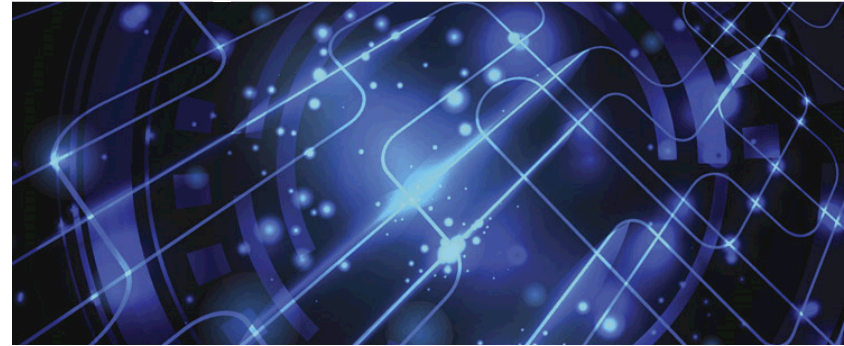
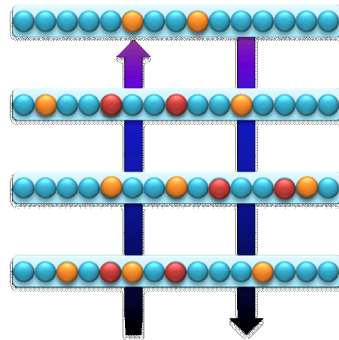
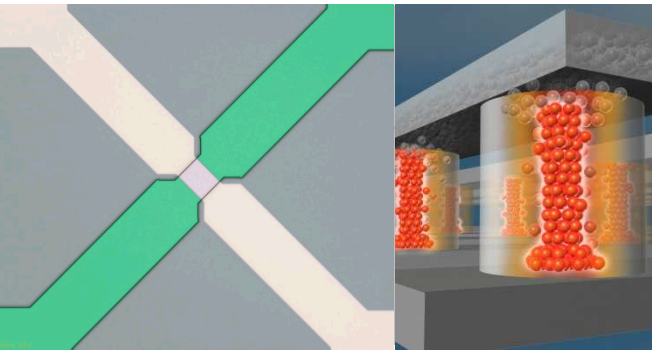


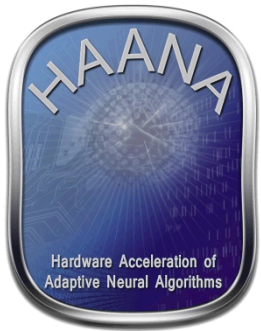
[Redacted text]

[Redacted text]



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



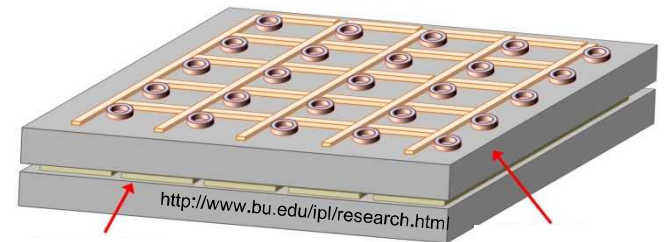
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

The Von Neumann Bottleneck



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

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



Processor Layer

Photonic Layer

Optical interconnects 100 fJ to 1 pJ

Communications require orders of magnitude more energy!

Use Resistive Memories for Local Computation

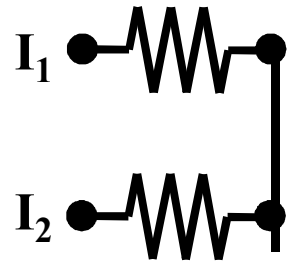


$$V = I \times R$$

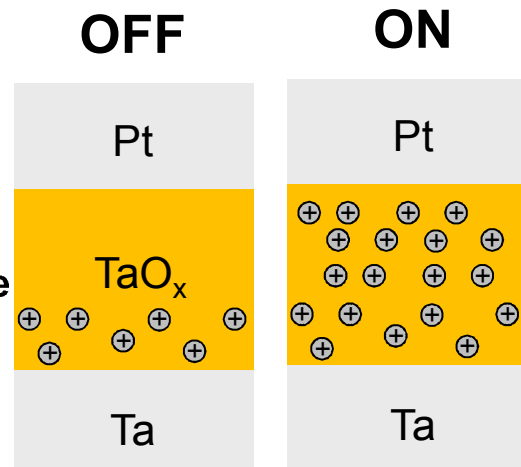
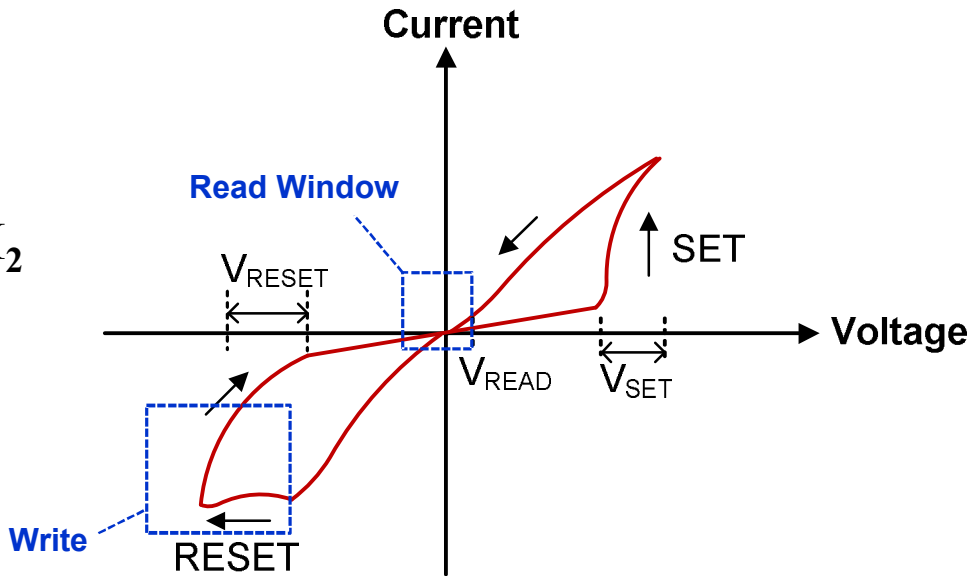
$$I = G \times V$$

← multiplication

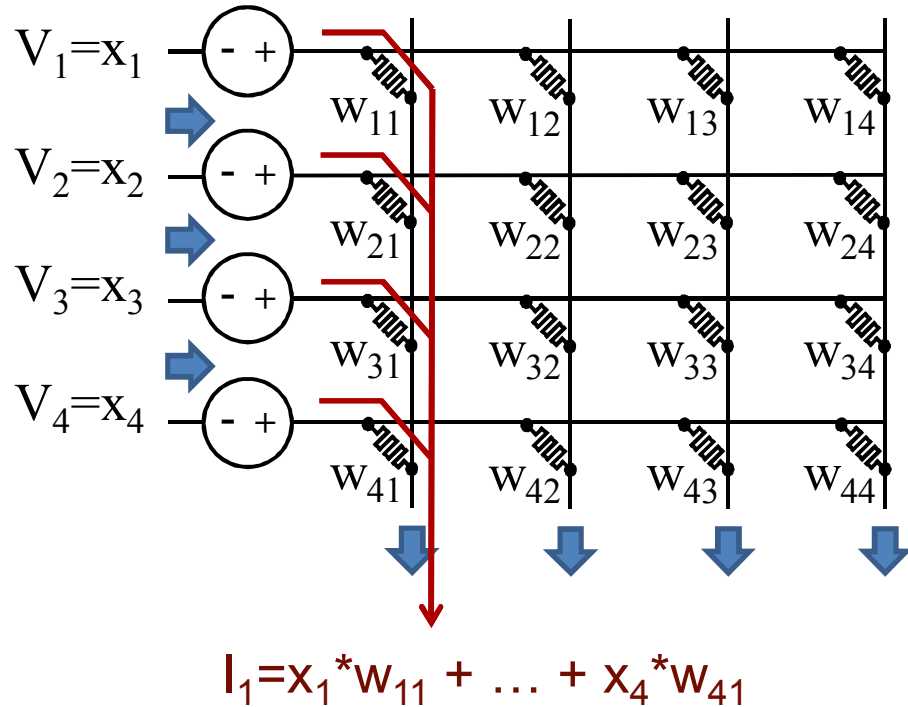
- A resistive memory or ReRAM is a programmable resistor
 - Apply small voltages allows the conductance to be read: $I = G \times V$
 - Apply large voltages to change the resistance



Addition: $I = I_1 + I_2$



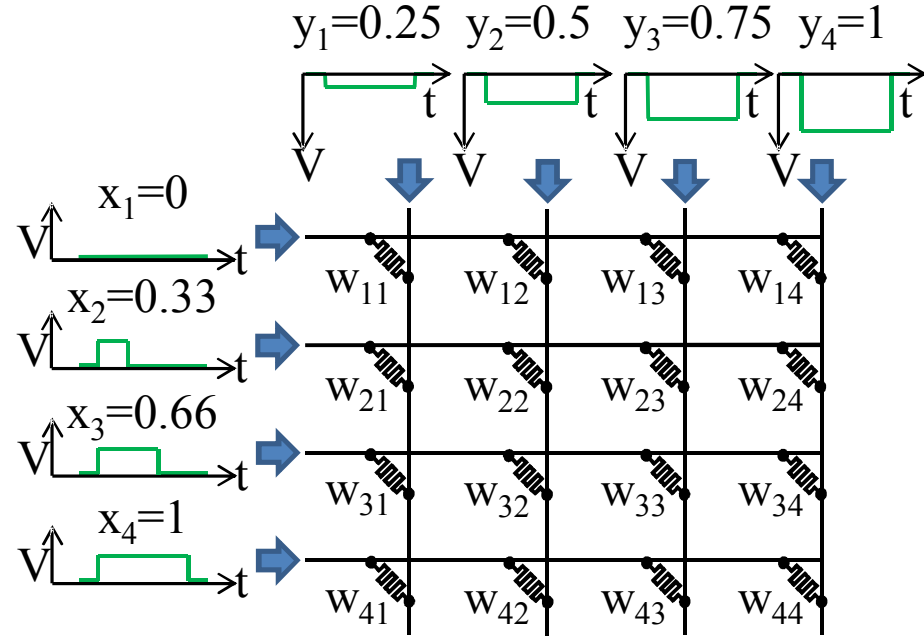
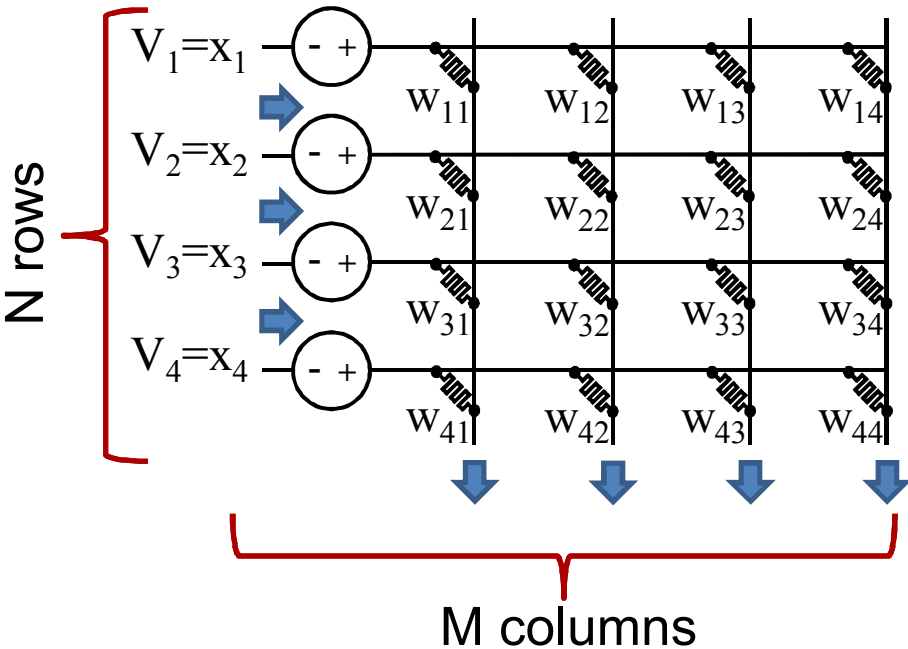
Directly Process in the Memory Itself



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

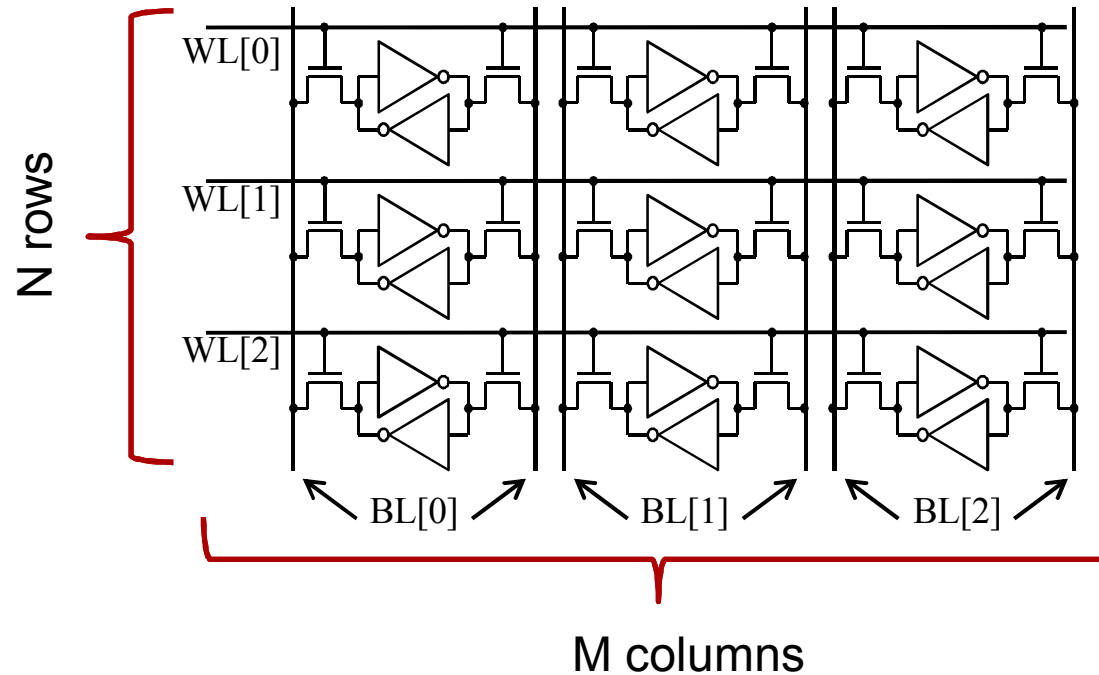
Crossbars Can Perform Parallel Reads and Writes



Energy to charge the crossbar is CV^2
 $E \propto C \propto \text{number of RRAMs} \propto N \times M$

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

SRAM Arrays Require Charging Columns Multiple Times



SRAMs must be read one row at a time, charging M columns
Each column wire length is $O(N)$.

