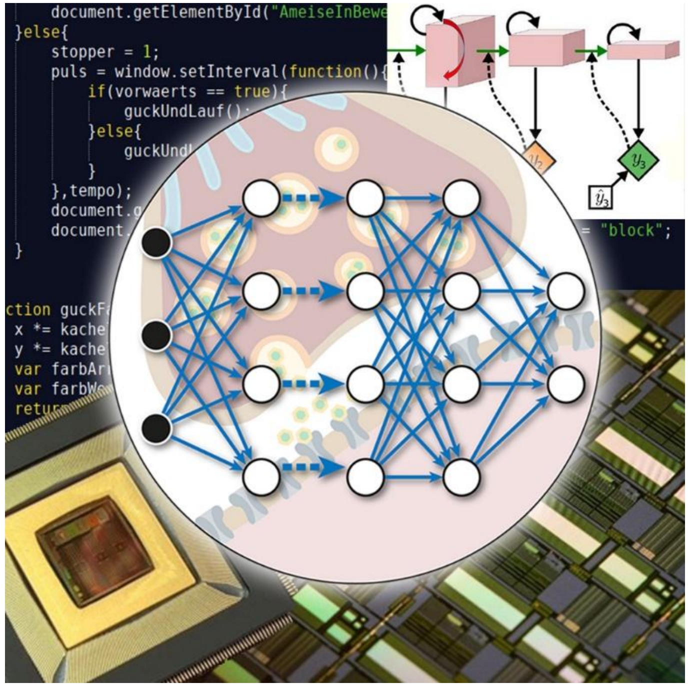


PERSPECTIVE AND CASE STUDIES IN COMPUTING WITH PHYSICAL ANALOG SYSTEMS

JOHN PAUL STRACHAN

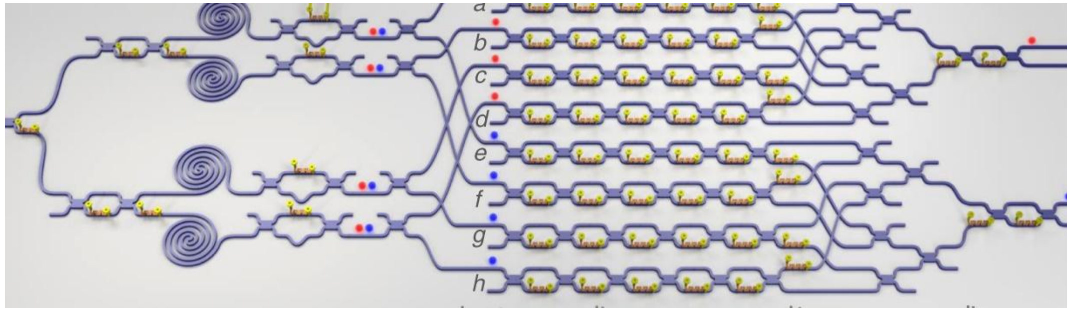
FORSCHUNGSZENTRUM JÜLICH (PGI-14)

RWTH AACHEN UNIVERSITY, GERMANY

Outline

- Introduction and viewpoint for physical computing
- Mixed analog-digital, in-memory computing (with and without errors)
  - Applications in A.I., Optimization
- Expanding our computing primitives – associative memories and complex neurons

Range of physical computing approaches under exploration (Not complete list)

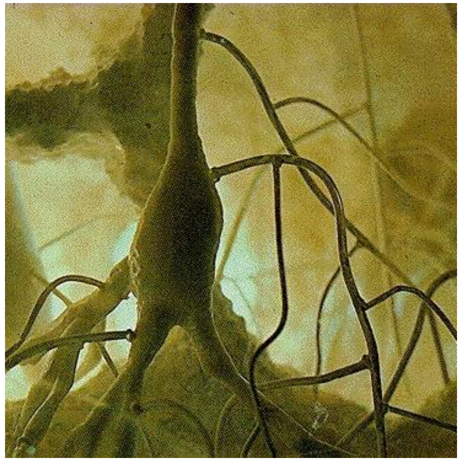


Qiang, X., et al. *Nature Photonics* (2018)

Optical/Photonic

… and more….

Brain-inspired/ Neuromorphic



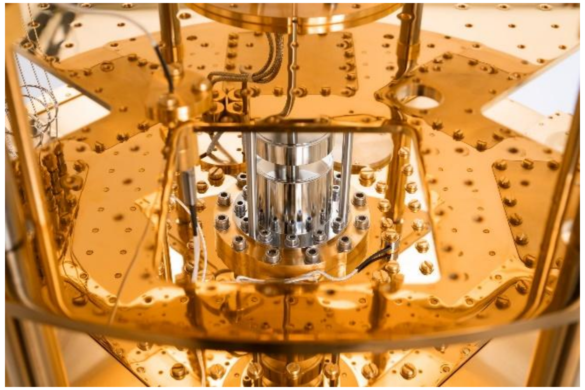
Frederic Vester 1978

Analog

Photo: Steven Fine, *Tide-predicting machine*

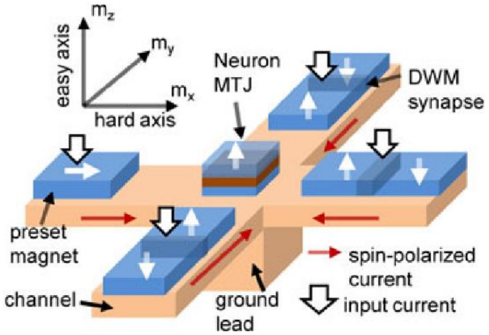
Future Computing paradigm??

Quantum



Forschungszentrum Jülich

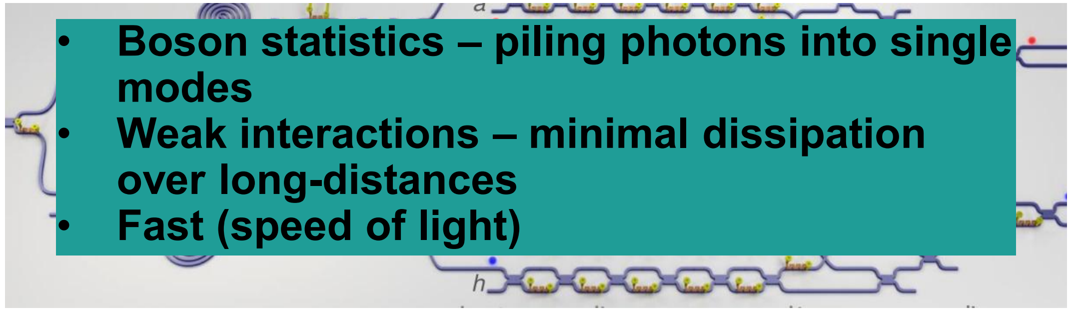
Magnetic/Spintronic



M Sharad, et al. *IEEE Trans Nano* (2012)

Range of physical computing approaches under exploration (Not complete list)

Novel Resources exploited



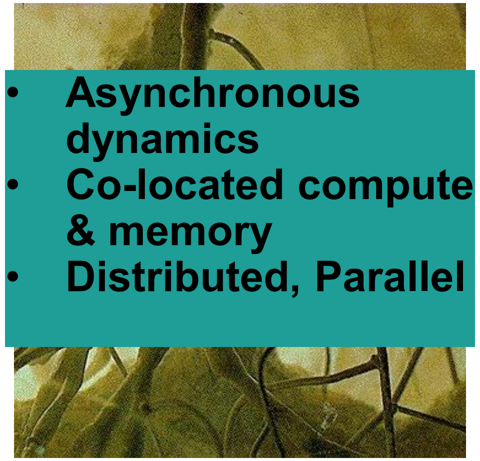
- **Boson statistics – piling photons into single modes**
- **Weak interactions – minimal dissipation over long-distances**
- **Fast (speed of light)**

Qiang, X., et al. *Nature Photonics* (2018)

Optical/Photonic

… and more….

Brain-inspired/ Neuromorphic

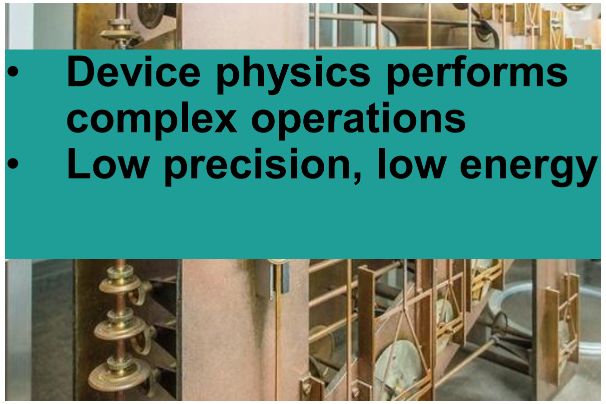


- **Asynchronous dynamics**
- **Co-located compute & memory**
- **Distributed, Parallel**

Frederic Vester 1978

Do I have to choose only one?

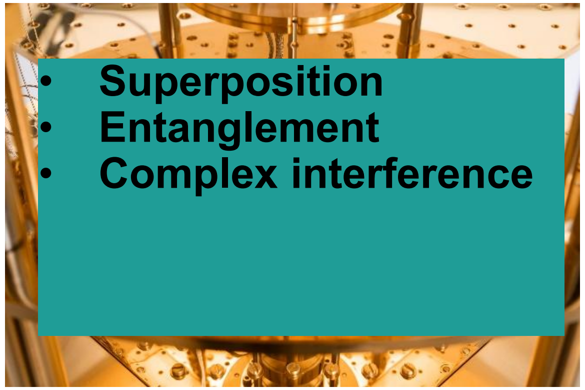
Analog



- **Device physics performs complex operations**
- **Low precision, low energy**

Photo: Steven Fine, *Tide-predicting machine*

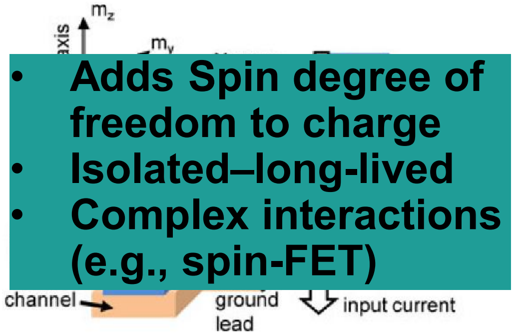
Quantum



- **Superposition**
- **Entanglement**
- **Complex interference**

Forschungszentrum Jülich

Magnetic/Spintronic



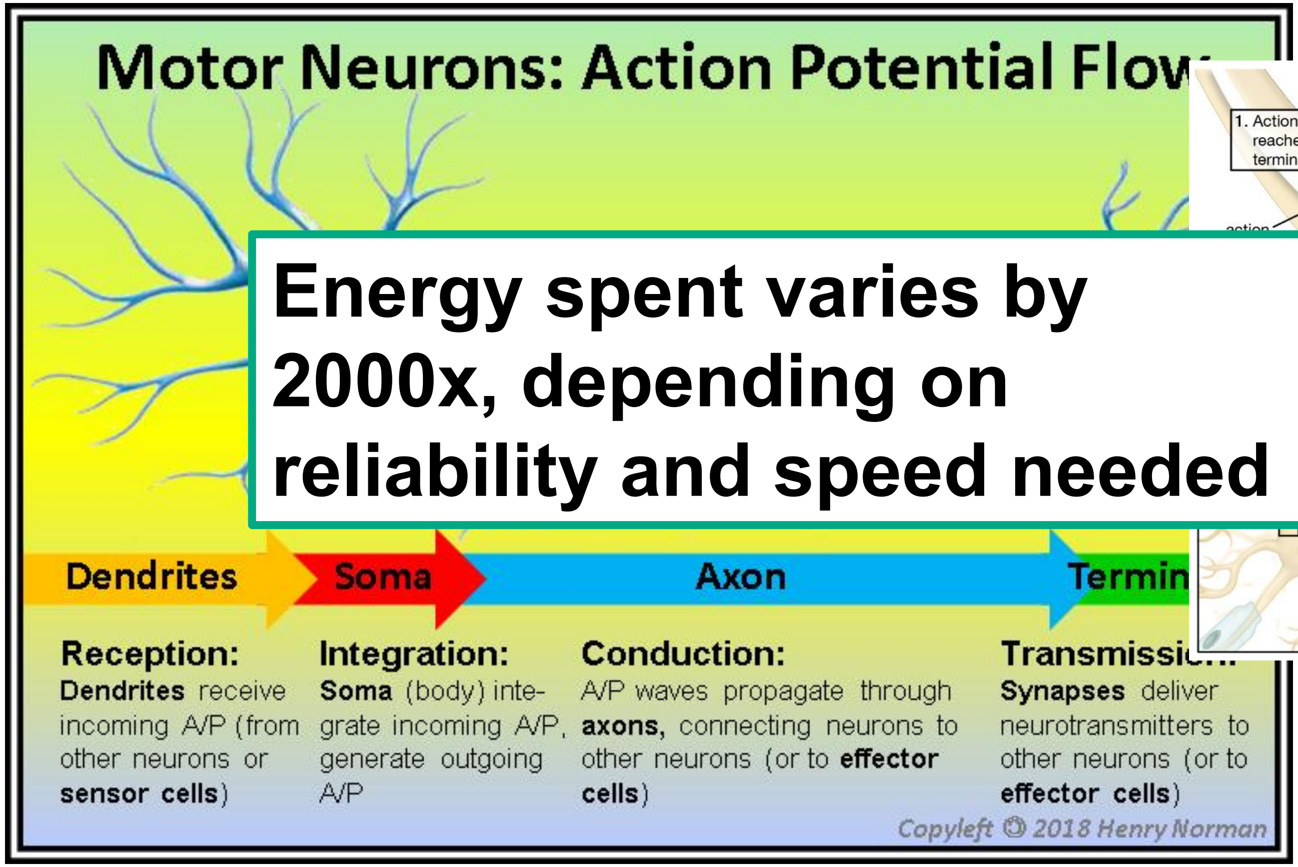
- **Adds Spin degree of freedom to charge**
- **Isolated–long-lived**
- **Complex interactions (e.g., spin-FET)**

M Sharad, et al. *IEEE Trans Nano* (2012)

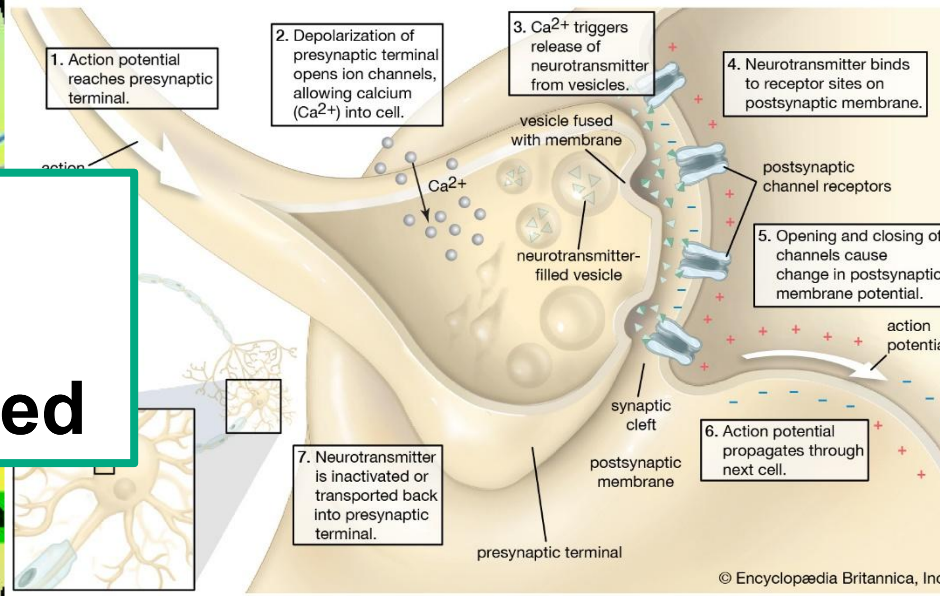
Brains leverage variable physics, time-scales, energy-scales

Example

Numbers from: *Principles of Neural Design*, Sterling and Laughlin (2015), *Chapters 5,6*



Energy spent varies by 2000x, depending on reliability and speed needed



Passive propagation through neuron body as **electrical signal**
Speed: 1 m/s
Energy: ~23 femtoJoules

Active signal propagation across axons as **Ionic currents**
Speed: 100 m/s
Energy: ~6000 femtoJoules
Fast, reliable, higher energy

Neurotransmitters move through **chemical diffusion** from vesicles to receptors
Speed: 0.001 m/s
Energy: ~ 2.3 femtoJoules
Slow, unreliable (noisy), very low energy

We have variety of applications for which we use computers

Data Heavy **(von Neumann bottleneck)**

Compute Heavy **(No von Neumann bottleneck)**

Computing Application	Characteristics					
	Compute Heavy	Data Heavy		Operational intensity FLOP/byte	Communication (Serial/Iterative)	Parallelism
Machine learning	hi					**high**
Graph problems	lo					high
Bayesian inference	hi					**medium**
Markov chain	hi					**high**
Data Bases (analytics)	lo					high
Data Bases (transactions)	med					medium
Search (indexing problem)	hi					**high**
Optimization problems (resource allocation)	hi					**low (worst case)**
Scientific Computing	high	low to high	low to high	medium	high	high
Finite Element Modelling	high	low	medium	medium	high	high
Email, chat, etc.	low	high	medium	low	high	low
Signal (image) processing	high	high	high	low	high	medium

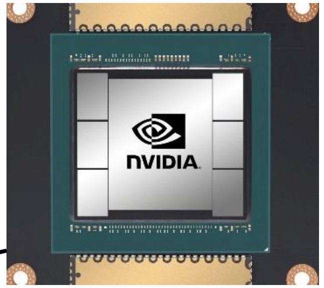
Version of the "No Free Lunch Theorem" applies here:

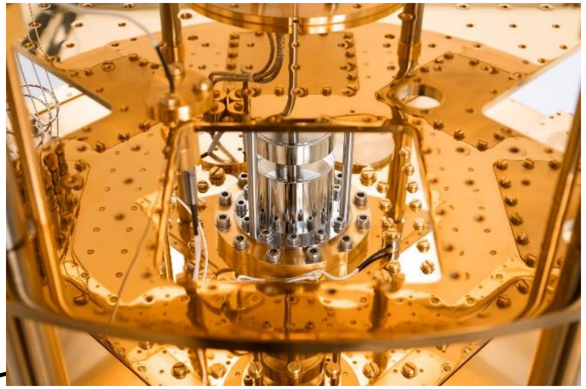
- It is unlikely a single *type* of computer is optimal for all applications

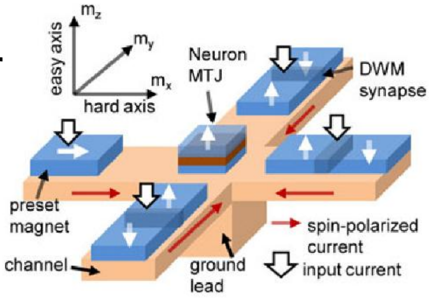
Milojicic, D., Bresniker, K., Campbell, G., Faraboschi, P., Strachan, J. P., & Williams, R. S.. Computing In-Memory, Revisited. *International Conference on Distributed Computing Systems (ICDCS) 2018.*

We have variety of applications for which we use computers

Computing Application
Machine learning
Graph problems
Bayesian inference
Markov chain
Data Bases (analytics)
Data Bases (transactions)
Search (indexing problem)
Optimization problems (resource allocation)
Scientific Computing
Finite Element Modelling
Email, chat, etc.
Signal (image) processing







Increasingly specialized and diverse computer hardware is the likely future.

End of Moore's Law and Dennard Scaling gives no good alternative.

This is a good thing:

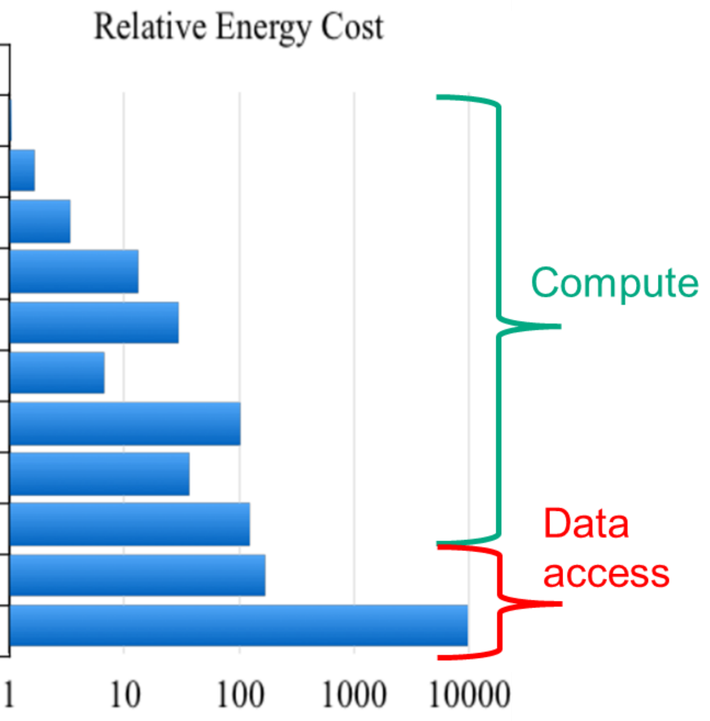
- Diverse ecosystem
- Increased efficiency and performance
- Drive breakthroughs in many fields of research

Who will be the next Nvidia?!

