Memory Sizing of a Scalable SRAM In-Memory Computing Tile Based Architecture

*Roman Gauchi[1], M. Kooli[1], P. Vivet[1], J.-P. Noel[1], E. Beigné[1], S. Mitra[3], H.-P. Charles[2]

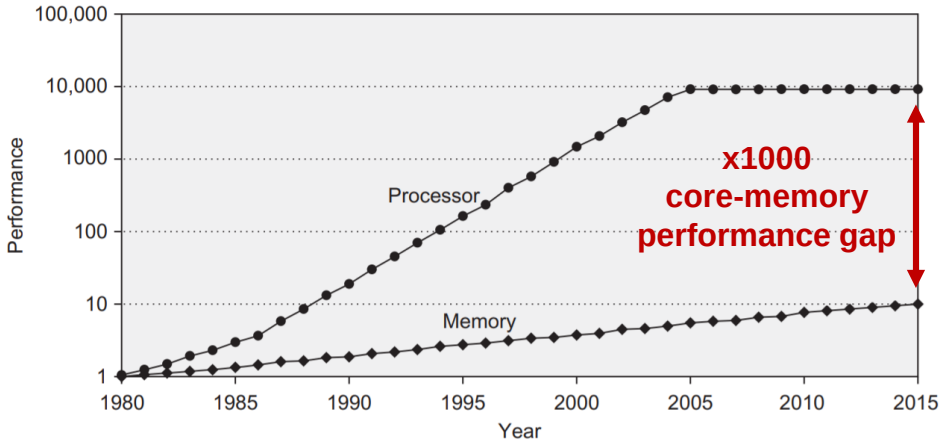
[1] Université Grenoble Alpes, CEA Leti, [2] University Grenoble Alpes, CEA List, [3] Stanford University

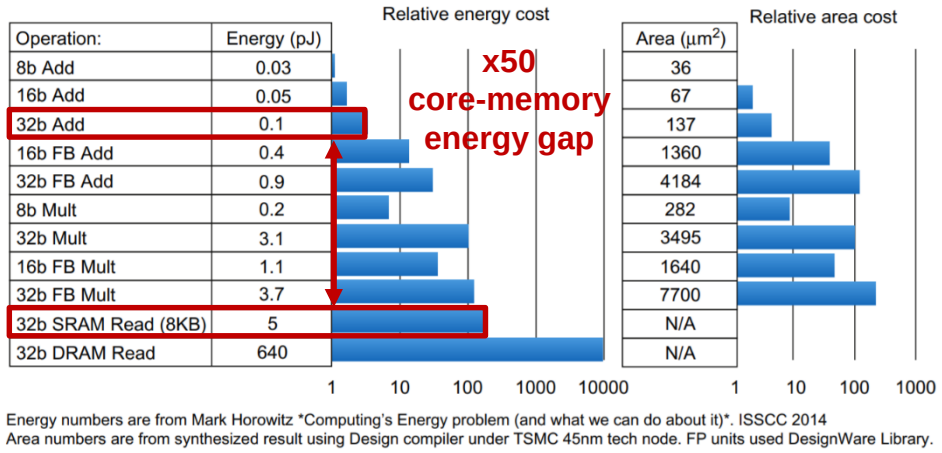
*Speaker

October 9, 2019

MOTIVATIONS: THE MEMORY WALL

Memory Wall (performance impacts) ➔ **But also** ➔ **Energy Wall** (power consumption impact)





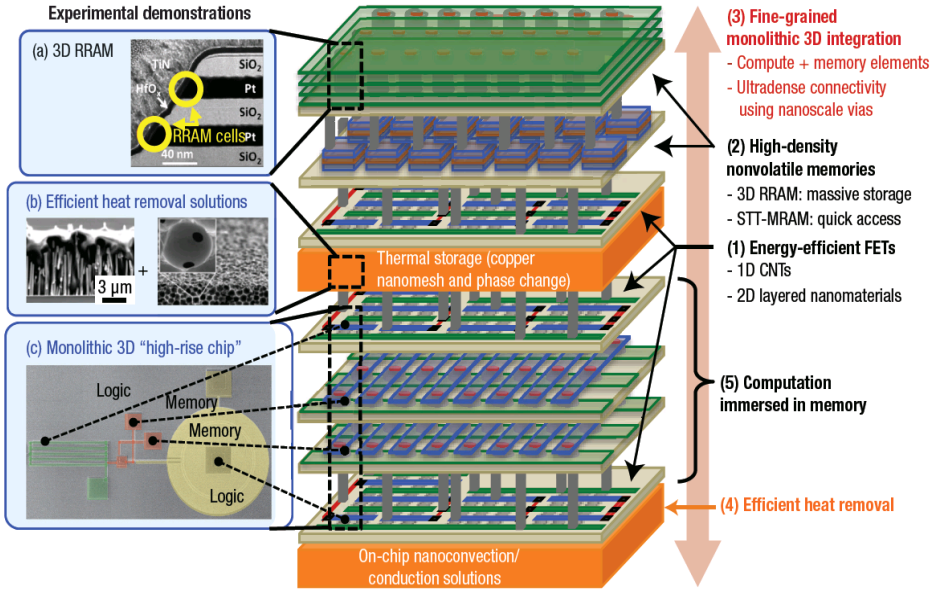
Energy numbers are from Mark Horowitz *Computing's Energy problem (and what we can do about it)*. ISSCC 2014
Area numbers are from synthesized result using Design compiler under TSMC 45nm tech node. FP units used DesignWare Library.

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

- **Computation latency** becomes limited by the **memory access time**…
- …and **memory energy** is much higher than the **computational energy**
- Moore's Law predicts that technology **will not scale anymore** (CMOS)

MOTIVATIONS: NEW ARCHITECTURES

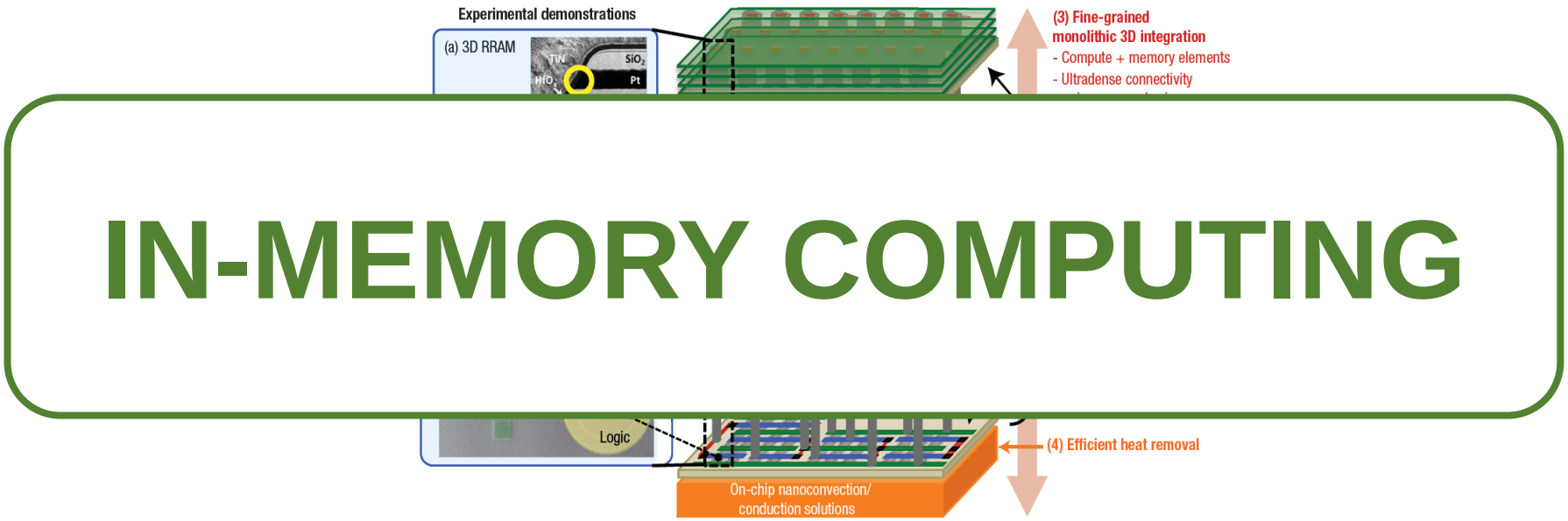
- **Problem**: Modern applications require **more and more data** to be processed
- **Possible solution**: New emerging **architectures** to break the "memory wall"



*M. Sabry Aly et al., **N3XT** Architecture, Stanford University, 2015*

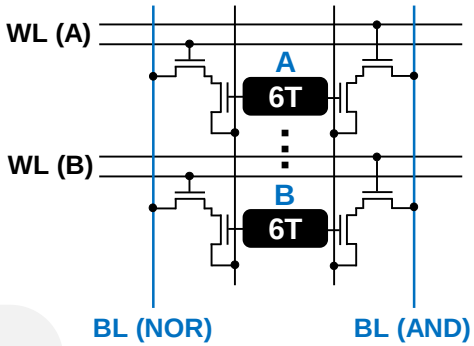
MOTIVATIONS: NEW ARCHITECTURES

- **Problem**: Modern applications require **more and more data** to be processed
- **Possible solution**: New emerging **architectures** to break the "memory wall"



*M. Sabry Aly et al., **N3XT** Architecture, Stanford University, 2015*

IN-MEMORY OR NEAR-MEMORY COMPUTING?



"pre-computation"
[1-4]

IMC

bit-cells array

Basic Logic
(nor, and)

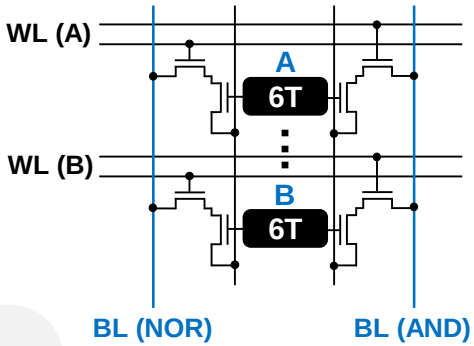
Interface levels

Computation complexity

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

[1] K. C. Akyel, DRC², 2016
[2] S. Aga, Compute Caches, 2017
[3] Y. Zhang, Recryptor, 2018
[4] A. Agrawal, X-SRAM, 2018

IN-MEMORY OR NEAR-MEMORY COMPUTING?



