Memristor based computation-in-memory Architectures for Edge AI
Opportunities and challenges

Said Hamdioui
Head of Quantum and Computer Engineering department
Delft University of Technology & Cognitive-IC
The Netherlands
S.Hamdioui@tudelft.nl

Next Talk: 17/May/2021, 4-5:30pm CET
Outline

• The opportunity and the challenges
  • IoT-edge partnership, HW challenges

• Computer architecture
  • Past and future & classification

• Computation-in-Memory CIM
  • Basics and classification

• CIM circuit design
  • Logic operations, Vector-matric multiplication

• CIM Potential
  • Design flow, application domains, potential improvements

• Challenges
  • The open questions

• Conclusion
The opportunity: IoT-edge partnership

Billions of IoT connected devices installed worldwide over the years 2015 to 2025.

Many requirements:
- Intelligence
- Energy constraints
- Local computing
- Data privacy
- Real-time decisions
- 24/7
The challenges: Intelligence

AlphaGo

AI surpassed human

Not suitable for Edge Application

AI requires huge resources
**The challenges:** The HW architecture and technology

### Architecture

<table>
<thead>
<tr>
<th>Operation</th>
<th>Energy (pJ) @TSMC 45nm</th>
<th>Cost * ALU</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALU: Add</td>
<td>0.1</td>
<td>1x</td>
</tr>
<tr>
<td>SRAM: read</td>
<td>5</td>
<td>50x</td>
</tr>
<tr>
<td>DRAM: read</td>
<td>640</td>
<td>6,400x</td>
</tr>
</tbody>
</table>

[Source: M. Horowitz et al., ISSCC, 2014]

1. Memory Wall
2. ILP Wall
3. Power Wall

[Source: D. Patterson, future of computer Architecture, 2006]

### Technology

1. Leakage Wall
2. Cost Wall
3. Reliability Wall

[Source: S. Hamdioui, et.al, DATE 2017]

**Need of new Architectures**

**Need of new Technologies**

**Need of Unconventional architectures using unconventional technologies**
Computer Architectures: Classification

- **COM: Computation-Out-Memory**
  1. Far (COM-F)
  2. Near (COM-N)

- **CIM: Computation-In-Memory**
  3. Periphery (CIM-P)
  4. Array (CIM-A)

- **Hybrid architectures**

- **Status**
  - COM: commercialized & conv technologies
  - CIM: Research, conv & unconv technologies

## Computer Architectures: Comparison

<table>
<thead>
<tr>
<th>Type</th>
<th>Off-chip Data mov.</th>
<th>Computations reqrs.</th>
<th>Available Bandwidth</th>
<th>Memory design effort</th>
<th>Scalability</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIM-A</td>
<td>No*</td>
<td>Yes</td>
<td>H. latency</td>
<td>Max</td>
<td>High</td>
</tr>
<tr>
<td>CIM-P</td>
<td>No*</td>
<td>Yes</td>
<td>H. Cost</td>
<td>High-Max</td>
<td>High</td>
</tr>
<tr>
<td>COM-N</td>
<td>Yes</td>
<td>NR</td>
<td>Low cost</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>COM-F</td>
<td>Yes</td>
<td>NR</td>
<td>Low cost</td>
<td>Low</td>
<td>Low</td>
</tr>
</tbody>
</table>

### Other attributes
- Endurance requirements

### Other metrics classification
- **Memory Technology**: SRAM, DRAM, MRAM, ReRAM, ...
- **Computation parallelism**: Task (e.g., multi-core), Data (e.g., SIMD), Instruction (e.g., VLIW)

---

Computer Architectures: overview

- Other attributes
- Endurance requirements
- Other metrics classification
- Memory Technology: SRAM, DRAM, MRAM, ReRAM
- Computation parallelism: Instruction, data, task

Computation-in-Memory: Basics

- Single or multi functions
  - \( r() \)
  - \( r() \) & \( f() \)?
  - \( p() \) & \( f() \)?
  - \( r() \) & \( p() \) & \( f() \)
  - Etc.

