eDRAM-CIM: Compute-In-Memory Design with Reconfigurable Embedded Dynamic Memory Array Realizing Adaptive Data Converters and Charge Domain Computing

Shanshan Xie\textsuperscript{1}, Can Ni\textsuperscript{1}, Aseem Sayal\textsuperscript{1}, Pulkit Jain\textsuperscript{2}, Fatih Hamzaoglu\textsuperscript{2}, Jaydeep P. Kulkarni\textsuperscript{1}

\textsuperscript{1}University of Texas at Austin, Austin, TX
\textsuperscript{2}Intel, Hillsboro, OR
Self Introduction

Shanshan Xie

Email: sxie@utexas.edu
Website: http://cyanxie.com/

Education

• B.S. 2018 Worcester Polytechnic Institute
• Ph.D. student at University of Texas at Austin since 2018
  • Circuit Research Lab: https://sites.utexas.edu/CRL/
  • Advisor: Professor Jaydeep P. Kulkarni

Experience

• Analog Design Intern, Texas Instruments, 2019
• Application Engineering Intern, Analog Devices Inc., 2018
• Test Engineering Intern, Analog Devices Inc., 2017
• Product Engineering Intern, Analog Devices Inc., 2016

Research Area

• Mixed signal approaches for ML accelerators
• Compute-in-memory techniques
Outline

- Compute-in-memory: motivation and challenges
- Proposed eDRAM-CIM macro
  - Overall architecture
  - Digital-to-analog conversion (DAC)
  - Multiply-accumulate-averaging (MAV) computation
  - Analog-to-digital conversion (ADC)
- Chip measurement and classification accuracy results
- Summary
Outline

- **Compute-in-memory: motivation and challenges**
- **Proposed eDRAM-CIM macro**
  - Overall architecture
  - Digital-to-analog conversion (DAC)
  - Multiply-accumulate-averaging (MAV) computation
  - Analog-to-digital conversion (ADC)
- **Chip measurement and classification accuracy results**
- **Summary**
Deep Neural Network Applications

AI High Performance Data Center

AI Cloud Computing

Edge Computing Face Detection

Edge Computing Object Detection
DNN Model Size Challenges

<table>
<thead>
<tr>
<th>Total Number of MACs</th>
<th>Energy Efficiency Challenges</th>
</tr>
</thead>
<tbody>
<tr>
<td>341k LeNet-5</td>
<td>① Data Intensive</td>
</tr>
<tr>
<td>724M AlexNet</td>
<td>② Huge Data Movement</td>
</tr>
<tr>
<td>2.8G OverFeat</td>
<td>③ Data Transfer Latency</td>
</tr>
<tr>
<td>3.9G ResNet50</td>
<td>④ Computation Energy Cost</td>
</tr>
<tr>
<td>1.43G GoogleNet v1</td>
<td>⑤ Memory Access Energy Cost</td>
</tr>
<tr>
<td>15.5G VGG16</td>
<td></td>
</tr>
</tbody>
</table>

Ref: Vivienne Sze, Proceedings of the IEEE 105.12 2017
## eDRAM-CIM Motivation

<table>
<thead>
<tr>
<th>Operation</th>
<th>Energy (pJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Computation Energy Cost</strong></td>
<td></td>
</tr>
<tr>
<td>Integer Add (32b)</td>
<td>0.1</td>
</tr>
<tr>
<td>Integer Multiply (32b)</td>
<td>3.1</td>
</tr>
<tr>
<td>Floating Point Add (32b)</td>
<td>0.9</td>
</tr>
<tr>
<td>Floating Point Multiply (32b)</td>
<td>3.7</td>
</tr>
<tr>
<td><strong>Memory Access Energy Cost</strong></td>
<td></td>
</tr>
<tr>
<td>8KB SRAM (64b)</td>
<td>10</td>
</tr>
<tr>
<td>1MB SRAM (64b)</td>
<td>100</td>
</tr>
<tr>
<td>DRAM</td>
<td>2000</td>
</tr>
</tbody>
</table>