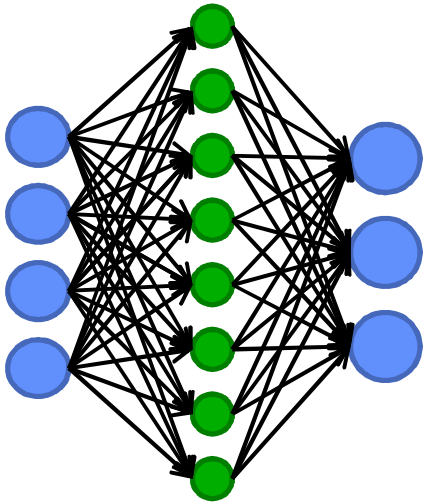
$$\text{Energy} = N \text{ Rows} \times M \text{ Columns} \times O(N) \text{ wire length}$$

$$\text{Energy} \sim O(N^2 \times M)$$

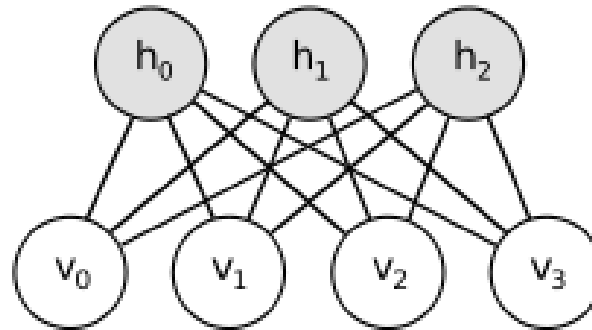
$O(N)$ times worse than a crossbar!

Want To Accelerate Many Different Neural Algorithms

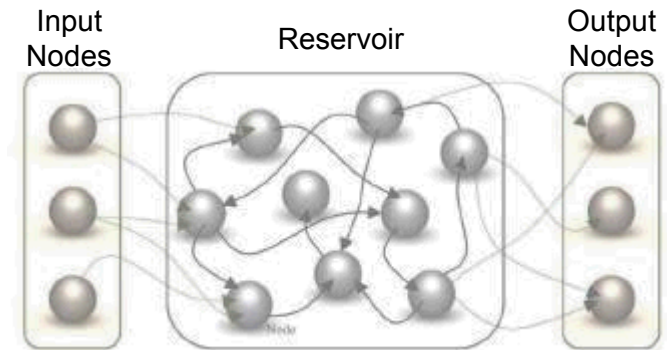
Backpropagation



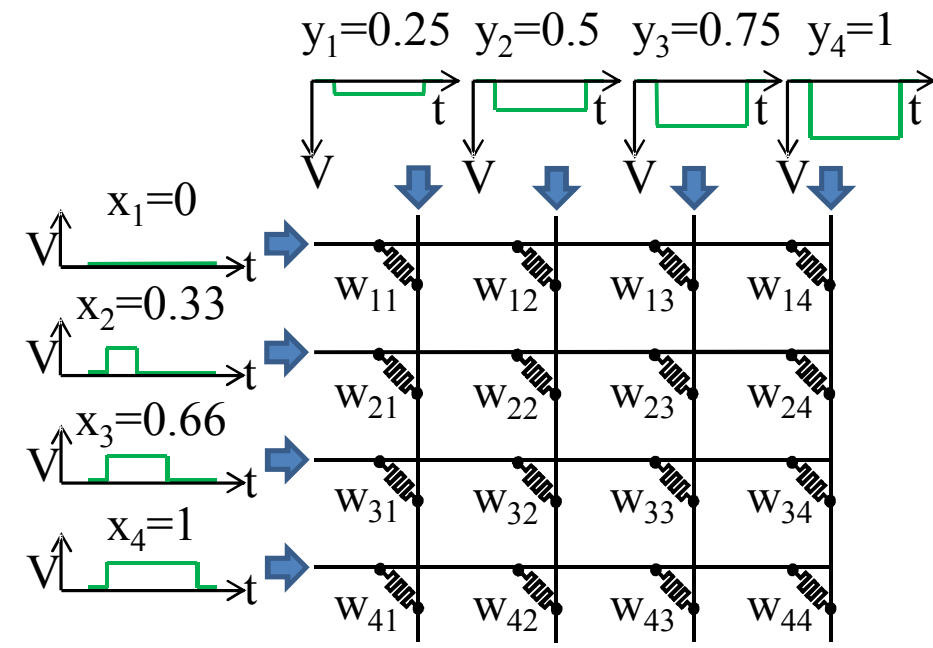
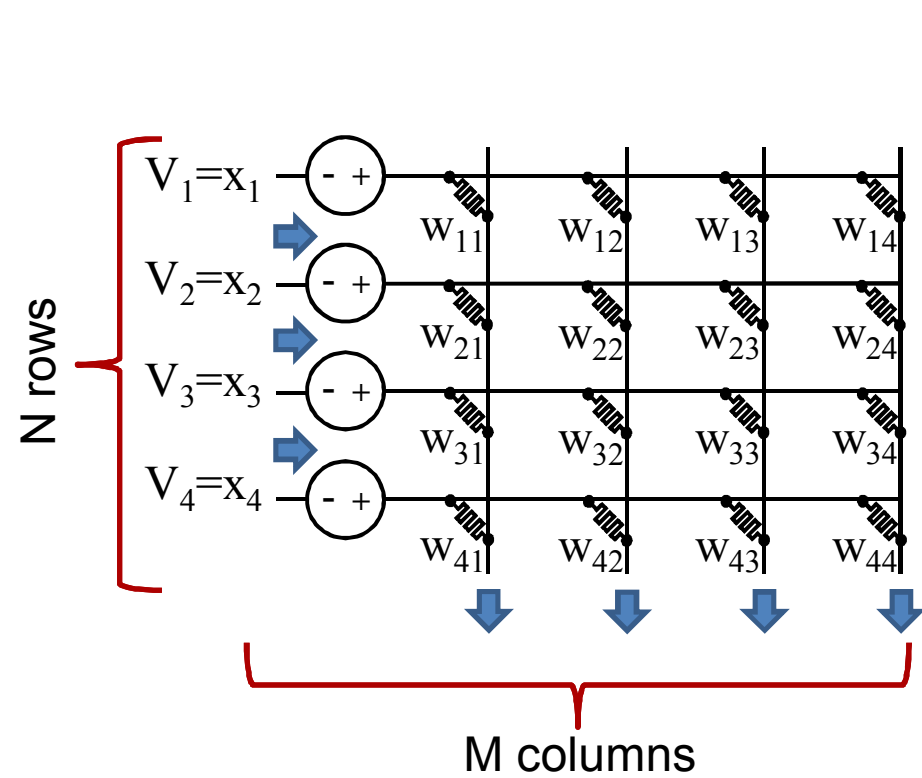
Sparse Coding



Liquid State Machine



Crossbars Can Perform Parallel Reads and Writes

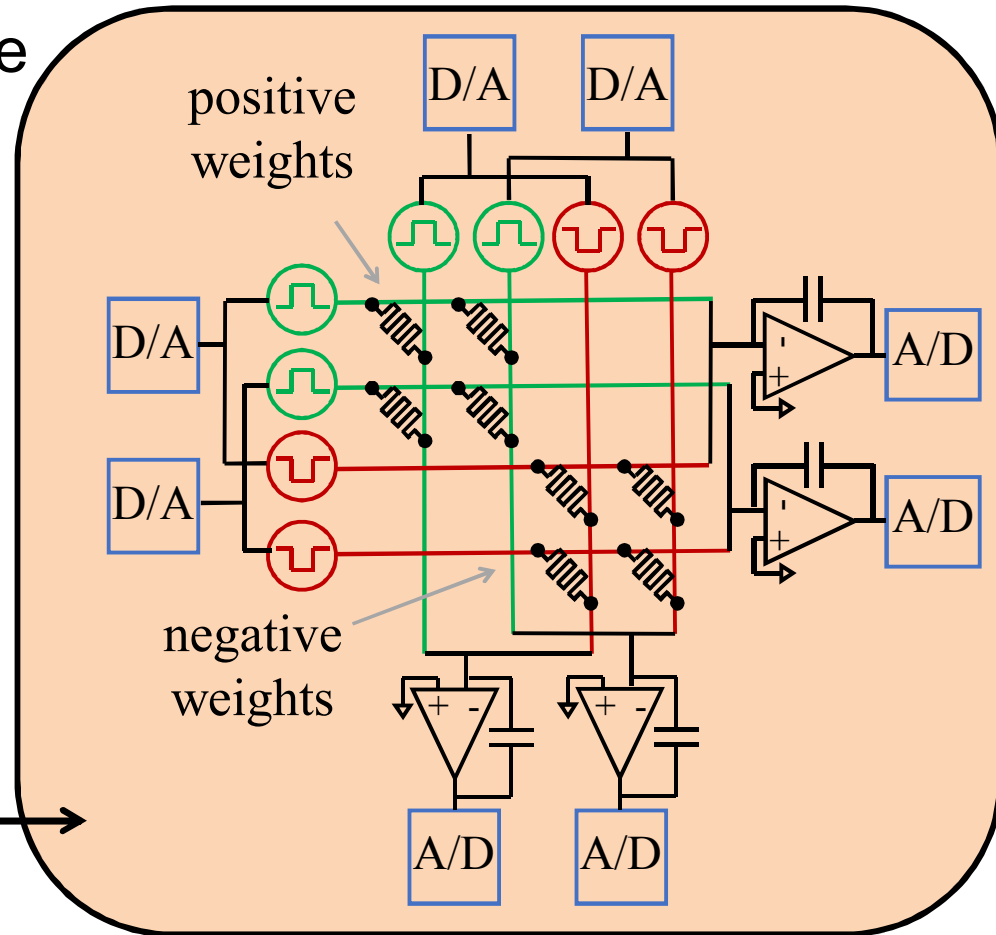
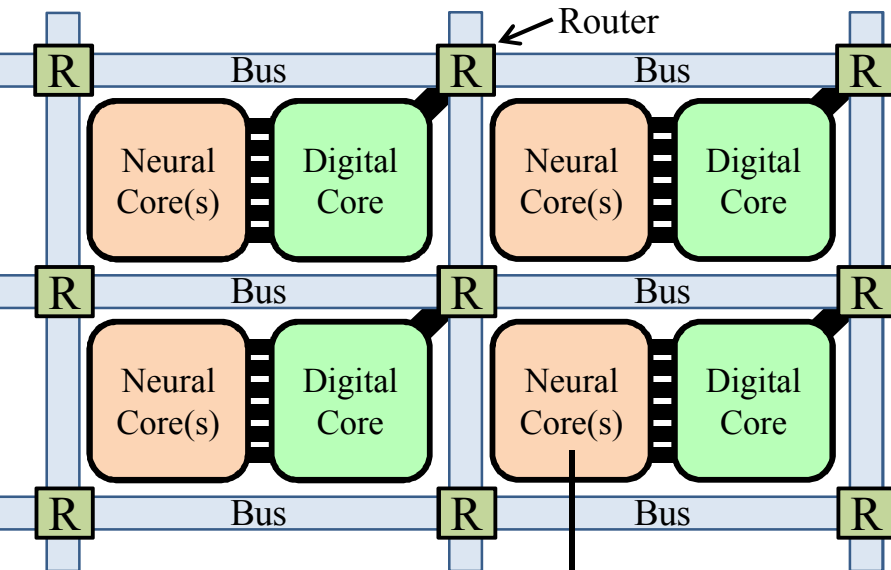


Energy to charge the crossbar is CV^2
 $E \propto C \propto \text{number of RRAMs} \propto N \times M$

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

General Purpose Neural Architecture

Run *any* neural algorithm on the same hardware



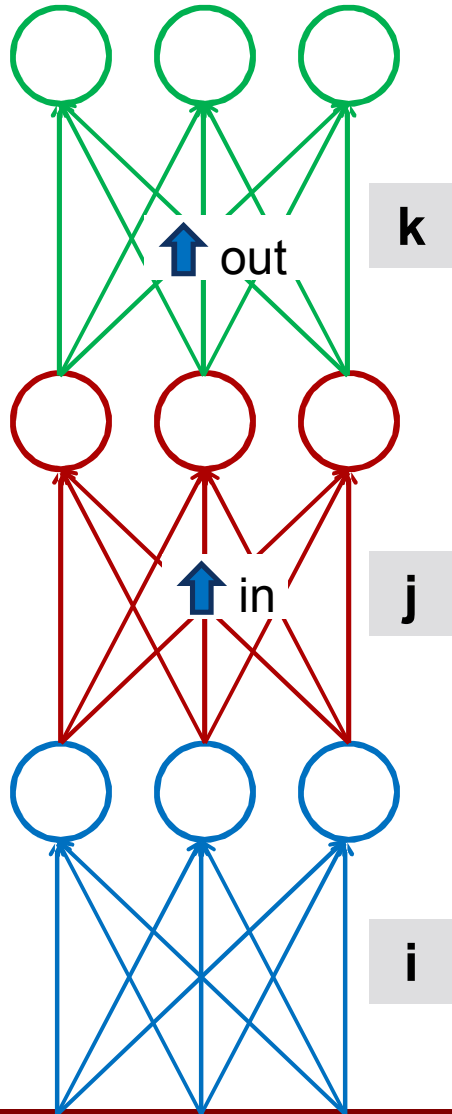
Neuromorphic core:

