More than the Machine – Using Memristors for Computing

Dietmar Fey
Department Computer Science – Chair for Computer Architecture
Friedrich-Alexander-University Erlangen-Nürnberg
What is the Machine?

- Who is aware of HP’s Machine?
What is the Machine?

- Utilisation of different technologies

Together...

Electrons for compute
Electrons like to interact; easily moved; interaction needed for compute

+ Ions for storage
Ions like to interact; stay put; good for storage

+ Photons to communicate
Photons don’t like to interact or stay put; good for long-distances

Source: P. Ranganathan, “Saving the world together, one server at a time…” ACACES 2011
What is the Machine?

- Special purpose cores arbitrarily connected with pool of non-volatile memories – the memristors
  - Access times between 0.3 and 3 ns (< below 250 ns)
  - Mostly flat memory model
    - Paging and TLBs shall become obsolete
    - Vision: Cache becomes non-volatile
What is the Machine?

- HP will provide first products of a complete new computer architecture within the next two to three years
- Up 160 racks based on memristors connected to a cluster
  - Data capacity up to 160 Petabyte
  - Size of a refrigerator
What is the Machine?

- Processor cores and memory connected via high speed fiber optics
  - Bandwidth of 6 Terabit / second

- Machine rack no server
  - Architecture flexible configurable from mobile device up to large computer
What is the Machine?

- Schedule for the revolution
  - New memory controllers
  - New OS for the Machine: Linux++ → Carbon
Outline

- What is the Machine?
- Memristor technology
- Digital Boolean logic with memristors
- Ternary Computing using memristors
- Conclusion
**Memristor technology**

- **Memristor - The missing 4th element**
  - Predicated by Leon Chua in 1971
    - Predicated by Leon Chua in 1971
    - Experimentally found in 2008 at HP Lab

**Memristor—The Missing Circuit Element**

**LETTERS**

*The missing memristor found*

Dmitri B. Strukov¹, Gregory S. Snider¹, Duncan R. Stewart¹ & R. Stanley Williams¹
Memristor technology

- Two principal kinds of memristors
  - Change resistivity of the device, e.g. due to ion transfer

\[ R_{MEM}(x) = R_{ON} \cdot x + R_{OFF} \cdot (1 - x), \]
\[ \text{where } x = \frac{w}{D} \in (0, 1) \]

Image from http://bit-player.org/2012/remember-the-memristor

Total memristance = sum of resistances of the doped und undoped regions
Memristor technology

- ResitiveRAM (ReRAM): Growing of a conducting filament due to depositions of cations

\[ I_{\text{fil,SET}} = j_{0,\text{el}} A_{\text{fil}} \left( \exp \left( -\frac{e^2}{k_B T} n_{\text{fil}} \right) - 1 \right) \]

\[ I_{\text{TU}} = \frac{C^3 \sqrt{2 m_{\text{eff}} \Delta W_0}}{2 \lambda} \left( \frac{e^2}{\hbar} \right) \exp \left( -\frac{4 \pi X}{\hbar} \sqrt{2 m_{\text{eff}} \Delta W_0} \right) A_{\text{fil}} V_{\text{TU}}. \]

More complicated model than to the HP model before

Images and equations taken from

PCCP

Switching kinetics of electrochemical metallization memory cells

Stephan Menzel,*, Stefan Tappertzhofen,† Rainer Waser* and Ilia Vakoss*
Memristor technology

- Modelling memristor behaviour
  - Used in a SPICE simulation
    \[
    \frac{dx}{dt} = k \cdot i(t) \cdot f(x), \quad k = \frac{\mu_v R_{ON}}{D^2}
    \]
  - Using a model for a non-linear dopant drift (window function)

Zdeněk BIOLEK, Dalibor BIOLEK Viera BIOLKOVÁ
SPICE Model of Memristor with Nonlinear Dopant Drift
RADIOENGINEERING, VOL. 18, NO. 2, JUNE 2009

Used window function

\[
f(x) = 1 - (2x - 1)^2^p
\]
Memristor technology

- Modelling and simulating memristors
  - Use an equivalent SPICE circuit model
  - Simplifies execution of mixed-signal simulations

Zdeněk BIOLEK, Dalibor BIOLEK Viera BIOLKOVÁ
SPICE Model of Memristor with Nonlinear Dopant Drift
RADIOENGINEERING, VOL. 18, NO. 2, JUNE 2009

* Memristor SPICE Model
  * For Transient Analysis only
  * created by Zdenek and Dalibor Biolek

