HEMAIA: A 16 nm Heterogeneous Multi-Accelerator SoC with Accelerator-Centric Execution

Ryan Antonio[1], Xiaoling Yi[1], Jun Yin[1], Ce Ma[1], Yunhao Deng[1], Fanchen Kong[1], Abdelrahman Ayman[1], Joren Dumoulin[1], Josse Van Delm[1], Tiago Knorst[3], Adrian Kneip[2], Guilherme Paim[4], Wim Dehaene[1], Marian Verhelst[1]
ESAT-MICAS, KU Leuven, Belgium[1]; EEMCS, TU Delft, The Netherlands[2];
INF-UFRGS, Brazil[3]; INESC-ID - Instituto Superior Técnico, Portugal[4];

***Abstract*—Neuro-symbolic AI demands heterogeneous multi-accelerator SoCs that integrate uniquely designed engines and orchestrate them efficiently. However, diverse accelerator data access patterns can reduce compute utilization to 10–30% and host-centric orchestration can consume 20–80% of end-to-end runtime. This paper presents HEMAIA, a 9 mm^2 TSMC 16 nm FinFET SoC addressing both through a reusable *accelerator shell* that enables accelerator-centric execution via a customizable data interface with N-dimensional affine-addressed streamers and autonomous embedded RISC-V cores. HEMAIA integrates a tensor core (TC), coarse-grained reconfigurable array (CGRA), and hyperdimensional computing (HDC) engine spanning the full neuro-symbolic compute spectrum. Aggregated accelerator performance and efficiency reach 1.91 TOPS and 2.17 TOPS/W, respectively. Accelerator-centric execution yields 69–98% per-accelerator utilization and system-level overhead below 38%, resulting in 1.27–1.80× latency speedup against host-centric execution and an additional 1.04–2.09× latency and 1.04–2.24× EDP improvement with further pipelining.**

***Index Terms*—Multi-accelerator heterogeneous SoC, AI accelerators, data-driven execution, accelerator shell**

I. Introduction

Neuro-symbolic (NeSy) AI extends classic deep neural networks (DNNs) to solve combined neural and reasoning problems [1]. NeSy augments neural inference with pre-/post-processing and structured symbolic operations, such as hyperdimensional encoding, similarity search, and circular convolutions that have no natural mapping onto conventional DNN accelerators [2]. Efficiently executing such pipelines demands heterogeneous multi-accelerator SoCs spanning the full neuro-symbolic compute spectrum, yet existing platforms [3]–[5] have largely focused on DNN-centric accelerators, and systematically expanding them to incorporate uniquely designed engines faces two systemic bottlenecks.

(1) Diverse data access patterns: accelerators expose fundamentally different memory layouts, access patterns, and per-cycle bandwidths, where incompatibilities between engines disrupt dataflow and collapse effective compute utilization to as low as 10–30% [6]. **(2) Host-centric orchestration:** resolving these mismatches falls on the host CPU, which manages every accelerator invocation, data layout transform, and synchronization event. This creates a scheduling burden that is difficult to parallelize with a single manager, and can consume 20–80% of total end-to-end runtime [7], making the host the dominant bottleneck as pipeline depth and kernel diversity grow.

This work presents **HEMAIA**, a 9 mm^2 heterogeneous multi-accelerator SoC in TSMC 16 nm FinFET that shifts this paradigm toward *accelerator-centric execution* through a novel *accelerator shell* providing: **(1) a customizable data interface** with multi-banked scratchpad memory, multi-width interconnects, and N-dimensional affine-addressed streamers parameterized to each accelerator's native bandwidth and access pattern; and **(2) autonomous local control**, where two embedded RISC-V cores per cluster independently manage kernel invocation and DMA transfers without host involvement, exposing a consistent programming abstraction across all engines. HEMAIA integrates a tensor core (TC) for matrix multiplication and convolution, a coarse-grained reconfigurable array (CGRA) for pre-/post-processing and irregular kernels, and a hyperdimensional computing (HDC) engine for binary symbolic operations. These collectively cover the compute demands of both classic AI and emerging NeSy workloads.