- 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



$O(N^2)$
Operations

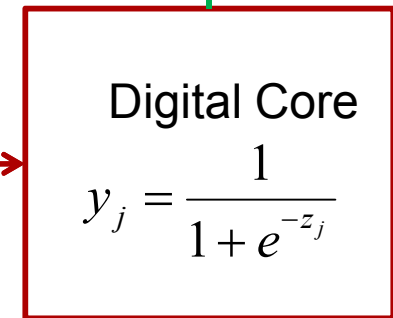
Neural Core



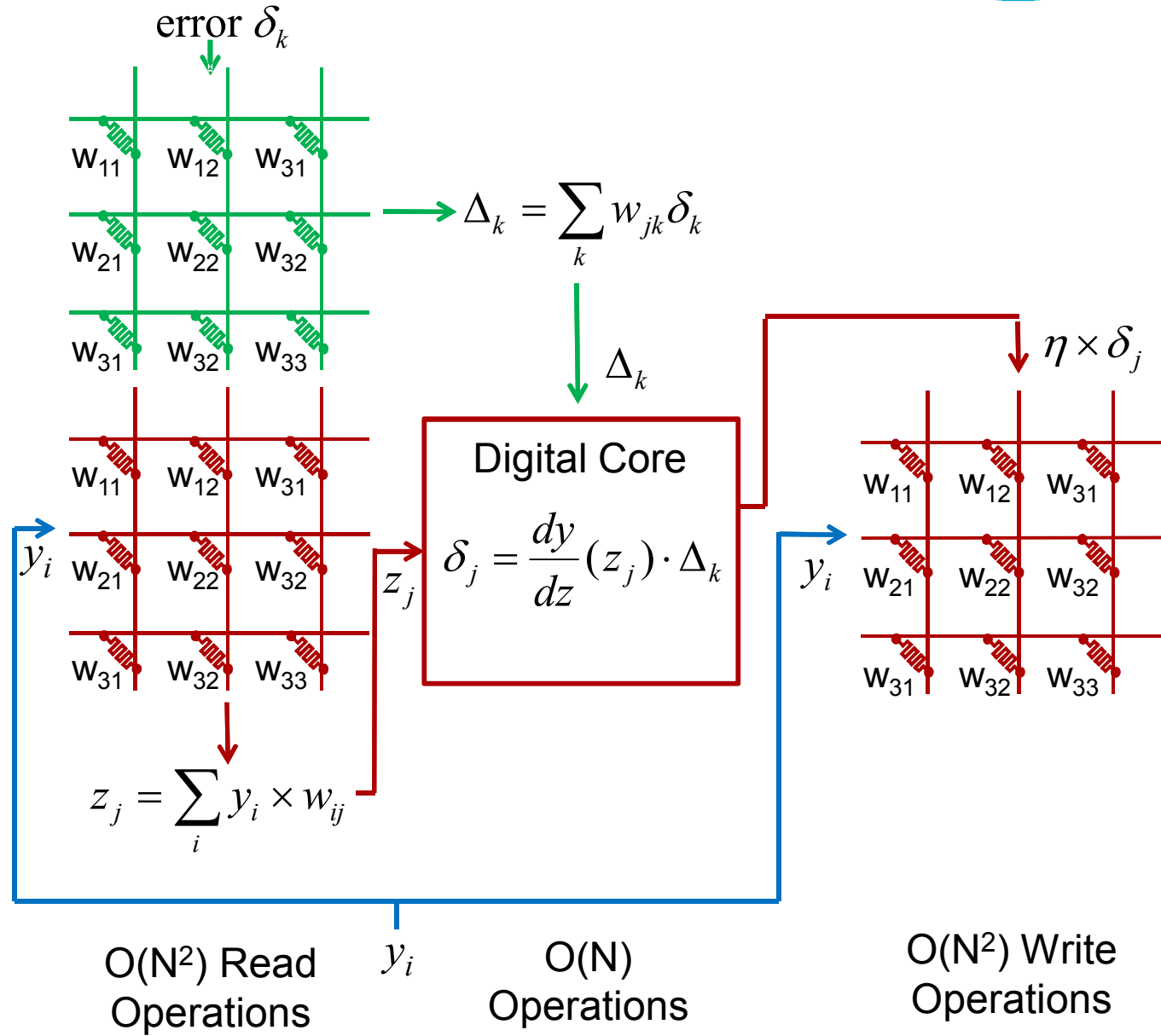
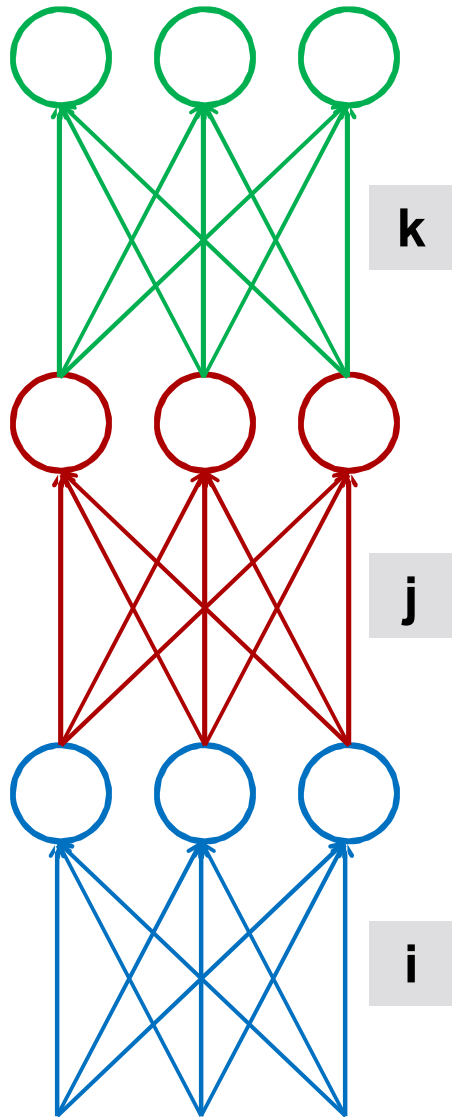
$$z_j = \sum_i y_i \times w_{ij}$$

$O(N)$
Operations

y_j

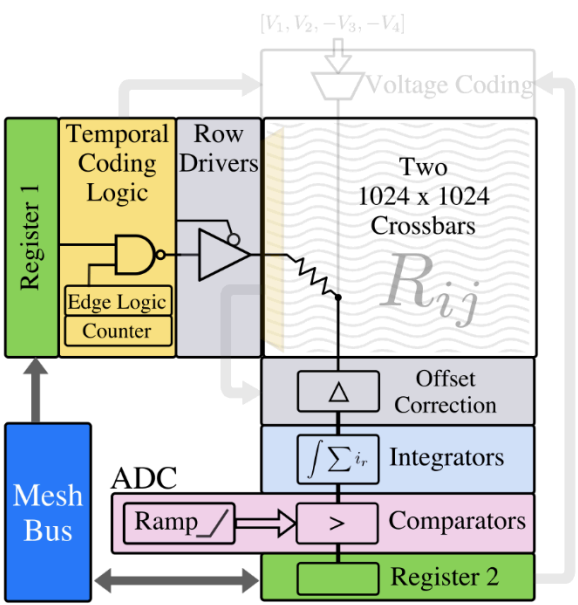


Back Propagation

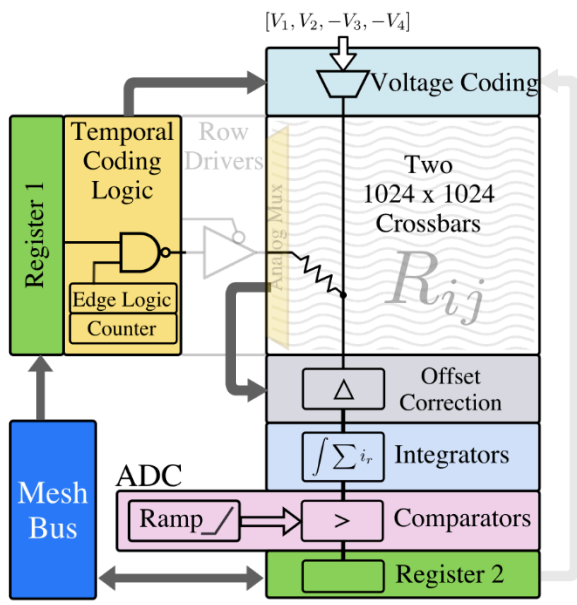


Design & Model Detailed Architecture

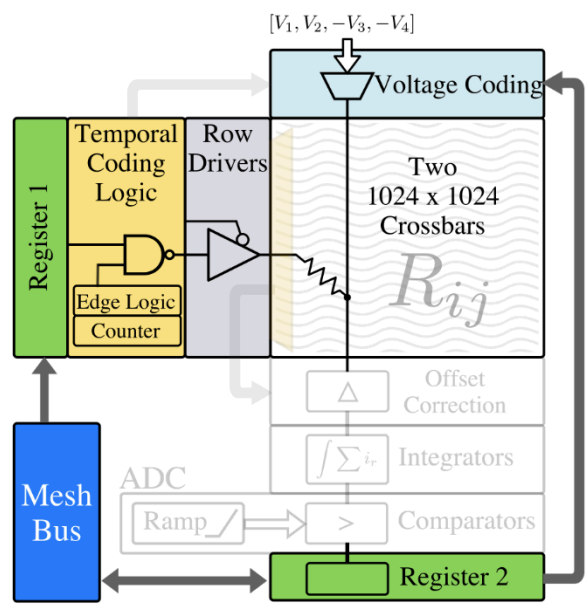
Vector Matrix Multiply



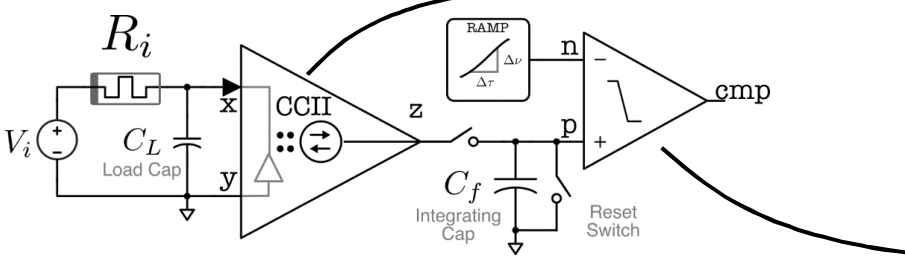
Matrix Vector Multiply



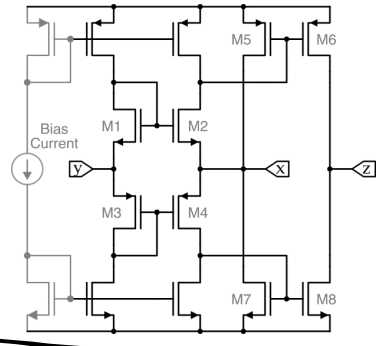
Outer product Update



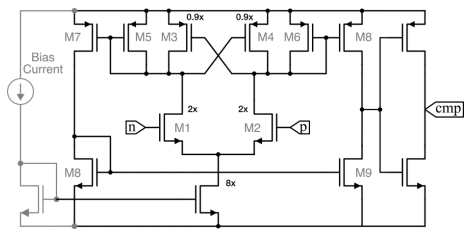
Neuron Circuitry



Current Conveyor Based Integrator

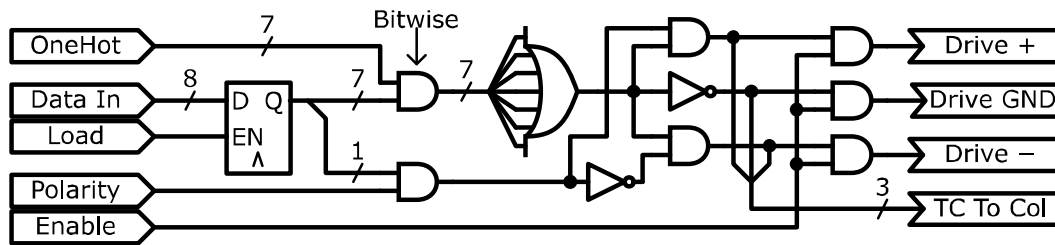


Comparator

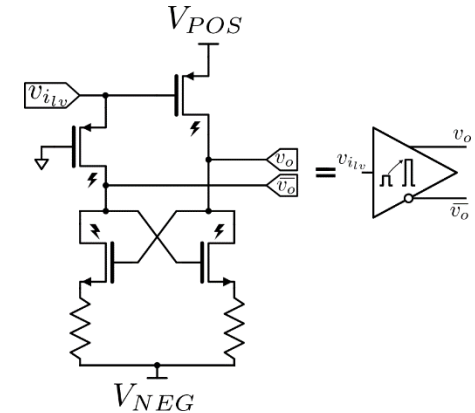


Row & Column Driver Circuitry

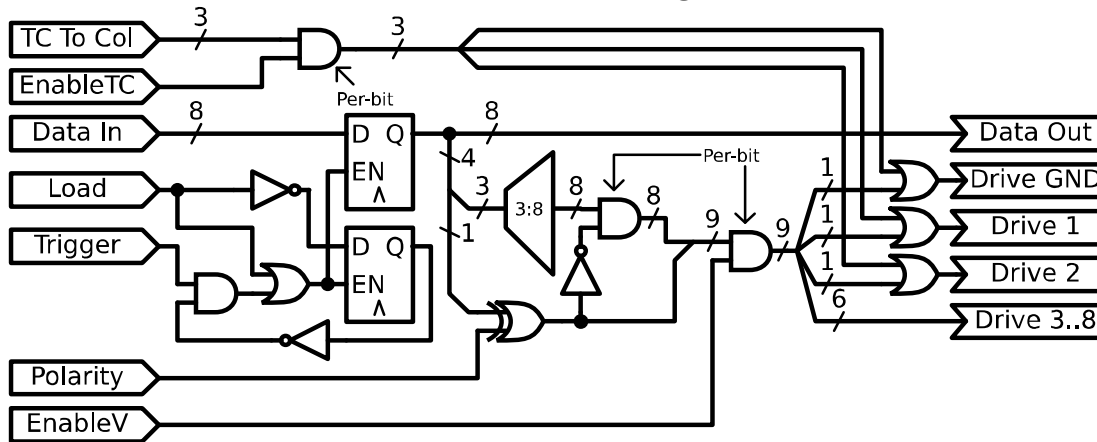
Row Driver Logic



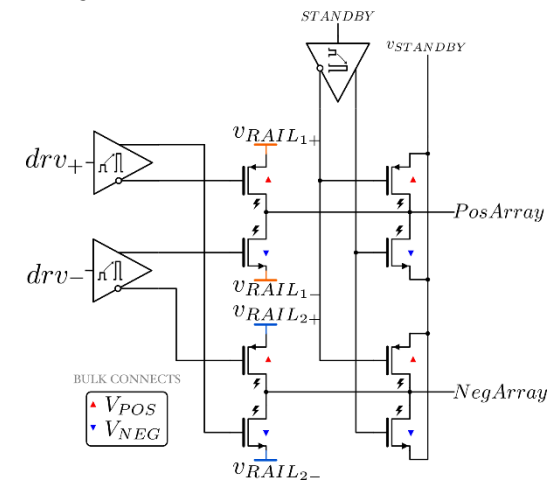
Voltage level shifter (drive high V transistor with low V)



