Machines are Beating Humans at Complex Inference Tasks

- game of Go is complex → huge search space: \( \sim 250^{150} \) (Go) vs. \( \sim 35^{80} \) (Chess)

- AlphaGo machine: 1202 CPUs+176 GPUs

- **HUGE Energy** (and latency) **Cost** \( \sim 10,000 \times \) more than human brain

- Critical issue at the Edge – IoT, wearables, autonomous
Energy Cost - Memory Access vs. Computation

\[ \frac{E_{\text{mem}}}{E_{\text{mac}}} \approx 10 \times \text{to} \ 100 \times \] (in SRAMs)

ML kernel-level energy breakdown for TM (8-b operands) [Kang, Shanbhag]

Dot product

<table>
<thead>
<tr>
<th></th>
<th>Integer</th>
<th>ADD</th>
<th>Mult</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 bits</td>
<td>0.03 pJ</td>
<td>0.2 pJ</td>
<td></td>
</tr>
<tr>
<td>32 bits</td>
<td>0.1 pJ</td>
<td>3 pJ</td>
<td></td>
</tr>
</tbody>
</table>

Computation energy (45nm)

<table>
<thead>
<tr>
<th></th>
<th>Memory</th>
<th>64 bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache 8 KB</td>
<td>10 pJ</td>
<td></td>
</tr>
<tr>
<td>Cache 32 KB</td>
<td>20 pJ</td>
<td></td>
</tr>
<tr>
<td>Cache 1 MB</td>
<td>100 pJ</td>
<td></td>
</tr>
<tr>
<td>DRAM</td>
<td>1.2 – 2.6 nJ</td>
<td></td>
</tr>
</tbody>
</table>

*Post-layout simulations with SRAM + synthesized logic in 65nm CMOS
Energy Costs in the Memory Hierarchy

Cost per bit ($)

Latency
- 10 – 30 ns
- 20-60-clock cycles
- 60 – 100 μs
- (12 – 20) × 10^4-clock cycles
- 1 ms
- 10^6-clock cycles

Bandwidth
- 25.6 Gb/s
- 16 Gb/s
- 8-12 Gb/s

Relative Energy Costs
- SRAM vs. compute: ≈ 10×–100×
- DRAM vs. compute: ≈ 500× [Sze, ISSCC’16]
- Flash vs. compute: ≈ 1000×

Key Question
How to reduce memory access costs?

[Yang, J. Joshua, Nature Nano, 2013]
To Speed Up AI, Mix Memory and Processing

New computing architectures aim to extend artificial intelligence from the cloud to smartphones

By Katherine Bourzac

If John von Neumann were designing a computer today, there’s no way he would build a thick wall between processing and memory. At least, that’s what computer engineer Naresh Shanbhag of the University of Illinois at Urbana-Champaign believes. The eponymous von Neumann architecture was published in 1945. It enabled the first stored-memory, reprogrammable computers—and it’s been the backbone of the industry ever since.

Now, Shanbhag thinks it’s time to switch to a design that’s better suited for today’s data-intensive tasks. In February, at the International Solid-State Circuits Conference (ISSCC), in San Francisco, he and others made their case for a new architecture that brings computing and memory closer together. The idea is not to replace the processor altogether but to add new functions to the memory that will make devices smarter without requiring more power.

Proposed Solution

Deep In-memory Architecture (DIMA)

https://spectrum.ieee.org/computing/hardware/to-speed-up-ai-mix-memory-and-processing

“breaching the memory wall”
The Deep In-memory Architecture (DIMA)

[Kang, et al., -ICASSP-2014] [Kang, et al., -JSSC-2018]

- Multi-row functional READ (reads multiple-bits/col/precharge)
- Analog, mixed-signal low-SNR fabric
- Bitline processing (SIMD analog processing)
- Cross bitline processing (analog averaging enhances SNR)
- Low complexity, low (decision) rate digital output

Inference/decisions
Functional Read (FR) – Voltage Mode (for SRAM)

- single FR → vector inner product
- multi-row access per precharge
- PWM, PAM, PWAM access pulses

Per-column dot-product

\[ \Delta V_{BL} = \frac{V_{pre}}{R_{BL}C_{BL}} \sum_{i=0}^{N-1} f(V_i)T_i d_i \]

\((T_i \ll R_{BL}C_{BL})\)