DRAM memory access is expensive: ~650X higher than computation energy cost

Ref: Mark Horowitz, ISSCC 2014
eDRAM Advantages for CIM

- Highest bitcell density
- High access speed
- Low read/write energy
- High write endurance
- High bandwidth

Desired attributes for CIM designs

Ref:
Gregory Fredeman, JSSC 2015;
Fatih Hamzaoglu, JSSC 2014;
Nasser Kurd, JSSC 2015
Our Approach: eDRAM based CIM

- Directly perform computation inside embedded DRAM (eDRAM)
- Reconfigure existing 1T1C eDRAM columns as charge domain computing units
- Reduce off-chip memory access and latency
Outline

- Compute-in-memory: motivation and challenges
- Proposed eDRAM-CIM macro
  - Overall architecture
  - Digital-to-analog conversion (DAC)
  - Multiply-accumulate-averaging (MAV) computation
  - Analog-to-digital conversion (ADC)
- Chip measurement and classification accuracy results
- Summary
eDRAM CIM Overall Architecture: DAC

1. Load Data to eDRAM Bitcell

2. Generate Analog Values Va & Va_bar

Load Input to DAC eDRAM array

*DAC: Digital-to-Analog Converter
eDRAM CIM Overall Architecture: MAV

*DAC: Digital-to-Analog Converter
*MAV: Multiply-Accumulation-Average

3. Load Weights to MAV Compute Array

4. Generate $V_{MAV}$
eDRAM CIM Overall Architecture: Partial Sum

*DAC: Digital-to-Analog Converter
*MAV: Multiply-Accumulation-Average

\[ \frac{1}{N} \sum_{i=1}^{N} V_{a,i} \times w_i \]

Input Data → Analog Output Conversion → MAV Operation → Avg. Pooling & ReLU Activation

DAC Control Logic
DAC Control Logic

DACPOS DACNEG

In-eDRAM MAV Compute Array & Averaging Circuit

Weight Array

V_{a, bar} → MAV Operation

\( V_a \) → Store Partial Sum

\( V_{MAV} \)
eDRAM CIM Overall Architecture: Partial Sum

*DAC: Digital-to-Analog Converter
*MAV: Multiply-Accumulation-Average
eDRAM CIM Overall Architecture: ADC

*DAC: Digital-to-Analog Converter
*MAV: Multiply-Accumulation-Average
*ADC: Analog-to-Digital Converter
16.2: eDRAM CIM: Compute-In-Memory Design with Reconfigurable Embedded Dynamic Memory Array
Realizing Adaptive Data Converters and Charge Domain Computing

eDRAM CIM Overall Architecture

DAC: Digital-to-Analog Converter
MAV: Multiply-Accumulation-Average
ADC: Analog-to-Digital Converter

1. Load Data to eDRAM Bitcell
2. Generate Analog Values Va & Va_bar
3. Load Weights to MAV Compute Array
4. Generate V_{MAV}
5. Generate V_{REF}
6. Final Y_{OUT}

Input Data → Analog Output Conversion

V_a & V_{a_bar} → MAV Operation

\[ \frac{1}{N} \sum_{i=1}^{N} V_{a,i} \times W_i \]

N = # dot products per MAV

Avg. Pooling & ReLU Activation

Digital Output Conversion

Y_{OUT}
In-eDRAM Differential Digital-to-Analog Converter (DAC)
In-eDRAM DAC – Positive DAC

\[ V_a = V_{DD} \times \frac{16 \times 1 + 8 \times X_3 + 4 \times X_2 + 2 \times X_1 + 1 \times X_0}{32} \]

© 2021 IEEE
International Solid-State Circuits Conference

16.2: eDRAM-CIM: Compute-In-Memory Design with Reconfigurable Embedded Dynamic Memory Array
Realizing Adaptive Data Converters and Charge Domain Computing
In-eDRAM DAC – Negative DAC

\[ V_{a\_bar} = V_{DD} \times \frac{8 \times X_3 + 4 \times X_2 + 2 \times X_1 + 1 \times X_0 + 1}{32} \]
Differential Analog Voltages, $V_a$ and $V_{a\_bar}$

\[
\begin{align*}
V_{\text{DD}} & \quad \text{Positive DAC} \\
V_a & \quad \text{Zero Reference}=V_{\text{DD}}/2 \\
V_{a\_bar} & \quad \text{Negative DAC} \\
V_{\text{SS}}
\end{align*}
\]

