Memory Architecture in Processing In Memory Computing.

A new approach to near memory architecture.

Morayo A Ogunsina
Department of Computer Engineering
Pennsylvania State University, The Behrend College
Erie, USA
mao5270@psu.edu

Abstract—**With the recent trend in hardware architecture, heterogeneous computing has provided efficient and capable solutions to computing problems that arise from delay in accessing data from the memory. This paper will examine the possibility of an addition or modification to the near memory architecture that can allow data not to be moved or fetched but "looked up" in the memory by the instruction that needs it. To evaluate this, focus will be on analysing the results of previous works in near memory computing that have established the basis of the reading and writing to a PIM core.**

The results to be included in this paper will indicate the viability of such architecture, its weaknesses and benefits also. This might include graphs, diagrams of processing flow etc. Most of the resources to use will be related to previous research works, textbooks and materials that help arrive at the anticipated conclusion. In the project, I plan to focus on how the processor sends out instruction to read from the memory, how it identifies the data and retrieves it back to the processor. To test for this, if the processor can use the right data in the memory without having to retrieve it, the project was successful. Success is measured on how high the percentage is in correctly using a data from memory without having to retrieve it. I expect to submit graphical, tabular and statistical evidence highlighting this result.

Keywords—PIM; heterogeneous computing; near memory computing; RAM; 3-D packaging.

I. Introduction

The idea behind the near memory computing is to make data access faster by using the concept of proximity. In a typical Von-Newman architecture, the processor and memory are separate and because the processor needs to fetch data and program from the memory unit, latency exist thereby causing a limitation in the throughput [1]. Latency is primarily caused because these bits of data and programs must move from one location to another. Now, there have been significant breakthroughs in improving memory architecture, but this is seen only in enabling more storage capacity for lesser space. The main problem, if an improvement on latency is to be made is to improve the speed of data access or transfer from the memory. An area of heterogeneous computing called the near memory computing or processing in memory (PIM), which has been a long-standing concept in the computing in early 1990's [2] has finally given a satisfactory solution to this. Previous work and research have shown that this technology is viable and can be implemented especially with the use of 3D packaged memory in which RAM chips are stacked on top of each other [1,3]. These researches also show that near memory computing can be used to achieve better concurrent data structure models which makes it possible to achieve parallel processing [3] and increased throughput. This is both interesting and important as such architecture will bypass the convention of moving data from the memory to the CPU and because such architecture might not require much resources for retrieving data since it is not retrieving it. This will reduce the number of resources needed for data retrieval and power consumed. In addition to this, this paper wishes to explore the possibility of adding some extra features to the PIM technology. This is a challenging problem because advances in this technology is relatively new despite it being a long-standing concept. There are still a lot of research being done in the computer architecture community.

"Monolithic computing units" (components such that all functionalities are interwoven and made to stand alone as a single unit) for example, an FPGA are placed near monolithic memory units all to minimize latency and distance travelled by the data to reach the processor [2]. The lesser the "distance travelled", the lesser the work and power consumed, and this reduces the overall cost of data access. Basically, it is bringing computation nearer to the memory unit. The paper has been specified into following sections: (1) problem formation to state and identify the problems and review relevant literature, (2) preliminary work, and (3) references.

II. Problem Formation

A. *Proposed modification to PIM Architecture*

The PIM technology can perhaps be modified to have some extra features. Traditionally, data is fetched from the memory to the processing unit by, but with the integration of a PIM core, data can be "looked up" without necessarily being fetched from the memory unit.

B. Analysis of Related Works on PIM Architecture

It is proposed that in a PIM architecture, the processing unit will operate on values stored in the memory, rather than loading values from registers that need to be operated on. Although, PIM allows for reduced latency in accessing data, using it as a co-processor that can assist the main processor has some benefits, for example in parallel processing [2]. There has been considerable progress in research where the PIM core technology has been used not only to break over the "Memory Wall" but also to open new memory architecture concepts such as virtual addressing to make smart-memory devices capable of multi-threaded control execution [5].

