Memristors: Not Just Memory

Shahar Kvatinsky
Advisors: Avinoam Kolodny, Uri Weiser, and Eby Friedman

Technion – Israel Institute of Technology
Electrical Engineering Department
November 2013
3 Fundamental Circuit Elements

Resistor

\[ v = R \cdot i \]

Georg Ohm 1827

Capacitor

\[ q = C \cdot v \]

don Kleist 1745

Inductor

\[ \phi = L \cdot i \]

Michael Faraday 1831
3 Fundamental Circuit Elements

Resistor

\[ v = R \cdot i \]

Capacitor

\[ q = C \cdot v \]

Inductor

\[ \varphi = L \cdot i \]
The Missing Circuit Element

- **Resistor**: \( v = R \cdot i \)

- **Capacitor**: \( q = C \cdot v \)

- **Inductor**: \( \phi = L \cdot i \)

- ?
The Memristor 1971

Resistor

\[ v = R \cdot i \]

Capacitor

\[ q = C \cdot v \]

Inductor

\[ \phi = L \cdot i \]

Memristor

\[ \phi = M \cdot q \]

Memristor Behavior

Decrease resistance
2008 Memristors are Real!

- Hewlett Packard

Fabricating Memristors

Diagram showing the layout of Metal5, Metal4, Metal3, Metal2, and Metal1 layers with Memristors and Via connections. At the bottom, there are connections labeled Drain, Source, and Gate.
More Memristors?

Resistive RAM (RRAM)

STT MRAM

Phase Change Memory (PCM)

SanDisk  SONY

Samsung  SAMSUNG

Panasonic  SAMSUNG

Toshiba  QUALCOMM

SK hynix  Crossbar
General Model – TEAM
ThrEshold Adaptive Memristor

- Tunable behavior
- Current threshold
- Low computational effort

Good accuracy
Fast simulations

Agenda

• What are memristors?

• **Background – modern computers**

• Memory intensive computing

• CPU improvement

• Beyond von Neumann

• Summary
von Neumann 1945

Input Device

A+B+C

Write A+B+C

Memory

Control Unit

A+B+C

Arithmetic/Logic Unit

A+B+C

Processor

Read A+B+C

Output Device
Memory Wall

- Performance bottleneck => Slow
- High power

Figure source: http://en.community.dell.com/techcenter/high-performance-computing/w/wiki/2284.aspx
Memory Hierarchy

- **CPU Registers**
- **L1 cache**
- **L2 cache**
- **Main memory (DRAM)**
- **SRAM**

**Capacity (Size):**
- Small: <500B
- Big: 64KB, 8MB, 4GB, 100GB, 100GB

**Speed:**
- Fast: 1-2ns
- Slow: 5ns, 100ns, 1ms
The Need for New Memory Technology

High static power consumption for volatile memories

Source: E. Esteve, IPNEST
Memristors are the Next Memory!

- Nonvolatile
- Low power
- High endurance
- Fast
- CMOS compatible
- Dense
The Way for Commercialization

HP 100TB Memristor drives by 2018 – if you're lucky, admits tech titan

Universal memory slow in coming

By Chris Mellor, 1st November 2013

SanDisk, Toshiba Announce 32Gb Bilayer Cross-point ReRAM

Feb 22, 2013 00:44
Masahide Kimura, Nikkei Electronics

Everspin throws first ST-MRAM chips down, launches commercial spin-torque memory era

By Steve Dent posted Nov 14th, 2012 at 12:50 PM

CROSSBAR EMERGES FROM STEALTH-MODE; UNVEILS CROSSBAR RRAM NON-VOLATILE MEMORY TECHNOLOGY

August 05, 2013

Sony Aims to Commercialize 16Gb ReRAM in 2015

Aug 21, 2013 14:47
Jyunichi Oshita, Nikkei BP Semiconductor Research
Agenda

• What are memristors?
• Background – modern computers

• Memory intensive computing

• Continuous flow multithreading

• Logic circuits with memristors
• Summary
Sea of Memory

• Dense and fast
Memristor-Based Memory

- Larger
- Not just X replacement
Memory Intensive Architectures

• Computation (breaking the wall)
  – Enhance computation
  – Within memory
Agenda

• What are memristors?
• Background – modern computers
• Memory intensive computing
• **CPU improvement**
• Beyond von Neumann
• Summary
The Processor

Fetch | Decode | Execute | Write Back

READ A (from memory)
C = A+B
D = C+B
WRITE D (to memory)
**Switch on Event Multithreading**

- Fetch
- Decode 1
- Decode 2
- Address G
- Mem access
- Execute
- Write back

**Thread A**

**Thread B**

**Simple**

**Low performance**

**High energy**

Cache miss!!!

