CHAIR FOR COMPILER CONSTRUCTION

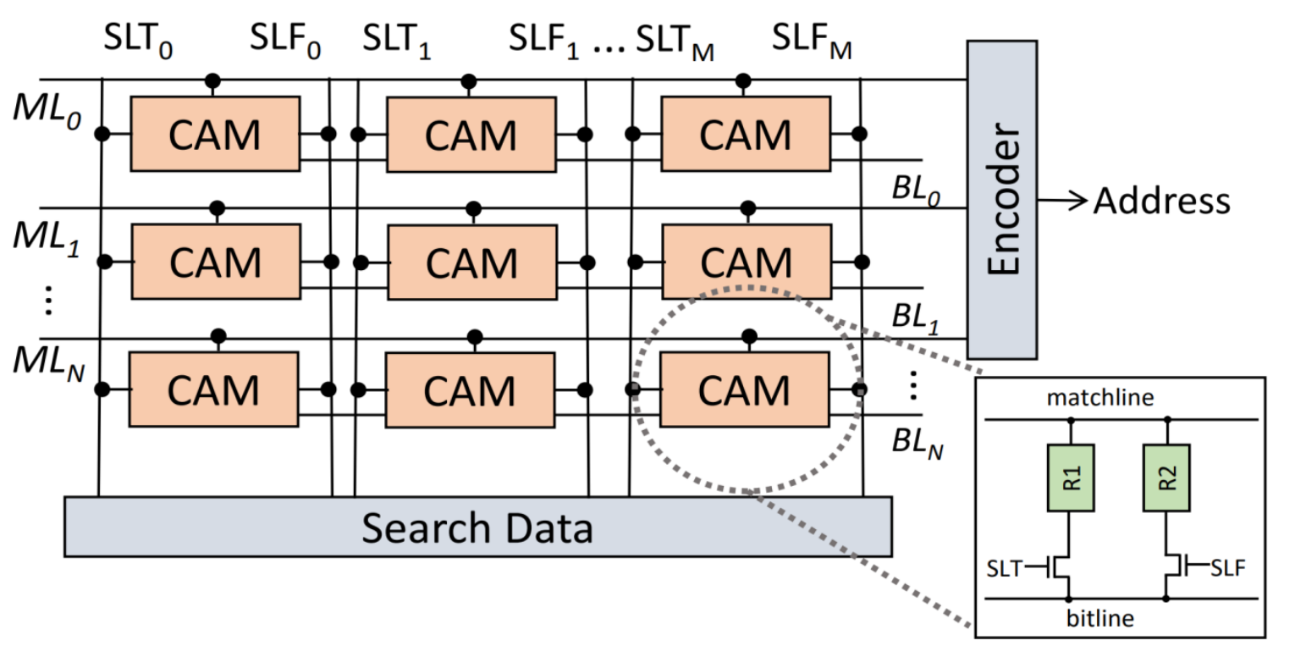
cfaed CENTER FOR ADVANCING ELECTRONICS DRESDEN

Architecture Optimization and Design Tools for CAM-based Accelerators

João Paulo C. de Lima (joao.lima@tu-dresden.de), Jeronimo Castrillon and Luigi Carro

Content-Addressable Memories

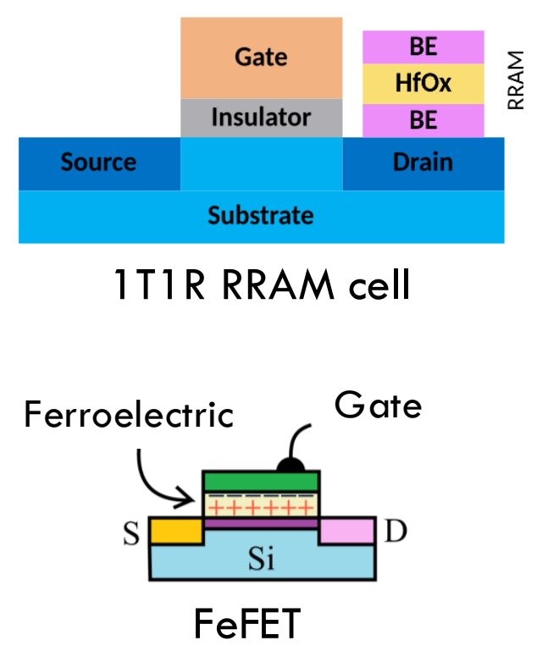
- ❑ CAM offers cross-memory lookup and pattern-matching acceleration in constant time O(1)
- ❑ In computing-in-memory, CAMs operate in various ways based on their cell precision, periphery circuitry, match type and external merging circuits
- ❑ Boolean function, Hamming distance, branch-split operation, automata, associative memory, …