"pre-computation"
[1-4]

IMC
IMC++

bit-cells array

word-line IOs

Basic Logic
(nor, and)

Logic & Arithmetic
(nand, or, add, sub, …)

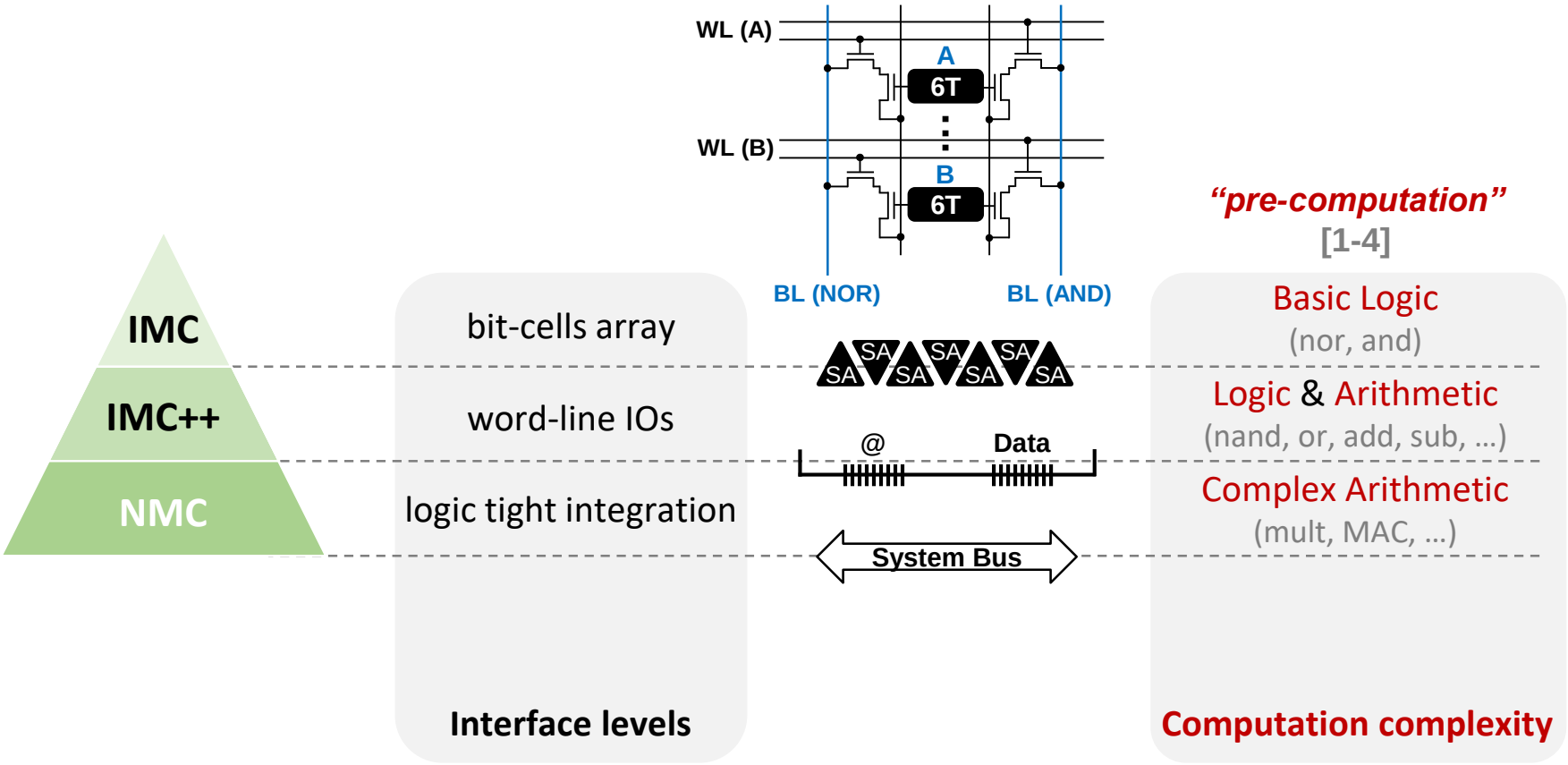
Interface levels

Computation complexity

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

[1] K. C. Akyel, DRC², 2016
[2] S. Aga, Compute Caches, 2017
[3] Y. Zhang, Recryptor, 2018
[4] A. Agrawal, X-SRAM, 2018

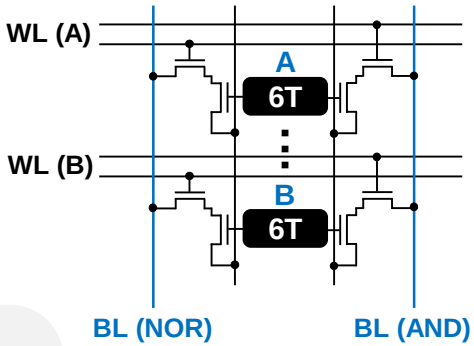
IN-MEMORY OR NEAR-MEMORY COMPUTING?



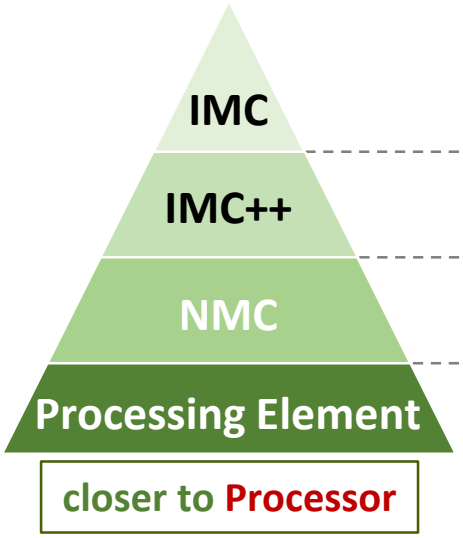
Concept: Bring computation closer to the memory (*SRAM-Based technology*)

[1] K. C. Akyel, DRC², 2016
[2] S. Aga, Compute Caches, 2017
[3] Y. Zhang, Recryptor, 2018
[4] A. Agrawal, X-SRAM, 2018

IN-MEMORY OR NEAR-MEMORY COMPUTING?



closer to Memory

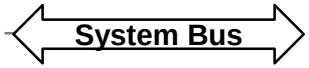


closer to Processor

bit-cells array

word-line IOs

logic tight integration



system integration

Interface levels

"pre-computation"
[1-4]

Basic Logic
(nor, and)

Logic & Arithmetic
(nand, or, add, sub, …)

Complex Arithmetic
(mult, MAC, …)

Everything else
(div, cos, sin, …)

Computation complexity

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

[1] K. C. Akyel, DRC², 2016
[2] S. Aga, Compute Caches, 2017
[3] Y. Zhang, Recryptor, 2018
[4] A. Agrawal, X-SRAM, 2018

STUDY OF DATA-CENTRIC APPLICATIONS

Applications	% of bitwise ops.	Bitwise ops.	Other ops.	Memory dependencies
AES	**66%**	shift, xor, and	add	key + message + SBox
Boolean Matrix Multiplication	**50%**	and, or	add, cmp	3 matrices
DNA pattern searching	**50%**	and, not, or, shift	add, sub	pattern + data base
Hamming distance (popcount)	**55~66%**	and, shift, xor	add, (sub, cmp)	pattern + text

Tool used: M. Kooli, et al. "Software Platform Dedicated for In-Memory Computing Circuit Evaluation", CEA Leti, RSP 2017

1 – explored kernels mostly exploit bitwise operations (IMC++ topology)

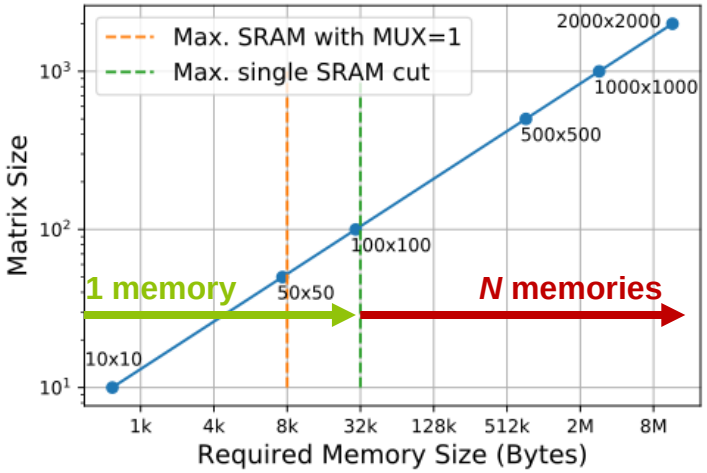
STUDY OF DATA-CENTRIC APPLICATIONS

Applications	% of bitwise ops.	Bitwise ops.	Other ops.	Memory dependencies
AES	**66%**	shift, xor, and	add	key + message + SBox
Boolean Matrix Multiplication	**50%**	and, or	add, cmp	3 matrices
DNA pattern searching	**50%**	and, not, or, shift	add, sub	pattern + data base
Hamming distance (popcount)	**55~66%**	and, shift, xor	add, (sub, cmp)	pattern + text

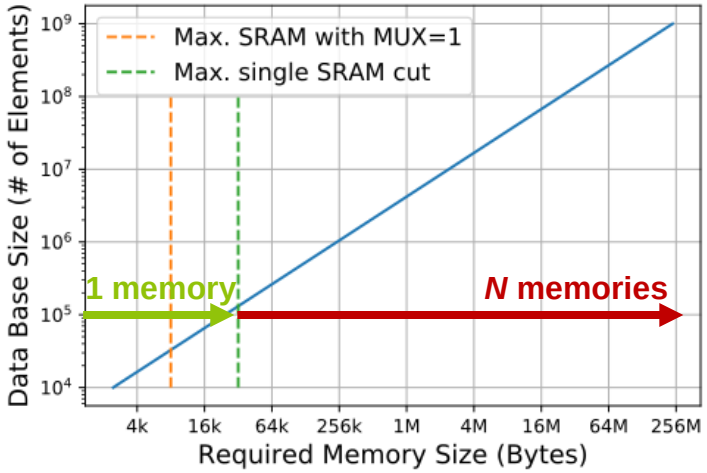
Tool used: M. Kooli, et al. "Software Platform Dedicated for In-Memory Computing Circuit Evaluation", CEA Leti, RSP 2017

1 – explored kernels mostly exploit **bitwise operations** (IMC++ topology)

Boolean Matrix Multiplication

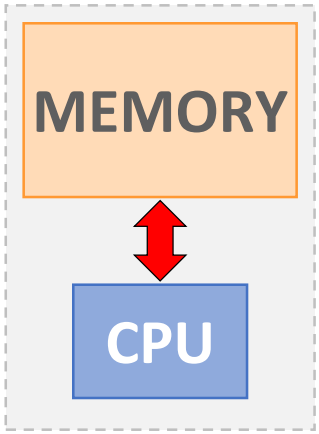


DNA Pattern Matching

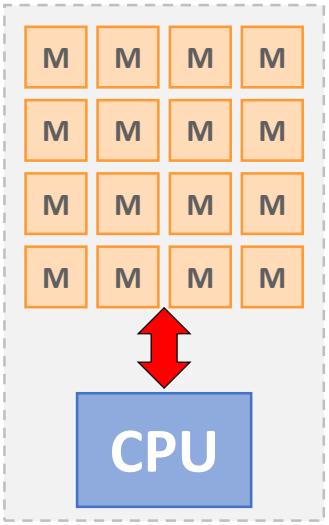


2 – single memory is not enough, we need more !

