

Multi-Partner Project: Outcomes of the ICSC Flagship 2 Project on Architectures and Design Methodologies to Accelerate AI Workloads

Cristina Silvano*, Fabrizio Ferrandi*, Serena Curzel*, Daniele Ielmini*,

Cristian Zambelli[†], Sebastiano Fabio Schifano[†],

Francesco Conti[‡], Angelo Garofalo[‡], Luca Benini[‡],

Maurizio Palesi[§], Giuseppe Ascia[§], Enrico Russo[§],

Fanny Spagnolo[¶], Pasquale Corsonello[¶], Stefania Perri[¶] and Fabio Frustaci[¶]

* Politecnico di Milano, Milano, Italy [†] Università degli Studi di Ferrara, Ferrara, Italy

[‡] Università di Bologna, Bologna, Italy [§] Università degli Studi di Catania [¶] Università della Calabria, Rende (CS), Italy

Email: *cristina.silvano@polimi.it

Abstract—Energy-efficient hardware accelerators specialized for AI tasks are now being deployed from low-power edge devices to large-scale high-performance computing systems and data centers. This paper presents the main outcomes of the Flagship 2 project of the ICSC Italian National Research Center for High Performance Computing, which focuses on the design techniques for heterogeneous hardware optimized for AI acceleration from the edge to the HPC. In particular, we describe the main challenges addressed and highlight some advances in architectures, technologies, and design methodologies tailored to accelerate deep learning, transformer-based, and generative AI models. We also summarize the most significant outcomes achieved through the close collaboration among the project partners, including the development of design techniques, tools, prototypes, IP cores, and models that collectively advance AI acceleration from the edge to the HPC contexts.

Index Terms—Heterogeneous Architectures, AI Accelerators, RISC-V Architecture, Emerging Memory Technologies, Edge Computing, High-Performance Computing.

I. INTRODUCTION

The ICSC Italian National Research Center for High Performance Computing, Big Data, and Quantum Computing is structured around a central supercomputing hub based in Bologna, supported by ten specialized research spokes, each dedicated to a specific thematic domain. Among these, Spoke 1 on Future HPC serves as the technological backbone of ICSC, focusing on the development of cutting-edge hardware and software technologies for next-generation HPC systems. Spoke 1 includes 15 Italian universities and 9 industrial partners and is organized into 5 flagship research projects. Among them, the Flagship 2 project specifically addresses the design of heterogeneous architectures to accelerate AI workloads. This initiative is driven by the rapid advancement of AI, in particular deep learning and transformer-based models, which increasingly demand energy-efficient, high-performance accelerators optimized for HPC systems [1]. Meeting these

requirements imposes a multidisciplinary approach that integrates competences on computer architecture, AI and machine learning algorithms, compiler technology, computational modeling, and approximate computing techniques. In recent years, various methodologies and tools for AI acceleration have emerged, including hardware–software co-design flows, High-Level Synthesis techniques, specialized compilers, and design space exploration frameworks for modeling and simulation. These different approaches share a common goal: maximizing computational parallelism and performance while minimizing the energy consumption. Based on these motivations, this paper presents the main research outcomes achieved by the Flagship 2 project during the past 3-year time frame.

II. TOOLCHAINS FOR DESIGN SPACE EXPLORATION AND HIGH-LEVEL SYNTHESIS FOR AI ACCELERATORS

The development of Design Space Exploration (DSE) and High-Level Synthesis (HLS) toolchains is crucial to enable designers to generate efficient reconfigurable accelerators for AI with minimal manual effort.

The toolchains developed within the ICSC Flagship 2 project allow designers to automatically explore the wide space of the architectural parameters and to adopt optimization strategies at a high level of abstraction through performance and resource estimations, and subsequently translate the desired design configuration into an efficient FPGA accelerator by the HLS phase.

The first newly developed toolchain is SPARTA [3], a methodology for the automated Synthesis of PARallel multi-Threaded Accelerators. SPARTA was integrated within the open-source tool Bambu [4] to be triggered when the input design contains OpenMP directives to parallelize part of the application. In this specialized HLS flow, parallel regions are first translated into calls to OpenMP runtime primitives by the front-end Clang compiler, and then implemented through corresponding low-level hardware components in the synthesis backend. Accelerators generated with SPARTA are based on a custom architecture that can exploit spatial parallelism and hide the latency of external memory accesses through context

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.

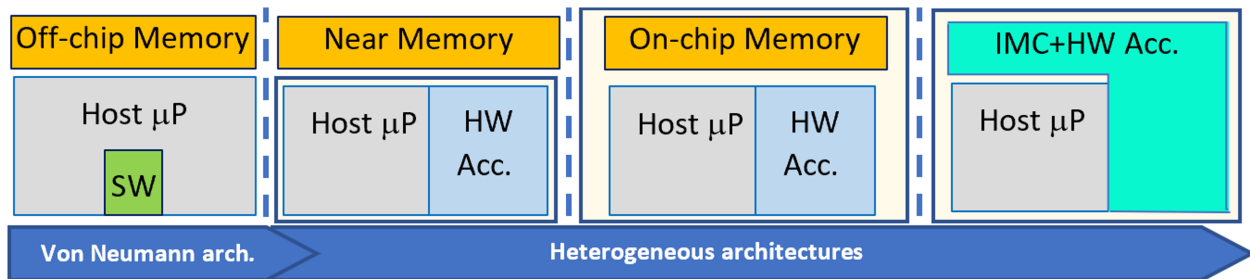


Fig. 1: The trend to merge processing and memory resources towards near-memory, on-chip memory and In-memory Computing architectures. Reprinted with permission from [2]. ©2024 IEEE.

switching. Moreover, SPARTA includes a custom Network-on-Chip connecting multiple external memory channels to each accelerator, memory-side caching, and on-chip private memories for each accelerator. SPARTA has primarily been tested on graph processing kernels to demonstrate its ability to generate efficient accelerators for irregular applications, suggesting that Graph Neural Networks could benefit from the acceleration of those kernels. However, SPARTA can be applied successfully for AI applications in general, since many high-level design frameworks for AI provide the possibility to compile the trained model into an OpenMP application.

