Panel: Is In-Memory computing a niche area or the main hardware platform for AI/ML?

Chair: Luca Benini

ETH Zürich, Università di Bologna

IMC is Hot – Will it live up to expectations?

1. Two paradigms: Large scale computing systems interspersed with memory (e.g., Cerebras) and large memory arrays with embedded processing and logic (e.g. Axcelera.ai). **Which one is the possible winner in various application domains?**

2. Advanced APU have been forced to reduce SRAM cache to accommodate new functionality at 4nm (3nm too). This is due **to the limited SRAM scaling**: taking 90 nanometer as a reference, 3 nanometers SRAM bitcell is 10X bigger than it should be according to Moore's Law (the hard rule of contacted gate…). **How, in your opinion, processor-centric computing can overcome this issue?** Please, bear in mind endurance, cost, and integrability constraints.

3. What is the role of **technology in IMC?** Will the architectural paradigm be a real motivator to **quit the CMOS-only design style? Which technologies be relevant as add-ons to CMOS and what are the limitations?**

4. With AI/ML dominating the application domains today, **what will the best approach to support and accelerate computation**?

5. **Digital vs. Analog/Mixed signal in-memory computing - who will win? Why?**

Our Panelists

- Daniele Ielmini, Politecnico di Milano
- Pierre-Emmanuel Gaillardon, University of Utah
- Shahar Kvatinsky, Technion
- Abu Sebastian, IBM Research, Zurich
- Amrita Mathuria, Kepler Computing
- Patrick Groeneweld, Cerebras

Academia

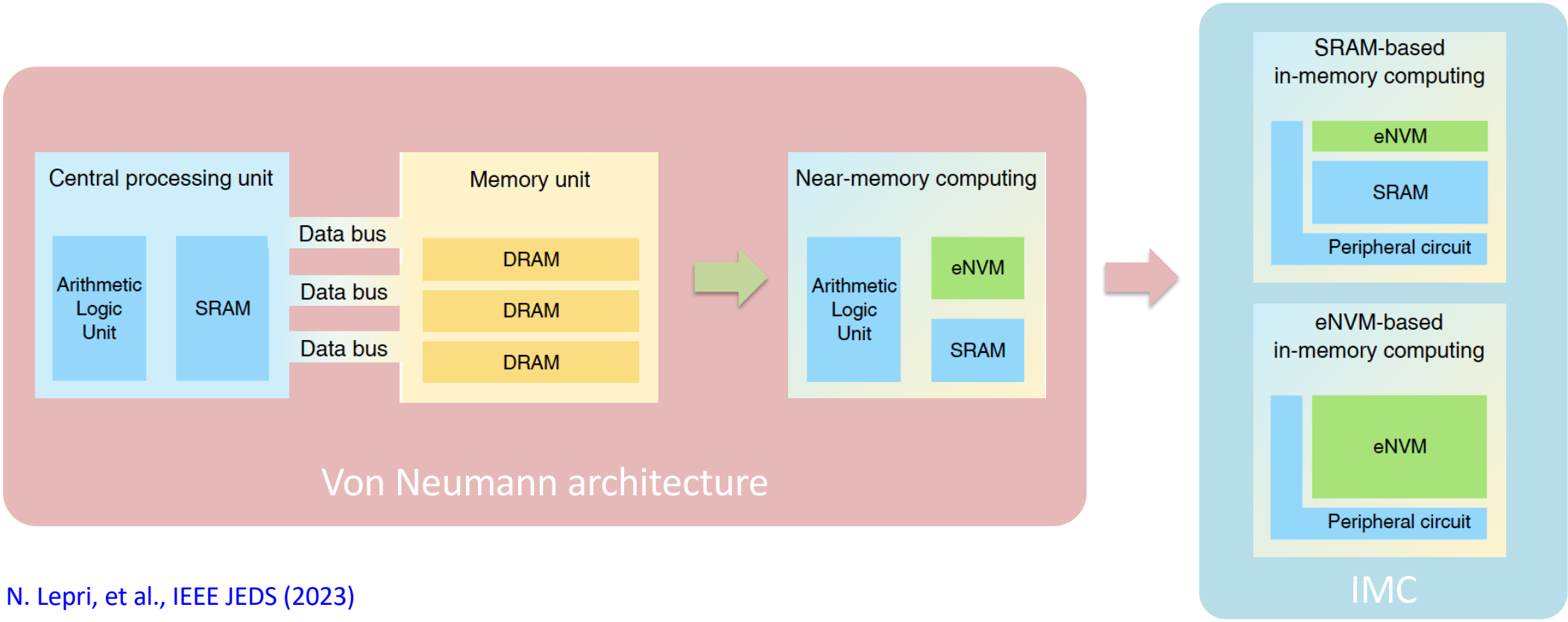
Industry

Is In-Memory computing a niche area or the main hardware platform for AI/ML?

Daniele Ielmini

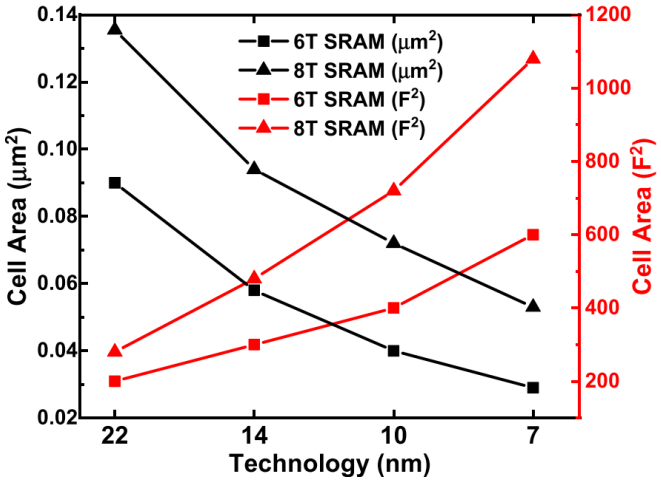
Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano
daniele.ielmini@polimi.it

In-memory computing (IMC)

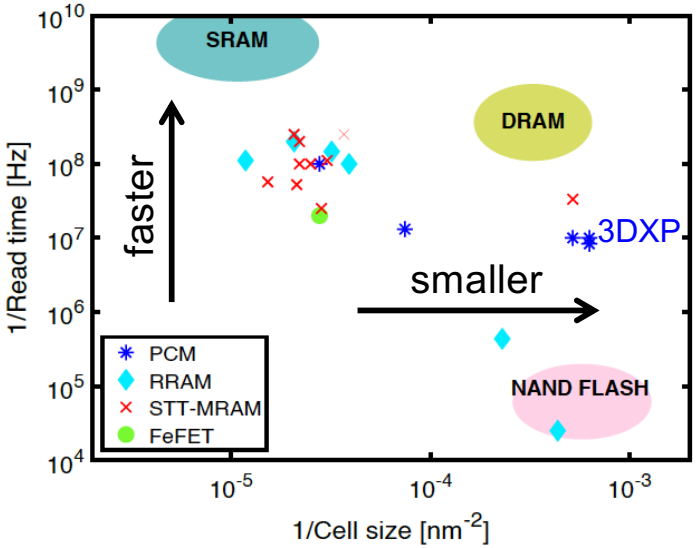


N. Lepri, et al., IEEE JEDS (2023)

Reduced SRAM scaling



A. Lu, et al., Front. Artif. Intell. 4:659060 (2021)



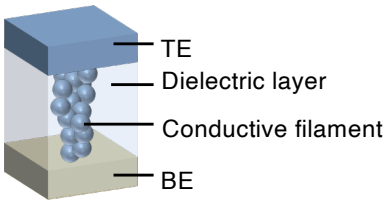
N. Lepri, et al., IEEE JEDS (2023)

- SRAM cannot be replaced by any other technologies
- Embedded NVM is non-volatile, but needs more scaling to support IMC:
  - Going beyond 1T1R
  - Multilevel cell operation

