Neuromorphic Computing with Resistive Synaptic Arrays: Devices, Circuits and Systems

Yu (Kevin) Cao, Shimeng Yu, Jae-sun Seo

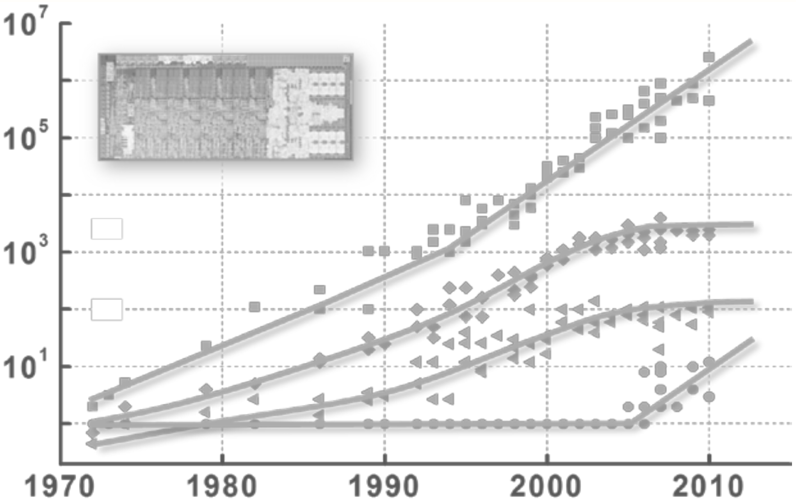
School of ECEE, Arizona State University

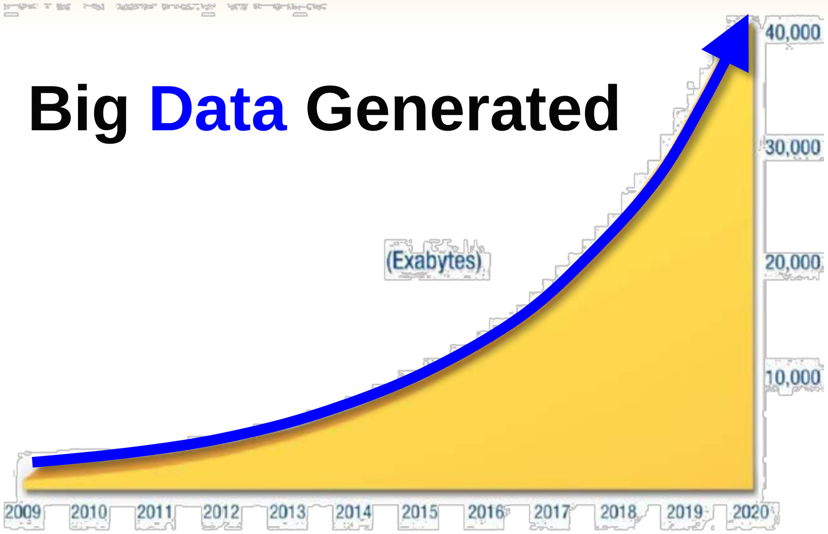
Outline

- Learning On-a-chip:

  Synaptic Devices and the Crosspoint Array
- Non-ideal **Device** Effects on Learning Accuracy
- Peripheral **Circuits** and Parallel Operation
- A **System**-level Benchmark Simulator
- Summary and Discussion

From Data to Information




Big Data Generated

Useful If Tagged and Analyzed 23%

Tagged 3%

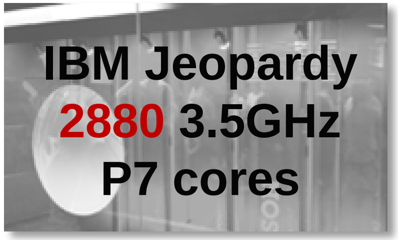
Analyzed 0.5%

Big Gap in Information Analysis!

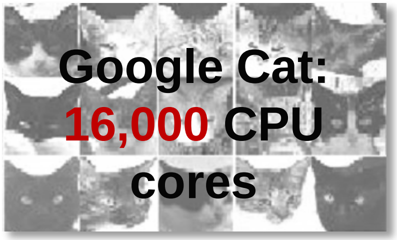
[IDC, December 2012]

Learning On-a-chip

- Deep learning in the cloud: expensive **computation**, **huge training** data, low **energy** efficiency, high precision



IBM Jeopardy
2880 3.5GHz
P7 cores



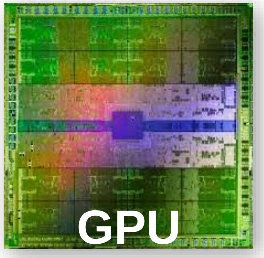
Google Cat:
16,000 CPU
cores

- Edge computing needs novel hardware / algorithms
  - **Local** to the sensor, **real-time**, **reliable**, low-power
  - **On-line**, personalized learning with continuous data

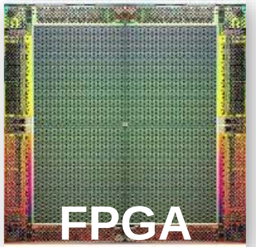
30
frames/s

Acceleration Need

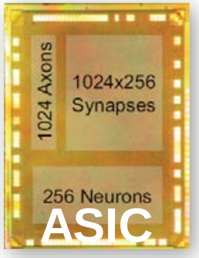
- 10^3 – 10^5 speedup required to achieve real-time training of HD images at 30 frames/second



