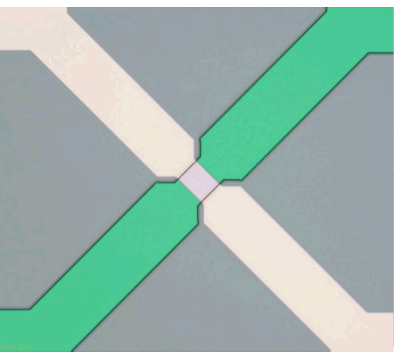
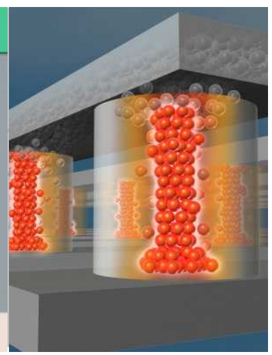
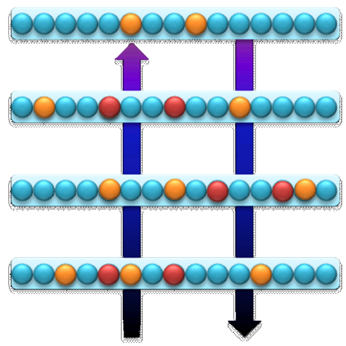
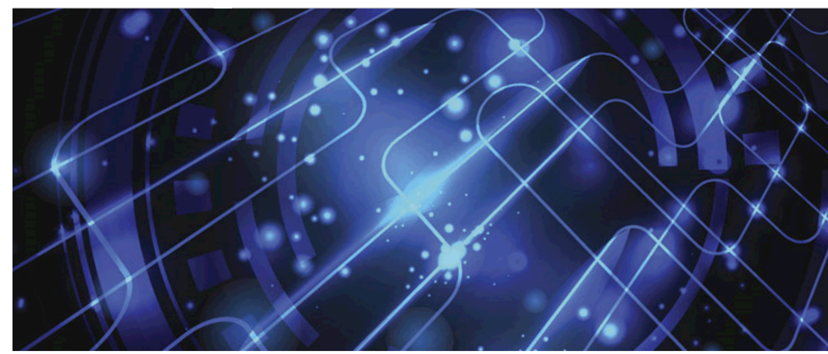
SAND2017-11349C









Designing an Analog Crossbar based Neuromorphic Accelerator

Sapan Agarwal, Alexander Hsia, Robin Jacobs-Gedrim, David R. Hughart, Steven J. Plimpton, Conrad D. James, Matthew J. Marinella
Sandia National Laboratories

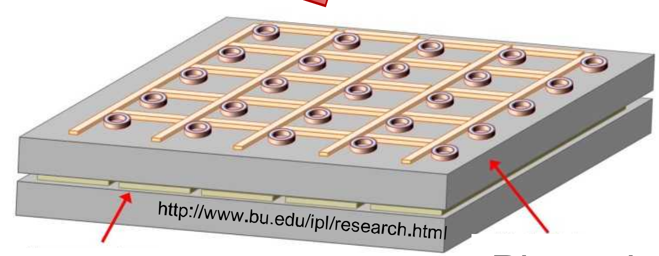
Hardware Acceleration of Adaptive Neural Algorithms

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

The Von Neumann Bottleneck

Current Transistors ~ 10 aJ
40kT Noise Limit ~ 0.2 aJ

Cross chip communications ~ 1 pJ
DRAM Access >10 pJ
Ethernet ~ 1nJ



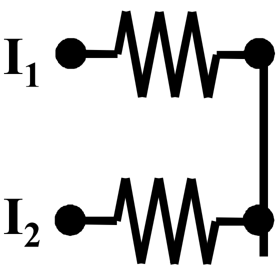
Communications require orders of magnitude more energy!

Optical interconnects 100 fJ to 1 pJ

Use Resistive Memories for Local Computation

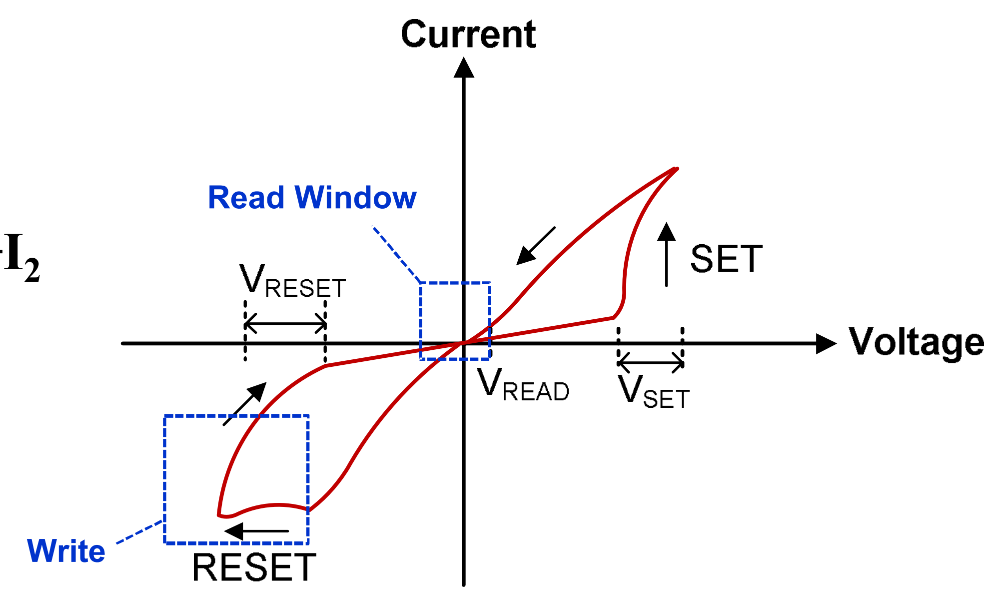
$\mathbf{V = I \times R}$

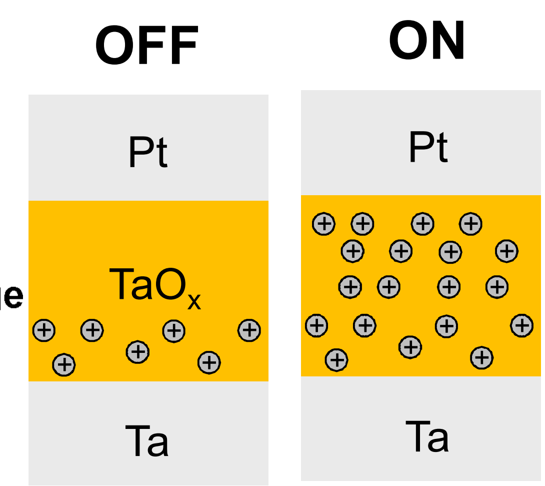
$\mathbf{I = G \times V}$ ← multiplication



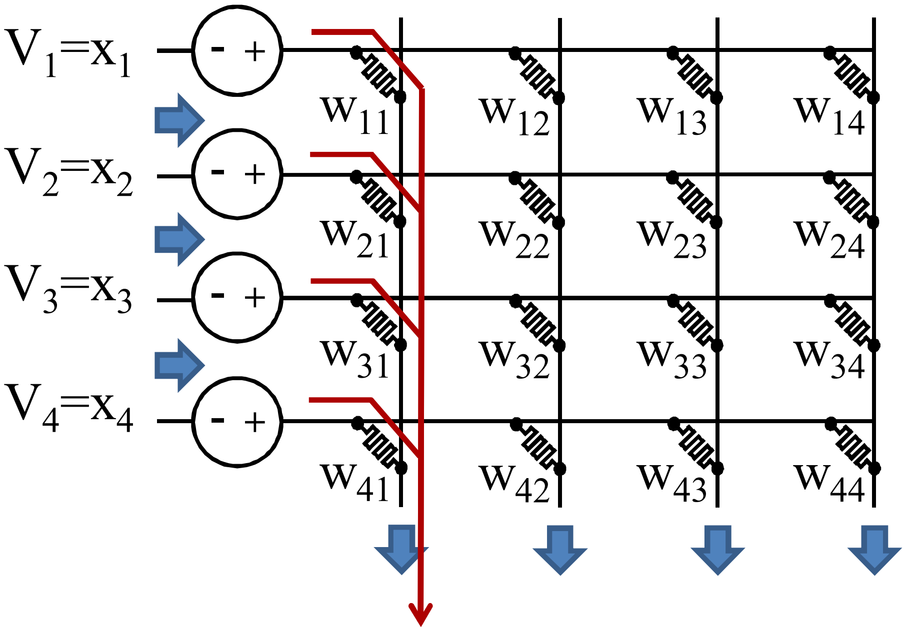
Addition: $\mathbf{I=I_1+I_2}$

- A resistive memory or ReRAM is a programmable resistor
  - Apply small voltages allows the conductance to be read: I = G × V
  - Apply large voltages to change the resistance





Directly Process in the Memory Itself

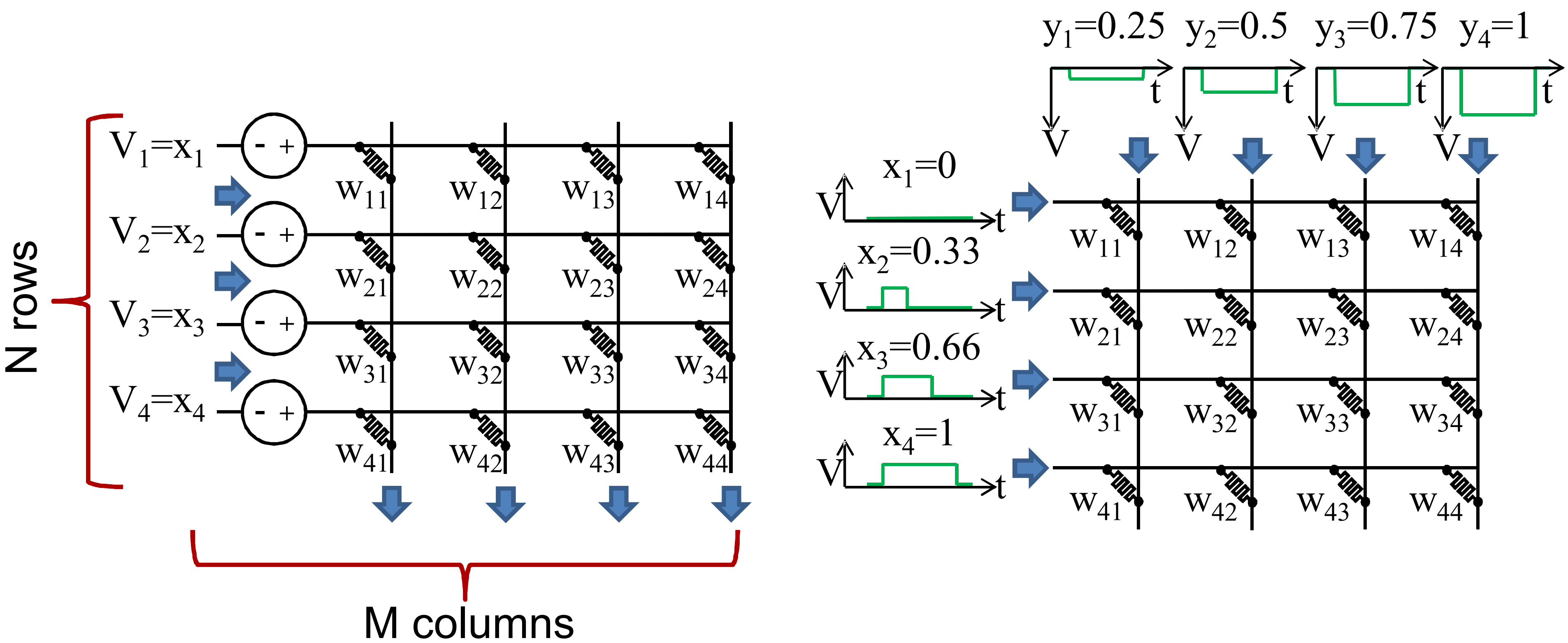


$I_1 = x_1 * w_{11} + \ldots + x_4 * w_{41}$

Analog is efficiently and naturally able to combine computation and data access

Effectively, large-scale processing in memory with a multiplier and adder at each real-valued memory location

Crossbars Can Perform Parallel Reads and Writes

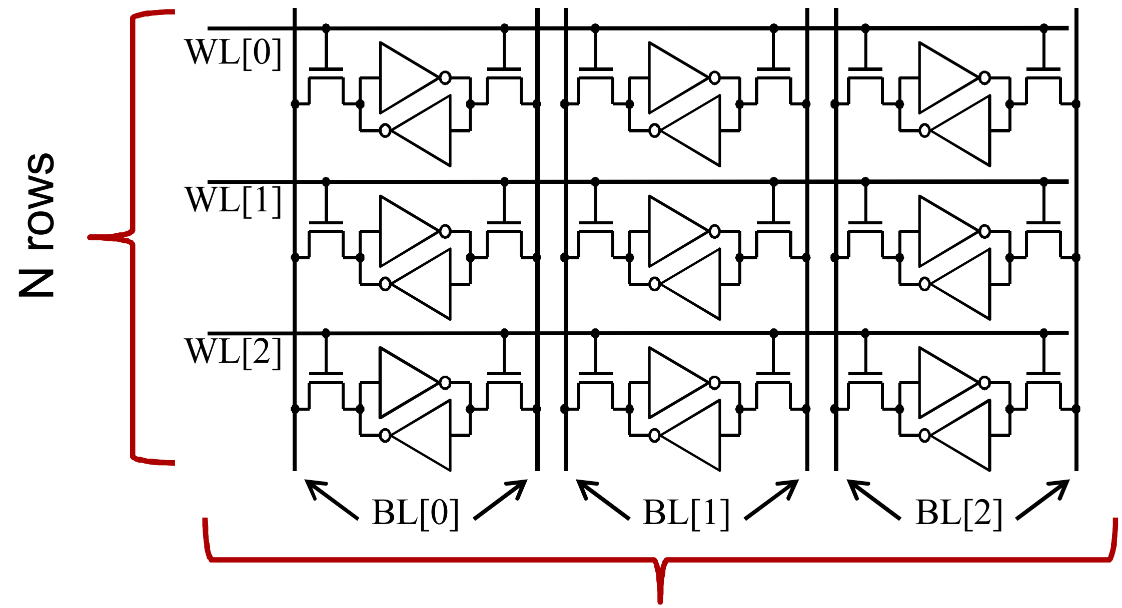


Energy to charge the crossbar is CV^2

$E \propto C \propto$ number of RRAMs $\propto N \times M$

$$E \sim O(N \times M)$$

SRAM Arrays Require Charging Columns Multiple Times



M columns

SRAMs must be read one row at a time, charging M columns

Each column wire length is O(N).