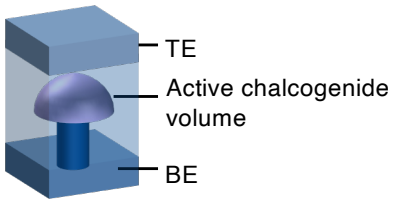
Role of technology in IMC

2-terminal devices

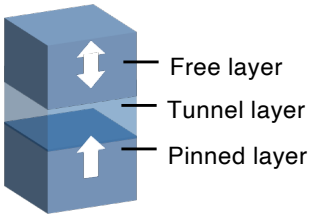
RRAM



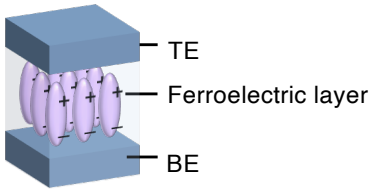
PCM



STT-MRAM

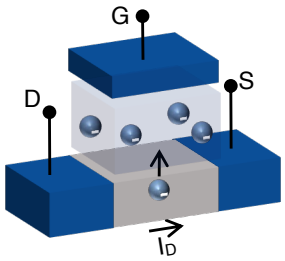


FERAM

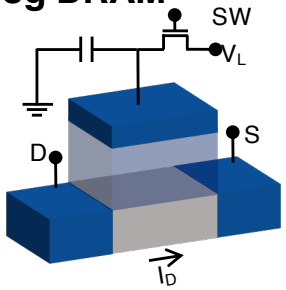


3-terminal devices

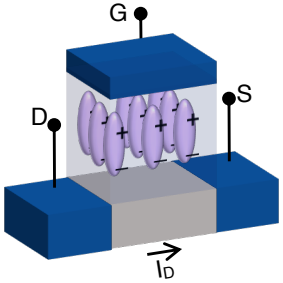
Flash



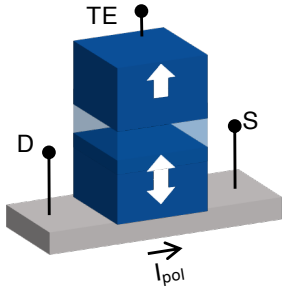
Analog DRAM



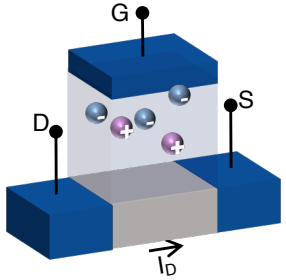
FEFET



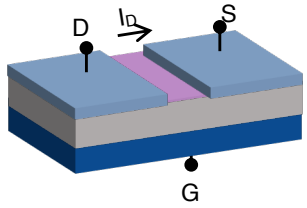
SOT-MRAM



ECRAM



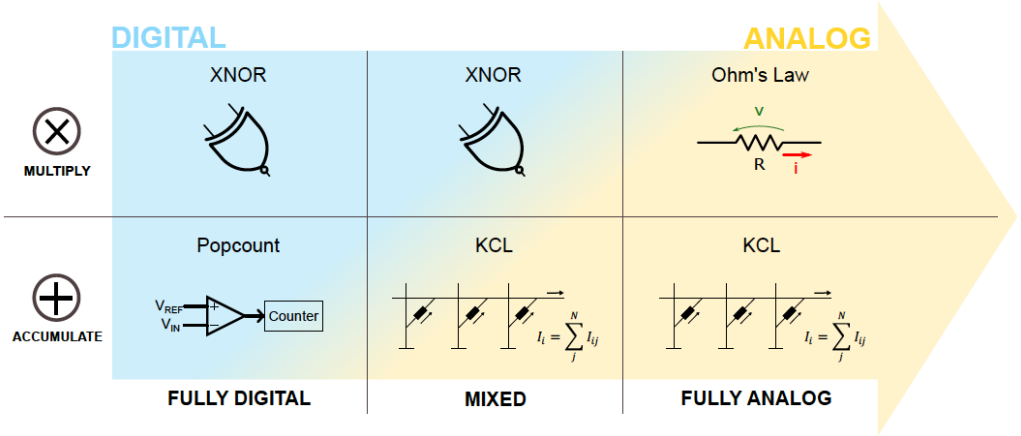
Memtransistor



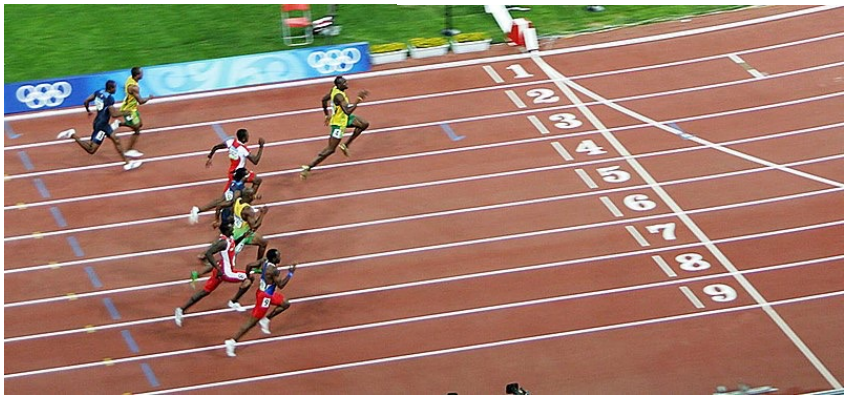
- Embedded NVM is pivotal in this revolution. The quest for best eNVM is open …

Digital vs. Analog/Mixed signal IMC: who will win? Why?

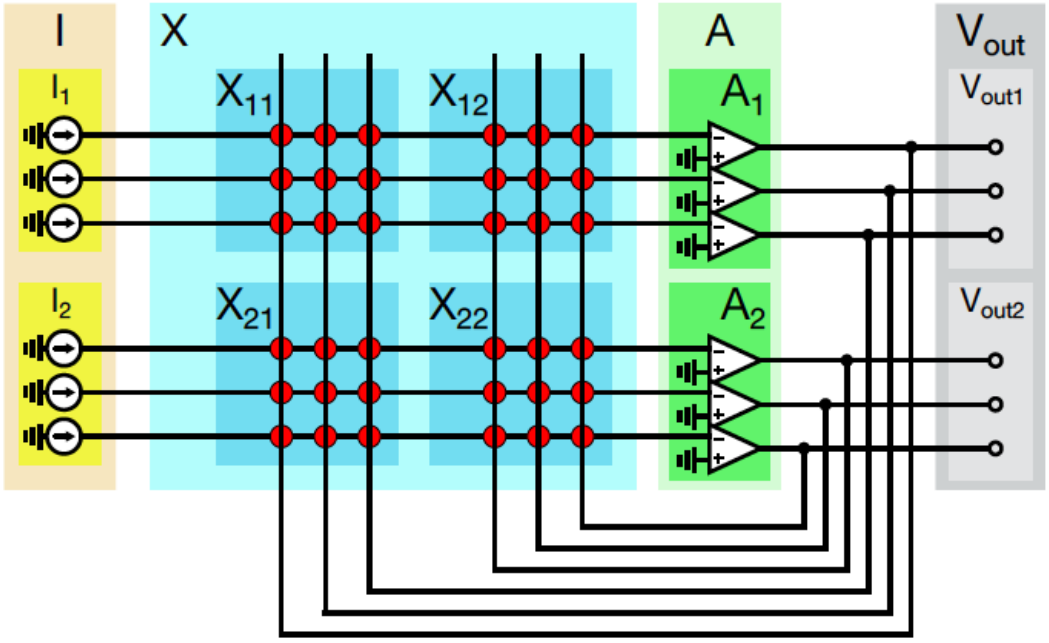
Digital IMC:
- leverage on CMOS technology (e.g., SRAM, logic circuits)
- Immune from noise, variations



Analog IMC:
- Nonvolatile weights
- Higher density
- Smaller complexity, e.g. MVM in O(1)



Best approach to accelerate AI/ML: O(1) complexity by closed-loop IMC



P. Mannocci, et al., IEEE JXCDC (2023)

- O(1) solution of not only MVM but also:
  - Matrix inversion
  - Pseudoinverse
  - Regularized regression
  - Singular value decomposition
  - …
- Full reconfigurability
- Applications range:
  - MIMO for 5G communications
  - Neural network training (echo state machine …)
  - PCA
  - …

Panel: Is In-Memory computing a niche area or the main hardware platform for AI/ML?

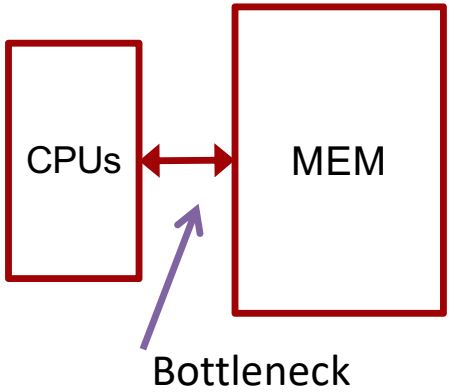
Pierre-Emmanuel Gaillardon

How to break the Memory Bottleneck?

High-performance computing / AI / etc. are very demanding in data manipulation directly hitting the Memory Bottleneck?

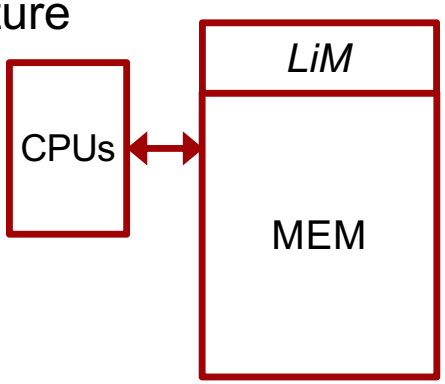
Emerging memories are highly promising devices:

- Non-volatile storage (1-bit or multi-bit)
- Energy improvements – Higher performance
- High density of integration – Low fabrication cost



No matter how good the memories are, the architecture is at the root of the problem

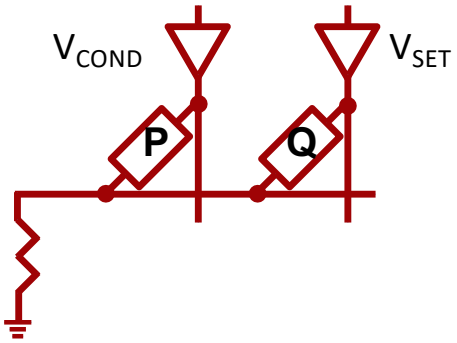
Possible Solution: Offload some processing tasks to the memory



Is Computing in the Memory the solution?

There exists a major body of work performing computation in memory elements (SRAM or emerging)

Ex: Logic implication



p	q	q_n
0	0	1
0	1	1
1	0	0
1	1	1

Theoretically interesting
but practical aspects are commonly overlooked

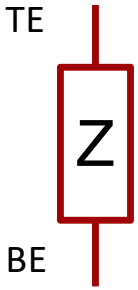
Open questions:

How to deploy the complex signals involved with the operation?
How to implement a dual-memory select with common ground?
How to schedule operations?

Etc.

Disruptive computing models do need holistic consideration of the design abstractions: **Device**, Basic computing engine, **Periphery**, **Tools**, etc

(1) The Quest towards Proper Logic Model



$V_{TE,BE} > V_{th}$

→ SET Operations

→ Z = 1

TE	BE	Z	Z_n
0	0	0	0
0	1	0	0
1	0	0	1
1	1	0	0

$Z_n = TE \cdot BE'$

$V_{TE,BE} < -V_{th}$

→ RESET Operations

→ Z = 0

TE	BE	Z	Z_n
0	0	1	1
0	1	1	0
1	0	1	1
1	1	1	1

$Z_n = TE + BE'$

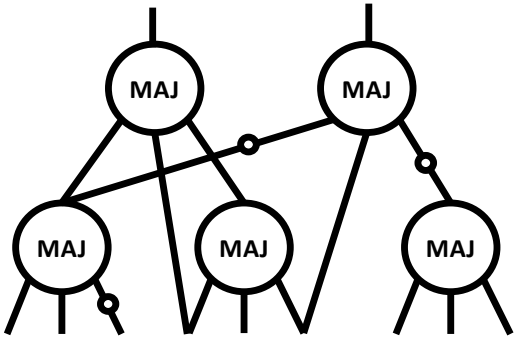
$Z_n =$ $(TE \cdot BE') \cdot Z' + (TE + BE') \cdot Z =$ **MAJ (TE, BE', Z)**

RRAM devices act as MAJ operators!

(In-memory computing with a functionally complete operator RM_3)

… To Unlock Superior Manipulation Techniques

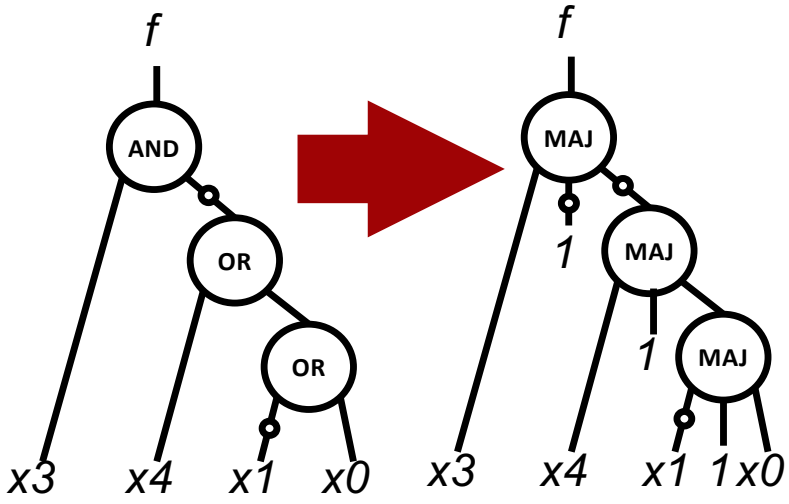
Definition: An MIG is a logic network consisting of 3-input majority nodes and regular/complemented edges.



