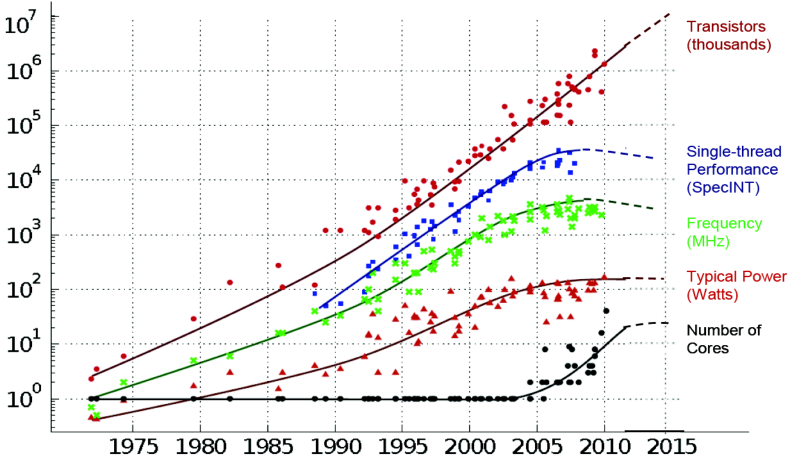
# Usages of Memristors

Bailey Denzer

University of Minnesota, Morris

December 5 2015

# 35 YEARS OF MICROPROCESSOR TREND DATA

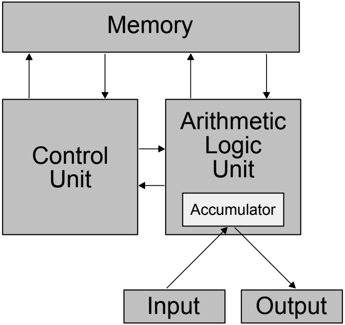


Original data collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond and C. Batten

# Von Neumann Architecture

Current computing architecture:

- Memory and processing are physically separate
- Transmission of information is becoming a bottleneck



## Main memory and Processor

Leakage is currently one of the main factors limiting increased computer processor performance

Main Memory:

- DRAM: Dynamic Random Access Memory
- Capacitors leak energy
- Need to be continually reset

Processor:

- Transistors
- Need a constant current to maintain state
- Has SRAM caches (Static Random Access Memory)

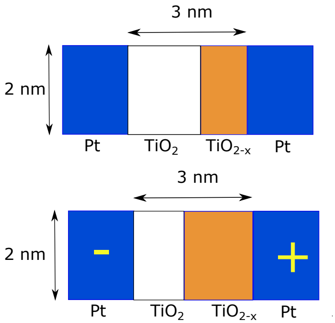
## Outline

1. Introduction
2. Background
   - Memristor
   - Crossbar Array
3. Memristors as Memory
4. Memristors for Logic Computations
5. Computation in Memory
   - Basic Design
   - Results on Large Data Sets
6. Conclusion

1. Introduction
2. Background
   - Memristor
   - Crossbar Array
3. Memristors as Memory
4. Memristors for Logic Computations
5. Computation in Memory
   - Basic Design
   - Results on Large Data Sets
6. Conclusion

# The Memristor

- mem: memory, -ristor: resistor
- Non-volatile, memory and switching device
- Originally used resistive $TiO_2$ and conductive $TiO_{2-x}$
- Applying current alters the state
- In the figure: nm (nano meters), Pt (platinum electrodes)

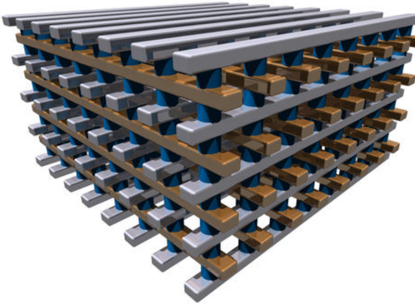


# History Abridged

- Originally theorized in 1971 by L. Chua
- Physical memristor construction published in 2008 by S. Williams et al.
- Long story short: the theoretical and 2008 memristor are not equivalent (S. Vongehr and X. Meng)
- More proper to call the memristor a resistance switching device
- Despite this, the memristor is a promising device