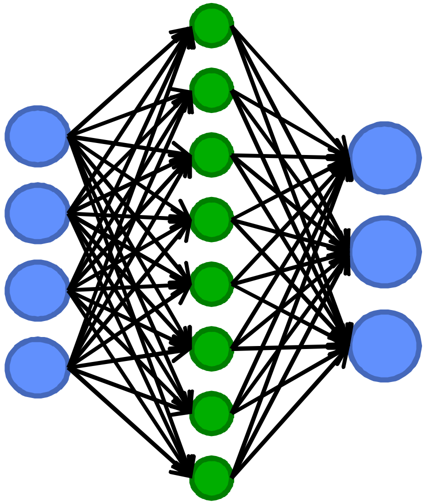
Energy = N Rows × M Columns × O(N) wire length

Energy ~ $O(N^2 \times M)$

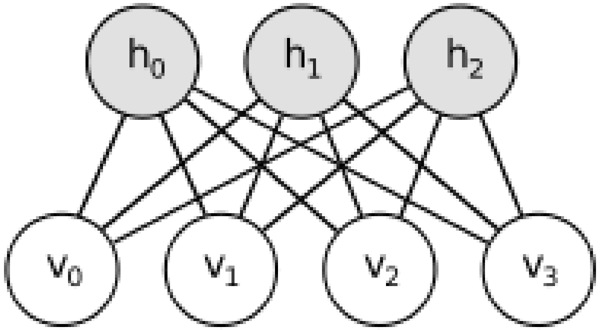
O(N) times worse than a crossbar!

Want To Accelerate Many Different Neural Algorithms

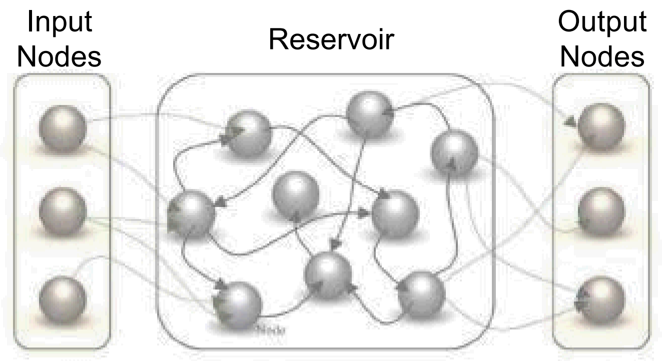
Backpropagation



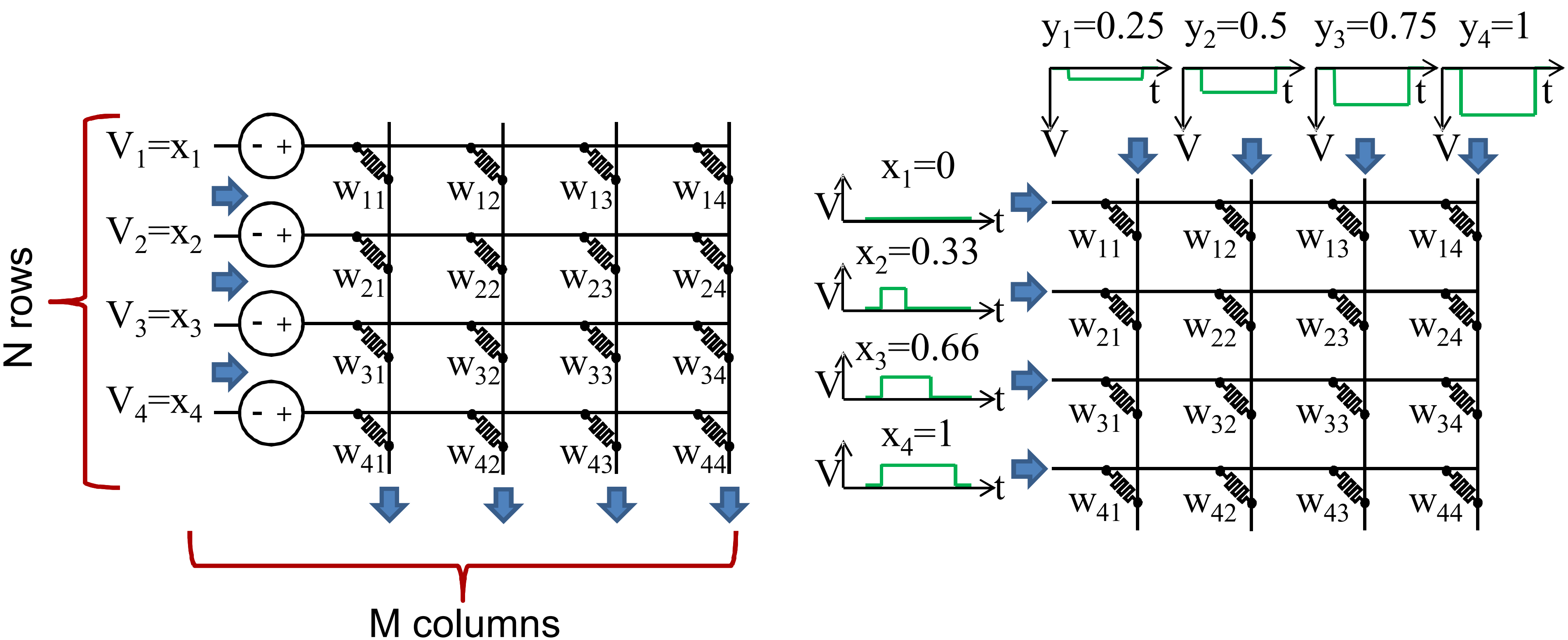
Sparse Coding



Liquid State Machine



Crossbars Can Perform Parallel Reads and Writes



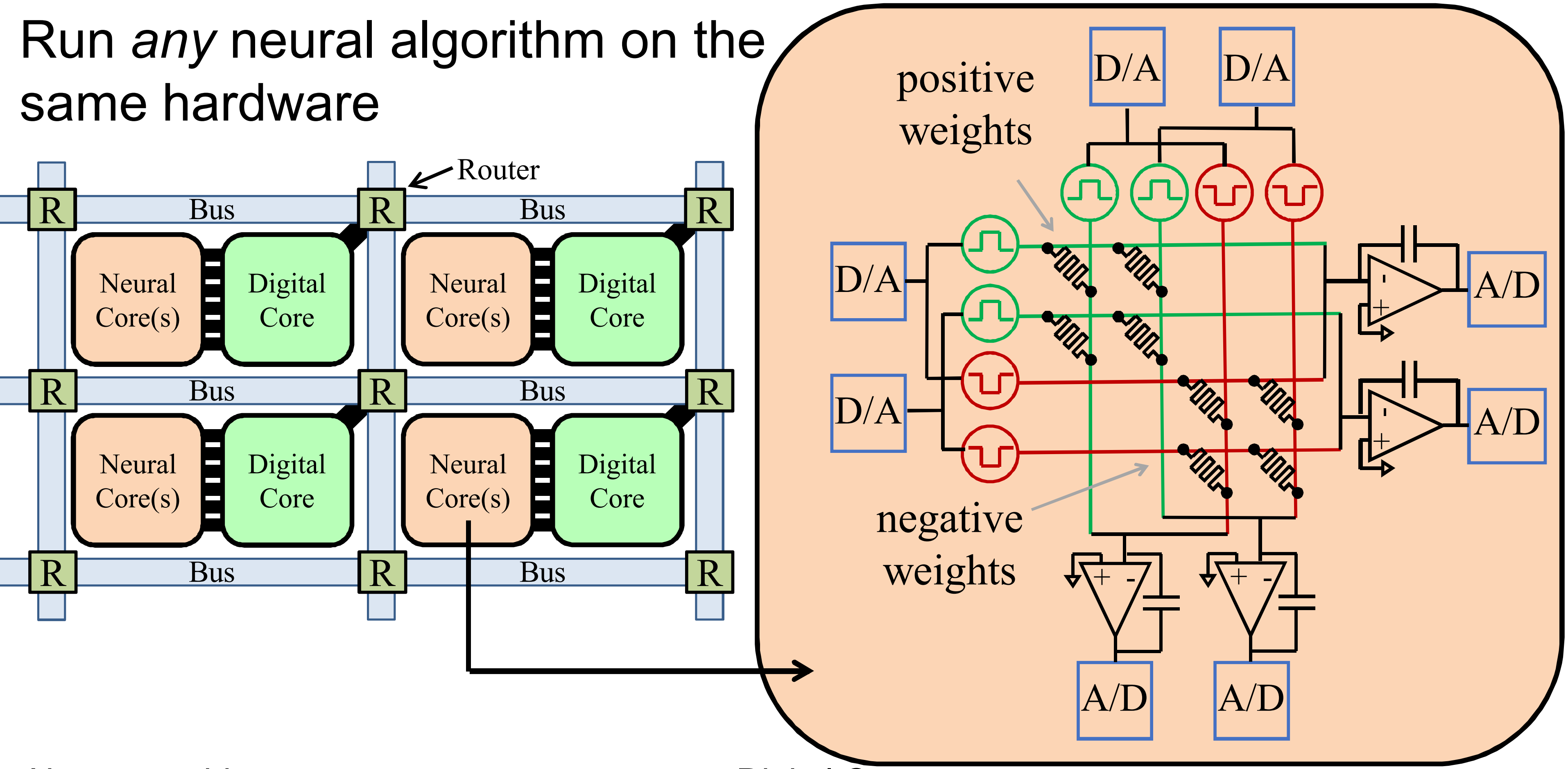
Energy to charge the crossbar is CV^2

$E \propto C \propto$ number of RRAMs $\propto N \times M$

$$E \sim O(N \times M)$$

General Purpose Neural Architecture

Run *any* neural algorithm on the same hardware



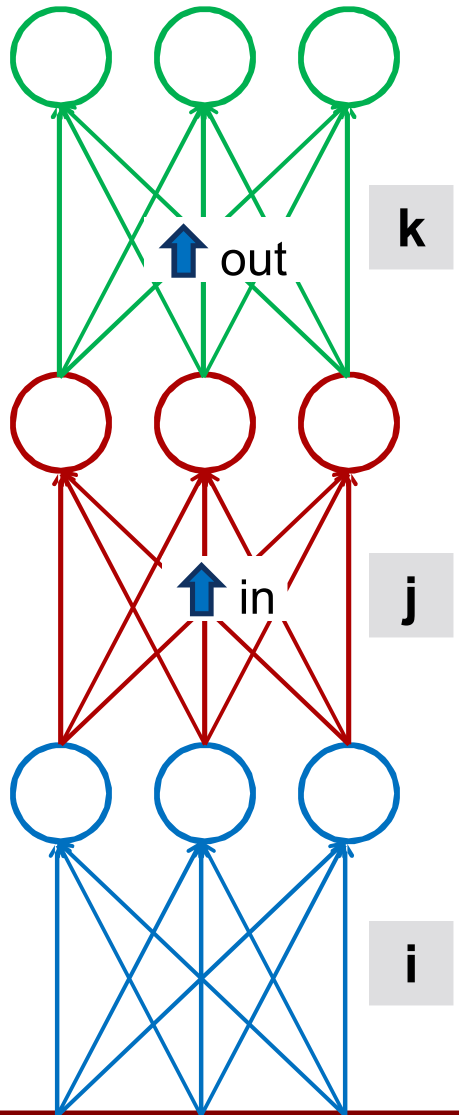
Neuromorphic core:

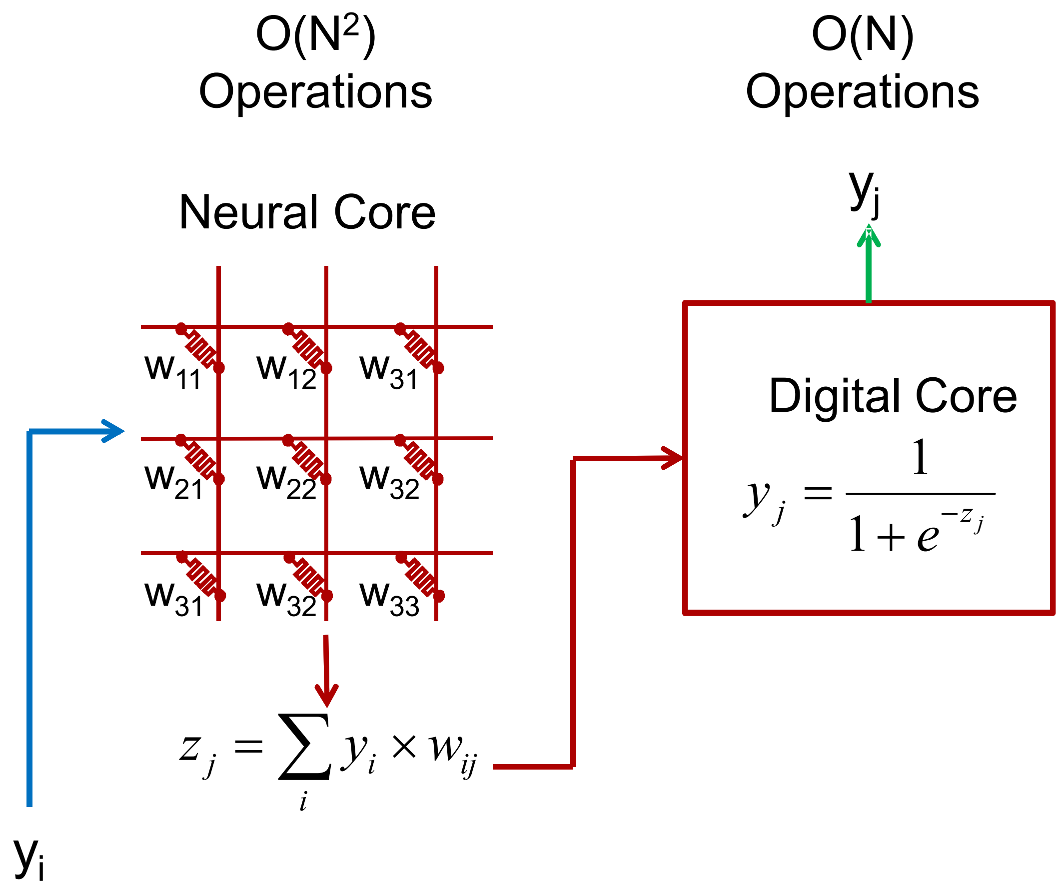
- Evaluate vector matrix multiplies along rows or columns
- Train based on input vectors

Digital Core:

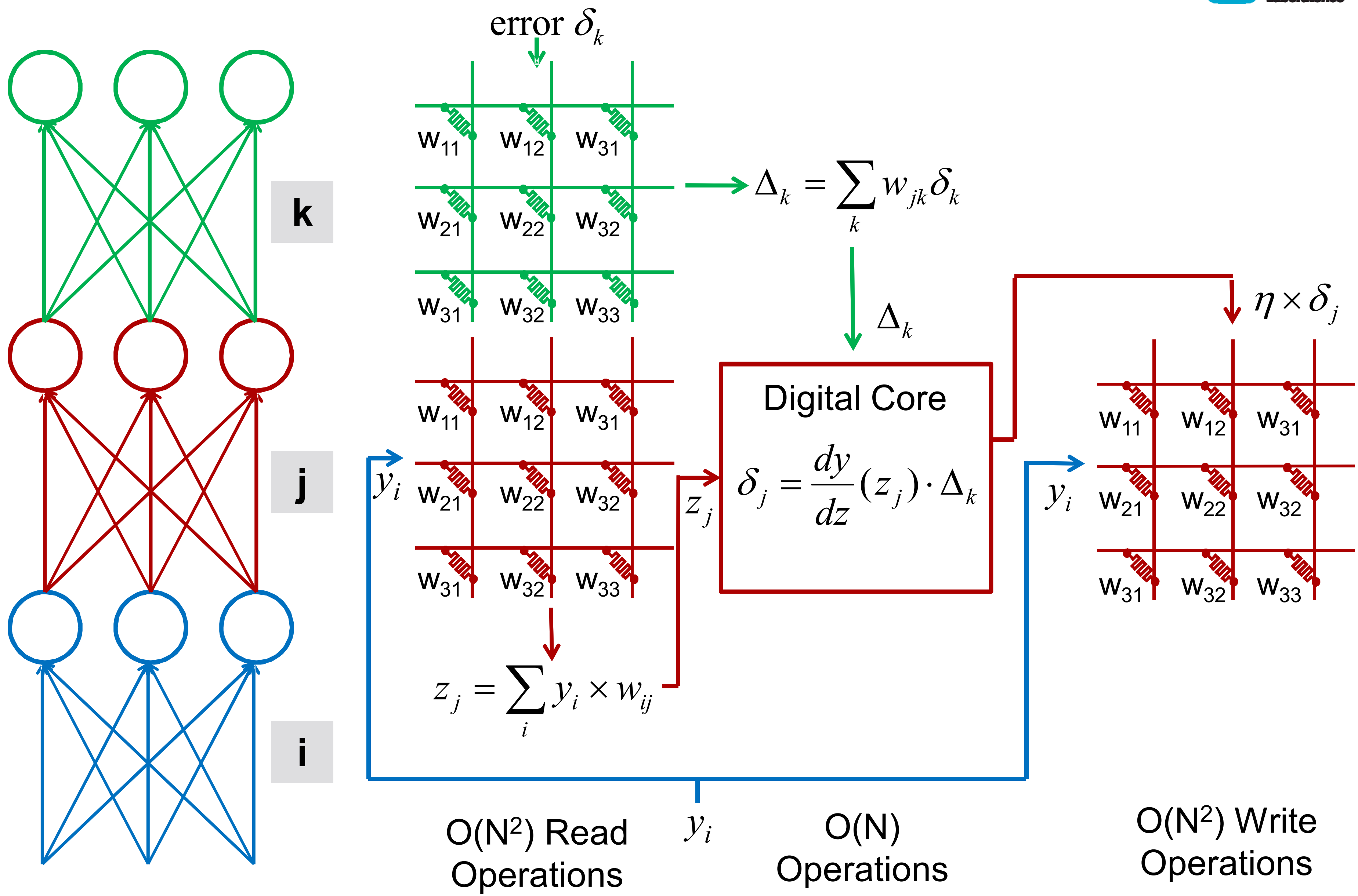
- Process neural core inputs/outputs
- For NxN crossbar, the crossbar accelerates $O(N^2)$ operations leaving only O(N) operations for the digital core

Can Run Neural Networks on this Architecture



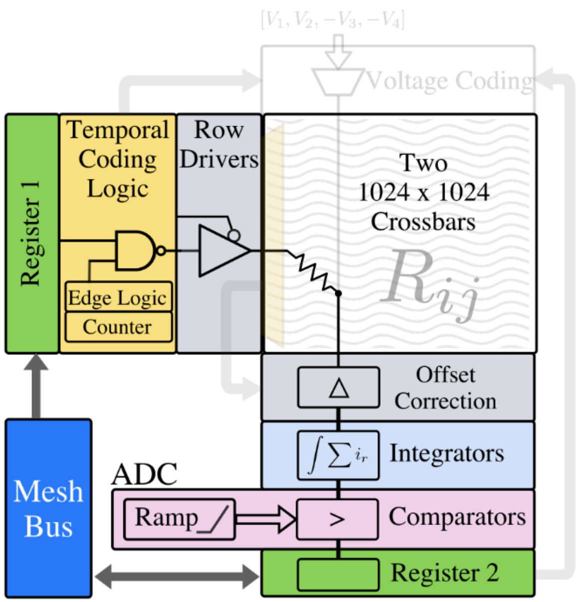


Back Propagation

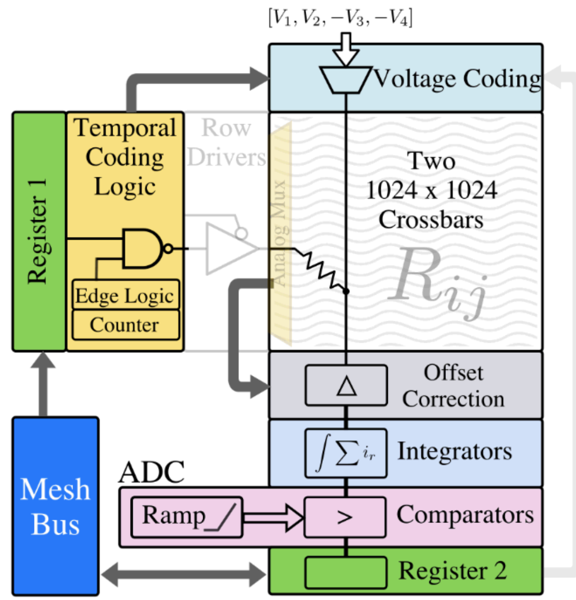


Design & Model Detailed Architecture

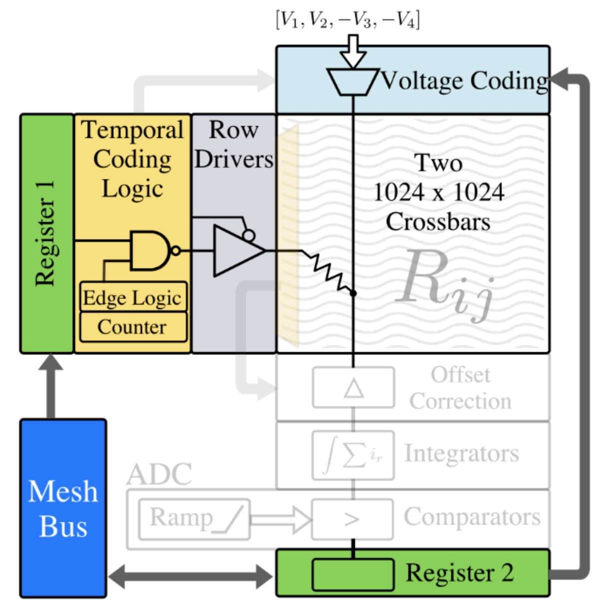
Vector Matrix Multiply



Matrix Vector Multiply



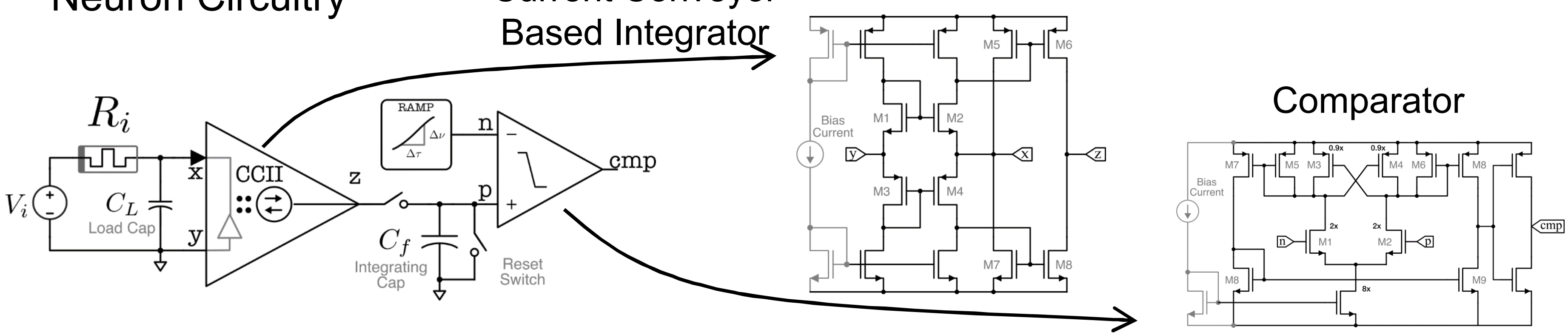
Outer product Update



Neuron Circuitry

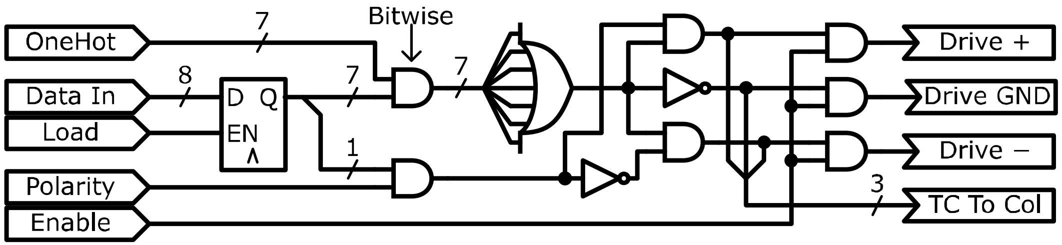
Current Conveyor Based Integrator

Comparator

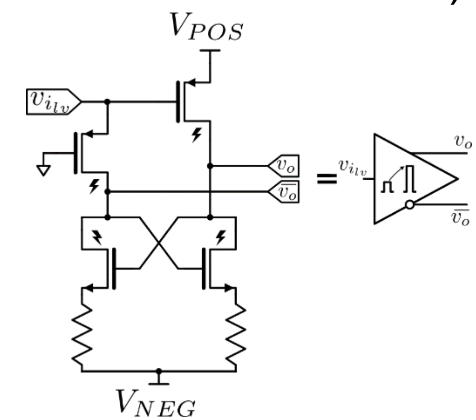


Row & Column Driver Circuitry

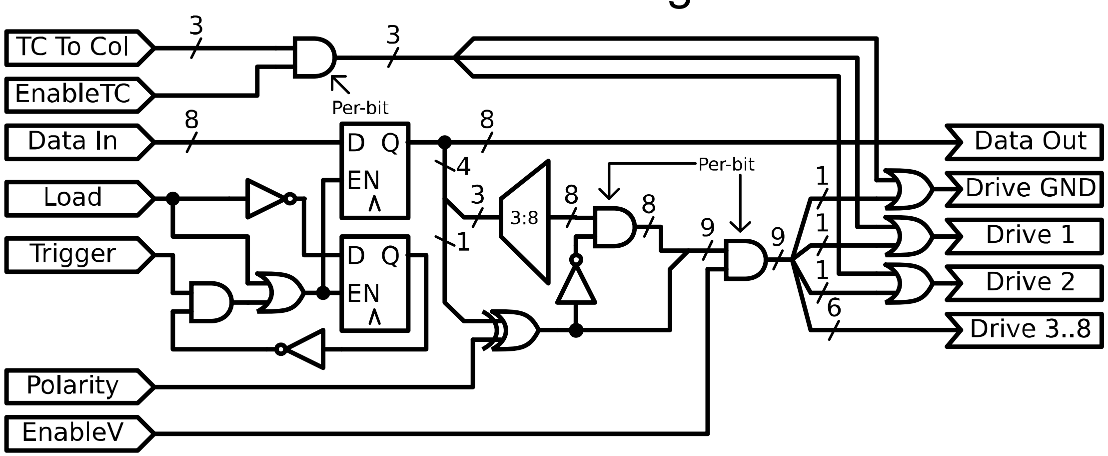
Row Driver Logic



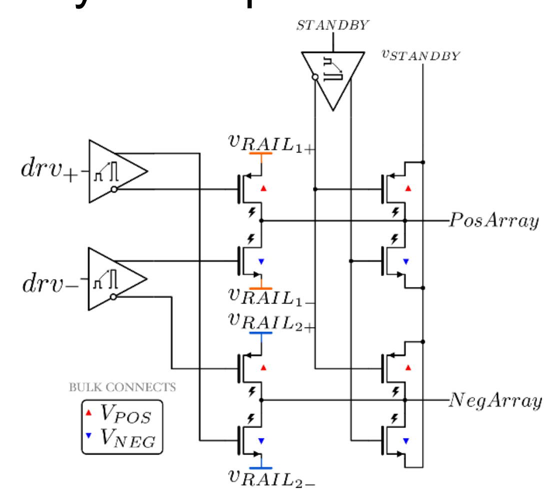
Voltage level shifter (drive high V transistor with low V)



Column Driver Logic



Array driver pass transistors



Compare Architectures

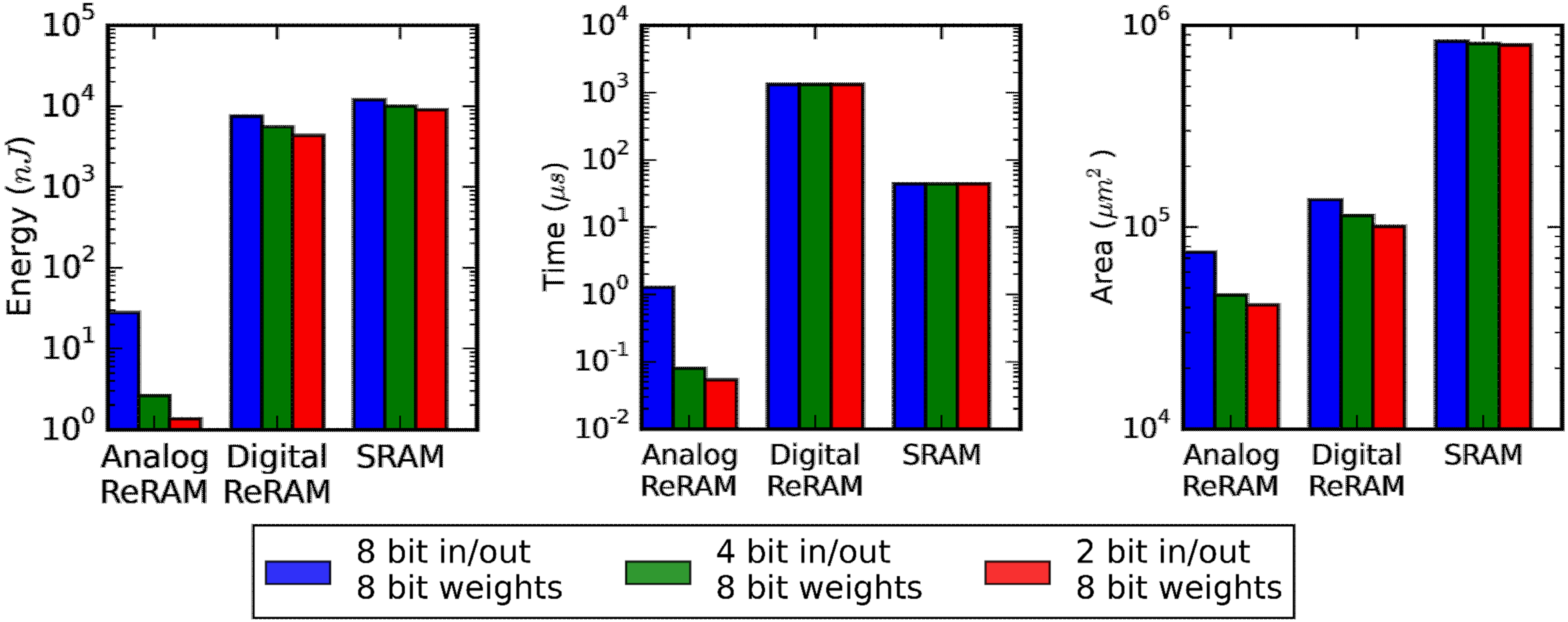
1024 x1024 = 1M array operations, sum over 1 training cycle, 3 operations:

- Vector Matrix Multiply
- Matrix Vector Multiply
- Outer Product Update

Energy
430 – 6,900X over SRAM

Latency
35 – 800X over SRAM

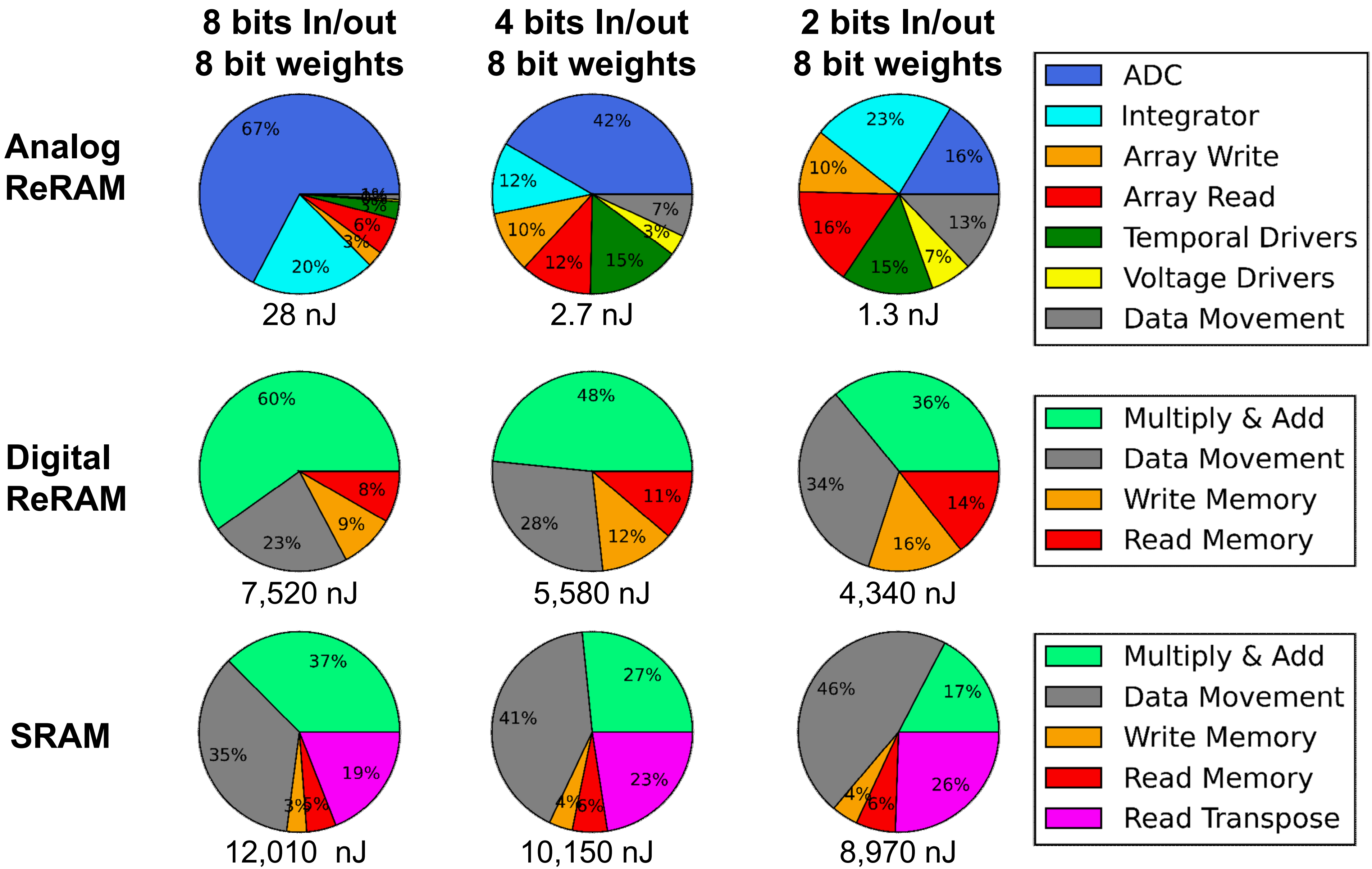
Area
11 – 20X over SRAM



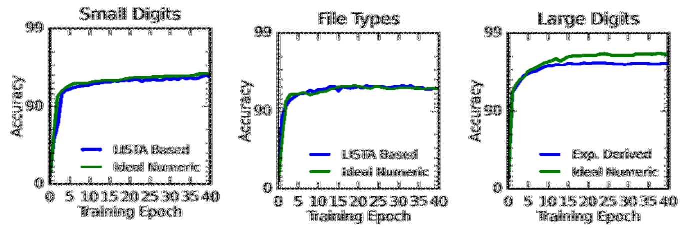
Used a commercial 14/16 nm PDK

***Requires 100 MΩ on state devices

Neural Core Energy Analysis



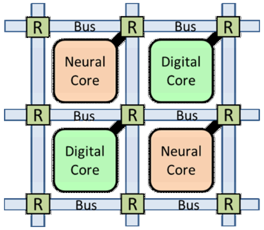
Multiscale Model of a Neural Training Accelerator



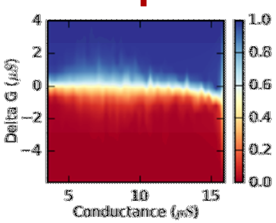
Target Algorithms
- Deep Learning
- Sparse Coding
- Liquid State Machines

Algorithms

Architecture

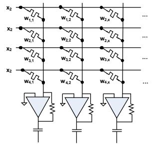


Modified McPAT/CACTI: Model performance and energy requirements



CROSS SIM

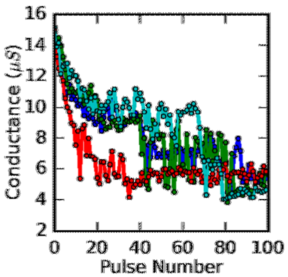
Sandia Cross-Sim: Translates device measurements and crossbar circuits to algorithm-level performance



Circuits

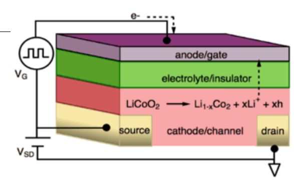
Xyce PARALLEL ELECTRONIC SIMULATOR

Sandia's Xyce Circuit Sim: Simulate crossbar circuits based on our devices

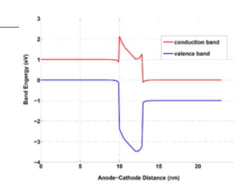


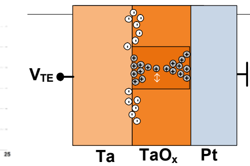
Memristor fabrication and measurements in MESAFab

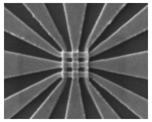
Devices

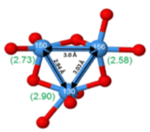


Drift-diffusion model of ReRAM band diagram & transport (REOS, Charon)





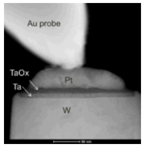


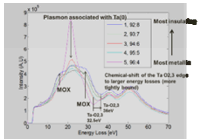


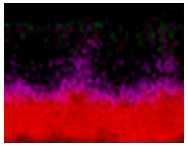
DFT of model of oxide physics, bands

Materials

In situ TEM of filament switching: Use DFT model to interpret EELS signature

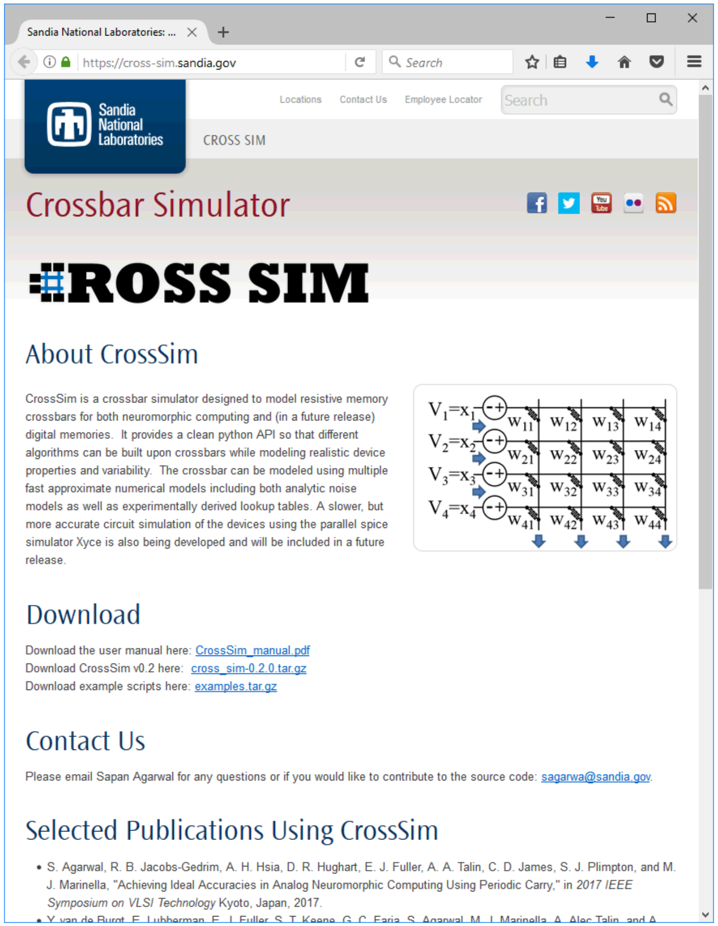






CROSS SIM

https://cross-sim.sandia.gov


Sandia National Laboratories: ...

https://cross-sim.sandia.gov

Locations Contact Us Employee Locator Search

Sandia National Laboratories

CROSS SIM

Crossbar Simulator

CROSS SIM

About CrossSim

CrossSim is a crossbar simulator designed to model resistive memory crossbars for both neuromorphic computing and (in a future release) digital memories. It provides a clean python API so that different algorithms can be built upon crossbars while modeling realistic device properties and variability. The crossbar can be modeled using multiple fast approximate numerical models including both analytic noise models as well as experimentally derived lookup tables. A slower, but more accurate circuit simulation of the devices using the parallel spice simulator Xyce is also being developed and will be included in a future release.

Download

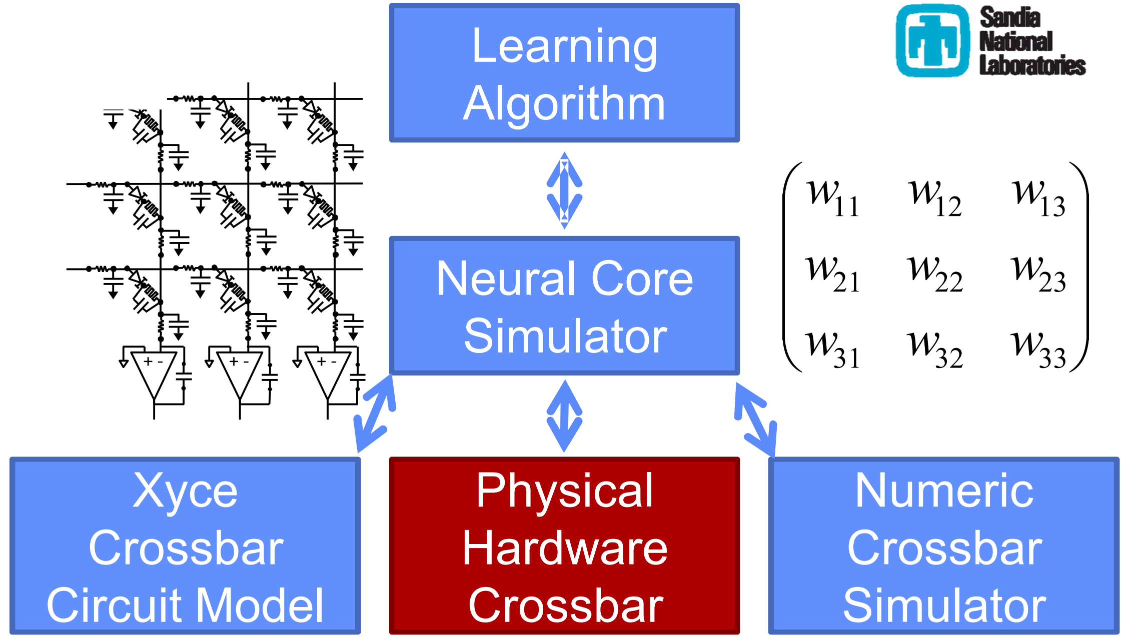
Download the user manual here: CrossSim_manual.pdf
Download CrossSim v0.2 here: cross_sim-0.2.0.tar.gz
Download example scripts here: examples.tar.gz

Contact Us

Please email Sapan Agarwal for any questions or if you would like to contribute to the source code: sagarwa@sandia.gov

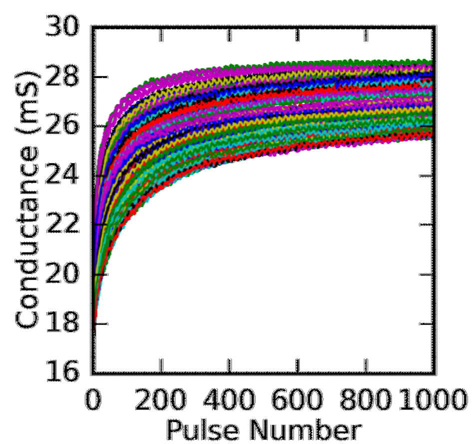
Selected Publications Using CrossSim

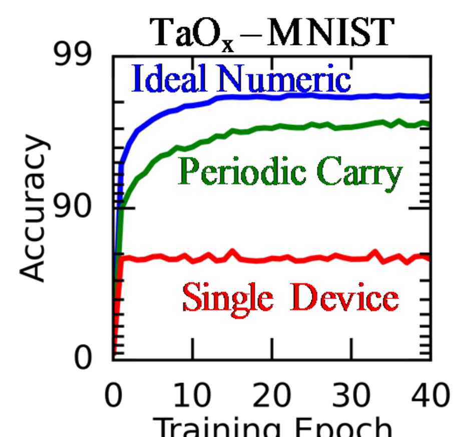
- S. Agarwal, R. B. Jacobs-Gedrim, A. H. Hsia, D. R. Hughart, E. J. Fuller, A. A. Talin, C. D. James, S. J. Plimpton, and M. J. Marinella, "Achieving Ideal Accuracies in Analog Neuromorphic Computing Using Periodic Carry," in *2017 IEEE Symposium on VLSI Technology* Kyoto, Japan, 2017.