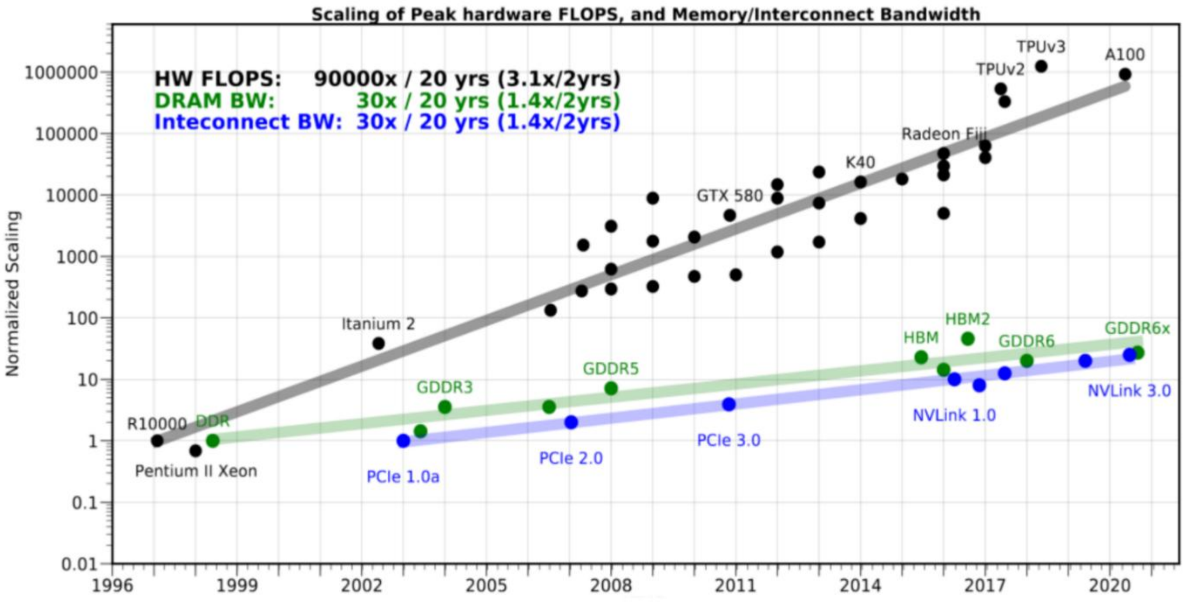
Key challenge to building more efficient hardware

Today: memory operations are far more expensive than computing operations

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



Source: W.Dally, Stanford



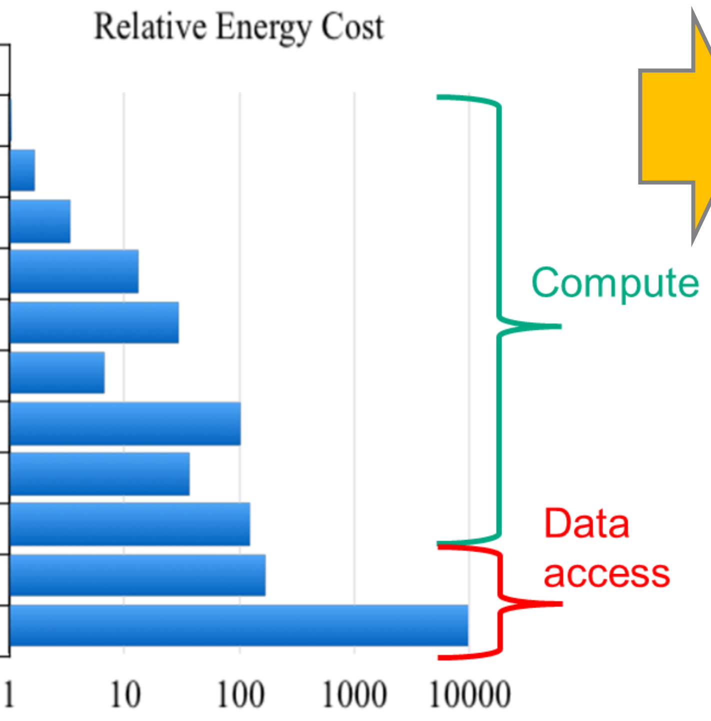
Over many decades:
Memory performance grew at 7% / year
CPU performance grew at 60% / year

Compute is free, memory is expensive!

Key challenge to building more efficient hardware

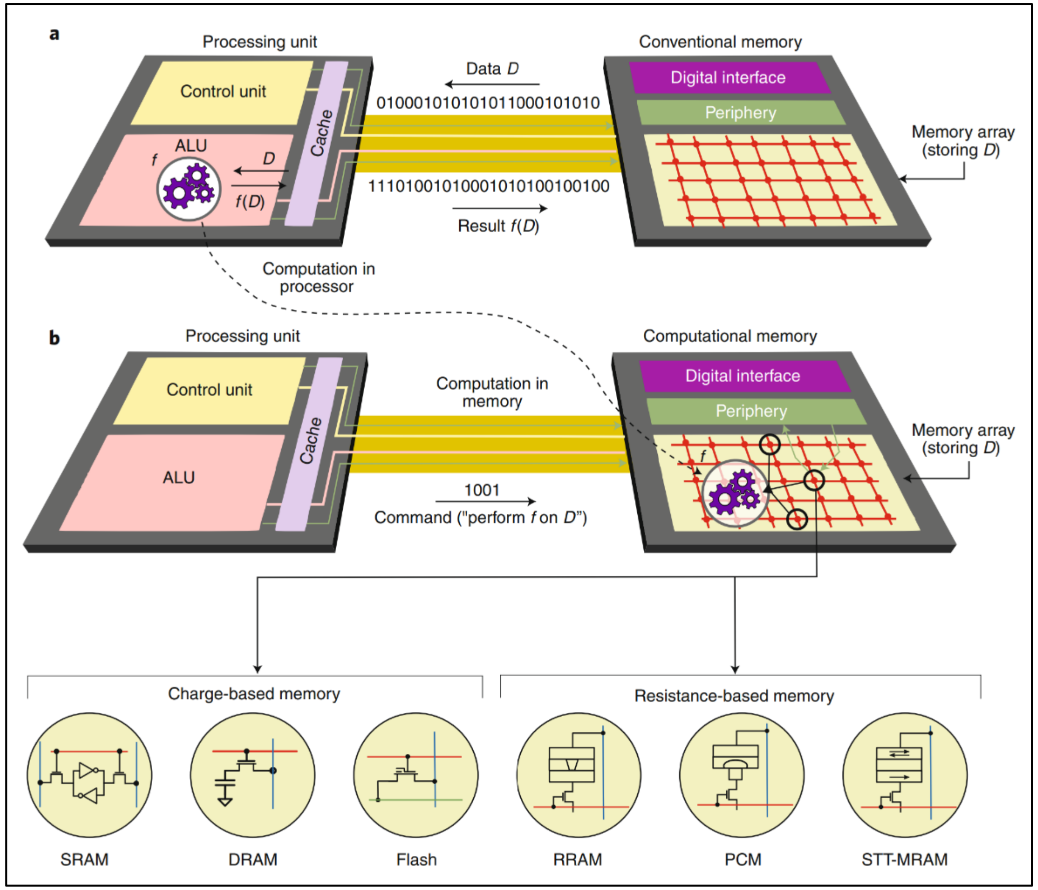
Today: memory operations are far more expensive than computing operations

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



Source: W.Dally, Stanford

- **Bring computation to the memory**
- **“In-memory computing”**

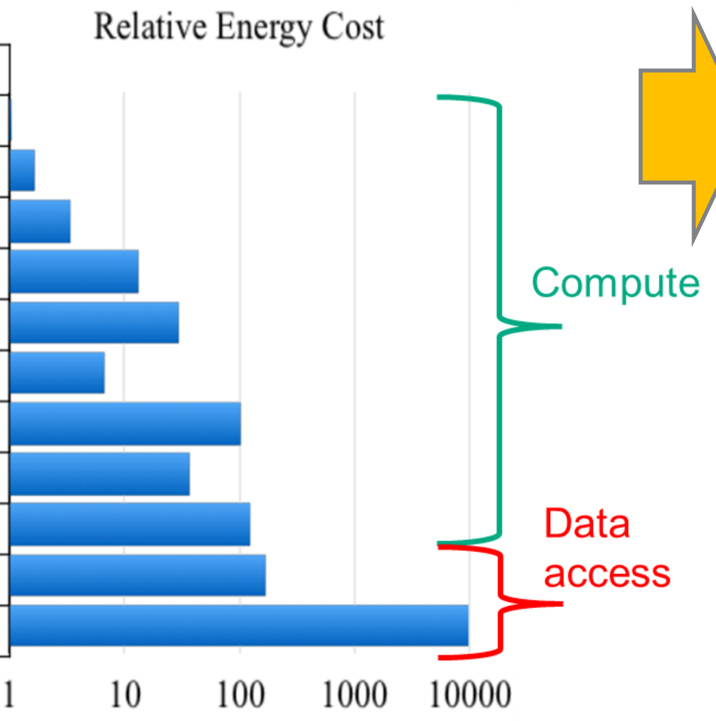


Sebastian, A., Le Gallo, M., Khaddam-Aljameh, R., & Eleftheriou, E. Memory devices and applications for in-memory computing. Nature Nanotechnology 2020

Key challenge to building more efficient hardware

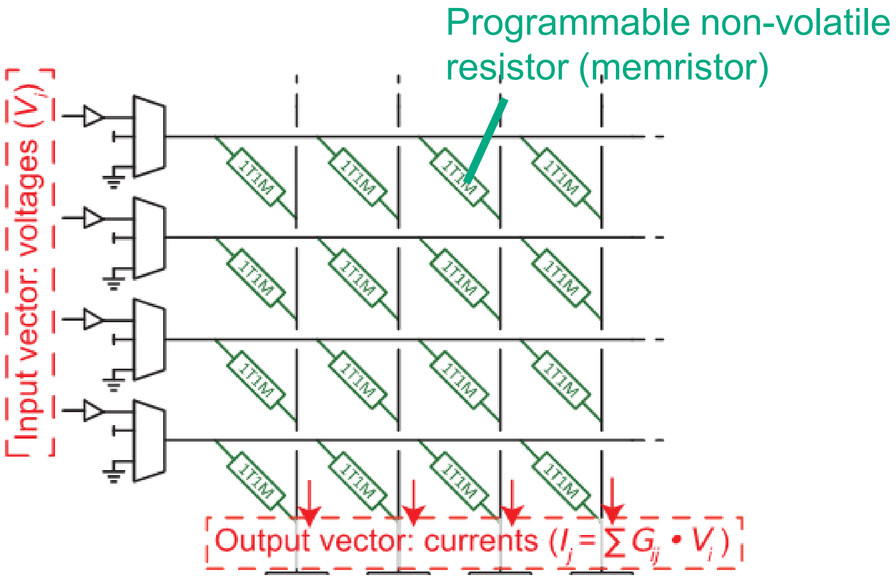
Today: memory operations are far more expensive than computing operations

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



Source: W.Dally, Stanford

Example: *Multiplication, Add, and Read* operations in crossbar memory circuits.
Leverage programmable resistive devices (memristors), ReRAM, PCM, Spintronic, Ferro-electric, etc

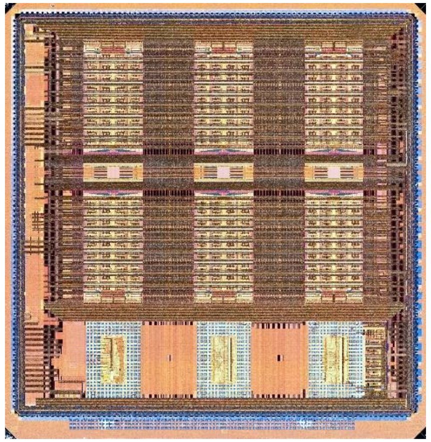


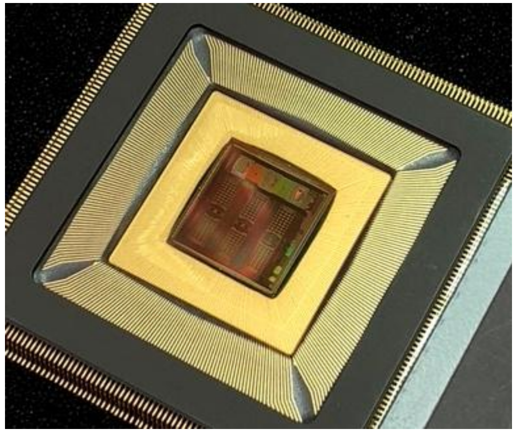
Vector-matrix product performed through Ohm's Law + Kirchhoff's Current Law
In-memory analog computing: Multiply & Add operation (@8-bit) in a 60x60 array consumes **<0.001 pJ per operation**

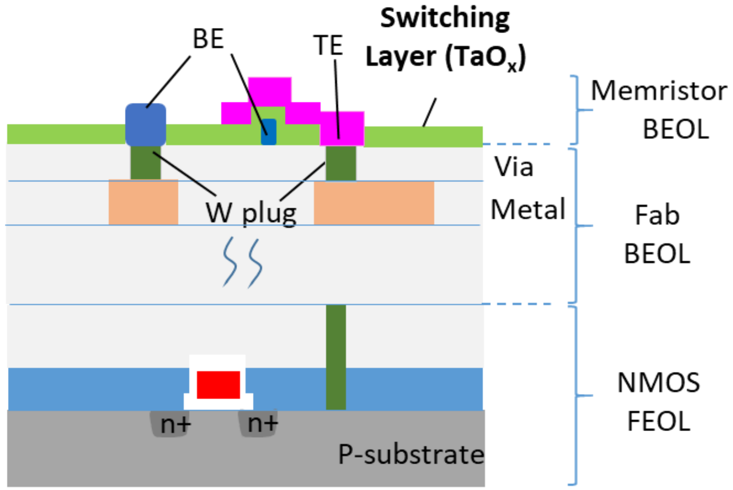
**Digital conversion not included here, but is in all future examples*

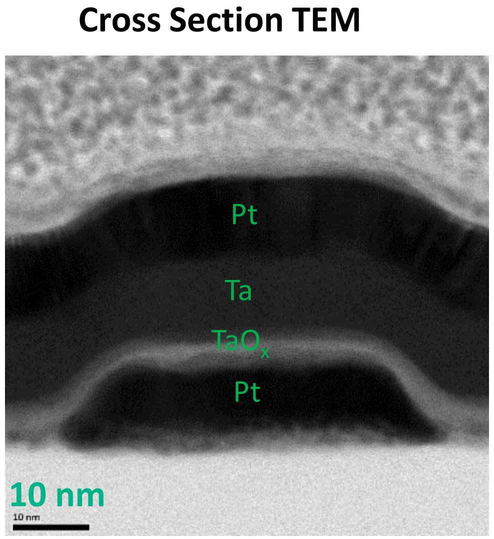
Engineering memristor crossbars for applications

Integrating CMOS and Memristors

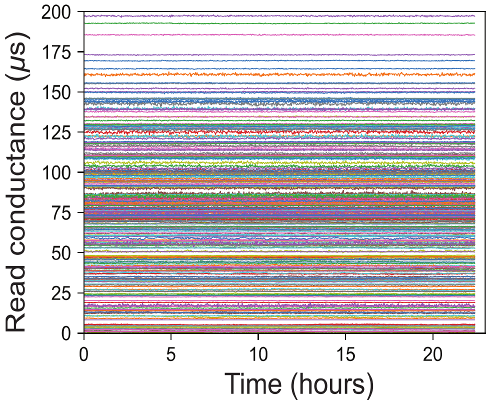


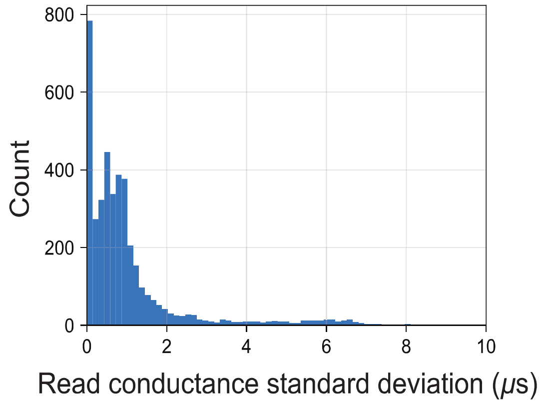






- In-memory analog multiply-accumulate (MAC) operations
- 25-50nm ReRAM/memristors, 180nm CMOS
- Down to 140ns Compute/Read operations, parallel across all arrays, results latched in Sample/Hold
- Multiple chips can be tiled and operate in parallel



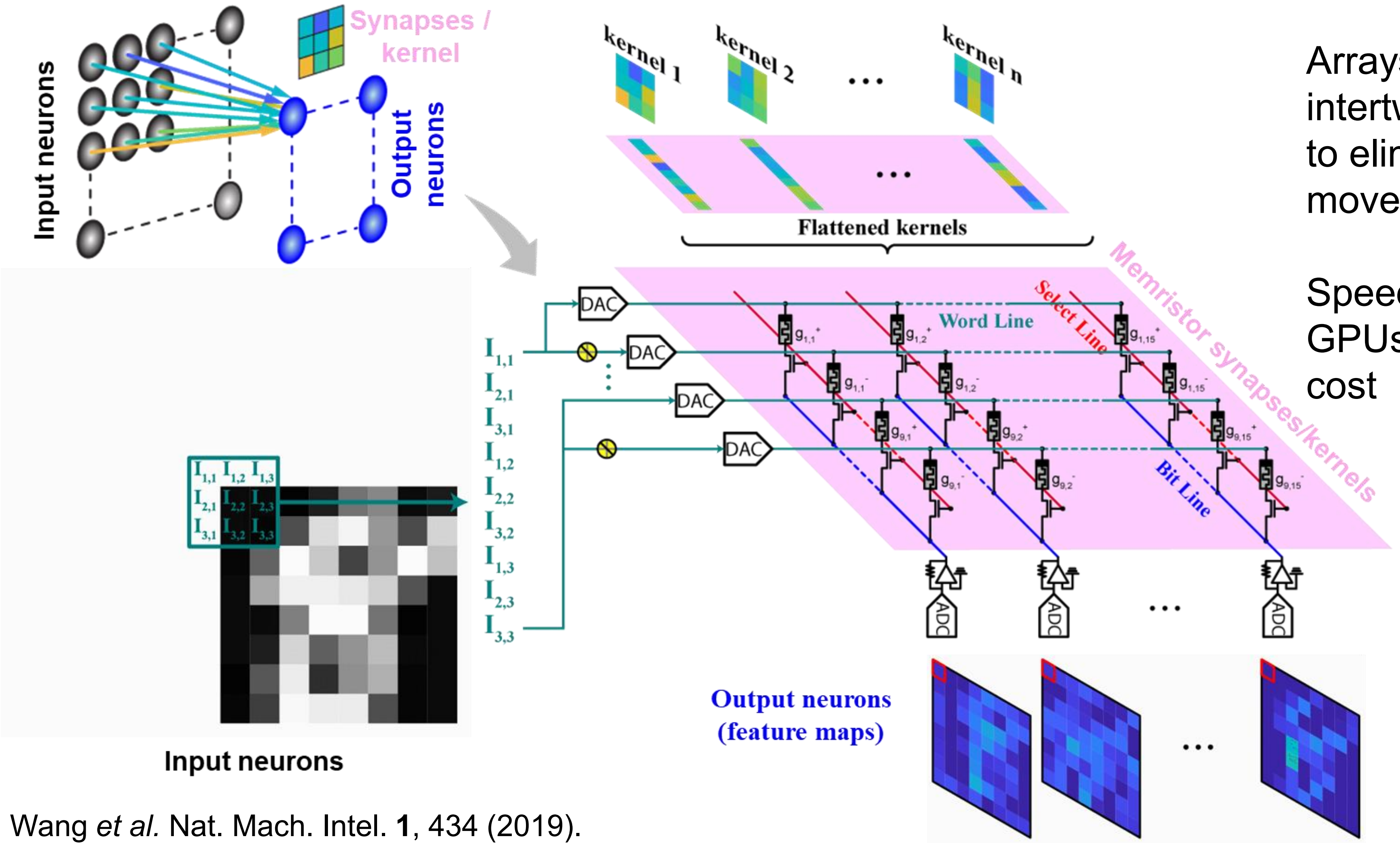


Analog programmable, $>10^5$ Conductance range
Stable conductances over years

Li, C. et al, **Inter. Mem. Workshop** (2020)
Li, C. et al, **Nature Electronics** (2018)
Hu, M., *et al*, **Advanced Materials** (2018)
Hu, M, et al, **DAC** (2016)

