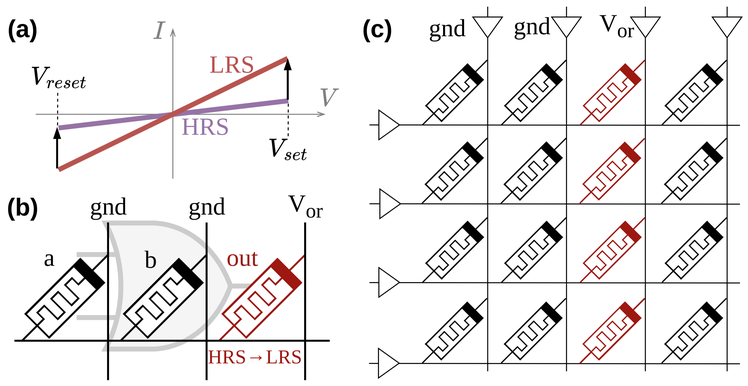
EDA LANDSCAPE FOR DIGITAL MEMRISTIVE IN-MEMORY COMPUTING

Fabian Seiler [fabian.seiler@ziti.uni-heidelberg.de], Nima TaheriNejad

ECLECTX Team, Heidelberg University, Germany

I. Digital Memristive Stateful Logic

Digital In-Memory Computing (IMC) mitigates the von Neumann bottleneck by performing computation directly within memory arrays, reducing costly data movement between processor and memory: **(a)** memristive devices store information through reversible switching between high- and low-resistance states, **(b)** Boolean operations are executed directly on these stored states using stateful logic, and **(c)** crossbar arrays enable highly parallel in-memory computation by operating on many devices simultaneously.

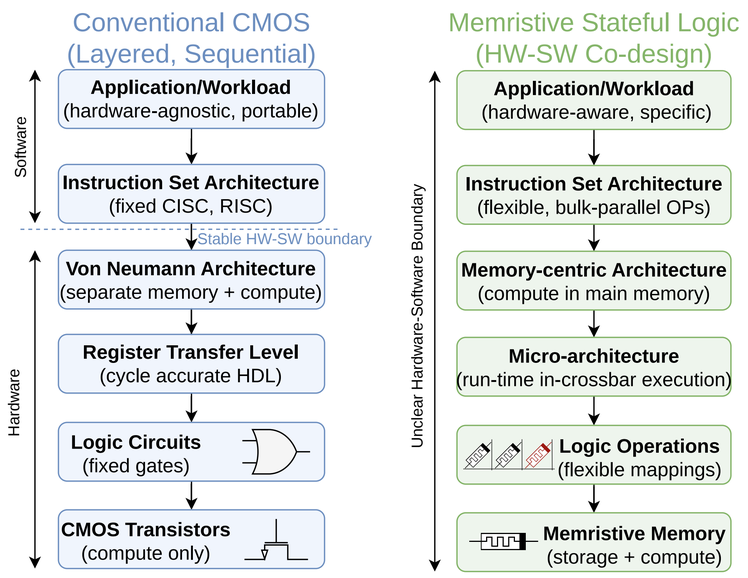


II. Hardware–Software Co-Design

Conventional CMOS relies on stable abstraction layers, where software primarily interacts with hardware through well-defined interfaces using the instruction set architecture (ISA).

In **memristive digital stateful logic**, software decisions must be co-developed with hardware capabilities. Algorithms, instruction sets, mapping strategies, and execution schedules are closely coupled to architectural organization and stateful operation primitives.

Memristive IMC requires **hardware–software co-design** across multiple abstraction layers instead of a sequential design flow.