- Prepare differential analog voltages ($V_a$ and $V_{a\_bar}$) for sign computation
- Positive DAC: $V_a$; Negative DAC: $V_{a\_bar}$
- Depends on sign of the weight, choose either $V_a$ or $V_{a\_bar}$
Positive DAC Demonstration
8b In-eDRAM DAC Dataflow – Step 1

Step 1: Load lower 4-bit to DAC

Load data on to memory array using eDRAM peripheral circuits (not shown in the diagram)

Turn on twice to make sure 16 capacitors are fully charged

Floating

Input Data

Constant ‘1’

Constant ‘0’
8b In-eDRAM DAC Dataflow – Step 2

Step 2: Charge Share

Data Flow

Step 1

Voltage is sampled on Cx1 capacitor

Step 2:

Activate all wordlines for charge share to convert digital data to analog voltage

V<sub>x16</sub>, V<sub>x1</sub>

V<sub>a</sub>
8b In-eDRAM DAC Dataflow – Step 3

Step 3: Load higher 4-bit to DAC

Constant ‘1’

Input Data

Constant ‘0’

W<5>
W<4>
W<3>
W<2>
W<1>
W<0>
Wlx16
Wlx1
DAC_CS

Vx16, Vx1

V_a/V_a_bar
8b In-eDRAM DAC Dataflow – Step 4

**Step 4: Charge Share**

**Data Flow**

Voltage is sampled on $C_{x16}$ capacitor.

- $V_{x16}$, $V_{x1}$
- $V_a / V_{a_{bar}}$
- $V_a$

**Step 1**

**Step 2**

**Step 3**

**Step 4**
8b In-eDRAM DAC Dataflow – Step 5

Step 5: Generate $V_a$ and $V_{a\_bar}$

Data Flow

- **Step 1**: Generate $V_{x16}$
- **Step 2**: Generate $V_{x1}$
- **Step 3**: Charge share to generate final analog voltage
- **Step 4**: $V_a$ generated
- **Step 5**: $V_{a\_bar}$ generated

$V_{x16}$, $V_{x1}$

DAC_CS

Charge share to generate final analog voltage
1T1C In-eDRAM Differential DAC Demonstration

- Signals match the expected waveforms
- $V_a$ and $V_{a\_bar}$ are symmetrical around $V_{DD}/2$ (600mV)
- Higher 4 bits are loaded last: minimize capacitor leakage on final charge share value

$V_{DD}/2 = 600mV$

$\text{Load 4b LSB} \quad \text{Load 4b MSB} \quad \text{CS} \quad \text{Generate} \quad V_a & V_{a\_bar}$

$\text{DAC\_CS}$

$\text{WL}<0>$

$\text{WL}<5>$

$\text{WL}<16>$

$\text{CS}$
In-eDRAM DAC Characterization: Measured Results

- Systematic deviation: positive DAC saturate at high value (can be correct during training)
In-eDRAM DAC Characterization: Zoomed In

- Good linearity, DNL<1LSB
In-eDRAM Multiply-Accumulate-Averaging (MAV) Computation
eDRAM-CIM: MAV Computation

Weights are stored in 1T1C eDRAM bitcell: $W_7$ (sign bit); $W_6$-$W_0$ (magnitude bits)

Analog Voltage from DAC
eDRAM-CIM: MAV Computation

Data Flow

$W_7$ (sign bit) is used to select between $V_a$ and $V_{a\_bar}$.

Voltage from DAC

Zero reference

Weight magnitude ($W_0 \sim W_6$) are used as MUX control signals.

