In- and Near-Memory Computing Using 2D/3D Resistive Memories

H.-S. Philip Wong, Haitong Li, Weier Wan
Stanford University
Big Data

Graph analytics

Machine learning

Image sources: Intel, bdtechtalks.com, Oracle, internetofbusiness.com
ABUNDANT-DATA COMPUTING

FAVORS DOMAIN-SPECIFIC ARCHITECTURES

System throughput / Watt

Adapted from AMD Hot Chips’19

Tasks

TPU

GPU

CPU

H.-S. Philip Wong
ABUNDANT-DATA COMPUTING
FAVORS DOMAIN-SPECIFIC ARCHITECTURES

System throughput / Watt

Adapted from AMD Hot Chips’19

Domain-specialization requires large number of transistors
PEAK THROUGHPUT CORRELATES WITH TRANSISTOR COUNT

Slope: 1.02
BANDWIDTH DEFICIT

H. S. Philip Wong, K. Akarvardar, DAC, plenary paper (2020)
In this N3XT System, high density on-chip nonvolatile memory is integrated with high speed on-chip nonvolatile memory and energy efficient logic (thin device layers). The system also features dense ILV connectivity (nanometer scale) and energy efficient memory access transistors. Nonvolatile memory cells are also present to enhance data retention and reliability. This integration is being worked on by W. Hwang, W. Wan, Y. Malviya, H. Li, M. Lee, M. Aly, H.-S. P. Wong, and S. Mitra at TSMC, with work in progress from 2017 to 2020.
EDGE INTELLIGENCE PIPELINE
TODAY: HEAVILY RELIES ON CLOUD PROCESSING

Sensor → Edge Computation → Cloud

Raw data → Meta-data

Decisions
WHY WE NEED EDGE COMPUTATION
REDUCED TOTAL COMPUTATION + COMMUNICATION ENERGY

Video Recognition Pipeline

100% cloud

Raw video

Background Subtraction

Feature Extraction

Classification

100% edge

More edge processing

Optimal Energy

Total Energy

Communication

Computation

Raw video

Background Subtraction

Feature Extraction

Classification

Keshavarzi, van den Hoek, IEEE Design & Test, April 2019
PATH TOWARDS EDGE INTELLIGENCE

Cloud-based → Edge Inferencing → Edge Training (Fine-tuning)

Efficiency & Privacy
EDGE INFERENCE WORKLOAD
WEIGHT MEMORY ACCESS DOMINATES ENERGY CONSUMPTION

Input data
• Intermittent
• Small batch size

Weight Memory (DRAM)

High-capacity external memory chip

Accelerator chip

SRAM Buffer
EDGE INFRINGEMENT WORKLOAD
WEIGHT MEMORY ACCESS DOMINATES ENERGY CONSUMPTION

Weight Memory (DRAM)
Reload weights for every input

Input data
• Intermittent
• Small batch size
Input data
- Intermittent
- Small batch size

Limited Energy-Efficiency
E.g. MobileNet
- 10M 4b-weights, 1 GOPS
- Even if compute is free

→ 2.5 TOPS/W (upper bound)

Source: B. Murmann (Stanford)
OVERCOME MEMORY-WALL BOTTLENECK

NEED TO STORE ALL WEIGHTS ON CHIP
## DNN MEMORY REQUIREMENTS

**LARGER MODEL → HIGHER ACCURACY**

<table>
<thead>
<tr>
<th>Application</th>
<th>Model</th>
<th>Model Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visual recognition</td>
<td>ResNet (CNN)</td>
<td>120 MB</td>
</tr>
<tr>
<td>Image captioning</td>
<td>CNN + LSTM</td>
<td>150 MB</td>
</tr>
<tr>
<td>Language model</td>
<td>LSTM</td>
<td>2.5 GB</td>
</tr>
</tbody>
</table>

**ImageNet Accuracy (%) vs. Model Size (MBytes, log scaled)**

- **MobileNet-v2 (2018)**
- **Inception-v4 (2016)**
- **MUXNet-xs (2020)**
- **Tiny DarkNet**

Source: W. Hwang, W. Wan, Prof. S. Mitra (Stanford)
ON-COMP SRAM CAPACITY SCALING TREND
FOR DATA-CENTER CPU & GPU

Estimated On-chip SRAM (MB)

Launch Year

2006 2009 2012 2015 2018

CPU

GPU

3.8 GBytes @ 1.4 nm node

Source: W. Hwang, Prof. S. Mitra (Stanford)
ON-CHIP MEMORY CAPACITY FOR EDGE SOC

INADEQUATE FOR DNN MODELS

- Die size: 14 mm$^2$
- 0.021$\mu$m$^2$ SRAM cell @ 5nm
- Assume 10% SOC area is weight memory
- Assume 80% SRAM area efficiency

→ 6.4 MB SRAM

Die shot of Apple W1 SoC (used in Airpod)

