Using magnetic tunnel junctions to compute like the brain

Mark Stiles with help from a cast of dozens

Why?

To understand how the brain works

To develop machine consciousness

To do cognitive computing more efficiently
Hardware for Artificial Intelligence – NIST Gaithersburg
Computers are designed to solve numerical problems, the brain excels at categorical problems.

<table>
<thead>
<tr>
<th>Digital Computer</th>
<th>Neuromorphic System</th>
</tr>
</thead>
<tbody>
<tr>
<td>High precision numerical</td>
<td>Categorical</td>
</tr>
</tbody>
</table>
Computers separate long term memory from processing, while in the brain, they are intimately connected.

Small scale structure of the brain:
- Neurons ($10^{11}$)
- Synapses ($10^{15}$)

von Neumann architecture:
- Central Processing Unit
- Control Unit
- Logic Unit
- Memory Unit
- Input
- Output

<table>
<thead>
<tr>
<th>Digital Computer</th>
<th>Neuromorphic System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory-Processor Bottleneck</td>
<td>Collocation of processor and memory</td>
</tr>
<tr>
<td>High precision numerical</td>
<td>Categorical</td>
</tr>
</tbody>
</table>
Computers use a rigid encoding scheme, the brain prioritizes resiliency with more flexible schemes.

**Binary Representation**

<table>
<thead>
<tr>
<th>64</th>
<th>32</th>
<th>16</th>
<th>8</th>
<th>4</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

**Clocked processing**


**Neuronal spiking**

Rossant et al., Frontiers in Neuroscience 5, 9 (2011)

<table>
<thead>
<tr>
<th>Digital Computer</th>
<th>Neuromorphic System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discrete, Deterministic, &amp; Synchronous</td>
<td>Analog, Stochastic, &amp; Asynchronous</td>
</tr>
<tr>
<td>Memory-Processor Bottleneck</td>
<td>Collocation of processor and memory</td>
</tr>
<tr>
<td>High precision numerical</td>
<td>Categorical</td>
</tr>
</tbody>
</table>
Neural networks provide the simplest implementation of brain-like cognitive computing (rate coding)

- Neurons output a rate representing a spike train.
- The rate from each neuron is multiplied by synaptic weights for each neuron pair.
- Neurons sum the incoming weights and output a non-linear function of the sum (activation function).
- Network non-linearly transforms space to bring similar inputs together
A whole range of approaches to more efficient computing

- Software
  - Machine Learning
    - Specialized Hardware, GPU, TPU
      - Specialty Chips, TrueNorth, Loihi, BrainScales
- Architectures
- Encoding
- Circuits
- Physical devices
CMOS has developed well-defined abstraction layers, but with novel devices, designing across the stack is necessary.

- Software
- Architectures
- Encoding
- Circuits
- Physical devices

Details depend on what the device can do.

What is important depends on what the architecture needs.
Magnetic tunnel junctions are multifunctional devices

- Non-volatile embedded memory
- Spin-torque nano-oscillators
- Superparamagnetic tunnel junctions

\[ R_{AP}, R_P \]

**Intel**: MRAM integrated into 22nm FinFET CMOS

Neuronal spiking

*Image: Wikipedia*

Rossant et al., Frontiers in Neuroscience 5, 9 (2011)
Bigger energy barriers mean longer data retention, but more energy needed to write.

- Energy barrier
  - 60 $kT$
  - 14 $kT$
  - 7 $kT$

- Retention Time
  - years
  - ms
  - $\mu$s

- Memory for storage (MRAM)
- Short term memory (embedded/cache)
- Superparamagnetic (random signals)

- low-resistance
- high-resistance

- Energy

- R (Ohms)
  - 200
  - 175
  - 150

- Time (ms)
  - 0
  - 100
  - 200
  - 300
  - 400
  - 500

- “1”
- “0”
Magnetic tunnel junctions can be controlled by current.

Electron flow through tunnel junction → Spin-transfer torque

Electron flow through heavy metal underlayer → Spin-orbit torque

Positive current

*low-resistance*

Energy

Parallel

Antiparallel

*high-resistance*

Negative current

*low-resistance*

Energy

Parallel

Antiparallel

*high-resistance*
Superparamagnetic magnetic tunnel junctions: Control rate with spin-transfer torque (or spin-orbit torque)

Mean Resistance (Probability)

\[ R_{AP} \]

\[ R_P \]

Rate (Hz)

Current (µA)

Resistance (Ω)

Mean Resistance (Probability)
Superparamagnetic magnetic tunnel junctions: Control rate with spin-transfer torque (or spin-orbit torque)

$I < 0$
Stabilizes parallel state

$I > 0$
Stabilizes antiparallel state

$I_{MTJ}$

$R_{AP}$
Mean Resistance (Probability)

Rate (Hz)

Mean Resistance (Probability)

Resistance ($\Omega$)