Stored data	R1	R2
0	LRS	HRS
1	HRS	LRS
x	HRS	HRS

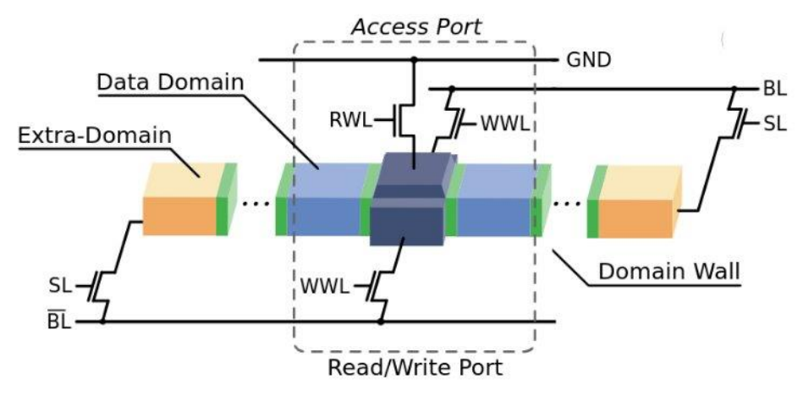
Search data	SLT	SLF
0	0	V_{search}
1	V_{search}	0
x	0	0

Search	Stored	Result
0	0	Match
0	1	Miss
1	0	Miss
1	1	Match
x	Any	Match
Any	x	Match



