P&S Processing-in-Memory

Real-World Processing-in-Memory Architectures:
UPMEM PIM Architecture

Dr. Juan Gómez Luna
Prof. Onur Mutlu
ETH Zürich
Spring 2023
23 March 2023
PIM Becomes Real

- **UPMEM**, founded in January 2015, announces the first real-world PIM architecture in 2016.

- UPMEM’s PIM-enabled DIMMs start getting commercialized in 2019.

- In early 2021, **Samsung** announces **FIMDRAM** at ISSCC conference.

- Samsung’s LP-DDR5 and DIMM-based PIM announced a few months later.

- In early 2022, **SK Hynix** announces **AiM** at ISSCC conference.

[startup plans to embed processors in DRAM](https://www.eenewsautomotive.com/news/startup-plans-embed-processors-dram-0#)
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

[Image of DRAM module with text overlay]


Samsung Function-in-Memory DRAM (2021)

- 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)

ISSCC 2021 / SESSION 25 / DRAM / 25.4

25.4 A 20nm 6GB Function-In-Memory DRAM, Based on HBM2 with a 1.2TFLOPS Programmable Computing Unit Using Bank-Level Parallelism, for Machine Learning Applications

Young-Cheon Kwon1, Suk Han Lee1, Jae-Hoon Lee1, Sang-Hyuk Kwon1, Ji Min Ryu1, Jong-Pil Son1, Seongill O1, Hak-Soo Yu1, Hassuk Lee1, Soo Young Kim1, Youngmin Cho1, Jin Guk Kim1, Jongyoon Choi1, Hyun-Sung Shin1, Jin Kim1, Beng Seng Phua1, Hyo Young Min Kim1, Myeong Jun Song1, Ahn Choi1, Daeho Kim1, Soo Young Kim1, Eun Bong Kim1, David Wang1, Shin Tae Kang1, Yuhwan Ro1, Seungwoo Seo1, Joon Ho Song1, Janyoun Youn1, Kyomin Sohn1, Nam Sung Kim1

1Samsung Electronics, Hwasung, Korea
2Samsung Electronics, San Jose, CA
3Samsung Electronics, Suwon, Korea
Samsung AxDIMM (2021)

- DIMM-based PIM
  - DLRM recommendation system

UPMEM PIM
Microarchitecture and ISA
Accelerator Model (I)

• UPMEM DIMMs coexist with conventional DIMMs

• Integration of UPMEM DIMMs in a system follows an accelerator model

• UPMEM DIMMs can be seen as a loosely coupled accelerator
  - Explicit data movement between the main processor (host CPU) and the accelerator (UPMEM)
  - Explicit kernel launch onto the UPMEM processors

• This resembles GPU computing
Accelerator Model (II)

- FIG. 6 is a flow diagram representing operations in a method of delegating a processing task to a DRAM processor according to an example embodiment.

```
FIG. 6

1. SOC loads data to be processed to DRAM memory bank.
2. SOC transmits data processing command to DRAM processor(s).
3. Data processing by DRAM processor(s).
4. Check if data processing is complete.
   - If not (N), go back to step 3.
   - If yes (Y), memory bank accessible by SOC.

Fig 6

```
System Organization (I)

- **FIG. 1** schematically illustrates a computing system comprising DRAM circuits having integrated processors according to an example embodiment.

Fig 1
System Organization (II)

- In a UPMEM-based PIM system UPMEM DIMMs coexist with regular DDR4 DIMMs
System Organization (III)

• 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
2,560-DPU System (II)
Vector Addition (VA)

- Our first programming example
- We partition the input arrays across:
  - DPUs
  - Tasklets, i.e., software threads running on a DPU
CPU-DPU/DPU-CPU Data Transfers

- **CPU-DPU and DPU-CPU transfers**
  - Between host CPU’s main memory and DPUs’ MRAM banks

- **Serial CPU-DPU/DPU-CPU transfers:**
  - A single DPU (i.e., 1 MRAM bank)

- **Parallel CPU-DPU/DPU-CPU transfers:**
  - Multiple DPUs (i.e., many MRAM banks)

- **Broadcast CPU-DPU transfers:**
  - Multiple DPUs with a single buffer
Inter-DPU Communication

- There is **no direct communication channel between DPUs**

- Inter-DPU communication takes places via the host CPU using CPU-DPU and DPU-CPU transfers

- Example communication patterns:
  - Merging of partial results to obtain the final result
    - Only DPU-CPU transfers
  - Redistribution of intermediate results for further computation
    - DPU-CPU transfers and CPU-DPU transfers
DRAM Processing Unit (I)

- FIG. 4 schematically illustrates part of the computing system of FIG. 1 in more detail according to an example embodiment.