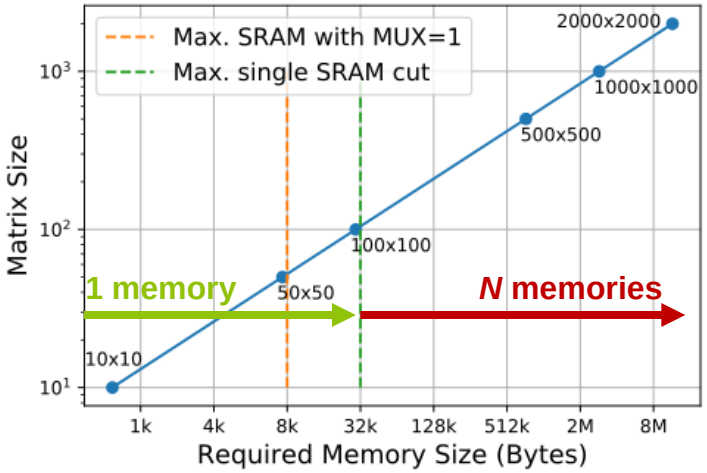
STUDY OF DATA-CENTRIC APPLICATIONS



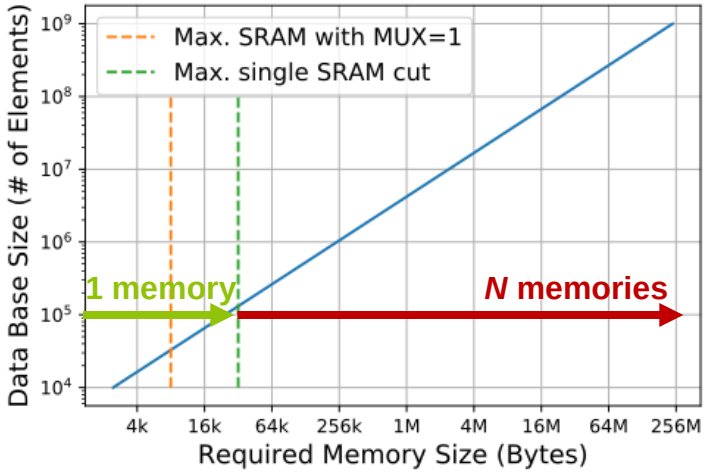
AT WHAT COST
IS IT SCALABLE?



Boolean Matrix Multiplication



DNA Pattern Matching



2 – single memory is not enough, we need more !

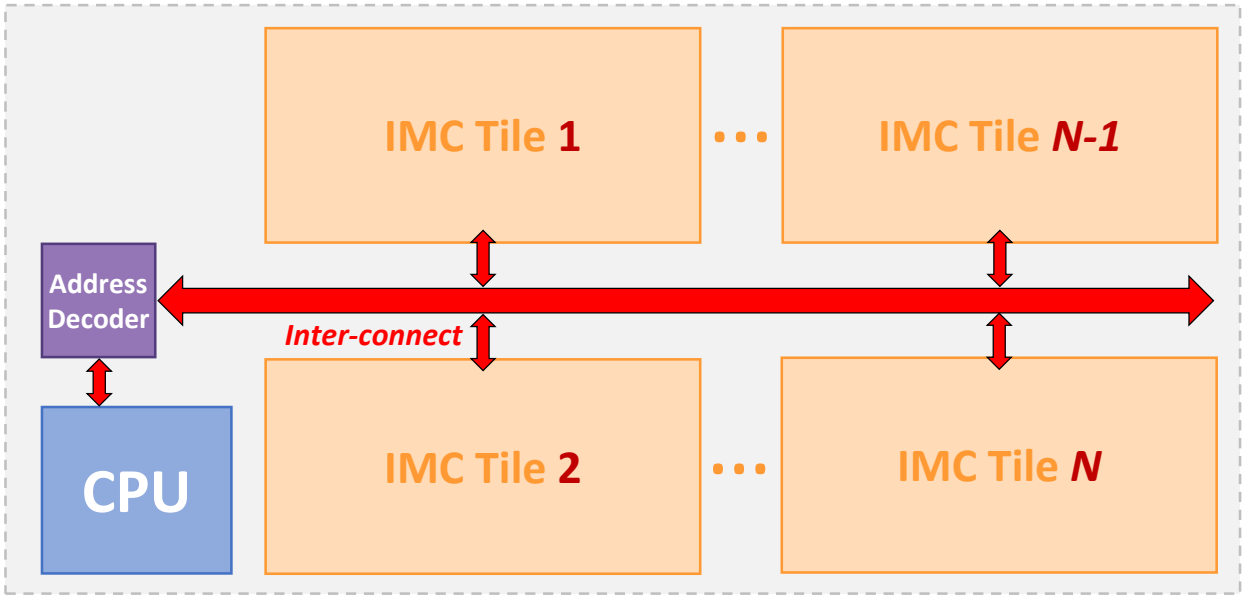
OUTLINE

1. Motivations & Applications
2. A 2-D Vector Scalable In-Memory Computing SRAM-Tile-Based Architecture
3. A methodology to evaluate the inter-connect cost
4. A fast and scalable wire cost model for architectural exploration
5. Conclusions & Perspectives

OUTLINE

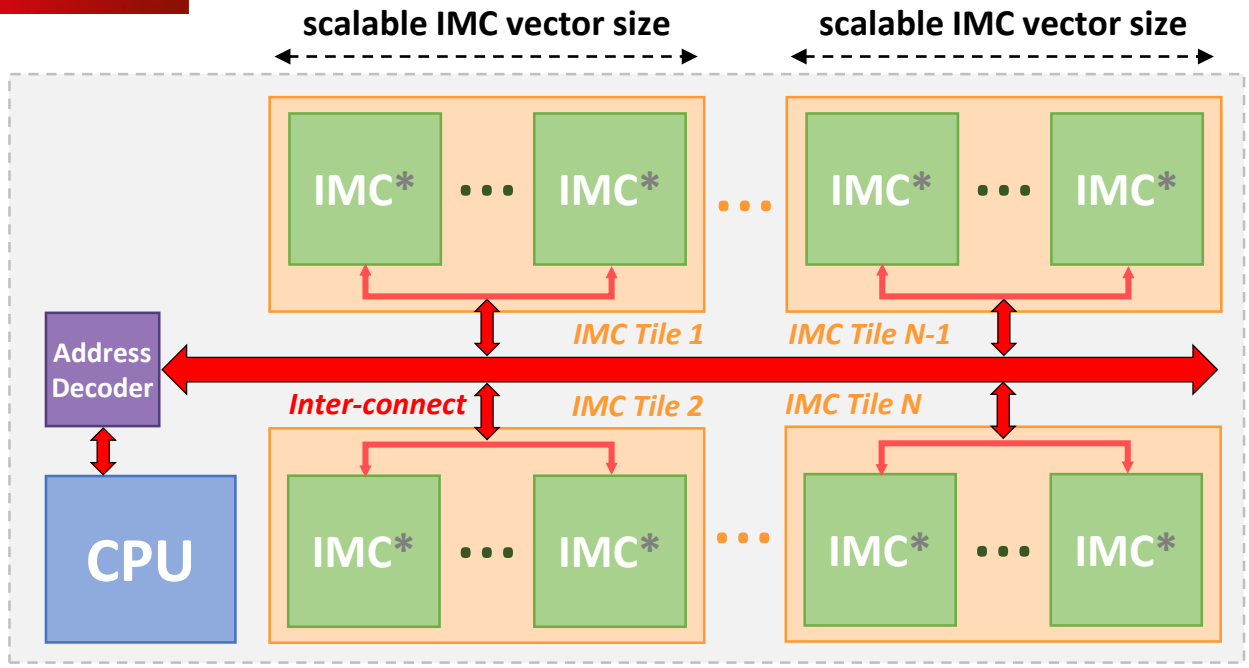
1. Motivations & Applications
2. A 2-D Vector Scalable In-Memory Computing SRAM-Tile-Based Architecture
3. A methodology to evaluate the inter-connect cost
4. A fast and scalable wire cost model for architectural exploration
5. Conclusions & Perspectives

A 2-D VECTOR SCALABLE IMC SRAM-TILE-BASED ARCHITECTURE



- A Central Processing Unit
- An Address Decoder for memory accessing
- A conventional Inter-connect
- *N* In-Memory Computing Tiles (IMC Tile)