Voltage from DAC

Zero reference

$V_{DD}/2$
eDRAM-CIM: MAV Computation

Data Flow

Reuse 1T1C eDRAM bitcell

Voltages are sampled on binary scaled 1T1C eDRAM capacitors

2:1 MUX: analog dot-product unit
**eDRAM-CIM: MAV Computation**

- Switches ($S_0$~$S_6$) are closed to charge share between all the 1T1C eDRAM cells
- $V_{MAV}$ = final MAV computation result
ReLU

Zero Reference $V_{DD}/2$

Final MAV Analog Results

Comparator

$V_{MAV} > V_{DD}/2$ → $V_{MAV}$

$V_{MAV} < V_{DD}/2$ → $V_{DD}/2$

• Comparator compares VMAV and zero reference voltage VDD/2
• If $V_{MAV} > V_{DD}/2$ → output $V_{MAV}$
• If $V_{MAV} < V_{DD}/2$ → output $V_{DD}/2$
In-eDRAM Analog-to-Digital Converter (ADC)
eDRAM-CIM: ADC Operation

V_{REF} Generation Waveforms for 1\textsuperscript{st} (b=1) conversion cycle

- Step 1: Charge corresponding bitcells
eDRAM-CIM: ADC Operation

**V_{REF} Generation Waveforms for 1^{st} (b=1) conversion cycle**

- Step 1: Charge corresponding bitcells
- Step 2: IDLE
eDRAM-CIM: ADC Operation

**V_{REF} Generation Waveforms for 1^{st} (b=1) conversion cycle**

- Step 1: Charge corresponding bitcells
- Step 2: IDLE
- Step 3: Charge Share to generate reference voltages

- \( V_{MAV} > V_{REF1} \)
- \( V_{MAV} < V_{REF2} \)
**eDRAM-CIM: ADC Operation**

- **V_{REF} Generation Waveforms for 1^{st} (b=1) conversion cycle**

- **V_{MAV}**
  - V_{REF1}
  - V_{REF2}
  - V_{REF3}

- **Y_{OUT}[7]**
- **SA2_OUT**
- **SA3_OUT**
- **Y_{OUT}[6]**

- **SA**
- \( b = 1,3,5,7 \)

- **Steps**
  - **Step 1**: Charge corresponding bitcells
  - **Step 2**: IDLE
  - **Step 3**: Charge Share to generate reference voltages
  - **Step 4**: Discharge and reset 1T1C eDRAM capacitors
2b/Conversion In-eDRAM ADC Waveforms

- 2b/conversion cycle
- 2 extra eDRAM columns
- 3 reference voltages ($V_{REF1}$~$V_{REF3}$)
- Minimize $V_{MAV}$ capacitor leakage during conversion
eDRAM-CIM: Adaptive ADC Motivation

Convolution Outputs (CiFAR-10 dataset Layer 1)

X = Clipping Threshold

\[ X = \text{Clipping Threshold} \]

\[ \mu = -0.07 \]

\[ \sigma = 25.7 \]

Infrequent appearing MAV computation outputs \( V_{\text{MAV}} \) are trimmed to mitigate ADC energy and latency.
Clipping In-eDRAM ADC Data Flow

- Step 1: Charge corresponding bitcells
- Step 2: IDLE
Clipping In-eDRAM ADC Data Flow

- Step 1: Charge corresponding bitcells
- Step 2: IDLE
- Step 3: Charge share to generate reference voltages
- Step 4: Compare $V_{\text{MAV}} > V_{\text{HB}}$ or $V_{\text{MAV}} < V_{\text{LB}}$? → Yes
- Step 5: Skip subsequent conversion & discharge BLs

$V_{\text{REF2}} = \frac{3}{4} V_{\text{DD}}$  
$V_{\text{REF3}} = \frac{1}{4} V_{\text{DD}}$

$V_{\text{REF3}} = V_{\text{LB}}$

$V_{\text{MAV}} > V_{\text{HB}}$

ADC_DONE

$V_{\text{REF2}} = V_{\text{HB}}$

$V_{\text{REF3}} = V_{\text{LB}}$

SA$_2$ _OUT

SA$_3$ _OUT

$V_{\text{OUT}}$
Clipping ADC Simulation Waveforms

- Skip subsequent ADC conversion cycles if $V_{MAV}$ is higher than $V_{HB}$ or lower than $V_{LB}$
- If $V_{MAV}$ is in conversion range, in-eDRAM ADC will perform regular SAR conversion
Clipping Threshold vs Accuracy and Energy

