An Energy-efficient Floating-Point DNN Processor using Heterogeneous Computing Architecture with Exponent-Computing-in-Memory

Juhyoung Lee, Juhyoung Lee, Jihoon Kim, Wooyoung Jo, Sangyeob Kim, Sangjin Kim, Donghyeon Han, Jinsu Lee and Hoi-Jun Yoo

Semiconductor System Lab.
School of EE, KAIST
Computing-in-Memory (CIM) Design

- Moving Computation into Memory

<Von-Neumann Arch.>

<Computing-in-Memory Arch.>

SRAM

Computing Unit

Processor

Energy Efficient Memory Structure

Computing Unit

Processor
Limitation of Previous CIM Hardwares

- Cannot Support Floating-point (FP) Data Representation
  - Achieved SOTA energy efficiency, but no fixed-point (FXP) precision

<table>
<thead>
<tr>
<th></th>
<th>2019</th>
<th>2020</th>
<th>2021</th>
</tr>
</thead>
<tbody>
<tr>
<td>Floating point (FP)</td>
<td>??</td>
<td>??</td>
<td>Proposed Processor</td>
</tr>
<tr>
<td>Fixed point (FXP)</td>
<td>ISSCC19 ` J.Wang</td>
<td>ISSCC19 ` J.Yang</td>
<td>ISSCC19 ` X.Si</td>
</tr>
<tr>
<td></td>
<td>VLSI20 ` J.Kim</td>
<td>ISSCC20 ` J.Su</td>
<td>ISSCC20 ` C.Xue</td>
</tr>
<tr>
<td></td>
<td>ISSCC21 ` H.Jia</td>
<td>ISSCC21 ` J.Yue</td>
<td></td>
</tr>
</tbody>
</table>

**Proposed Processor:**
- HEMTC Cluster 1

ISSCC19, J.Wang
ISSCC19, J.Yang
ISSCC19, X.Si
VLSI20, J.Kim
ISSCC20, J.Su
ISSCC20, C.Xue
ISSCC21, H.Jia
ISSCC21, J.Yue
Abstract of Proposed FP CIM Processor

(1) Heterogeneous FP Computing Arch.
: Separate optimization of FP computing
: Realize 2 cycles FP MAC w/ CIM 😊

(2) Exponent Computing-in-Memory
: In-memory AND/NOR + BL charge reusing
: Total memory power 46.4 % ↓

(2) Mantissa Free Exponent Calculation
: Removing redundant normalization
: Total MAC power 14.4 % ↓

**Chip Photo & Spec**

**Key Feature List**

<table>
<thead>
<tr>
<th>Technology</th>
<th>28nm Logic CMOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Die Area</td>
<td>1.62 mm × 3.6 mm</td>
</tr>
<tr>
<td>Precision</td>
<td>BFloat16</td>
</tr>
<tr>
<td>Energy Efficiency [TFLOPS/W]</td>
<td>1.43* – 13.7** @ (40MHz, 0.76V)</td>
</tr>
</tbody>
</table>

* = Activation Sparisty 0%, ** = Activation Sparisty 90%
(1 MAC = 2 OP)
Floating-point (FP) Computing System

- Memory Part: Same Op. for Mantissa and Exponent
Heterogeneous Characteristics!

- **Exp.**: memory Intensive $\Leftrightarrow$ **Man.**: computation Intensive
  - Hard to optimize both mantissa and exponent w/ CIM at once! 😁
Previous FP CIM Architecture*

- Unified FP Computing-in-memory Architecture
  - Complex mantissa op. ➔ Repetitive simple op. ➔ Throughput ↓ 😞

<Prev. FP Computing Arch.>

@\( T=0-T_e \) ➔ @\( T=T_e-T_e+T_m \)

<Operation Cycles Analysis>

1) BFP16 Multiplication
   : Mantissa Mult. ➔ Repetitive Addition
   > 100 cycles required

2) FP32 Accumulation
   : Mantissa Shift
   Mantissa Norm. ➔ Repetitive Addition
   > 4900 cycles required

Total > 5000 cycles required 😞

Proposed Heterogeneous FP CIM Arch.

- Only exponent operations optimized w/ CIM
  - Cycles for FP MAC: >5000 cycles ➔ 2 cycles 😊
Overall Processor Architecture

- 16 Heterogeneous-exponent-mantissa-training-core (HEMTC)
- Aggregation & Activation Core, Top RISC Controller, etc ...