X. Sheng, et al, **Adv. Electron. Mater.** (2019)

Example: acceleration of deep neural networks



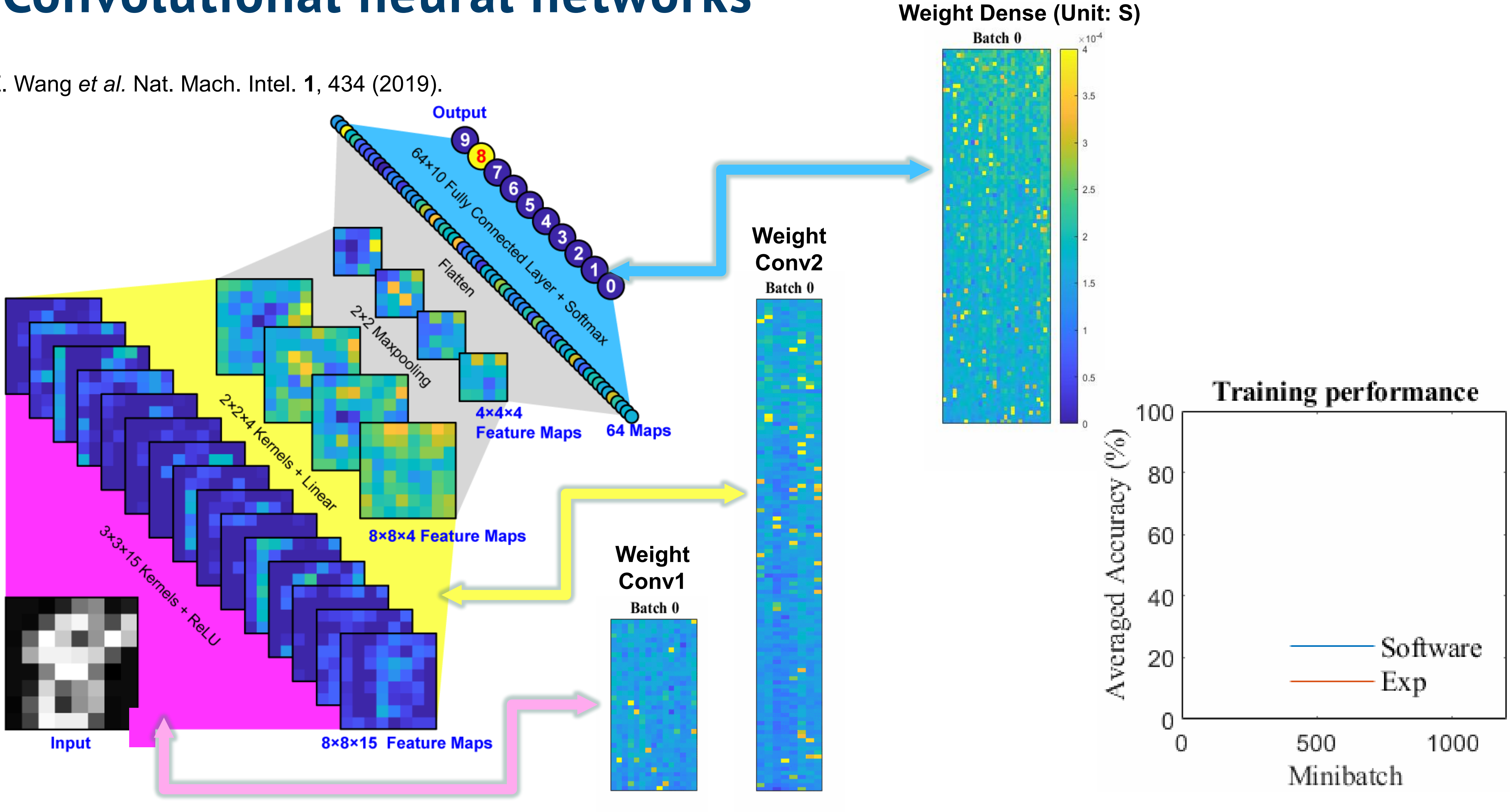
Z. Wang *et al.* Nat. Mach. Intel. **1**, 434 (2019).

Arrays of memristors intertwined with computing to eliminate expensive data movement.

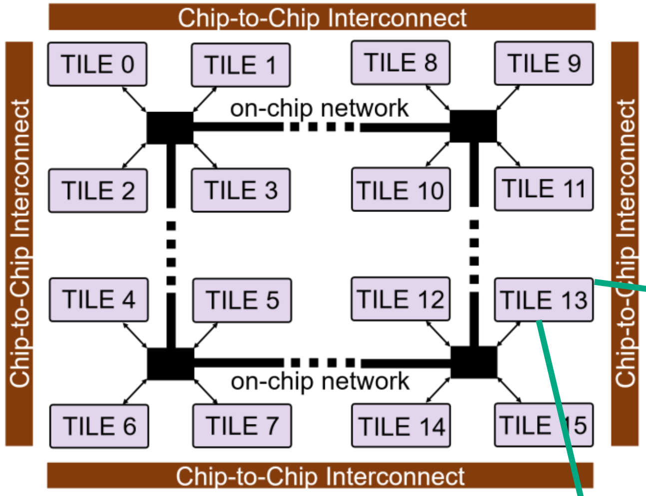
Speed-ups 10-100x over GPUs possible, with lower cost

Convolutional neural networks

Z. Wang *et al.* Nat. Mach. Intel. **1**, 434 (2019).



Scalable and Flexible Architectures Developed for Machine Learning



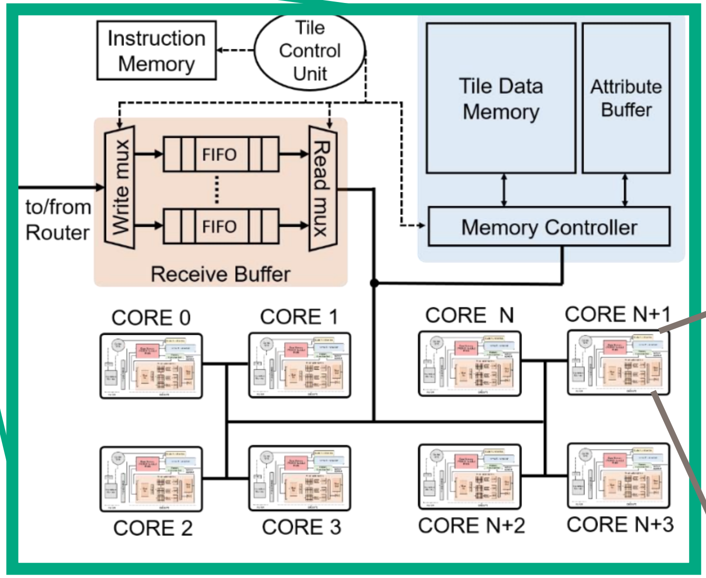
Node

Each "node" consists of multiple tiles with an on-chip network

In-memory ReRAM crossbars (each at 128x128) performing matrix vector multiplications

Digital units for logic, scalar, and vector ops

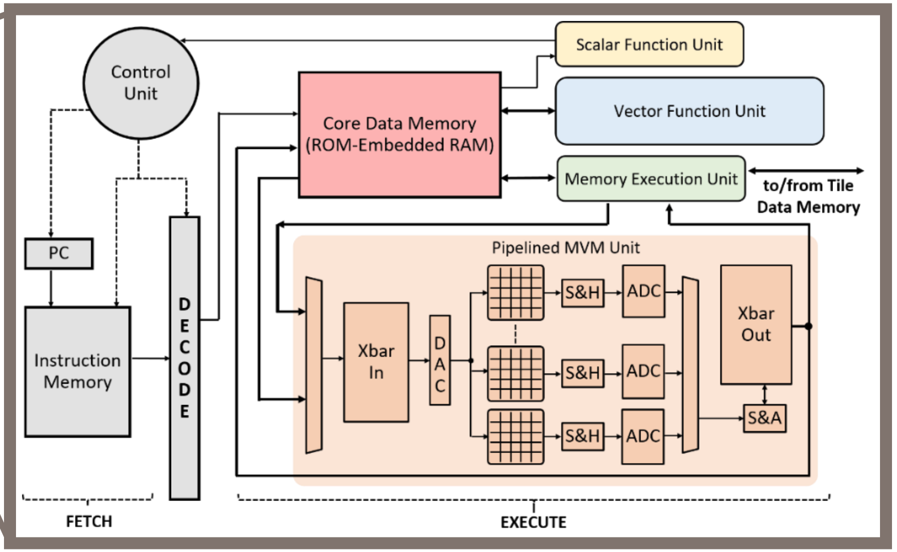
3-stage pipeline, instruction decoder, and instruction memory



Tile

Estimate @22nm CMOS
~250 mm^2, <50 Watts
46,000 crossbars
100 Trillion Ops/sec (8bit operations)
→ *Outperform GPUs in Throughput/Watt and cost*

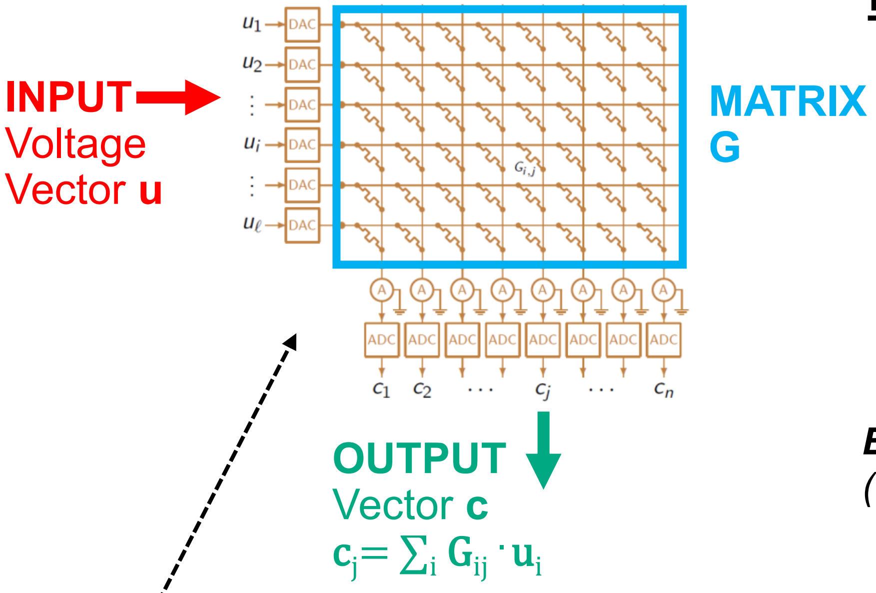
Core



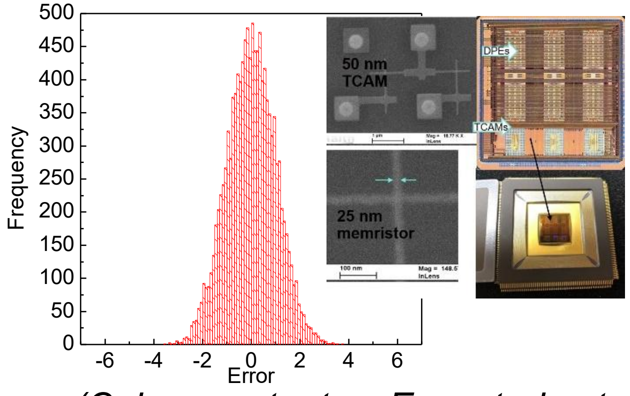
Within each "tile" are multiple "cores" with a shared memory for holding data to/from other tiles and to/from cores.

A Shafiee, *et al*, *ISCA* (2016)
A Nag, *et al. IEEE Micro* (2018)
A Ankit. *et al*, *ASPLOS* (2019)
A Ankit. *et al*, *IEEE Trans. on Computers* (2020)

Challenge: Presence of "Noise" in analog computing

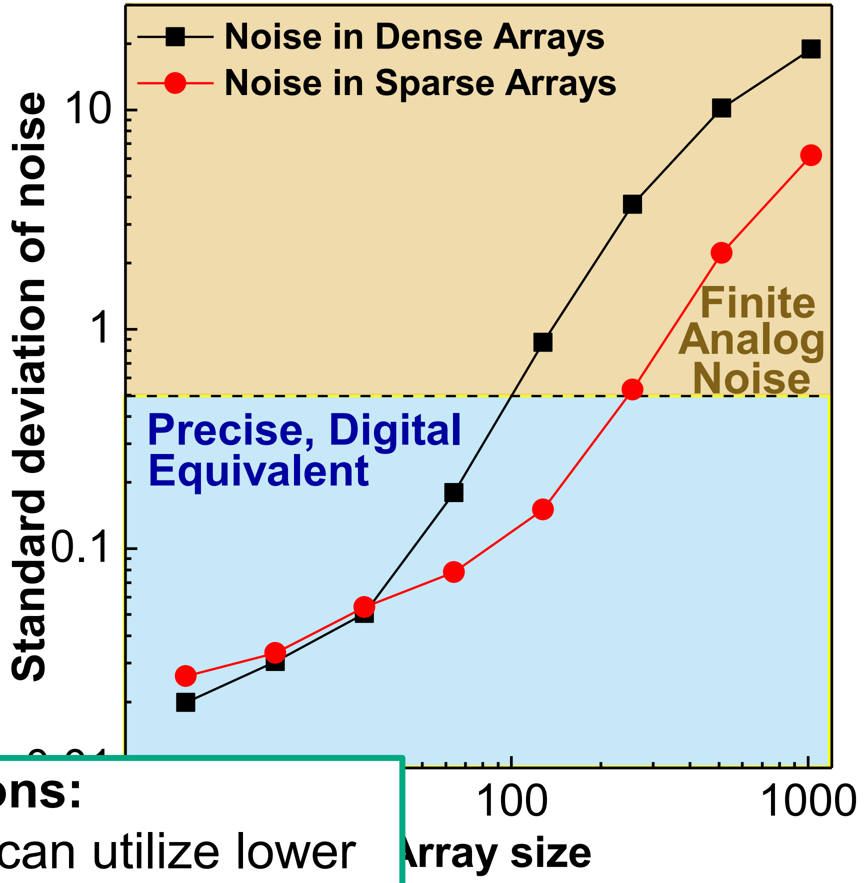


Experimental Data:



Error *(Column outputs – Expected outputs)*
(128 rows)

M. Hu, et. al, Adv. Mater. 2018
F. Cai, et al., Nature Electr 2020



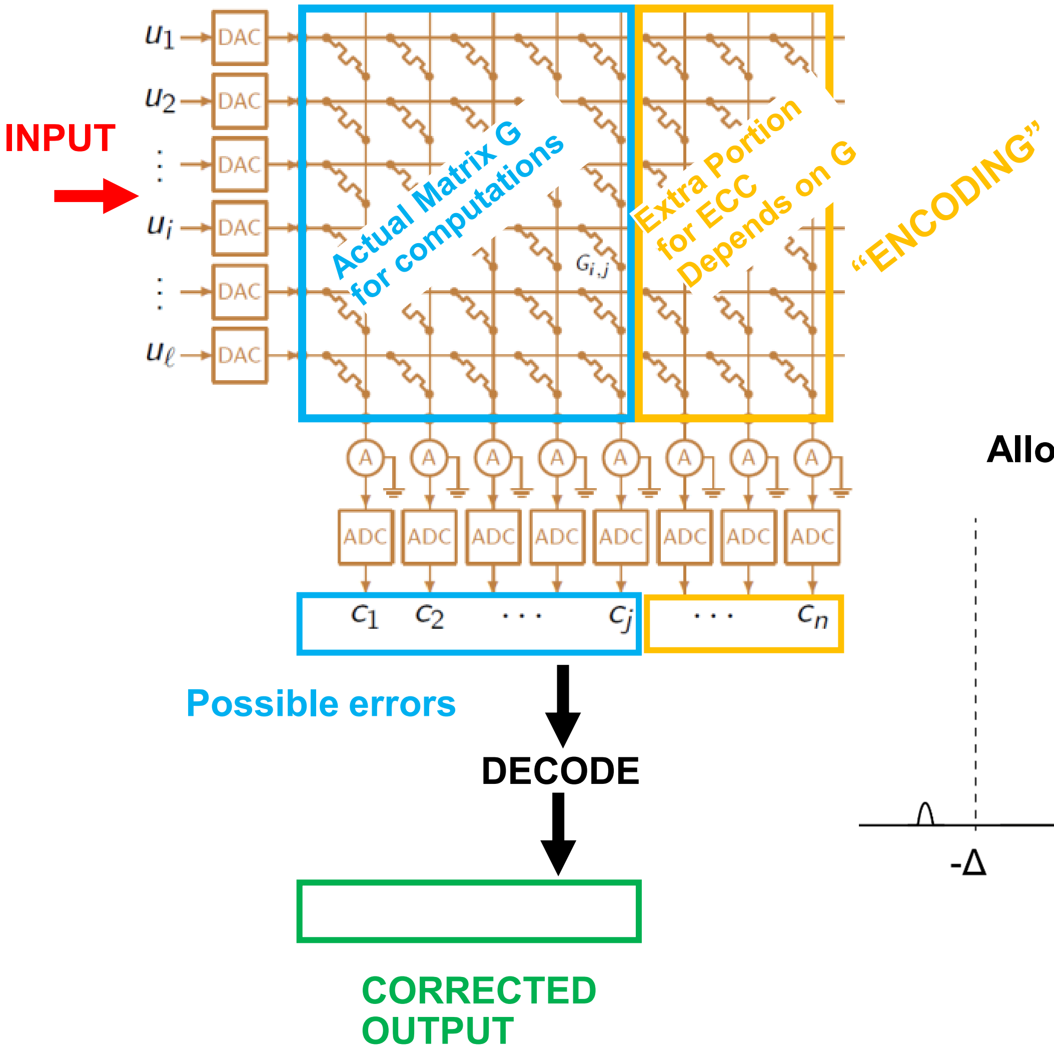
Many sources of errors:
- Interconnecting wires ≠ 0 resistance
- Device resistances are finite
- Nonlinear devices G = G(V)
- Johnson/Shot noise sources
- Temperature effects
- Device resistance fluctuations

- Comput
- Comput
- But nois

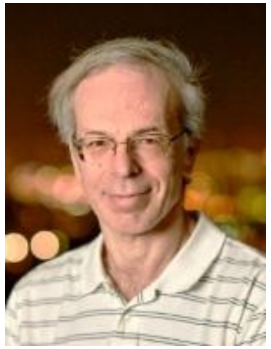
$\frac{\ldots tions}{\ldots nsumed} = \frac{N^2}{N}$

Two Research Directions:
- Find applications that can utilize lower precision and analog noise
- Find ways to reduce/correct these errors

Analog Error Correcting Codes (analog ECC)



Novel encoding schemes invented by Prof Ronny Roth, Technion
Summer visitor

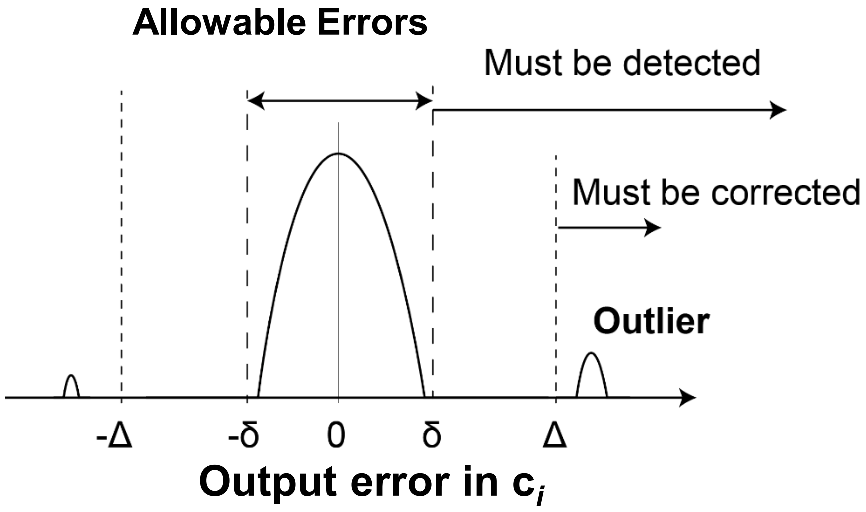


Roth, Ron M *IEEE Trans. on Info. Theory* (2018)
Roth, Ron M. *IEEE Trans. on Info. Theory* (2020)