Optimum clipping threshold = 60%

*ADC with clipping energy/without clipping energy
Clipping vs Non-Clipping ADC Tradeoff

- Throughput is improved by 1.14X while incurring 3.79% (1.21%) drop in Top-1 (Top-5) in CIFAR-10.
In-eDRAM ADC Measurement Results

- Linear for both clipping and non-clipping in-eDRAM ADC
Outline

- Compute-in-memory: motivation and challenges
- Proposed eDRAM-CIM macro
  - Overall architecture
  - Digital-to-analog conversion (DAC)
  - Multiply-accumulate-averaging (MAV) computation
  - Analog-to-digital conversion (ADC)
- Chip measurement and classification accuracy results
- Summary
Chip Micrograph

- 65nm native CMOS process
- Supply Voltage: 1~1.2V
- Input Clock Frequency: 50~200MHz
- Bitcell Size: 22.08um²
- Macro Area: 0.57mm²
- eDRAM Capacitor Type: Metal-Oxide-Metal (MOM)
Test-chip Area Breakdown

Unit: mm²

0.61

0.06

Weight array

0.12

DAC array

0.22

Weight array

0.08

MAV computation array

0.09

ADC test structure

DAC

Weight array

scanchain1

scanchain2

Main ADC

MAV computation array

MAV test structure

Input image array

eDRAM

ADC array

DAC array

Non-Macro:
scan-chains & test structures

eDRAM-CIM
Macro Area

0.669mm

0.669mm

1.759mm

© 2021 IEEE International Solid-State Circuits Conference

16.2: eDRAM-CIM: Compute-In-Memory Design with Reconfigurable Embedded Dynamic Memory Array
Realizing Adaptive Data Converters and Charge Domain Computing

© 2021 IEEE International Solid-State Circuits Conference
CIFAR-10 Classification Details

- 4 convolution layers, 2 average pooling layers, and 2 fully connected layers
- Input precision: 8 bits
- Weight precision: signed 8 bits
- Output precision: 8 bits
Classification Accuracy on CIFAR-10 Dataset

- Measured Top-5 accuracy is dropped 0.3% with non-clipping ADC and dropped 1.51% with clipping ADC
- Comparing with 8b software accuracy, eDRAM-CIM preserved good accuracy results
## Comparison with Prior Works

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Technology</strong></td>
<td>65nm</td>
<td>28nm</td>
<td>28nm</td>
<td>22nm</td>
<td>65nm</td>
</tr>
<tr>
<td><strong>Memory Cell Structure</strong></td>
<td>1T1C eDRAM</td>
<td>6T SRAM</td>
<td>6T + Local Computing SRAM</td>
<td>1T1R SLC ReRAM</td>
<td>6T SRAM</td>
</tr>
<tr>
<td><strong>Array Size</strong></td>
<td>16Kb</td>
<td>64Kb</td>
<td>64Kb</td>
<td>2Mb</td>
<td>128Kb</td>
</tr>
<tr>
<td><strong>Input Precision (bit)</strong></td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td><strong>Weight Precision (bit)</strong></td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td><strong>Supply Voltage (V)</strong></td>
<td>1~1.2</td>
<td>0.85~1.0</td>
<td>0.7~0.9</td>
<td>0.8</td>
<td>1</td>
</tr>
<tr>
<td><strong>Dataset</strong></td>
<td>CIFAR-10</td>
<td>CIFAR-10</td>
<td>MNIST</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Model</strong></td>
<td>CNN: 4 CONV + 2 Pooling + 2 FC</td>
<td>CNN: ResNet-20</td>
<td>CNN: ResNet-20</td>
<td>N/A</td>
<td>SVM</td>
</tr>
</tbody>
</table>

1 measured at 1.1V  
2 Scaled to 65nm, assume energy $\propto$ (Tech.)$^2$ [7]  
3 Limited by clocking infrastructure, chip size, technology and bit cell area  
4 $\text{FoM} = \text{input precision} \times \text{weight precision} \times \text{energy efficiency}$ (scaled to 65nm)  
5 Top-1 or Top-5 is not mentioned.
## Comparison with Prior Works

