PhD Defense Thesis

EXPLORATION OF RECONFIGURABLE TILES OF COMPUTING-IN-MEMORY ARCHITECTURE FOR DATA-INTENSIVE APPLICATIONS

Presented by **Roman GAUCHI**

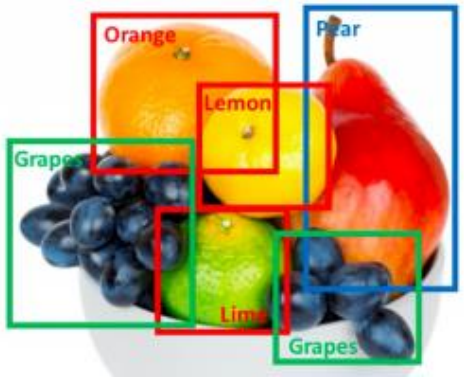
March 22, 2021 – Grenoble, FRANCE

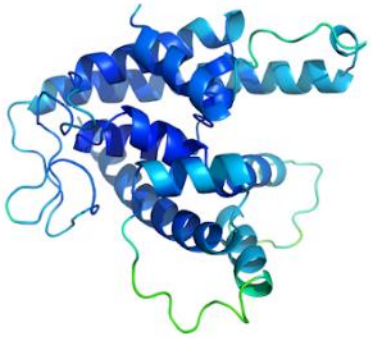
Mr. **Henri-Pierre CHARLES**, CEA, France	*as Co-director*
Mr. **Subhasish MITRA**, Stanford, USA	*as Co-director*
Mr. **Pascal VIVET**, CEA, France	*as Supervisor*

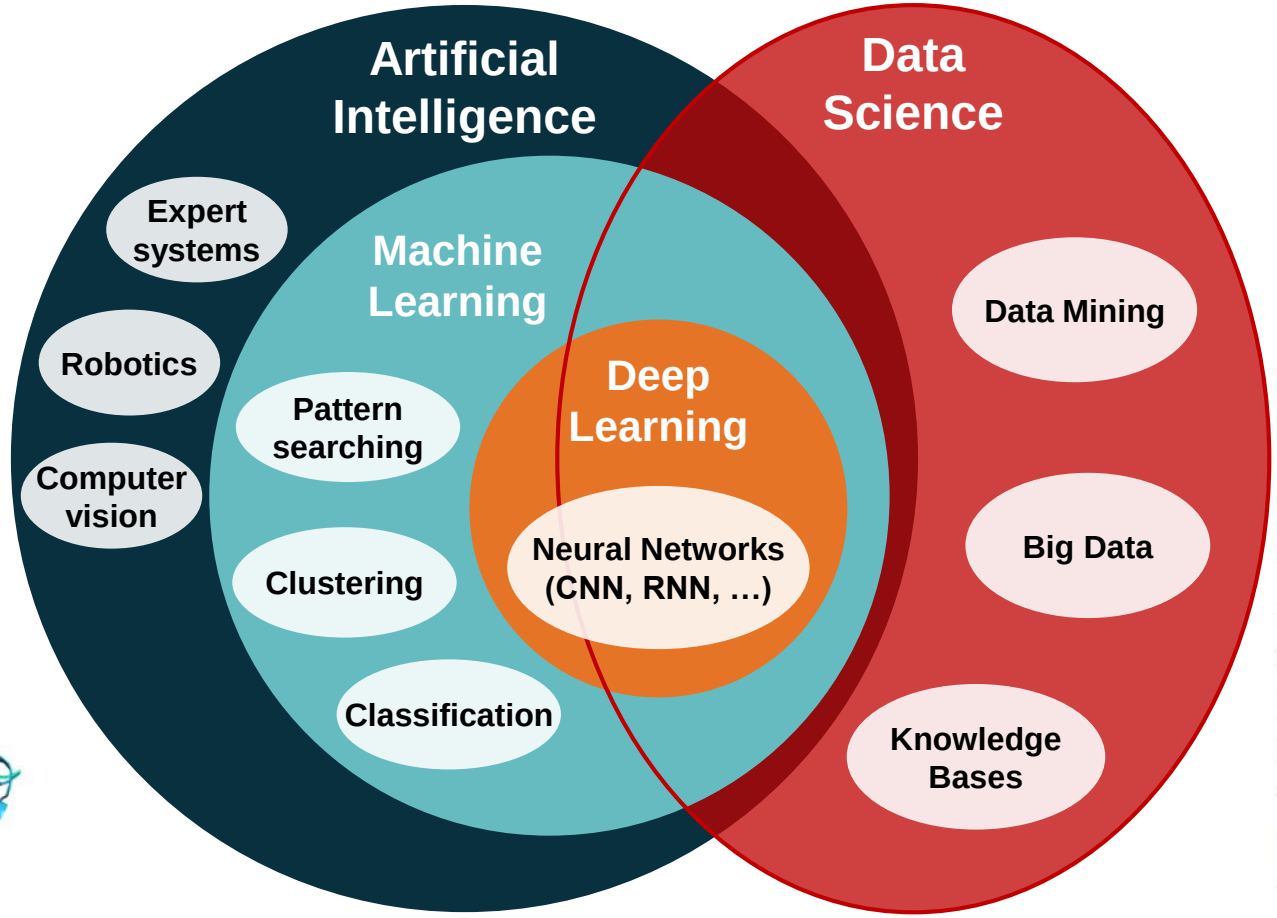
Mr. **Ian O'CONNOR**, EEA/INL, France	*as Reviewer*
Mr. **Gilles SASSATELLI**, LIRMM, France	*as Reviewer*
Mr. **Frédéric ROUSSEAU**, TIMA, France	*as Examiner*
Mrs. **Edith BEIGNE**, Facebook, USA	*as Examiner*
Mr. **Alexandre LEVISSE**, EPFL, Switzerland	*as Examiner*

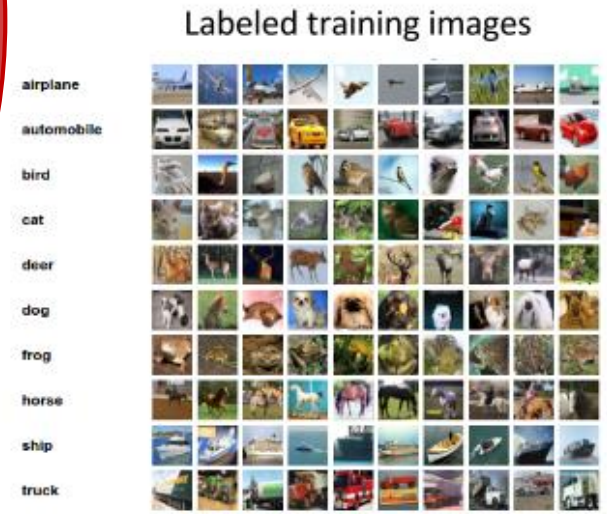
PhD Thesis of the Université Grenoble Alpes (UGA), prepared at the CEA LIST
Doctoral School of Electronique, Electrotechnique, Automatique, Traitement du Signal (EEATS)

BIG DATA APPLICATIONS





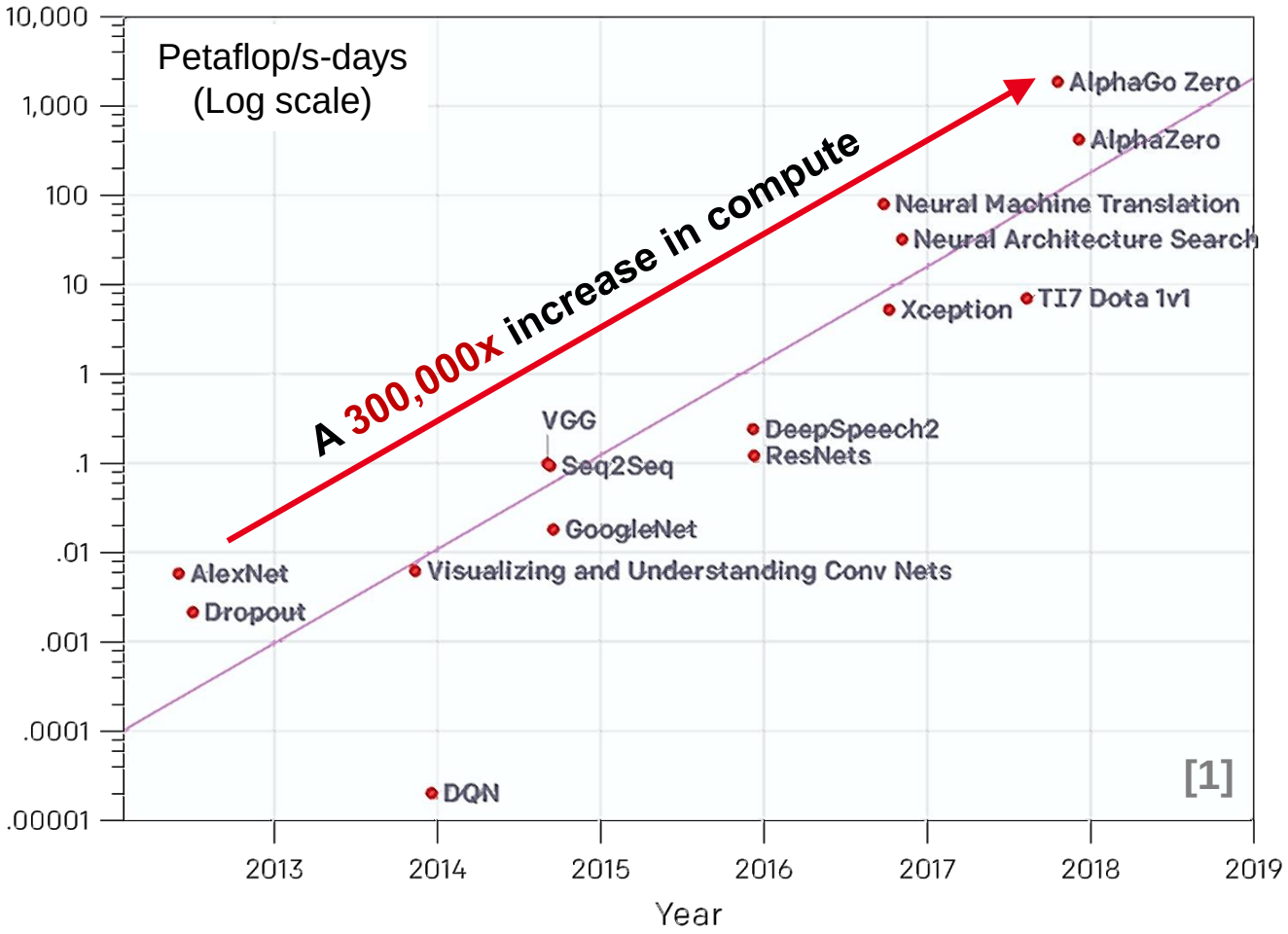




MOTIVATIONS

DEMANDS FOR DEEP NEURAL NETWORKS

➔ **Modern applications compute on more and more data**



Deep Neural Networks for Natural Language Processing

Model	Training (h)	Power (W)
ELMo	336	**517**
Transformer	84	**1,515**
BERT	79	**12,041**

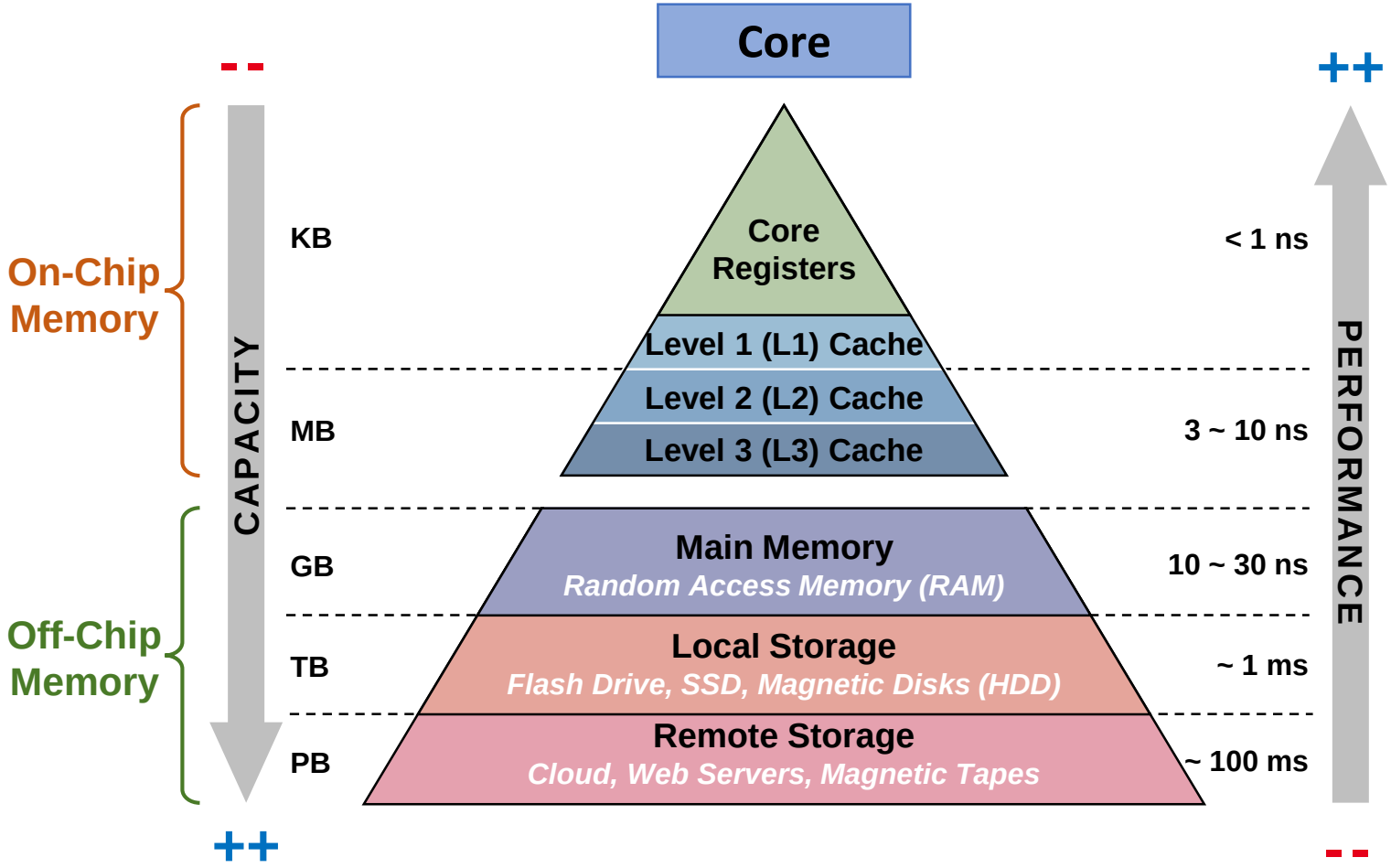
[2]

[1] source: https://openai.com/blog/ai-and-compute/
[2] "*Energy and Policy Considerations for Deep Learning in NLP*", E. Strubell et al., ACL, 2019

MOTIVATIONS

TRADITIONAL MEMORY HIERARCHY

➔ **high-performance on-chip memory & high-capacity off-chip memory**



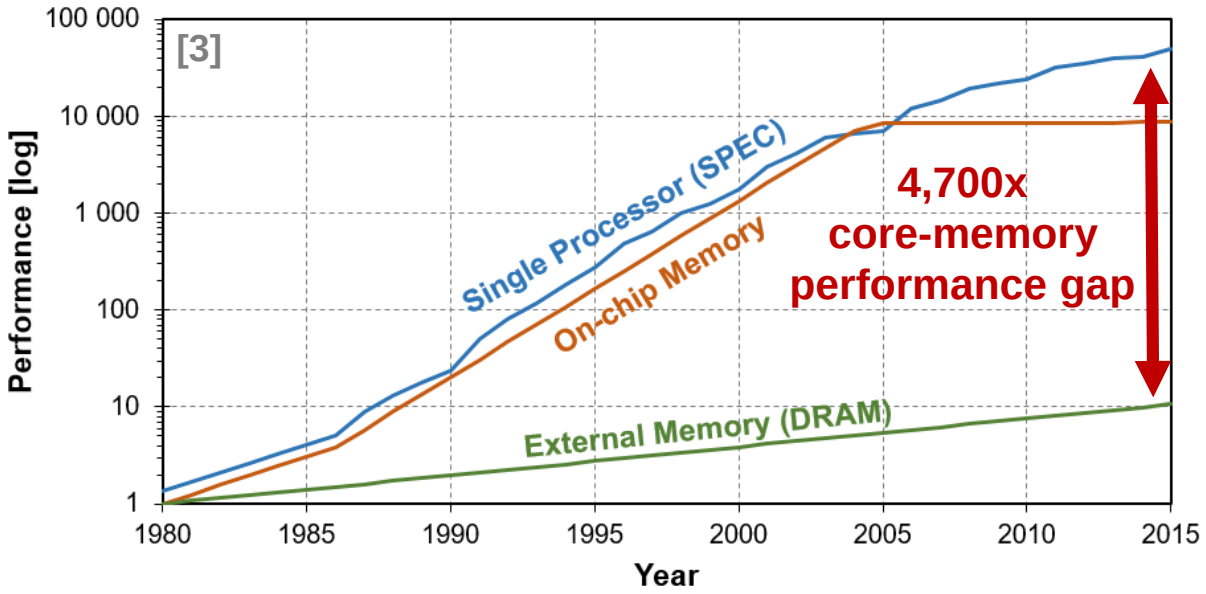
MOTIVATIONS

THE MEMORY & ENERGY WALL

➔ **Computation latency** becomes limited by the **memory access time**…
…and memory energy is much higher than the computational energy