- CIM-P
  - **Major** changes in periphery
    - \( p() \) and/or \( r() \)
  - Basic: no changes in array
  - Hybrid: some changes in array

Computation-in-Memory: Classification

<table>
<thead>
<tr>
<th>CIM-A</th>
<th>CIM-P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output in <strong>Array</strong></td>
<td>Output in <strong>Periphery</strong></td>
</tr>
<tr>
<td>CIM-Ar</td>
<td>CIM-Ah</td>
</tr>
<tr>
<td>Input: All <strong>resistive</strong></td>
<td>Input: Hybrid</td>
</tr>
</tbody>
</table>

- **Example**
  - CIM-Ar: $O_1 = I_1 \text{ OR } I_2$
  - CIM-Ah: $O_2 = I_2 \text{ OR } I_3$

*Source: M. Abu Lebdeh, et. al, “Memristive device based circuits for computation-in-memory architectures”, ISCAS 2019*
## Computation-in-Memory: Classification

<table>
<thead>
<tr>
<th>CIM-A</th>
<th>CIM-P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output in <strong>array</strong></td>
<td>Output in <strong>Periphery</strong></td>
</tr>
<tr>
<td>CIM-Ar</td>
<td>CIM-Pr</td>
</tr>
<tr>
<td>Input: All <strong>resistive</strong></td>
<td>Input: All <strong>resistive</strong></td>
</tr>
<tr>
<td>CIM-Ah</td>
<td>CIM-Ph</td>
</tr>
<tr>
<td>Input: Hybrid</td>
<td>Input: Hybrid</td>
</tr>
</tbody>
</table>

**Example**
- CIM-Pr: \( O_1 = I_1 \text{ OR } I_2 \)
- CIM-Ah: \( O_2 = I_2 \text{ OR } I_3 \)

[Source: M. Abu Lebdeh, et. al, “Memristive device based circuits for computation-in-memory architectures”, ISCAS 2019]
### Computation-in-Memory: Classification

<table>
<thead>
<tr>
<th>CIM-A</th>
<th>CIM-P</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Output in</strong></td>
<td><strong>Output in</strong></td>
</tr>
<tr>
<td><strong>array</strong></td>
<td><strong>Periphery</strong></td>
</tr>
<tr>
<td><strong>CIM-Ar</strong></td>
<td><strong>CIM-Pr</strong></td>
</tr>
<tr>
<td>Input: All <strong>resistive</strong></td>
<td>Input: All <strong>resistive</strong></td>
</tr>
<tr>
<td></td>
<td>Pinatubo [2016]</td>
</tr>
<tr>
<td></td>
<td>OR, AND, XOR</td>
</tr>
<tr>
<td></td>
<td>Scouting [2017]</td>
</tr>
<tr>
<td></td>
<td>OR, AND, XOR</td>
</tr>
<tr>
<td><strong>CIM-Ah</strong></td>
<td><strong>CIM-Ph</strong></td>
</tr>
<tr>
<td>Input: Hybrid</td>
<td>Input: Hybrid</td>
</tr>
<tr>
<td></td>
<td>VMM [2016, 2017]</td>
</tr>
<tr>
<td></td>
<td>BCMM V [2017]</td>
</tr>
<tr>
<td></td>
<td>BDP [2018]</td>
</tr>
</tbody>
</table>

**SNIDER** [2005]  
e.g., NAND  
**IMPLY** [2010]  
e.g., IMP, NAND  
**MAGIC** [2014]  
e.g., NOT, NOR  
**Fast Boolean** [2015]  

**Resi. Accu.** [2003]  
**Majority Logic** [2016]  

**Recent work mainly on CIM-P?**

[Source: M. Abu Lebdeh, et. al, “Memristive device based circuits for computation-in-memory architectures”, ISCAS 2019]
CIM circuit design: CIM-Ar

- **Snider logic**
  - Primitive logic operations: NAND, INV
  - 2 Control voltages: Vw, Vh
    - Vw > Vth > Vh
    - Vw − Vh < Vth

- **Working principal** (e.g., 2NAND)
  1. **Store**
     1. Program p to Ron
     2. Program q to Roff
  2. **Compute**
     1. Initialize f to Roff
     2. Process NAND
        - Vx ≈ Vh; Vw − Vx = Vw − Vh < Vth
        - => f = Roff

