The Bitlet Model:
Defining a Litmus Test for the Bitwise Processing-in-Memory Paradigm

Kunal Korgaonkar, Ronny Ronen, Anupam Chattopadhyay and Shahar Kvatinsky

Postdoc at Technion (PhD, UC San Diego)

Technion, UC San Diego and NTU
Dec 16, 2019
CATC, Haifa, 2019

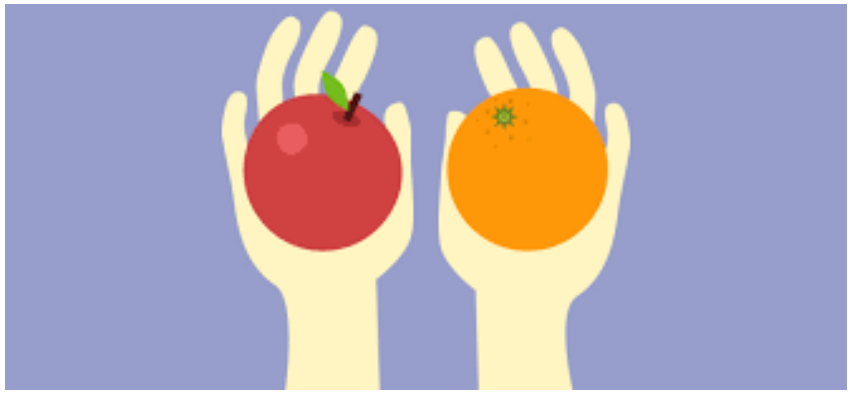
ASIC²
ARCHITECTURES
SYSTEMS
INTELLIGENT COMPUTING
INTEGRATED CIRCUITS

TECHNION
Israel Institute of Technology

Bitlet Overview

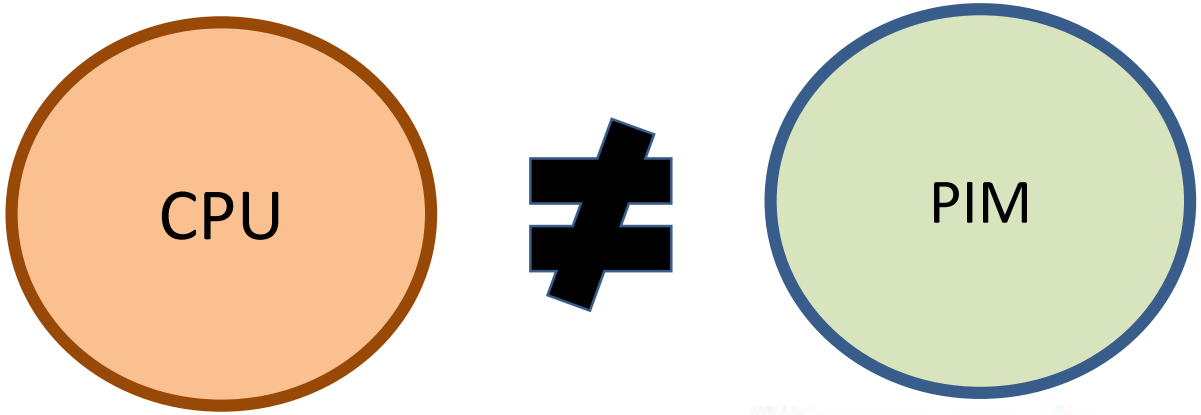
What is Bitlet?

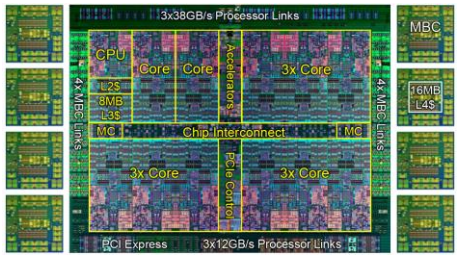
- A model for high-level performance and power comparison between **Processing -in-Memory (PIM)** and **CPU**
- Provide a simple operational view of PIM and CPU for relevant workloads
- Highlight PIM performance and efficiency limits



ASIC²

Bitlet Model: Motivation





Requirements:

- Requests to compare PIM vs. CPU performance and power
- Look for real tradeoffs/sweet-spot as opposed to just looking for winning configurations
- Look for simple, flexible and easily to use solution

ASIC²

Bitlet Model: Motivation

- OOO complexity
- Execution unit diversity
- Memory hierarchy differences

ASIC²

Bitlet Model: Motivation

- OOO complexity
- Execution unit diversity
- Memory hierarchy differences

- Non-CMOS execution units
- System software different
- Parallelism is higher but with constraints

ASIC²

Bitlet Model: Motivation

What we have accomplished:

- **Adopted an analytical approach (e.g. Roofline)**
- **Reduced the overall model to just "6 parameters" instead of 1000s**

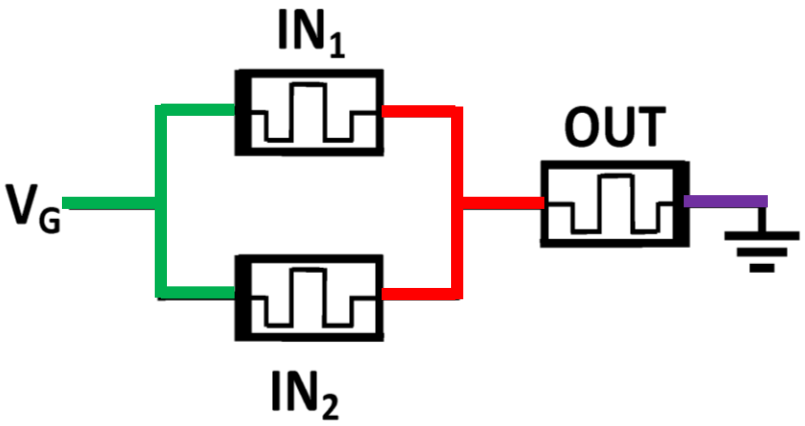
Roofline: an insightful visual performance model for multicore architectures, David Patterson et al., CACM 2009

Outline

- Goals
- **From Memristor to Processing-In-Memory**
- The Bitlet Model
  - Performance
  - Power Limited Performance
- Comments & Conclusions

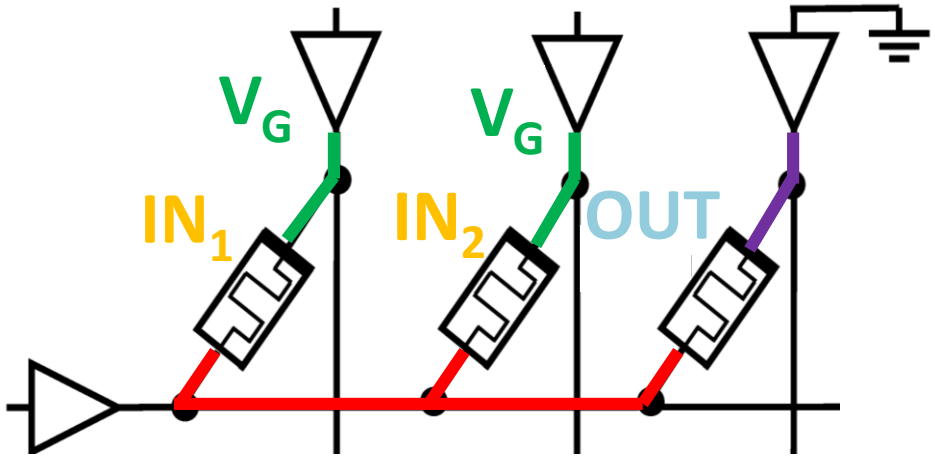
Memristor to Processing-in-Memory

MAGIC GATE (e.g. in-memory logic family)