The integration of generated accelerators in complex SoCs is made possible by selecting standard interface protocols (e.g., AXI4) in the synthesis options. Moreover, users can run complex system-level simulations where the host processor and the RTL simulation of the accelerator interact through Inter-Process Communication [5]. The required register-transfer level testbench is automatically generated, and simulated results are automatically verified against software results as well.

Another tool developed in the ICSC Flagship 2 project is LEMON [6], an open-source framework for optimally mapping tensor workloads on spatial accelerators via integer linear programming (ILP). LEMON takes as input a description of a tensor computation and the architecture of the spatial accelerator, including the detailed energy access cost and bandwidth of each memory buffer in the memory hierarchy. It then formulates and solves an ILP to determine the best combination of loop tiling, unrolling, and iteration ordering for the computation. This enables LEMON to minimize both the expected execution latency and the overall energy consumption, taking into account the unique characteristics of the accelerator’s memory system. By jointly considering data movement, memory bandwidth, and computation, LEMON enables design-time exploration of mapping strategies that are provably optimal under the given hardware constraints. This is particularly relevant for AI workloads, whose performance is often bottlenecked by memory accesses rather than raw compute. The adoption of LEMON in the DSE flow enables a more accurate and efficient mapping of tensor workloads, such as convolutions and matrix multiplications, onto reconfigurable and spatial accelerators. LEMON is available as open-source software¹.

¹<https://github.com/haimrich/lemon>

Building upon LEMON, other DSE tools have also been developed. Notably, MOHaM [7] extends the methodology to enable DSE of multi-accelerator systems, specifically tailored for multi-DNN workloads. MOHaM can leverage LEMON’s optimal mapping strategies at the individual accelerator level and can coordinate the scheduling and resource allocation across multiple accelerators, supporting the efficient deployment and hardware design for complex AI applications that require concurrent execution of several neural networks.

III. DESIGNING EFFICIENT IMC ARCHITECTURES

As schematized in Fig. 1, IMC is an emerging paradigm where, in contrast to conventional approaches, data processing is largely executed within the memory array, thus minimizing the data movement and the associated latency and energy consumption [8]. IMC is particularly efficient for data-intensive workloads, such as training and inference of AI models. Both volatile memories, such as static random access memories (SRAMs), and emerging nonvolatile memories (NVMs), such as phase change memories (PCMs) and resistive switching memories (RRAMs), have been proposed as computational memory devices supporting IMC [9]. Recently, SRAM-based digital IMC (DIMC) has been proposed with outstanding energy-efficient characteristics [2], [10].

This project focuses on the development of IMC accelerators that address fundamental design challenges at device, circuit, and architecture levels. Among the research activities, we explored RRAM-, PCM-, and SRAM-based IMC concepts as well as SRAM-based DIMC architectures.

The research activity of the ICSC Flagship 2 achieved significant results with the design of new SRAM bitcell topologies and the introduction of a new framework tool to support the design of SRAM-based DIMC architectures. The new tool is called In-Memory-computing-CACTI (IMCACTI), and it is an enhanced extension of CACTI [11]. Our approach allows integrating the computational logic, specifically an AND gate and a Full-Adder (FA), inside each bitcell of the SRAM array and provides an efficient operational modeling framework for digital SRAM-based IMC macros. Its purpose is to serve as an effective plugin for analyzing at the circuit-level abstraction within a realistic in-memory computing scenario. IMCACTI is the first tool, to the best of our knowledge, suitable to model digital SRAM-based IMC macros with embedded logic within

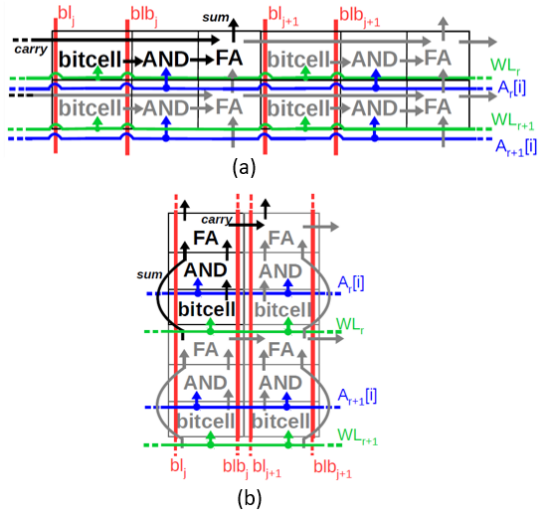


Fig. 2: Modeled layout orientations: a) horizontal; b) vertical.