Detailed but slow **Fast but approximate**

Measured Devices **Algorithmic Performance**





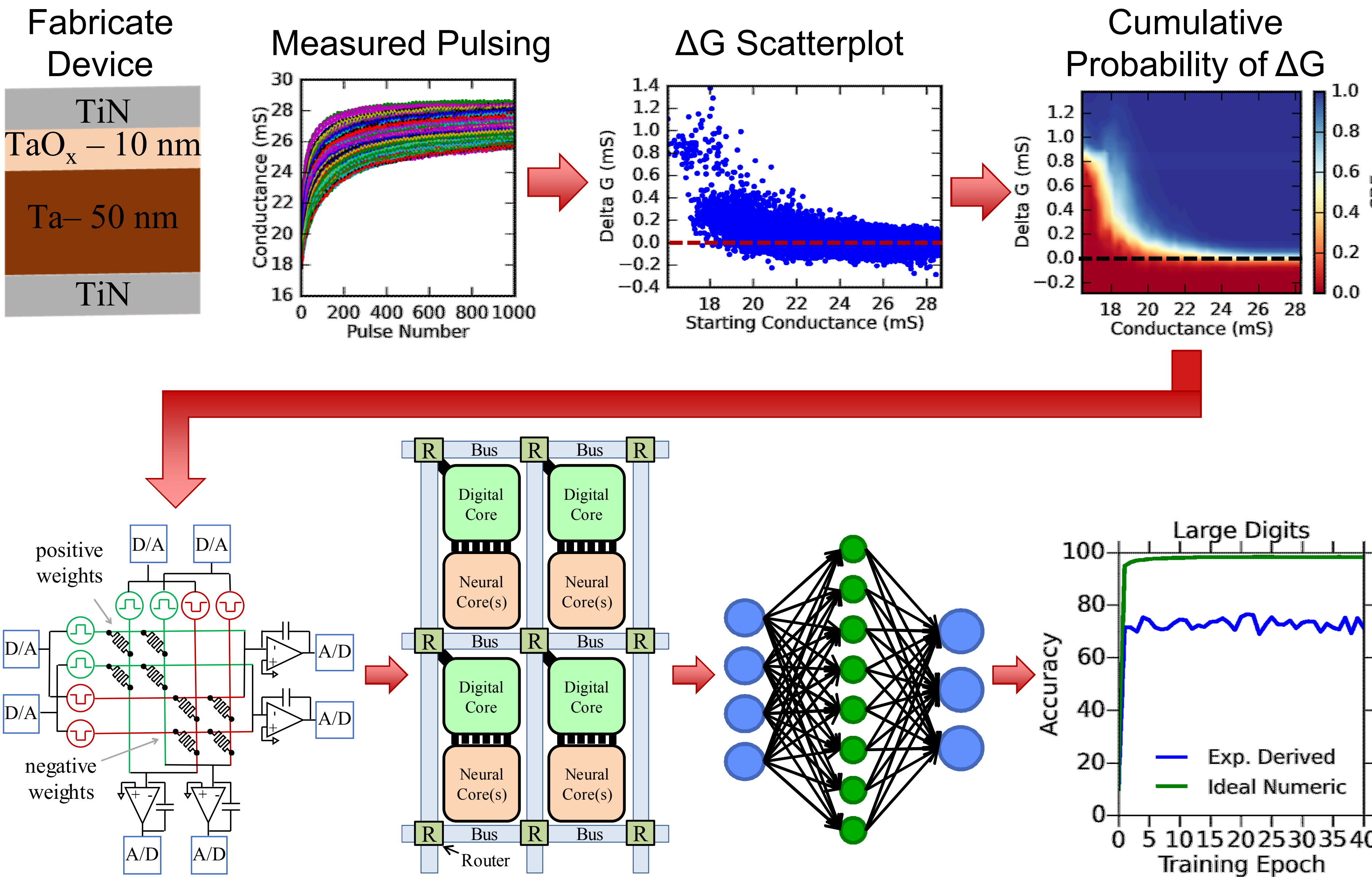
Simple Python API:

```python
# Do a matrix vector multiplication
result = neural_core.run_xbar_mvm(vector)
```

Simple API to model crossbars

```
# ************** set parameters defining the crossbar

params.algorithm_params.weights.sim_type = “XYCE” # Use a XYCE based sim
params.algorithm_params.weights.maximum = 10  # clipping limits
params.algorithm_params.weights.minimum = -10  # clipping limits
params.xyce_parameters.xbar.device.TAHA_A1 = 4e-4  # Xyce Parameters
…
…

# ************** API for running neural operations
#                          All crossbar details are transparent to the user

# Create a neural_core object that models a crossbar
neural_core = MakeCore(params=params)

neural_core.set_matrix(weights) # set the initial weights
result = neural_core.run_xbar_vmm(vector) # Do a vector matrix multiply
result = neural_core.run_xbar_mvm(vector) # Do the transpose, a matrix vector mult.
neural_core.update_matrix(vector1,vector2) # Do an outer product update
```

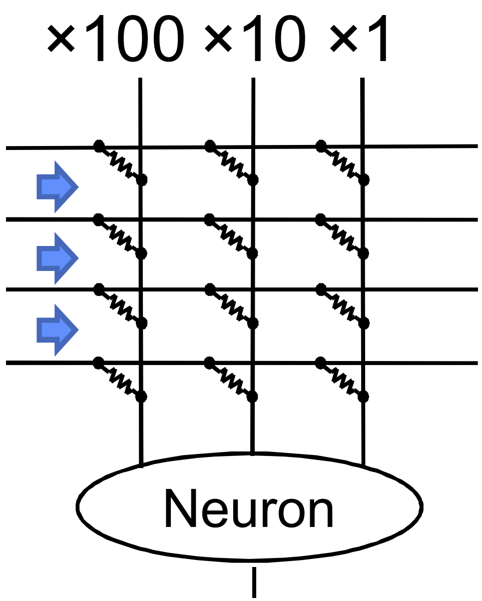
Go from Measurement to Accuracy



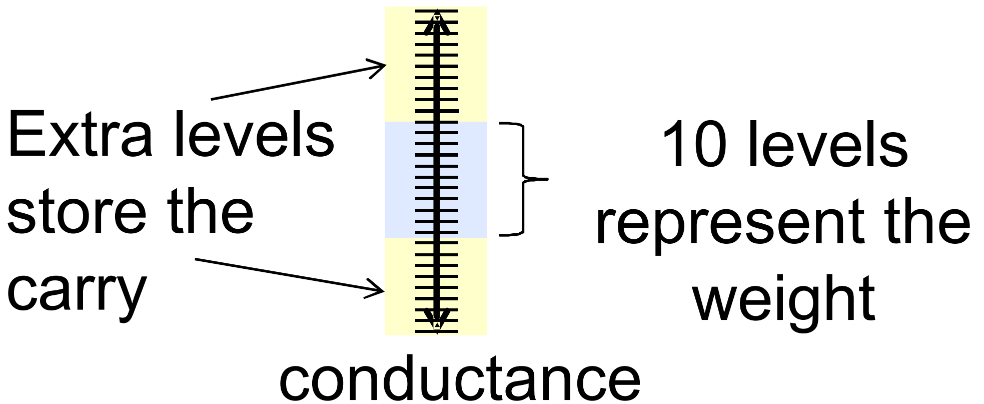
Multi-ReRAM Synapse: Periodic Carry

If we need more bits per synapse, use multiple memristors

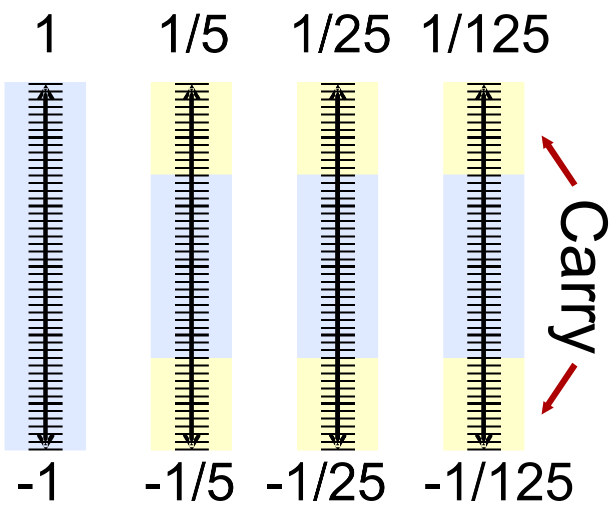
- Three 10 level ReRAMs could represent 1-1000!
- Adding to the weight requires reading every ReRAM to account for any carries and serially programming each ReRAM: VERY EXPENSIVE

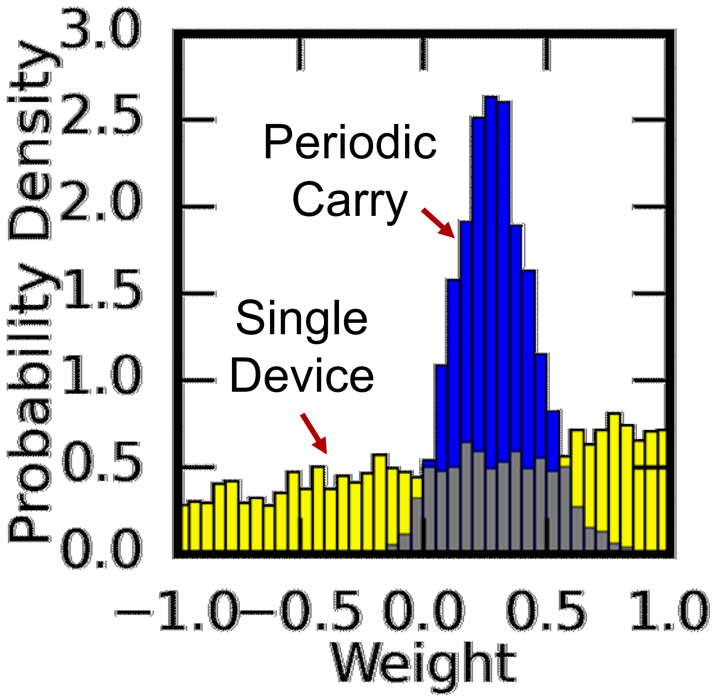


- Use >10 levels to represent a base 10 system
- Ignore carry and program the crossbar in parallel.
- Periodically (once every few hundred cycles) read the ReRAM and perform the carry



Periodic Carry Compensates for Write Noise





Read and reset every 100 pulses

Do 300,000 small (0.02% of weight range) updates

- net of 1500 positive training pulses

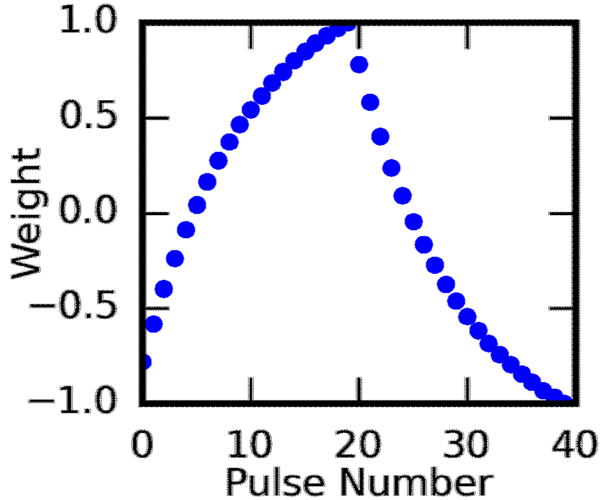
Learn from a 0.5% Signal

Noise Sigma = 1.4% for single device

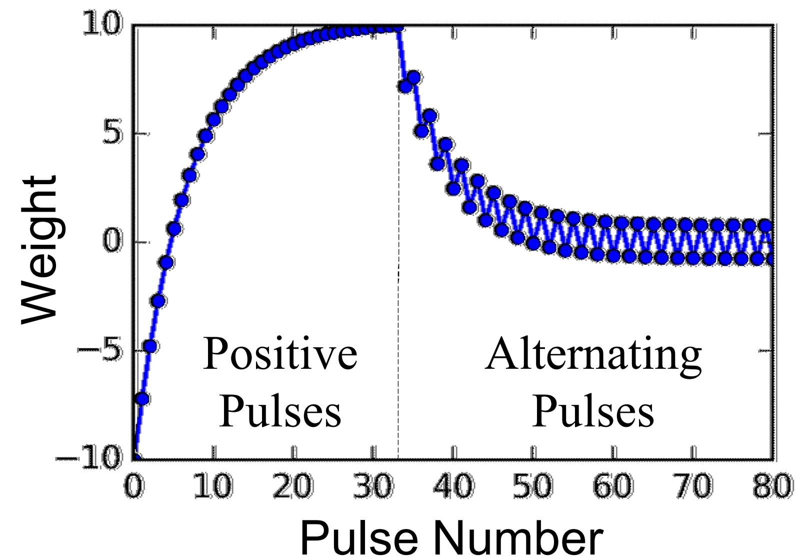
- (from $\sigma_{noise}/G_{range} = 0.1\sqrt{\Delta G/G_{range}}$)
- Write noise applied during updates and carries