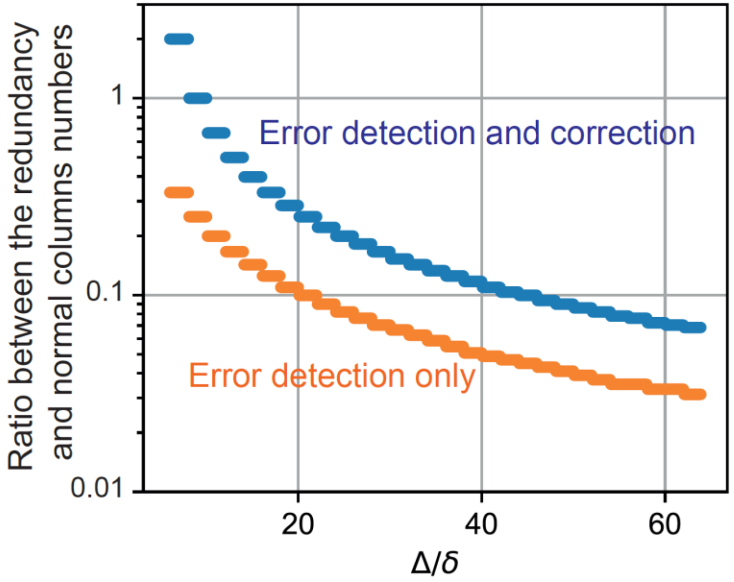
ECC for *computation*, works for all inputs

Two different ECC schemes

1) Integer precise computations (2018)
2) Tolerate small analog errors, detect/correct large deviations (2020)



Adjustable ratio Δ/δ

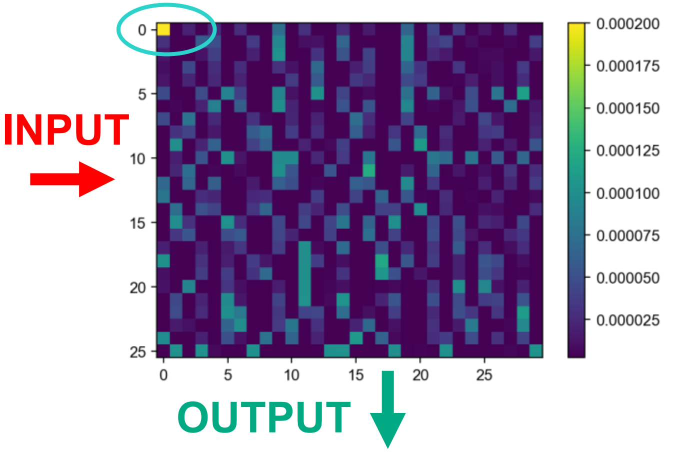


Smaller ratio → larger overhead

C. Li, R. Roth, C. Graves, X. Sheng, J.P. Strachan, IEDM 2020

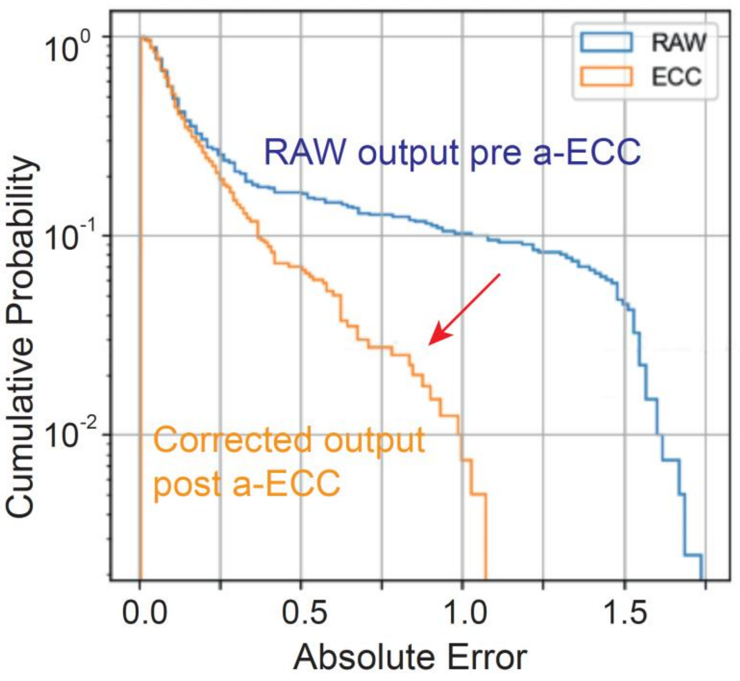
Demonstrations of Analog Error Correcting Codes (analog ECC)

Test analog-ECC on “Device Failures”

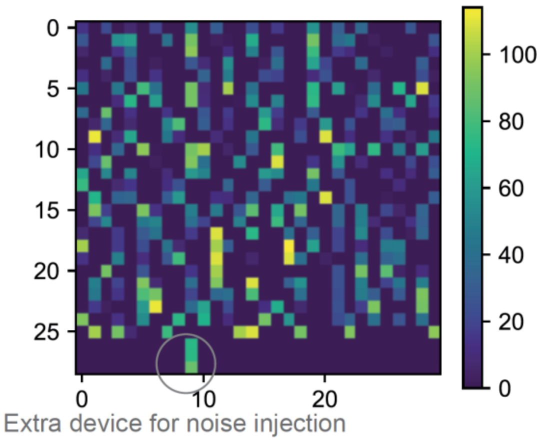


Procedure:

- Turn random devices far ON (low resistance)
- Apply random inputs
- Measure (out – expected out) with and without analog-ECC

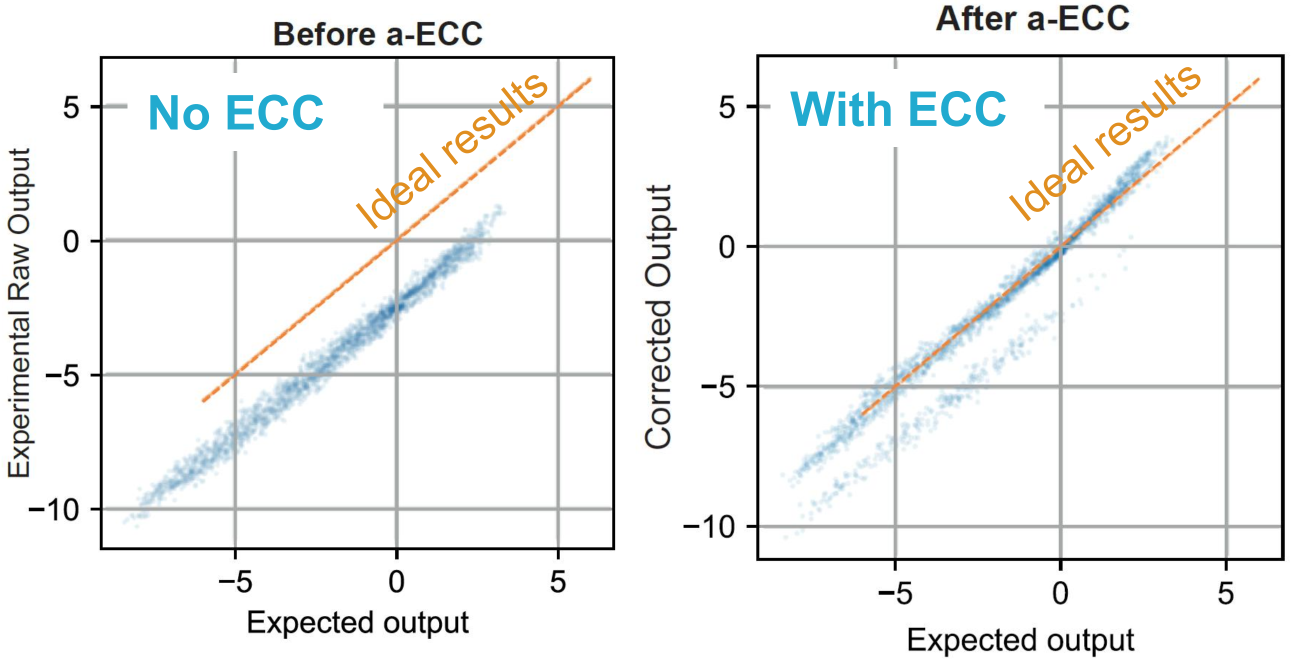


Test analog-ECC on “Noise Injection”



Procedure:

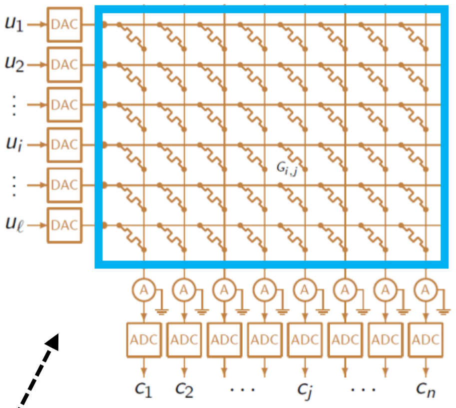
- Use extra rows to inject noise into select columns
- Apply random inputs
- Measure (out – expected out) with and without analog-ECC



C. Li, R. Roth, C. Graves, X. Sheng, J.P. Strachan, IEDM 2020

Challenge: Presence of “Noise” in analog computing

INPUT
Voltage
Vector **u**

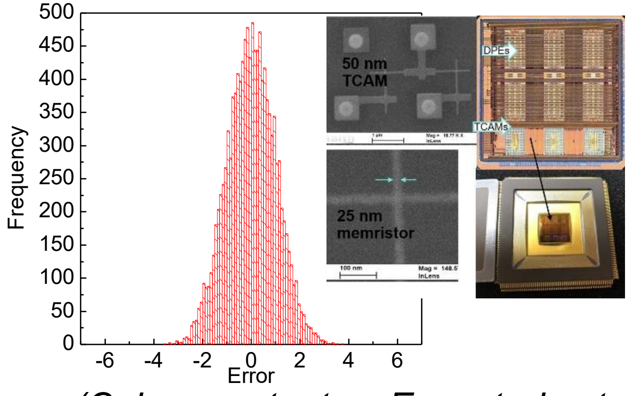


MATRIX
G

OUTPUT
Vector **c**
$c_j = \sum_i G_{ij} \cdot u_i$

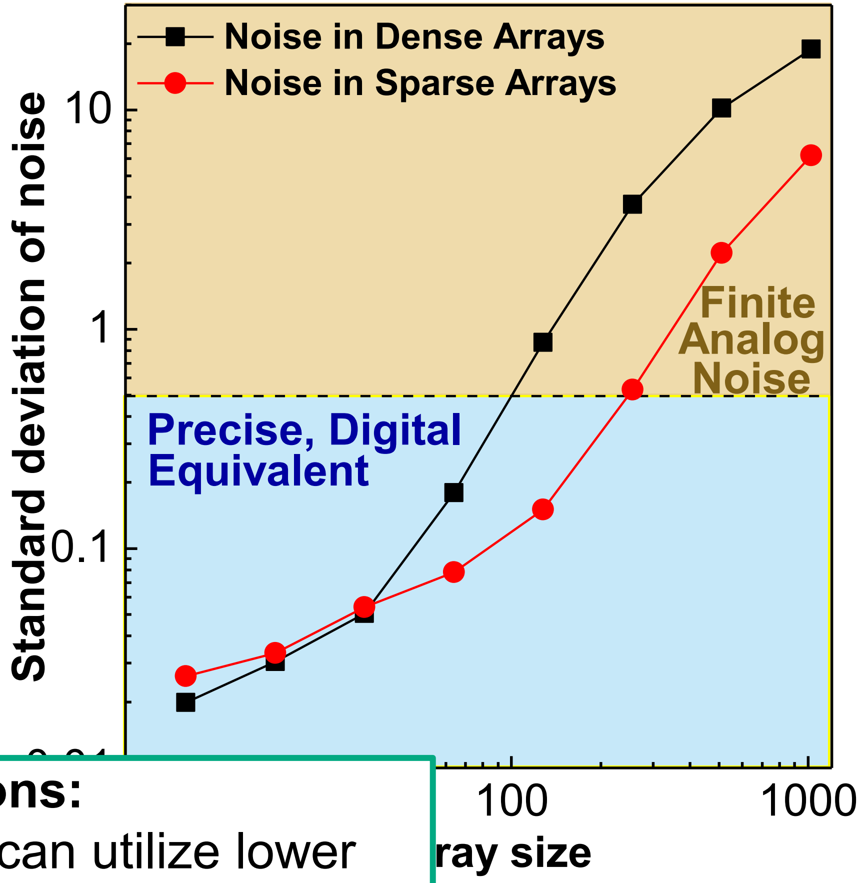
Many sources of errors:
- Interconnecting wires ≠ 0 resistance
- Device resistances are finite
- Nonlinear devices G = G(V)
- Johnson/Shot noise sources
- Temperature effects
- Device resistance fluctuations

Experimental Data:



Error *(Column outputs – Expected outputs)*
(128 rows)

M. Hu, et. al, Adv. Mater. 2018
F. Cai, et al., Nature Electr 2020



- Comput
- Comput
- But nois

ions / umed $= \frac{N^2}{N}$

Two Research Directions:
- ➡ Find applications that can utilize lower precision and analog noise
- ▪ Find ways to reduce/correct these errors

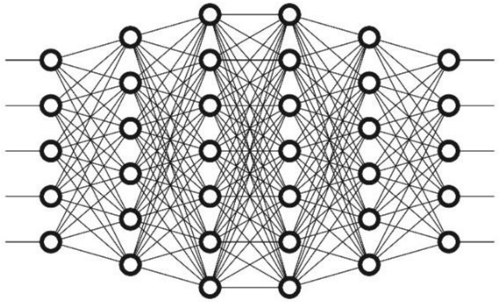
C. Li, R. Roth, C. Graves, X. Sheng, J.P. Strachan, IEDM 2020
Roth, Ron M. *IEEE Trans. on Info. Theory* (2020)

Today's hardest and most interesting problems: <u>Optimization</u>

Learning in biology itself is an optimization process; as is the **Training of Artificial Neural Networks**

Practical industrial problems in scheduling, routing, financial portfolios, system design, etc



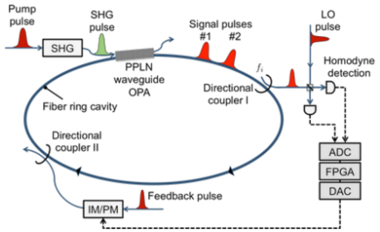

Many challenging "non-convex" problems: k-SAT, Graph Coloring, Traveling Salesman, Mixed-Integer Linear Programming, Knapsack, Weighted MaxCut, ….

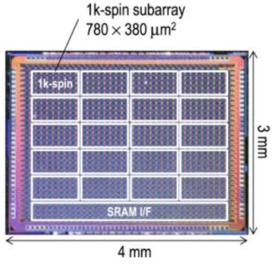
Interesting ones are NP-complete or NP-hard

Many heuristic "algorithms" and approaches:

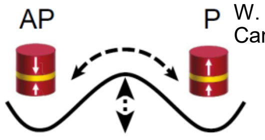
Local Search, Simulated Annealing, Evolutionary Algorithms, Boltzmann machines, Hopfield networks, etc.

Many physical systems and device implementations:

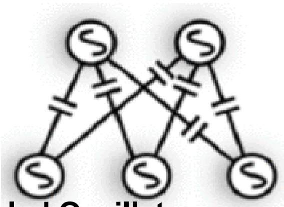




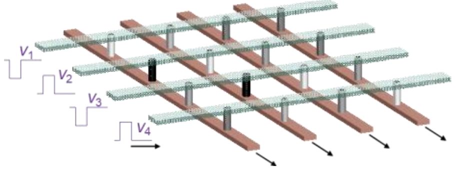
Optical and CMOS Ising Machines



P-bits, Magnetic Tunnel Junctions
W. A. Borders, A. Z. Pervaiz, S. Fukami, K. Y. Camsari, H. Ohno, and S. Datta, Nature (2019)



Coupled Oscillators
Parihar, Abhinav, et al. *Scientific reports* 7.1 (2017)



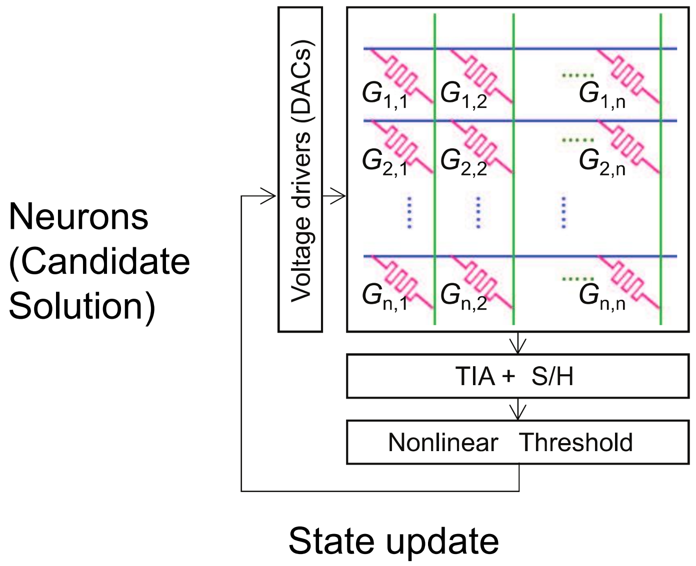
Analog in-memory computing

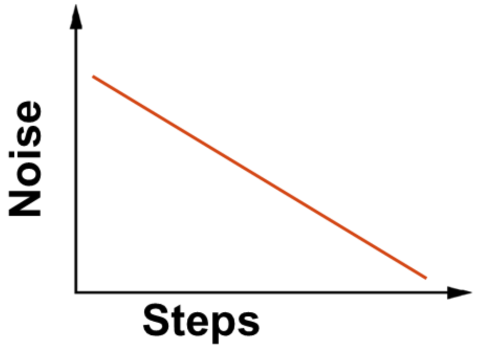
And Many Others!

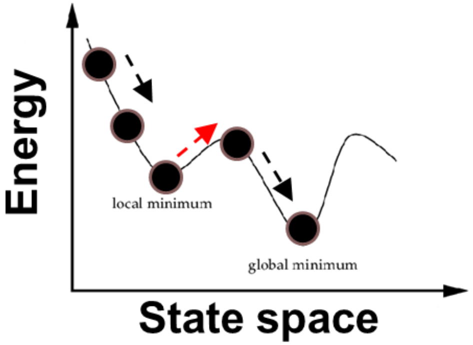
Goal: Provide *some* speed-up and energy reduction for these hard problems

Our HW approach: in-memory, mixed analog-digital circuits

- Massive connection weights need to be in some compact programmable memory
- Ideally, an analog (multi-bit) tunable Memristive crossbar implements the couplings. Flash, SRAM, and other memory tech are options.
- Additional CMOS circuitry for neurons/spins, updates, and control







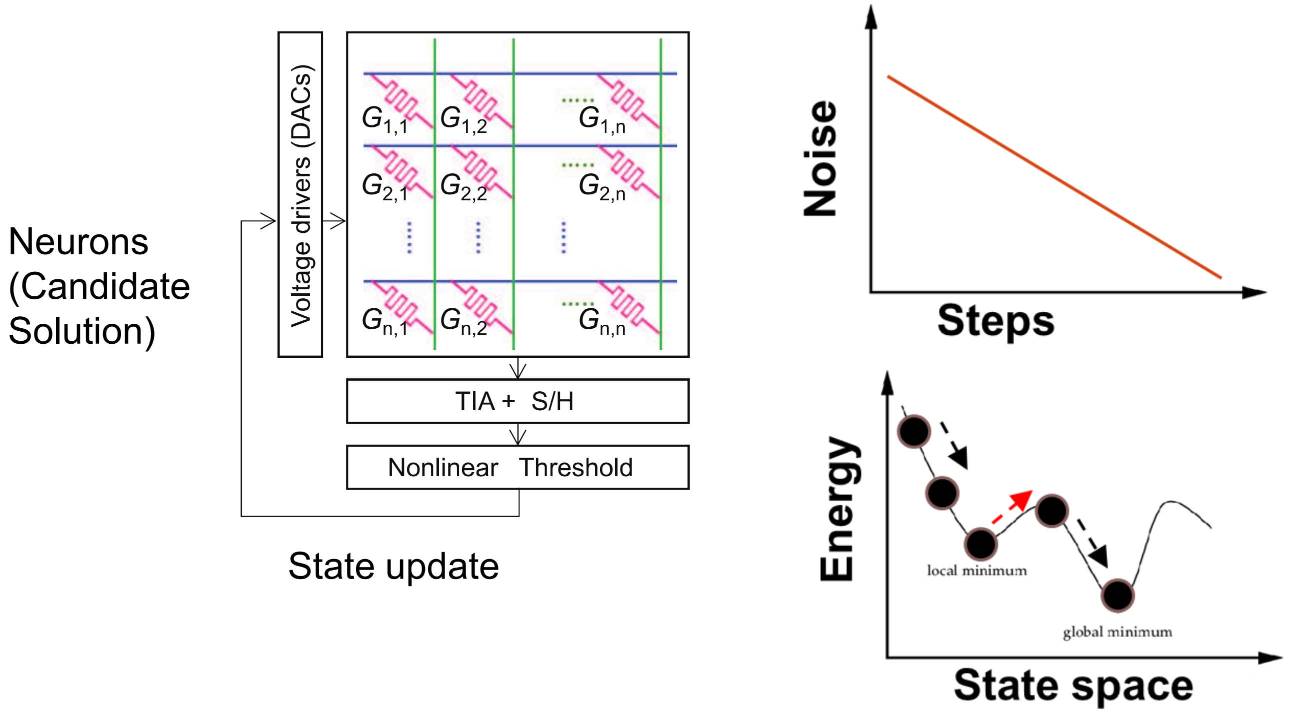
Proposed QUBO/Ising/Hopfield solver

What advantages?

