Removing the Golden Handcuffs: Computing at the End of Moore’s Scaling

R. Stanley Williams
HPE Senior Fellow
What’s Next?

The end of Moore’s law could be the best thing that has happened in computing since the beginning of Moore’s law.
Motivation

- non von Neumann (and > Turing?) computing
- brain inspiration: connectome, Hodgkin-Huxley, STDP - Heb
- nonlinear dynamics, nanoscale physics, materials that compute
Future exponential increases in computer performance and efficiency will require multiple advances:

- Memory driven computing – no von Neumann bottleneck
- Dot-product engine: memristor-based vector-matrix multiplication accelerator for neural nets and signal processing
- Mimicking Synapse and Neuron Dynamics with Memristors
- Chaos as a computing resource for constrained optimization problem solving (Hopfield network)
The Chua Lectures: A 12-Part Series with HPE Labs
From Memristors and Cellular Nonlinear Networks to the Edge of Chaos
https://www.youtube.com/playlist?list=PLtS6YX0Y0X4eAQ6IrOZSta3xjRXzpcXy
or enter “The Chua Lectures” into your favorite browser

‘Linearize then analyze’ is not valid for understanding nanodevices or neurons – a nonlinear dynamical theory of electronic circuits is needed, and was developed 50 years ago by Leon Chua.

Connect the math to materials and nanostructures.
In-memory computation:

Analog linear algebra using Ohm’s and Kirchoff’s Laws

Boolean logic with conditional memristor operations
Dot Product Engine: memristor arrays accelerate vector-matrix multiplication

- Parallel multiply & add through Kirchoff’s and Ohm’s laws
- Memristors as highly scalable, tunable analog resistors
  High ON/OFF ratio (~10^5), supporting multiple levels
- Well suited for streaming workloads like neural nets
- Many ways to scale up
  Memristor levels, array size, wire pitch, 3D layer, DAC/ADC speed & width etc.
- Performance (execution time) improvements >1000x and energy efficiency >100x over GPUs for particular applications
- Commercial products in development

\[ I_j^0 = \sum_i G_{ij} \cdot V_i^1 \]

Input Voltage vector

Output current

John Paul Strachan - support from IARPA
Programming memristor arrays for computation

Analog Memory

64x128 array = 8192 memristors
Each “pixel” is a continuously tunable memristor cell
Grayscale - 182 distinct conductance levels (~7-8 bits)
Each memristor can be reprogrammed >1e6 times

Closed-loop programming with 100ns pulses
64 levels (6 bits)

E.J. Merced-Grafals, N. Dávila, N. Ge, R.S. Williams, J.P. Strachan
Nanotechnology 27, 365202 (2016).
Real-time signal processing on a DPE system (UMass collaboration) – instantaneous cosine transform

Time-domain inputs (applied to rows)

Freq-domain (DCT) outputs (read from columns)

Real-time experimental data from DPE

Software cosine transform

C. Li, et al., Nat. Elect. in press
UMass Amherst
Image Compression with cosine transform on the DPE

Compress at the edge and transmit only high-value bits

C. Li, et al., *Nat. Elect.* in press
UMass Amherst
Convolutional Filters with the DPE

Eight different filters applied in parallel

With larger cross bar, can do entire set of operations in one clock cycle/step

C. Li, et al., *Nat. Elect.* in press

Umass Amherst
Single Layer CNN for MNIST classification – standard comparison

(a) VMM Output Current (mA) for digits 0-9

(b) Recognition accuracy for different image numbers

90% accuracy by ignoring defects

Single-layer is highly sensitive to even a few defects

Dealing with defects (stuck memristors) - retraining

Retrain the network - surrounding weights can compensate for bad devices

![Normalized accuracy vs Defect Percentage](image)