1T1R RRAM cell

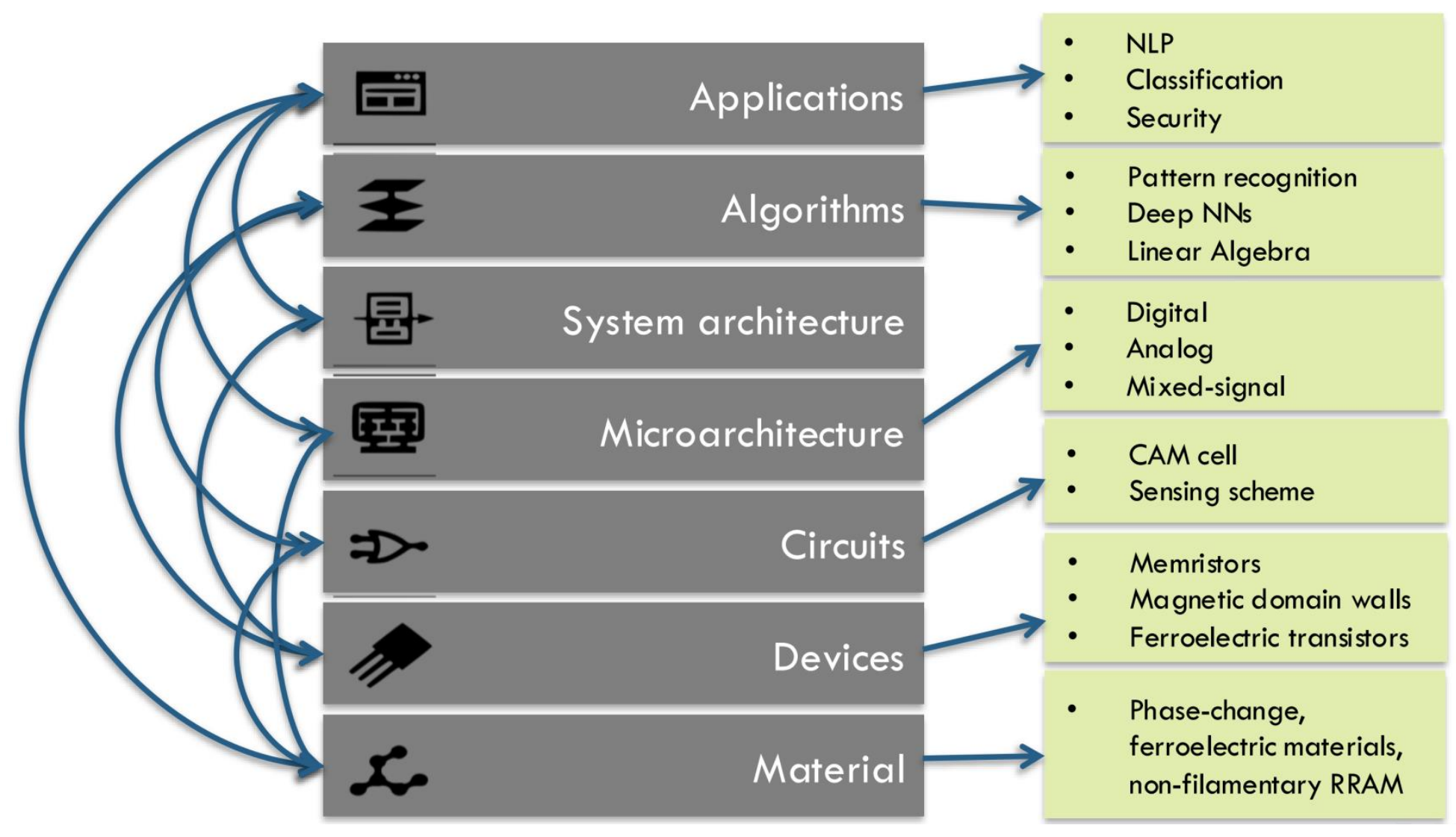
FeFET



Magnetic domain wall devices (aka Racetrack memories)

Cross-layer Stack Design

- ❑ An omnidirectional co-design approach is needed as aspects of one the components influence others directly
- ❑ Focus on three general-purpose computing models



