SSSLab 超级科学软件实验室 Super Scientific Software Laboratory

中国石油大学 CHINA UNIVERSITY OF PETROLEUM BEIJING

ReRAM-Based Process-In-Memory Accelerator for Iterative Solvers: A Systematic Survey

Boyu Geng[1], Mingjia Fan[1], Zhou Jin[2], Weifeng Liu[1]

[1]SSSLab, Dept.of CST, China University of Petroleum-Beijing, China, [2]College of Integrated Circuits, Zheiiang University, Hangzhou. China

Introduction

Iterative solvers are foundational for large-scale scientific computing, enabling solutions to sparse linear systems critical in simulations, machine learning, and optimization. However, traditional von Neumann architectures suffer from the "memory wall" due to data movement bottlenecks. ReRAM-based Process-In-Memory (PIM) architectures offer a breakthrough by integrating computation within memory, enabling matrix operations at O(1) complexity. This poster surveys advances in ReRAM-PIM for iterative solvers, categorizing key innovations and outlining future challenges.

- Data Movement Overhead: Frequent transfers between CPU and memory limit performance.
- Precision vs. Efficiency Trade-off: High-precision floating-point arithmetic strains conventional hardware.
- Sparsity and Irregularity: Sparse matrix operations require non-deterministic memory access, degrading efficiency.
- Scalability: Analog ReRAM non-idealities (e.g., conductance drift) hinder large-scale deployment.

TABLE I: Summary of ReRAM based PIM accelerators for iterative solvers.

Category	Citation	Solver	Baseline	Speedup	Energy Efficiency	Iteration	Sparsity	Year
Mixedprecision strategy	[17]	CG	NVIDIA Tesla K10 GPU	1500×	8.5×	√	√	2015
	[18]	GMRES	IBM POWER8 CPU, NVIDIA P100 GPU	6.3−17.5× (CPU), 3.6−7.8× (GPU)	6.8 − 24×	√	√	2018
	[19]	Block-Jacobi preconditioned flexible GMRES	2.3 GHz 8-Core Intel i9 machine equipped with GB of system memory	FGMRES required 2× to 4× fewer FLOPS than PGMRES + ILU	——	√	×	2023
Based on the feedback circuit theory	[20]	——	——	O(1)	——	×	×	2019
	[21]	——	——	O(logN) or O(1) (model covariance), O((1/λ_{min}))	——	×	√	2020
	[22]	Eigenvector	——	O(1)	——	×	×	2020
	[23]	Jacobi iterative method	NVIDIA Tesla P40 GPU	100×	1000×	×	√	2021
	[24]	Least-Squares Fitting	NVIDIA K40m GPU	132−3282×	8201−96738×	×	×	2022
	[25]	——	original AMC	——	1.6 − 1.67×	×	×	2024
Enabling floating-point computations	[26]	CG, BiCG	NVIDIA Tesla P100 GPU	10.3×	10.9×	√	√	2018
	[27]	CG, BiCGSTAB	NVIDIA Tesla V100 GPU, PIM accelerator [26]	12.59× (CG GPU), 12.94× (CG PIM), 13.34× (BiCGSTAB GPU), 15.98× (BiCGSTAB PIM)	——	√	√	2023
Dealing with irregularity and sparsity with CAM	[28]	AMG	AMD 2nd EPYC 7702 CPU, NVIDIA Tesla A100 GPU	10× (CPU), 100× (GPU)	100× (CPU), 1000× (GPU)	√	×	2023
	[29]	JPCG	AMD 2nd EPYC 7702 CPU, NVIDIA Tesla A100 GPU, Xilinx Alveo U280 FPGA	1000× (CPU), 10× (GPU), 10× (FPGA)	100× (CPU), 100× (GPU), 10× (FPGA)	√	√	2024

ReRAM-PIM Contributions

1. Mixed-Precision Strategy

- Method: Hybrid analog-digital workflows for iterative solvers.
- Richter et al. : ReRAM-based preconditioning for Conjugate Gradient (CG).
- Kalantzis et al. : Block-Jacobi preconditioning for GMRES using ReRAM/PCM arrays.
- Le Gallo et al. : Mixed-precision GMRES with analog matrix-vector multiplication.
- Advantage: Combines ReRAM's low-precision analog acceleration with CPU-based high-precision refinement to balance efficiency and accuracy.
- Performance:
- CG: 1500× speedup, 8.5× energy efficiency vs. NVIDIA Tesla K10 GPU.
- GMRES: 6.3–17.5× (CPU) / 3.6–7.8× (GPU) speedup, 6.8–24× energy efficiency.
- Flexible GMRES: 2–4× FLOPS reduction vs. ILU-preconditioned GMRES.