Column Driver Logic



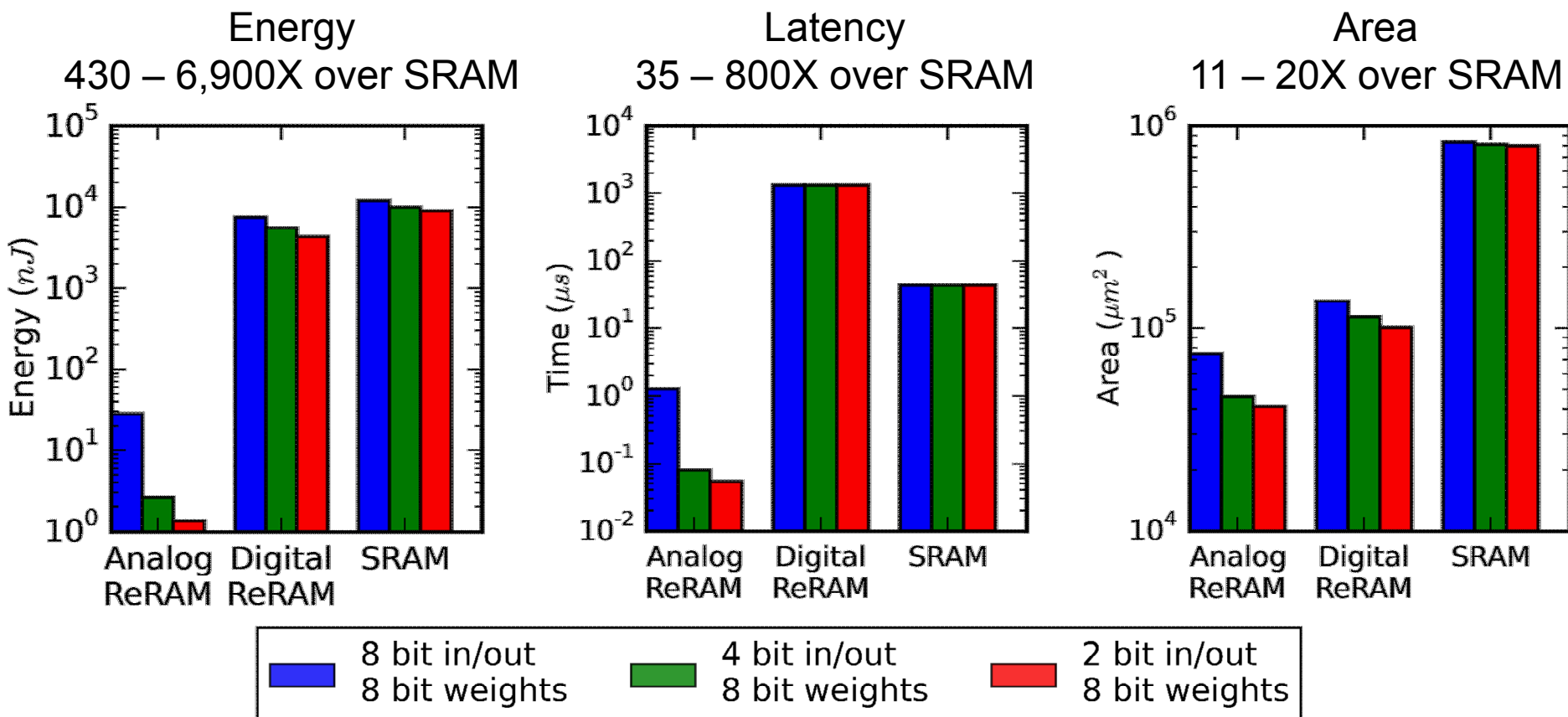
Array driver pass transistors



Compare Architectures

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

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

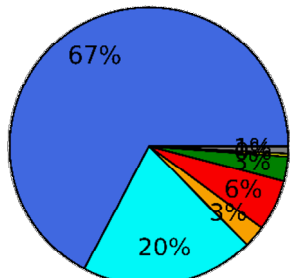


Used a commercial 14/16 nm PDK

***Requires 100 MΩ on state devices

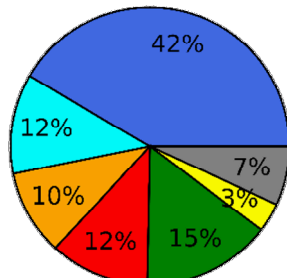
Neural Core Energy Analysis

**8 bits In/out
8 bit weights**



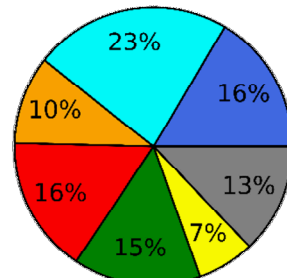
28 nJ

**4 bits In/out
8 bit weights**

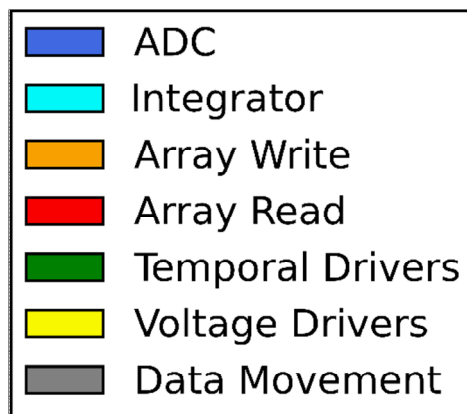


2.7 nJ

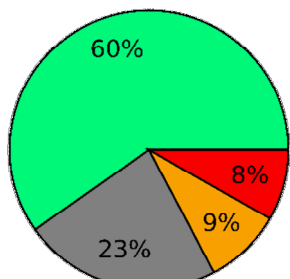
**2 bits In/out
8 bit weights**



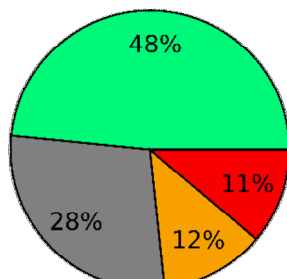
1.3 nJ



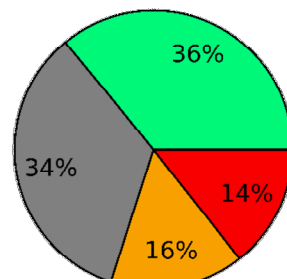
**Analog
ReRAM**



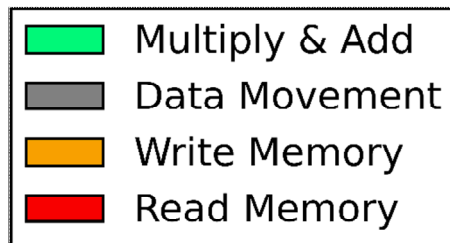
7,520 nJ



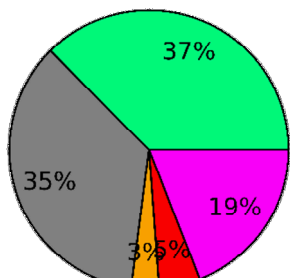
5,580 nJ



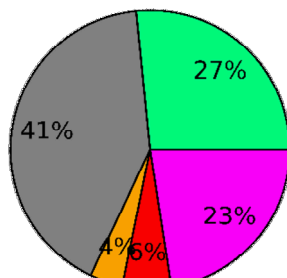
4,340 nJ



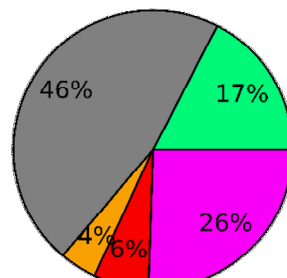
**Digital
ReRAM**



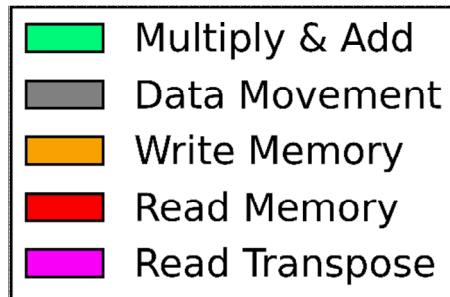
12,010 nJ



10,150 nJ

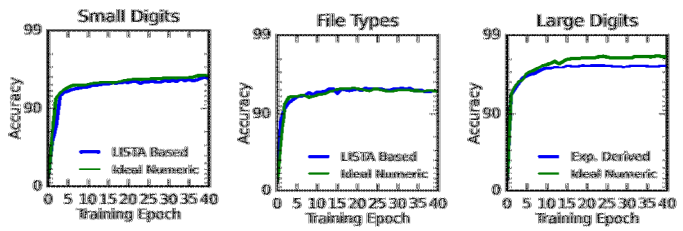


8,970 nJ



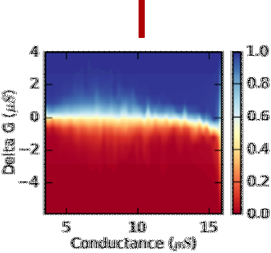
SRAM

Multiscale Model of a Neural Training Accelerator



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

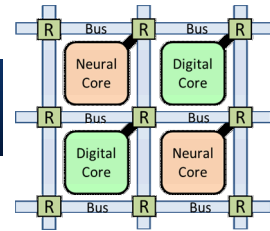
Algorithms



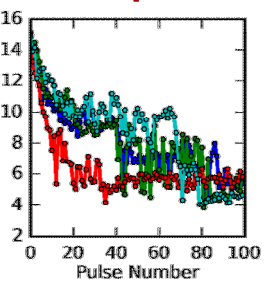
#ROSS SIM

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

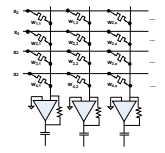
Architecture



Modified McPAT/CACTI:
Model performance and energy requirements



Memristor fabrication and measurements in MESA Fab

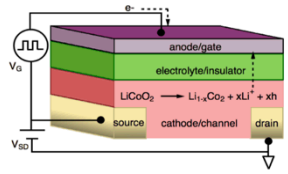


Circuits

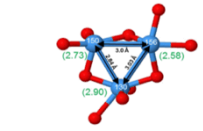
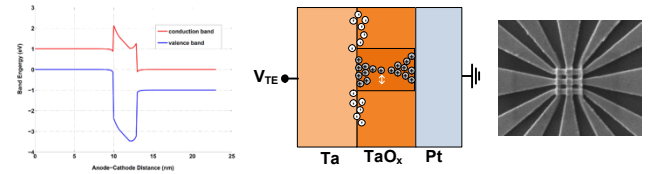


Sandia's Xyce Circuit Sim: Simulate crossbar circuits based on our devices

Devices



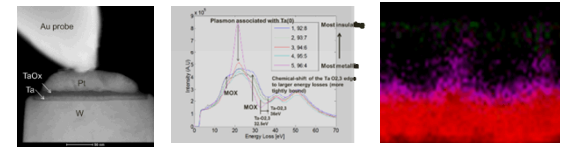
Drift-diffusion model of ReRAM band diagram & transport (REOS, Charon)



DFT of model of oxide physics, bands

Materials

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



#ROSS SIM



<https://cross-sim.sandia.gov>

Crossbar Simulator

#ROSS 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.

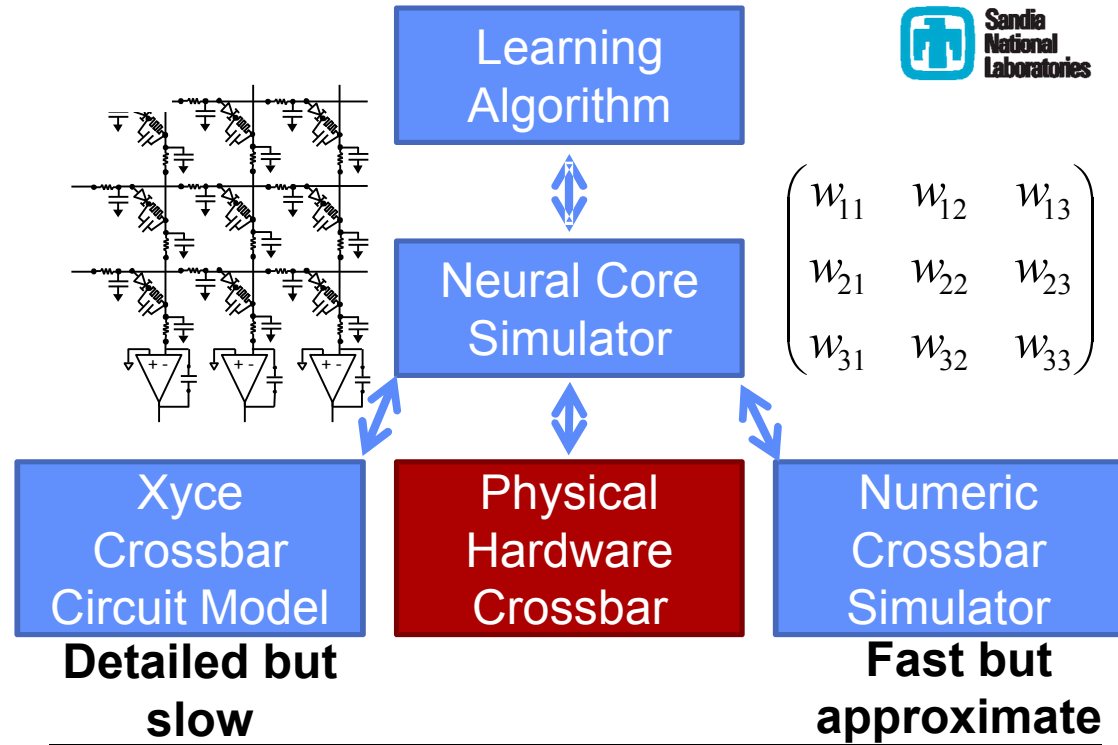
$$\begin{matrix}
 V_1 = X_1 & \oplus & & & & \\
 V_2 = X_2 & \oplus & W_{11} & W_{12} & W_{13} & W_{14} \\
 V_3 = X_3 & \oplus & W_{21} & W_{22} & W_{23} & W_{24} \\
 V_4 = X_4 & \oplus & W_{31} & W_{32} & & W_{33} & W_{34} \\
 & & W_{41} & W_{42} & W_{43} & W_{44}
 \end{matrix}$$

Download

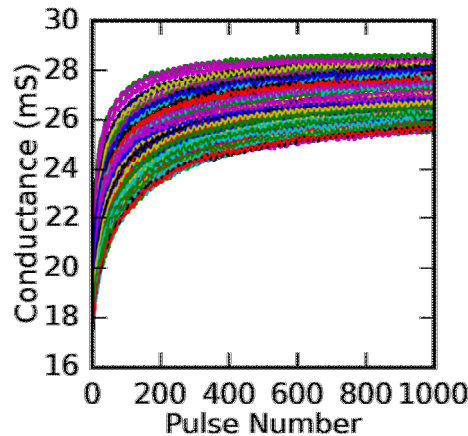
Contact Us

Selected Publications Using CrossSim

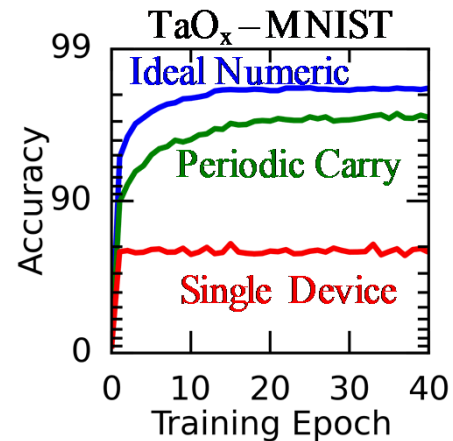
- S. Agarwal, R. B. Jacobs-Gedrim, A. H. Hsia, D. R. Hughtart, E. J. Fuller, A. A. Talin, C. D. James, S. J. Plimpton, and M. J. Marinella, "Achieving Ideal Accuracies in Analog Neuromorphic Computing Using Periodic Carry," in 2017 IEEE Symposium on VLSI Technology Kyoto, Japan, 2017.
- Y. van de Burst, E. Lubberman, E. J. Fuller, S. T. Keane, G. C. Eria, S. Agarwal, M. J. Marinella, A. Alar-Talin, and A.



