SHWCIM: A Scalable Heterogeneous Workload Computing-in-memory Architecture

Yanfeng Yang[1], Yi Zou[2,*], Zhibiao Xue[3], Liuyang Zhang[4]
[1,2,3]*School of Microelectronics, South China University of Technology*, Guangzhou, China
[4]*National Key Laboratory of Spintronics, Hangzhou International Innovation Institute, Beihang University*, Hangzhou, China
{mi_yyf, mixuezhibiao}@mail.scut.edu.cn, zouyi@scut.edu.cn, liuyang.zhang@buaa.edu.cn

***Abstract*—This study introduces HWCIM, a SRAM-based Computing-In-Memory core, and SHWCIM, a CIM-capable Coarse-Grained Reconfigurable Architecture, to enhance resource utilization, multi-functionality, and on-chip memory size in SRAM-based CIM designs. Evaluated using the SMIC 55nm process, HWCIM achieves 1.6× lower power, 2.8× higher energy efficiency, and up to 4.1× smaller area compared to previous CIM and CGRA works. Additionally, SHWCIM delivers an average 105.9× speedup over existing CGRAs and consumes 2–5× less energy than the Nvidia A40 GPU on realistic workloads.**

I. Introduction

Advancements in computing have enabled AI/ML breakthroughs requiring large datasets and complex algorithms, driving the need for energy-efficient, low-latency hardware. Computing-in-memory (CIM) integrates memory and computation to boost efficiency and speed by reducing data transfers, utilizing technologies like SRAM, DRAM, MRAM, PCM, ReRAM, and Flash [1]. SRAM-based CIM is favored for low voltage, speed, durability, and CMOS compatibility but struggles with utilization, flexibility, and memory constraints.

We introduce SHWCIM, a scalable CIM-based CGRA that overcomes these challenges. Heterogeneous Workload CIM (HWCIM) enhances utilization by supporting memory and computation modes for vector/matrix operations. SHWCIM's PEs offer 1–64 bit precision for sparse and dense data. Evaluations on 15 workloads demonstrate SHWCIM outperforms existing CGRA PEs in throughput, energy efficiency, and power. RTL modeling validates its suitability for memory-constrained applications.

II. Background and Motivation

Neural network algorithms, especially matrix-vector multiplications, demand high computational power and parallelism. Traditional processors, with separate memory and computation, incur high power consumption and latency due to frequent data transfers. CIM integrates memory and computation, enhancing efficiency and speed. SRAM-based CIM technology has rapidly progressed, achieving milestones such as the University of Michigan and Hokkaido University's SRAM CIM chips, ISSCC's focus on CIM, TSMC's CIM macros from 22nm to 4nm, and MediaTek's 12nm designs in 2023 [1]. These advancements are driven by SRAM's process compatibility, flexibility, and reliability.

Reconfigurable computing, like CGRA, improves efficiency by mapping algorithms to customizable engines, utilizing spatial and temporal parallelism to reduce delays and overhead. CGRA's flexibility, energy efficiency, and post-silicon reconfigurability make it superior to ASICs for specific applications.

However, SRAM-based CIM faces challenges in:

- **Resource utilization:** Fixed-size SRAM arrays limit performance and energy efficiency for varying operator sizes.
- **Architecture:** Limited on-chip memory creates bottlenecks, increasing off-chip memory access.
- **Computational completeness:** While CIM excels in tensor operations, it lacks support for vector, scalar, and nonlinear computations.

To address these issues, we propose integrating SRAM CIM into CGRA, enhancing flexibility, throughput, and efficiency. By using SRAM CIM as PEs, the architecture allows dynamic allocation between computing and memory, overcoming memory constraints and supporting diverse workloads through scalable PE arrays.

