Check for updates

ACM DIGITAL LIBRARY

Association for Computing Machinery

acm open

Latest updates: https://dl.acm.org/doi/10.1145/3729215

PDF Download
3729215.pdf
15 February 2026
Total Citations: 35
Total Downloads: 14088

Published: 13 June 2025
Online AM: 18 April 2025
Accepted: 22 March 2025
Revised: 04 March 2025
Received: 22 June 2023

Citation in BibTeX format

SURVEY

A Survey on Deep Learning Hardware Accelerators for Heterogeneous HPC Platforms

CRISTINA SILVANO, Politecnico di Milano, Milan, MI, Italy

DANIELE IELMINI, Politecnico di Milano, Milan, MI, Italy

FABRIZIO FERRANDI, Politecnico di Milano, Milan, MI, Italy

LEANDRO FIORIN, Politecnico di Milano, Milan, MI, Italy

SERENA CURZEL, Politecnico di Milano, Milan, MI, Italy

LUCA BENINI, University of Bologna, Bologna, BO, Italy

View all

Open Access Support provided by:

University of Turin

Istituto Nazionale di Fisica Nucleare, Sezione di Ferrara

Polytechnic of Turin

University of Bologna

Tor Vergata University of Rome

University of Calabria

View all

ACM Computing Surveys, Volume 57, Issue 11 (November 2025)
https://doi.org/10.1145/3729215
EISSN: 1557-7341

A Survey on Deep Learning Hardware Accelerators for Heterogeneous HPC Platforms

CRISTINA SILVANO, DEIB, Politecnico di Milano, Milano, Italy
DANIELE IELMINI, DEIB, Politecnico di Milano, Milano, Italy
FABRIZIO FERRANDI, DEIB, Politecnico di Milano, Milano, Italy
LEANDRO FIORIN, DEIB, Politecnico di Milano, Milano, Italy
SERENA CURZEL, DEIB, Politecnico di Milano, Milano, Italy
LUCA BENINI, Alma Mater Studiorum Università di Bologna, Bologna, Italy
FRANCESCO CONTI, Alma Mater Studiorum Università di Bologna, Bologna, Italy
ANGELO GAROFALO, Alma Mater Studiorum Università di Bologna, Bologna, Italy
CRISTIAN ZAMBELLI, Università degli Studi di Ferrara, Ferrara, Italy
ENRICO CALORE, INFN, Ferrara, Italy and Università degli Studi di Ferrara, Ferrara, Italy
SEBASTIANO SCHIFANO, Università degli Studi di Ferrara, Ferrara, Italy
MAURIZIO PALESI, Università degli Studi di Catania, Catania, Italy
GIUSEPPE ASCIA, Università degli Studi di Catania, Catania, Italy
DAVIDE PATTI, Università degli Studi di Catania, Catania, Italy
NICOLA PETRA, Università degli Studi di Napoli Federico II, Napoli, Italy
DAVIDE DE CARO, Università degli Studi di Napoli Federico II, Napoli, Italy
LUCIANO LAVAGNO, Politecnico di Torino, Torino, Italy
TEODORO URSO, Politecnico di Torino, Torino, Italy

Authors' Contact Information: Cristina Silvano, DEIB, Politecnico di Milano, Milano, Lombardia, Italy; e-mail: cristina.silvano@polimi.it; Daniele Ielmini, DEIB, Politecnico di Milano, Milano, Lombardia, Italy; e-mail: daniele.ielmini@polimi.it; Fabrizio Ferrandi, DEIB, Politecnico di Milano, Milano, Lombardia, Italy; e-mail: fabrizio.ferrandi@polimi.it; Leandro Fiorin, DEIB, Politecnico di Milano, Milano, Lombardia, Italy; e-mail: leandro.fiorin@polimi.it; Serena Curzel, DEIB, Politecnico di Milano, Milano, Lombardia, Italy; e-mail: serena.curzel@polimi.it; Luca Benini, Alma Mater Studiorum Universitì di Bologna, Bologna, Emilia-Romagna, Italy; e-mail: luca.benini@unibo.it; Francesco Conti, Alma Mater Studiorum Università di Bologna, Bologna, Emilia-Romagna, Italy; e-mail: f.conti@unibo.it; Angelo Garofalo, Alma Mater Studiorum Università di Bologna, Bologna, Emilia-Romagna, Italy; e-mail: angelo.garofalo@unibo.it; Cristian Zambelli, Università degli Studi di Ferrara, Ferrara, Emilia-Romagna, Italy; e-mail: cristian.zambelli@unife.it; Enrico Calore, INFN, Ferrara, Italy and Università degli Studi di Ferrara, Ferrara, Italy; e-mail: enrico.calore@fe.infn.it; Sebastiano Schifano, Università degli Studi di Ferrara, Ferrara, Italy; e-mail: sebastiano.fabio.schifano@unife.it; Maurizio Palesi, Università degli Studi di Catania, Catania, Sicilia, Italy; e-mail: maurizio.palesi@unict.it; Giuseppe Ascia, Università degli Studi di Catania, Catania, Sicilia, Italy; e-mail: giuseppe.ascia@unict.it; Davide Patti, Università degli Studi di Catania, Catania, Sicilia, Italy; e-mail: davide.patti@unict.it; Nicola Petra, Università degli Studi di Napoli Federico II, Napoli, Campania, Italy; e-mail: nicola.petra@unina.it; Davide De Caro, Università degli Studi di Napoli Federico II, Napoli, Campania, Italy; e-mail: dadecaro@unina.it; Luciano Lavagno, Politecnico di Torino, Torino, Piemonte, Italy; e-mail: luciano.lavagno@polito.it; Teodoro Urso, Politecnico di Torino, Torino, Piemonte, Italy; e-mail: teodoro.urso@polito.it; Valeria Cardellini, University of Rome Tor Vergata, Roma, Italy; e-mail: cardellini@ing.uniroma2.it; Gian Carlo Cardarilli, University of Rome Tor Vergata, Roma, Italy; e-mail: g.cardarilli@uniroma2.it; Robert Birke, Università degli Studi di Torino, Torino, Piemonte, Italy; e-mail: robert.birke@unito.it; Stefania Perri, Università degli Studi della Calabria, Arcavacata di Rende, Italy; e-mail: stefania.perri@unical.it.

This work is licensed under a Creative Commons Attribution 4.0 International License.

© 2025 Copyright held by the owner/author(s).
ACM 0360-0300/2025/06-ART286
https://doi.org/10.1145/3729215

VALERIA CARDELLINI, University of Rome Tor Vergata, Roma, Italy
GIAN CARLO CARDARILLI, University of Rome Tor Vergata, Roma, Italy
ROBERT BIRKE, Università degli Studi di Torino, Torino, Italy
STEFANIA PERRI, Università degli Studi della Calabria, Arcavacata di Rende, Italy

Recent trends in deep learning (DL) have made hardware accelerators essential for various high-performance computing (HPC) applications, including image classification, computer vision, and speech recognition. This survey summarizes and classifies the most recent developments in DL accelerators, focusing on their role in meeting the performance demands of HPC applications. We explore cutting-edge approaches to DL acceleration, covering not only GPU- and TPU-based platforms but also specialized hardware such as FPGA- and ASIC-based accelerators, Neural Processing Units, open hardware RISC-V-based accelerators, and co-processors. This survey also describes accelerators leveraging emerging memory technologies and computing paradigms, including 3D-stacked Processor-In-Memory, non-volatile memories like Resistive RAM and Phase Change Memories used for in-memory computing, as well as Neuromorphic Processing Units, and Multi-Chip Module-based accelerators. Furthermore, we provide insights into emerging quantum-based accelerators and photonics. Finally, this survey categorizes the most influential architectures and technologies from recent years, offering readers a comprehensive perspective on the rapidly evolving field of deep learning acceleration.

CCS Concepts: • **Computer systems organization → Architectures**; • **Hardware → Reconfigurable logic and FPGAs**; **Emerging technologies**; **Very large scale integration design**; *Power and energy*; • **Computing methodologies → Machine learning;**

Additional Key Words and Phrases: Hardware accelerators, high-performance computing, deep learning, deep neural networks, emerging memory technologies

ACM Reference Format:
Cristina Silvano, Daniele Ielmini, Fabrizio Ferrandi, Leandro Fiorin, Serena Curzel, Luca Benini, Francesco Conti, Angelo Garofalo, Cristian Zambelli, Enrico Calore, Sebastiano Schifano, Maurizio Palesi, Giuseppe Ascia, Davide Patti, Nicola Petra, Davide De Caro, Luciano Lavagno, Teodoro Urso, Valeria Cardellini, Gian Carlo Cardarilli, Robert Birke, and Stefania Perri. 2025. A Survey on Deep Learning Hardware Accelerators for Heterogeneous HPC Platforms. *ACM Comput. Surv.* 57, 11, Article 286 (June 2025), 39 pages. https://doi.org/10.1145/3729215

1 Introduction

With the advent of the Exascale era, we have witnessed a growing convergence between **High-Performance Computing** (**HPC**) and **Artificial Intelligence** (**AI**). The increasing computing power of HPC systems, combined with their ability to manage vast amounts of data, has driven the development of more and more sophisticated **machine learning** (**ML**) techniques. **Deep Learning** (**DL**), a subset of ML, utilizes **Deep Neural Networks** (**DNNs**) with multiple layers of artificial neurons to mimic the human brain behavior by learning from large datasets. Thanks to advancements in technology and system architecture, HPC nodes now integrate not only an increasing number of high-end parallel processors but also specialized co-processors such as **Graphics Processing Units (GPUs)** and vector/tensor computing units. This supercomputing power has significantly accelerated both the training and inference phases of DNN models used in several application scenarios. The introduction of the pioneering AlexNet [109] model at the ImageNet challenge in 2012 marked a turning point, demonstrating the power of GPU acceleration in deep learning. Since then, numerous DNN models have been developed for various tasks including image recognition and classification, Natural Language Processing (NLP), and Generative AI. These applications demand specialized *hardware accelerators*, to efficiently handle the heavy

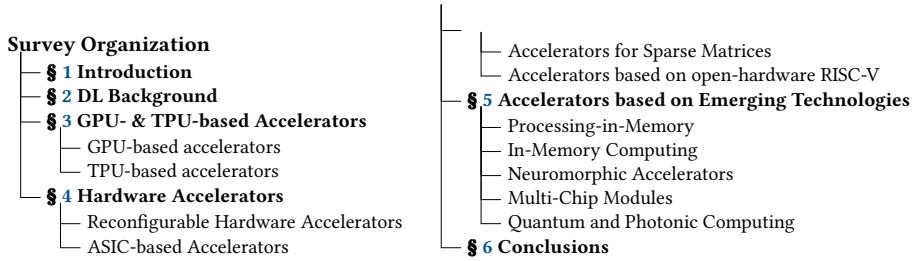


Fig. 1. Organization of the survey.

computational workload of DNN algorithms. Today, DL accelerators are deployed across a wide range of computing systems spanning from ultra-low-power resource-constrained devices to high-performance servers, HPC infrastructures, and large-scale data centers.

Scope of the survey. This survey is an attempt to provide an extensive overview of the most influential architectures to accelerate DL for high-performance applications. The survey highlights various approaches that support DL acceleration including GPU-based accelerators, Tensor Processor Units, FPGA-based accelerators, and ASIC-based accelerators, such as Neural Processing Units and specialized co-processors based on the open-hardware RISC-V architecture. The survey also includes accelerators based on emerging technologies and computing paradigms, such as 3D-stacked PIM, emerging non-volatile memories such as the **Resistive Random Access Memory** (**RRAM**) and the **Phase Change Memory** (**PCM**), Neuromorphic Processing Units, and Multi-Chip Modules.

Overall, we have reviewed the research on DL accelerators from the past two decades, covering a significant period of literature in this field. Being DL acceleration a prolific and rapidly evolving field, we do not claim to cover exhaustively all the research works appeared so far, but we focused on the most influential contributions. Moreover, this survey can be leveraged as a connecting point for some previous surveys on AI and DL accelerators [28, 61, 83, 168] and other surveys focused on some more specific aspects of DL, including the architecture-oriented optimization of sparse matrices [170] and the Neural Architecture Search [32]. Another research trend in state-of-the-art AI architecture design addresses transformer models. A recent survey on the full stack of optimizations on transformer inference has recently been published in [105].

Organization of the survey. The survey is structured in different sections and sub-sections belonging to the areas of computer architecture and hardware design, as shown in Figure 1, To this aim, we organized the material in a way that all research papers corresponding to multiple types of sections are cited under each section. Moreover, for each section, we have selectively chosen the most notable and influential works and, for each work, we focused on its most innovative contributions.

To conclude, we hope this survey could be useful for a wide range of readers, including computer architects, hardware developers, HPC engineers, researchers, and technical professionals. A major effort was spent to use a clear and concise technical writing style: we hope this effort could be useful in particular to the young generations of master's and Ph.D. students. To facilitate the reading, a list of acronyms is reported in Table 1.

2 Deep Learning Background

Deep Learning [114, 175] is a subset of ML methods that can automatically discover the representations needed for feature detection or classification from large data sets, by employing multiple layers

Table 1. List of Acronyms

Acronym	Acronym	Acronym
AI: Artificial Intelligence	**ASIC:** Application Specific Integrated Circuit	**BRAM:** Block Random Access Memory
CMOS: Complementary Metal Oxide Semiconductor	**CNN:** Convolutional Neural Network	**CPU:** Central Processing Unit
DL: Deep Learning	**DP:** Double Precision	**DNN:** Deep Neural Network
DRAM: Dynamic Random Access Memory	**EDA:** Electronic Design Automation	**FLOPS:** Floating Point Operations per Second
FMA: Fused Multiply-Add	**FPGA:** Field-Programmable Gate Array	**GEMM:** General Matrix Multiply
GP-GPU: General-Purpose Graphics Processing Unit	**GPU:** Graphics Processing Unit	**HBM:** High Bandwidth Memory
HDL: Hardware Description Language	**HLS:** High Level Synthesis	**HMC:** Hybrid Memory Cube
HPC: High-Performance Computing	**MLP:** Multi-Layer Perceptron	**NPU:** Neural Processing Unit
IMC: In-Memory Computing	**IoT:** Internet of Things	**ISA:** Instruction Set Architecture
MCM: Multi-Chip Module	**ML:** Machine Learning	**PCM:** Phase Change Memory
PIM: Processing In-Memory	**QC:** Quantum Computing	**QNN:** Quantized Neural Network
QPU: Quantum Processing Unit	**RAM:** Random Access Memory	**RRAM:** Resistive RAM
RISC: Reduced Instruction Set Computer	**RNN:** Recurrent Neural Network	**SNN:** Spiking Neural Network
SoC: System on Chip	**SRAM:** Static Random Access Memory	**TPU:** Tensor Processing Unit

of processing to extract progressively higher-level features. The most recent works in literature clearly show that two main DL topologies have emerged as dominant: DNNs and Transformers.

Concerning DNNs, there are three types of DNNs mostly used today: Multi-Layer Perceptrons (MLPs), Convolutional Neural Networks (CNNs), and Recurrent Neural Networks (RNNs). MLPs [171] are feed-forward ANNs composed of a series of fully connected layers, where each layer is a set of nonlinear functions of a weighted sum of all outputs of the previous one. On the contrary, in a CNN [115], a convolutional layer extracts the simple features from the inputs by executing convolution operations. Each layer is a set of nonlinear functions of weighted sums of different subsets of outputs from the previous layer, with each subset sharing the same weights. Each convolutional layer in the model can capture a different high-level representation of input data, allowing the system to automatically extract the features of the inputs to complete a specific task, e.g., image classification, face authentication, and image semantic segmentation. Finally, RNNs [175] address the time-series problem of sequential input data. Each RNN layer is a collection of nonlinear functions of weighted sums of the outputs of the previous layer and the previous state, calculated when processing the previous samples and stored in the RNN's internal memory. RNN models are widely used in NLP for natural language modeling, word embedding, and machine translation. More details on concepts and terminology related to DNNs are provided in Section A.1 of Appendix A.

Each type of DNN is especially effective for a specific subset of cognitive applications. Depending on the target application, and the resource constraints of the computing system, different DNN models have been deployed. Besides DNNs, Transformer-based models [204] recently captured a great deal of attention. Transformers were originally proposed for NLP [204], and are designed to recognize long-distance dependencies between data by *attention layers*, where the weights used to linearly transform input data are computed dynamically based on the input data itself. While DNNs use convolutional layers to perform "local" operations on small portions of the input, Transformers use attention layers to perform "global" operations on the whole input. Although quite different, DNNs and Transformers share many underlying principles (such as gradient descent training, and reliance on linear algebra), and many of the DL-dedicated architectures described in this survey address both types of topologies.

3 GPU- and TPU-Based Accelerators

3.1 GPU-Based Accelerators

GPUs are specific-purpose processors introduced to compute efficiently graphics-related tasks, such as 3D rendering. They became widely used since the nineties as co-processors, working alongside

Table 2. MLPerf Training v2.1 Benchmark Results (Minutes)

	ImageNet ResNet	KiTS19 3D U-Net	OpenImages RetinaNet	COCO Mask R-CNN	LibriSpeech RNN-T	Wikipedia BERT	Go Minigo
8 × A100	30.8	25.6	89.1	43.1	32.5	24.2	161.6
8 × H100	14.7	13.1	38.0	20.3	18.2	6.4	174.6

Central Processing Units (CPUs) to offload graphics-related computations. The introduction of programmable shaders into GPU architectures increased their flexibility paving the way for their adoption to perform general-purpose computations. Despite being specifically designed for computer graphics, their highly parallel architecture is well suited to tackle a wide range of applications. Consequently, in the early 2000s, GPUs started to be used to accelerate data-parallel computations not necessarily related to graphics. This practice is commonly referred to as General-Purpose computing on Graphics Processing Units GPUs (GP-GPU) and started to be increasingly popular in the early 2010s with the advent of the CUDA language. The technological development of the last ten years significantly increased the computing power of GPUs, which, due to their highly parallel nature, are incidentally very well suited to accelerate neural network training algorithms. The availability of such computing power allowed more complex neural network models to become practically usable, fostering the development of DNNs.

The impressive results obtainable with DNNs, followed by significant investments in this market sector, induced hardware manufacturers to modify GPU architectures in order to be even more optimized to compute such workloads, as an example implementing the support for lower-precision computations. This led to a de-facto co-design of GPU architectures and neural network algorithms implementations, which is nowadays significantly boosting the performance, accuracy, and energy efficiency of AI applications. The basic features of GPU architectures able to boost the performance of HPC and DL applications are briefly reviewed in Section A.2 of the Appendix A.

GPUs can execute multiple, simultaneous computations. This enables the distribution of training processes and can significantly speed up ML operations. With GPUs, it is possible to cumulate many cores that use fewer resources without sacrificing neither efficiency nor power.

The performance of GPU accelerators could be compared in different ways. As a first approximation, their theoretical peak performance and memory bandwidth could be used. Anyhow several other architectural characteristics could affect the final performance of actual algorithm implementation. To get a better overview of their expected performance, running a specific workload, it could be preferable to use reference benchmarks, possibly made of representative sets of commonly used algorithm implementations. For this reason, different benchmarks have been developed, each of them able to test the obtainable performance concerning a given workload characteristic, or a given set of application kernels. In the context of ML, one of the most used benchmarks is MLPerf [135], which has a specific set of training phase tasks [134]. Its results on two different systems, embedding the latest GPU architecture and its predecessor (i.e., NVIDIA Hopper and Ampere) are shown in Table 2, highlighting on average an approximate 2× factor of performance improvement. Different vendors, like AMD and Intel, have also developed GP-GPU architectures mostly oriented to HPC and more recently to AI computing. Yet the terminology used by different vendors is not the same, they share most of the hardware details. For example, AMD names Compute Unit which NVIDIA calls Streaming Multiprocessor, and Intel calls Compute Slice or **Execution-Unite (EU)**. Furthermore, NVIDIA names Warp the set of instructions scheduled and executed at each cycle, while AMD uses the term Wavefront, and Intel uses the term EU-Thread. Concerning the execution model, NVIDIA uses the **Single Instruction Multiple Thread (SIMT)**, while AMD and Intel use the **Single Instruction Multiple Data (SIMD)** [102]. In Table 3, we report the

Table 3. Selected Features of the Most Recent GP-GPU Systems

Model	NVIDIA H100	AMD Instinct MI250X	Intel Arc 770
Clock [GHz]	1.6	1.7	2.4
Peak Performance in Double Precision [TFLOPS]	30	47.9	4.9
Peak Performance in Single Precision [TFLOPS]	60	95.8	19.7
Peak Performance in FP16 [TFLOPS]	120	383	39.3
Max Memory [GB]	80 HBM2e	128GB HBM2e	16GB GDDR6
Mem BW [TB/s]	2.0	3.2	0.56
Thermal Design Power (TDP) [Watt]	350	560	225

main hardware features of the three most recent GP-GPU architectures developed by NVIDIA [8], AMD [7] and Intel [89]. We compare the peak performance related to the 32-bit single- and 64-bit double-precision, and the peak performance achieved using half-precision. The comparison evidences that the higher the memory bandwidth provided to sustain DL workloads like model training, the higher the power consumption. Further, a huge number of parallel resources intended as physical cores are mandatory to achieve high computing performance to reduce model training time, however at the expense of reduced energy efficiency.

3.2 TPU-based Accelerators

Tensor Processing Units (TPUs) dedicated to training and inference have been proposed very early after the emergence of the first large CNN-based applications. This is due to the observation that these workloads are dominated by linear algebra kernels that can be refactored as matrix multiplications (particularly if performed in batches) and that their acceleration is particularly desirable for high-margin applications in data centers. More recently, the emergence of exponentially larger models with each passing year (e.g., the GPT-2, GPT-3, GPT-4 Transformer-based large language models) required a continuous investment in higher-performance training architectures.

Google showcased the first TPU [96, 98] at ISCA in 2017, but according to the original paper the first deployment occurred in 2015 – just three years after the "AlexNet revolution". Their last TPU v4 implementation outperforms the previous TPU v3 by 2.1x and improves performance/Watt by 2.7x [97]. The architecture of the TPU presented in [96, 98] is centered on a large (256×256) systolic array operating on 8-bit integers and targeting exclusively data center inference applications; this is coupled with a large amount of on-chip SRAM for activations (24 MiB) and a high-bandwidth (30 GiB/s) dedicated path to off-chip L3 DRAM for weights. The next design iterations (TPUv2, TPUv3) [99] forced to move from an inference-oriented design to a more general engine tuned for both inference and training, employing the 16-bit BF16 floating-point format, more cores (2 per chip) using each one or two 4× smaller arrays than TPUv1 (128×128, to reduce under-usage inefficiencies). TPUv2/v3 also introduced high-bandwidth memory support, which results in more than 20× increase in the available off-chip memory bandwidth. In 2019, Habana Labs and Intel proposed Goya and Gaudi as microarchitectures for the acceleration of inference [136]. Goya [136] relies on PCIe 4.0 to interface to a host processor and exploits a heterogenous design approach comprising of a large **General Matrix Multiply** (**GEMM**) engine, TPUs, and a large shared DDR4 memory pool. Each TPU also incorporates its local memory that can be either hardware-managed or fully software-managed, allowing the compiler to optimize the residency of data and reduce movement. Each of the individual TPUs is a VLIW design optimized for AI applications and especially for training. The TPU supports mixed-precision operations including 8-bit, 16-bit, and 32-bit SIMD vector operations for both integer and floating-point. Gaudi has an enhanced version of the TPUs and uses HBM global memories rather than the DDR used in Goya, increasing the support toward bfloat16 data types and including more operations and functionalities dedicated to

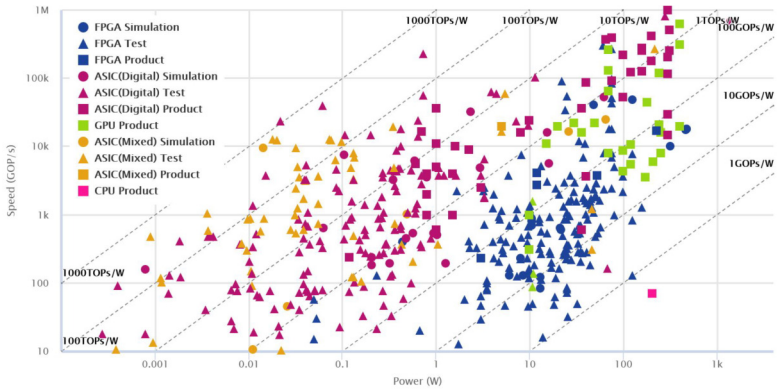


Fig. 2. Overview on state-of-the-art Neural Network accelerators based on available data collected in [76]. Legenda: *Simulation* means GOPS/W values collected from post-layout simulation; *Test* means from prototype devices; *Product* means from off-the-shelf devices.

training operations. While Google and Intel rely on a mixture of in-house designs and GPUs, the other main data center providers typically rely on NVIDIA GPUs to serve DL workloads. Starting from the Volta architecture [34] and continuing with Ampere [35] and Hopper [33, 52], NVIDIA has embedded inside the GPU Streaming Multiprocessors the counterpart of smaller TPUs, i.e., *TensorCores*.