Very optimistic assumptions!
"NEW" MEMORIES
RANDOM ACCESS, NON-VOLATILE, NO ERASE BEFORE WRITE, ON-CHIP INTEGRATION

STT-MRAM
Spin transfer
torque magnetic
random access
memory

PCM
Phase change
memory

RRAM
Resistive switching
random access
memory

CBRAM
Conductive bridge
random access
memory

FEFET
Ferro-electric
field effect
transistor

RRAM: SWITCHING BEHAVIORS

RRAM

Pristine “0”

Set/forming “1”

Reset “0”

H.-S. Philip Wong

RRAM: MEMORY CHARACTERISTICS

Key properties:
- 10 ns read/write, $V_{\text{PROG}} \sim 1 - 2$ V
- Write current $\sim nA - 10's \mu A$
- 1E12 cycles endurance @ device
- 1T1R, $\sim 6F^2$
- 3D RRAM (similar to 3D NAND)
- Scalable to $< 5$ nm & smaller
- Fab-friendly, easy to embed

<table>
<thead>
<tr>
<th>Speed</th>
<th>0.1’s ns</th>
<th>1 - 10’s ns</th>
<th>10’s -100’s ns</th>
<th>0.1-10’s ms</th>
<th>&gt; 100’s ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU cycles</td>
<td>1</td>
<td>10 - 100</td>
<td>&gt;100</td>
<td>$10^6 - 10^8$</td>
<td>$&gt;10^9$</td>
</tr>
<tr>
<td>Size (bytes)</td>
<td>100</td>
<td>K</td>
<td>M</td>
<td>G</td>
<td>T</td>
</tr>
</tbody>
</table>

Top Electrode
Bottom Electrode

- TiN, Al
- HfO$_x$, TaO$_x$, AlO$_x$, AlO$_x$Ny
- TiN

Main Memory (DRAM)
Secondary Storage (Flash, Disk)
Tertiary Storage (Disk, RAID)
3D VERTICAL RRAM

Metal Pillar Electrode

Memory cell

Metal Plane Electrode

S. Yu, H.-Y. Chen et al., Symp. VLSI Tech. 2013
3D VERTICAL RRAM

Stanford:
IEDM ’12, ’13, ’14
VLSI ’13, ’14, ’16
DATE ’15, Nature Comm ’15
PROGRESS TOWARDS A PRACTICAL TECHNOLOGY
MEMORY-LOGIC INTEGRATION

Capacity limited by SRAM scaling

Logic

SRAM

Off-chip DRAM
MEMORY-LOGIC INTEGRATION

Capacity limited by SRAM scaling

Off-chip DRAM

Higher memory capacity

Reduce off-chip memory access
MEMORY-LOGIC INTEGRATION

Capacity limited by SRAM scaling

Logic
SRAM
Off-chip DRAM

3D on top of logic

HD NVM
Logic
SRAM
HD NVM
Off-chip DRAM

Reduce off-chip memory access
MEMORY-LOGIC INTEGRATION

Capacity limited by SRAM scaling

Off-chip DRAM

3D on top of logic

Compute-memory integration in 3D
NEW TECHNOLOGY-SYSTEM DESIGN SPACE

- More memory:
  - Logic
  - SRAM
  - HD NVM

- Off-chip DRAM

- More compute:
  - Logic
  - SRAM
  - HD NVM

- No off-chip DRAM

- From chip to system

NEW TECHNOLOGY-SYSTEM DESIGN SPACE

- More memory
  - Logic
  - SRAM
  - HD NVM

- Off-chip DRAM

- More SRAM
  - Logic
  - SRAM

- Off-chip DRAM

- No off-chip DRAM
  - Logic
  - SRAM
  - HD NVM

From chip to system

NEW TECHNOLOGY-SYSTEM DESIGN SPACE

- More memory
  - Logic
  - SRAM
  - HD NVM

- Off-chip DRAM

- More compute
  - Logic
  - SRAM
  - HD NVM

- No off-chip DRAM
  - Logic
  - SRAM
  - HD NVM

From chip to system

NEW TECHNOLOGY-SYSTEM DESIGN SPACE

- More memory
  - Logic
  - SRAM
  - HD NVM

- Off-chip DRAM

- More compute
  - Logic
  - SRAM
  - HD NVM

- No off-chip DRAM
  - Logic
  - SRAM
  - HD NVM

From chip to system
CASE STUDY: DNN ACCELERATOR + NVM
RESNET-50, GOOGLENET, MOBILENET, FASTERRCNN, YOLO-TINY

Dataflows:
Output stationary
Weight stationary

Memory

On-chip
Off-chip

LPDDR3 DRAM

Weight SRAM

PE

Input Fmap SRAM

Output Fmap SRAM

{32, 64, 128, 256, 1024} KB

16 × 16
24 × 24
32 × 32

{32, 64, 128, 256, 1024} KB

MRAM
3D VRRAM
eDRAM/SRAM
MODELING AND EVALUATION METHODOLOGY

Evaluation framework built upon open-source SCALE-Sim from Arm Research
https://github.com/ARM-software/SCALE-Sim
ENERGY-AREA PARETO FRONTIERS

colormap represents total SRAM capacity in a specific design w/ off-chip DRAM

Baseline designs with DRAM, running inference with ResNet-50
CASE STUDY: DNN ACCELERATOR + NVM
SYSTOLIC ARRAY ARCHITECTURE, RESNET-50 INFERENCE