# Crossbar Array

- Scalable, layerable
- Potential to store Petabits of memory within a $\mathrm{cm}^3$
- Can be mass produced using current techniques



1. Introduction
2. Background
   - Memristor
   - Crossbar Array
3. **Memristors as Memory**
4. Memristors for Logic Computations
5. Computation in Memory
   - Basic Design
   - Results on Large Data Sets
6. Conclusion

# Other Potential Non-volatiles

STT-RAM

- Spin transfer torque random access memory
- New implementation of magnetoresistive RAM
- Requires a significant current to alter state

PCM

- Phase change memory
- Alters state by melting into either a conductive or resistive state

# Comparison to other prospective memories

- Retention: How long the device will retain its state after a write
- Endurance: How many writes before the device fails
- 3D capability: Whether or not the device can be constructed in layers

| | Memristor | PCM | STT-RAM | DRAM | Flash | HD |
|---|---|---|---|---|---|---|
| Chip area per bit ($F^2$) | 4 | 8-16 | 14-64 | 6-8 | 4-8 | |
| Energy per bit (pJ) | 0.1-3 | 2-100 | 0.1-1 | 2-4 | $10^1 – 10^3$ | $10^6 – 10^7$ |
| Read time (ns) | <10 | 20-70 | 10-30 | 10-50 | 25,000 | 5-8 X $10^6$ |
| Write time (ns) | 20-30 | 50-500 | 13-95 | 10-50 | 200,000 | 5-8 X $10^6$ |
| Retention | 10 years | <10 years | Weeks | < Second | ~10 years | ~10 years |
| Endurance (cycles) | $10^{12}$ | $10^7 – 10^8$ | $10^{15}$ | $10^{17}$ | $10^3 – 10^6$ | $10^{15}$ |
| 3D capability | Yes | No | No | No | Yes | |

## Newer Implementation

Tantalum-Oxide based memristor: S. Williams et al.

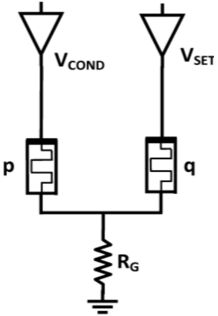
- Switching times from 105 to 120 pico seconds
- Associated energies: 1.9 and 5.8 pico joules
- The device in question was 2 micro meters in length

1. Introduction
2. Background
   - Memristor
   - Crossbar Array
3. Memristors as Memory
4. **Memristors for Logic Computations**
5. Computation in Memory
   - Basic Design
   - Results on Large Data Sets
6. Conclusion

## Material Implication Logic

IMP: $(p \rightarrow q) \vdash (\neg p \vee q)$

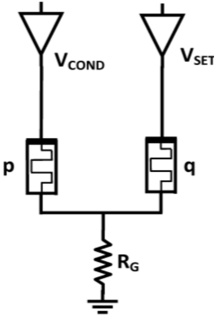
- Kvatinsky et al.
- Like the nand gate, the imply gate implements any boolean function
- Memristors P and Q with initial values p and q
- Memristor Q is treated as the output
- $V_{\mathrm{cond}}$ and $V_{\mathrm{set}}$ are voltages where $(|V_{\mathrm{cond}}| < |V_{\mathrm{set}}|)$
- $R_G$ has a resistance between $R_{\mathrm{Off}}$ and $R_{\mathrm{On}}$



# Material Implication Logic

IMP

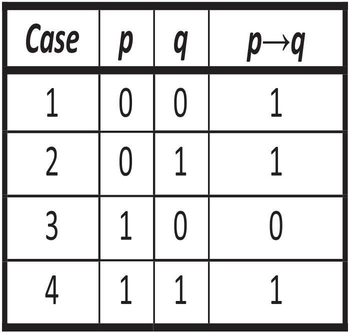
- Apply voltage $V_{\text{cond}}$ to P and $V_{\text{set}}$ to Q



# Material Implication Logic

IMP

- If $p = q = 0$ (high resistance)
- Voltage on Q is approximately $V_{\mathrm{set}}$
- Q is switched to 1.

