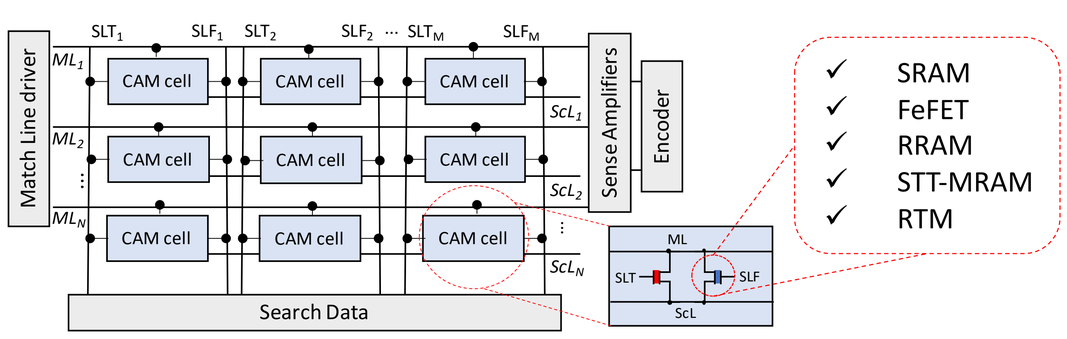
# Efficient Associative Processing with RTM-TCAMs

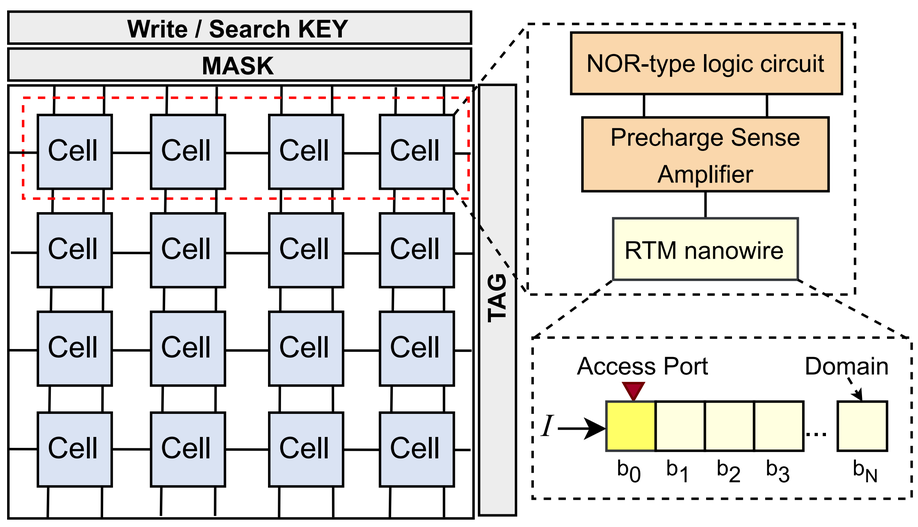
Contact: Joao Paulo C. de Lima (joao.lima@tu-dresden.de), Asif Ali Khan (asif_ali.khan@tu-dresden.de), Hamid Farzaneh (hamid.farzaneh@tu-dresden.de), Jeronimo Castrillon (jeronimo.castrillon@tu-dresden.de)

## Motivation

- Content-Addressable Memories (CAMs) perform search in constant time
- CAMs generalize to Associative Processors (APs) for data-intensive SIMD tasks
- CAM can be multi-bit or analog, implemented with CMOS/NVMs techs.
- Racetrack Memories (RTMs) are exceptionally well-suited for bit-serial processing in the AP model




## Associative Processors (AP) and Observations




- CIM paradigm for arbitrary computations in word-parallel and bit-serial manner
- Boolean functions are broken down into ***search*** and ***write*** operations
- The ***TAG*** register stores the match results for updating the data pattern in all tagged rows in the next write operation

1-bit full adder (FA) example: truth table and sequence of search-write operations

| A | B | $C_{in}$ | S | $C_{out}$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

A and B are stored in the same row

```
Search 001--  Input pattern: A=0, B=0, Cin=1
Search 010--  Input pattern: A=0, B=1, Cin=0
Search 100--  Input pattern: A=1, B=0, Cin=0
Search 111--  Input pattern: A=1, B=1, Cin=1
Write  ---1-  Computation result: Sum=1
Search -11--  Input pattern:  B=1, Cin=1
Search 11---  Input pattern: A=1, B=1
Search 1-1--  Input pattern: A=1, Cin=1
Write  ----1  Computation result: Cout=1
```