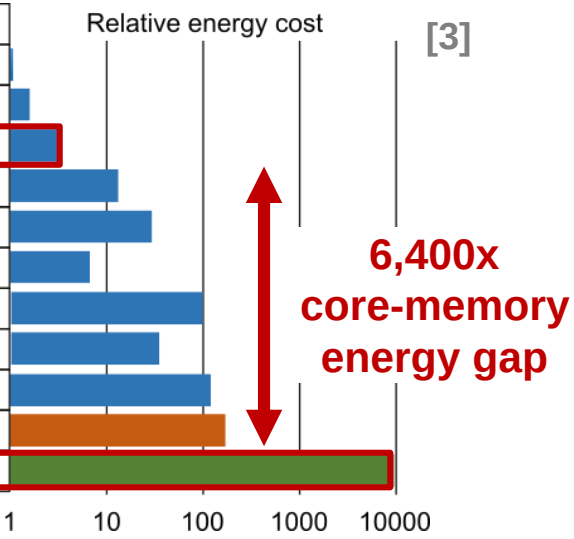
➔ Slowdown of Dennard Scaling ➔ Transistors are **not scaling anymore** (CMOS)

The Memory Wall: **Performance impacts**



The Memory Wall: **Energy impacts**

Operation:	Energy (pJ)
8b Add	0.03
16b Add	0.05
32b Add	0.1
16b FP Add	0.4
32b FP Add	0.9
8b Mult	0.2
32b Mult	3.1
16b FP Mult	1.1
32b FP Mult	3.7
32b SRAM Read (8KB)	5
32b DRAM Read	640



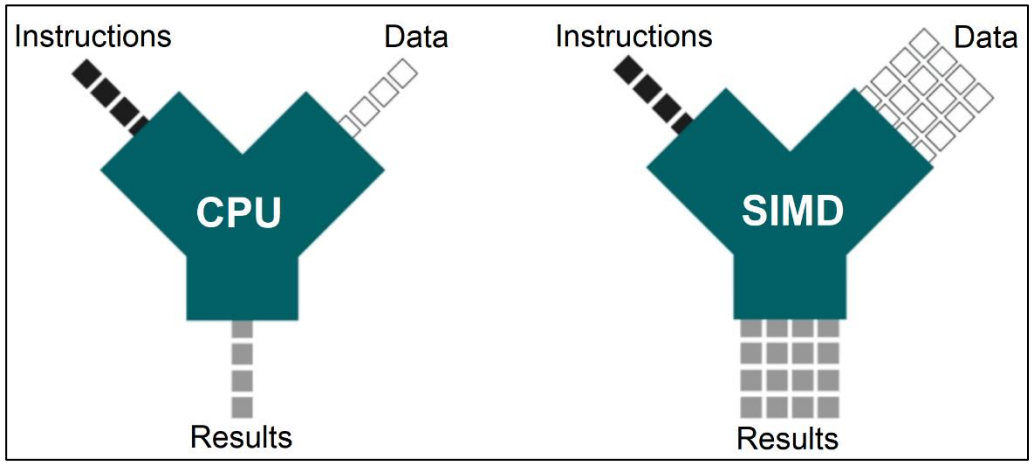
[3] "*Computer Architecture: A Quantitative Approach*", J. L. Hennessy and D. A. Patterson, 6th edition, 2018

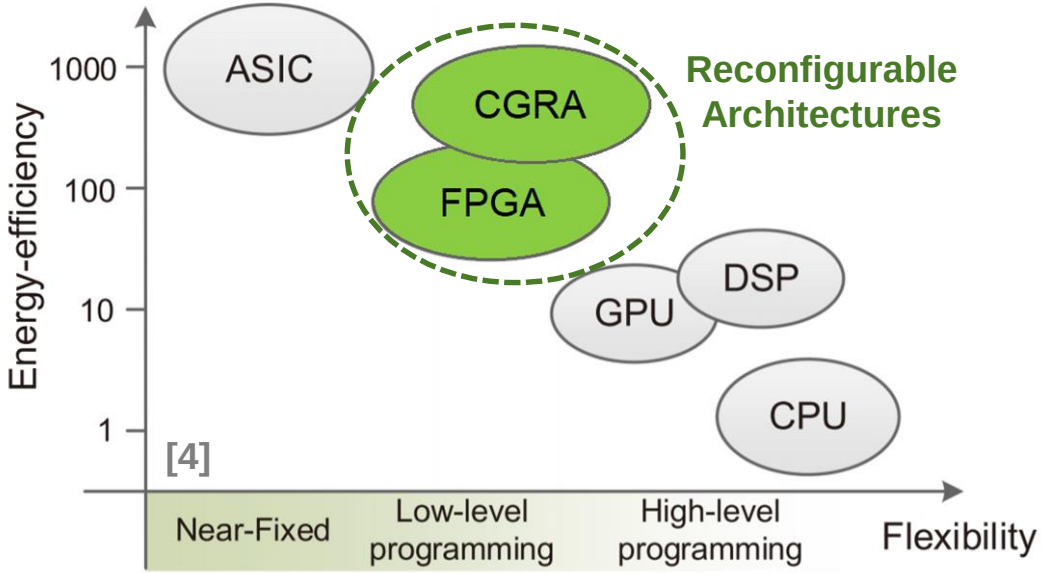
STATE-OF-THE-ART

DISTRIBUTED AND RECONFIGURABLE ARCHITECTURES

- **Exploit Data-Level Parallelism to improve data-intensive applications performance**
- **Vector processors & Parallel engines**
  - **SIMD**: *Single Instructions Multiple Data*
  - Higher bandwidth and scalable instructions
  - For massive data and compute parallelism
- **Reconfigurable architectures (FPGA, CGRA)**
  - Energy-efficient architectures
  - Enhanced data path reconfiguration

Vector processing and reconfigurability exist, but with limited vector sizes

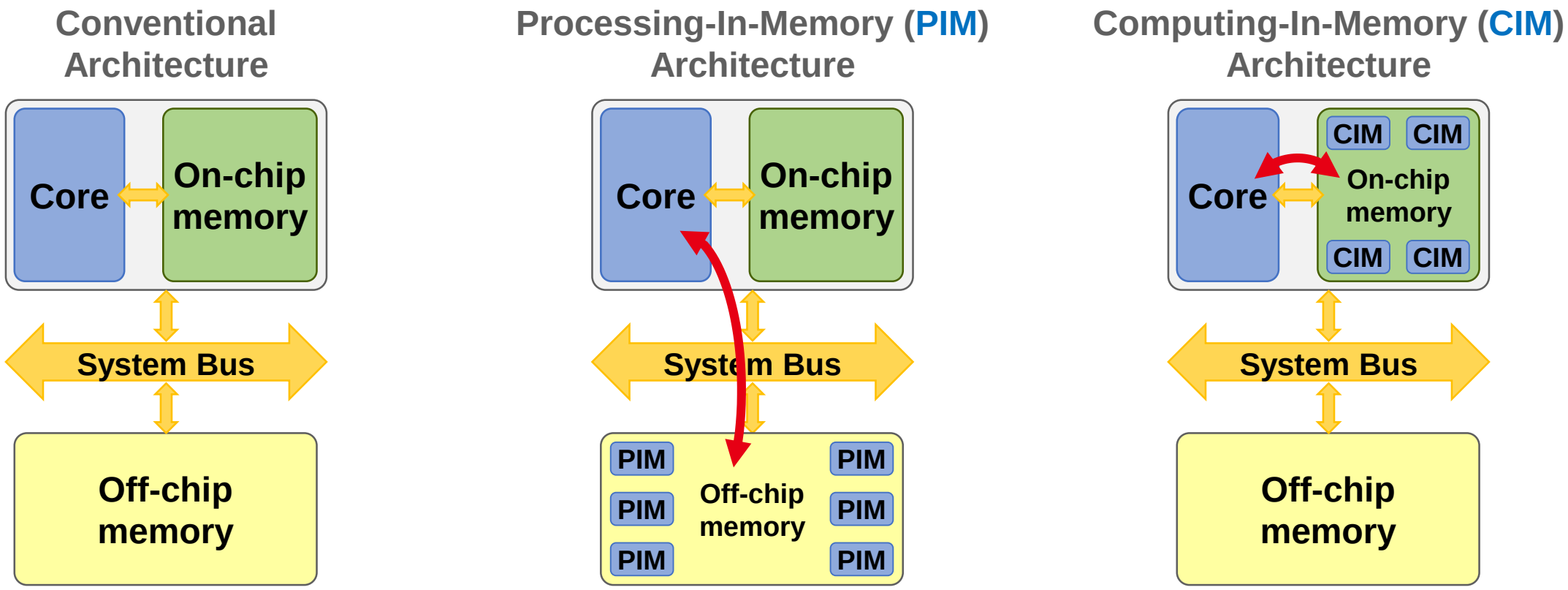




[4] "*A Survey of Coarse-Grained Reconfigurable Architecture and Design: Taxonomy, Challenges, and Applications*", L. Lui et al., ACM Computing Surveys, 2019

STATE-OF-THE-ART

COMPUTATION IMMERSED IN MEMORY APPROACHES



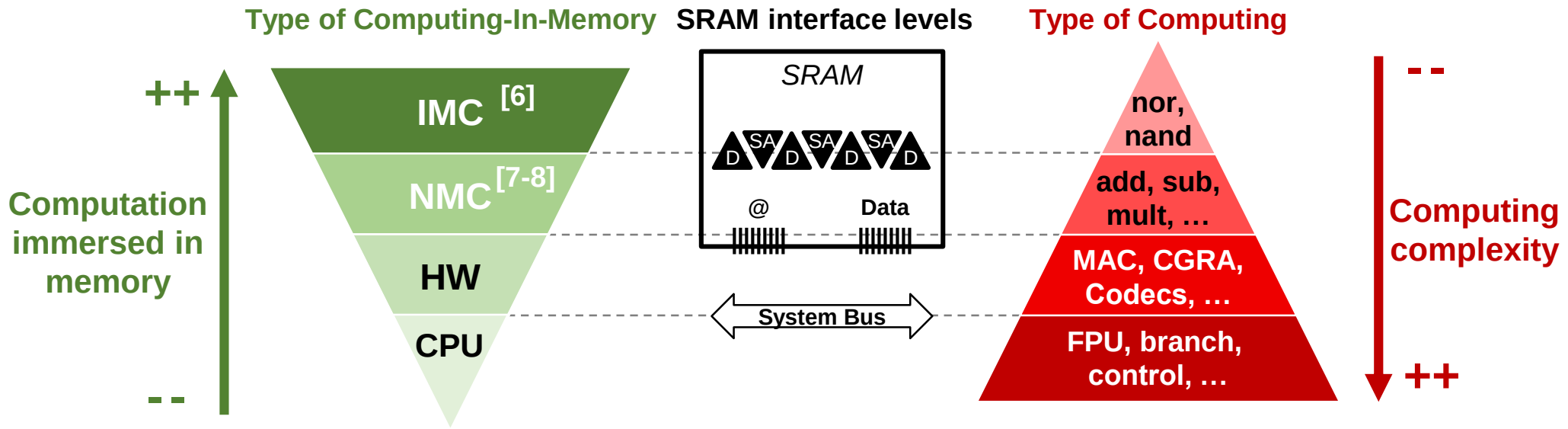
- **PIM** approach ➔ relax the **off-chip** "memory wall" (*apply to DRAM* [5])
- **CIM** approach ➔ relax the **on-chip** "memory wall" (*apply to SRAM or NVM*)

[5] "*BLAST on UPMEM*", D. Lavenier et al., Research Report, INRIA Rennes, 2016.

STATE-OF-THE-ART

COMPUTING-IN MEMORY TAXONOMY, IN- OR NEAR- MEMORY COMPUTING ?

➔ **Concept**: Bring computation closer to the memory (*SRAM-based technology*)



IMC *In-Memory Computing*
NMC *Near-Memory Computing*
HW *Hardware accelerator integrating SRAM*
CPU *Central Processing Unit*

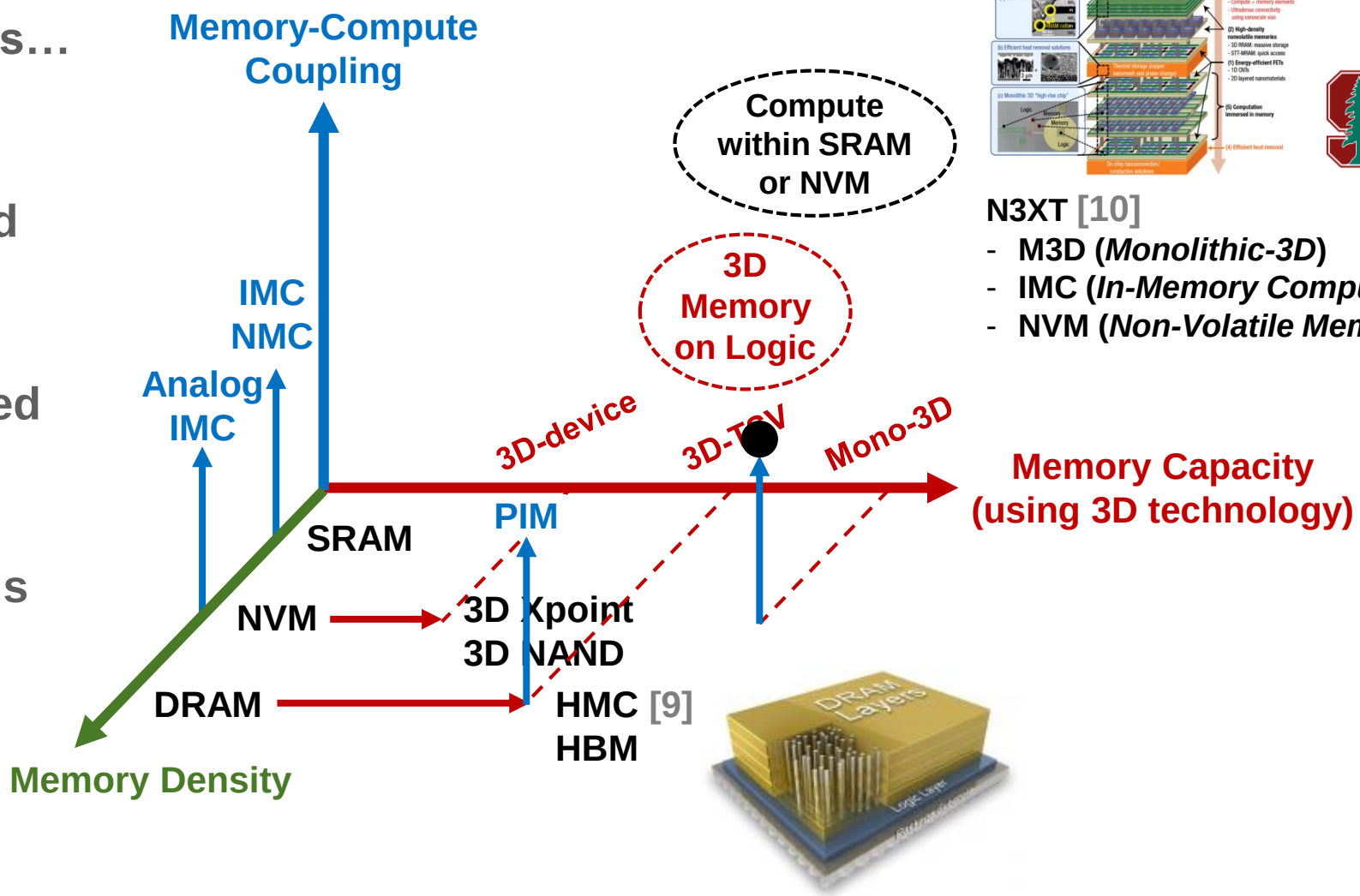
[6] "*A 35.6 TOPS/W/mm² 3-Stage Pipelined Computational SRAM With Adjustable Form Factor for Highly Data-Centric Applications*", J.-P. Noel et al., L-SSC, 2020
[7] "*Neural cache: bit-serial in-cache acceleration of deep neural networks*", R. Das et al., ISCA, 2018
[8] "*Computational SRAM Design Automation using Pushed-Rule Bitcells for Energy-Efficient Vector Processing*", J.-P. Noel et al., DATE, 2019

STATE-OF-THE-ART

NEW TECHNOLOGIES TO IMPROVE MEMORY-COMPUTE COUPLING

More-than-Moore technologies…
What for ?