Periodic Carry Mitigates Write Nonlinearity

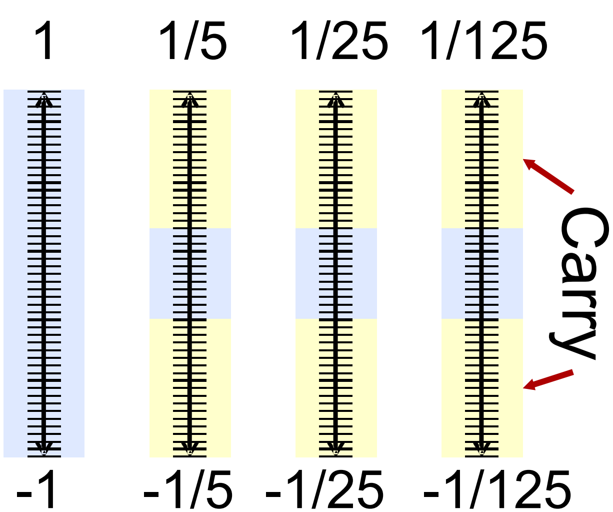
Write Nonlinearity

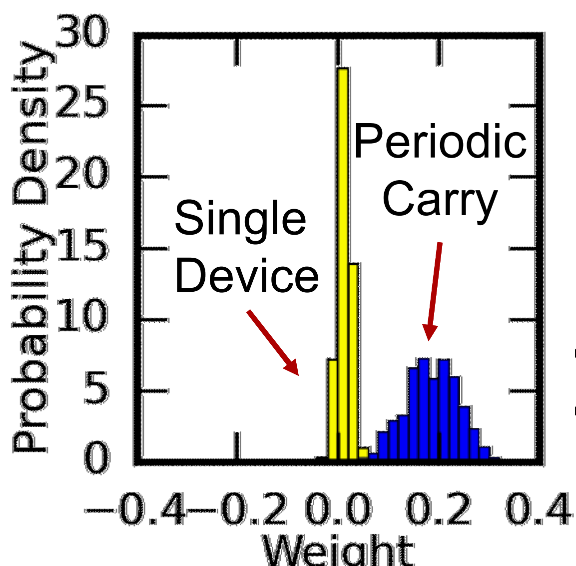


Alternating Pulses Cause Weight Decay



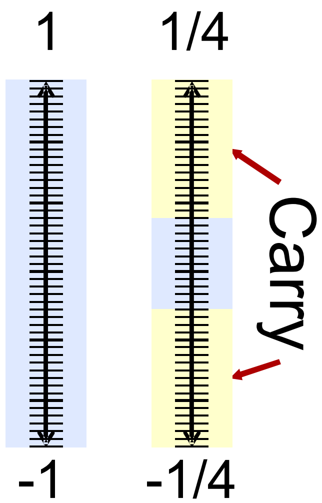
Use center linear range of weights



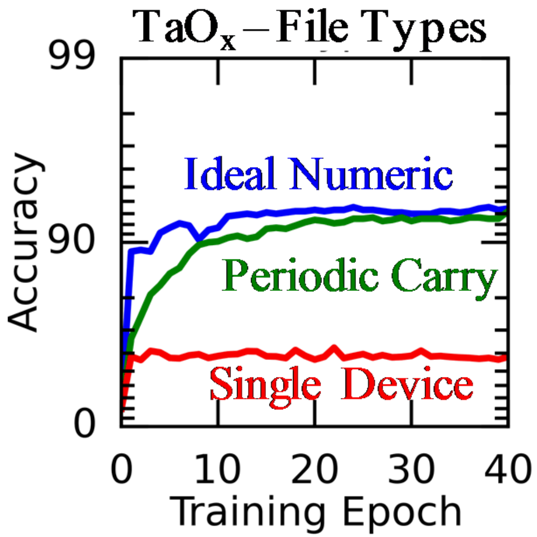


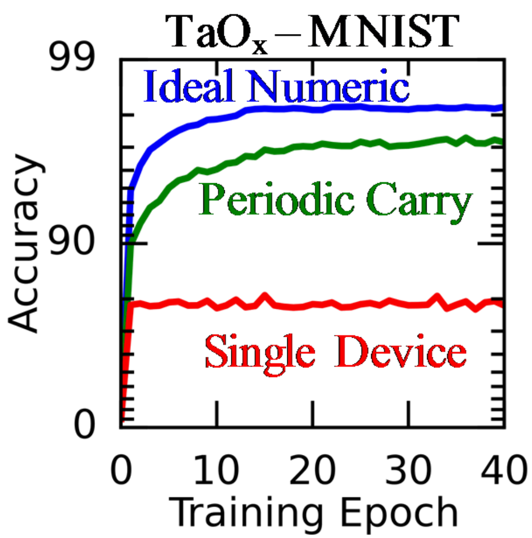
- Train with 1% signal
- Ideal result is 0.6

TaO_x Results



Carry once every
1000 updates

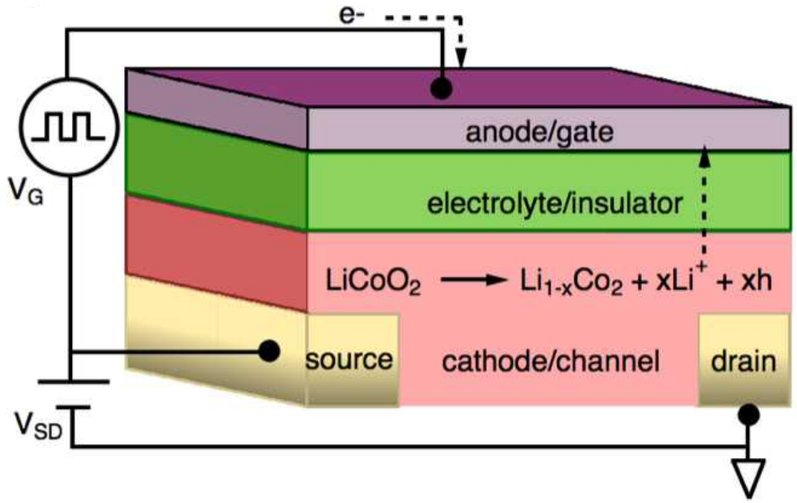


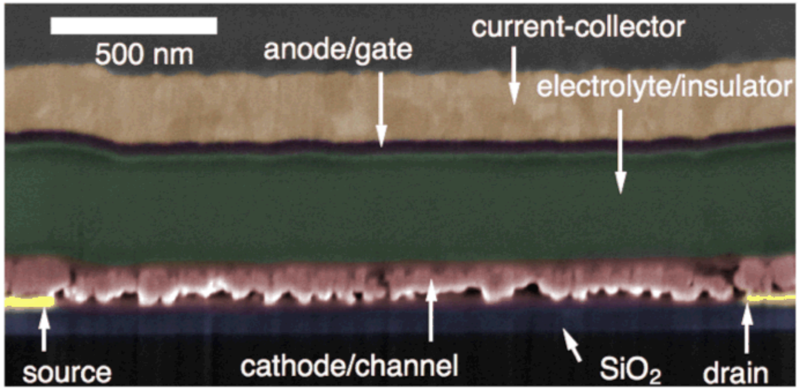


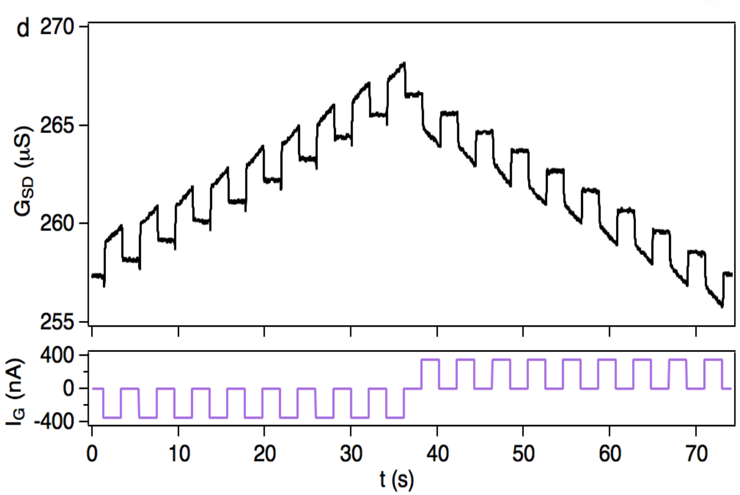
A/D and D/A is modeled, Serial operations modeled

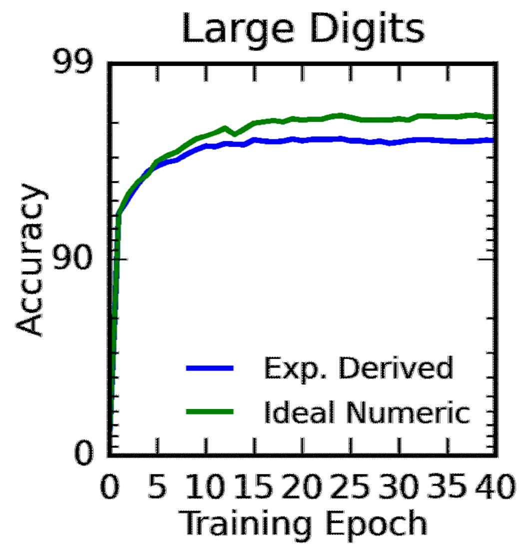
- When resetting weight, need to adjust pulse size based on current state to compensate for nonlinearity
- When reading a single weight, need to adjust readout range to be smaller (change capacitor on the integrator)

Li-Ion Synaptic Transistor for Analog Computation (LISTA)





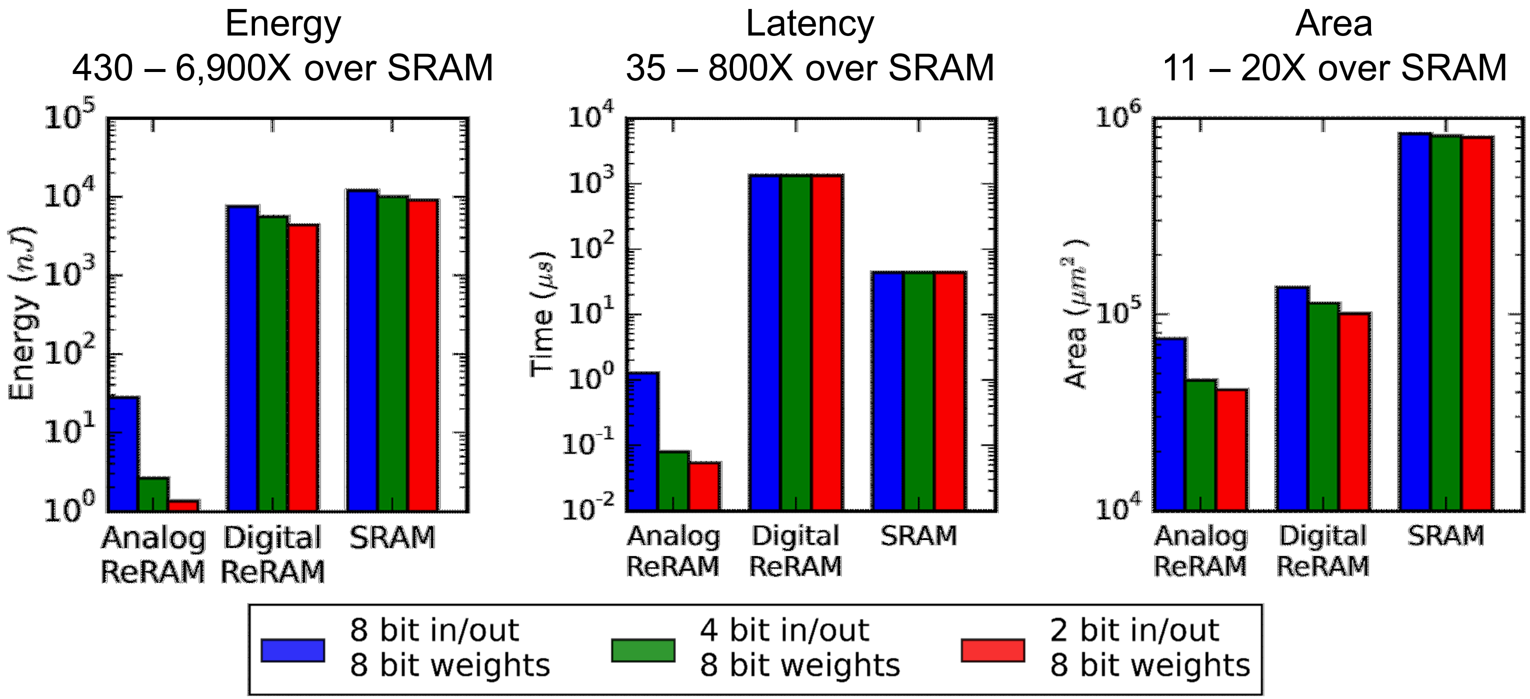




Off by ~ 1% from Ideal

E. J. Fuller, et al, "Li-Ion Synaptic Transistor for Low Power Analog Computing," *Advanced Materials,* vol. 29, no. 4, p. 1604310, 2017.

Summary

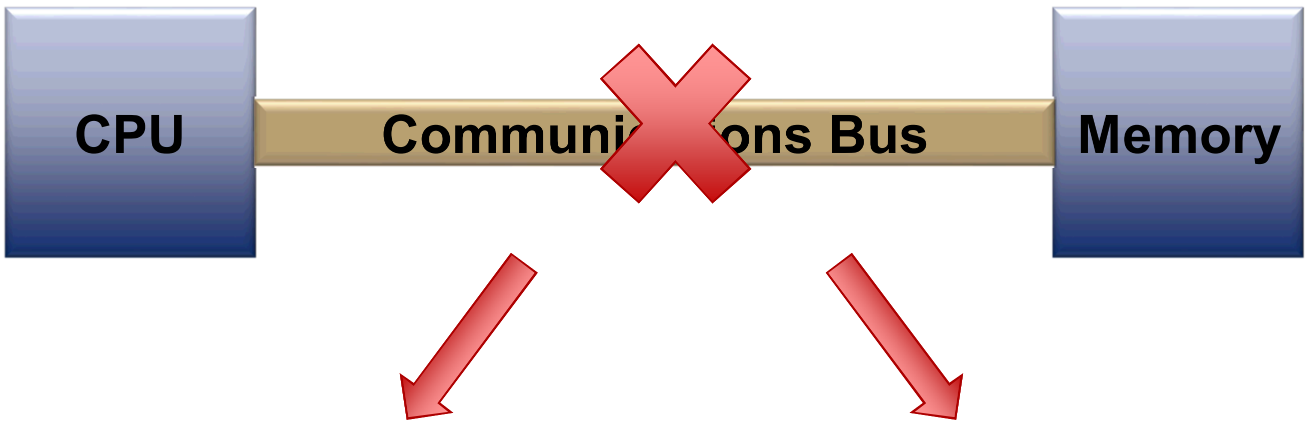


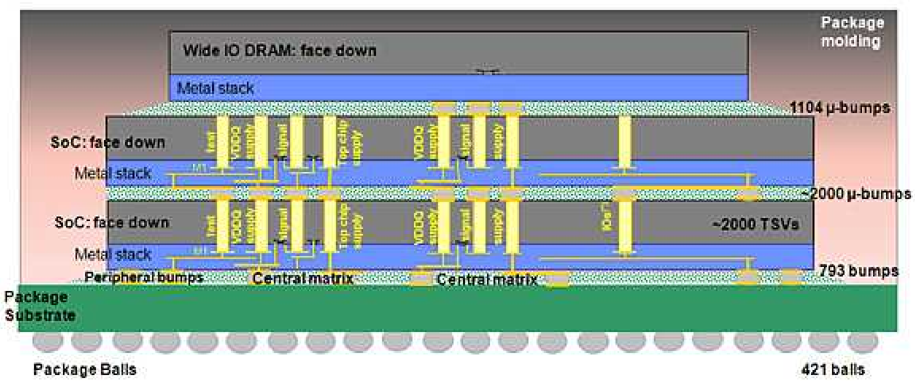
- Fundamental O(N) energy scaling advantage
- Use CrossSim to co-design materials to algorithms
  - Use periodic carry to overcome noise devices
- Need high resistance 10-100 MΩ Devices
- Need low write nonlinearities

