Challenges and Opportunities: From Near-memory Computing to In-memory Computing

Soroosh Khoram, Yue Zha, Jialiang Zhang, Jing Li
Department of Electrical and Computer Engineering
University of Wisconsin-Madison
{khoram, yzha3}@wisc.edu, {jialiang.zhang,jli}@ece.wisc.edu

ABSTRACT

The confluence of the recent advances in technology and the ever-growing demand for large-scale data analytics created a renewed interest in a decades-old concept, processing-in-memory (PIM). PIM, in general, may cover a very wide spectrum of compute capabilities embedded *in close proximity to* or even *inside* the memory array. In this paper, we present an initial taxonomy for dividing PIM into two broad categories: 1) Near-memory processing and 2) In-memory processing. This paper highlights some interesting work in each category and provides insights into the challenges and possible future directions.

Keywords

In-memory processing; Near-memory processing; Nonvolatile Memory; 3D Integration

1. INTRODUCTION

The rapid explosion in data, while creating opportunities for new discoveries, is also posing unprecedented demand for computing capability to handle the ever-growing data volume, velocity, variety and veracity (also known as "*four V*"), from ubiquitous and networked devices to the warehouse-scale computers [1]. As the traditional benefits for expanding the processing capability of computers through technology scaling has diminished with the end of Dennard scaling, limitations in traditional compute system, also known as "Memory Wall" [2] and "Power Wall" [3] are being outpaced by the growth of Big Data to the point where a new paradigm is needed.

As such, processing in memory (PIM), a decades-old concept, has reignited interest among industry and academic communities, largely driven by the recent advances in technology (e.g., die stacking, emerging nonvolatile memory) and the ever-growing demand for large-scale data analytics. In this paper, we classify the existing PIM work into two broad categories: 1) near-memory processing (NMP) and 2) in-memory processing (IMP). In the following sections, we will present an overview of research progress on both types.

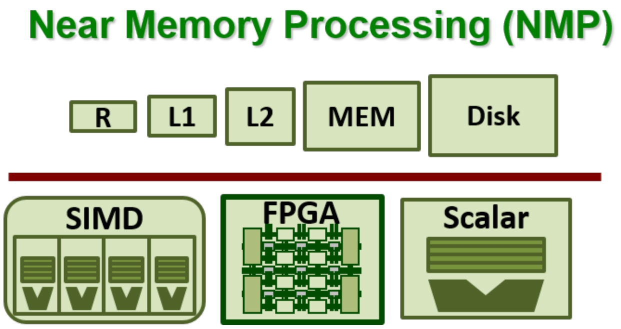

Figure 1: Conceptual diagram of Near-Memory Processing (NMP). Monolithic compute unit (multi-core, vector unit, GPU, FPGA, CGRA, ASIC etc.) are placed in close proximity to monolithic memory.

2. NEAR-MEMORY PROCESSING

The first category of PIM is near-memory processing (NMP). The underlying principle of NMP, as shown in Figure 1, is *processing in proximity of memory* – by physically placing monolithic compute units (multi-core, GPU, FPGA, ASIC, CGRA etc.) closer to monolithic memory – to minimize data transfer cost.

The original idea of implementing this type of PIM dates back to early 1990's. Since then, there has been great interest in the potential of integrating compute capabilities in large DRAM memories. Multiple research teams built NMP designs and prototypes, and confirmed speed-up in a range of applications [4, 5, 6, 7, 8, 9, 10]. Among them, EXECUBE [4], IRAM [5, 6], DIVA[7], FlexRAM[8] etc. are the representative early proposals. However, the implementation of NMP experienced great challenges in cost and manufacturability. Therefore, even with great potentials, the concept of NMP has never been embraced commercially in the early days.

Nevertheless, the practicality concerns and cost limitations of NMP are alleviated with recent advances in die-stacking technology [11, 12, 13]. Several specialized NMP systems were developed for important domains of applications [14, 15, 16, 17, 18, 19, 20]. In addition, advanced memory modules such as Hybrid Memory Cube (HMC)[21], High Bandwidth Memory (HBM) [22] and Bandwidth Engine (BE2)[23] have been developed by major memory vendors and made their commercial success. For instance, HMC that stacks multiple DRAM dies on top of a CMOS logic layer using through-silicon-via (TSV) technology effectively addressed the previous limitations of implementing NMP. HMC not only provides much better random access per-

ISPD '17, March 19–22, 2017, Portland, OR, USA.
© 2017 ACM. ISBN 978-1-4503-4696-2/17/03. . . $15.00
DOI: http://dx.doi.org/3036669.3038242

formance compared to traditional DDR DRAM due to its higher memory-level parallelism [2], but also supports near-memory operations, such as read-modify-write, locking, etc., on the base logic layer, making it possible for accelerating these operations near memory.