- **Memory density** is increased with memory technology
- **Memory capacity** is increased thanks to 3D integration
- **Memory-compute coupling** is improved with computation immersed in memory



N3XT [10]
- M3D (*Monolithic-3D*)
- IMC (*In-Memory Computing*)
- NVM (*Non-Volatile Memory*)

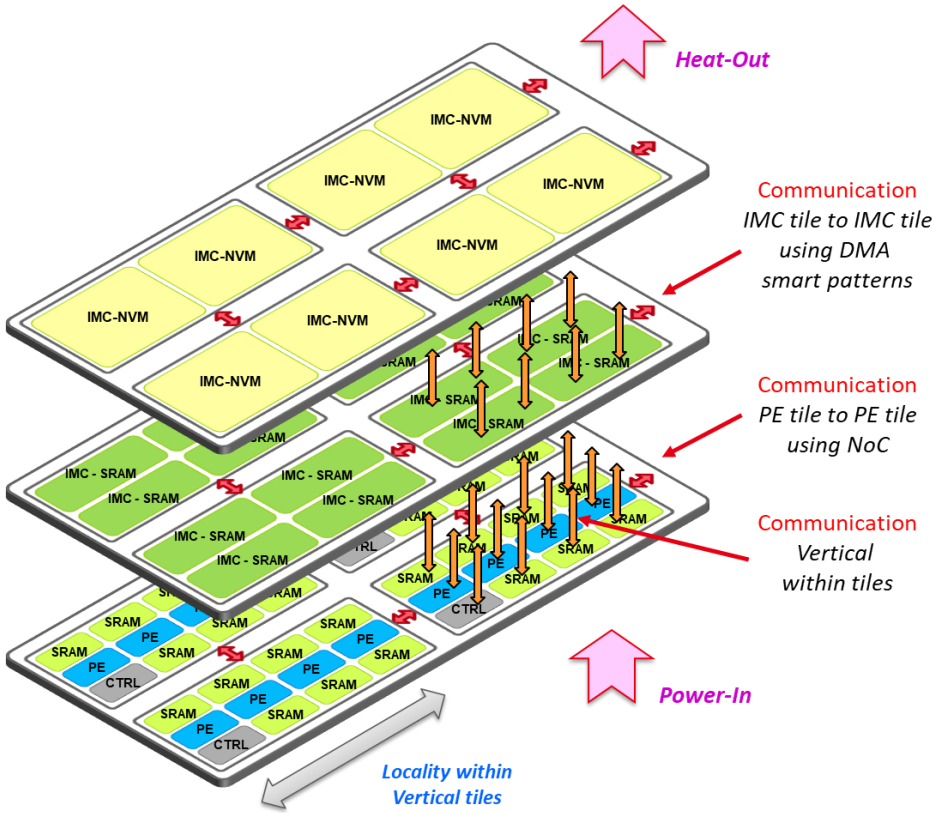
[9] "*Hybrid Memory Cube (HMC)*", J. T. Pawlowski, HCS, 2011
[10] "*Energy-Efficient Abundant-Data Computing: The N3XT 1000x*", M. M. Sabry Aly et al., Computer, 2015

LONG-TERM PERSPECTIVES

A DREAM: A 3D-STACKED DISTRIBUTED COMPUTING ARCHITECTURE

Possible solution to break the "Memory Wall"

➔ **Emerging architectures coupled with emerging technologies**



(1) 3D-stacked architecture

(2) 3D fine-grained integration (M3D)

(3) Efficient heat dissipation

(4) High-density emerging NVM

(5) On-chip In-/Near- Memory Computing

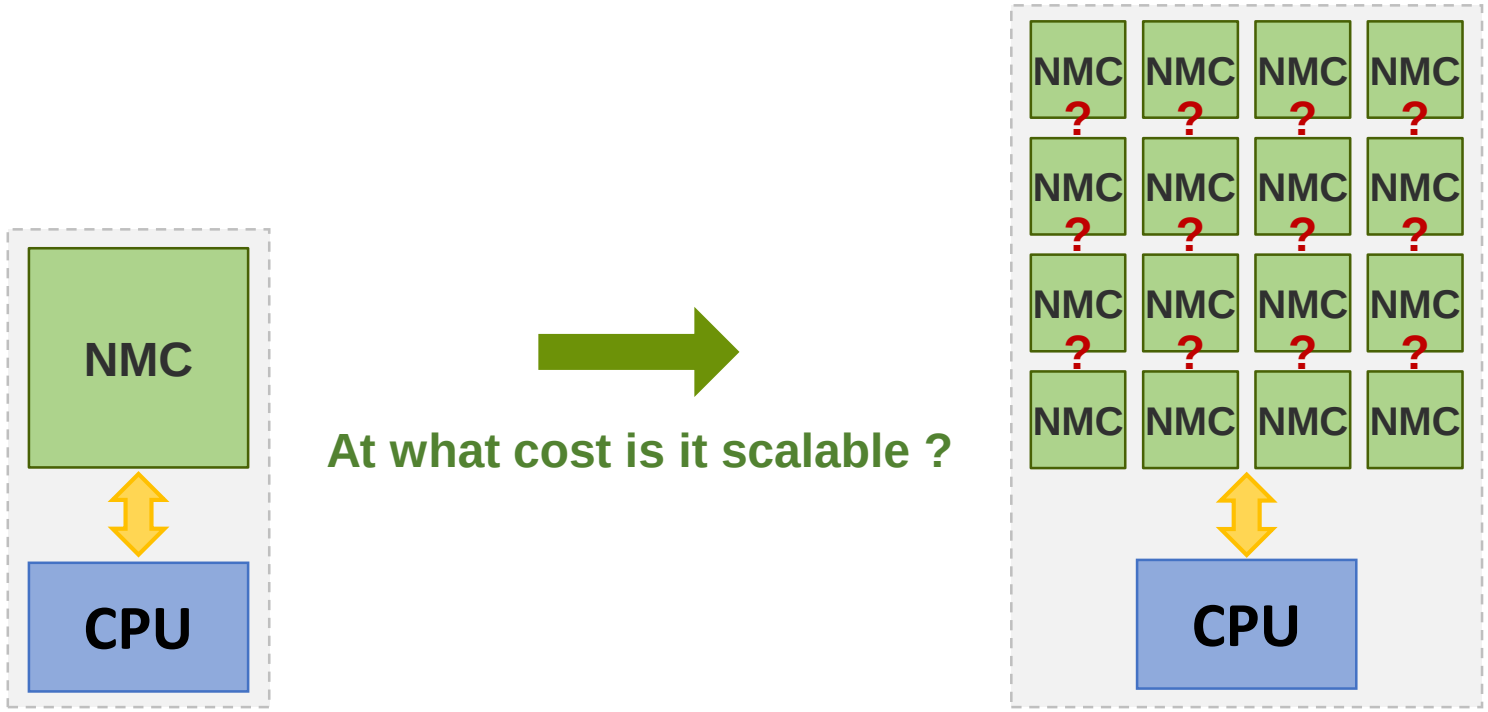
(6) Reconfigurable architecture

(7) Distributed computing

(8) General-purpose programming model

➔ **PhD objectives & contributions**

PHD OBJECTIVES



Existing works

PhD objectives

- How to scale out NMC and make it reconfigurable?
- How to program and configure such architecture?
- What are the sizing and limiting parameters of this architecture?

PHD CONTRIBUTIONS

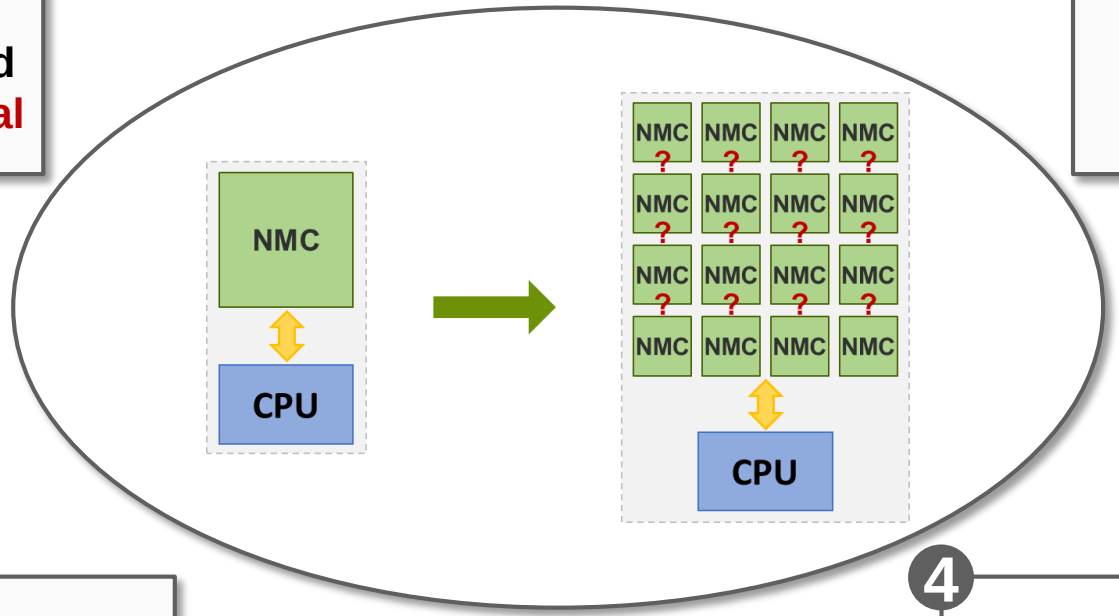
1 **METEOR**:
A Reconfigurable Memory-based Computing **Architecture Proposal**

2 **Software Integration** for
Scalable Vector Computing

3 **Design Space Exploration**
of the Memory Interconnect

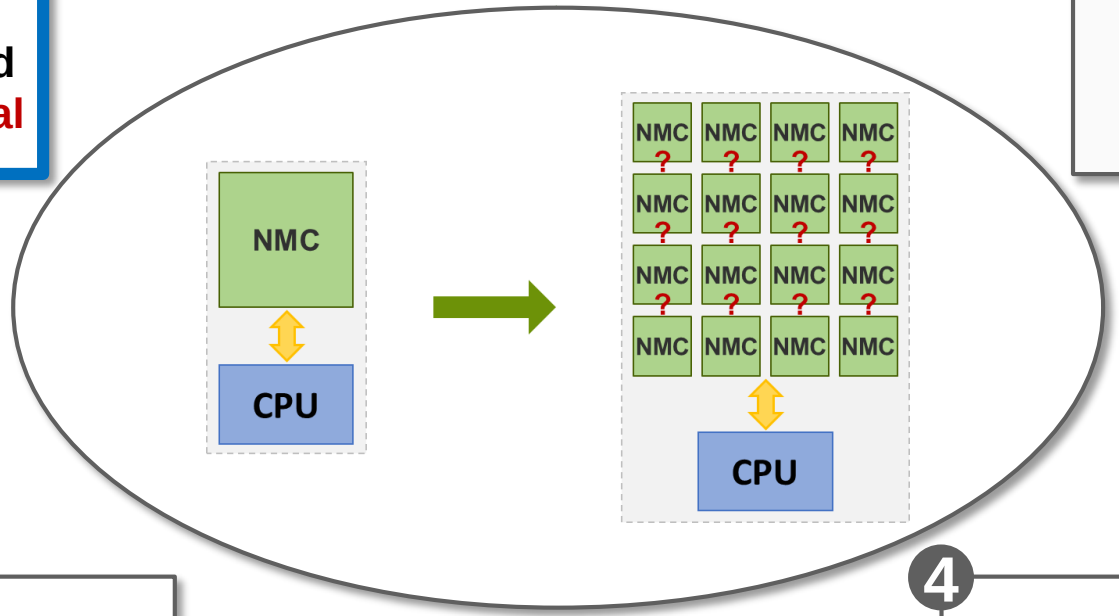
4 **ArchSim**:
An IMC-NMC Software-Hardware
Simulation Platform

5 METEOR Architecture
Exploration Results



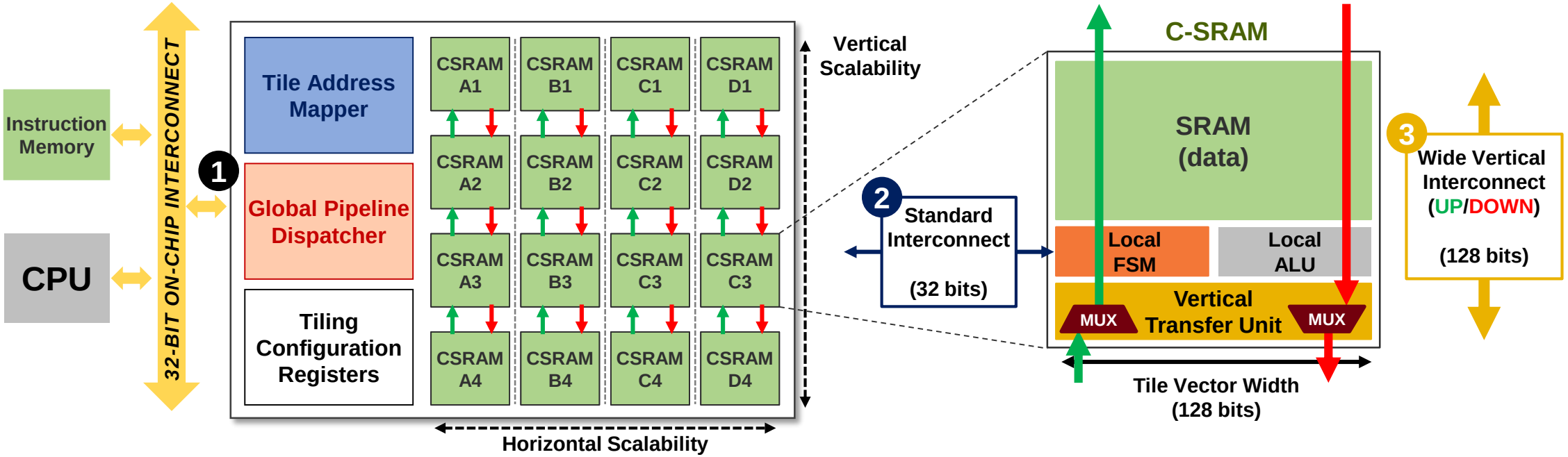
OUTLINE

1. **METEOR**: A Reconfigurable Memory-based Computing **Architecture Proposal**
2. Software Integration for Scalable Vector Computing
3. Design Space Exploration of the Memory Interconnect
4. ArchSim: An IMC-NMC Software-Hardware Simulation Platform
5. METEOR Architecture Exploration Results



METEOR

A Reconfigurable Tiles of C-SRAM Architecture



2D Scalable Cluster Architecture

- **Tile Address Mapper (TAM)**
  - Horizontal & Vertical configurability
- **Global Pipeline Dispatcher (GPD)**
  - Data Hazards & address conflicts
- **Tiling Configuration Registers (TCR)**
  - Grid shape and vector size

NMC Tile

- **C-SRAM (Compute-SRAM)** [11]
  - Arithmetic & Logical instructions
  - Pipelined instructions (5 stages)
- **Vertical Transfer Unit**
  - Inter-tile vertical communication

Interconnects

1. **METEOR standard memory interface**
2. **C-SRAM standard interconnect** [12]
   - 32-bit read/write accesses
3. **Wide Vertical Interconnect**
   - 128-bit bi-directional transfers

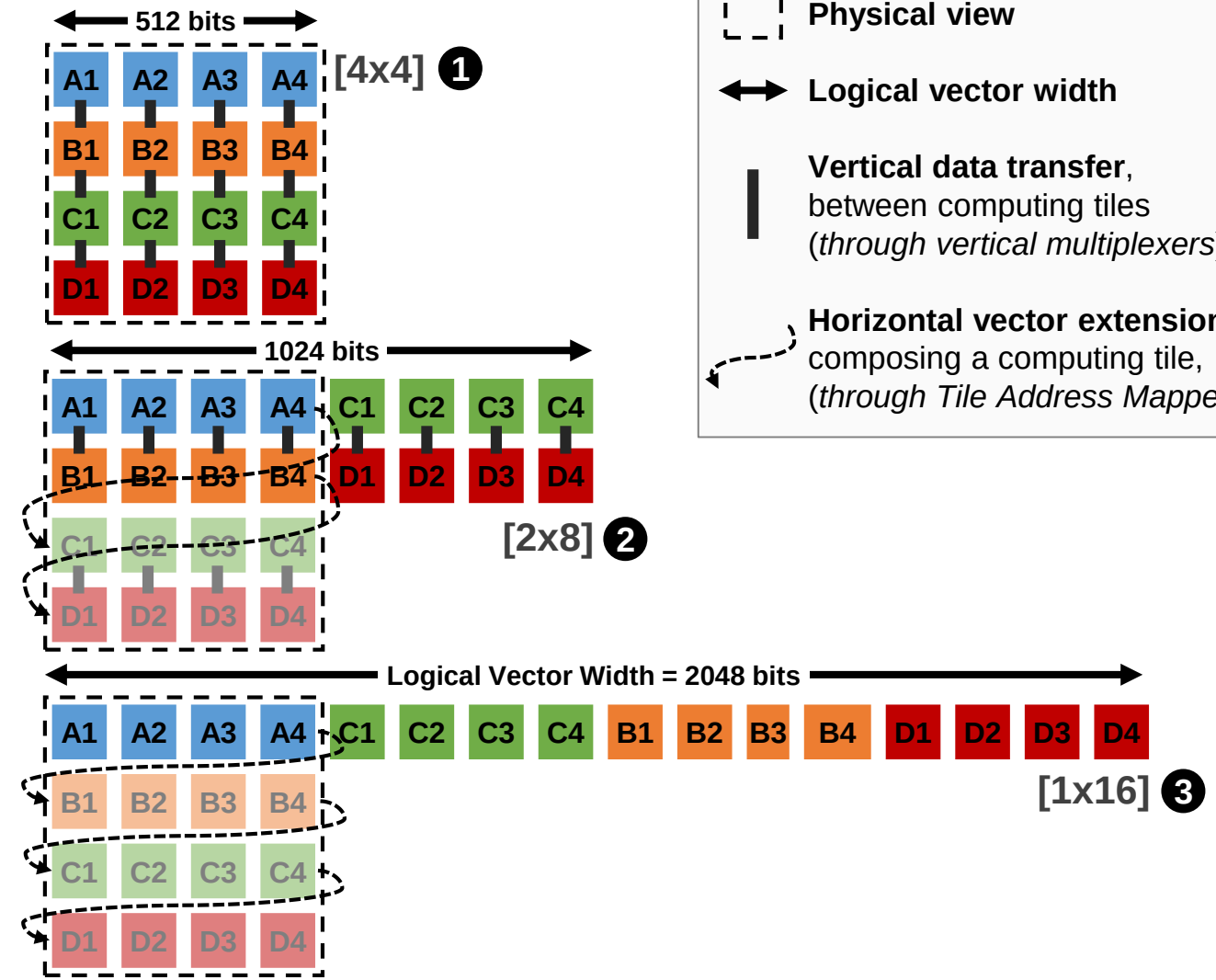
[11] "*Computational SRAM Design Automation using Pushed-Rule Bitcells for Energy-Efficient Vector Processing*", J.-P. Noel et al., DATE, 2019
[12] "*Memory Sizing of a Scalable SRAM In-Memory Computing Tile Based Architecture*", R. Gauchi et al., VLSI-SoC, 2019

