2023 IEEE 30th International Conference on High Performance Computing, Data, and Analytics (HiPC) | 979-8-3503-8322-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/HIPC58850.2023.00020

A 118 GOPS/mm^2 3D eDRAM TensorCore Architecture for Large-scale Matrix Multiplication

Mengtian Yang*
University of Texas at Austin
mengtian.yang@utexas.edu

Yipeng Wang*
University of Texas at Austin
yipeng.wang@utexas.edu

Jaydeep P. Kulkarni
University of Texas at Austin
jaydeep@austin.utexas.edu

***Abstract*—The computational demands for recent large transformer-based language models and Neural Radiance Fields (NeRF) have rapidly increased, impacting applications like conversational AI and Mixed Reality (MR). Current accelerator architectures struggle to cope with the vast computational requirements, creating a gap with slowly growing hardware resources. This paper proposes repurposing memory components as high-density computational units, leveraging recent advancements in Back-End-Of-Line (BEOL) transistors and monolithic 3D integration techniques. An ultra-high density monolithic 3D eDRAM is presented as a reconfigurable matrix multiplication unit, co-designed with analog computation circuits, achieving energy efficiency up to 2.41 TOPS/W, performance up to 1.71 TOPS on bfloat16, and compute intensity up to 118 GOPS/mm^2. A comprehensive multi-cube(core) architecture is also devised and optimized with bit stationary tensorcore dataflow. We evaluate the proposed architecture on state-of-the-art machine learning models: NeRF and LLaMa-7B, improving the computation density by up to 6.59x and 1.12x compared with GPU and state-of-the-art vector processor designs, respectively.**

***Index Terms*—ML accelerator, matrix multiplication, monolithic 3D, eDRAM, Compute-in-memory, CIM**

I. Introduction

In recent years, the surge in the adoption of large transformer-based language models and Neural Radiance Fields (NeRF) based rendering has been closely tied to their intensive computational demands. While these tools have spearheaded breakthroughs in fields like natural language processing (NLP) and 3D reconstruction, such as conversational AI and MR [1], [2], the computational challenges they introduce cannot be overlooked. For instance, to achieve high-quality responses, models necessitate more than 175 billion parameters and a staggering computational power in the ballpark of hundreds of petaflops [3]. The sheer magnitude of these models stretches the limits of present-day accelerator architectures, underscoring the pressing need for extreme compute throughput.

Currently, most commercial products depend primarily on off-chip DRAM access and centralized matrix multiplication units. Further improvement of computation resources are limited by area and power density. Recent explorations involve implementing Single Instruction, Multiple Data (SIMD) units placed near High Bandwidth Memory (HBM) IO boundaries for machine learning workloads [4], [5], which extends the computation to off-chip. However, further exploration of increasing flexible on-chip computation units is necessary to address the rapidly changing requirements on the algorithmic side effectively.

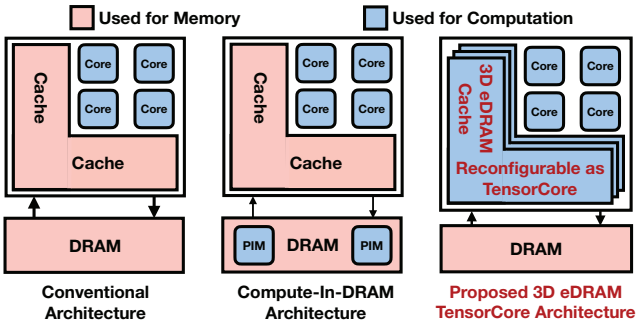

Fig. 1. Design overview of proposed 3D eDRAM TensorCore architecture.

In modern System-on-Chips (SoCs), a significant portion of the on-chip area is consumed by memory, which restricts the available area for computation, thereby limiting the chip's overall throughput potential. To address this, we propose repurposing memory components as high-density computational units that benefit from innovations in Back-End-Of-Line (BEOL) transistors and monolithic 3D integration methods. As Fig. 1 shows, Our proposal is focused on the potential of ultra-high density, monolithic-3D embedded DRAM (eDRAM) as a powerful compute engine for matrix multiplication. We aim to co-design a compact eDRAM integrated with analog compute circuits, thereby amplifying data reuse and curbing energy consumption. The emergence of monolithic 3D integration of BEOL transistors for high-density tasks further bolsters this approach. By facilitating matrix multiplication across dimensions that transcend 2D limitations, this vertical integration not only facilitates substantial data reuse benefits but also opens up new opportunities for architectural advancements.