- Memristive crossbars provide **fully-connected, multi-bit, programmable** connectivity
  - 3D integration potential
- Need **noise** for probabilistic sampling
- High **Parallelism**: Can update batches of neurons in parallel, many crossbars running in parallel
- Favorable **energy/latency** trade-off at lower precision
- **Flexibility** to support many heuristics
  - Local Search, WalkSAT, Stochastic and Simulated annealing, Parallel Tempering, etc.

Our HW approach: in-memory, mixed analog-digital circuits

- Massive connection weights need to be in some compact programmable memory
- Ideally, an analog (multi-bit) tunable Memristive crossbar implements the couplings. Flash, SRAM, and other memory tech are options.
- Additional CMOS circuitry for neurons/spins, updates, and control



Proposed QUBO/Ising/Hopfield solver

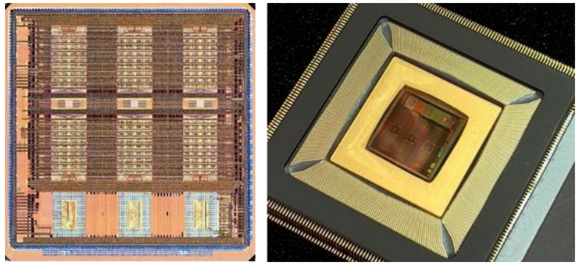
What disadvantages?

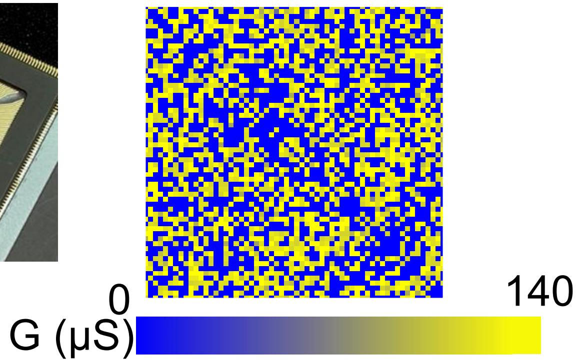
- Memristive technologies under development
- Memristor precision and fluctuations could be problematic

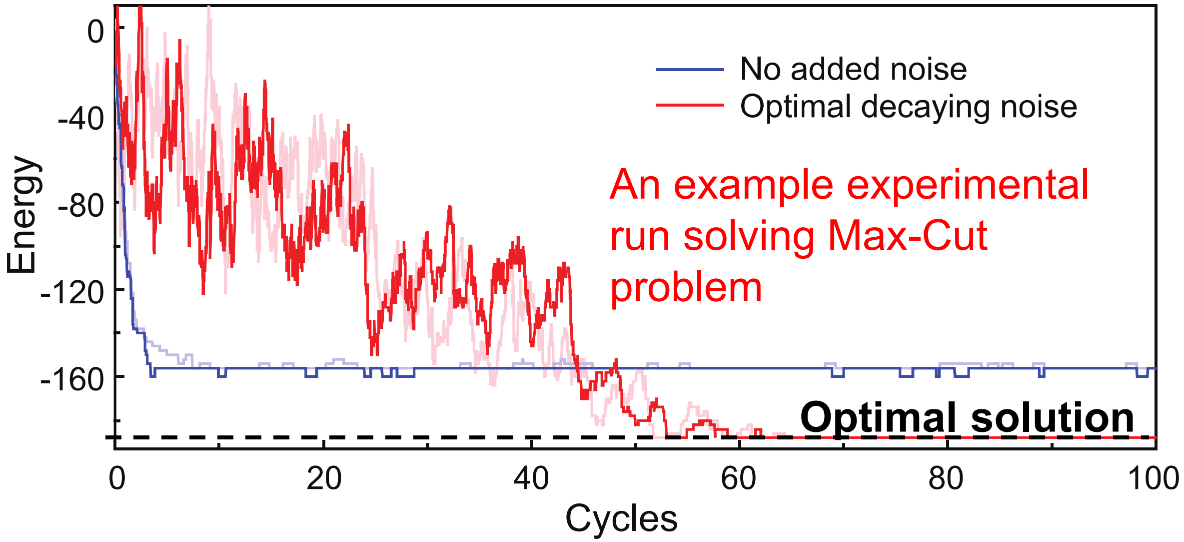
Experimental demonstrations and performance comparisons

Solving NP-hard Max-Cut problems – 60x60 problem

Experimental Memory Pattern







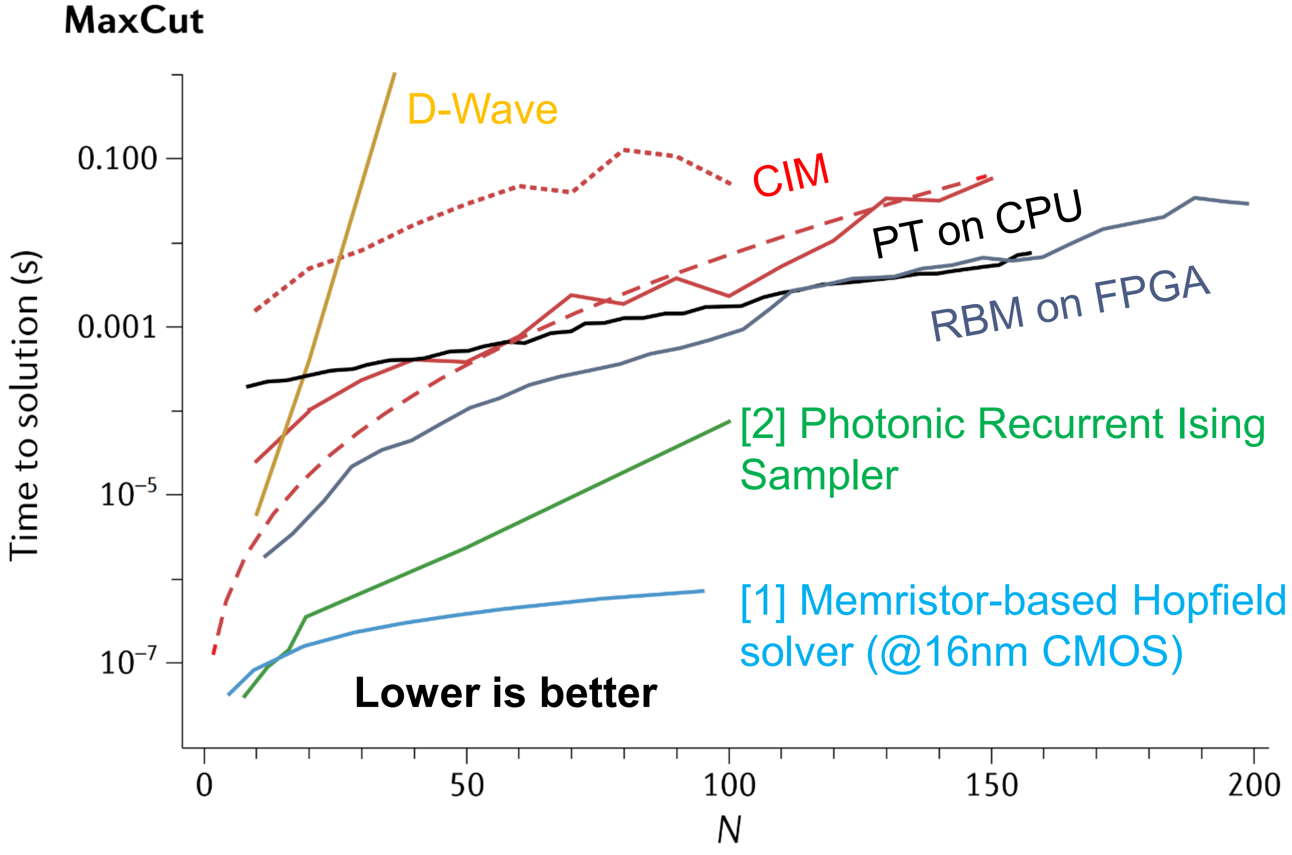


Figure from N. Mohseni, P.L. McMahon, T. Byrnes. *Nature Reviews Physics* (2022)

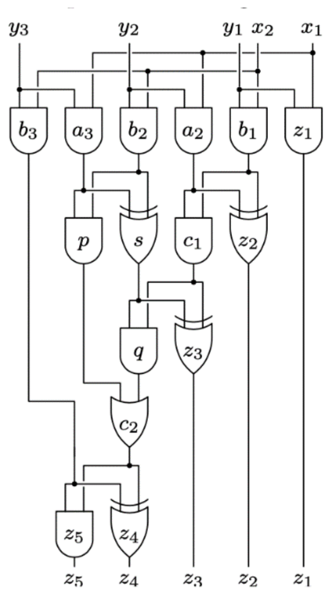
[1] F. Cai, S. Kumar,T. Van Vaerenbergh,… JP Strachan, *Nature Electronics (2020)*
[2] C. Roques-Carmes, et al. *Nature Comm.* (2020)

Harder problem: 3SAT

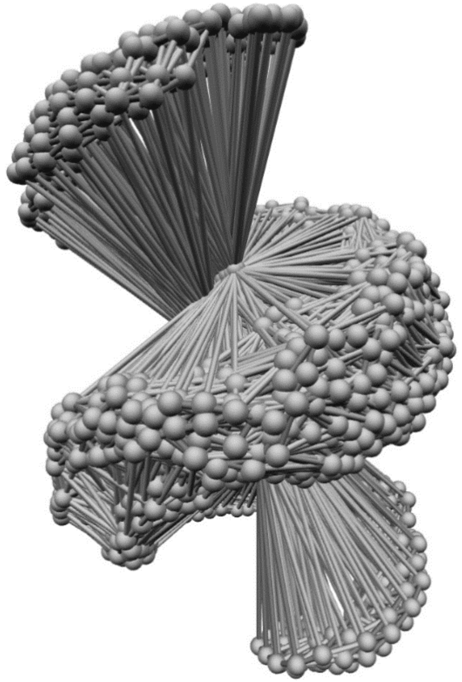
OR AND

3SAT: find boolean assignment of variables x_i to satisfy: $(x_1 \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee x_4 \vee \overline{x_5}) \wedge \ldots$

"Hard" problems have many more clauses than variables (~*4.25x*) → many interdependencies



Fault testing a circuit (*)



SAT formulation

* from D. Knuth, TAOCP, Volume 4, Fascicle 6, 2015

Harder problem: 3SAT

3SAT: find boolean assignment of variables x_i to satisfy: $\underset{\text{OR}}{(x_1 \vee x_2 \vee x_3)} \overset{\text{AND}}{\wedge} (\overline{x_1} \vee x_4 \vee \overline{x_5}) \wedge \ldots$

Convert into Energy/Loss minimization problem

Minimize $E = \underbrace{(1 - x_1) * (1 - x_2) * (1 - x_3)}_{} + (x_1) * (1 - x_4) * (x_5) + \ldots.$

$\Rightarrow (1 - x_1 - x_2 - x_3 + x_1 x_2 + x_1 x_3 + x_2 x_3 - x_1 x_2 x_3)$

Pair-wise 2nd order coupling

3rd order coupling

Ising / QUBO hardware cannot compute 3rd order products

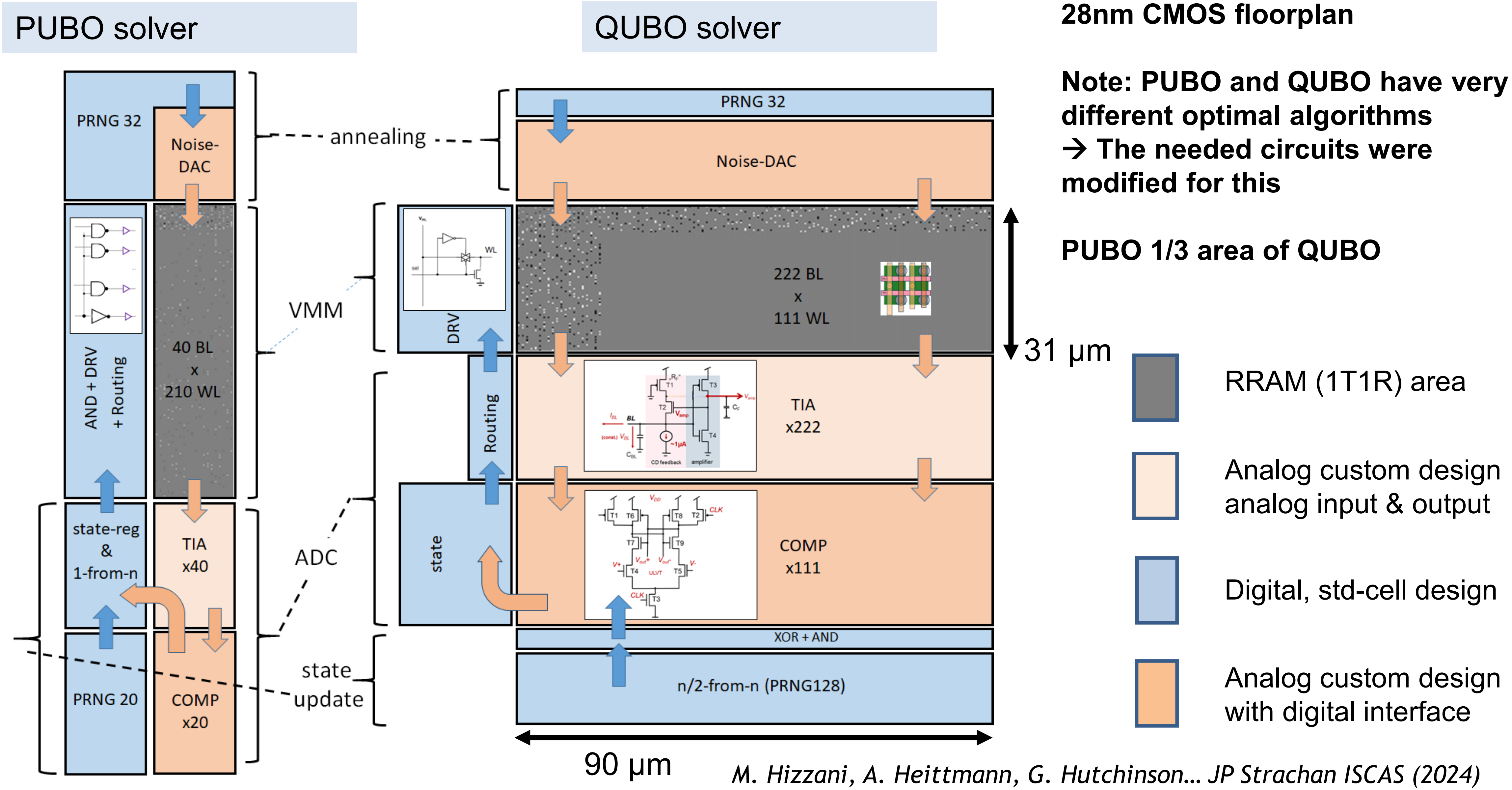
Introduce auxiliary variable $y = x_1 x_2$

$\Rightarrow \quad E_{QUBO} = (1 - x_1 - x_2 - x_3 + x_1 x_2 + x_1 x_3 + x_2 x_3 - y\, x_3) + \lambda\,(x_1 x_2 - 2x_1 y - 2x_2 y + 3y)$

Need to add penalty terms to enforce $y = x_1 x_2$

Or $E_{PUBO} = (1 - x_1 - x_2 - x_3 + x_1 x_2 + x_1 x_3 + x_2 x_3 - x_1 x_2 x_3)$ keep 3rd order terms, but more complex operations needed!

Mixed analog-digital design of PUBO and QUBO



28nm CMOS floorplan

Note: PUBO and QUBO have very different optimal algorithms → The needed circuits were modified for this

PUBO 1/3 area of QUBO

M. Hizzani, A. Heittmann, G. Hutchinson… JP Strachan ISCAS (2024)

Performance comparisons of optimization solvers

Previous QUBO memristor solver — Higher order solver

	Coherent Ising Machine (CIM)[7, 11]	D-Wave 2000Q[8,10]	sparse Ising Machine (sIM)[9]: FPGA sIM	sparse Ising Machine (sIM)[9]: Nanodevice sIM	mem-SO-HNN[4, 14]	Augmented Ising Machine (AIMs)[10]	This Work: WalkSAT/SKC	This Work: HO-HNN
Spin representation	Coherent light	Superconducting qubits	Digital bits	CMOS-MTJ p-bit	Digital bits	Analog charge based	Digital bits	
Coupling representation	Coupling matrix in FPGA	Flux storage	Sparse coupling matrix in FPGA		Memristor crossbar	Custom CMOS coupling cell	Memristor crossbar	
Connectivity	All-to-all	Sparse	Sparse	Sparse	All-to-all	Sparse	All-to-all	
Dynamics	discrete-time	continuous-time	discrete-time	continuous-time	discrete-time	continuous-time	discrete-time	
Interaction	High order	Second order	Second order	Second order	Second order	Third order	High order	
Frequency	1GHz		30MHz	1GHz	500 MHz		500 MHz	
Time-to-Solution (TTS)	388.9us (N=100)	44ms (N=20)	*<L> = 2555.09s (N=100)	*<L> = 77.77s (N=100)	12.12ms (N=100)	73.8us (N=100)	1.8us (N=20) **51.8us** (N=100) ***<L>=22us**(N=100)	**0.77us** (N=20) 113.4us (N=100)
Power	50W	25kW	75W	**38.7mW**	317.2mW	300mW (N=500)	**13.35mW** (N=20) 66.4mW (N=100)	42.4mW (N=20) 209mW (N=100)
Energy-to-Solution (ETS)	19.445mJ (N=100)	1.1kJ	191.632kJ (N=100)	3J (N=100)	3.84mJ (N=100)		**24.03nJ** (N=20) **3.43uJ** (N=100) ***<E>=1.46uJ** (N=100)	32.6nJ (N=20) 23.7uJ (N=100)
Area	1 km fibre ring cavity	>10m^2 room			0.0224 mm^2	6.76 mm^2	0.0214 mm^2 (N=100)	**0.0197 mm^2** (N=100)
Solutions per second per watt	51.427	9x10^{-4}	5.21x10^{-6}	0.33	260.114		**4.1x10^7** (N=20) **3x10^5** (N=100)	3x10^7 (N=20) 4.2x10^4 (N=100)
Solutions per second per mm^2					3.7x10^3	2x10^3	**9x10^5**	4.5x10^5