To start with, PEIs (PIM enabled instructions) can be used to extend the ISA to support PIMs. This will facilitate the computer memory architecture to carry out some simple PIM related operations and make it easier to use virtual memory and do in-memory computations [6]. So far, the "Looking up" of data can be possible if it is localized since that is the basis PIM works on. The PIM simply brings the computation to the data instead of moving the data to the point of computation. Now, for very little amounts of data, none of this matter as the latency is not quite as bad, but it becomes problematic when dealing with large data, hence the need for PIM cores.

III. PRELIMINARY WORK

A. Viability of PIM architecture

The PIM has been tested to be a very viable and feasible technology which can help overcome computing problems especially the "Memory Wall" problem. The technology of integrating the processor and memory on a single chip has brought about a lot of benefits to the microprocessor industry as it has lowered the latency in data access from the memory by the processor, increased low energy and power consumption, and even giving way for higher bandwidth.
Aside from this, new research [6] has suggested that PIM architecture can be modified in such a way to decide if it is necessary to execute instructions in the memory or processor by data locality which implies that such architecture could be made to provide an illusion that its operations were done as if they were processor instructions.

B. Data Access and Memory Architecture

The way data is retrieved from memory by the processor is the next step in determining if my original hypothesis is feasible or correct; the memory architecture implemented in a PIM will determine if "looing up" data will help reduce the latency that occurs during its access. The data access for this paper is limited to L1, L2 i.e having associative access, and L3 i.e. hierarchies whose access method is to identify data locations exactly in memory so, to have a smaller access time, the data must be very close to the accessor – the processor. (Check Fig 2).

C. Some problems facing PIMs

Although PIMs have been a conceptual solution to solving the problem of latency and low bandwidth, etcetera, there have been breakthroughs that have implemented the architecture, and have it integrated into some modern processor chips. Some few examples are IT-SRAM, Terasys, IRAM, DIVA (Check Fig. 3), PRAM, the LINDEN DAAM, and in some embedded devices. These projects addressed the implementation of PIMs from different perspectives and through different means, for example, integrating on-board memory into a traditional CPU or distributing computation across the main memory which is controlled by a different CPU.
Despite all this, there were still some problems which are largely related to the fundamental construct of memory architecture design and CPU logic. Some of these are:

- Difficulties in manufacturing high speed logic on the same die with hid density memory;
- Problems with processor-memory performance gap; the performance of processor grows at an exponential rate and far exceeds that of memory performance.
- Split of computer architecture into two industries each centered towards manufacturing memory and manufacturing processors;

D. Research Challenges

It has been quite difficult to find the right research material to use to carry out this research as the field of PIM is still an emerging one and although great advances have been made over this area of computing, much is still being explored on the subject. Materials that were eventually found to carry out this research were very technical and beyond the scope of an undergraduate degree in Computer Engineering, nevertheless, the subject seemed a very interesting one which was why I decided to research into it.

IV. FINAL RESULTS AND CONCLUSIONS

After all the relevant literature have been analysed for any possibility of modifying the processing-in-memory architecture to access data without necessarily fetching it, i.e. if it is possible to "look up" the data, the conclusive results are given in this section.

The purpose of this paper was to investigate a new architecture design and computing technology, PIM and determine if it is possible to modify such architecture to retrieve data differently. From my understanding by reading research articles and online publications, it is quite impossible to do a "lookup" without going to fetch the data from the location it is stored even if it is inside the processor. The best solution, as offered by PIM is to bring the computation to the memory and perform the necessary instruction executions. It is on the other hand, very possible that further research might prove successful in finding more ways to utilize the PIM architecture to its fullest capacity through manipulations to its design, executions and processes, such that “looking up” data is possible.

A. Figures, Tables and Diagrams

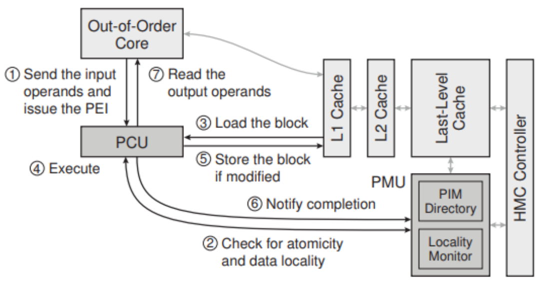

Figure 1: Host Side PEI source: [6]

