Computational memory: A stepping stone to non-von Neumann computing?

Abu Sebastian
IBM Research – Zurich

Stanford EE380, 7th March 2018

IBM

IBM Research - Zurich

IBM Research - Zurich

Outline

- Motivation for in-memory computing
- Constituent elements of computational memory
- Computational memory: Logical operations
- Computational memory: Arithmetic operations
- Computational memory: Computing with device dynamics
- Mixed-precision in-memory computing
- Summary & Outlook

2.5 exabytes of data created every day.

1 exabyte is 1,000,000,000,000,000,000 or 10^{18}

90% of the data in the world today has been created in the last **two years** alone.

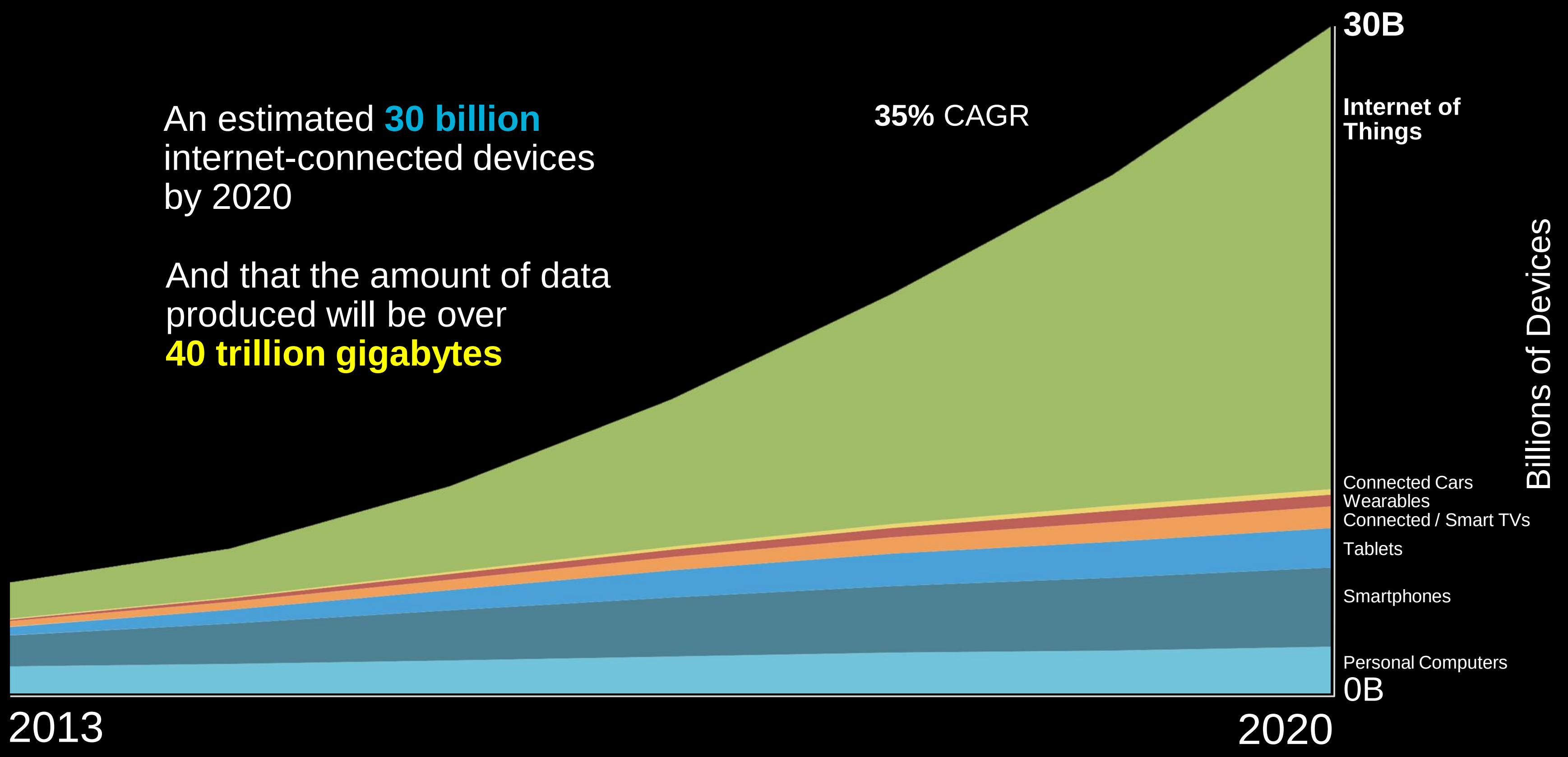
Global scientific output **doubles every nine years**

>2.5 million papers per year
>300,000 US Patents annually
>100 million substances in CAS Registry, growing exponentially

Internet of Things (IoT)

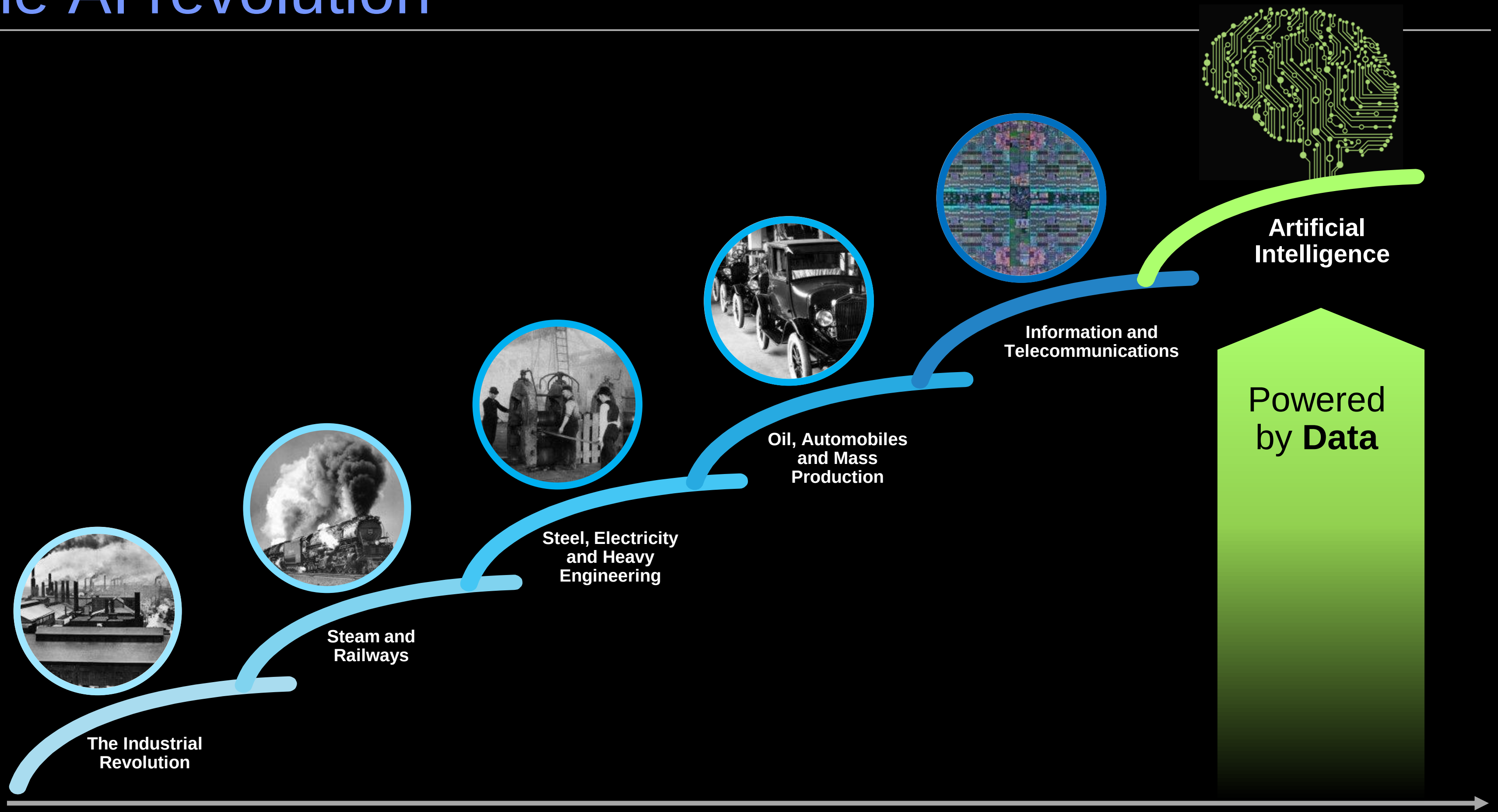
An estimated **30 billion** internet-connected devices by 2020

And that the amount of data produced will be over **40 trillion gigabytes**



Source: BI Intelligence Estimates

The AI revolution



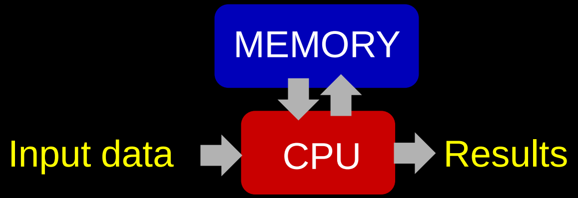
The computing challenge

~20W ~80,000 W

IBM's Watson in *Jeopardy!*

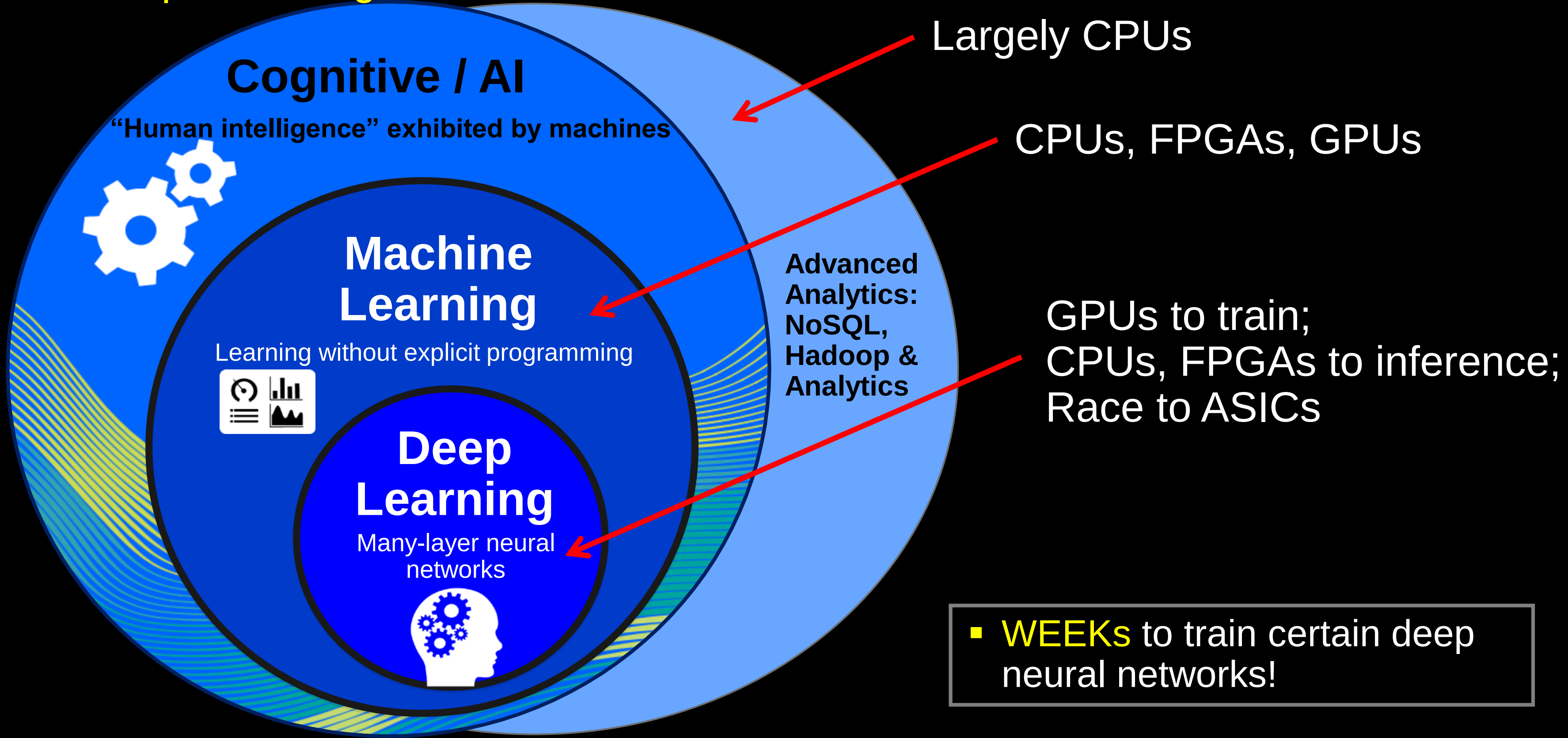
2880 processor threads
16 terabytes of RAM
80 kW of power
20 tons of air-conditioned cooling capacity

Conventional von Neumann computing architecture



The computing challenge

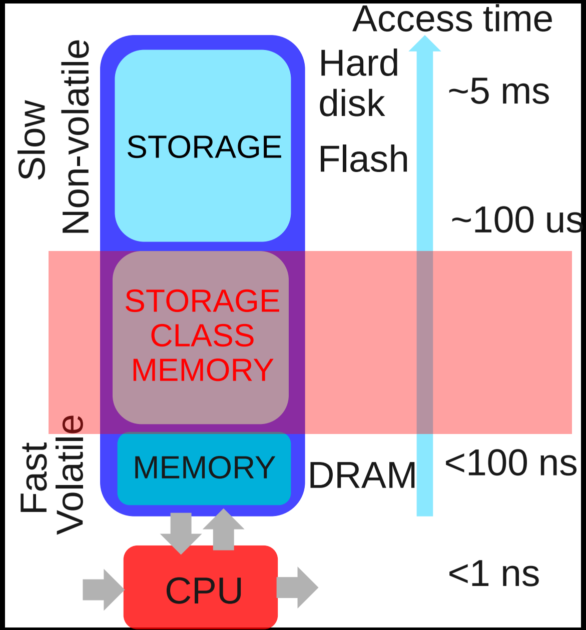
Landscape of AI Algorithms



- WEEKs to train certain deep neural networks!

