P&S Processing-in-Memory

Real-World Processing-in-Memory Architectures: Alibaba Hybrid Bonding PNM Engine

Dr. Juan Gómez Luna
Prof. Onur Mutlu

ETH Zürich
Spring 2023
27 April 2023
### UPMEM PIM System Organization

- A UPMEM DIMM contains **8 or 16 chips**
  - Thus, **1 or 2 ranks** of 8 chips each
- Inside each PIM chip there are:
  - **8 64MB banks** per chip: **Main RAM (MRAM)** banks
  - **8 DRAM Processing Units (DPUs)** in each chip, 64 DPUs per rank
PIM units respond to standard DRAM column commands (RD or WR)

- Compliant with unmodified JEDEC controllers

- They execute one wide-SIMD operation commanded by a PIM instruction with deterministic latency in a lock-step manner

- A PIM unit can get 16 16-bit operands from IOSAs, a register, and/or the result bus
AiM: System Organization

- GDDR6-based AiM architecture
AxDIMM Design: Hardware Architecture

As illustrated in Figure 4, the AxDIMM system is an FPGA-based DDR4-compatible board with standard DIMM interface that serves as a non-acceleration mode, the host can directly access the DDR4-compatible FPGA module instantiated on the AxDIMM board. The interfaces of the Rank-NMP module, (c) Host-NMP offloading model.

The execution of Rank-NMP modules is supported in two modes: (a) non-acceleration mode, and (b) acceleration mode.

**Figure 4:** (a) DDR4-compatible FPGA-based DDR4 Slave PHY (64 bits) (b) Standard DIMM Interface and (c) Host-NMP offloading model.

The memory interface generator (MIG) supports the interface with the two internal ranks, and the non-acceleration mode, the host can directly access the DRAM devices and the Psum buffer, the instruction decoder holds the intermediate Psum values for embedding pooling operations.

The execution of Rank-NMP modules is supported in two modes: (a) non-acceleration mode, and (b) acceleration mode.

NMP-Inst (64 bits):

<table>
<thead>
<tr>
<th>OpCode</th>
<th>Locality</th>
<th>PSUM Tag</th>
<th>Trace End</th>
<th>Reserved</th>
<th>Row Addr</th>
<th>BG</th>
<th>BA</th>
<th>Col Addr</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 bit</td>
<td>1 bit</td>
<td>12 bit</td>
<td>1 bit</td>
<td>17 bit</td>
<td>17 bit</td>
<td>2 bit</td>
<td>2 bit</td>
<td>10 bit</td>
</tr>
</tbody>
</table>

Alibaba 3D Logic-to-DRAM Hybrid Bonding with Processing-near-Memory Engine
Hybrid Bonding with PnM Engine (ISSCC 2022)

ISSCC 2022 / SESSION 29 / ML CHIPS FOR EMERGING A

29.1 184QPS/W 64Mb/mm² 3D Logic-to-DRAM Hybrid Bonding with Process-Near-Memory Engine for Recommendation System

Dimin Niu¹, Shuangchen Li¹, Yuhao Wang¹, Wei Han¹, Zhe Zhang², Yijin Guan², Tianchan Guan³, Fei Sun¹, Fei Xue¹, Lide Duan¹, Yuanwei Fang¹, Hongzhong Zheng¹, Xiping Jiang⁴, Song Wang⁴, Fengguo Zuo⁴, Yubing Wang⁴, Bing Yu⁴, Qiwei Ren⁴, Yuan Xie¹

¹Alibaba DAMO Academy, Sunnyvale, CA; ²Alibaba DAMO Academy, Beijing, China
³Alibaba DAMO Academy, Shanghai, China; ⁴UniIC, Xian, China

https://doi.org/10.1109/ISSCC42614.2022.9731694
Processing-in-Memory for Machine Learning

- Memory bandwidth is not enough for many ML workloads

Solution Needed
- Limited by physical limit
- New chip architecture or new memory technology

Scaling Out Solution
- More hardware
- More computation time

750x / 2 years
AI model computation requirement

3.1x / 2 years
Hardware computation capability

1.4x / 2 years
Memory system capability

Natural Language Processing

Recommendation Systems

Graph Neural Network

Multi-Task Online Inference

Niu et al., 184QPS/W 64Mb/mm² 3D Logic-to-DRAM Hybrid Bonding with Process-Near-Memory Engine for Recommendation System, ISSCC 2022
Processing-in-Memory Classification