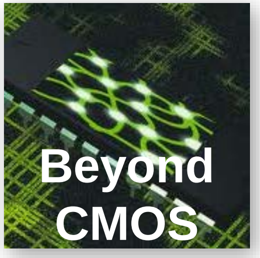
10 – 30 X



10 – 50 X

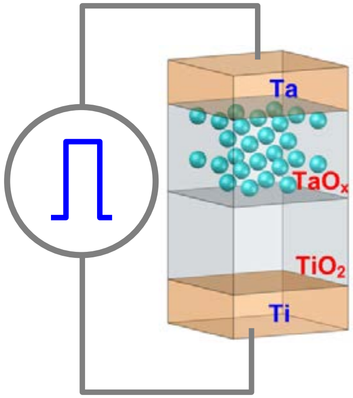


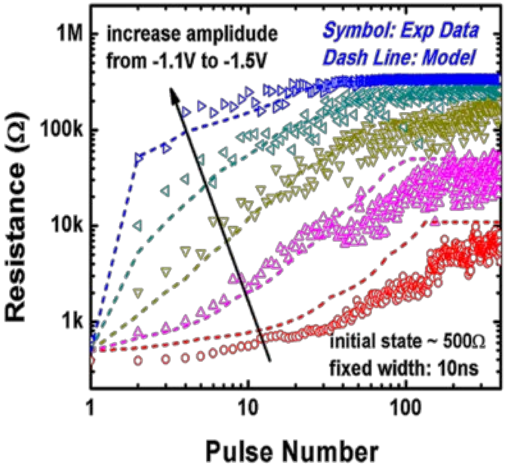
10^2 – 10^3 X

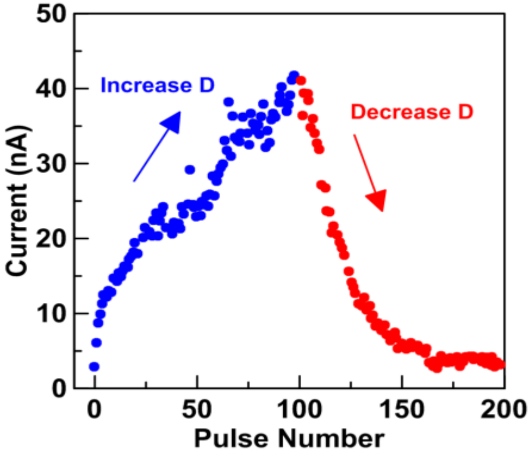


>10^3 X

- Device beyond CMOS: RRAM to emulate the synapse



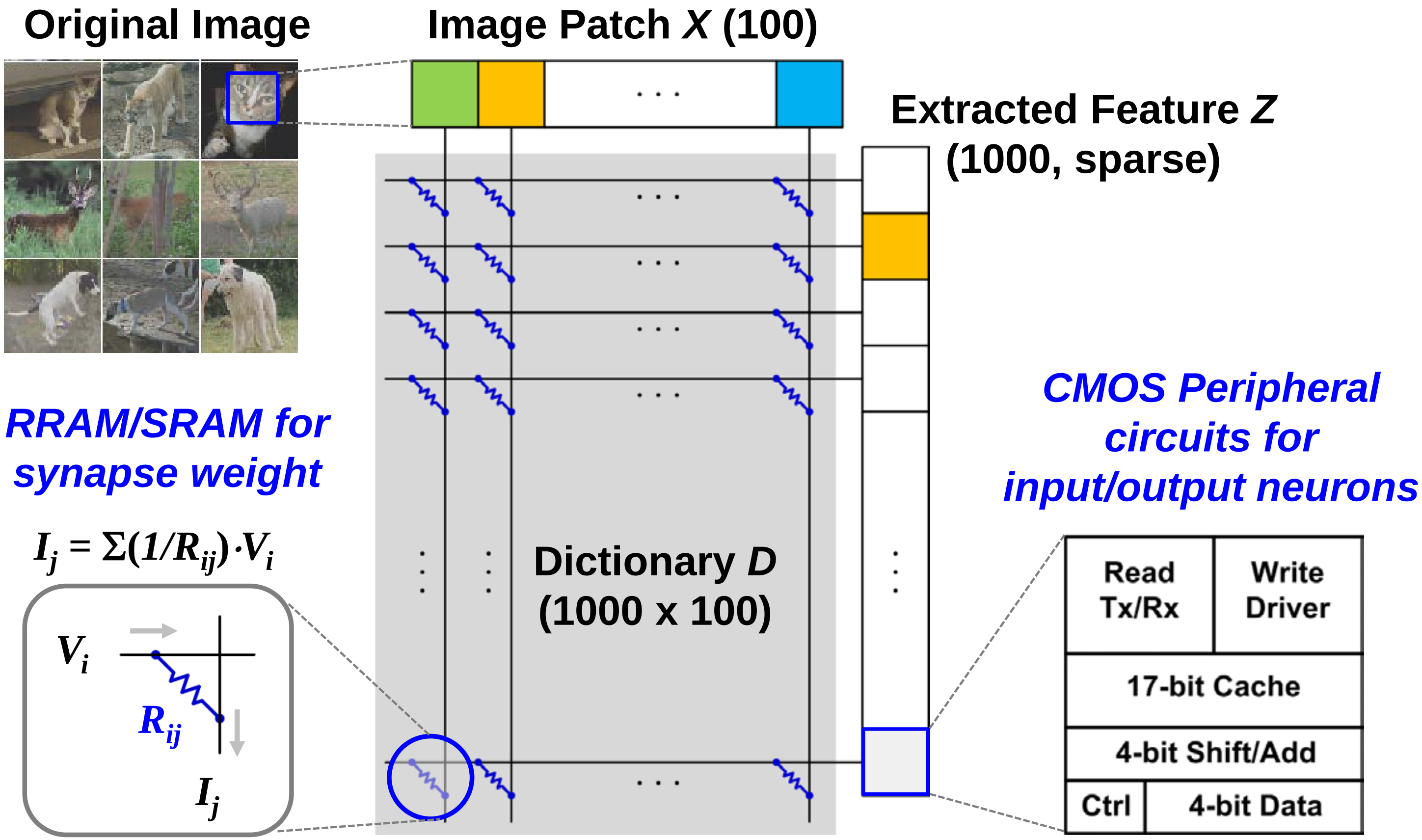




[S. H. Jo *et al*., Nano Letter 2009]

Resistive Crosspoint Array

- A biomimetic solution: RRAM for synapse, crosspoint for dense interconnection; not necessarily spiking neurons



Original Image

Image Patch *X* (100)

Extracted Feature *Z* (1000, sparse)

RRAM/SRAM for synapse weight

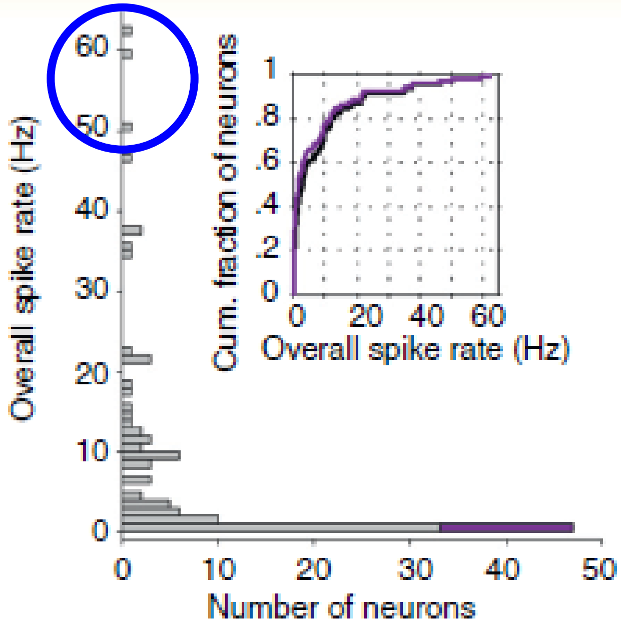
$I_j = \Sigma(1/R_{ij}) \cdot V_i$

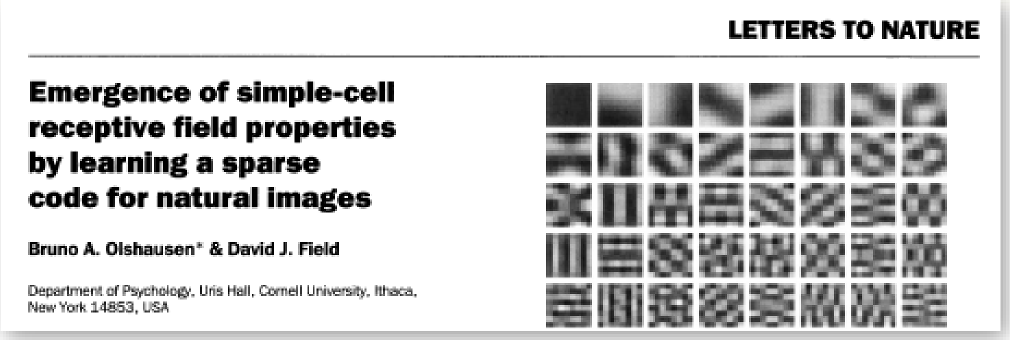
Dictionary *D* (1000 x 100)

CMOS Peripheral circuits for input/output neurons

Read Tx/Rx	Write Driver
17-bit Cache	
4-bit Shift/Add	
Ctrl	4-bit Data

Sparse Coding





LETTERS TO NATURE

Emergence of simple-cell receptive field properties by learning a sparse code for natural images