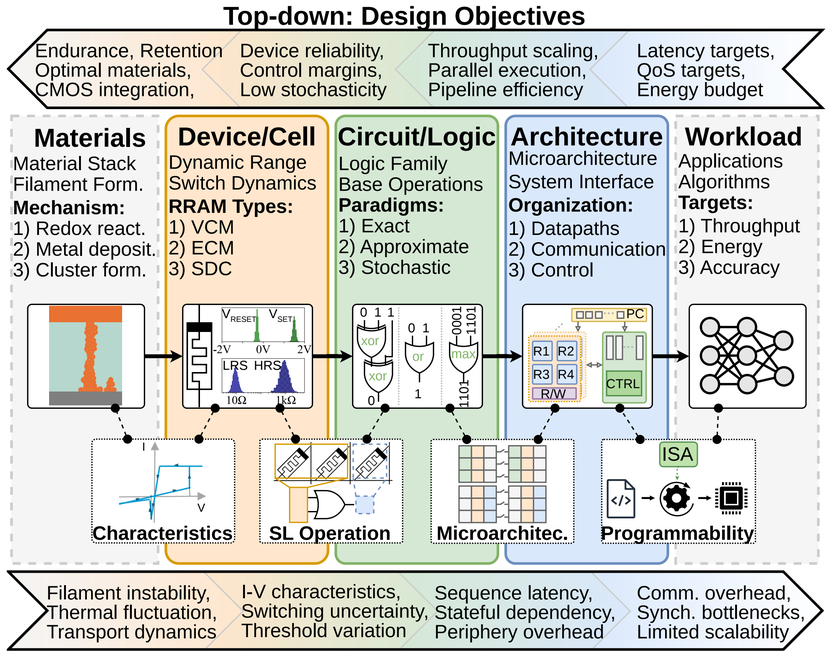


III. Cross-Layer Dependencies

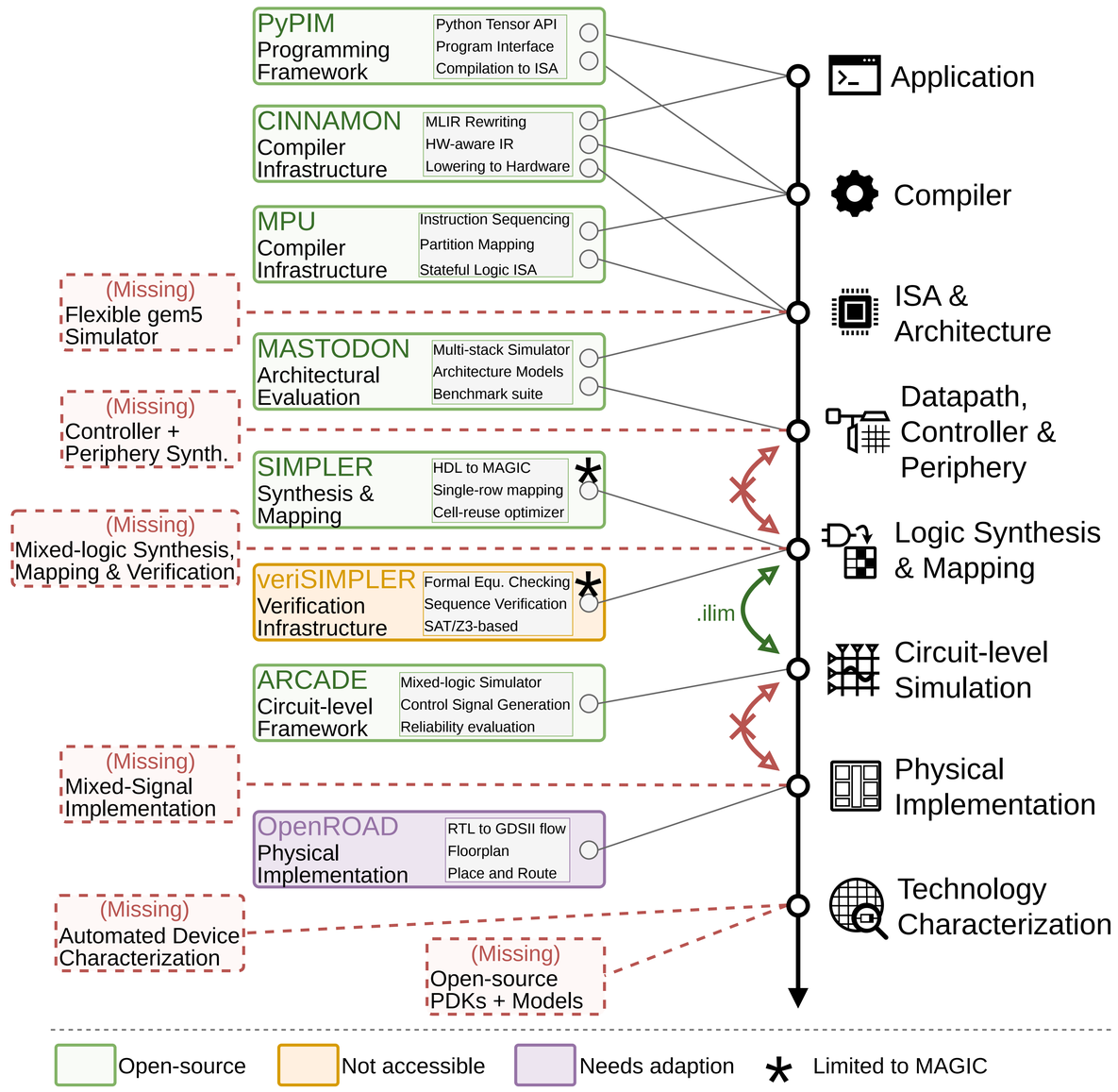
Conventional CMOS design relies on stable abstraction layers and well-defined interfaces, allowing each layer to be developed largely independently within a sequential EDA flow.