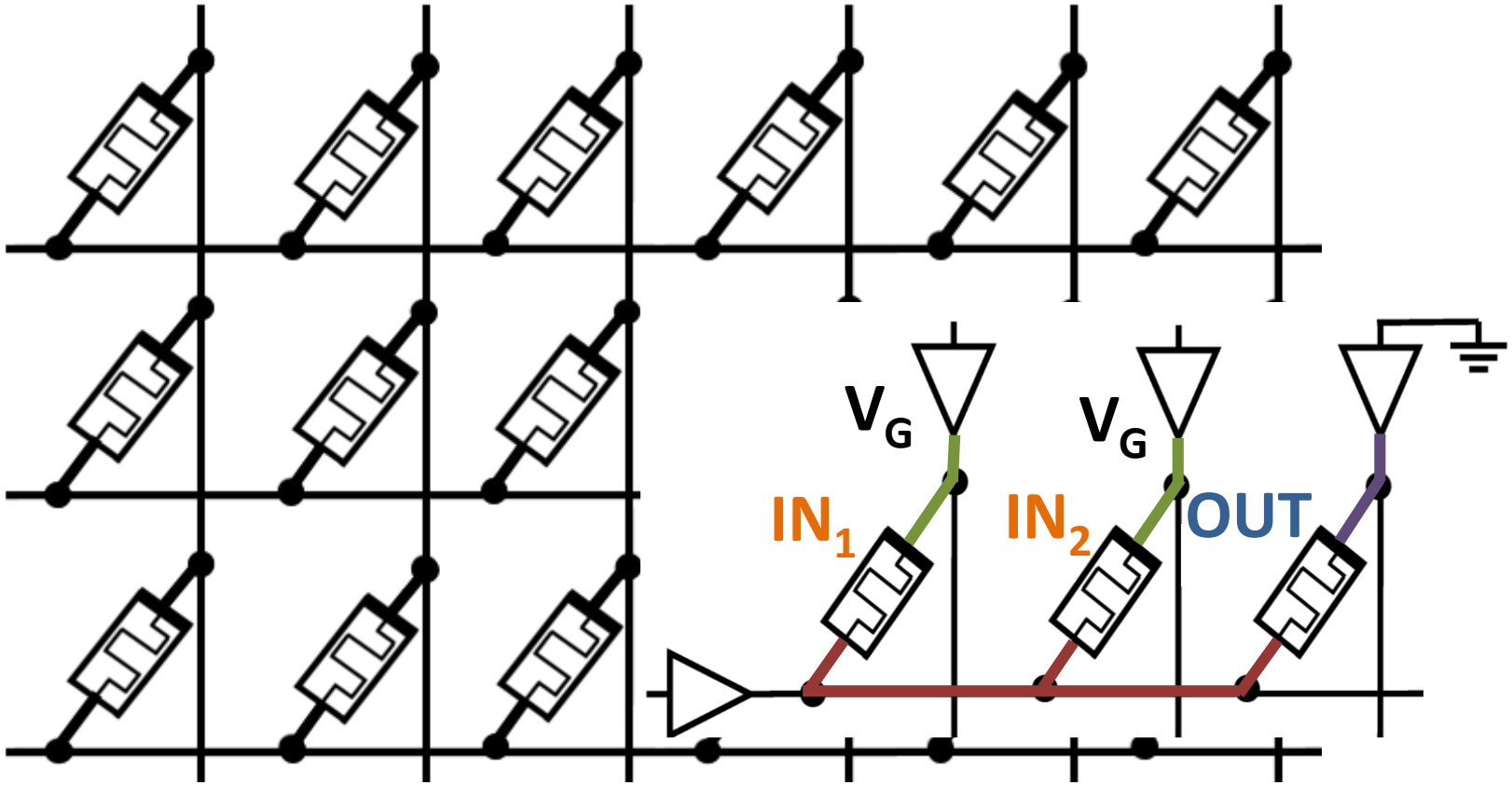
Functionally complete

Crossbar Compatible

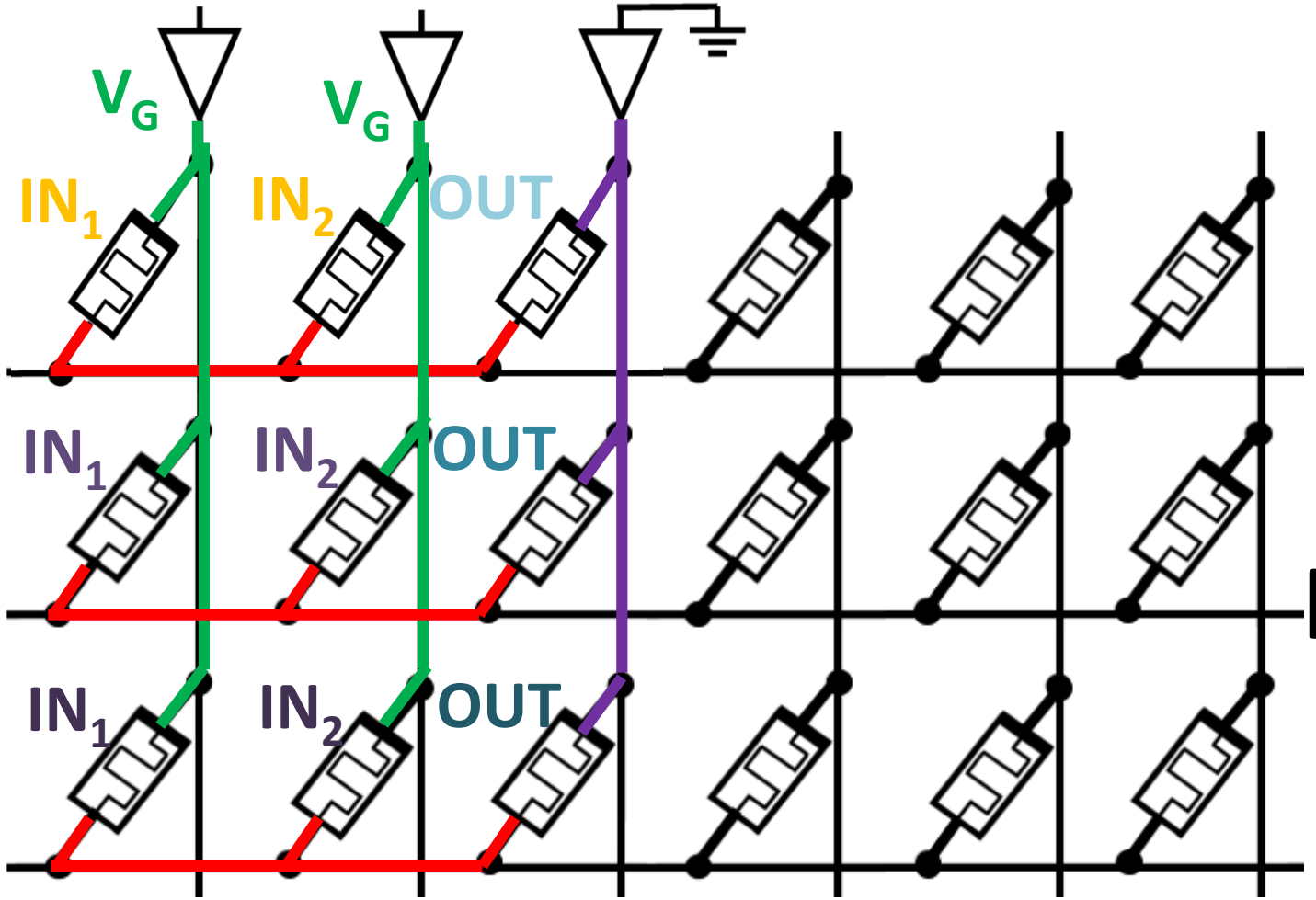


ASIC²

Logic Gate in a Memristive Memory Array



Parallel Execution of Logic Gates



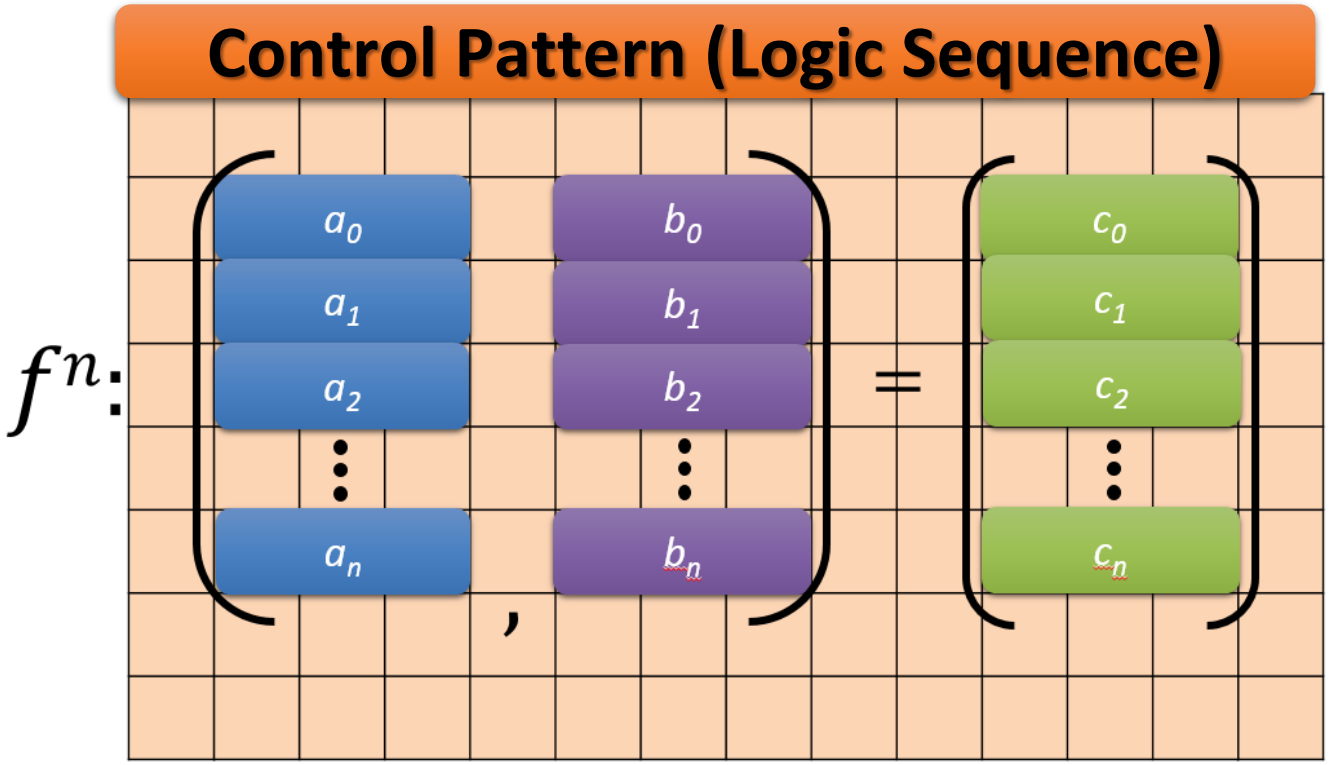
Efficient SIMD Realization

N. Talati *et al.*, "Logic Design within Memristive Memories using Memristor-Aided loGIC (MAGIC)," *TNANO*, 2016

Row-Based Computing: Throughput Improvement