GraphCore Colossus Mk1 and Mk2 IPUs [92, 108] target the GNNs, DNNs, and Transformers training employing a tiled many-core architecture. GraphCore focuses on a high power- and cost-efficient memory hierarchy that does not rely on high-bandwidth off-chip HBM, but on cheaper DRAM chips combined with a large amount of on-chip SRAM (in the order of 1 GiB per chip).

IBM Research focused on reducing the data precision used for training [1, 206], by introducing Hybrid-FP8 formats in training ASICs and tensor processors. Further improvements were achieved with Cambricon-Q [225], which exploits the statistical properties of tensors to minimize bandwidth and maximize efficiency. Finally, Gemmini [66, 72] and RedMulE [198, 199] introduce tensor processor hardware IPs (respectively, generated from a template and hand-tuned) that can be integrated inside System-on-Chips, similarly to what NVIDIA does with TensorCores. Further details on TPU architectures are provided in Section A.2 of the Appendix A.

4 Hardware Accelerators

Typical HPC workloads, like genomics, astrophysics, finance, and cyber security, require the elaboration of massive amounts of data and they can take advantage of DL methods with results that can surpass human ability [11, 73, 175, 189]. However, an ever-increasing computing power, a rapid change of the data analysis approaches, and the introduction of novel computational paradigms are needed. DL models can be efficiently supported by optimized hardware platforms providing high levels of parallelism and a considerable amount of memory resources. These platforms can be developed using CPUs, GPUs, FPGAs, and ASICs [47, 51, 73, 127, 131, 193, 213].

Figure 2 presents a comparison of state-of-the-art architectures in terms of speed *(Giga Operations per Second)* versus power consumption *(Watt)*. The diagonal dashed lines represent energy efficiency levels in *(GOPS/W)* and *TOPS/W* (Tera Operations per Second per Watt), with higher slopes indicating better energy efficiency. The most energy-efficient devices are ASICs and GPUs clustering in the high range of energy efficiency (1–100 TOPS/W) and mainly located in the top right region characterized by the highest computational throughput. Powerful GPUs are generally

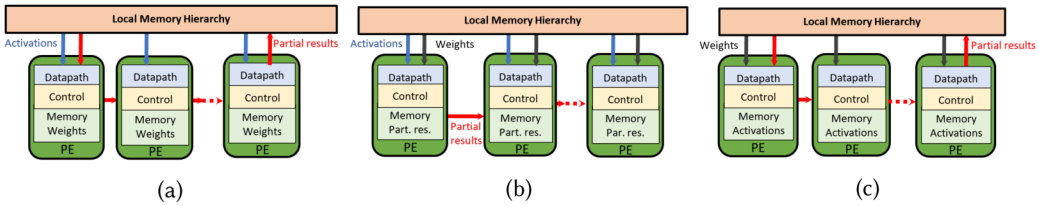


Fig. 3. Dataflows in DL accelerators: (a) Weights stationary; (b) Output stationary; (c) Input stationary.

preferred for heavier tasks like training and running large, complex models built on large datasets. Conversely, FPGAs are well suited to accelerate specific inference tasks that privilege lower power consumption over processing speeds. From the figure, FPGAs (represented in blue) are well distributed across different power and performance levels, mostly clustering in the lower-to-mid range of energy efficiency (from 10 GOPS/W to 1 TOPS/W).

Independently of the technology, a common problem in the design of accelerators is the energy and latency cost of accessing the off-chip DRAM memory, in particular considering the significant amount of data that HPC applications need to process. As sketched in Figure 3, different data reuse and stationary strategies can be exploited to reduce the number of accesses [22, 77, 153, 156, 174, 192]. In weight stationary dataflow, convolutional weights are fixed and stored in the local memory of the **Processing Elements** (**PEs**) and reused on input activations uploaded step-by-step from the external DRAM. Conversely, in output stationary dataflow, partial outputs from PEs are stored locally and reused step-by-step until the computation is completed. Then, just the final results are moved to the external DRAM. An efficient alternative is input stationary dataflow: input activations are stored in the PEs' local memory, while weights are uploaded from the external DRAM and sent to the PEs. Another approach common to many accelerators is the introduction of *quantization* to reduce the data type width [67, 129]. Integer or fixed-point data formats are generally preferred to the more computationally intensive floating-point ones. This guarantees better memory occupation and lower computational cost [95]. Extreme quantization techniques that use only one bit for the data stored (Binary Neural Networks [164]) are widely used for very large networks but to get a comparable accuracy they require 2-11× the number of parameters and operations [200], making them not suitable for complex problems.

4.1 Reconfigurable Hardware Accelerators

FPGAs and Coarse-Grained Reconfigurable Arrays (CGRAs) are highly sought-after solutions to hardware accelerate a wide range of applications. The main feature of such platforms is the ability to support different computational requirements by repurposing the underlying hardware accelerators at deploy-time and also at runtime. More details on FPGA technologies and related EDA frameworks are respectively provided in Sections A.3 and A.4 of Appendix A. Several FPGA-based hardware accelerators for DL are structured as heterogeneous embedded systems [2, 120, 130, 159, 219] that mainly consist of a general-purpose processor, for running the software workload; a computational module, designed to speed up common DL operators, like convolutions [166, 205], de-convolutions [23, 179], pooling, fully connected operations, activation, and softmax functions [190, 191]; and a memory hierarchy to optimize data movement to/from the external DRAM to store data to be processed and computational results. A typical approach to accelerate convolutions consists of a systolic array architecture (SA), a regular pattern that can be easily replicated [216]. Each PE in the array is a SIMD vector accumulation module where inputs and weights are supplied at each cycle by shifting them from the horizontally and vertically adjacent PEs. The use of pipelined groups of PEs with short local communication and regular architecture enables a high clock frequency and limited global data transfer.

Table 4. Summary of ASIC-Based AI-Accelerators

Design	Process [nm]	Area [mm^2]	Peak Perf. [TOPS]	Energy Eff. [TOPS/W]	Maturity-level
Samsung [187]	8	5.5	933	6.9	Silicon
UM+NVIDIA [223]	16	2.4	480	-	Silicon
MediaTek [124]	7	3.04	880	3.6	Silicon
Alibaba [94]	12	709	700	825	Silicon
Samsung [151]	5	5.46	1196	29.4	Silicon
Samsung [152]	4	4.74	1197	39.3	Silicon
DaDianNao [27]	28	67.7	5.59	-	Layout
ShiDianNao [27]	65	4.86	0.19	-	Layout
Cambricon [224]	65	6.38	1.1	-	Layout
EIE [82]	28	63.8	0.002	0.18	Simulation
Eyeriss [29]	65	16	0.03	0.07	Layout
STM [46]	28	34.8	0.75	2.9	Silicon
IBM [147]	14	9.84	3	1.1	Silicon
IBM [116]	7	19.6	16.3	3.58	Silicon

Although FPGAs have traditionally been proposed as accelerators for edge applications, they are starting to be adopted also in data centers. Microsoft's Project Brainwave [55] uses several FPGA boards to accelerate the execution of RNNs in the cloud, exploiting the reconfigurability to adapt the platform to different DL models. One way to face the limitations imposed by the capability of FPGAs to effectively map very large DL models is to use a deeply pipelined multi-FPGA design. Recent studies focus on optimizing this type of architecture and maximizing the overall throughput [167, 181, 222]. In these application contexts, CGRAs represent an alternative to FPGAs, providing reconfigurability with coarser-grained functional units. They are based on an array of PEs, performing the basic arithmetic, logic, and memory operations at the word level and using a small register file as temporary data storage. Neighboring PEs are connected through reconfigurable routing that allows transferring intermediate results of the computations towards the proper neighbors for the next computational step. CGRAs can represent a vaible solution to accelerate dense linear algebra applications, such as ML, image processing, and computer vision [18, 63].

4.2 ASIC-Based Accelerators

To comply with the computational capabilities required by AI workloads, new powerful processing architectures are upcoming. Among them, there are two different types of Neural Processing Units: single-chip NPUs and NPUs integrated in the general purpose CPU. One of the main trends toward the next generation of laptops follows the second option by pushing the performance of AI workloads by integrating into the general-purpose CPU not only a GPU to accelerate graphics but also an NPU. This is the case of the recent Lunar Lake Intel processor architecture [88]. Table 4 is an attempt to offer a common ground of different types of AI-accelerators in terms of process technology node, area, peak performance, energy efficiency and maturity level.

The purpose of an integrated NPU is to accelerate the performance and improve the energy efficiency of specific AI-tasks offloaded from the CPU [187]. In particular, NPUs are designed to accommodate a reasonable amount of multiply/accumulate (MAC) units, which are the PEs devised in the convolutional and fully-connected layers of DNNs [29, 46].

Each PE contains a synaptic weight buffer and MAC units to perform the computation of a neuron, namely, multiplication, accumulation, and an activation function (e.g., sigmoid). A PE can

be realized with full-CMOS optimized circuits to trade off speed and power consumption. One of the most popular approaches adopted to this aim is referred to as *approximate computing paradigm* to approximate the design at the cost of an acceptable accuracy loss. Representative approximate computing techniques suitable to design efficient arithmetic data paths are overviewed in Section A.5 of the Appendix A.

An alternative method to design PEs consists in using emerging non-volatile memories such as RRAM and PCM to perform *in situ* matrix-vector multiplication as in the RENO chip [128] or as in the MAC units proposed in References [145, 218]. The advantage of these architectures is that only the input and final output are digital; the intermediate results are all analog and are coordinated by analog routers. Data converters (DACs and ADCs) are required only when transferring data between the NPU and the CPU with an advantage in terms of energy efficiency (the work in [218] reports an energy efficiency of 53.17 TOPS/W), although there are insufficient experimental data to support this evidence in comparison with full-digital NPUs.

In the DNN landscape, single-chip domain-specific accelerators achieved great success in both cloud and edge scenarios. These custom architectures offer better performance and energy efficiency concerning CPUs/GPUs thanks to an optimized data flow (or data reuse pattern) that reduces off-chip memory accesses, while improving the system efficiency [28]. The DianNao series represents a full digital stand-alone DNN accelerator that introduces a customized design to minimize the memory transfer latency and enhance the system efficiency. DaDianNao [27] targets the datacenter scenario and integrates a large on-chip **embedded dynamic random access memory** (**eDRAM**) to avoid the long main memory access time. The same principle applies to the embedded scenario. ShiDianNao [27] is a DNN accelerator dedicated to CNN applications. Using a weight-sharing strategy, its footprint is much smaller than the previous design. It is possible to map all of the CNN parameters onto a small on-chip static random access memory (SRAM) when the CNN model is small. In this way, ShiDianNao avoids expensive off-chip DRAM access time and achieves 60 times more energy efficiency compared to DianNao. Furthermore, domain-specific instruction set architectures (ISAs) have been proposed to support a wide range of NN applications. Cambricon [224] and EIE [82] are examples of architectures that integrate scalar, vector, matrix, logical, data transfer, and control instructions. Their ISA considers data parallelism and the use of customized vector/matrix instructions.

Eyeriss is another notable accelerator [29] that can support high throughput inference and optimize system-level energy efficiency, also including off-chip DRAMs. The main features of Eyeriss are a spatial architecture based on an array of 168 PEs that creates a four-level memory hierarchy, a dataflow that reconfigures the spatial architecture to map the computation of a given CNN and optimize towards the best energy efficiency, a network-on-chip (NoC) architecture that uses both multi-cast and point-to-point single-cycle data delivery, and **run-length compression** (**RLC**) and PE data gating that exploit the statistics of zero data in CNNs to further improve EE.

In Reference [46], STMicroelectronics introduced the Orlando system-on-chip, a 28nm FDSOI-based CNN accelerator integrating an SRAM-based architecture with low-power features and adaptive circuitry to support a wide voltage range. Such a DNN processor provides an energy-efficient set of convolutional accelerators supporting kernel compression, an on-chip reconfigurable data-transfer fabric, a power-efficient array of DSPs to support complete real-world computer vision applications, an ARM-based host subsystem with peripherals, a range of high-speed I/O interfaces, and a chip-to-chip multilink to pair multiple accelerators together.

IBM presented in Reference [147] a processor core for AI training and inference tasks applicable to a broad range of neural networks (such as CNN, LSTM, and RNN). High compute efficiency is achieved for robust FP16 training via efficient heterogeneous 2-D systolic array-SIMD compute engines that leverage DLFloat16 FPUs. A modular dual-corelet architecture with a shared scratchpad

memory and a software-controlled network/memory interface enables scalability to many-core SoCs for scale-out paradigms. In 2022, IBM also presented a 7-nm four-core mixed-precision AI chip [116] that demonstrates leading-edge power efficiency for low-precision training and inference without model accuracy degradation. The chip is based on a high-bandwidth ring interconnect to enable efficient data transfers, while workload-aware power management with clock frequency throttling maximizes the application performance within a given power envelope.

Qualcomm presented an AI core that is a scalar 4-way VLIW architecture that includes vector/tensor units and lower precision to enable high-performance inference [24]. The design uses a 7 nm technology and is sought to be integrated into the AI 100 SoC to reach up to 149 TOPS with a power efficiency of 12.37 TOPS/W.

4.3 Accelerators for Sparse Matrices

Network pruning and zero-valued activations introduce sparsity that can be exploited by hardware accelerators to achieve compute and data reduction. This section overviews accelerator architectures designed to manage sparse matrices. Definitions, storage formats appropriate for sparse matrices, and their impacts on the computational complexity of DL models are discussed in Appendix A.6.

Eyeriss [29] targets CNN acceleration by storing in DRAM only nonzero-valued activations in **Compressed Sparse Columns (CSC)** format and by skipping zero-valued activations to save energy. Eyeriss v2 [30], which targets DNNs on mobile devices, also supports sparse network models. It utilizes the CSC format to store weights and activations, which are kept compressed not only in memory but also during processing. To improve flexibility, it uses a hierarchical mesh for the PEs interconnections. By means of these optimizations, Eyeriss v2 is significantly faster and more energy-efficient than the original Eyeriss.

Cnvlutin [4] uses hierarchical data-parallel units, skips computation cycles for zero-valued activations and employs a co-designed data storage format based on Compressed Sparse Rows (CSR) to compress the activations in DRAM. However, it does not consider the sparsity of the weights. On the contrary, Cambricon-X architecture [224] enables the PEs to store the compressed weights in CSR format for asynchronous computation. However, it does not exploit activation sparsity. EIE [82], besides compressing the weights through a variant of CSC sparse matrix representation and skipping zero-valued activations, employs a scalable array of PEs, each storing a partition of the DNN in SRAM that allows obtaining significant energy savings with respect to DRAM. NullHop [2] applies the Compressed Image Size (CIS) format to the weights and skips the null activations, similarly to EIE. Sparse CNN (SCNN) [150] is an accelerator architecture for inference in CNNs. It employs a cluster of asynchronous PEs comprising several multipliers and accumulators. SCNN exploits sparsity in both weights and activations, which are stored in the classic CSR representation. It employs a Cartesian product-based computation architecture that maximizes the reuse of weights and activations within the cluster of PEs; the values are delivered to an array of multipliers, and the resulting scattered products are summed using a dedicated interconnection mesh. By exploiting two-sided sparsity, SCNN improves performance and energy over dense architectures. SparTen [71] is based on SCNN [150]. It addresses some considerable overheads of SCNN in performing the sparse vector-vector dot product by improving the distribution of the operations to the multipliers and allows using any convolutional stride. It also addresses unbalanced sparsity distribution across the PEs employing an offline software scheme. The PermDNN architecture [43] uses permuted diagonal matrices to not generate load imbalance which is caused by the irregularity of unstructured sparse DNN models.

SqueezeFlow [121] exploits concise convolution rules to benefit from the reduction of computation and memory accesses as well as the acceleration of existing dense CNN architectures without intrusive PE modifications. The **Run Length Compression** (**RLC**) format is used to compress

activations and weights. A different strategy is pursued by the **Unique Weight CNN** (**UCNN**) accelerator [85], which proposes a generalization of the sparsity problem. Rather than considering only the repetition of zero-valued weights, UCNN exploits repeated weights with any value by reusing CNN sub-computations and reducing the model size in memory. SIGMA [163] is characterized by a flexible and scalable architecture that offers high utilization of its PEs regardless of kernel shape (i.e., matrices of arbitrary dimensions) and sparsity pattern. It targets the acceleration of GEMMs with unstructured sparsity. Bit-Tactical [42] uses a static scheduling middleware and a co-designed hardware front-end, with a lightweight sparse shuffling network that comprises two multiplexers per activation input. Unlike SIGMA and other accelerators, Bit-tactical leverages scheduling in software to align inputs and weights. Flexagon [142] is a reconfigurable accelerator capable of performing sparse-sparse matrix multiplication computation by using the particular data flow that best matches each case.

Besides the design of specialized hardware accelerators to exploit model sparsity, a parallel trend is to use GPU architectures. Pruned sparse models with unstructured sparse patterns introduce irregular memory accesses that are unfriendly on commodity GPU architectures. The first direction to tackle this issue is at the software layer, using pruning algorithms that enforce a particular sparsity pattern, such as tile sparsity [75], on the model that allows leveraging existing GEMM accelerators. A second direction is to introduce new architectural support, such as Sparse Tensor Cores [140]. The NVIDIA Ampere architecture introduces this design with a fixed 50% weight pruning target and achieves a better accuracy and performance tradeoff. However, sparsity from activations, which are dynamic and unpredictable, is challenging to leverage on GPUs. Indeed, the current Sparse Tensor Core can take advantage of weight sparsity. Reconfigurability appears to be a keyword for the design of new sparse accelerators because some network models exhibit *dynamic sparsity* [53], where the position of non-zero elements changes over time.

4.4 Accelerators Based on Open-Hardware RISC-V

RISC-V is an open-source, modular instruction set architecture (ISA) that is gaining popularity in computer architecture research due to its flexibility and suitability for integration with acceleration capabilities for DL. The RISC-V ISA is designed with a small, simple core that can be extended with optional instruction set extensions (ISEs) to support various application domains. RISC-V offers several advantages for DL acceleration research. First, the modular nature of the ISA allows researchers to easily integrate acceleration capabilities as ISEs, which can be customized to suit the specific needs of different DL models. Second, RISC-V supports a set of standard interfaces, such as AXI4, that can be used to interface with external acceleration units integrated on the same System-on-Chip at various levels of coupling. This makes it easy to integrate specialized DL hardware accelerators into RISC-V-based systems. Moreover, the defining feature of the RISC-V ISA is its openness, meaning that anybody can design a RISC-V implementation without paying royalties or needing a particular license. Thanks to this non-technical advantage against other ISAs (such. as ARM, x86), RISC-V has gained significant attention from academia and emerging startups. Figure 4 reports a synthetic taxonomy of representative RISC-V-based accelerators for DL.

4.4.1 RISC-V ISA Extensions for (Deep) Learning. Works in [36, 132] propose ISA extensions for *posit* numbers which can be used to do weight compression. Posit numbers need fewer bits to obtain the same precision or dynamic range of IEEE floats allowing them to store more weights in a same-sized memory. For example, the work in [36] provides an efficient conversion between 8- or 16-bit posits and 32-bit IEEE floats or fixed point formats with little loss in precision leading to a 10x speedup in inference time. Other works directly address the compute-intensive parts of different neural networks, in particular CNNs, GCNs, and transformers. The new Winograd-based

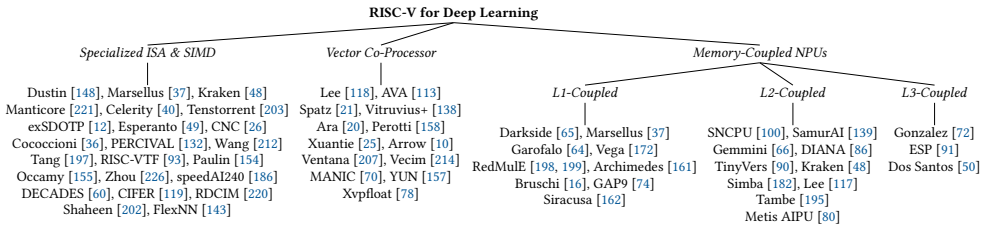
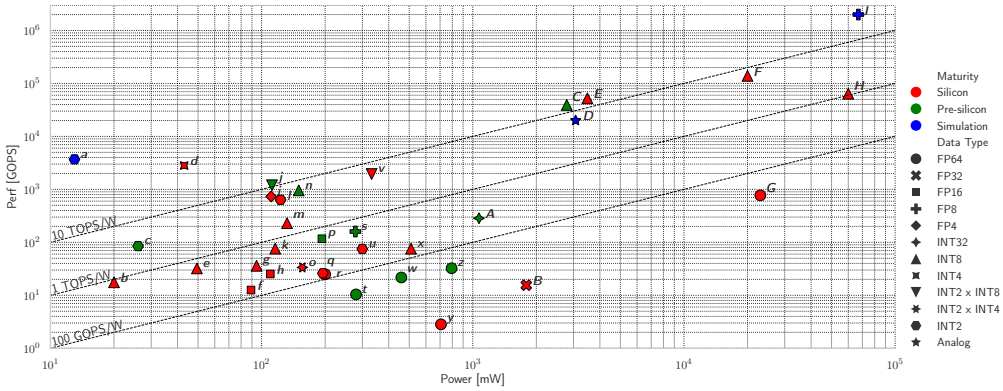


Fig. 4. Taxonomy of RISC-V based acceleration units discussed in Section 4.4.

convolution instruction proposed in [212] enables to compute a convolution producing a 2×2 output using a 3×3 kernel on a 4×4 input in a single instruction using 19 clock cycles instead of multiple instructions totaling 140 cycles using the standard RISC-V ISA. The set of general-purpose instructions for GCNs designed in [197] mitigate the compute inefficiencies in aggregating and combining feature vectors. As such the authors combine the software programmability given by the RISC-V ISA with the compute efficiency of GCN accelerators. Similarly, [93] focuses on transformer models. Notably, the extension comprises instructions to accelerate the well-known ReLU activation and softmax functions. Paulin et al. [154] performs a similar task but focuses on RNNs.