![Graph showing energy consumption vs. area for different memory technologies.](chart)

- Baseline: off-chip DRAM
- 3D VRRAM
- MRAM
- Hybrid VRRAM/DRAM

1.83X reduction in energy for hybrid VRRAM/DRAM compared to baseline.
ENERGY-AREA TRADEOFFS WITH 3D VRRAM

AREA & ENERGY SAVINGS: HIGH DENSITY NVM ON CHIP; 4X LESS SRAM

- 33% Area & Energy Savings: High density NVM on chip; 4X less SRAM.
- 416 KB SRAM (w/o VRRAM) → 96 KB SRAM (w/ VRRAM).

Diagram showing the tradeoffs between area and energy for different memory technologies, including baseline off-chip DRAM, 3D VRRAM, MRAM, and Hybrid VRRAM/DRAM.
LARGE MEMORY CAPACITY:
X, Y, Z, M

- Two dimensional down scaling
- Store multiple bits per memory cell
- 3D layers

Inspired by Jian Chen (Western Digital), *IEDM Short Course*, 2019
RRAM DENSITY LIMITED BY ACCESS TRANSISTOR

Limited by access transistor width/# fins (provide enough write current)

Limited by Contact Gate Pitch (CGP) & Metal Pitch
RRAM DENSITY VS. SRAM DENSITY
LOWERING THE PROGRAMMING CURRENT

RRAM programming current limits density
MORE CAPACITY WITH MULTI-BIT PER CELL

![Graph showing conductance distribution with multi-bit per cell capacity improvement.](image_url)
TBIT-CLASS 3D VERTICAL RESISTIVE SWITCHING MEMORY

Metal Pillar Electrode

Metal Wordplane (WP)

Wordline (WL)

Bitline (BL)

Wordplane Connection (WPC)

Memory Cell

Pillar Driver: FinFET

Wordplane (WP) Connection

x

y

z

Metal Wordplane (WP)

Wordline (WL)

Bitline (BL)
3D RRAM DENSITY VS SRAM DENSITY

GOING 3D

1 layer of RRAM

2 layers of RRAM

Density (MB/mm²)

Technology node

16nm 10nm 7nm 5nm

RRAM

RRAM

RRAM

RRAM

1.7x 3.5x 7x
DIGITAL ACCELERATOR W/ RRAM

RRAM Weight Memory

SRAM Buffer
DIGITAL ACCELERATOR W/ RRAM

LOAD WEIGHTS FROM RRAM TO DIGITAL PE ARRAY

• Batch-1 inference: no weight reuse
  → reload weights for every input

• RRAM weight memory (MBs) >> SRAM buffer (KBs)
  → Weight memory access dominates energy consumption
COMPUTE-IN-MEMORY USING RRAM
WEIGHT-STATIONARY -- ELIMINATES WEIGHT MEMORY ACCESS

Analog matrix-vector multiplication
MULTI-CORE PIPELINED DNN INFERENCE
HIGHER THROUGHPUT & LOWER LATENCY
DESIGN CHALLENGES FOR CIM HARDWARE
ADC LIMITS ENERGY EFFICIENCY

- CIM activates multiple rows & columns simultaneously → Large current
  - dominates energy consumption
DESIGN CHALLENGES FOR CIM HARDWARE

ADC LIMITS ENERGY EFFICIENCY

- CIM activates multiple rows & columns simultaneously → Large current
  - dominates energy consumption

- High precision ADC dominates area and energy consumption
DESIGN CHALLENGES FOR CIM HARDWARE

RECONFIGURABILITY

- Re-programming RRAMs is expensive
DESIGN CHALLENGES FOR CIM HARDWARE

RECONFIGURABILITY

- Re-programming RRAMs is expensive

- Design favors larger array size to amortize large ADC energy & area
  - Inflexible for small weights & vector operations (e.g. depthwise convolution)
RECONFIGURABLE RRAM CIM CHIP
256×256 RRAM ON 130nm CMOS

Chip micrograph

Cross-section TEM
CMOS at the bottom

ENERGY-EFFICIENCY
Static voltage-mode sensing scheme

RECONFIGURABILITY
Dataflow reconfigurability realized by Transposable Neurosynaptic Core
ENERGY-EFFICIENCY
Static voltage-mode sensing scheme