* Ron, Roff - Resistance in ON / OFF States
* Rinit - Resistance at T=0
* D - Width of the thin film
* \( \mu v \) - Migration coefficient
* \( p \) - Parameter of the WINDOW-function
  * for modeling nonlinear boundary conditions
* \( x \) - \( W/D \) Ratio, \( W \) is the actual width
  * of the doped area (from 0 to D)
* \( C_x \)

* RESISTIVE PORT OF THE MEMRISTOR *
  ***************************************
  \( E_{mem} \) plus aux value = \{-I(Emem)\*V(x)\*(Roff-Ron)\}
  Roff aux minus \{Roff\}

* Flux computation *
  **********************************************
  Eflux flux 0 value = \{SDT(V(plus,minus))\}

* Charge computation *
  **********************************************
  Echarge charge 0 value = \{SDT(I(Emem))\}

* WINDOW FUNCTIONS *
  FOR NONLINEAR DRIFT MODELING *
  **********************************************

*window function, according to Joglekar
  \( \text{fun}(x,p) = \{1-(2^*x-1)\}^2*p\} \)
*proposed window function
  \( \text{fun}(x,i,p) = \{1-(x-stp(-i))\}^2*p\) ENDs memristor

**Slide 13**
Memristor technology

- Modelling multi-bit feature
  - Demonstration in a SPICE simulation
Outline

- What is the Machine?
- Memristor technology
- Digital Boolean logic with memristors
- Ternary Computing using memristors
- Conclusion
Digital Boolean logic with memristors

- Different branches of computing with memristors

- Memristive Computing
  - Analog Computing
    - Neural networks, Neuromorphic processing, STDP
  - Digital Computing
    - Hybrid approaches: CMOS+memristors
    - CMOS-circuit like equivalent memristor networks
    - Ratioed Logic
    - IMPLY Logic

Digital Boolean logic with memristors

- **Ratioed Logic**

  - Creating simple AND- and OR- gates by (mem)resistive networks
  - Making following abstraction
    - Current flowing into the device: memristance ↓
    - Current flowing out of the device: memristance ↑

  *S. KVATINSKY, N. WALD, G. SATAT, A. KOLODNY, U.C. WEISER, G.E. FRIEDMAN
  MRL – Memristor Ratioed Logic
Digital Boolean logic with memristors

- Example for OR and AND gate for input $V_{in1} = 1$ and $V_{in2} = 0$

**OR gate**

**AND gate**
Digital Boolean logic with memristors

- IMPLY Logic
  - Based on conditional toggling (kind of 3-phase logic)
    - Initializing certain states in memristors by input data
    - Apply constant voltages ($V_{\text{cond}}$ and $V_{\text{set}}$) that possibly change states
    - Reading out the state (applying voltage that does not change states)
Digital Boolean logic with memristors

- State changing after time

![Diagram depicting the relationship between voltage and current over time, showing the behavior of memristors in the context of digital logic.](image)

- **V**<sub>SET</sub> makes q′ ← 1 if p is opened

- **V**<sub>COND</sub> let p′ ← p

- Behaviour corresponds exactly to IMPLY logic
  - \( p \rightarrow q = q' \)
- Can be expanded to NAND by subsequent IMP operations
Outline

- What is the Machine?
- Memristor technology
- Boolean logic with memristors
- Ternary Computing using memristors
- Conclusion
Ternary computing using memristors

- **Ternary computers**
  - Since the days of Konrad Zuse and John v. Neumann
    - Binary computers
  - Ternary system
    - Differentiates between 3 and not 2 states
  - 17th century: Caramuel y Lobkowitz
    - Investigated number system with digits 0, 1, and 2

http://ternary.3neko.ru/history_of_ternary.html
Ternary computing using memristors

- 18th century: Abraham Gotthelf Kästner
  - each number weighted sum of multiples of 3
  - Weights were -1, 0, and +1

- Donald Knuth
  - Denoted that as balanced ternary system

  - Fast carry-free addition with signed-digit (SD) numbers
  - Difficult to implement in digital electronics

- 1988: Parhami
  - Binary SD number system

- 1958: Brousentsov
  - SETUN ternary computer
Ternary computing using memristors

- May be a renaissance of ternary computers?
  - CMOS compatible,
  - fast,
  - Energy-poor,
  - multi-bit storing capable non-volatile memory cells like memristors

- Hybrid CMOS-memristor approach

Using the multi-bit feature of memristors for register files in signed-digit arithmetic units

Dietmar Fey
Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Department Computer Science 3, Chair for Computer Architecture, Martensstr. 3, 91054 Erlangen, Germany
Ternary computing using memristors