From AOIG to MIG by direct transposition

Theorem: MIGs include AOIGs
MIGs include AIGs

MIGs are at least as compact as AOIGs
Exploiting the MAJ functionality unlock better representations



(2) The Bigger Picture

- Are we essentially moving the problem around?
  - Indeed, in-memory computing still needs control
  - Instead of moving data, we are moving scheduling
- Isn't in-memory computing just a form a parallelism?
- How much does it costs us to align the data to truly benefit from in-memory computing?
- While many modern applications use more regularl memory access, they are still not truly aligned, in very large amount (not fitting all on-chip) and with duplication still requiring data transfers.

My Take-Away

- In whatever form or shape, our industry must find ways to reduce data transfers costs (in-memory computing, near-memory computing, lower energy links, etc.)

- The problem can only be addressed with a holistic perspective (so we do not reduce the data transfer at the expense of control cost)

Thank you for your attention

Questions?

Laboratory for NanoIntegrated Systems
Department of Electrical and Computer Engineering
MEB building – University of Utah – Salt Lake City – UT – USA

In-Memory computing a niche area or the main hardware platform for AI/ML?

Shahar Kvatinsky

Technion – Israel Institute of Technology

April 2023

ASIC^2

ARCHITECTURES
SYSTEMS
INTELLIGENT COMPUTING
INTEGRATED CIRCUITS

TECHNION
Israel Institute
of Technology

Who Will Win?

- Analog vs. digital
- Data centric vs. compute centric
- Logic and memory technology

PIM Throughput Potential (Digital Example)

- 16-bit OR/ADD/MUL OC = 144/32/1600
- 32-bit OR/ADD/MUL OC = 288/64/6000

PIM Throughput: TPpim = (R*P*XB)/(OC*CT)

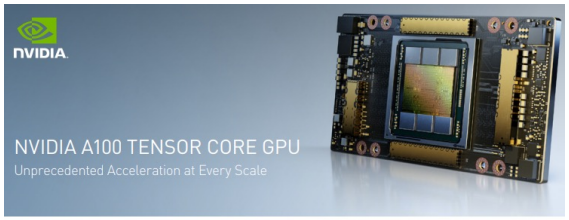
PIM throughput as function of #crossbars and operation type (Example)
R= 1024, P=1, CT = 10ns, Col = 1024, Switch energy = 0.1pJ

		TPpim (TOPS)		
OP Type	OC cyc	XB=1K	XB=32K	XB=1M
16-bit OR	32	3.28	104.9	3,355
32-bit OR	64	1.64	52.4	1,678
16-bit ADD	144	0.73	23.3	746
32-bit ADD	288	0.36	11.7	373
16-bit MUL	1600	0.066	2.1	67
32-bit MUL	6400	0.016	0.5	17
Size (GB)		0.125	4	128
Power (W)		10	336	10,737

#Columns = 1024
Switch energy = 0.1pJ

Is that Good Enough?

Is that Good Enough?



The Most Powerful Compute Platform for Every Workload

The NVIDIA A100 Tensor Core GPU delivers unprecedented acceleration—at every scale—to power the world's highest-performing elastic data centers for AI, data analytics, and high-performance computing (HPC) applications. As the engine of the NVIDIA data center platform, A100 provides up to 20X higher performance over the prior NVIDIA Volta™ generation. A100 can efficiently scale up or be partitioned into seven isolated GPU instances with Multi-Instance GPU (MIG), providing a unified platform that enables elastic data centers to dynamically adjust to shifting workload demands.

NVIDIA A100 Tensor Core technology supports a broad range of math precisions, providing a single accelerator for every workload. The latest generation A100 80GB doubles GPU memory and debuts the world's fastest memory bandwidth at 2 terabytes per second (TB/s), speeding time to solution for the largest models and most massive datasets.

A100 is part of the complete NVIDIA data center solution that incorporates building blocks across hardware, networking, software, libraries, and optimized AI models and applications from the NVIDIA NGC™ catalog. Representing the most powerful end-to-end AI and HPC platform for data centers, it allows researchers to deliver real-world results and deploy solutions into production at scale.

NVIDIA A100 TENSOR CORE GPU SPECIFICATIONS (SXM4 AND PCIE FORM FACTORS)

	A100 40GB PCIe	A100 80GB PCIe	A100 40GB SXM	A100 80GB SXM
FP64	9.7 TFLOPS			
FP64 Tensor Core	19.5 TFLOPS			
FP32	19.5 TFLOPS			
Tensor Float 32 (TF32)	156 TFLOPS | 312 TFLOPS*			
BFLOAT16 Tensor Core	312 TFLOPS | 624 TFLOPS*			
FP16 Tensor Core	312 TFLOPS | 624 TFLOPS*			
INT8 Tensor Core	624 TOPS | 1248 TOPS*			
GPU Memory	40GB HBM2	80GB HBM2e	40GB HBM2	80GB HBM2e
GPU Memory Bandwidth	1,555GB/s	1,935GB/s	1,555GB/s	2,039GB/s
Max Thermal Design Power (TDP)	250W	300W	400W	400W
Multi-Instance GPU	Up to 7 MIGs @ 5GB	Up to 7 MIGs @ 10GB	Up to 7 MIGs @ 5GB	Up to 7 MIGs @ 10GB
Form Factor	PCIe		SXM	
Interconnect	NVIDIA® NVLink® Bridge for 2 GPUs: 600GB/s ** PCIe Gen4: 64GB/s		NVLink: 600GB/s PCIe Gen4: 64GB/s	
Server Options	Partner and NVIDIA-Certified Systems™ with 1-8 GPUs		NVIDIA HGX™ A100-Partner and NVIDIA-Certified Systems with 4,8, or 16 GPUs NVIDIA DGX™ A100 with 8 GPUs	

* With sparsity
** SXM4 GPUs via HGX A100 server boards; PCIe GPUs via NVLink Bridge for up to two GPUs

NVIDIA A100 TENSOR CORE GPU | DATA SHEET | JUN21 | 1

Nvidia A100 GPU[1]

PIM 32K crossbars[2]

MUL FP (TFLOPF)	ADD INT (TOPS)
0.6	**11.7**
3.9	**23.3**
8.4	**46.5**

- Assume same
- Assume same

PIM is not Competitive on Compute Throughput

[1] https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/a100/pdf/nvidia-a100-datasheet-us-nvidia-1758950-r4-web.pdf

[2] *derived from: O. Leitersdorf et al., "AritPIM: High-Throughput In-Memory Arithmetic," IEEE TETC, 2023*

So, what do we Miss?

The memory bandwidth (BW) is the bottleneck!

TPgpu = BW/DIO (data in/out)

OP Type	OC cyc	TPpim (TOPS) XB=32K	Data I/O (B)	Tpgpu (TOPS)	PIM/GPU TOPS Ratio
16-bit OR	32	104.9	6	0.34	309
32-bit OR	64	52.4	12	0.17	309
16-bit ADD	144	23.3	6	0.34	69
32-bit ADD	288	11.7	12	0.17	69
16-bit MUL	1600	2.1	6	0.34	6
32-bit MUL	6400	0.5	12	0.17	3
Power (W)		336		300-400	

PIM vs. GPU throughput on memory intensive kernel (V = V op V)

- GPU is limited when the data is big
- PIM competes when compute/byte is low!
  - **Bulk bitwise – yes, Matrix-Multiply – no**...

NVIDIA A100 TENSOR CORE GPU SPECIFICATIONS (SXM4 AND PCIE FORM FACTORS)

	A100 40GB PCIe	A100 80GB PCIe	A100 40GB SXM	A100 80GB SXM
FP64	9.7 TFLOPS			
FP64 Tensor Core	19.5 TFLOPS			
FP32	19.5 TFLOPS			
Tensor Float 32 (TF32)	156 TFLOPS | 312 TFLOPS*			
BFLOAT16 Tensor Core	312 TFLOPS | 624 TFLOPS*			
FP16 Tensor Core	312 TFLOPS | 624 TFLOPS*			
INT8 Tensor Core	624 TOPS | 1248 TOPS*			
GPU Memory	40GB HBM2	80GB HBM2e	40GB HBM2	80GB HBM2e
GPU Memory Bandwidth	1,555GB/s	1,935GB/s	1,555GB/s	2,039GB/s
Max Thermal Design Power (TDP)	250W	300W	400W	400W

PIM Advantage – Eliminating Data Transfers!

PIM Example: AritPIM[1]

- AritPIM – PIM efficient implementation of the 4 basic arithmetic operations (add/sub/mul/div) for fixed/float data types

Operation	PIM/GPU TOPS/Watt ratio[3]	
	16 Bit	32 Bit
Fixed Add	80.2	78.1
Fixed Subtract	72.5	70.1
Fixed Multiply	9.5	4.9
Fixed Divide	3.6	1.7
Floating Add Unsigned	18.7	19.0
Floating Add Signed	11.2	11.0
Floating Multiply	11.6	3.9
Floating Divide	8.1	3.0

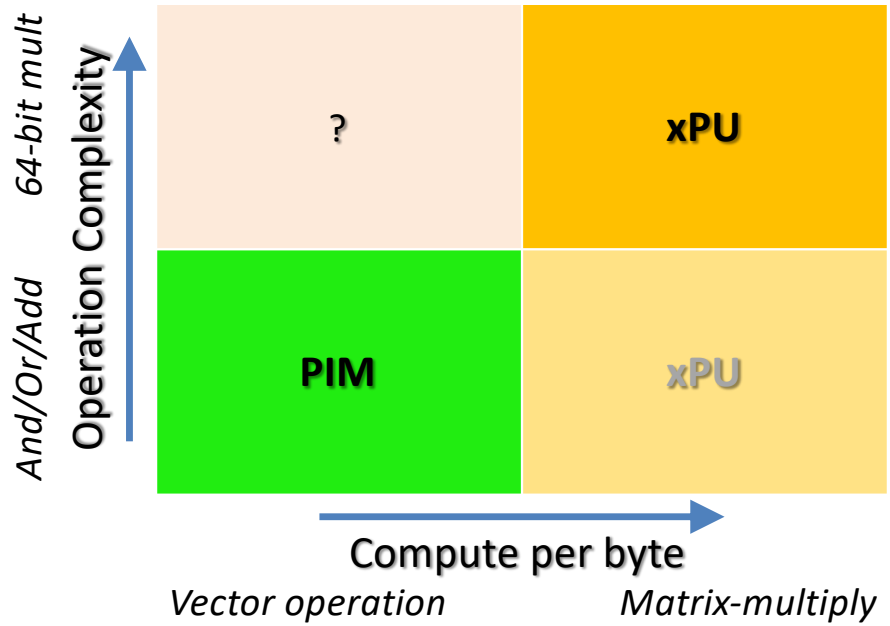
PIM vs. GPU (A100 40GB PCI)[2] Throughput/Watt Ratio
VectorC = VectorA op VectorB

PIM Advantage shows in Simpler, Shorter operations

[1] *O. Leitersdorf et al., "AritPIM: High-Throughput In-Memory Arithmetic," IEEE TETC, 2023*
[2] *https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/a100/pdf/nvidia-a100-datasheet-us-nvidia-1758950-r4-web.pdf*
[3] *A100-40GB max BW=1555GB, A100-80GB max BW=2039GB. Actual TOPS/TFLOPS are 85% of theoretical TOPS/TFLOPS*

Lessons so Far…

- PIM promise is in **eliminating data transfer** (low compute/byte)
- PIM is better in working on **simple operation** (ideally bit-wise)



Thanks!