All solvers benchmarked on hard instances of 3-SAT problems

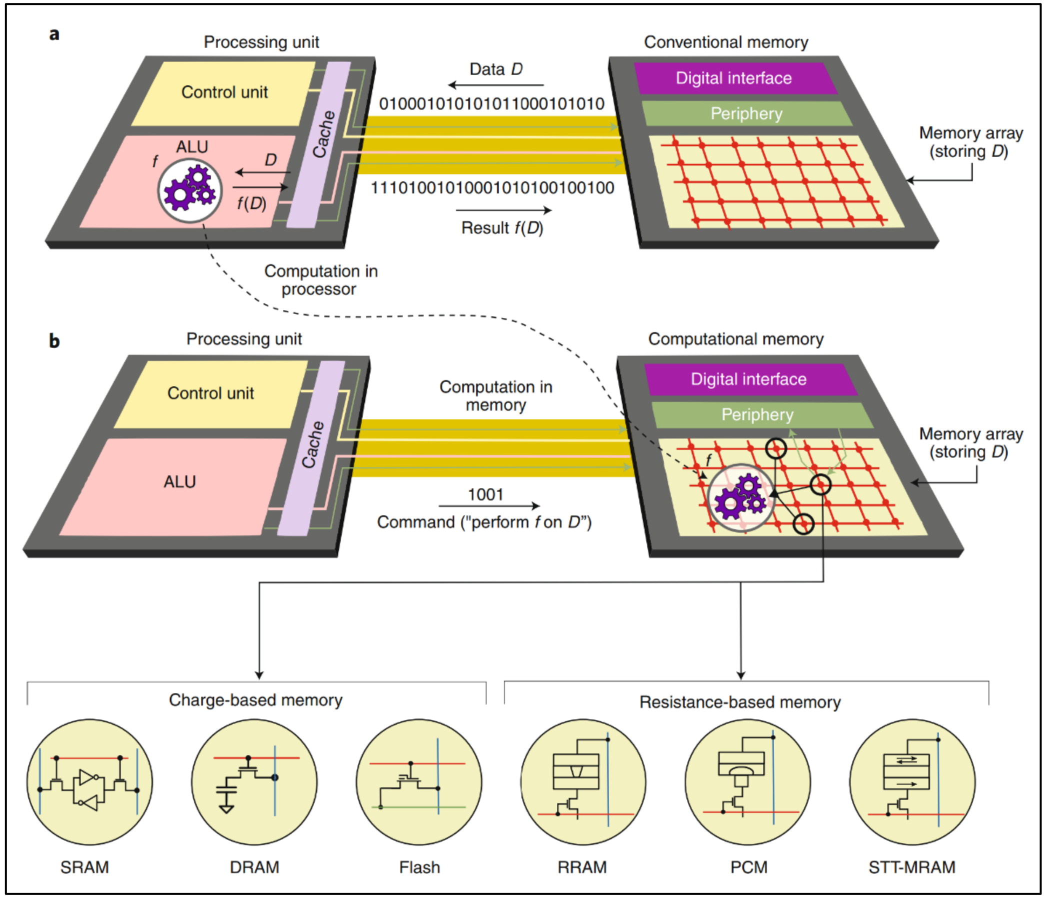
T. Bhattacharya et al., Nature Communications, 15, 8211 (2024)

Best Time-to-solution

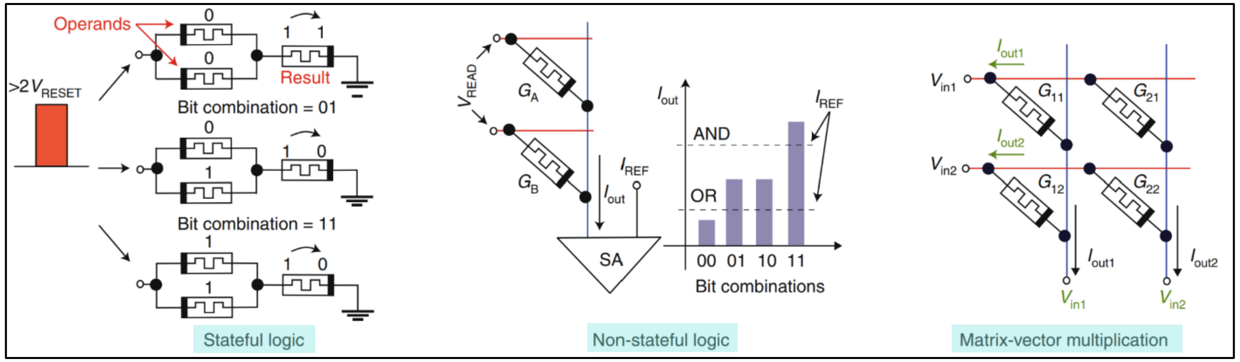
Best Energy-to-solution

Best solutions per second per Watt / mm^2

What other powerful computing primitives should we build?



Beyond memristor crossbar, a few computing primitives listed:

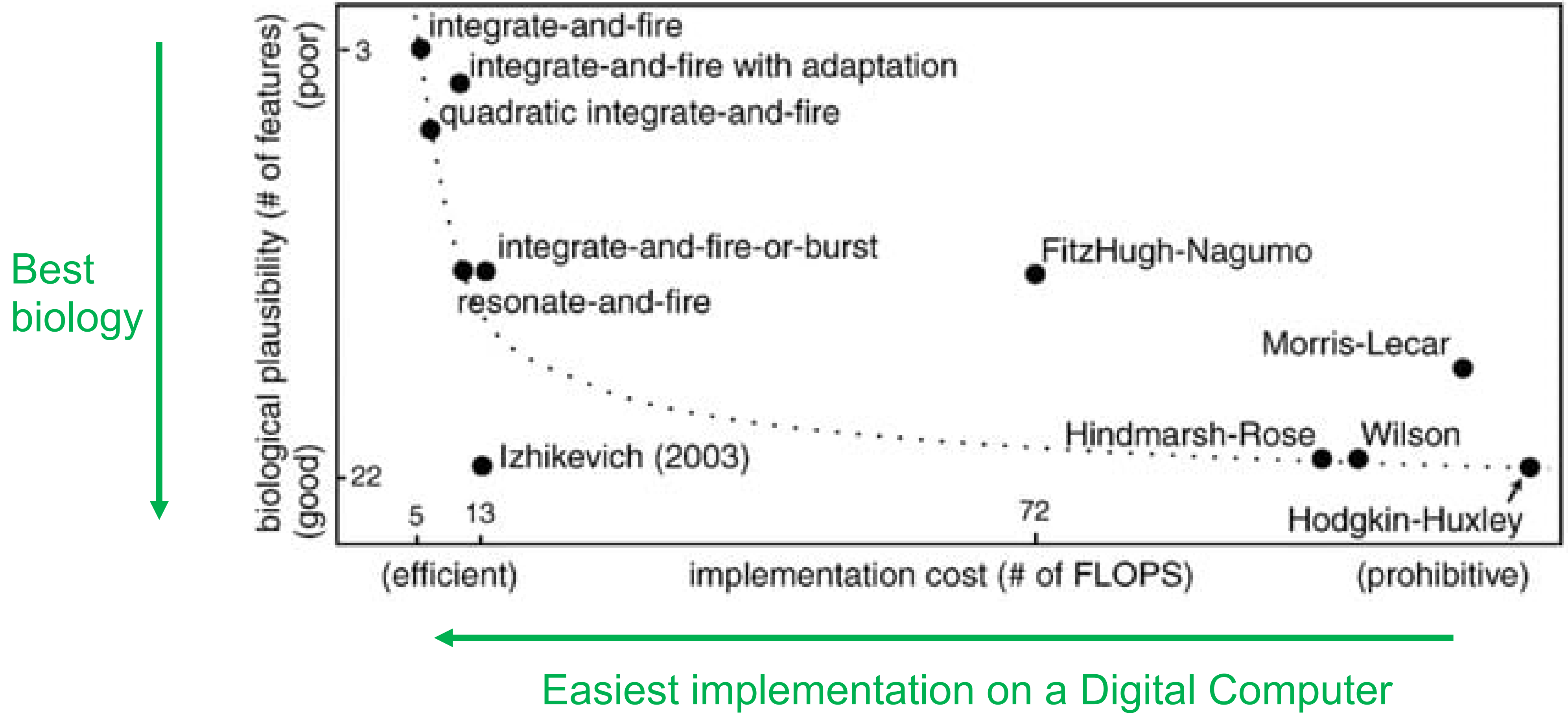


My proposal for valuable computing primitives:

- Complex dynamical neuron circuits (beyond integrate and fire neuron)
- Associative memories and Content-addressable memories

Sebastian, A., Le Gallo, M., Khaddam-Aljameh, R., & Eleftheriou, E. Memory devices and applications for in-memory computing. Nature Nanotechnology 2020

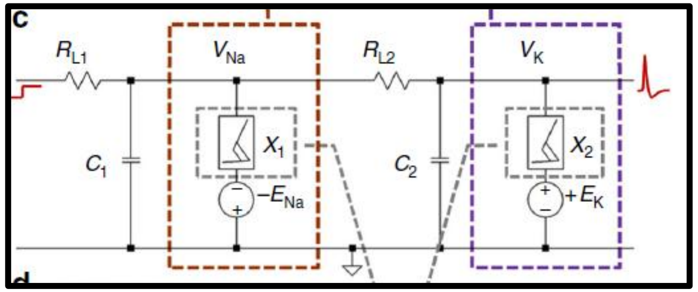
Complex neuron models – trade-off with performance?



Yamazaki, K.; Vo-Ho, V.-K.; Bulsara, D.; Le, N. Spiking Neural Networks and Their Applications: A Review. Brain Sci. (2022).

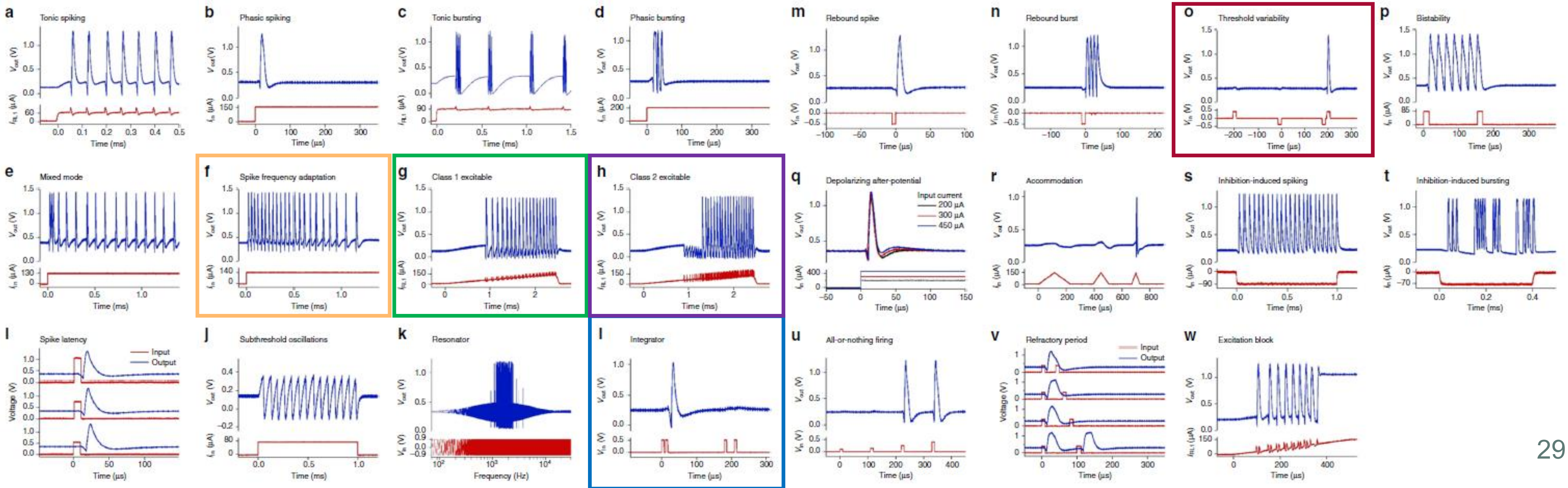
Easy in analog: Mott memristors in Hodgkin-Huxley circuit

Circuit with two VO_2 memristors, two batteries, two capacitors, two resistors.

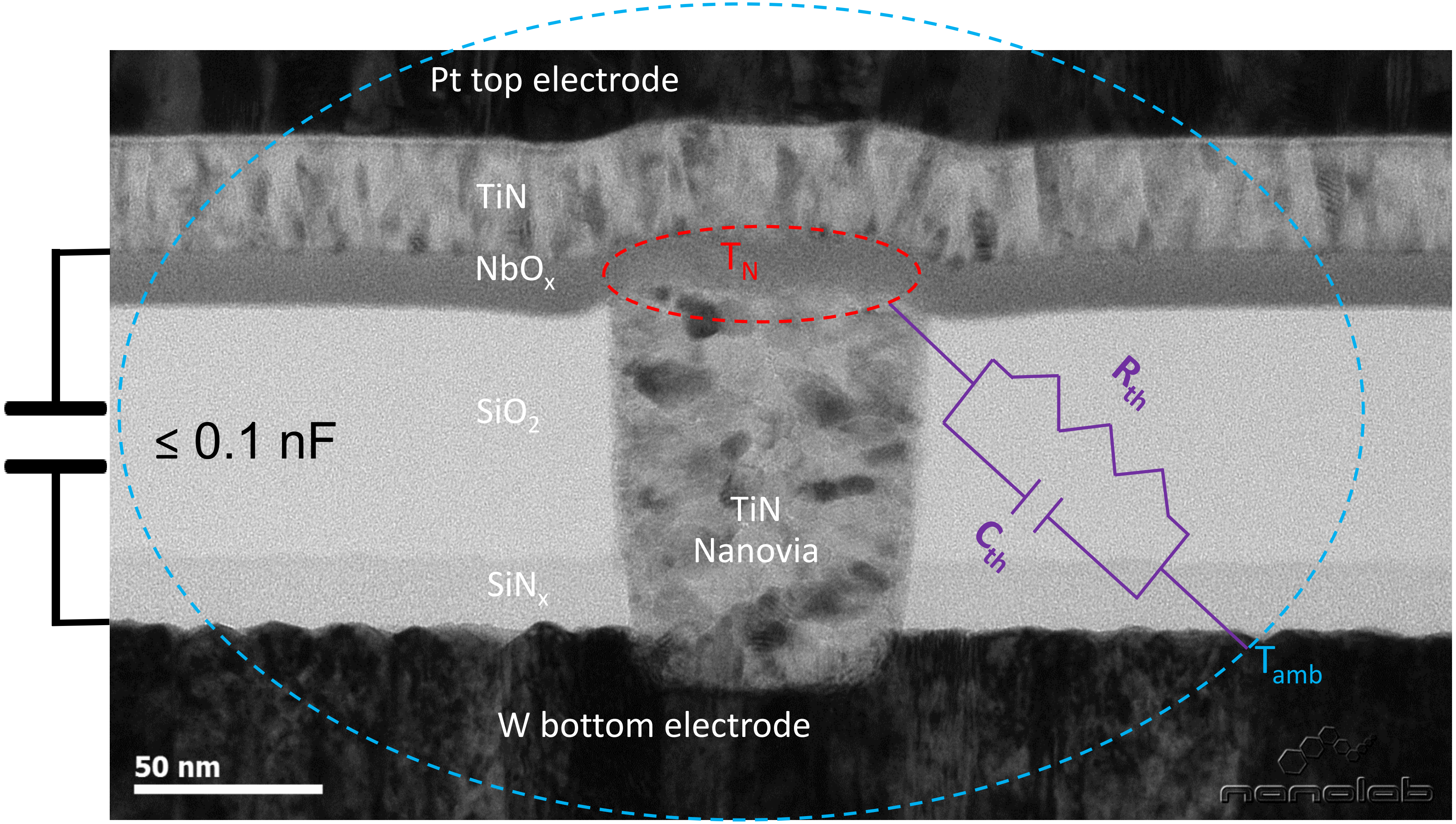


- Experimentally exhibits 23 types of known biological neuron behaviors.
- Potential for performing complex and useful compute operations
- Has local state-variables – type of in-memory conputing

Wei Yi, et al, *Nature Comm.* **9**, 4661 (2018).



Mott Memristor as part of Hodgkin-Huxley Neuron



Can store information in structural phase, ionic configuration, etc

Nanoscale in size, <10 nm

1000 times faster than a neuron

1% of the energy per spike

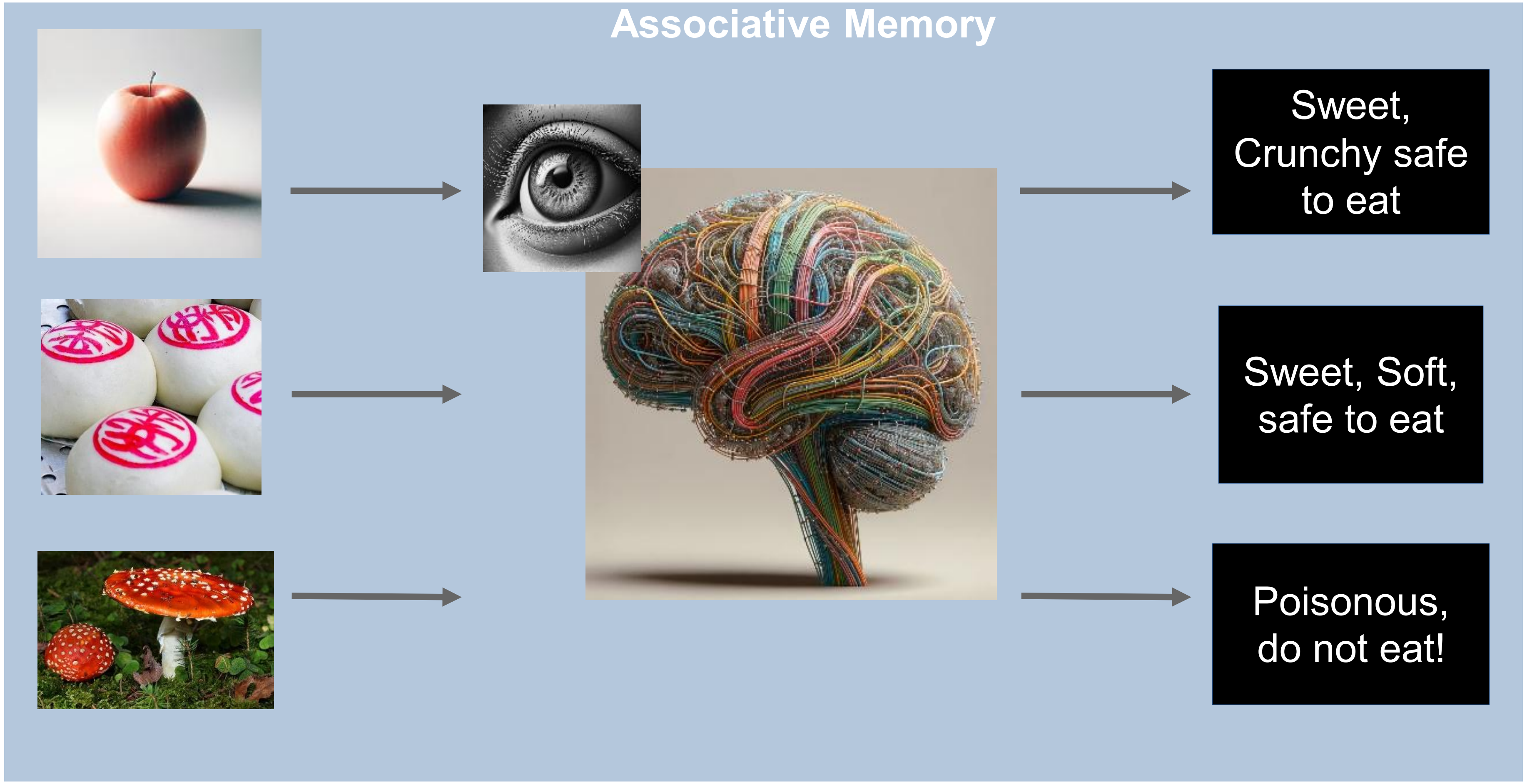
Dark field cross-sectional TEM image of NbO_x memristor. The heated region is thermally connected to T_{amb} through the effective thermal resistance, R_{th}, and thermal capacitance, C_{th}.