- NLP
- Classification
- Security

- Pattern recognition
- Deep NNs
- Linear Algebra

- Digital
- Analog
- Mixed-signal

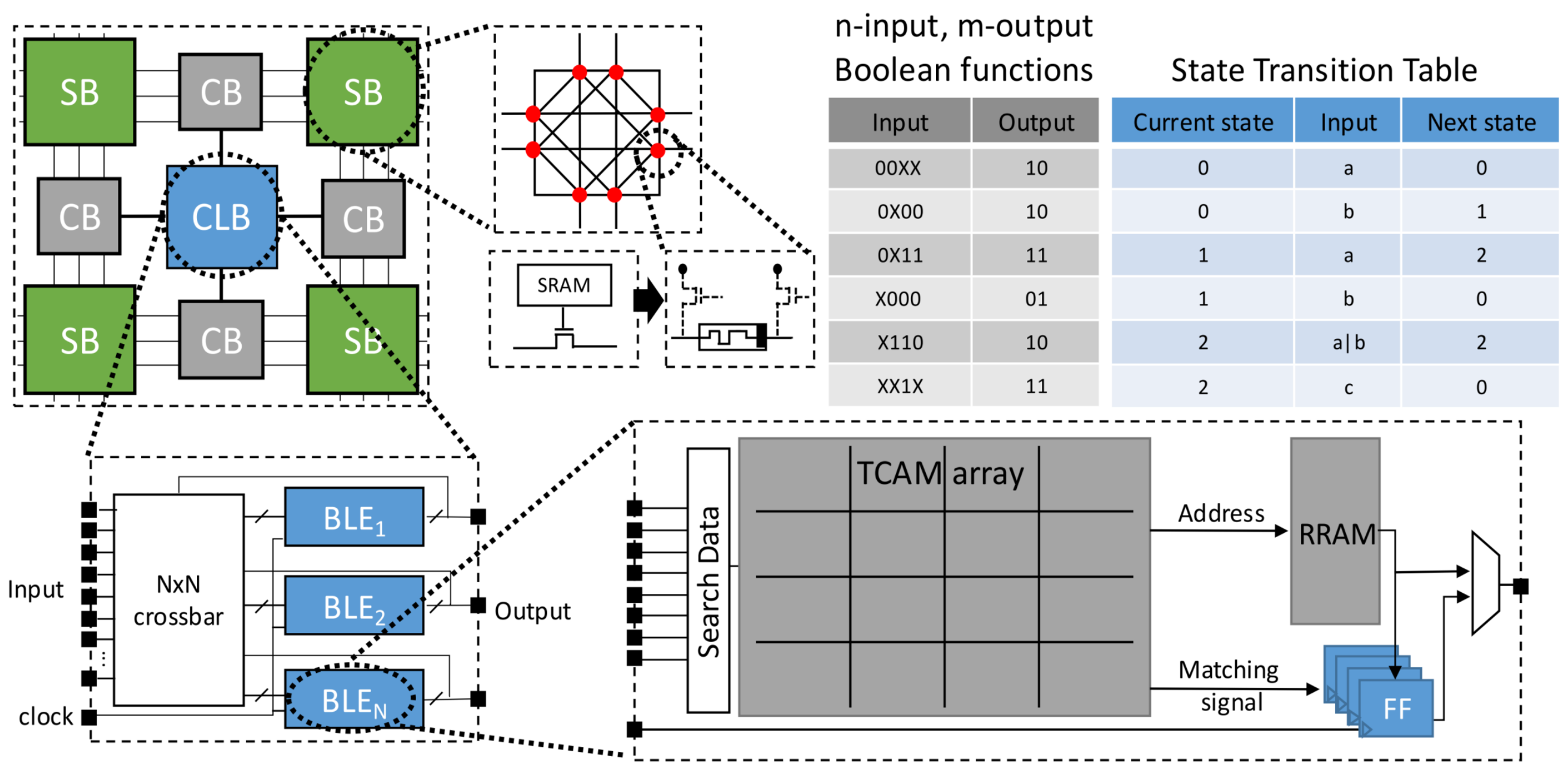
- CAM cell
- Sensing scheme

- Memristors
- Magnetic domain walls
- Ferroelectric transistors

- Phase-change, ferroelectric materials, non-filamentary RRAM

CAMs as Configurable Logic Blocks

- ❑ A reconfigurable fabric using TCAMs and RRAM-based interconnect
- ❑ Enables large input functions, often achieving higher density than LUTs
- ❑ Leverages reuse of reconfiguration data for endurance-aware flow



n-input, m-output Boolean functions

Input	Output
00XX	10
0X00	10
0X11	11
X000	01
X110	10
XX1X	11

State Transition Table

Current state	Input	Next state
0	a	0
0	b	1
1	a	2
1	b	0
2	a|b	2
2	c	0

$$\sigma(x_1, x_2, x_3) = \bar{x}_1 * x_2 * \bar{x}_3 + x_1 * \bar{x}_2 * x_3 + x_1 * x_2 * x_3$$