METEOR

Horizontal and Vertical Reconfigurability

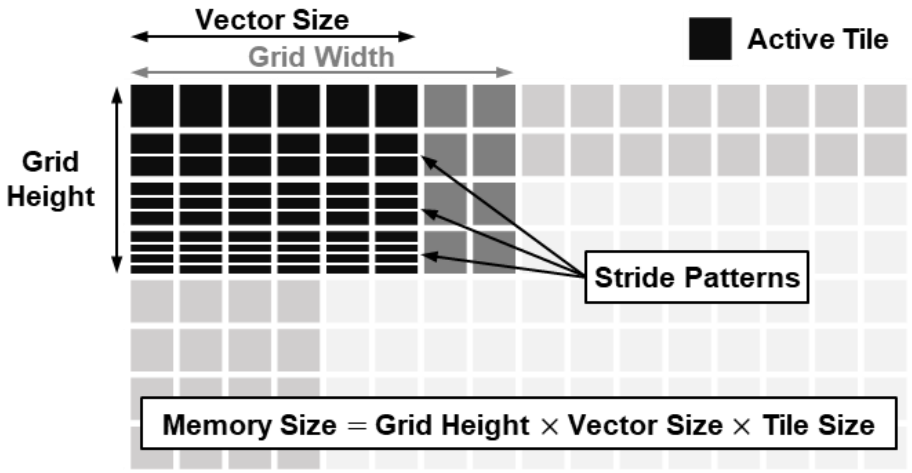
➔ **Concept**: The physical grid can be logically reconfigured

- **Vertical Vector Extension** create **more vectors** through the vertical connections
  - Increase inter-tile com' & compute
- **Horizontal Vector Extension** create **larger vectors** through the *Tile Address Mapper*
  - Increase Data-Level Parallelism



METEOR

TILING CONFIGURATION PARAMETERS



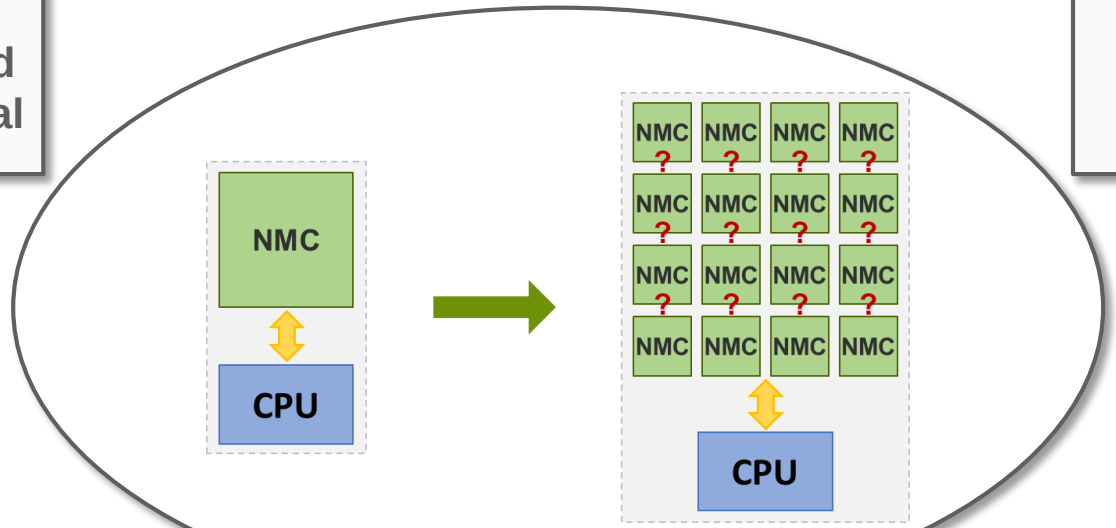
Parameter	Description	Reg #	Type	RD flag
	Setter			
Grid Width	set the physical vector size (multiple of tile vector size)	1	U	0
Vector Size	set the logical active vector size (inferior to grid size)	2	U	0
Stride Pattern	set the read/write address stride pattern	3	U	0
	Getter			
Grid Width	get the physical vector size (multiple of tile vector size)	1	U	1
Vector Size	get the logical active vector size (inferior to grid size)	2	U	1
Stride Pattern	get the read/write address stride pattern	3	U	1
Memory Size	get the total cluster active memory size	4	U	1
Grid Height	get the physical number of tile available	5	U	1

- The **Grid Layout** is configured with the *Tiling Configuration Registers* (TCR)
- **Dynamic vector reconfiguration** increase the **data & compute parallelisms**

➔ **What is the programming model ?**

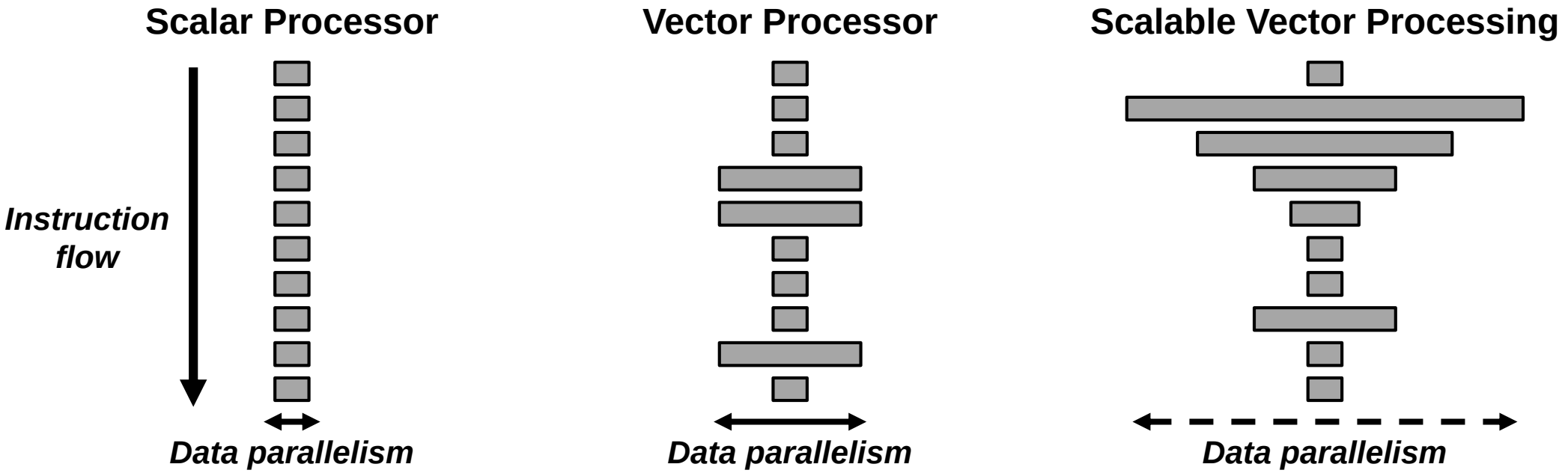
OUTLINE

1. METEOR: A Reconfigurable Memory-based Computing Architecture Proposal
2. **Software Integration** for Scalable Vector Computing
3. Design Space Exploration of the Memory Interconnect
4. ArchSim: An IMC-NMC Software-Hardware Simulation Platform
5. METEOR Architecture Exploration Results



SOFTWARE INTEGRATION

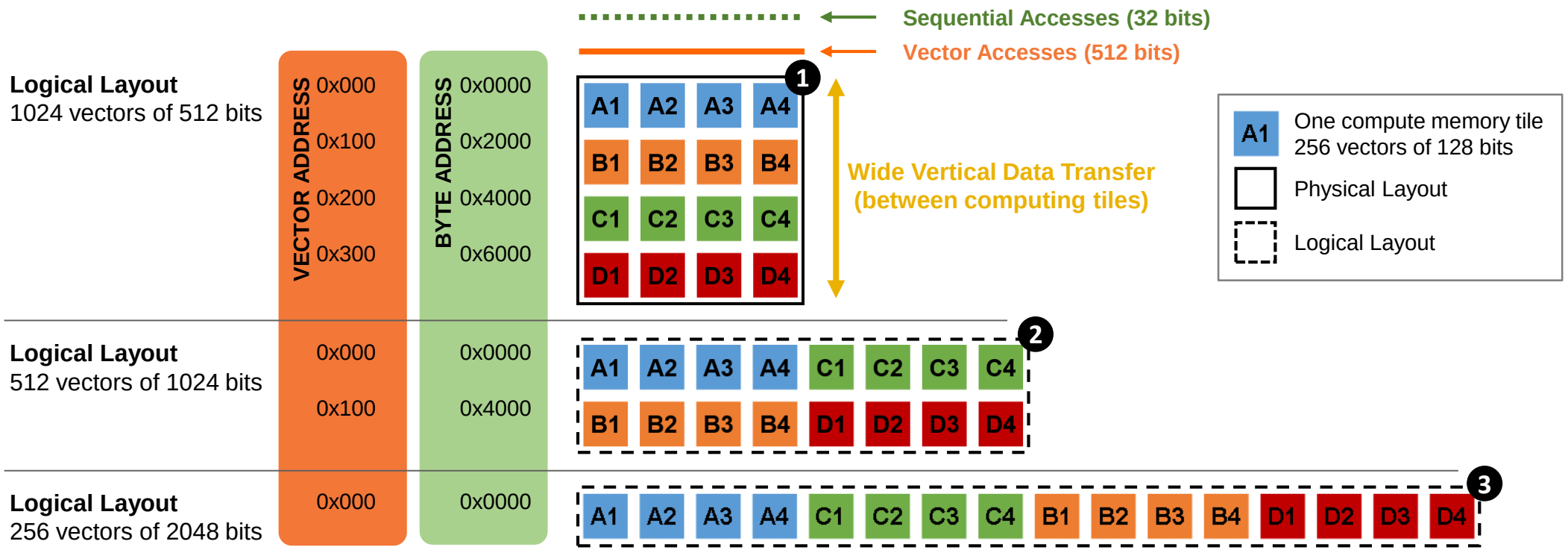
EXECUTION MODELS



- **Scalar Processors**
  - Fixed width instruction ➔ memory wall bottleneck
- **Vector Processors (e.g. SIMD)**
  - Data-level parallelism ➔ **fixed** and **limited vector sizes** (*typically 512 bits*)
- **Scalable Vector Processing (e.g. METEOR)**
  - Dynamic vector configuration ➔ exploit **massive data-level parallelism**

SOFTWARE INTEGRATION

PROGRAMMER'S VIEW: PROGRAMMABLE AS A VECTOR MACHINE



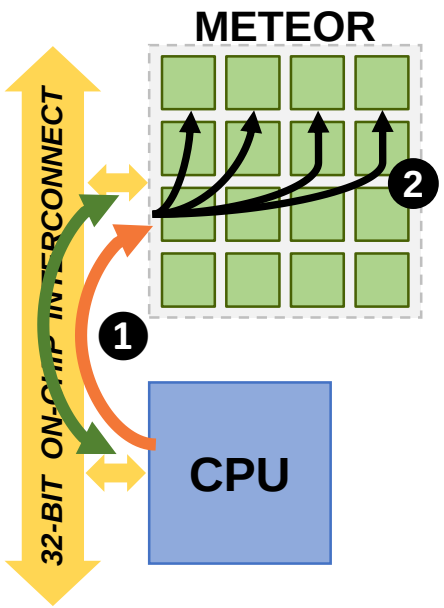
- **Vector Data Approach** ➔ manipulate large data (*in computing mode*)
- **Vertical Communication** ➔ dense interconnections through multiple tiles
- **Dynamic Vector Reconfiguration** ➔ extend vector size → reduce data movement
- **At maximum vector width** ➔ NMC instruction is broadcasted on all tiles

- Vector programming model equivalent to SIMD ➔ using same kernel, with available compiler tool chain

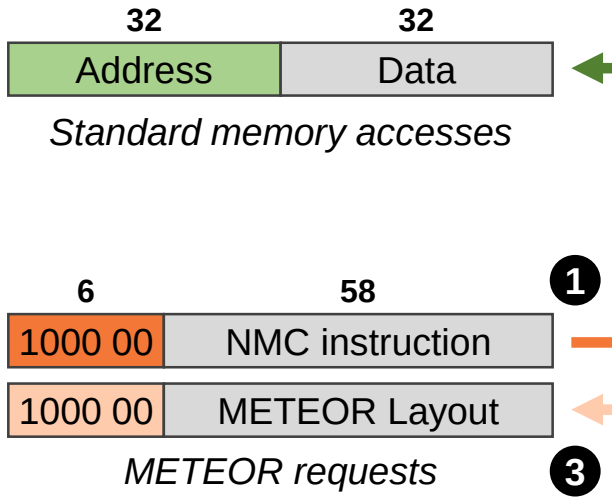
SOFTWARE INTEGRATION

INSTRUCTION SET ARCHITECTURE (ISA)

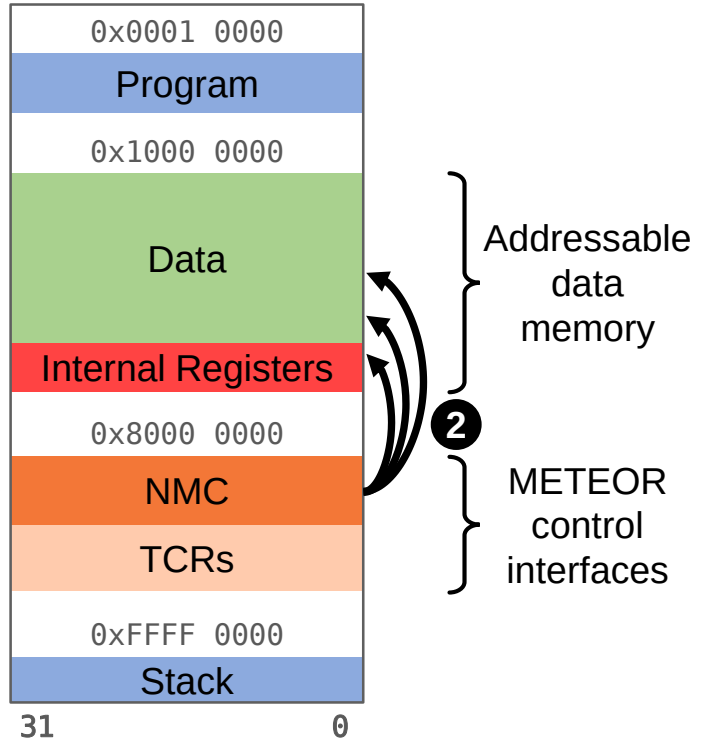
Multi-tile Architecture



32-bit System Bus Memory Requests



System Memory Map



- **Standard memory accesses**
  - Load/Store a data from/to a specific memory segment
- **Send NMC instructions**
  - ❶ Store a data → **METEOR control interface**
  - ❷ Dispatch the NMC instructions → physical memory tiles targeted
- **Send METEOR layout configurations**
  - ❸ Load/Store a data → **METEOR** *Tiling Configuration Registers* (TCRs)

SOFTWARE INTEGRATION

SCALABLE VECTOR PROCESSING

vmul8 instruction: C macro of 8-bit chunk multiply operation on a METEOR vector

```
#define vmul8(_result, _operA, _operB) do {                                    \
  uint64_t            cm_word = MAKE_ISA_RTYPE(OPC_MUL8,_result,_operA,_operB); \
  volatile uintptr_t* cm_addr = (uintptr_t*)((uint32_t)(cm_word >> 32));        \
  uintptr_t           cm_data = (uint32_t)(cm_word);                            \
  *cm_addr = cm_data;          /* NMC compute is equivalent to a store a data */ \
  asm("" ::: "memory");        /* memory fence */                               \
} while(0)
```

- **NMC instruction as a C macro**
  - Build at compile time
  - Support of 54 instructions in total
- **METEOR request set the vector size**
  - Reconfiguration at runtime

- **New METEOR-specific vector type**
  - Vector attributes for alignment
  - Section attributes for allocation
- **Smooth software integration for vector engine compatibility**
  - C macros ➔ SIMD intrinsics

Example of 8-bit word NxN matrix multiplication

```
typedef VectorLine Mat[N][N / VECTOR_SIZE];    /* matrix data type */
__attribute__((section(".csram"))) Mat A,B,C;  /* store and align data */

for(int i = 0; i < N; i++)
  for(int j = 0; j < N; j++) {
    int sum = 0; VectorLine tmp;
    for(int k = 0; k < N / VECTOR_SIZE; k++) {
      vmul8(tmp[k].v, A[i][k].v, B[j][k].v);  /* vector instruction */
      vreduce_add8(sum, tmp[k].v);            /* dedicated micro-code */
    }
    C[i][j / VECTOR_SIZE].i8[j % VECTOR_SIZE] = sum;
}
```