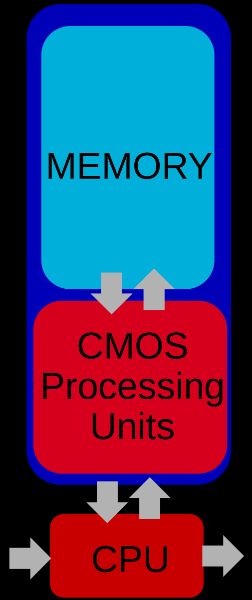
Advances in von Neumann computing

Storage class memory



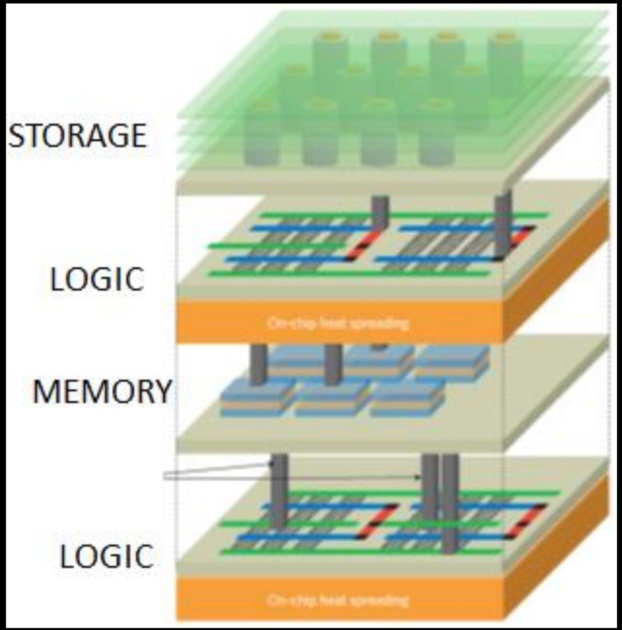
Burr *et al.*, *IBM J. Res. Dev.*, 2008

Processor-in-memory (near memory computing)



Vermij *et al.*, *Proc. ACM CF*, 2016

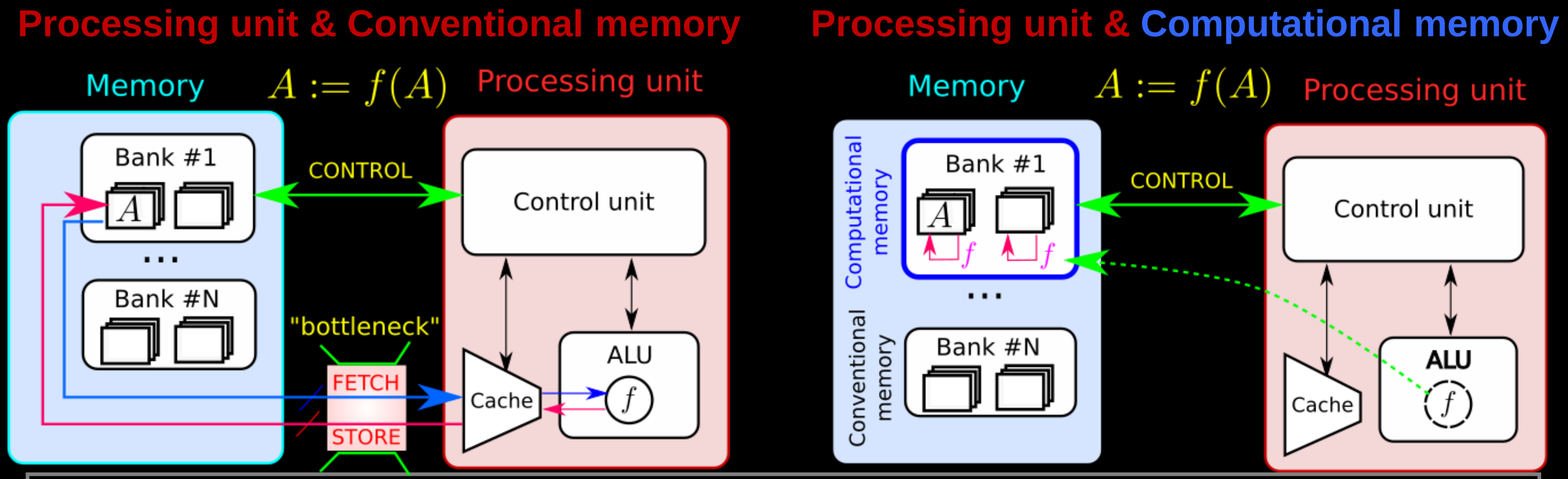
Monolithic 3D integration



Wong, Salahuddin, *Nature Nano.*, 2015

- Still confined within the von Neumann paradigm
- Minimize the time and distance to memory access

Beyond von Neumann: In-memory computing



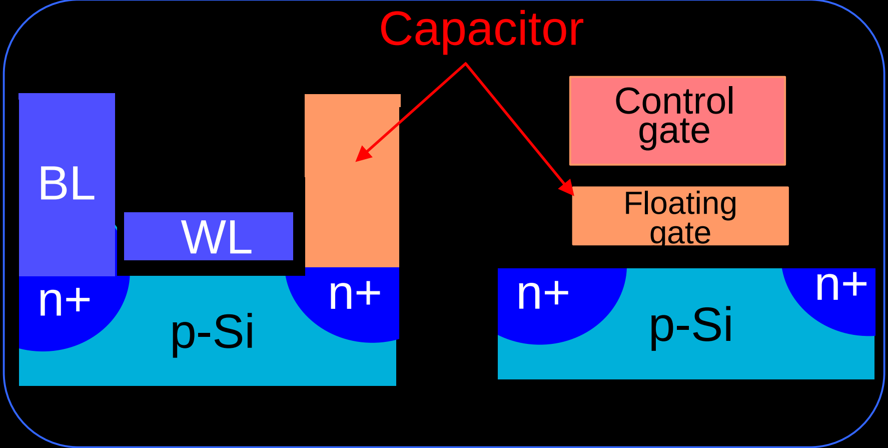
- Perform "certain" computational tasks using "certain" memory cores/units without the need to shuttle data back and forth in the process
  - ✓ Logical operations
  - ✓ Arithmetic operations
  - ✓ Machine learning algorithms
- Exploits the physical attributes and state dynamics of the memory devices

Outline

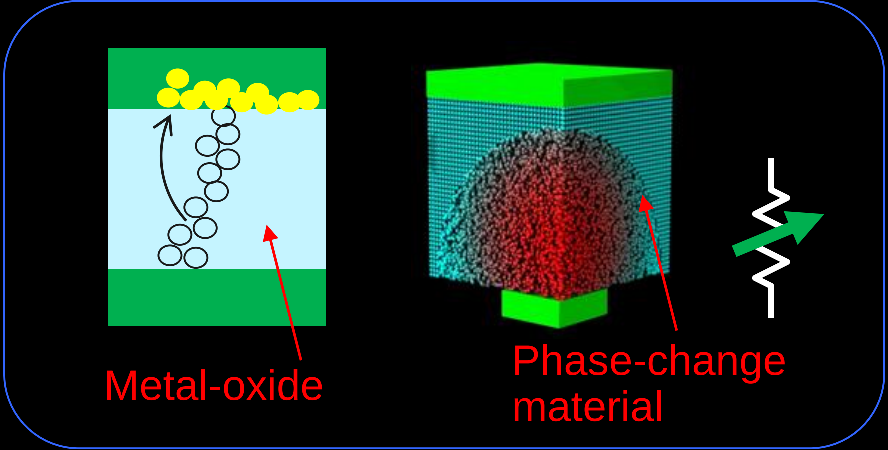
- Motivation for in-memory computing
- Constituent elements of computational memory
- Computational memory: Logical operations
- Computational memory: Arithmetic operations
- Computational memory: Computing with device dynamics
- Mixed-precision in-memory computing
- Summary & Outlook

Constituent elements of computational memory

“Charge on a capacitor”

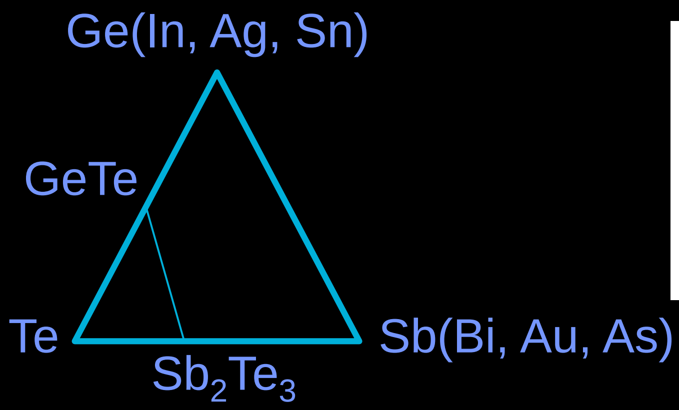


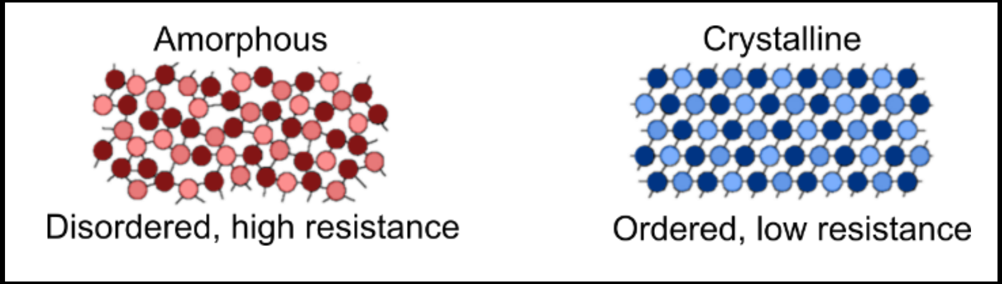
“Alternate atomic arrangements”



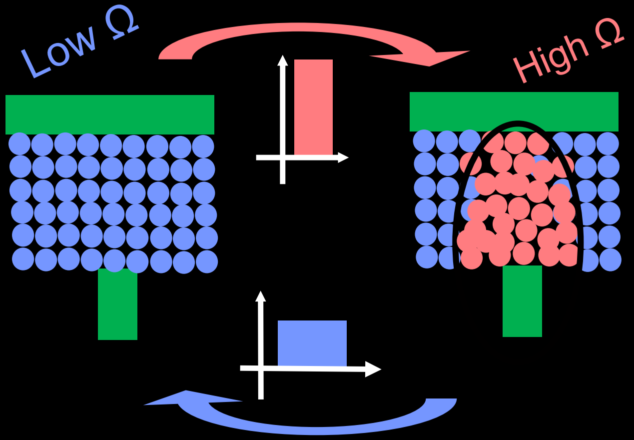
- Difference in atomic arrangements induced by the application of electrical pulses and measured as a difference in electrical resistance
- Resistive memory devices or memristive devices
- Based on physical mechanisms such as ionic drift and phase transition

Phase-change memory

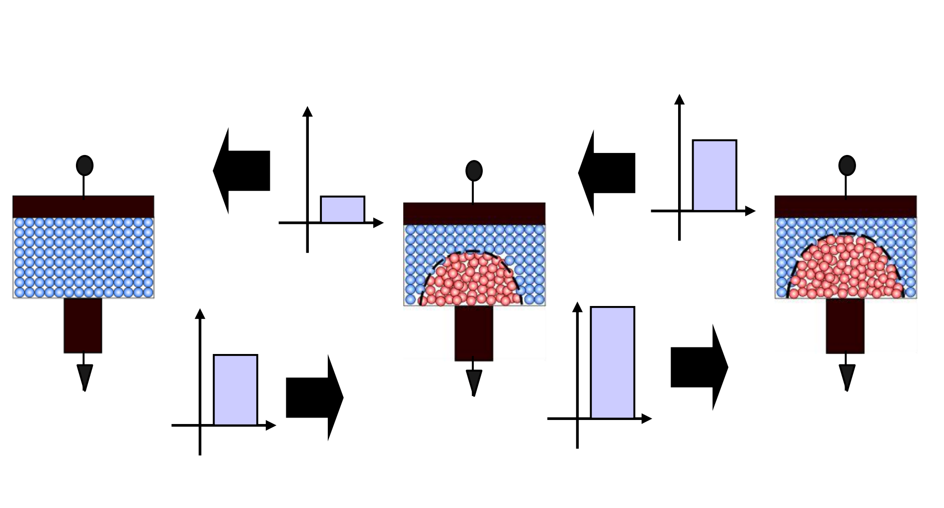


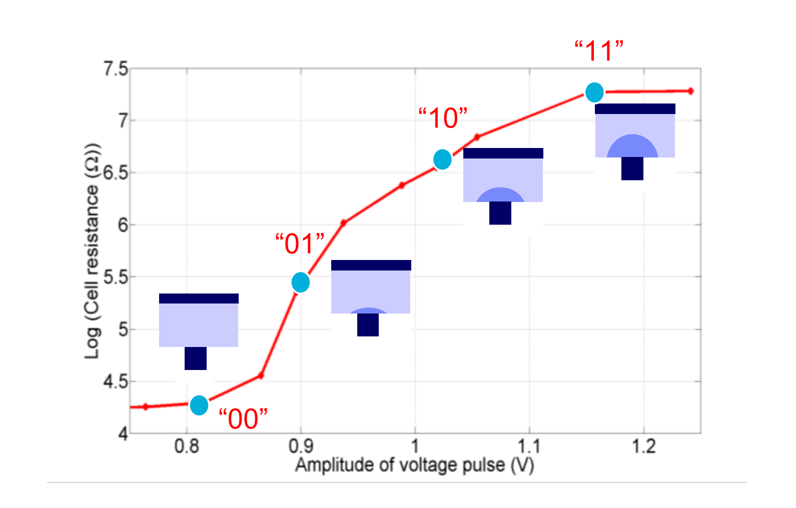


- A nanometric volume of phase-change material between two electrodes
- "WRITE" Process
  - ✓ By applying a voltage pulse the material can be changed from the crystalline phase (SET) to the amorphous phase (RESET)
- "READ" process
  - ✓ Low-field electrical resistance



Multi-level storage capability

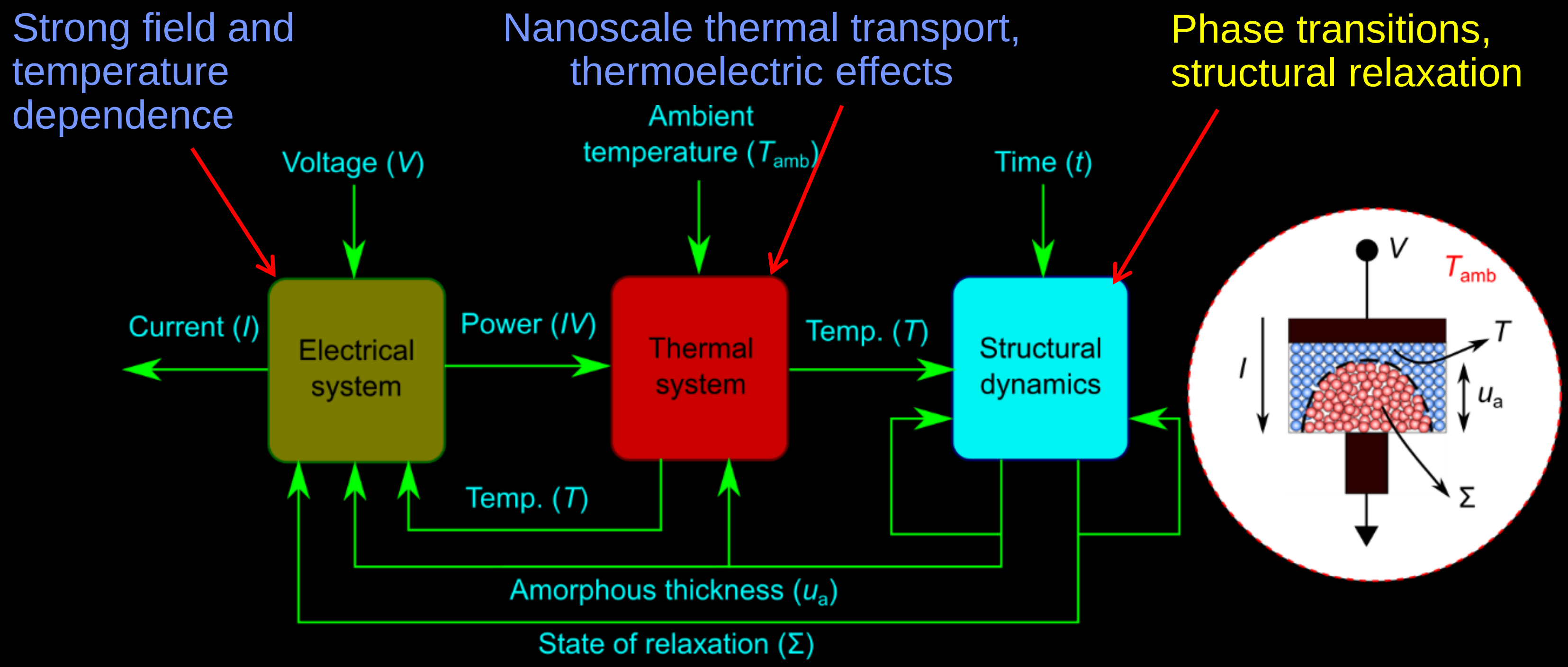




- Possible to achieve intermediate phase configurations
- Can achieve a continuum of resistance/conductance levels
- Essentially an analog storage device!