A 2-D VECTOR SCALABLE IMC SRAM-TILE-BASED ARCHITECTURE

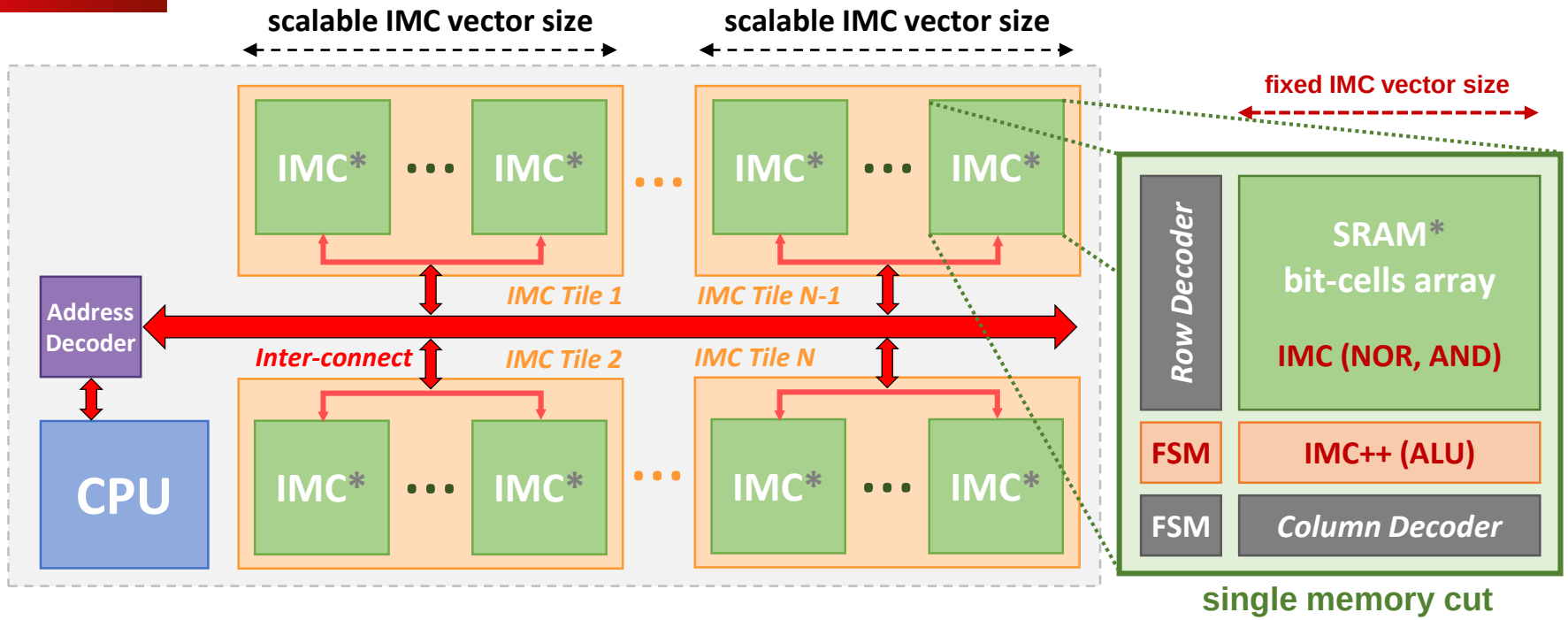


*IMC++ SRAM-based [1]

- A Central Processing Unit
- An Address Decoder for memory accessing
- A conventional Inter-connect
- *N* In-Memory Computing Tiles (IMC Tile)
  - IMC++ SRAM-based cut

[1] K. C. Akyel, DRC², 2016

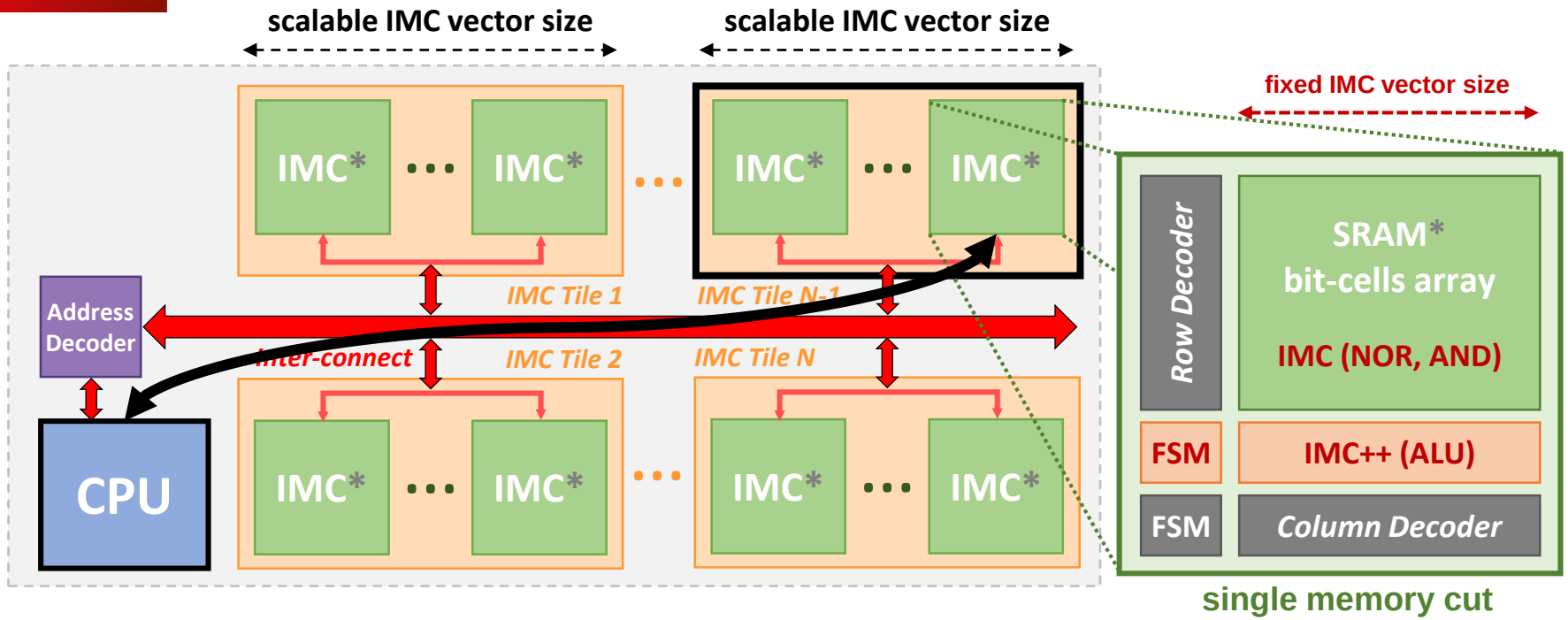
A 2-D VECTOR SCALABLE IMC SRAM-TILE-BASED ARCHITECTURE



- A Central Processing Unit
- An Address Decoder for memory accessing
- A conventional Inter-connect
- *N* In-Memory Computing Tiles (IMC Tile)
  - IMC++ SRAM-based cut
  - support basic logic and arithmetic operations (IMC++)

[1] K. C. Akyel, DRC², 2016

A 2-D VECTOR SCALABLE IMC SRAM-TILE-BASED ARCHITECTURE



- A **Central Processing Unit**
- An **Address Decoder** for memory accessing
- A conventional **Inter-connect**
- *N* **In-Memory Computing Tiles** (IMC Tile)
  - **IMC++ SRAM-based** cut
  - support basic logic and arithmetic operations (IMC++)

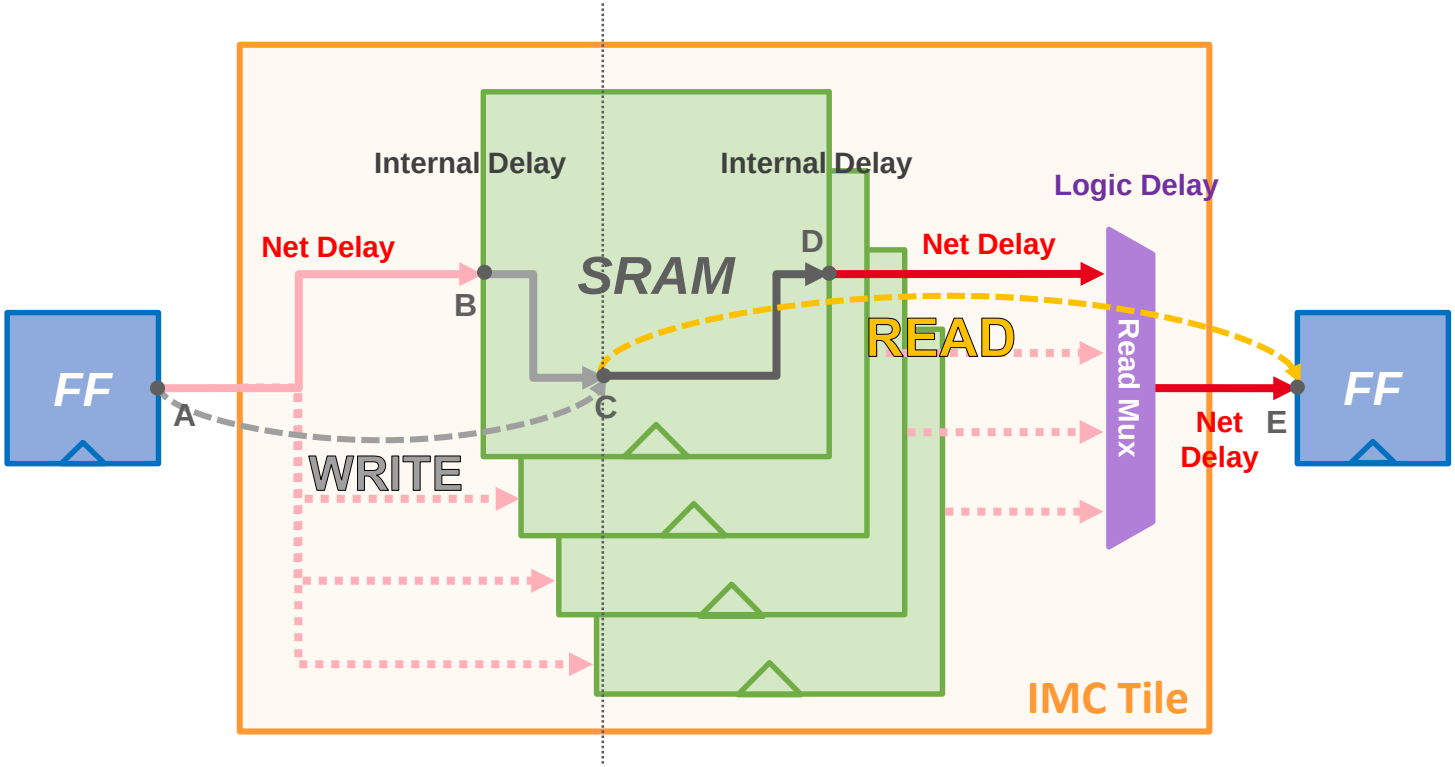
What is the wiring cost ?