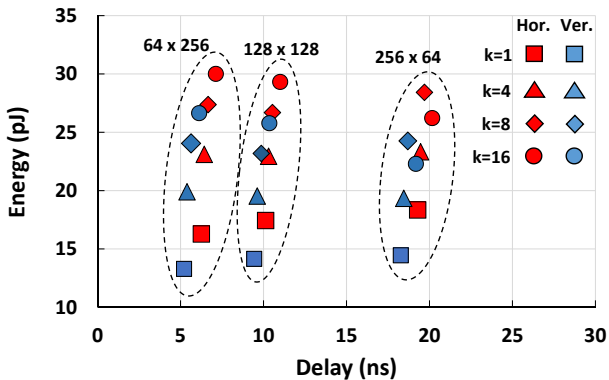


Fig. 3: Energy-delay tradeoffs for various configurations and layout strategies.

the memory array. IMCACTI evaluates area, delay, and energy metrics, and, as a further novelty with respect to state-of-the-art frameworks, it considers the layout styles depicted in Fig. 2 to efficiently support the modeling of systolic array architectures with configurable weight bit-widths k .

Figure 3 reports some exploration results collected for a 2kB memory array at various values of k for different layout styles, and various aspect ratios (128×128 , 256×64 , and 64×256). Results show that the total energy dissipation can significantly increase with k . As an example, for the 128×128 configuration with the horizontal layout, the energy increases by 68% with k varying from 1 to 16. On the contrary, the total delay varies with k only slightly. Indeed, it depends mostly on how many FAs are series-connected along multiple rows to implement the addition tree, which is independent of k . The proposed exploration approach represents a valid design support to select the most efficient configuration to meet some specific requirements. As an example, for $k = 1$ and $k = 16$, the lowest energy consumption is achieved by the 64×256 and the 256×64 configuration, respectively, with the vertical layout.

At the circuit level, a new 9T memory bitcell has been designed and characterized by using the ST 28nm 1V FDSOI

technology. Proper models have been integrated into IMCACTI to support the novel circuit topologies. Preliminary results show how to reduce the energy consumption of a DIMC architecture by approximately 30% compared to conventional 6T and 8T cell topologies, without compromising performance. The work is still ongoing, and the next main objectives are: validating the newly introduced models for other technologies, like the 14nm and 22nm; optimizing the elementary full-adder architecture, and further reducing the energy consumption of the adder trees. Upon acceptance of the paper [12], IMCACTI will be released as open-source via a publicly accessible repository.

IV. FPGA-BASED ACCELERATORS FOR APPROXIMATE COMPUTING

The proposed accelerators exploit approximate computing within critical layers typically employed in Deep Learning models, such as convolutions (CONVs), transposed convolutions (TCONVs) [13], [14], [15], [16], pooling, fully connected operations, and SoftMax function [17]. To offer flexibility, our accelerators were designed to be integrated as IPs within heterogeneous FPGA-based hardware platforms. Our recent activities have been focused on non-linear activation functions that play a crucial role in the overall achieved performance of DNN models. A proper activation function not only enables faster convergence during the training process but also helps to reduce the complexity of the model, getting the same or even better accuracy results. The Swish activation function [18] proposed by Google team as a non-monotonic and smoothed alternative to ReLU, demonstrates superior performance in deeper models and across a number of challenging datasets.

$$f_1(x) = \frac{x}{1 + e^{-x}} \quad (1)$$

However, the Swish function (1) involves complex exponentiation and division operations, which prevent its extensive use in applications constrained in computational resources and/or power budgets. To cope with this limitation, some hardware-friendly approximation strategies [19]–[21] have been successfully demonstrated in the recent past. They rely on replacing exponentiation and division units by either shifting operations [19] or piecewise linear functions [20], [21]. However, both solutions still involve product operations.

$$f_4(x) = \begin{cases} 0 & x \leq -4 \\ \frac{-x}{16} - 0.375 & -4 < x \leq -1 \\ \frac{x}{4} - 0.0375 & -1 < x \leq 0 \\ \frac{x}{2} + \frac{x}{4} - 0.0375 & 0 < x \leq 1 \\ x + \frac{x}{16} - 0.375 & 1 < x \leq 4 \\ x & x > 4 \end{cases} \quad (2)$$

To address this issue, we proposed a new multiplier-free approximation of the Swish function and its custom hardware architecture [22]. Our piecewise linear function given in 2 was devised to operate only on power-of-two factors and to increase the reuse of computational paths between different segments. As a consequence, the proposed circuit includes only four

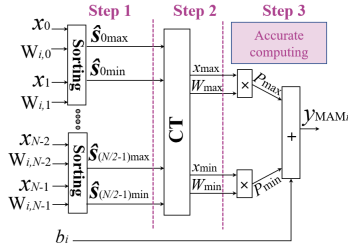


Fig. 4: Processing pipeline in the proposed HIMAM.

adders and one comparator. When implemented on a Xilinx Artix-7 FPGA chip, our 16-bit design exhibits a Power-Delay-Product (PDP) of 10.4 pJ, which is about 20 times lower than the best competitor [21]. The ASIC-based implementation carried out by using a 28 nm FDSOI technology process occupies $357.408 \mu\text{m}^2$ and dissipates only 0.176 mW, thus saving more than 65% and 57% of area and power, respectively, while experiencing a lower error with respect to [21].