Measured Devices



Algorithmic Performance



Simple Python API:

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
...
...

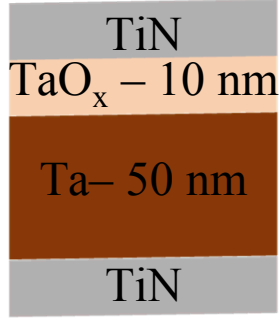
# ***** API for running neural operations
#           All crossbar details are transparent to the user

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

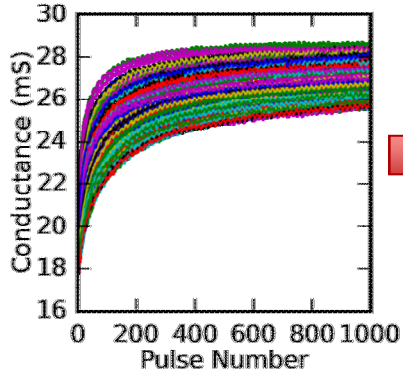
neural_core.set_matrix(weights) # set the initial weights
result = neural_core.run_xbar_vmm(vector) # Do a vector matrix multiply
result = neural_core.run_xbar_mvm(vector) # Do the transpose, a matrix vector mult.
neural_core.update_matrix(vector1,vector2) # Do an outer product update
```

Go from Measurement to Accuracy

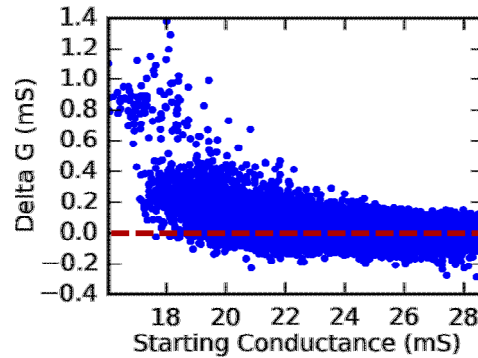
Fabricate Device



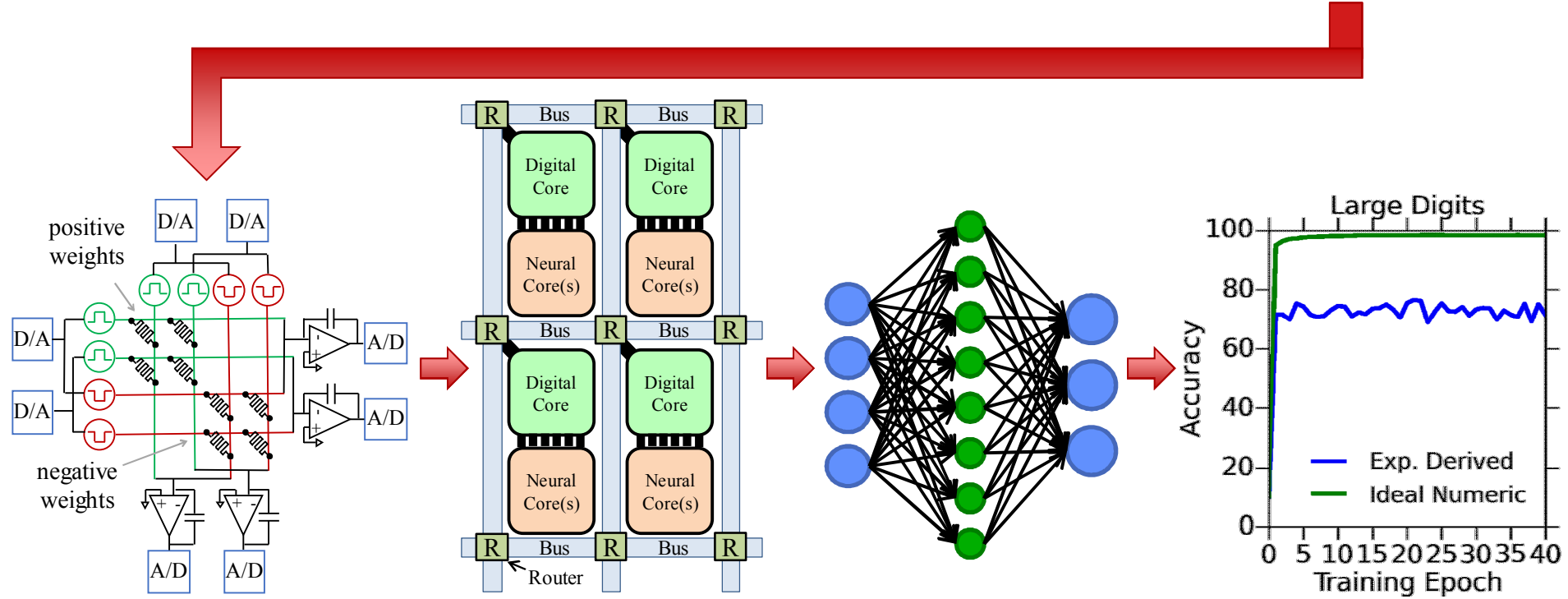
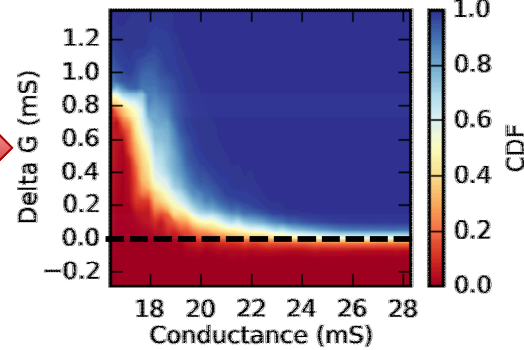
Measured Pulsing



ΔG Scatterplot



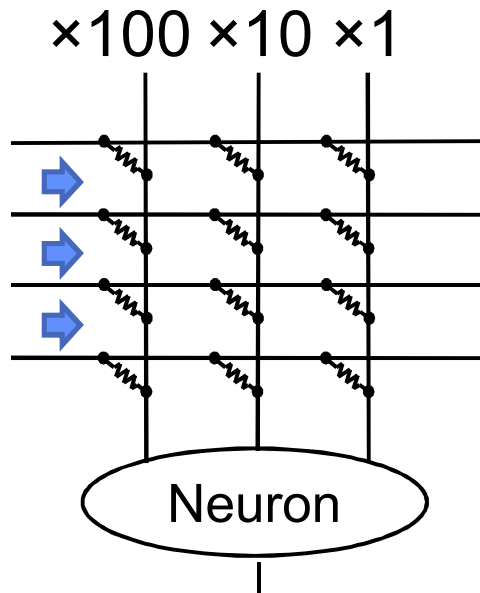
Cumulative Probability of ΔG



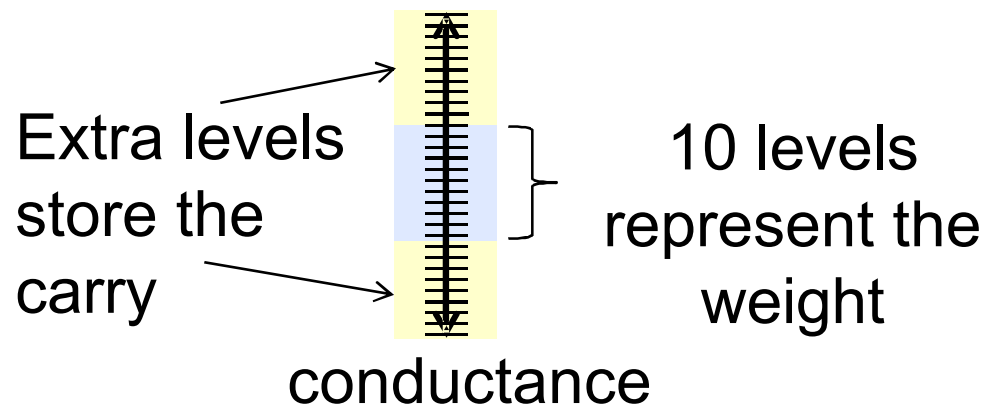
Multi-ReRAM Synapse: Periodic Carry

If we need more bits per synapse, use multiple memristors

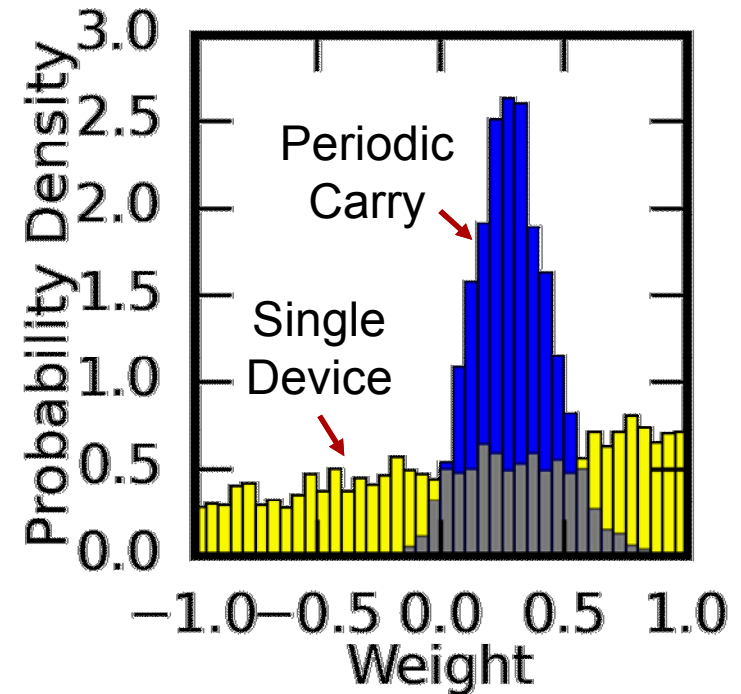
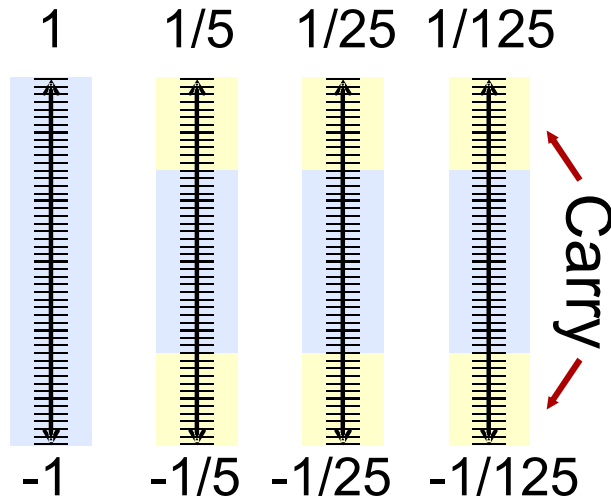
- Three 10 level ReRAMs could represent 1-1000!
- Adding to the weight requires reading every ReRAM to account for any carries and serially programming each ReRAM: VERY EXPENSIVE



- 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

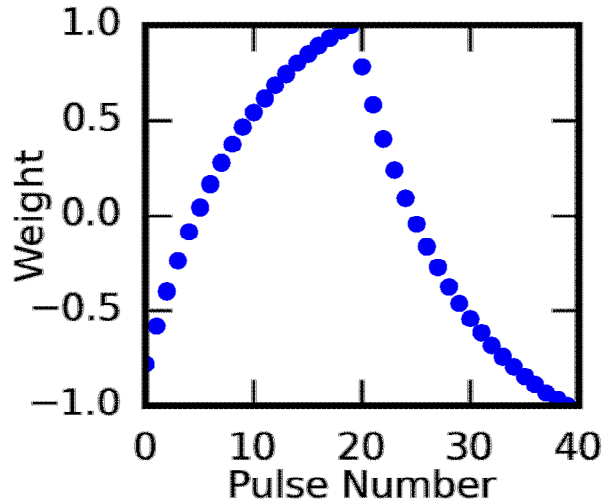
Noise Sigma = 1.4% for single device

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

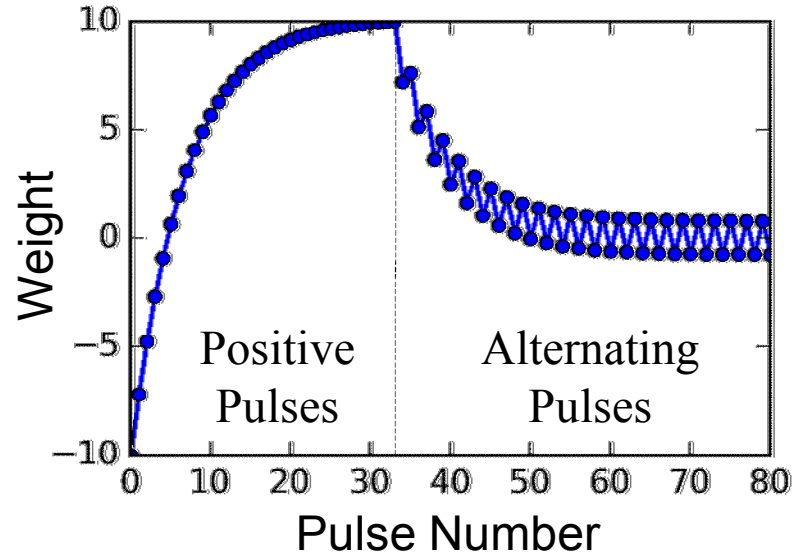
