Efficient Associative Processing with RTM-TCAMs

João Paulo C. de Lima[*†], Asif Ali Khan[*], Hamid Farzaneh[*], Jeronimo Castrillon[*†]
[*]Chair for Compiler Construction, Technische Universität Dresden, Dresden, Germany
[†]Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI), Dresden, Germany
{first.lastname}@tu-dresden.de

I. Introduction

Computing-in-memory (CIM) revolutionizes the traditional computing paradigm by leveraging memory cells to store data and conduct computations in-place, thereby eliminating the von Neumann bottleneck. Most emerging nonvolatile memory (NVM) technologies, including resistive RAM (RRAM), phase change memory (PCM), ferroelectric FETs (FeFETs), and magnetic racetrack memories (RTMs), are CIM-compatible and have been effectively harnessed by previous research to accelerate applications across different domains. Among the most promising CIM circuits, content-addressable memories (CAMs) [1] enable performing search operations in constant time and have received considerable interest in pattern matching and similarity search in application domains such as network security, data analysis, machine learning, and bioinformatics [2]. Nevertheless, their true potential extends beyond these specific applications, as they can be generalized to realize associative processors (APs) for data-intensive SIMD tasks [2], [3].

In the AP model, each CAM row can be viewed as one SIMD unit (see Fig.1a). To perform a boolean function with an AP, the truth table of an expression is broken down into simpler search and write primitives (Fig.1d and Fig.1e), and the generated instructions are stored in the control unit of the AP. These operations are then recursively evaluated. Naively, each evaluation step (i.e., one for each entry in the truth table) consists of a search for the input pattern (e.g., 101--) followed by a write operation of the output pattern (e.g., ---1-). However, the sequence of evaluation steps can be optimized by reducing the number of search and write operations, e.g., by aggregating multiple searches before a single write or by matching multiple input patterns in a single search operation in ternary CAMs (TCAMs)(Fig. 1e) [3].

Despite recent optimizations, the write operation remains a fundamental and frequently performed operation in APs. For APs based on NVM technologies, particularly ReRAM and PCM, this is a major challenge as the write operation in these devices is not only slow and energy-inefficient but also reduces their lifetime. This work improves upon the state-of-the-art AP model by employing racetrack memory (RTM)-based TCAM arrays. It leverages the inherent shift operations offered by RTMs for bitwise processing in APs, capitalizing on their exceptional endurance and less expensive write operations compared to other NVMs. From the RTM perspective, we demonstrate how the bit-serial processing in the AP model is an ideal use case for the inherently sequential RTMs. We also demonstrate that in certain cases, the shift operations in RTMs can eliminate the intermediate write operations in RTM-APs and further diminish the number of writes, thereby reducing AP latency and power consumption.

Observations of the AP model: First, APs support only bitwise operations and require storing data in CAM in a way that each row stores bits from different operands at the same bit position (see Fig. 1a and Fig. 1g). In the case of NVM technologies like RRAM, which can store multiple bits per cell, their multi-level cell operation is incompatible with the execution model of APs. Consequently, they need to operate in a single bit per cell mode, significantly losing their capacity.

Second, performing arithmetic operations with multi-bit numbers often involves carrying or borrowing from adjacent bits. However, implementing carry-in and carry-out signals on APs is costly, as it requires frequent writes of intermediate results to the CAM array.

Both issues are due to the limitations imposed by the underlying memory technologies. We demonstrate how to overcome them by effectively exploiting the RTM operations in the AP implementation.

II. RTM-based TCAMs for APs

Fig. 1a illustrates an AP consisting of a TCAM array, which allows comparing a query (search key) to the entire data in parallel and storing the match results in the TAG circuitry, which drives updating the data pattern in all tagged rows. Fig. 1b presents the various components of the TCAM cell incorporating a NOR-type logic circuit inspired by [1], [4].

A single cell in RTM is a magnetic nanowire that can store up to a hundred data bits and has one or more access ports, as demonstrated in Fig. 1c [5]. Accessing a particular bit in the nanowire (read or write) requires it to be shifted and aligned to the access port position. Earlier research has successfully implemented ultra-dense and low-power TCAMs using RTM devices [1], [4], [6]. We leverage these designs in our RTM-based APs. The proposed design stores operands sequentially in the nanowires (see Fig. 1g). Since RTM nanowires can store up to a hundred bits, RTM-TCAMs are significantly denser than other technologies, with the potential for achieving a hundredfold increase in density. Note that the ternary states, i.e., 0, 1, and X of the TCAM cells, are represented by two nanowires in RTM-TCAMs. However, for better readability, we show only a single nanowire.