Burr *et al*., IEEE JETCAS, 2016; Sebastian *et al*., Proc. E\PCOS, 2016

Rich dynamic behavior



- Feedback interconnection of electrical, thermal and structural dynamics

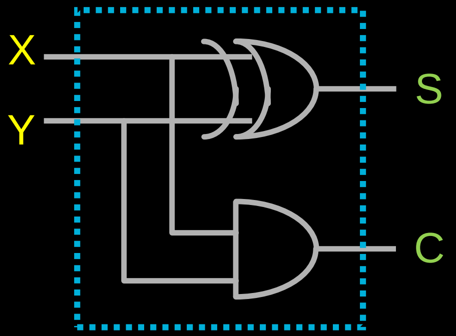
Sebastian *et al.*, *Nature Comm.*, 2014; Le Gallo *et al.*, *New J. Phys.*, 2015; Le Gallo *et al.*, *JAP*, 2016; Sebastian *et al.*, *IRPS 2015*

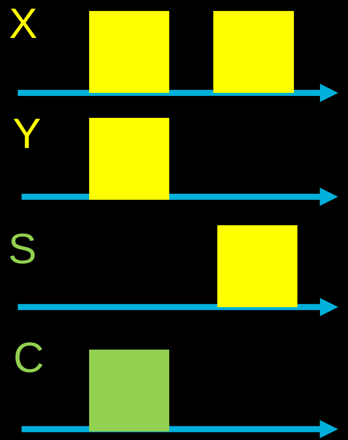
Outline

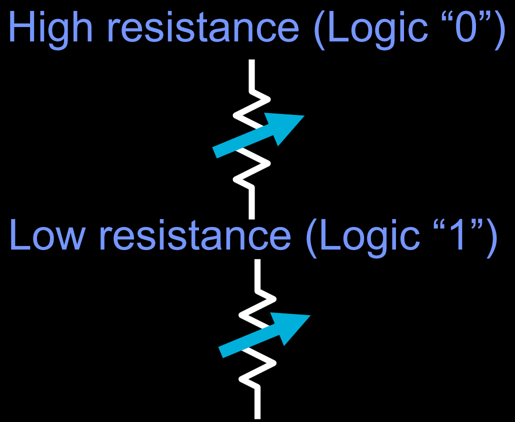
- Motivation for in-memory computing
- Constituent elements of computational memory
- Computational memory: Logical operations
- Computational memory: Arithmetic operations
- Computational memory: Computing with device dynamics
- Mixed-precision in-memory computing
- Summary & Outlook

Abu Sebastian, IBM Research - Zurich

Logic design using resistive memory devices







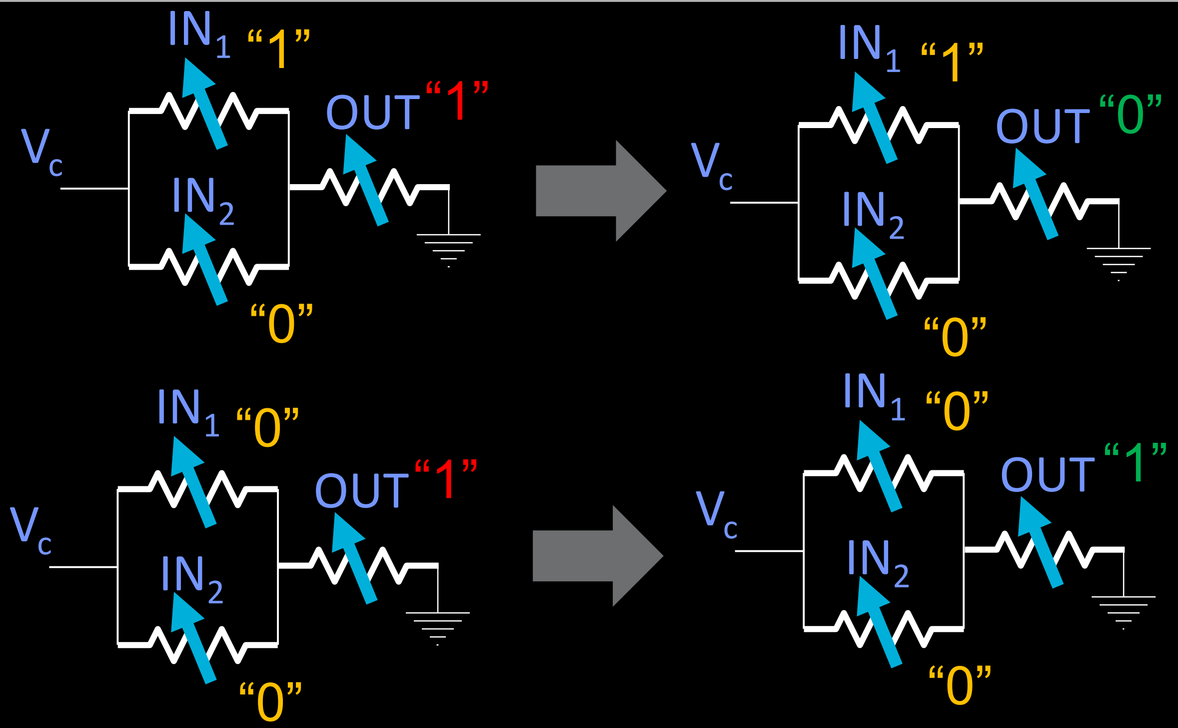
- Voltage serves as the single logic state variable in conventional CMOS
- CMOS gates regenerate this state variable during computation
- How about using the resistance state of memristive devices as a state variable?
- Can toggle the states by applying voltage signals; only binary storage required
- Logical operations enabled by the interaction between voltage and resistance state variables

Borghetti *et al.*, *Nature*, 2010 Vourkas, Sirakoulis, *IEEE CAS Magazine*, 2017

Stateful logic

NOR

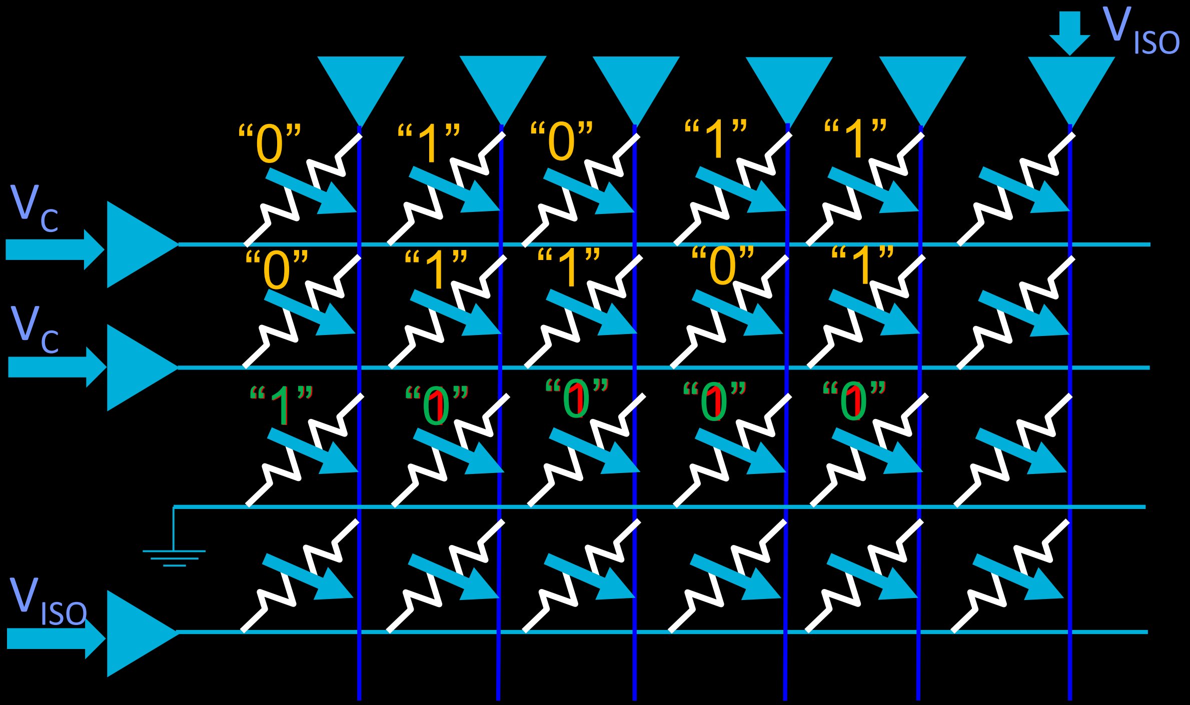
IN_1	IN_2	OUT
0	0	1
0	1	0
1	0	0
1	1	0



- Stateful logic exhibited by certain memristive logic families
- The Boolean variable is represented only in terms of the resistance state

Kvatinsky *et al.*, *IEEE TCAS*, 2014

Bulk bitwise operations



- Can perform bulk bit-wise operations in a cross-bar array
- Each processing task can be divided into a sequence of such operations

Talati *et al.*, *IEEE Trans. on Nanotech.*, 2016

Outline

- Motivation for in-memory computing
- Constituent elements of computational memory
- Computational memory: Logical operations
- Computational memory: Arithmetic operations
- Computational memory: Computing with device dynamics
- Mixed-precision in-memory computing
- Summary & Outlook

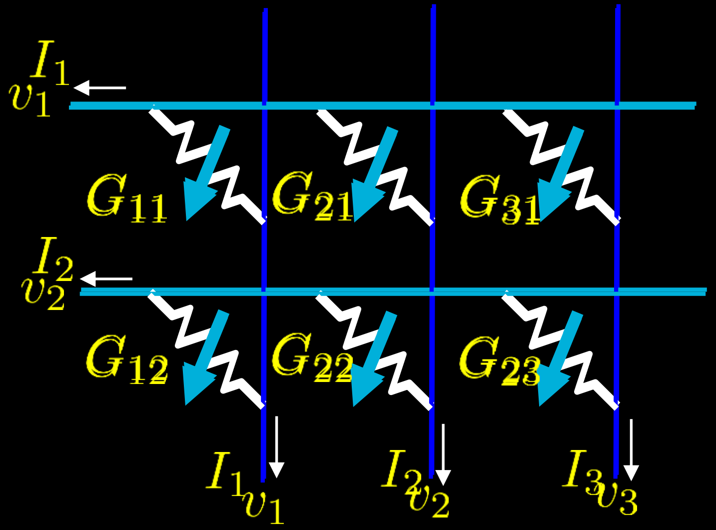
Matrix-vector multiplication

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \\ A_{31} & A_{32} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

MAP to conductance values

MAP to read voltage

DECIPHER from the current

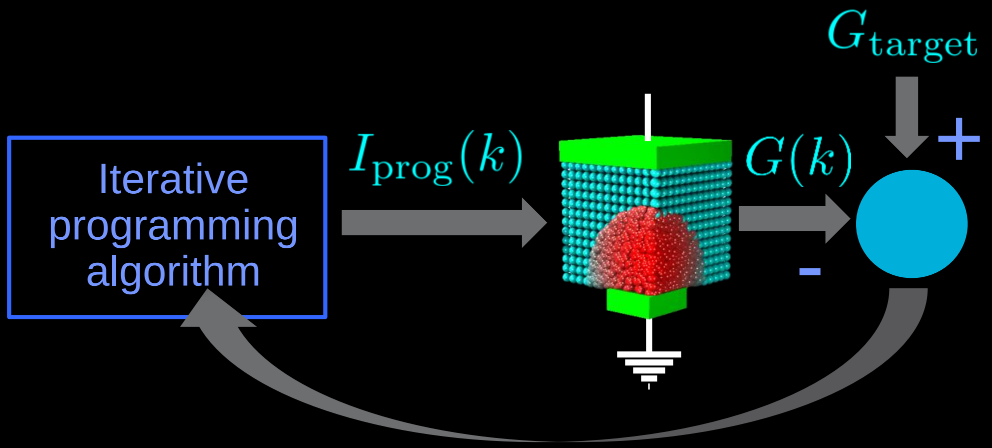


Burr *et al*., *Adv. Phys:* X, 2017

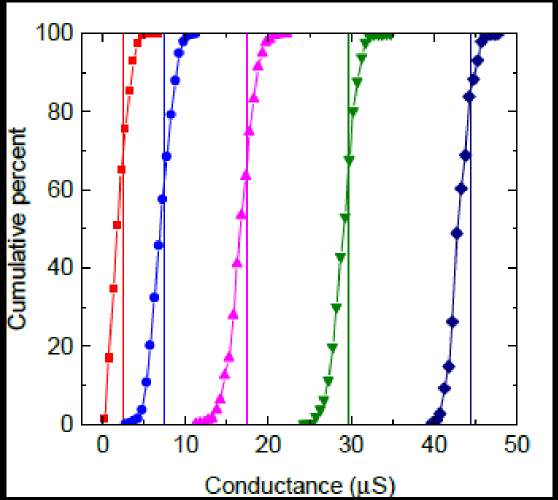
Zidan *et al*., *Nature Electronics*, 2018

- By arranging the memristive devices in a cross-bar configuration, one can perform matrix-vector operation with O(1) complexity
- Exploits multi-level storage capability and Kirchhoff's circuits laws
- Can also implement multiplication with the matrix transpose

Storing a matrix element in a PCM device

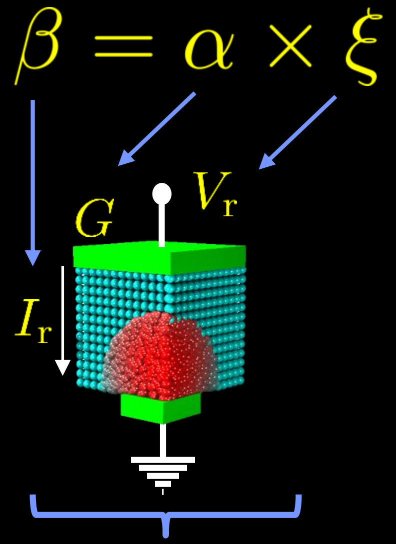


Distribution of conductance values in a large array



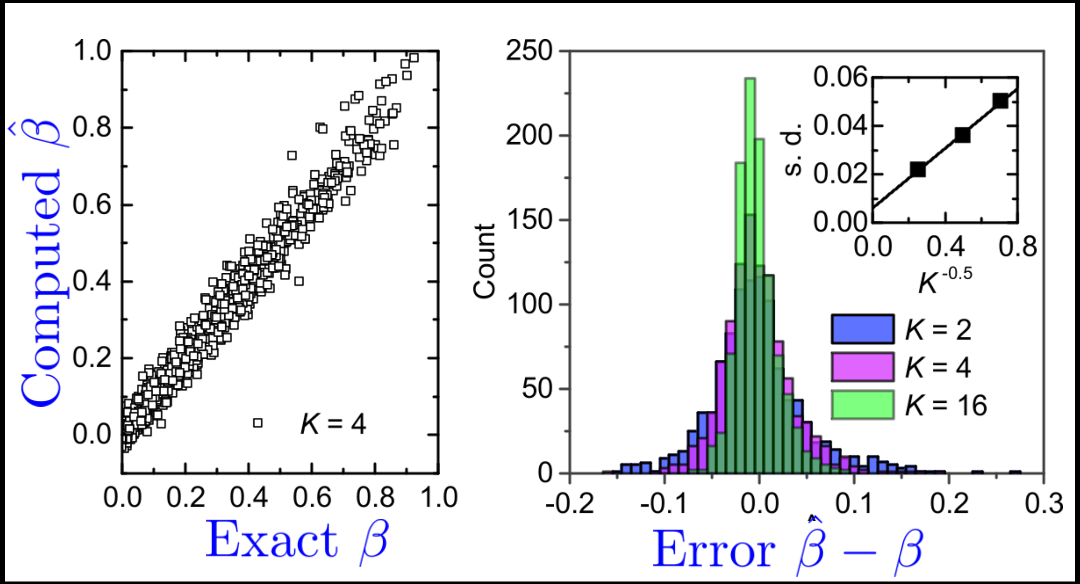
- An iterative programming scheme is typically used to store the matrix elements in a PCM device

