Abraham Mitchell

GRSD 51103

October 20, 2024

1. BRAMAC: Compute-in-BRAM Architectures for Multiply-Accumulate on FPGAs
Y. Chen and M. S. Abdelfattah, "BRAMAC: Compute-in-BRAM architectures for multiply-accumulate on FPGAs," in *Proc. IEEE 31st Annu. Int. Symp. Field-Programmable Custom Comput. Mach.*, 2023, pp. 1–10.

This paper introduces BRAMAC, a novel architecture for multiply-accumulate (MAC) operations directly within the BRAMs of FPGAs. The authors propose two variants, BRAMAC-2SA and BRAMAC-1DA, which allow BRAMs to perform both storage and computation functions simultaneously. The approach aims to improve FPGA throughput, particularly in deep neural network (DNN) inference, by leveraging both bit-serial and bit-parallel computations within a dummy BRAM array.

The paper is highly relevant to my research because it tackles the challenge of underutilized on-chip memory (BRAM) and demonstrates how this memory can be exploited for computational tasks. The methods proposed align well with focusing on maximizing FPGA resource utilization, particularly for CNNs.

This work provides actionable insights for my ongoing FPGA research. The hybrid dataflow and tiling strategies discussed may help address similar bottlenecks in machine learning accelerators. The open-source code on the author's GitHub repository ensures reproducibility, which could be useful for benchmarking experiments based on BRAMAC.

2. CoMeFa: Compute-In-Memory Blocks for FPGAs
A. Arora, T. Anand, et al., "CoMeFa: Compute-In-Memory blocks for FPGAs," in *Proc. IEEE Int. Symp. Field-Programmable Custom Comput. Mach.*, 2022, pp. 1–8.

The CoMeFa paper proposes enhancing FPGA BRAMs by embedding programmable processing elements (PEs) within them to form compute-in-memory (CIM) blocks. These blocks support bit-serial computation, maximizing parallelism for deep learning tasks and reducing data movement across the FPGA. The architecture introduces two variants: CoMeFa-D (delay-optimized) and CoMeFa-A (area-optimized), providing flexibility based on the use case.

This work is directly aligned with my research interests in FPGAs, more specifically in processor-in-memory (PIM) architectures and memory-efficient computing. The authors provide an innovative approach to minimizing routing delays by performing computations within memory units.

The CoMeFa framework offers practical solutions that I could utilize in reference to memory width reconfiguration. Having two distinct architectures that balance power vs area efficiency is very useful to have as FPGAs are often size limited and larger components are also often difficult to implement into working designs correctly.

3. Processing in Memory: The Tipping Point
P. Radojković, P. Carpenter, P. Esmaili-Dokht, et al., "Processing in Memory: The tipping point," ETP4HPC White Paper, 2021.

This white paper explores the growing importance and technological readiness of Processing-in-Memory (PIM) architectures. It highlights how PIM overcomes the limitations of the von Neumann bottleneck, making it suitable for high-performance computing (HPC) and AI applications. The paper discusses the challenges of building a PIM ecosystem, including software development, data placement, and performance benchmarks. It also outlines industrial prototypes and real-world use cases to illustrate the maturity of PIM technology.

The paper provides a comprehensive overview of PIM technology, which is very useful for filling in any knowledge gaps I may have for this highly specialized field. Its insights into the role of PIM in AI, big data, and HPC make it relevant for optimizing memory bandwidth and on-chip computation. I believe the focus on the European research ecosystem adds credibility to the paper by showcasing ongoing efforts to push PIM technology into production.

This resource will be valuable for my future research, particularly in understanding the broader trends in PIM adoption and benchmarking methodologies. Its exploration of challenges like data coherence and application profiling can be used for finding solutions in FPGA memory utilization problems.

4. In-Memory Intelligence
T. Finkbeiner, G. Hush, P. Lea, J. Leidel, and T. Manning, "In-Memory Intelligence," *IEEE Micro*, vol. 37, no. 4, pp. 20–30, Aug. 2017, doi: 10.1109/MM.2017.123456.

This paper describes the Micron In-Memory Intelligence (IMI) architecture, which places single-instruction multiple-data (SIMD) computing elements directly on DRAM arrays to reduce data movement. IMI targets many areas, such as big-data applications in vision, neural networks, and encryption. The IMI authors combines bit-serial computation, a custom instruction set, and a very long instruction word(VLIW control unit to support efficient parallelism. The work emphasizes the advantages of PIM in terms of performance and energy efficiency.

The paper offers insights onto an interesting, albeit outdated, PIM architecture. The technical details on bit-serial processing, custom ISAs, and memory management strategies, show that designers of PIM architectures are coming to a general consensus on the way to tackle common problems of these architectures. The IMI's focus on reducing data transfers is particularly interesting for BRAM utilization.

This paper provides a lot of information on architectural ideas and tools for FPGA PIM architectures. However, the paper is now fairly old, and a lot of the ideas either introduced in this paper or that were relevantly new when the paper was published have now been improved upon. While being helpful for new researchers learning about the field and what techniques have been used but is not a good resource for understanding the current cutting-edge technologies in PIM.