Bruno A. Olshausen* & David J. Field

Department of Psychology, Uris Hall, Cornell University, Ithaca, New York 14853, USA

$$\min_{D,Z}\frac{1}{n}\sum_{i=1}^{n}\left(\underset{\textbf{Reconstruction Error}}{\underline{\frac{1}{2}\parallel D\cdot Z_i - x_i \parallel^2}} + \underset{\textbf{Sparseness}}{\underline{\lambda|Z_i|_1}}\right)$$

- High power efficiency
- No backward propagation
- Scalable to multi-layers

X: input vector

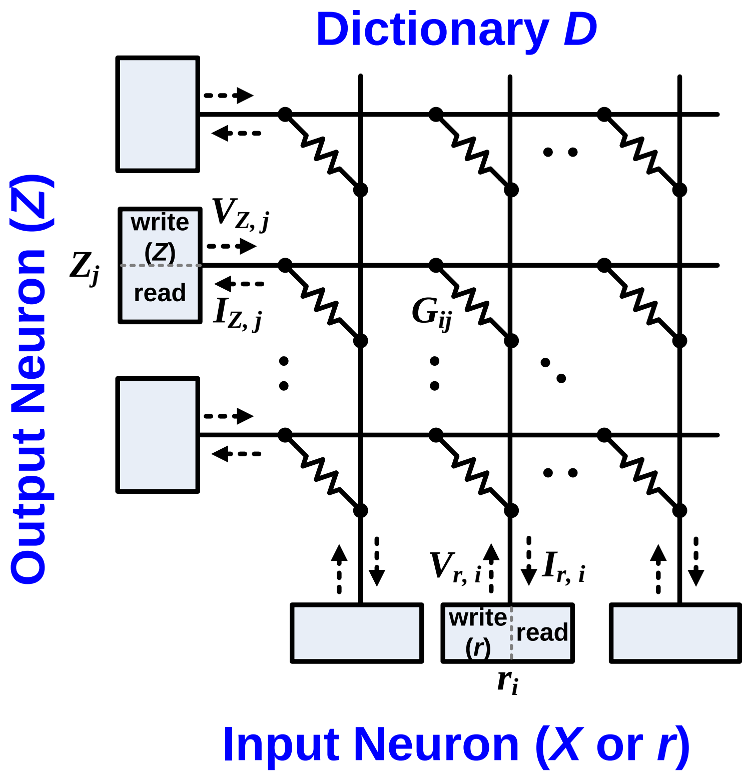
Z: feature vector (output)

D: dictionary (weight matrix)

[D. H. O’Connor *et al*., Neuron 2010; B. A. Olshausen, D. J. Field, Nature 1996]

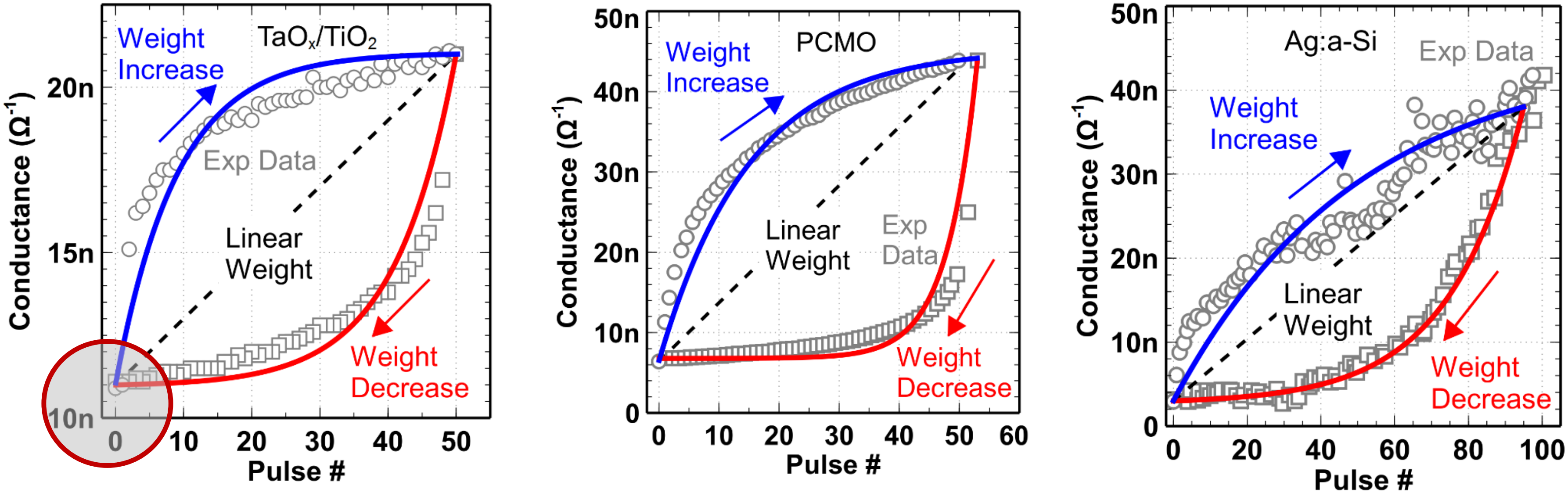
Analog Memory and Computing

- All cells are DC connected, no sneak path for read
- The value of *Z*, *X* (or *r*) represented by the number of voltage pulses; *D* by the RRAM conductance



Task	Operations
$\boldsymbol{D \cdot Z}$	$I_{r,i} = \sum_i G_{ij} \cdot V_{Z,j}$
$\boldsymbol{D^T \cdot r}$	$I_{Z,j} = \sum_j G_{ij} \cdot V_{r,ij}$
D **update**	$\Delta G_{ij} = \eta \cdot r \cdot Z$

Realistic Device Properties

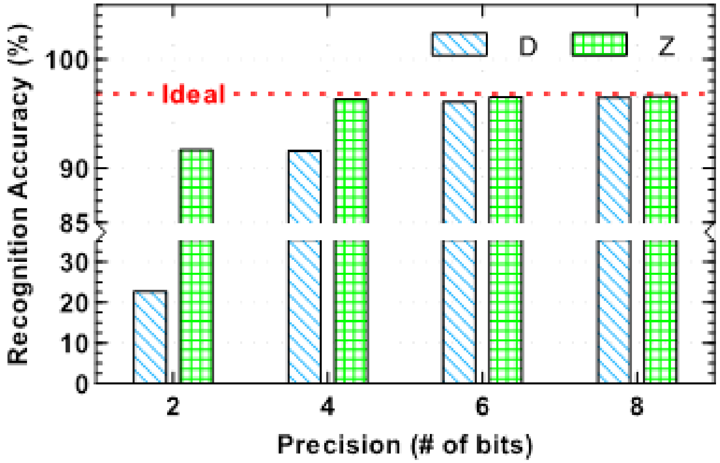


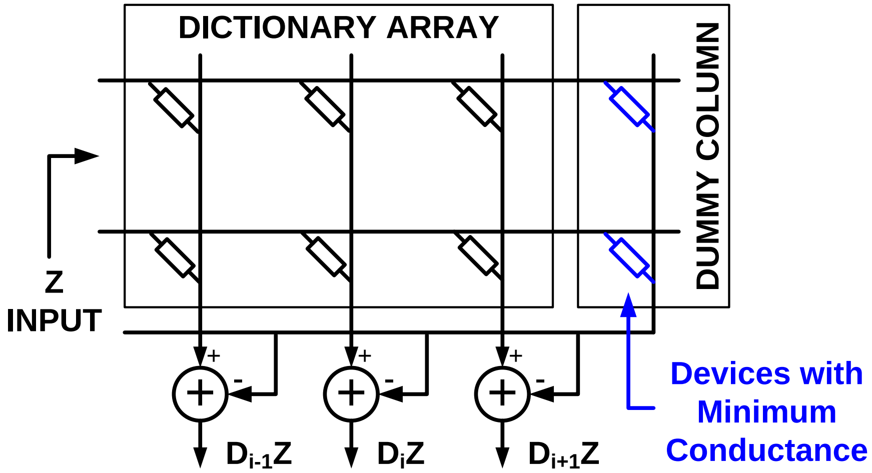
- Non-zero off-state conductance; limited levels / precision
- Device variations; nonlinearity in weight update
- Experiment with unsupervised **sparse coding + MNIST** to study their impact on learning accuracy