Many ISA extensions focus on low-bit-width arithmetics to accelerate inference of **Quantized Neural Networks** (**QNNs**), often combined with multi-core parallel execution to further boost performance and efficiency. Several developments augment the PULP RI5CY core used in Vega [172] to improve its energy efficiency on QNNs. Marsellus [37] (16 cores) and Kraken [48] (8 cores) use *Xpulpnn*, an ISA extension for low-bitwidth (2/4-bit) integer dot-products used to accelerate symmetric precision QNNs, which is further extended in *FlexNN* [143]. Dustin [148] (16 cores) also exploits a similar concept, but it also introduces a lockstep mechanism to operate all the cores in a SIMD fashion, further increasing their efficiency. Shaheen [202] exploits the same techniques in a more powerful SoC dedicated to applications for unmanned aerial vehicles. Many architectures, like Manticore [221], Occamy [155], CIFER [119], and DECADES [60], exploit ISA extensions for faster RISC-V based DL workload execution in the context of many-core architectures where a large number of cores cooperate On the other hand, this approach is also popular with emerging commercial platforms, such as Celerity [40], Esperanto [49], Tenstorrent [203], and speedAI240 [186]. All these architectures are targeted at datacenter-based training and batch inference of large DNNs, Transformers, and **Large Language Models** (**LLMs**): hence, they typically focus on floating point multiply-accumulate and dot-product operations, such as exSDOTP [12]. Finally, a growing trend is to integrate digital **in-memory computing** (**IMC**) devices inside the pipeline of RISC-V processors as instruction set extensions. Two notable early examples at the silicon prototype maturity level are given by RDCIM [220] and Zhou et al. [226].

4.4.2 RISC-V Vector Co-Processors. Vector co-processors represent a sort of natural architectural target for DL-oriented RISC-V acceleration. AVA [113], Vitruvius+ [138], Ara [20, 157, 158], Xvpfloat [78], MANIC [70] are academic vector co-processors meant to accelerate the full RISC-V *V* extension for vectorizable applications, including DL. Commercial RISC-V vector processors mainly targeted at HPC markets, such as Xuantie [25], and Ventana [207], have recently started appearing. In addition, vector co-processors explicitly tailored for DL, like Spatz [21] and Arrow [10], have been developed. The former, in particular, focuses only on a subset of the *V* extension and on 32-bit data, capturing better opportunities for energy efficiency. Further pushing this trend, Vecim [214] uses an IMC array to implement part of a reduced-precision (FP16) DL-dedicated vector extension for RISC-V.



a: Zhou et al. [226], ***b***: TinyVers [90], ***c***: FlexNN [143], ***d***: RDCIM [220], ***e***: Vega [172], ***f***: Darkside [65], ***g***: SamurAI [139], ***h***: Vecim [214], ***i***: Tambe et al. [195], ***j***: Archimedes [161], ***k***: SNCPU [100], ***l***: Marsellus [37], ***m***: DIANA [86], ***n***: Garofalo et al. [64], ***o***: Dustin [148], ***p***: RedMulE [198, 199], ***q***: Shaheen [202], ***r***: Manticore [221], ***s***: exSDOTP [12], ***t***: NewAra [158], ***u***: Kraken [48], ***v***: Siracusa [162], ***w***: Vitruvius+ [138], ***x***: CNC [26], ***y***: YUN [157], ***z***: Ara [20], ***A***: Spatz [21], ***B***: CIFER [119], ***C***: Axelera AI [215], ***D***: Bruschi et al. [16], ***E***: Metis AIPU [80], ***F***: Esperanto [49], ***G***: Occamy [155], ***H***: Lee et al. [117], ***I***: speedAI240 [186]

Fig. 5. Performance and power consumption of SotA DL accelerators based on open-HW RISC-V.

4.4.3 RISC-V Memory-coupled Neural Processing Units (NPUs). Concerning the tightest kind of memory coupling, at L1, most proposals in the state-of-the-art are based on the Parallel Ultra-Low Power (PULP) template, and devote significant effort to enabling fast communication between RISC-V cores and accelerators. Representative system architectures designed in this way are available at several levels of maturity, like the prototypes Vega [172] and Darkside [65], the commercial products GreenWaves Technologies GAP9 [74], Archimedes [161] and Siracusa [162], and the simulation templates [64] and [16].

Moving the shared memory from L1 to L2/L3, there are other NPU solutions in the state-of-the-art. For example, SNCPU [100], can act as either a set of 10 RISC-V cores or be reconfigured in a systolic NPU. In [72] and [66], systolic arrays generated by Gemmini are coupled with a RISC-V core by exploiting a shared L3 or L2 memory, respectively. Simba [182] is also meant to be scaled towards server-grade performance using the integration of chiplets on multi-chip modules. ESP [68, 69] and [195] also focus on integrating hardware accelerators and NPUs in large-scale Network-on-Chips using RISC-V cores as computing engines. Axelera AI propose a so-far unique architecture that uses a L2 shared-memory accelerator exploiting digital SRAM-based IMC, called Metis AIPU [80]. SamurAI [139], TinyVers [90], and DIANA [86] build up AI-IoT systems composed of a microcontroller and L2-coupled NPUs. Kraken [48] couples the RISC-V ISA-extended cluster with specialized L2-coupled Spiking Neural Network (SNN) and Ternary Neural Network (TNN) accelerators.

4.4.4 Summary. Figure 5 clearly shows that RISC-V-based solutions occupy essentially the full spectrum of DL architectures ranging from 10 mW microcontrollers up to 100 W SoCs meant to be integrated as part of HPC systems. So far, most of the research has focused on the lower end of this spectrum, striving for the best energy efficiency. We can observe how efficiency is strongly correlated with architectural techniques yielding accuracy (e.g., data bit-width reduction & quantization) and with the usage of emerging computational paradigms such as in-memory computing. Table 5 compares the above discussed architectures quantitatively and reports their highest performance and energy efficiency values.

5 Accelerators Based on Emerging Technologies

To design efficient DNN hardware accelerators, combining optimized memory architectures and processing modules is crucial to achieve high speed at reasonable costs and power dissipation.

Table 5. Summary of RISC-V Deep Learning Acceleration Architectures

Category	Accelerator	Tech [nm]	Area [mm2]	Freq [MHz]	Voltage [V]	Power [mW]	Perf [GOPS]	Eff [GOPS/W]	# MAC units	Data Type	Maturity
ISA	Dustin [148]	65	10	205	1.2	156	33.6	215	128	INT2 x INT4	Silicon
	Kraken (RISC-V cores) [48]	22	9	330	0.8	300	75	750	128	INT2	Silicon
	Manticore [221]	22	888	500	0.6	200	25	188	24	FP64	Pre-silicon
	Celerity [40]	16	25	1050	-	1900	-	-	496	INT32	Silicon
	Tenstorrent [203]	12	477	-	-	-	92000	-	-	FP16	Silicon
	exSDOTP [12]	12	0.52	1260	0.8	*278*	160	575	16	FP8	Pre-silicon
	Esperanto [49]	7	570	1000	-	20000	139000	6.95	69632	INT8	Silicon
	CNC [26]	4	1.92	1150	0.85	510	75.8	149	512	INT8	Silicon
	Occamy [155]	12	146	1000	0.8	23000	770	28.1	432	FP64	Silicon
	Zhou et al. [226]	28	-	50	-	13	3690	285	144000 1–bit	INT2 (IMC)	Simulation
	speedAI240 [186]	7	-	1300	-	67000	2000000	30000	372000	FP8	Simulation
	DECADES [60]	12	62	911	1.2	-	1460	-	60	INT64	Silicon
	CIFER [119]	12	16	1195	0.8	1792	15.54	6.63	14	FP32	Silicon
	RDCIM [220]	55	9.8	200	1.2	43	2820	66300	524288 1–bit	INT4 (IMC)	Silicon
	Shaheen [202]	22	9	500	0.8	195	26	133	8	INT2	Silicon
	FlexNN [143]	22	0.55	463	0.65	26	85	3260	128	INT2	Pre-silicon
Vector	Lee et al. [117]	14	181	2000	0.8	60000	64000	1450	16384	INT8	Silicon
	AVA [113]	22	3.9	-	-	-	-	-	-	FP64	Pre-silicon
	Spatz [21]	22	20	594	0.8	1070	285	266	256	INT32	Pre-silicon
	Vitruvius+ [138]	22	1.3	1400	0.8	459	21.7	47.3	8	FP64	Pre-silicon
	Ara [20]	22	10735 kGE	1040	0.8	794	32.4	40.8	16	FP64	Pre-silicon
	Perotti et al. [158]	22	0.81	1340	0.8	280	10.4	37.1	4	FP64	Pre-silicon
	Vecim [214]	65	4	250	1	110	25.3	230	4	FP16 (IMC)	Silicon
	MANIC [70]	22	0.57	48.9	1.05	2	0.512	256	1	INT32	Silicon
	YUN [157]	65	6	280	1.2	707	2.83	4	4	FP64	Silicon
	Xvpfloat [78]	7	0.14	1250	0.675	-	-	-	1	FP64	Pre-silicon
L1 NPU	Darkside [65]	65	3.85	200	1.2	89.1	12.6	152	32	FP16	Silicon
	Marsellus (NPU) [37]	22	18.7	420	0.8	123	637	7600	10368 1–bit	INT2	Silicon
	Garofalo et al. [64]	22	30	500	0.8	150	958	6390	36 (DW)	INT8	Pre-silicon
	Vega [172]	22	12	450	0.8	49.4	32.2	651	27	INT8	Silicon
	RedMulE [198, 199]	22	0.73	613	0.8	193	117	608	96	FP16	Pre-silicon
	Archimedes [161]	22	3.38	270	0.65	112	1198	10.6	5184	INT2 x INT8	Pre-silicon
	Bruschi et al. [16]	5	480	-	-	3070	20000	6500	3.35×10^7	Analog (IMC)	Simulation
	Siracusa [162]	16	16	360	0.8	332	1950	7000	10368 1×8–bit	INT2 x INT8	Silicon
L2 NPU	SNCPU [100]	65	4.47	400	1	116	75.9	655	100	INT8	Silicon
	SamurAI [139]	28	4.52	350	0.9	94.7	36	380	64	INT8	Silicon
	Gemmini [66]	22	1.03	1000	-	-	-	-	256	INT8	Pre-silicon
	DIANA (digital) [86]	22	10.24	280	0.8	132	230	1740	256	INT8	Silicon
	DIANA (analog) [86]	22	10.24	350	0.8	132	18100	176000	256	Analog (IMC)	Silicon
	TinyVers [90]	22	6.25	150	0.8	20	17.6	863	64	INT8	Silicon
	Simba [182]	16	6	161	0.42	-	-	9100	1024	INT8	Silicon
	Metis AIPU [80]	12	144	800	0.68	3490	52400	15000	-	INT8 (IMC)	Silicon
	Tambe et al. [195]	12	4.59	717	1	111	734	6612	-	FP4	Silicon
L3 NPU	Gonzalez et al. [72]	22	16	961	-	-	-	106.1	256	INT8	Silicon
	ESP [91]	12	21.6	1520	1	1830	-	-	3x NVDLA	INT8	Silicon
	Dos Santos et al. [50]	12	64	1600	1	4330	-	-	4x NVDLA	INT8	Silicon

Such architectures must be designed taking into account the large amount of memory necessary to store the input feature maps, weights, and intermediate results generated by the convolutional layers in a DNN. Moreover, managing DNN computational models causes a large number of data movements between the memory and the processing elements, often posing a challenge in terms of achievable speed performance, energy consumption, and memory bandwidth. For these reasons, several innovative memory architectures and technologies have recently emerged to increase memory capacity and data bandwidth, to reduce memory access latency, and potentially to improve the power efficiency. In this Section, we discuss several technologies: Processing-in-Memory and In-Memory Computing (see Figure 6), Neuromorphic accelerators, approaches based on Multi-Chip Modules, and Quantum and Photonic computing.

5.1 Processing-in-Memory

Processing-in-Memory (PIM) solutions are mostly implemented on DRAM modules. PIMs' computing elements can compute in parallel in all subarrays/banks, accessing data through the internal DRAM buses, and reducing the amount of data transferred between host and memory. Depending on where the computation is performed, we can identify three main categories of PIMs [173]: (1) In-subarray PIMs (the compute occurs at the local sense amplifiers), (2) bank-level PIMs (processing

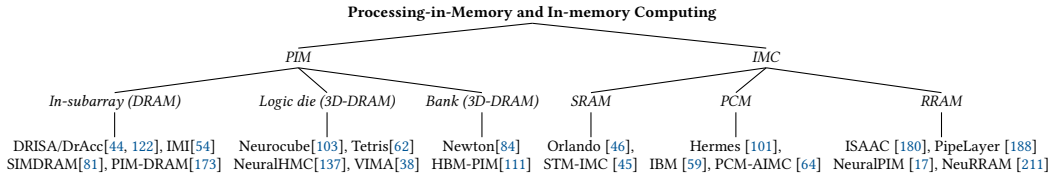


Fig. 6. Taxonomy of accelerators based on the emerging memory technologies discussed in Sections 5.1 and 5.2.

Table 6. Summary of Processing-in-Memory DNN Accelerators

PIM	Year	Integration Level	Mem. Tech.	Functions	Data Type	Tech. Node	Performance [GOPs/s]	Power [W]	Maturity
DRISA/DrAcc[44, 122]	2017	In-subarray	2D DRAM	XOR	variable	-	-	-	Simulation
IMI[54]	2017	In-subarray	2D DRAM	boolean	variable	-	-	-	Simulation
SIMDRAM[81]	2021	In-subarray	2D DRAM	MAJ/NOT	variable	-	-	-	Simulation
PIM-DRAM[173]	2021	In-subarray	2D DRAM	ADD/AND	variable	-	-	-	Simulation
Neurocube[103]	2016	Logic die	HMC	MAC	16-bit fixed point	15nm	132	3.4 + HMC	Layout
Tetris[62]	2017	Logic die	HMC	ALU/MAC	16-bit fixed point	45nm	-	8.42	Simulation
NeuralHMC[137]	2019	Logic die	HMC	MAC	32-bit floating point	-	-	-	Simulation
VIMA[38]	2021	Logic die	HMC	ALU/MULT/DIV	32-bit integer/floating point	-	-	3.2 + HMC	Simulation
Newton[84]	2020	Bank	HBM	MAC	bfloat16	-	-	-	Simulation
HBM-PIM[111]	2020	Bank	HBM	ALU/MAC	16-bit floating point	20nm	1200	-	Silicon

logic is integrated into each DRAM die at the level of the memory banks, after the column decoder and selector blocks), and (3) logic-die level PIMs (compute cores are embedded into the logic die of a 3D-stacked memory block). Table 6 presents a summary of the three types of PIM accelerators.

3D-stacked memory blocks rely on the possibility of stacking layers of conventional 2D DRAM or other memory types together with one or more optional layers of logic circuits. These logic layers are often implemented with different process technologies and can include buffer circuitry, test logic, and PEs. Two main 3D stacked memory standards have been recently proposed: the Hybrid Memory Cube (HMC) and the High Bandwidth Memory (HBM). They both provide highly parallel access to the memory, a sought-after characteristic in the highly parallel architecture of the DNN accelerators. The PEs of 3D stacked DNN accelerators can be embedded in the logic die or in the memory dies, significantly reducing the latency of accessing data in main memory, and improving the system energy efficiency. However, as detailed in Section A.7 of Appendix A, there are some challenges and limitations to consider when using this technology [104].

Most in-subarray PIMs for DNNs rely on solutions similar to Ambit [178] and RowClone [177] for implementing the computing elements. Ambit exploits the analog operation of DRAM technology to perform bit-wise AND, OR and NOT operations completely inside the DRAM. RowClone is a mechanism that efficiently copies rows inside the same DRAM subarray by exploiting the vast internal DRAM bandwidth without CPU intervention.

DRISA[122] leverages these technologies by implementing bit-wise XORs, and by expressing more complex functions as sequences of such a basic operation. Additional logic (e.g., shifters) and modifications in the memory controllers are needed for driving the execution of operation opcodes. Higher bit-widths are supported, but with the execution time increasing exponentially. However, multiple subarrays and banks provide large parallelism and large computational throughput. While DRISA evaluates the implementation of CNNs with binary weights, DrAcc [44] focuses on CNNs with ternary weights.

The Micron **In-Memory Intelligence** (**IMI**) architecture [54] is built on simple bit-serial computing elements placed below standard DRAM array's sense-amplifiers and provides the memory block with the ability for massive SIMD parallelism by supporting vector instructions over an entire bank. Complex operations are implemented as serial sequences of basic logic functions, such as

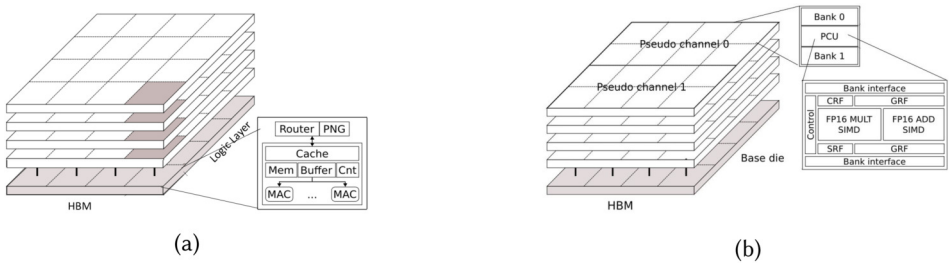


Fig. 7. (a) Neurocube architecture. (b) HBM-PIM architecture.

XOR and AND. A control unit is attached to each DRAM bank and translates the IMI instruction to be executed into row-cycles that control the SIMD computing elements.

SIMDRAM [81] is a flexible general-purpose processing-using-DRAM framework that provides a flexible mechanism to support the implementation of arbitrary user-defined operations as sequences of basic functions including MAJ and NOT. The sequence of DRAM commands generated by the framework are executed by a control unit located inside the memory controller, which manages the computation from start to end.

PIM-DRAM [173] is an in-subarray PIM, which performs multiplications as sequences of bit-wise in-subarray addition and AND operations while the accumulation and activation functions are performed in the bank architecture. The processing happens by enabling multiple wordlines at the same time to leverage the large available internal DRAM bandwidth.

A first example of a logic die-level PIM implementation is Neurocube [103] that, as shown in Figure 7(a), is embedded into the logic die of an HMC, and consists of a cluster of PEs connected by a 2D mesh Network-on-Chip (NoC). The PE is composed of a row of multiply accumulator (MAC) units, a cache memory, a temporal buffer, and a memory module for storing shared synaptic weights. Each PE is associated with a single memory vault and can operate independently and communicate through the TSVs and the vault controller. A host communicates with the Neurocube through the external links of the HMC to configure the Neurocube for different neural network architectures. Each vault controller in the HMC has an associated programmable neuro sequence generator (PNG), i.e., a programmable state machine that controls the data movements required for neural computation. Neurocube implements an output stationary dataflow.

Tetris [62] uses an HMC memory stack organized into 16 vaults. Each vault is associated with a PE, connected to the vault controller, and composed of a systolic array of 14×14 PEs and a small SRAM buffer, shared among the PEs. A 2D mesh NoC connects all the PEs. The dimension of the buffers in the logic layer is reduced and optimized to take into account the lower cost of accessing the DRAM layers, as well as the area constraints of the 3D package. Each PE has a register file and a MAC locally storing the inputs/weights and performing computations. Tetris implements a row stationary dataflow that maps 1D convolutions onto a single PE. A 2D convolution is orchestrated on the 2D array interconnect so that the data propagation among PEs remains local. In [62], an optimal scheduling is discussed to maximize on-chip reuse of weights and/or activations, and resource utilization. However, a programming model is not presented.

NeuralHMC [137] adopts a weight-sharing pipelined MAC to lower the cost of accessing weight data, by reducing the original 32-bit floating-point weights to a 5 or 8-bit cluster index, saving memory consumption. Moreover, it allows reducing and optimizing packet scheduling and on-chip communication in multi-HMC architectures.

The HIVE architecture [5] extends the HMC ISA for performing common vector operations directly inside the HMC. By migrating ML kernels on near-data processing (NDP) architectures

Table 7. Summary of IMC Accelerators Based on RRAM and PCM Memories

Accelerator	Technology	Process	Application	Area [mm^2]	Power [mW]	Performance [GOPS]	EE [GOPS/W]	AE [GOPS/mm^2]	Maturity-level
ISAAC [180]	RRAM+CMOS	32 nm	CNN	85.4	65800	-	380.7	466.8	Simulation
PipeLayer [188]	RRAM+CMOS	-	CNN	82.63	-	-	140	1485	Simulation
NeuralPIM [17]	RRAM+CMOS	32 nm	CNN+RNN	86.4	67700	-	2040.6	1904	Simulation
PRIME [31]	RRAM+CMOS	65 nm	MLP+CNN	-	-	-	2100	1230	Simulation
NeuRRAM [211]	RRAM+CMOS	130 nm	CNN+RNN+RBN	159	49.7	2135	43000	-	Layout
Hermes [101]	PCM+CMOS	14 nm	MLP+CNN+LSTM	-	-	-	10500	1590	Silicon

capable of large-vector operations, the Vector-In-Memory Architecture (VIMA) proposed in [38] supports all ARM NEON Integer and floating-point instructions, operating over vectors of 8 KB of data by fetching data over the 32 channels (vaults) of the HMC in parallel. In this way, it leads to a significant speed-up and energy reduction with respect to an x86 baseline.

Several accelerators adopting the bank-level PIM approach can be found in the literature. The Newton fixed data flow accelerator proposed in [84] employs only MAC units, buffers, and a DRAM-like command interface with the host CPU, avoiding the overhead and granularity issues of launching the kernel and switching between the PIM/non-PIM operational modes. The output vector write traffic is reduced by means of an unusually wide interleaved layout (DRAM row-wide). Moreover, input/output vectors have high reuse while the matrix has no reuse.

HBM-PIM [111] implements a **function-in-memory DRAM** (**FIMDRAM**) that integrates a 16-wide SIMD engine within the memory banks exploits bank-level parallelism to provide 4× higher processing bandwidth than an off-chip memory solution (Figure 7(b)). Each computing unit (PCU) is shared among two banks, and there are 8 PCUs per pseudo-channel. The PCU is divided into a register group, an execution unit, a decoding unit for parsing instructions needed to perform operations, and interface units to control data flow. The register group consists of a command-register file for instruction memory (CRF), a general-purpose register file for weight and accumulation (GRF), and a scalar register file to store constants for MAC operations (SRF). The PIM controller is integrated to support the programmability of the PCU and the seamless integration with the host by determining the switching between the PIM/non-PIM operational modes. If the PIM mode is asserted, the PCUs execute the instructions pre-stored in the CRF, incrementing the program counter every time a DRAM's read command is issued. 3D-stacked PIM has also been proposed for accelerating applications loosely related to DNNs. We present a brief overview of these accelerators in Section A.7 of the Appendix A.

5.2 In-Memory Computing

IMC has been proposed to break both the memory and the compute wall in data-driven AI workloads, using either SRAM or emerging memory technologies (such as PCM and RRAM described in Section A.8 of Appendix A) integrated in a dedicated accelerator (Table 7).