- Execution of a single instance in each row in parallel
- Implement desired function using sequence of basic logic gate level operations

$$Throughput = \frac{\#instances}{Latency}$$



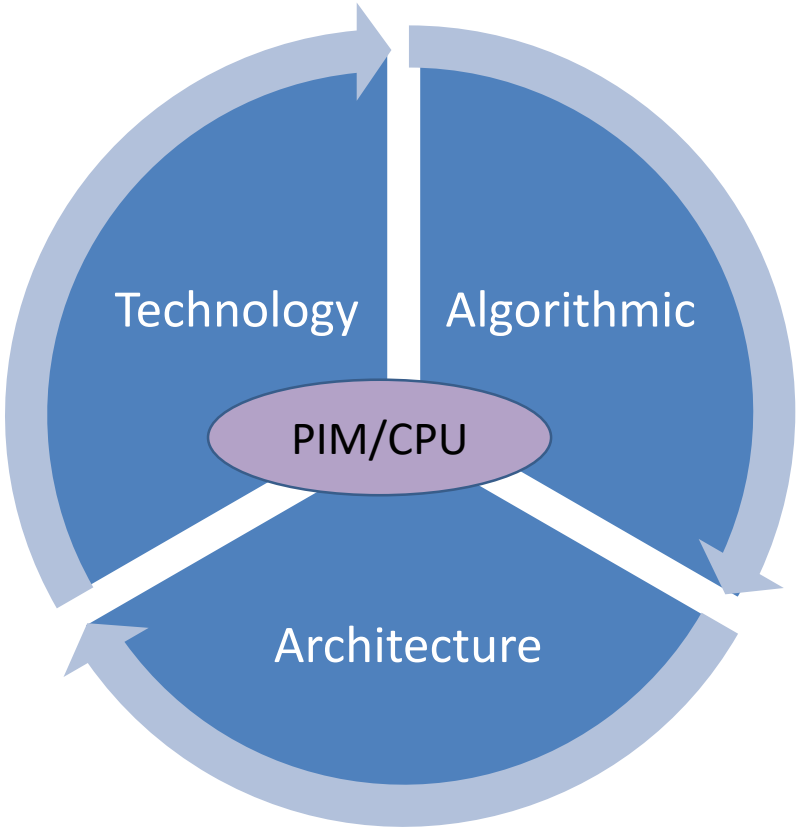
ASIC²

Outline

- Goals
- From Memristor to Processing-In-Memory
- **The Bitlet Model**
  - **Performance**
  - **Power Limited Performance**
- Comments & Conclusions

Implementation parameters: the three `forces'

