Memory Integrated Neural Network Accelerators

February 26, 2013

Tarek Taha

Electrical and Computer Engineering Department, University of Dayton
Objective

Design multicore neural network processors

Examine:
- Routing
- Memory options
- Analog and Digital options
Related Work: SpiNNaker

- Each chip contains 18 ARM cores and a network router
- Geared towards brain scale simulation
- Cores share 128MB SDRAM stacked within the package
- Multiple chips connected in a 2D torroidal network
Related Work: FACETS

- Mixed signal ASIC wafer
- Geared towards brain scale simulation
- 200k neurons and 50 million synapses
- Synaptic weight is represented in a 4-bit SRAM with a 4-bit DAC
Related Work: IBM design

- Crossbar memory based digital core
- Each core has capacity to simulate 256 integrate and fire neuron
- 1024 synapse per neuron
- One bit per synapse
Implementing Synapses

- Memory elements to implement synapse:

<table>
<thead>
<tr>
<th></th>
<th>DRAM</th>
<th>SRAM</th>
<th>Memristor crossbar</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Off-chip</td>
<td>On-chip</td>
<td>On-chip</td>
</tr>
<tr>
<td>Density (Gb/cm²)</td>
<td>6.67</td>
<td>0.338</td>
<td>12</td>
</tr>
<tr>
<td>Read time</td>
<td>~100</td>
<td>0.3</td>
<td>0.3</td>
</tr>
</tbody>
</table>

- Memristors can also mimic neural computing circuits
Digital Core Design

Towards Memristor based Neuromorphic Architectures

Tarek Taha
Digital Core Configurations

- Configurations:
  - 2 bit per synapse, 1 bit per neuron
    - Multipliers not needed.
  - 4 bits per synapse
    - Multiplier and adder needed

- Analysis
  - SRAM array power and area calculated from CACTI.
  - Only SRAM array, predecoder, and sense amplifiers considered.
  - Static and dynamic powers both considered
Routing Analysis

- Model developed to determine routing network bandwidth requirements
- Router and link powers calculated from Orion

![Routing Analysis Diagram]

- Model developed to determine routing network bandwidth requirements
- Router and link powers calculated from Orion
A generalized memristor SPICE model has been developed
- Capable of modeling several different published memristor devices for a variety of different voltage inputs
- The result of our model has been compared to the published characterization data using a quantifiable error percentage

Variation tolerance studies have been performed at the device and circuit level
- Layer thickness variation
- Variation in state variable dynamics has also been studied
  - Caused by inconsistent stoichiometric ratio in the metal-oxide layer
  - Leads to variability in the number oxygen vacancies in each device on a wafer
Model Equations

Current-Voltage (I-V) Relationship

\[ I(t) = \begin{cases} 
    a_1 x(t) \sinh(bV(t)), & V(t) \geq 0 \\
    a_2 x(t) \sinh(bV(t)), & V(t) < 0 
\end{cases} \]

State Variable Motion

\[ \frac{dx}{dt} = \eta g(V(t)) f(x(t)) \]

Voltage Threshold

\[ g(V(t)) = \begin{cases} 
    A_p (e^{V(t)} - e^{V_p}), & V(t) > V_p \\
    -A_n (e^{-V(t)} - e^{V_n}), & V(t) < -V_n \\
    0, & -V_n \leq V(t) \leq V_p 
\end{cases} \]

Non-Linear Drift

\[ f(x) = \begin{cases} 
    e^{-\alpha_p (x-x_p)} w_p(x,x_p), & x \geq x_p \\
    1, & x < x_p \\
    e^{\alpha_n (x+x_n-1)} w_n(x,x_n), & x \leq 1 - x_n \\
    1, & x > 1 - x_n 
\end{cases} \]

\[ w_p(x,x_p) = \frac{x_p - x}{1 - x_p} + 1 \]

\[ w_n(x,x_n) = \frac{x}{1 - x_n} \]
Modeling – Sinusoidal Inputs

Boise State Device

Pino Compact Model

Univ. of Dayton Model

HP Labs Model

Biolek Model
Modeling – Sweeping Inputs

Boise State Device

Univ. of Dayton Model*

HP Labs Model

Joglekar Model

Biolek Model


UD Model: Boise State Device

Model result

- Voltage (V)
- Current (mA)

Data from [1]

Model matches the target data with 6.64% error [1]

- Top Electrode Voltage (V)
- Top Electrode Current (mA)

Model matches the target data with 6.66% error [1]