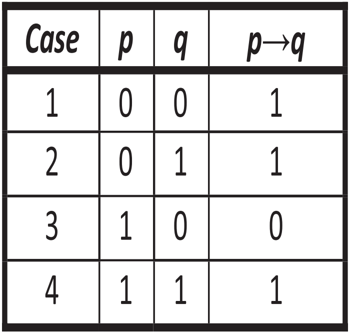


| Case | $p$ | $q$ | $p \to q$ |
|---|---|---|---|
| 1 | 0 | 0 | 1 |
| 2 | 0 | 1 | 1 |
| 3 | 1 | 0 | 0 |
| 4 | 1 | 1 | 1 |

# Material Implication Logic

IMP

- If p = 0 and q = 1,
- the state of Q is unchanged

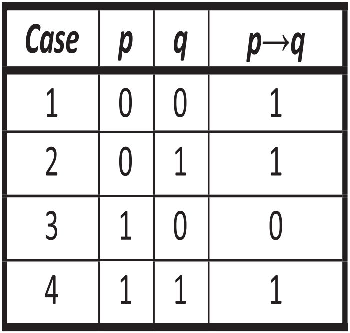


| Case | $p$ | $q$ | $p \to q$ |
|---|---|---|---|
| 1 | 0 | 0 | 1 |
| 2 | 0 | 1 | 1 |
| 3 | 1 | 0 | 0 |
| 4 | 1 | 1 | 1 |

# Material Implication Logic

IMP

- If p = 1 (low resistance)
- Voltage on Q is $V_{\text{set}} - V_{\text{cond}}$
- Voltage on Q is small enough that the state of Q is unchanged.



| *Case* | *p* | *q* | *p→q* |
|---|---|---|---|
| 1 | 0 | 0 | 1 |
| 2 | 0 | 1 | 1 |
| 3 | 1 | 0 | 0 |
| 4 | 1 | 1 | 1 |

1 Introduction

2 Background
- Memristor
- Crossbar Array

3 Memristors as Memory

4 Memristors for Logic Computations

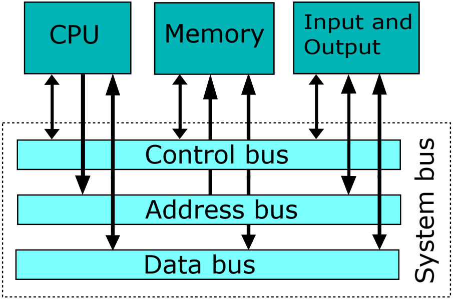
5 Computation in Memory
- Basic Design
- Results on Large Data Sets

6 Conclusion

# Von Neumann Bottleneck

Issues:

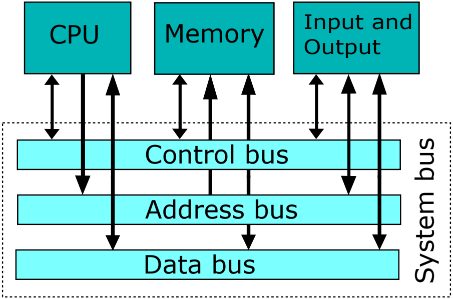
- Information between processor and main memory is shared by a system bus
- The bus can only access the processor or memory, one at a time



# Von Neumann Bottleneck

Issues:

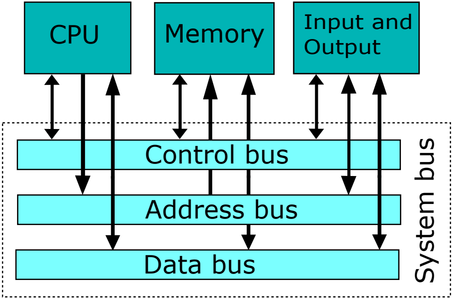
- Processor will be idle while the bus is getting information from main memory
- Study from 1996 found that 3 out of 4 CPU cycles spent waiting for memory
- Processor processing and main memory transfer rate are more than the data transfer rate of the bus