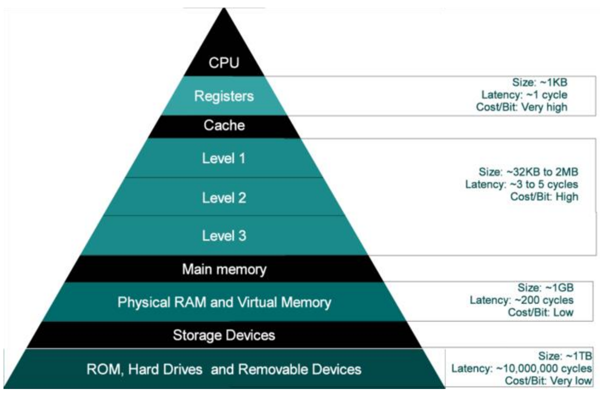

Figure 2: Memory Hierarchy. Source:

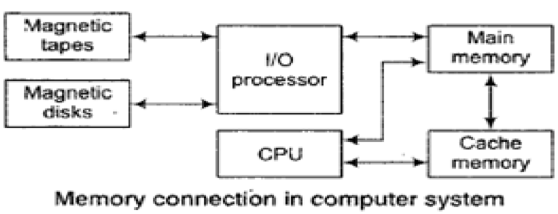

Figure 2: memory connection. Source: Gradeup.co on memory architecture

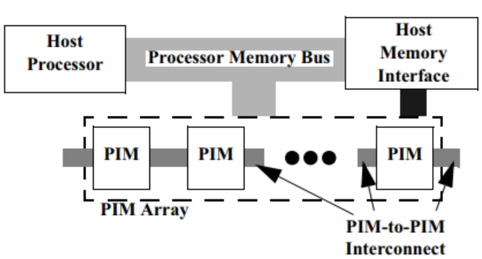

Figure 3: DIVA PIM architecture connection

B. Further Research

Further research is definitely needed to come to a very precise or accurate understanding of how data is accessed in a PIM core, how data is used for in-memory computations, etcetera which will give a clearer description of the problem and solution to achieving “looking up” data without going to access it. As stated earlier, more technical skill is needed for this as it is beyond the scope of an undergraduate degree in Computer Engineering.

C. Final Remarks

Judging from the relevant research results used for this paper, PIM is a very viable technology, one of few solutions to the Memory wall problem and although it is making strides in the computer architecture industry, it is quite difficult to envision it as the solution to completely solving the problem of latency and power consumption and generally, the memory wall problem. Its approach is simply manipulation of some computer system design technology such as caches.

The main idea here lies in the speculative fact that processors cannot get any faster than they are because they

References

[1] Liu, Zhiyu, et al, “Concurrent Data Structures for Near-Memory Computing,” Proceedings of the 29th ACM Symposium on Parallelism in Algorithms and Architectures, SPAA '17, 2017.

[2] Nikilesh Balakrishnan, et al, “Recent Advances in Computer Architecture: The Opportunities and Challenges for Provenance,” Recent Advances in Computer Architecture: The Opportunities and Challenges for Provenance, UNISEX, 2015.

[3] “Near-Memory Data Services,” IEEE Journals & Magazine, ieeexplore.ieee.org/document/7419157/.

[4] Yue Zha, Jialiang Zhang, Jing Li, Soroosh Khoram, “Challenges and Opportunities : From Near-memory Computing to In-Memory Computing”, Proceedings of the 2017 ACM International Symposium on Physical Design 2017, pp 43- 46.

[5] Granacki, John & Hall, Mary & Draper, Jeffrey & Lacoss, Jeff & Chame, Jacqueline. (2004). DIVA (Data Intensive Architecture). 404.

[6] J. Ahn, S. Yoo, O. Mutlu and K. Choi, "PIM-enabled instructions: A low-overhead, locality-aware processing-in-memory architecture," *2015 ACM/IEEE 42nd Annual International Symposium on Computer Architecture (ISCA)*, Portland, OR, 2015, pp. 336-348. doi: 10.1145/2749469.2750385.

[7] Han, Lei & Shen, Zhaoyan & Liu, Duo & Shao, Zili & Howie Huang, H & Li, Tao. (2018). A Novel ReRAM-Based Processing-in-Memory Architecture for Graph Traversal. ACM Transactions on Storage. 14. 1-26. 10.1145/3177916.