At UW-Madison, our research group was trying to understand what an NMP computer architecture might entail by combining the flexibility of modern FPGA with emerging memory module, i.e. HMC. Our initial efforts are on understanding the needs of the driving applications for HMC and developing collaborative software/ hardware techniques for efficient algorithmic mapping. In particular, we demonstrated the very first near-memory graph processing system on a real FPGA-HMC platform based on software/hardware co-design and co-optimization. The work aimed to tackle a challenging problem in processing large-scale sparse graphs, which have been broadly applied in a wide range of applications from machine learning to social science but are inherently difficult to process efficiently. It is not only due to their large memory footprint, but also that most graph algorithms entail memory access patterns with poor locality and a low compute-to-memory access ratio. To address these challenges, we leveraged the exceptional random access performance of HMC technology combined with the flexibility and efficiency of FPGA. A series of innovations were applied, including new data structure/algorithm and a platform-aware graph processing architecture. Our implementation achieved 166 million edges traversed per second (MTEPS) using GRAPH500 benchmark on a random graph with a scale of 25 and an edge factor of 16, which significantly outperforms CPU and other FPGA-based graph processors.

In another project, we tackled the challenge from a different angle. In particular, we demonstrated a high-performance near-memory OpenCL-based FPGA accelerator for deep learning. We applied a combination of theoretical and experimental approaches. Based on a comprehensive analysis, we identified that the key performance bottleneck is the on-chip memory bandwidth, largely due to the scarce memory resources in modern FPGA and the memory duplication policy in current OpenCL execution model. We proposed a new kernel design to effectively address such limitation and achieved substantially improved memory utilization, which further results in a balanced data-flow between computation, on-chip, and off-chip memory access. We implemented our design on an Altera Arria 10 GX1150 board and achieved 866 Gop/s floating point performance at 370MHz working frequency and 1.79 Top/s 16-bit fixed-point performance at 385MHz. To the best of our knowledge, our implementation achieves the best power efficiency and performance density compared to existing work.

3. IN-MEMORY PROCESSING

In-memory processing (IMP), as the second category of PIM, grew out of NMP from *processing in proximity of memory* to *processing inside memory* which seamlessly embeds computation in memory array, as depicted in Figure 2. As the compute units become more tightly coupled with memory, one can exploit more fine-grained parallelism for better performance and energy efficiency. In this section, we present the enabling technology for IMP and the exploratory IMP architectures.

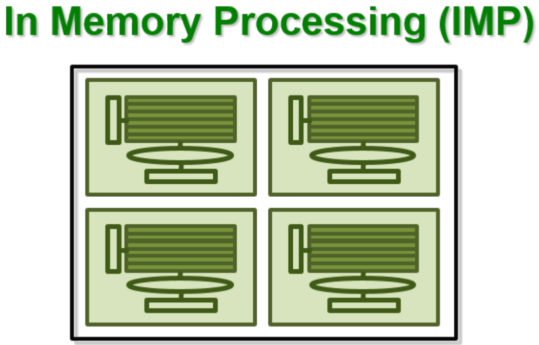

Figure 2: Conceptual diagram of In-Memory Processing (IMP). Compute units are seamlessly embedded into memory array to better exploit the internal memory bandwidth.

Technology: The last decade has seen significant progress in emerging nonvolatile memory technologies (NVMs) including Spin Torque Transfer RAM (STT RAM)[24], phase change memory (PCM)[25] and resistive RAM (RRAM)[26]. Until now, the key industry players have all demonstrated Gb-scale capacity in advanced technology nodes, including 1Gb PCM at 45nm by Micron [27], 8Gb PCM at 20nm by Samsung [28], 32Gb RRAM at 24nm by Toshiba/Sandisk [29], 16Gb conductive bridge (CBRAM, a special type of RRAM) at 27nm by Micron/Sony [30], and most recently 128Gb 3D XPoint technology by Micron/Intel [31].

Even with successful commercialization, the insertion of these technologies to exiting computer systems as a direct drop-in replacement turns out not being effective. The fundamental reasons for that are 1) Technically, the inherent nature of these technologies does not align well with either main memory or persistent storage in terms of cost-per-bit, latency, power, endurance, and retention. 2) Economically, besides more investment to existing memory manufacturing facilities for producing these new technologies, it is difficult to convince end-users to switch to a new technology as long as they can still use DRAM or Flash for the same purpose, unless significant benefits are provided. Therefore, it is challenging for any of the emerging NVMs to take over the dominant mature market of DRAM or Flash. However, we envision that to enable a wide adoption of these NVM technologies, a potentially viable path is to explore non-traditional usage models or new paradigms beyond traditional memory applications, for instance, IMP. We believe that the emerging NVMs will become an enabling technology for IMP.