Full-digital IMC designs offer a fast path for the integration of the next generation of neural processing systems like NPUs. Recently, STMicroelectronics proposed a scalable and design time parametric IMC-NPU relying on digital SRAM IMC for edge AI [45]. This architecture is the evolution of the Orlando SoC [46] and is specialized in accelerating the inference workloads. When manufactured in 18 nm FDSOI technology, this IMC-NPU achieves an energy efficiency of 77 TOPS/W and an area efficiency of 13.6 TOPS/mm^2. With its four key features and dedicated hardware able to lower the activity of the memory early terminating the operations when needed, NeuroCIM [106] achieves 310.4 TOPS/W. The ISAAC non-volatile inference-based machine on RRAM technology [180] is a tile-based architecture for CNN processing which combines the data encoding and the processing steps within *in situ* MAC units (IMA). The design is pipelined

fetching data from an external eDRAM chip to the computing tile. The data format in ISAAC is fixed at 16-bit. During computation, at each clock cycle, 1-bit is given as input to the IMA, whose result is converted to the digital format, thus requiring 16 clock cycles to process the input. Such a design allows implementing the computation on different tiles in a fully pipelined approach to increase computing performance and throughput. The PipeLayer [188] architecture introduces intra-layer parallelism and an inter-layer pipeline for tiled architecture, using duplicates of processing units featuring the same weights to process multiple data in parallel. Designs like PRIME [31] take part of the RRAM memory arrays to serve as acceleration instead of adding an extra processing unit for computation. As outlined in Reference [17], existing PIM RRAM accelerators suffer from frequent and energy-intensive analog-to-digital (A/D) conversions, severely limiting their performance. To efficiently accelerate DL tasks by minimizing the required A/D conversions, a new architecture was presented with analog accumulation and neural-approximated peripheral circuits. The new dataflow introduced in [17] remarkably reduces the required A/D conversions for matrix-vector multiplications by extending shift-and-add operations to the analog domain before the final quantization. A summary of the technological features in major RRAM accelerators can be found in [184].

The first PCM-based silicon demonstrator for DNN inference is Hermes [101] which consists of a 256x256 PCM cross-bar and optimized ADC circuitry to reduce the read-out latency and energy penalty. The SoC is implemented in 14nm technology, showing 10.5 TOPS/W energy efficiency and performance density of 1.59 TOPS/mm^2. The same 256x256 PCM cross-bar has been integrated into a scaled-up mixed-signal architecture that targets the inference of **long short-term memory** (**LSTM**) and ResNet-based neural networks [59]. The chip, implemented in the same 14nm technology, consists of 64 analog cores interconnected via an on-chip communication network and complemented with digital logic to execute activation functions, normalization, and other kernels than **Matrix-Vector Multiplications** (**MVMs**). The accelerator achieves a peak throughput of 63.1 TOPS with an energy efficiency of 9.76 TOPS/W for 8-bit input/8-bit output MVM operations.

Besides silicon stand-alone demonstrators, the PCM technology is evaluated from a broader perspective in heterogeneous architectures that target different classes of devices, from IoT end nodes to many-core HPC systems. Such studies aim to highlight and overcome the system-level challenges that arise when PCM technology is integrated into more complex mixed-signal systems. For example, Garofalo et al. [64] analyze the limited flexibility of AIMC cores that can only sustain MVM-oriented workloads, but they are inefficient in executing low-reuse kernels and other ancillary functions such as batch-normalization and activation functions. To better balance Amdahl's effects that show up on the execution of end-to-end DNN inference workloads, they propose as a solution an analog-digital edge system that complements the computing capabilities of PCM-based accelerators with the flexibility of general-purpose cores. The architecture, benchmarked on a real-world MobileNetV2 model, demonstrates significant advantages over purely digital solutions. Bruschi et al. [16] leave the edge domain to study the potentiality of PCM-based AIMC in much more powerful HPC many-core systems. The work presents a general-purpose chipset-oriented architecture of 512 processing clusters, each composed of RISC-V cores for digital computations and nvAIMC cores for analog-amenable operations, such as 2D convolutions. This system is benchmarked on a ResNet18 DNN model, achieving 20.2 TOPS and 6.5 TOPS/W.

5.3 Neuromorphic Accelerators

Neuromorphic computing represents a paradigm shift from Von Neumann-based architectures to distributed and co-integrated memory and PEs [56]. Neuromorphic chip architectures enable the hardware implementation of spiking neural networks (SNNs) [168] and advanced bio-inspired

Table 8. Summary of Neuromorphic Accelerators

Chip name	Technology	Cores	Core Area [mm^2]	Neurons per core	Synapses per core	Weights storage	Supply Voltage [V]	Energy per SOP [J]
SpiNNaker [149]	0.13 μm	18	3.75	1000	-	Off-chip	1.2	>11.3n/26.6n
[176]	45 nm SOI	1	0.8	256	64k	1-bit SRAM	0.53 - 1.0	-
ODIN [57]	28 nm FDSOI	1	0.086	256	64k	(3+1)-bits (SRAM)	0.55 - 1.0	8.4p/12.7p
MorphIC [58]	65 nm LP	4	0.715	512	528k	1-bit (SRAM)	0.8 - 1.2	30p/51p
TrueNorth [3]	28 nm	4096	0.095	256	64k	1-bit (SRAM)	0.7 - 1.05	26p
Loihi [41]	14 nm FinFET	128	0.4	1024	1M	1- to 9 bits (SRAM)	0.5 - 1.25	>23.6p

computing systems that have the potential to achieve even higher energy efficiency with respect to DNN stand-alone accelerators described so far [3].

SpiNNaker chip [149] is a digital architecture designed on a 130 nm technology for SNN and neuroscience simulation acceleration. It is based on a distributed von Neumann approach using a globally asynchronous locally synchronous (GALS) design for efficient handling of asynchronous spike data. The SpiNNaker 2 system [126] uses a 22 nm technology and embeds 4 ARM Cortex M4F cores out of the planned 152 per chip. The objective is to simulate two orders of magnitude more neurons per chip compared to [149]. However, it has been demonstrated that GPU-based accelerators compare favorably to a SpiNNaker system when it comes to large SNN and cortical-scale simulations [107].

In comparison with the above-described accelerators, full-custom digital hardware leads to higher-density and more energy-efficient neuron and synapse integration for SNN [56]. The 45 nm design in Reference [176] is a small-scale architecture embedding 256 **Leaky-Integration-Fire** (**LIF**) neurons and up to 64k synapses based on the **Stochastic Synaptic Time Dependant Plasticity** (**S-STDP**) concept. Due to its reasonably high neuron and synapse densities and energy-efficiency, this design is an ideal choice for edge computing scenarios. At the same integration scale, the ODIN chip embeds 256 neurons and 64k **Spike Driven Synaptic Plasticity** (**SDSP**)-based 4-bit synapses in a 28 nm CMOS process [57]. A first attempt to scale up the NPU for SNN is represented by the 65 nm MorphIC chip, which is based on the ODIN core integrated into a quadcore design [58].

Two large-scale neuromorphic platforms required for cognitive computing applications, are currently offered: the 28 nm IBM TrueNorth [3] and the 14 nm Intel Loihi [41]. TrueNorth is a GALS design embedding as high as 1M neurons and 256M binary non-plastic synapses per chip, where neurons rely on a custom model that allows modifying their behaviors by combining up to three neurons [19]. Loihi is a fully asynchronous design embedding up to 180k neurons and 114k (9-bit) to 1M (binary) synapses per chip. Neurons rely on a LIF model with a configurable number of compartments to which several functionalities such as axonal and refractory delays, spike latency, and threshold adaptation have been added. The spike-based plasticity rule used for synapses is programmable. Loihi will evolve then in the new Loihi 2 neuromorphic chip and TrueNorth into the NorthPole platform [169].

Digital designs for neuromorphic chips can obtain versatility with a joint optimization of power and area efficiencies. This flexibility is highlighted with platforms going from versatility-driven (e.g., SpiNNaker) to efficiency-driven (e.g., ODIN and MorphIC), through platforms aiming at a well-balanced trade-off on both sides (e.g., Loihi). Table 8 summarizes the main characteristics of the neuromorphic chips described so far with particular insight on the Energy per spike operation (SOP).

5.4 Accelerators Based on Multi-Chip Modules

The alternate multichip-module (**MCM**) silicon interposer-based integration technology, described in Section A.9 of Appendix A, offers several advantages over single-chip designs, including increased functionality, reduced power consumption, higher performance, improved reliability, and cost

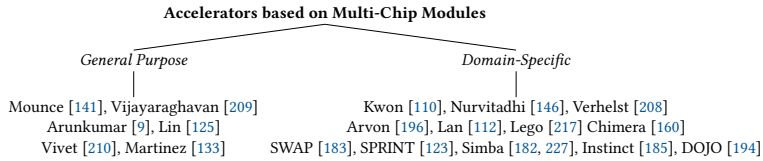


Fig. 8. Taxonomy of MCM based accelerators discussed in Section 5.4.

savings. By utilizing MCM, designers can combine multiple chips and functionalities into a single package, resulting in a reduced overall footprint and cost. Furthermore, MCM-based designs can utilize off-the-shelf components and existing manufacturing processes and technology, contributing to cost savings in overall manufacturing.

Figure 8 illustrates a comprehensive taxonomy of the MCM-based designs explored in this survey. Specifically, Section A.9.2 of Appendix A discusses a collection of representative general-purpose MCM-based designs, while this section concentrates on MCM-based DNN accelerators.

In the realm of DL, chiplet-based design is utilized to create hardware accelerators that are both efficient and scalable. The chiplet-based design proposed in [110] is a viable solution to provide higher performance at a lower cost compared to IP-based design. In [110], various aspects of designing a chiplet AI processor are considered, including incorporating NPU chiplets, HBM chiplets, and 2.5D interposers, ensuring signal integrity for high-speed interconnections, power delivery network for chiplets, bonding reliability, thermal stability, and interchiplet data transfer on heterogeneous integration architecture.

At the aim of balancing both data movement and compute capabilities of data-intensive DL algorithms, keeping the entire DL model on-chip is becoming the new norm for real-time services to avoid expensive off-chip memory accesses. In [146] it is shown how the integration of FPGA with ASIC chiplets enhances on-chip memory capacity and bandwidth and provides compute throughput that outperforms GPU-based platforms (NVIDIA Volta). Specifically, the GPU and chiplet-based FPGA computing capabilities are 6% and 57% of their peak, respectively. Moreover, the FPGA achieves a delay that is 1/16 and energy efficiency that is 34x better than the GPU.

In accordance with the recent trend in DL accelerators, chiplet integration is considered a promising implementation strategy for both homogeneous and heterogeneous multi-core accelerators to further increase throughput and match the ever-growing computational demands [208]. In Reference [112], a chiplet-based architecture is proposed for a multi-core neuromorphic processor with a chip-package co-design flow. The proposed design is reusable for different neuromorphic computing applications by scaling the number of chips in a package and by reusing existing IPs from different technology nodes with 2.5D integration technology. The MCM architecture presented in Reference [217] is a promising approach to address the issue of using modern DNN accelerators in multi-tenant DNN data centers, but it leaves the challenge of distributing DNN model layers with different parameters across chiplets still open. When a dynamic scheduler is used to comply with the size of DNN model layers and increase chiplet utilization, the Lego MCM architecture achieves a 1.51x speedup over a monolithic DNN accelerator. Chimera [160] is a non-volatile chip for DNN training and inference that does not require off-chip memory. Multiple Chimera accelerator chiplets can be combined in a multi-chip system to enable inference on models larger than the single-chip memory with only 5% energy overhead. The Arvon accelerator [196] is a heterogeneous System-in-Package that integrates a 14-nm FPGA chiplet with two 22-nm DSP chiplets using embedded multidie interconnect bridge technology. Arvon is designed to support various workloads, including neural network processing through

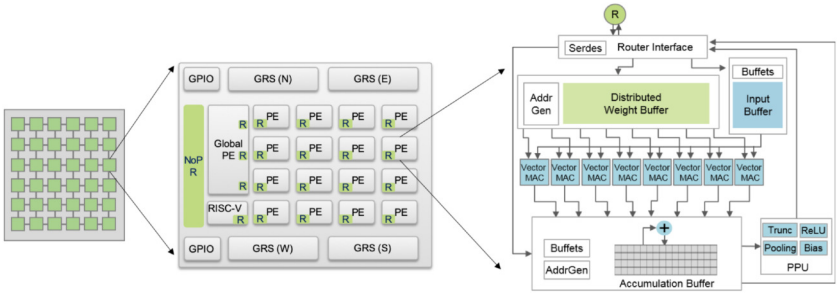


Fig. 9. Simba architecture [182] from left to right: package with 36 chiplets, chiplet, and processing element.

a specific compilation procedure that optimizes workload distribution across the FPGA and DSP chiplets.

Notable examples of commercial chiplet-based hardware accelerators include AMD's Instinct [185] and Tesla's DOJO [194]. The AMD Instinct MI300 series are designed for HPC and AI at exascale levels. They utilize a modular chiplet architecture that integrates data center-class CPUs, GPU accelerated compute, AMD Infinity Cache, and 8-stack HBM3 memory in a single package. Specifically, the MI300X model targets traditional dual-processor CPU servers with eight GPU accelerators, host DDR, and device HBM for large AI model training and inference. The MI300A model combines three Zen 4 CPU chiplets with six CDNA 3 GPU chiplets for high-density HPC systems, facilitating seamless CPU-GPU interaction without explicit data transfers. The Tesla's DOJO system is an exascale computer designed for ML training applications. It features a homogeneous modular architecture with a hierarchical chiplet-based organization. Each chiplet, named D1 die, contains 354 DOJO nodes, each functioning as a full-fledged computer with dedicated processing pipelines, local memory, and network interfaces. A training tile is formed by integrating 25 D1 dies within a single package.

SWAP [183] is a DNN inference accelerator based on the 2.5D integration of multiple resistive RAM chiplets. In [183], a design space exploration flow is proposed to optimize the interconnection Network-on-Package, minimizing inter-chiplet communications and enabling link pruning. Further improvements are achieved in SPRINT [123], where a photonic-based interconnects is used as an alternative to metallic-based inter-chiplet networks.

Finally, as a representative chiplet-based DNN hardware accelerator, we report Simba [182, 227]. Simba is a scalable DNN accelerator consisting of 36 chiplets connected in a mesh network on a multi-chip module using ground-referenced signaling. Simba enables flexible scaling for efficient inference on a wide range of DNNs, from mobile to data center domains. The prototype achieves high area efficiency, energy efficiency, and peak performance for both one-chiplet and 36-chiplet systems. Simba architecture is shown in Figure 9. It implements a tile-based architecture and adopts a hierarchical interconnect to efficiently connect different PEs. This hierarchical interconnect consists of a **network-on-chip** (**NoC**) that connects PEs on the same chiplet and a **network-on-package** (**NoP**) that connects chiplets together on the same package. Each Simba chiplet contains an array of PEs, a global PE, an NoP router, and a controller, all connected by a chiplet-level interconnect. Table 9 presents a summary of the key characteristics of a representative subset of chiplet-based DNN accelerators that were reviewed earlier.

5.5 Accelerators Based on Quantum Computing and Photonic Computing

Before concluding this Section, we would like to introduce some open challenges on two promising technologies to further speed up AI workloads: Quantum Computing and Photonic Computing.

Table 9. Summary of Chiplet-Based DNN Accelerators

	Simba [182]	Chimera [160]	Arvon [196]	Instinct [185]	DOJO [194]
Technology	16nm	40nm	14nm FPGA 22nm DSP	6nm FinFET	7nm
Area	6 mm^2*	29.2 mm^2	32.3 mm^2	-	645 mm^2
Power Efficiency	9.1 TOPS/W**	2.2 TOPS/W	1.8 TFLOPS/W	0.7 TFLOPS/W	0.6 TFLOPS/W
Throughput	4–128 TOPS	0.92 TOPS	4 TFLOPS	383 TFLOPS	362 TFLOPS
Frequency	161 MHz–1.8 GHz	200 MHz	800 MHz	1.7–2.1 GHz	2 GHz
Precisions	int8	int8, fp16	fp16	multi	multi
On-chip Memory	752 KiB*	2.5 MB†	-	4 MB	1.25 MB
Chiplet Bandwidth	100 GB/s	1.9 Gbps	7.68 Tb/s	17.2 TB/s	18 TB/s
Interconnect	Wired Mesh (GRS)	Wired App. specific‡	Wired (EMIB)	Infinity Fabric	2D mesh
Applications	CNN Inference	Inference Training	NN, Comm. signal proc.	Inference Training	Training

*One chiplet, **When operating at a minimum voltage of 0.42 V with a 161 MHz PE frequency.
†2 MB RRAM, 0.5 MB SRAM, ‡C2C links (77 pJ/bit, 1.9 Gbits/s).

There is a general agreement that Quantum computers will not replace conventional computing systems, but they will be used in combination with supercomputers to accelerate some hard-to-compute problems. Quantum computers will play the role of unconventional accelerators to outperform conventional supercomputers, thanks to the improved parallelism that enables the so-called *quantum speedup*. Governments, supercomputing centers, and companies around the world have also started to investigate *How*/*When*/*Where* quantum processing units (QPUs) could fit into HPC infrastructures to speed up some heavy tasks, such as DL workloads. Emerging trends and commercial solutions related to *hybrid* quantum-classical supercomputers are described in Reference [87]. To address this challenging trend, in October 2022, the EuroHPC Joint Undertaking initiative selected six supercomputing centers across the European Union to host quantum computers and simulators. IBM Research was the first provider to offer a cloud-based QC service. IBM Qiskit [165] is an open-source SDK based on a library of quantum gates/circuits: Remote users can develop quantum programs and execute them on quantum simulators and cloud-based quantum processors. Cloud providers have also jumped into the quantum race. As an example, Amazon Braket [14] is a QC service based on different types of quantum systems and simulators, including the quantum annealer from D-Wave. On this trend, there is a general agreement that GPUs will play a key role in hybrid quantum-classical computing systems. GPU company NVIDIA offers CuQuantum DGX hardware appliance integrating a software container on a full-stack quantum circuit simulator: The system uses NVIDIA's A100 GPUs to accelerate quantum simulation workloads.

Recently, a survey on QC technologies appeared in Reference [79], while another survey on QC frameworks appeared in Reference [201]. More specifically, there is a promising research trend on *Quantum Machine Learning* [13] which aim at developing quantum algorithms that outperform classical computing algorithms on ML tasks such as recommendation systems. More in detail, classical DNNs inspired the development of *Deep Quantum Learning* methods. The main advantage of these methods is that they do not require a large, general-purpose quantum computer. Quantum annealers, such as the D-Wave commercial solutions [39], are well-suited for implementing deep quantum learners. Quantum annealers are special-purpose quantum processors that are significantly easier to construct and scale up than general-purpose quantum computers. Following this research trend, Google proposed TensorFlow Quantum (TFQ) [15], an open-source quantum machine learning library for prototyping hybrid quantum-classical ML models.

The second challenging and promising research direction is represented by the use of Photonic Computing to further accelerate DL tasks. Photonic Computing relies on the computation of electromagnetic waves typically via non-linear modulation and interference effects. It was originally introduced in the 1980s to address optical pattern recognition and optical Fourier transform processing [6]. Despite the potential advantages of processing parallelism and speed, optical computing has never translated into a widely adopted commercial technology. Only recently, due to the emergence of data-intensive computing tasks, such as AI, optical computing has seen a renewed interest. There are two main advantages of optical computing, namely (i) the inherent speed of signal transmission, where light pulses can be transferred without the typical RC delays and IR drop of electrical interconnects, and (ii) the inherent parallelism, where multiple wavelengths, polarizations, and modes can be processed by the same hardware (e.g., waveguides, interferometers, etc.), without interfering with each other. These properties can provide strong benefits to data-intensive computing tasks such as DL. Photonic computing represents a promising platform for accelerating AI. For instance, it has been estimated that photonic MAC operations can show significant improvements over digital electronics in terms of energy efficiency ($> 10^2$), speed ($> 10^3$), and compute density ($> 10^2$) [144]. However, there are still many challenges to developing an industrially feasible photonic system. The main challenge is the area/energy inefficiency of processing across the mixed optical/electronic domain. Optical-electrical conversion and vice versa result in considerable overhead in terms of area and power consumption. To bridge this gap, the trend is developing silicon **photonic integrated circuits** (**PICs**) with increasing robustness, manufacturability, and scalability. Photonic computing essentially operates in the analog domain, thus accuracy is deeply affected by accumulated noise and imprecision of optical devices, such as electro-optic and phase change modulators. These challenges, similar to those arising in analog IMC, might be mitigated by hardware-aware training and system-level calibration techniques.

6 Conclusions

The DL ecosystem based on advanced computer architectures and memory technologies spans from edge and IoT computing solutions to high-performance servers, supercomputers, and up to large data centers for data analytics. In this context, the main objective of this survey is to provide an overview of the leading computing platforms utilized for accelerating the execution and enhancing the energy efficiency of DL applications. Although GPUs have been crucial to boosting the DL revolution, especially for crunching in parallel a large amount of data, they are not the *panacea* for all types of AI-applications. There are plenty of much smaller and customized AI accelerators, boosting their energy efficiency to make them suitable for mobile resource-constrained devices at a reasonable market price and without relying on sending data to the cloud. This survey reviews not only GPU-based solutions and Tensor Processor Units, but also ASIC- and FPGA-based accelerators, Neural Processing Units, and customized co-processors based on the open-hardware RISC-V architecture. To push further on the more advanced AI solutions, the survey also describes accelerators based on emerging computing paradigms and technologies, such as 3D-stacked processing in memory, emerging non-volatile memories, Multi-Chip Modules, chiplets, quantum-and photonic-based accelerating solutions.

Acknowledgments

This work has been supported by the Spoke 1 on *Future HPC* of the Italian Research Center on High-Performance Computing, Big Data and Quantum Computing (ICSC) funded by MUR Mission 4 - Next Generation EU.

References

[1] Ankur Agrawal, Sae Kyu Lee, Joel Silberman, Matthew Ziegler, Mingu Kang, Swagath Venkataramani, Nianzheng Cao, Bruce Fleischer, Michael Guillorn, Matthew Cohen, Ophir Erez, Thomas Fox, George Gristede, Howard Haynie, Vicktoria Ivanov, Siyu Koswatta, Shih-Hsien Lo, Martin Lutz, Gary Maier, Alex Mesh, Yevgeny Nustov, Scot Rider, Marcel Schaal, Michael Scheuermann, Xiao Sun, Naigang Wang, Fanchieh Yee, Ching Zhou, Vinay Shah, Brian Curran, Vijayalakshmi Srinivasan, Pong-Fei Lu, Sunil Shukla, Kailash Gopalakrishnan, and Leland Chang. 2021. 9.1 A 7nm 4-core AI chip with 25.6 TFLOPS hybrid FP8 training, 102.4 TOPS INT4 inference and workload-aware throttling. In *2021 IEEE International Solid-State Circuits Conference (ISSCC)*, Vol. 64. IEEE, 144–146. DOI: https://doi.org/10.1109/ISSCC42613.2021.9365791

[2] Alessandro Aimar, Hesham Mostafa, Enrico Calabrese, Antonio Rios-Navarro, Ricardo Tapiador-Morales, Iulia-Alexandra Lungu, Moritz B. Milde, Federico Corradi, Alejandro Linares-Barranco, Shih-Chii Liu, and Tobi Delbruck. 2019. NullHop: A flexible convolutional neural network accelerator based on sparse representations of feature maps. *IEEE Transactions on Neural Networks and Learning Systems* 30, 3 (2019), 644–656. DOI: https://doi.org/10.1109/TNNLS.2018.2852335

[3] Filipp Akopyan, Jun Sawada, Andrew Cassidy, Rodrigo Alvarez-Icaza, John Arthur, Paul Merolla, Nabil Imam, Yutaka Nakamura, Pallab Datta, Gi-Joon Nam, Brian Taba, Michael Beakes, Bernard Brezzo, Jente B. Kuang, Rajit Manohar, William P. Risk, Bryan Jackson, and Dharmendra S. Modha. 2015. TrueNorth: Design and tool flow of a 65 mW 1 million neuron programmable neurosynaptic chip. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 34, 10 (2015), 1537–1557. DOI: https://doi.org/10.1109/TCAD.2015.2474396

[4] Jorge Albericio, Patrick Judd, Tayler Hetherington, Tor Aamodt, Natalie Enright Jerger, and Andreas Moshovos. 2016. Cnvlutin: Ineffectual-neuron-free deep neural network computing. In *Proceedings of the 43rd International Symposium on Computer Architecture (ISCA'16)*. 13. DOI: https://doi.org/10.1109/ISCA.2016.11

[5] Marco A. Z. Alves, Matthias Diener, Paulo C. Santos, and Luigi Carro. 2016. Large vector extensions inside the HMC. In *2016 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. 1249–1254.

