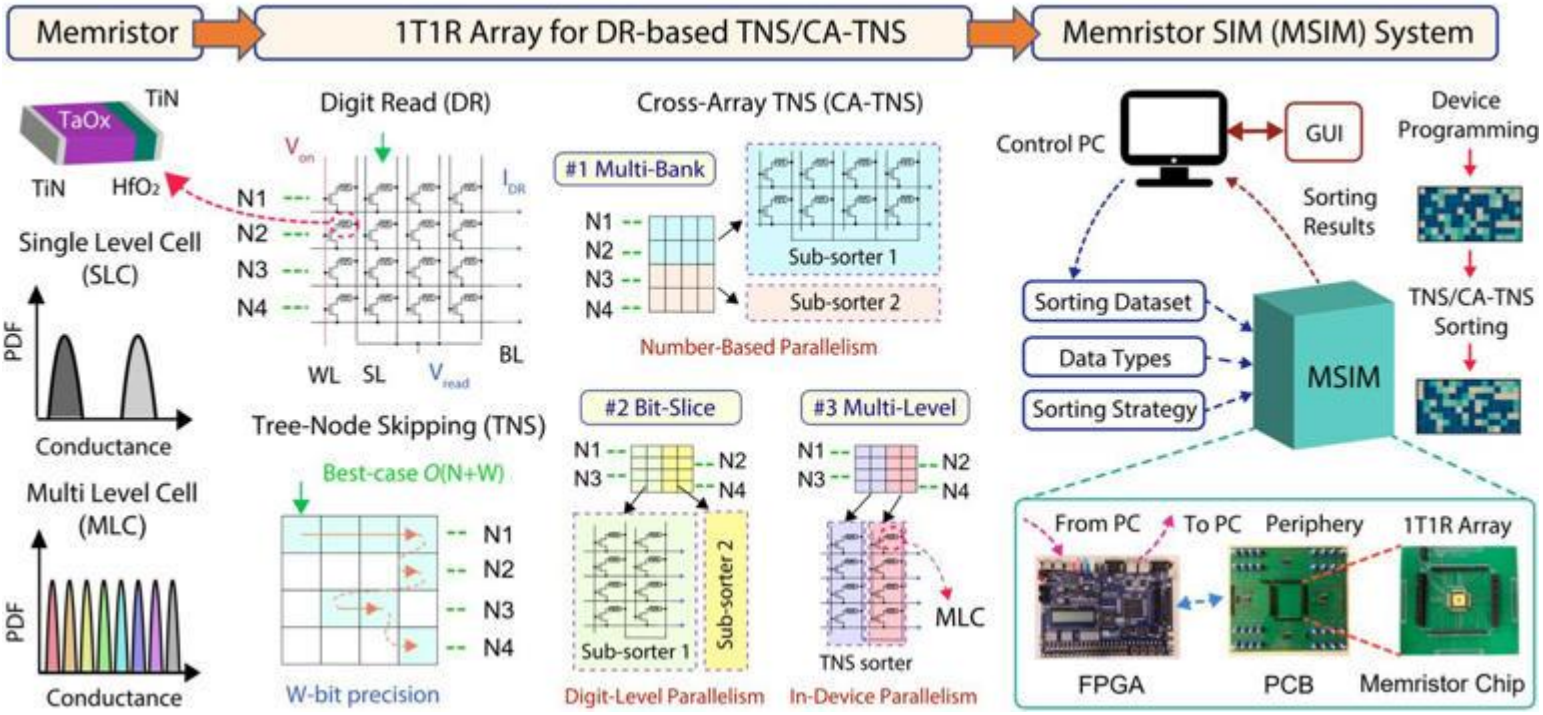
New reconfigurable memristor-based system enables in-memory data sorting