Scalar multiplication using PCM devices



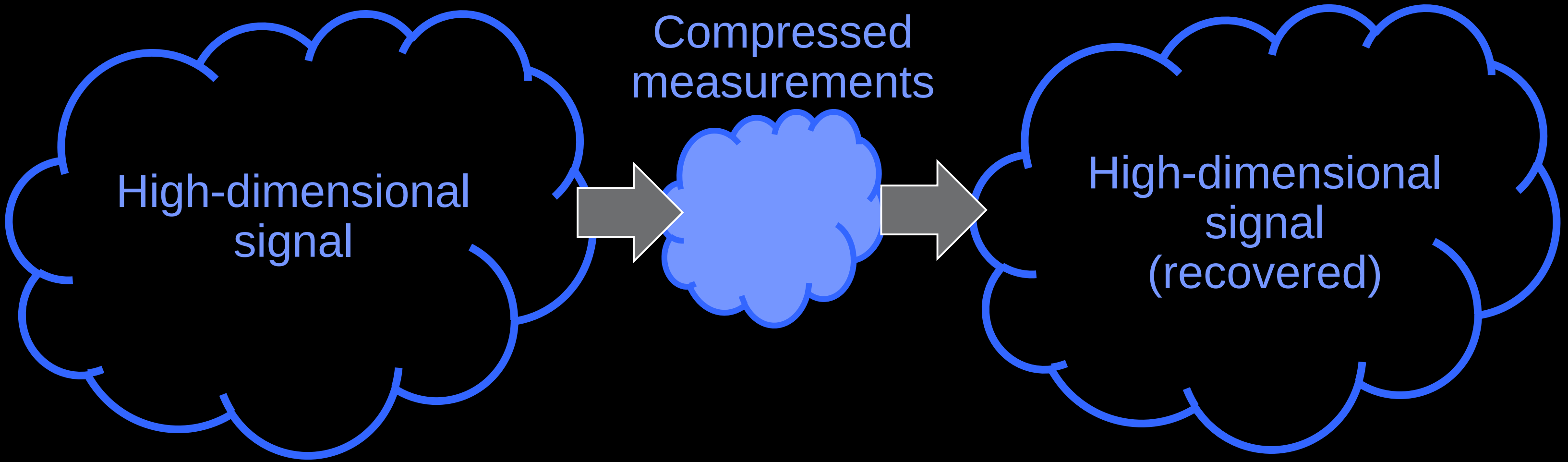
Averaged on K devices

1024 combinations of α and ξ



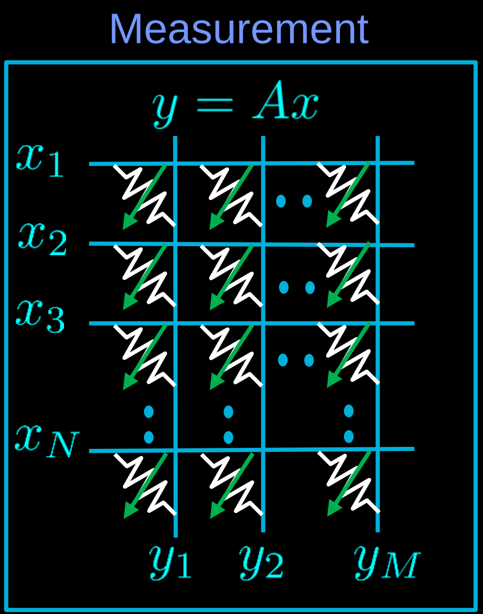
- Experimental characterization of scalar multiplication based on Ohm's law

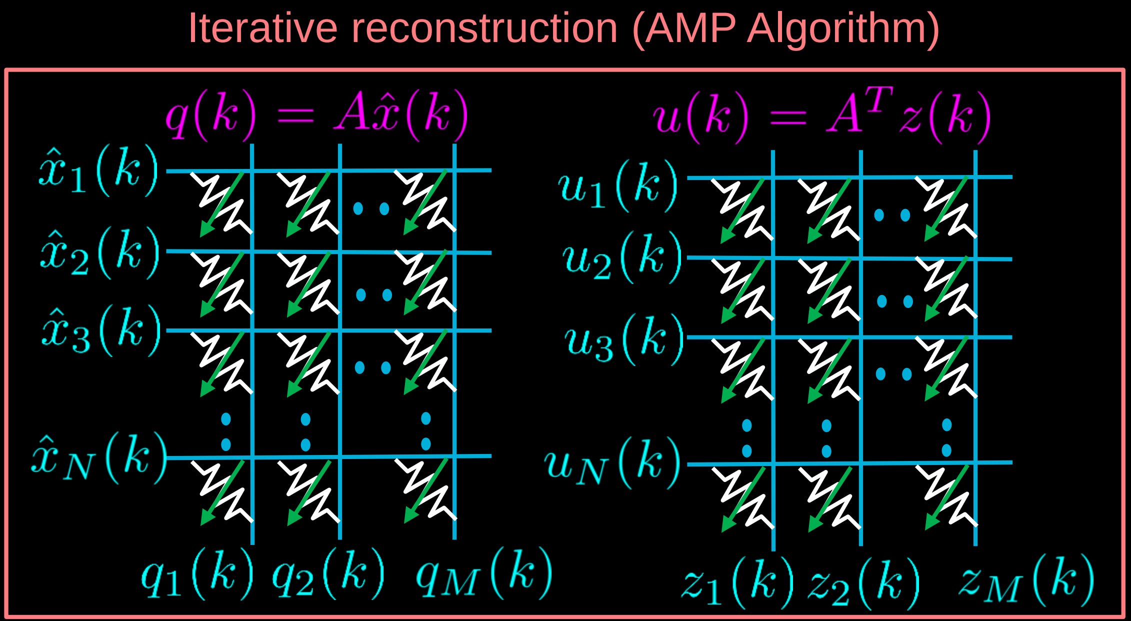
Application: Compressed sensing and recovery



- Compressed sensing: Acquire a large signal at sub-Nyquist sampling rates and subsequently reconstruct that signal accurately
- Sampling and compression done simultaneously
- Used in various applications such as MRI, facial recognition, holography, audio restoration or in mobile-phone camera sensors (allows significant reduction in the acquisition energy per image)

Compressed sensing using computational memory

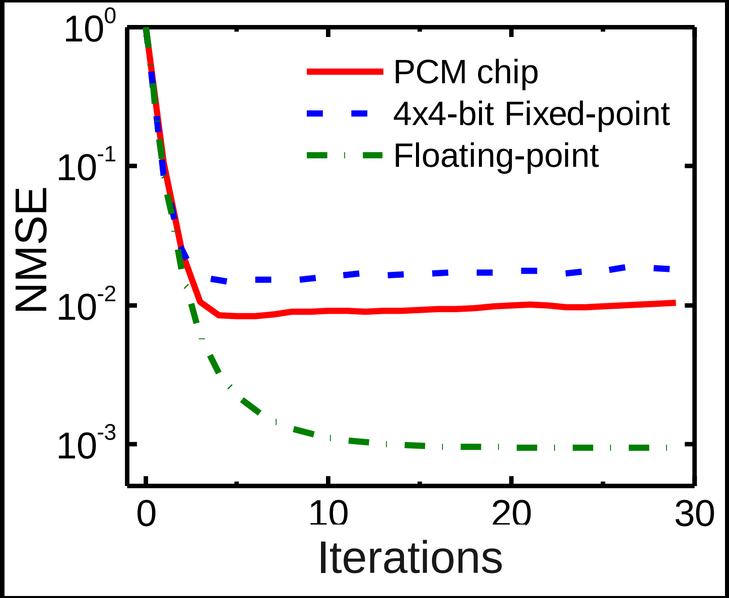




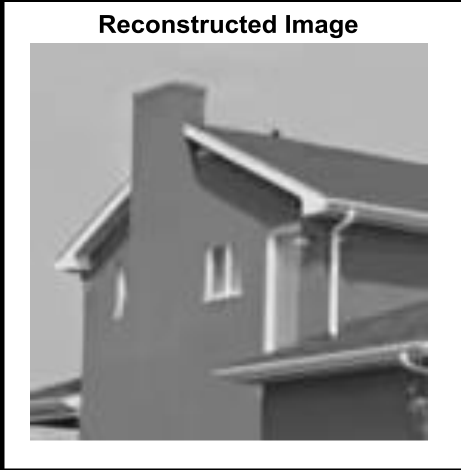
- Store the measurement matrix in a cross-bar array of resistive memory devices
- The same array used for both compression and reconstruction
- Reconstruction complexity reduction: O(NM) → O(N)

Le Gallo *et al.*, *Proc. IEDM*, 2017

Compressive imaging: Experimental results



Experimental result: 128X128 image, 50% sampling rate,
Computation memory unit with 131,072 PCM devices

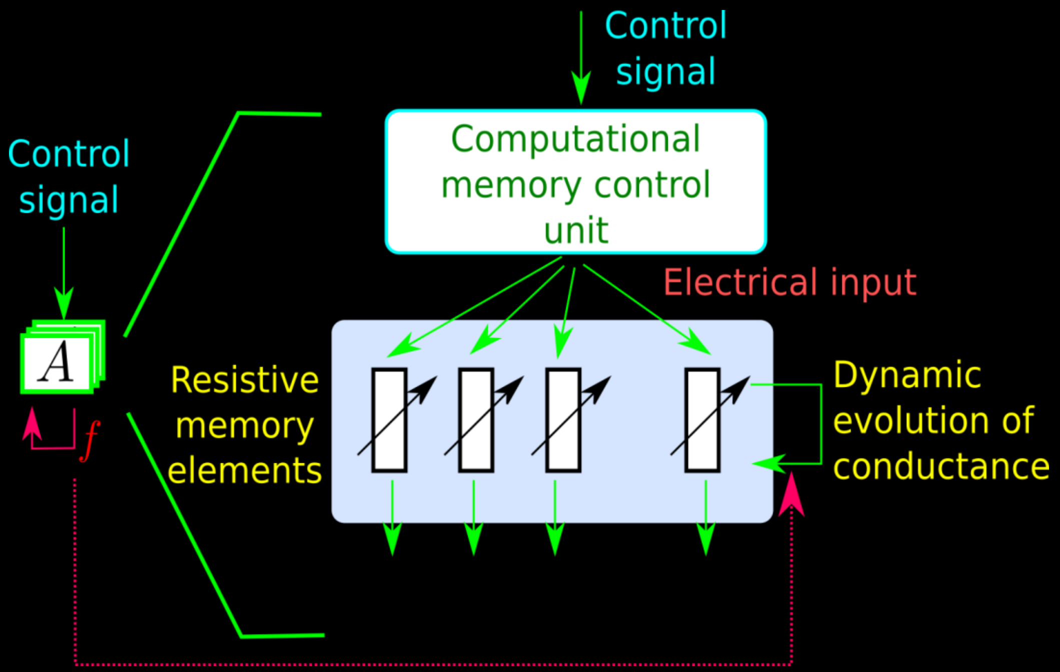


- Reasonable reconstruction accuracy achieved despite inaccuracies
- Estimated power reduction of 50x compared to using an optimized 4-bit FPGA matrix-vector multiplier that delivers same reconstruction accuracy at same speed

Outline

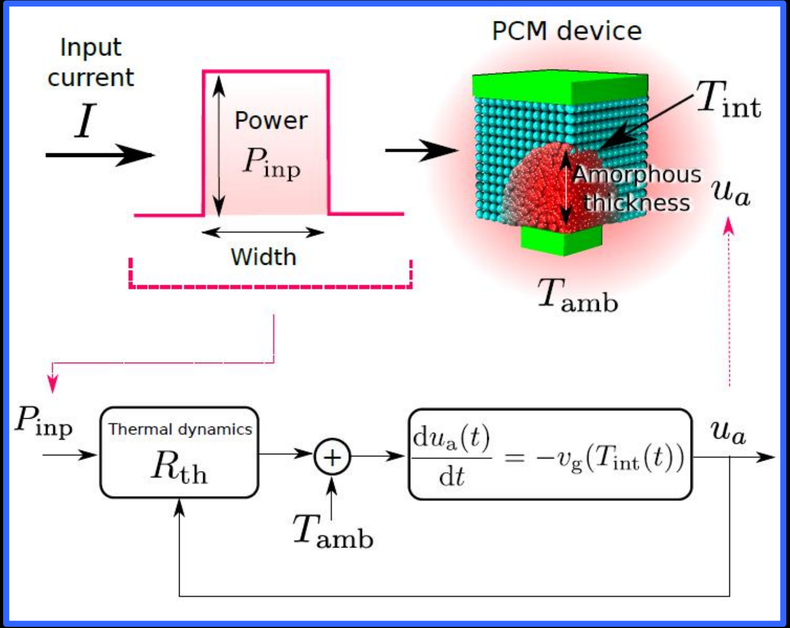
- Motivation for in-memory computing
- Constituent elements of computational memory
- Computational memory: Logical operations
- Computational memory: Arithmetic operations
- Computational memory: Computing with device dynamics
- Mixed-precision in-memory computing
- Summary & Outlook

Can we compute with device dynamics?