III. A Scalable Heterogeneous Workload Computing-In-Memory Architecture

As shown in Fig. 2, the Scalable Heterogeneous Workloads Central Computing-In-Memory (SHWCIM) architecture comprises a Separate/Gather Unit (SGU), Management Unit (MU), Switch, and Compute-In-Memory (CIM) Core. The SGU handles data precision and sparsity, while SHWCIM functions as a RISC-V ISA extension co-processor. The MU decodes instructions and manages the CIM Core, termed Heterogeneous Workloads CIM (HWCIM), which includes a 512×256 8T SRAM array, multiple Multi-Functional Units (MFUs) aligned with data bit-width for parallel processing, an adder tree, and peripherals. This setup enables the storage of two 256×256 matrices and performs addition, multiplication, subtraction, and comparison operations.

Additionally, the MU features Scratchpad Memory (SPM), a CIM-Mapped Register, a Crossbar, two Data Interaction Units (DIUs), and CIM Managers, which route commands and data to multiple HWCIMs via the Switch. As illustrated in Fig. 3, MFUs utilize complementary word lines and an add-1 operation for two's complement, allowing the adder

*Corresponding author. This work is partly funded by SCUT Startup Research Fund No. K3200890 and Shenzhen Bureau of Science and Technology Research Fund No. KJZD20240903102708012.

978-3-9826741-0-0/DATE25/© 2025 EDAA

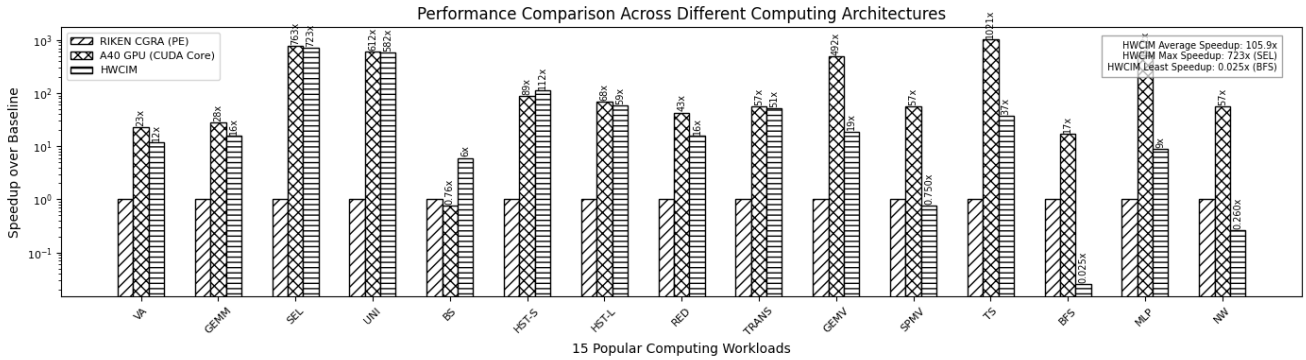

Fig. 1. Performance comparison between RIKEN CGRA [2], A40 GPU, and HWCIM, using RIKEN CGRA (PE) as the baseline.

circuit to efficiently perform both addition and subtraction. Multiplication is achieved by ANDing the WBL and RWL lines to generate partial sums processed by the adder tree. Comparison operations are executed as signed subtractions, determining equality or magnitude based on the results.

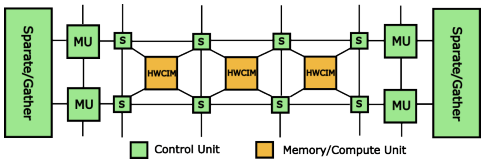

Fig. 2. Illustration of the proposed SHWCIM.

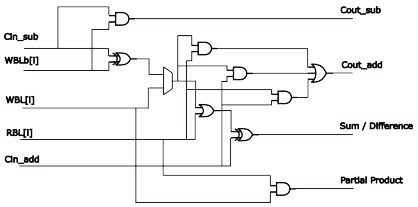

Fig. 3. Illustration of the Single Multi-Functional Unit.