2. Based on the Feedback Circuit Theory

- Method: Circuit-level equation solving via Ohm-Kirchhoff laws.
- Sun et al. : Linear system/eigenvector solving in O(1)/O(logN) time.
- Song et al. : Closed-loop Jacobi iteration mapped to ReRAM crossbars.
- Chen et al. : Least-squares solver using analog matrix inversion.
- Pan et al. : BlockAMC architecture for scalable analog matrix computation.
- Advantage: Eliminates iterative steps by leveraging analog circuit equilibrium.
- Performance:
- Jacobi: 100× speedup, 1000× energy efficiency vs. NVIDIA Tesla P40 GPU.
- Least-squares: 132–3282× speedup, 8201–96,738× energy efficiency vs. NVIDIA K40m GPU.
- BlockAMC: 1.6–1.67× throughput improvement vs. baseline AMC.

3. Enabling Floating-Point Computations

- Method: Hardware-software co-design for high-precision arithmetic.
- Feinberg et al. : Exponent truncation with mantissa padding for CG/BiCG.
- Song et al. : ReFloat data format with exponent locality optimization.
- Advantage: Enables scientific-grade precision on ReRAM arrays.
- Performance:
- CG/BiCG: 10.3× speedup, 10.9× energy efficiency vs. NVIDIA Tesla P100 GPU.
- ReFloat: 12.59–15.98× speedup vs. NVIDIA V100 GPU and prior ReRAM designs.

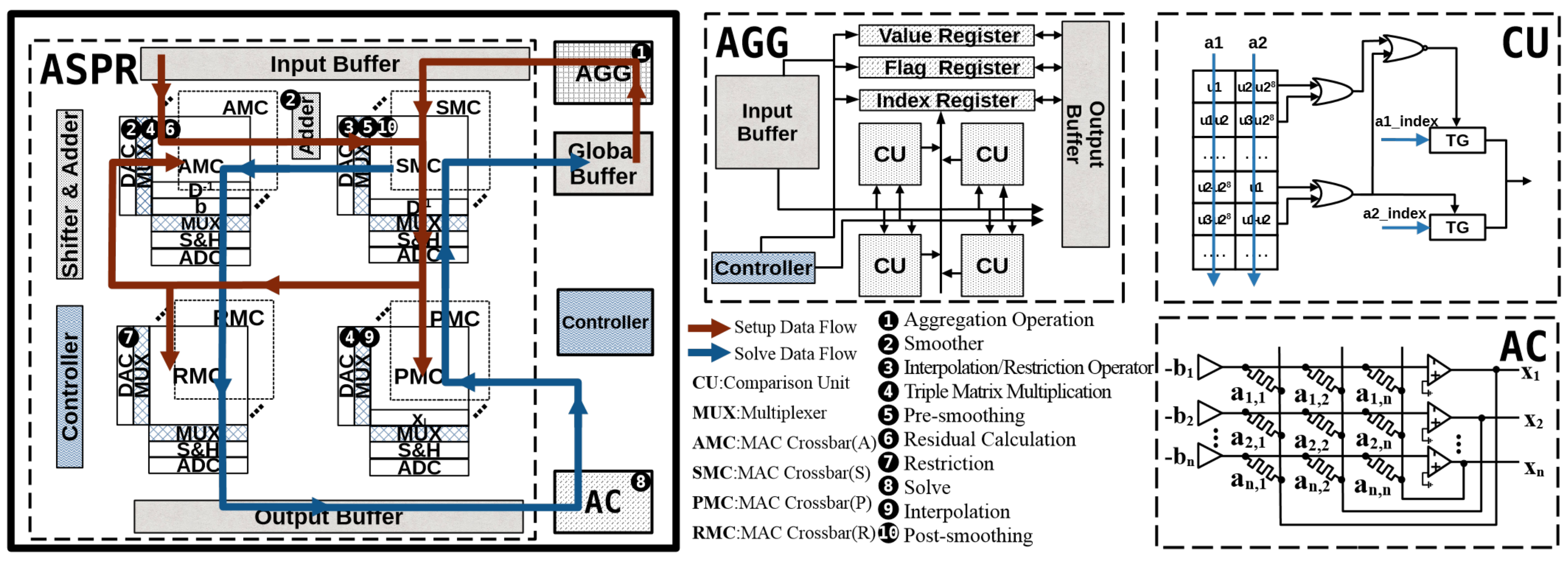

Fig. 3: The architecture of AmgR [28].