Traditional

Memory ↔ Computation

2D CIM

Memory & Computation

2D PNM

Memory ↔ Computation

2.5D PNM

Interposer

3D HB-PNM

Memory

Computation

3D TSV-PNM

Memory

Memory

Memory

Computation

PNM: Process Near Memory
CIM: Compute In Memory

HB: Hybrid Bonding
TSV: Through-silicon Via

Niu et al., 184QPS/W 64Mb/mm2 3D Logic-to-DRAM Hybrid Bonding with Process-Near-Memory Engine for Recommendation System, ISSCC 2022
Two PIM Approaches

5.2. Two Approaches: Processing Using Memory (PUM) vs. Processing Near Memory (PNM)

Many recent works take advantage of the memory technology innovations that we discuss in Section 5.1 to enable and implement PIM. We find that these works generally take one of two approaches, which are categorized in Table 1: (1) processing using memory or (2) processing near memory. We briefly describe each approach here. Sections 6 and 7 will provide example approaches and more detail for both.

Table 1: Summary of enabling technologies for the two approaches to PIM used by recent works. Adapted from [341] and extended.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Example Enabling Technologies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processing Using Memory</td>
<td>SRAM</td>
</tr>
<tr>
<td></td>
<td>DRAM</td>
</tr>
<tr>
<td></td>
<td>Phase-change memory (PCM)</td>
</tr>
<tr>
<td></td>
<td>Magnetic RAM (MRAM)</td>
</tr>
<tr>
<td></td>
<td>Resistive RAM (RRAM)/memristors</td>
</tr>
<tr>
<td>Processing Near Memory</td>
<td>Logic layers in 3D-stacked memory</td>
</tr>
<tr>
<td></td>
<td>Silicon interposers</td>
</tr>
<tr>
<td></td>
<td>Logic in memory controllers</td>
</tr>
<tr>
<td></td>
<td>Logic in memory chips (e.g., near bank)</td>
</tr>
<tr>
<td></td>
<td>Logic in memory modules</td>
</tr>
<tr>
<td></td>
<td>Logic near caches</td>
</tr>
<tr>
<td></td>
<td>Logic near/in storage devices</td>
</tr>
</tbody>
</table>

PIM Review and Open Problems

A Modern Primer on Processing in Memory

Onur Mutlu\textsuperscript{a,b}, Saugata Ghose\textsuperscript{b,c}, Juan Gómez-Luna\textsuperscript{a}, Rachata Ausavarungnirun\textsuperscript{d}

SAFARI Research Group

\textsuperscript{a}ETH Zürich
\textsuperscript{b}Carnegie Mellon University
\textsuperscript{c}University of Illinois at Urbana-Champaign
\textsuperscript{d}King Mongkut’s University of Technology North Bangkok

Onur Mutlu, Saugata Ghose, Juan Gomez-Luna, and Rachata Ausavarungnirun, "A Modern Primer on Processing in Memory"

HB-PNM: Overall Architecture (I)

- 3D-stacked logic die and DRAM die vertically bonded by hybrid bonding (HB)
HB-PNM: Overall Architecture (II)

- Match engine and neural engine for matching and ranking in a recommendation system

![Diagram of the overall architecture](image)
Recommendation Systems
Feature Generation + Matching & Ranking

- Recommendation system
  - Feature generation
    - Classification, object detection, feature extraction
  - Compute-bound
  - Good fit for GPU
  - Matching and ranking
    - Coarse-grained matching, fine-grained ranking
  - Memory-bound
    - Most latency (89.87%) and energy (82.97%)
    - Typically run on CPU

Niu et al., 184QPS/W 64Mb/mm2 3D Logic-to-DRAM Hybrid Bonding with Process-Near-Memory Engine for Recommendation System, ISSCC 2022
Matching & Ranking

- **Coarse-grained matching**
  - Binary feature vectors with 512 dimensions
  - Distance calculation
  - Top-1000 items selected from 40K items

- **Fine-grained ranking**
  - Features with 8 bits x 1024 dimensions
  - 3-layer MLP (2048-256-64-1) for similarity prediction
  - Top-100 ranking results from 1K items
3D Logic-to-DRAM Hybrid Bonding
HB-PNM: Chip Implementation

- 3D Logic-to-DRAM Hybrid Bonding
  - Face-to-face hybrid wafer bonding

![Cross-section illustration of 3D Logic-to-DRAM Hybrid Bonding](image)
HB-PNM Architecture
HB-PNM Architecture

- DRAM die composed of 6x6 1Gb DRAM cores
  - 8 banks per core
  - 128-bit I/O per bank
  - On-chip ECC (8 Mb per 128 Mb)

HB imposes design constraints on location of memory controllers (MC) and PHY.
HB-PNM: Logic Die

- **Match engine and neural engine** for matching and ranking in a recommendation system
  - Direct access to their counterpart DRAM blocks
  - Access to other DRAM blocks via on-chip bus
**HB-PNM Logic Die: Dual-mode Interface**