[8] Ghose, Saugata et al. “Enabling the Adoption of Processing-in-Memory: Challenges,Mechanisms,Future Research Directions” *CoRR* abs/1802.00320 (2018): n. pag.

[9] Piotr Mitros. “Computing In Memory”. [http://www.ai.mit.edu/projects/aries/course/notes/pim.html retrieved 01/04/2018](http://www.ai.mit.edu/projects/aries/course/notes/pim.html).

A General Survey of Hand Pose Estimation Projects

Morayo Ogunsina
Department of Computer Science.
California State University, Los Angeles, USA.
mogunsi@calstatela.edu

April 12, 2022

Abstract

This paper reviews some of the existing and relatively new work done in the area of hand pose estimation and prediction, with a focus on sign language translation and hand emoji prediction, in relevance to the ongoing *Emoji-Hand* project. These previous works are centered around certain computer vision, and machine learning tasks consisting of but not limited to 3D hand mesh reconstruction and keypoint mapping, emoji shape matching, prediction of different hand poses, and hand shape recognition. First, hand pose estimation is introduced by highlighting some relevant works done on the topic, and in some cases, the use they have been developed for. The paper will also discuss the methods, approaches, challenges, and limitations of the research. Finally, the paper concludes with a recommendation for some of these methods and approaches for use in the *Emoji-Hand* project.

I. Introduction

Hand pose estimation research focuses on representing a human hand, including the joint locations, orientation and articulation in space, as a mesh model or a feature outline. It also involves the simulation of hand parts in a 3D space. Hand pose estimation tasks such as 3D hand mesh reconstruction [2] and keypoint mapping, emoji matching and association [8], estimation of different hand poses, and 3D hand shape recognition [10]. Hand mesh reconstruction and hand pose estimation involve estimating and representing the 3D surface and location information of the hand joint and surface in a graph-like format and generating the 3D hand mesh in this graph from associated image features [2] [10]. For hand pose estimation and reconstruction, depth sensors are used to provide some representation of hand joint location for further processing, although more research has been geared towards the estimating hand poses and reconstructing 3D meshes from RGB cameras since they are widely available [2]. Keypoint mapping and estimation involves the determination of the kinematic structure of the joints a hand affords [10]. These motion parameters reflect the degrees of freedom (DoF) of a hand structure, allowing proper representation of the complex motion associated with a hand [10]. Hand emoji prediction involves hand shape recognition and mapping a standard emoji to its outline, as predicted by a deep neural network [4] [5] [8]. These research topics can be considered as important foundations for the development of robust and accessible applications for use in areas like sign language translation [3].

Modern hand pose estimation tasks are made possible through a singular or a combination of several computer vision (CV) and pattern recognition tasks, largely aided by Deep Learning and low cost computation and edge computing. It has become increasingly easy to generate predictive models that solve certain hand pose estimation problems and researchers leveraged this to optimize existing solutions and explore new topics in the field of hand pose estimation. In certain use or research cases, like sign language translation, gesture interaction, and even in virtual reality and hand emoji mapping, natural language processing (NLP) tasks have also been utilized to develop robust and accessible solutions [6].

II. Related Works

Classical methods and algorithms in machine learning and CV were some of the first approaches available to hand pose estimation problems and later, as computation became cheaper, faster and powerful, deep learning gained popularity with researchers who utilized them in CV, and NLP tasks relevant to estimation.

Wu *et al.* [10] explored articulated hand motion problems centered on high dimensionality introduced by the complex motion of the hand, and particle degeneracy which is a problem of the sampling process. In the paper, 3 (static constraints, joint correlation constraint, and *purposive constraints*) constraints which prevents feasible hand articulation from spanning the entire joint angle space were highlighted. The approach was to use a sequential Monte Carlo tracking algorithm as part of a divide and conquer strategy to separate the hand articulations restructure them, thereby reducing the complexity caused by many DoFs[1]. the hand affords. An importance function was tied to the for sampling process for the Monte Carlo algorithm. and thus the representation of the hand articulation is more detailed, as smaller number of points are used for efficient tracking.