<table>
<thead>
<tr>
<th>Defect Percentage (Stuck ON)</th>
<th>Worst Case, No Retraining</th>
<th>Worst Case, w/ Retraining</th>
<th>Best Case, No Retraining</th>
<th>Best Case, w/ Retraining</th>
</tr>
</thead>
<tbody>
<tr>
<td>5%</td>
<td>0.993</td>
<td>0.994</td>
<td>0.9872</td>
<td>0.9882</td>
</tr>
<tr>
<td>10%</td>
<td>0.9872</td>
<td>0.9882</td>
<td>0.9852</td>
<td>0.9852</td>
</tr>
<tr>
<td>15%</td>
<td>0.9852</td>
<td>0.9852</td>
<td>0.9772</td>
<td>0.9812</td>
</tr>
<tr>
<td>20%</td>
<td>0.9772</td>
<td>0.9812</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

C Liu, M Hu, JP Strachan, H Li, *DAC 2017*
“ISAAC” architecture: arbitrary precision and size DPEs

Store and re-use Kernels in non-volatile memristors – reduce data fetching

Heavy pipelining

Speedup of >5,000x over GPUs and 800x less energy

Speedup of 15x over Digital ASIC and 5.5x less energy

128x128 memristor Xbar arrays, 2 bits per cell, operating at 10 MHz

Modified Hopfield network for optimization problems

Chaos as a computing resource

Turing believed that intelligence required randomness and would necessarily lead to mistakes

Chua has shown that neurons are ‘poised on the edge of chaos’, and that chaos leads to complexity and emergent behavior

Epilepsy occurs when the neurons in the brain synchronize, i.e. lack of chaos

May allow construction of a highly scalable “annealing” machine
Benchmarks to compare to other Classical and Quantum Annealing machines
**NbO$_2$ Mott memristor: chaotic oscillator with DC bias**

Nanoscale oscillator

- In situ and in operando scanning transmission x-ray microscopy (STXM) at ALS critical for analysis of memristor operation

Dynamical Behavior:

- Oscillations - 1.02 V
- Chaos 1.03 V

Lyapunov exponent

$$\lambda = \lim_{t \to \infty} \lim_{\delta Z_0 \to 0} \frac{1}{t} \ln \frac{|\delta Z(t)|}{|\delta Z_0|}$$

Frequency analysis

*S Kumar, et al. Nature 2017*
Modified Hopfield network for optimization problems

Encode any TSP instance in the DPE weight matrix

Defines an “energy” of the system to be minimized

\[ E = -\frac{1}{2} \sum_{i} \sum_{j} s_{i,j} \sum_{k} \sum_{l} s_{k,l} w_{(i,j),(k,l)} + \sum_{i} \sum_{j} s_{i,j} \theta \]

Follow update rule: \( s_{i,j} = \begin{cases} 1 & \text{if } Ws'_{i,j} > \theta \\ -1 & \text{if } Ws'_{i,j} < \theta \end{cases} \)

\( \Gamma = 0 \) (no chaos) \hspace{1cm} \( \Gamma \neq 0 \) (chaos)

S Kumar, et al. Nature 2017

Traveling Salesman problem

NP hard: \( O(n^32^n) \)
Modified Hopfield network for optimization problems

Modified Hopfield network for optimization problems

Traveling Salesman problem

$O(n2^n)$?

Synapses
Memristor DPE

$s'_{i,j}$

Neurons
Chaotic NbO$_2$ nanodevices

$\Gamma=0$ (no chaos)

Example solutions w/ and w/o chaos

$\Gamma=2.3$

Optimal solution

S Kumar, et al. Nature 2017
Stateful Boolean logic

Conditional SET – using sneak paths to compute

Configured two and three memristor AND, OR, NOT, IMP and NIMP
Stateful Implication Logic

Conditional switching
There is plenty of room for creativity!

Computation inside memristor crossbars:

- Analog linear algebra – filters, signal and image processing, CNN
- Boolean logic
  - Two and three memristor conditional switching
Acknowledgments

John Paul Strachan
Miao Hu
Suhas Kumar
Catherine Graves
Emmanuelle Merced
Dejan Milojicic
Ali Shafiee
Naveen Muralimanohar
Gary Gibson
Max Zhang

J. Joshua Yang
Qiangfei Xia
Can Li
Yunning Li
Saumil Joshi
Zhongrui Wang

Izzat El Hajj (Univ Illinois Urbana-Champaign)
Pedro Bruel (IME, Universidade de São Paulo)
Sergey Saveliev (Loughborough U, UK)

Funding assistance