https://cross-sim.sandia.gov

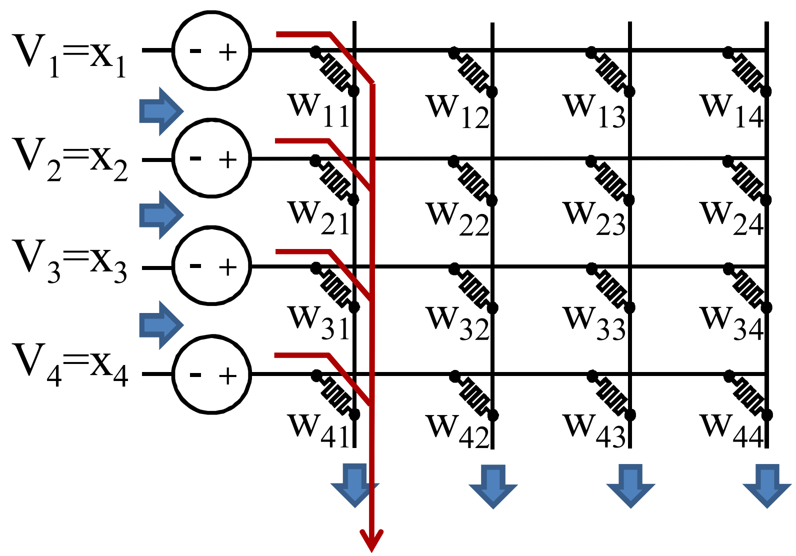
Extra Slides

Overcoming the Power Limit



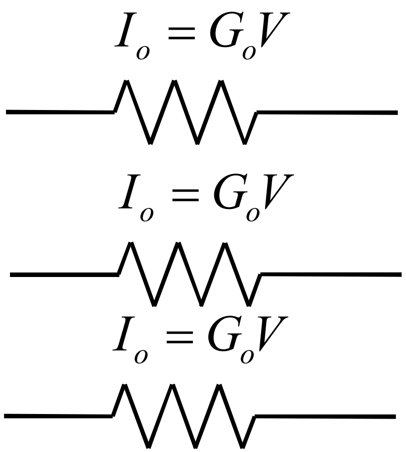


Richard Goering, “Three Die Stack -- A Big Step “Up” for 3D-ICs with TSVs” Cadence blog



Integrate Processing and Memory

The Noise Limited Energy to Read a Crossbar Column is Independent of Crossbar Size



$$\text{Thermal Noise} = \left\langle \Delta I^2 \right\rangle$$
$$= N \times \left(4 k_b T \times G_o \times \Delta f\right)$$

$$SNR^2 = \frac{(N I_o)^2}{\left\langle \Delta I^2 \right\rangle}$$

$$\frac{1}{\Delta f} = 4 k_b T \times SNR^2 \times \frac{1}{V^2 G_o \times N}$$

Measure N resistors and determine the total output current with some signal to noise ratio (SNR)*

What is the minimum energy?

$$Energy = \underbrace{V^2 G_O \times N}_{\text{Power in each resistor} \times \text{number of resistors}} \times \frac{1}{\Delta f}$$

Power in each resistor × number of resistors

Determined by noise and SNR

If we double the number of resistors, we can double the speed to get the same energy and SNR.

This is because the noise scales as sqrt(N) while the signal scales as N

$$Energy = 4 k_b T \times SNR^2$$

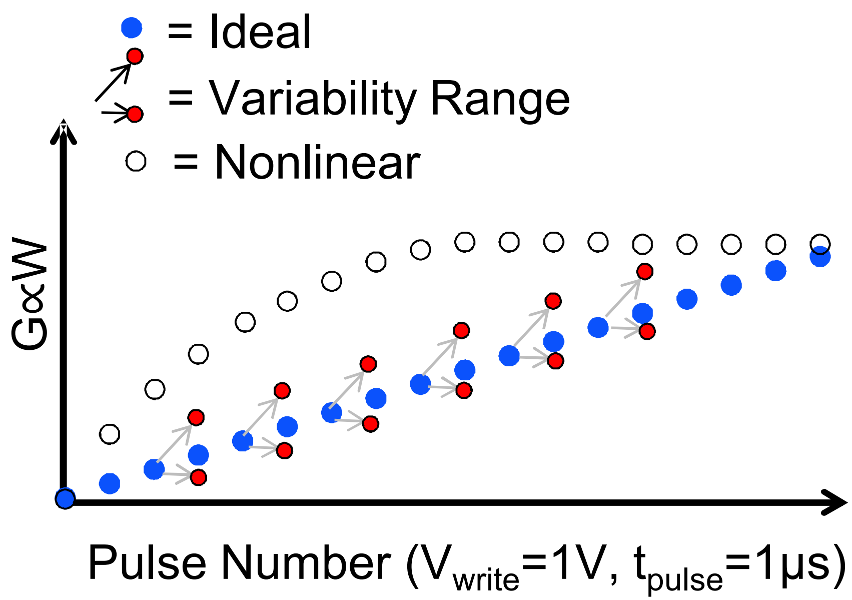
*we are assuming we need some fixed precision on the output, and don't need full floating point accuracy

Experimental Device Non-idealities

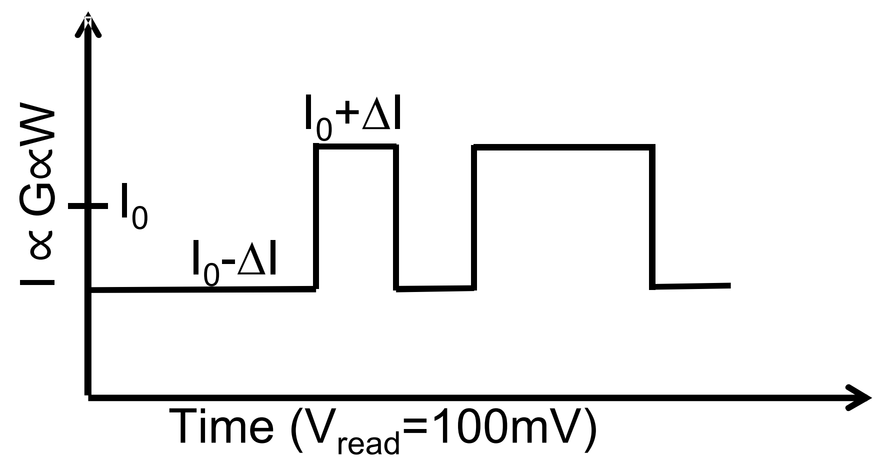
Device: Write Variability, Write Nonlinearity, Asymmetry, Read Noise

Circuit: A/D, D/A noise, parasitics

Variability and Nonlinearity

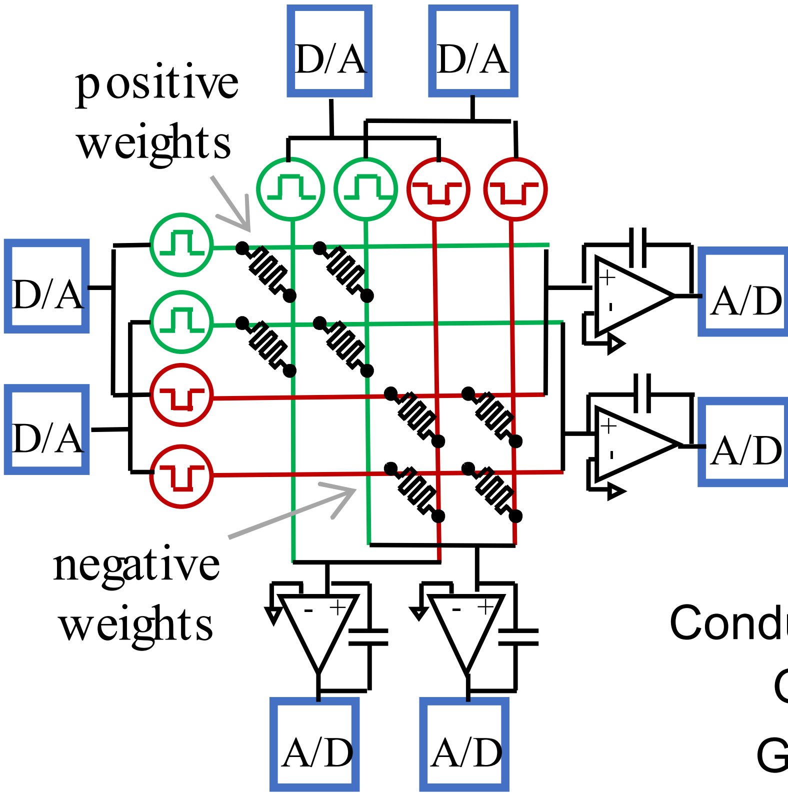


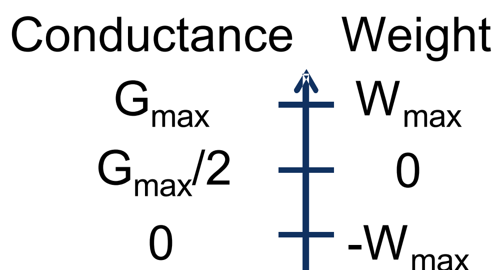
Read Noise

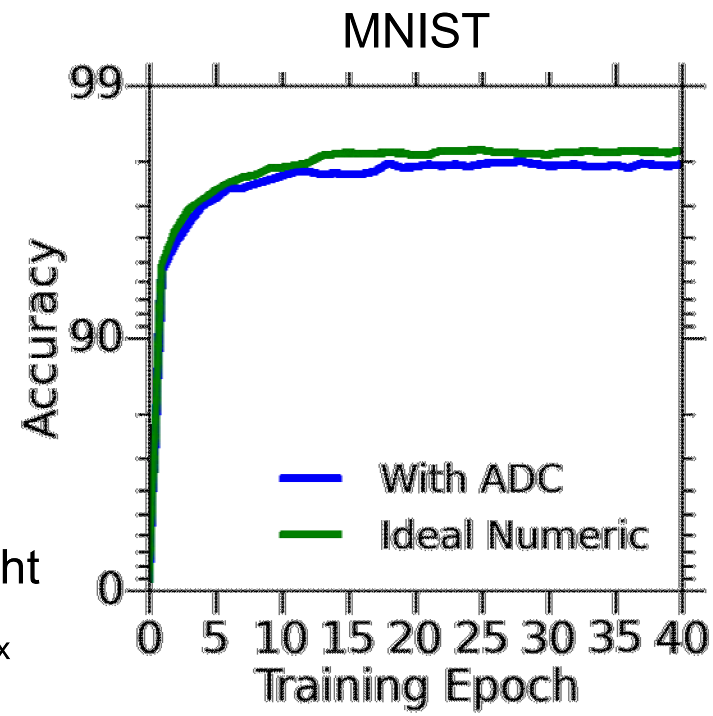


Full System Simulation

A/D & D/A Have Minimal Impact



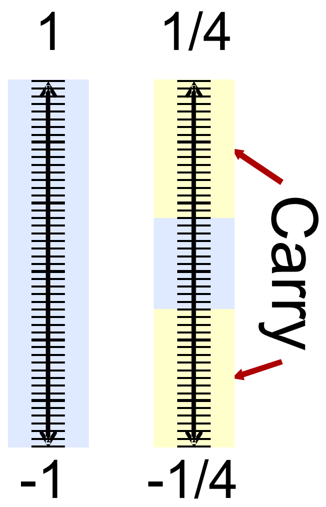


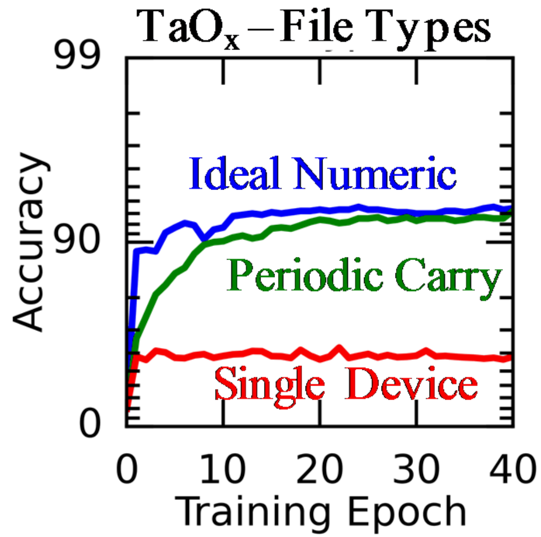


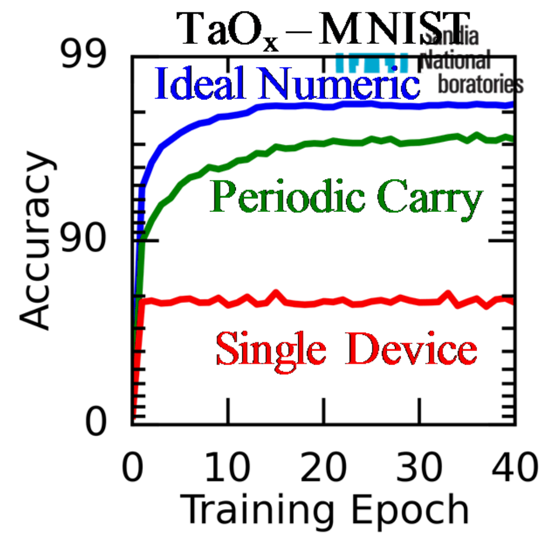
	Range	Bits
Row Input	-1 to 1	8
Col Output	-6 to 6	8
Col Input	-1 to 1	8
Row Output	-4 to 4	8
Row Update	-0.01 to 0.01	7
Col Update	-1 to 1	5

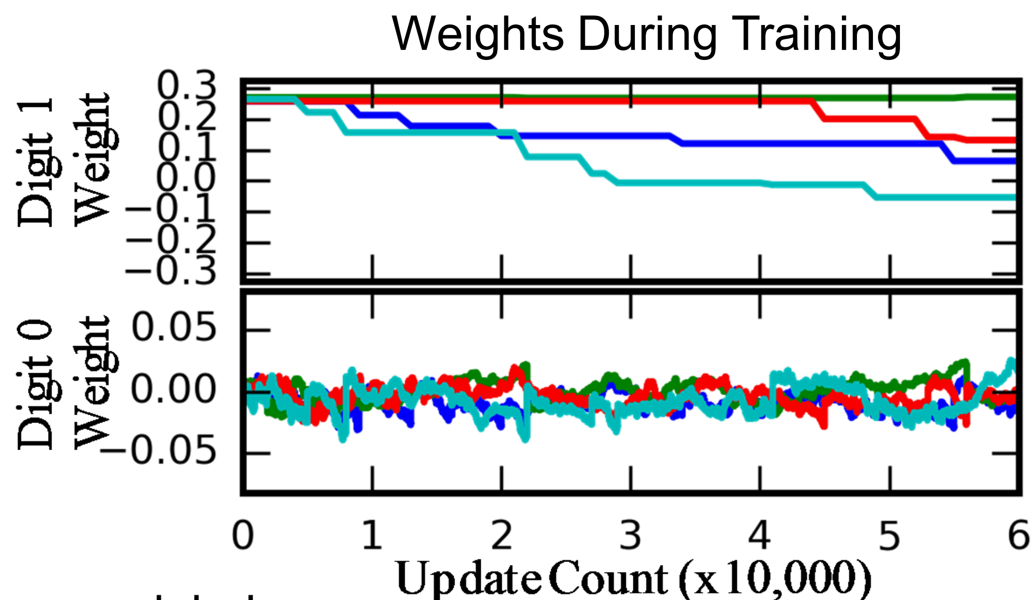
Data set	#Training/Test Examples	Network Size
File Types	4,501 / 900	256×512×9
MNIST	60,000 /10,000	784×300×10