RECONFIGURABILITY
Dataflow reconfigurability realized by Transposable Neurosynaptic Core
\[
(V_1 \ \cdots \ V_m) \begin{pmatrix}
G_{11} & \cdots & G_{1n} \\
\vdots & \ddots & \vdots \\
G_{m1} & \cdots & G_{mn}
\end{pmatrix} = (Y_1 \ \cdots \ Y_n)
\]

\(Y_i\) can be current or voltage depending on the sensing scheme.
CONVENTIONAL CURRENT-MODE SENSING

- Drive all BLs to $V_{\text{read}}$

X

0 $\rightarrow$ BL1

1 $\rightarrow$ BL2

Voltage Clamp | Voltage Clamp

Current Sense-amp
CONVENTIONAL CURRENT-MODE SENSING

- Drive all BLs to $V_{\text{read}}$
- Clamp SLs to $V_{\text{ref}}$

$X$

$0 \rightarrow \{\begin{array}{c}
\text{BL1} \quad V_{\text{read}} \\
\end{array} \rightarrow \{\begin{array}{c}
\text{G}^+ \\
\text{SL}^+ \\
\text{G}^- \\
\text{SL}^- \\
\text{V}_{\text{ref}} \\
\text{V}_{\text{ref}} \\
\text{Voltage Clamp} \\
\text{Voltage Clamp} \\
\text{Current Sense-amp}
\end{array} \}
\}$

$1 \rightarrow \{\begin{array}{c}
\text{BL2} \quad V_{\text{read}} \\
\end{array} \rightarrow \{\begin{array}{c}
\text{G}^+ \\
\text{SL}^+ \\
\text{G}^- \\
\text{SL}^- \\
\text{V}_{\text{ref}} \\
\text{V}_{\text{ref}} \\
\text{Voltage Clamp} \\
\text{Voltage Clamp} \\
\text{Current Sense-amp}
\end{array} \}
\}$
CONVENTIONAL CURRENT-MODE SENSING

- Drive all BLs to $V_{\text{read}}$
- Clamp SLs to $V_{\text{ref}}$
- Activate WLs with “1” input
CONVENTIONAL CURRENT-MODE SENSING

- Drive all BLs to $V_{\text{read}}$
- Clamp SLs to $V_{\text{ref}}$
- Activate WLs with “1” input
- Sense SL current as output of MVM

$$I_{SL} = (V_{\text{read}} - V_{\text{ref}}) \sum_i X_i G_i$$
LIMITATIONS OF CURRENT-MODE SENSING

Limited Energy-Efficiency
- Static current flows in array during the entire period of sensing
  
  E.g. Apply 0.5V across 100KΩ RRAM for 10ns
  \[ \rightarrow 25\text{fJ} / \text{MAC} \rightarrow < 80\text{TOPS/W} \]
LIMITATIONS OF CURRENT-MODE SENSING

Limited Throughput

- Need large transistors at peripheral circuits to source large current
- To pitch match, peripherals shared and time-multiplexed by multiple columns
- Limits throughput
**DYNAMIC VOLTAGE-MODE SENSING**

- Utilize charge/discharge of SL capacitance

\[
\Delta V = V_{pre} - V_{precharge}
\]

Senset when \(\Delta V\) is large

**Graph:**
- Red line: \(VSL^+ : \sum \Delta V_i = 200 \mu S\)
- Blue line: \(VSL^- : \sum \Delta V_i = 300 \mu S\)

**Legend:**
- **ΔV**
- Sense when **ΔV** is large

**Diagram:**
- Precharge
- Voltage Sense-amp
- Precharge

*Stanford University*

2015.04.15

H.-S. Philip Wong
DYNAMIC VOLTAGE-MODE SENSING

- Utilize charge/discharge of SL capacitance
- SLs discharge at different speeds

\[ \Delta V \]

\[ V_{pre} \]

\[ VSL^+ : \sum X_i G_i^+ = 200\mu S \]

\[ VSL^- : \sum X_i G_i^- = 300\mu S \]

Sense when \( \Delta V \) is large
DYNAMIC VOLTAGE-MODE SENSING

- Utilize charge/discharge of SL capacitance
- SLs discharge at different speeds
- Sense voltage difference of $SL^+$ and $SL^-$

$V_{pre}$

$V_{SL}^+$: $\sum X_i G_i^+ = 200\, \mu S$

$V_{SL}^-: \sum X_i G_i^- = 300\, \mu S$

Sense when $\Delta V$ is large
LIMITATION OF DYNAMIC VOLTAGE-SENSING

- $\Delta V_{SL}$ not linear w.r.t ideal MVM results $\sum_i X_i (G_i^+ - G_i^-)$
- $\Delta V_{SL}$ can be different even under the same ideal MVM results

$\Rightarrow$ Only able to sense 1-bit output (sign)

$$\Delta V_{SL}(t) = V_{pre}(e^{-\frac{t}{C_{SL}}} \sum_i X_i G_i^+ - e^{-\frac{t}{C_{SL}}} \sum_i X_i G_i^-)$$