ASIC² ARCHITECTURES SYSTEMS INTELLIGENT COMPUTING INTEGRATED CIRCUITS

GENPRO
The Israeli RISC-V Consortium

Prime Minister's Office
National Cyber Bureau

Hiroshi Fujiwara Cyber Security Research Center

HiPer

משרד המדע והטכנולוגיה
Israel Ministry of Science and Technology

RBNI

TCE technion computer engineering center

Horizon 2020 European Union Funding for Research & Innovation

ICRI-CI Intel Collaborative Research Institute Computational Intelligence

Advanced Circuit Research Center ACRC

European Research Council

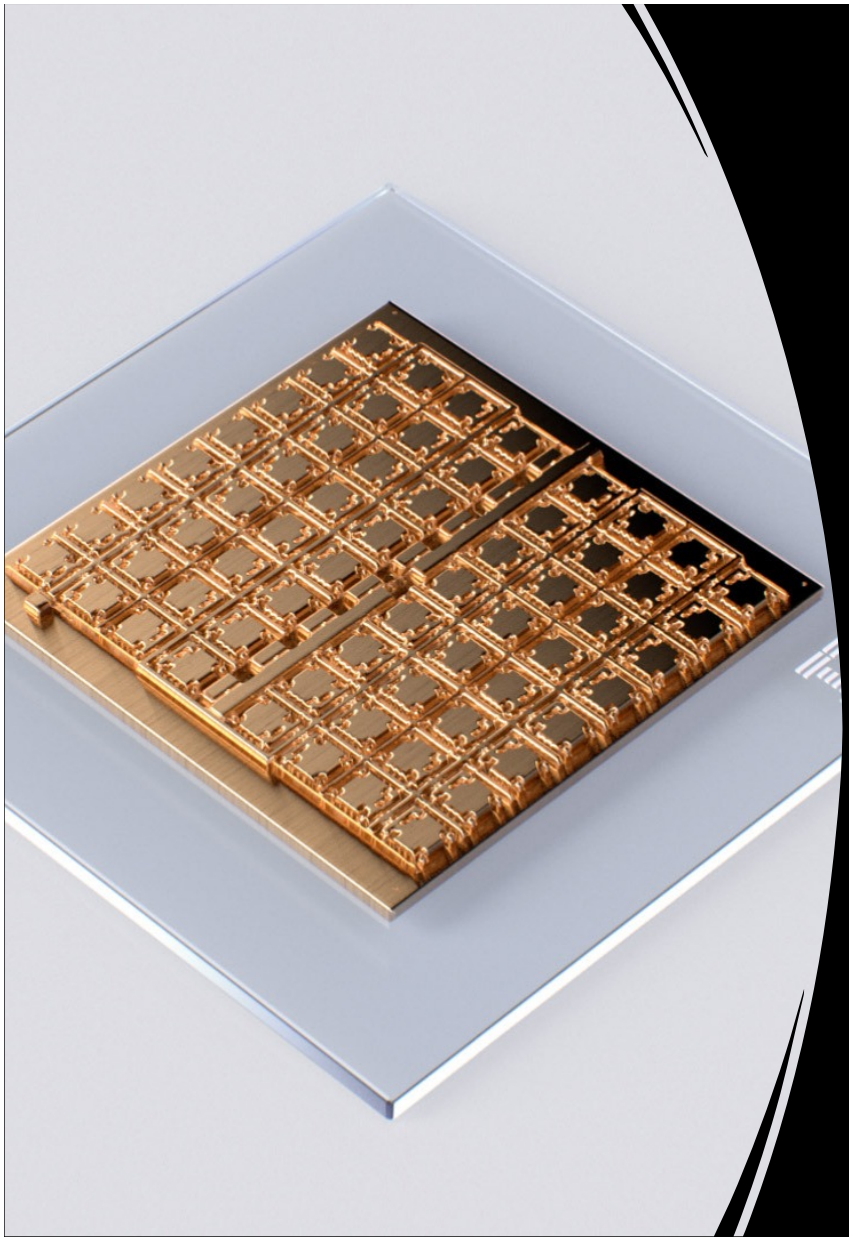
HUAWEI

רשות החדשנות Israel Innovation Authority

CISCO

WD

BSF United States – Israel Binational Science Foundation



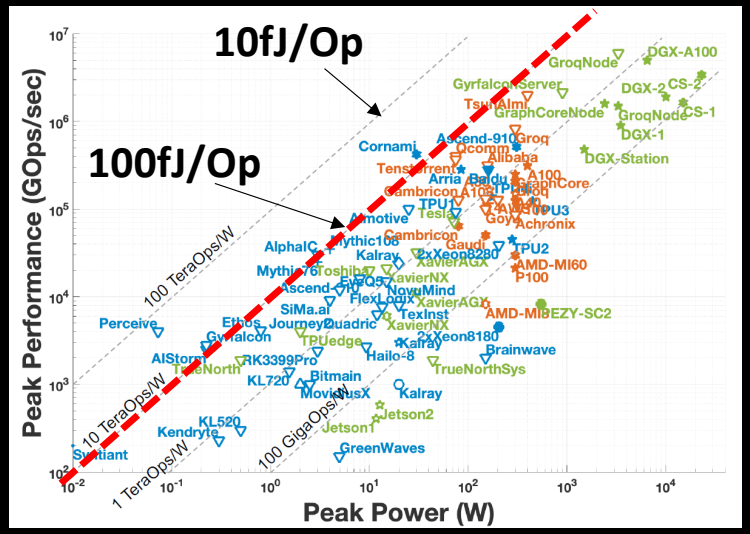
In-memory computing for AI/ML Panel Discussion

Abu Sebastian
Distinguished Scientist
IBM Research - Zurich

Emerging technologies and applications
The elephants in the technology room, Apr. 21, 2023

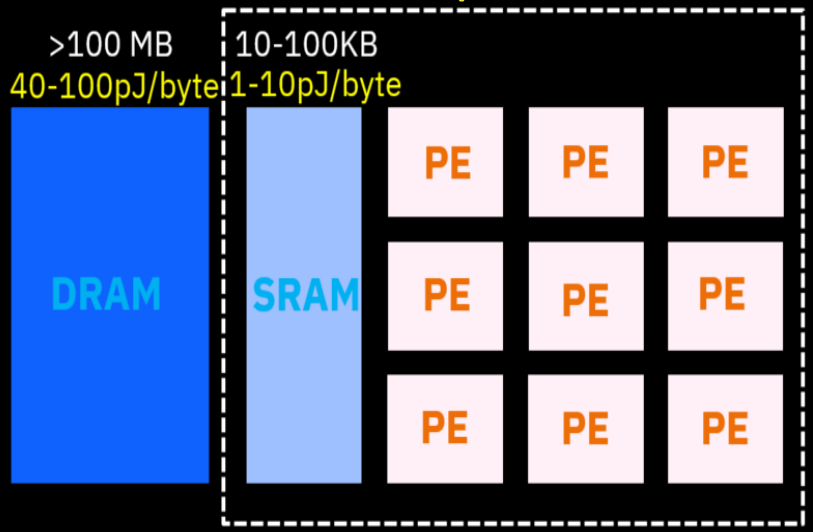
Why in-memory computing for AI/ML?

DNN accelerator zoo



Reuther et al., "AI accelerator survey and trends", IEEE HPEC (2021)

>> 300fJ/Operation

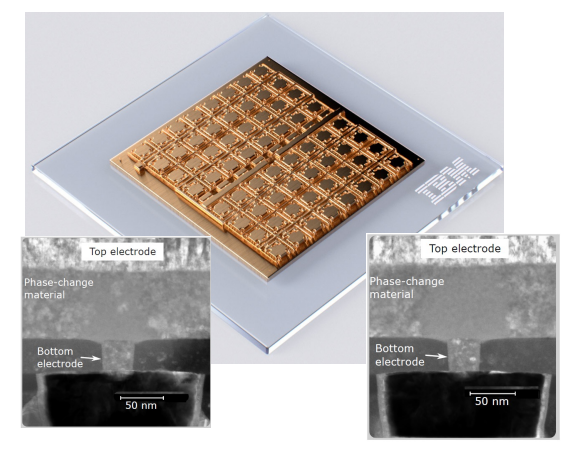


Murmann, IEEE TVLSI (2020)

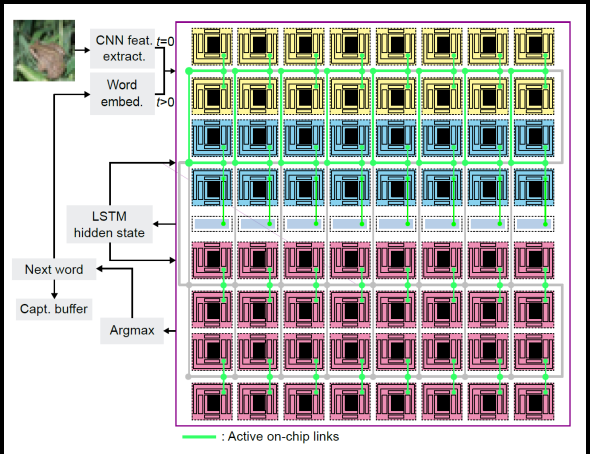
- In-memory computing (digital or mixed-signal) widely considered the next big idea beyond reduced-precision arithmetic
- Imprecise and in-place computation are hallmarks of how brain computes
- IMC being explored using both volatile (SRAM + switched-cap) as well as non-volatile memory (NVM) technology