TaO_x Results







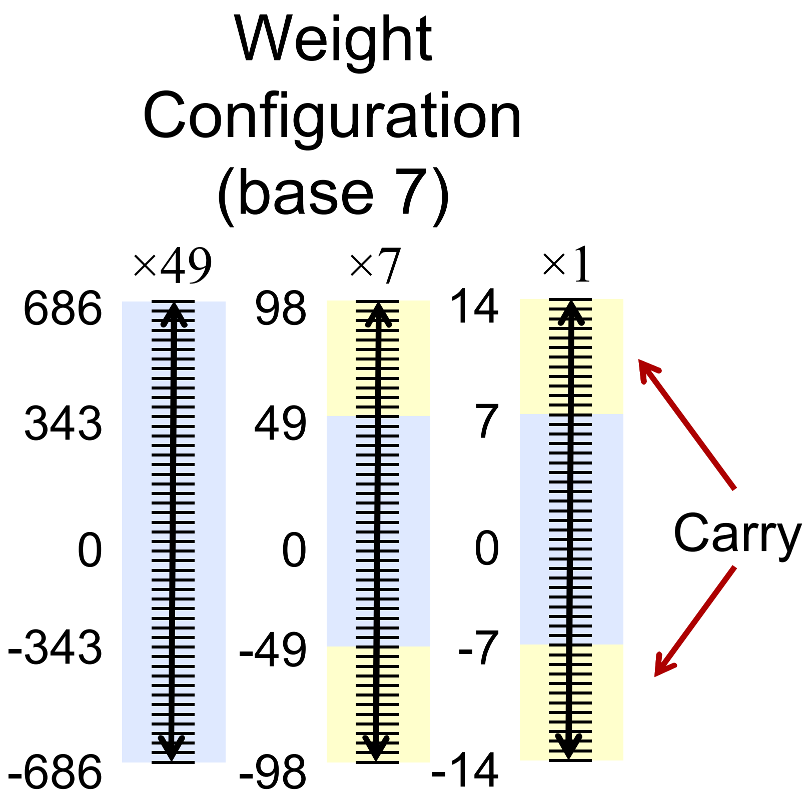


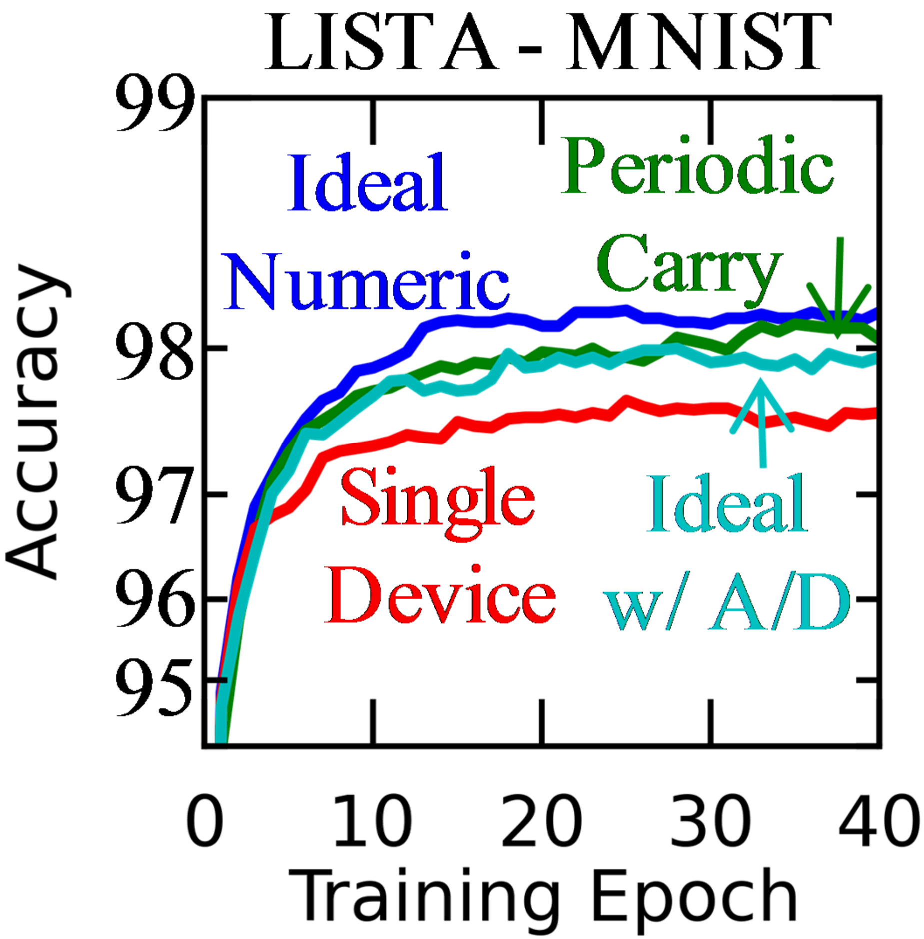
Carry once every 1000 updates for the LSB, and every 2 updates on others

A/D and D/A is modeled, serial operations modeled

- When resetting weight, need to adjust pulse size based on current state to compensate for nonlinearity
- When reading a single weight, need to adjust readout range to be smaller (change capacitor on the integrator)

LISTA Results

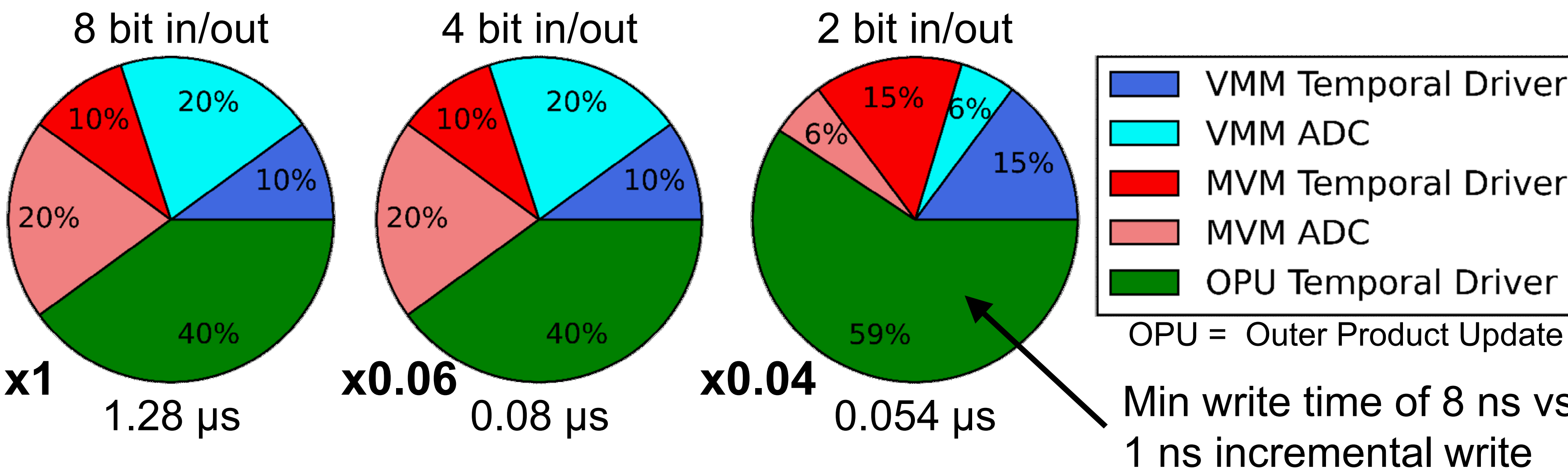




- Carry once every 1000 updates
- Use a single device per weight and subtract a reference current

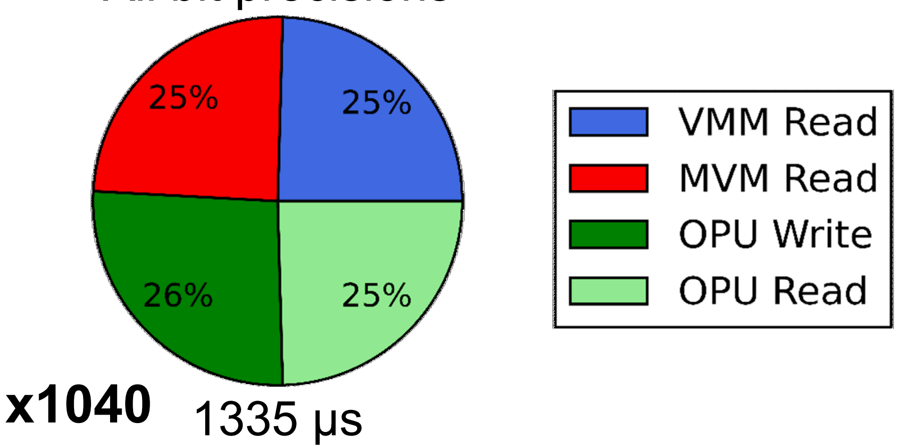
Neural Core Latency Analysis

Analog ReRAM



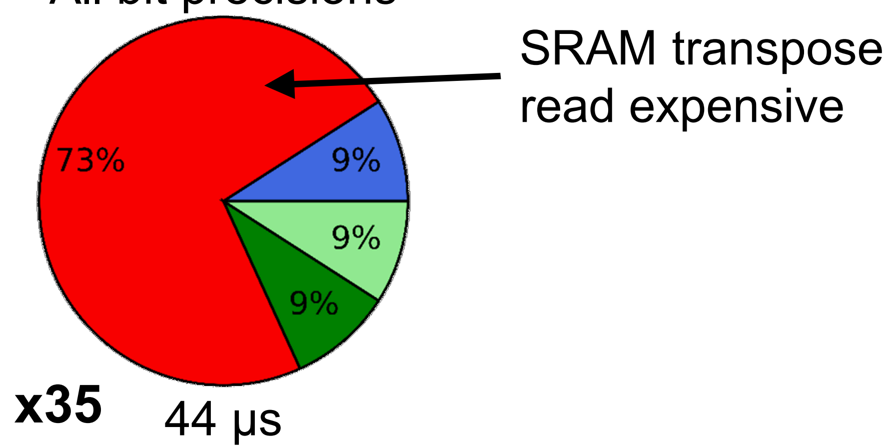
Digital ReRAM

All bit precisions

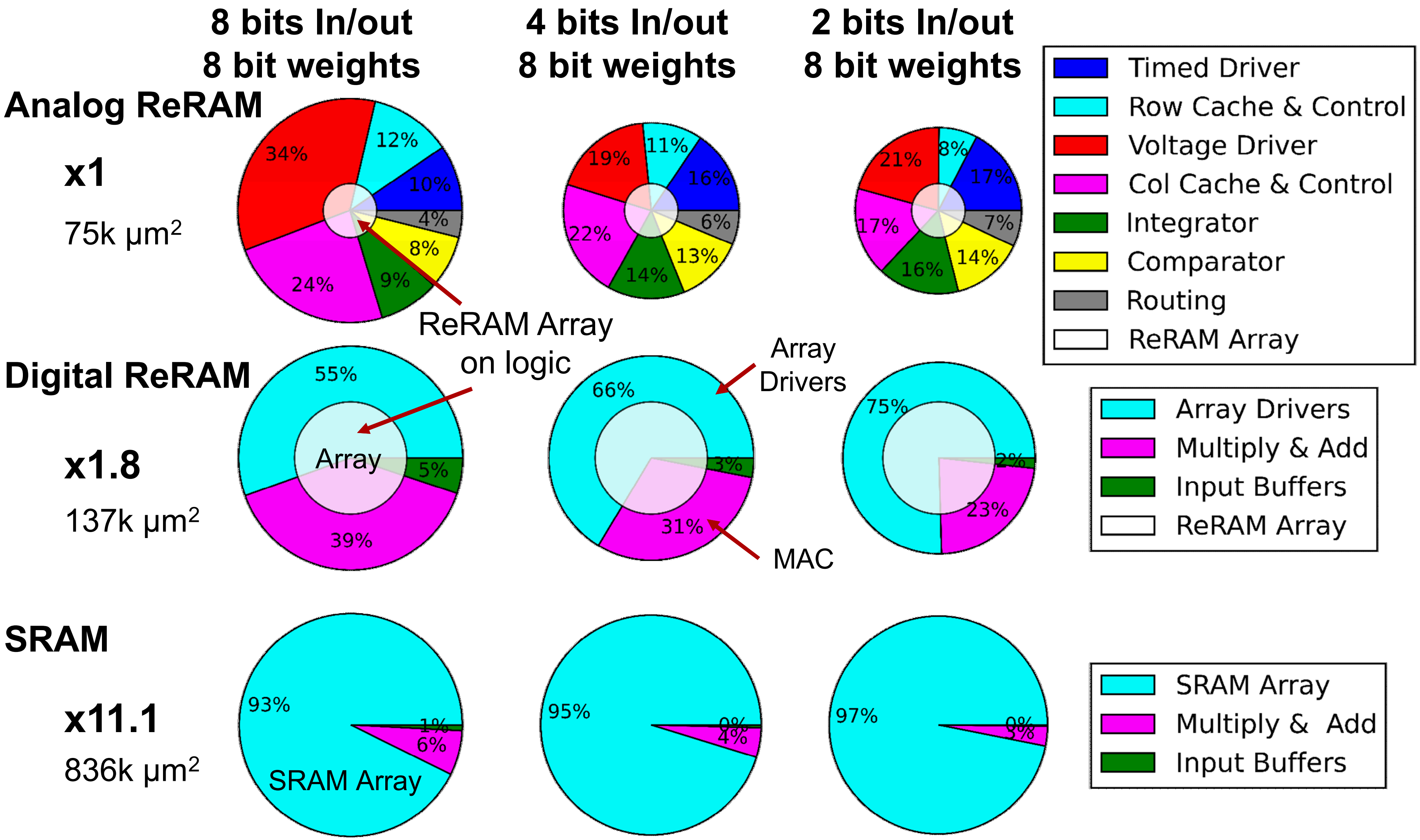


SRAM

All bit precisions



Neural Core Area Analysis



For the ReRAM, high voltage transistors require 8X area, improving this could give ~2X area savings