Learn from a 0.5% Signal

Periodic Carry Mitigates Write Nonlinearity

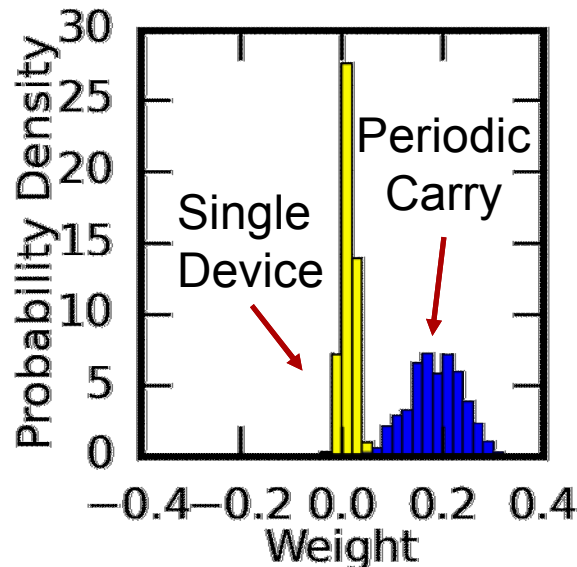
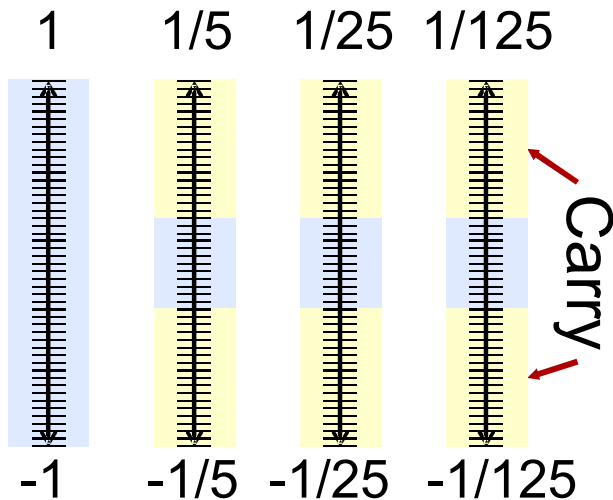
Write Nonlinearity



Alternating Pulses Cause Weight Decay

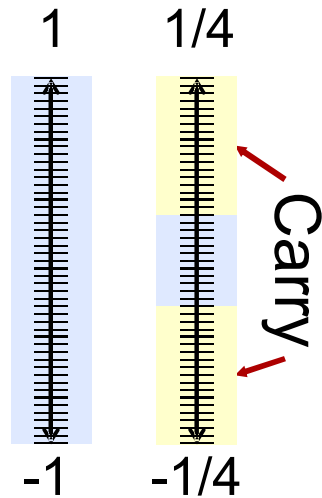


Use center linear range of weights

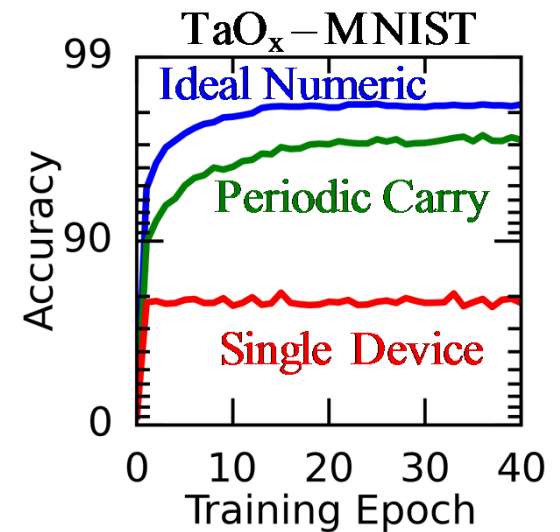
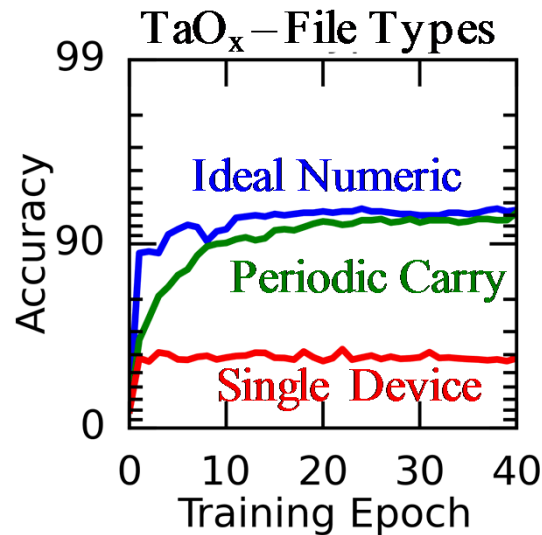


- Train with 1% signal
- Ideal result is 0.6

TaO_x Results



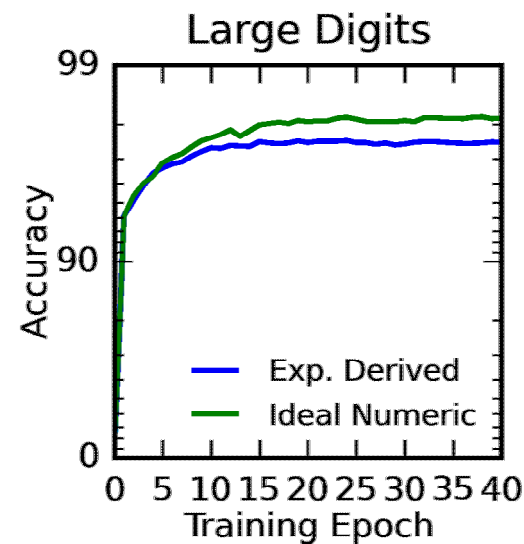
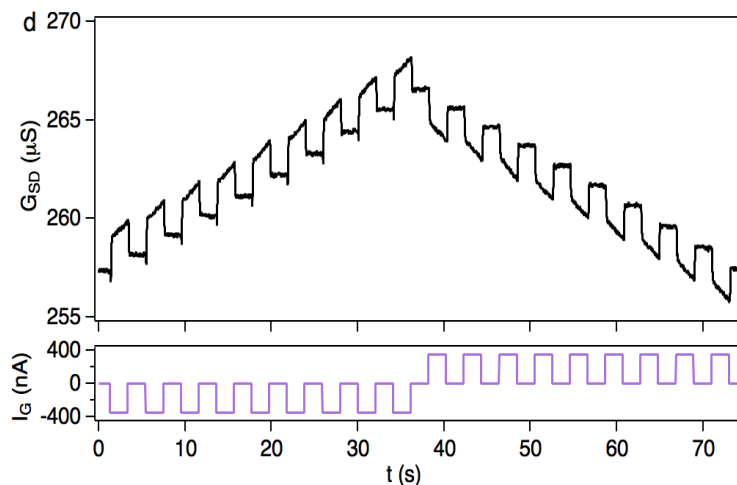
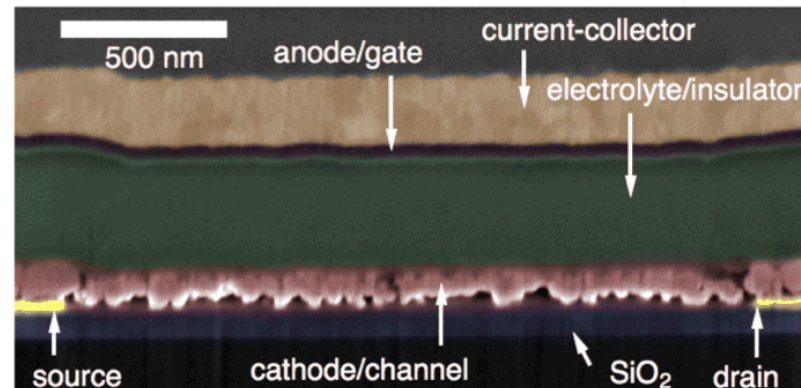
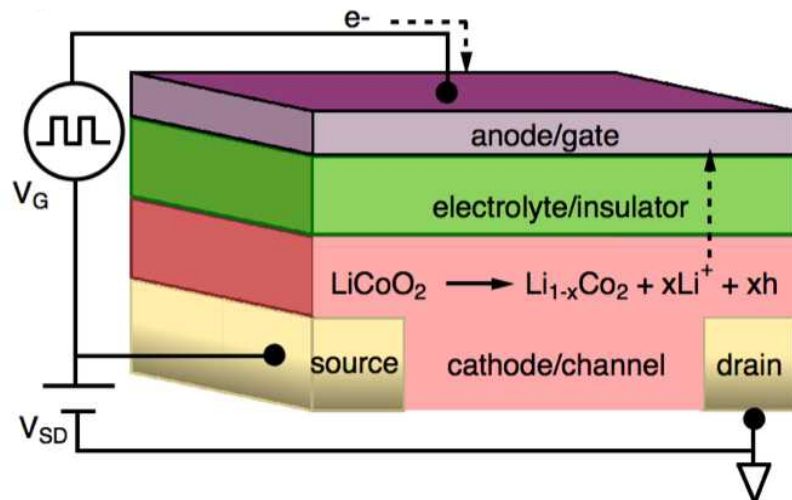
Carry once every
1000 updates



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)

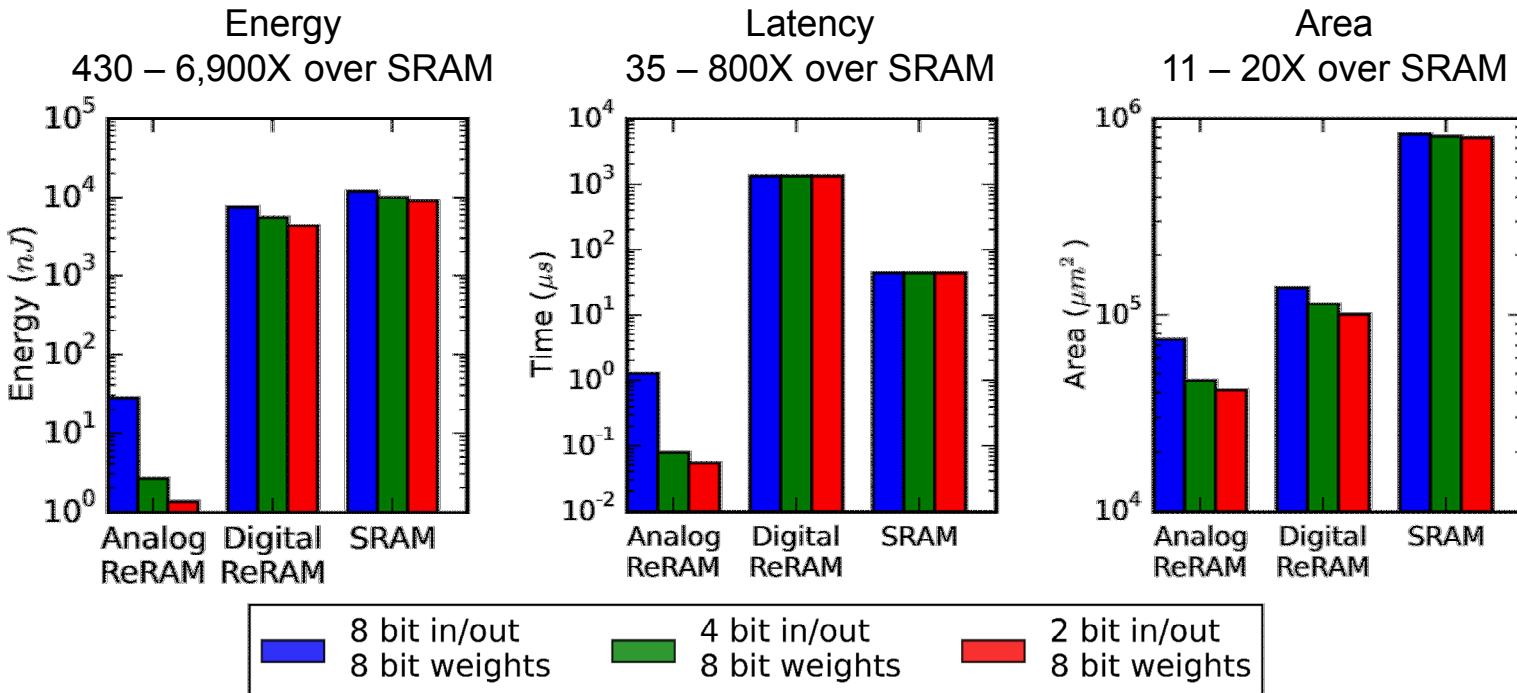
Li-Ion Synaptic Transistor for Analog Computation (LISTA)



Off by $\sim 1\%$
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



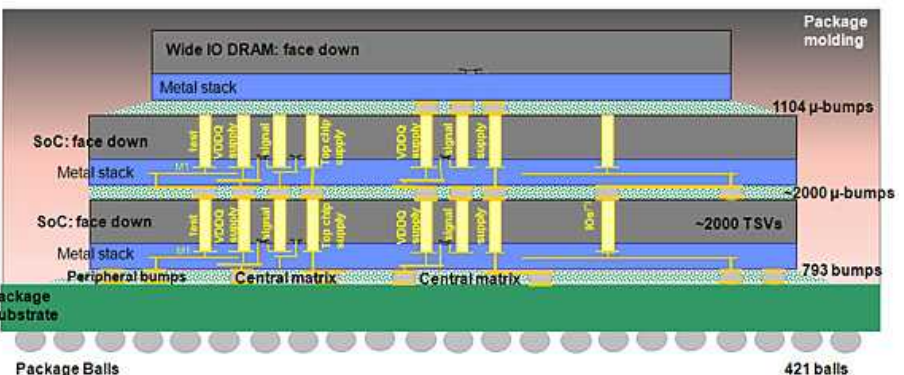
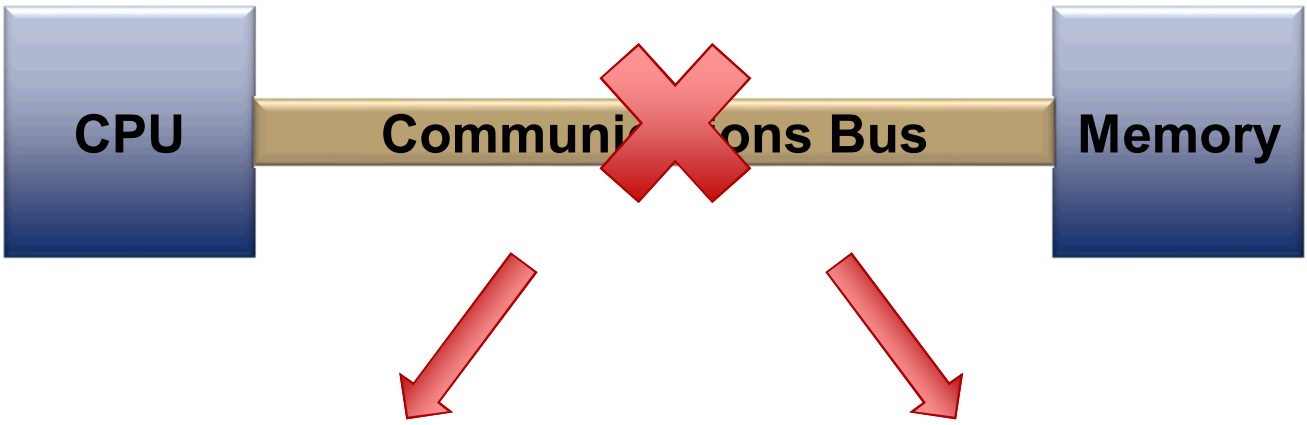
- Fundamental $O(N)$ energy scaling advantage
- Use CrossSim to co-design materials to algorithms
 - Use periodic carry to overcome noise devices
- Need high resistance 10-100 M Ω Devices
- Need low write nonlinearities

CROSS SIM

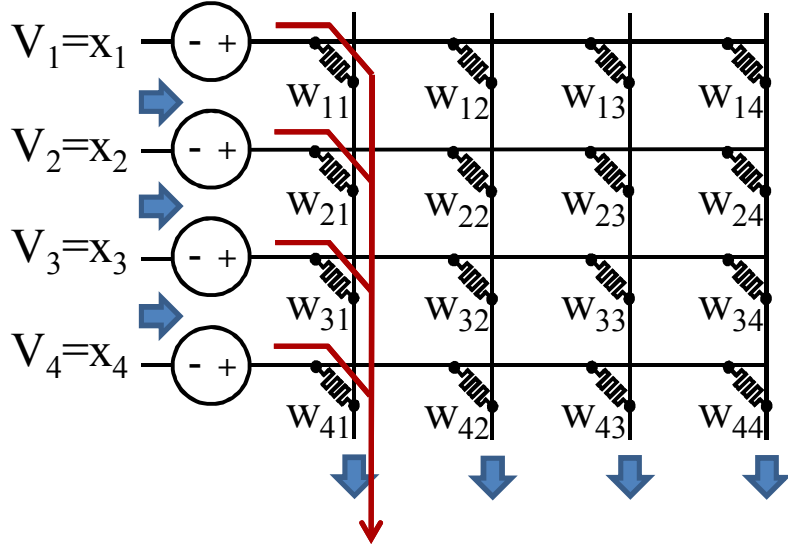
<https://cross-sim.sandia.gov>

Extra Slides

Overcoming the Power Limit

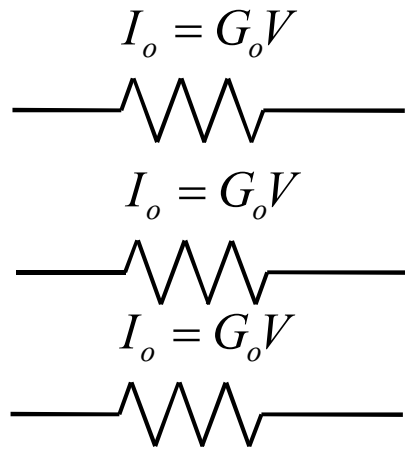


Richard Goering, "Three Die Stack -- A Big Step "Up" for 3D-ICs with TSVs" Cadence blog



Integrate Processing and Memory

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



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