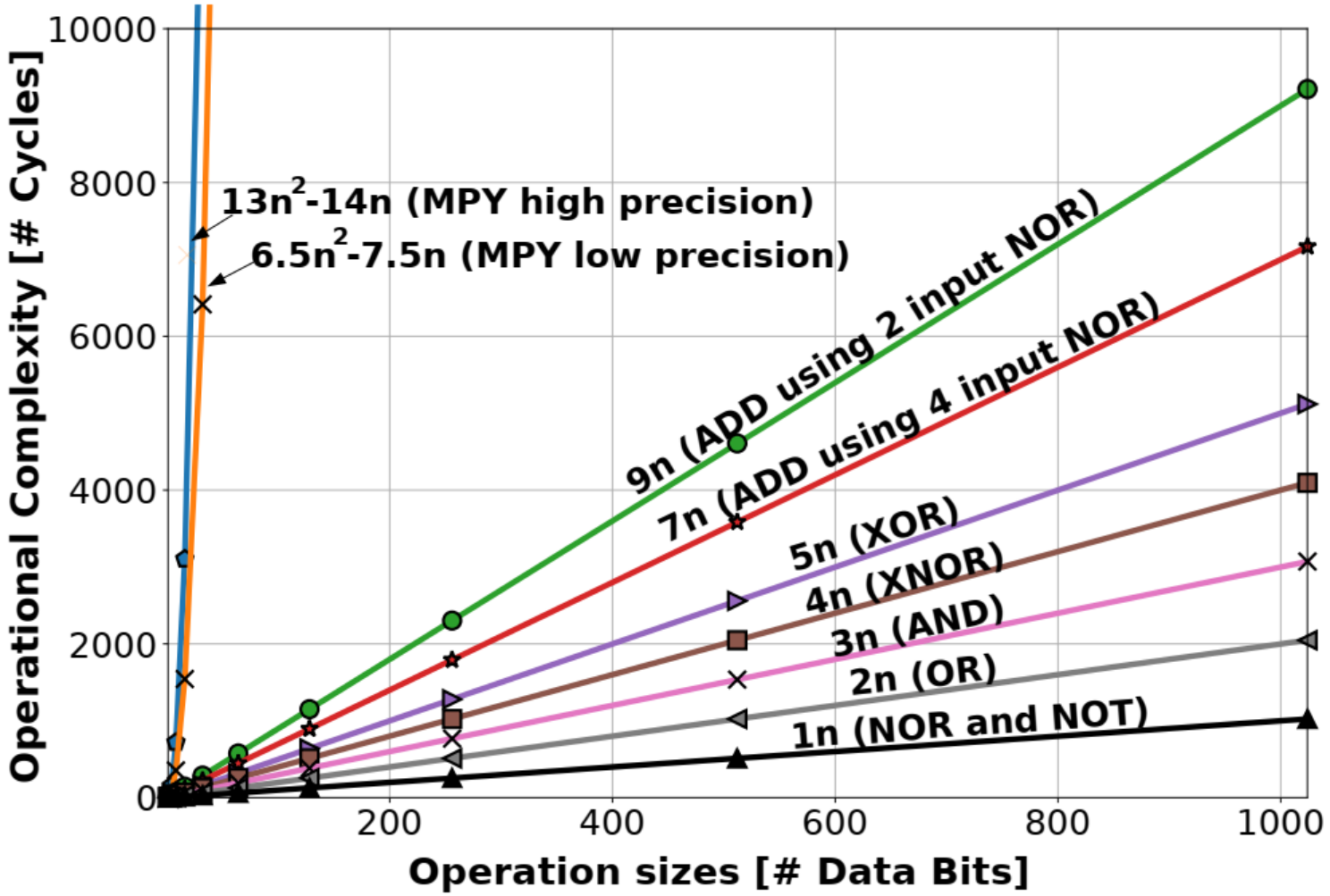
- Technology
  - Cycle time (10 ns)
  - Density of Memories (1 TB, 1Kx1K bits)
- Algorithmic
  - Logic synthesis (assuming logic seq.)
  - Code compilation
  - Data movements
- Architecture
  - Memory interfaces
  - Memory bandwidth



ASIC^2

PIM Performance: Operation Complexity (OC)

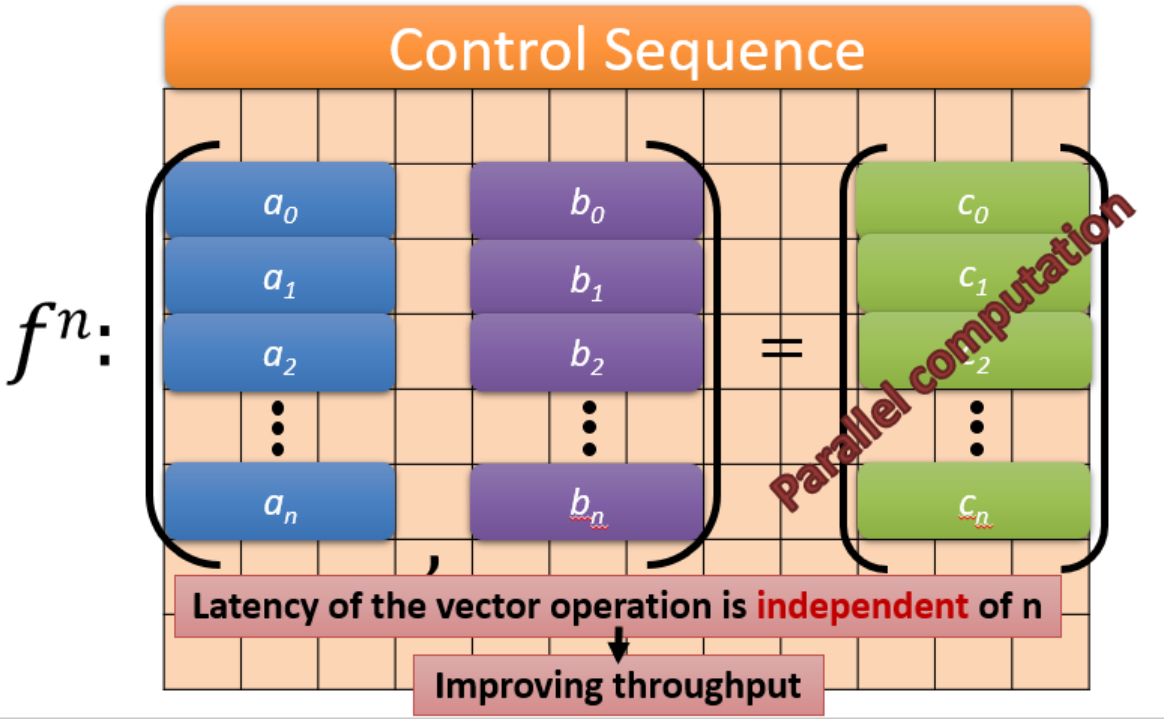
- **Definition**:
  OC: Number of basic gates (e.g. NOR gate) needed to implement a computation
- PIM performance highly (inversely) depends on OC.
- Examples for OC:
  - N-bit full adder: 9*N
  - N-bit multiply: $13*N^2-14*N$
- In most simple operation, OC per bit is independent of N



ASIC²

PIM Performance: Parallelism

- A Memory Array (MAT) consists of Rows and Columns
- All Rows in a MAT can compute in parallel (e.g., 256, 1024 rows)
- All MATs in system can compute in parallel (e.g., 1K-32K MATs)
  - 1GB Memory can contain 8K MATs (1024x1024) or 32K (512x512)



PIM Performance: Parameters

PIM-related parameters of the Bitlet model

Parameter name	Notation	Value(s)	Type
Operation complexity	OC	1 - 32k cycles	Algorithmic
PIM cycle time	CT	10 ns	Technological
Memory array count	MAT	1k - 16k	Architectural
Memory array dimensions	ROW × COL	1024 x 1024	Technological

The number of parallel computations /

latency of single computation

Throughput-PIM(Op) = (ROWs*MATs)/(OC*CT)