Another approximate circuit integrated in the ICSC Flagship 2 project is HIMAM [23]: the first Hardware Implementation of the Multiply-and-Max/Min (MAM) layers, recently proposed as an effective alternative to the traditional Multiply-and-Accumulate (MAC) layers used in DNNs. Most recently, Prono *et al.* [24] introduced the new MAM computational layer to take into account that, once a MAC-based neuron is pruned, the amount of data to be aggregated is lower than the non-pruned case. The general idea behind the MAM paradigm is to explicitly identify which inputs are used to compute the output and which are not, thus resulting in a paradigm more prone to aggressive pruning compared to the traditional MAC-based counterpart. Our custom hardware architecture performs the processing steps schematized in 4. The novel approach operating on floating-point data was devised to efficiently support the multiply then compare-and-add pipeline involved in MAM layers, by exploiting the observation that most of the involved product operations can be simplified through proper approximate techniques. We analyzed the novel architecture for input parallelism levels ranging from 4 to 16 and considering both 32-bit (FP32) and 16-bit (FP16) floating-point arithmetic. When implemented on a Xilinx Zynq Ultrascale+ FPGA device, the 4-input FP32 configuration of the proposed circuit occupies just 3374 LUTs and dissipates 174 mW, which is at least 58.1% and 41.2% lower than the state-of-the-art, respectively. The corresponding ASIC implementation carried out by using a 28-nm FDSOI technology process achieves an energy efficiency of 377.35 GFLOPS/W and confirms its advantages over MAC-based competitors.

V. HETEROGENEOUS CPU-GPU-ASIC-FPGA PLATFORMS FOR AI AND HPC APPLICATIONS

Optimizing AI performance and energy efficiency requires comprehensive adjustments across the software/hardware stack, as well as the end-to-end data flow (from data host to accelerator). Our ICSC Flagship 2 initiative involved a benchmarking campaign on a key DL model for medical image segmentation. We utilized specialized profiling tools to assess performance

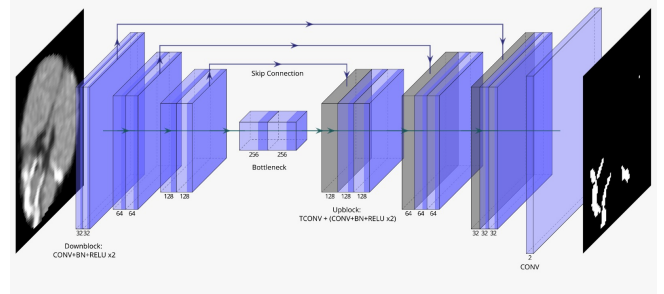


Fig. 5: Architecture of the 3D-UNet Medical CNN. Adapted with permission from [27] under Creative Commons License (CC BY 4.0).

characteristics and energy efficiency metrics for CPU, GPU, and FPGA architectures across the training and inference stages of the DL pipeline [25], [26]. The data represent a crucial reference for future optimization and tradeoff analyses.

Furthermore, we investigated the inference performance of FPGA-based Deep Processing Units (DPUs) on the AMD Alveo U55C, using calcium segmentation in cardiac CT scans as the benchmark [27]. We successfully ported and deployed a U-Net DNN model onto the DPUs (see Fig. 5). A comparison of its accuracy, throughput, and energy efficiency against four generations of GPUs and a recent dual 32-core CPU platform demonstrated that the U-Net runs effectively on DPUs using 8-bit integer computation with accuracy comparable to floating-point GPU/CPU models. We achieved maximum prediction accuracy through hyperparameter optimization and model size reduction via pruning. Crucially, we quantized the model using different numerical schemes to exploit the low-precision processing capabilities of DPUs and GPUs. The DPUs delivered a competitive inference latency of ~ 3.5 ms and a throughput of ~ 4.2 kFPS. This boosted the performance of a 64-core CPU system by $\sim 10\%$ (latency) and $2\times$ (throughput) but did not surpass the performance of GPUs running at the same numerical precision. Considering the energy efficiency, the improvements are approximately a factor $6.7X$ compared to the CPU, and $1.6X$ compared to the NVIDIA P100 GPU manufactured with the same technological process (16 nm). Different GPU generations have been tested to provide a wide range of technologies for the benchmarking campaign, as shown in Fig.6.

After the identification of performance bottlenecks, we enhanced the overall DL performance by addressing the I/O path. This was achieved through custom solutions based on the Computational Storage paradigm [28] and the exploration of Persistent Memory or low-latency SSDs. These I/O optimizations yielded a training time reduction of up to 10% and an inference throughput improvement of up to 10%.

VI. DESIGN OF ACCELERATORS BASED ON RISC-V OPEN-HARDWARE ARCHITECTURE

As highlighted in [1], most of the currently available RISC-V ecosystems for deep learning accelerators operate within a relatively constrained power envelope, typically clustered in

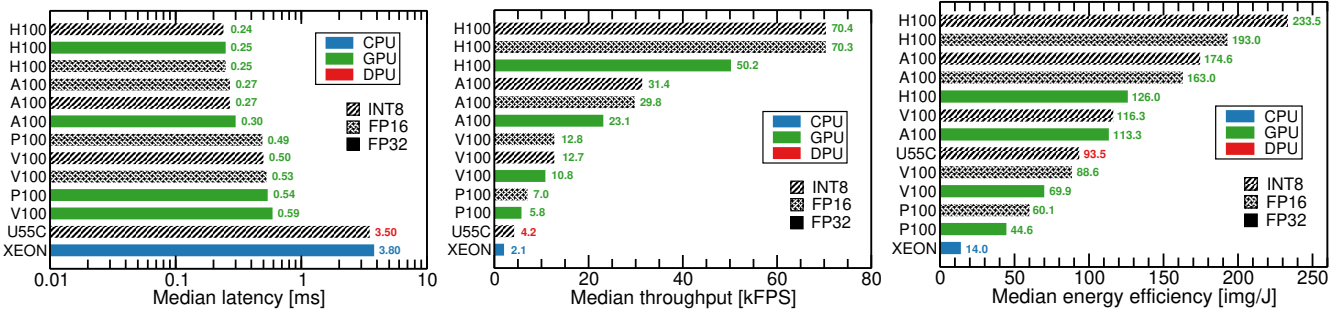


Fig. 6: Comparison results among CPU, GPUs, and DPU deployed on FPGA as median latency, median throughput, and median energy efficiency. Adapted with permission from [27] under Creative Commons License (CC BY 4.0).