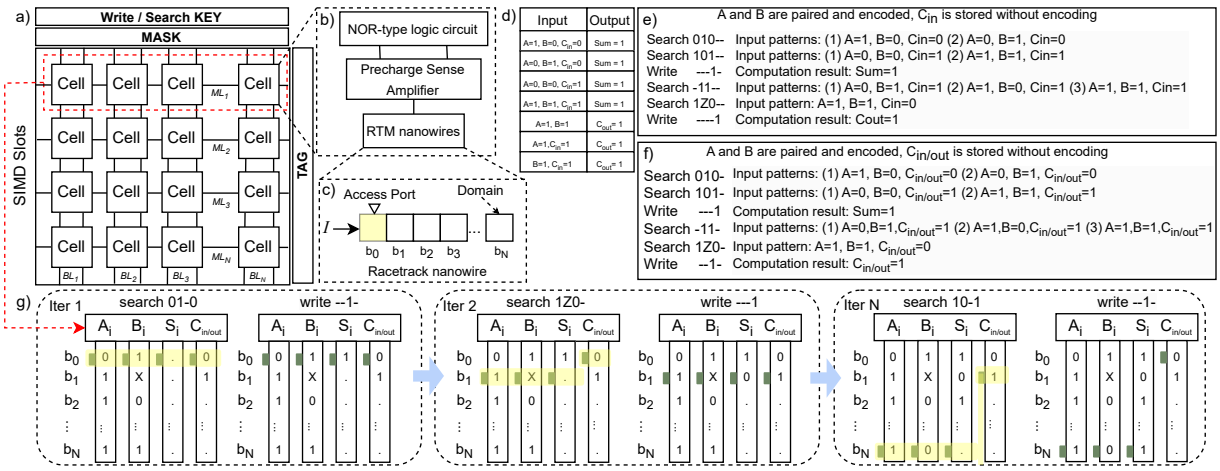

Fig. 1. a) AP organization consisting of a TCAM array and registers, b) generic TCAM cell structure using Racetrack Memories, c) racetrack device, d) compacted truth table of a 1-bit full adder, e) optimized sequence of operations in the Hyper-AP [3] for performing Single-Search-Multi-Pattern and Multi-Search-Single-Write, f) proposed sequence of operations replacing C_{in} and C_{out} with a single $C_{in/out}$ column, g) snapshot of search and write operations highlighting the shift operation in the $C_{in/out}$ column.

To explain the proposed design, let us consider a simple running example of a 1-bit full adder (see Fig. 1d). This truth table is optimized to generate the sequence of operations/instructions in Fig. 1e, which are then embedded in the AP controller. In a naive AP implementation, this would take fourteen steps, one search and one write for each truth table entry. However, with prior optimizations, e.g., in the Hyper-AP model, these operations can be reduced to only six as listed in Fig. 1f (cf. [3] for details).

Optimizations: We observe that C_{in} and C_{out} columns are subject to frequent writes and could share the same column (i.e., $C_{in/out}$) without any impact on correctness. Fig. 1f presents our optimized sequence replacing C_{in} and C_{out} with a single $C_{in/out}$ column. Similarly, $C_{in/out}$ always flips between 0 and 1 in the running example. Since the computation is performed in a bit-serial fashion, this (carry) bit is updated after a search operation at each bit position and is subsequently consumed in the search operation at the next bit position. In our proposed design, we pre-store 1 and 0 in $C_{in/out}$ column (see Fig. 1g) and adjust the access port position to the bit that needs to be read from the nanowire instead of writing it.

Fig. 1g illustrates multiple iterations on n-bit operands, where each iteration performs the sequence of six operations from Fig. 1f, occurring in parallel in all TCAM rows. In the first iteration (Iter 1), the key *01-0* matches the stored data and causes the output S_i to be set to 1 by writing *--1-*. At the end of each iteration, all RTM nanowires advance to the next bit position. However, $C_{in/out}$ nanowire does not move to the next position as it is not updated by the previous search. In the next iteration (Iter 2), the pattern *1Z0-*, in which Z matches only stored X's, matches the stored data (highlighted in yellow) and causes the $C_{in/out}$ column to be shifted from 0 to 1 by the pattern *---1*. Note that the $C_{in/out}$ column need not to be aligned to the other nanowires' domain walls to perform search operations. Similarly, in the last iteration (iter N), *10-1* matches the store data, sets 1 to the output S_i, and shifts $C_{in/out}$ to 0 to align to the access port for future computations.

III. Conclusions and Outlook

RTM-APs support storing and manipulating multiple bits per cell, distinguishing them from other NVM technologies. The capability of shifting data in nanowires is particularly well-suited for carry propagate signals in bit-wise operations, eliminating the requirement for writing intermediate results. Our preliminary results for the running example show a significant reduction by up to 2× in the write operations leading to an estimated energy reduction of over 20% (compared to a Hyper-AP using RTMs). Moving forward, we plan to conduct a comprehensive design space exploration, e.g., by changing the RTM technology, number of access ports, and bits per nanowire. Additionally, we aim to target more use cases and explore optimizations such as efficient data mapping to the TCAM arrays and instructions scheduling from higher-level abstractions like compilers and programming languages.

Acknowledgments

This work is partially funded by the German Research Council (DFG) through the CO4RTM project(450944241).

References

[1] K. P. Gnawali et al., "Low power spintronic ternary content addressable memory," *IEEE Trans. on Nanotech.*, vol. 17, no. 6, pp. 1206–1216, 2018.

[2] E. Garzón and et. al, "Am4: Mram crossbar based cam/tcam/acam/ap for in-memory computing," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 13, no. 1, pp. 408–421, 2023.

[3] Y. Zha and J. Li, "Hyper-ap: Enhancing associative processing through a full-stack optimization," in *2020 ACM/IEEE 47th Annual Intl. Symp. on Computer Architecture (ISCA)*. IEEE, 2020, pp. 846–859.

[4] Y. Zhang, J. Nan, G. Wang, X. Zhang, Y. Zhang, and W. Zhao, "Ring-shaped content addressable memory based on spin orbit torque driven chiral domain wall motions," in *2019 IEEE/ACM Intl. Symp. on Nanoscale Architectures (NANOARCH)*, 2019, pp. 1–2.

[5] R. Bläsing et al., "Magnetic racetrack memory: From physics to the cusp of applications within a decade," *Proceedings of the IEEE*, vol. 108, no. 8, pp. 1303–1321, 2020.

[6] P. Junsangsri, J. Han, and F. Lombardi, "A non-volatile low-power tcam design using racetrack memories," in *2016 IEEE 16th Intl. Conf. on Nanotechnology (IEEE-NANO)*. IEEE, 2016, pp. 525–528.