Architecture: The exploratory IMP architectures reported in literature can be further divided into the several types: **1)** One type is to utilize the inherent dot-product capability of the crossbar structure to accelerate matrix multiplication, which is a key computational kernel in a wide array of applications including deep learning, optimization, etc. Representative work includes PRIME [32], ISAAC [8], and memristive boltzmann machine [33]. By augmenting RRAM crossbar design with various digital or analog circuits in the periphery, these architectures can realize different accelerator functions that are built atop matrix multiplication. **2)** Another type is to implement a neuromorphic system, which exploits the analog nature of NVM array to implement synaptic network in order to mimic the fuzzy, fault-tolerant and stochastic computation of the human brain, without sacrificing its space or energy efficiency

[8, 32, 34, 35]. **3)** The third type is associative processor (AP), also known as nonvolatile Content addressable memory (nv-CAM) or Ternary content addressable memory (nv-TCAM), which supports associative search to locate data records by content rather than address. We demonstrated the very first large-scale PCM TCAM chip and prove the feasibility of implementing in-memory processing using emerging NVM in a cost-effective manner. Other representative work includes RRAM-based TCAM [36], AC-DIMM [37], and RRAM-based associative processor [38]. **4)** The fourth type is reconfigurable architecture (RA). Representative work includes nonvolatile field programmable gate array (nv-FPGA [39, 40]) and reconfigurable in-memory computing architecture that combines the best advantages of TCAM and FPGA [41].

Both AP (Type-3) and RA (Type-4) show great promise in implementing the concept of in-memory processing without necessarily incurring high cost. Specifically, they do not need expensive mixed-signal circuits (A/D, D/A) as Type-1 and Type-2 and thus, their adoption barrier is lower than Type-1 and Type-2. However, all of these types need to address a common challenge in operational robustness due to the limited ON/OFF resistance ratio of NVM technologies (except for CBRAM), which can be mitigated by advanced material engineering [42], cell design [43, 44, 45], and coding technique [46].

Among all the work, we would like to specifically highlight an interesting reconfigurable in-memory computing architecture (Type-4) developed by us. It shares some similarities to FPGA in morphable data-flow architecture but also radically differs from it by providing: 1) flexible on-chip storage, 2) flexible routing resources, and 3) enhanced hardware security. For the first time, it exploits a continuum of IMP capabilities across the whole spectrum, ranging from 0% (pure data storage) to 100% (pure compute engine), or intermediate states in between (partial storage and partial computation). Such superior programmability blurs the boundary between computation and storage. We believe it may open up rich research opportunities in driving new reconfigurable architecture, design tools, and developing new data-intensive applications, which were not generally considered to be suitable for FPGA-like accelerations.

4. OTHER CHALLENGES

PIM, including both NMP and IMP, offers a promising approach to overcome the challenges posed by emerging data-intensive applications. In our view, NMP has a relatively low adoption barrier than IMP, as there is no need to change the internal memory architecture, whereas IMP fully exploits the internal memory bandwidth to achieve more parallelism.

To make NMP or IMP practically viable, there are other challenges that need to be addressed, including virtual memory support to ensure a unified address space, memory/cache coherence, fault tolerance, security and privacy, thermal and power constraints, compatibility with modern programming models, etc.. All of them will require collaborative efforts between technologies, IC designers and system engineers.

5. REFERENCES

[1] Ling Liu. Computing infrastructure for big data processing. volume 7, pages 165–170, 2013.

[2] Wm. A. Wulf and Sally A. McKee. Hitting the memory wall: Implications of the obvious. *SIGARCH Comput. Archit. News*, 23(1):20–24, March 1995.

[3] T. Kuroda. Low-power, high-speed cmos vlsi design. pages 310–315, 2002.

[4] Peter M. Kogge. Execube-a new architecture for scaleable mpps. In *Proceedings of the 1994 International Conference on Parallel Processing - Volume 01*, ICPP '94, pages 77–84. IEEE Computer Society, 1994.

[5] David Patterson, Thomas Anderson, Neal Cardwell, et al. A case for intelligent ram. *IEEE Micro*, 17(2):34–44, March 1997.