Learnings from own research

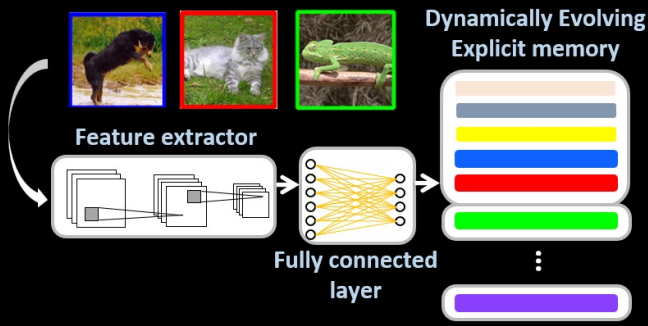
IBM HERMES Project Chip



Conventional deep neural networks



Deep neural networks + "something"



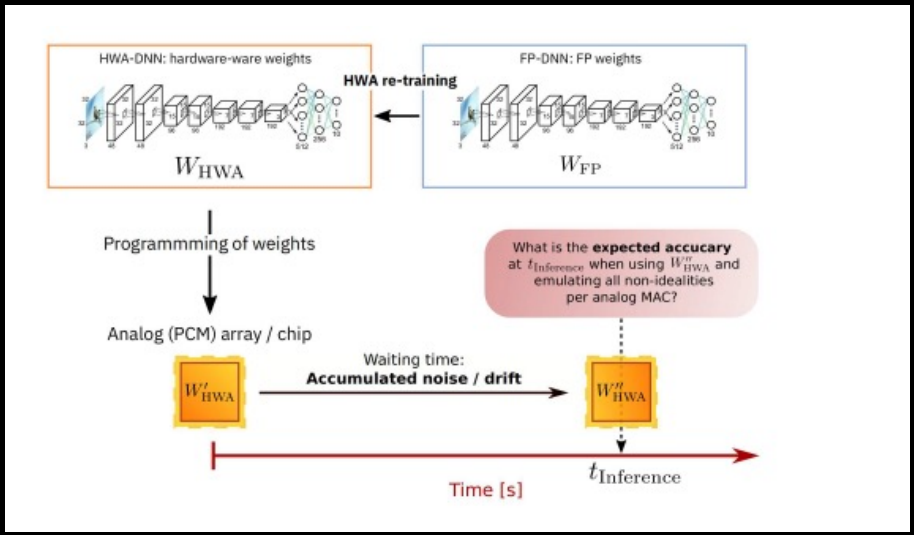
Le Gallo et al., A 64-core mixed-signal in-memory compute chip based on phase-change memory for deep neural network inference (2022)

- Mixed-signal IMC is feasible with embedded non-volatile memory in advanced technology nodes
- 64-core chip with embedded phase-change memory in 14nm CMOS technology
- Seamless integration of analog IMC-tiles with custom digital processing units and a communication fabric

- Highest reported accuracy on CIFAR-10 using NVM-based IMC
- Applications in DNN + Dynamically Evolving Hyper-dimensional Explicit Memory for few-shot continual learning

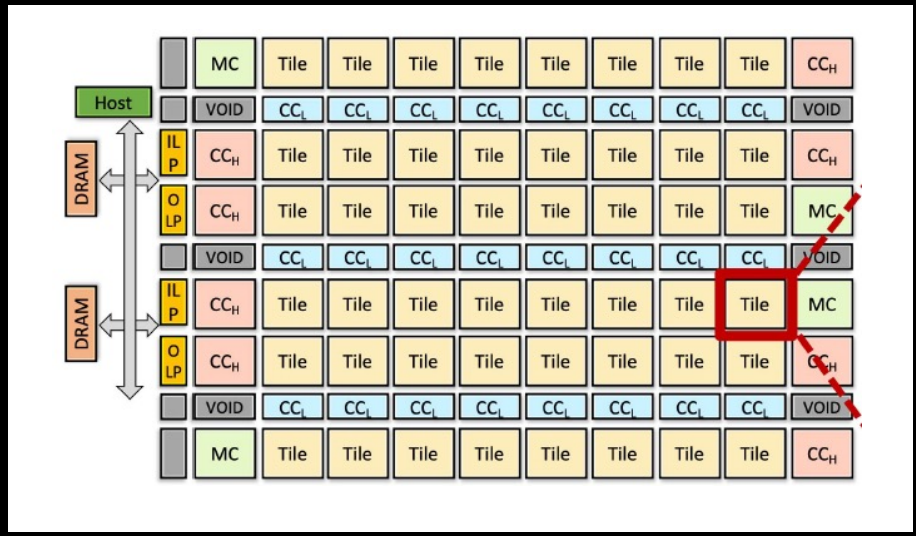
Learnings from own research

Training methodology



Rasch et al., Hardware-aware training for large-scale and diverse deep learning inference workloads using in-memory computing-based accelerators (2023)

Architecture



Jain et al., A Heterogeneous and Programmable Compute-In-Memory Accelerator Architecture for Analog-AI Using Dense 2-D Mesh (2022)

- Custom training required to maintain software-equivalent accuracies
- Highly scalable architecture with heterogenous compute blocks and massively parallel communication ideal for IMC with NVM

In-memory computing: Assessment

- SRAM-based IMC (digital or mixed-signal) on the verge of being commercialized or being integrated into DNN accelerators
- NVM-based IMC (mostly mixed-signal) is getting mature for embedded applications
- Specific NVM technology is mostly irrelevant (PCM, ReRAM, FeFET, Flash etc.)
- To realize the full potential of IMC, weight capacity is essential
- Highly dense 2D NVM or 3D NVM-based IMC would be a critical breakthrough (Hinton's "GPT on a toaster!")
  - ✓ Non-volatile weight storage
  - ✓ 10-100 TOPS/W
  - ✓ TDP of 1-10W
  - ✓ Weight capacity in the order of hundreds of billions of weights
- But could IMC become a key component of future AI/ML Hardware platforms?

If weight capacity issue persists, IMC will remain a niche technology for edge applications, else it could be a key component of future AI/ML platforms