Another compute primitive: associative memories

Modern computers use Addressed Memory (RAM): **Input address → Output data**

Memories in brains are very different

Input data **Output / Association**



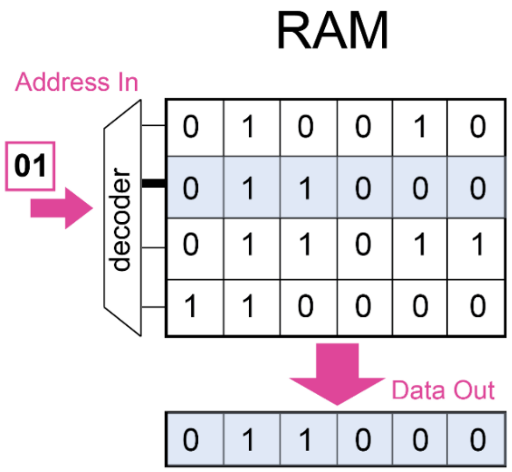
Building Addressed Memory versus Associative Memory

Addressed Memory
RAM (Random Access Memory)

- **Input**: Address, **Output**: Contents at this address

What is needed to build it:

1) Addressing circuitry (periphery)
2) Storage



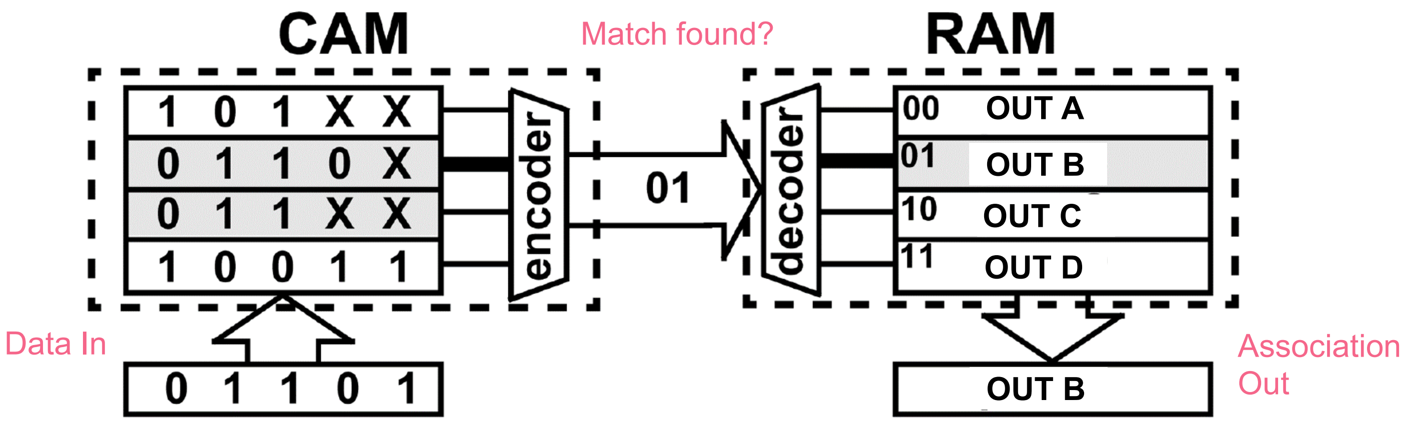
Associative Memory
CAM (Content Addressable Memory) and Ternary CAM

- **Input**: Search word (content), **Output**: address of a match

  Ternary CAMs (**TCAM**) have third state: 0,1, and X = 'don't care'

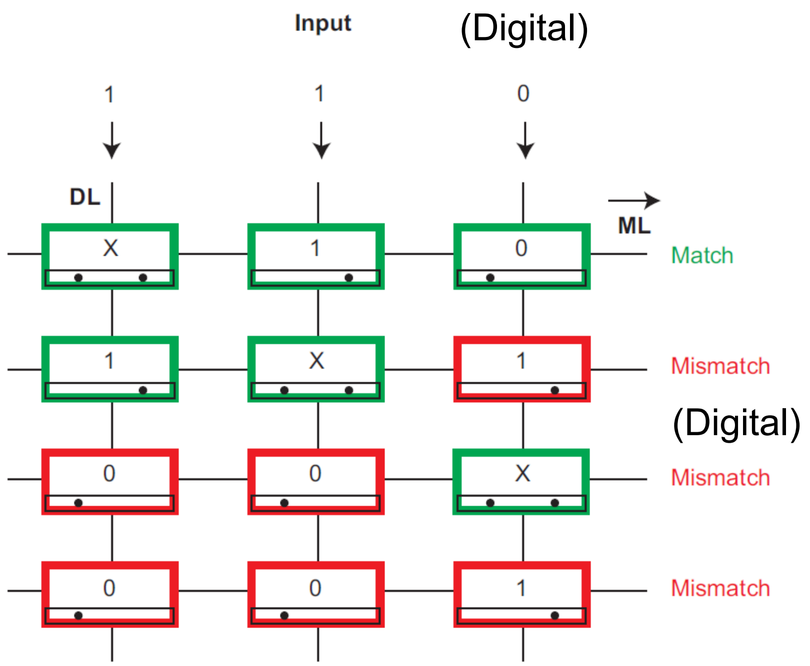
What is needed to build it:

1) Storage
2) Comparison circuitry (at every word)
3) Association circuitry



Our Analog Content Addressable Memory (aCAM) proposal

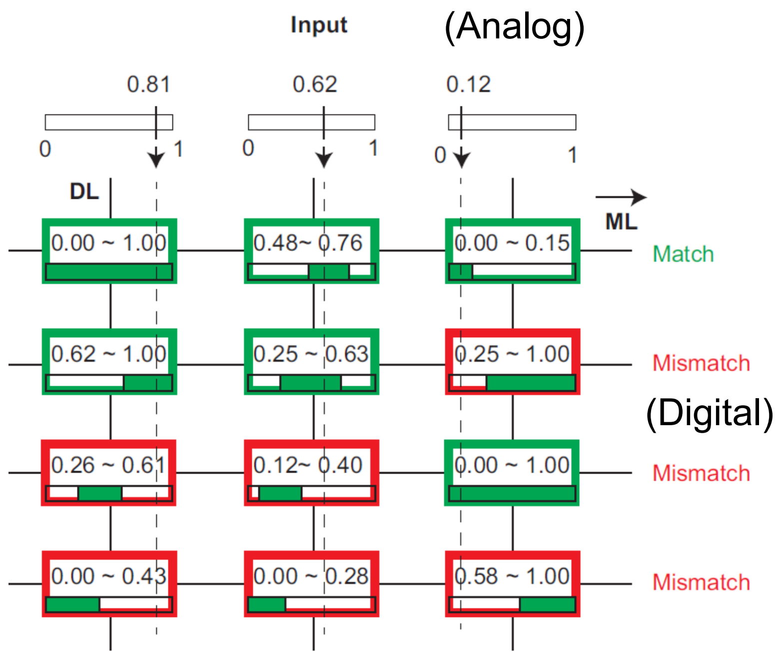
Digital CAM



Digital CAM / TCAM

Stores: 0,1,or 'X'
Input Search: 0,1, or 'X'

Analog CAM (aCAM)



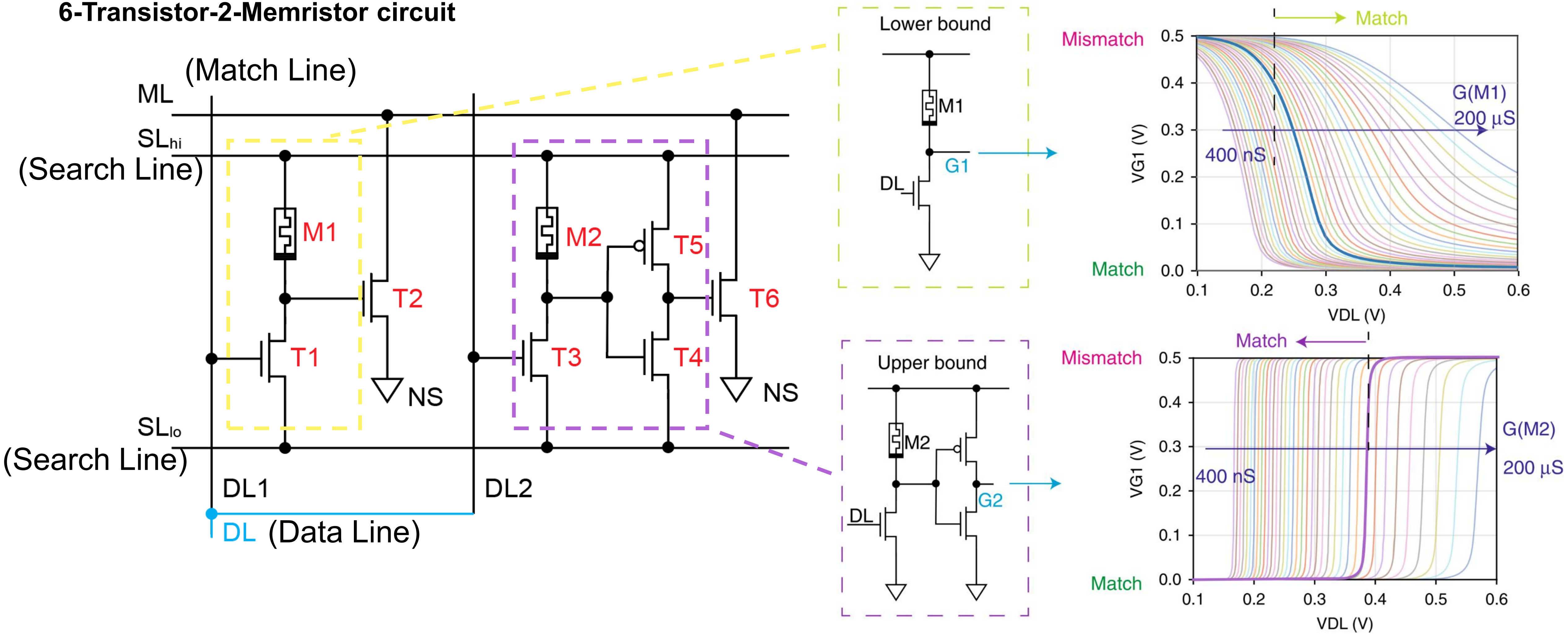
Analog CAM

Stores: analog ranges
Input Search: analog values (multi-bit)

Can Li et al, Nature Communications 11, 1 (2020)

Analog Content Addressable Memory (aCAM)

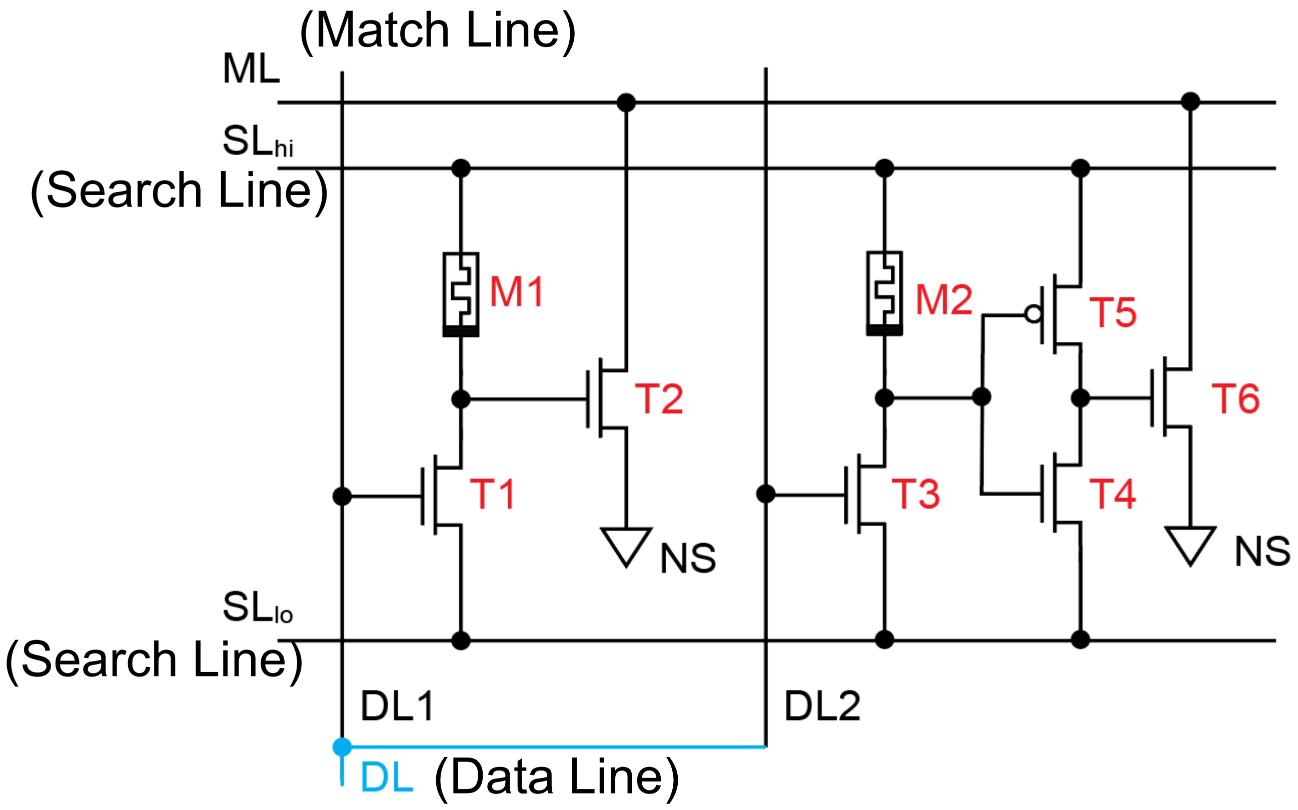
Our implementation of analog CAM
6-Transistor-2-Memristor circuit



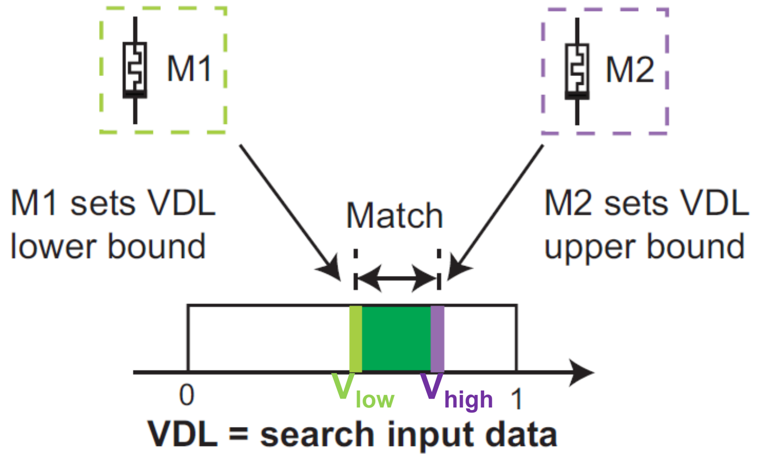
C. Li, et al., *Nature Communications* (2020)

Analog Content Addressable Memory (aCAM)

Our implementation of analog CAM
6-Transistor-2-Memristor circuit



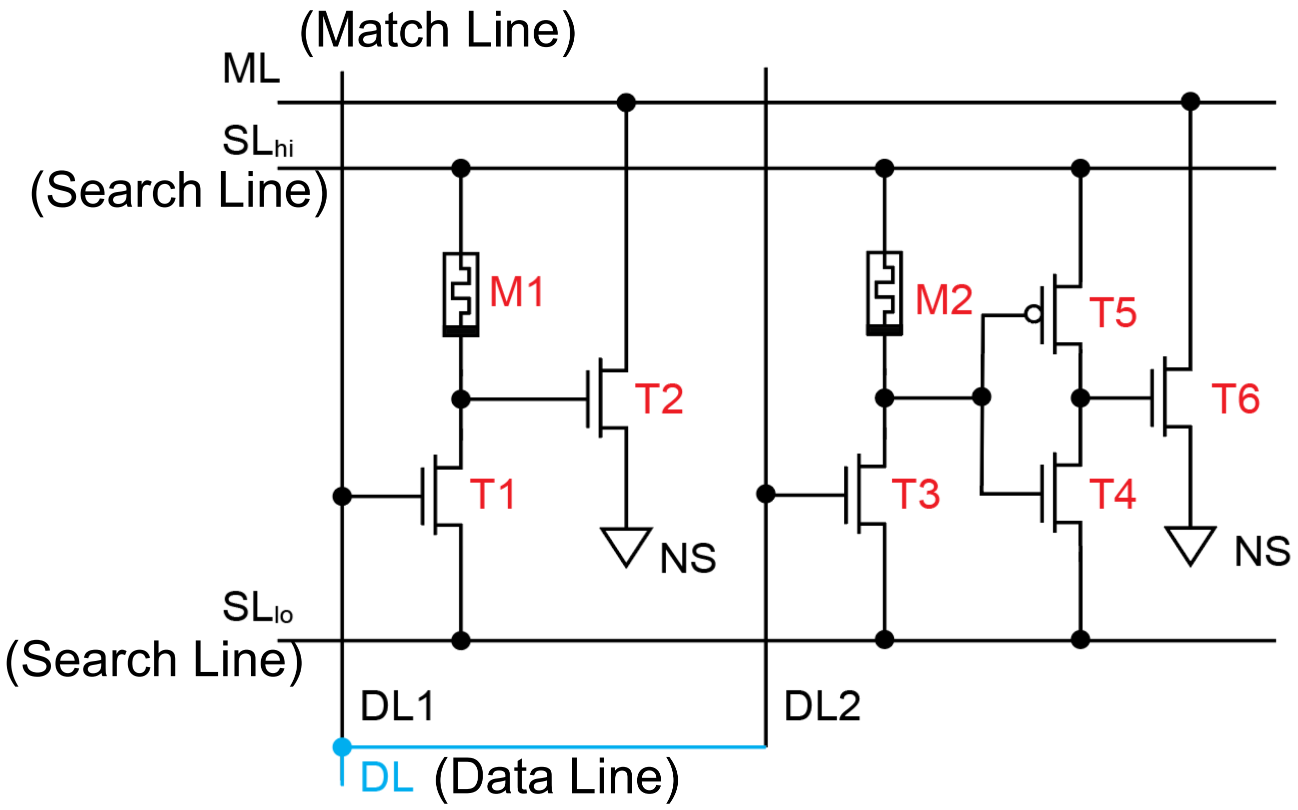
1) Performs inequality test operations
Is V_{low} < Input < V_{high} ?



C. Li, et al., *Nature Communications* (2020)

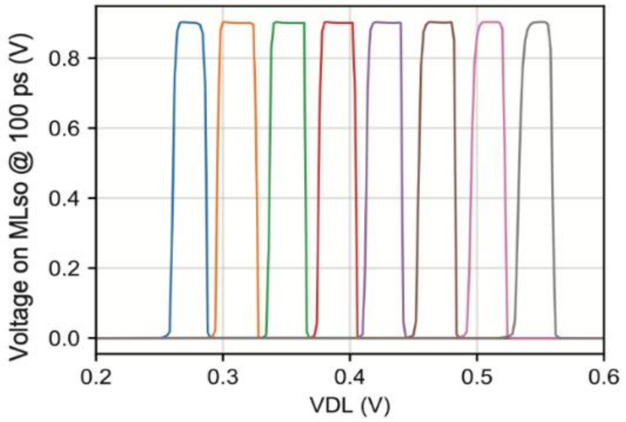
Analog Content Addressable Memory (aCAM)

Our implementation of analog CAM
6-Transistor-2-Memristor circuit

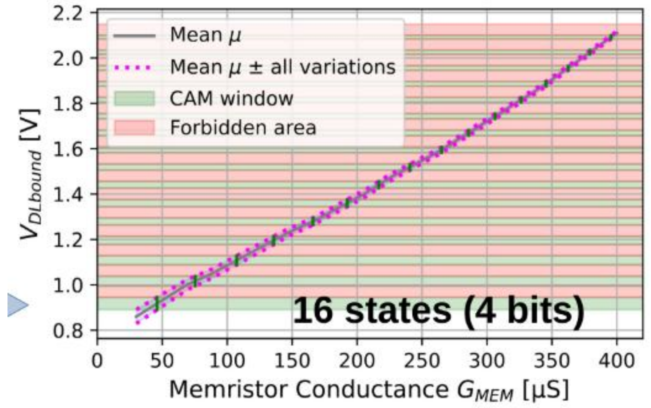


C. Li, et al., *Nature Communications* (2020)

2) Can operate as a multi-bit CAM



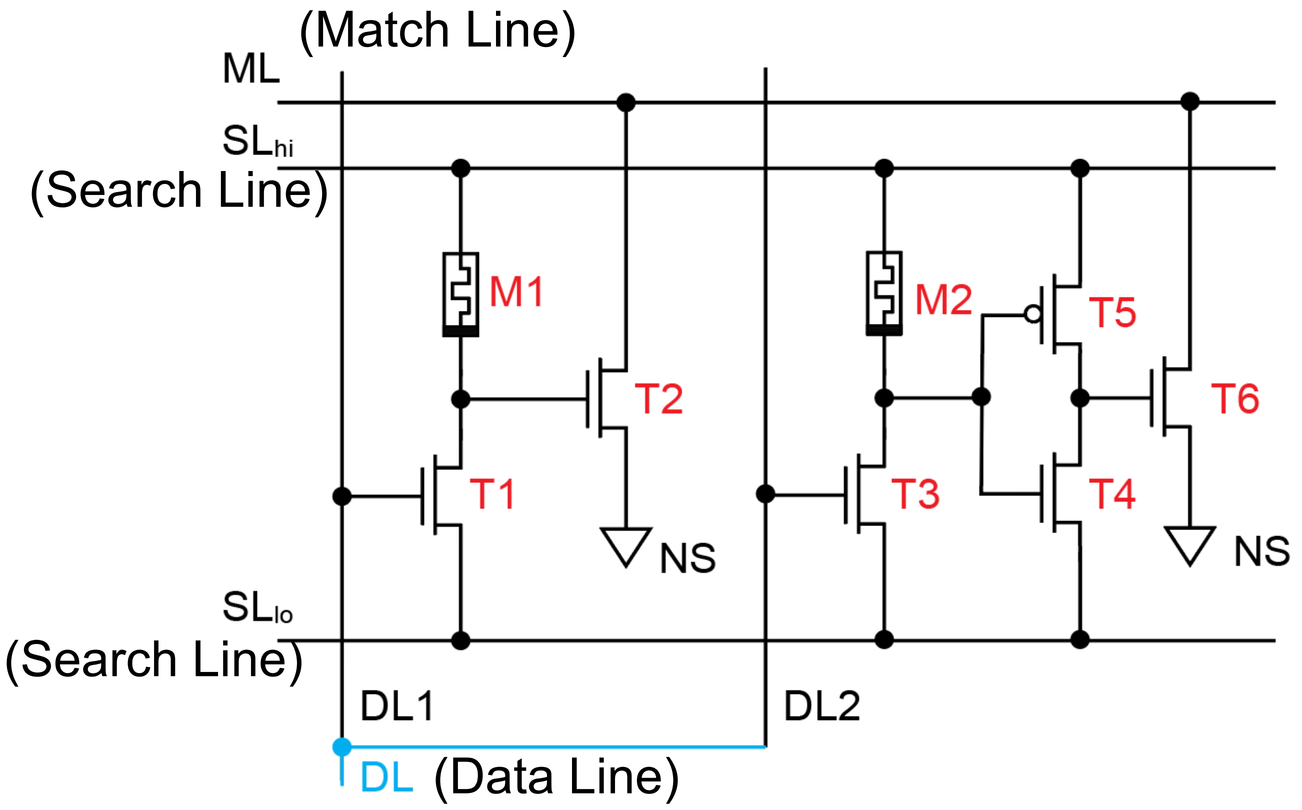
Showed up to 16 levels (4 bits)