UD Model: Univ of MI Device

Our model result

Characterization data from [1]

Model matches the target data with 6.21% error

Towards Memristor based Neuromorphic Architectures

UD Model: HP Labs Device

Model matches the target data with 11.58% error [1]

Model matches the target data with 13.6% error (8.72% when not considering the largest outlier [2])

Model matches the target data with 4.60% error

Towards Memristor based Neuromorphic Architectures


Memristor Modeling

- Our model is more robust and accurate than other models
- Results are based on a 16 memristor crossbar circuits with a 10 ns switching time

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Biolek</th>
<th>Joglekar</th>
<th>Laiho</th>
<th>Chang</th>
<th>Our Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time Before Error (µs)</td>
<td>0.517</td>
<td>0.517</td>
<td>15</td>
<td>0.135</td>
<td>15 (Finished: no error)</td>
</tr>
<tr>
<td>Accuracy</td>
<td>Low</td>
<td>Low</td>
<td>Medium</td>
<td>Medium</td>
<td>High</td>
</tr>
<tr>
<td>Generality</td>
<td>High</td>
<td>High</td>
<td>Medium</td>
<td>Medium</td>
<td>High</td>
</tr>
</tbody>
</table>

Memristor Models Included in the Study
Crossbar Write Technique

- Write method used stops state change of unselected devices
  - Although this does not solve the problem for reads
Unwanted Current Paths

- Alternate current paths in the resistor array consume power and degrade signal.

Correct read between \(a_1\) and \(b_1\)

Incorrect read between \(a_1\) and \(b_1\) due to the alternate current path
Tiled Memristor Digital Memory

- Full crossbar simulated in SPICE
- Wire resistances simulated
- Untiled memory array write energy very high
- Read noise margin low
- Designed a tiled memristor memory array

Untiled memory array

Tiled memory array

• Full crossbar simulated in SPICE
• Wire resistances simulated
• Untiled memory array write energy very high
• Read noise margin low
• Designed a tiled memristor memory array
Memristor Memory Properties

- Training energy increases with crossbar size.
- Novel segmented memristor crossbar memory.
- 4x4 crossbar: 2 bits per transistor
- 8x8 crossbar: 4 bits per transistor

<table>
<thead>
<tr>
<th>Crossbar size</th>
<th>Training energy per synapse</th>
<th>Read energy per synapse</th>
</tr>
</thead>
<tbody>
<tr>
<td>4x4</td>
<td>1.90 pJ</td>
<td>1.84 fJ</td>
</tr>
<tr>
<td>8x8</td>
<td>5.07 pJ</td>
<td>2.00 fJ</td>
</tr>
</tbody>
</table>
Classifier Simulation

- 2 layer neural classifier
  - Based on a 16 memristor crossbar
  - Iteratively trained through MATLAB and SPICE
Simulation Setup

- **Simulation platform**
  - Uses SPICE to produce accurate crossbar circuit simulation
  - Output voltages and synaptic weights are analyzed in MATLAB

**MATLAB / C**

1. Generate weights and write to a SPICE configuration file
2. Call SPICE program with the new weight configuration file
3. Read weights from file
4. Simulate crossbar with different inputs and record outputs
5. Export output voltages to file
6. Evaluate output voltages from export file
7. Revise weights based on outputs
8. Go to step 2

**SPICE**

3. Read weights from file
4. Simulate crossbar with different inputs and record outputs
5. Export output voltages to file
Image Conversion

- Image data is converted to a SPICE waveform
- Each row in the image is converted to a voltage of 16 intervals
- Entire image is represented by 4 voltage pulses
Simulation Result (1/2)

- Crossbar bar is performing as a linear separator
- Can differentiate between all 4 image classes
Crossbar bar is performing as a linear separator
Can differentiate between all 4 image classes

Class 3
Class 4

Voltage (V)
Output Neuron
Voltage (V)
Output Neuron
Parallel operation

- Read enable signal and diode allow multiple crossbars to be read in parallel
- Multipliers and adders not needed

[[Diagram of parallel operation]]
Analog Memristor Core

- Analog crossbar array replaces digital memory AND multiply-add units
- Full crossbar can be evaluated in parallel
- Most of the core can be shutdown once computations are done

**Diagram:**

- **SRAM array**
- **Memristor crossbar array**
- **Pre-synaptic Neuron Values (PSNV)**
- **Outputs**