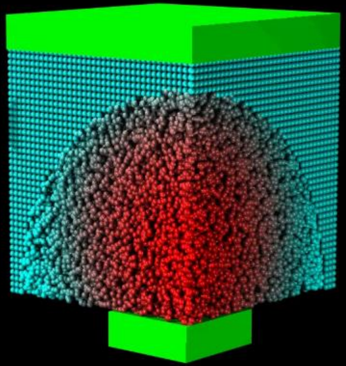


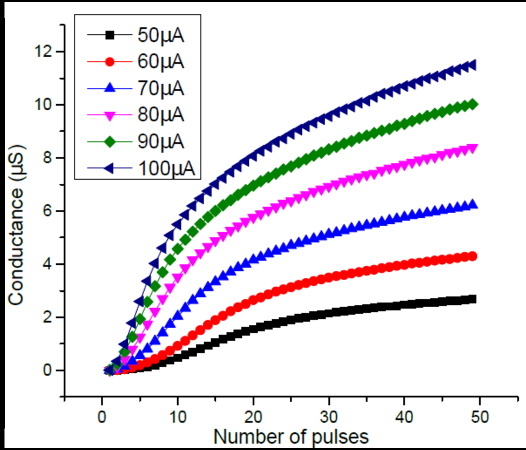
- Depending on the operation, a suitable electrical signal is applied
- The conductance of the devices evolves in accordance with the electrical input
- The result of the operation is imprinted in the memory devices

Crystallization dynamics in PCM



A nanoscale non-volatile integrator
"Accumulative behavior"

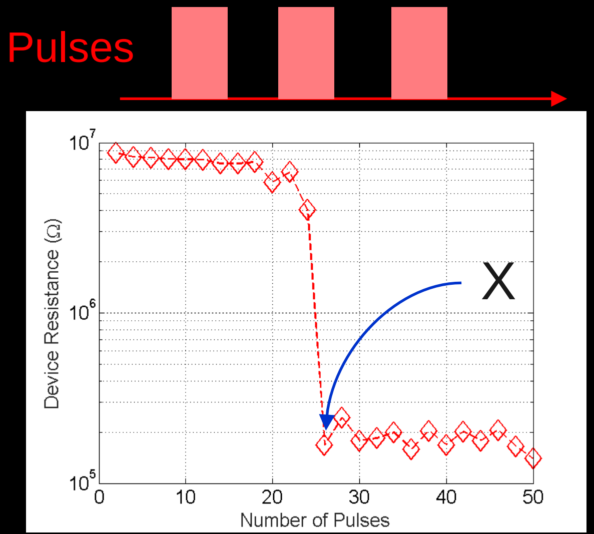


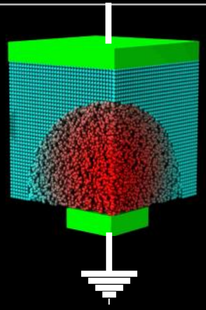


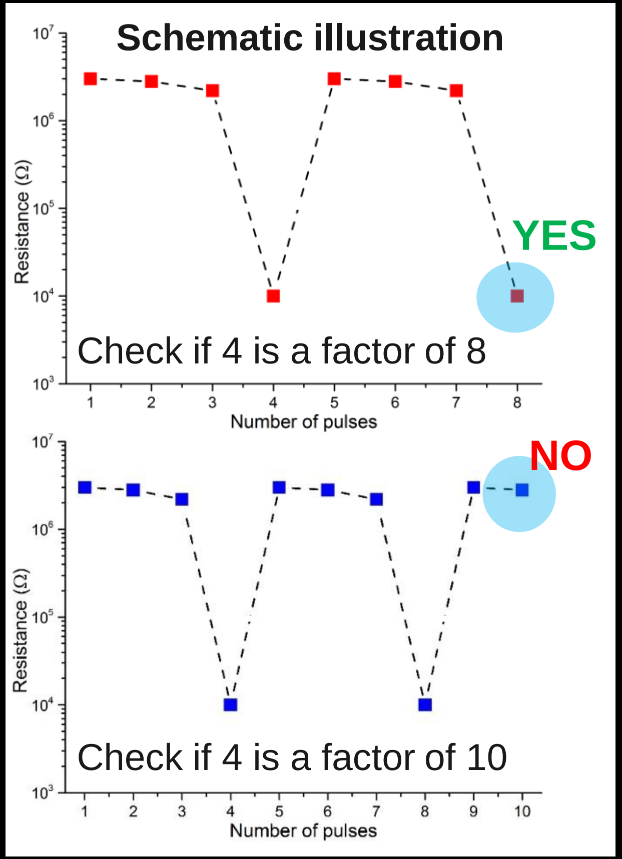
Sebastian *et al.*, *Nature Communications*, 2014

- With successive application of current pulses, we get progressive crystallization
- Higher amplitude → More crystallization and high conductance

Example 1: Finding the factors of numbers



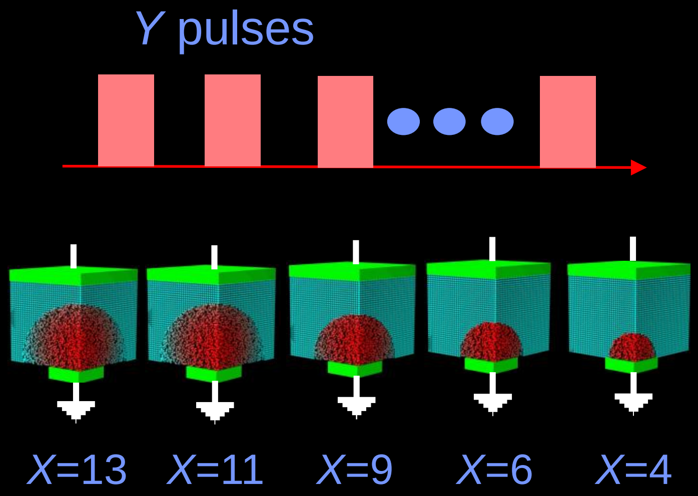




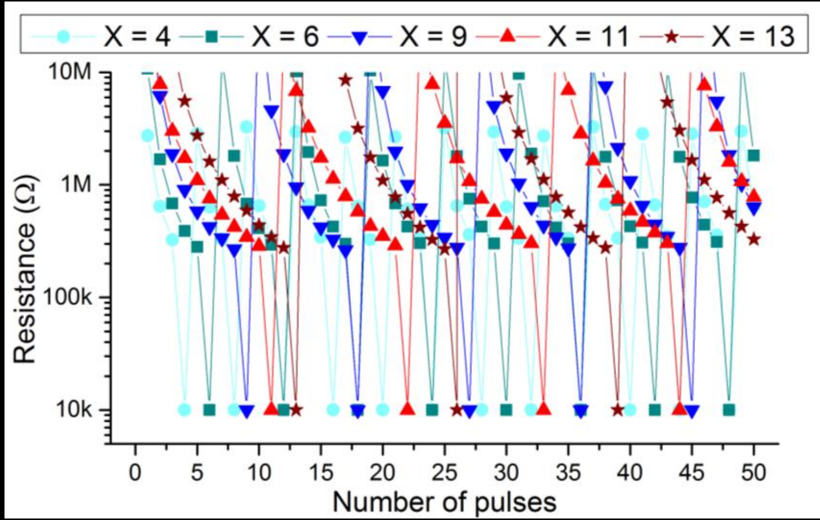
- Assume that a PCM device goes to a low resistance state by the application of *X* number of pulses
- To check if *X* is a factor of *Y*, apply *Y* number of pulses and check if the device is in the low resistance state after the application of the pulses

Hosseini *et al.*, *EDL*, 2017

Finding the factors of numbers in parallel

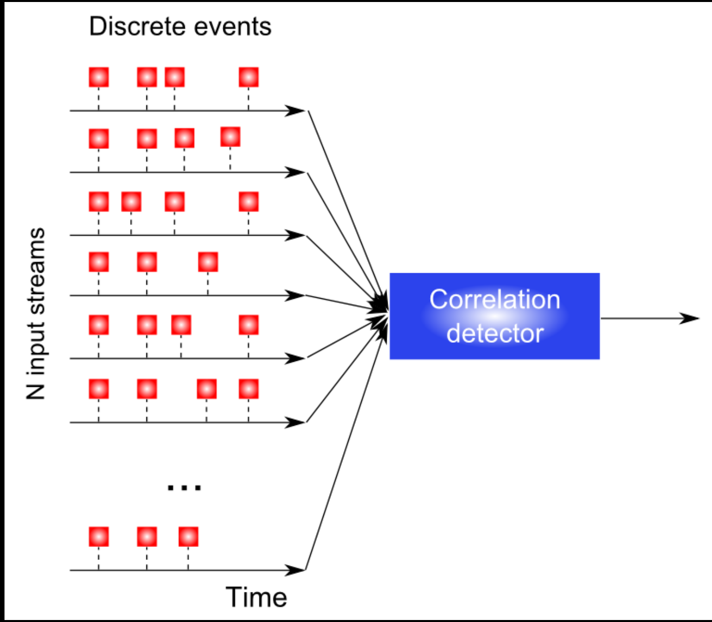


Experimental results



- Can perform this operation to find factors of a number in parallel
- Simple demonstration of the ability to perform higher-level computational primitives
- Multiple devices needed to increase the accuracy

Example 2: Unsupervised learning of correlations



Algorithmic goals

- Find temporal correlations between event-based data streams in an unsupervised manner
- Gain selectivity specifically to the correlated inputs
- Observe variations in the activity of the correlated input
- Quickly react to occurrence of coincident inputs in the correlated inputs
- Continuously and dynamically re-evaluate the learned statistics

Use only unsupervised learning & consume very low power

FINANCE

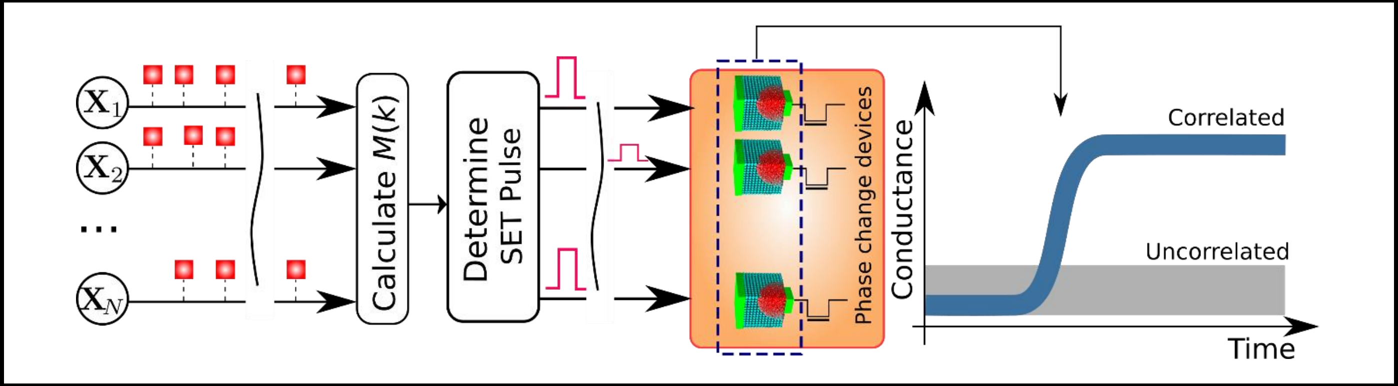
SCIENCE

MEDICINE

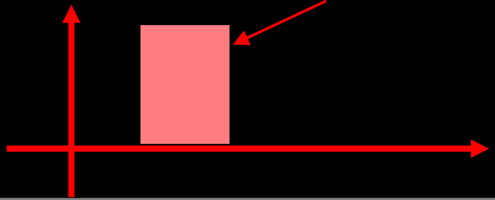
BIG DATA

...AND MORE

Realization using computational memory



Modulate the amplitude or width based on $M(k) = \sum_{j=1}^{N} X_j(k)$

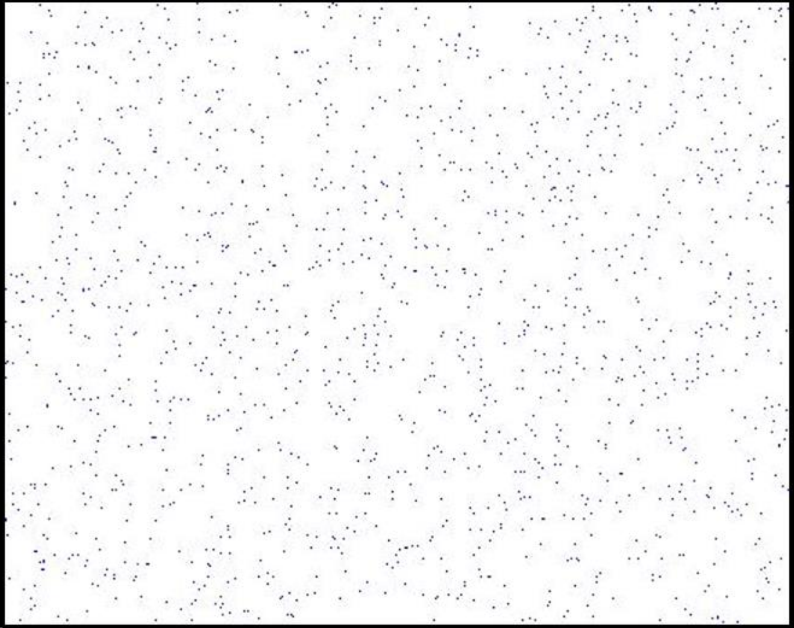


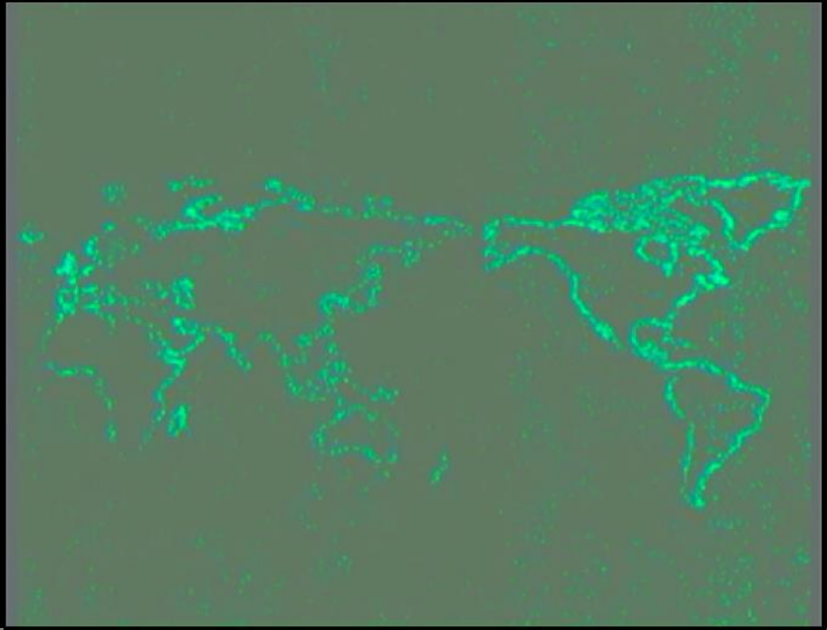
- Devices interfaced to the correlated processes go to a high conductance state

$$
\begin{aligned}
\Delta u_{a_i}(K) &= \sum_{k=1}^{K} \delta u_{a_i}(k) X_i(k) \\
&= C\mathcal{G} \sum_{k=1}^{K} \sum_{j=1}^{N} X_i(k) X_j(k) \\
&= C\mathcal{G} \sum_{j=1}^{N} \sum_{k=1}^{K} X_i(k) X_j(k) \\
&= KC\mathcal{G} \sum_{j=1}^{N} \hat{R}_{ij} \\
&= KC\mathcal{G} \hat{W}_i .
\end{aligned}
$$

Sebastian *et al.*, *Nature Communications*, 2017

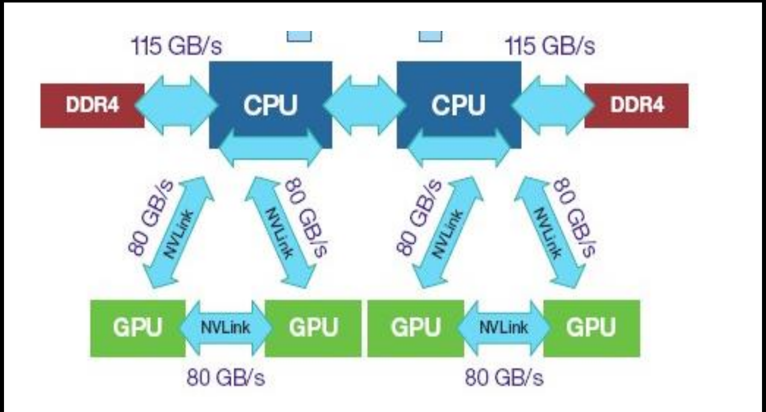
Experimental results (1 Million PCM devices)

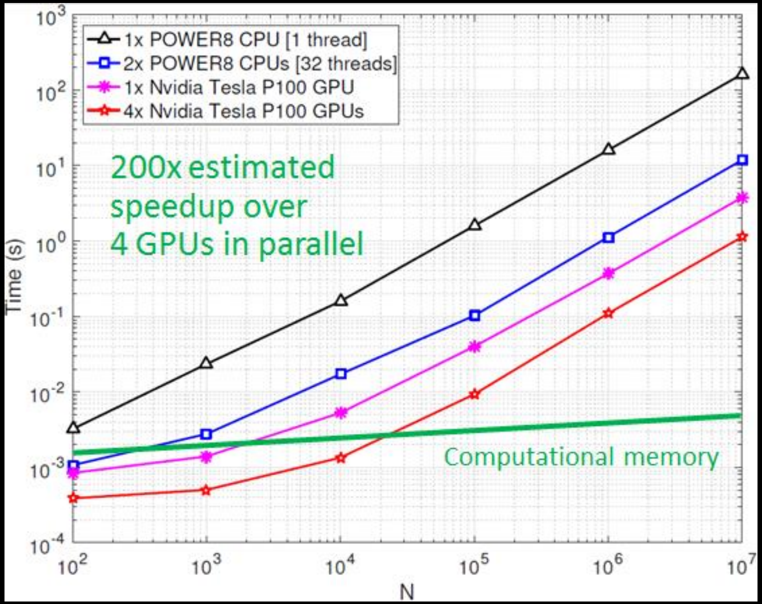




- A million pixels representing a million binary random processes
- The million processes assigned to a million PCM devices in a PCM chip
- The PCM devices interfaced to the correlated processes go to a high conductance state
- Result of the computation imprinted on the devices!