the 100 mW to 1 W range. These designs target low-power inference at the edge, but do not fully address the needs of high-performance deep learning workloads. The ICSC Flagship 2 project broadens this scope by targeting architectures exceeding 1 W total power, where significantly higher performance becomes achievable for HPC-class deep learning inference. In this context, one of the key research directions concerns the definition, modeling, and validation of a scalable system architecture — referred to as the Scalable Compute Fabric (SCF) — built upon the RISC-V open ISA and designed to seamlessly grow as performance requirements increase. The SCF comprises a heterogeneous acceleration platform integrated on a single silicon chip or chiplet. A Linux-capable host/controller processor, based on the CVA6 microarchitecture, orchestrates a large number of acceleration components. These include multiple Compute Units (CUs) and specialized hardware engines of differing granularity, interconnected through a high-bandwidth structure such as a hierarchical AXI fabric [29], [30] or a Network-on-Chip solution [31] that ensures scalable communication. CUs are designed as clusters of one or more compute-focused RISC-V cores—such as Snitch².

Building on the SCF template and the RISC-V-centric approach outlined in this section, we designed an open-source MAGIA³, a multi-tile hardware framework for design space explorations of heterogeneous solutions for HPC AI. The MAGIA architecture (shown in Fig. 7), is a scalable template of heterogeneous RISC-V accelerated tiles, organized around a flexible Network-on-Chip [31], and an efficient hardware synchronization mechanism, namely *FractalSynch* [32]. Moreover, MAGIA comes with a complete SDK for software development and exploration. The current reference tile of MAGIA integrates a CV32E40P RISC-V core, extended with the X-IF interface, that eases integration of embedded hardware accelerators or DMA engines through standard and custom RISC-V ISA extensions. This core is used to control the program execution, to program the DMA, and RedMule [33], a mixed-precision matrix engine capable of delivering up to 58.5 GFLOPS (FP16) or 117 GFLOPS (FP8) at 22 nm with efficiencies in the 1.19–1.67 TFLOPS/W range.

An important characteristic of MAGIA is its modularity,

allowing these tiles to be extended with more accelerators, such as the Softex introduced to accelerate Softmax, and other non-linearities in transformer workloads, or to replace the tiles by alternative compute engines, including vector processors [34], other tensor accelerators, or full PULP clusters [35], enabling systematic system-level exploration of heterogeneous fabrics.

MAGIA integrates Fractal Synch. This lightweight hierarchical hardware synchronization mechanism outperforms software-based and NoC-based synchronization, while guaranteeing scalability to a higher number of tiles in the system. On tile meshes from 2×2 up to 16×16 tiles, FractalSync delivers $43 \times$ speedup in barrier time compared to a baseline implementation using software atomic memory operations (AMOs). For an 8×8 mesh, the best AMO/NoC software scheme takes 614 cycles per barrier, while FractalSync completes in 18 cycles, a $34 \times$ speedup; on 16×16 , software rises to 1462 cycles, while FractalSync needs 34 cycles, a $43 \times$ speedup. Even at modest scales (4×4), software requires 347 cycles vs 10 cycles with FractalSync ($34 \times$), and at 2×2 it's 119 cycles vs 6 cycles ($19 \times$). These results highlight how software barriers grow steeply with mesh size, whereas FractalSync stays near-constant/slow-growing thanks to its hierarchical design, all with negligible area cost (sync network 0.007%, full NoC 1.7% of MAGIA).

As a demonstration of the results achieved, we also evaluated the integration of the automatic design space exploration of DIMC accelerators described in Section III in RISC-V architectures. The focus of this activity is to adopt a near-memory computing approach to minimize data movement, thus reducing latency and improving energy efficiency, particularly for emerging workloads such as foundation models. We also plan to explore a tightly-coupled integration approach to embed our DIMC solutions directly within the processor core, as a specialized unit of the processor dataflow. This will enable the efficient execution of deep learning tasks through direct data transfers within the execution flow of the processor. As recently demonstrated in [36], [37], this approach can provide significant advantages. Indeed, the custom arithmetic block proposed in [37] for Bfloat16 exponentiation can be efficiently integrated into the Floating-Point Unit (FPU) of the RISC-V cores of a compute cluster. Through the extended FPU depicted in Fig. 8, with a negligible area overhead of 1%. The optimized software