ASIC²

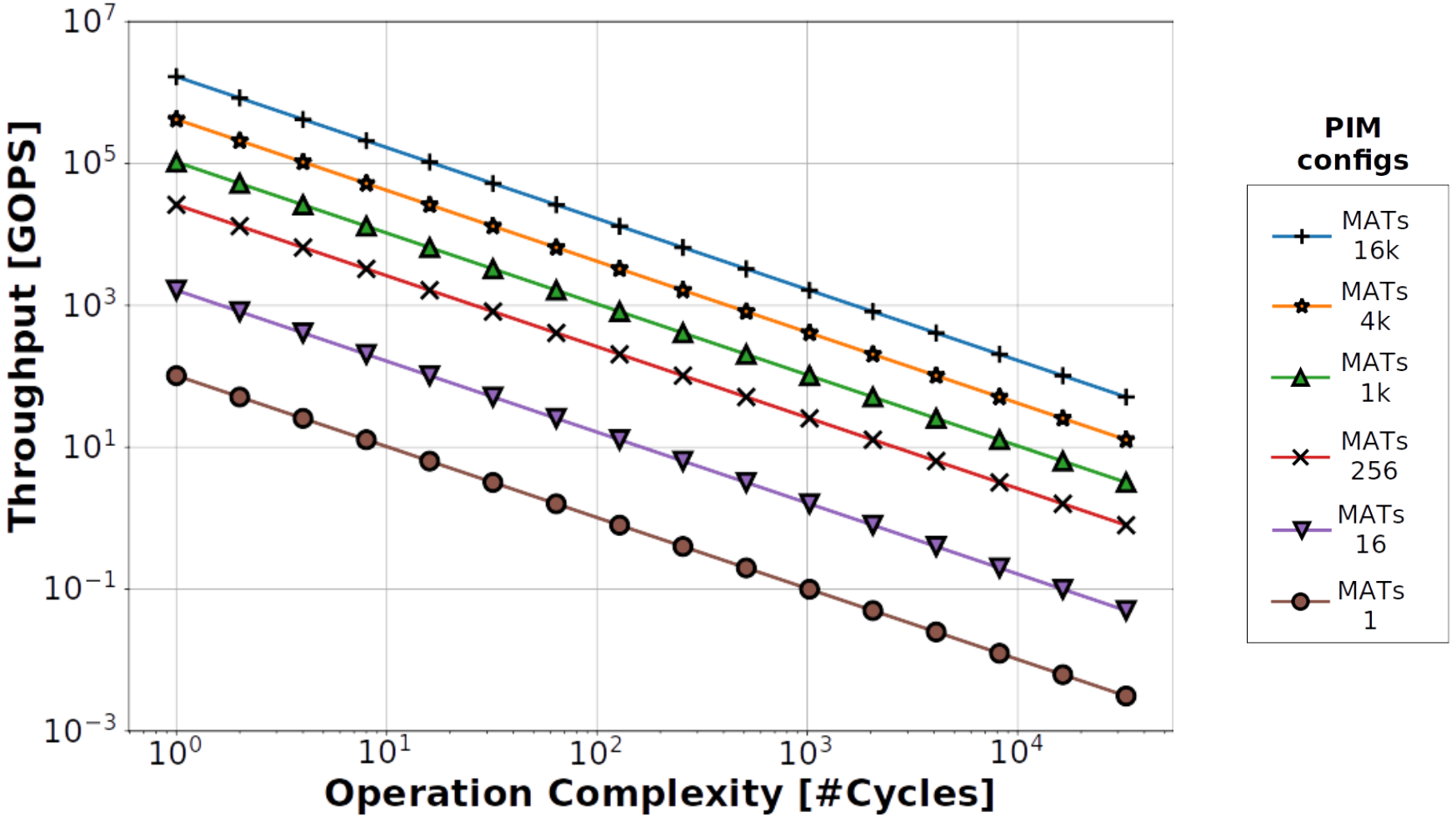
PIM Performance: Trends

- Varying Parameters:
  - **OC**
  - **#MATs**

Fixed parameters:

- **CT: 10ns;**
- **ROWs: 1024**

PIM improves when # MATs increases and OC decreases



CPU Performance: Parameters

CPU-related parameters of the Bitlet model

Parameter name	Notation	Value(s)	Type
Memory bandwidth	BW	1Tbps to 16Tbps	Architectural
Data in-out bits	DIO	24, 48	Algorithmic

- Assumption: Memory Bandwidth (BW) is the **only** bottleneck for CPU performance
  - On the other extreme, computation is so dense, that PIM cannot likely compete
- CPU computation throughput is (inversely) depends on Data in-out (DIO):
  - The number of bits transferred for each element (Data in-out: DIO).

Bandwidth / Data-per-element

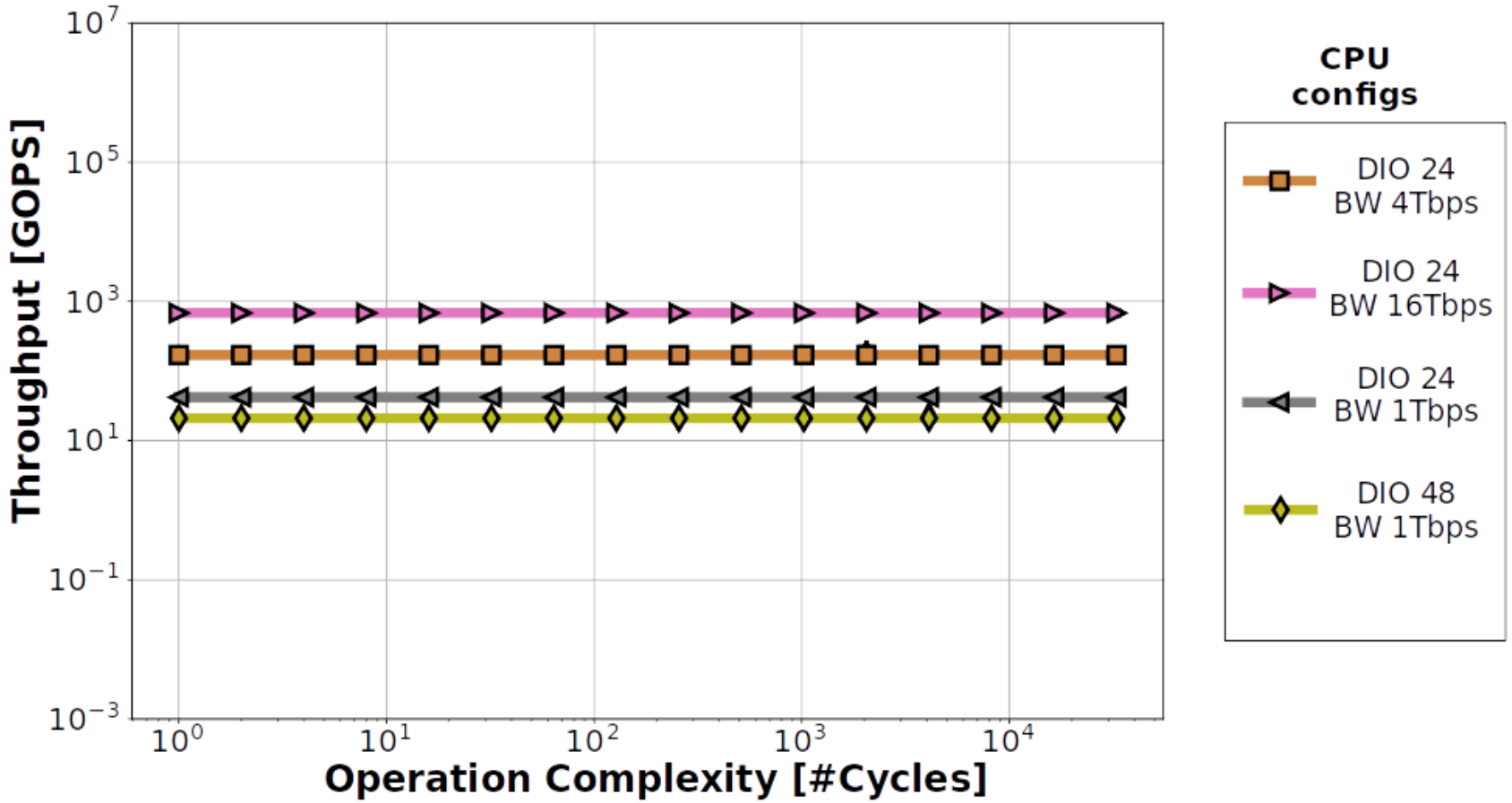
Throughput-CPU(Op) = BW/DIO
(independent of actual operation!)