Comparative study



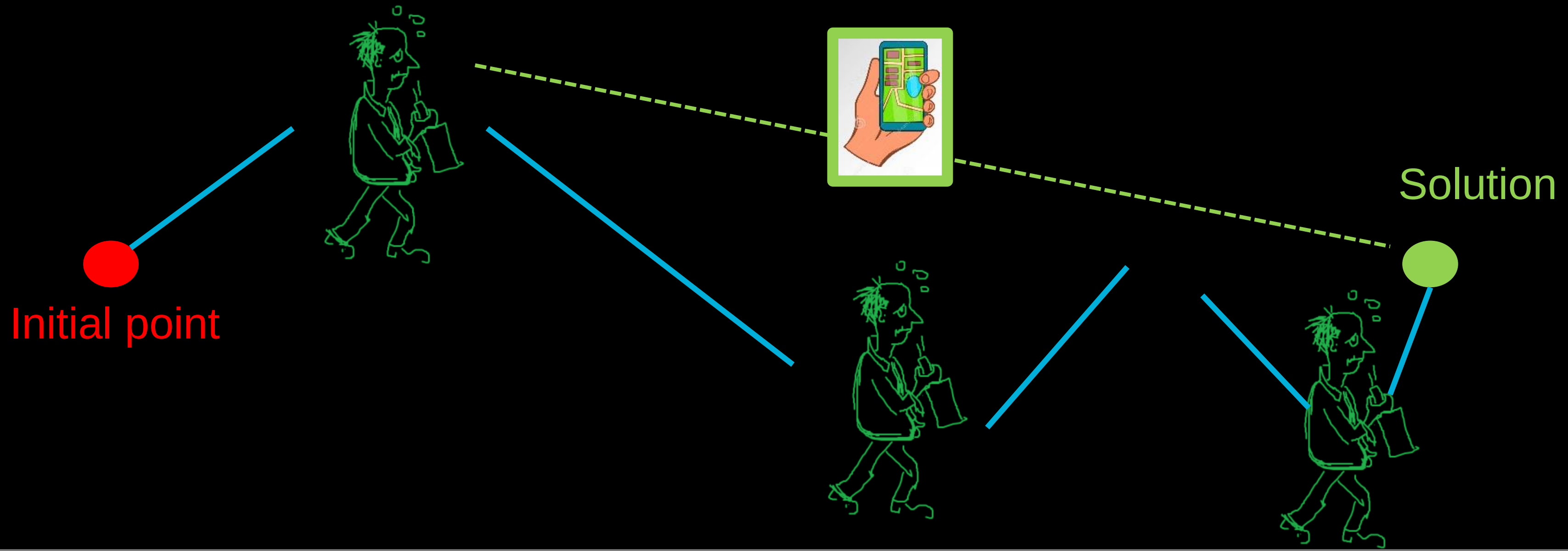


- We expect a 200x improvement in computation time!
- Peak dynamic power on the order of watts compared to hundreds of Watts

Outline

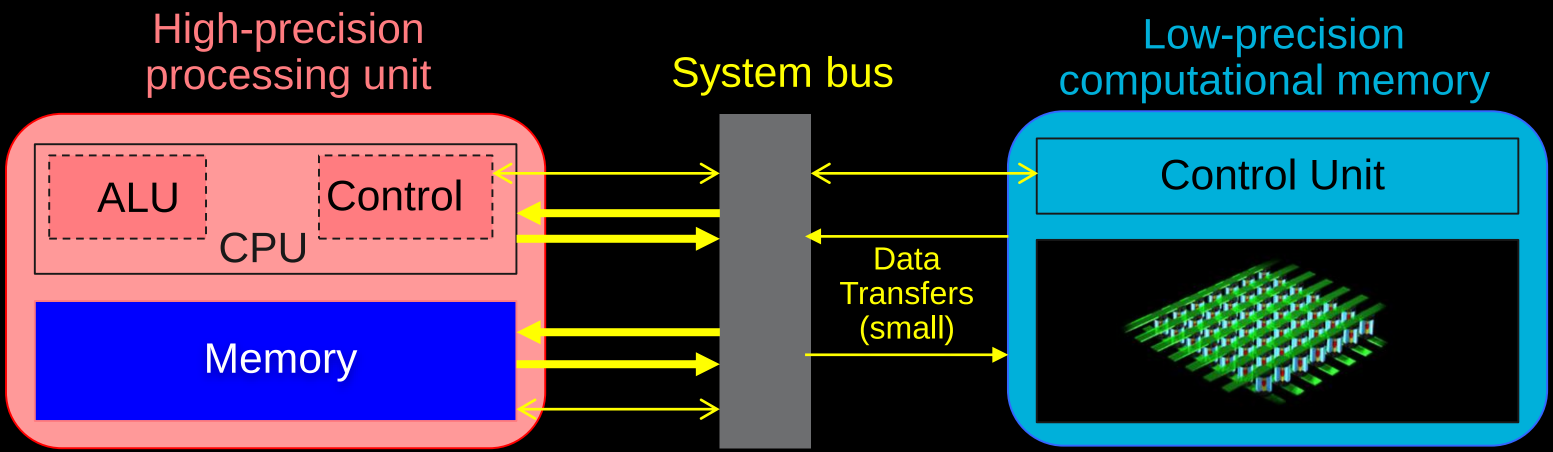
- Motivation for in-memory computing
- Constituent elements of computational memory
- Computational memory: Logical operations
- Computational memory: Arithmetic operations
- Computational memory: Computing with device dynamics
- Mixed-precision in-memory computing
- Summary & Outlook

The challenge of imprecision!



- Many computational tasks can be formulated as a sequence of low- and high-precision components
  - ✓ Step 1: An approximate solution is obtained (high computational load)
  - ✓ Step 2: Resulting error in the overall objective is calculated accurately (low comp. load)
  - ✓ The approximate solution is adapted (repeating step 1)

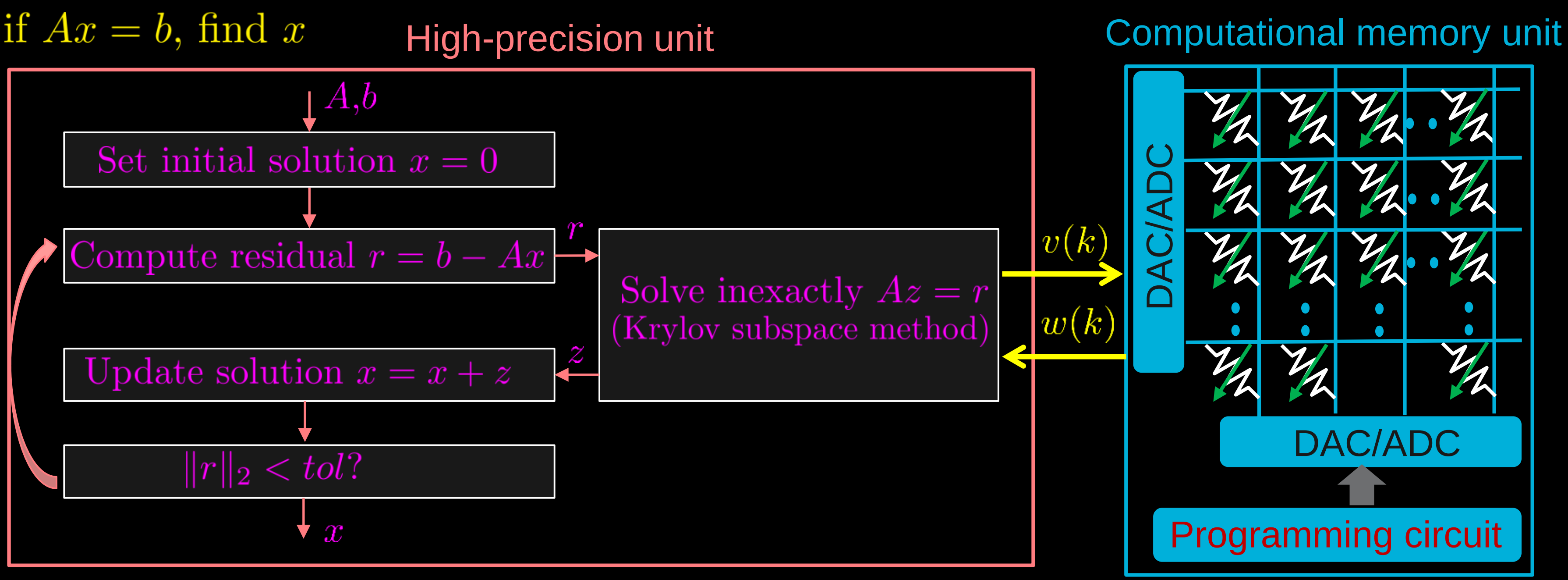
Mixed-precision in-memory computing



Le Gallo *et al*., "Mixed-precision in-memory computing", ArXiv, 2017

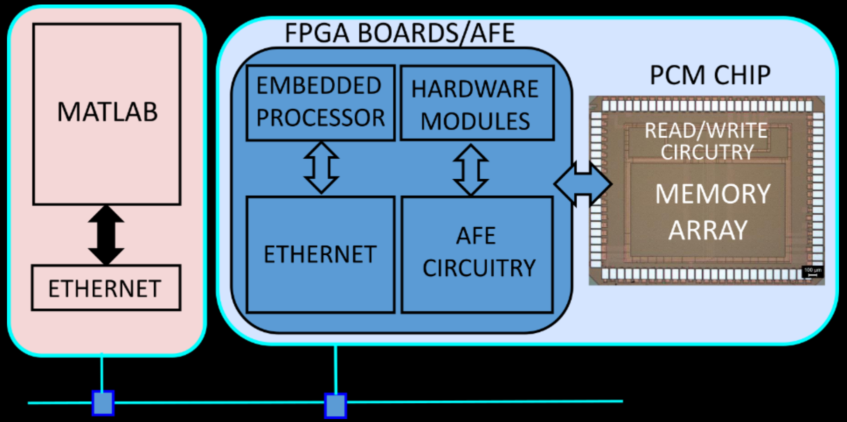
- Use a low precision computational memory unit to obtain the approximate solution
- A von Neumann machine to calculate the error precisely
- Bulk of the computation still realized in computational memory
- Significant areal/power/speed improvements retained while addressing the key challenge of inexactness associated with computational memory

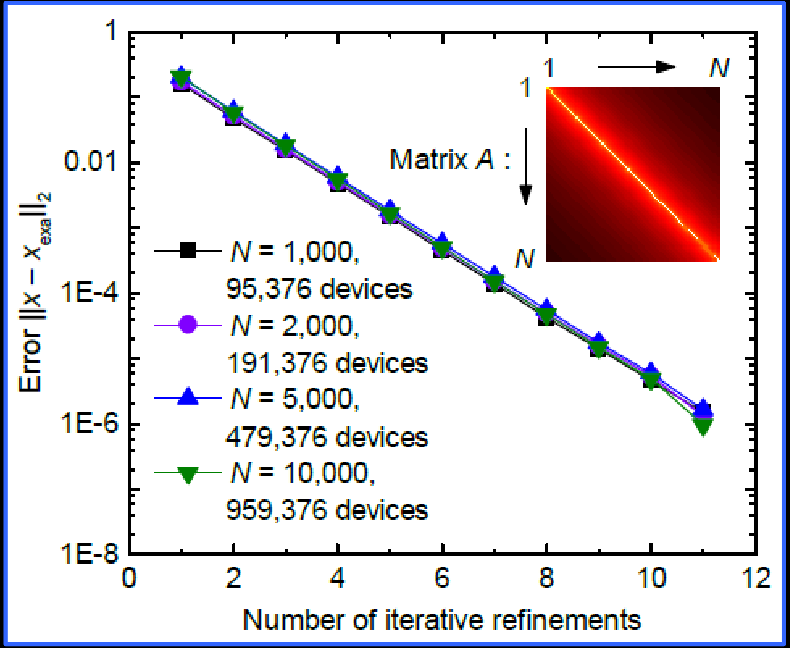
Application 1: Mixed-precision linear solver



- Solution iteratively updated with low-precision error-correction terms
- Correction terms are obtained using an inexact inner solver
- The matrix multiplications in the inner solver are performed using computational memory