OUTLINE

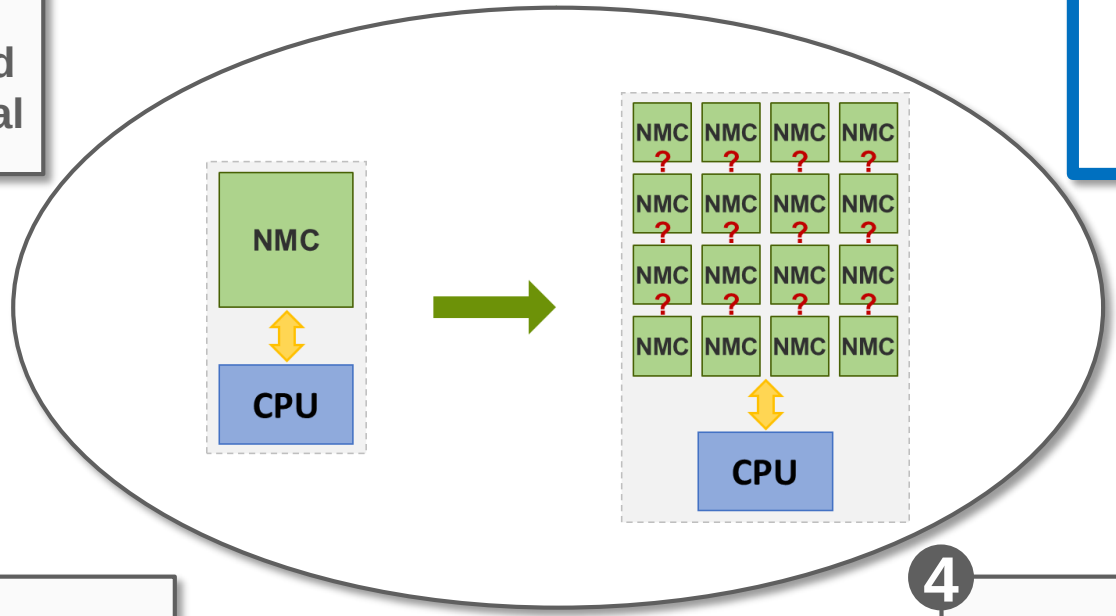
1. METEOR:
A Reconfigurable Memory-based Computing Architecture Proposal

2. Software Integration for Scalable Vector Computing

3. **Design Space Exploration** of the Memory Interconnect

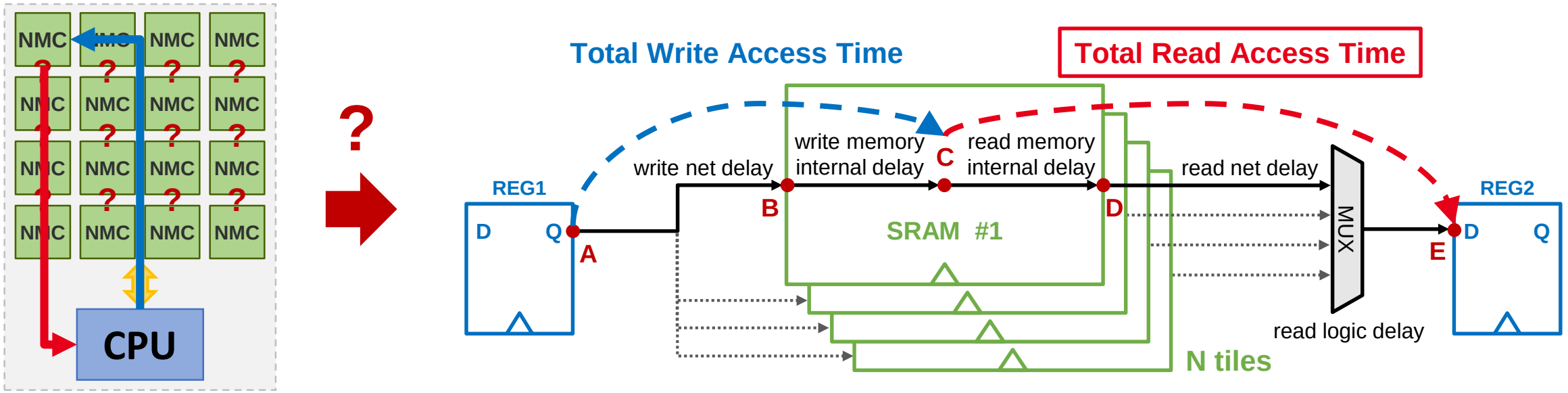
4. ArchSim:
An IMC-NMC Software-Hardware Simulation Platform

5. METEOR Architecture Exploration Results



MEMORY INTERCONNECT & SIZING

MULTIPLE MEMORY TILE EXPLORATION



- Objectives
  - Develop a **memory interconnect model** to design **larger memory architecture**
- WNS (*worst negative slack*) path extraction ➔ **Total Read Access Time**
- Extract **Power**, **Performance** & **Area** (PPA) trade-offs of various **memory architectures**

MEMORY INTERCONNECT & SIZING

EXPLORATION METHODOLOGY

Software & Design Parameters

Specification

- Physical layout
- Memory tile size
- # of tiles

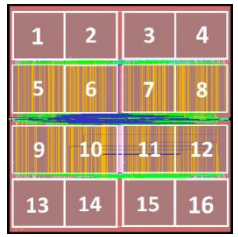
inputs

Physical Digital Design Flow

1. **Behavior Model** (*VHDL*)
2. **Logic Synthesis** (*netlist*)
3. **Place & Route** (*physical layout*)
4. **Power Analysis** (*activity report*)

P&R optimizations

- 75% area density
- WNS: read data path
- 28 nm FDSOI (ST)



extractions

Analysis & Modeling

Wiring Model (*Python*)

Design Tools

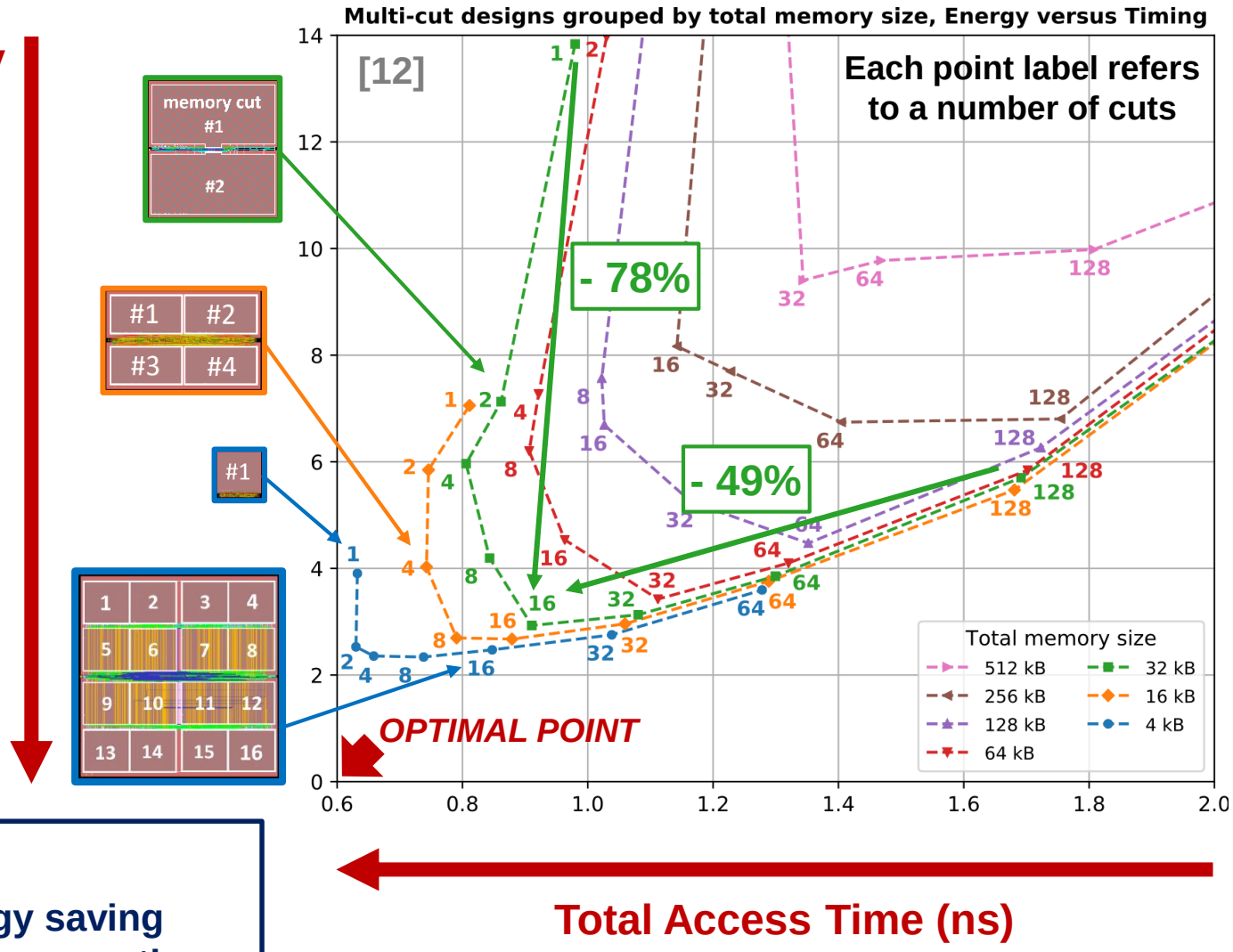
1. ModelSim (*Mentor Graphics*)
2. Design Compiler (*Synopsys*)
3. Innovus (*Cadence*)
4. PrimeTime-PX (*Synopsys*)

- **Exploration of 22 memory architectures (*various size & configurations*)**
- **Model and wire estimations are obtained by polynomial regression algorithms**

MEMORY INTERCONNECT & SIZING

EXPERIMENTAL & MODELING RESULTS

- PPA trade-offs design exploration combining **experimental** & **model** results
- **Fragmenting into smaller cuts:**
  - Reduce the total **dynamic energy** at the expense of **leakage power**
- **Too many small memories:**
  - Increase the **total access time**



Example: for a 32 kB total memory size

- 1-cut ➔ 16-cut memory: 78% better in energy saving
- 128-cut ➔ 16-cut memory: 49% better in read access time

[12] "*Memory Sizing of a Scalable SRAM In-Memory Computing Tile Based Architecture*", R. Gauchi & al, VLSI-SoC, 2019

OUTLINE

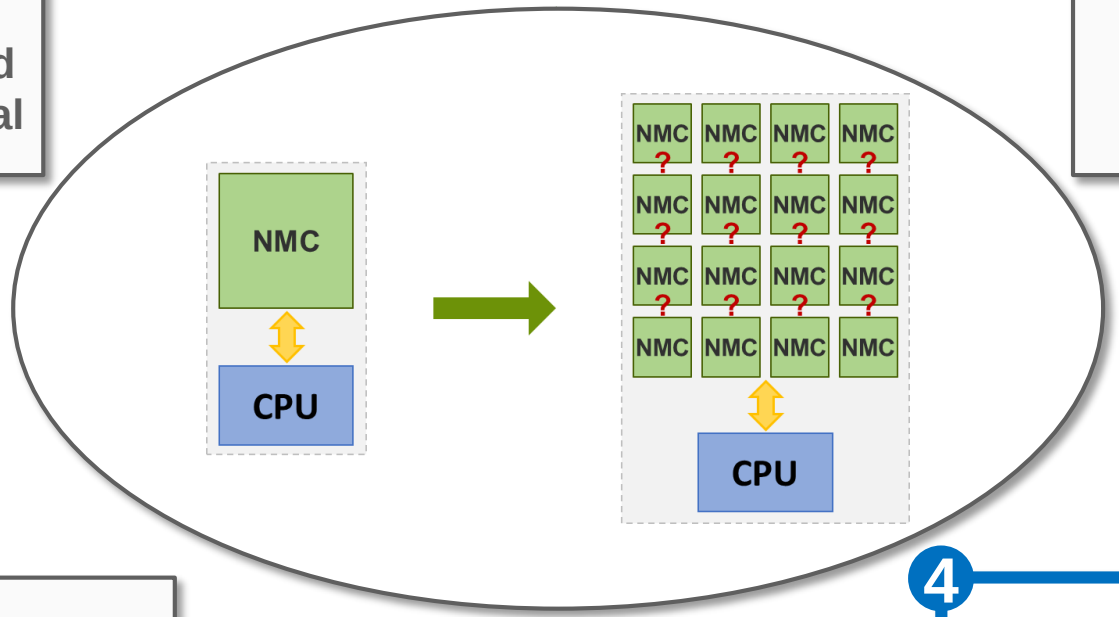
1. METEOR:
A Reconfigurable Memory-based Computing Architecture Proposal

2. Software Integration for Scalable Vector Computing

3. Design Space Exploration of the Memory Interconnect

4. **ArchSim:**
An IMC-NMC Software-Hardware Simulation Platform

5. METEOR Architecture Exploration Results



ARCHSIM

A Software-Hardware Simulation Platform Overview

Hardware Layer

- SystemC/TLM models
- Model of C-SRAM, Interconnects & Pipeline Control
- Instruction Set Simulator (ISS)
- Timing & Power annotations

Instruction Set Architecture (ISA)

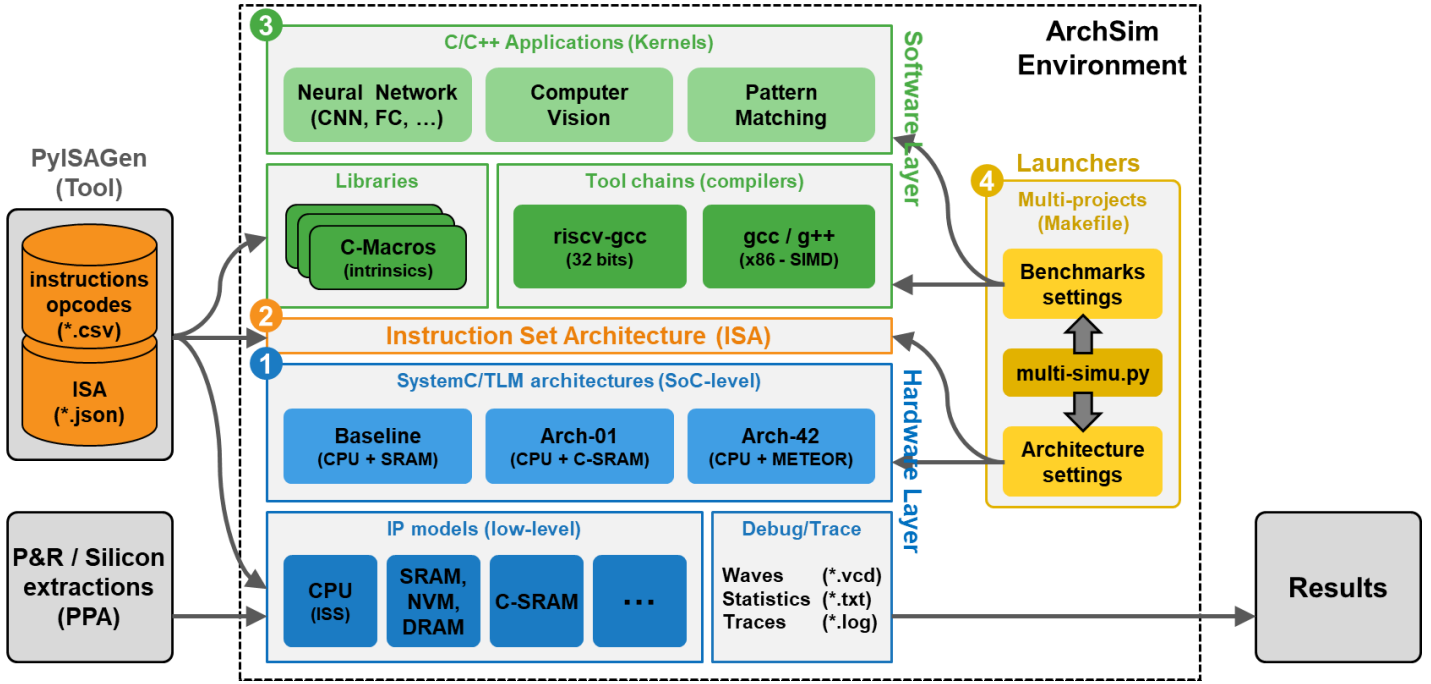
- HW/SW interoperability
- PyISAGen: automatic ISA generator for HW & SW layers compatibility

Software Layer

- C-SRAM/METEOR ISA library
- Linker script for memory mapping
- User program (kernels) with SIMD compatibility

Launchers & Results

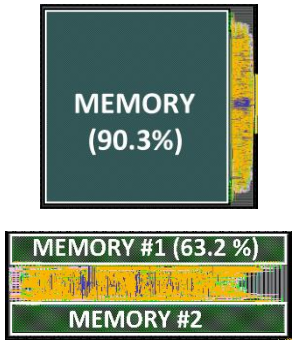
- Launch compilation & benchmark flows
- Extract system energy & performance



ARCHSIM

How it is Working ?

