RETROSPECTIVE: Sparse ReRAM Engine: Joint Exploration of Activation and Weight Sparsity in Compressed Neural Networks

Tzu-Hsien Yang*, Hsiang-Yun Cheng†, Chia-Lin Yang*, I-Ching Tseng*,
Han-Wen Hu‡, Hung-Sheng Chang‡, and Hsiang-Pang Li‡
*National Taiwan University, †Academia Sinica, ‡Macronix International Co., Ltd.

I. Background and Motivation

When we began this work in 2018, deep neural networks (DNNs) had gained significant popularity, and researchers were actively exploring innovative architectural designs to achieve energy efficiency. To address the intensive computing and memory demands of DNNs, several studies proposed computing-in-memory (CIM) designs to accelerate DNN inference. Particularly, in ISCA 2016, two papers [O9][O40] first introduced this idea to the computer architecture community. These pioneering studies suggested leveraging the inherent matrix-vector multiplication computing capability of emerging memristor devices, such as resistive random access memory (ReRAM), to overcome the memory wall challenge and enable highly parallel computations. The revisit of the CIM concept [1], along with its successful combination with AI workloads, has opened up a prominent research domain.

Despite memristor-based CIM showing promising potential for AI acceleration, there remain challenges to overcome in practice. First, the non-ideality of memristor devices makes it difficult for an analog-to-digital converter (ADC) to accurately read out the matrix-vector multiplication results on bitlines. Our analysis, conducted through DL-RSIM [O31] (a simulation framework we developed), revealed that the inference accuracy drops considerably when too many wordlines are activated concurrently due to the accumulated per-cell current deviation (e.g., 16 wordlines, even with technology advances enabling cell variation reduction). Second, deploying a high-frequency ADC with high bit-resolution to enable the entire memristor crossbar array to operate in one cycle leads to significant power and area overhead.

Considering the non-ideality of memristor devices and the overheads associated with ADC, practical matrix-vector multiplication must proceed at a smaller granularity in memristor-based CIM. This was demonstrated by the state-of-the-art ReRAM macro designed for DNN acceleration at that time, which operated only a 9x8 sub-region in a crossbar array per cycle [O6]. However, when we developed our idea, existing designs published in the computer architecture community overlooked these hardware constraints and assumed that the entire crossbar array could be operated in a single cycle. We were the first to recognize that it is more practical to operate only a smaller section of a crossbar array in a single cycle, and we defined this small section that can be operated per cycle as the Operation Unit (OU).

Although the OU-based architecture is more practical, it is not necessarily better than the over-idealized one that operates an entire crossbar array in a single cycle when considering the impact on performance. In comparison to the over-idealized architecture, an OU-based design benefits from better accuracy and lower resolution requirements on ADCs, as fewer wordlines are activated per cycle. However, while an OU-based design could potentially achieve a shorter cycle time by adopting lower-resolution ADCs with faster sensing speed, it also requires more cycles to complete the same amount of computations due to limited wordline and bitline-level parallelism. Consequently, the significantly increased number of cycles overshadows the advantage of a shorter cycle time, making an OU-based design likely to deliver lower performance than an over-idealized counterpart if no optimization mechanism is applied to reduce computations.

The comparison between the OU-based design and the over-idealized counterpart involves a tradeoff between accuracy and performance. This motivates us to explore whether we can resolve this dilemma through architectural design, and the collaboration between academics and industry enables us to derive a practical yet efficient solution. During our investigation, we discovered that the OU-based architecture naturally allows the exploitation of fine-grained sparsity to skip more redundant computations, as each OU operates independently. By effectively harnessing the fine-grained sparsity offered by the OU-based design, it becomes possible to achieve a practical design with satisfactory inference accuracy while attaining comparable performance and energy efficiency with the over-idealized counterpart. To accomplish this objective, we propose the Sparse ReRAM Engine, which stands as one of the pioneering works that avoids overlooking the impact of hardware constraints and aims to develop a practical yet energy-efficient memristor-based CIM design.

II. Sparse ReRAM Engine

The Sparse ReRAM Engine is the first practical memristor-based CIM design that jointly exploits weight and activation sparsity to enable energy-efficient DNN inference while mitigating accuracy loss caused by non-ideal devices/circuits. Our key findings and innovations include the following:

Practical OU-based CIM. Unlike prior over-idealized designs that overlook the overhead of ADC and the accumulated effect of per-cell current deviation on inference accuracy, we were the first to recognize the practicality of operating only a smaller section of a crossbar array (OU) instead of the entire array in one cycle. Additionally, we emphasized the need for architectural solutions to reduce computations in OU-based designs, overcoming the limitation of computation parallelism that hinders their potential to achieve comparable or even better energy efficiency than the over-idealized counterpart.

OU-level fine-grained sparsity exploitation. In an over-idealized design, redundant computations can only be skipped when the entire wordline/bitline cells contain zeros or when the input bits to the crossbar array are all zeros in the same cycle due to the tightly coupled crossbar structure. However, we discovered that the OU-based architecture naturally enables the exploitation of fine-grained sparsity, as each OU is operated independently. This characteristic allows for the skipping of more zeros, creating new performance improvement opportunities for OU-based designs.