Time (s)

Current ($\mu$A)

$R_A P$

$I > 0$
Stabilizes antiparallel state

$I < 0$
Stabilizes parallel state
Computing with superparamagnetic tunnel junctions

- Random number generation
  
  \[ p = 0.5 \]

- Probabilistic computing
  
- Rate coding

 Won Ho Choi et al., 2014 IEEE IEDM pp. 12.5.1-12.5.4,


 Borders et al., Nature 573, 390-393 (2019)
Many reasons to incorporate Complementary Metal Oxide Semiconductor (CMOS)

- **pmos**
  - On for $V_g = 0$
  - Off for $V_g = V_{dd}$

- **nmos**
  - Off for $V_g = 0$
  - On for $V_g = V_{dd}$

Author: Cephidan, https://commons.wikimedia.org/wiki/File:LDD-MOS_transistor_-_CMOS_with_STI.svg
Many reasons to incorporate Complementary Metal Oxide Semiconductor (CMOS)

- **V\text{g}**
  - pmos: On for $V_g = 0$
  - Off for $V_g = V_{dd}$
  - nmos: Off for $V_g = 0$
  - On for $V_g = V_{dd}$

**Inverter**

- Output voltage from voltage pins, not input
- One transistor always off except during switching, → 30 aJ and up

What can other devices do better?
- Non-volatility (local memory)
- Plasticity (local learning)
- Stochasticity
- Oscillators

In 2014, 250 $\times$ 10$^{18}$ transistors manufactured → cheap substrate
Energies vary over many orders of magnitude

- **1 nJ**
- **1 GeV**
- **1 pJ**
- **1 MeV**
- **1 fJ**
- **1 keV**
- **1 aJ**
- **1 eV**
- **1 zJ**
- **1 meV**

- **Neural spike**
- **MRAM switching**
- **Synaptic event**
- **Gate transition**
- **MRAM barrier**
- **Ion channel**
- **Room temperature**
- **Liquid Helium temperature**

Room temperature

Gate transition

MRAM barrier

Ion channel

MRAM switching

Synaptic event

Neural spike

Liquid Helium temperature
Population coding with superparamagnetic tunnel junctions

- Rate/Population Coding
- Transition rate, binary
- Pre-charge sense amplifier
- Superparamagnetic tunnel junctions, Short term memory

A. Mizrahi, T. Hirtzlin, A. Fukushima, H. Kubota, S. Yuasa, J. Grollier, and D. Querlioz, Nature Comm. 9, 1533 (2018);

Bigger energy barriers mean longer data retention, but more energy needed to write.

- **Energy barrier**
  - 60 kT
  - 14 kT
  - 7 kT

- **Retention Time**
  - years
  - ms
  - μs

- **Memory for storage (MRAM)**
- **Short term memory** (synaptic weights)
- **Superparamagnetic** (spiking neurons)

- **Energy**
  - low-resistance
  - high-resistance

- **Graph**
  - R (Ohms)
  - Time (ms)
  - AP
  - P
Population coding – represent continuous degrees of freedom with multiple spike rates

\[ \sum w_i r_i \rightarrow H \]

Neurons

Directions

\[ \theta \]

Weights

\[ +180^\circ \]

\[ -180^\circ \]

Firing rate

Tuning curves

\[ \text{Direction} \]

\[ \text{Neuron} \]

\[ \text{Firing rate} \]
Simple sense, respond, measure, and learn test

Sensory input

Motor response

T. Hirtzlin at C2N:

- SMTJ-based vs. CMOS only
  - Area: 12,000 μm² vs. 20,000 μm²
  - Energy: 7.8 nJ vs. 20 nJ

Key advantage = stochastic analog to digital conversion

Weights stored in stable MTJs. Can we save energy there?
Continuous learning: the key to a robust system that can adapt to changes

Usually system is only trained once because it requires a lot of energy
BUT unable to adapt to changes!

System equipped with continuous learning
Continuous learning overcomes unreliable synapses

Here, $\Delta E = 15 \, k_BT$ magnetic tunnel junctions can be used for memory with no precision loss.
Using unreliable synapses lowers the energy consumption

Write current
\[ I \propto \Delta E_w \]
(Sato et al., APL 2014)

Write energy
\[ Energy \propto \Delta E_w^2 \]
Using unreliable synapses lowers the energy consumption

Write current

\[ I \propto \Delta E_w \]

(Sato et al., APL 2014)

Write energy

\[ \text{Energy} \propto \Delta E_w^2 \]
Using unreliable synapses lowers the energy consumption

Write current
\[ I \propto \Delta E_w \]
(Sato et al., APL 2014)

Write energy
\[ Energy \propto \Delta E_w^2 \]

Example:
For 3% precision
→ 54 neurons in each population (2916 weights)
→ \( \Delta E_w = 12 k_B T \)
Stochastic computing with superparamagnetic tunnel junctions

- Neural network
- Stochastic bit streams
- AND and OR gates, PCSA
- Superparamagnetic tunnel junctions

Stochastic computing represents numbers as probabilistic bit streams

- Resilient to white noise
- Incorrect bit $\rightarrow$ error $\approx O(1)$
- Compare to binary: error $\approx O(2^N)$
- Naturally coherent with bilevel devices (magnetic tunnel junctions).
- Interpreting these “spike trains” as rates or probabilities restricts us to $0 \leq p \leq 1$. 