[6] D. Patterson, T. Anderson, N. Cardwell, et al. Intelligent ram (iram): chips that remember and compute. In *1997 IEEE International Solids-State Circuits Conference. Digest of Technical Papers*, pages 224–225, Feb 1997.

[7] Mary Hall, Peter Kogge, Jeff Koller, et al. Mapping irregular applications to diva, a pim-based data-intensive architecture. In *Proceedings of the 1999 ACM/IEEE Conference on Supercomputing*, SC '99. ACM, 1999.

[8] J. Torrellas. Flexram: Toward an advanced intelligent memory system: A retrospective paper. In *2012 IEEE 30th International Conference on Computer Design (ICCD)*, pages 3–4, Sept 2012.

[9] D. G. Elliott, M. Stumm, W. M. Snelgrove, et al. Computational ram: implementing processors in memory. *IEEE Design Test of Computers*, 16(1):32–41, Jan 1999.

[10] K. Mai, T. Paaske, N. Jayasena, et al. Smart memories: a modular reconfigurable architecture. In *Proceedings of 27th International Symposium on Computer Architecture (IEEE Cat. No.RS00201)*, pages 161–171, June 2000.

[11] M. G. Farooq, T. L. Graves-Abe, W. F. Landers, et al. 3d copper tsv integration, testing and reliability. In *2011 International Electron Devices Meeting*, pages 7.1.1–7.1.4, Dec 2011.

[12] Y. Liu, W. Luk, and D. Friedman. A compact low-power 3d i/o in 45nm cmos. In *2012 IEEE International Solid-State Circuits Conference*, pages 142–144, Feb 2012.

[13] Antonis Papanikolaou, Dimitrios Soudris, and Riko Radojcic. *Three Dimensional System Integration: IC Stacking Process and Design*. Springer Publishing Company, Incorporated, 1st edition, 2010.

[14] R. Balasubramonian, J. Chang, T. Manning, et al. Near-data processing: Insights from a micro-46 workshop. *IEEE Micro*, 34(4):36–42, July 2014.

[15] Seth H Pugsley, Jeffrey Jestes, Huihui Zhang, et al. Ndc: Analyzing the impact of 3d-stacked memory+ logic devices on mapreduce workloads. In *Performance Analysis of Systems and Software (ISPASS), 2014 IEEE International Symposium on*, pages 190–200. IEEE, 2014.

[16] M. Wordeman, J. Silberman, G. Maier, et al. A 3d system prototype of an edram cache stacked over processor-like logic using through-silicon vias. In *2012 IEEE International Solid-State Circuits Conference*, pages 186–187, Feb 2012.

[17] Q. Zhu, B. Akin, H. E. Sumbul, et al. A 3d-stacked

logic-in-memory accelerator for application-specific data intensive computing. In *2013 IEEE International 3D Systems Integration Conference (3DIC)*, pages 1–7, Oct 2013.

[18] Q. Zhu, T. Graf, H. E. Sumbul, et al. Accelerating sparse matrix-matrix multiplication with 3d-stacked logic-in-memory hardware. In *2013 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–6, Sept 2013.

[19] V. Seshadri, K. Hsieh, A. Boroum, et al. Fast bulk bitwise and and or in dram. *IEEE Computer Architecture Letters*, 14(2):127–131, July 2015.

[20] Vivek Seshadri, Yoongu Kim, Chris Fallin, et al. Rowclone: Fast and energy-efficient in-dram bulk data copy and initialization. In *Proceedings of the 46th Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO-46, pages 185–197. ACM, 2013.

[21] J Thomas Pawlowski. Hybrid memory cube (hmc). In *IEEE Hot Chips*, 2011.

[22] D. U. Lee, K. W. Kim, K. W. Kim, et al. 25.2 a 1.2v 8gb 8-channel 128gb/s high-bandwidth memory (hbm) stacked dram with effective microbump i/o test methods using 29nm process and tsv. In *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pages 432–433, Feb 2014.

[23] M. J. Miller. Bandwidth engine 2; serial memory chip breaks 2 billion accesses/sec. In *2011 IEEE Hot Chips 23 Symposium (HCS)*, pages 1–23, Aug 2011.

[24] Weisheng Zhao et al. Spin transfer torque (stt)-mram–based runtime reconfiguration fpga circuit. *TECS*, 9(2):14, 2009.

[25] Benjamin C Lee et al. Architecting phase change memory as a scalable dram alternative. In *ACM SIGARCH Computer Architecture News*, volume 37, pages 2–13. ACM, 2009.

[26] H-S Philip Wong et al. Metal–oxide rram. *Proceedings of the IEEE*, 100(6):1951–1970, 2012.