- $\Delta V_{SL}$ not linear w.r.t ideal MVM results
- $\Delta V_{SL}$ can be different even under the same ideal MVM results

$\Rightarrow$ Only able to sense 1-bit output (sign)

$\Delta V_{SL}(t)$

- Same differential conductance
- Total activated conductance differs by 100$\mu$S
STATIC VOLTAGE-MODE SENSING

Weights mapped as differential conductance between RRAMs on adjacent rows
STATIC VOLTAGE-MODE SENSING

- Drive BL pairs to $V_{\text{ref}} \pm X_i \times V_{\text{read}}$

W. Wan et al., VLSI 2020
STATIC VOLTAGE-MODE SENSING

- Drive BL pairs to $V_{\text{ref}} \pm X_i \times V_{\text{read}}$
- Activate all WLs

W. Wan et al., VLSI 2020
**STATIC VOLTAGE-MODE SENSING**

- Drive BL pairs to $V_{\text{ref}} \pm X_i \times V_{\text{read}}$
- Activate all WLs
- Turn off WLs when SL voltage settles

---

- **Drive BL pairs to** $V_{\text{ref}} \pm X_i \times V_{\text{read}}$
- **Activate all WLs**
- **Turn off WLs when SL voltage settles**

---

*W. Wan et al., VLSI 2020*
**STATIC VOLTAGE-MODE SENSING**

- Drive BL pairs to $V_{ref} \pm X_i \times V_{read}$
- Activate all WLs
- Turn off WLs when SL voltage settles
- Charges that remain on SL capacitance are sampled and sensed

$V_{ref}$

$V_{ref}$

$V_{ref} + V_{read}$

$V_{ref} - V_{read}$

### Graphical Representation

- **Activate WLs**
- **Turn off WLs**

### Time (ns)

0.0  0.5  1.0  1.5  2.0  2.5  3.0

0.86  0.88  0.90
\[
V_{SL} = V_{read} \frac{\sum_i X_i (G^+_i - G^-_i)}{\sum_i (G^+_i + G^-_i)} + V_{ref}
\]

- Drive BL pairs to \(V_{ref} \pm X_i \times V_{read}\)
- Activate all WLs
- Turn off WLs when SL voltage settles
- Charges that remain on SL capacitance are sampled and sensed

Total conductance along a column
MULTI-BIT OUTPUT SENSING CAPABILITY
OVERCOMES THE LIMITATIONS OF DYNAMIC VOLTAGE-SENSING

- $V_{SL}$ linear w.r.t. ideal MVM (for columns having same total conductance)
- $V_{SL}$ stays the same under the same ideal MVM results

$$V_{SL} = V_{\text{read}} \frac{\sum_i X_i (G_i^+ - G_i^-)}{\sum_i (G_i^+ + G_i^-)} + V_{\text{ref}}$$

Total conductance along a column

![Graph showing 32 output levels (5-bit)]
ENERGY-EFFICIENCY BENEFIT

Array turned off as soon as SL voltage settles → Less energy consumption

* Includes only energy from RRAM array
**VOLTAGE-SENSING NEURON CIRCUIT**

- **Compact**: Single amplifier reused for CDS, integration & comparison
- **Reconfigurable**: activation functions: binary, ternary, ReLU, tanh
- **Energy-efficient**: amplifier can operate in sub-threshold regime

---

- **COMPACT**: Single amplifier reused for CDS, integration & comparison
- **RECONFIGURABLE**: activation functions: binary, ternary, ReLU, tanh
- **ENERGY-EFFICIENT**: amplifier can operate in sub-threshold regime

---

W. Wan *et al.*, ISSCC 2020, K. Rajkumar *et al.*, ICONS 2020
RECORD ENERGY-EFFICIENCY AMONG RRAM CIM CHIPS

(1 TMACS/W = 2 TOPS/W)

Precision: (input, weight, output)

Energy Efficiency (TMACS/W)

- 74.00 (1b, analog, 1b)
- 60.69 (1b, 2b, 6b)
- 39.20 (1b, ternary, 3b)
- 26.58 (1b, ternary, 3b)
- 0.89 (4b, analog, 13b)
- 10.35 (1b, analog, 1b)
- 33.25 (1b, analog, 1b)

ISSCC'20 130nm
ISSCC'20 22nm
ISSCC'20 130nm
ISSCC'19 55nm
Nat. El. 19 180nm
VLSI'18 180nm
VLSI'18 40nm
ENERGY-EFFICIENCY
Static voltage-mode sensing scheme

RECONFIGURABILITY
Dataflow reconfigurability realized by Transposable Neurosynaptic Core
VERSATILE DATAFLOW DIRECTION
NOT IMPLEMENTED BY CONVENTIONAL CIM ARCHITECTURES

- Existing CIM architectures mostly support only feed-forward neural networks
- Drivers and ADCs on different sides of synapse array
VERSATILE DATAFLOW INCURS OVERHEAD
FORWARD & BACKWARD MVM