ASIC²

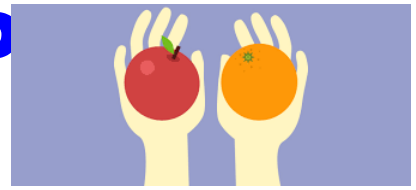
CPU Performance: Trends

CPU improves when BW increases or when DIO decreases

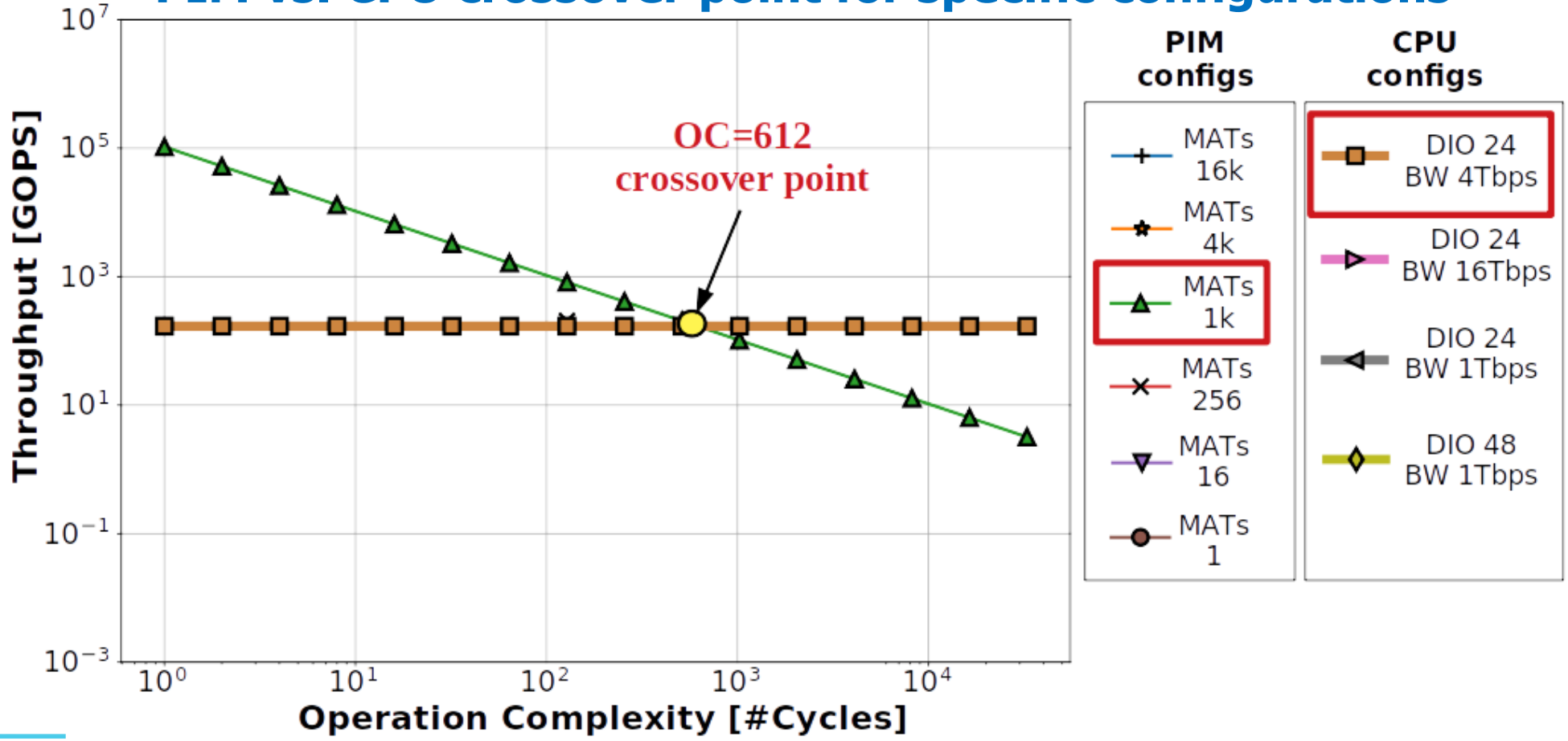
- Varying Parameters:
  - **DIO**
  - **BW**



Performance: PIM vs. CP (Crossovers Points)



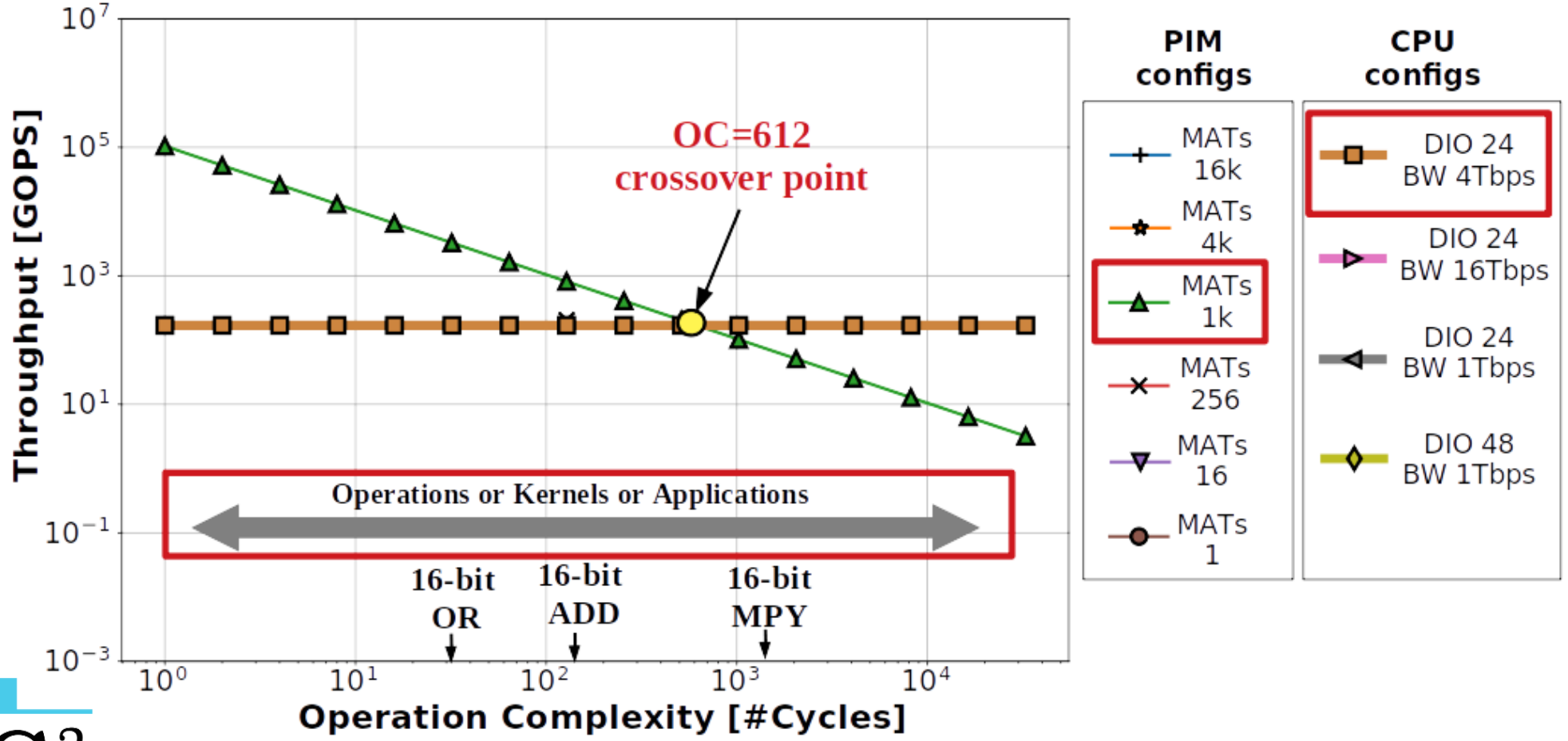
PIM vs. CPU crossover point for specific configurations



ASIC²

Performance: PIM vs. CPU (Operations/Kernels/etc.)