\(\Delta V_{BL} \propto\) multiple-bits per column
Bit-line Analog Processors

computations dominating machine learning

<table>
<thead>
<tr>
<th>Kernel</th>
<th>BLP</th>
<th>CBLP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manhattan distance</td>
<td>subtract-compare</td>
<td>aggregation</td>
</tr>
<tr>
<td>Euclidean distance</td>
<td>subtract-square</td>
<td>aggregation</td>
</tr>
<tr>
<td>Dot product</td>
<td>multiply</td>
<td>weighted aggregation</td>
</tr>
<tr>
<td>Hamming distance</td>
<td>XOR</td>
<td>aggregation</td>
</tr>
</tbody>
</table>

charge-redistribution based cross bitline aggregation

\[ SAD(\overline{D}, \overline{P}) = \sum_{l=1}^{L} |D_l - P_l| \]
SRAM DIMA IC Prototypes

with 16kB standard 6T SRAM in 65nm CMOS

multi-functional

FIRST random forest IC

on-chip learning

iso-accuracy comparisons wrt. post-layout 8-b digital processor & measured SRAM read energy

SVM, TM, k-NN, MF;
MIT-CBCL, MNIST, ..;
energy savings = 10X;
EDP reduction = 50X;

RF with 64 trees;
KUL traffic sign;
energy savings = 3X
EDP reduction = 7X

8b, 128-dim SVM; MIT-CBCL dataset;
SGD-based learning;
energy savings = 21X;
EDP reduction = 100X;

[JSSC January 2018]

[ESSCIRC 2017,
JSSC (special issue) May 2018]

[ISSCC 2018, JSSC special issue (Invited)]
ISCAS 2018 - Migrate DIMA into Flash

Challenges in Flash (relative to SRAM) → DIMA Solution

• NAND flash bit cell is $(32 \times)$ smaller than SRAM → multi-col processing
• Much larger BL caps → use current sensing
• High $V_t$ variations → use high dimensional vector processing
• Slower logic devices → use mixed-signal analog circuits
NAND Flash-based DIMA

- **Multi-col functional read:** converts a stored word into output voltage
- **Multi-BL Processor (MBLP):** performs scalar mixed signal multiplication
- **Cross BL Processing:** aggregation via charge sharing
MC Functional Read – Current Mode

- Current sensing - BL not discharged
- Binary word stored horizontally
- Use time modulated SEL signals

\[
\Delta V_{OUT} = \frac{I_{ON}}{C_{OUT}} \sum_{i=0}^{N-1} T_i d_i \propto D
\]
Simulation Methodology

**Challenge:** need to reflect device/process non-idealities at the system level

**Solution:** use device models + array parameters to estimate energy, delay and behavior

- **behavioral models** - $V_t$ variations, ICI, pattern dependency, read/pgm disturbance
- **energy and delay models:** estimated from circuit simulations & analysis
Architectural Set-up

- 32nm node; 16kB/page; 64 pages/block; 3000 blocks/plane; 4 planes/IC
- 200×320 8-b images; one image stored in 4 pages in 4 planes
- I/O limited to 800MB/s (ONFI 4)
Machine Learning Applications

Face detection via linear SVM

decision rule
\[ w^T x + b \geq 0 \rightarrow \text{face} \]
else \( \rightarrow \no\text{face} \)

Face recognition via \( k\)-NN

Face database \( (d) \) stored in NAND flash

“Based on 3 closest images pick person 1”

- Caltech 101 dataset
- extended Yale B dataset (2336 test images; 28 classes)
- input buffer stores weights
- input buffer stores reference image
Simulation Results - Accuracy

- Detection accuracy robust to $V_t$ variations in the typical range
- SVM accuracy: 92%; $k$-NN accuracy (top-3): 95%
Energy & Throughput Benefits

Single NAND IC (4 planes/IC)

Normalized Energy

SSD (16 ICs/package)

Normalized Energy

- 8.3×
- 23×

9× 15× → throughput enhancement
Summary & Future Work

• Deep In-memory Architecture – energy (23×) and throughput (15×) enhancement for SSDs.
• DIMA in other technologies – FD-SOI, MRAM, eDRAM, DRAM, emerging devices (e.g., RRAM)
• Scaled-up DIMA – multi-bank architectures for DNNs
• Robustifying DIMA – using Shannon-inspired statistical error compensation, on-chip learning [ISSCC 2018]
• Programmable DIMA – programming models, compilers (with Adve, Kim (UIUC)) [ISCA 2018]
• Inference algorithms for DIMA – analog data flow
• DIMA physical compilers – auto. synth. of DIMA cores
Acknowledgements

This work was supported in part by Systems on Nanoscale Information fabriCns (SONIC), one of the six SRC STARnet Centers, sponsored by MARCO and DARPA.

http://shanbhag.ece.illinois.edu