In this paper, we first introduce the eDRAM tensor core circuit and its 3D BEOL integration. We then present a dataflow supporting INT8 and BF16 general-purpose matrix multiplication. Subsequently, we assess the advantages of this approach over the traditional processors in terms of critical metrics such as energy efficiency, performance, and compute intensity. In addition, we devise a comprehensive architecture, including cache, datapath, and DRAM access components. Moreover, we evaluate the effectiveness of the proposed ap-

*Equally contributed to this work.

2640-0316/23/$31.00 ©2023 IEEE
DOI 10.1109/HiPC58850.2023.00020

proach using state-of-the-art machine learning models such as NeRF and LLaMA-7B language model.

II. 3D eDRAM Tensor Core Circuit and Dataflow

Monolithic 3D memory technology offers higher bit-density and higher energy efficiency compared to tiled 2D memories and other 3D integration methods [6], thus meeting the density and energy efficiency needs of future ML workloads, exhibiting low write energy, and high write endurance due to many weight and activation updates (write operations) in each layer's computation. Monolithic 3D silicon SRAMs face low temperature processing challenges, incur higher transistor count and leakage current compared to other back-end integrated resistive, magnetic, and phase change memories [7]. Although commercially available, these memories exhibit poor write-endurance (10^4-10^6), and high write-energy (>10pJ), limiting their use in large scale accelerator designs requiring frequent, energy-efficient weight/activation updates.

A promising 3D memory technology suitable for extreme-scale designs is eDRAM, which employs back-end integration-compatible Indium Gallium Zinc Oxide (IGZO) transistors [8]. eDRAM features an intrinsic contention-free write mechanism, resulting in a low write energy (approximately 1pJ). Furthermore, IGZO eDRAM has shown an impressive 6-8 orders of magnitude higher write-endurance compared to other back-end integrated memories [9], [10]. Consequently, IGZO eDRAM presents a distinct value proposition characterized by low write energy, high write endurance, and remarkable bit density.

This work introduces a monolithic 3D eDRAM approach using BEOL-integrated 2T only gain-cell bitcell topology to maximize memory density. Fig. 2 illustrates our inner-product style dataflow that extends from traditional 2D Compute-in-memory (CIM) [11] dataflow. Within this framework, 3D weights are retained in the columns of the 3D eDRAM array for inner product matrix multiplication. Inputs are fed into

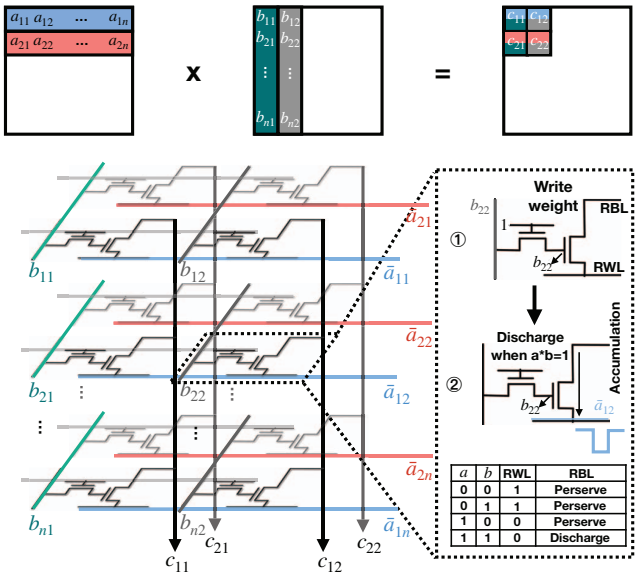

Fig. 2. Extend from 2D to 3D: inner-product style dataflow.