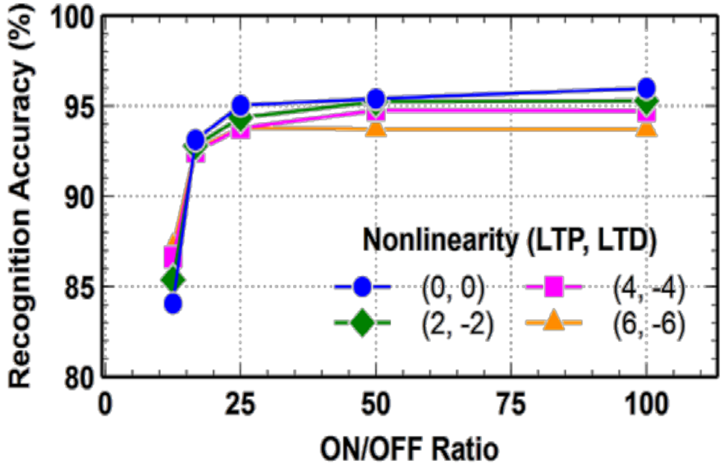
[B. A. Olshausen, D. J. Field, Nature 1996]

Non-zero Off-state and Precision

- Solution: spatial redundancy to solve non-zero off-state
- Fixed-point computing
  - Weight (*D*): **6 bits (64 levels)**
  - Output (*Z*): 4 bits
  - On/off ratio needs to be > 25