**Logic states:**
- In/out: Roff=1; Ron=0

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Suitable for crossbar -> density

Requires multiple accesses

Requires array redesign

[Source: Snider et.al, APA, 2005; Xie et al. ICCD, 2015]
**CIM circuit design: CIM-Ah**

- **Majority logic** [Gaillordon et.al, DATE, 2016]
  - Logic operation: Majority Function
  - Inputs: p, q, z
  - Out: \( z = MAJ(p, \bar{q}, z) \)
  - 2 Control voltages: Vw, GND
    - Vw > Vth

- **Working principal**
  1. Program z to Roff
  2. Process MAJ
     - Apply \( Vp = Vw \) & \( Vq = GND \)
     - \( Vw - 0 = Vw > Vth \)

\[ \Rightarrow z = Ron \]

---

**Logic states:**
- \( \text{In } p \& q: Vw=1; \text{GND}=0 \)
- \( \text{In } z: Ron=1; \text{Roff}=0 \)
- \( \text{Out: } Ron=1; \text{Roff}=0 \)

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>z</th>
<th>( \text{MAJ}(p, \bar{q}, z) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

+ Suitable for crossbar
- Needs data movement \((p,q)\)
- Parallelism?
CIM circuit design: CIM-Pr example

- **Scouting Logic**

Read a memory cell

Parallelism
- -> performance

Operations
- 1 cycle: E efficiency

Reduced com.
- -> E efficiency

Read intensive

Endurance
- Data alignment
- SA & AD medication

OR operation

---

**CIM circuit design: CIM-Pr example**

Read & operate on two cells

**Read & operate on two cells**

- Less requirements on endurance
- Once access per operation (single step)
- No major changes in array
- Modification of SA and AD selection
- Data alignment

**AND operation**

<table>
<thead>
<tr>
<th>Input</th>
<th>00</th>
<th>10/01</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>0</td>
<td>(2V_r/R_H)</td>
<td>(V_r/R_L)</td>
</tr>
</tbody>
</table>

**XOR operation**

<table>
<thead>
<tr>
<th>Input</th>
<th>00</th>
<th>10/01</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>0</td>
<td>(2V_r/R_H)</td>
<td>(V_r/R_L)</td>
</tr>
</tbody>
</table>

Vector dot product: \( \mathbf{a} \cdot \mathbf{b} \)

- \( \mathbf{a} \cdot \mathbf{b} = a_1 b_1 + a_2 b_2 + a_3 b_3 \)
- Inputs: voltage and resistance
- Output: voltage
- Result: \( \text{OR} \)
- Complexity: \( O(1) \)

Vector matrix multiplication

- \( \mathbf{v} = \mathbf{a} \cdot \mathbf{B} = [\mathbf{a} \cdot \mathbf{b}_1, \mathbf{a} \cdot \mathbf{b}_2, \ldots, \mathbf{a} \cdot \mathbf{b}_N] \)
- Extendable to matrix x matrix
- Complexity
  - \( O(1) \): vector * matrix
  - \( O(n) \): matrix * matrix

+ Less requirements on endurance
+ Reduced computing complexity
+ No major changes in array
- Modification of Periphery
- Data alignment

CIM circuit design: CIM-P²

- Multiply-Accumulate
  - \( I_i = \sum V_j \cdot G_{ij} \)
  - \( I_1 = V_1 \cdot G_{11} + V_2 \cdot G_{21} + \ldots + V_m \cdot G_{m1} \)
  - Complexity: \( O(1) \)

---

[A. Shafiee et al., "ISAAC: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars," in ISCA, 2016]
CIM circuit design: ADC design / classification

| Sensing Stage | Input | Analog data $I_{\text{column}}$
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacitor</td>
<td>Intermediate analog data Voltage, Current, Time</td>
<td></td>
</tr>
<tr>
<td>Resistor</td>
<td>Intermediate discrete data Pulses, Interim (1010..)</td>
<td></td>
</tr>
<tr>
<td>……</td>
<td>Output</td>
<td>E.g. Digital bit-stream 010....</td>
</tr>
</tbody>
</table>

- Six classes: (Voltage, Current, Time) x (Pulse, Interim)