- **Dual-mode interface** can switch between
  - All 8 banks in lock-step for full bandwidth
  - Single channel (1 of 8 banks)
HB-PNM: Match Engine

- Responsible for coarse-grained matching

Overall Architecture

- Memory
  - 4 x 1 Gb blocks with 4096 bits I/O
  - 38.4 GB/s on-chip bandwidth per block

- Compute
  - Match Engine: Coarse-grained Matching
  - Neural Engine: Fine-grained Ranking

Dual-mode Interface

Match Engine (ME)

Logic Die

Utility
  - PERF.
  - DEBUG
  - MODE

Max-heap Top-K Engine

Distance Calculator

FIFO

AddrGen

REGs

SPI bridge

QSPI

I/F Bridge

Memory Controller

DRAM-0

Memory Controller

DRAM-1

GPIO

Niu et al., 184QPS/W 64Mb/mm2 3D Logic-to-DRAM Hybrid Bonding with Process-Near-Memory Engine for Recommendation System, ISSCC 2022
Match Engine: Distance Calculator

- **Distance calculator** obtains similarity between input query and feature vectors
  - It computes **Hamming distance of two 512-bit vectors**
  - Distance is filtered by root of max-heap
Match Engine: Top-K Engine (II)

- **Max-heap hardware block** and data structure with 1000 nodes <address, distance> for the 1000 shortest distances

### Distance calculator

- **Addr**
- **Data**
- **Query Data**

  - Bit-wise Hamming Distance
  - Partial Popcount 0
  - Partial Popcount 1

### Top-K

1. **Start**
2. **Reset**
3. **Input?**
   - **Odd clk cycle?**
     - **N**
     - **Y**
       - Root.Value = Distance
       - Root.Metadata = Addr
4. **For all nodes on odd layer**
   - **< child nodes?**
     - **Y**
     - **N**
       - **< left child nodes?**
         - **Y**
         - **N**
5. **For all nodes on even layer**

If Distance < left child, swap left child; Else, swap right child
HB-PNM: Neural Engine

- Responsible for similarity prediction for fine-grained ranking
Neural Engine: Vector Processing Unit

- **Activations** based on LUTs
  - Support for GeLU and Exp

- **Transpose**
  - Transpose 16x16 matrix with ping-pong array
  - 2D register file array
  - Row-based writes and column-based reads
Neural Engine: GEMM

- 32x32 INT8 fully-pipelined systolic array
  - Partial sums accumulated in INT32 accumulator
## HB-PNM: Key Feature Summary

### Comparison table

<table>
<thead>
<tr>
<th></th>
<th>2D CIM *</th>
<th>UPMEM PIM **</th>
<th>A100 GPU ***</th>
<th>FIMDRAM ****</th>
<th>This work</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type of Memory</strong></td>
<td>SRAM</td>
<td>DDR4</td>
<td>HBM2</td>
<td>HBM2</td>
<td>LPDDR4</td>
</tr>
<tr>
<td><strong>Technology (Memory/Logic)</strong></td>
<td>16nm</td>
<td>2xnm / 2xnm</td>
<td>1y# / 7nm</td>
<td>20nm / 20nm</td>
<td>25nm / 55nm</td>
</tr>
<tr>
<td><strong>Capacity</strong></td>
<td>4.5 Mb</td>
<td>8GB / DIMM</td>
<td>80GB</td>
<td>6GB / cube</td>
<td>4.5GB</td>
</tr>
<tr>
<td><strong>Bandwidth</strong></td>
<td>-</td>
<td>128GB/s / DIMM</td>
<td>1935GB/s</td>
<td>1200GB/s / cube#</td>
<td>38.4GB/s / 1Gb</td>
</tr>
<tr>
<td><strong>Frequency (Logic)</strong></td>
<td>200MHz</td>
<td>500MHz</td>
<td>1410MHz</td>
<td>300MHz</td>
<td>300MHz</td>
</tr>
<tr>
<td><strong>Bandwidth/Capacity (a.u.)</strong></td>
<td>-</td>
<td>16</td>
<td>24.2</td>
<td>200</td>
<td>307</td>
</tr>
</tbody>
</table>

** F. Devaux et al, Hotchip 2019
*** J. Choquette et al, Hotchip 2020
**** Y. C. Kwon et al, ISSCC 2021
Longer Lecture on HB-PNM

Neural Engine: GEMM

- 32x32 INT8 fully-pipelined systolic array
  - Partial sums accumulated in INT32 accumulator

https://youtu.be/8MM6_36LmWQ
Processing-in-Memory Classification

Traditional

Memory <-> Computation

CPU

GPU

FPGA

2D CIM

Memory & Computation

PUM (e.g., SIMDRAM, NVM...)