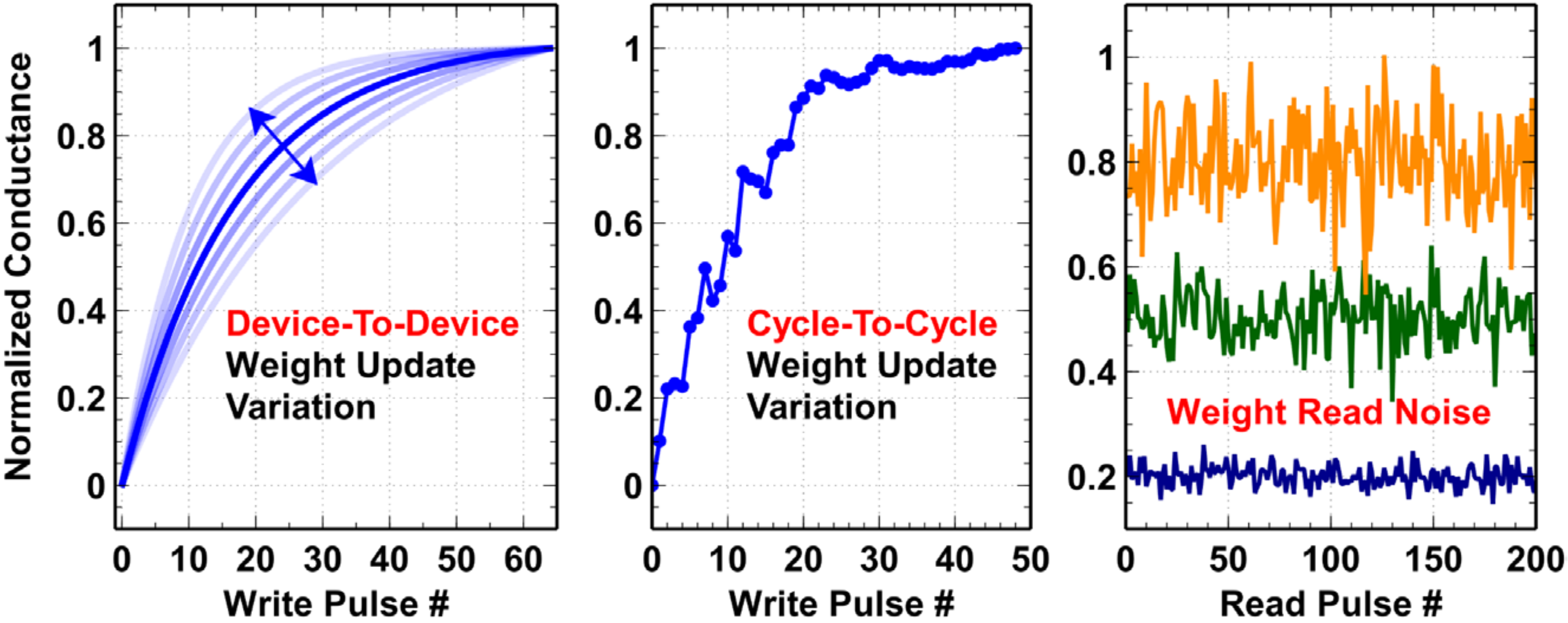




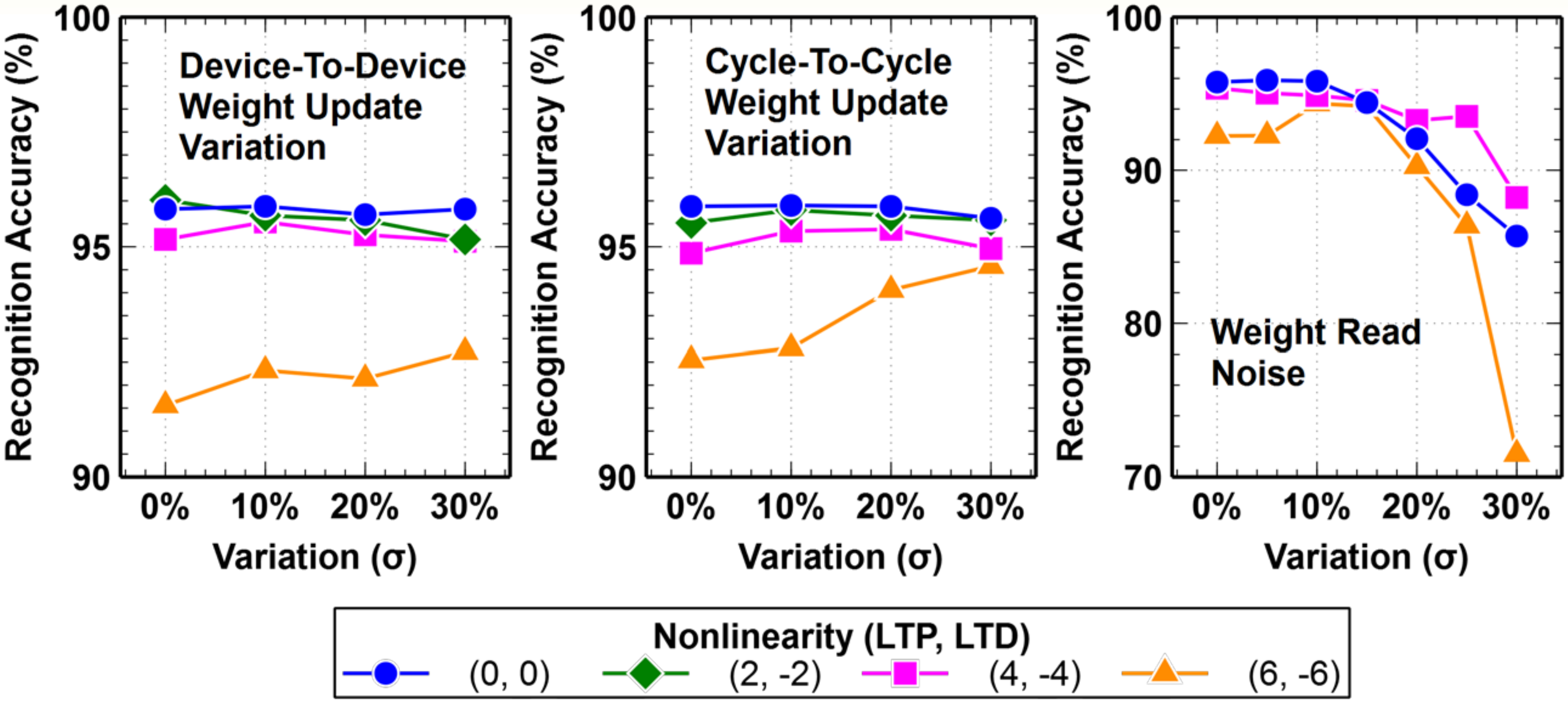


Device Variations

- Weight update variation: device-to-device and cycle-to-cycle
  - Device nonlinearity has moderate impact on the accuracy
- Weight read noise



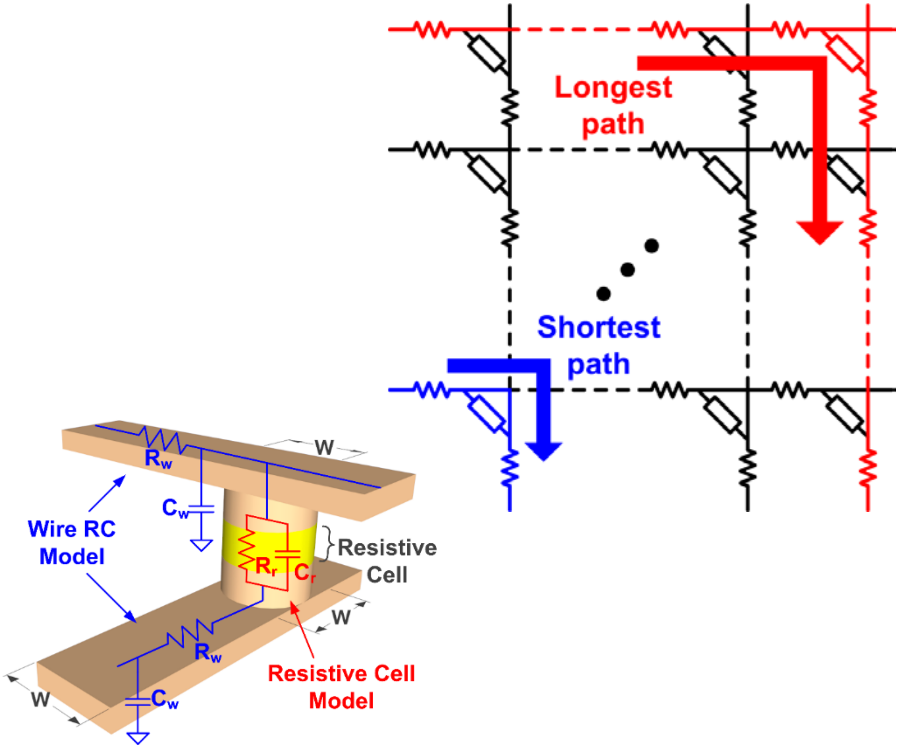
Impact on the Accuracy

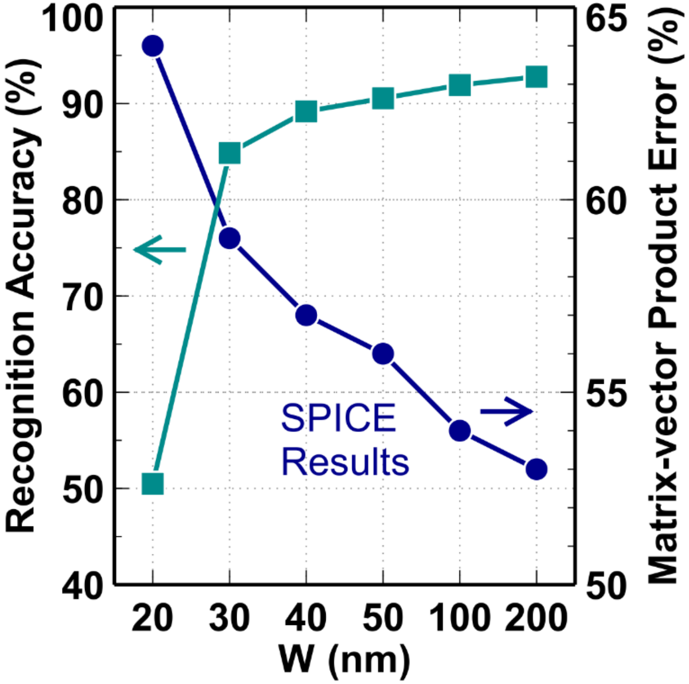


- Impact of weight update variation: **moderate**
- Impact of weight read noise: **significant**
- Solution: multiple cells to minimize the variation

Interconnect Resistance

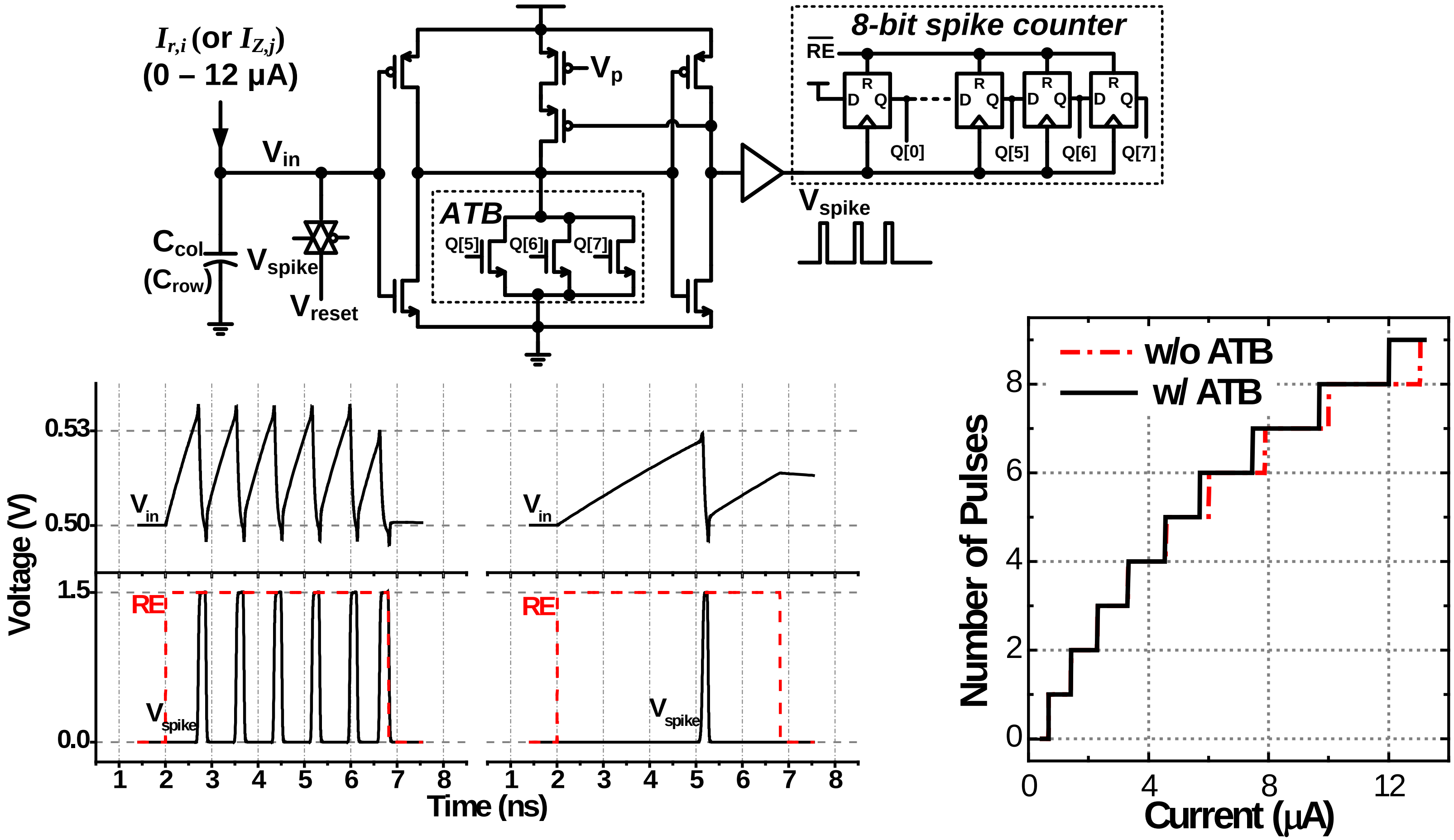
- Wire resistance is in series with RRAM resistance
  - RC delay is not an issue
- Solution: scaling up the wire