KEPLER

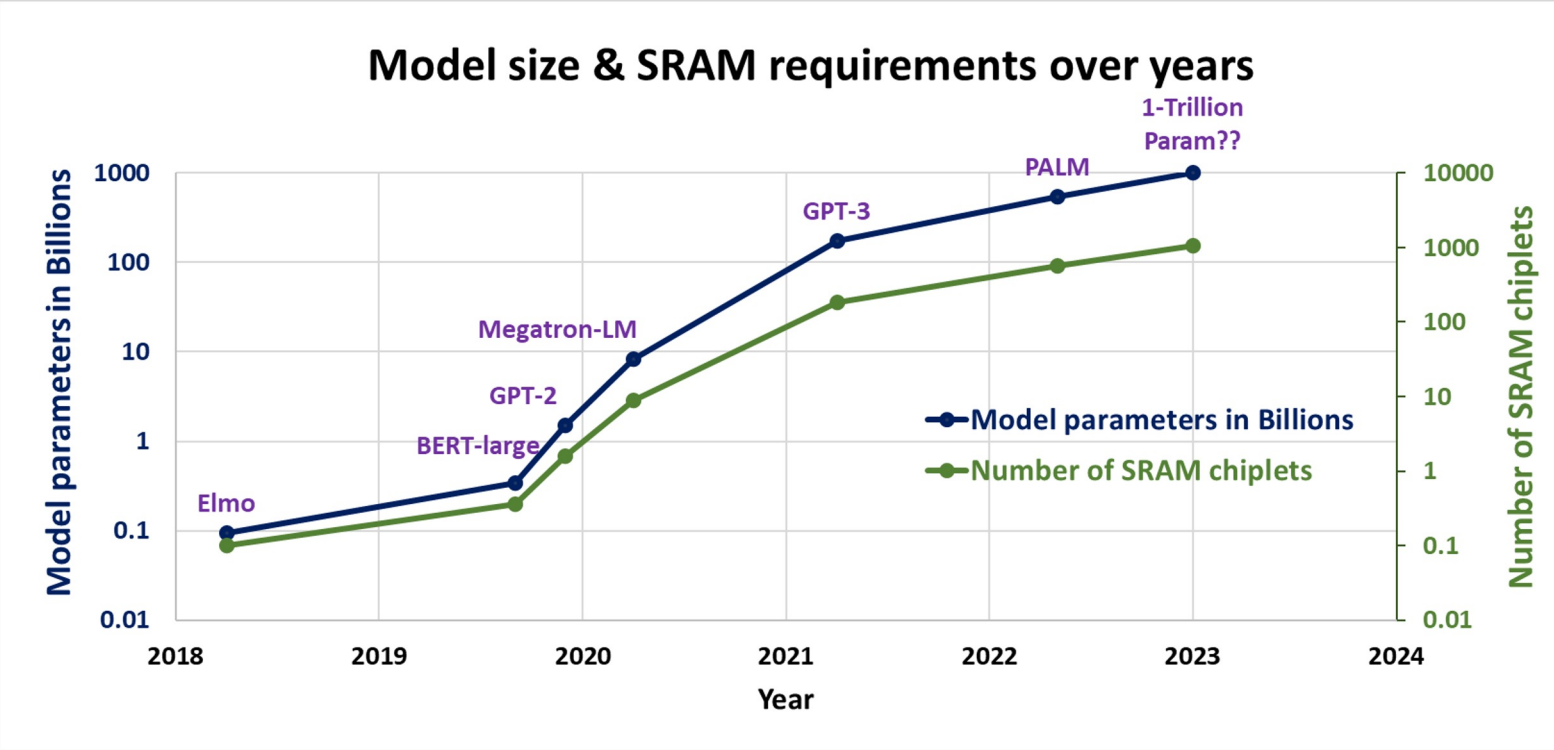
Enable the next generation of computing

In-memory computing panel

Amrita Mathuriya

Problem 1: Cost of SRAM

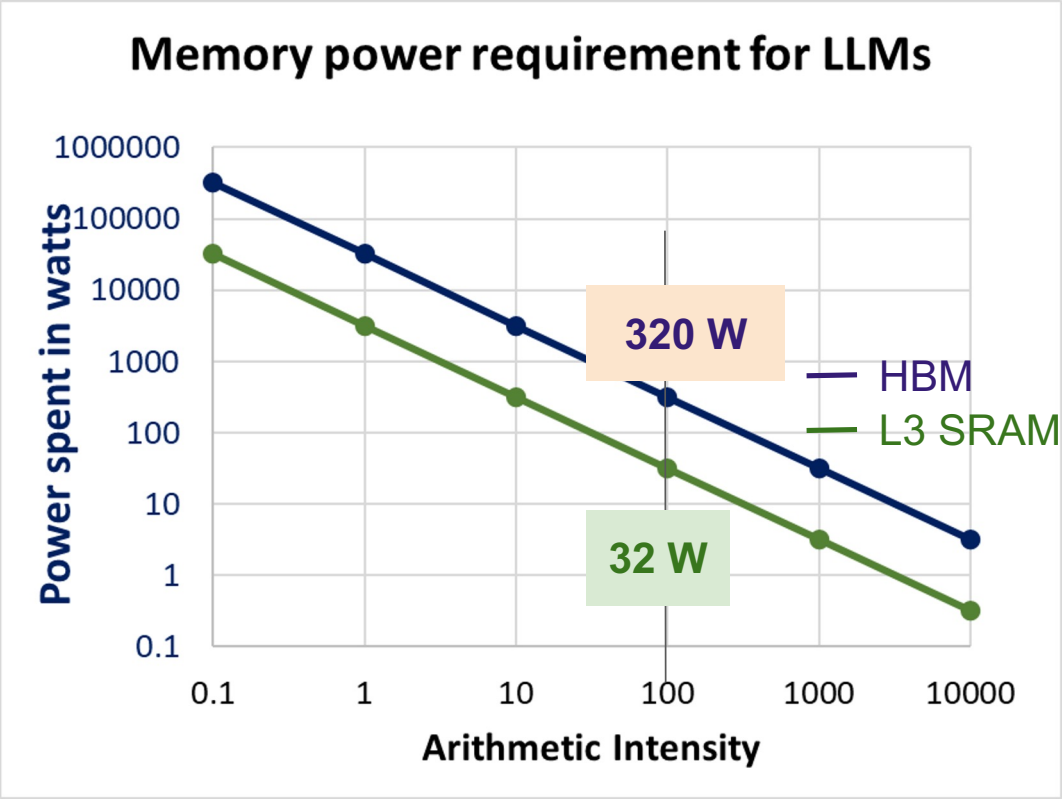
KEPLER Computing



For a Trillion parameter model, it will take > 1000 SRAM chiplets of 200mm^2.

Problem 2: Power of DRAM

KEPLER Computing

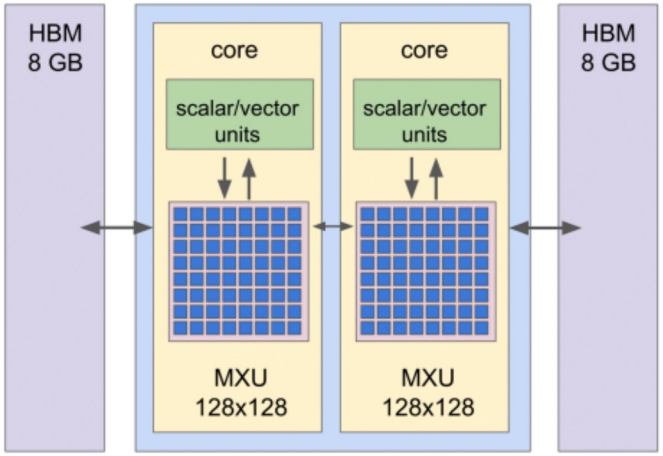


- System TOPs = 1000
- Arithmetic intensity = 100
- Bandwidth required for 100% utilization = 10 TB/s
- Only feasible with on-chip data-movement!

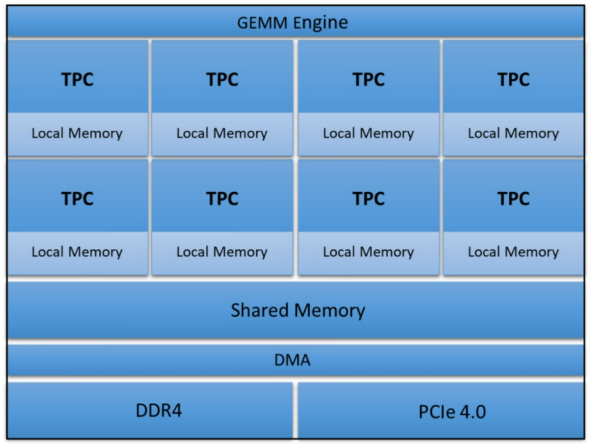
On-chip memory enables 10x power reduction for data-movement.

Progress towards in-memory compute

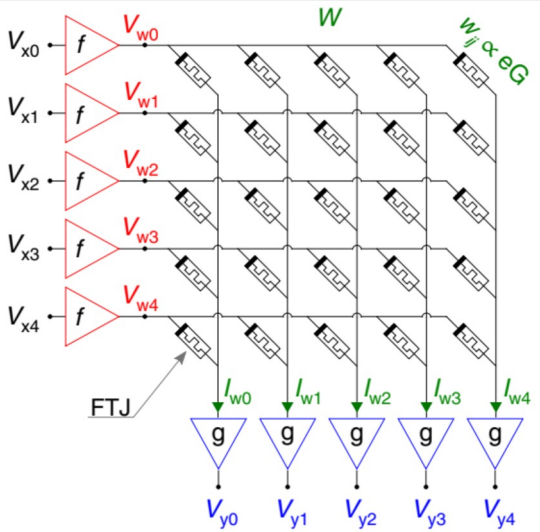
KEPLER



Google TPU[1]



Habana[2]



In-memory compute[3]

Large processor cores with HBM

Small cores with small memory chunks

Weight stationary flow with SRAM and NV Memories

https://www.anandtech.com/show/12429/google-cloud-announces-cloud-tpu-beta-availability

https://www.gazettabyte.com/home/2018/11/15/habana-labs-unveils-its-ai-processor-plans.html

Berdan, Radu, et al. "Low-power linear computation using nonlinear ferroelectric tunnel junction memristors." *Nature Electronics* 3.5 (2020): 259-266.

Challenges for in-memory compute

KEPLER

1. Digital SRAM computing :
   - **Engineering/Architecture problem:** Compiler. Efficiency with flexibility.
   - Suffers from large bitcell size, large static power.
     - **Research:** Technology scaling.
2. Analog computing:
   - **Co-design process and arch:** Precision, variations and SNR
   - Suffers from getting multi-bit precision implemented reliably.
3. Emerging memories
   - **R&D:** High density, endurance, integration with CMOS on advanced nodes
   - There is a need for a reliable, scalable, high density and fast memory.

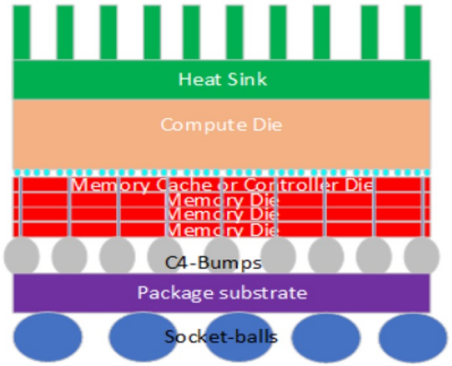
How to go about the problem?

KEPLER

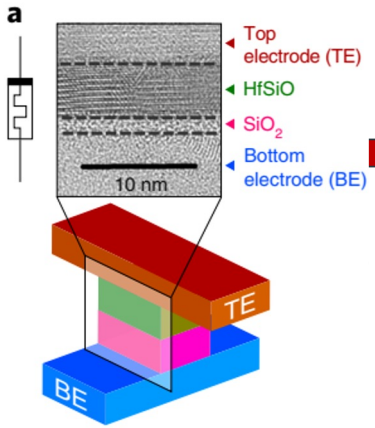
Combine the following approaches:

1. 3D-stacking of chiplets => Facilitates introduction of new memory technologies and lower cost!
2. High density bitcell compared to SRAM bitcell!
3. Analog in-memory compute => High compute density!
4. Multi-bit weight cell => even high density!