$\rho = 0.2$

\[ \text{Time $\rightarrow$} \]

\[ \begin{array}{c}
\text{observed} = 0.19 \\
0.35 \\
0.63 \\
0.69
\end{array} \]

$\rho = 0.4$

$\rho = 0.5$

$\rho = 0.7$
Unbiased superparamagnetic tunnel junction plus CMOS for low energy random bitstreams

Superparamagnetic tunnel junction

$I_{MTJ} \ll I_C$

$R_{AP} > R_{ref} > R_P$

Low energy random bits

$p = 0.5$
AND gates for multiplication
– Programmable bitstream generator: 10 fJ per bit

Example

\[\begin{align*}
&1001010100 \ (0.4) \quad 1010001110 \ (0.5) \\
\times &1010001110 \ (0.5) \\
= &1000000100 \ (0.2)
\end{align*}\]

\[\begin{align*}
p_{\text{AND}} &= p_a p_b
\end{align*}\]

Typical stochastic computing uses Linear Feedback Shift Registers
→ Short period pseudorandom numbers
Synapses from bitstream generators (weights) and AND gates (multiplication)

Bit stream from neuron

AND-gate synapse
Synaptic weight:

Programmable Bit stream generator
Neurons from OR gates for nonlinear summation

Example

\[\begin{align*}
1001010100 & \quad (0.4) \\
+ 1010001110 & \quad (0.5) \\
= 1011011110 & \quad (0.7)
\end{align*}\]

\[p_{\text{OR}} = p_a + p_b - p_a p_b\]

OR-gate neuron does not work with correlated bit streams
Low area, high efficiency neural network with stochastic information throughout the calculation.

- AND-gate synapse
  - Synaptic weight: Bit stream from programmable SMTJ generator

- OR-gate neuron

Bit stream from neuron
Efficient random bitstream generation and avoiding domain translation – energy efficiency on a standard problem

- Energy/performance tradeoff by tuning circuit parameters.
- Running for longer or shorter total time ($\sim N$) gives energy/performance tradeoff.

Future: implement primitives to check feasibility with realistic MTJs

A standard deep neural network structure

LeNet5 MNIST Inference Error (%)

Energy per inference (nJ)

Stochastic computing implementations

- SC-DCNN6
- ReLU
- Tanh
- SC-DCNN11
- Logistic
- HEIF

N = 1024
N = 32
N = 64
256
128
64

Standard test dataset

- PML · NDCD · Alternative Computing Group
Collaborators

Brian Hoskins, **Matthew Daniels**, Guru Khalsa, Mark Anders, Jonathan Goodwill, Jabez McClelland, Nikolai Zhitenev, William Rippard, Matthew Pufall, Emilie Jué, **Mark Stiles**

**Alice Mizrahi, Advait Madhavan, Philippe Talatchian**

Siyuan Huang, Gina Adam

George Tzimpragos, Tim Sherwood

Jacob Torrejon, Mathieu Riou, Flavio Abreu Araujo, Paolo Bortolotti, Vincent Cros, Julie Grollier

Tifenn Hirtzlin, Damien Querlioz

Sumito Tsunegi, Kay Yakushiji, Akio Fukushima, Hitoshi Kubota, Shinji Yuasa

Energy Frontier Research Center
Department of Energy

Vivek Amin, Jonathan Gibbons, Paul Haney, Axel Hoffmann, Julie Grollier
Using magnetic tunnel junctions to compute like the brain

- Emulating features of the brain can increase efficiency for cognitive computing.
- Applications require designing across the computational stack
  - Architecture
  - Encoding
  - Circuitry
  - Devices
- CMOS is likely to play an important role in any room-temperature computing scheme.
- Magnetic tunnel junctions
  - Multifunctional
  - Already in CMOS Fabs

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Encoding</th>
<th>Devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population coding</td>
<td>Fluctuation rates</td>
<td>Superparamagnetic tunnel junctions</td>
</tr>
<tr>
<td>Synaptic weights</td>
<td>Binary</td>
<td>Unstable magnetic tunnel junctions</td>
</tr>
<tr>
<td>Neural network with stochastic computing</td>
<td>Random bitstreams</td>
<td>Superparamagnetic tunnel junctions, Repurposed CMOS Logic gates</td>
</tr>
</tbody>
</table>