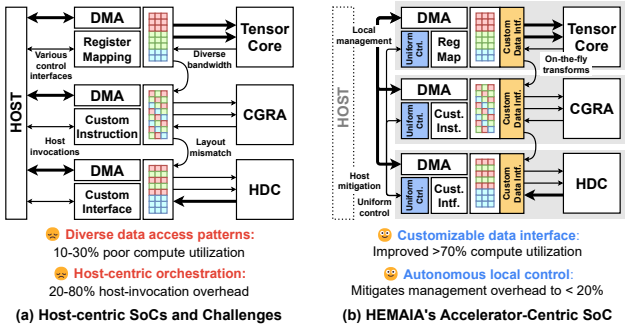

Fig. 1: Challenges in a paradigm shift from a host-centric SoC into an accelerator-centric SoC to support a diversity of accelerators.

II. HEMAIA Architecture

Fig. 2a shows the HEMAIA SoC, built around a modified Occamy platform [8]. It has a shared 1 MB L2 memory as the primary inter-accelerator data staging area over a dual-width AXI NoC (64-bit narrow control, 512-bit wide data fabric). A CVA6 RISC-V host is present only for convenient debug monitoring and cluster-level clock gating, but it plays no role in steady-state execution.

HEMAIA integrates three custom-designed accelerators (detailed in Fig. 2a–b): the TC, CGRA, and HDC, whose computational diversity poses the core integration challenge across two axes. First, data access requirements diverge sharply in both bandwidth and access layout. The TC demands wide 512-bit streams for weight tiles alongside narrow 64-bit activation feeds with regular, affine-addressed tiling patterns suited to dense matrix multiplication. Convolutions are achieved via on-

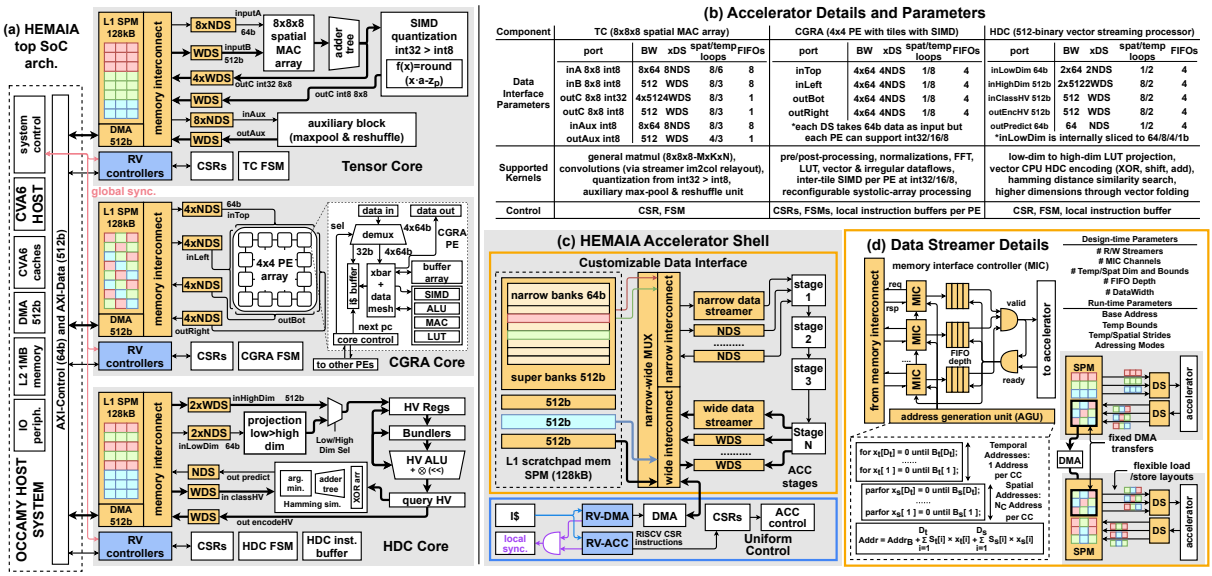

Fig. 2: HEMAIA SoC architecture, accelerator details, and microarchitectures, and accelerator shell details.

the-fly `im2col` re-layout [6]. The TC also processes $8\times8\times8$ matrices in an output-stationary dataflow. The CGRA exposes a four-directional mesh of narrow 64-bit ports per PE tile edge, reflecting its tile-to-tile dataflow and support for irregular and reconfigurable kernels while also managing internal SIMD units inside each PE. The HDC engine operates on wide 512-bit binary hypervectors for high-throughput encoding and similarity search, yet simultaneously consumes narrow 64-bit low-dimensional inputs for its projection stage. Second, control interfaces diverge equally: the TC uses a flat CSR-and-FSM model, the CGRA requires per-PE instruction buffers to express reconfigurable dataflow, and the HDC engine maintains a local instruction buffer for HDC encoding. Resolving both data transfers and accelerator invocation locally, without host involvement, is essential to realizing the accelerator-centric execution paradigm.