Neuron Circuits: Parallel Read

- A **current-to-digital converter**, operating as the Integrate-and-Fire neuron model

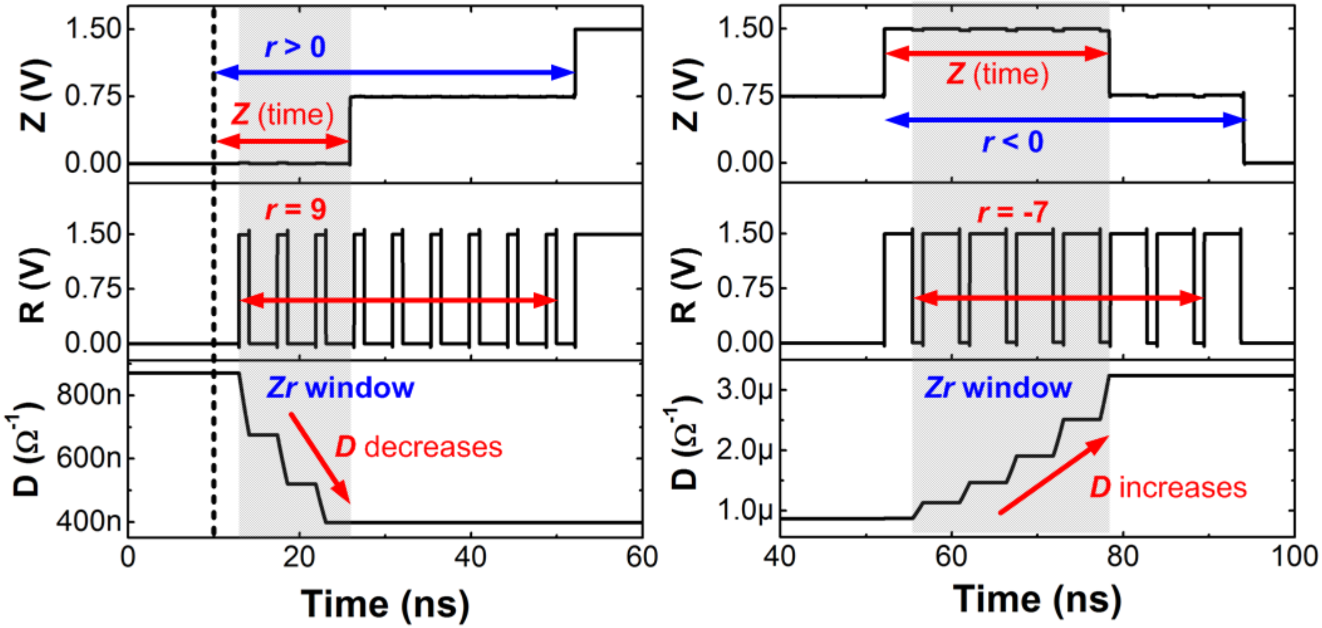


Neuron Circuits: Parallel Write

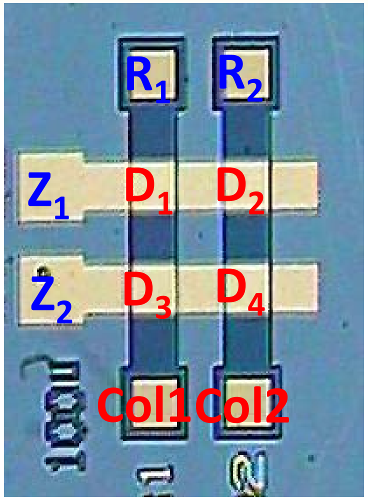
- Write RRAM through the **spiking rate** between input (X or r) and output (Z) neurons

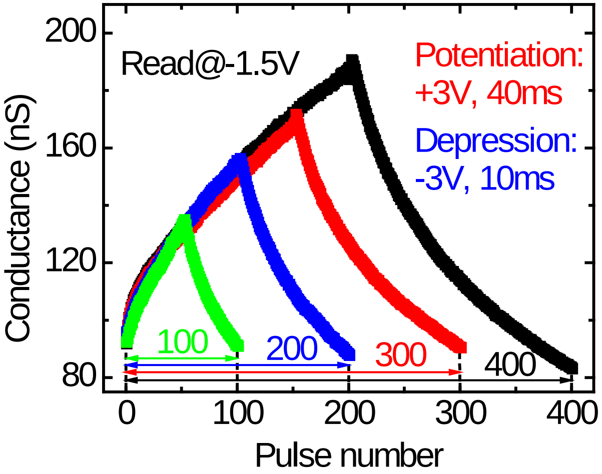
$$\Delta G_{ij} \propto pulse\ width = \boldsymbol{Write\ Time} \cdot \boldsymbol{Firing\ Rate} = \eta \cdot Z \cdot r$$

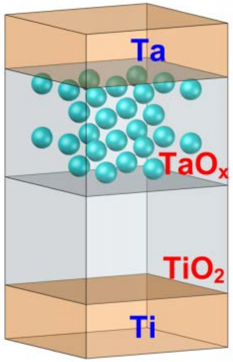
  - *Z* value for the time window to write
  - *r* value for the pulse number (firing rate)