[6] Pierre Ambs. 2010. Optical computing: A 60-year adventure. *Advances in Optical Technologies* 2010, 372652 (May 2010). DOI: https://doi.org/10.1155/2010/372652

[7] AMD. 2021. AMD Instinct MI200 series accelerator. (Jan 2021). Retrieved May 25, 2023 from https://www.amd.com/system/files/documents/amd-instinct-mi200-datasheet.pdf

[8] Michael Andersch, Greg Palmer, Ronny Krashinsky, Nick Stam, Vishal Mehta, Gonzalo Brito, and Sridhar Ramaswamy. 2022. NVIDIA Hopper Architecture In-Depth. (Mar 2022). Retrieved Apr 16, 2023 from https://developer.nvidia.com/blog/nvidia-hopper-architecture-in-depth/

[9] Akhil Arunkumar, Evgeny Bolotin, Benjamin Cho, Ugljesa Milic, Eiman Ebrahimi, Oreste Villa, Aamer Jaleel, Carole-Jean Wu, and David Nellans. 2017. MCM-GPU: Multi-chip-module GPUs for continued performance scalability. In *2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA)*. 320–332.

[10] Imad Al Assir, Mohamad El Iskandarani, Hadi Rayan Al Sandid, and Mazen A. R. Saghir. 2021. Arrow: A RISC-V vector accelerator for machine learning inference. arXiv:2107.07169. Retrieved from https://arxiv.org/abs/2107.07169

[11] Yoshua Bengio. 2009. Learning deep architectures for AI. *Foundations and Trends in Machine Learning* 2, 1 (2009), 1–127. DOI: https://doi.org/10.1561/2200000006

[12] Luca Bertaccini, Gianna Paulin, Tim Fischer, Stefan Mach, and Luca Benini. 2022. MiniFloat-NN and ExSdotp: An ISA extension and a modular open hardware unit for low-precision training on RISC-V cores. In *2022 IEEE 29th Symposium on Computer Arithmetic (ARITH)*. 1–8. DOI: https://doi.org/10.1109/ARITH54963.2022.00010

[13] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. 2017. Quantum machine learning. *Nature* 549, 7671 (sep 2017), 195–202. DOI: https://doi.org/10.1038/nature23474

[14] Braket 2023. Quantum Computing Service—Amazon Braket—AWS. (2023). Retrieved from https://aws.amazon.com/braket/

[15] Michael Broughton, Guillaume Verdon, Trevor McCourt, Antonio J. Martinez, Jae Hyeon Yoo, Sergei V. Isakov, Philip Massey, Ramin Halavati, Murphy Yuezhen Niu, Alexander Zlokapa, Evan Peters, Owen Lockwood, Andrea Skolik, Sofiene Jerbi, Vedran Dunjko, Martin Leib, Michael Streif, David Von Dollen, Hongxiang Chen, Shuxiang Cao, Roeland Wiersema, Hsin-Yuan Huang, Jarrod R. McClean, Ryan Babbush, Sergio Boixo, Dave Bacon, Alan K. Ho, Hartmut Neven, and Masoud Mohseni. 2021. TensorFlow Quantum: A Software Framework for Quantum Machine Learning. (2021). arXiv:2003.02989. DOI: https://doi.org/10.48550/arXiv.2003.02989

[16] Nazareno Bruschi, Giuseppe Tagliavini, Angelo Garofalo, Francesco Conti, Irem Boybat, Luca Benini, and Davide Rossi. 2023. End-to-end DNN Inference on a massively parallel analog in memory computing architecture. In *Design, Automation & Test in Europe Conference & Exhibition, DATE*. IEEE, 1–6. DOI: https://doi.org/10.23919/DATE56975.2023.10137208

[17] Weidong Cao, Yilong Zhao, Adith Boloor, Yinhe Han, Xuan Zhang, and Li Jiang. 2022. Neural-PIM: Efficient processing-in-memory with neural approximation of peripherals. *IEEE Trans. Comput.* 71, 9 (2022), 2142–2155. DOI: https://doi.org/10.1109/TC.2021.3122905

[18] Alex Carsello, Kathleen Feng, Taeyoung Kong, Kalhan Koul, Qiaoyi Liu, Jackson Melchert, Gedeon Nyengele, Maxwell Strange, Keyi Zhang, Ankita Nayak, Jeff Setter, James Thomas, Kavya Sreedhar, Po-Han Chen, Nikhil Bhagdikar, Zachary Myers, Brandon D'Agostino, Pranil Joshi, Stephen Richardson, Rick Bahr, Christopher Torng, Mark Horowitz, and Priyanka Raina. 2022. Amber: A 367 GOPS, 538 GOPS/W 16nm SoC with a coarse-grained reconfigurable array for flexible acceleration of dense linear algebra. In *2022 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*. 70–71. DOI: https://doi.org/10.1109/VLSITechnologyandCir46769.2022.9830509

[19] Andrew S. Cassidy, Paul Merolla, John V. Arthur, Steve K. Esser, Bryan Jackson, Rodrigo Alvarez-Icaza, Pallab Datta, Jun Sawada, Theodore M. Wong, Vitaly Feldman, Arnon Amir, Daniel Ben-Dayan Rubin, Filipp Akopyan, Emmett McQuinn, William P. Risk, and Dharmendra S. Modha. 2013. Cognitive computing building block: A versatile and efficient digital neuron model for neurosynaptic cores. In *2013 International Joint Conference on Neural Networks (IJCNN)*. 1–10. DOI: https://doi.org/10.1109/IJCNN.2013.6707077

[20] Matheus Cavalcante, Fabian Schuiki, Florian Zaruba, Michael Schaffner, and Luca Benini. 2020. Ara: A 1-GHz+ scalable and energy-efficient RISC-V vector processor with multiprecision floating-point support in 22-Nm FD-SOI. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 28, 2 (Feb. 2020), 530–543. DOI: https://doi.org/10.1109/TVLSI.2019.2950087

[21] Matheus Cavalcante, Domenic Wüthrich, Matteo Perotti, Samuel Riedel, and Luca Benini. 2022. Spatz: A compact vector processing unit for high-performance and energy-efficient shared-L1 clusters. In *41st IEEE/ACM International Conference on Computer-Aided Design*. ACM, San Diego California, 1–9. DOI: https://doi.org/10.1145/3508352.3549367

[22] Lukas Cavigelli and Luca Benini. 2017. Origami: A 803-GOp/s/W convolutional network accelerator. *IEEE Transactions on Circuits and Systems for Video Technology* 27, 11 (2017), 2461–2475. DOI: https://doi.org/10.1109/TCSVT.2016.2592330

[23] Jung-Woo Chang, Keon-Woo Kang, and Suk-Ju Kang. 2020. An energy-efficient FPGA-based deconvolutional neural networks accelerator for single image super-resolution. *IEEE Transactions on Circuits and Systems for Video Technology* 30, 1 (2020), 281–295. DOI: https://doi.org/10.1109/TCSVT.2018.2888898

[24] Karam Chatha. 2021. Qualcomm®cloud Al 100 : 12TOPS/W scalable, high performance and low latency deep learning inference accelerator. In *2021 IEEE Hot Chips 33 Symposium (HCS)*. 1–19. DOI: https://doi.org/10.1109/HCS52781.2021.9567417

[25] Chen Chen, Xiaoyan Xiang, Chang Liu, Yunhai Shang, Ren Guo, Dongqi Liu, Yimin Lu, Ziyi Hao, Jiahui Luo, Zhijian Chen, Chunqiang Li, Yu Pu, Jianyi Meng, Xiaolang Yan, Yuan Xie, and Xiaoning Qi. 2020. Xuantie-910: A commercial multi-core 12-stage pipeline out-of-order 64-bit high performance RISC-V processor with vector extension : Industrial product. In *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*. 52–64. DOI: https://doi.org/10.1109/ISCA45697.2020.00016

[26] Gregory K. Chen, Phil C. Knag, Carlos Tokunaga, and Ram K. Krishnamurthy. 2022. An eight-core RISC-V processor with compute near last level cache in intel 4 CMOS. *IEEE Journal of Solid-State Circuits* 58, 4 (2022), 1–12. DOI: https://doi.org/10.1109/JSSC.2022.3228765

[27] Yunji Chen, Tianshi Chen, Zhiwei Xu, Ninghui Sun, and Olivier Temam. 2016. DianNao family: Energy-efficient hardware accelerators for machine learning. *Commun. ACM* 59, 11 (Oct 2016), 105–112. DOI: https://doi.org/10.1145/2996864

[28] Yiran Chen, Yuan Xie, Linghao Song, Fan Chen, and Tianqi Tang. 2020. A survey of accelerator architectures for deep neural networks. *Engineering* 6, 3 (2020), 264–274. DOI: https://doi.org/10.1016/j.eng.2020.01.007

[29] Yu-Hsin Chen, Tushar Krishna, Joel S. Emer, and Vivienne Sze. 2017. Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks. *IEEE Journal of Solid-State Circuits* 52, 1 (Jan. 2017), 127–138. DOI: https://doi.org/10.1109/JSSC.2016.2616357

[30] Yu-Hsin Chen, Tien-Ju Yang, Joel Emer, and Vivienne Sze. 2019. Eyeriss v2: A flexible accelerator for emerging deep neural networks on mobile devices. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 9, 2 (2019), 292–308. DOI: https://doi.org/10.1109/JETCAS.2019.2910232

[31] Ping Chi, Shuangchen Li, Cong Xu, Tao Zhang, Jishen Zhao, Yongpan Liu, Yu Wang, and Yuan Xie. 2016. PRIME: A novel processing-in-memory architecture for neural network computation in ReRAM-based main memory. In *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*. 27–39. DOI: https://doi.org/10.1109/ISCA.2016.13

[32] Krishna Teja Chitty-Venkata and Arun K. Somani. 2022. Neural architecture search survey: A hardware perspective. *ACM Comput. Surv.* 55, 4, Article 78 (nov 2022), 36 pages. DOI: https://doi.org/10.1145/3524500

[33] Jack Choquette. 2023. NVIDIA hopper H100 GPU: Scaling performance. *IEEE Micro* 43, 3 (2023), 1–13. DOI: https://doi.org/10.1109/MM.2023.3256796

[34] Jack Choquette, Olivier Giroux, and Denis Foley. 2018. Volta: Performance and programmability. *IEEE Micro* 38, 2 (March 2018), 42–52. DOI: https://doi.org/10.1109/MM.2018.022071134

[35] Jack Choquette, Edward Lee, Ronny Krashinsky, Vishnu Balan, and Brucek Khailany. 2021. 3.2 The A100 datacenter GPU and ampere architecture. In *2021 IEEE International Solid- State Circuits Conference (ISSCC)*, Vol. 64. 48–50. DOI: https://doi.org/10.1109/ISSCC42613.2021.9365803

[36] Marco Cococcioni, Federico Rossi, Emanuele Ruffaldi, and Sergio Saponara. 2022. A lightweight posit processing unit for RISC-V processors in deep neural network applications. *IEEE Transactions on Emerging Topics in Computing* 10, 4 (2022), 1898–1908. DOI: https://doi.org/10.1109/TETC.2021.3120538

[37] Francesco Conti, Davide Rossi, Gianna Paulin, Anaelo Garofalo, Alfio Di Mauro, Georg Rutishauer, Gian marco Ottavi, Manuel Eggimann, Hayate Okuhara, Vincent Huard, Olivier Montfort, Lionel Jure, Nils Exibard, Pascal Gouedo, Mathieu Louvat, Emmanuel Botte, and Luca Benini. 2023. 22.1 A 12.4TOPS/W @ 136GOPS AI-IoT system-on-chip with 16 RISC-V, 2-to-8b precision-scalable DNN acceleration and 30%-boost adaptive body biasing. In *2023 IEEE International Solid- State Circuits Conference (ISSCC)*. 21–23. DOI: https://doi.org/10.1109/ISSCC42615.2023.10067643

[38] Aline S. Cordeiro, Sairo R. dos Santos, Francis B. Moreira, Paulo C. Santos, Luigi Carro, and Marco A. Z. Alves. 2021. Machine learning migration for efficient near-data processing. In *2021 29th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*. 212–219. DOI: DOI:https://doi.org/10.1109/PDP52278.2021.00041

[39] D-WAVE 2023. D-Wave Systems - The Practical Quantum Computing Company. (2023). https://www.dwavesys.com/

[40] Scott Davidson, Shaolin Xie, Christopher Torng, Khalid Al-Hawai, Austin Rovinski, Tutu Ajayi, Luis Vega, Chun Zhao, Ritchie Zhao, Steve Dai, Aporva Amarnath, Bandhav Veluri, Paul Gao, Anuj Rao, Gai Liu, Rajesh K. Gupta, Zhiru Zhang, Ronald Dreslinski, Christopher Batten, and Michael Bedford Taylor. 2018. The celerity open-source 511-Core RISC-V tiered accelerator fabric: Fast architectures and design methodologies for fast chips. *IEEE Micro* 38, 2 (March 2018), 30–41. DOI: https://doi.org/10.1109/MM.2018.022071133

[41] Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, Yuyun Liao, Chit-Kwan Lin, Andrew Lines, Ruokun Liu, Deepak Mathaikutty, Steven McCoy, Arnab Paul, Jonathan Tse, Guruguhanathan Venkataramanan, Yi-Hsin Weng, Andreas Wild, Yoonseok Yang, and Hong Wang. 2018. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro* 38, 1 (2018), 82–99. DOI: https://doi.org/10.1109/MM.2018.112130359

[42] Alberto Delmas Lascorz, Patrick Judd, Dylan Malone Stuart, Zissis Poulos, Mostafa Mahmoud, Sayeh Sharify, Milos Nikolic, Kevin Siu, and Andreas Moshovos. 2019. Bit-tactical: A software/hardware approach to exploiting value and bit sparsity in neural networks. In *Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'19)*. 749–763. DOI: https://doi.org/10.1145/3297858.3304041

[43] Chunhua Deng, Siyu Liao, Yi Xie, Keshab K. Parhi, Xuehai Qian, and Bo Yuan. 2018. PermDNN: Efficient compressed DNN architecture with permuted diagonal matrices. In *51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO-51)*. 189–202. DOI: https://doi.org/10.1109/MICRO.2018.00024

[44] Quan Deng, Lei Jiang, Youtao Zhang, Minxuan Zhang, and Jun Yang. 2018. DrAcc: A DRAM based accelerator for accurate CNN inference. In *Proceedings of the 55th Annual Design Automation Conference (DAC'18)*. Association for Computing Machinery, New York, NY, USA, Article 168, 6 pages. DOI: https://doi.org/10.1145/3195970.3196029

[45] Giuseppe Desoli, Nitin Chawla, Thomas Boesch, Manui Avodhyawasi, Harsh Rawat, Hitesh Chawla, VS Abhijith, Paolo Zambotti, Akhilesh Sharma, Carmine Cappetta, Michele Rossi, Antonio De Vita, and Francesca Girardi. 2023. A 40-310TOPS/W SRAM-based all-digital up to 4b in-memory computing multi-tiled NN accelerator in FD-SOI 18nm for deep-learning edge applications. In *2023 IEEE International Solid-State Circuits Conference (ISSCC)*. 260–262. DOI: https://doi.org/10.1109/ISSCC42615.2023.10067422

[46] Giuseppe Desoli, Nitin Chawla, Thomas Boesch, Surinder-pal Singh, Elio Guidetti, Fabio De Ambroggi, Tommaso Majo, Paolo Zambotti, Manuj Ayodhyawasi, Harvinder Singh, and Nalin Aggarwal. 2017. A 2.9TOPS/W deep convolutional neural network SoC in FD-SOI 28nm for intelligent embedded systems. In *2017 IEEE International Solid-State Circuits Conference (ISSCC)*. 238–239. DOI: https://doi.org/10.1109/ISSCC.2017.7870349

[47] Pudi Dhilleswararao, Srinivas Boppu, M. Sabarimalai Manikandan, and Linga Reddy Cenkeramaddi. 2022. Efficient hardware architectures for accelerating deep neural networks: Survey. *IEEE Access* 10 (2022), 131788–131828. DOI: https://doi.org/10.1109/ACCESS.2022.3229767

[48] Alfio Di Mauro, Moritz Scherer, Davide Rossi, and Luca Benini. 2022. Kraken: A direct event/frame-based multi-sensor fusion SoC for ultra-efficient visual processing in Nano-UAVs. In *2022 IEEE Hot Chips 34 Symposium (HCS)*. 1–19. DOI: https://doi.org/10.1109/HCS55958.2022.9895621

[49] David R. Ditzel and the Esperanto team. 2022. Accelerating ML recommendation with over 1,000 RISC-V/tensor processors on esperanto's ET-SoC-1 chip. *IEEE Micro* 42, 3 (May 2022), 31–38. DOI: https://doi.org/10.1109/MM.2022.3140674

[50] Maico Cassel Dos Santos, Tianyu Jia, Joseph Zuckerman, Martin Cochet, Davide Giri, Erik Jens Loscalzo, Karthik Swaminathan, Thierry Tambe, Jeff Jun Zhang, Alper Buyuktosunoglu, Kuan-Lin Chiu, Giuseppe Di Guglielmo, Paolo Mantovani, Luca Piccolboni, Gabriele Tombesi, David Trilla, John-David Wellman, En-Yu Yang, Aporva Amarnath, Ying Jing, Bakshree Mishra, Joshua Park, Vignesh Suresh, Sarita Adve, Pradip Bose, David Brooks, Luca P. Carloni, Kenneth L. Shepard, and Gu-Yeon Wei. 2024. 14.5 A 12nm Linux-SMP-Capable RISC-V SoC with 14 accelerator types, distributed hardware power management and flexible noc-based data orchestration. In *2024 IEEE International Solid-State Circuits Conference (ISSCC)*, Vol. 67. 262–264. DOI: https://doi.org/10.1109/ISSCC49657.2024.10454572

[51] Li Du, Yuan Du, Yilei Li, and Mau-Chung Frank Chang. 2017. A reconfigurable streaming deep convolutional neural network accelerator for internet of things. *IEEE Transactions on Circuits and Systems I: Regular Papers* PP (07 2017). DOI: https://doi.org/10.1109/TCSI.2017.2735490

[52] Anne C. Elster and Tor A. Haugdahl. 2022. Nvidia hopper GPU and grace CPU highlights. *Computing in Science & Engineering* 24, 2 (March 2022), 95–100. DOI: https://doi.org/10.1109/MCSE.2022.3163817

[53] William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research* 23, 120 (2022), 1–39. Retrieved from http://jmlr.org/papers/v23/21-0998.html

[54] Tim Finkbeiner, Glen Hush, Troy Larsen, Perry Lea, John Leidel, and Troy Manning. 2017. In-memory intelligence. *IEEE Micro* 37, 4 (2017), 30–38. DOI: https://doi.org/10.1109/MM.2017.3211117

[55] Jeremy Fowers, Kalin Ovtcharov, Michael Papamichael, Todd Massengill, Ming Liu, Daniel Lo, Shlomi Alkalay, Michael Haselman, Logan Adams, Mahdi Ghandi, Stephen Heil, Prerak Patel, Adam Sapek, Gabriel Weisz, Lisa Woods, Sitaram Lanka, Steven K. Reinhardt, Adrian M. Caulfield, Eric S. Chung, and Doug Burger. 2018. A configurable cloud-scale DNN processor for real-time AI. In *ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*. 1–14.

[56] Charlotte Frenkel, David Bol, and Giacomo Indiveri. 2021. Bottom-up and top-down neural processing systems design: Neuromorphic intelligence as the convergence of natural and artificial intelligence. *CoRR* abs/2106.01288 (2021).