$x_1\ x_2\ x_3$	z
000	0
001	0
010	1
011	0
100	0
101	1
110	0
111	1

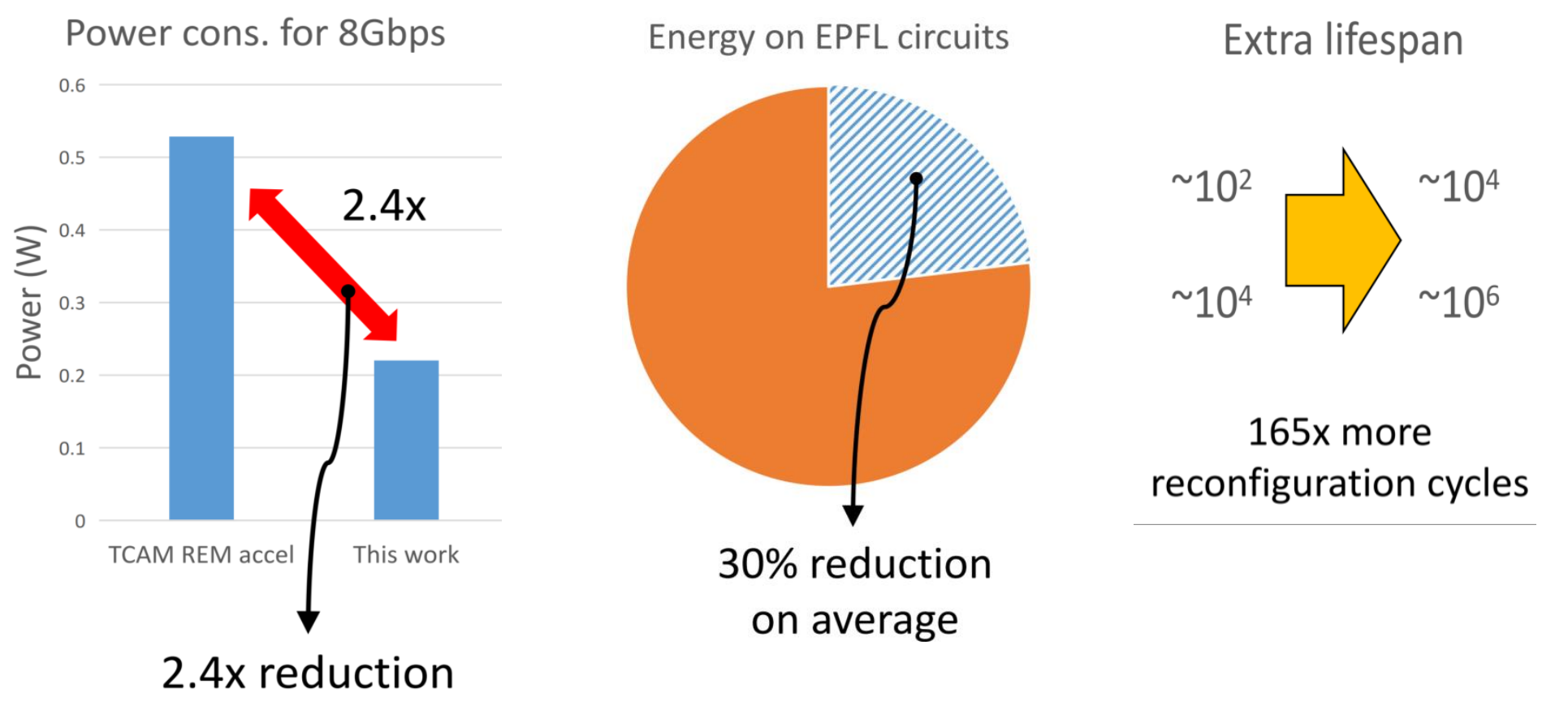
LUT = 0x25

$x_1\ x_2\ x_3$
010
101
111

BCAM

$x_1\ x_2\ x_3$
010
1X1

TCAM



2.4x reduction

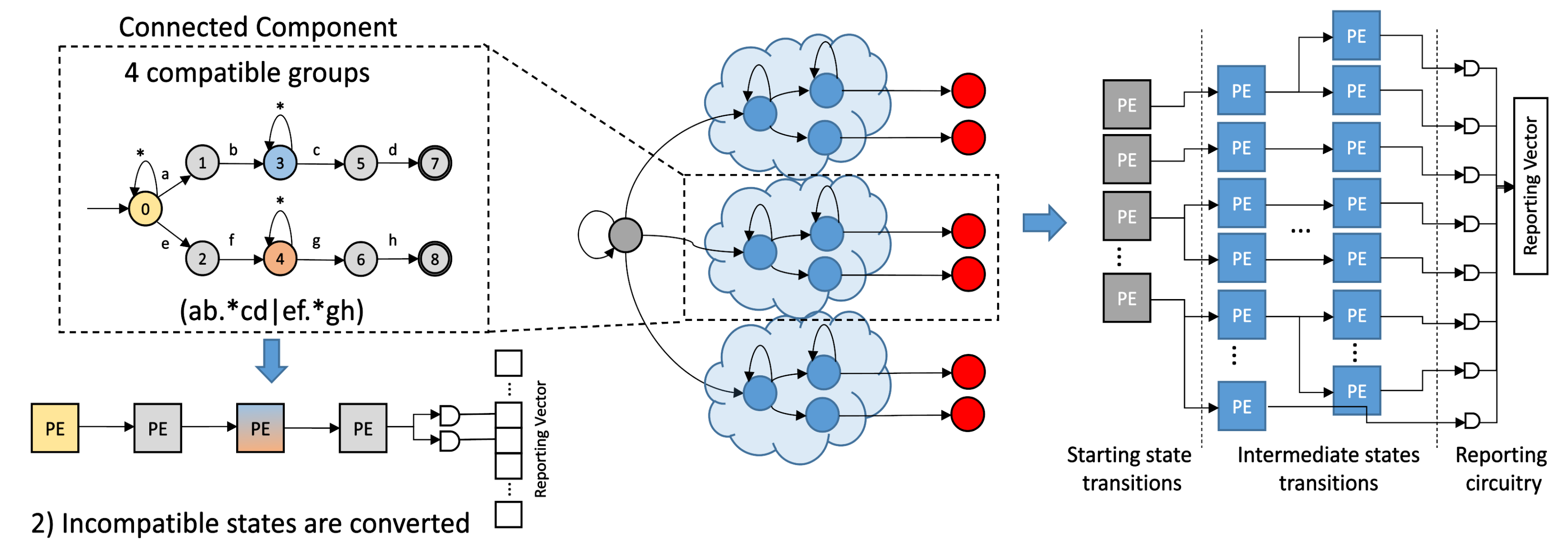
30% reduction on average

165x more reconfiguration cycles

Lima, J.P.C., et al. "Endurance-aware RRAM-based Reconfigurable Architecture using TCAM Arrays", FPL'20.

Scalable NFA-based Pattern Matching