4. Dealing with Irregularity and Sparsity with CAM

- Method: Content-addressable memory (CAM) for irregular/sparse operations.
- Fan et al. : AmgR architecture for Algebraic Multigrid (AMG).
- Fan et al. : ReCG framework for sparse Conjugate Gradient.
- Advantage: Parallel pattern-matching resolves unstructured data access.
- Performance:
- AMG: 10× (CPU) / 100× (GPU) speedup, 100–1000× energy efficiency.
- ReCG: 1000× (CPU) / 10× (GPU/FPGA) speedup, 100–10× energy efficiency.

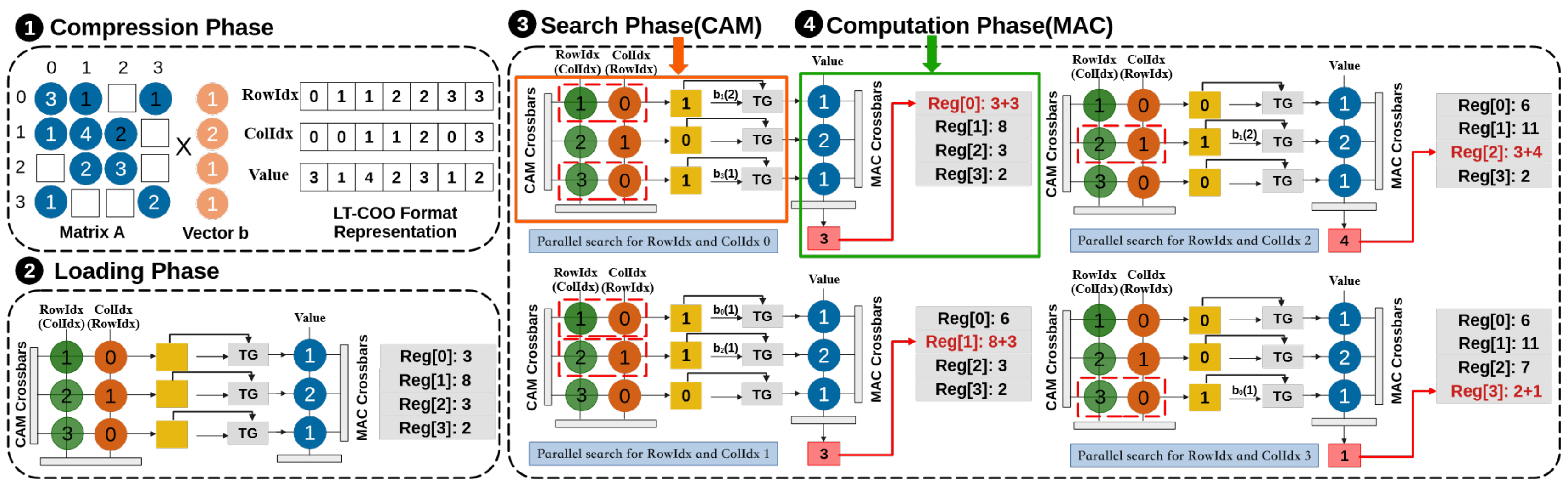

Fig. 4: Implement SpMV on ReRAM, including Compression, Loading, Search, and Computation Phases

Conclusion

ReRAM-PIM architectures revolutionize iterative solvers by merging computation and memory, achieving orders-of-magnitude gains in speed and energy efficiency. Critical advancements in mixed-precision, feedback circuits, floating-point support, and CAM-based sparsity handling lay the groundwork for broader adoption in scientific computing. Future work must address precision, scalability, and operator complexity to unlock ReRAM's full potential.

Acknowledgments:

Supported by NSFC (U23A20301, 62234010), State Key Lab of Computer Architecture.

Contact:

Zhou Jin (z.jin@zju.edu.cn)

References:

1. Richter, K. Pas, X. Guo, R. Patel, J. Liu, E. Ipek, and E. G. Friedman, "Memristive accelerator for extreme scale linear solvers," in GOMACTech, 2015.
2. Z. Sun, G. Pedretti, E. Ambrosi, A. Bricalli, W. Wang, and D. Ielmini, "Solving matrix equations in one step with cross-point resistive arrays," in Proceedings of the National Academy of Sciences (PNAS), 2019.
3. B. Feinberg, U. K. R. Vengalam, N. Whitehair, S. Wang, and E. Ipek, "Enabling scientific computing on memristive accelerators," in International Symposium on Computer Architecture (ISCA), 2018.
4. M. Fan, X. Tian, Y. He, J. Li, Y. Duan, X. Hu, Y. Wang, Z. Jin, and W. Liu, "AmgR: Algebraic multigrid accelerated on ReRAM," in Design Automation Conference (DAC), 2023.
5. M. Fan, X. Chen, D. Yang, Z. Jin, and W. Liu, "ReCG: ReRAM-accelerated sparse conjugate gradient," in Design Automation Conference (DAC), 2024.

Full list in accompanying paper.