

# In-Memory Processing: A look into "A Logic-In-Memory Computer", by Harold S. Stone, IEEE 1970

Peppas Athanasios, Iliadis Thrasyvoulos-Fivos

**Abstract**—In this work we discuss the use that Processing-in-Memory designs can have in modern systems. Starting from the very early steps towards PIM, we take a critical look to "A Logic-In-Memory Computer" and try to evaluate the foundations that were laid on processing in-memory architectures. We pinpoint the persistent ideas from the 70s up to today in the work as well as touch on some inaccuracies and overlooked problems. After that, we show ways that accelerate and ease the embedding of PIM units in contemporary systems, such as 3D-stacked Memories and DRAM usages.

## I. INTRODUCTION

### A. Processing-in-Memory

From the time of the article's writing to today Processing-In-Memory (or In-Memory Processing, PIM, Near-Data-Processing, NDP) is a developing field. Generally in computer science, Processing-In-Memory is a group of technologies for the processing of data stored in an in-memory database. Modern PIM architectures rely on the use of RAM because of its faster data accessing times and the emerging demands in Business Intelligence, but older systems implemented such systems based on disk storage and other slower hardware.

The problem that PIM technologies are trying to address is reducing the movement of data for computation. Modern architectures are designed with little consideration on data movement optimization. The CPU is treated as the only place that computations can take place, and data can be processed only after they are moved into proper registers (similarly for accelerators, e.g. GPUs), therefore stressing the bandwidth between CPU and memory. Apart from the bandwidth power consumption levels are also increasing. These drawbacks are particularly apparent in mobile devices and server setups, with a recent work showing that *"more than 62% of the entire system energy of a mobile device is spent on data*

*movement between the processor and the memory hierarchy for widely-used mobile workloads"*[1], [2].

The result of this is data access being a key bottleneck, especially with emergin data-heavy applications ,as well as energy consumption becoming a limiter. With data movement being very costly in terms of bandwidth, these design trends are forcing the field towards other more aware in terms of data movement architectures, one of which is the In-Memory-Processing model [2].

### B. PIM current trends

Main trends in the field

In Section II we show the main points of "A Logic-In-Memory Computer" and give an overview of the key ideas that emerge from it. Section III is where we evaluate the strengths and weaknesses of the original work, while in Section IV we discuss problems that were overlooked and give our recommendations on work to be done.

## II. PAPER OVERVIEW

Overview of the sections of the paper, Rent's rule, microelectronic memories etc.

The goal of the article written by Harold S. Stone is bringing into consideration some options on embedding logic-in-memory arrays inside conventional computers of the time. Using a contemporary computer, the IBM 360/85, as an example the paper describes a way to enhance the performance of the cache memory by arming it with some processing capabilities.

### A. Cache Memory Benefits

Overview of the IBM 360/85 using a microelectronic memory as a cache

### *B. Logic-In-Memory Array as Cache*

Overview of using a microelectronic PIM as a cache and what instructions we could use

### *C. Cache Control*

Overview of giving some control of the cache to programmers

## III. EVALUATION

Critical evaluation of the ideas in the article

## IV. FURTHER PROBLEMS

Problems with bandwidth described in literature

## REFERENCES

- [1] A. Boroumand, S. Ghose, M. Patel, H. Hassan, B. Lucia, K. Hsieh, K. T. Malladi, H. Zheng, and O. Mutlu, "Lazypim: An efficient cache coherence mechanism for processing-in-memory," *IEEE Computer Architecture Letters*, vol. 16, no. 1, pp. 46–50, 2017.
- [2] O. Mutlu, S. Ghose, J. Gómez-Luna, and R. Ausavarungnirun, "A modern primer on processing in memory," 2021.