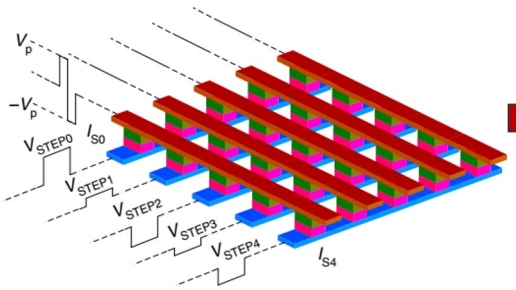
Example with an emerging memory



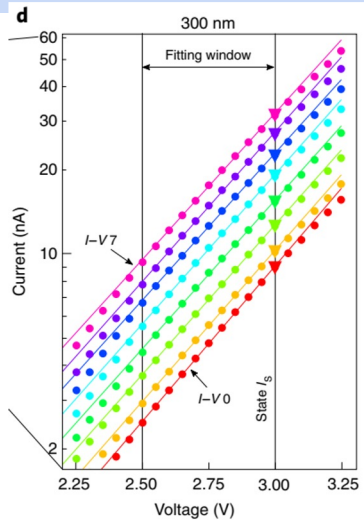
Multiple stacked memory dies



Dense bit-cell



Analog MAC



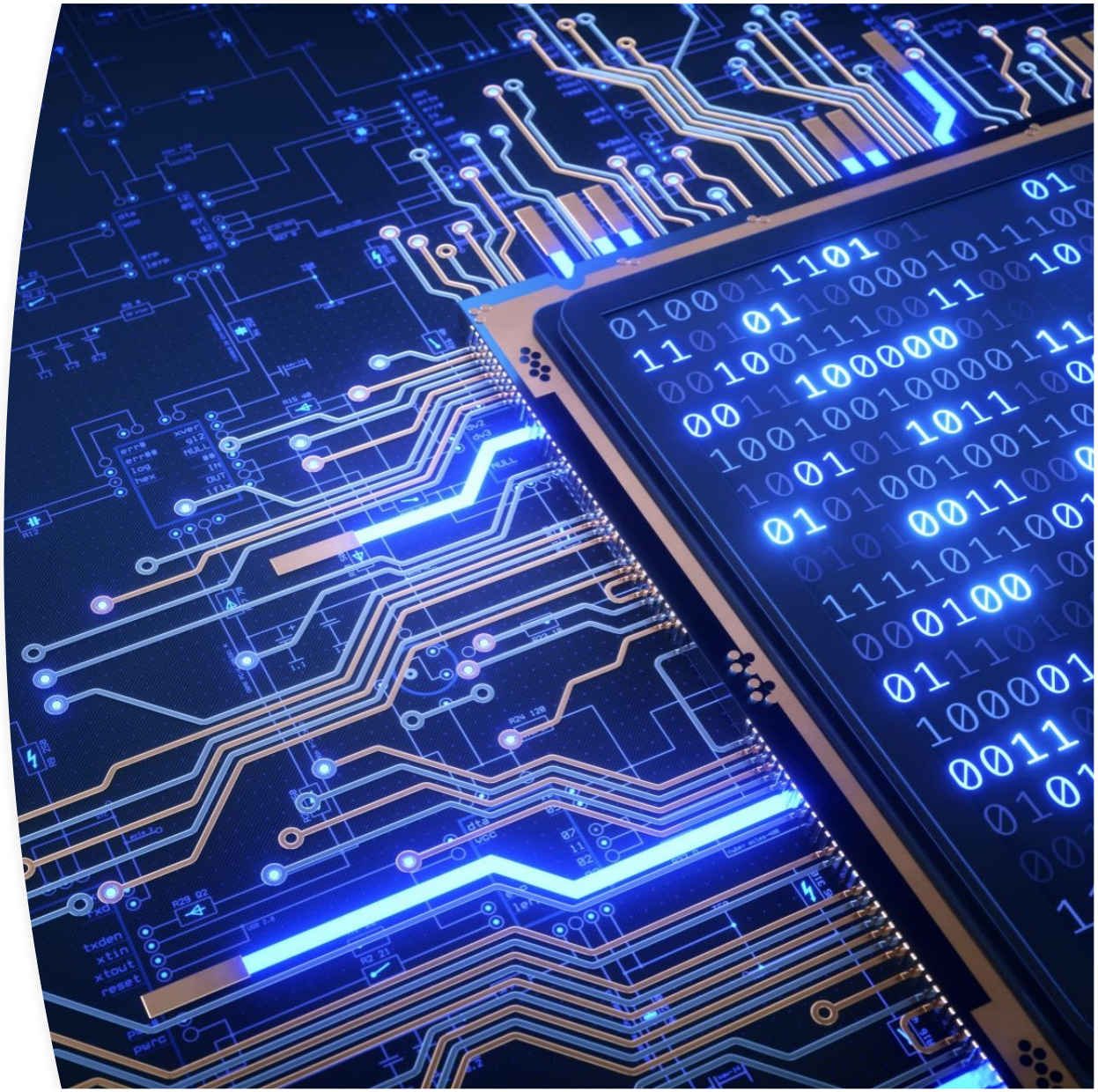
Multi-bit precision

Berdan, Radu, et al. "Low-power linear computation using nonlinear ferroelectric tunnel junction memristors." *Nature Electronics* 3.5 (2020): 259-266.

In-Memory computing a niche area or the main hardware platform for AI/ML?

Patrick Groeneveld

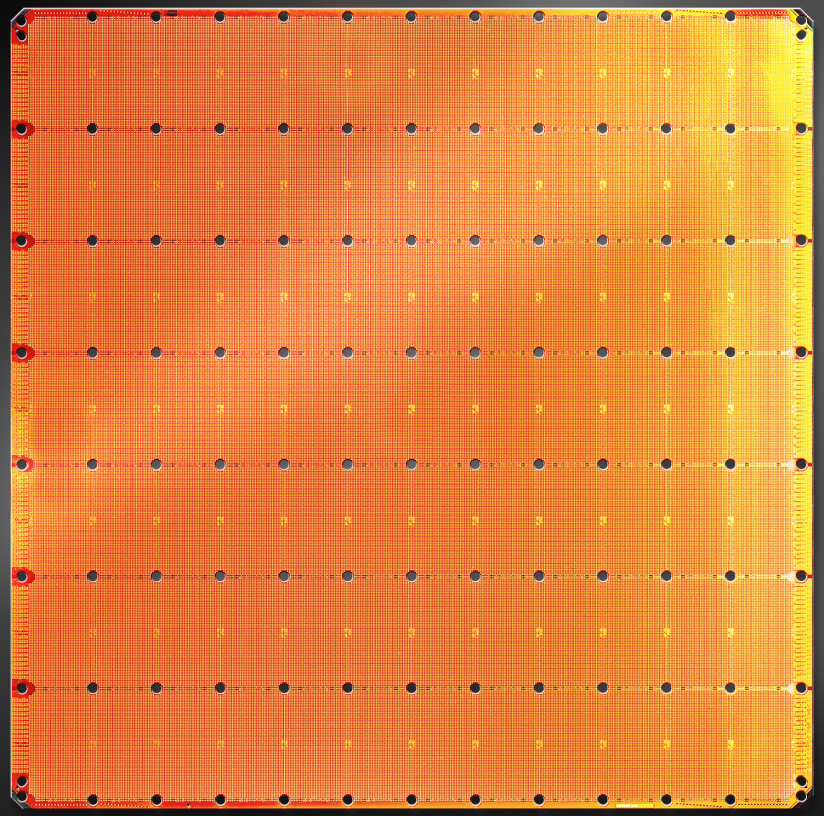
April 22, 2023



Cerebras
Wafer Scale Engine
AI Supercomputer

- Focus: Supercomputer for ML training
- Single 21.5cm by 21.5cm chip in 7nm
  - **2.4 Trillion transistors**
- 850,000 ‘AI processor cores’
  - In a 800 by 1060 array
- Total on-chip memory:
  ~40Gb fast SRAM
- ~100 PetaByte/second fabric bandwidth