PIM vs. CPU for different operations or kernels or applications



PIM vs. CPU Under Limited Power

Power-related parameters of the Bitlet model

Parameter name	Notation	Value(s)	Type
Energy for PIM op (*OC*=1)	E^{PIM}	0.1pJ	Technological
Energy for memory bit transfer	E^{CPU}	15pJ	Technological
Thermal design power	*TDP*	20-160W	Architectural

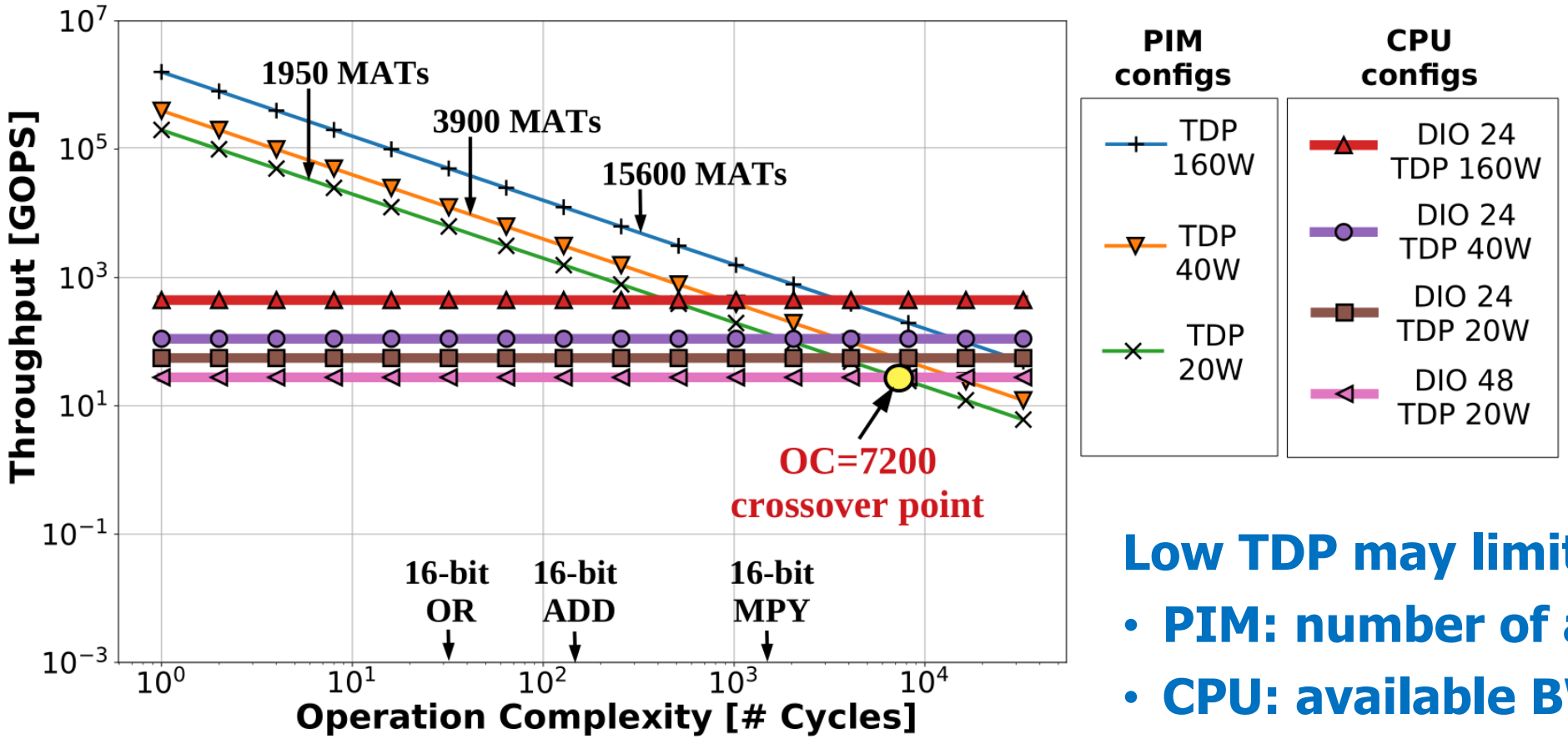
- Both PIM and CPU are limited by the system Thermal Design Power (TDP)
- PIM: TDP may limit # of active MATs
- CPU: TDP may limit memory bandwidth

TDP / energy per operation

$$PL\text{-}Throughput\text{-}PIM(Op) = Min(..., TDP/(E^{PIM}*OC)$$
$$PL\text{-}Throughput\text{-}CPU(Op) = Min(...; TDP/E^{CPU}*DIO\ )$$

ASIC²

Performance under Limited Power: PIM vs. CPU



Low TDP may limit:

- **PIM: number of active MATs**
- **CPU: available BW!**

ASIC²

Outline

- Goals
- From Memristor to Processing-In-Memory
- The Bitlet Model
  - Performance
  - Power Limited Performance
- **Comments & Conclusions**

ASIC²

Comments

- Scope to extend the PIM and CPU model when:
  - e.g. BW becomes huge.
  - e.g. PIM technology/architecture improves significantly.
- Model considers perfect parallelism, and also, accounts for data placement (details in the **arxiv report**)
- Can be extended to evaluate PIM's role in heterogenous computing.

Conclusions

- Simple, flexible, abstract model for PIM evaluation.
- Useful to understand PIM limitation+tradeoffs vs. CPU .
- Proved useful in assessing several external PIM ideas

Arxiv: https://arxiv.org/pdf/1910.10234.pdf

ASIC²

Thanks!

ARCHITECTURES
SYSTEMS
INTELLIGENT COMPUTING
INTEGRATED CIRCUITS

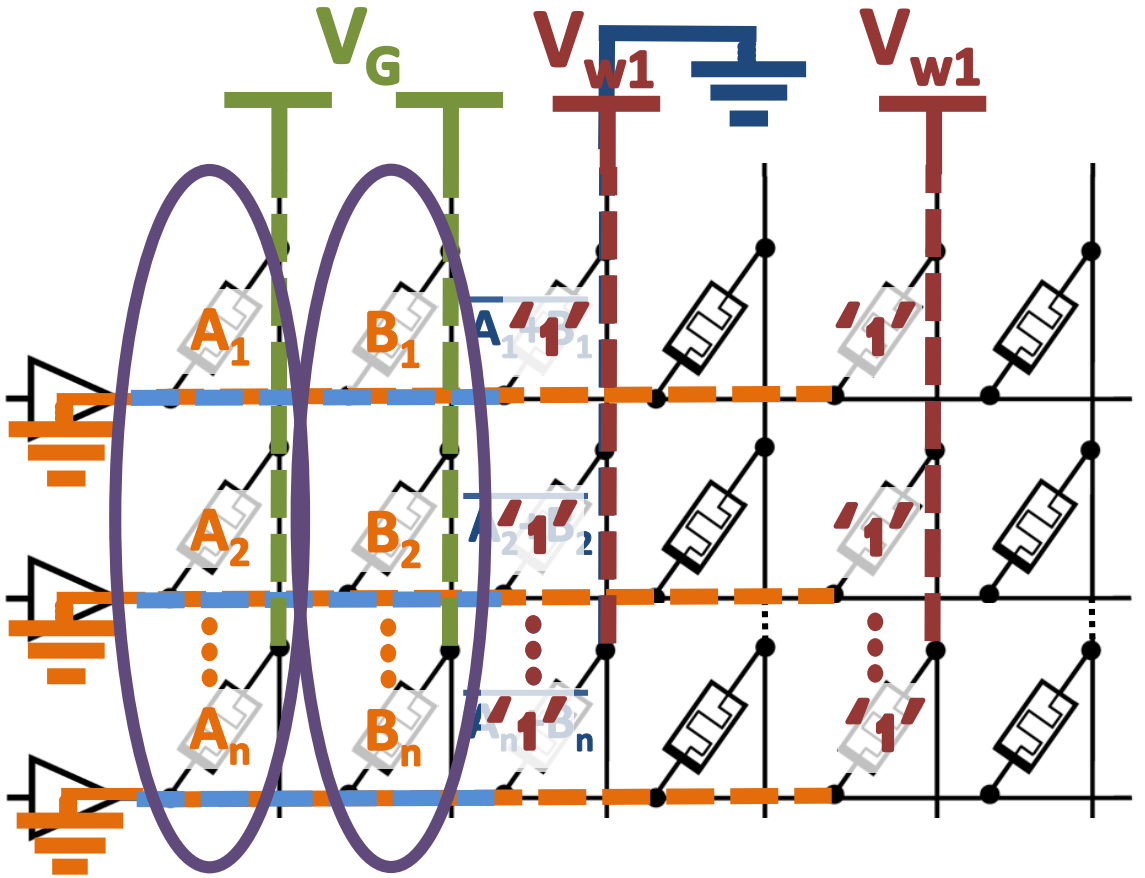
ASIC² Summer 2018

Questions!