- Signed-digit number representation to base 2
  \[ w(a) = \sum_{i=0}^{n} a_i \cdot 2^i \]
  \[ a = (a_{n-1}, ..., a_0), \quad a_i \in \{-1, 0, 1\} \]

  Example: \(10\overline{1} = 1 \times 2^2 + 0 \times 2^1 - 1 \times 2^0 = 4 - 1 = 3\)  \(\overline{1} = -1\)
  
  \(1\overline{1}1 = 5 - 2 = 3\)
  
  \(011 = 2 + 1 = 3\)

- Used digital coding for signed digits (SD)

<table>
<thead>
<tr>
<th>a⁺</th>
<th>a⁻</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Not used</td>
</tr>
</tbody>
</table>
### Ternary computing using memristors

#### Carry-free addition in \( O(1) \)

<table>
<thead>
<tr>
<th>Binary addition</th>
<th>Signed digit addition</th>
</tr>
</thead>
</table>
| \[
  \begin{array}{cccc}
    1 & 0 & 1 & 1 \\
    + & 0 & 1 & 0 & 1 \\
    C & 0 & 0 & 0 & 1 - \\
    S & - & 1 & 1 & 1 & 0 \\
    C & 0 & 0 & 1 & 0 - \\
    S & - & 1 & 1 & 0 & 0 \\
    C & 0 & 1 & 0 & 0 - \\
    S & 0 & 1 & 0 & 0 & 0 \\
    S & 1 & 0 & 0 & 0 & 0
  \end{array}
\] | \[
  \begin{array}{cccc}
    1 & 0 & 1 & 1 \\
    + & 0 & 1 & 0 & 1 \\
    C & 1 & 1 & 1 & 1 & 0 \\
    Z \bar{0} \bar{1} \bar{1} \bar{1} 0 \\
    S \bar{1} 0 0 0 0 0 = (16)_{10}
  \end{array}
\] |

- \( O(n) \)
  - Best case: \( \log(n) \)

- \( O(1) \)

<table>
<thead>
<tr>
<th>( x_i )</th>
<th>( y_i )</th>
<th>( z_i )</th>
<th>( c_{i+1} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( c_i )</th>
<th>( z_i )</th>
<th>( s_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
<td>0</td>
</tr>
</tbody>
</table>
Ternary computing using memristors

- Addition / subtraction of (i) a SD number $a$ and a binary number $B$ and (ii) two SD numbers $c$ and $z$

(i)

\[
\begin{align*}
    c_i^+ &= a_i^+ \lor \overline{B_i \land a_i^-} \\
    z_i^- &= (a_i^+ \lor a_i^-) \oplus B_i \\
    \land: \text{ and} \quad \lor: \text{ or} \\
    \oplus: \text{ exor}
\end{align*}
\]

(ii)

\[
\begin{align*}
    s_i^+ &= z_i^- \land c_{i-1}^+ \\
    s_i^- &= c_{i-1}^+ \land z_i^- \\
    \land
\end{align*}
\]

- $a - B$: Subtraction can be simply reduced to addition

\[
a - B = (-1) \cdot ((-1) \cdot a + B)
\]

- Negative complement simply by exchange positive and negative part

\[
\begin{array}{c}
    a_i^+ \\
    \downarrow \quad \downarrow
\end{array}
\begin{array}{c}
    a_i^- \\
    \uparrow \quad \uparrow
\end{array}
\begin{array}{c}
    a_i^- \\
    \downarrow \quad \downarrow
\end{array}
\begin{array}{c}
    a_i^+ \\
    \uparrow \quad \uparrow
\end{array}
\]
Ternary computing using memristors

- Schematic of a digit processor cell
2 Signed-digit (SD) arithmetic

- Corresponding gate logic for an SD adder / subtractor cell
  - Completely implemented in SPICE
Ternary computing using memristors

- Schematic of a digit processor cell

  - Several cells are connected side-by-side to a row
Ternary computing using memristors

- Modelling multi-bit feature
  - Interfacing to produce binary input for digital processing circuit
Ternary computing using memristors

- Memristor-based SD arithmetic unit

![Diagram of memristor-based SD arithmetic unit]
Ternary computing using memristors

- Simulation result

![Simulation result diagram showing the simulation of ternary computing using memristors. The diagram includes voltage levels and states for variables a, b, and c, demonstrating addition and subtraction operations.]
Conclusion

- Possible computer architecture revolution happens?
- Core technology are NVM like memristors
- Proposal for first memristive Boolean logic gates
- Renaissance or break-through for ternary computers

Outlook
- First simple gates have to be realised
- Devices have to be improved
- From gates to complex systems