²<https://github.com/pulp-platform/snitch>

³<https://github.com/pulp-platform/MAGIA>

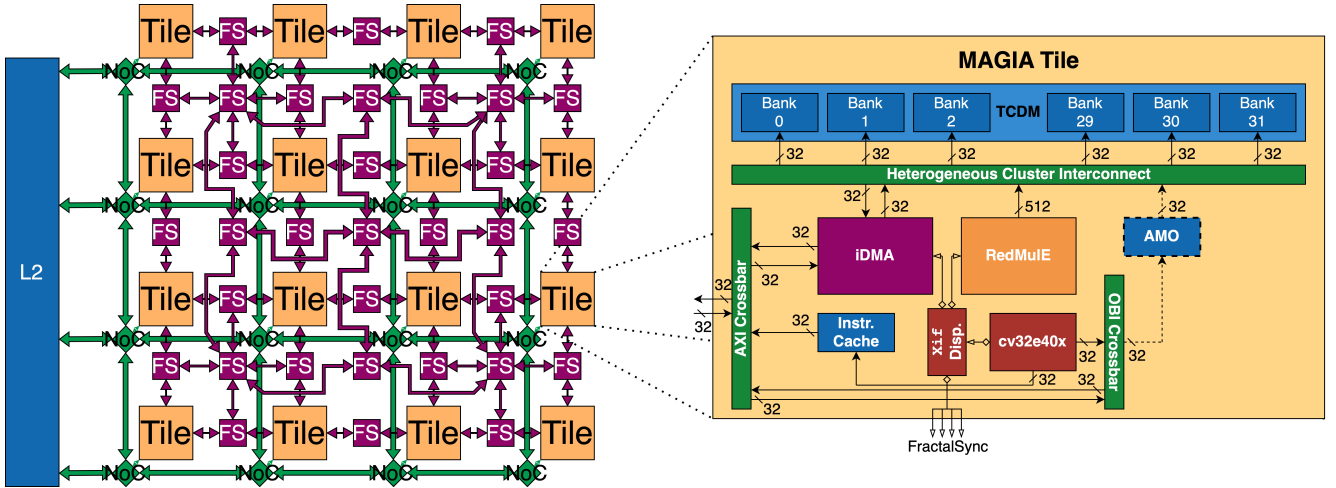


Fig. 7: Architecture of the MAGIA system. [32].

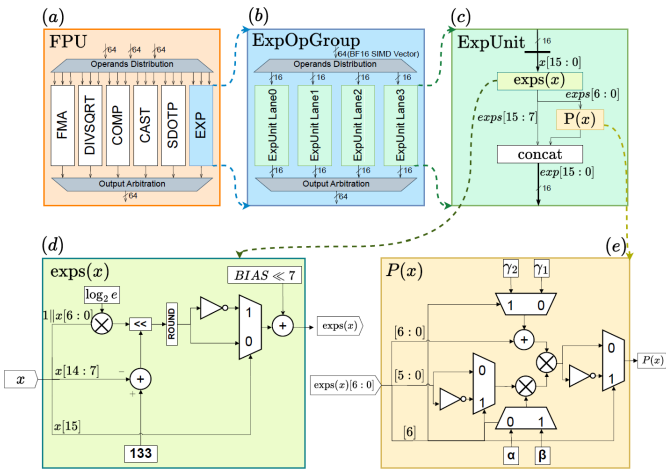


Fig. 8: The extended FPU architecture. [36]

kernels to leverage the extension execute Softmax with 162.7 \times less latency and 74.3 \times less energy compared to the baseline cluster, achieving an 8.2 \times performance improvement and 4.1 \times higher energy efficiency for the FlashAttention-2 kernel in GPT-2 configuration. Moreover, the proposed approach enables a multi-cluster system to efficiently execute end-to-end inference of pre-trained Transformer models, such as GPT-2, GPT-3 and ViT, achieving up to 5.8 \times and 3.6 \times reduction in latency and energy consumption, respectively, without requiring re-training and with negligible accuracy loss.

Another efficient approach to minimize memory transactions was demonstrated in [36], which presents a new flexible algorithm tailored to RISC-V SoCs with a multi-level memory hierarchy for automatic fusion between tiled layers. The Fused-Tiled Layers (FTL) algorithm fuses and optimizes layer tiling using linear transformations to reduce data movement in scratchpad-based memory hierarchies.

FTL has been integrated into Deeploy3, an open-source DNN deployment framework that generates optimized bare-metal

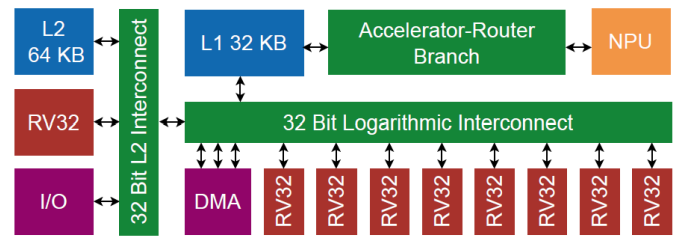


Fig. 9: Overview of the modified Siracusa SoC. Reprinted with permission from [38]. ©2024 IEEE.

C requiring minimal runtime support. We relied on kernels using the extended RV32IMCF-XpulpV2 ISA featuring hardware loops, post-increment load/store, and Single Instruction Multiple Data (SIMD) statements. To move tiles of tensors across the memory hierarchy, we used Direct Memory Access (DMA) engines that rely on the flexibility of RISC-V systems to perform 3D transfers. The modified SoC structure is illustrated in Fig. 9. We leveraged the flexibility and efficiency of a RISC-V (RV32) heterogeneous SoC to integrate FTL in an open-source deployment framework, which we tuned for RISC-V targets. We demonstrated that FTL brings up to 60.1% runtime reduction for a typical Multi-Layer Perceptron (MLP) stage of Vision Transformers (ViTs) due to the reduction of off-chip transfer and on-chip data movement by 47.1%.