DRAM Processing Unit (II)

PIM Chip

Control/Status Interface

DDR4 Interface

DISPATCH
FETCH1
FETCH2
FETCH3
READOP1
READOP2
READOP3
FORMAT
ALU1
ALU2
ALU3
ALU4
MERGE1
MERGE2
DMA Engine

PIM Chip

24-KB IRAM

64-KB WRAM

64-MB DRAM Bank (MRAM)

Register File

Pipeline

64 bits

x8

SAFARI
DPU Pipeline

• In-order pipeline
  - Up to 425 MHz

• Fine-grain multithreaded
  - 24 hardware threads

• 14 pipeline stages
  - DISPATCH: Thread selection
  - FETCH: Instruction fetch
  - READOP: Register file
  - FORMAT: Operand formatting
  - ALU: Operation and WRAM
  - MERGE: Result formatting
DPU Instruction Set Architecture

• Specific 32-bit ISA
  - Aiming at scalar, in-order, and multithreaded implementation
  - Allowing compilation of 64-bit C code
  - LLVM/Clang compiler

Instruction Set Architecture

This section covers the architecture concepts required to understand and use UPMEM DPU processor as a software developer. It is also providing an exhaustive list of the available processor instructions.

Software developers should use this section as a reference manual to develop or debug assembly code.

Resources overview

Thread registers

The system is composed of 24 hardware threads. Each of them owns a set of private resources:

• 24 general purpose 32-bits registers named $r0$ through $r23$
• A 16-bits wide program counter, named PC. Notice that the PC value does not address an instruction in memory, but the index of such an instruction directly. For example, a PC equal to 1 represents the second instruction in the DPU’s program memory.
• Two persistent flags, keeping information about the previous result of an arithmetic or logical instruction:
  - ZF: last result is equal to zero

https://sdk.upmem.com/2021.2.0/201_IS.html#
Microbenchmark for INT32 ADD Throughput

```c
#define SIZE 256
int* bufferA = mem_alloc(SIZE * sizeof(int));
for(int i = 0; i < SIZE; i++){
    int temp = bufferA[i];
    temp += scalar;
    bufferA[i] = temp;
}
```

Compiled code (UPMEM DPU ISA)

```assembly
move r2, 0
.LBB0_1:  // Loop header
  lsl_add r3, r0, r2, 2 // Address calculation
  lw r4, r3, 0 // Load from WRAM
  add r4, r4, r1 // Add
  sw r3, 0, r4 // Store to WRAM
  add r2, r2, 1 // Index update
  jneq r2, 256, .LBB0_1 // Conditional jump
```
DPU: Arithmetic Throughput vs. Operational Intensity

**PIM Chip**

Control/Status Interface

DDDR4 Interface

DISPATCH

FETCH1

FETCH2

FETCH3

READOP1

READOP2

READOP3

FORMAT

ALU1

ALU2

ALU3

ALU4

MERGE1

MERGE2

64-KB WRAM

DMA Engine

64-MB DRAM Bank (MRAM)

24-KB IRAM
Upcoming Lectures

- Microbenchmarking of the UPMEM DPU
  - Compute throughput
  - MRAM and WRAM bandwidth
  - Arithmetic intensity versus compute throughput
- Programming an UPMEM-based PIM system
- Introduction to Samsung’s, SK Hynix’s, and Alibaba’s PIM devices
Understanding a Modern PIM Architecture

Benchmarking a New Paradigm: Experimental Analysis and Characterization of a Real Processing-in-Memory System

JUAN GÓMEZ-LUNA¹, IZZAT EL HAJJ², IVAN FERNANDEZ¹,³, CHRISTINA GIANNOLA¹,⁴, GERALDO F. OLIVEIRA¹, AND ONUR MUTLU¹

¹ETH Zürich
²American University of Beirut
³University of Malaga
⁴National Technical University of Athens

Corresponding author: Juan Gómez-Luna (e-mail: juang@ethz.ch).

https://github.com/CMU-SAFARI/prim-benchmarks
Understanding a Modern PIM Architecture

Juan Gómez Luna, Izzat El Hajj, Ivan Fernandez, Christina Giannoula, Geraldo F. Oliveira, Onur Mutlu

https://github.com/CMU-SAFARI/prim-benchmarks

SAFARI Live Seminar: Understanding a Modern Processing-in-Memory Architecture

2,579 views • Streamed live on Jul 12, 2021

Onur Mutlu Lectures
18.7K subscribers

https://www.youtube.com/watch?v=D8Hiy2iU9l4&list=PL5Q2soXY2Zi_tOTAYm--dYByNPL7JhwR9
2,560-DPU System (I)

- UPMEM-based PIM system with 20 UPMEM DIMMs of 16 chips each (40 ranks)
  - P21 DIMMs
  - Dual x86 socket
    - UPMEM DIMMs coexist with regular DDR4 DIMMs
  - 2 memory controllers/socket (3 channels each)
  - 2 conventional DDR4 DIMMs on one channel of one controller

* There are 4 faulty DPUs in the system that we use in our experiments. Thus, the maximum number of DPUs we can use is 2560 - 4 = 2556.
P&S Processing-in-Memory
Real-World Processing-in-Memory Architectures:
UPMEM PIM Architecture

Dr. Juan Gómez Luna
Prof. Onur Mutlu
ETH Zürich
Spring 2023
23 March 2023