C-SRAM PPA numbers
from P&R extractions
(GF 22nm FDSOI)



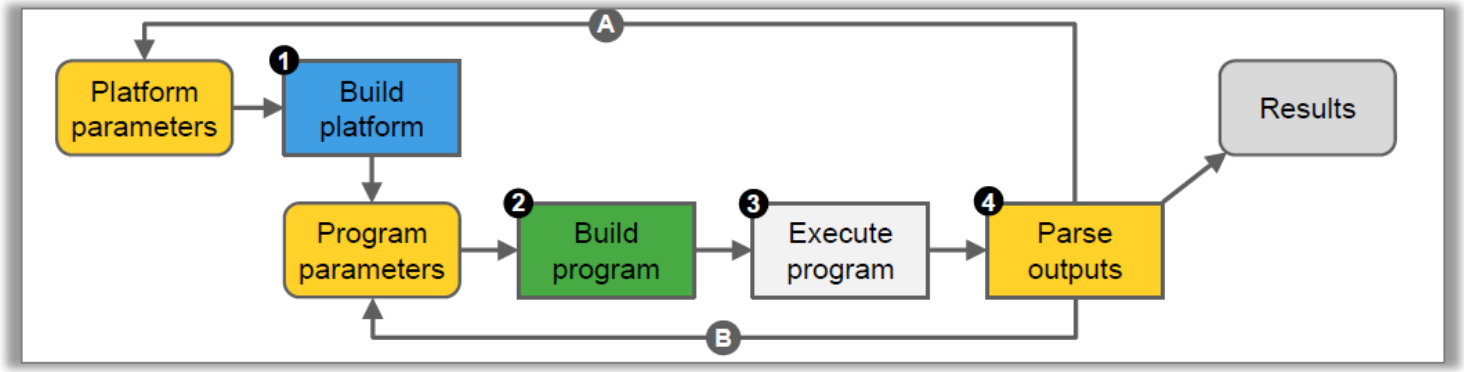
Multi-tile interconnect and configuration

C-SRAM (versus baseline SRAM)			
Leakage (%)	Dynamic Energy (%)	Density (%)	Area (%)
+81 %	+14 %	-38 %	+62 %
+73 %	+13 %	-35 %	+53 %
+57 %	+11 %	-30 %	+42 %
+48 %	+11 %	-17 %	+20 %
+32 %	+11 %	-10 %	+11 %
+30 %	+10 %	-7 %	+8 %

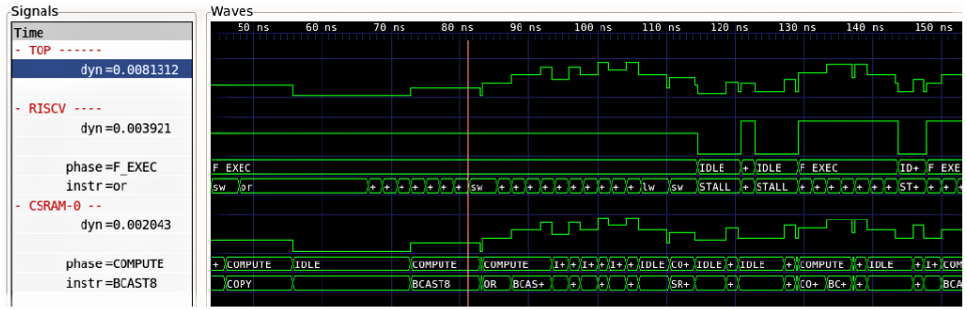
User program (RISCV/SIMD-compatible)

```
typedef VectorLine Mat[N][N / VECTOR_SIZE];    /* matrix data type */
__attribute__((section(".csram"))) Mat A,B,C;  /* store and align data */

for(int i = 0; i < N; i++)
  for(int j = 0; j < N; j++) {
    int sum = 0; VectorLine tmp;
    for(int k = 0; k < N / VECTOR_SIZE; k++) {
      vmul8(tmp[k].v, A[i][k].v, B[j][k].v);  /* vector instruction */
      vreduce_add8(sum, tmp[k].v);            /* dedicated micro-code */
    }
    C[i][j / VECTOR_SIZE].i8[j % VECTOR_SIZE] = sum;
}
```



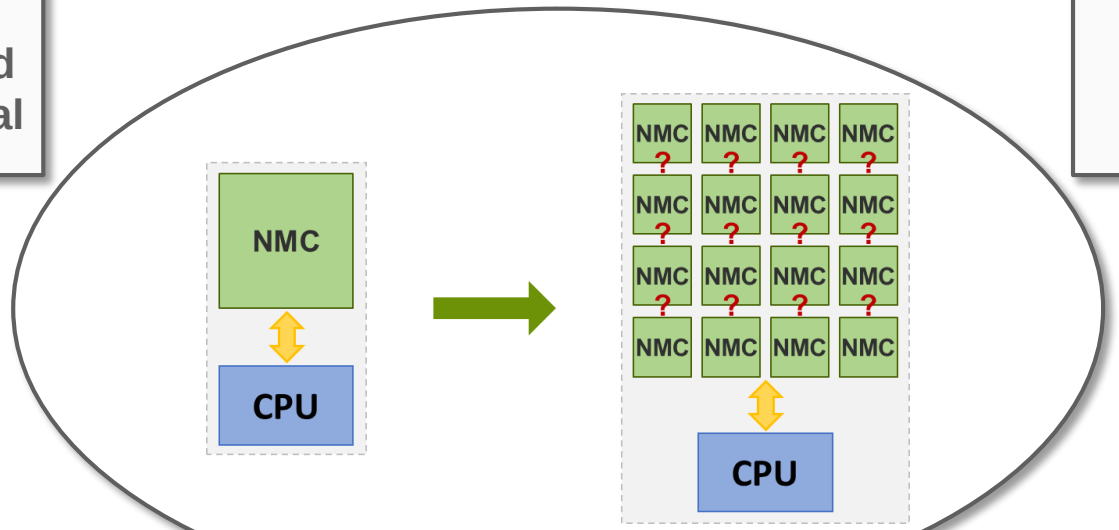
System energy & performance extractions



- **Platform modeling inputs:**
  - Circuit level numbers from C-SRAM, RISC-V core
  - Memory interconnect & tile configuration
- **Run user program onto the platform flow**
- **Extract and Visualize the PPA results**

OUTLINE

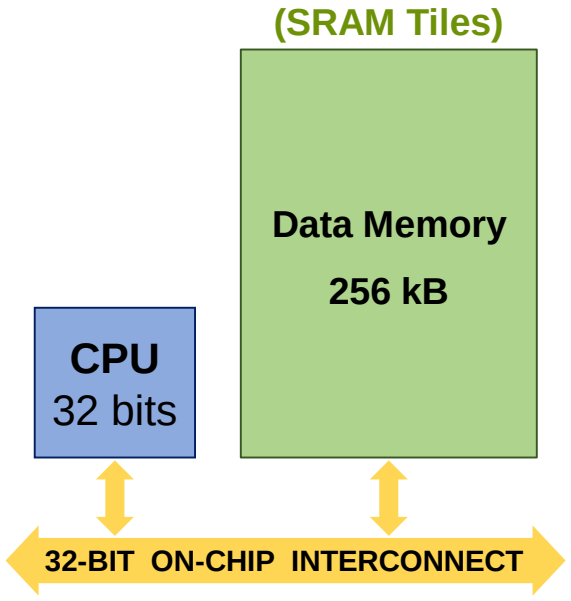
1. METEOR: A Reconfigurable Memory-based Computing Architecture Proposal
2. Software Integration for Scalable Vector Computing
3. Design Space Exploration of the Memory Interconnect
4. ArchSim: An IMC-NMC Software-Hardware Simulation Platform
5. METEOR Architecture Exploration Results



ARCHITECTURE EXPLORATION RESULTS

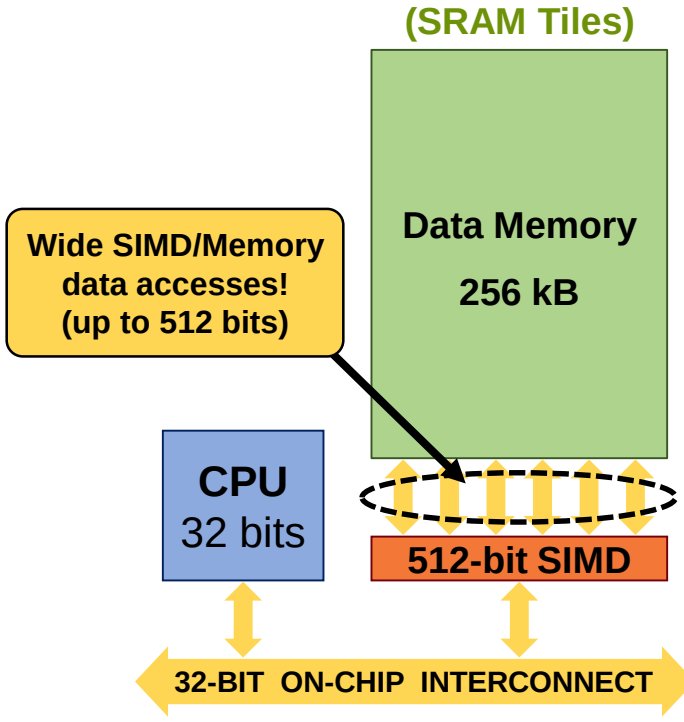
EXPLORATION BENCHMARKING

32-bit Scalar Architecture (Baseline)



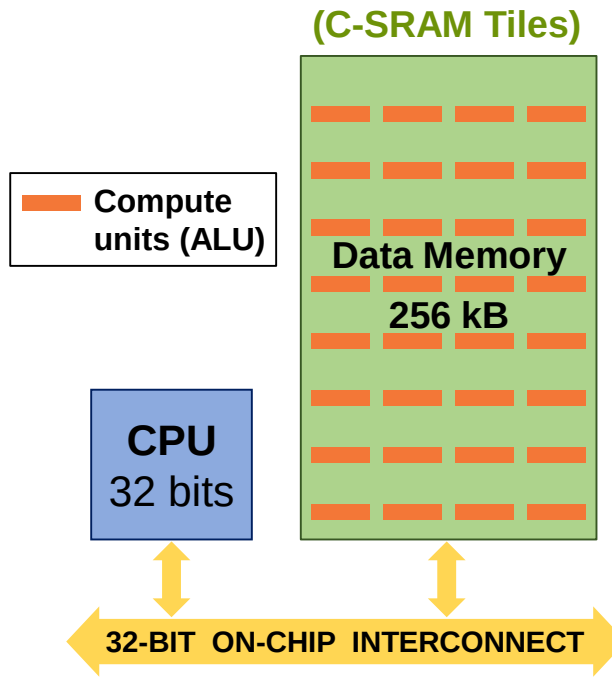
Evaluated using the proposed HW/SW simulation platform [13]

512-bit SIMD Architecture



Evaluated using Pin [14], an Intel simulation platform

METEOR Architecture (up to 8,192 bits)



Evaluated using the proposed HW/SW simulation platform [13]

[13] "*Reconfigurable Tiles of Computing-In-Memory SRAM Architecture for Scalable Vectorization*", R. Gauchi et al., ISLPED, 2020
[14] "*Pin: Building Customized Program Analysis Tools with Dynamic Instrumentation*", C.-K. Luk et al., ACM SIGPLAN Notices, 2005

ARCHITECTURE EXPLORATION RESULTS

KERNEL OVERVIEW

1. Linear Complexity Kernel ➔ Shift-OR

- **Application:** Bio-informatics
- **NMC instructions:** OR, AND, SRL, SLL
- **Description:** *Match a pattern in a DNA sequence*

2. Quadratic Complexity Kernel ➔ Atax [15]

- **Application:** Image computing & Neural Network
- **NMC instructions:** ADD8, MUL8
- **Description:** *Matrix transpose and vector multiplication*

3. Cubic Complexity Kernel ➔ GEMM [15]

- **Application:** Neural Network
- **NMC instructions:** ADD8, MUL8
- **Description:** *matrix multiplication as $C = \alpha.A.B + \beta.C$*

Application Scope	Kernel Name	Operations	Memory Footprint
DNA pattern searching	Shift-OR	**41n**	n
Computer Vision	Atax	**$4n^2$**	$n^2 + 3n$
Neural Network	GEMM	**$3n^3 + n^2$**	$3n^2$

Kernel Complexity:

Linear time **Quadratic time** **Cubic time**

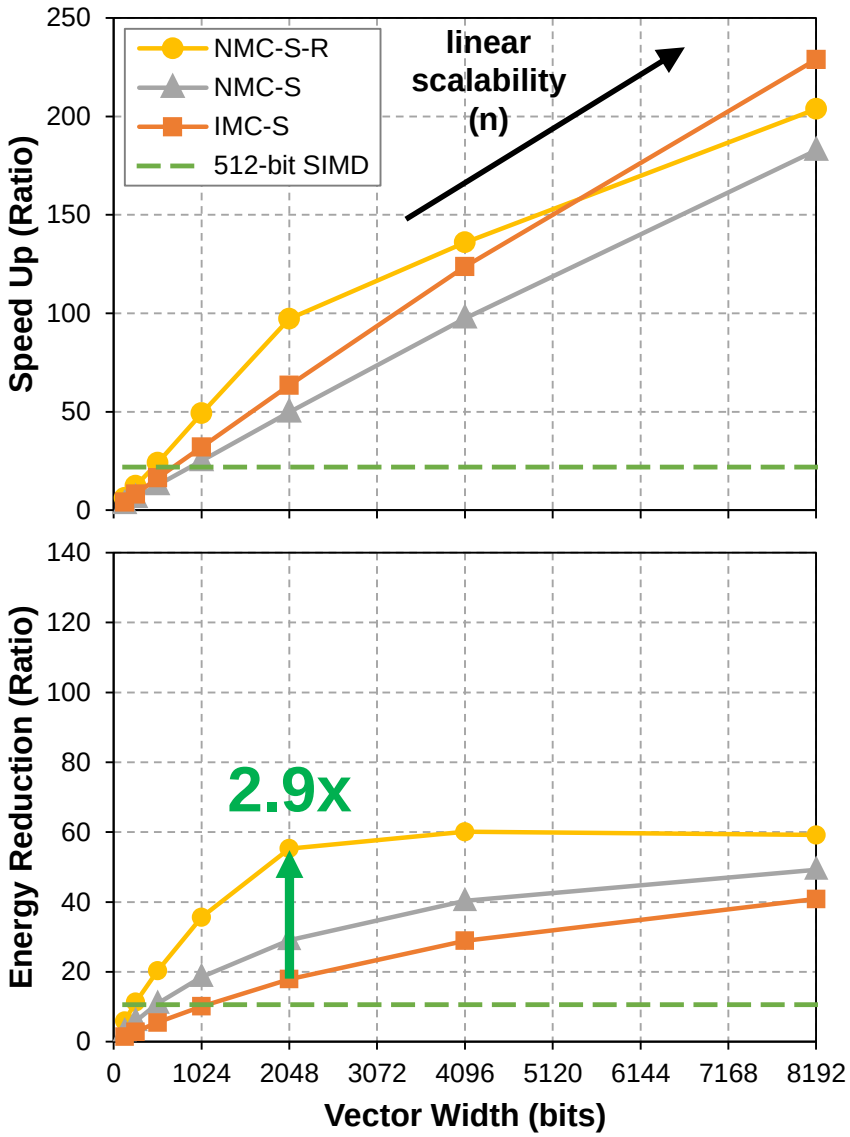
*The problem size parameters are assumed to take the same input value **n***

[15] "*PolyBench/C, the Polyhedral Benchmark suite*", L.-N. Pouchet., 2015

ARCHITECTURE EXPLORATION RESULTS

IN- OR NEAR- MEMORY COMPUTING?

Architecture name	*Vector configuration*	*Pipeline support*	*Clock cycles*
NMC-S-R	Static	YES	5
NMC-S	Static	NO	5
IMC-S	Static	NO	3

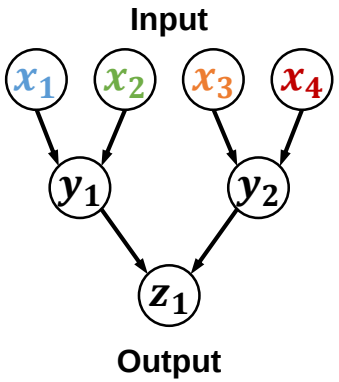


- **Exploration of a linear complexity kernel (Shift-OR)**
  - **512-bit NMC** better in performance & energy than **512-bit SIMD**
  - **NMC** is **2.9x** better in energy savings compared to **IMC** (2048 bits)
- **Pipeline support (Register)**
  - Operand forwarding ➔ reduce timings
  - Memory access savings ➔ reduce energy
- **IMC is not suitable for high computational complexity**
  - Multiplier implementation requires a high latency

ARCHITECTURE EXPLORATION RESULTS

BENEFITS OF DYNAMIC VECTOR RECONFIGURATION

Architecture	Vector	Register
NMC-D-R	Dynamic	YES
NMC-D	Dynamic	NO
NMC-S-R	Static	YES
NMC-S	Static	NO