1. Duplicate ADC/neurons on both sides of RRAM array
2. Route output from both sides to the same set of ADC/neurons

Forward & backward
e.g. RBM / back-propagation

Implementation 1

Implementation 2

GEN: Forward MVM
INF: Backward MVM

VERSATILE DATAFLOW INCURS OVERHEAD
FORWARD & BACKWARD MVM

1. Duplicate ADC/neurons on both sides of RRAM array
2. Route output from both sides to the same set of ADC/neurons

Forward & backward
e.g. RBM / back-propagation

Implementation 1

Implementation 2

GEN: Forward MVM
INF: Backward MVM
VERSATILE DATAFLOW INCURS OVERHEAD

RECURRENT MVM

Requires porting outputs from one side to the input of the other side

Recurrent e.g. RNN / LSTM

\[ W \]

Implementation

Weights

ADC / Neuron

Registers

Requires porting data

REC: Recurrent MVM
**TRANSPOSABLE NEUROSYNAPTIC CORE**

RECONFIGURABLE DATAFLOW W/ MINIMAL OVERHEAD

- **GEN**: Forward MVM
- **INF**: Backward MVM
- **REC**: Recurrent MVM

- No additional ADCs
- No extra routing outside array

W. Wan et al., ISSCC 2020
INTERLEAVED RRAM & CMOS NEURON

Conventional design: Neuron outside of RRAM array

W. Wan et al., ISSCC 2020
BI-DIRECTIONAL DUAL NEURON I/O PORTS

- **Dual**: Neuron in sub-core \((j, k)\) connects to BL \((16j+k)\) and SL \((16k+j)\)
- **Bi-directional**: Neuron uses its connecting BL & SL for both input & output

![Diagram of BI-DIRECTIONAL DUAL NEURON I/O PORTS](image-url)
RECONFIGURABLE DATAFLOW DIRECTION

<table>
<thead>
<tr>
<th>pulse input</th>
<th>neuron input switch</th>
<th>neuron output switch</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SL</td>
<td>BL</td>
<td>BL</td>
<td>Forward (SL-BL) MVM</td>
</tr>
<tr>
<td>BL</td>
<td>SL</td>
<td>SL</td>
<td>Backward (BL-SL) MVM</td>
</tr>
<tr>
<td>BL</td>
<td>SL</td>
<td>BL</td>
<td>BL Recurrent MVM</td>
</tr>
<tr>
<td>SL</td>
<td>BL</td>
<td>SL</td>
<td>SL Recurrent MVM</td>
</tr>
<tr>
<td>BL</td>
<td>BL</td>
<td></td>
<td>Directly driving neurons</td>
</tr>
<tr>
<td>SL</td>
<td>SL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LFSR</td>
<td>SL</td>
<td></td>
<td>Pseudo-random-noise injection into neurons</td>
</tr>
</tbody>
</table>

W. Wan et al., ISSCC 2020
RECONFIGURABLE DATAFLOW DIRECTION

<table>
<thead>
<tr>
<th>pulse input</th>
<th>neuron input switch</th>
<th>neuron output switch</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SL</td>
<td>BL</td>
<td>BL</td>
<td>Forward (SL-BL) MVM</td>
</tr>
<tr>
<td>BL</td>
<td>SL</td>
<td>SL</td>
<td>Backward (BL-SL) MVM</td>
</tr>
<tr>
<td>BL</td>
<td>SL</td>
<td>BL</td>
<td>BL Recurrent MVM</td>
</tr>
<tr>
<td>SL</td>
<td>BL</td>
<td>SL</td>
<td>SL Recurrent MVM</td>
</tr>
<tr>
<td>BL</td>
<td>BL</td>
<td></td>
<td>Directly driving neurons</td>
</tr>
<tr>
<td>SL</td>
<td>SL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LFSR</td>
<td>SL</td>
<td></td>
<td>Pseudo-random-noise injection into neurons</td>
</tr>
</tbody>
</table>

Input Stage

Output Stage

RECONFIGURABLE DATAFLOW DIRECTION

W. Wan et al., ISSCC 2020
# RECONFIGURABLE DATAFLOW DIRECTION

<table>
<thead>
<tr>
<th>pulse input</th>
<th>neuron input switch</th>
<th>neuron output switch</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SL</td>
<td>BL</td>
<td>BL</td>
<td>Forward (SL-BL) MVM</td>
</tr>
<tr>
<td>BL</td>
<td>SL</td>
<td>SL</td>
<td>Backward (BL-SL) MVM</td>
</tr>
<tr>
<td>BL</td>
<td>SL</td>
<td>BL</td>
<td>BL Recurrent MVM</td>
</tr>
<tr>
<td>SL</td>
<td>BL</td>
<td>SL</td>
<td>SL Recurrent MVM</td>
</tr>
<tr>
<td>BL</td>
<td>BL</td>
<td></td>
<td>Directly driving neurons</td>
</tr>
<tr>
<td>SL</td>
<td>SL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LFSR</td>
<td>SL</td>
<td></td>
<td>Pseudo-random-noise injection into neurons</td>
</tr>
</tbody>
</table>