Mixed-precision linear solver: Experimental results

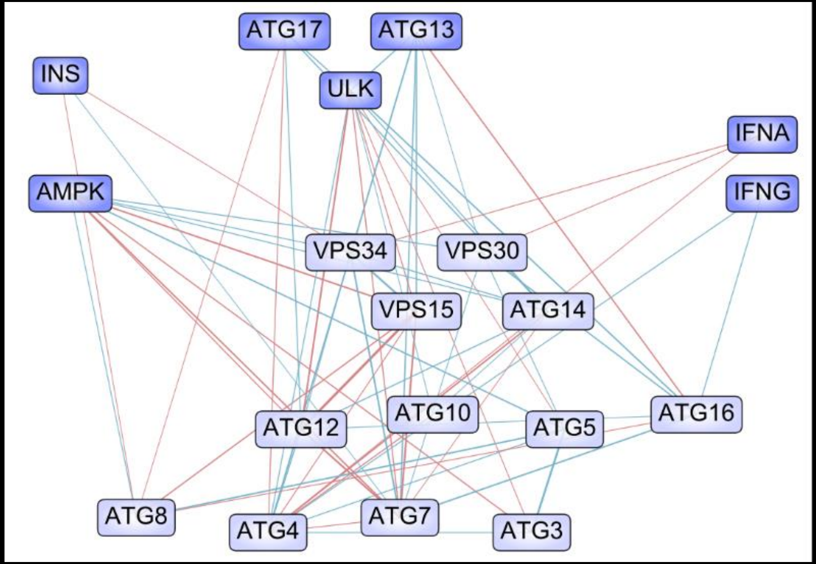




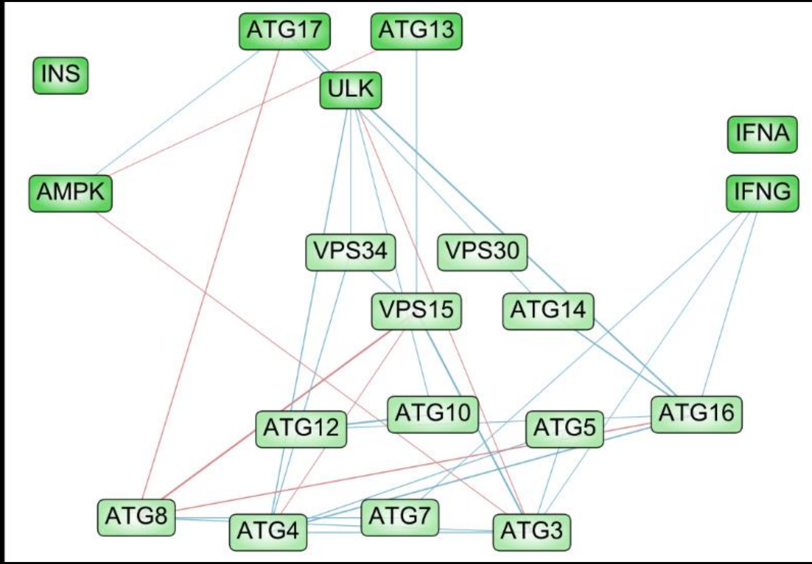
- Experimental results using model covariance matrices of different sizes
- The matrix multiplications in the inner solver are performed using PCM devices (90 nm)
- High-precision iterative refinement ensures that the accuracy is not limited by the precision of the computational memory unit

Application to gene interaction networks

Normal tissue



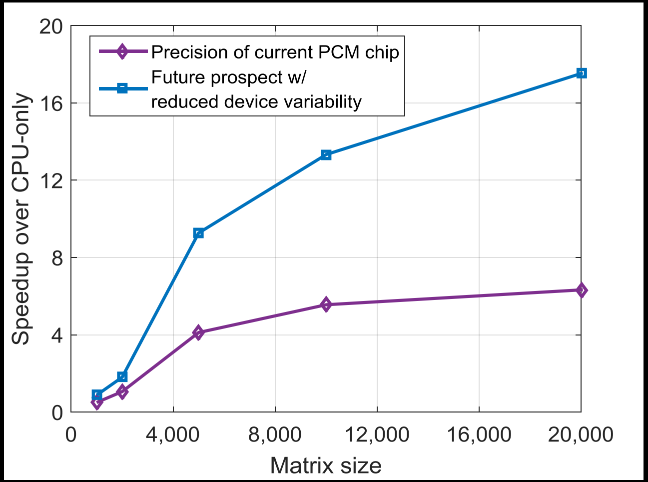
Cancer tissue

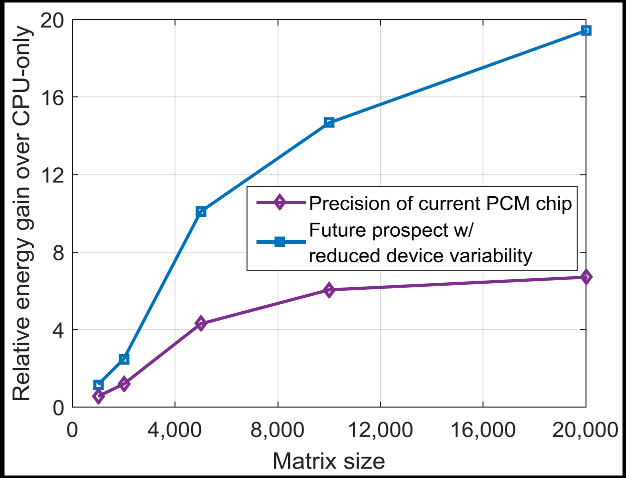


- Gene interaction network (interactome) from RNA expression measurements
- The inverse covariance from RNA measurements of 946 tumor cells and 946 normal cells calculated with mixed-precision in-memory computing

Comparative study

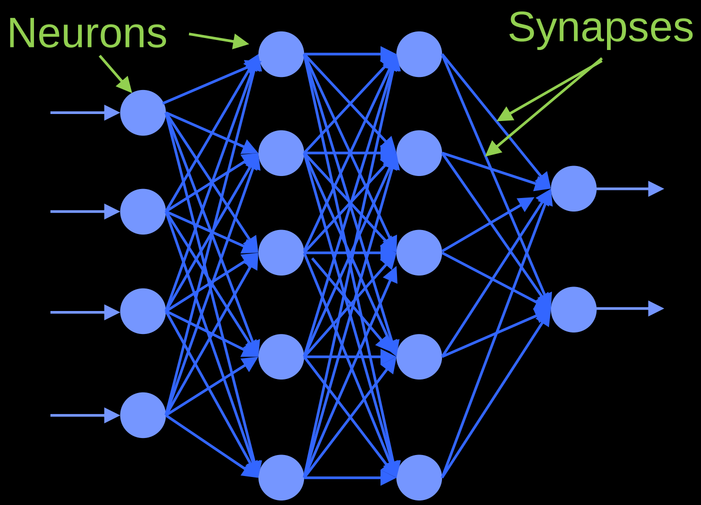
System-level measurements: POWER8 CPU as high-precision processing unit, simulated in-memory computing unit

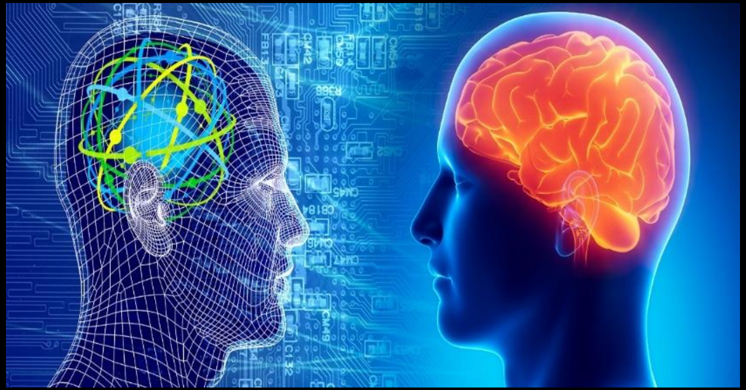




- Significant improvement in time/energy to solution predicted for large matrices over CPU-only and GPU-only implementations
- More accurate in-memory computing → Higher gain in performance

Application 2: Training deep neural networks

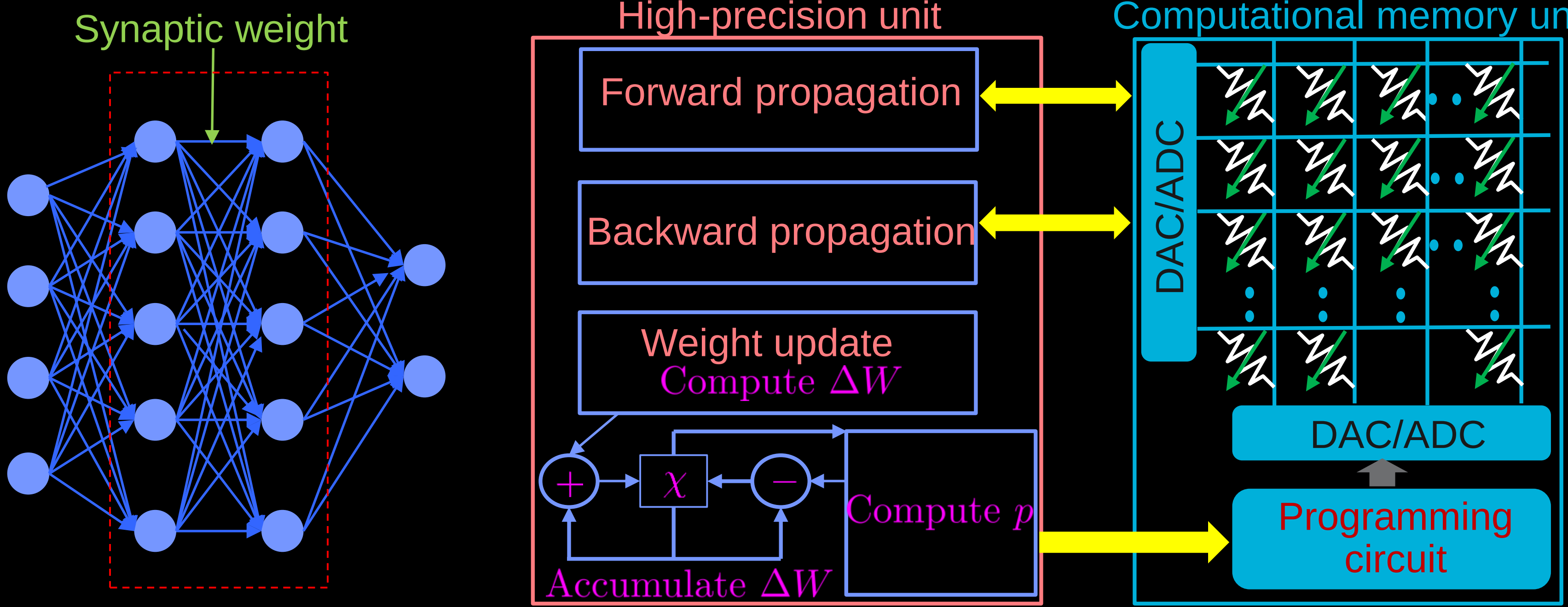




- Multiple layers of parallel processing units (neurons) interconnected by plastic synapses
- By tuning the synaptic weights (training), able to solve certain classification tasks remarkably well
- Training based on a global supervised learning algorithm → Backpropagation
- Brute force optimization: Multiple days or weeks to train state-of-the-art networks on von Neumann machines (CPU,GPU clusters)

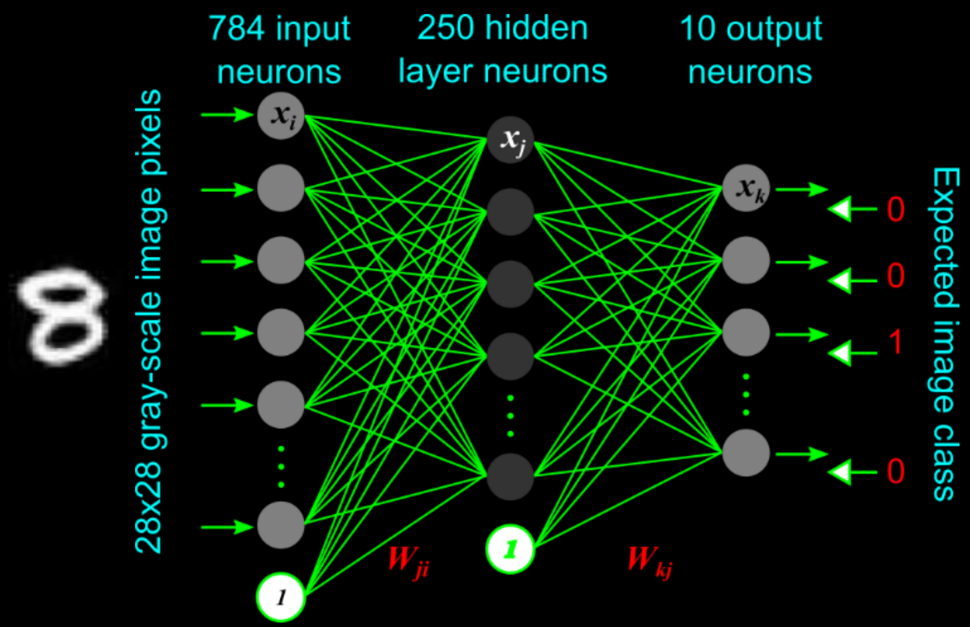
LeCun, Bengio, Hinton, *Nature*, 2015

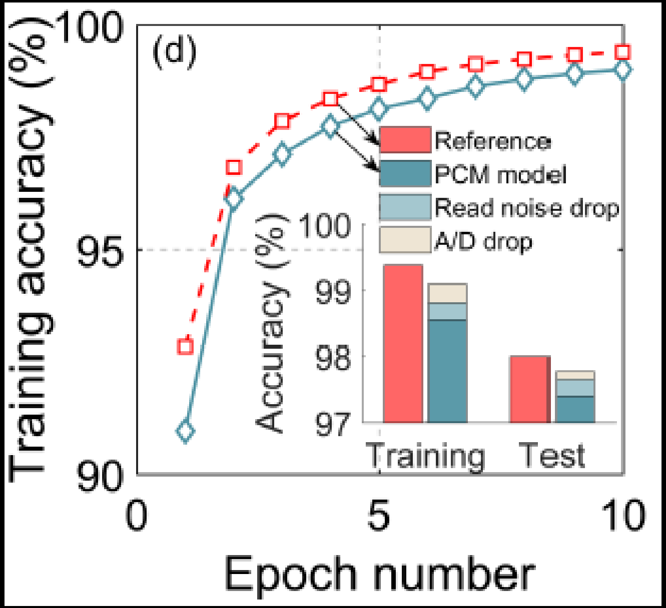
Mixed-precision deep learning



- Synaptic weights always reside in the computational memory
- Forward/backward propagation performed in place (with low precision)
- The desired weight updates accumulated in high precision
- Programming pulses issued to the memory devices to alter the synaptic weights

Results





- MNIST handwritten digit classification problem
- Two PCM devices in differential configurations to represent a synapse
- Device-model-based network simulation achieves 97.78% test accuracy

Nandakumar *et al.*, arXiv:1712.01192, 2017

Summary

- Immense computing challenge associated with the explosive growth of data-centric AI applications
- Computational memory: A memory unit that performs certain computational tasks in place
- Resistive memory devices are considered to play a key role in computational memory
- Computational memory: Logical operations
  - Resistance as a logic state variable enables seamless integration of processing and storage
- Computational memory: Arithmetic operations
  - Matrix-vector multiplications can be performed with O(1) complexity
  - Wide range of applications in optimization problems such as compressed sensing and recovery
- Computational memory: Computing with device dynamics
  - The accumulative behavior exhibited by certain memory devices can be used to perform rather high-level computational tasks such as finding factors of numbers in parallel and unsupervised learning of temporal correlations
- Mixed-precision in-memory computing
  - A significant step towards tackling the imprecision associated with computational memory
  - Applications include solving systems of linear equations and training deep neural networks

Outlook: The evolution of our computing systems

STORAGE
(e.g. Flash, HDD)
(nonvolatile, slow)

STORAGE-CLASS MEMORY

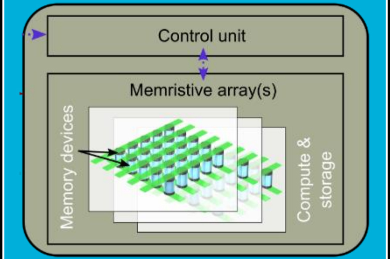
CMOS processing units

MEMORY (e.g. DRAM)
(volatile, fast)

von Neumann accelerators
(e.g. GPUs, ASICs)