ASIC^2

Example: In-Memory Parallel Execution

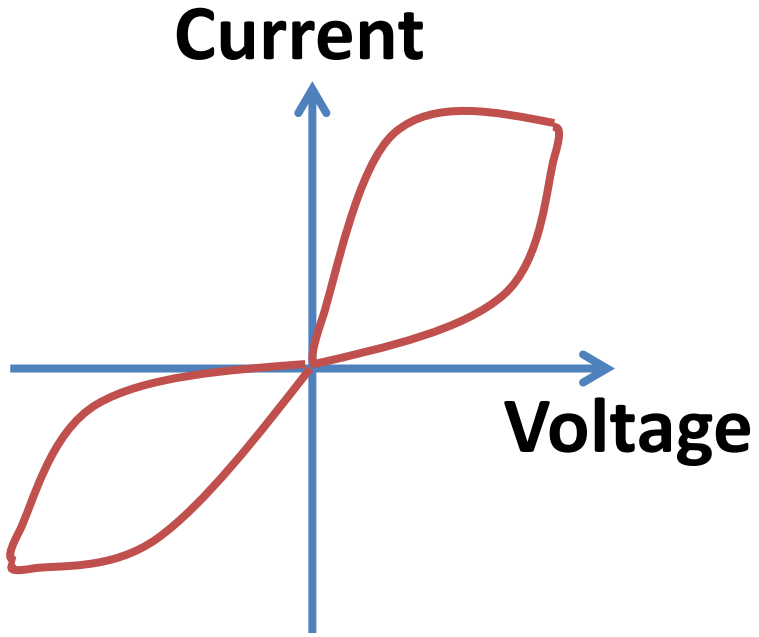
Initialize NOR(A_i,B_i) OUT to '1'



$$OR(A_i, B_i)$$
$$\forall i = 1, \ldots, N$$

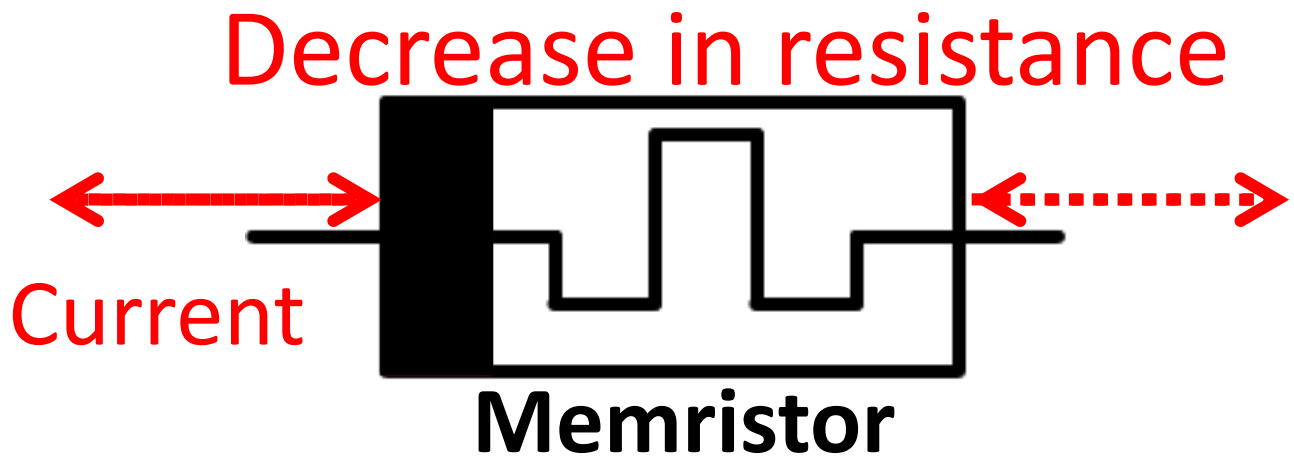
$$A \text{ or } B = not(A \text{ nor } B)$$

Memristor to Processing-in-Memory (1)



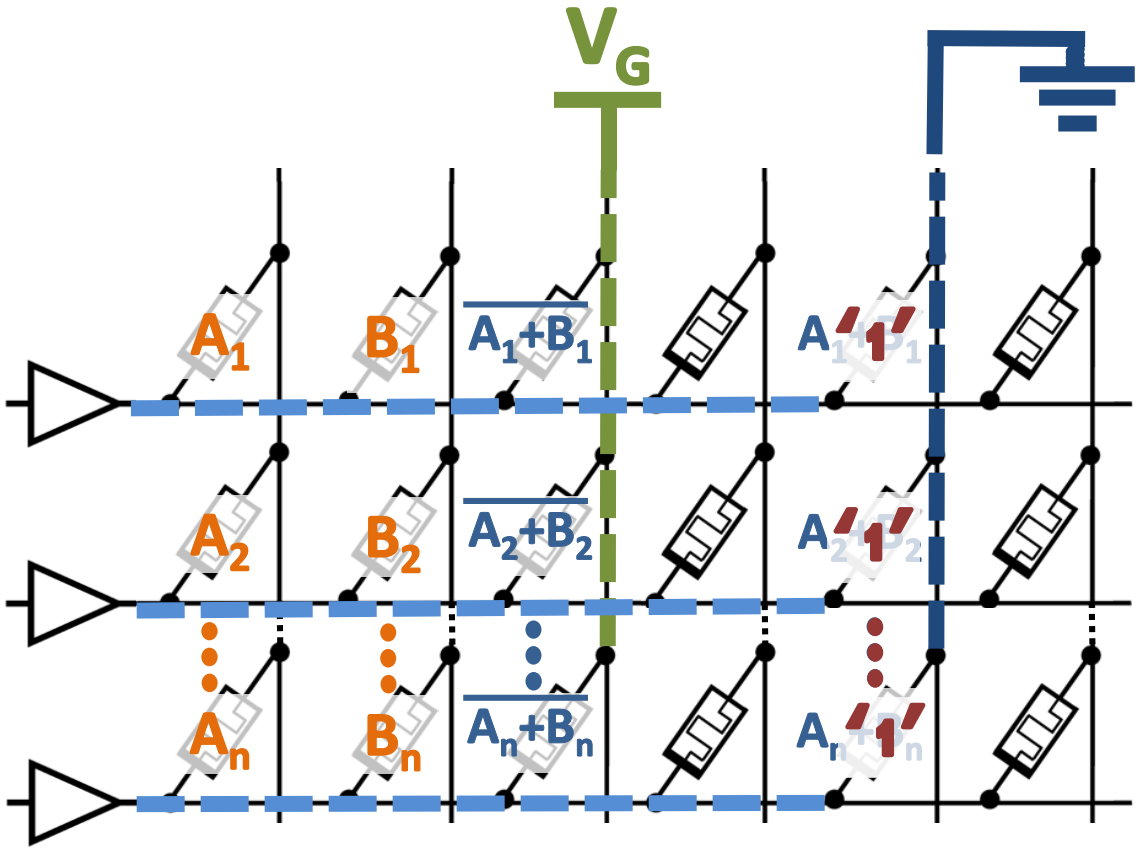
Low resistive state (R_{ON}, LRS)

High resistive state (R_{OFF}, HRS)



Example: In-Memory Parallel Execution

NOT($\overline{A+B}$)



$$OR(A_i, B_i)$$
$$\forall i = 1, \ldots, N$$

$$A \text{ or } B = not(A \text{ nor } B)$$

Total*:

Latency: 2 steps

Area: (2+2) cells

Energy: 2 operations

* Per element, ignoring initialization

ASIC^2