![Diagram of processor architecture with details on HEMTC clusters, network interfaces, shared bus, and various memory and processor components.]
Key Features for Energy-Efficient FP CIM

- **1. Exponent-Computing-in-Memory (ECIM)** ➔ Exponent
ECIM Architecture

- **512×128 Weight ECIM**
  - Inference, Back-propagation
- **128×128 Output ECIM**
  - Weight-gradient
- **Shared Exponent Peripheral**
  - Finalize exponent op.
- **CLA Decoder, WL Driver**
- **Normal I/O Interface**
ECIM Architecture - Detail

- **CIM Local Array (CLA)**
  - 32 6T bitcells
  - \( V_{DD} \) Pre-charger

- **Near-Memory Logic**
  - Simplified exp. Adder
  - Exp. comparator

- **Features for Low Memory Power**
  - 1) In-memory AND/NOR op. @ CLA
  - 2) BL charge reusing @ hierarchical BL
3 Stage 6T-SRAM In-memory AND/NOR Operations

- Preventing result corruption by lowering WL voltage

<table>
<thead>
<tr>
<th>Stored Bit</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precharge</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>LBL</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>LBLB</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ V_{th} \approx V_{WL} \approx V_{NML} + V_{th} \]

\* \( V_{NML} \) = GBL driver's noise margin low
Operation w/ Bitline Charge Reusing

- Charge Reusing w/ Hierarchical Bitline Structure
  - Condition: Previous CLA’s and/or result = current CLA’s and/or result

<ECIM Status>

<Operational Waveforms>

| Operation Scenario | @T=0: Cell bit = 0, Precharge bit = 1 | @T=1: Cell bit = 1, Precharge bit = 0 | Reuse! | Reuse! |
Result of ECIM

- Reducing BL Pre-charge Power Consumption
  - >84 % of GBL charge reused, 46.4 % of memory access power ↓ 😊

<GBL Charge Reuse Ratio (%)>

<Normalized MEM Power *>

* Measured @ 512 Ch_{in}, 16 Ch_{out}, No Sparsity, 512×128 bit ECIM, 32 cells in CLA
Mantissa-Free-Exponent-Calculation

- **Key Concept: Removing Redundant Norm. Process**
  - Normalize every MAC
    - Overflow counter + Accumulation register
  - Normalize once after accumulation finished
    - Pipelining btw/ ECIM & Mantissa PE available 😊
    - Shortening critical path 😊

### <FP MAC w/ MFEC>
- Exp.Subtractor
- Mantissa Shifter → Mantissa Adder
- Reg. → Ovf Counter → Accum Reg.
- Exponent Updater → Norm. & Round

### <FP Op. Pipeline>
- FP Multiplier
- FP Mult.
- FP Mult.
- FP Add.
- FP Add.
- FP Add.
- FP Norm.
Result of MFEC

- Removing Redundant Normalization Process
  - Reduce Power and Area of Mantissa MAC 😊

<Normalized MAC Power>

<Normalized MAC Area>

Total 14.4% Reduction
Total 11.7% Reduction
- **28.6 ~ ×274** Higher Energy Efficiency than Previous FP CIM

---

**Chip Specifications**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Technology</strong></td>
<td>28nm Logic CMOS</td>
</tr>
<tr>
<td><strong>Die Area</strong></td>
<td>1.62 mm × 3.6 mm</td>
</tr>
<tr>
<td><strong>Precision</strong></td>
<td>BFloat16</td>
</tr>
<tr>
<td><strong>Supply Voltage</strong></td>
<td>0.76 V ~ 1.1 V</td>
</tr>
<tr>
<td><strong>Frequency</strong></td>
<td>~ 250 MHz</td>
</tr>
<tr>
<td><strong>Peak Performance</strong></td>
<td>119.4* – 662.0* @ 250MHz</td>
</tr>
<tr>
<td><strong>Power Consumption [mW]</strong></td>
<td>1.2* – 2.1* @ (5MHz, 0.76V)</td>
</tr>
<tr>
<td></td>
<td>91.0* – 156.1 * @ (250MHz, 1.1V)</td>
</tr>
<tr>
<td><strong>Energy Efficiency [TFLOPS/W]</strong></td>
<td>1.43* – 13.7* @ (40MHz, 0.76V)</td>
</tr>
<tr>
<td></td>
<td>0.76* – 7.3* @ (250MHz, 1.1V)</td>
</tr>
</tbody>
</table>

* = Activation Sparisty 0%, ** = Activation Sparisty 90% (1 MAC = 2 OP)
Conclusion

- An Energy-Efficient Floating-Point (FP) CIM Processor

  - For High Energy-Efficiency
    - Heterogenous FP computing arch. → MAC cycle: >5000 → 2 cycle
    - Mantissa-free-exponent-calculation → MAC power 14.4 % ↓
    - Exponent Computing-in-Memory → MEM power 46.4 % ↓

A 13.7 TFLOPS/W Floating-point Computing-in-Memory Processor for DNN Training
Thank You!

▪ Questions? Feel Free to Contact Me!
  – E-mail: juhyoung@kaist.ac.kr
  – LinkedIn: https://www.linkedin.com/in/juhyounglee
  – Zoom Meeting: https://us02web.zoom.us/j/3466650389?pwd=c1RWaXFTWGljaVU1MiNtcDhKaGg0dz09 (Password: HC_JHLEE)

▪ Acknowledgement
  – This work was supported by the Samsung Electronics.