What is the minimum energy?

$$Energy = \underbrace{V^2 G_o \times N}_{\text{Power in each resistor} \times \text{number of resistors}} \times \frac{1}{\Delta f} \leftarrow \text{Determined by noise and SNR}$$

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

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

$$\text{Thermal Noise} = \langle \Delta I^2 \rangle$$

$$= N \times (4k_b T \times G_o \times \Delta f)$$

$$SNR^2 = \frac{(NI_o)^2}{\langle \Delta I^2 \rangle}$$

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

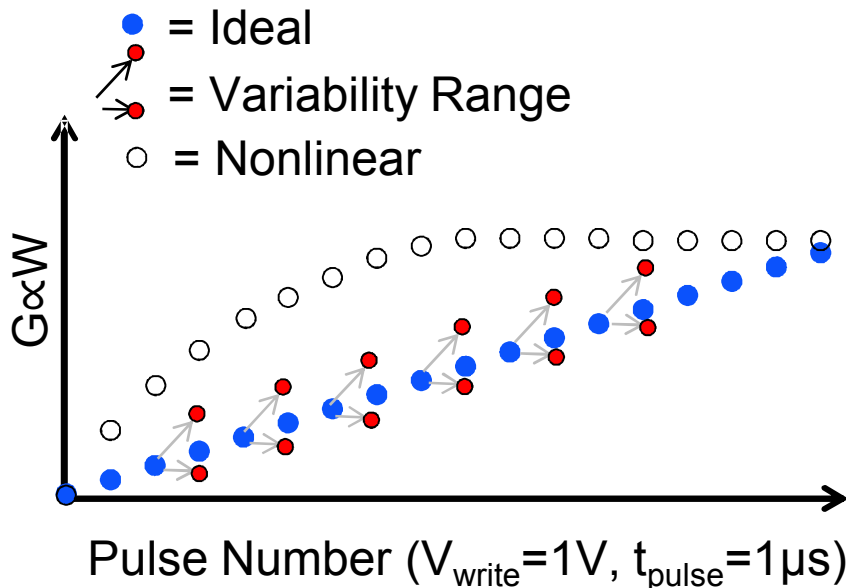
*we are assuming we need some fixed precision on the output, and don't need full floating point accuracy

Experimental Device Non-idealities

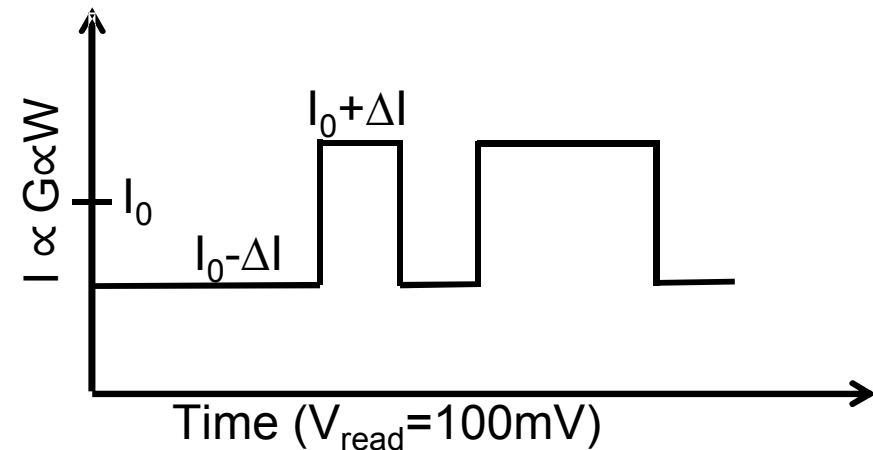
Device: **Write Variability, Write Nonlinearity, Asymmetry, Read Noise**

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

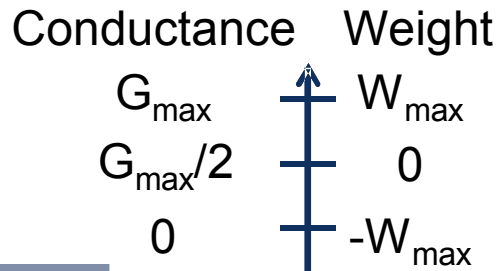
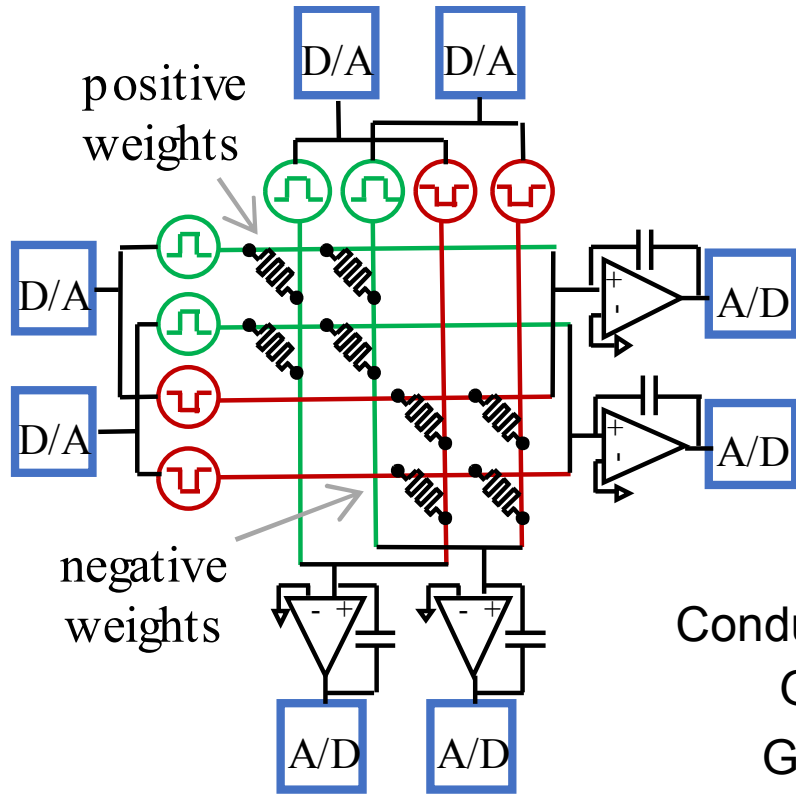
Variability and Nonlinearity



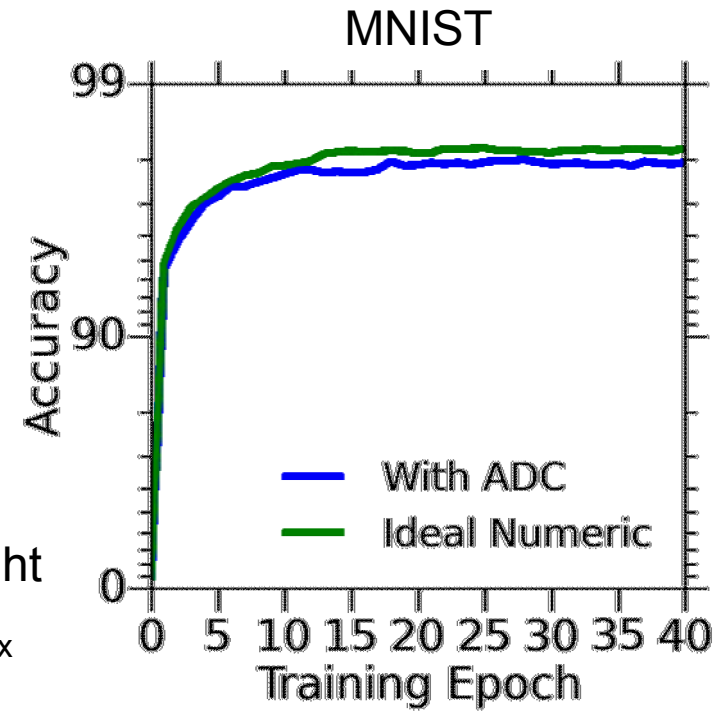
Read Noise



Full System Simulation



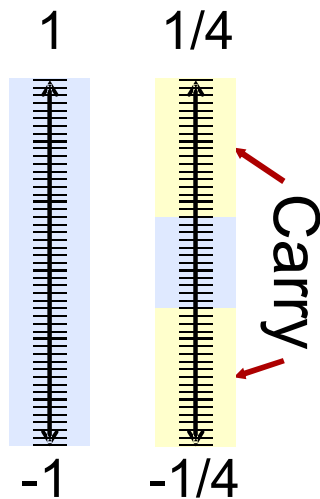
A/D & D/A Have Minimal Impact



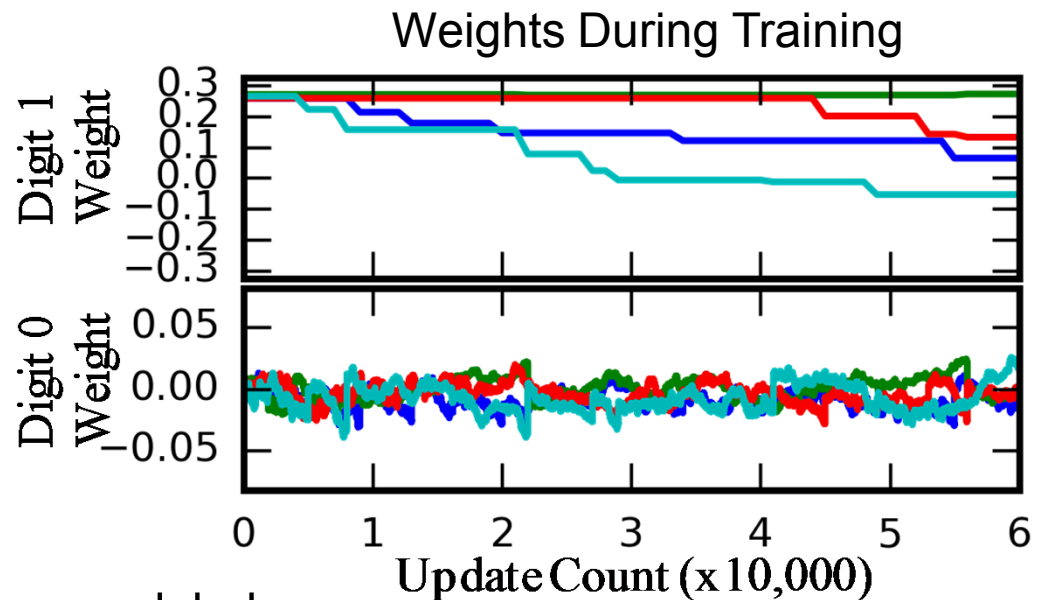
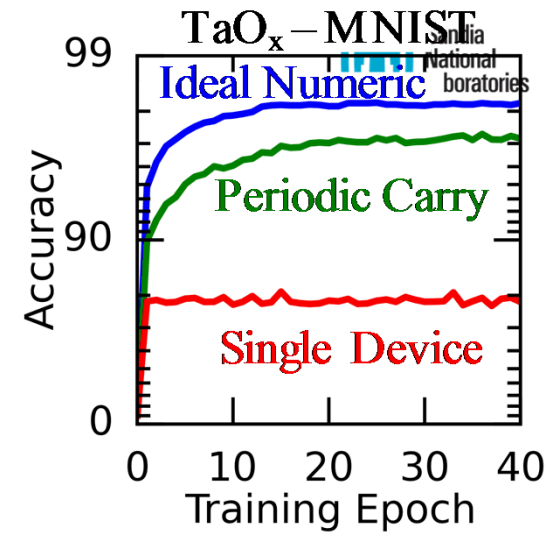
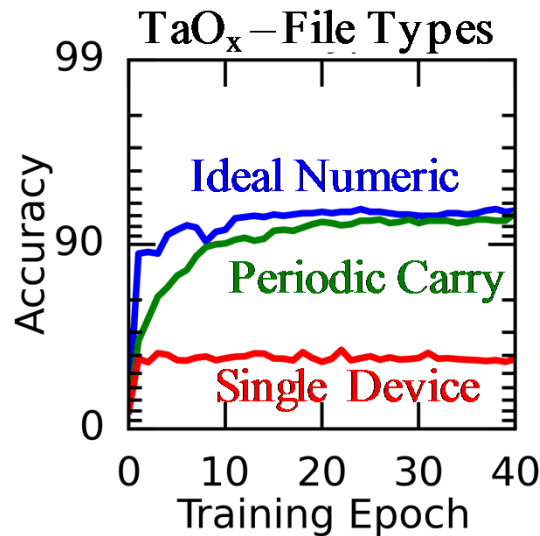
	Range	Bits
Row Input	-1 to 1	8
Col Output	-6 to 6	8
Col Input	-1 to 1	8
Row Output	-4 to 4	8
Row Update	-0.01 to 0.01	7
Col Update	-1 to 1	5

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



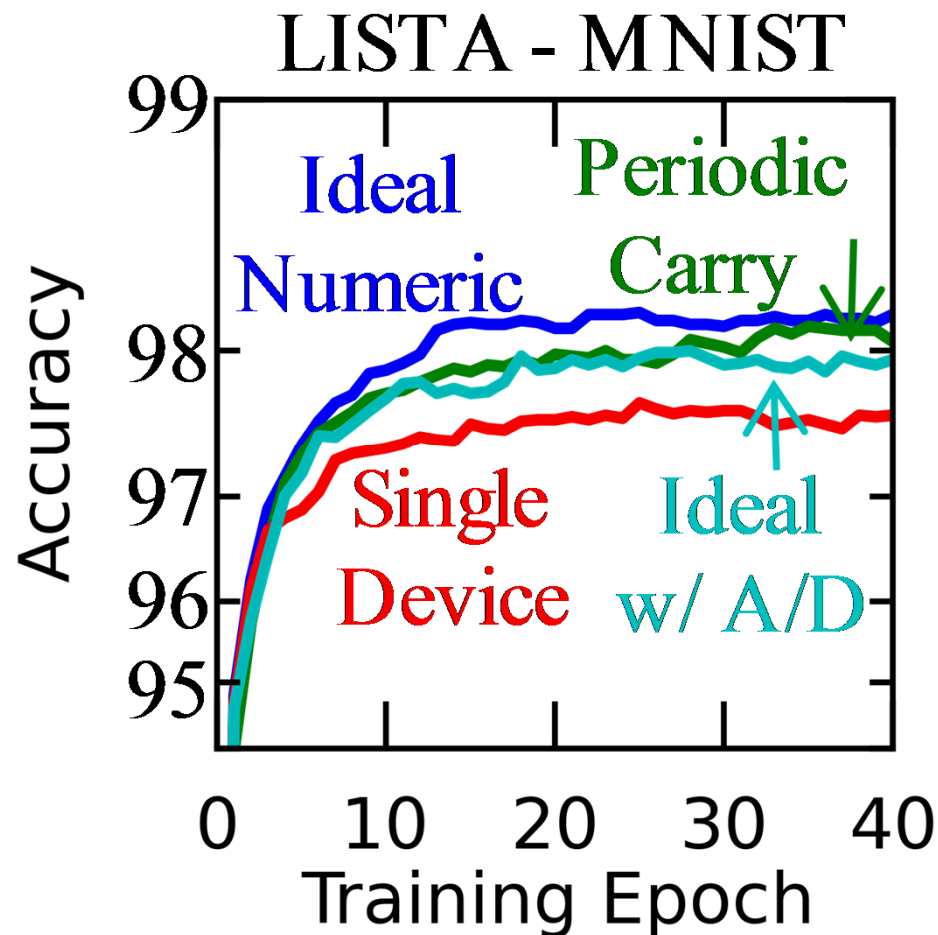
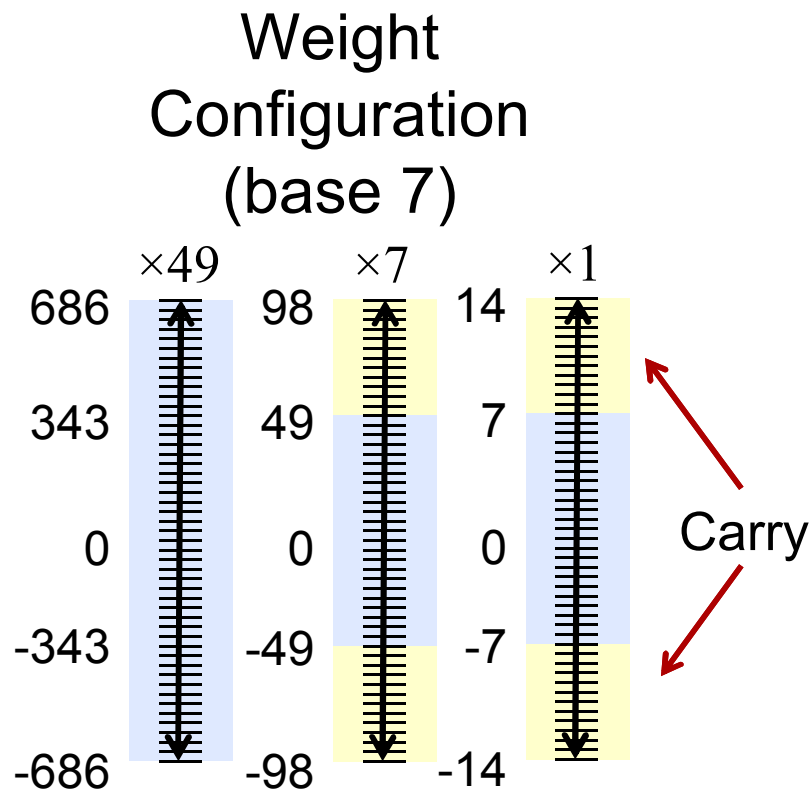