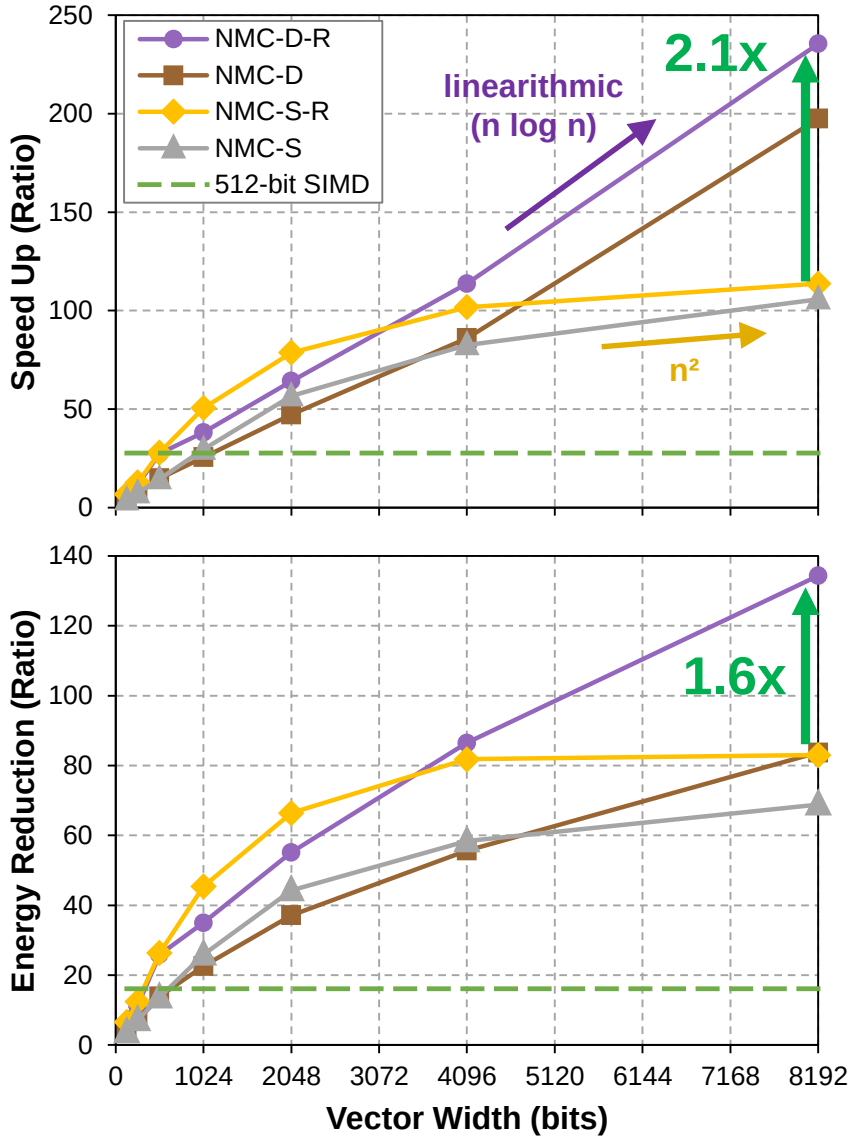
$$z_1 = x_1 + x_2 + x_3 + x_4$$

```
#define vreduce_add8(_8b_dest, _v_src) \
do { \
  _cm_copy_rm(    R0, _v_src ); \
  set_mvl(              1024 ); \
  _cm_add8_rrr(   R0, R0, R1 ); \
  set_mvl(               512 ); \
  _cm_add8_rrr(   R0, R0, R2 ); \
  _cm_hswap64_rr( R1,     R0 ); \
  _cm_add8_rrr(   R0, R0, R1 ); \
  _cm_hswap32_rr( R1,     R0 ); \
  _cm_add8_rrr(   R0, R0, R1 ); \
  _cm_srli32_rr(  R1, R0, 16 ); \
  _cm_add8_rrr(   R0, R0, R1 ); \
  _cm_srli16_rr(  R1, R0,  8 ); \
  _cm_add8_rrr(   R0, R0, R1 ); \
  set_mvl(              2048 ); \
  _8b_dest = CRegs[0]->i8[0]  + \
             CRegs[0]->i8[16] + \
             CRegs[0]->i8[32] + \
             CRegs[0]->i8[48]; \
} while(0)
```

METEOR: dynamic vector reconfiguration

C-SRAM: internal registers & instructions

Last CPU reduction (*5 memory accesses*)

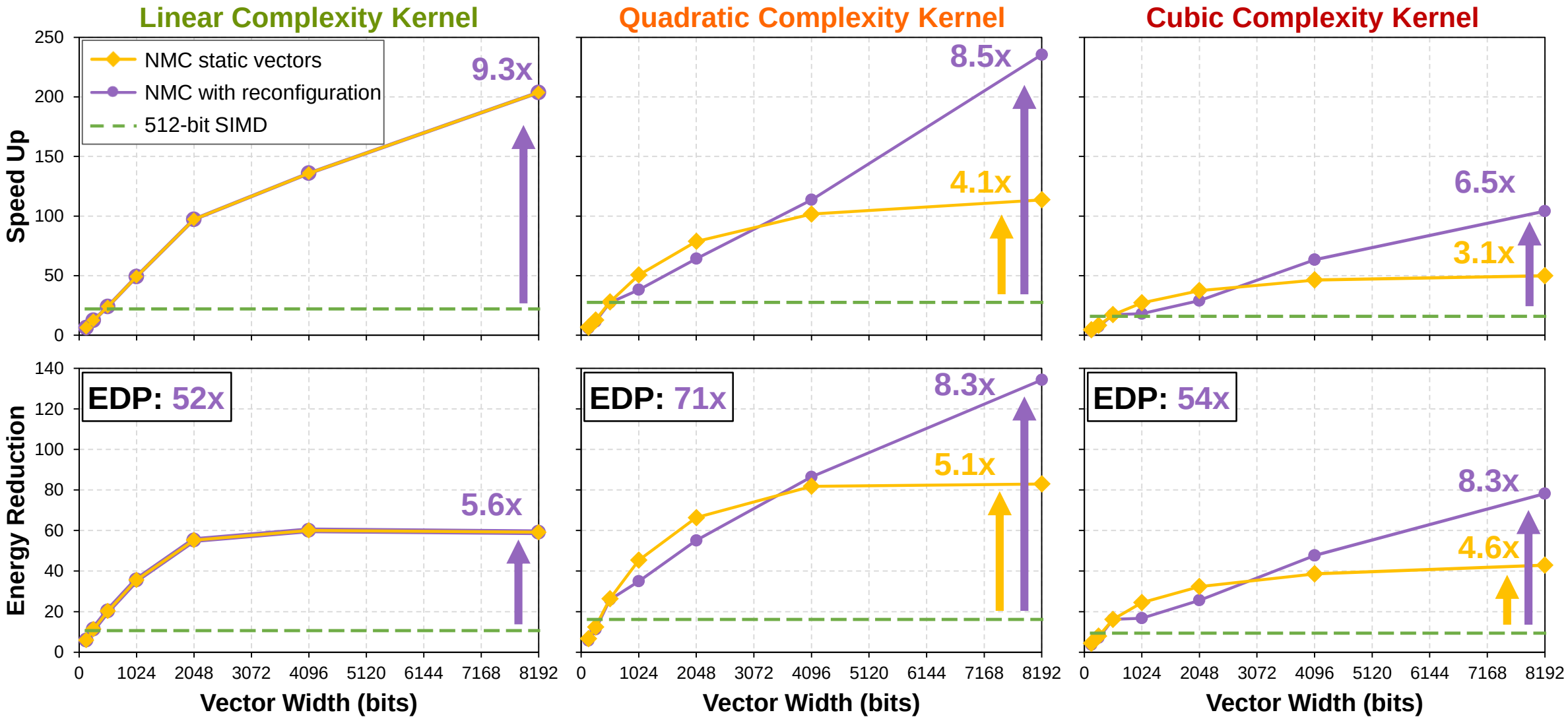


- **Exploration of a quadratic complexity kernel (Atax)**
  - Addition reduce operation (METEOR-specific)
  - Combine **METEOR reconfigurability** & **C-SRAM instructions**

- **METEOR vector reconfiguration**
  - Better performance by **2.1x** & energy consumption by **1.6x** compared to static vectors (*NMC only*)

ARCHITECTURE EXPLORATION RESULTS

COMPARED TO THE 32-BIT SCALAR ARCHITECTURE (BASELINE)



ARCHITECTURE EXPLORATION RESULTS

PERFORMANCE SUMMARY

METEOR architecture compared to 512-bit SIMD architecture

Kernel Family	Speed Up			Energy Reduction			Energy Delay Product		
	W_{512}	W_{2048}	W_{Best}	W_{512}	W_{2048}	W_{Best}	W_{512}	W_{2048}	W_{Best}
Linear	0.8	3.7	9.3	1.1	4.3	5.6	0.9	15.8	51.8
Quadratic	0.9	2.4	8.5	1.4	3.5	8.3	1.4	8.6	71.2
Cubic	0.9	2.2	4.0	1.4	3.1	5.1	1.3	6.7	20.2

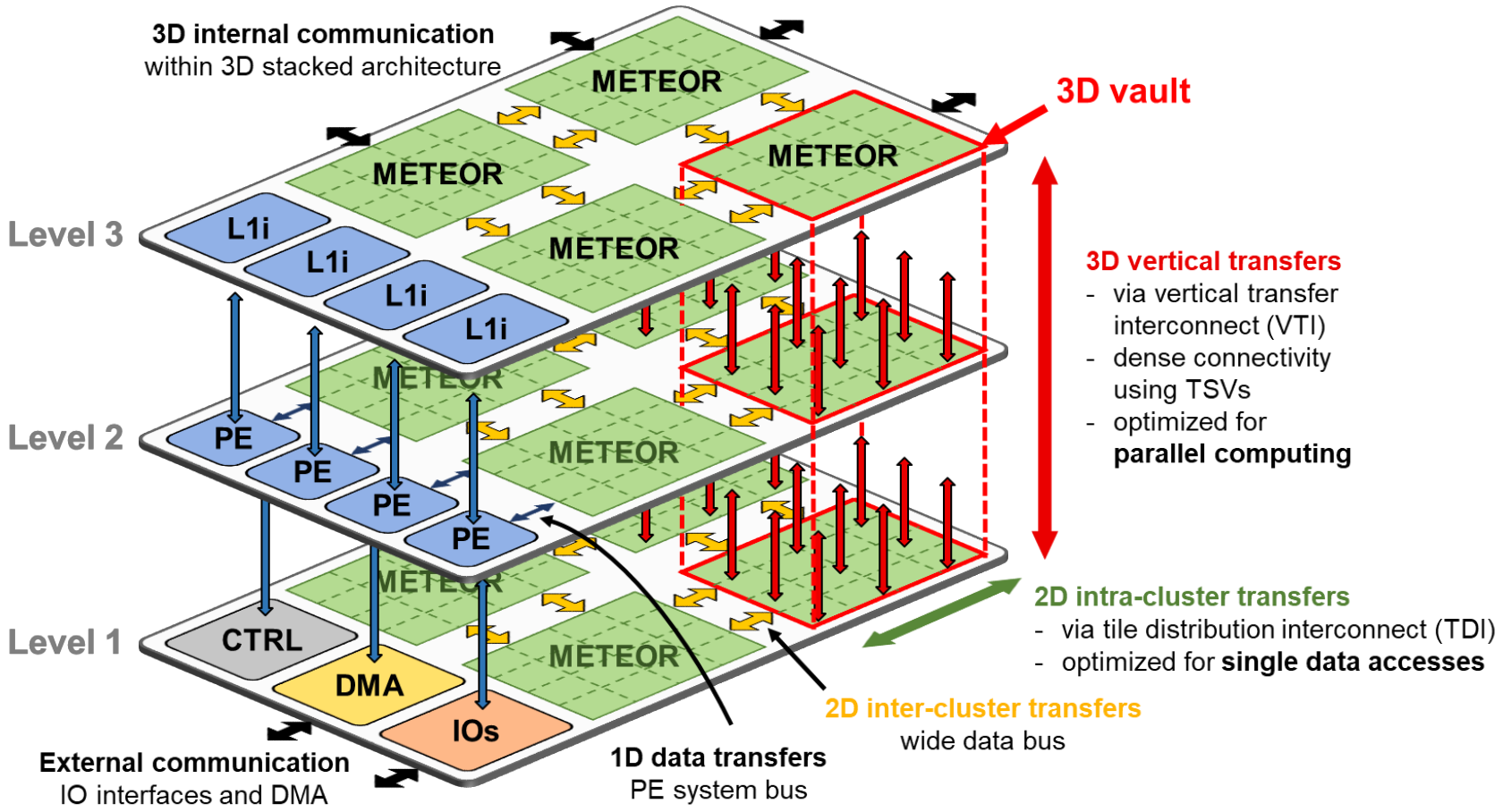
52x

71x

W_{512} ➔ vector width of 512 bits
W_{2048} ➔ vector width of 2048 bits
W_{Best} ➔ the largest vector width (up to 8192 bits)

- **Maximum performance gains, thanks to:**
  - Large vectorization in the application kernel (*up to 8192 bits*)
  - Vertical data communication between memory tiles
  - Reduced data movement (CPU⇔Memory) ➔ METEOR dynamic reconfiguration

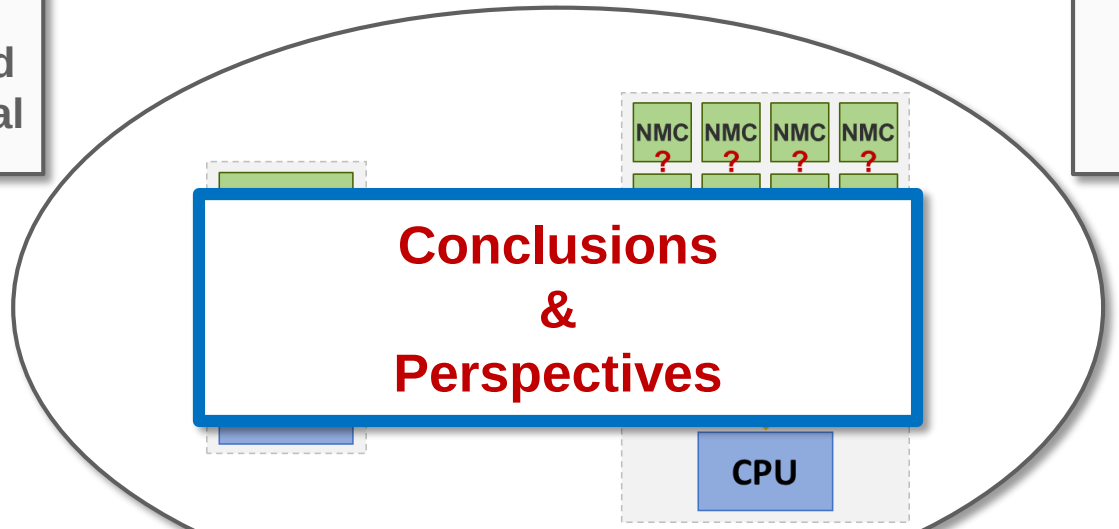
TOWARD A 3D METEOR ARCHITECTURE



- **Vector transfers** for immersed computing through **3D vertical paths** & **2D intra-cluster**
- Combine **best memory interconnect** trade-off and **high data parallelism**

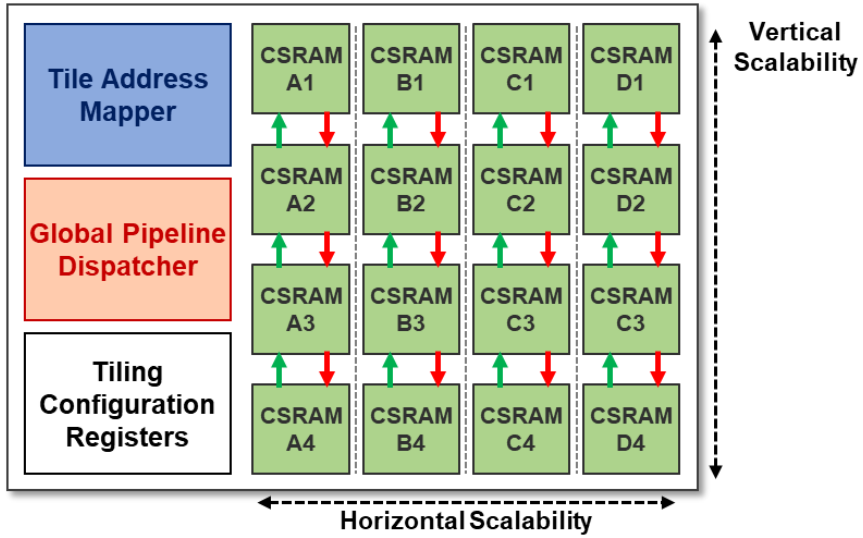
OUTLINE

1. METEOR: A Reconfigurable Memory-based Computing Architecture Proposal
2. Software Integration for Scalable Vector Computing
3. Design Space Exploration of the Memory Interconnect
4. ArchSim: An IMC-NMC Software-Hardware Simulation Platform
5. METEOR Architecture Exploration Results



Conclusions & Perspectives

CONCLUSIONS



- **In the context of the More-than-Moore architecture**
  - **Single memory** tile is **not scalable** for data-centric applications
  - Computation immersed in memory approaches are **limited without** using a **dense inter-connect**
  - **Extended Benefits** of NMC thanks to local **configurable control**
- **Contributions**
  - **METEOR: 2D Reconfigurable NMC-based Architecture** (*patented*)
  - **Specific ISA** for software integration of IMC & NMC architectures
  - **Memory interconnect model** for multi-tile memory architecture sizing
  - **ArchSim: hardware-software simulation platform** for architecture explorations
  - **512-bit SIMD** vs. **METEOR architecture**
    - EDP gains up to **52x** for **linear** and **71x** for **quadratic** complexity kernels
- **Perspectives**
  - Design & measure METEOR architecture
  - Explore METEOR within complete **memory hierarchy** (off chip access, DMA, caches, etc.)
  - Continue to explore METEOR immersed in **3D architectures**