9 operations

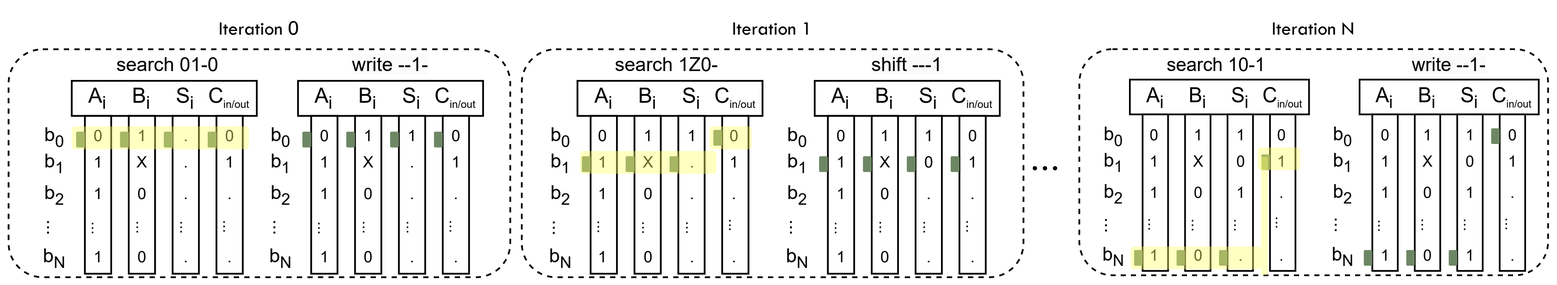
- Hyper-AP [1] enables Single-Search-Multi-Pattern and Multi-Search-Single-Write, reducing the1-bit FA to 6 operations

**Observations**

- APs are inherently bitwise and each CAM row stores bits from different operands at the same bit position
- Carrying or borrowing from adjacent bits require frequent writes of intermediate results to the CAM

## Racetrack Memory-based AP optimizations and preliminary results

- $C_{in}$ and $C_{out}$ columns can share the same column (i.e., $C_{in/out}$) without any impact on correctness
- Since $C_{in/out}$ does not need to be permanently stored, the access port can be positioned at pre-stored 0 or 1 bits

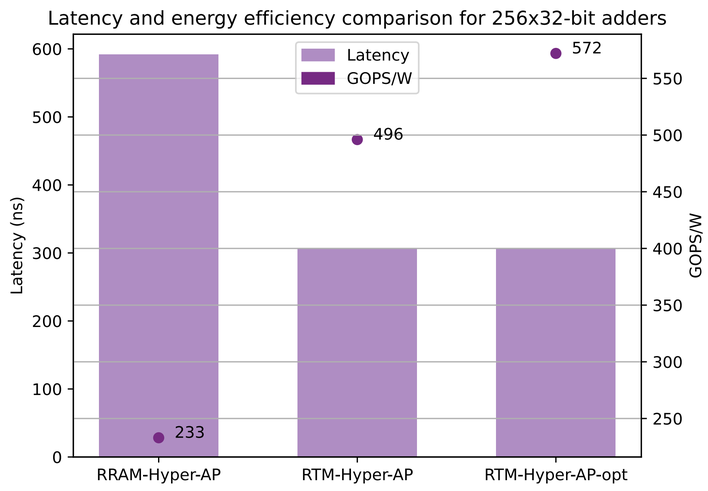



Optimized sequence of search-write operations

A and B are paired and encoded as in Hyper-AP [1]

```
Search 01-0   Input patterns: (1) A=1, B=0, Cin/out=0 (2) A=0, B=1, Cin/out=0
Search 10-1   Input patterns: (1) A=0, B=0, Cin/out=1 (2) A=1, B=1, Cin/out=1
Write  --1-   Computation result: Sum=1
Search -11-   Input patterns: (1) A=0,B=1,Cin/out=1 (2) A=1,B=0,Cin/out=1
              (3) A=1,B=1,Cin/out=1
Search 1Z-0   Input pattern: A=1, B=1, Cin/out=0
Shift  ---1   Computation result: Cin/out=1
```

** Z matches only "X" and "-" bit column is not selected for search or write*




- RTM-AP effectively exploits the inherent multi-bit RTM cells compared to other NVM-APs
- It significantly *reduces the number of bits for $C_{in}/C_{out}$ signals* and *eliminate write operations for $C_{in/out}$ signals* by using less-costly shift operations
- **Future work:** Efficient data mapping to the CAM arrays and supporting complex application kernels

1. Y. Zha & J. Li. Hyper-AP: Enhancing associative processing through a full-stack optimization. In: ISCA, 2020.
2. K. P. Gnawali, S. N. Mozaffari & S. Tragoudas, "Low power spintronic ternary content addressable memory ". In IEEE TNANO, 2018.
3. P. Junsangsri, J. Han & F. Lombardi, "A non-volatile low-power TCAM design using racetrack memories". In IEEE-NANO, 2016.