The key architectural innovation is encapsulating each engine in a reusable parametric *accelerator shell* (Fig. 2c) that directly addresses both axes. **(1) Customizable data interface:** Inspired by the tightly coupled memory-to-accelerator interface [5], we implement more features to resolve diverging bandwidth and layout requirements. Each shell provides a 128 kB private SPM with 32 narrow banks (64-bit) grouped into 4 wide super-banks (512-bit), connected through a design-time configurable narrow-wide interconnect tunable to each accelerator's native access width. Narrow (NDS) and wide (WDS) run-time programmable data streamers [6] (Fig. 2c–d) supply continuous data via N-dimensional affine AGUs with configurable strides, on-the-fly layout transforms, and FIFO-buffered valid-ready channels that decouple computation from memory latency. Fig. 2d also illustrates how the streamers enable varied dataflow and access patterns, continuously feeding accelerators in the correct access order regardless of the underlying memory layout. **(2) Autonomous local control:** Two lightweight RISC-V IMC cores [9] manage steady-state

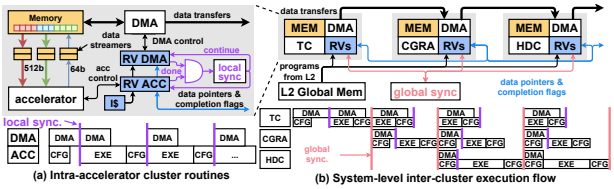

Fig. 3: Intra-accelerator routines and inter-accelerator flows.

execution per cluster: `RV-DMA` for data transfers and `RV-ACC` for accelerator configuration, streamer programming, and kernel launch, issuing all configurations via standard CSR instructions and providing a consistent programming abstraction across all accelerators.

Fig. 3 illustrates intra- and inter-cluster execution. Within each cluster (Fig. 3a), RV-DMA and RV-ACC pipeline data transfers and streamer–kernel configuration and execution in parallel, with a software-defined local synchronizer enforcing data coherence. The data interface is aligned to the accelerator's native bandwidth, while the configured data streamers fetch data at different layouts. Across clusters (Fig. 3b), interaction reduces to pointer and flag exchanges coordinated by a memory-mapped global synchronizer register. The entire system program is stored in L2 global memory and fetched and cached autonomously by each cluster's RISC-V cores, requiring no host involvement after boot. This architecture enables an accelerator-centric paradigm, where each engine operates independently at full datapath efficiency while the system composes seamlessly into productive end-to-end pipelines.

III. Evaluation

A. Silicon Implementation

The HEMAIA SoC was fabricated in TSMC 16 nm FinFET, achieving 650MHz at 0.8V. It was synthesized and retimed with AOCV sign-off using a mixture of LVT and SVT cells to balance speed and leakage. Silicon fabrication is essential to validate the accelerator-centric concept since the complex-

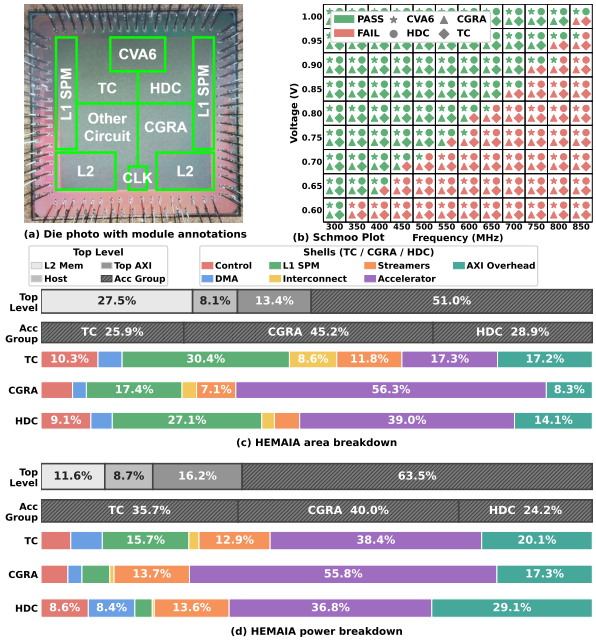

Fig. 4: Die photo with per core schmoo plot, area, and power breakdowns.

ity of inter-accelerator orchestration cannot be fully observed through RTL simulation alone. Fig. 4 presents the annotated die photo, schmoo plot, and hierarchical area and power breakdowns, with HEMAIA spanning 325 MHz at 0.6 V as its most energy-efficient point to 825 MHz at 1.0 V for peak performance. The area breakdown shows that autonomous local control and the customizable data interface account for ≈15–25% of per-cluster area. This overhead is amortized as accelerator datapaths scale. The power breakdown confirms accelerator datapaths and memories as the dominant contributors at over 50%, followed by the data interface, streamers, interconnect, and DMA. Synthesis inserted clock gating was omitted to sustain the target operating frequency. This results in elevated AXI and host dynamic power of ≈25%; resynthesis with switching-annotated simulations confirms this would reduce to below 5%.