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Technology</strong></td>
<td>65nm</td>
<td>28nm</td>
<td>28nm</td>
<td>22nm</td>
<td>65nm</td>
</tr>
<tr>
<td><strong>Memory Cell Structure</strong></td>
<td>1T1C eDRAM</td>
<td>6T SRAM</td>
<td>6T + Local Computing SRAM</td>
<td>1T1R SLC ReRAM</td>
<td>6T SRAM</td>
</tr>
<tr>
<td><strong>Array Size</strong></td>
<td>16Kb</td>
<td>64Kb</td>
<td>64Kb</td>
<td>2Mb</td>
<td>128Kb</td>
</tr>
<tr>
<td><strong>Input Precision (bit)</strong></td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td><strong>Weight Precision (bit)</strong></td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td><strong>Supply Voltage (V)</strong></td>
<td>1~1.2</td>
<td>0.85~1.0</td>
<td>0.7~0.9</td>
<td>0.8</td>
<td>1</td>
</tr>
<tr>
<td><strong>Dataset</strong></td>
<td>CIFAR-10</td>
<td>CIFAR-10</td>
<td>MNIST</td>
<td>N/A</td>
<td>SVM</td>
</tr>
<tr>
<td><strong>Model</strong></td>
<td>CNN: 4 CONV + 2 Pooling + 2 FC</td>
<td>CNN: ResNet-20</td>
<td>CNN: ResNet-20</td>
<td>N/A</td>
<td>SVM</td>
</tr>
<tr>
<td><strong>Measured Accuracy</strong></td>
<td>80.1% (Top-1), 98.1 % (Top-5)</td>
<td>91.91%</td>
<td>92.02%</td>
<td>N/A</td>
<td>83.27%</td>
</tr>
</tbody>
</table>

---

1 measured at 1.1V  
2 Scaled to 65nm, assume energy $\propto$ (Tech.)  
3 Limited by clocking infrastructure, chip size, technology and bit cell area  
4 FoM = input precision x weight precision x energy efficiency (scaled to 65nm)  
5 Top-1 or Top-5 is not mentioned
# Comparison with Prior Works

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Technology</strong></td>
<td>65nm</td>
<td>28nm</td>
<td>28nm</td>
<td>22nm</td>
<td>65nm</td>
</tr>
<tr>
<td><strong>Memory Cell Structure</strong></td>
<td>1T1C eDRAM</td>
<td>6T SRAM</td>
<td>6T + Local Computing SRAM</td>
<td>1T1R SLC ReRAM</td>
<td>6T SRAM</td>
</tr>
<tr>
<td><strong>Array Size</strong></td>
<td>16Kb</td>
<td>64Kb</td>
<td>64Kb</td>
<td>2Mb</td>
<td>128Kb</td>
</tr>
<tr>
<td><strong>Input Precision (bit)</strong></td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td><strong>Weight Precision (bit)</strong></td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td><strong>Supply Voltage (V)</strong></td>
<td>1~1.2</td>
<td>0.85~1.0</td>
<td>0.7~0.9</td>
<td>0.8</td>
<td>1</td>
</tr>
<tr>
<td><strong>Dataset</strong></td>
<td>CIFAR-10</td>
<td>CIFAR-10</td>
<td>MNIST</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Model</strong></td>
<td>CNN: 4 CONV + 2 Pooling + 2 FC</td>
<td>CNN: ResNet-20</td>
<td>CNN: ResNet-20</td>
<td>N/A</td>
<td>SVM</td>
</tr>
<tr>
<td><strong>Measured Accuracy</strong></td>
<td>80.1% (Top-1), 98.1 % (Top-5)</td>
<td>91.91%</td>
<td>92.02%</td>
<td>N/A</td>
<td>83.27%</td>
</tr>
<tr>
<td><strong>Throughput (GOPS)</strong></td>
<td>1.34.71</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>4</td>
</tr>
<tr>
<td><strong>Average Energy Efficiency (TOPS/W)</strong></td>
<td>14.76</td>
<td>7.3 (1.35)</td>
<td>14.08 (2.61)</td>
<td>28.93 (3.31)</td>
<td>3.125</td>
</tr>
<tr>
<td><strong>GOPS/mm²</strong></td>
<td>8.26</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>2.78</td>
</tr>
<tr>
<td><strong>4FoM</strong></td>
<td>304.6</td>
<td>86.4</td>
<td>167</td>
<td>53</td>
<td>201.6</td>
</tr>
</tbody>
</table>