5. Accelerating Neural Network Inference with Processing-in-DRAM: From the Edge to the Cloud

G. F. Oliveira, J. Gómez-Luna, S. Ghose, A. Boroumand, and O. Mutlu, "Accelerating neural network inference with processing-in-DRAM: From the edge to the cloud," *IEEE Micro*, vol. 42, no. 1, pp. 36–47, 2022

This paper is an evaluation on three state-of-the-art PIM architectures used for edge and clod computing. These architectures are UPMEM, Mensa, and SIMDRAM, each with its own benefits and short comings. UPMEM is easily scalable but not the fastest, MENSA, a 3-d based processor, is customizable but hard to implement, and SIMDRAM is fast but has limited applications. The paper compares the three PIM architectures to more general-purpose architectures for CPUs, GPUs, and TPUs to highlight the benefits when using each for the tasks they were designed for.

This paper is very relevant for processor-in-memory and FPGA architectures. It not only provides insights into PIM solutions but also discusses the challenges of memory-bound workloads, such as CNNs and LSTMs. It is also useful for my lab because it gives a clear breakdown of the pros and cons of each of these architectures. While nothing I do will be exactly the same as any of these, reading this paper has helped me identify areas that will need improvement on future works.

6. FABulous: An Embedded FPGA Framework

D. Koch, N. Dao, B. Healy, J. Yu, and A. Attwood, "FABulous: An embedded FPGA framework," in *Proc. ACM/SIGDA Int. Symp. Field-Programmable Gate Arrays*, 2021, pp. 50–59.

This paper presents FABulous, an open-source framework for designing embedded FPGAs (eFPGAs). FABulous aims to make the integration of FPGA fabrics into ASICs easier. The framework supports various customization levels, including logic, arithmetic, and memory tiles. The architecture emphasizes the ability to optimize area, performance, and power while maintaining portability across different process nodes. The authors demonstrate FABulous with a case study that integrates a RISC-V core with an eFPGA fabric to extend instruction sets.

FABulous is a FPGA tool that has been gaining traction in the field for its ability to simplify tile design. The integration of open-source tools, such as Yosys and nextpnr, allows FABulous to do highly specialized creation and analysis tasks. Furthermore, the study's focus on embedding reconfigurable hardware into ASICs provides new avenues for exploring the integration of machine learning accelerators.

FABulous is a tool that I have actively been using this semester for continuing upon a PIM architecture developed by another student on my lab who recently defended his PhD. The paper itself is not the best resource for understanding how to use the tool, but it does well in explaining why it is helpful. In all honestly, using FABulous has an extreme learning curve, even for someone with an undergraduate background in the field, but after becoming familiar with it, the benefits become apparent.

7. Survey on Processing-in-Memory Techniques: Advances and Challenges

K. Asifuzzaman, N. R. Miniskar, A. R. Young, F. Liu, and J. S. Vetter, "Survey on processing-in-memory techniques: Advances and challenges," *Memories: Materials, Devices, Circuits, and Systems*, vol. 1, no. 1, pp. 1–20, 2023

This survey, supported by Oak Ridge National Laboratory and the US Department of Engery, provides a comprehensive analysis of recent advances in processing-in-memory (PIM) techniques. It discusses emerging memory technologies such as DRAM, ReRAM, and STT-MRAM, and classifies PIM architectures based on whether processing occurs at the data array, row buffer, or near memory. Additionally, the survey explores challenges such as workload analysis, memory mapping, runtime support, and coherency issues, offering insights into how PIM can complement or replace conventional von Neumann architectures.

This paper is a great introduction for new researchers into the PIM field as it not only discusses memory-centric computing but also touches on emerging technologies and architectural models. The detailed categorization of PIM designs and evaluation methodologies makes it a useful resource for benchmarking work.

This survey will serve as a foundational reference for my research. Its exploration of different architectural models and performance benchmarks aligns is a needed relief from other highly specific work on optimization solutions for BRAM usage within FPGAs. The insights into OS support and application partitioning will be particularly useful as I continue in my research.

8. FPGA Architecture: Principles and Progression

A. Boutros and V. Betz, "FPGA architecture: Principles and progression," *IEEE Circuits Syst. Mag.*, vol. 21, no. 2, pp. 45–60, 2021

This paper offers a comprehensive review of FPGA architecture evolution over the past 30 years. It covers the fundamental design principles of FPGAs, including programmable logic blocks, routing architectures, DSP blocks, and memory subsystems. The authors highlight key advancements such as fracturable lookup tables (LUTs) and configurable block RAMs (BRAMs), emphasizing the trade-offs between flexibility, performance, and area. The paper also discusses emerging trends like heterogeneous multi-die FPGA systems, the increasing integration of hard blocks, and the role of CAD tools in FPGA architecture design and optimization.

This paper is useful for its focus on memory reconfiguration and FPGA resource optimization. The detailed analysis of BRAMs, DSP blocks, and programmable logic offers insights into how on-chip resources can be reconfigured to achieve better efficiency. The exploration of trade-offs between different FPGA architectures directly supports efforts to improve accelerators and on-chip memory utilization.

This paper will serve as a valuable reference for my ongoing research on FPGA-based architectures. It offers a clear roadmap of the challenges and innovations in FPGA design, which can guide exploration of memory configurations and PIM systems. The insights on CAD tools will also help better leverage tools like Synopsys in synthesis and optimization processes.