VII. CONCLUSIONS

This paper summarizes the main research outcomes of the multi-partner FlagShip 2 project of the Italian National Research Center ICSC. We outlines the key challenges tackled in developing: new toolchains for design space exploration; innovative, energy-efficient IMC architectures; heterogeneous platforms combining CPUs, GPUs, ASICs, and FPGAs—for AI and HPC workloads; and accelerators based on RISC-V architectures.

REFERENCES

- [1] C. Silvano *et al.*, “A survey on deep learning hardware accelerators for heterogeneous hpc platforms,” *ACM Comput. Surv.*, vol. 57, June 2025.
- [2] S. Perri, C. Zambelli, D. Ielmini, and C. Silvano, “Digital In-Memory Computing to Accelerate Deep Learning Inference on the Edge,” in *2024 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pp. 130–133, 2024.
- [3] G. Gozzi, M. Fiorito, S. Curzel, C. Barone, V. G. Castellana, M. Minutoli, A. Tumeo, and F. Ferrandi, “Sparta: High-level synthesis of parallel multi-threaded accelerators,” *ACM Trans. Reconfigurable Technol. Syst.*, July 2024. Just Accepted.
- [4] F. Ferrandi, V. G. Castellana, S. Curzel, P. Fezzardi, M. Fiorito, M. Lattuada, *et al.*, “Bambu: an Open-Source Research Framework for the High-Level Synthesis of Complex Applications,” in *DAC 2021: 58th ACM/IEEE Design Automation Conference*, 2021.
- [5] M. Fiorito, S. Curzel, and F. Ferrandi, “Poster: A system-level hw/sw co-simulation framework for hls-generated accelerators,” in *Proceedings of the 22nd ACM International Conference on Computing Frontiers*, CF ’25, (New York, NY, USA), p. 216–217, Association for Computing Machinery, 2025.
- [6] E. Russo, M. Palesi, G. Ascia, D. Patti, S. Monteleone, and V. Catania, “Memory-aware dnn algorithm-hardware mapping via integer linear programming,” in *Proceedings of the 20th ACM International Conference on Computing Frontiers*, pp. 134–143, 2023.
- [7] A. Das, E. Russo, and M. Palesi, “Multi-objective hardware-mapping co-optimisation for multi-dnn workloads on chiplet-based accelerators,” *IEEE Transactions on Computers*, vol. 73, no. 8, pp. 1883–1898, 2024.
- [8] D. Ielmini and H.-S. P. Wong, “In-memory computing with resistive switching devices,” *Nature Electronics*, vol. 1, pp. 333–343, 2018.
- [9] N. Lepri, A. Glukhov, L. Cattaneo, M. Farronato, P. Mannocci, and D. Ielmini, “In-Memory Computing for Machine Learning and Deep Learning,” *IEEE Journal of the Electron Devices Society*, vol. 11, pp. 587–601, 2023.
- [10] G. Desoli, N. Chawla, T. Boesch, M. Avodhyawasi, H. Rawat, H. Chawla, V. Abhijith, P. Zambotti, A. Sharma, C. Cappelletta, M. Rossi, A. De Vita, and F. Girardi, “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)*, pp. 260–262, 2023.
- [11] R. Balasubramonian, A. B. Kahng, N. Muralimanohar, A. Shafiee, and V. Srinivas, “Cacti 7: New tools for interconnect exploration in innovative off-chip memories,” *ACM Trans. Archit. Code Optim.*, vol. 14, June 2017.
- [12] C. Silvano, S. Perri, P. Corsonello, and F. Frustaci, “Imcacti: a design space exploration framework for digital in-memory computing architectures,” *Submitted to IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2025.
- [13] F. Spagnolo, P. Corsonello, F. Frustaci, and S. Perri, “Design of a Low-Power Super-Resolution Architecture for Virtual Reality Wearable Devices,” *IEEE SENSORS JOURNAL*, vol. 23, no. 8, pp. 9009–9016, 2023.
- [14] W. Chang, K.-W. Kang, and S.-J. Kang, “An Energy-Efficient FPGA-Based Deconvolutional Neural Networks Accelerator for Single Image Super-Resolution,” *IEEE Trans. Circ. Syst. Video Tech.*, vol. 30, no. 1, p. 281–295, 2020.
- [15] C. Sestito, F. Spagnolo, and S. Perri, “Design of Flexible Hardware Accelerators for Image Convolutions and Transposed Convolutions,” *Journal of Imaging*, vol. 7, no. 10, 2021.
- [16] L. Chang, X. Zhao, and J. Zhou, “ADAS: A High Computational Utilization Dynamic Reconfigurable Hardware Accelerator for Super Resolution,” *ACM Trans. Reconf. Techn. Syst.*, vol. Early Access, 2022.
- [17] F. Spagnolo, S. Perri, and P. Corsonello, “Aggressive Approximation of the SoftMax Function for Power-Efficient Hardware Implementations,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 3, p. 1652–1656, 2022.
- [18] P. Ramachandran, B. Zoph, and Q. Le, “Searching for activation functions,” in *Intern. Conf. Learning Representations Workshops*, 2018.
- [19] B. Kang, N. Kim, J. Lee, and H. Kim, “Hardware-friendly activation functions for hybridvit models,” in *Proc. 20th Intern. SoC Design Conf.*, pp. 147–148, 2023.