[1] K. C. Akyel, DRC², 2016

OUTLINE

1. Motivations & Applications
2. A 2-D Vector Scalable In-Memory Computing SRAM-Tile-Based Architecture
3. **A methodology to evaluate the inter-connect cost**
4. A fast and scalable wire cost model for architectural exploration
5. Conclusions & Perspectives

MEASURING THE WORST TIMING PATH

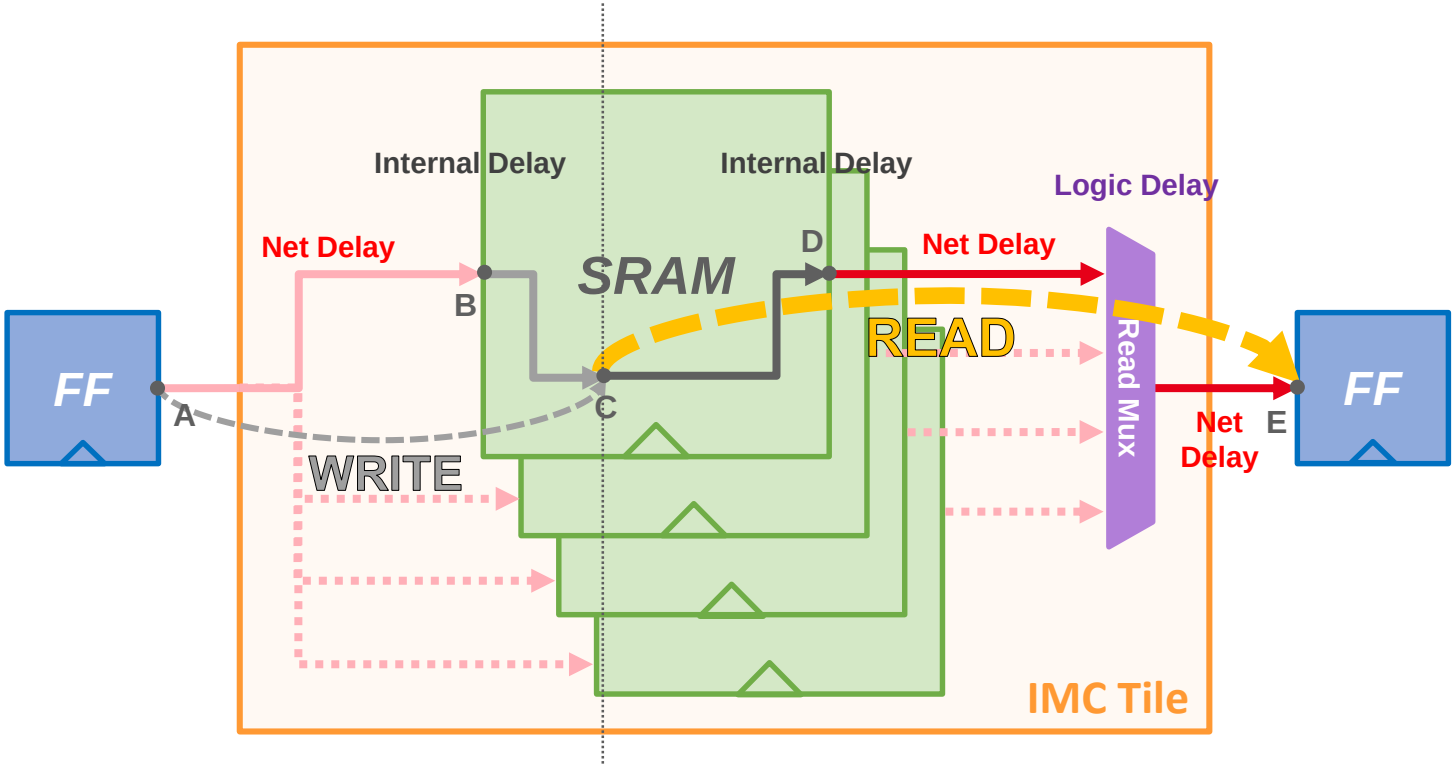


Total Write Access Time: $T_{AC} = T_{AB} + T_{BC}$ = write net delay + write memory access

Total Read Access Time: $T_{CE} = T_{CD} + T_{DE}$ = read memory access + read net delay + read logic delay

The proposed wiring exploration consider SRAM cut ⇔ IMC

MEASURING THE WORST TIMING PATH



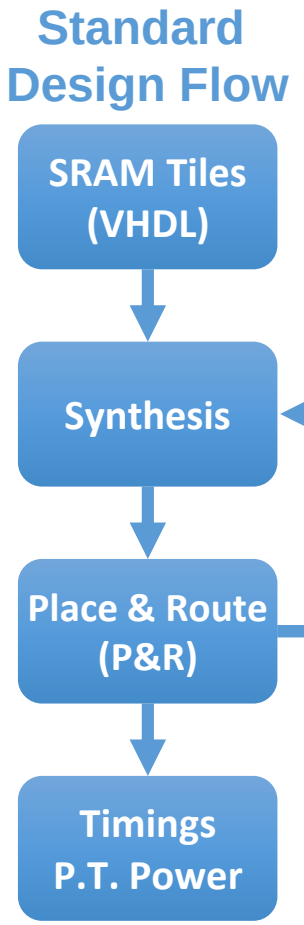
Total Write Access Time: $T_{AC} = T_{AB} + T_{BC}$ = write net delay + write memory access

Total Read Access Time: $T_{CE} = T_{CD} + T_{DE}$ = read memory access + read net delay + read logic delay

The Worst Timing Path

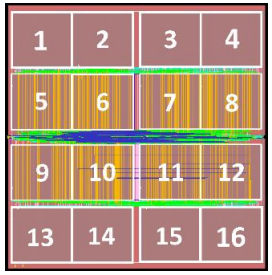
The proposed wiring exploration consider SRAM cut ⇔ IMC

STANDARD METHODOLOGY



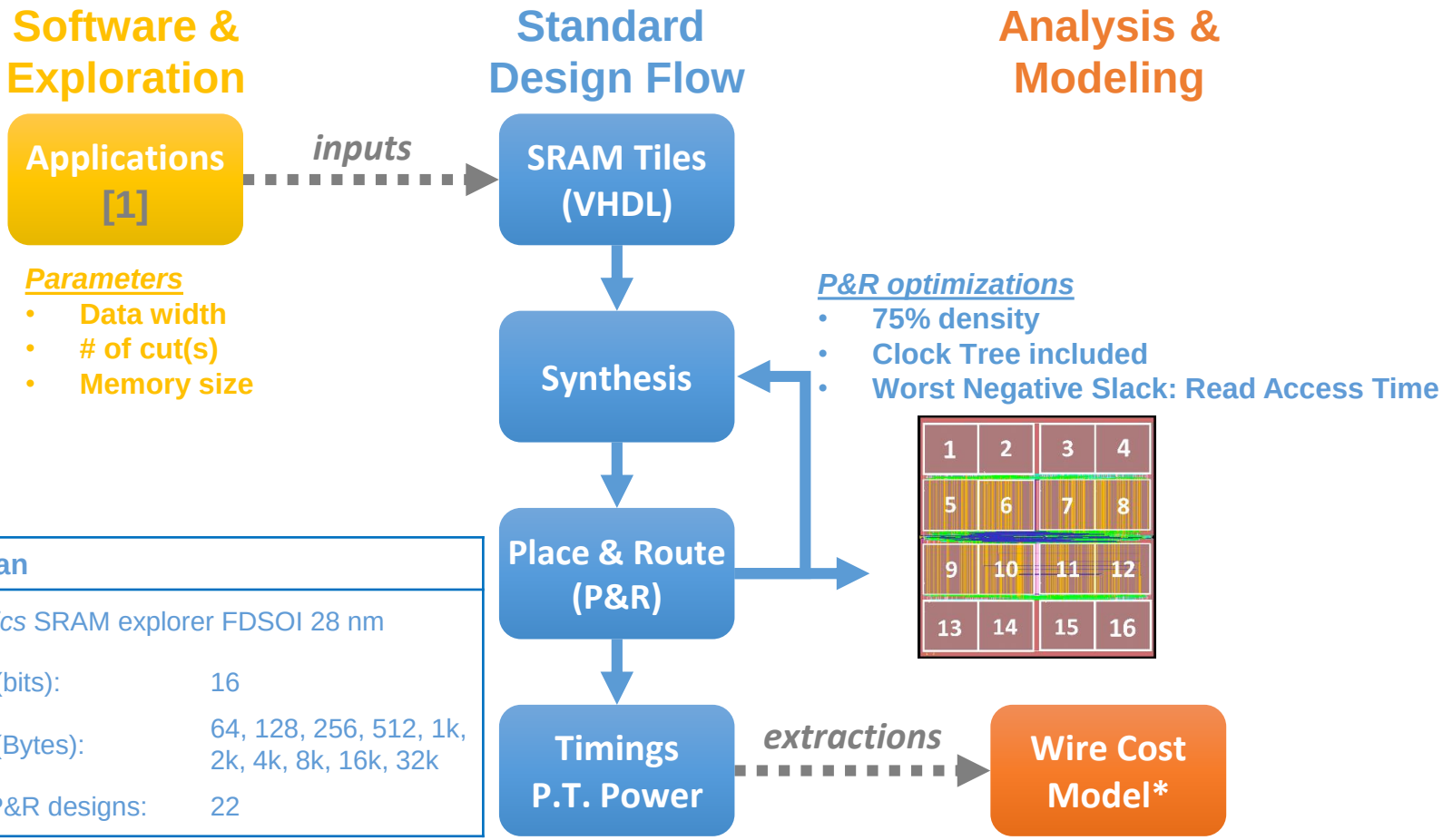
P&R optimizations

- 75% density
- Clock Tree included
- Worst Negative Slack: Read Access Time



e.g. floorplan of 16 SRAM cuts

METHODOLOGY USED IN THE STUDY



Experimental Plan	
- *ST Microelectronics* SRAM explorer FDSOI 28 nm	
- Fixed data width (bits):	16
- Memory cut size (Bytes):	64, 128, 256, 512, 1k, 2k, 4k, 8k, 16k, 32k
- Total number of P&R designs:	22