[57] Charlotte Frenkel, Martin Lefebvre, Jean-Didier Legat, and David Bol. 2019. A 0.086-mm^2 12.7-pJ/SOP 64k-Synapse 256-neuron online-learning digital spiking neuromorphic processor in 28-nm CMOS. *IEEE Transactions on Biomedical Circuits and Systems* 13, 1 (2019), 145–158. DOI: https://doi.org/10.1109/TBCAS.2018.2880425

[58] Charlotte Frenkel, Jean-Didier Legat, and David Bol. 2019. MorphIC: A 65-nm 738k-Synapse/mm^2 quad-core binary-weight digital neuromorphic processor with stochastic spike-driven online learning. *IEEE Transactions on Biomedical Circuits and Systems* 13, 5 (2019), 999–1010. DOI: https://doi.org/10.1109/TBCAS.2019.2928793

[59] Manuel Le Gallo, Riduan Khaddam-Aljameh, Milos Stanisavljevic, Athanasios Vasilopoulos, Benedikt Kersting, Martino Dazzi, Geethan Karunaratne, Matthias Braendli, Abhairaj Singh, Silvia M Mueller, et al. 2022. A 64-core mixed-signal in-memory compute chip based on phase-change memory for deep neural network inference. arXiv:2212.02872. Retrieved from https://arxiv.org/abs/2212.02872

[60] Fei Gao, Ting-Jung Chang, Ang Li, Marcelo Orenes-Vera, Davide Giri, Paul J. Jackson, August Ning, Georgios Tziantzioulis, Joseph Zuckerman, Jinzheng Tu, Kaifeng Xu, Grigory Chirkov, Gabriele Tombesi, Jonathan Balkind, Margaret Martonosi, Luca Carloni, and David Wentzlaff. 2023. DECADES: A 67mm2, 1.46TOPS, 55 giga cache-coherent 64-Bit RISC-V instructions per second, heterogeneous manycore SoC with 109 Tiles Including Accelerators, Intelligent Storage, and eFPGA in 12nm FinFET. In *2023 IEEE Custom Integrated Circuits Conference (CICC)*. 1–2. DOI: https://doi.org/10.1109/CICC57935.2023.10121257

[61] Jianhua Gao, Weixing Ji, Fangli Chang, Shiyu Han, Bingxin Wei, Zeming Liu, and Yizhuo Wang. 2023. A systematic survey of general sparse matrix-matrix multiplication. *ACM Comput. Surv.* 55, 12, Article 244 (2023), 36 pages. DOI: https://doi.org/10.1145/3571157

[62] Mingyu Gao, Jing Pu, Xuan Yang, Mark Horowitz, and Christos Kozyrakis. 2017. TETRIS: Scalable and efficient neural network acceleration with 3D memory. *SIGARCH Comput. Archit. News* 45, 1 (2017), 751–764. DOI: https://doi.org/10.1145/3093337.3037702

[63] Mingyu Gao, Xuan Yang, Jing Pu, Mark Horowitz, and Christos Kozyrakis. 2019. TANGRAM: Optimized coarse-grained dataflow for scalable NN accelerators. In *Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'19)*. Association for Computing Machinery, New York, NY, USA, 807–820. DOI: https://doi.org/10.1145/3297858.3304014

[64] Angelo Garofalo, Gianmarco Ottavi, Francesco Conti, Geethan Karunaratne, Irem Boybat, Luca Benini, and Davide Rossi. 2022. A heterogeneous in-memory computing cluster for flexible end-to-end inference of real-world deep neural networks. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 12, 2 (June 2022), 422–435. DOI: https://doi.org/10.1109/JETCAS.2022.3170152

[65] Angelo Garofalo, Yvan Tortorella, Matteo Perotti, Luca Valente, Alessandro Nadalini, Luca Benini, Davide Rossi, and Francesco Conti. 2022. DARKSIDE: A heterogeneous RISC-V compute cluster for extreme-edge on-chip DNN inference and training. *IEEE Open Journal of the Solid-State Circuits Society* 2 (2022), 231–243. DOI: https://doi.org/10.1109/OJSSCS.2022.3210082

[66] Hasan Genc, Seah Kim, Alon Amid, Ameer Haj-Ali, Vighnesh Iyer, Pranav Prakash, Jerry Zhao, Daniel Grubb, Harrison Liew, Howard Mao, Albert Ou, Colin Schmidt, Samuel Steffl, John Wright, Ion Stoica, Jonathan Ragan-Kelley, Krste Asanovic, Borivoje Nikolic, and Yakun Sophia Shao. 2021. Gemmini: Enabling systematic deep-learning architecture evaluation via full-stack integration. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*. 769–774. DOI: https://doi.org/10.1109/DAC18074.2021.9586216

[67] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W. Mahoney, and Kurt Keutzer. 2021. A Survey of Quantization Methods for Efficient Neural Network Inference. (2021). arXiv:2103.13630. Retrieved from https://arxiv.org/abs/2103.13630

[68] Davide Giri, Kuan-Lin Chiu, Giuseppe Di Guglielmo, Paolo Mantovani, and Luca P. Carloni. 2020. ESP4ML: Platform-based design of systems-on-chip for embedded machine learning. In *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. 1049–1054. DOI: https://doi.org/10.23919/DATE48585.2020.9116317

[69] Davide Giri, Kuan-Lin Chiu, Guy Eichler, Paolo Mantovani, and Luca P. Carloni. 2021. Accelerator integration for open-source soc design. *IEEE Micro* 41, 4 (July 2021), 8–14. DOI: https://doi.org/10.1109/MM.2021.3073893

[70] Graham Gobieski, Oguz Atli, Cagri Erbagci, Ken Mai, Nathan Beckmann, and Brandon Lucia. 2023. MANIC: A 19\mu\mathrmW @ 4MHz, 256 MOPS/mW, RISC-V microcontroller with embedded MRAM main memory and vector-dataflow co-processor in 22nm Bulk finFET CMOS. In *2023 IEEE International Symposium on Circuits and Systems (ISCAS)*. 1–4. DOI: https://doi.org/10.1109/ISCAS46773.2023.10181809

[71] Ashish Gondimalla, Noah Chesnut, Mithuna Thottethodi, and T. N. Vijaykumar. 2019. SparTen: A sparse tensor accelerator for convolutional neural networks. In *52nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'52)*. 151–165. DOI: https://doi.org/10.1145/3352460.3358291

[72] Abraham Gonzalez, Jerry Zhao, Ben Korpan, Hasan Genc, Colin Schmidt, John Wright, Ayan Biswas, Alon Amid, Farhana Sheikh, Anton Sorokin, Sirisha Kale, Mani Yalamanchi, Ramya Yarlagadda, Mark Flannigan, Larry Abramowitz, Elad Alon, Yakun Sophia Shao, Krste Asanovic, and Borivoje Nikolic. 2021. A 16mm ^{2} 106.1 GOPS/W heterogeneous RISC-V multi-core multi-accelerator soc in low-power 22nm FinFET. In *ESSCIRC 2021 - IEEE 47th European Solid State Circuits Conference (ESSCIRC)*. IEEE, Grenoble, France, 259–262. DOI: https://doi.org/10.1109/ESSCIRC53450.2021.9567768

[73] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

[74] GreenWaves Technologies GAP9 Processor. 2023. Retrieved from https://greenwaves-technologies.com/gap9_processor/. (2023). Accessed: 2023-04-18.

[75] Cong Guo, Bo Yang Hsueh, Jingwen Leng, Yuxian Qiu, Yue Guan, Zehuan Wang, Xiaoying Jia, Xipeng Li, Minyi Guo, and Yuhao Zhu. 2020. Accelerating sparse DNN models without hardware-support via tile-wise sparsity. In *International Conference for High Performance Computing, Networking, Storage and Analysis (SC'20)*. Article 16, 15 pages.

[76] K. Guo, W. Li, K. Zhong, Z. Zhu, S. Zeng, S. Han, Y. Xie, P. Debacker, M. Verhelst, and Y. Wang. 2023. Neural Network Accelerator Comparison. (2023). Retrieved May 9, 2024 from https://nicsefc.ee.tsinghua.edu.cn/project.html

[77] Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. 2015. Deep learning with limited numerical precision. In *32nd International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Vol. 37. 1737–1746.

[78] Eric Guthmuller, César Fuguet, Andrea Bocco, Jérôme Fereyre, Riccardo Alidori, Ihsane Tahir, and Yves Durand. 2024. Xvpfloat: RISC-V ISA extension for variable extended precision floating point computation. *IEEE Trans. Comput.* 73, 7 (July 2024), 1683–1697. DOI: https://doi.org/10.1109/TC.2024.3383964

[79] Laszlo Gyongyosi and Sandor Imre. 2019. A survey on quantum computing technology. *Computer Science Review* 31 (2019), 51–71. DOI: https://doi.org/10.1016/j.cosrev.2018.11.002

[80] Pascal Alexander Hager, Bert Moons, Stefan Cosemans, Ioannis A. Papistas, Bram Rooseleer, Jeroen Van Loon, Roel Uytterhoeven, Florian Zaruba, Spyridoula Koumousi, Milos Stanisavljevic, Stefan Mach, Sebastiaan Mutsaards, Riduan Khaddam Aljameh, Gua Hao Khov, Brecht Machiels, Cristian Olar, Anastasios Psarras, Sander Geursen, Jeroen Vermeeren, Yi Lu, Abhishek Maringanti, Deepak Ameta, Leonidas Katselas, Noah Hütter, Manuel Schmuck, Swetha Sivadas, Karishma Sharma, Manuel Oliveira, Ramon Aerne, Nitish Sharma, Timir Soni, Beatrice Bussolino, Djordje Pesut, Michele Pallaro, Andrei Podlesnii, Alexios Lyrakis, Yannick Ruiner, Martino Dazzi, Johannes Thiele, Koen Goetschalckx, Nazareno Bruschi, Jonas Doevenspeck, Bram Verhoef, Stefan Linz, Giuseppe Garcea, Jonathan Ferguson, Ioannis Koltsidas, and Evangelos Eleftheriou. 2024. 11.3 Metis AIPU: A 12nm 15TOPS/W 209.6TOPS SoC for Cost- and energy-efficient inference at the edge. In *2024 IEEE International Solid-State Circuits Conference (ISSCC)*, Vol. 67. 212–214. DOI: https://doi.org/10.1109/ISSCC49657.2024.10454395

[81] Nastaran Hajinazar, Geraldo F. Oliveira, Sven Gregorio, João Dinis Ferreira, Nika Mansouri Ghiasi, Minesh Patel, Mohammed Alser, Saugata Ghose, Juan Gómez-Luna, and Onur Mutlu. 2021. SIMDRAM: A framework for bit-serial SIMD processing using DRAM. In *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'21)*. Association for Computing Machinery, New York, NY, USA, 329–345. DOI: https://doi.org/10.1145/3445814.3446749

[82] Song Han, Xingyu Liu, Huizi Mao, Jing Pu, Ardavan Pedram, Mark A. Horowitz, and William J. Dally. 2016. EIE: Efficient inference engine on compressed deep neural network. In *43rd International Symposium on Computer Architecture (ISCA'16)*. 243–254. DOI: https://doi.org/10.1109/ISCA.2016.30

[83] Mehdi Hassanpour, Marc Riera, and Antonio González. 2022. A survey of near-data processing architectures for neural networks. *Machine Learning and Knowledge Extraction* 4, 1 (2022), 66–102. DOI: https://doi.org/10.3390/make4010004

[84] Mingxuan He, Choungki Song, Ilkon Kim, Chunseok Jeong, Seho Kim, Il Park, Mithuna Thottethodi, and T. N. Vijaykumar. 2020. Newton: A DRAM-maker's accelerator-in-memory (AiM) architecture for machine learning. In *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. 372–385. DOI: https://doi.org/10.1109/MICRO50266.2020.00040

[85] Kartik Hegde, Jiyong Yu, Rohit Agrawal, Mengjia Yan, Michael Pellauer, and Christopher W. Fletcher. 2018. UCNN: Exploiting computational reuse in deep neural networks via weight repetition. In *45th Annual International Symposium on Computer Architecture (ISCA'18)*. 674–687. DOI: https://doi.org/10.1109/ISCA.2018.00062

[86] Pouya Houshmand, Giuseppe M. Sarda, Vikram Jain, Kodai Ueyoshi, Ioannis A. Papistas, Man Shi, Qilin Zheng, Debjyoti Bhattacharjee, Arindam Mallik, Peter Debacker, Diederik Verkest, and Marian Verhelst. 2023. DIANA: An end-to-end hybrid digital and ANAlog neural network soc for the edge. *IEEE Journal of Solid-State Circuits* 58, 1 (Jan. 2023), 203–215. DOI: https://doi.org/10.1109/JSSC.2022.3214064

[87] HPCWIRE 2022. Quantum computers emerging as accelerators in HPC. (2022). https://www.hpcwire.com/2022/06/07/quantum-computers-emerging-as-accelerators-in-hpc/

[88] Intel. Lunar Lake processor specifications. (n.d.). Retrieved Jun 29, 2024 from https://download.intel.com/newsroom/2024/client-computing/Lunar-Lake-Architecture-Fact-Sheet.pd

[89] Intel. 2022. Intel Arc A770 Graphics 16GB. (Jul 2022). Retrieved May 25, 2023 from https://ark.intel.com/content/www/us/en/ark/products/229151/intel-arc-a770-graphics-16gb.html

[90] Vikram Jain, Sebastian Giraldo, Jaro De Roose, Linyan Mei, Bert Boons, and Marian Verhelst. 2023. TinyVers: A tiny versatile system-on-chip with state-retentive eMRAM for ML inference at the extreme edge. *IEEE Journal of Solid-State Circuits* (2023), 1–12. DOI: https://doi.org/10.1109/JSSC.2023.3236566

[91] Tianyu Jia, Paolo Mantovani, Maico Cassel Dos Santos, Davide Giri, Joseph Zuckerman, Erik Jens Loscalzo, Martin Cochet, Karthik Swaminathan, Gabriele Tombesi, Jeff Jun Zhang, Nandhini Chandramoorthy, John-David Wellman, Kevin Tien, Luca Carloni, Kenneth Shepard, David Brooks, Gu-Yeon Wei, and Pradip Bose. 2022. A 12nm agile-designed soc for swarm-based perception with heterogeneous IP blocks, a reconfigurable memory hierarchy, and an 800MHz multi-plane NoC. In *ESSCIRC 2022- IEEE 48th European Solid State Circuits Conference (ESSCIRC)*. 269–272. DOI: https://doi.org/10.1109/ESSCIRC55480.2022.9911456

[92] Zhe Jia, Blake Tillman, Marco Maggioni, and Daniele Paolo Scarpazza. 2019. Dissecting the Graphcore IPU Architecture via Microbenchmarking. (Dec. 2019). arXiv:1912.03413. DOI: https://doi.org/10.48550/arXiv.1912.03413

[93] Qiang Jiao, Wei Hu, Fang Liu, and Yong Dong. 2021. RISC-VTF: RISC-V based extended instruction set for transformer. In *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 1565–1570. DOI: https://doi.org/10.1109/SMC52423.2021.9658643

[94] Yang Jiao, Liang Han, Rong Jin, Yi-Jung Su, Chiente Ho, Li Yin, Yun Li, Long Chen, Zhen Chen, Lu Liu, Zhuyu He, Yu Yan, Jun He, Jun Mao, Xiaotao Zai, Xuejun Wu, Yongquan Zhou, Mingqiu Gu, Guocai Zhu, Rong Zhong, Wenyuan Lee, Ping Chen, Yiping Chen, Weiliang Li, Deyu Xiao, Qing Yan, Mingyuan Zhuang, Jiejun Chen, Yun Tian, Yingzi Lin, Wei Wu, Hao Li, and Zesheng Dou. 2020. A 12nm programmable convolution-efficient neural-processing-unit chip achieving 825TOPS. In *2020 IEEE International Solid- State Circuits Conference - (ISSCC)*. 136–140. DOI: https://doi.org/10.1109/ISSCC19947.2020.9062984

[95] Qing Jin, Jian Ren, Richard Zhuang, Sumant Hanumante, Zhengang Li, Zhiyu Chen, Yanzhi Wang, Kaiyuan Yang, and Sergey Tulyakov. 2022. F8Net: Fixed-point 8-bit only multiplication for network quantization. In *International Conference on Learning Representations*. Retrieved from https://openreview.net/forum?id=_CfpJazzXT2

[96] Norman Jouppi, Al Borchers, Rick Boyle, Pierre-luc Cantin, Clifford Chao, Chris Clark, Jeremy Coriell, Mike Daley, Matt Dau, Jeffrey Dean, Ben Gelb, Cliff Young, Tara Ghaemmaghami, Rajendra Gottipati, William Gulland, Robert Hagmann, C. Ho, Doug Hogberg, John Hu, and Nan Boden. 2017. In-datacenter performance analysis of a tensor processing unit. In *44th Annual International Symposium on Computer Architecture*. Association for Computing Machinery, 1–12. DOI: https://doi.org/10.1145/3079856.3080246

[97] Norm Jouppi, George Kurian, Sheng Li, Peter Ma, Rahul Nagarajan, Lifeng Nai, Nishant Patil, Suvinay Subramanian, Andy Swing, Brian Towles, Clifford Young, Xiang Zhou, Zongwei Zhou, and David A Patterson. 2023. TPU v4: An optically reconfigurable supercomputer for machine learning with hardware support for embeddings. In *50th Annual International Symposium on Computer Architecture*. 14. DOI: https://doi.org/10.1145/3579371.3589350

[98] Norman Jouppi, Cliff Young, Nishant Patil, and David Patterson. 2018. Motivation for and evaluation of the first tensor processing unit. *IEEE Micro* 38, 3 (May 2018), 10–19. DOI: https://doi.org/10.1109/MM.2018.032271057

[99] Norman P. Jouppi, Doe Hyun Yoon, George Kurian, Sheng Li, Nishant Patil, James Laudon, Cliff Young, and David Patterson. 2020. A domain-specific supercomputer for training deep neural networks. *Commun. ACM* 63, 7 (June 2020), 67–78. DOI: https://doi.org/10.1145/3360307

[100] Yuhao Ju and Jie Gu. 2023. A systolic neural CPU processor combining deep learning and general-purpose computing with enhanced data locality and end-to-end performance. *IEEE Journal of Solid-State Circuits* 58, 1 (Jan. 2023), 216–226. DOI: https://doi.org/10.1109/JSSC.2022.3214170

[101] Riduan Khaddam-Aljameh, Milos Stanisavljevic, Jordi Fornt Mas, Geethan Karunaratne, Matthias Brändli, Feng Liu, Abhairaj Singh, Silvia M Müller, Urs Egger, Anastasios Petropoulos, Theodore Antonakopoulos, Kevin Brew, Samuel Choi, Injo Ok, Fee Li Lie, Nicole Saulnier, Victor Chan, Ishtiaq Ahsan, Vijay Narayanan, S. R. Nandakumar, Manuel Le Gallo, Pier Andrea Francese, Abu Sebastian, and Evangelos Eleftheriou. 2022. HERMES-core–A 1.59-TOPS/mm 2 PCM on 14-nm CMOS in-memory compute core using 300-ps/LSB linearized CCO-based ADCs. *IEEE Journal of Solid-State Circuits* 57, 4 (2022), 1027–1038.

[102] Mahmoud Khairy, Amr G. Wassal, and Mohamed Zahran. 2019. A survey of architectural approaches for improving GPGPU performance, programmability and heterogeneity. *J. Parallel and Distrib. Comput.* 127 (2019), 65–88. DOI: https://doi.org/10.1016/j.jpdc.2018.11.012

[103] Duckhwan Kim, Jaeha Kung, Sek Chai, Sudhakar Yalamanchili, and Saibal Mukhopadhyay. 2016. Neurocube: A programmable digital neuromorphic architecture with high-density 3D memory. In *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*. 380–392. DOI: https://doi.org/10.1109/ISCA.2016.41

[104] Donghyuk Kim, Chengshuo Yu, Shanshan Xie, Yuzong Chen, Joo-Young Kim, Bongjin Kim, Jaydeep P. Kulkarni, and Tony Tae-Hyoung Kim. 2022. An overview of processing-in-memory circuits for artificial intelligence and machine learning. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 12, 2 (2022), 338–353. DOI: https://doi.org/10.1109/JETCAS.2022.3160455

[105] Sehoon Kim, Coleman Hooper, Thanakul Wattanawong, Minwoo Kang, Ruohan Yan, Hasan Genc, Grace Dinh, Qijing Huang, Kurt Keutzer, Michael W. Mahoney, Yakun Sophia Shao, and Amir Gholami. 2023. Full Stack Optimization of Transformer Inference: a Survey. (2023). arXiv:cs.CL/2302.14017 https://arxiv.org/abs/2302.14017

[106] Sangyeob Kim, Sangjin Kim, Soyeon Um, Soyeon Kim, Kwantae Kim, and Hoi-Jun Yoo. 2022. Neuro-CIM: A 310.4 TOPS/W neuromorphic computing-in-memory processor with low WL/BL activity and digital-analog mixed-mode neuron firing. In *2022 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*. 38–39. DOI: https://doi.org/10.1109/VLSITechnologyandCir46769.2022.9830276

[107] James C. Knight and Thomas Nowotny. 2018. GPUs outperform current hpc and neuromorphic solutions in terms of speed and energy when simulating a highly-connected cortical model. *Frontiers in Neuroscience* 12 (2018). DOI: https://doi.org/10.3389/fnins.2018.00941

[108] Simon Knowles. 2021. Graphcore. In *2021 IEEE Hot Chips 33 Symposium (HCS)*. 1–25. DOI: https://doi.org/10.1109/HCS52781.2021.9567075

[109] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 1097–1105. Retrieved from http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[110] Youngsu Kwon, Jinho Han, Yongcheol Peter Cho, Juyeob Kim, Jaehoon Chung, Jaewoong Choi, Sujin Park, Igyeong Kim, Hyunjeong Kwon, Jinkyu Kim, Hyunmi Kim, Won Jeon, Youngdeuk Jeon, Minhyung Cho, and Minseok Choi. 2023. Chiplet heterogeneous-integration AI processor. In *2023 International Conference on Electronics, Information, and Communication (ICEIC)*.

[111] Young-Cheon Kwon, Suk Han Lee, Jaehoon Lee, Sang-Hyuk Kwon, Je Min Ryu, Jong-Pil Son, O Seongil, Hak-Soo Yu, Haesuk Lee, Soo Young Kim, Youngmin Cho, Jin Guk Kim, Jongyoon Choi, Hyun-Sung Shin, Jin Kim, BengSeng Phuah, HyoungMin Kim, Myeong Jun Song, Ahn Choi, Daeho Kim, SooYoung Kim, Eun-Bong Kim, David Wang, Shinhaeng Kang, Yuhwan Ro, Seungwoo Seo, JoonHo Song, Jaeyoun Youn, Kyomin Sohn, and Nam Sung Kim. 2021. 25.4 A 20nm 6GB function-in-memory DRAM, based on HBM2 with a 1.2TFLOPS programmable computing unit using bank-level parallelism, for machine learning applications. In *2021 IEEE International Solid-State Circuits Conference (ISSCC)*, Vol. 64. 350–352. DOI: https://doi.org/10.1109/ISSCC42613.2021.9365862

[112] Jingjing Lan, Vishnu P. Nambiar, Rheeshaalaen Sabapathy, Mihai Dragos Rotaru, and Anh Tuan Do. 2021. Chiplet-based architecture design for multi-core neuromorphic processor. In *2021 IEEE 23rd Electronics Packaging Technology Conference (EPTC)*. 410–412. DOI: https://doi.org/10.1109/EPTC53413.2021.9663898

[113] Cristóbal Ramírez Lazo, Enrico Reggiani, Carlos Rojas Morales, Roger Figueras Bagué, Luis A. Villa Vargas, Marco A. Ramírez Salinas, Mateo Valero Cortés, Osman Sabri Ünsal, and Adrián Cristal. 2022. Adaptable register file organization for vector processors. In *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. 786–799. DOI: https://doi.org/10.1109/HPCA53966.2022.00063

[114] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436.

[115] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324. DOI: https://doi.org/10.1109/5.726791

[116] Sae Kyu Lee, Ankur Agrawal, Joel Silberman, Matthew Ziegler, Mingu Kang, Swagath Venkataramani, Nianzheng Cao, Bruce Fleischer, Michael Guillorn, Matthew Cohen, Silvia M. Mueller, Jinwook Oh, Martin Lutz, Jinwook

Jung, Siyu Koswatta, Ching Zhou, Vidhi Zalani, Monodeep Kar, James Bonanno, Robert Casatuta, Chia-Yu Chen, Jungwook Choi, Howard Haynie, Alyssa Herbert, Radhika Jain, Kyu-Hyoun Kim, Yulong Li, Zhibin Ren, Scot Rider, Marcel Schaal, Kerstin Schelm, Michael R. Scheuermann, Xiao Sun, Hung Tran, Naigang Wang, Wei Wang, Xin Zhang, Vinay Shah, Brian Curran, Vijayalakshmi Srinivasan, Pong-Fei Lu, Sunil Shukla, Kailash Gopalakrishnan, and Leland Chang. 2022. A 7-nm four-core mixed-precision AI Chip With 26.2-TFLOPS Hybrid-FP8 Training, 104.9-TOPS INT4 inference, and workload-aware throttling. *IEEE Journal of Solid-State Circuits* 57, 1 (2022), 182–197. DOI: https://doi.org/10.1109/JSSC.2021.3120113

[117] Sang Min Lee, Hanjoon Kim, Jeseung Yeon, Juyun Lee, Younggeun Choi, Minho Kim, Changjae Park, Kiseok Jang, Youngsik Kim, Yongseung Kim, Changman Lee, Hyuck Han, Won Eung Kim, Rui Tang, and Joon Ho Baek. 2022. A 64-TOPS energy-efficient tensor accelerator in 14nm With reconfigurable fetch network and processing fusion for maximal data reuse. *IEEE Open Journal of the Solid-State Circuits Society* 2 (2022), 219–230. DOI: https://doi.org/10.1109/OJSSCS.2022.3216798

[118] Yunsup Lee, Andrew Waterman, Rimas Avizienis, Henry Cook, Chen Sun, Vladimir Stojanović, and Krste Asanović. 2014. A 45nm 1.3 GHz 16.7 double-precision GFLOPS/W RISC-V processor with vector accelerators. In *ESSCIRC 2014-40th European Solid State Circuits Conference (ESSCIRC)*. IEEE, 199–202.