Major challenge: appropriate accuracy at low energy & high speed

[Source: A. Signh, et. al, "SRIF: Scalable and reliable integrate and fire circuit ADC for memristor-based CIM architectures", TCAS-1, 2021]
[Source: A. Signh, et. al, "Low Power Memristor-based Computing for Edge-AI Applications", ISCAS 2021]
CIM circuit design: CIM-P

Non-Transposed $B$ Matrix

\[
\begin{bmatrix}
a_1 & a_2 & a_3 \\
b_1 & b_4 & b_7 \\
b_2 & b_5 & b_8 \\
b_3 & b_6 & b_9 \\
\end{bmatrix}
= 
\begin{bmatrix}
c_1 \\
c_2 \\
c_3 \\
\end{bmatrix}
\]

Transposed $B$ Matrix

\[
\begin{bmatrix}
a_1 & a_2 & a_3 \\
b_1 & b_4 & b_7 \\
b_2 & b_5 & b_8 \\
b_3 & b_6 & b_9 \\
\end{bmatrix}^T
= 
\begin{bmatrix}
u_1 \\
u_2 \\
u_3 \\
\end{bmatrix}
\]

Can configurable periphery enable different functions? At what cost?

**Computation-in-Memory: CIM-A v CIMP designs**

---

**Shmoo plot for IMPLY (CIM-Ar)**

<table>
<thead>
<tr>
<th>Voltage</th>
<th>10</th>
<th>50</th>
<th>100</th>
<th>200</th>
<th>500</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>1.4</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>1.5</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>1.6</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>1.7</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>1.8</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>1.9</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>2</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>2.1</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>2.2</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>2.3</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>2.4</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>2.5</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>2.6</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>2.7</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>2.8</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>2.9</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>3</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

---

**Shmoo plot for Scouting OR (CIM-Pr)**

<table>
<thead>
<tr>
<th>Voltage</th>
<th>10</th>
<th>50</th>
<th>100</th>
<th>200</th>
<th>500</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>0.3</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>0.4</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>0.5</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>0.6</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>0.7</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>0.8</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>0.9</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>1</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>1.1</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>1.2</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>1.3</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>1.4</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>1.5</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>1.6</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>1.7</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>1.8</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>1.9</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
</tbody>
</table>

---

**CIM-P operates at low $R_{off}/R_{on}$ ratio & low voltages**

**CIM-P is less sensitive to device variations**

(Source: Ongoing work at TUDelft, not published yet)
Computation-in-Memory: Design flow

Application domain

Kernel → Architecture

Circuit Design

Memory Array → Periphery

Technology
- RRAM
- PCRAM
- STT-MRAM
- More?

Structure
- Hierarchy
- Crossbar
- CMOS-Accessed
- More?

Analog/mixed signal
- Drivers
- DAC/ADC
- SA
- More?

Digital
- Controller
- AD
- More?

Requirements
- Data types
- Operations
- Parallelism
- Memory size


Holistic approach needed
**CIM potential: Database query example**

**Data Set**

<table>
<thead>
<tr>
<th>Dist</th>
<th>Size</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>55</td>
<td>2016</td>
</tr>
<tr>
<td>B</td>
<td>23</td>
<td>2014</td>
</tr>
<tr>
<td>C</td>
<td>43</td>
<td>2015</td>
</tr>
<tr>
<td>D</td>
<td>60</td>
<td>2016</td>
</tr>
<tr>
<td>E</td>
<td>25</td>
<td>2000</td>
</tr>
<tr>
<td>F</td>
<td>34</td>
<td>2001</td>
</tr>
<tr>
<td>G</td>
<td>18</td>
<td>2012</td>
</tr>
<tr>
<td>H</td>
<td>30</td>
<td>2011</td>
</tr>
</tbody>
</table>

**QUERY:** find DATA satisfy \{far or large\}

### Bitmap Indexing

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>Far</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Near</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Large</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Medium</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Small</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>New</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Old</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**OR** \[1 0 1 1 0 0 0 0\]

### Circuit Design: Scouting

[Diagram of a circuit with labels and values]

### Requirements:

1. **Kernel:** bitwise OR
2. **Architecture:** CIM-Pr
   - I1 and I2: stored in a database (in memory)
   - O: read out

---

[Source: Hamdioui, et.al, "Applications of computation-in-memory architectures based on memristive devices", DATE 2019]
**CIM potential:** Database query example

*Delay:* Up to 35X improvements!