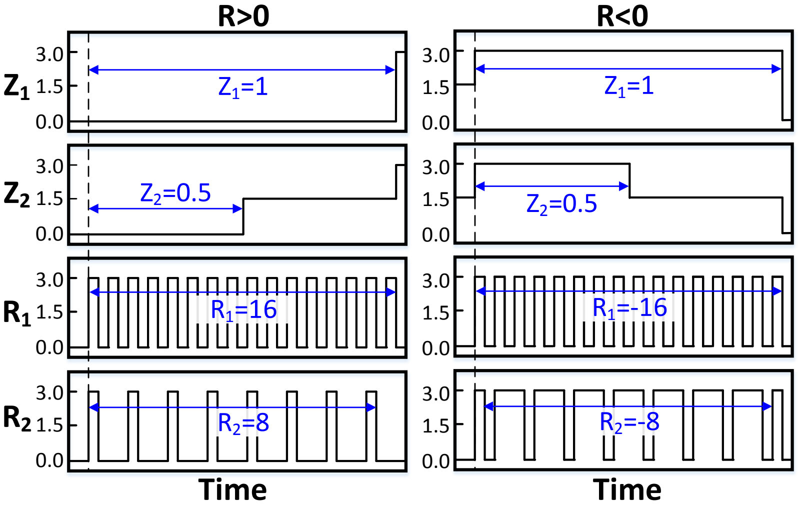


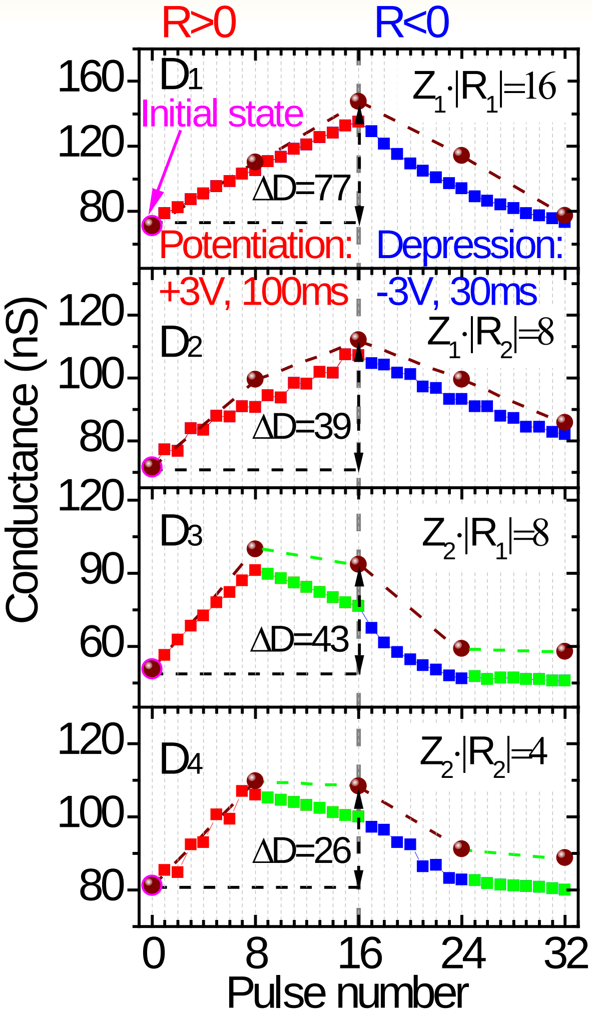
Parallel Operation: O(1)