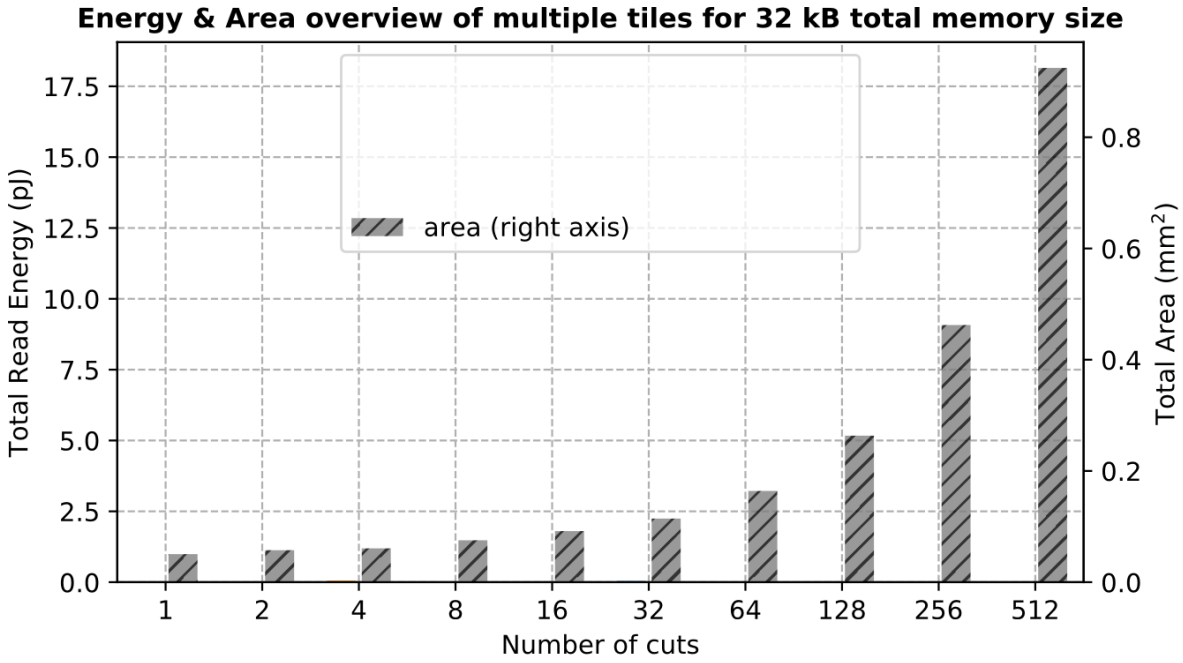
[1] M. Kooli, "Soft. Platf. for IMC Eval.", 2017

OUTLINE

1. Motivations & Applications
2. A 2-D Vector Scalable In-Memory Computing SRAM-Tile-Based Architecture
3. A methodology to evaluate the inter-connect cost
4. **A fast and scalable wire cost model for architectural exploration**
5. Conclusions & Perspectives

EXPERIMENTAL RESULTS

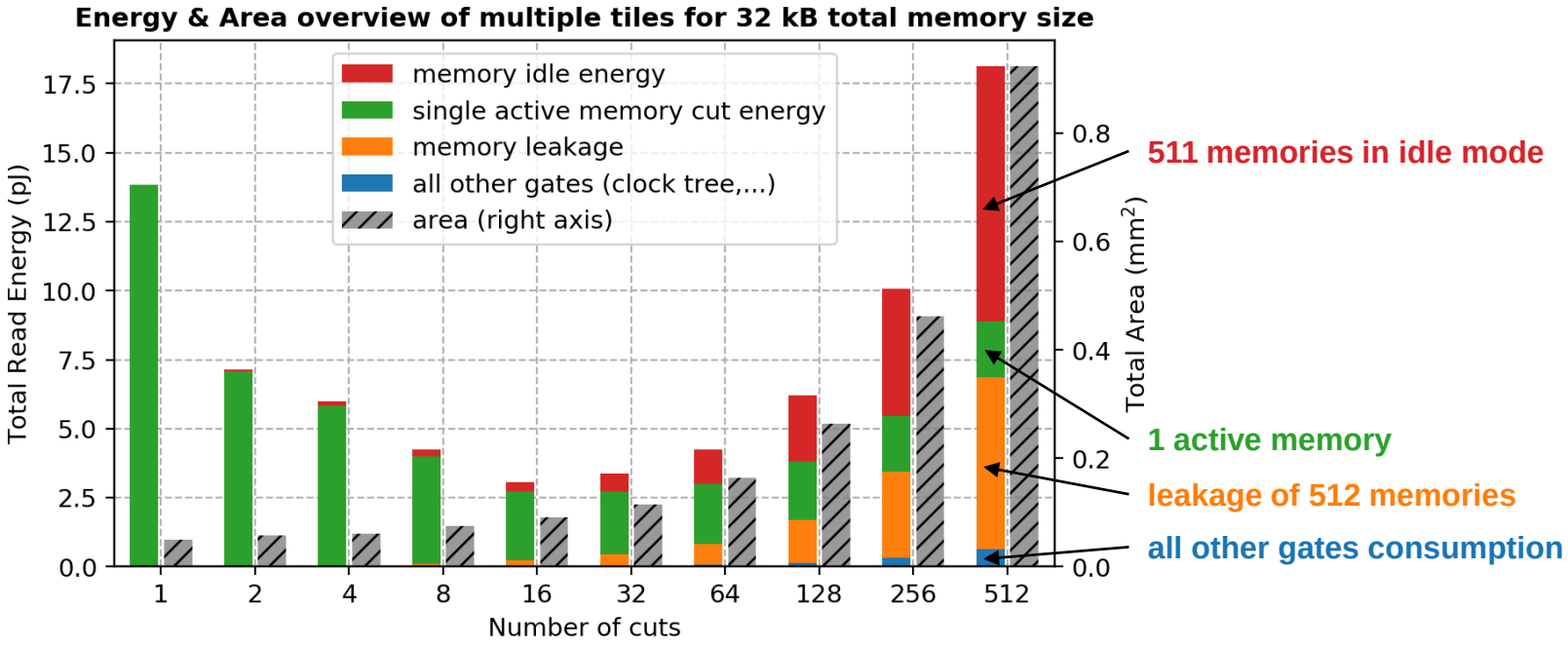
ENERGY & AREA



- For the **same total memory size** (32 kB) [1 core + variable # of cuts]
  - Area is steadily increasing

EXPERIMENTAL RESULTS

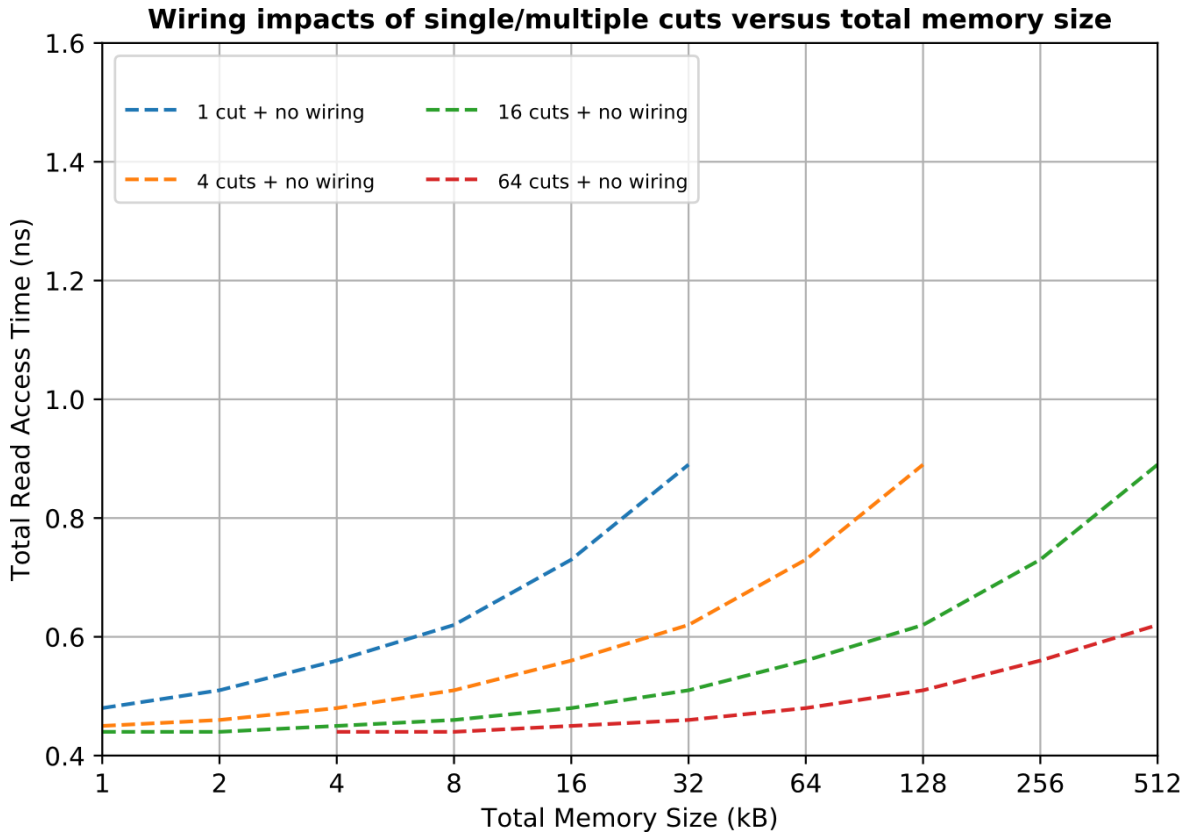
ENERGY & AREA



- For the **same total memory size** (32 kB) [1 core + variable # of cuts]
  - Area is steadily increasing
  - For 512 cuts design ➔ **>50% idle energy**
- **Optimal** trade-off can change with regard to the **memory size**

EXPERIMENTAL RESULTS

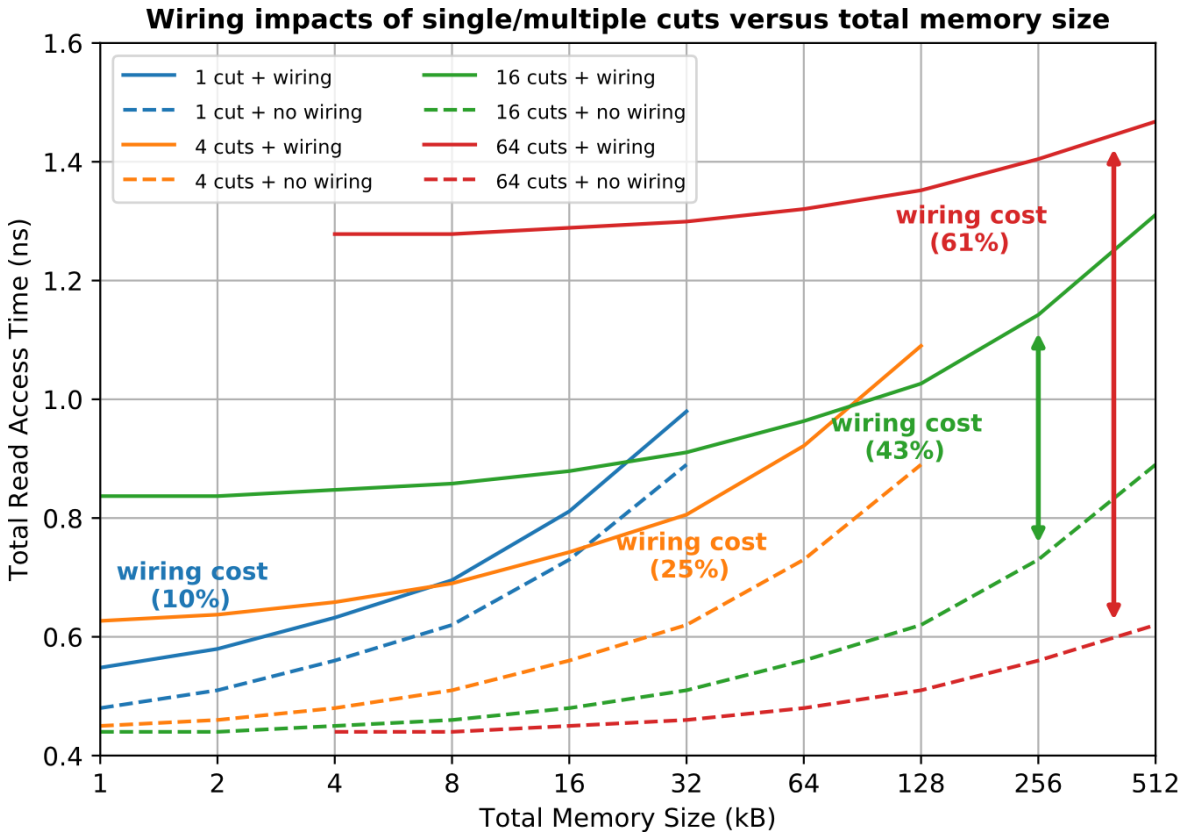
TIMING (NO WIRING)