Memory Controller
Multistate Pipeline Register

Novel Memory Structure

- Fetch
- Decode 1
- Decode 2
- Address G
- Mem access
- Execute
- Write back

Continuous Flow MT (CFMT)

Fetch
Decode 1
Decode 2
Address G
Mem access
Execute
Write back

Thread A
Thread B

Cache miss!!!

CFMT – A Novel μArchitecture

- Simplicity of SoE MT
- Novel memory structure - MPR
- No pipeline flush
  - Enhance performance
  - Reduce energy

2X performance improvement

Logic with Memristors

• Memristors can be building blocks in logic circuits
  – With/without CMOS

• Memristor functionalities
  – Input
  – Output
  – Perform logic operation
  – State register (latch, Flip-Flop)
  – Configurable switch

Memristor Ratioed Logic (MRL) Enhancing Computation

- Similar to CMOS logic
- Using CMOS for inversion and amplification
- Memristors operate only as computational elements

MRL AND Operation

\[ V_{OUT} = V_{CC} \cdot \frac{R_{ON}}{R_{ON} + R_{OFF}} \approx V_{CC} \cdot \frac{R_{ON}}{R_{OFF}} \ll V_{CC} \]

<table>
<thead>
<tr>
<th>IN₁</th>
<th>IN₂</th>
<th>AND</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Increase resistance:

No current

Decrease resistance:

\[ R_{OFF} \gg R_{ON} \]
Need for Amplification

- Chain of memristor-only logic gates

\[ V_{\text{CC}} \]

\[ V_{\text{ON}} \]

\[ V_{\text{OFF}} \]

\[ R_{\text{ON}} = 100 \, \Omega \]

\[ R_{\text{OFF}} = 10 \, \text{k}\Omega \]

Extending CMOS Technology

• Integrating MRL with CMOS logic:
  — Signal restoration
  — Inversion

• Increase logic density
Agenda

- What are memristors?
- Background – modern computers
- Memory intensive computing
- CPU improvement
- **Beyond von Neumann**
- Summary
Logic within the Memory
Beyond von Neumann Architecture

Input Device -> Control Unit -> Arithmetic/Logic Unit -> Output Device

Memory

Logic within the Memory

Example – Image Processing

- CPU
- Accelerators – GPU
- Logic within the memory
Logic within Memristor Memory

- Based on memristor-based crossbar memory
- \( R_{\text{ON}} \rightarrow \text{logical '1'}, \ R_{\text{OFF}} \rightarrow \text{logical '0'} \)

Logic within Memory
Our Contribution

• Crossbar analysis and design
• Design methodology for IMPLY
• Current research
  – Parallel and dynamic computation within memory
  – New logic families
Neural Networks with Memristors
Mimicking the Brain

• The brain – no separation between computing and memory

• Machine learning algorithms – recognition

• Hardware implementation:
  – Real parallelism

Hebbian Learning with Memristors

- Implementing many learning algorithms
- Significant reduction in area:
  - 3 components vs. 70

Agenda

- What are memristors?
- Background – modern computers
- Memory intensive computing
- CPU improvement
- Beyond von Neumann

- Summary
Acknowledgements

• My advisors: Prof. Avinoam Kolodny, Prof. Uri Weiser, and Prof. Eby Friedman
• Prof. Yoav Etsion, Prof. Yuval Cassuto, Prof. Levi Schächter, Prof. Shuki Bruck, Dr. Eitan Yaakobi, Dr. Dotan Di Castro, Dr. Asaf Gal, Dr. Daniel Soudry, Yuval Nacson, Ravi Patel, Yifat Levy
• All my friends and colleagues at the TCE and EE
• My wife Noa, and my daughters Or and Neta
Summary

• Memristors are not just the next memory

Enhancing CPU performance

Breaking the memory wall

Memory Intensive

Hybrid CMOS-memristor logic

Logic within memory

Novel memory structures

Neuromorphic
Thanks!

http://memristor.shorturl.com
skva@tx.technion.ac.il