P. Manea, C. Sudarshan, F. Cueppers, J.P. Strachan, *NanoARCH* (2023); *Best paper award*

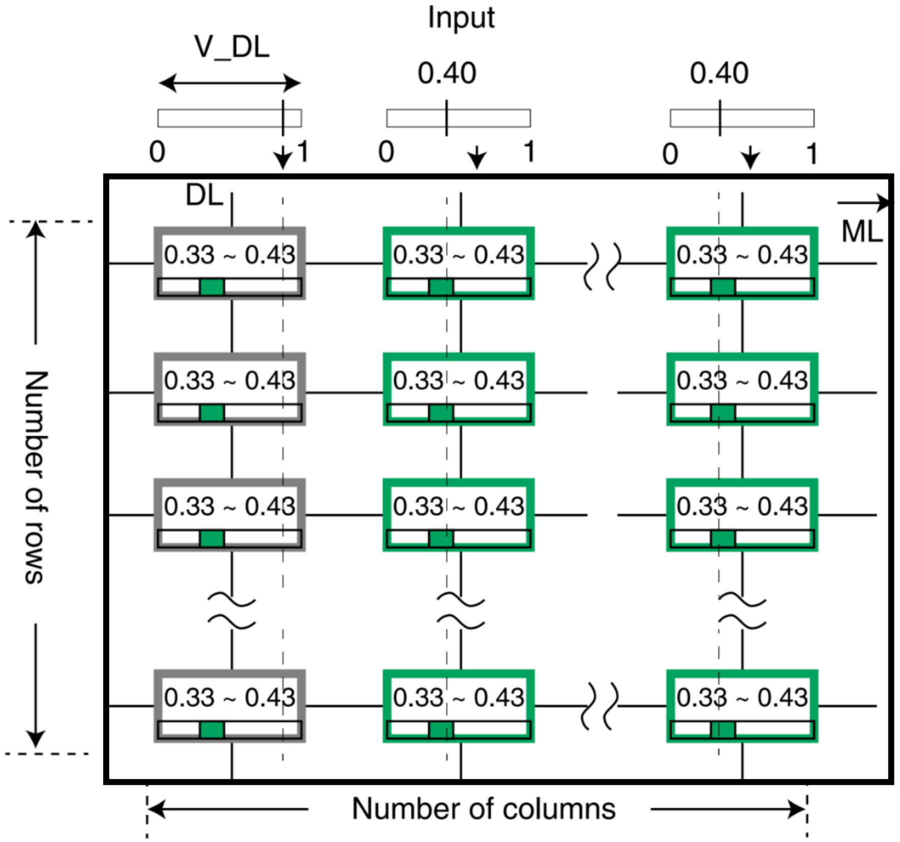
Analog Content Addressable Memory (aCAM)

Our implementation of analog CAM
6-Transistor-2-Memristor circuit

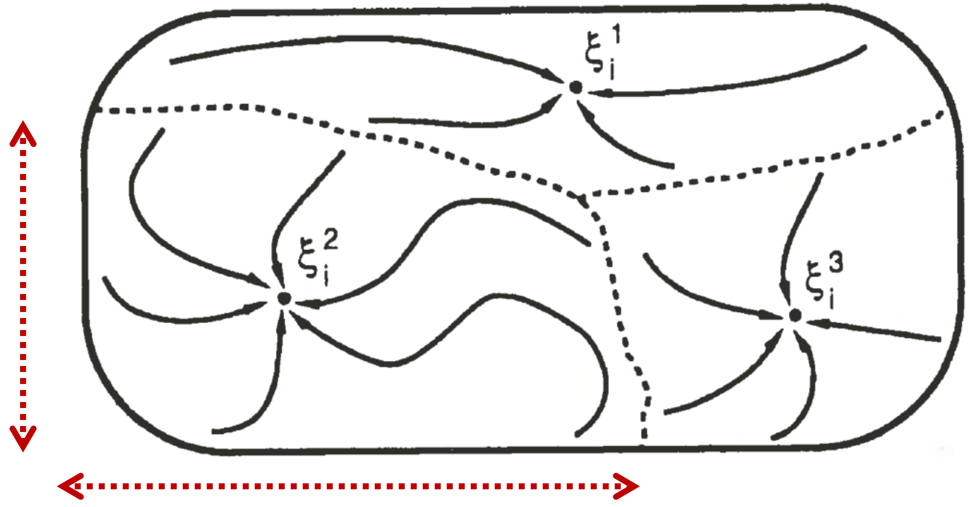


C. Li, et al., *Nature Communications* (2020)

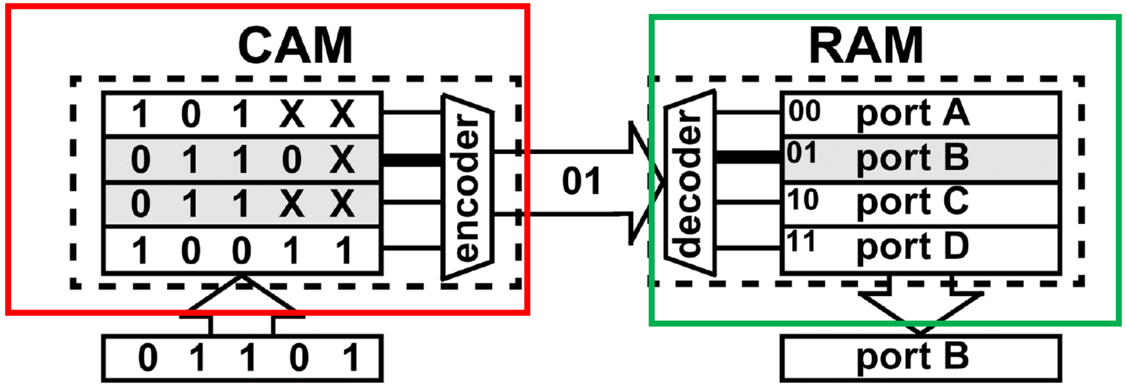
aCAM cells laid out in an array.
Allows high dimensional data to be stored as words of "intervals"



Connection to Hopfield Networks and (dense) Modern Hopfield Networks



=

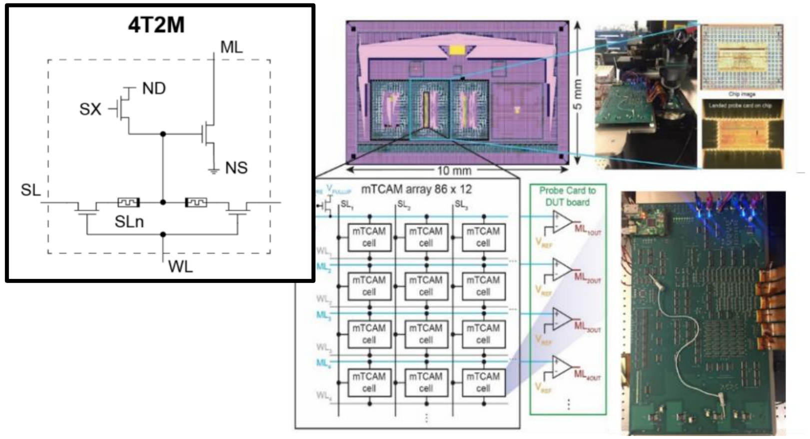


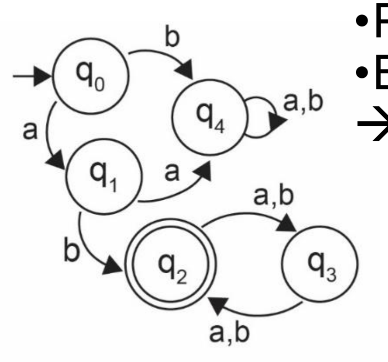
- ξ^1, ξ^2, ξ^3 are the attractor states
- Regions bounded by the dashed lines are the basins of attraction

- Analog CAM intervals can match (high-dimensional) boundaries of the basins of attraction
- Activates a RAM with the stored (clean) attractor states
- Exponential capacity
- No dynamics needed: single step look-up, then memory activation

(analog) Associative Memories enable many applications

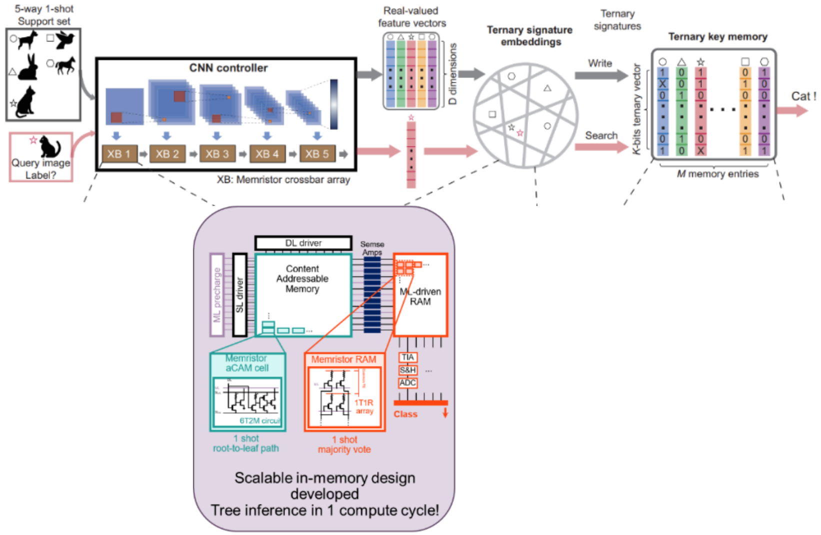
We have built prototype Associative Memories (180nm CMOS), testing multiple applications





Associative Memories implement **Finite State Machines**
•FSM has applications in Genomics, Network Security, etc
•Put FSM state transitions in a rapid-lookup CAM
•Benchmarked versus state-of-the-art FPGA
→25x improved Throughput/Watt, reduced chip area/cost

C. E. Graves, *et al*, **NANOARCH** (2018);
C. E. Graves, *et al*, **ICRC** (2018);
C. E. Graves, *et al*, **IEEE TNano**, (2019)
C. E. Graves, *et al.*, **Adv. Mater** (2020)



Machine Learning applications: Decision Trees, Random Forests

Tested on traffic-sign classification task

- **914x higher Throughput** (Decisions/second) over digital ASIC
- **15x lower Energy per Decision** over digital ASIC

G. Pedretti, C. Graves, S. Serebryakov, R. Mao, X. Sheng, M. Foltin, C. Li, J.P. Strachan, *Nature Communications (2021)*

Few-shot Learning application:

Mao, Ruibin, et al. "Experimentally realized memristive memory augmented neural network." *Nature Communications (2022)*

Modified analog CAM for Transformers

Combine

- Memristor crossbar arrays for Feed-Forward Network Layers (~75% computations)
- Modified Dynamic aCAM based Attention

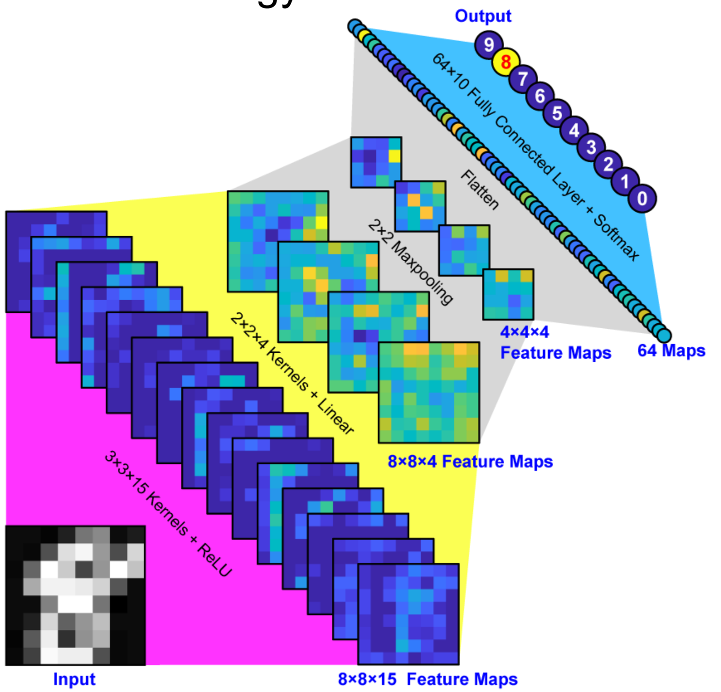
Nathan Leroux

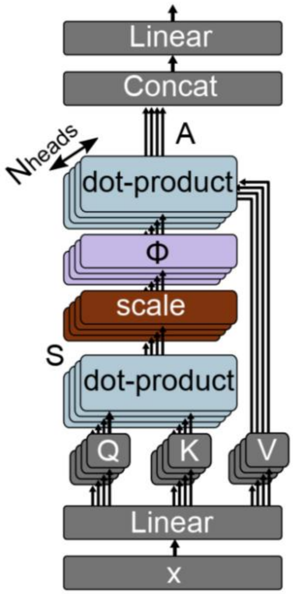
Paul Manea

Feed-forward Linear Layers

Compute in Memristor device arrays

Offers low-energy read

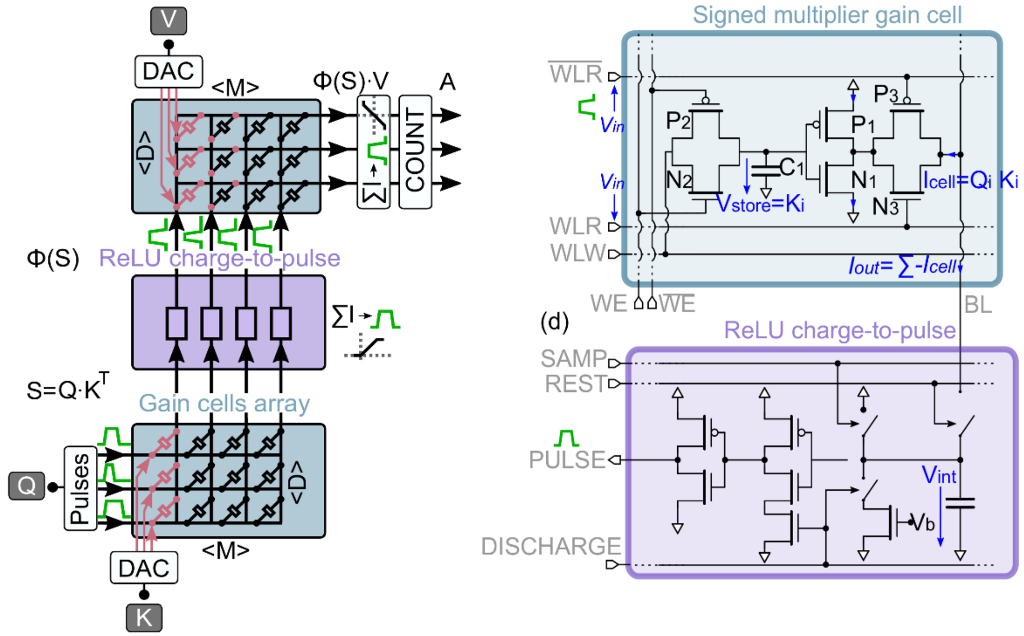




Attention layers

Compute in Capactive gain-cell arrays

Lower energy & fast re-programming



Leroux, N., Manea, P. P., Sudarshan, C., Finkbeiner, J., Siegel, S., Strachan, J. P., & Neftci, E. *"Analog In-Memory Computing Attention Mechanism for Fast and Energy-Efficient Large Language Models."* (2025), *Nature Comp. Sci.*

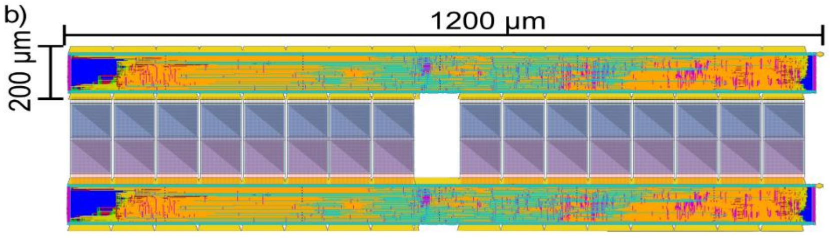
Modified analog CAM for Transformers


Nathan Leroux

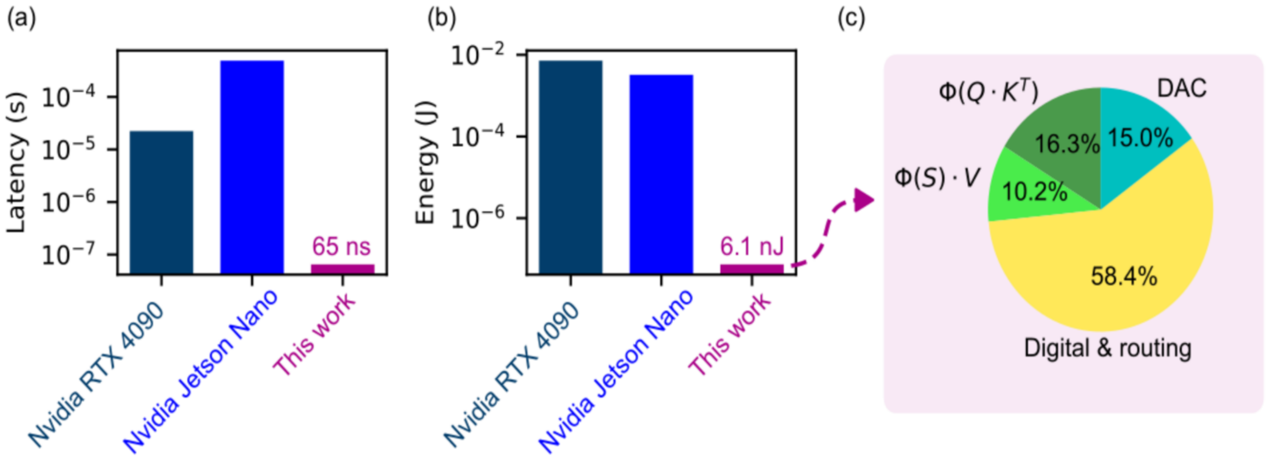

Paul Manea

Combine

- Memristor crossbar arrays for Feed-Forward Network Layers (~75% computations)
- Modified Dynamic aCAM based Attention



Deployed on GPT2 sized LLMs (>1 Billion)
Retrained LLM with Spice-based circuit characteristics
Full accuracy reached
Inference Latency – 65ns, 6.1 nJ



Leroux, N., Manea, P. P., Sudarshan, C., Finkbeiner, J., Siegel, S., Strachan, J. P., & Neftci, E. "Analog In-Memory Computing Attention Mechanism for Fast and Energy-Efficient Large Language Models." (2025), Nature Comp. Sci,

Thank you!

Jülich / RWTH Aachen

Paul Manea
Nathan Leroux
Mohammad Hizzani
Arne Heittmann
Ming-Jay Yang
Emre Neftci
Dmitrii Dobrynin
Jan Finkbeiner
Sebastian Siegel
Chirag Sudarshan

UCSB

Dima Strukov
George Hutchinson
Tinish Battacharya

HP Labs

Giacomo Pedretti
Masoud Mohseni
Fabian Bohm
Thomas Van Vaerenbergh
Jim Ignowski
Ray Beausoleil

Key Collaborators

Can Li (Univ HK)
Catherine Graves (Google DeepMind)
J. Joshua Yang (USC)
Qiangfei Xia (UMass Amherst)
Wei Lu (U Michigan)
Shimeng Yu (Georgia Tech)