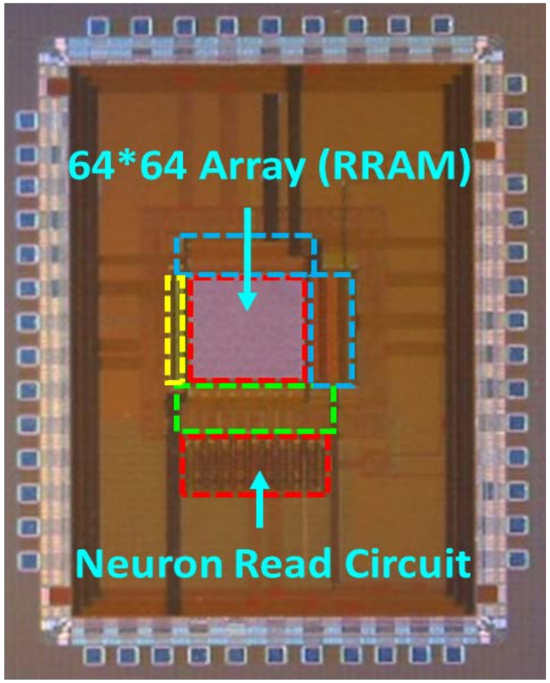




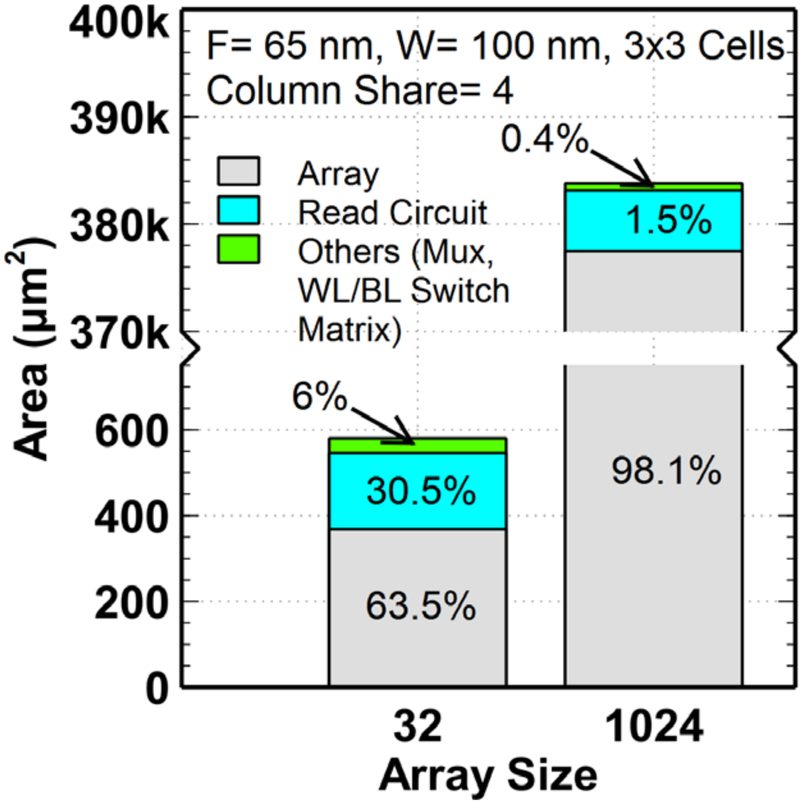


Array Size

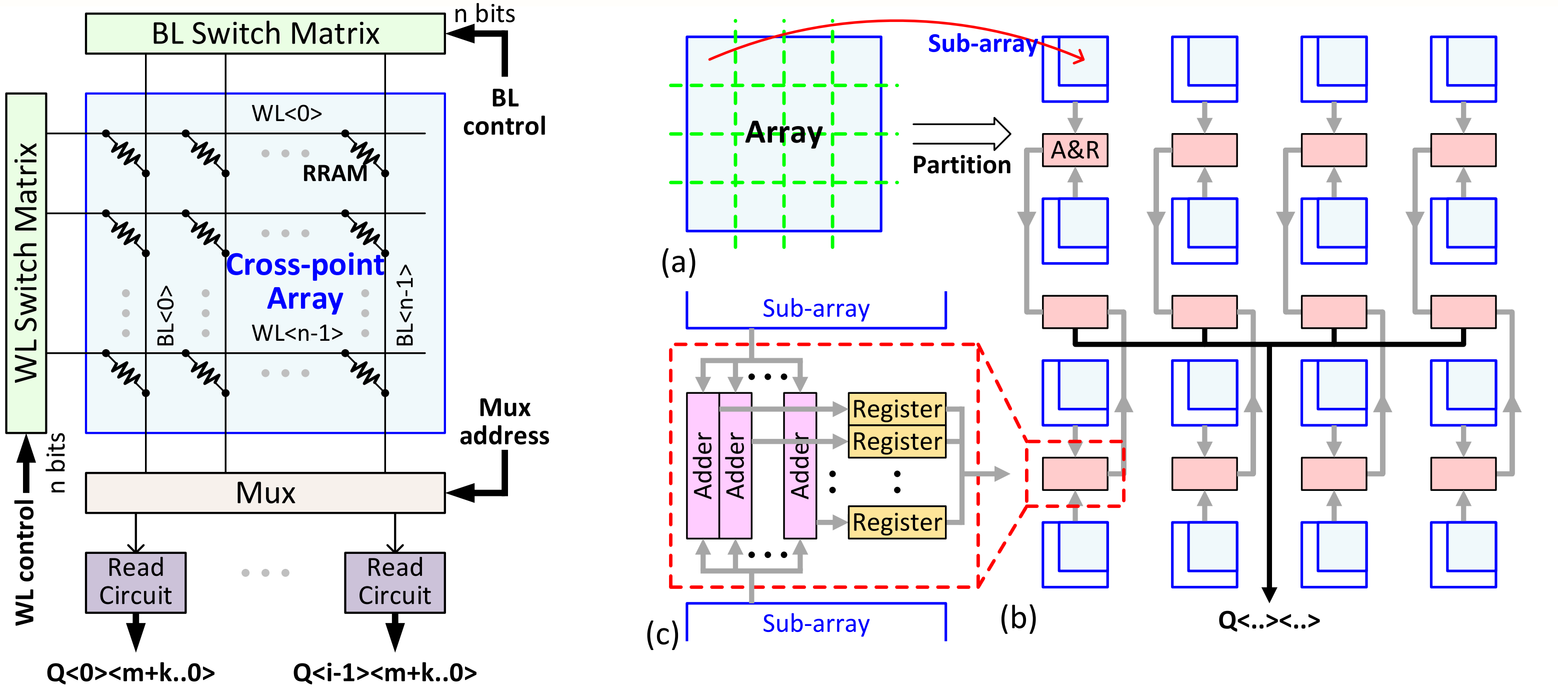
- Peripheral circuits consume significant area
- Solution: scaling up the array size; non-CMOS neurons



130nm 1T1R array



System Simulator for Benchmark



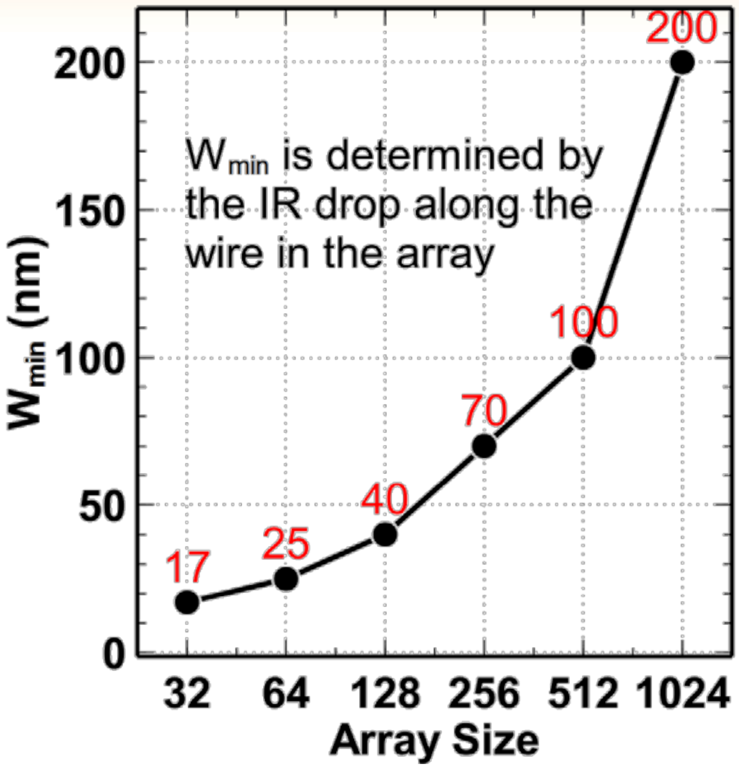
- Built on the template of CACTI and NVSim
- Metrics include area, latency, leakage power, dynamic power, etc. for a given array size, device type and node

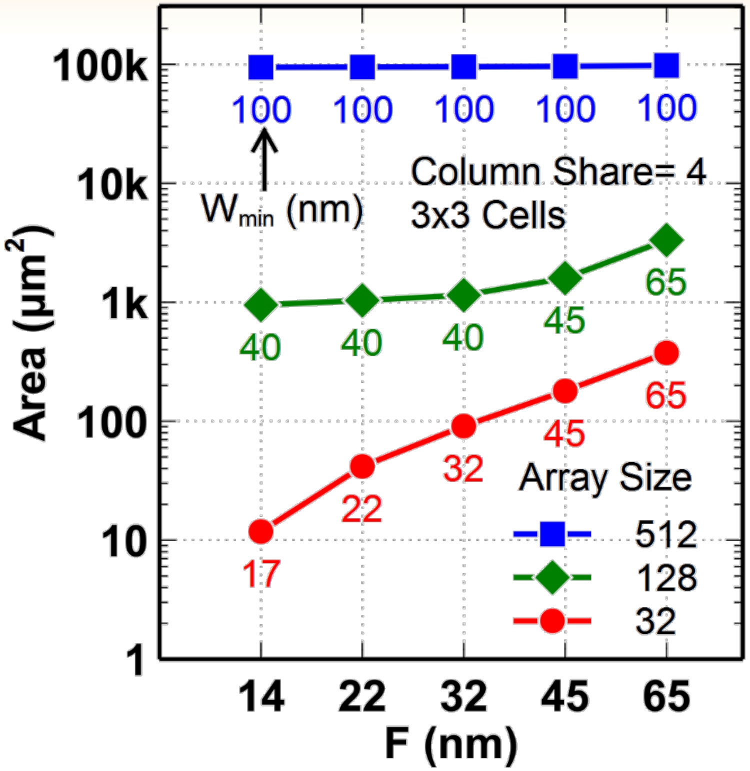
Example: A 256 x 256 Array

Architecture (array size=256^2)	Area	Read Latency	Read Energy	Write Latency	Write Energy	Leakage
SRAM Array (row-by-row)	39638.07 μm^2	**393.38 ns**	**15.14 nJ**	114.55 ns	**1.9 nJ**	**3247.93 μW**
1T1R Array (row-by-row)	5601.04 μm^2	75.51 ns	1.84 nJ	**10311.42 ns**	15.22 nJ	11.17 μW
Cross-point Array (fully parallel)	6551.49 μm^2	70.63 ns	1.68 nJ	160 ns	10.62 nJ	2.07 μW

Sparse Coding	SRAM	1T1R	Cross-point	Improvement
Update Z (200 Read)	78.7 μs	15.1 μs	14.1 μs	
Update D (1 Write)	115 ns	10.3 μs	160 ns	
Time for 1 Iteration	78.8 μs	25.4 μs	**14.2 μs**	**5.5×**

Technology Scaling





- Large array does not scale well due to wire width relaxation
- Solution: partition of large array into multiple small arrays with technology scaling

Future Needs

- Synaptic Device: variation control, read noise reduction, better endurance (habituation), more levels (>4-bit)
- Circuits and Architecture: larger array, **peripheral device/circuits**, physical design, multi-array architecture
- Neuromorphic Algorithm: brain-inspired algorithm for low precision, compact network, and high energy efficiency