B. Per-Accelerator Efficiency

Fig. 5(a) shows per-accelerator throughput (TOPS) and compute utilization across end-to-end AI workloads and kernels: CNN and transformer pipelines, FFT and interpolation, and binary HDC classification. The customizable data interface supplies data at each accelerator's native bandwidth, yielding compute utilizations of **69–98%** across all three engines. Residual loss is workload-inherent, where kernels exceeding the local SPM require L2-to-SPM tiling and introduce unavoidable bank conflicts. Fig. 5(b) shows performance–efficiency curves across operating points, where higher utilization directly translates to improved performance and energy efficiency. These per-accelerator gains carry directly into system-level improvements.

C. End-to-End Pipeline Performance

Fig. 6 presents the evaluated workloads spanning both NeSy variants and classic ML pipelines, demonstrating HEMAIA's

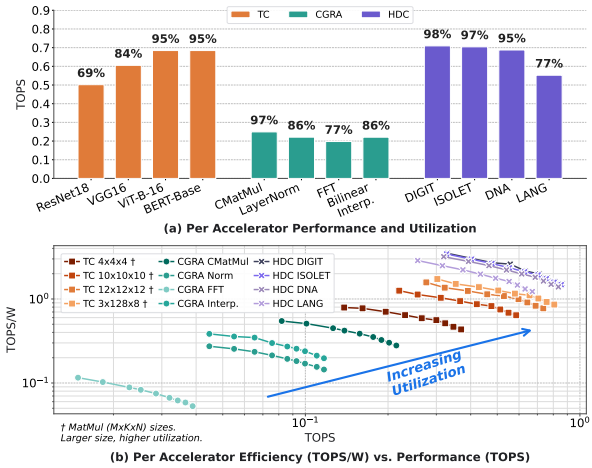

Fig. 5: (a) Per accelerator performance, compute utilization, and (b) energy-efficiency curves.

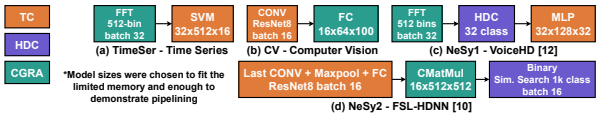

Fig. 6: Benchmark workload scenarios and their corresponding accelerator task allocation in the HEMAIA multi-accelerator dataflow.

breadth across modern and established AI workloads. Fig. 7 illustrates the progressive speedup and execution distribution as accelerators are activated incrementally for the NeSy-2 pipeline. When the HDC task falls back to the CVA6 host, for example, emulating the absence of a dedicated accelerator, it bottlenecks the entire pipeline, underscoring the importance of full accelerator coverage in a heterogeneous SoC.

Fig. 8 compares latency and EDP across four execution modes: (1) a GPU (Jetson Orin Nano Super Developer Kit) as a common edge inference baseline, (2) sequential host-centric accelerator invocation emulated via CVA6 memory-mapped load/store instructions, (3) sequential, and (4) pipelined accelerator-centric execution. Specialized accelerators yield substantial gains over the GPU in latency (1.78–9.83×) and EDP (3.7–75.41×). The GPU incurs 20–80% control and data-movement overhead and performs poorly on irregular kernels such as FFT and memory-bottlenecked symbolic operations [2]. Host-centric execution similarly suffers 30–60% overhead from round-trips of multiple cycles per CSR issue across the 10s–100s of registers required per accelerator configuration. Transitioning to accelerator-centric execution reduces the overhead down to 0.1–38%, yielding 1.27–1.80× latency speedup. The residual overhead is workload-size dependent, where the amount of data processing is minimal, such as in the TimeSer case, where configuration costs constitute a larger share of the runtime. Further pipelining improves latency by 1.04–2.09× and EDP by 1.04–2.24×. Pipelining only yields minimal gains on compute-dominated kernels such as FFT in TimeSer and convolutions in CV that take ≈95% of the execution time. This analysis clearly shows that HEMAIA supports both classic and modern NeSy pipelines within a single unified system, validating the heterogeneous multi-accelerator SoC as a practical substrate for the full AI compute spectrum.