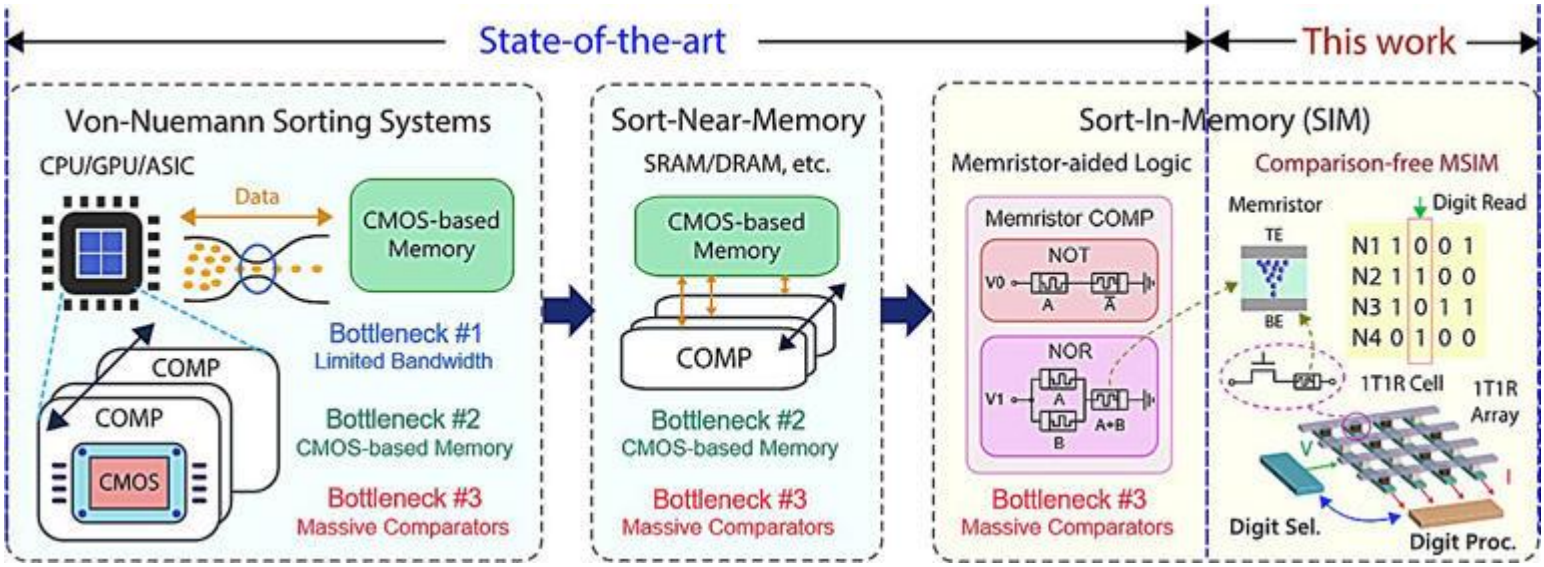
Enabled by memristor devices, this work develops a 1T1R array for digit read (DR), TNS/CA-TNS techniques for comparison-free SIM and an end-to-end MSIM system for practical demonstration. Credit: Yu et al.

Organizing data in a specific order, also known as sorting, is a central computing operation performed by a wide range of systems. Conventional hardware systems rely on separate components to store and sort data, which limits their speed and energy efficiency.

Researchers at Peking University have recently developed a new reconfigurable sort-in-memory system that relies on memristors to in-situ sort stored data. Their proposed system, outlined in a paper published in *Nature Electronics* and led by Professor Yuchao Yang, was found to store and sort data both quickly and energy-efficiently.

"The original idea comes from the fact that although operations like matrix multiplication and convolution have been widely implemented in CIM (Computing-in-Memory) systems, sorting has long been regarded as a 'hard nut to crack' in computing-in-memory technology due to its unique computational characteristics," Yaoyu Tao, corresponding author of the paper, told TechXplore.

"Firstly, traditional sorting hardware involves extensive comparison and select logic, conditional branching, or swap operations, featuring irregular control flow that fundamentally differs from the linear operations that CIM (Compute-in-Memory) excels at. Secondly, sorting with large amounts of data often leads to highly dynamic memory access patterns that conflict with CIM's structured memory access designs."



Sorting tasks are ubiquitous in numerous applications. CPU/GPU or ASIC-based sorting systems employ massive comparison units. The performance is limited by CMOS devices and the bandwidth between memory and comparison units. Sort-near-memory alleviates the bandwidth bottleneck. Sort-in-memory based on memristor-aided logic utilizes memristors but still relies on comparison operations. Our comparison-free MSIM with TNS/CA-TNS strategies resolves the three bottlenecks. Credit: Yu et al.

Most algorithms for sorting developed to date were found to also exhibit strong data dependencies. This essentially means that the operations they perform rely on the results of earlier operations, which in turn makes them hard to upscale and reduces the advantage of CIM systems in tasks that entail performing several operations simultaneously.

"The reason sorting remains an unresolved challenge in CIM development lies in its 'unstructured' and 'control-intensive' computational nature, which inherently

conflicts with current CIM design principles centered on in-memory linear acceleration," explained Tao.