*Energy:* Up to 140X improvements!

CIM potential: Database query demo

• Manufactured PCM + circuit simulation

4x8 1R crossbar

Calibrated model

41x303 crossbar circuit

65nm periphery
• Sense amplifier
• Cascade logic
• etc.

Ron/Roff: 20K/1M
Vread: 0.2V

Results

• **11-step** query in **36 ns**
• Total energy: **20 pJ** (power: 558 μW)

Throughput: **92.9 GOPS**
Efficiency: **166 TOPS/W**

6fj/ operation

[Source: Iason Giannopoulos et al., “In-Memory Database Query”, Advanced Intelligent Systems, open access, 2020]
Potential of CIM: Enabling edge/IoT applications

Enable deployment of AI at the edge
Challenges

- Profilers & compilers
- Mapping applications on architecture
- EDA tool chains
- Bridging algorithms to the architecture and even to the device characteristics
- Simulators

Tools & Application

- Micro vs macro architectures
- Intra- and inter-communication
- Virtual memory address translation mechanism
- Coherence and synchronization of CIM kernels
- Design exploration

### Challenges

**Circuit design**
- Fast and energy efficient signal conversion circuits (DAC, ADC)
- High precision programming of NVM
- Precise measurement of current
  - Vector x matrix: output as current
- Design for non-idealities
- Managing sneak paths

**Technology**
- Device-to-device variability, R ratio
- Multi-state behavior
- Energy switching
- Threshold behaviors
- Endurance
- 3D Integration & stacking, yield

[A. Shafiee et al., “ISAAC: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars,” in ISCA, 2016]
Challenges

- There are "known unknowns"
- Defects not fully understood
- Fault models & test solutions
- DFX, BIST, BISR
- Memory vs Comp configuration

Test

- Endurance
- Fault tolerance
- Cycle-to-cycle variability
- Degradation
- Intermittent unpredicted switching

[Source: M. Fieback, et. al, "Intermittent Undefined State Fault in RRAMs", ETS 2021, Nominated for BPA]
Testing CIM device

- Test for two configurations
  - **Memory config**: Read, Write
  - **Computation config**: Read, Write, Compute

- Use typical structural approach

[Source: M Fieback, et.al, “Testing Scouting Logic based Computation-In-Memory Architectures”, ETS 2020]
Testing CIM device: Device-Aware-Test

- **State of the art**
  - Defects modeled as linear resistors
  - Inappropriate for many defects
  - Cannot guarantee 0DPM

- **CIM uses new device technologies**
  - New failure mechanisms/ modes

- **Need of accurate defect modeling**
  - Incorporate defective behavior in device model
  - Crucial and critical for high quality test solutions

R-model fails to detect all defects

Device-Aware Test

[Source: L. Wu, et.al, ITC’20, ITC’21; S. Hamdioui, et.al, ITC’19; M Fieback, et.al, ITC’19, ETS’20, ETS’21]
Conclusion

- **Emerging applications requires new chips**
  - IoT-Edge partnership: Data-centric computing & Huge data storage
  - Rethink: micro-architectures, design, Reliability, security, communication,
  - Efficiency: Order of fJ/operation

- **Key enablers: new architectures and technologies**
  - Arch: CIM, artificial neural network, bio-inspired NN, AP, ...
  - Tech: NV memories, 3D processing/stacking, photonics, ...

- **Huge potential**
  - Order of magn. improvements in computational efficiency
  - Huge storage, reduced communication
  - Cheap implementation of basic operations

- **BUT still many challenges**
  - Technology, circuit, architecture, prog models, Testing, Reliability
  - Full application evaluation rather than small isolated kernels
Acknowledgement

http://www.mnemosene.eu/

Horizon 2020
European Union Funding
for Research & Innovation

MNEMOSENE

ETH Zürich

RWTH

RHEINISCH-WESTFÄLISCHE
TECHNISCHE HOCHSCHULE
AACHEN

umec

IBM

TU/e

Technische Universität
Eindhoven
University of Technology

Where innovation starts

Thanks