[119] Ang Li, Ting-Jung Chang, Fei Gao, Tuan Ta, Georgios Tziantzioulis, Yanghui Ou, Moyang Wang, Jinzheng Tu, Kaifeng Xu, Paul Jackson, August Ning, Grigory Chirkov, Marcelo Orenes-Vera, Shady Agwa, Xiaoyu Yan, Eric Tang, Jonathan Balkind, Christopher Batten, and David Wentzlaff. 2023. CIFER: A cache-coherent 12-Nm 16-Mm2 SoC With Four 64-Bit RISC-V Application Cores, 18 32-Bit RISC-V Compute Cores, and a 1541 LUT6/Mm2 Synthesizable eFPGA. *IEEE Solid-State Circuits Letters* 6 (2023), 229–232. DOI: https://doi.org/10.1109/LSSC.2023.3303111

[120] Gang Li, Zejian Liu, Fanrong Li, and Jian Cheng. 2021. Block convolution: Towards memory-efficient inference of large-scale CNNs on FPGA. *CoRR* abs/2105.08937 (2021). arXiv:2105.08937 https://arxiv.org/abs/2105.08937

[121] Jiajun Li, Shuhao Jiang, Shijun Gong, Jingya Wu, Junchao Yan, Guihai Yan, and Xiaowei Li. 2019. SqueezeFlow: A sparse CNN accelerator exploiting concise convolution rules. *IEEE Trans. Comput.* 68, 11 (2019), 1663–1677. DOI: https://doi.org/10.1109/TC.2019.2924215

[122] Shuangchen Li, Dimin Niu, Krishna T. Malladi, Hongzhong Zheng, Bob Brennan, and Yuan Xie. 2017. DRISA: A DRAM-based reconfigurable in-situ accelerator. In *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO-50'17)*. Association for Computing Machinery, New York, NY, USA, 288–301. DOI: https://doi.org/10.1145/3123939.3123977

[123] Yuan Li, Ahmed Louri, and Avinash Karanth. 2021. SPRINT: A high-performance, energy-efficient, and scalable chiplet-based accelerator with photonic interconnects for CNN inference. *IEEE Transactions on Parallel and Distributed Systems* 33, 10 (2021), 2332–2345.

[124] Chien-Hung Lin, Chih-Chung Cheng, Yi-Min Tsai, Sheng-Je Hung, Yu-Ting Kuo, Perry H Wang, Pei-Kuei Tsung, Jeng-Yun Hsu, Wei-Chih Lai, Chia-Hung Liu, Shao-Yu Wang, Chin-Hua Kuo, Chih-Yu Chang, Ming-Hsien Lee, Tsung-Yao Lin, and Chih-Cheng Chen. 2020. A 3.4-to-13.3TOPS/W 3.6TOPS dual-core deep-learning accelerator for versatile AI applications in 7nm 5G smartphone SoC. In *2020 IEEE International Solid- State Circuits Conference - (ISSCC)*. 134–136. DOI: https://doi.org/10.1109/ISSCC19947.2020.9063111

[125] Mu-Shan Lin, Tze-Chiang Huang, Chien-Chun Tsai, King-Ho Tam, Cheng-Hsiang Hsieh, Tom Chen, Wen-Hung Huang, Jack Hu, Yu-Chi Chen, Sandeep Kumar Goel, Chin-Ming Fu, Stefan Rusu, Chao-Chieh Li, Sheng-Yao Yang, Mei Wong, Shu-Chun Yang, and Frank Lee. 2019. A 7nm 4GHz Arm-core-based CoWoS chiplet design for high performance computing. In *2019 Symposium on VLSI Circuits*.

[126] Chen Liu, Guillaume Bellec, Bernhard Vogginger, David Kappel, Johannes Partzsch, Felix Neumärker, Sebastian Höppner, Wolfgang Maass, Steve B. Furber, Robert Legenstein, and Christian G. Mayr. 2018. Memory-efficient deep learning on a SpiNNaker 2 prototype. *Frontiers in Neuroscience* 12 (2018), 840. DOI: https://doi.org/10.3389/fnins.2018.00840

[127] L. Liu, J. Zhu, Z. Li, Y. Lu, Y. Deng, J. Han, S. Yin, and S. Wei. 2019. A survey of coarse-grained reconfigurable architecture and design: Taxonomy, challenges, and applications. *ACM Comput. Surv.* 52, 6 (2019), 1–39. DOI: https://doi.org/10.1145/3357375

[128] Xiaoxiao Liu, Mengjie Mao, Beiye Liu, Hai Li, Yiran Chen, Boxun Li, Yu Wang, Hao Jiang, Mark Barnell, Qing Wu, and Jianhua Yang. 2015. RENO: A high-efficient reconfigurable neuromorphic computing accelerator design. In *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*. 1–6. DOI: https://doi.org/10.1145/2744769.2744900

[129] Zechun Liu, Kwang-Ting Cheng, Dong Huang, Eric Xing, and Zhiqiang Shen. 2022. Nonuniform-to-uniform quantization: Towards accurate quantization via generalized straight-through estimation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 4932–4942. DOI: https://doi.org/10.1109/CVPR52688.2022.00489

[130] Yufei Ma, Yu Cao, Sarma Vrudhula, and Jae-sun Seo. 2018. Optimizing the convolution operation to accelerate deep neural networks on FPGA. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 26, 7 (2018), 1354–1367. DOI: https://doi.org/10.1109/TVLSI.2018.2815603

[131] Raju Machupalli, Masum Hossain, and Mrinal Mandal. 2022. Review of ASIC accelerators for deep neural network. *Microprocessors and Microsystems* 89 (2022), 104441. DOI: https://doi.org/10.1016/j.micpro.2022.104441

[132] David Mallasén, Raul Murillo, Alberto A. Del Barrio, Guillermo Botella, Luis Piñuel, and Manuel Prieto-Matias. 2022. PERCIVAL: Open-source posit RISC-V core with quire capability. *IEEE Transactions on Emerging Topics in Computing* 10, 3 (2022), 1241–1252. DOI: https://doi.org/10.1109/TETC.2022.3187199

[133] P. Y. Martinez, Y. Beilliard, M. Godard, D. Danovitch, D. Drouin, J. Charbonnier, P. Coudrain, A. Garnier, D. Lattard, P. Vivet, S. Cheramy, E. Guthmuller, C. Fuguet Tortolero, V. Mengue, J. Durupt, A. Philippe, and D. Dutoit. 2020. ExaNoDe: Combined integration of chiplets on active interposer with bare dice in a multi-chip-module for heterogeneous and scalable high performance compute nodes. In *2020 IEEE Symposium on VLSI Technology*.

[134] Peter Mattson, Christine Cheng, Gregory Diamos, Cody Coleman, Paulius Micikevicius, David Patterson, Hanlin Tang, Gu-Yeon Wei, Peter Bailis, Victor Bittorf, et al. 2020. Mlperf training benchmark. *Proceedings of Machine Learning and Systems* 2 (2020), 336–349.

[135] Peter Mattson, Vijay Janapa Reddi, Christine Cheng, Cody Coleman, Greg Diamos, David Kanter, Paulius Micikevicius, David Patterson, Guenther Schmuelling, Hanlin Tang, Gu-Yeon Wei, and Carole-Jean Wu. 2020. MLPerf: An industry standard benchmark suite for machine learning performance. *IEEE Micro* 40, 2 (2020), 8–16. DOI: https://doi.org/10.1109/MM.2020.2974843

[136] Eitan Medina. 2019. [Habana labs presentation]. In *2019 IEEE Hot Chips 31 Symposium (HCS)*. 1–29. DOI: https://doi.org/10.1109/HOTCHIPS.2019.8875670

[137] Chuhan Min, Jiachen Mao, Hai Li, and Yiran Chen. 2019. NeuralHMC: An efficient HMC-based accelerator for deep neural networks. In *24th Asia and South Pacific Design Automation Conference*. ACM, 394–399. DOI: https://doi.org/10.1145/3287624.3287642

[138] Francesco Minervini, Oscar Palomar, Osman Unsal, Enrico Reggiani, Josue Quiroga, Joan Marimon, Carlos Rojas, Roger Figueras, Abraham Ruiz, Alberto Gonzalez, Jonnatan Mendoza, Ivan Vargas, César Hernandez, Joan Cabre, Lina Khoirunisya, Mustapha Bouhali, Julian Pavon, Francesc Moll, Mauro Olivieri, Mario Kovac, Mate Kovac, Leon Dragic, Mateo Valero, and Adrian Cristal. 2023. Vitruvius+: An area-efficient RISC-V decoupled vector coprocessor for high performance computing applications. *ACM Transactions on Architecture and Code Optimization* 20, 2 (March 2023), 28:1–28:25. DOI: https://doi.org/10.1145/3575861

[139] Ivan Miro-Panades, Benoit Tain, Jean-Frédéric Christmann, David Coriat, Romain Lemaire, Clément Jany, Baudouin Martineau, Fabrice Chaix, Guillaume Waltener, Emmanuel Pluchart, Jean-Philippe Noel, Adam Makosiej, Maxime Montoya, Simone Bacles-Min, David Briand, Jean-Marc Philippe, Yvain Thonnart, Alexandre Valentian, Frédéric Heitzmann, and Fabien Clermidy. 2022. SamurAI: A versatile IoT node with event-driven wake-up and embedded ML acceleration. *IEEE Journal of Solid-State Circuits* 58, 6 (2022), 1–0. DOI: https://doi.org/10.1109/JSSC.2022.3198505

[140] Asit K. Mishra, Jorge Albericio Latorre, Jeff Pool, Darko Stosic, Dusan Stosic, Ganesh Venkatesh, Chong Yu, and Paulius Micikevicius. 2021. Accelerating sparse deep neural networks. arXiv:2104.08378. Retrieved from https://arxiv.org/abs/2104.08378

[141] Gabriel Mounce, Jim Lyke, Stephen Horan, Wes Powell, Rich Doyle, and Rafi Some. 2016. Chiplet based approach for heterogeneous processing and packaging architectures. In *2016 IEEE Aerospace Conference*. 1–12.

[142] Francisco Muñoz Martínez, Raveesh Garg, Michael Pellauer, José L. Abellán, Manuel E. Acacio, and Tushar Krishna. 2023. Flexagon: A multi-dataflow sparse-sparse matrix multiplication accelerator for efficient DNN processing. In *28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3 (ASPLOS 2023)*. 252–265. DOI: https://doi.org/10.1145/3582016.3582069

[143] Alessandro Nadalini, Georg Rutishauser, Alessio Burrello, Nazareno Bruschi, Angelo Garofalo, Luca Benini, Francesco Conti, and Davide Rossi. 2023. A 3 TOPS/W RISC-V Parallel cluster for inference of fine-grain mixed-precision quantized neural networks. In *2023 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. 1–6. DOI: https://doi.org/10.1109/ISVLSI59464.2023.10238679

[144] Mitchell A. Nahmias, Thomas Ferreira de Lima, Alexander N. Tait, Hsuan-Tung Peng, Bhavin J. Shastri, and Paul R. Prucnal. 2020. Photonic multiply-accumulate operations for neural networks. *IEEE Journal of Selected Topics in Quantum Electronics* 26, 1 (2020), 1–18. DOI: https://doi.org/10.1109/JSTQE.2019.2941485

[145] P. Narayanan, S. Ambrogio, A. Okazaki, K. Hosokawa, H. Tsai, A. Nomura, T. Yasuda, C. Mackin, S. C. Lewis, A. Friz, M. Ishii, Y. Kohda, H. Mori, K. Spoon, R. Khaddam-Aljameh, N. Saulnier, M. Bergendahl, J. Demarest, K. W. Brew, V. Chan, S. Choi, I. Ok, I. Ahsan, F. L. Lie, W. Haensch, V. Narayanan, and G. W. Burr. 2021. Fully on-chip MAC at 14 nm enabled by accurate row-wise programming of PCM-based weights and parallel vector-transport in duration-format. *IEEE Transactions on Electron Devices* 68, 12 (2021), 6629–6636. DOI: https://doi.org/10.1109/TED.2021.3115993

[146] Eriko Nurvitadhi, Dongup Kwon, Ali Jafari, Andrew Boutros, Jaewoong Sim, Phillip Tomson, Huseyin Sumbul, Gregory Chen, Phil Knag, Raghavan Kumar, Ram Krishnamurthy, Sergey Gribok, Bogdan Pasca, Martin Langhammer, Debbie Marr, and Aravind Dasu. 2019. Why compete when you can work together: FPGA-ASIC integration for persistent

RNNs. In *2019 IEEE 27th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*. 199–207.

[147] Jinwook Oh, Sae Kyu Lee, Mingu Kang, Matthew Ziegler, Joel Silberman, Ankur Agrawal, Swagath Venkataramani, Bruce Fleischer, Michael Guillorn, Jungwook Choi, Wei Wang, Silvia Mueller, Shimon Ben-Yehuda, James Bonanno, Nianzheng Cao, Robert Casatuta, Chia-Yu Chen, Matt Cohen, Ophir Erez, Thomas Fox, George Gristede, Howard Haynie, Vicktoria Ivanov, Siyu Koswatta, Shih-Hsien Lo, Martin Lutz, Gary Maier, Alex Mesh, Yevgeny Nustov, Scot Rider Marcel Schaal, Michael Scheuermann, Xiao Sun, Naigang Wang, Fanchieh Yee, Ching Zhou, Vinay Shah, Brian Curran, Vijayalakshmi Srinivasan, Pong-Fei Lu, Sunil Shukla, Kailash Gopalakrishnan, and Leland Chang. 2020. A 3.0 TFLOPS 0.62V scalable processor core for high compute utilization AI training and inference. In *2020 IEEE Symposium on VLSI Circuits*. 1–2. DOI: https://doi.org/10.1109/VLSICircuits18222.2020.9162917

[148] Gianmarco Ottavi, Angelo Garofalo, Giuseppe Tagliavini, Francesco Conti, Alfio Di Mauro, Luca Benini, and Davide Rossi. 2023. Dustin: A 16-cores parallel ultra-low-power cluster with 2b-to-32b fully flexible bit-precision and vector lockstep execution mode. *IEEE Transactions on Circuits and Systems I: Regular Papers* (2023), 1–14. DOI: https://doi.org/10.1109/TCSI.2023.3254810

[149] Eustace Painkras, Luis A. Plana, Jim Garside, Steve Temple, Francesco Galluppi, Cameron Patterson, David R. Lester, Andrew D. Brown, and Steve B. Furber. 2013. SpiNNaker: A 1-W 18-core system-on-chip for massively-parallel neural network simulation. *IEEE Journal of Solid-State Circuits* 48, 8 (2013), 1943–1953. DOI: https://doi.org/10.1109/JSSC.2013.2259038

[150] Angshuman Parashar, Minsoo Rhu, Anurag Mukkara, Antonio Puglielli, Rangharajan Venkatesan, Brucek Khailany, Joel Emer, Stephen W. Keckler, and William J. Dally. 2017. SCNN: An accelerator for compressed-sparse convolutional neural networks. In *Proceedings of the 2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA'17)*. 27–40. DOI: https://doi.org/10.1145/3079856.3080254

[151] Jun-Seok Park, Jun-Woo Jang, Heonsoo Lee, Dongwoo Lee, Sehwan Lee, Hanwoong Jung, Seungwon Lee, Suknam Kwon, Kyungah Jeong, Joon-Ho Song, SukHwan Lim, and Inyup Kang. 2021. A 6K-MAC feature-map-sparsity-aware neural processing unit in 5nm flagship mobile soc. In *2021 IEEE International Solid- State Circuits Conference (ISSCC)*, Vol. 64. 152–154. DOI: https://doi.org/10.1109/ISSCC42613.2021.9365928

[152] Jun-Seok Park, Changsoo Park, Suknam Kwon, Hyeong-Seok Kim, Taeho Jeon, Yesung Kang, Heonsoo Lee, Dongwoo Lee, James Kim, YoungJong Lee, Sangkyu Park, Jun-Woo Jang, SangHyuck Ha, MinSeong Kim, Jihoon Bang, Suk Hwan Lim, and Inyup Kang. 2022. A multi-mode 8K-MAC HW-utilization-aware neural processing unit with a unified multi-precision datapath in 4nm flagship mobile soc. In *2022 IEEE International Solid-State Circuits Conference (ISSCC)*, Vol. 65. 246–248. DOI: https://doi.org/10.1109/ISSCC42614.2022.9731639

[153] Seong-Wook Park, Junyoung Park, Kyeongryeol Bong, Dongjoo Shin, Jinmook Lee, Sungpill Choi, and Hoi-Jun Yoo. 2015. An energy-efficient and scalable deep learning/inference processor with tetra-parallel MIMD architecture for big data applications. *IEEE Transactions on Biomedical Circuits and Systems* 9, 6 (2015), 838–848. DOI: https://doi.org/10.1109/TBCAS.2015.2504563

[154] Gianna Paulin, Renzo Andri, Francesco Conti, and Luca Benini. 2021. RNN-based radio resource management on multicore risc-v accelerator architectures. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 29, 9 (Sept. 2021), 1624–1637. DOI: https://doi.org/10.1109/TVLSI.2021.3093242

[155] Gianna Paulin, Paul Scheffler, Thomas Benz, Matheus Cavalcante, Tim Fischer, Manuel Eggimann, Yichao Zhang, Nils Wistoff, Luca Bertaccini, Luca Colagrande, Gianmarco Ottavi, Frank K. Gürkaynak, Davide Rossi, and Luca Benini. 2024. Occamy: A 432-Core 28.1 DP-GFLOP/s/W 83% FPU Utilization Dual-Chiplet, Dual-HBM2E RISC-V-Based accelerator for stencil and sparse linear algebra computations with 8-to-64-bit floating-point support in 12nm FinFET. In *2024 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*. 12. DOI: https://doi.org/10.1109/VLSITechnologyandCir46783.2024.10631529

[156] Maurice Peemen, Arnaud A. A. Setio, Bart Mesman, and Henk Corporaal. 2013. Memory-centric accelerator design for convolutional neural networks. In *2013 IEEE 31st International Conference on Computer Design (ICCD)*. 13–19. DOI: https://doi.org/10.1109/ICCD.2013.6657019

[157] Matteo Perotti, Matheus Cavalcante, Alessandro Ottaviano, Jiantao Liu, and Luca Benini. 2023. Yun: An open-source, 64-Bit RISC-V-based vector processor with multi-precision integer and floating-point support in 65-Nm CMOS. *IEEE Transactions on Circuits and Systems II: Express Briefs* 70, 10 (Oct. 2023), 3732–3736. DOI: https://doi.org/10.1109/TCSII.2023.3292579

[158] Matteo Perotti, Matheus Cavalcante, Nils Wistoff, Renzo Andri, Lukas Cavigelli, and Luca Benini. 2022. A "New Ara" for vector computing: An open source highly efficient RISC-V V 1.0 vector processor design. In *2022 IEEE 33rd International Conference on Application-specific Systems, Architectures and Processors (ASAP)*. 43–51. DOI: https://doi.org/10.1109/ASAP54787.2022.00017

[159] Stefania Perri, Cristian Sestito, Fanny Spagnolo, and Pasquale Corsonello. 2020. Efficient deconvolution architecture for heterogeneous systems-on-chip. *Journal of Imaging* 6, 9 (2020), 85. DOI: https://doi.org/10.3390/jimaging6090085

[160] Kartik Prabhu, Albert Gural, Zainab F Khan, Robert M Radway, Massimo Giordano, Kalhan Koul, Rohan Doshi, John W Kustin, Timothy Liu, Gregorio B. Lopes, Victor Turbiner, Win-San Khwa, Yu-Der Chih, Meng-Fan Chang, Guénolé Lallement, Boris Murmann, Subhasish Mitra, and Priyanka Raina. 2022. CHIMERA: A 0.92-TOPS, 2.2-TOPS/W edge AI accelerator with 2-MByte on-chip foundry resistive RAM for efficient training and inference. *IEEE Journal of Solid-State Circuits* 57, 4 (2022), 1013–1026.

[161] Arpan Prasad, Luca Benini, and Francesco Conti. 2023. Specialization meets flexibility: A heterogeneous architecture for high-efficiency, high-flexibility AR/VR processing. In *Proceedings of the 2023 Design Automation Conference (DAC 2023), to Appear.*

[162] Arpan Suravi Prasad, Moritz Scherer, Francesco Conti, Davide Rossi, Alfio Di Mauro, Manuel Eggimann, Jorge Tomás Gómez, Ziyun Li, Syed Shakib Sarwar, Zhao Wang, Barbara De Salvo, and Luca Benini. 2024. Siracusa: A 16 Nm heterogenous RISC-V SoC for extended reality with At-MRAM neural engine. *IEEE Journal of Solid-State Circuits* (2024), 1–15. DOI: https://doi.org/10.1109/JSSC.2024.3385987

[163] Eric Qin, Ananda Samajdar, Hyoukjun Kwon, Vineet Nadella, Sudarshan Srinivasan, Dipankar Das, Bharat Kaul, and Tushar Krishna. 2020. SIGMA: A sparse and irregular GEMM accelerator with flexible interconnects for DNN training. In *2020 IEEE International Symposium on High Performance Computer Architecture (HPCA'20)*. 58–70. DOI: https://doi.org/10.1109/HPCA47549.2020.00015

[164] Haotong Qin, Ruihao Gong, Xianglong Liu, Xiao Bai, Jingkuan Song, and Nicu Sebe. 2020. Binary neural networks: A survey. *Pattern Recognition* 105 (2020), 107281. DOI: https://doi.org/10.1016/j.patcog.2020.107281

[165] QISKIT 2023. IBM Qiskit Simulator. (2023). https://qiskit.org/

[166] Jiantao Qiu, Jie Wang, Song Yao, Kaiyuan Guo, Boxun Li, Erjin Zhou, Jincheng Yu, Tianqi Tang, Ningyi Xu, Sen Song, Yu Wang, and Huazhong Yang. 2016. Going deeper with embedded FPGA platform for convolutional neural network. *2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays* (2016).

[167] Atul Rahman, Sangyun Oh, Jongeun Lee, and Kiyoung Choi. 2017. Design space exploration of FPGA accelerators for convolutional neural networks. In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*. 1147–1152. DOI: https://doi.org/10.23919/DATE.2017.7927162

[168] Nitin Rathi, Indranil Chakraborty, Adarsh Kosta, Abhronil Sengupta, Aayush Ankit, Priyadarshini Panda, and Kaushik Roy. 2023. Exploring neuromorphic computing based on spiking neural networks: Algorithms to hardware. *ACM Comput. Surv.* 55, 12 (2023), 1–49. DOI: https://doi.org/10.1145/3571155

[169] IBM Research. A new chip architecture points to faster, more energy-efficient AI. (n.d.). Retrieved from https://research.ibm.com/blog/northpole-ibm-ai-chi

[170] Albert Reuther, Peter Michaleas, Michael Jones, Vijay Gadepally, Siddharth Samsi, and Jeremy Kepner. 2022. AI and ML accelerator survey and trends. In *2022 IEEE High Performance Extreme Computing Conference (HPEC)*. 1–10. DOI: https://doi.org/10.1109/HPEC55821.2022.9926331

[171] F. Rosenblatt. 1957. *The perceptron - A perceiving and recognizing automaton*. Technical Report 85-460-1. Cornell Aeronautical Laboratory, Ithaca, New York.