**Legend:**
- LFSR: Linear Feedback Shift Register
- SL: Source Line
- BL: Bit Line
- MVM: Matrix Vector Multiplication
- RRAM: Resistive Random Access Memory

W. Wan et al., ISSCC 2020
DEMONSTRATE VERSATILE DATAFLOW
RESTRICTED BOLTZMANN MACHINE (RBM) FOR IMAGE RECOVERY

Inference: back-and-forth Gibbs sampling between visible & hidden neurons

Sample from gray-level image
Reconstruct w/ RBM
Average

225 Visible Neurons (15x15 pixels)

“good” pixels
Driven to ground-truth

“bad” pixels
Reconstruct by Gibbs sampling

60 Hidden Neurons

W. Wan et al., ISSCC 2020
MEASUREMENT SETUP

Measurement Board

Test Chip

FPGA Test Chip

Measurement Board
Click on the image to corrupt pixels!

- New Image
- Clear
- Recover

# Gibbs Steps: 6
# Samples: 10

W. Wan et al., ISSCC 2020
ISSCC & VLSI Chip Demo Video: https://youtu.be/b7ITxmfaLBk
IMAGE RECOVERY PERFORMANCE

Reconstruction Error after 2 Gibbs Steps

- Corrupted Image: 2.45
- Software (64b-float): 1.81
- Software (7-level quantized): 1.89
- Our Chip: 1.91

W. Wan et al., ISSCC 2020
BRAIN-INSPIRED HYPER-DIMENSIONAL (HD) COMPUTING

- Data representation, learning, inference: all in HD space
- Language, visual, bio-signal processing, robotics/control

\[
\begin{bmatrix}
1 & 1 & 0 & \ldots & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & \ldots & 1 & 1 & 0 & 1
\end{bmatrix}
\]

Source: A. Mitrokhin, et al., Science Robotics, 2019 (University of Maryland)

Source: Prof. Jan Rabaey, UC Berkeley

INSIDE THE BOX
MULTIPLY-ACCUMULATE-PERMUTE (MAP)

Multiply: bitwise XOR

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Accumulate

<table>
<thead>
<tr>
<th>1</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Permute

<table>
<thead>
<tr>
<th>1</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
HD COMPUTING: MAPPING PHASE

Length: kilobits

Random projection
- Alphabet $\rightarrow$ 26 HD vectors

High dimensionality: $\sim$kbits
- Resilient to errors

High-dimensional (HD) vectors
HD COMPUTING: LEARNING PHASE

Training text: English
HD COMPUTING: LEARNING PHASE

MAP Kernel

Multiply: Bitwise XOR
Accumulation
Permute: Circular Shift

Store in Associative Memory

Training text: English
HD Vector: entire English language
HD COMPUTING: LEARNING PHASE

Training text: Spanish
HD Vector: entire Spanish language

MAP Kernel
- Multiply: Bitwise XOR
- Accumulation
- Permute: Circular Shift

Store in Associative Memory
HD COMPUTING: INFERENCe PHASE

Test sentence: infer language

MAP Kernel

Multiply: Bitwise XOR
Accumulation
Permute: Circular Shift

HD Vector for English
HD Vector for Spanish

Associative search/compare
**UTILITYING RRAM STOCHASTICITY**

GENERATE HD VECTORS DIRECTLY IN MEMORY

---

High resistance state: 0

Low resistance state: 1

\[ V_{TE} < V_{SET} \]

\[ P_{SET} < 1 \]

\[ 1001100111......010011101 \]

(HD vector)

\[ P_{SET}: \text{SET probability (switching from} \ '0' \ \text{to} \ '1') \]

---

**Exp. data**

**Tunable probabilities**

**SET Probability**

**Pulse Amplitude (V)**

**Pulse Width (ns)**

---

H. Li, ... H.-S. P. Wong, *IEDM, 2016*
MAP KERNELS WITHIN 3D VERTICAL RRAM

- Exploit 3D vertical structure for native MAP kernels in memory
- Extensive measurements up to $10^{12}$ cycles for reliable operations

Essential HD kernels experimentally realized within 3D VRRAM
MULTIPLY: XOR LOOKUPS

![Image of RRAM cell with Pillar electrode, Plane electrode, Word line (WL), Bit line (BL), Select line (SL), and RRAM cell.]