- ❑ STAP: a memristive Scalable TCAM-based Automata Processor for NFA processing without exponential memory space
- ❑ Requires less memory to represent state matching and state transition compared to traditional Micron's AP model



Starting state transitions

Intermediate states transitions

Reporting circuitry

2) Incompatible states are converted to allow only one active state at a time

1) An automaton is divided into Connected Components

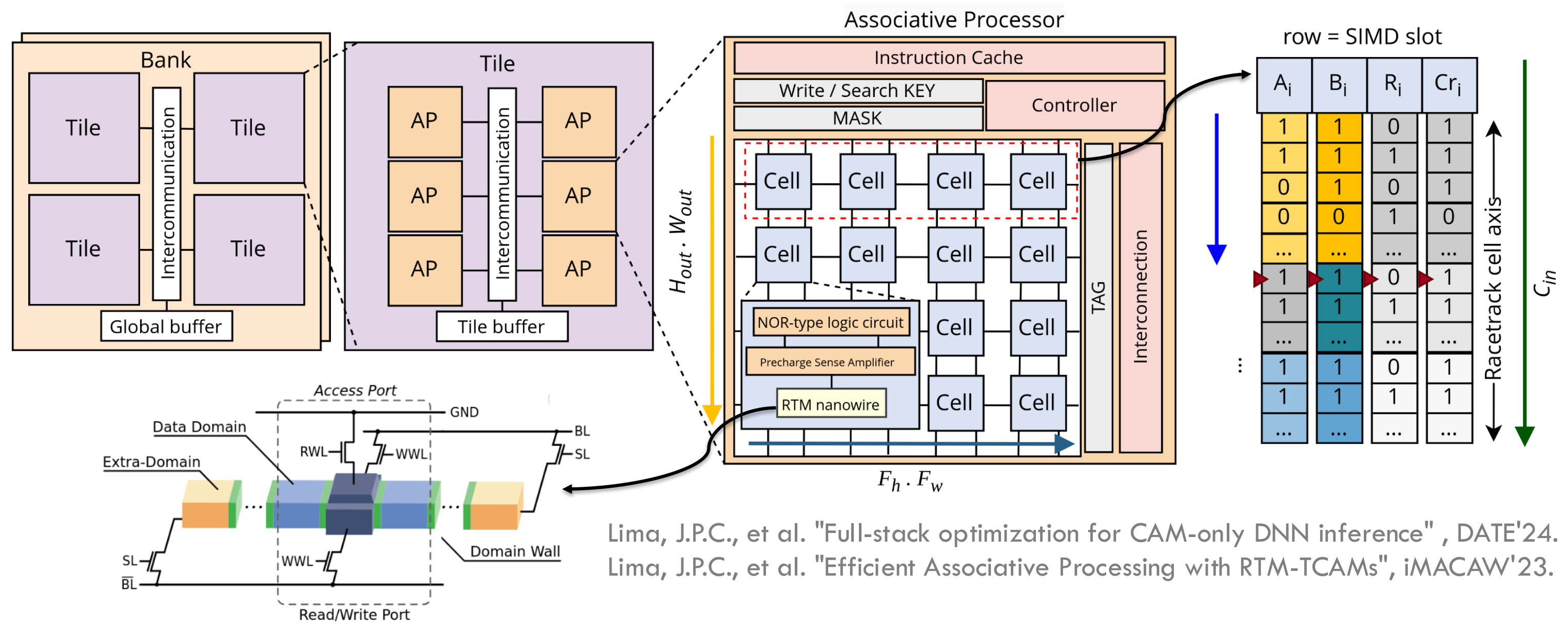
3) CCs are greedly assigned to PEs to increase resource utilization