Joint exploitation of weight and activation sparsity. To effectively take advantage of the fine-grained sparsity offered by the OU-based design, we proposed an innovative method to jointly exploit weight and activation sparsity. In an OU-based design, weight compression can be performed in either the row or column dimension. For activation compression, instead of simply skipping an OU computation when inputs to all the wordlines of the OU are zeros, we proposed forming an OU unit at runtime to activate non-contiguous wordlines with non-zero input values in the same cycle. This method works perfectly with row-wise weight compression, enabling the simultaneous exploration of weight and activation sparsity.

III. Looking Back

The design of the Sparse ReRAM Engine has inspired a new direction in neural network compression, with the goal of maximizing the energy efficiency improvements achievable through fine-grained operations in practical designs. Several studies have been conducted to develop pruning algorithms that are co-designed with the underlying practical architecture, allowing for better exploitation of fine-grained sparsity. One such study by Yuan et al. [2] introduced a framework for generating memristor-based CIM-friendly DNN models. They employed a fragment polarization technique to address the issue of signed weight representation and combined it with structured pruning and quantization methods. In a similar vein, Yang et al. [3] utilized reinforcement learning to automatically determine the pruning policy and quantization bit-width for the OU-based CIM architecture. Besides eliminating computations with zero-valued weights and inputs, researchers have also explored opportunities for sharing computed results with repetitive weight [4] and input patterns [5] at OU or finer levels, aiming to reduce redundant computations.

As one of the pioneering works aiming to develop a practical yet energy-efficient memristor-based CIM design, the Sparse ReRAM Engine has garnered significant attention from the computer architecture community, emphasizing the importance of addressing the negative impacts of non-ideal devices/circuits and ADC overheads. Since its publication in 2019, researchers have proposed various methods to overcome these hardware constraints. One potential approach is redesigning dataflow and weight encoding. For instance, Chou et al. [6] introduced an architecture that connects each ReRAM CIM array with a buffer ReRAM array, utilizing an analog summation scheme to extend processing in the analog domain and reduce the required analog-digital conversions. Andrulis et al. [7] designed a weight encoding method enabling an adaptive architecture to use efficient low-resolution ADCs without retraining. Another strategy is employing energy-efficient circuit interfaces like time-domain interfaces with analog local buffers to reduce analog-digital conversion overheads [8].

IV. Conclusion

In the big data era, the need to efficiently process vast amounts of data drives a disruptive change in computing platform design. Bringing computation closer to data with a memory-centric approach is an attractive solution to mitigate costly data movement across memory channels. The inherent computing-in-memory capability of emerging memory technologies allows for this paradigm shift. However, challenges arise due to the non-ideality of current devices/circuits and analog-digital conversion overheads, making practical implementation difficult. The Sparse ReRAM Engine demonstrates how leveraging fine-grained sparsity in neural networks can achieve a reliable and energy-efficient practical design. We anticipate that the key design concept used in the Sparse ReRAM Engine, which tailors the system to practical hardware configurations and exploits software features to accommodate hardware constraints, will play a crucial role in accelerating the adoption of memory-centric computing in the future.

References

[1] H. S. Stone, "A logic-in-memory computer," *IEEE Transactions on Computers*, vol. C-19, no. 1, pp. 73–78, 1970.

[2] G. Yuan *et al.*, "FORMS: Fine-grained polarized ReRAM-based in-situ computation for mixed-signal DNN accelerator," in *ACM/IEEE International Symposium on Computer Architecture (ISCA)*, 2021, pp. 265–278.

[3] S. Yang *et al.*, "APQ: Automated DNN pruning and quantization for ReRAM-based accelerators," *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 9, pp. 2498–2511, 2023.

[4] Y. Zhang *et al.*, "A practical highly paralleled ReRAM-based DNN accelerator by reusing weight pattern repetitions," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 4, pp. 922–935, 2022.

[5] C.-Y. Tsai *et al.*, "RePIM: Joint exploitation of activation and weight repetitions for in-ReRAM DNN acceleration," in *ACM/IEEE Design Automation Conference (DAC)*, 2021, pp. 589–594.

[6] T. Chou *et al.*, "CASCADE: Connecting RRAMs to extend analog dataflow in an end-to-end in-memory processing paradigm," in *IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2019, p. 114–125.

[7] T. Andrulis *et al.*, "RAELLA: Reforming the arithmetic for efficient, low-resolution, and low-loss analog PIM: No retraining required!" in *ACM/IEEE International Symposium on Computer Architecture (ISCA)*, 2023.

[8] W. Li *et al.*, "TIMELY: Pushing data movements and interfaces in PIM accelerators towards local and in time domain," in *ACM/IEEE International Symposium on Computer Architecture (ISCA)*, 2020, p. 832–845.