XOR/XNOR table

<table>
<thead>
<tr>
<th>LUT Evaluation Cycle (#)</th>
<th>Resistance (Ω)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10^3</td>
<td>10</td>
</tr>
<tr>
<td>10^6</td>
<td>100</td>
</tr>
<tr>
<td>10^9</td>
<td>10k</td>
</tr>
<tr>
<td>10^12</td>
<td>1M</td>
</tr>
</tbody>
</table>

MULTIPLY: XOR LOOKUPS

![Image of XOR/XNOR table with entries for different combinations of A, B, C, and D, showing the resistance values for each combination.]
ACCUMULATE: CURRENT SUMMING

“Physically” adding up ‘0’s and ‘1’s stored along a vertical pillar

Exp. data

Current (µA)

Addition Cycle (#)
PERMUTE: IN-MEMORY BIT COPY/SHIFT

Measured LRS (~10kΩ)  
Measured HRS (400kΩ-1MΩ)

L4  L3  L2  L1
0  0  0  1
0  gnd 1  0
0  gnd 1  0
1  VDD 0  0

Bit 1 up

L4  L3  L2  L1
0  0  0  1
0  gnd 1  0
0  gnd 1  0
1  VDD 0  0

L4  L3  L2  L1
0  VDD 1  1
0  VDD 1  1
0  VDD 1  1
1  1 1 1

Bit 0 down

L4  L3  L2  L1
0  VDD 1  1
0  VDD 1  1
0  VDD 1  1
1  1 1 1

L4  L3  L2  L1
0  VDD 1  1
0  VDD 1  1
0  VDD 1  1
1  1 1 1

(a) Measured LRS (~10kΩ)  
Measured HRS (400kΩ-1MΩ)

L4  L3  L2  L1
0  gnd 1  0
0  gnd 1  0
0  gnd 1  0
1  VDD 0  0

Bit 1

L4  L3  L2  L1
0  gnd 1  0
0  gnd 1  0
0  gnd 1  0
1  VDD 0  0

L4  L3  L2  L1
0  VDD 1  1
0  VDD 1  1
0  VDD 1  1
1  1 1 1

(b) Measured LRS (~10kΩ)  
Measured HRS (400kΩ-1MΩ)

L4  L3  L2  L1
0  VDD 1  1
0  VDD 1  1
0  VDD 1  1
1  1 1 1

L4  L3  L2  L1
0  VDD 1  1
0  VDD 1  1
0  VDD 1  1
1  1 1 1
CHECKERBOARD: 64 VERTICAL PILLARS

### Measured XOR results
**Digits:** XOR inputs

<table>
<thead>
<tr>
<th>0</th>
<th>Measured HRS (500kΩ-1.5MΩ)</th>
<th>1</th>
<th>Measured LRS (9-14kΩ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>00</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>10</td>
<td>00</td>
<td>01</td>
</tr>
<tr>
<td>01</td>
<td>11</td>
<td>10</td>
<td>00</td>
</tr>
<tr>
<td>00</td>
<td>01</td>
<td>11</td>
<td>10</td>
</tr>
</tbody>
</table>

### Measured summed current
**Digits:** 4-bit vector inputs

<table>
<thead>
<tr>
<th>0</th>
<th>Measured summing current (mA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0001</td>
</tr>
<tr>
<td>0001</td>
<td>0011</td>
</tr>
<tr>
<td>0011</td>
<td>0111</td>
</tr>
<tr>
<td>0111</td>
<td>1111</td>
</tr>
<tr>
<td>1111</td>
<td>1110</td>
</tr>
<tr>
<td>1110</td>
<td>1100</td>
</tr>
<tr>
<td>1100</td>
<td>1000</td>
</tr>
<tr>
<td>1000</td>
<td>0000</td>
</tr>
</tbody>
</table>

### Measured permuted states
**Digits:** initial states

<table>
<thead>
<tr>
<th>0</th>
<th>Measured HRS (400kΩ-1.5MΩ)</th>
<th>1</th>
<th>Measured LRS (9-14kΩ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>L4</td>
<td>L1</td>
<td>L1</td>
<td>L1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**H.-S. Philip Wong**

Stanford University
END-TO-END SYSTEM-TECHNOLOGY CO-OPTIMIZATION

Machine Learning SoC

On-chip Memory

CIM accelerators

Digital accelerators

Divide workloads

System integration: combine the best characteristics
THE FUTURE IS SYSTEM INTEGRATION

High Density On-Chip NVM
(CIM accelerators + digital storage)

Dense Connectivity
(Nanometer Scale Inter-Level Vias)

Si Logic Die (digital accelerators)

Energy-Efficient Peripheral Logic
(memory access, CIM)

High-Speed On-Chip NVM
(accelerator scratchpads)

N3XT graphic: S. Mitra, H.-S. P. Wong (Stanford)
Students

Weier Wan

Haitong Li
Collaborators

Subhasish Mitra

Max Shulaker

Mohamed S. Aly
Colleagues at TSMC

TSMC R&D Center, Hsinchu, Taiwan
(Under construction)
Collaborators

Gert Cauwenberghs  Huaqiang Wu  Bin Gao  Priyanka Raina
Sponsors

Stanford SystemX Alliance

Non-Volatile Memory Technology Research Initiative (NMTRI)
End of Talk

Questions?