PHD PUBLICATIONS & PATENTS

- **Papers**
  - **R. Gauchi**, V. Egloff, M. Kooli, J.-P. Noel, B. Giraud, P. Vivet, S. Mitra and H.-P. Charles, *Reconfigurable Tiles of Computing-In-Memory SRAM Architecture for Scalable Vectorization,* ISLPED, 2020
  - J.-P. Noel, M. Pezzin, **R. Gauchi**, J.-F. Christmann, M. Kooli, H.-P. Charles, L. Ciampolini, M. Diallo, F. Lepin, B. Blampey, P. Vivet, S. Mitra and B. Giraud, *A 35.6TOPS/W/mm2 3-Stage Pipelined Computational SRAM with adjustable Form Factor for Highly Data-Centric Applications*, L-SSC, 2020
  - J.-P. Noel, V. Egloff, M. Kooli, **R. Gauchi**, J.-M. Portal, H.-P. Charles, P. Vivet and B. Giraud, *Computational SRAM Design Automation using Pushed-Rule Bitcells for Energy-Efficient Vector Processing*, DATE, 2020
  - **R. Gauchi**, M. Kooli, P. Vivet, J.-P. Noel, E. Beigné, S. Mitra and H.-P. Charles, *Memory Sizing of a Scalable SRAM In-Memory Computing Tile Based Architecture*, VLSI-SoC, 2019
- **Patents**
  - **R. Gauchi**, P. Vivet, H.-P. Charles and S. Mitra, *Module mémoire reconfigurable adapté à mettre en œuvre des opérations de calcul*, 2020
  - M. Kooli, **R. Gauchi**, and P. Vivet., Module mémoire adapté à mettre en œuvre des fonctions de calcul". FR2014174. 2020.
- **Posters**
  - **R. Gauchi**, V. Egloff, M. Kooli, P. Vivet, J.-P. Noel, S. Mitra and H.-P. Charles, *Exploration of a Scalable Vector-based In-Memory Computing Architecture via a System-on-Chip Evaluation Framework*, DAC, WiP Poster Session, 2020
  - **R. Gauchi**, V. Egloff, M. Kooli, J.-P. Noel, B. Giraud, P. Vivet, S. Mitra and H.-P. Charles, *Exploration of a Scalable Vector-Tile-based In-Memory Computing Architecture*, DATE, Computation-In-Memory Workshop, 2020

THANK YOU
FOR
YOUR ATTENTION

BACKUP SLIDES

Commissariat à l'énergie atomique et aux énergies alternatives
Institut List | CEA SACLAY NANO-INNOV | BAT. 861 – PC142
91191 Gif-sur-Yvette Cedex - FRANCE
www-list.cea.fr

IN-MEMORY COMPUTING (IMC)

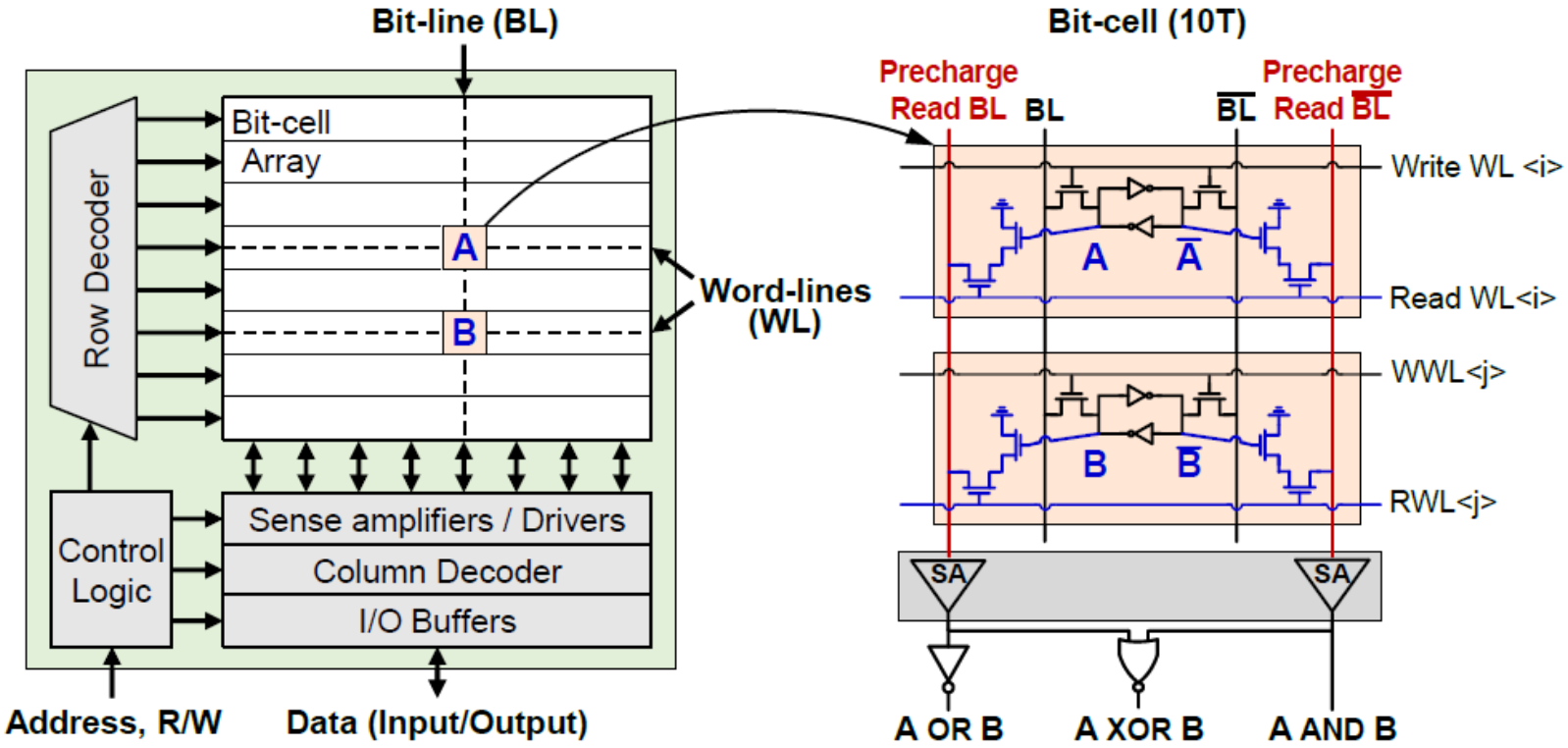


Figure 2.4: Modified SRAM architecture to enable In-Memory Computing (IMC) [52].

INSTRUCTION SET ARCHITECTURE (ISA)

Figures 5.3 & 5.4: IMC/NMC ISA integrated on a standard 32-bit system bus.

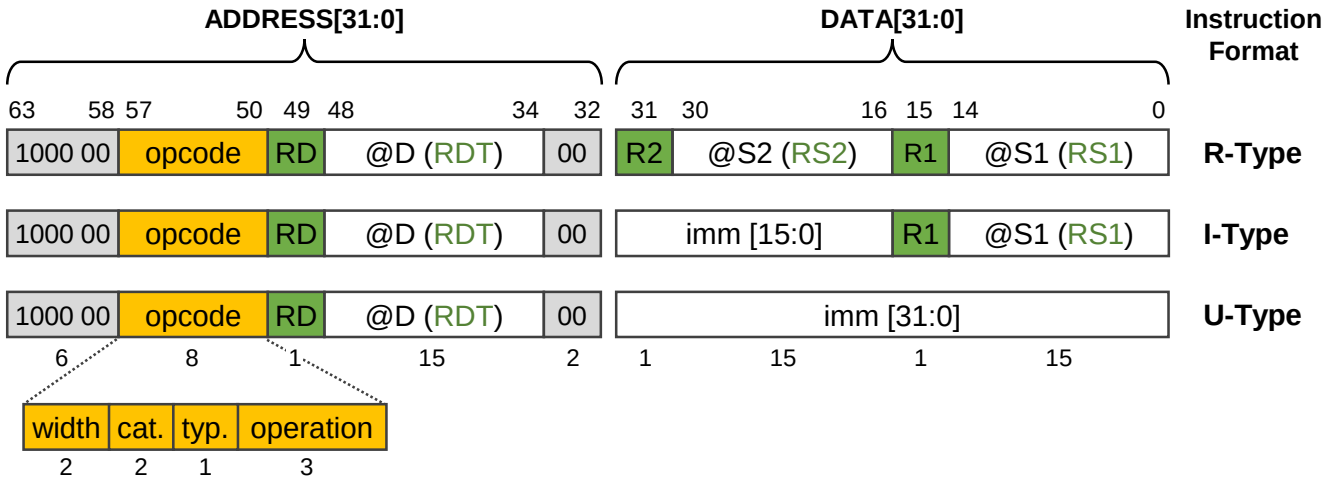


Table 5.2: Instruction summary of METEOR (54 instructions in total).

Category	Type	Width (bits)	Operations (28)
memory	I	Line	copy, hswap64, hswap128
	R	8, 16, 32	copyeq, copygeq, copygt, copyleq, copylt, copyneq
	U	8, 16, 32	bcast
logical	I	8, 16, 32	slli, srli
	I	Line	not, redor
	R	Line	and, or, xor, nand, nor, xnor
arithmetical	I	8, 16, 32	abs
	R	8, 16, 32	add, sub, cmp
	R	8	fxadd, fxmul, mul
CSR	U	32	vreg

- *RD ➔ Register Destination Enable
- *R1 ➔ Register Source 1 Enable
- *R2 ➔ Register Source 2 Enable

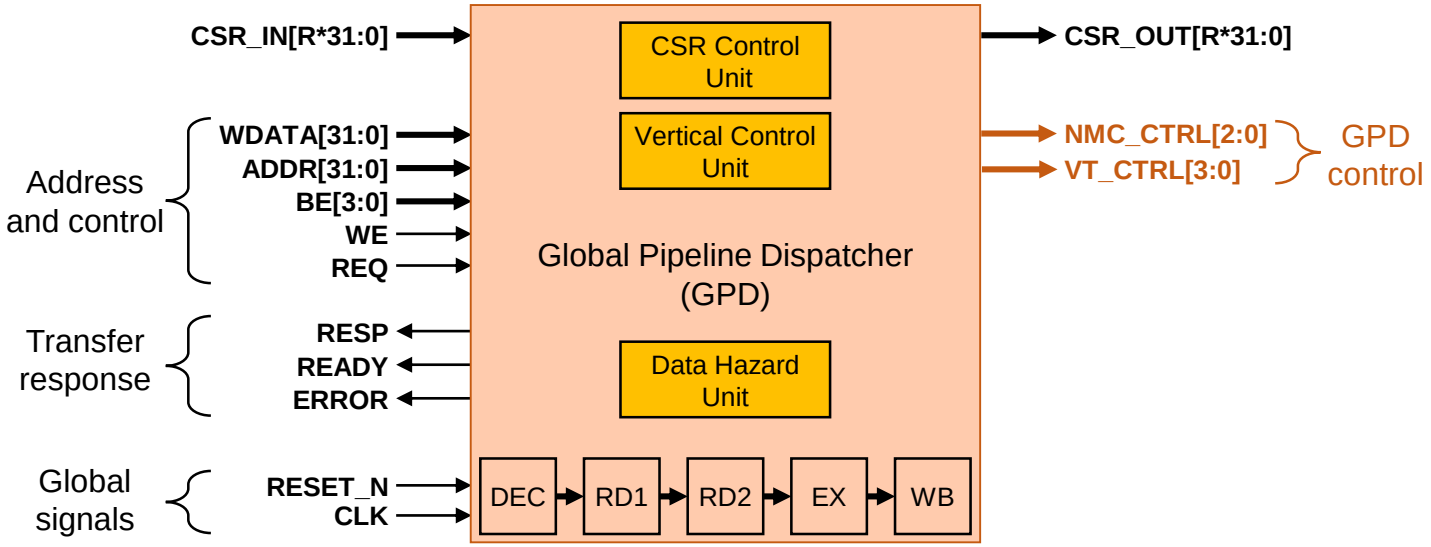
MEMORY INTERCONNECT MODEL

EXTRACTED NUMBERS EXAMPLE FOR A 4-KB SRAM

Table 7.3: Additional wiring cost of a 4-kB SRAM in a multi-tile architecture.

Total Memory Size (kB)	# of cut	Timing (%)	Dynamic Energy (%)	Leakage (×)
4	1	10 %	+0.1 %	1.02 ×
8	2	17 %	1 %	2.02 ×
16	4	25 %	3 %	4.02 ×
32	8	34 %	7 %	8.02 ×
64	16	43 %	14 %	16.02 ×
128	32	53 %	26 %	32.02 ×
256	64	61 %	42 %	64.02 ×
512	128	71 %	61 %	128.02 ×

GLOBAL PIPELINE DISPATCHER (GPD)



Targeted pipeline stream (with vertical transfer)

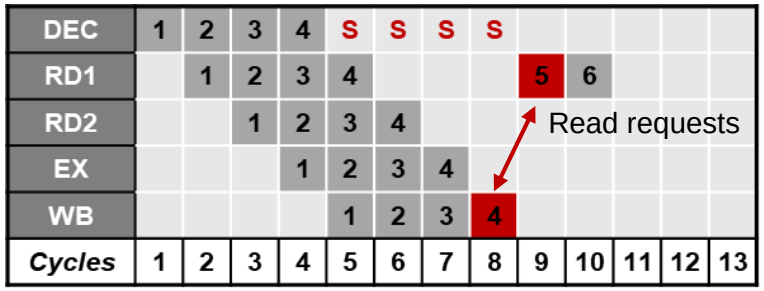
Global Pipeline Dispatcher

- Can stall the CPU via the load/store unit
- Handle address conflicts between C-SRAM instructions and memory reads/writes
- No forwarding control & no reordering (it's not a processor)

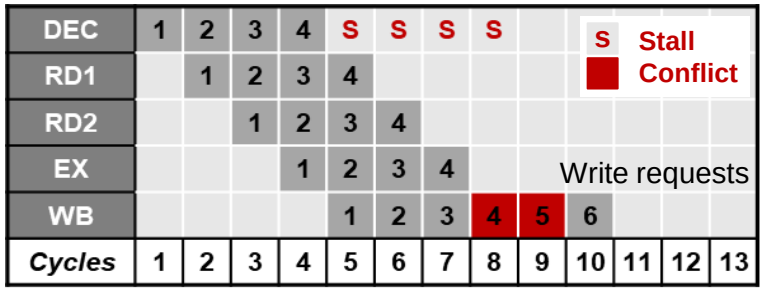
- **Data Hazard Unit (*Stall Control*)**
  - Data hazards: **RAW** (Read After Write), **WAW** (Write After Write), **WAR** (Write After Read)
  - No forwarding (*connect new value directly to the next stage*) & no reordering (it's not a processor)

- **Vertical Control Unit (*Synchronize Transfers*)**
  - In case of vertical data transfer, can stall the processor

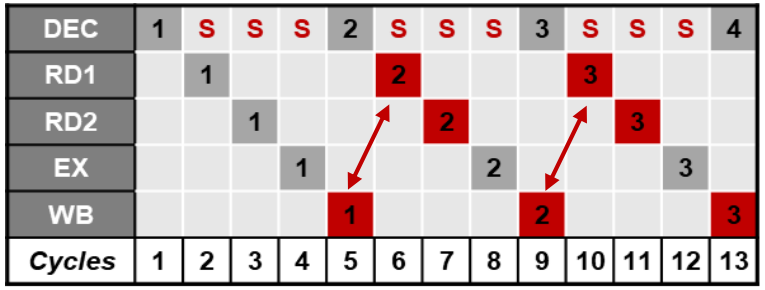
GPD TO AVOID DATA HAZARDS



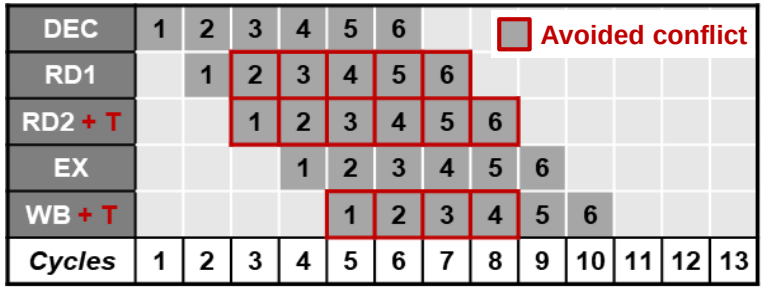
5-stage pipeline Read-After-Write (RAW) hazard



5-stage pipeline Write-After-Write (WAW) hazard

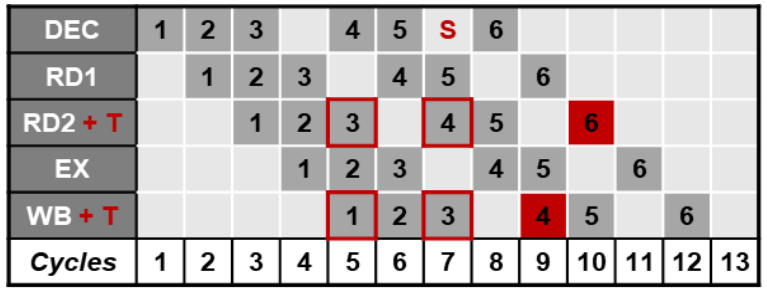


5-stage pipeline (1-port SRAM)

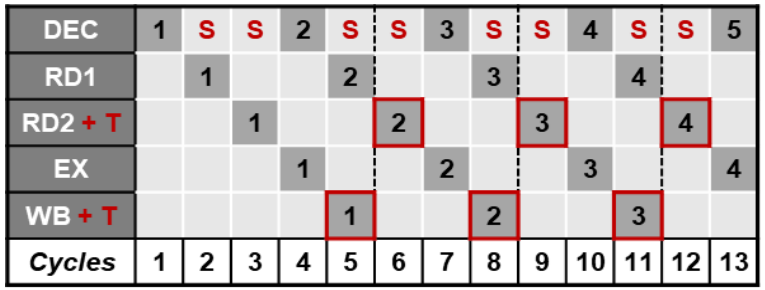


5-stage pipeline with vertical transfer (1-port SRAM)

GPD TO AVOID VECTOR TRANSFERS CONFLICTS



Up-Up or Down-Down Transfer Hazards



Tiling Reconfiguration in the pipeline flow

PYISAGEN ENVIRONMENT