[27] C. Villa, D. Mills, G. Barkley, et al. A 45nm 1gb 1.8v phase-change memory. In *2010 IEEE International Solid-State Circuits Conference - (ISSCC)*, pages 270–271, Feb 2010.

[28] Y. Choi, I. Song, M. H. Park, et al. A 20nm 1.8v 8gb pram with 40mb/s program bandwidth. In *2012 IEEE International Solid-State Circuits Conference*, pages 46–48, Feb 2012.

[29] T. Y. Liu, T. H. Yan, R. Scheuerlein, et al. A 130.7mm2 2-layer 32gb reram memory device in 24nm technology. In *2013 IEEE International Solid-State Circuits Conference Digest of Technical Papers*, pages 210–211, Feb 2013.

[30] M. Adams. 2015 winter analyst conference, 2 2015. Micron Technology, Inc.

[31] Micron Technology, Inc. Breakthrough Nonvolatile Memory Technology. http://www.micron.com/about/innovations/3d-xpoint-technology. Accessed: 2015-10-30.

[32] Natalie Enright Jerger, Li-Shiuan Peh, and Mikko Lipasti. Virtual circuit tree multicasting: A case for on-chip hardware multicast support. volume 36, pages 229–240. ACM, June 2008.

[33] G. Khodabandehloo, M. Mirhassani, and M. Ahmadi. Analog implementation of a novel resistive-type sigmoidal neuron. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 20(4):750–754, April 2012.

[34] Yu Wang, Tianqi Tang, Lixue Xia, et al. Energy efficient rram spiking neural network for real time classification. In *Proceedings of the 25th Edition on Great Lakes Symposium on VLSI*, GLSVLSI '15, pages 189–194. ACM, 2015.

[35] Chenchen Liu, Bonan Yan, Chaofei Yang, et al. A spiking neuromorphic design with resistive crossbar. In *Proceedings of the 52Nd Annual Design Automation Conference*, DAC '15, pages 14:1–14:6. ACM, 2015.

[36] Qing Guo, Xiaochen Guo, Yuxin Bai, et al. A resistive tcam accelerator for data-intensive computing. In *Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO-44, pages 339–350. ACM, 2011.

[37] Qing Guo, Xiaochen Guo, Ravi Patel, et al. Ac-dimm: associative computing with stt-mram. In *ACM SIGARCH Computer Architecture News*, volume 41, pages 189–200. ACM, 2013.

[38] L. Yavits, S. Kvatinsky, A. Morad, et al. Resistive associative processor. *IEEE Computer Architecture Letters*, 14(2):148–151, July 2015.

[39] Y. Tsuji, X. Bai, A. Morioka, et al. A 2x logic density programmable logic array using atom switch fully implemented with logic transistors at 40nm-node and beyond. In *2016 IEEE Symposium on VLSI Circuits (VLSI-Circuits)*, pages 1–2, June 2016.

[40] Jason Cong and Bingjun Xiao. Fpga-rpi: A novel fpga architecture with rram-based programmable interconnects. *IEEE Trans. Very Large Scale Integr. Syst.*, 22(4):864–877, April 2014.

[41] Yue Zha and Jing Li. Reconfigurable in-memory computing with resistive memory crossbar. In *Proceedings of the 35th International Conference on Computer-Aided Design*, page 120. ACM, 2016.

[42] M. J. Lee, C. B. Lee, S. Kim, et al. Stack friendly all-oxide 3d rram using gainzno peripheral tft realized over glass substrates. In *2008 IEEE International Electron Devices Meeting*, pages 1–4, Dec 2008.

[43] S. H. Jo, T. Kumar, S. Narayanan, et al. Cross-point resistive ram based on field-assisted superlinear threshold selector. *IEEE Transactions on Electron Devices*, 62(11):3477–3481, Nov 2015.

[44] J. Zhou, K. H. Kim, and W. Lu. Crossbar rram arrays: Selector device requirements during read operation. *IEEE Transactions on Electron Devices*, 61(5):1369–1376, May 2014.

[45] S. H. Jo, T. Kumar, C. Zitlaw, et al. Self-limited rram with on/off resistance ratio amplification. In *2015 Symposium on VLSI Technology (VLSI Technology)*, pages T128–T129, June 2015.

[46] J. Li, R. K. Montoye, M. Ishii, et al. 1 mb 0.41 um2 2t-2r cell nonvolatile tcam with two-bit encoding and clocked self-referenced sensing. *IEEE Journal of Solid-State Circuits*, 49(4):896–907, April 2014.