Digital stateful logic tightly couples device physics, logic operations, architecture, and software, creating **cross-layer dependencies** where decisions at one layer directly influence the performance and functionality of others [1].

These dependencies highlight the need for **emerging cross-layer EDA tools** that support **exploration**, **evaluation**, and **automation** across the complete design flow of memristive in-memory computing systems.



IV. Current EDA Landscape



V. Open Challenges and Outlook

Open Challenges

- **Limited cross-layer integration:** Existing frameworks [2, 3, 4] typically target only one or two abstraction layers, while digital memristive stateful logic exhibits strong dependencies spanning devices, circuits, architectures, compilers, and applications. Effective design-space exploration requires bidirectional feedback across layers.
- **Restricted flexibility:** Most available tools are developed for specific logic families, architectures, or execution models, limiting their applicability to the broader spectrum of memristive computing paradigms and emerging hardware platforms.
- **Missing automation across the design flow:** Several stages remain largely manual, including mixed-logic synthesis, controller and peripheral generation, technology characterization, and physical implementation, creating barriers to scalable system development.
- **Lack of interoperability:** Existing frameworks operate largely as standalone solutions with limited compatibility. Standardized intermediate representations and exchange formats are required to enable seamless communication and reuse across tools. Emerging formats such as `.ilim` provide a promising foundation by capturing mapped stateful logic sequences across multiple abstraction layers.
- **Insufficient support for physical realization:** While higher-level design automation is advancing, the lower layers remain underdeveloped. Automated mixed-signal and mixed-logic implementation flows, comparable to those available for CMOS digital design, are still largely missing.

Outlook

The next generation of EDA infrastructures should evolve toward an open, end-to-end, and cross-layer design ecosystem for memristive in-memory computing. Such an ecosystem must support continuous optimization from materials and devices up to applications and workloads, enabling automated exploration of design trade-offs across all abstraction layers. Standardized representations, interoperable toolchains, and open-source models will be key enablers for community-driven development. Closing the remaining gaps in synthesis, verification, controller generation, physical implementation, and technology characterization will ultimately pave the way for a complete RTL-to-fabrication flow for digital memristive systems. The long-term vision is a mature EDA stack that makes memristive computing platforms as accessible, portable, and scalable as contemporary CMOS-based design flows.

References

[1] Fabian Seiler et al. ‘‘Cross-Layer Design Approaches of Stateful Digital Memristive In-Memory Computing’’. In: *Submitted to the 39th IEEE International System-on-Chip Conference (SoCC)*. 2026.

[2] Chandan Kumar Jha et al. ‘‘Devices, Synthesis, Circuits, and Verification for Resistive RAM-Based Digital Computing-in-Memory’’. In: *ICEdge*. 2025.

[3] Asif Ali Khan et al. ‘‘CINM (Cinnamon): A Compilation Infrastructure for Heterogeneous Compute In-Memory and Compute Near-Memory Paradigms’’. In: *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 4*. 2025.

[4] Minh S. Q. Truong et al. ‘‘The Memory Processing Unit: A Generalized Interface for End-to-End In-Memory Execution’’. In: *HPCA*. 2026.

Acknowledgement

We appreciate the support provided for this work by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – under Project 550835853.