Carry once every 1000 updates for the LSB, and every 2 updates on others



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)

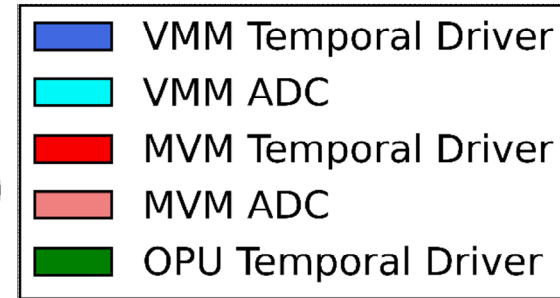
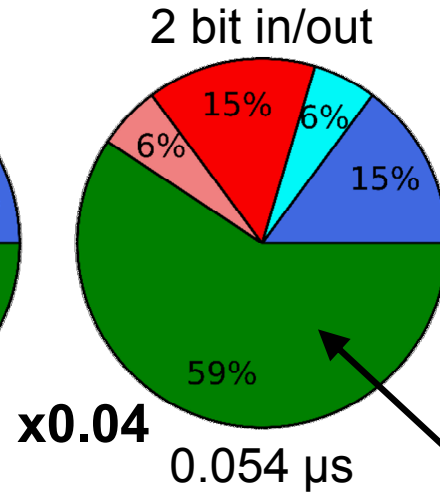
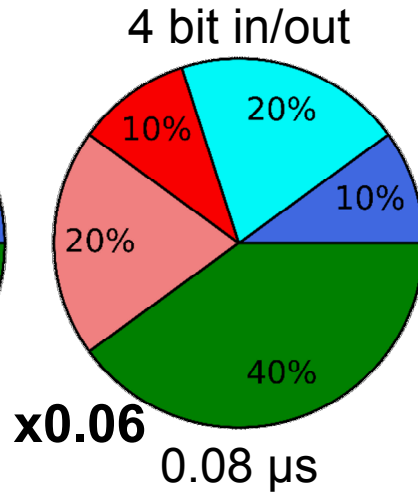
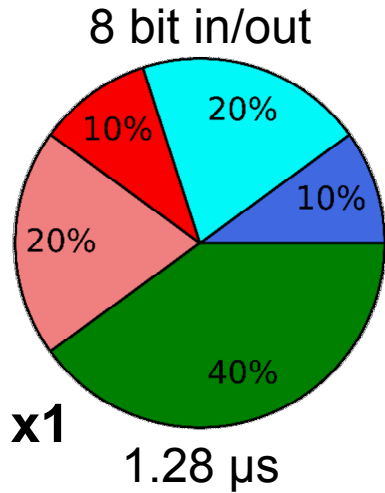
LISTA Results



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

Neural Core Latency Analysis

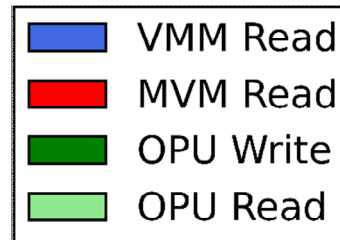
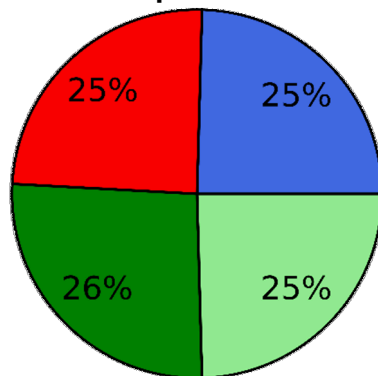
Analog ReRAM



OPU = Outer Product Update
Min write time of 8 ns vs 1 ns incremental write

Digital ReRAM

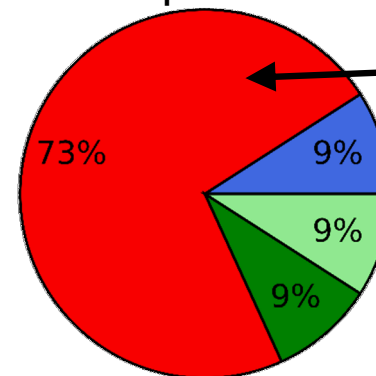
All bit precisions



x1040
1335 μ s

SRAM

All bit precisions



SRAM transpose read expensive

x35
44 μ s

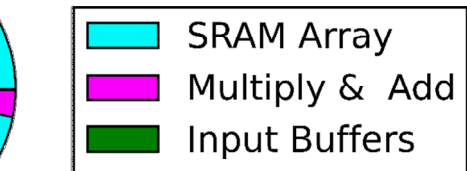
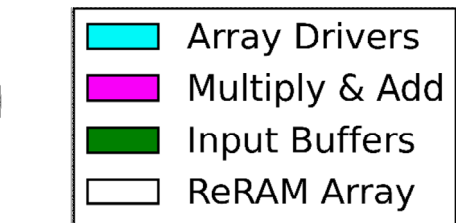
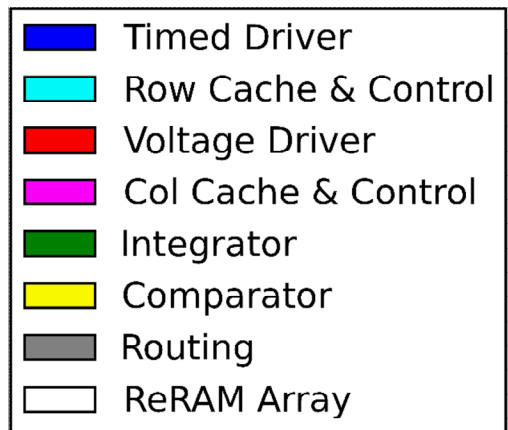
Neural Core Area Analysis



**8 bits In/out
8 bit weights**

**4 bits In/out
8 bit weights**

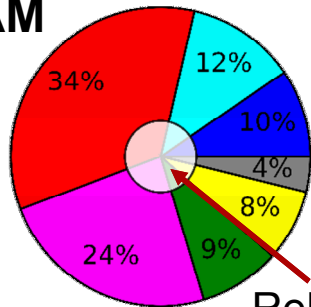
**2 bits In/out
8 bit weights**



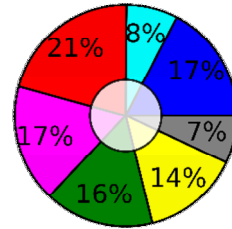
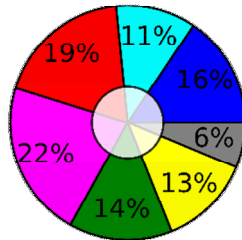
Analog ReRAM

x1

75k μm^2



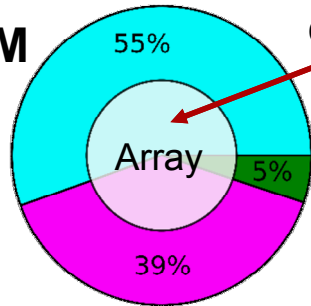
ReRAM Array



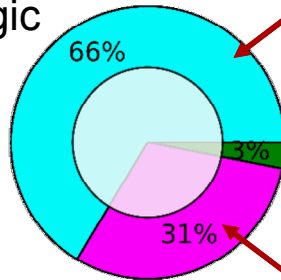
Digital ReRAM

x1.8

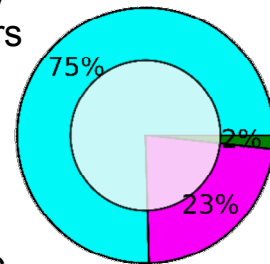
137k μm^2



on logic



Array Drivers

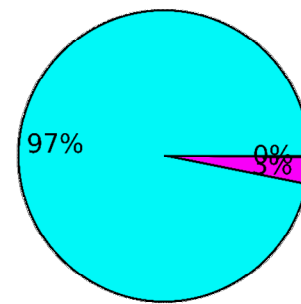
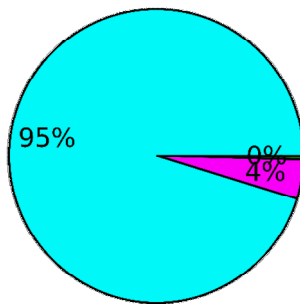
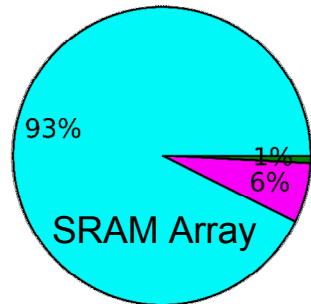


MAC

SRAM

x11.1

836k μm^2



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