1. Measured at 1.1V  
2. Scaled to 65nm, assume energy $\propto$ (Tech.)² [7]  
3. Limited by clocking infrastructure, chip size, technology and bit cell area  
4. $4\text{FoM} = \text{input precision } \times \text{weight precision } \times \text{energy efficiency}$ (scaled to 65nm)  
5. Top-1 or Top-5 is not mentioned
## Test-chip Summary

<table>
<thead>
<tr>
<th></th>
<th>Test-chip Summary</th>
<th>Advanced Technology Node Estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technology</td>
<td>65nm CMOS</td>
<td>22nm tri-gate CMOS</td>
</tr>
<tr>
<td>Total eDRAM Size</td>
<td>16Kb</td>
<td>38.4MB</td>
</tr>
<tr>
<td>Bitcell Size (µm²)</td>
<td>22.08</td>
<td>0.029</td>
</tr>
<tr>
<td>Unit eDRAM Capacitor Size (fF)</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>eDRAM Capacitor Type</td>
<td>Metal-oxide-metal (MOM)</td>
<td>Capacitor-over-bitline (COB)</td>
</tr>
<tr>
<td>Input Clock Frequency</td>
<td>50 MHz (measured)</td>
<td>1.6GHz</td>
</tr>
<tr>
<td></td>
<td>200MHz (simulation)</td>
<td></td>
</tr>
<tr>
<td>GOPS/mm²</td>
<td>8.26</td>
<td>1305</td>
</tr>
<tr>
<td>MAV Operation Latency (ns)</td>
<td>10 (2 cycles)</td>
<td>1.25 (2 Cycles)</td>
</tr>
<tr>
<td>¹Throughput @ 1.1V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>w/o ADC Clipping</td>
<td>4.037 GOPS</td>
<td>19.5 TOPS</td>
</tr>
<tr>
<td>with 60% ADC Clipping</td>
<td>4.71 GOPS</td>
<td>22.2 TOPS</td>
</tr>
<tr>
<td>³Average Energy Efficiency (TOPS/W) @ 1.1V</td>
<td>4.76</td>
<td>11.12</td>
</tr>
<tr>
<td>Energy/MAV (fJ/MAV) with 60% ADC Clipping @ 1.1V</td>
<td>30.6</td>
<td>⁴0.21</td>
</tr>
</tbody>
</table>

¹Assume 1MAV = 2 operations
²Assume 30% utilization of 128MB (77mm²) [6] eDRAM memory for CIM usage
³Excluded scan-chains power
⁴Increased parallelism in large-capacity eDRAM array

© 2021 IEEE
International Solid-State Circuits Conference

16.2: eDRAM-CIM: Compute-In-Memory Design with Reconfigurable Embedded Dynamic Memory Array
Realizing Adaptive Data Converters and Charge Domain Computing

57 of 60
Outline

- Compute-in-memory: motivation and challenges
- Proposed eDRAM-CIM macro
  - Overall architecture
  - Digital-to-analog conversion (DAC)
  - Multiply-accumulate-averaging (MAV) computation
  - Analog-to-digital conversion (ADC)
- Chip measurement and classification accuracy results
- Summary
Summary

• 1T1C eDRAM bit cell: good candidate for compute-in-memory design
• Repurposing existing 1T1C eDRAM bitcell
• Intrinsic charge share operation in eDRAM bitcell
• Support 8b input and 8b signed/unsigned weight for MAV operations
• Measured Top-5 (Top-1) peak accuracy 98.1% (80.1%) on CIFAR-10
• Measured average energy efficiency 4.76 TOPS/W
• Measured throughput 4.71 GOPS
• FoM metrics can exceed significantly in an advanced eDRAM technology
• Potential for scalability to advanced eDRAM technology node
Acknowledgements

• Clifford Ong for technical discussions
• Intel for funding support

Thank you for your attention!