- No wiring cost, only the **read access time of a single memory tile**

EXPERIMENTAL RESULTS

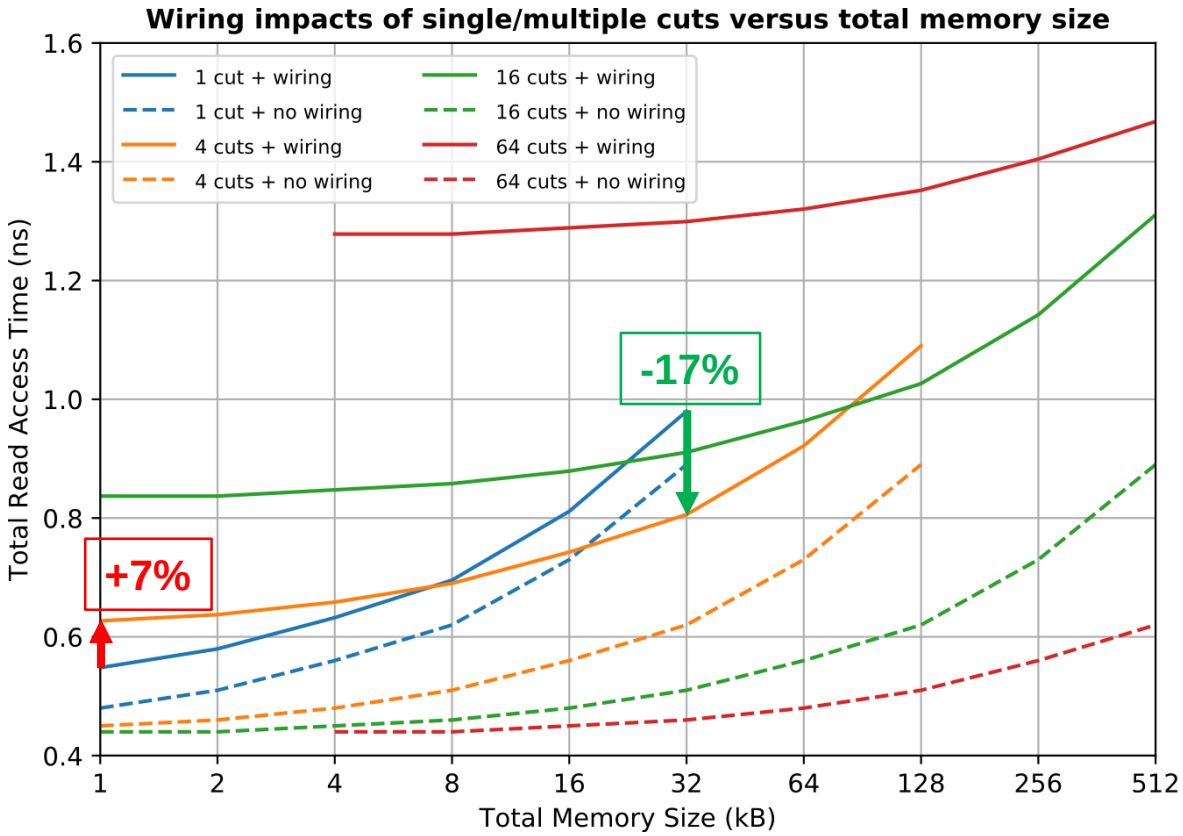
TIMING (WITH WIRING)



- **Wiring cost impact**
  - **64-cut (no wiring) ➔ 64-cut (+wiring) : +61% in timing**

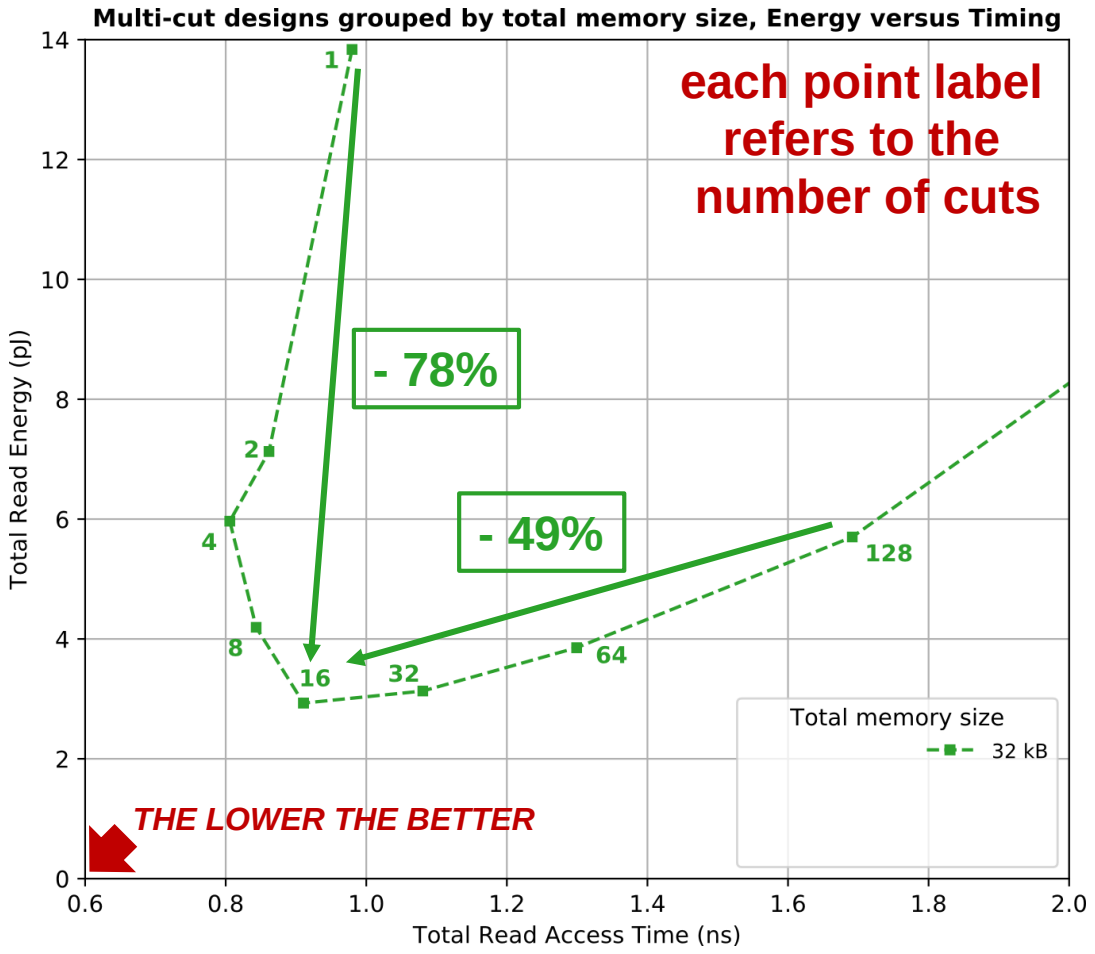
EXPERIMENTAL RESULTS

TIMING (WITH WIRING)



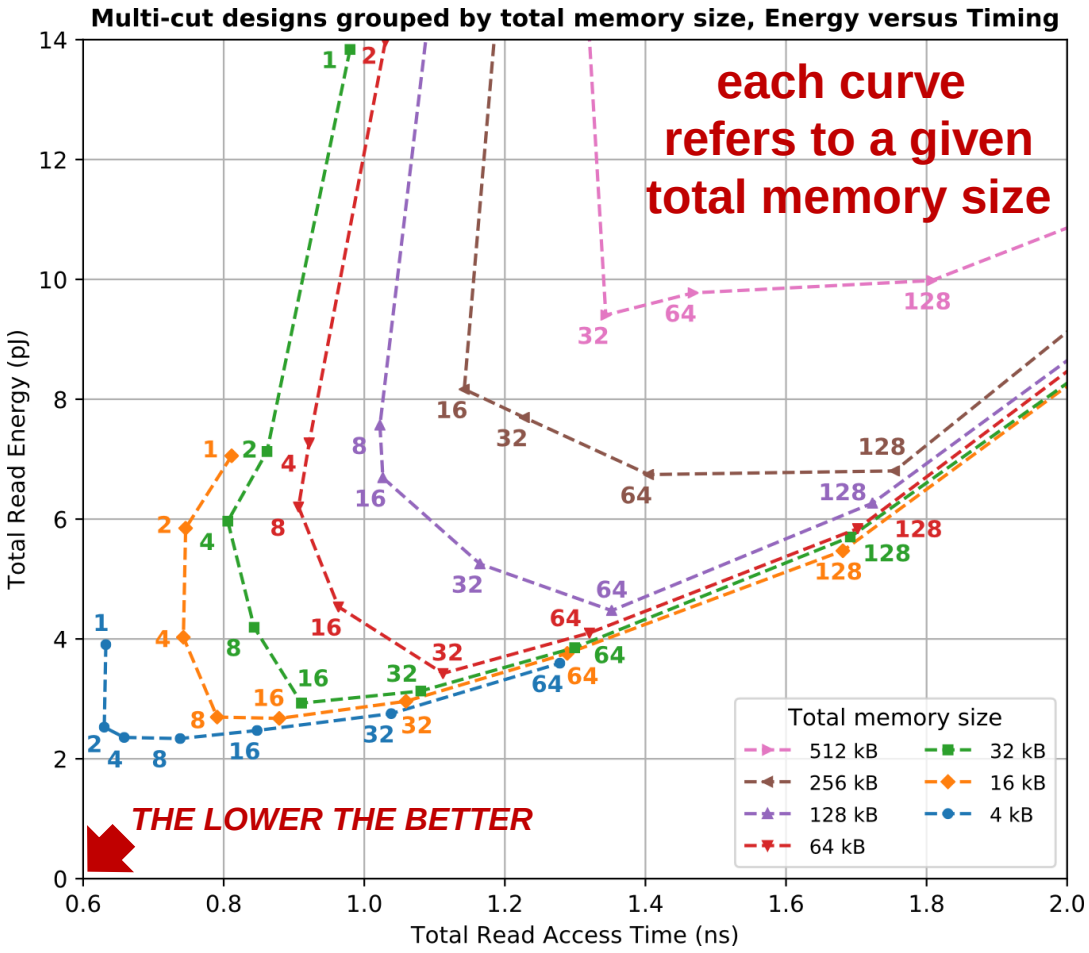
- **Wiring cost impact**
  - **64-cut (no wiring) ➔ 64-cut (+wiring) : +61% in timing**
  - **@ 1kB: 1-cut ➔ 4-cut: +7% in timing**
  - **@ 32kB: 1-cut ➔ 4-cut: -17% in timing**