High-speed memory

COMPUTATIONAL MEMORY



NEUROMORPHIC CO-PROCESSORS

CENTRAL PROCESSING UNIT

Acknowledgements

- Exploratory memory & cognitive technologies
  - Manuel Le Gallo
  - Irem Boybat
  - Nandakumar SR
  - Iason Giannopoulos
  - Timoleon Moraitis
  - Riduan Khaddam-Aljameh
  - Stanislaw Wozniak
  - Varaprasad Jonnalagadda
  - Angeliki Pantazi
  - Giovanni Cherubini
  - Evangelos Eleftheriou
- Costas Bekas, Foundations of cognitive solutions
- Nikolaos Papandreou, Tom Parnell, Cloud storage and analytics
- Matt Brightsky, IBM T.J. Watson Research Center
- University of Patras, RWTH Aachen, NJIT, Oxford, Exeter, EPFL, ETH

erc
European Research Council
Established by the European Commission

THE FRAMEWORK PROGRAMME FOR RESEARCH AND INNOVATION
HORIZON 2020

FNSNF
FONDS NATIONAL SUISSE
SCHWEIZERISCHER NATIONALFONDS
FONDO NAZIONALE SVIZZERO
SWISS NATIONAL SCIENCE FOUNDATION

References

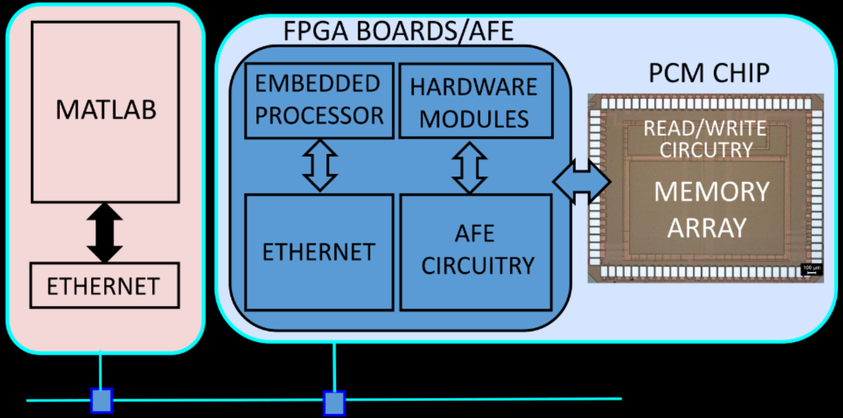
- Ferrucci, D.A., 2012. Introduction to “this is Watson”. *IBM Journal of Research and Development*, *56*(3,4), pp.1-1
- Burr, G.W., Kurdi, B.N., Scott, J.C., Lam, C.H., Gopalakrishnan, K. and Shenoy, R.S., 2008. Overview of candidate device technologies for storage-class memory. *IBM Journal of Research and Development*, *52*(4.5), pp.449-464
- Vermij, E., Hagleitner, C., Fiorin, L., Jongerius, R., van Lunteren, J. and Bertels, K., 2016, May. An architecture for near-data processing systems. In *Proceedings of the ACM International Conference on Computing Frontiers* (pp. 357-360)
- Wong, H.S.P. and Salahuddin, S., 2015. Memory leads the way to better computing. *Nature Nanotechnology*, *10*(3), p.191
- Shulaker, M.M., Hills, G., Park, R.S., Howe, R.T., Saraswat, K., Wong, H.S.P. and Mitra, S., 2017. Three-dimensional integration of nanotechnologies for computing and data storage on a single chip. *Nature*, *547*(7661), p.74.
- Traversa, F.L. and Di Ventra, M., 2015. Universal memcomputing machines. *IEEE transactions on neural networks and learning systems*, *26*(11), pp.2702-2715.
- Seshadri, V., Lee, D., Mullins, T., Hassan, H., Boroumand, A., Kim, J., Kozuch, M.A., Mutlu, O., Gibbons, P.B. and Mowry, T.C., 2016. Buddy-ram: Improving the performance and efficiency of bulk bitwise operations using DRAM. *arXiv preprint arXiv:1611.09988*.
- Bavandpour, M., Mahmoodi, M.R. and Strukov, D.B., 2017. Energy-Efficient Time-Domain Vector-by-Matrix Multiplier for Neurocomputing and Beyond. *arXiv preprint arXiv:1711.10673*.
- Borghetti, J., Snider, G.S., Kuekes, P.J., Yang, J.J., Stewart, D.R. and Williams, R.S., 2010. ‘Memristive’ switches enable ‘stateful’ logic operations via material implication. *Nature*, *464*(7290), p.873
- Vourkas, I. and Sirakoulis, G.C., 2016. Emerging memristor-based logic circuit design approaches: A review. *IEEE Circuits and Systems Magazine*, 16(3), pp.15-30
- Kvatinsky, S., Belousov, D., Liman, S., Satat, G., Wald, N., Friedman, E.G., Kolodny, A. and Weiser, U.C., 2014. MAGIC—Memristor-aided logic. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 61(11), pp.895-899
- Talati, N., Gupta, S., Mane, P. and Kvatinsky, S., 2016. Logic design within memristive memories using memristor-aided loGIC (MAGIC). *IEEE Transactions on Nanotechnology*, *15*(4), pp.635-650.

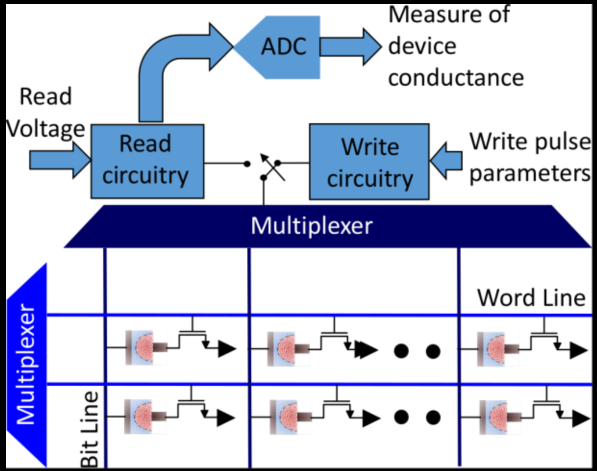
References

- Burr, G.W., Shelby, R.M., Sebastian, A., Kim, S., Kim, S., Sidler, S., Virwani, K., Ishii, M., Narayanan, P., Fumarola, A. and Sanches, L.L., 2017. Neuromorphic computing using non-volatile memory. *Advances in Physics: X*, *2*(1), pp.89-124
- Hu, M., Strachan, J.P., Li, Z., Grafals, E.M., Davila, N., Graves, C., Lam, S., Ge, N., Yang, J.J. and Williams, R.S., 2016, June. Dot-product engine for neuromorphic computing: programming 1T1M crossbar to accelerate matrix-vector multiplication. In *Proceedings of the 53rd annual design automation conference* (p. 19). ACM.
- Zidan, M.A., Strachan, J.P. and Lu, W.D., 2018. The future of electronics based on memristive systems. *Nature Electronics*, *1*(1), p.22.
- Le Gallo, M., Sebastian, A., Cherubini, G., Giefers, H., Eleftheriou, E., 2017. *Proc. International Electron Devices Meeting*
- Hosseini, P., Sebastian, A., Papandreou, N., Wright, C.D. and Bhaskaran, H., 2015. Accumulation-based computing using phase-change memories with FET access devices. *IEEE Electron Device Letters*, *36*(9), pp.975-977
- Sebastian, A., Tuma, T., Papandreou, N., Le Gallo, M., Kull, L., Parnell, T. and Eleftheriou, E., 2017. Temporal correlation detection using computational phase-change memory. *Nature Communications,* 8, article 1115
- Boybat, I., Gallo, M.L., Moraitis, T., Parnell, T., Tuma, T., Rajendran, B., Leblebici, Y., Sebastian, A. and Eleftheriou, E., 2017. Neuromorphic computing with multi-memristive synapses. arXiv preprint arXiv:1711.06507.
- Le Gallo, M., Sebastian, A., Mathis, R., Manica, M., Giefers, H., Tuma, T., Bekas, C., Curioni, A. and Eleftheriou, E., 2017. Mixed-Precision In-Memory Computing. *arXiv preprint arXiv:1701.04279*.
- LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. *Nature*, *521*(7553), p.436.
- Nandakumar, S. R., Le Gallo, M., Boybat, I., Rajendran, B., Sebastian, A. and Eleftheriou, E., 2017. Mixed-precision training of deep neural networks using computational memory. *arXiv preprint arXiv:1712.01192*
- Merolla, P.A., Arthur, J.V., Alvarez-Icaza, R., Cassidy, A.S., Sawada, J., Akopyan, F., Jackson, B.L., Imam, N., Guo, C., Nakamura, Y. and Brezzo, B., 2014. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197), pp.668-673
- Tuma, T., Pantazi, A., Le Gallo, M., Sebastian, A. and Eleftheriou, E., 2016. Stochastic phase-change neurons. *Nature Nanotechnology*, 11(8), p.693

BACK-UP

Experimental platform

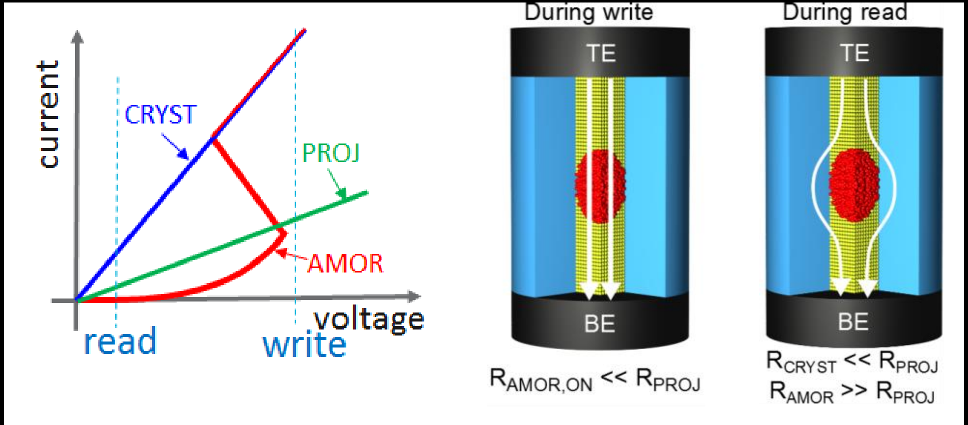


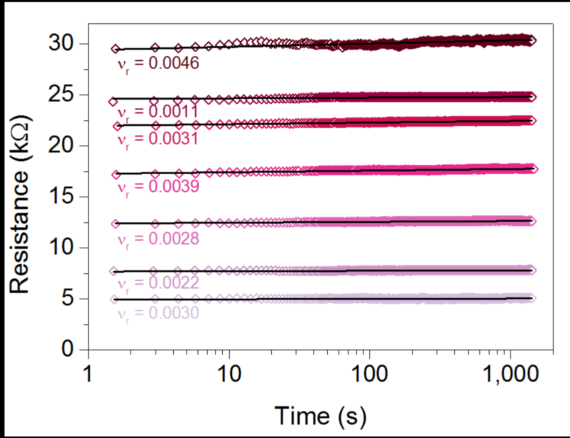


- Experimental platform built around a prototype multi-level PCM chip that comprises 3 million devices
- The PCM chip is organized as a matrix of world lines and bit lines
- It also integrates the associated read/write circuitries

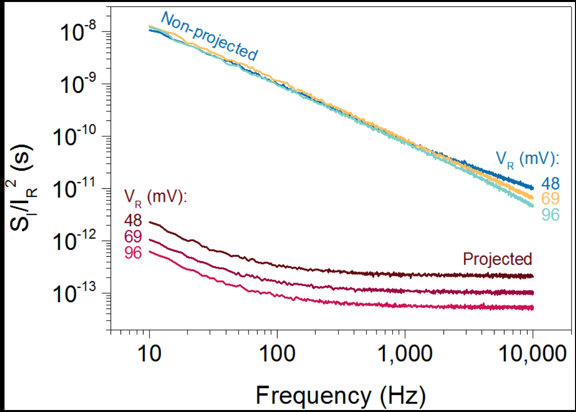
Sebastian *et al.*, *Nature Communications*, 2017

Projected memory



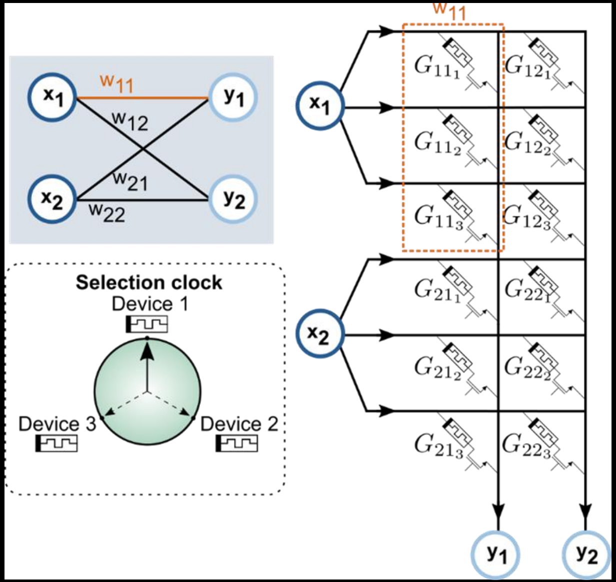


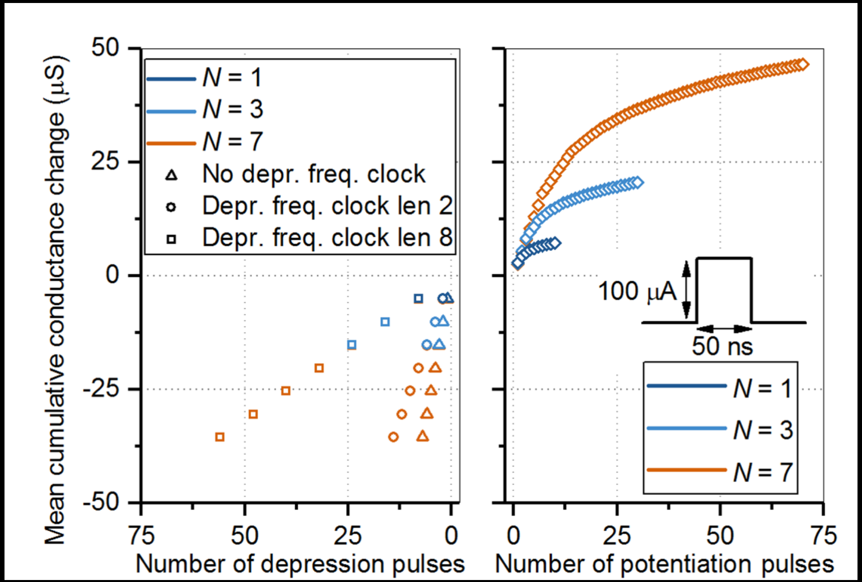
- Carefully designed layer of non-insulating projection segment parallel to the phase-change segment
- Write operation not affected
- During read, the current flows around the amorphous phase
- Significant reduction in noise, drift and drift variability expected



Koelmans *et al.*, *Nature Communications*, 2015

Multi-memristive architectures





- Represent weights/matrix elements using multiple devices
- Only a subset of the devices programmed at any instance, but all devices read in parallel
- A global clock-based arbitration for device selection and to tune the conductance response curve

Boybat *et al*., *arXiv:1711.06507*, 2017