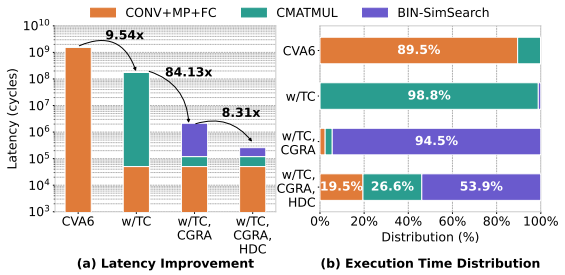

Fig. 7: Progressive speedup as accelerators are incrementally activated for the NeSy2 workload, starting from CVA6-only execution.

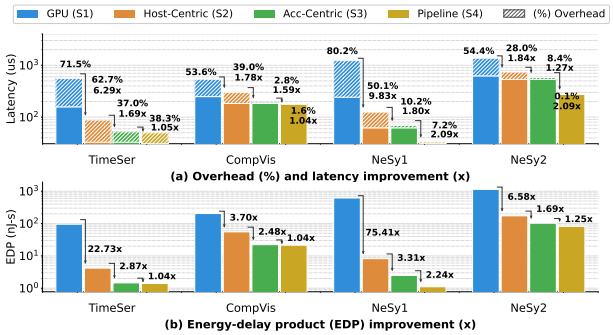

Fig. 8: System latency and EDP across workloads, with control overhead (%) from configurations and DMA transfers, and effective improvement (×).

IV. Comparison with State-of-the-Art (SoTA)

Table I benchmarks HEMAIA against heterogeneous SoCs [3]–[5] and dedicated NeSy accelerators [10], [11]. The two extremes of the flexibility-efficiency tradeoff. All existing heterogeneous SoCs are host-centric, making the orchestration strategy the primary differentiator. Cumulatively, HEMAIA's pipelined accelerators can yield 2.4–15.9× throughput gains and 2.19–13.4× performance density (TOPS/mm^2), with broadly comparable TOPS/W across works. The primary power overhead stems from the AXI interconnect (see Section III-A). To the best of our knowledge, HEMAIA is the only platform that removes the host from steady-state execution through hardware-supported accelerator-centric execution. In contrast with existing frameworks (ESP tiles, RoCC, HWPE [3]–[5]), HEMAIA's accelerator shell parametrically generates a data interface tailored to each engine's native access pattern and embeds autonomous RISC-V-based local control, both configurable from a single file. This enables flexibility of heterogeneous SoCs and the efficiency of dedicated accelerators with a clear path for system expansion.

V. Conclusion

This paper presented **HEMAIA**, a TSMC 16 nm FinFET heterogeneous multi-accelerator SoC demonstrating accelerator-centric execution across a TC, CGRA, and HDC engine, supporting both neuro-symbolic and classic AI workloads. A reusable accelerator shell wraps each engine with a parametrically generated customizable data interface and autonomous embedded RISC-V cores, removing the host from steady-state execution and enabling near-peak per-accelerator efficiency. Post-silicon measurements confirm 1.91 TOPS at 2.17 TOPS/W, 69–98% compute utilization, and system-level orchestration overhead of 0.1–38%. The accelerator-centric execution yields 1.27–1.80× latency speedup, with an additional 1.04–2.09× latency and 1.04–2.24× EDP improvement through pipelining. HEMAIA demonstrates that accelerator-centric heterogeneous SoCs effectively balance efficiency and flexibility across the full AI compute spectrum.