3D HB-PNM

Memory

Computation

3D TSV-PNM

Memory

Memory

Memory

Computation

PUM : Process Near Memory
CIM: Compute In Memory

HB : Hybrid Bonding
TSV: Through-silicon Via

Niu et al., 184QPS/W 64Mb/mm2 3D Logic-to-DRAM Hybrid Bonding with Process-Near-Memory Engine for Recommendation System, ISSCC 2022
UPMEM Processing-in-DRAM Engine (2019)

- **Processing in DRAM Engine**
  - Includes **standard DIMM modules**, with a **large number of DPU processors** combined with DRAM chips.

- **Replaces standard DIMMs**
  - DDR4 R-DIMM modules
    - 8GB+128 DPUs (16 PIM chips)
    - Standard 2x-nm DRAM process
  - **Large amounts of** compute & memory bandwidth

---

CPU (x86, ARM, RV…)


FIMDRAM: Chip Structure

- FIMDRAM based on HBM2

[3D Chip Structure of HBM with FIMDRAM]

Chip Specification

- 128DQ / 8CH / 16 banks / BL4
- 32 PCU blocks (1 FIM block/2 banks)
- 1.2 TFLOPS (4H)
- FP16 ADD / Multiply (MUL) / Multiply-Accumulate (MAC) / Multiply-and-Add (MAD)
AiM: Chip Implementation

- 4 Gb AiM die with 16 processing units (PUs)

AiM Die Photograph

1 Process Unit (PU) Area

<table>
<thead>
<tr>
<th>Area</th>
<th>Area (mm²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>0.19</td>
</tr>
<tr>
<td>MAC</td>
<td>0.11</td>
</tr>
<tr>
<td>Activation Function (AF)</td>
<td>0.02</td>
</tr>
<tr>
<td>Reservoir Cap.</td>
<td>0.05</td>
</tr>
<tr>
<td>Etc.</td>
<td>0.01</td>
</tr>
</tbody>
</table>

Reservoir Cap. 26%
MAC 58%
AF 11%
Etc. 5%
Samsung AxDIMM (2021)

- DIMM-based PIM
  - DLRM recommendation system

Variety of Current Real PIM Architectures

- UPMEM PIM, Samsung HBM-PIM, Samsung AxDIMM, SK Hynix AiM, Alibaba HB-PNM

• Differences
  - Near-bank (UPMEM, FIMDRAM, AiM, HB-PNM) vs. near-chip (AxDIMM)
  - General-purpose (UPMEM) vs. special-function (FIMDRAM, AiM, HB-PNM)
  - FGMT (UPMEM) vs. SIMD (FIMDRAM, AiM, AxDIMM) vs. systolic array (HB-PNM)
  - Natively integer (UPMEM, HB-PNM) vs. floating point (FIMDRAM)
    - FP16 (FIMDRAM) vs. BF16 (AiM) vs. FP32 (AxDIMM)
  - DDR4 (UPMEM, AxDIMM) vs. LPDDR4 (HB-PNM) vs. HBM2 (FIMDRAM) vs. GDDR6 (AiM)
Common Characteristics

• These PIM systems have some common characteristics:

1. There is a host processor (CPU or GPU) with access to (1) standard main memory, and (2) PIM-enabled memory

2. PIM-enabled memory contains multiple PIM processing elements (PEs) with high bandwidth and low latency memory access

3. PIM PEs run only at a few hundred MHz and have a small number of registers and small (or no) cache/scratchpad

4. PIM PEs may need to communicate via the host processor
A State-of-the-Art PIM (PNM) System

These PIM systems have some common characteristics:

1. There is a host processor (CPU or GPU) with access to (1) standard main memory, and (2) PIM-enabled memory
2. PIM-enabled memory contains multiple PIM processing elements (PEs) with high bandwidth and low latency memory access
3. PIM PEs run only at a few hundred MHz and have a small number of registers and small (or no) cache/scratchpad
4. PIM PEs may need to communicate via the host processor
Upcoming Lectures

- PUM architectures and prototypes

- Case studies
  - SpMV on UPMEM PIM architecture
  - Neural network accelerators for the edge
  - Hybrid transactional and analytical processing (HTAP) databases

- Enabling the adoption of PIM
P&S Processing-in-Memory

Real-World Processing-in-Memory Architectures: Alibaba Hybrid Bonding PNM Engine

Dr. Juan Gómez Luna
Prof. Onur Mutlu

ETH Zürich
Spring 2023
27 April 2023