[172] Davide Rossi, Francesco Conti, Manuel Eggiman, Alfio Di Mauro, Giuseppe Tagliavini, Stefan Mach, Marco Guermandi, Antonio Pullini, Igor Loi, Jie Chen, Eric Flamand, and Luca Benini. 2022. Vega: A ten-core soc for IoT endnodes with DNN acceleration and cognitive wake-up from MRAM-based state-retentive sleep mode. *IEEE Journal of Solid-State Circuits* 57, 1 (Jan. 2022), 127–139. DOI: https://doi.org/10.1109/JSSC.2021.3114881

[173] Sourjya Roy, Mustafa Ali, and Anand Raghunathan. 2021. PIM-DRAM: Accelerating machine learning workloads using processing in commodity DRAM. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 11, 4 (2021), 701–710. DOI: https://doi.org/10.1109/JETCAS.2021.3127517

[174] Murugan Sankaradas, Venkata Jakkula, Srihari Cadambi, Srimat Chakradhar, Igor Durdanovic, Eric Cosatto, and Hans Peter Graf. 2009. A massively parallel coprocessor for convolutional neural networks. In *20th IEEE International Conference on Application-Specific Systems, Architectures and Processors*. IEEE Computer Society, USA, 53–60. DOI: https://doi.org/10.1109/ASAP.2009.25

[175] Jürgen Schmidhuber. 2015. Deep learning in neural networks: An overview. *Neural Networks* 61 (2015), 85–117. DOI: https://doi.org/10.1016/j.neunet.2014.09.003

[176] Jae-sun Seo, Bernard Brezzo, Yong Liu, Benjamin D. Parker, Steven K. Esser, Robert K. Montoye, Bipin Rajendran, José A. Tierno, Leland Chang, Dharmendra S. Modha, and Daniel J. Friedman. 2011. A 45nm CMOS neuromorphic chip with a scalable architecture for learning in networks of spiking neurons. In *2011 IEEE Custom Integrated Circuits Conference (CICC)*. 1–4. DOI: https://doi.org/10.1109/CICC.2011.6055293

[177] Vivek Seshadri, Yoongu Kim, Chris Fallin, Donghyuk Lee, Rachata Ausavarungnirun, Gennady Pekhimenko, Yixin Luo, Onur Mutlu, Phillip B. Gibbons, Michael A. Kozuch, and Todd C. Mowry. 2013. RowClone: Fast and energy-efficient in-DRAM bulk data copy and initialization. In *2013 46th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. 185–197.

[178] Vivek Seshadri, Donghyuk Lee, Thomas Mullins, Hasan Hassan, Amirali Boroumand, Jeremie Kim, Michael A. Kozuch, Onur Mutlu, Phillip B. Gibbons, and Todd C. Mowry. 2017. Ambit: In-memory accelerator for bulk bitwise operations using commodity DRAM technology. In *2017 50th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. 273–287.

[179] Cristian Sestito, Fanny Spagnolo, and Stefania Perri. 2021. Design of flexible hardware accelerators for image convolutions and transposed convolutions. *Journal of Imaging* 7, 10 (2021), 210. Retrieved from https://www.mdpi.com/2313-433X/7/10/210

[180] Ali Shafiee, Anirban Nag, Naveen Muralimanohar, Rajeev Balasubramonian, John Paul Strachan, Miao Hu, R. Stanley Williams, and Vivek Srikumar. 2016. ISAAC: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars. In *43rd International Symposium on Computer Architecture*. 14–26. DOI: https://doi.org/10.1109/ISCA.2016.12

[181] Junnan Shan, Mario R. Casu, Jordi Cortadella, Luciano Lavagno, and Mihai T. Lazarescu. 2019. Exact and heuristic allocation of muiti-kernel applications to multi-FPGA platforms. In *2019 56th ACM/IEEE Design Automation Conference (DAC)*. 1–6.

[182] Yakun Sophia Shao, Jason Clemons, Rangharajan Venkatesan, Brian Zimmer, Matthew Fojtik, Nan Jiang, Ben Keller, Alicia Klinefelter, Nathaniel Pinckney, Priyanka Raina, Stephen G. Tell, Yanqing Zhang, William J. Dally, Joel Emer, C. Thomas Gray, Brucek Khailany, and Stephen W. Keckler. 2019. Simba: Scaling deep-learning inference with multi-chip-module-based architecture. In *52nd Annual IEEE/ACM International Symposium on Microarchitecture*. Association for Computing Machinery, 14–27. DOI: https://doi.org/10.1145/3352460.3358302

[183] Harsh Sharma, Sumit K Mandal, Janardhan Rao Doppa, Umit Y Ogras, and Partha Pratim Pande. 2022. SWAP: A server-scale communication-aware chiplet-based manycore PIM accelerator. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 41, 11 (2022), 4145–4156.

[184] Kamilya Smagulova, Mohammed E. Fouda, Fadi Kurdahi, Khaled N. Salama, and Ahmed Eltawil. 2023. Resistive neural hardware accelerators. *Proc. IEEE* 111, 5 (2023), 500–527. DOI: https://doi.org/10.1109/JPROC.2023.3268092

[185] Alan Smith, Eric Chapman, Chintan Patel, Raja Swaminathan, John Wuu, Tyrone Huang, Wonjun Jung, Alexander Kaganov, Hugh McIntyre, and Ramon Mangaser. 2024. 11.1 AMD InstinctTM MI300 series modular chiplet package – HPC and AI accelerator for exa-class systems. In *2024 IEEE International Solid-State Circuits Conference (ISSCC)*, Vol. 67. 490–492. DOI: https://doi.org/10.1109/ISSCC49657.2024.10454441

[186] Martin Snelgrove and Robert Beachler. 2023. speedAI240: A 2-Petaflop, 30-Teraflops/W At-memory inference acceleration device with 1456 RISC-V cores. *IEEE Micro* 43, 3 (May 2023), 58–63. DOI: https://doi.org/10.1109/MM.2023.3255864

[187] Jinook Song, Yunkyo Cho, Jun-Seok Park, Jun-Woo Jang, Sehwan Lee, Joon-Ho Song, Jae-Gon Lee, and Inyup Kang. 2019. An 11.5TOPS/W 1024-MAC butterfly structure dual-core sparsity-aware neural processing unit in 8nm flagship mobile soc. In *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*. 130–132. DOI: https://doi.org/10.1109/ISSCC.2019.8662476

[188] Linghao Song, Xuehai Qian, Hai Li, and Yiran Chen. 2017. PipeLayer: A pipelined ReRAM-based accelerator for deep learning. In *2017 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. 541–552. DOI: https://doi.org/10.1109/HPCA.2017.55

[189] Fanny Spagnolo, Stefania Perri, and Pasquale Corsonello. 2020. Design of a real-time face detection architecture for heterogeneous systems-on-chips. *Integration* 74 (2020), 1–10. DOI: https://doi.org/10.1016/j.vlsi.2020.04.008

[190] F. Spagnolo, S. Perri, and P. Corsonello. 2022. Aggressive approximation of the SoftMax function for power-efficient hardware implementations. *IEEE Transactions on Circuits and Systems II: Express Briefs* 69, 3 (2022), 1652–1656. DOI: https://doi.org/10.1109/TCSII.2021.3120495

[191] F. Spagnolo, S. Perri, and P. Corsonello. 2022. Approximate down-sampling strategy for power-constrained intelligent systems. *IEEE Access* 10 (2022), 7073–7081. DOI: https://doi.org/10.1109/ACCESS.2022.3142292

[192] Vinay Sriram, David Cox, Kuen Tsoi, and Wayne Luk. 2011. Towards an embedded biologically-inspired machine vision processor. In *2010 International Conference on Field-Programmable Technology*. 273–278. DOI: https://doi.org/10.1109/FPT.2010.5681487

[193] V. Sze, Y. H. Chen, T. J. Yang, and J. S. Emer. 2017. Efficient processing of deep neural networks: A tutorial and survey. *Proc. IEEE* 105, 12 (2017), 2295–2329. DOI: https://doi.org/10.1109/JPROC.2017.2761740

[194] Emil Talpes, Debjit Das Sarma, Doug Williams, Sahil Arora, Thomas Kunjan, Benjamin Floering, Ankit Jalote, Christopher Hsiong, Chandrasekhar Poorna, Vaidehi Samant, John Sicilia, Anantha Kumar Nivarti, Raghuvir Ramachandran, Tim Fischer, Ben Herzberg, Bill McGee, Ganesh Venkataramanan, and Pete Banon. 2023. The microarchitecture of DOJO, Tesla's exa-scale computer. *IEEE Micro* 43, 3 (2023), 31–39. DOI: https://doi.org/10.1109/MM.2023.3258906

[195] Thierry Tambe, Jeff Zhang, Coleman Hooper, Tianyu Jia, Paul N. Whatmough, Joseph Zuckerman, Maico Cassel Dos Santos, Erik Jens Loscalzo, Davide Giri, Kenneth Shepard, Luca Carloni, Alexander Rush, David Brooks, and Gu-Yeon Wei. 2023. 22.9 A 12nm 18.1TFLOPs/W sparse transformer processor with entropy-based early exit, mixed-precision predication and fine-grained power management. In *2023 IEEE International Solid- State Circuits Conference (ISSCC)*. IEEE, San Francisco, CA, USA, 342–344. DOI: https://doi.org/10.1109/ISSCC42615.2023.10067817

[196] Wei Tang, Sung-Gun Cho, Tim Tri Hoang, Jacob Botimer, Wei Qiang Zhu, Ching-Chi Chang, Cheng-Hsun Lu, Junkang Zhu, Yaoyu Tao, Tianyu Wei, Naomi Kavi Motwani, Mani Yalamanchi, Ramya Yarlagadda, Sirisha Rani Kale, Mark Flanigan, Allen Chan, Thungoc Tran, Sergey Shumarayev, and Zhengya Zhang. 2024. Arvon: A heterogeneous system-in-package integrating FPGA and DSP chiplets for versatile workload acceleration. *IEEE Journal of Solid-State Circuits* 59, 4 (2024), 1235–1245. DOI: https://doi.org/10.1109/JSSC.2023.3343457

[197] Wenkai Tang and Peiyong Zhang. 2022. GPGCN: A general-purpose graph convolution neural network accelerator based on RISC-V ISA extension. *Electronics* 11, 22 (2022). DOI: https://doi.org/10.3390/electronics11223833

[198] Yvan Tortorella, Luca Bertaccini, Luca Benini, Davide Rossi, and Francesco Conti. 2023. RedMule: A mixed-precision matrix-matrix operation engine for flexible and energy-efficient on-chip linear algebra and TinyML training acceleration. *CoRR* abs/2301.03904, arXiv:2301.03904 (Jan. 2023). DOI: https://doi.org/10.48550/arXiv.2301.03904 arXiv:arXiv:2301.03904

[199] Yvan Tortorella, Luca Bertaccini, Davide Rossi, Luca Benini, and Francesco Conti. 2022. RedMulE: A compact FP16 matrix-multiplication accelerator for adaptive deep learning on RISC-V-based ultra-low-power SoCs. In *2022 Conference & Exhibition on Design, Automation & Test in Europe*. European Design and Automation Association, 1099–1102.

[200] Yaman Umuroglu, Nicholas J. Fraser, Giulio Gambardella, Michaela Blott, Philip Leong, Magnus Jahre, and Kees Vissers. 2017. FINN: A framework for fast, scalable binarized neural network inference. In *2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. ACM, 65–74. DOI: https://doi.org/10.1145/3020078.3021744

[201] Paramita Basak Upama, Md Jobair Hossain Faruk, Mohammad Nazim, Mohammad Masum, Hossain Shahriar, Gias Uddin, Shabir Barzanjeh, Sheikh Iqbal Ahamed, and Akond Rahman. 2022. Evolution of Quantum Computing: A Systematic Survey on the Use of Quantum Computing Tools. (2022). arXiv:cs.SE/2204.01856

[202] Luca Valente, Alessandro Nadalini, Asif Hussain Chiralil Veeran, Mattia Sinigaglia, Bruno Sá, Nils Wistoff, Yvan Tortorella, Simone Benatti, Rafail Psiakis, Ari Kulmala, Baker Mohammad, Sandro Pinto, Daniele Palossi, Luca Benini, and Davide Rossi. 2024. A heterogeneous RISC-V based SoC for secure nano-UAV navigation. *IEEE Transactions on Circuits and Systems I: Regular Papers* 71, 5 (May 2024), 2266–2279. DOI: https://doi.org/10.1109/TCSI.2024.3359044

[203] Jasmina Vasiljevic, Ljubisa Bajic, Davor Capalija, Stanislav Sokorac, Dragoljub Ignjatovic, Lejla Bajic, Milos Trajkovic, Ivan Hamer, Ivan Matosevic, Aleksandar Cejkov, Utku Aydonat, Tony Zhou, Syed Zohaib Gilani, Armond Paiva, Joseph Chu, Djordje Maksimovic, Stephen Alexander Chin, Zahi Moudallal, Akhmed Rakhmati, Sean Nijjar, Almeet Bhullar, Boris Drazic, Charles Lee, James Sun, Kei-Ming Kwong, James Connolly, Miles Dooley, Hassan Farooq, Joy Yu Ting Chen
Matthew Walker, Keivan Dabiri, Kyle Mabee, Rakesh Shaji Lal, Namal Rajatheva, Renjith Retnamma, Shripad Karodi, Daniel Rosen, Emilio Munoz, Andrew Lewycky, Aleksandar Knezevic, Raymond Kim, Allan Rui, Alexander Drouillard, and David Thompson. 2021. Compute substrate for software 2.0. *IEEE Micro* 41, 2 (March 2021), 50–55. DOI: https://doi.org/10.1109/MM.2021.3061912

[204] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 5998–6008.

[205] Stylianos I. Venieris and Christos-Savvas Bouganis. 2019. fpgaConvNet: Mapping regular and irregular convolutional neural networks on FPGAs. *IEEE Transactions on Neural Networks and Learning Systems* 30, 2 (2019), 326–342. DOI: https://doi.org/10.1109/TNNLS.2018.2844093

[206] Swagath Venkataramani, Vijayalakshmi Srinivasan, Wei Wang, Sanchari Sen, Jintao Zhang, Ankur Agrawal, Monodeep Kar, Shubham Jain, Alberto Mannari, Hoang Tran, Yulong Li, Eri Ogawa, Kazuaki Ishizaki, Hiroshi Inoue, Marcel Schaal, Mauricio Serrano, Jungwook Choi, Xiao Sun, Naigang Wang, Chia-Yu Chen, Allison Allain, James Bonano, Nianzheng Cao, Robert Casatuta, Matthew Cohen, Bruce Fleischer, Michael Guillorn, Howard Haynie, Jinwook Jung, Mingu Kang, Kyu-hyoun Kim, Siyu Koswatta, Saekyu Lee, Martin Lutz, Silvia Mueller, Jinwook Oh, Ashish Ranjan, Zhibin Ren, Scot Rider, Kerstin Schelm, Michael Scheuermann, Joel Silberman, Jie Yang, Vidhi Zalani, Xin Zhang, Ching Zhou, Matt Ziegler, Vinay Shah, Moriyoshi Ohara, Pong-Fei Lu, Brian Curran, Sunil Shukla, Leland Chang, and Kailash Gopalakrishnan. 2021. RaPiD: AI accelerator for ultra-low precision training and inference. In *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*. 153–166. DOI: https://doi.org/10.1109/ISCA52012.2021.00021

[207] Ventana Micro. 2023. Retrieved from https://www.ventanamicro.com/. (2023). Accessed: 2023-04-18.

[208] Marian Verhelst, Man Shi, and Linyan Mei. 2022. ML processors are going multi-core: A performance dream or a scheduling nightmare? *IEEE Solid-State Circuits Magazine* 14, 4 (2022), 18–27.

[209] Thiruvengadam Vijayaraghavan, Yasuko Eckert, Gabriel H. Loh, Michael J. Schulte, Mike Ignatowski, Bradford M. Beckmann, William C. Brantley, Joseph L. Greathouse, Wei Huang, Arun Karunanithi, Onur Kayiran, Mitesh Meswani, Indrani Paul, Matthew Poremba, Steven Raasch, Steven K. Reinhardt, Greg Sadowski, and Vilas Sridharan. 2017. Design and analysis of an APU for exascale computing. In *2017 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. 85–96.

[210] Pascal Vivet, Eric Guthmuller, Yvain Thonnart, Gael Pillonnet, César Fuguet, Ivan Miro-Panades, Guillaume Moritz, Jean Durupt, Christian Bernard, Didier Varreau, Julian Pontes, Sébastien Thuries, David Coriat, Michel Harrand, Denis Dutoit, Didier Lattard, Lucile Arnaud, Jean Charbonnier, Perceval Coudrain, Arnaud Garnier, Frédéric Berger, Alain Gueugnot, Alain Greiner, Quentin L. Meunier, Alexis Farcy, Alexandre Arriordaz, Séverine Chéramy, and Fabien Clermidy. 2021. IntAct: A 96-core processor with six chiplets 3D-stacked on an active interposer with distributed interconnects and integrated power management. *IEEE Journal of Solid-State Circuits* 56, 1 (2021), 79–97.

[211] Weier Wan, Rajkumar Kubendran, Clemens Schaefer, Sukru Burc Eryilmaz, Wenqiang Zhang, Dabin Wu, Stephen Deiss, Priyanka Raina, He Qian, Bin Gao, Siddharth Joshi, Huaqiang Wu, H.-S. Philip Wong, and Gert Cauwenberghs. 2022. A compute-in-memory chip based on resistive random-access memory. *Nature* 608, 7923 (01 Aug 2022), 504–512. DOI: https://doi.org/10.1038/s41586-022-04992-8

[212] Shihang Wang, Jianghan Zhu, Qi Wang, Can He, and Terry Tao Ye. 2021. Customized instruction on RISC-V for winograd-based convolution acceleration. In *2021 IEEE 32nd International Conference on Application-specific Systems, Architectures and Processors (ASAP)*. 65–68. DOI: https://doi.org/10.1109/ASAP52443.2021.00018

[213] Yizhi Wang, Jun Lin, and Zhongfeng Wang. 2019. FPAP: A folded architecture for energy-quality scalable convolutional neural networks. *IEEE Transactions on Circuits and Systems I: Regular Papers* 66 (2019), 288–301.

[214] Yipeng Wang, Mengtian Yang, Chieh-Pu Lo, and Jaydeep P. Kulkarni. 2024. 30.6 Vecim: A 289.13GOPS/W RISC-V Vector co-processor with compute-in-memory vector register file for efficient high-performance computing. In *2024 IEEE International Solid-State Circuits Conference (ISSCC)*. IEEE, San Francisco, CA, USA, 492–494. DOI: https://doi.org/10.1109/ISSCC49657.2024.10454387

[215] Sally Ward-Foxton. 2022. Axelera Demos AI Test Chip After Taping Out in Four Months. (2022).

[216] Xuechao Wei, Cody Hao Yu, Peng Zhang, Youxiang Chen, Yuxin Wang, Han Hu, Yun Liang, and Jason Cong. 2017. Automated systolic array architecture synthesis for high throughput CNN inference on FPGAs. In *2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)*. 1–6. DOI: https://doi.org/10.1145/3061639.3062207

[217] Zhou Yu Xuan, Ching-Jui Lee, and Tsung Tai Yeh. 2022. Lego: Dynamic tensor-splitting multi-tenant DNN models on multi-chip-module architecture. In *2022 19th International SoC Design Conference (ISOCC)*. 173–174. DOI: https://doi.org/10.1109/ISOCC56007.2022.10031596

[218] Cheng-Xin Xue, Wei-Hao Chen, Je-Syu Liu, Jia-Fang Li, Wei-Yu Lin, Wei-En Lin, Jing-Hong Wang, Wei-Chen Wei, Ting-Wei Chang, Tung-Cheng Chang, Tsung-Yuan Huang, Hui-Yao Kao, Shih-Ying Wei, Yen-Cheng Chiu, Chun-Ying Lee, Chung-Chuan Lo, Ya-Chin King, Chorng-Jung Lin, Ren-Shuo Liu, Chih-Cheng Hsieh, Kea-Tiong Tang, and Meng-Fan Chang. 2019. A 1Mb multibit ReRAM computing-in-memory macro with 14.6ns parallel MAC computing time for CNN based AI edge processors. In *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*. 388–390. DOI: https://doi.org/10.1109/ISSCC.2019.8662395

[219] Amir Yazdanbakhsh, Kambiz Samadi, Nam Sung Kim, Hadi Esmaeilzadeh, Hajar Falahati, and Philip J. Wolfe. 2018. GANAX: A unified MIMD-SIMD acceleration for generative adversarial networks. In *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*. 650–661. DOI: https://doi.org/10.1109/ISCA.2018.00060

[220] Wente Yi, Kefan Mo, Wenjia Wang, Yitong Zhou, Yejun Zeng, Zihan Yuan, Bojun Cheng, and Biao Pan. 2024. RDCIM: RISC-V supported full-digital computing-in-memory processor with high energy efficiency and low area overhead. *IEEE Transactions on Circuits and Systems I: Regular Papers* 71, 4 (April 2024), 1719–1732. DOI: https://doi.org/10.1109/TCSI.2024.3350664

[221] Florian Zaruba, Fabian Schuiki, and Luca Benini. 2021. Manticore: A 4096-Core RISC-V chiplet architecture for ultraefficient floating-point computing. *IEEE Micro* 41, 2 (March 2021), 36–42. DOI: https://doi.org/10.1109/MM.2020.3045564

[222] Chen Zhang, Di Wu, Jiayu Sun, Guangyu Sun, Guojie Luo, and Jason Cong. 2016. Energy-efficient CNN implementation on a deeply pipelined FPGA cluster. In *2016 International Symposium on Low Power Electronics and Design*. ACM, 326–331. DOI: https://doi.org/10.1145/2934583.2934644

[223] Jie-Fang Zhang, Ching-En Lee, Chester Liu, Yakun Sophia Shao, Stephen W. Keckler, and Zhengya Zhang. 2019. SNAP: A 1.67—21.55TOPS/W sparse neural acceleration processor for unstructured sparse deep neural network inference in 16nm CMOS. In *2019 Symposium on VLSI Circuits*. C306–C307. DOI: https://doi.org/10.23919/VLSIC.2019.8778193

[224] Shijin Zhang, Zidong Du, Lei Zhang, Huiying Lan, Shaoli Liu, Ling Li, Qi Guo, Tianshi Chen, and Yunji Chen. 2016. Cambricon-X: An accelerator for sparse neural networks. In *2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO) (MICRO-49)*. Article 20, 12 pages. DOI: https://doi.org/10.1109/MICRO.2016.7783723

[225] Yongwei Zhao, Chang Liu, Zidong Du, Qi Guo, Xing Hu, Yimin Zhuang, Zhenxing Zhang, Xinkai Song, Wei Li, Xishan Zhang, Ling Li, Zhiwei Xu, and Tianshi Chen. 2021. Cambricon-Q: A hybrid architecture for efficient training. In *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*. 706–719. DOI: https://doi.org/10.1109/ISCA52012.2021.00061

[226] Haoxiang Zhou, Haiqiao Hong, Dingbang Liu, Hang Liu, Yu Xia, Kai Li, Jun Liu, Shaobo Luo, Wei Mao, and Hao Yu. 2023. RISC-V based fully-parallel SRAM computing-in-memory accelerator with high hardware utilization and

data reuse rate. In *2023 IEEE 5th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*. 1–5. DOI: https://doi.org/10.1109/AICAS57966.2023.10168630

[227] Brian Zimmer, Rangharajan Venkatesan, Yakun Sophia Shao, Jason Clemons, Matthew Fojtik, Nan Jiang, Ben Keller, Alicia Klinefelter, Nathaniel Pinckney, Priyanka Raina, Stephen G. Tell, Yanqing Zhang, William J. Dally, Joel S. Emer, C. Thomas Gray, Stephen W. Keckler, and Brucek Khailany. 2020. A 0.32–128 TOPS, scalable multi-chip-module-based deep neural network inference accelerator with ground-referenced signaling in 16 nm. *IEEE Journal of Solid-State Circuits* 55, 4 (2020), 920–932. DOI: https://doi.org/10.1109/JSSC.2019.2960488

Received 22 June 2023; revised 4 March 2025; accepted 22 March 2025