- [20] A. Howard, M. Sandler, L.-C. C. G. Chu, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, and Q. L. ad H. Adam, “Hardware-friendly activation functions for hybridvit models,” in *Proc. IEEE/CVF Intern. Conf. Computer Vision*, pp. 1314–1324, 2019.
- [21] K. Choi, S. Kim, J. Kim, and I.-C. Park, “Hardware-friendly approximation for swish activation and its implementation,” *IEEE Trans. Circuits and Systems II: Express Briefs*, pp. 1314–1324, 2024.
- [22] F. Spagnolo, S. Perri, and P. Corsonello, “Approximate swish activation function for low-energy yet low-error vlsi implementations,” in *Proc. IEEE Annual Symposium on VLSI (ISVLSI)*, 2025.
- [23] F. Spagnolo, P. Corsonello, and S. Perri, “Himam: Hardware implementation of multiply-and-max/min layers for energy-efficient dnn inference,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 72, no. 8, pp. 1083–1087, 2025.
- [24] L. Prono, P. Bich, C. Boretti, M. Mangia, F. Pareschi, and R. R. G. Setti, “A multiply-and-max/min neuron paradigm for aggressively prunable deep neural networks,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 8, pp. 14414–14427, 2025.
- [25] G. Minghini, A. U. Cavallo, A. Miola, V. Sisini, E. Calore, F. Fortini, R. Micheloni, P. Rizzo, S. F. Schifano, F. V. D. Segal, and C. Zambelli, “An HPC Pipeline for Calcium Quantification of Aortic Root From Contrast-Enhanced CCT Scans,” *IEEE Access*, vol. 11, pp. 101309–101319, 2023.
- [26] V. Sisini, A. Miola, G. Minghini, E. Calore, A. U. Cavallo, S. F. Schifano, and C. Zambelli, “Segmentation of Aortic Valve Calcium Lesions Using FPGA Accelerators,” in *Parallel Processing and Applied Mathematics (R. Wyrzykowski, J. Dongarra, E. Deelman, and K. Karczewski, eds.)*, pp. 113–128, Springer Nature Switzerland, 2025.
- [27] V. Sisini, A. Miola, G. Minghini, E. Calore, A. U. Cavallo, S. F. Schifano, and C. Zambelli, “Benchmarking a DNN for aortic valve calcium lesions segmentation on FPGA-based DPU using the vitis AI toolchain,” *Future Generation Computer Systems*, vol. 175, p. 108115, 2026.
- [28] C. Zambelli, R. Bertaggia, L. Zuolo, R. Micheloni, and P. Olivo, “Enabling computational storage through fpga neural network accelerator for enterprise ssd,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 66, no. 10, pp. 1738–1742, 2019.
- [29] N. Bruschi, G. Tagliavini, A. Garofalo, F. Conti, I. Boybat, L. Benini, and D. Rossi, “End-to-End DNN Inference on a Massively Parallel Analog In Memory Computing Architecture,” Nov. 2022.
- [30] G. Paulin *et al.*, “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)*, pp. 1–2, June 2024.
- [31] T. Fischer, M. Rogenmoser, M. Cavalcante, F. K. Gürkaynak, and L. Benini, “FlooNoC: A Multi-Tb/s Wide NoC for Heterogeneous AXI4 Traffic,” *IEEE Design & Test*, vol. 40, pp. 7–17, Dec. 2023.
- [32] V. Isachi, A. Nadalini, R. F. Gallotta, A. Garofalo, F. Conti, and D. Rossi, “Fractalsync: Lightweight scalable global synchronization of massive bulk synchronous parallel ai accelerators,” in *Proceedings of the 22nd ACM International Conference on Computing Frontiers*, pp. 84–87, 2025.
- [33] Y. Tortorella, L. Bertaccini, L. Benini, D. Rossi, and F. Conti, “RedMule: A Mixed-Precision Matrix-Matrix Operation Engine for Flexible and Energy-Efficient On-Chip Linear Algebra and TinyML Training Acceleration,” Jan. 2023.
- [34] M. Cavalcante, D. Wüthrich, M. Perotti, S. Riedel, and L. Benini, “Spatz: A Compact Vector Processing Unit for High-Performance and Energy-Efficient Shared-L1 Clusters,” in *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design*, (San Diego California), pp. 1–9, ACM, Oct. 2022.
- [35] F. Conti, G. Paulin, A. Garofalo, D. Rossi, A. Di Mauro, G. Rutishauser, G. Ottavi, M. Eggiman, H. Okuhara, and L. Benini, “Marsellus: A heterogeneous risc-v ai-iot end-node soc with 2–8 b dnn acceleration and 30%-boost adaptive body biasing,” *IEEE Journal of Solid-State Circuits*, vol. 59, no. 1, pp. 128–142, 2023.
- [36] J. Victor J. B., B. Alessio, C. Francesco, and B. Luca, “Fused-Tiled Layers: Minimizing Data Movement on RISC-V SoCs with Software-Managed Caches,” *arXiv preprint arXiv:2504.03676*, 2025.
- [37] W. Run, I. Gamze, B. Andrea, P. Viviane, C. Francesco, G. Angelo, and B. Luca, “VEXP: A Low-Cost RISC-V ISA Extension for Accelerated Softmax Computation in Transformers,” *arXiv preprint arXiv:2504.11227*, 2025.
- [38] A. S. Prasad, M. Scherer, F. Conti, D. Rossi, A. D. Mauro, M. Eggimann, J. T. Gómez, Z. Li, S. S. Sarwar, Z. Wang, B. D. Salvo, and L. Benini, “SiraCusa: A 16 nm Heterogenous RISC-V SoC for Extended Reality With At-MRAM Neural Engine,” *IEEE Journal of Solid-State Circuits*, pp. 1–15, 2024.