# Von Neumann Bottleneck

Some current mitigations:

- SRAM based caches between the CPU and main memory
- Providing the CPU a limited stack or scratch-pad memory



# Computation in Memory Architecture

Computation in Memory (CIM) as propsed by Hamdioui et al. aims to combine main memory and processing into a single crossbar array. The memory/computing crossbar array allows for:

- Massive parallelism
- Little to no power leakage (no longer accessing SRAM caches)
- Performance improvement at lower energy and area (no communication bottleneck)

# Specifics of Architectures Compared

Von Neumann architecture:

- 22nm multi-core implementation
- 64 clusters
- Each cluster has 32 ALUs that share an 8KB L1 cache

Computation in Memory architecture:

- 5nm memristor crossbar
- Crossbar size is equal to total cache size of the VN architecture

## Data Sets to Compare VN vs CIM

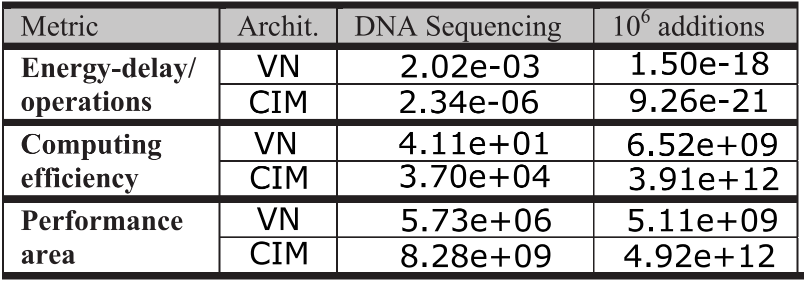
DNA comparison:

- A sorted index of reference DNA is created in order to identify the locations of matches or mismatches in another sequence.
- Comparing 200 GB of DNA to a 3 GB healthy reference.

$10^6$ parallel additions

# Results

- Energy-delay / operations: Energy consumed per operation (joules)
- Computing efficiency: Number of operations per energy (n/J)
- Performance Area: Operations per area



| Metric | Archit. | DNA Sequencing | $10^6$ additions |
|---|---|---|---|
| **Energy-delay/ operations** | VN | 2.02e-03 | 1.50e-18 |
| | CIM | 2.34e-06 | 9.26e-21 |
| **Computing efficiency** | VN | 4.11e+01 | 6.52e+09 |
| | CIM | 3.70e+04 | 3.91e+12 |
| **Performance area** | VN | 5.73e+06 | 5.11e+09 |
| | CIM | 8.28e+09 | 4.92e+12 |

1. Introduction
2. Background
   - Memristor
   - Crossbar Array
3. Memristors as Memory
4. Memristors for Logic Computations
5. Computation in Memory
   - Basic Design
   - Results on Large Data Sets
6. Conclusion

# The Good

- On their own memristors seem to be a very competitive memory
- They are scalable and potentially cheap to manufacture
- Computation in memory architecture has excellent potential

## The Bad

- Completely new logic operations could require significant changes to software
- Combined with a complete overhaul of the architecture, CIM could take some time to be developed

# Discussion

Questions?

## References

- S. Hamdioui, L. Xie, H. A. D. Nguyen, M. Taouil, K. Bertels, H. Corporaal, H. Jiao, F. Catthoor, D. Wouters, L. Eike, and J. van Lunteren. Memristor based computation-in-memory architecture for data-intensive applications. 2015
- S. Kvatinsky, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser. Memristor-based material implication (imply) logic: Design principles and methodologies. 2014
- A. C. Torrezan, J. P. Strachan, G. Medeiros-Ribeiro, and R. S. Williams. Sub-nanosecond switching of a tantalum oxide memristor. 2011
- S. Vongehr and X. Meng. The missing memristor has not been found. 2015
- L. O. Chua. Memristor - the missing circuit element. 1971
- D. B. Strukov, X. Meng, D. R. Stewart, and R. S. Williams. The missing memristor found. 2008