Designs	Throughput (Gbps)	Power (W)	Area (mm²)	State density (cell/state)	Endurance cycles
Grapefruit (FPGA)	6.9	5.3	n/a	n/a	10^{16}
eAP (DRAM)	20.0	29.6	5.4	2346	10^{16}
RRAM-AP	24.0	0.6	3.1	2346	~50
16-bit STAP	35.2	6.9	19.6	1694	10^4

Lima, J.P.C., et al. "STAP: An architecture and design tool for automata processing on memristor TCAMs", JETC'21.

In-memory general-purpose SIMD processing

- ❑ Associative Processors iteratively evaluate LUTs and update data in place
- ❑ Same search/write pattern is applied simultaneously to all CAM rows
- ❑ RTM-CAMs offer multi-level capability needed for bitwise processing

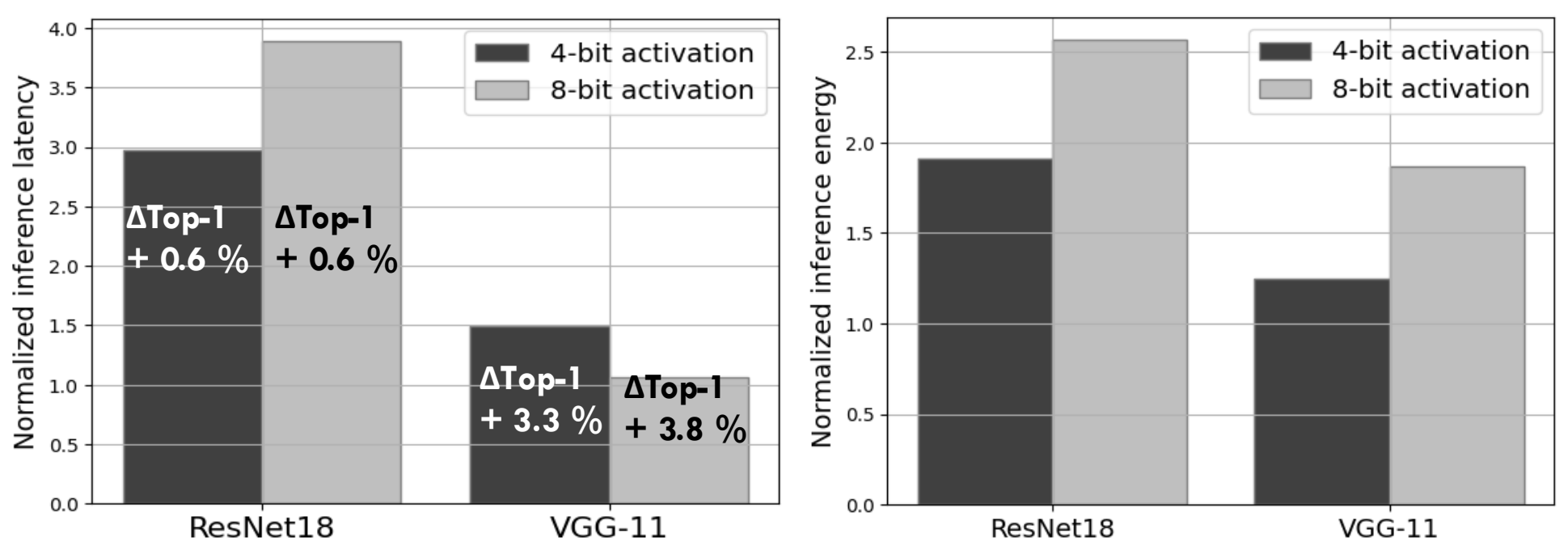


Lima, J.P.C., et al. "Full-stack optimization for CAM-only DNN inference" , DATE'24.
Lima, J.P.C., et al. "Efficient Associative Processing with RTM-TCAMs", iMACAW'23.