- Decoder
- Pre-synaptic neuron number
- Pre-synaptic neuron value

- **Operations:**
  - Multiply
  - Add
  - Accumulate

- **Connections:**
  - Pre-synaptic neuron values
  - SRAM array
  - Memristor crossbar array
  - Outputs
Digital Memristor Core

- Digital crossbar array replaces SRAM memory array
- One row of synapses processed per cycle
- Most of the core can be shutdown once computations are done
Core Characteristics

Cycle Time: 5ns (200MHz clock)
Technology: 40nm

Cores for 1 bit per neuron case (multiplies not needed):

<table>
<thead>
<tr>
<th>Config.</th>
<th>Synaptic memory device</th>
<th>Memory cells per synapse</th>
<th>Bits per synapse</th>
<th>Adder</th>
<th>Core area (mm²)</th>
<th>Energy per neuron (pJ)</th>
<th>Core throughput (neurons/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Memristor</td>
<td>1</td>
<td>Analog</td>
<td>Current add</td>
<td>0.037</td>
<td>23</td>
<td>263.9×10⁶</td>
</tr>
<tr>
<td>2</td>
<td>Memristor</td>
<td>2</td>
<td>2 bits</td>
<td>12 bit adder</td>
<td>0.098</td>
<td>240</td>
<td>42.1×10⁶</td>
</tr>
<tr>
<td>3</td>
<td>SRAM</td>
<td>2</td>
<td>2 bits</td>
<td>12 bit adder</td>
<td>0.288</td>
<td>377</td>
<td>42.1×10⁶</td>
</tr>
</tbody>
</table>

Cores for 4 bits per neuron case:

<table>
<thead>
<tr>
<th>Config.</th>
<th>Synaptic memory device</th>
<th>Memory cells per synapse</th>
<th>Bits per synapse</th>
<th>Multiplier</th>
<th>Adder</th>
<th>Core area (mm²)</th>
<th>Energy per neuron (pJ)</th>
<th>Core throughput (neurons/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Memristor</td>
<td>1</td>
<td>Analog</td>
<td>Memr. resist.</td>
<td>Current add</td>
<td>0.058</td>
<td>26</td>
<td>66.3×10⁶</td>
</tr>
<tr>
<td>5</td>
<td>Memristor</td>
<td>4</td>
<td>4 bits</td>
<td>4 bit multiplier</td>
<td>18 bit adder</td>
<td>0.179</td>
<td>391</td>
<td>28.6×10⁶</td>
</tr>
<tr>
<td>6</td>
<td>SRAM</td>
<td>4</td>
<td>4 bits</td>
<td>4 bit multiplier</td>
<td>18 bit adder</td>
<td>0.513</td>
<td>780</td>
<td>26.6×10⁶</td>
</tr>
</tbody>
</table>

Energy considers leakage, data transfer to/from router, and control circuits.
General Purpose Systems Comparison

- Programmed commercial systems:
  - Data set was small enough to fit in cache/onboard memory
  - Simulated neural networks with 1024 synapses per neuron

- NVIDIA Tesla M2070 GPGPU
  - 225W max power
  - 448 cores
  - 575 MHz
  - CUDA program

- Intel Xeon X5650 processor
  - 95W max power
  - Six cores
  - 2.66 GHz
  - SSE instructions modeled
Comparison: 1 Bit Neurons

Example 1:
- 25,600 neurons
- 100,000 iterations/s
- 1024 synap./neuron

<table>
<thead>
<tr>
<th>Configuration</th>
<th># of chips</th>
<th>Chip area (mm²)</th>
<th>% Active</th>
<th>Power (W)</th>
<th>Power eff. over Xeon</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memristor Analog</td>
<td>1</td>
<td>3.7</td>
<td>9.7%</td>
<td>0.07</td>
<td>253,489</td>
</tr>
<tr>
<td>Memristor Digital</td>
<td>1</td>
<td>9.7</td>
<td>60.8%</td>
<td>0.62</td>
<td>27,546</td>
</tr>
<tr>
<td>SRAM</td>
<td>1</td>
<td>35.2</td>
<td>60.8%</td>
<td>1.13</td>
<td>15,099</td>
</tr>
<tr>
<td>NVIDIA M2070</td>
<td>12</td>
<td>529.0</td>
<td>99.2%</td>
<td>2700.00</td>
<td>6</td>
</tr>
<tr>
<td>Intel Xeon X5650</td>
<td>179</td>
<td>240.0</td>
<td>99.9%</td>
<td>17005.00</td>
<td>1</td>
</tr>
</tbody>
</table>