Liuhao *et al.*'s [2] approach was different for the same set of problem in hand pose estimation. They first represented the 3D hand mesh as a graph-structured data, taking inspiration from works on Graph CNN such as [9]. Using an RGB image as input, 3D hand mesh vertices are generated in graph form. This approach ensures better representation of high-varied 3D hand pose as well as capture more local details.

Zimmermann *et al.* [11] used a deep network for detecting keypoints from single RGB images, and in effect, generating 3D hand pose. The use of deep neural networks are useful in bypassing occlusion, multiple ambiguities and strong articulation problems. It was one of the first works done on estimating 3D hand images from RGB cameras and encouraged the use of single-depth cameras thereby providing a more robust use coverage, particularly for sign language projects.

[1]Degrees of Freedom

Zimmermann *et al.* [11] extended the capabilities of its hand pose estimation system with a classifier for gesture recognition, an important precursor to sign language recognition.

Another related work in hand pose estimation that reflects the versatile nature of deep neural networks is found in [1], in which Transfer learning, a variation of Deep Learning was used to classify hand poses. The learning model is trained on a hand washing pose dataset.

Successful applications of hand pose estimation research has made it possible to explore development of accessible technologies, for example in sign language translation and hand emoji association as seen in [6][7] [4] and [8]. These projects either utilize deep learning and to develop an hand emoji-based hand pose predictive system or optimize existing solutions. For example, [4] and [8] made use of CNNs[2] to predict and associate standard emojis to hand signs. In [6], an NLP library which returns recognized hand signs as words,and established the importance of pretraining unlabelled and the development of a cross-lingual transfer from one sign language to another. This is particularly significant due to the concentration of available research in one sign language (American Sign Language). This research opens up transferability of solutions.

III. Recommendations and Conclusions

For the purpose of the *Emoji-Hand* project, some of the most suitable approaches that can guarantee significant success is the use of deep learning or some variations of deep learning such as Transfer Learning. Deep Learning and Computer Vision libraries have much to offer towards deploying rapid solutions for hand pose estimation tasks. It is also advisable to explore different approaches to dataset acquisition for the purpose of arriving at a robust solution.

In this paper, we highlighted some relevant works in the field of hand pose estimation as also explored certain works that have extended or optimized existing solutions for use in accessible technologies such as sign language recognition and translation systems.

[2]Convolutional Neural Networks

References

[1] Rashmi Bakshi. Hand pose classification based on neural networks, 2021.

[2] Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 3d hand shape and pose estimation from a single RGB image. *CoRR*, abs/1903.00812, 2019.

[3] Susan Goldin-Meadow and Diane Brentari. Gesture, sign and language: The coming of age of sign language and gesture studies. *The Behavioral and brain sciences*, -1:1–82, 10 2015.

[4] Jung Koh, Josh Cherian, Paul Taele, and Tracy Hammond. Developing a hand gesture recognition system for mapping symbolic hand gestures to analogous emojis in computer-mediated communication. *ACM Transactions on Interactive Intelligent Systems*, 9:1–35, 03 2019.

[5] Jaya Sahoo, Allam Prakash, Paweł Pławiak, and Saunak Samantray. Real-time hand gesture recognition using fine-tuned convolutional neural network. *Sensors*, 22:706, 01 2022.

[6] Prem Selvaraj, Gokul N. C., Pratyush Kumar, and Mitesh M. Khapra. Openhands: Making sign language recognition accessible with pose-based pretrained models across languages. *CoRR*, abs/2110.05877, 2021.

[7] Sharvani Srivastava, Amisha Gangwar, Richa Mishra, and Sudhakar Singh. Sign language recognition system using tensorflow object detection API. *CoRR*, abs/2201.01486, 2022.

[8] Neil Abraham Usha Kiruthika, Mayank Mohan. Hand gesture recognition for emoji prediction. *International Journal for Research in Applied Science and Engineering Technology*, 8:1310–1317, 06 2020.

[9] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, W. Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. *ArXiv*, abs/1804.01654, 2018.

[10] Ying Wu and John Lin. Analyzing and capturing articulated hand motion in image sequences. *IEEE transactions on pattern analysis and machine intelligence*, 27:1910–22, 01 2006.

[11] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4913–4921, 2017.