WIRE MODEL EXPLORATION BASED ON EXPERIMENTAL RESULTS



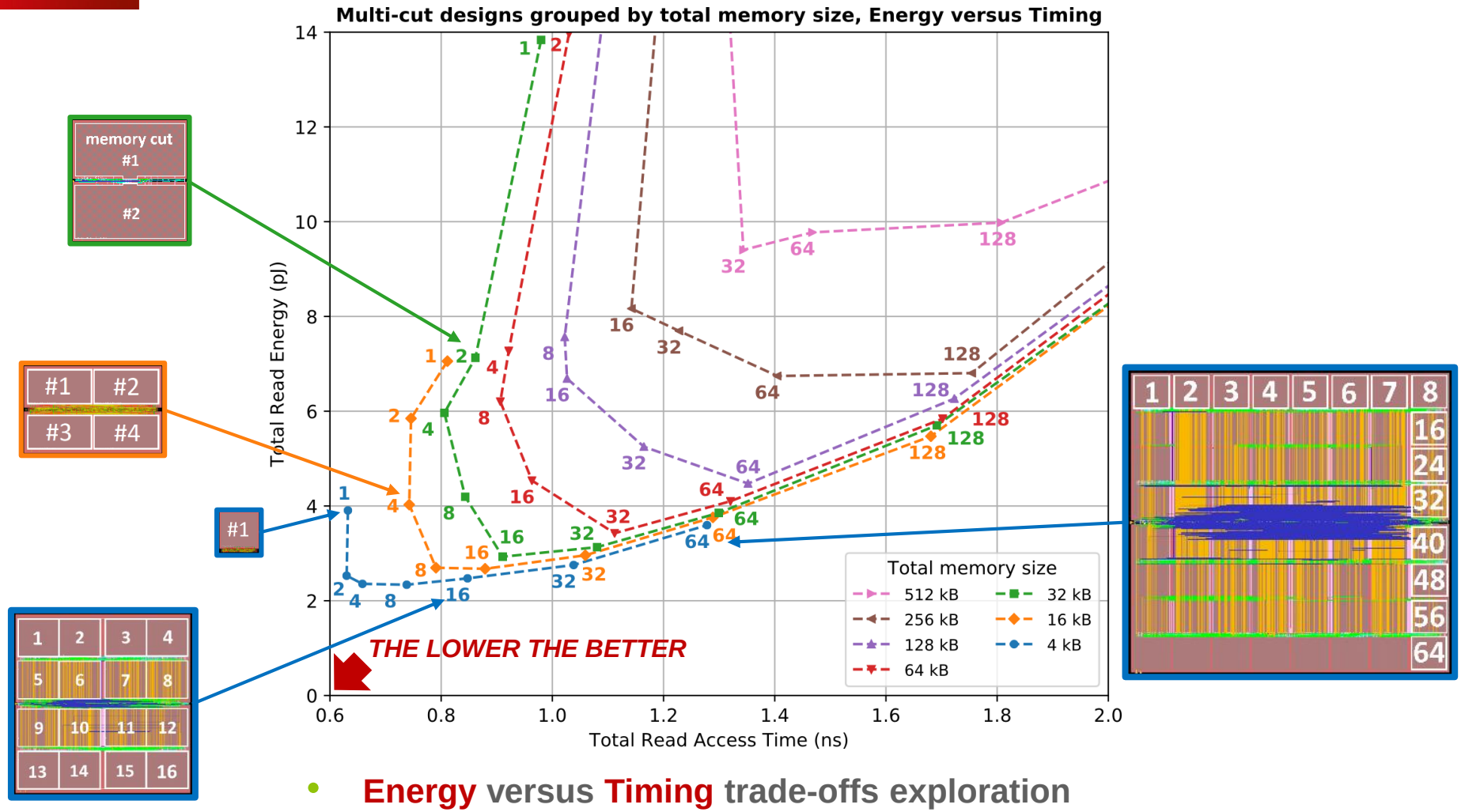
- Example: for a **32 kB** total memory size
  - **1-cut** ➔ **16-cut** memory: **78%** better in energy saving
  - **128-cut** ➔ **16-cut** memory: **49%** better in read access time

WIRE MODEL EXPLORATION BASED ON EXPERIMENTAL RESULTS



- **Energy** versus **Timing** trade-offs exploration

WIRE MODEL EXPLORATION BASED ON EXPERIMENTAL RESULTS



- **Energy** versus **Timing** trade-offs exploration
- Include **real P&R designs** to calibrate the model

OUTLINE

1. Motivations & Applications
2. A 2-D Vector Scalable In-Memory Computing SRAM-Tile-Based Architecture
3. A methodology to evaluate the inter-connect cost
4. A fast and scalable wire cost model for architectural exploration
5. **Conclusions & Perspectives**

CONCLUSIONS & PERSPECTIVES

- **We proposed**
  - A 2-D Vector Scalable IMC SRAM-Tile-Based Architecture
  - A new **model of the wiring cost** calibrated on P&R extractions
- **Conclusions**
  - **Inter-connect** should be estimated to evaluate **emerging IMC** architectures
  - By splitting the memory into multiple cuts, we achieve:
    - **-78%** in energy consumption and **-49%** in read access time
- **Future works**
  - Wiring cost must be accurately evaluated for **3D** emerging technologies
  - These models allow to **quantify IMC gains** in a larger system

Thank you for your attention

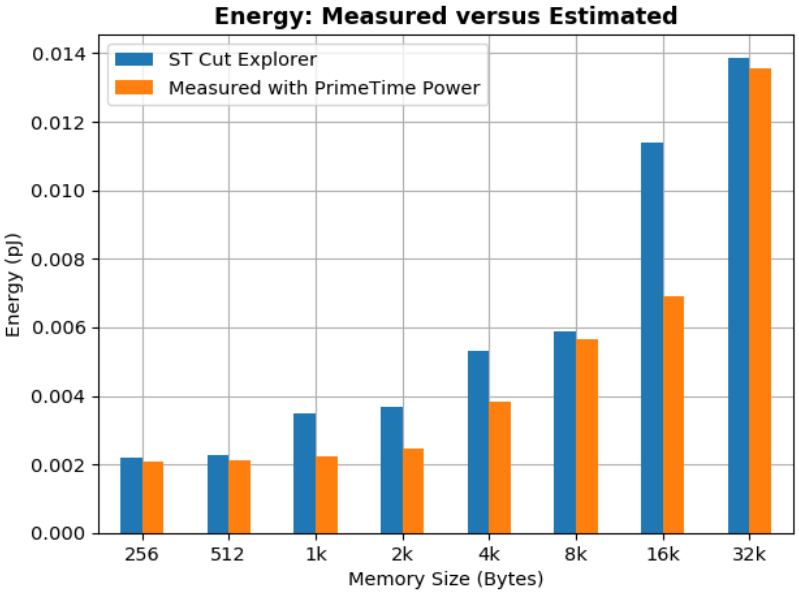
Questions ?

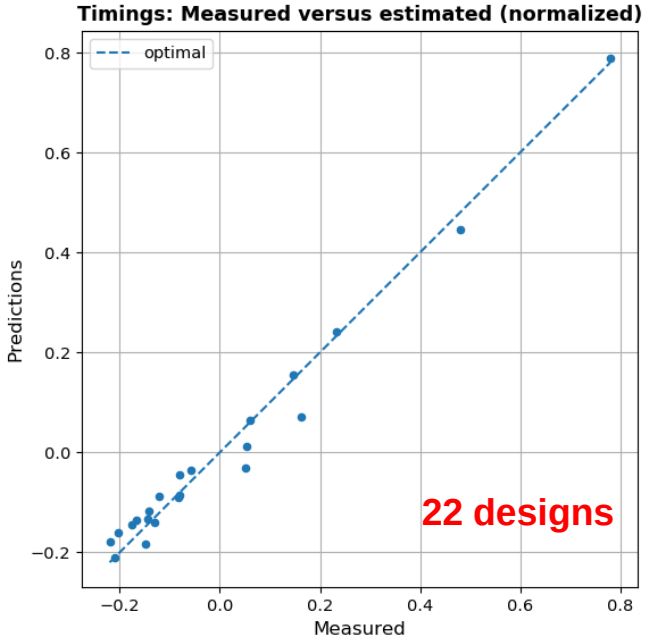
Speaker: Roman GAUCHI (roman.gauchi@cea.fr)

Leti, technology research institute
Commissariat à l'énergie atomique et aux énergies alternatives
Minatec Campus | 17 rue des Martyrs | 38054 Grenoble Cedex | France
www.leti.fr

Backup Slides

MODEL CALIBRATION





Energy model (Errors)

- **Maximum error: 39.44 %**
- **Average error: 17.35 %**
- **Median error: 19.53 %**

Timing model (Errors)

- **Maximum error: 11.22 %**
- **Average error: 4.30 %**
- **Median error: 3.68 %**

Conditions of experience:

- **ST Microelectronics SRAM explorer FDSOI 28 nm, SPHD (Single Port High Density)**
- **Data width fixed: 16 bits**
- **Memory size used: {64, 128, 256, 512, 1k, 2k, 4k, 8k, 16k, 32k} Bytes**
- **Total number of P&R designs: 22 designs**