Comparison with XBar-based accelerators:

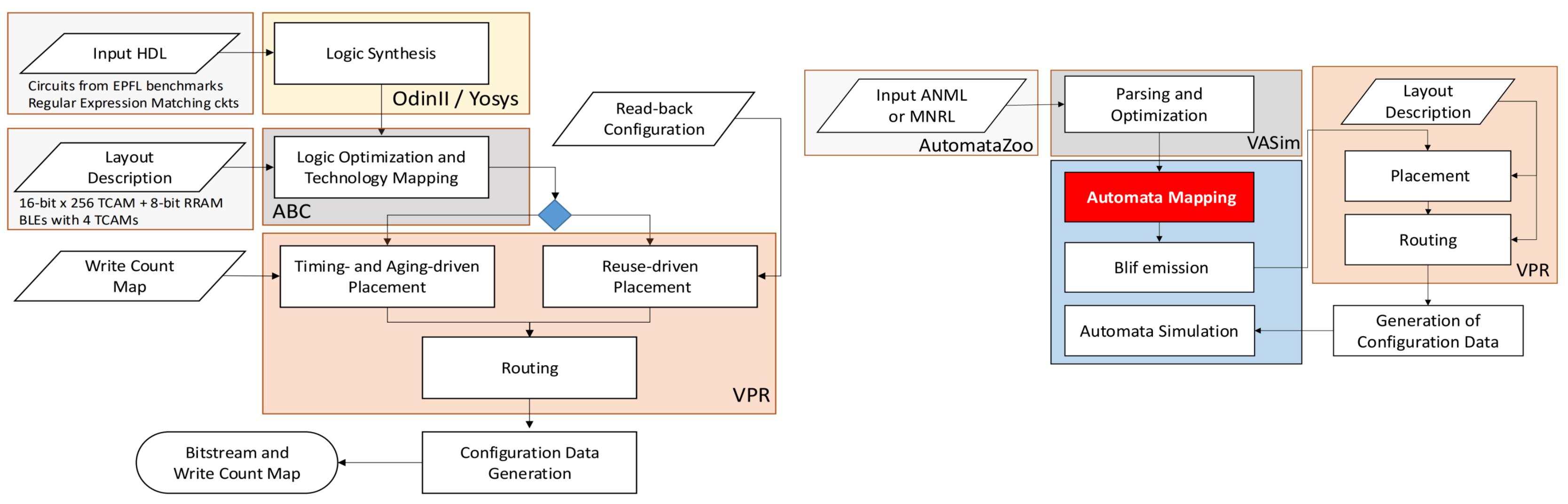
Data movement for partial sums

- RTM-based CAM: 3% total energy
- RRAM Xbar-based accelerators: 40%



Programmability and Tool Suite

- ❑ Tools for spatial CAM-based architectures are built upon existing tools designed for FPGAs and Micron's AP



Input HDL
Circuits from EPFL benchmarks
Regular Expression Matching ckts
Logic Synthesis
OdinII / Yosys
Layout Description
16-bit x 256 TCAM + 8-bit RRAM BLEs with 4 TCAMs
Logic Optimization and Technology Mapping
ABC
Read-back Configuration
Write Count Map
Timing- and Aging-driven Placement
Reuse-driven Placement
Routing
VPR
Configuration Data Generation
Bitstream and Write Count Map
Input ANML or MNRL
AutomataZoo
Parsing and Optimization
VASim
Automata Mapping
Blif emission
Automata Simulation
Layout Description
Placement
Routing
VPR
Generation of Configuration Data

- ❑ C4CAM[1] advances mapping of some code patterns onto CAM primitives
- ❑ Hybrid approach can combine all CAM computing models to implement multiple kernels in more complex applications
- ❑ Making strides on matching technology, circuits, architecture, algorithms and optimizations for efficient CAM-based accelerator design

[1] Farzaneh, H., Lima, J.P.C., et al. "C4CAM: A Compiler for CAM-based In-memory Accelerators", ASPLOS'24.

cfaed.tu-dresden.de