"Overcoming the implementation bottleneck of sorting in CIM would thus not only solve a critical engineering challenge but also represent a major step toward making CIM a general-purpose intelligent computing technology. The primary goal of our research is to realize 'in-situ' sorting and develop sort-in-memory techniques that are compatible with state-of-the-art compute-in-memory techniques."

The new reconfigurable sort-in-memory system developed by Tao and his colleagues as part of this recent study is made up of one-transistor-one-resistor (1T1R) memristor arrays and peripheral circuits. The peripheral circuits are divided into three distinct modules, referred to as a digit processor, a digit selector and a state controller.

"These components are reconfigurable to access the 1T1R memristor arrays and support variable data types in the literature, including unsigned, two's complement, or sign-and-magnitude fixed-point or floating-point numbers to meet the needs of real-world sorting applications," explained Tao.

"Their unique advantages/characteristics include the support for three different concurrency enhancement methodologies, multi-bank strategy for higher number-level parallelism, bit-slice strategy for higher bit-level parallelism and multi-level strategy for higher in-device parallelism."

In initial tests, the new sort-in-memory scheme devised by this team of researchers yielded highly promising results, as it required significantly less energy than previously introduced approaches to sorting data. In addition, the scheme is highly versatile and adaptable, allowing it to be integrated with various other systems and tailored to meet the demands of specific real-world problems.

"The scheme enables three parallelism enhancement strategies, including multi-bank, bit-slice and multi-level conductance, and is compatible with state-of-the-art compute-in-memory for matrix-vector multiplications," said Tao. "Sorting algorithms play a vital role in data processing by rapidly ranking massive volumes of candidate data to identify the most relevant items for further analysis.

"In scenarios like large language model training, robotic path planning, and reinforcement learning search, quickly evaluating and ranking multiple decisions or actions is both indispensable and highly time-consuming. Yet sorting operations involve non-linear operations and irregular data access patterns, and currently lack general-purpose, efficient hardware primitives for sorting."

Currently available processing-in-memory (PIM) architectures are known to have significant limitations, including an inability to efficiently sort large amounts of data. This has so far limited their deployment to a restricted number of scenarios.

The new approach developed by Tao and his colleagues could help overcome the shortcomings of these PIM architectures, thereby enhancing the efficiency with which systems store and sort data when tackling a wider range of tasks. In the future, it could be deployed in health care facilities and manufacturing sites, or it might also be used to efficiently organize scientific databases and optimize smart transportation solutions.

"We are now working on improving the sort-in-memory system and integrating it into state-of-the-art compute-in-memory systems for AI or other emerging scenarios," added Tao. "Our ultimate goal is to deploy this technology into more general hardware systems where sorting becomes a computational bottleneck."

Written for you by our author Ingrid Fadelli,edited by Gaby Clark, and fact-checked and reviewed by Andrew Zinin—thisarticle is the result of careful human work. We rely on readers like you to keep independent science journalism alive.If this reporting matters to you,please consider a donation (especially monthly).

More information: Lianfeng Yu et al, A fast and reconfigurable sort-in-memory system based on memristors, *Nature Electronics* (2025). DOI: 10.1038/s41928-025-01405-2.

A fast and reconfigurable sort-in-memory system based on memristors

Published: 25 June 2025

- Lianfeng Yu, Teng Zhang, Zeyu Wang, Xile Wang, Zelun Pan, Bowen Wang, Zhaokun Jing, Jiaxin Liu, Yuqi Li, Ziang Xie, Yihang Zhu, Bonan Yan, Yaoyu Tao & Yuchao Yang

Nature Electronics (2025)

Abstract

Sorting is a fundamental task in modern computing systems. Hardware sorters are typically based on the von Neumann architecture, and their performance is limited by the data transfer bandwidth and CMOS memory. Sort-in-memory using memristors could help overcome these limitations, but current systems still rely on comparison operations so that sorting performance remains limited. Here we describe a fast and reconfigurable sort-in-memory system that uses digit reads of one-transistor–one-resistor memristor arrays. We develop digit-read tree node skipping, which supports various data quantities and data types. We extend this approach with the multi-bank, bit-slice and multi-level strategies for cross-array tree node skipping. We experimentally show that our comparison-free sort-in-memory system can improve throughput by ×7.70, energy efficiency by ×160.4 and area efficiency by ×32.46 compared with conventional sorting systems. To illustrate the potential of the approach to solve practical sorting tasks, as well as its compatibility with other compute-in-memory schemes, we apply it to Dijkstra's shortest path search and neural network inference with in situ pruning.