TABLE I: Comparison with State-of-the-Art

Works 'Year	[3] '24	[4] '25	[5] '23	[10] '24	[11] '26	This Work
Type	Hetero SoC	Hetero SoC	Hetero SoC	Acc	Acc	Hetero SoC
Application	General	General/Audio	General/DNN	NeSy	CNN/NeSy	General/AI
Tech (nm)	12	16	16	40	40	16nm FinFet
Voltage (V)	0.5-1	0.55-1.10	0.6-0.8	0.9	0.73-1.1	0.6-1.0
Freq. (MHz)	680-1620	50-1250	360-530	100	50-250	325-850
Power (mW)	83-4330	10-875	151-332	59	73-321	90-1500
Area (mm²)	64	4	16	11.3	27.56	9
Cores	23 cores CPUs, DNN, DSP, Crypto	4xRocket w/ vector acc., FFT, Conv, I2S Audio	1 RV32IMC, 8 RV32IMFC xPULPnnv2, Neureka DNN	CNN, HDC	NN, VSA	CVA6, 6xRV32IMC, TC, CGRA, HDC
Memories	7.8 MB SPM + Caches	256kB L2 32kB SPM	2MB L2 256kB L1 4MB SRAM 4MB NVM	349kB	5.625MB RRAM 0.85MB SRAM	1MB L2 4x128kB SPM >64kB caches
Network BW	6x64b NoC Planes	128b TileLink	32b & 64b AXI	-	-	64b & 512b AXI
TOPS	-	0.006-0.120	RVs-0.0036§ CNN-0.371§ cluster-0.698§	CNN-0.154 HDC-0.025	0.788	TC-0.819^ CGRA-0.217^ HDC-0.870^ Total-1.9^
TOPS/W	-	0.137-0.558	RVs-0.310§ CNN-2.9§ cluster-2.68§	CNN-5.7 HDC-0.780	0.800-10.8	TC-1.750°ᵛ CGRA-0.540°ᵛ HDC-3.39°ᵛ Total-2.17ᵛ
TOPS/mm²	-	0.000375-0.0963*‡	0.0652 §*‡	0.015.8‡	0.029‡	0.211^
TOPS/W/mm²	-	0.034.25-0.110*‡	0.245 §*‡	0.573‡	0.391‡	0.139*ᵛ (0.240)†ᵛ
Architecture Framework						
Orchestration	Host Centric	Host Centric	Host Centric	-	-	Accelerator Centric
Integration	ESP Tiles	Chipyard/RoCC Template	PULP/HWPE Template	-	-	HEMAIA Shell
Parametrizable	✓	✓	✓	-	-	✓

§ int8 only. * Combined host core and accelerator performance and power. † Excluding host power.
° Per accelerator power only. ‡ Computed based on the respective original works.
^ Total TOPS at 1.0V and 850 MHz. ˅ Total TOPS / Total Power at 0.6V and 300MHz.

Acknowledgements

This project has been partly funded by the European Research Council (ERC) under grant agreement No. 101088865, the Flanders AI Research Program, and long-term structural Methusalem funding by the Flemish Government. Circuit fabrication was supported by TSMC via the University Program.

References

[1] M. Hersche, et. al., "A Neuro-vector-symbolic Architecture for Solving Raven's Progressive Matrices", Nat Mach Intell 5, 2023.
[2] Z. Wan, et. al., "Towards Efficient NeSy AI: From Workload Characterization to Hardware Architecture", IEEE TCASAI 2024.
[3] M. C. Dos Santos et al., "14.5 A 12nm Linux-SMP-Capable RISC-V SoC with 14 Accelerator Types, Distributed Hardware Power Management and Flexible NoC-Based Data Orchestration", ISSCC 2024.
[4] E. Gao et al., "COSMIC: A Multi-Vector-Core Heterogeneous RISC-V SoC for Intelligent Audio DSP in Intel 16", ESSCIRC 2025
[5] M. Scherer et al., "Siracusa: A Low-Power On-Sensor RISC-V SoC for Extended Reality Visual Processing in 16nm CMOS", ESSCIRC 2023.
[6] X. Yi, et al., "DataMaestro: A Versatile and Efficient Data Streaming Engine Bringing Decoupled Memory Access To Dataflow Accelerators," DAC 2025.
[7] Van Delm, Josse, et al., "The Configuration Wall: Characterization and Elimination of Accelerator Configuration Overhead." ASPLOS 2026.
[8] 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," VLSI 2024.
[9] F. Zaruba, et. al., "Snitch: A Tiny Pseudo Dual-Issue Processor for Area and Energy Efficient Execution of Floating-Point Intensive Workloads," TCAS1 2021.
[10] H. Yang et al., "FSL-HDnn: A 5.7 TOPS/W End-to-end Few-shot Learning Classifier Accelerator with Feature Extraction and Hyperdimensional Computing," ESSERC 2024.
[11] C. -K. Liu et al., "A 40-nm Programmable Heterogeneous SoC With 5.625/0.85 MB RRAM/SRAM for Accelerating Neuro-Symbolic AI Models," JSSC 2026.
[12] M. Imani, et. al., "VoiceHD: Hyperdimensional Computing for Efficient Speech Recognition," ICRC 2017.