IV. Evaluation of HWCIM and SHWCIM

We evaluate HWCIM and SHWCIM in two stages. HWCIM is fabricated using the SMIC 55nm process, supporting vector and matrix operations, operating at 224 MHz, occupying 3.69 mm², consuming 5.2 mW, supporting 0.75-1.1 V, and achieving 2.3 TOPS/W. Compared to recent CIM works and CGRA PEs (Table I), it demonstrates 1.6x lower power, 2.8x higher energy efficiency, and is 2.8–4.1x smaller in area. SHWCIM is preliminarily evaluated on the Xca7a200t FPGA with RTL modeling, integrating a RISC-V CPU to issue instructions and data, using two 8KB RAM models to simulate HWCIM behavior, and tested across PE array sizes of 3x3, 8x8, and 16x16, achieving a 2-5x power reduction compared to the A40 GPU. Benchmarking across 15 workloads (Figure 1) shows that HWCIM matches the A40 GPU's CUDA cores and outperforms RIKEN CGRA PEs, especially in Binary Search. For less CIM-suitable tasks, HWCIM still surpasses CGRA but remains below GPU performance.

	VLSI [3]	**JSSC [5]**	**TPDS [4]**	**IPDPS [2]**	**This Work**
CIM	Y	Y	N	N	Y
Tech	65nm	28nm	45nm	7nm	55nm
Freq(MHz)	245	50	N/A	288	224
Area(mm²)	1.52	4.74	15.20	12.42	3.69
Power(mW)	8.4	N/A	484	250.8	5.2
Vdd(V)	1	0.8	3.3	3.3	0.75-1.1
Eff.(TOPS/W)	0.8	1.14	N/A	0.14	2.3
Ops Supported	Bool Vec Shift Vec Arith Vec	1b×1b MAC Ref Volt Gen A/D Switch	Add Sub Mul Comp	Add Sub Mul Fused Mul	Vec Add Vec Sub Vec-Mat Mul Mat-Mat Mul

TABLE I
Comparison of This Work with Previous Studies

V. Conclusion

We propose SHWCIM, a CGRA architecture using HWCIM CIM cores as PEs to address SRAM-based CIM challenges: low utilization, limited computation, and on-chip memory bottlenecks. HWCIM supports vector and matrix operations with optimized circuit compactness. SHWCIM features a scalable, workload-adaptive design with a custom RISC-V extension and centralized address allocation. Evaluated with the SMIC 55nm process, HWCIM achieves 1.6× lower power, 2.8× higher energy efficiency, and is 2.8–4.1× smaller than previous CIM and CGRA PEs. Benchmarking on 15 workloads shows HWCIM matches A40 GPU performance and outperforms CGRA PEs. SHWCIM demonstrates feasibility and potential for memory-constrained applications.

References

[1] Kazi Asifuzzaman, Narasinga Rao Miniskar, Aaron R Young, Frank Liu, and Jeffrey S Vetter. A survey on processing-in-memory techniques: Advances and challenges. *Memories-Materials, Devices, Circuits and Systems*, 4:100022, 2023.

[2] Emanuele Del Sozzo, Xinyuan Wang, Boma Adhi, Carlos Cortes, Jason Anderson, and Kentaro Sano. Exploration of trade-offs between general-purpose and specialized processing elements in hpc-oriented cgra. In *2024 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 668–680. IEEE, 2024.

[3] Yuhao Ju, Yijie Wei, Xi Chen, and Jie Gu. A general-purpose compute-in-memory processor combining cpu and deep learning with elevated cpu efficiency and enhanced data locality. In *2023 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*, pages 1–2, 2023.

[4] Di Mou, Bo Wang, and Dajiang Liu. Sc-cgra: An energy-efficient cgra using stochastic computing. *IEEE Transactions on Parallel and Distributed Systems*, 2024.

[5] Chun-Yen Yao, Tsung-Yen Wu, Han-Chung Liang, Yu-Kai Chen, and Tsung-Te Liu. A fully bit-flexible computation in memory macro using multi-functional computing bit cell and embedded input sparsity sensing. *IEEE Journal of Solid-State Circuits*, 58(5):1487–1495, 2023.