Example 2:
- 1,706,667 neurons
- 1500 iterations/s
- 1024 synap./neuron

<table>
<thead>
<tr>
<th>Configuration</th>
<th># of chips</th>
<th>Chip area (mm²)</th>
<th>% Active</th>
<th>Power (W)</th>
<th>Power eff. over Xeon</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memristor Analog</td>
<td>2</td>
<td>248</td>
<td>0.15%</td>
<td>0.70</td>
<td>24,395</td>
</tr>
<tr>
<td>Memristor Digital</td>
<td>2</td>
<td>333</td>
<td>0.91%</td>
<td>1.25</td>
<td>13,633</td>
</tr>
<tr>
<td>SRAM</td>
<td>5</td>
<td>388</td>
<td>0.91%</td>
<td>28.02</td>
<td>607</td>
</tr>
<tr>
<td>NVIDIA M2070</td>
<td>12</td>
<td>529</td>
<td>99.2%</td>
<td>2700.00</td>
<td>6</td>
</tr>
<tr>
<td>Intel Xeon X5650</td>
<td>179</td>
<td>240</td>
<td>99.90%</td>
<td>17005.00</td>
<td>1</td>
</tr>
</tbody>
</table>
### Example 1:
- 25,600 neurons
- 100,000 iterations/s
- 1024 synap./neuron

<table>
<thead>
<tr>
<th>Configuration</th>
<th># of chips</th>
<th>Chip area (mm²)</th>
<th>% active</th>
<th>Power (W)</th>
<th>Power eff. over Xeon</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memristor Analog</td>
<td>1</td>
<td>5.9</td>
<td>38.6%</td>
<td>0.07</td>
<td>234,859</td>
</tr>
<tr>
<td>Memristor Digital</td>
<td>1</td>
<td>18.2</td>
<td>89.6%</td>
<td>0.62</td>
<td>16,968</td>
</tr>
<tr>
<td>SRAM</td>
<td>1</td>
<td>29.1</td>
<td>89.6%</td>
<td>1.13</td>
<td>8,215</td>
</tr>
<tr>
<td>NVIDIA M2070</td>
<td>12</td>
<td>529.0</td>
<td>99.2%</td>
<td>2700.00</td>
<td>6</td>
</tr>
<tr>
<td>Intel Xeon X5650</td>
<td>179</td>
<td>240.0</td>
<td>99.9%</td>
<td>17005.00</td>
<td>1</td>
</tr>
</tbody>
</table>

### Example 2:
- 1,706,667 neurons
- 1500 iterations/s
- 1024 synap./neuron

<table>
<thead>
<tr>
<th>Configuration</th>
<th># of chips</th>
<th>Chip area (mm²)</th>
<th>% active</th>
<th>Power (W)</th>
<th>Power eff. over Xeon</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memristor Analog</td>
<td>1</td>
<td>395</td>
<td>0.58%</td>
<td>0.70</td>
<td>24,210</td>
</tr>
<tr>
<td>Memristor Digital</td>
<td>4</td>
<td>303</td>
<td>1.34%</td>
<td>1.63</td>
<td>10,419</td>
</tr>
<tr>
<td>SRAM</td>
<td>9</td>
<td>383</td>
<td>1.34%</td>
<td>48.67</td>
<td>349</td>
</tr>
<tr>
<td>NVIDIA M2070</td>
<td>12</td>
<td>529</td>
<td>99.2%</td>
<td>2700.00</td>
<td>6</td>
</tr>
<tr>
<td>Intel Xeon X5650</td>
<td>179</td>
<td>240</td>
<td>99.9%</td>
<td>17005.00</td>
<td>1</td>
</tr>
</tbody>
</table>
Comparison: Digital vs Analog cores

256 Neurons with 1024 synapses each.

Synapses hold 16 possible values.

**Digital SRAM**

- Neuron number
- 1024x1024 SRAM array
- Adder and multiplier needed
- 1 synapse processed /cycle (per neuron)
- 29.1 mm²
- 1.13 W

**Analog**

- Current summation
- 1024 synapses processed/cycle (per neuron)
- 5.9 mm²
- 0.07 W

- 6x area
- 16x more energy*

*Energy calculation considers leakage and data routing power
Conclusion

- Specialized cores very efficient for neural network acceleration
- Further improvements possible for both SRAM and memristor cores
- Memristor cores efficient when carrying out recognition tasks

Acknowledgement: This work was partially supported by an NSF Award