Cerebras Wafer Scale Engine



20,000 Watt

Cerebras WSE

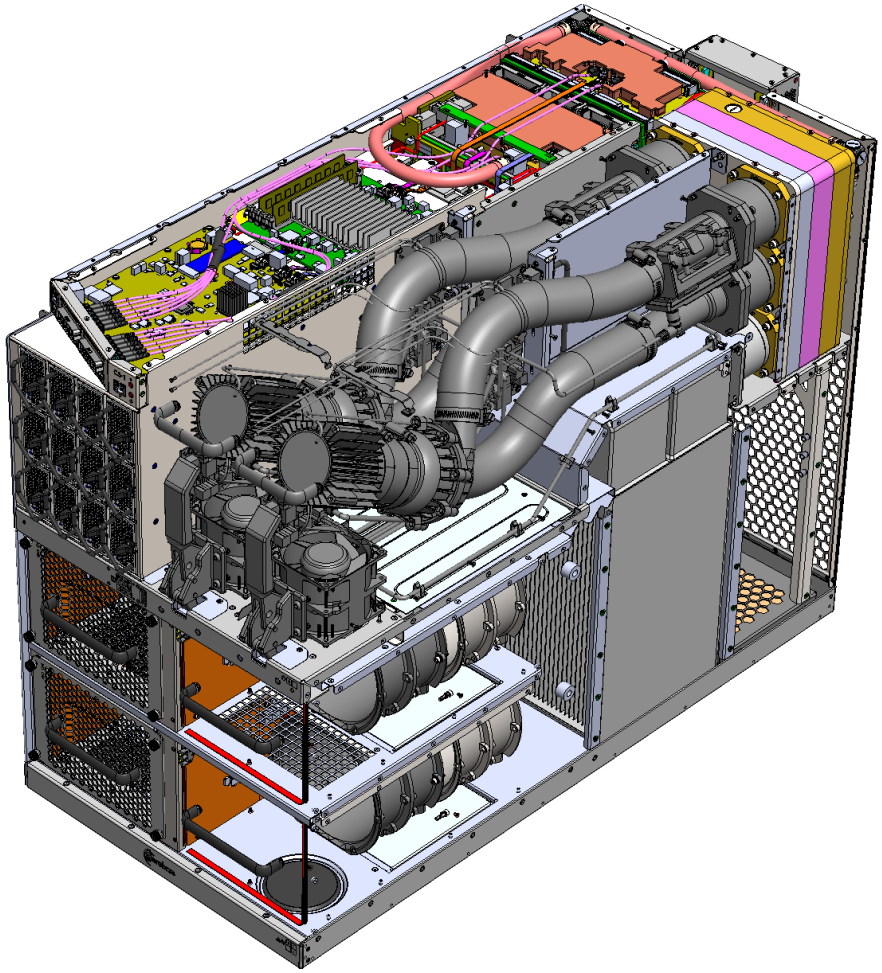
46,225 mm² Silicon

Largest GPU

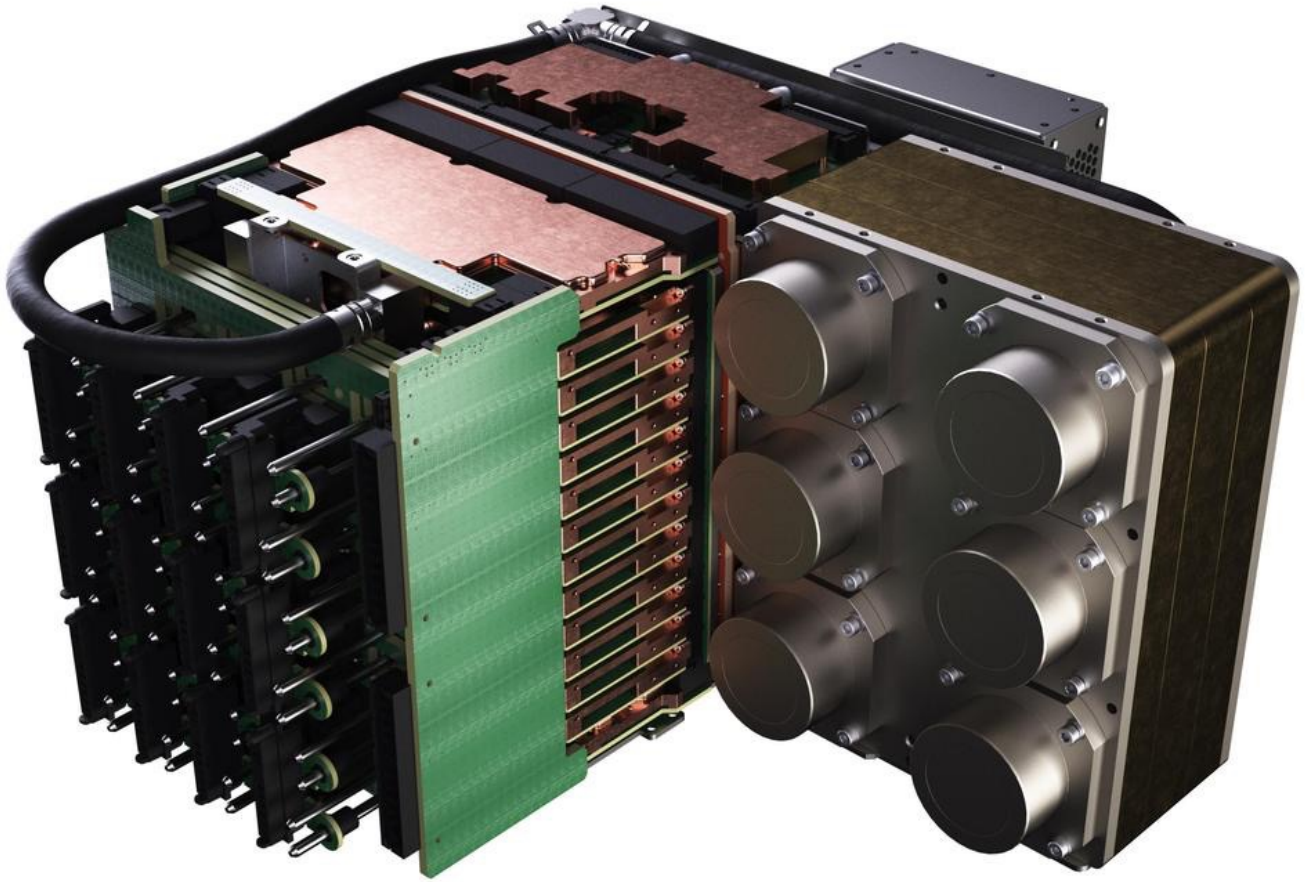
815 mm² Silicon

200 Watt

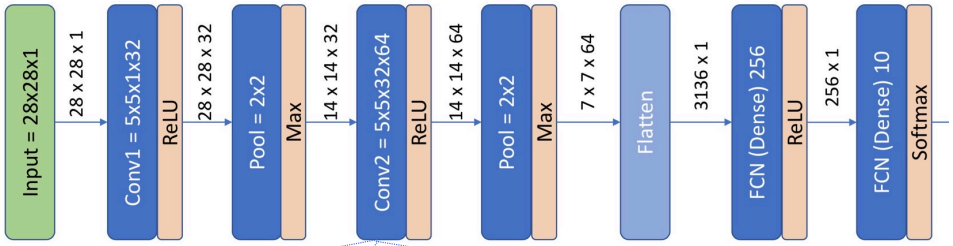
WSE Hardware: The box



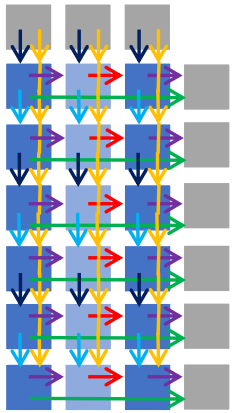
The ‘Engine Block’ that holds the WSE



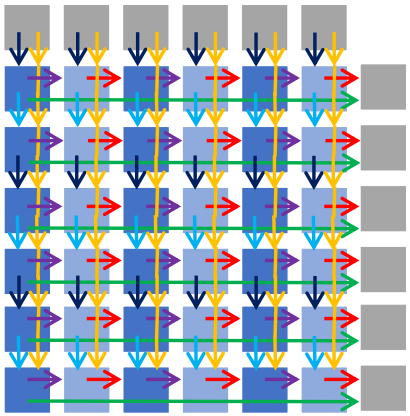
Area vs Performance of a Kernel



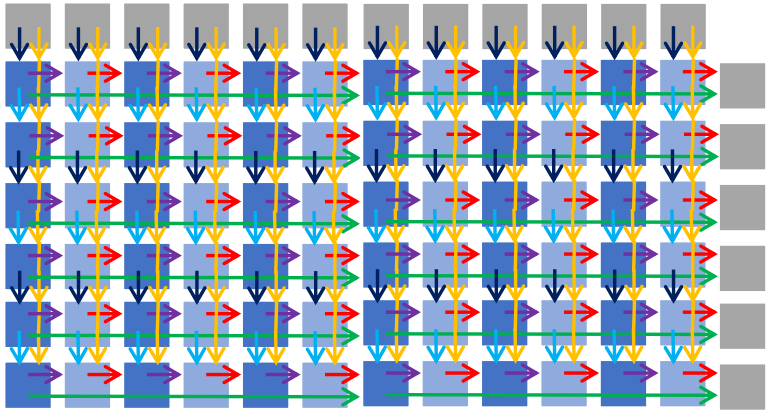
2X base task per PE as 3X6 core
Expect ~2X slower, 2X memory



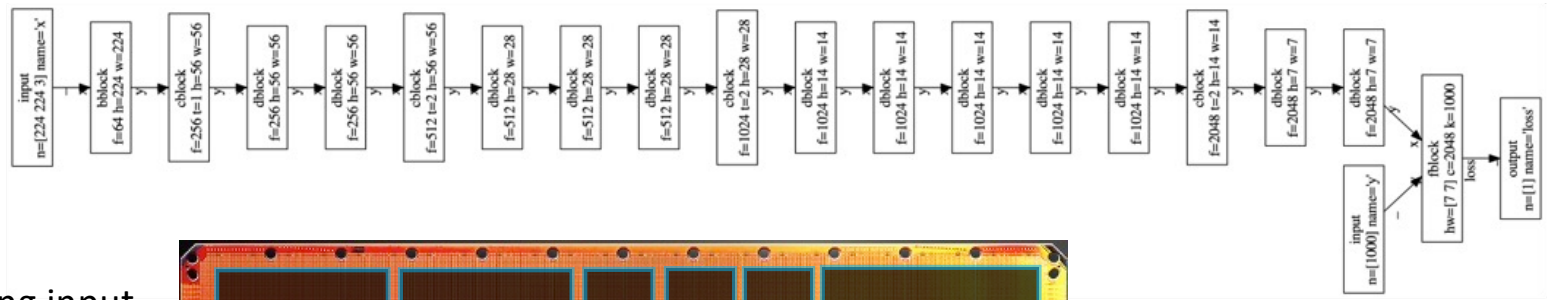
Base task, 6X6 core



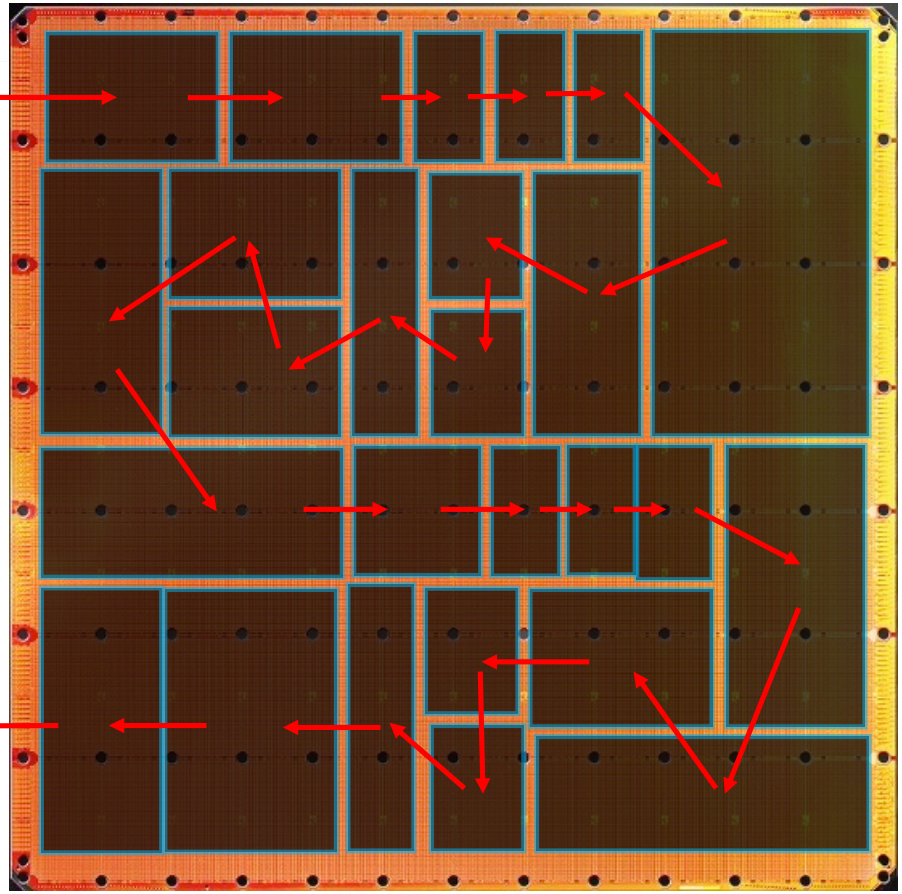
½ X base task per PE as 12X6 core
Expect ~2X faster, ½ memory



Layer-pipelined execution



Continuously Streaming input



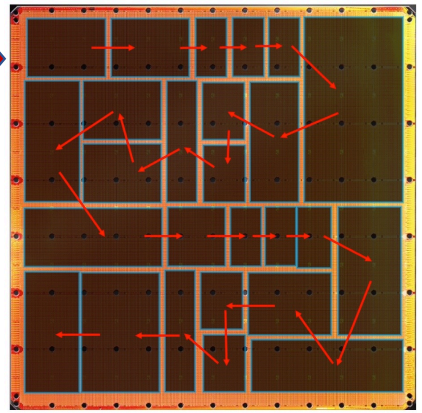
cat

Continuously Streaming output

Different methods depending on model

Typical sized models

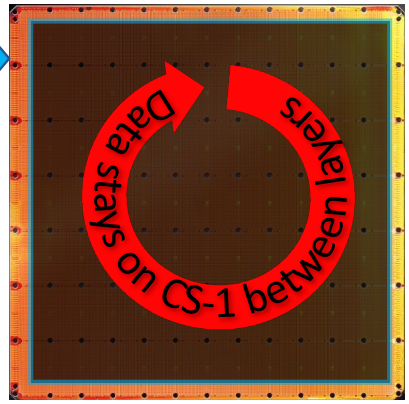
Stream Unlimited **Data**



- Layers run pipelined: the entire model runs in parallel
- **Data streaming** into CS-2
- Stationary: Weights/parameters
- Leverages activation and data sparsity
- Exceptional throughput, Billion parameter models

Extremely large models (LLM)

Stream Unlimited **Weights**



- Layers run sequential, one at a time on CS-2
- Batch of data stays stationary
- **Weight streaming** into CS-2
- Leverages weight sparsity
- Exceptional capacity, **Trillion** parameter models

Addressing the Panel Questions

- Large Homogenous Multiprocessor or true in-memory compute?
  - The first!
- The effects of the SRAM problem
  - Adjust ratio memory-compute: stream everything
- CMOS-only or novel devices for in-memory compute
  - CMOS-only will have the edge
- Digital or analog multiplication?
  - Digital, no question about that
- What will the best approach to support and accelerate computation?
  - Mapping of PyTorch/Tensorflow into compute kernels is underappreciated problem