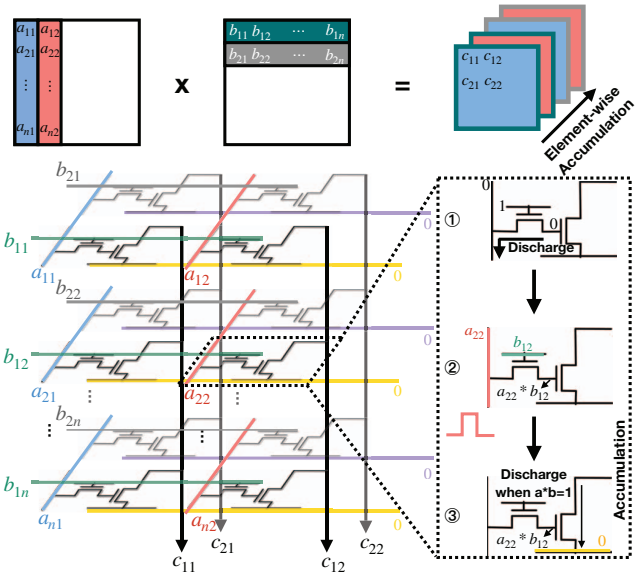

Fig. 3. Proposed 3D eDRAM TensorCore: outer-product style dataflow.

the read-wordline (RWL) systematically. On the bit cell level, WBL and WWL are routed along the X and Y axes, while RBL extends in the Z direction. The write access transistor is utilized for writing one multiplier (b_{22} in Fig. 5) into the 3D array, while the complement of another multiplier ($\bar{a}_{12}$ in Fig. 5) is fed through the RWL and performs multiplication through the read operation. The outcome of the multiplication process determines the discharging action of the read-bitline (RBL), resulting in layer-wise accumulation similar to previous models. In this operational mode, the weight remains static within the array, reducing the requirement for high input bandwidth. However, unlike 3T [12] cell designs, 2T cell designs exhibit RBL limited bitline swing due to the unselected cell sneak-path leakage current, which could impose computation accuracy issues. Additionally, a small voltage swing implies an insufficient read-sensing margin.

To surmount these challenges, we propose a higher performance outer-product dataflow which is shown in Fig. 3 when reconfiguring the 3D eDRAM cache as TensorCore. This mode of operation feeds and broadcasts 2D multipliers through the array in the X and Y directions while results accumulate in the Z direction, enabling parallel outer product matrix multiplication. Instead of performing multiplication through read operation in inner-product dataflow, this proposed dataflow uses the memory write stage for multiplication to achieve a full RBL swing, leakage-free accumulation. Meanwhile, outer-production achieves the best parallelism among result matrix elements, maximizing the utilization of all 3D RBLs. In executing multiply and accumulate (MAC) operations, WBL is initially pre-discharged, and WWL is set high to discharge all storage nodes. Concurrently, RWLs are driven to the ground, and RBL is precharged to Vcc. Single-bit multiplier values are then delivered to WBL as pulses and WWL as steady voltages. A storage node charges only when both WBL and WWL

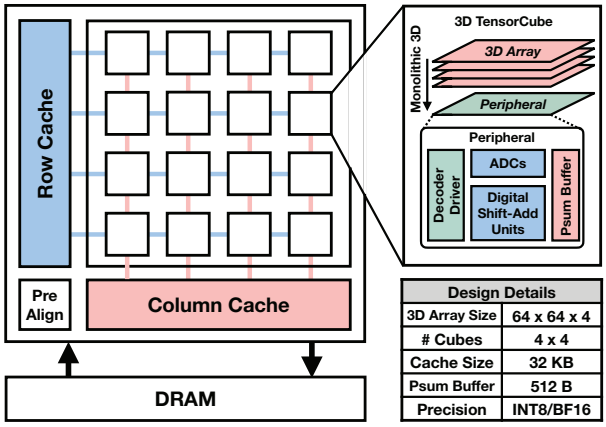

Fig. 4. Proposed 3D eDRAM TensorCore architecture floorplan, single compute cube design, and design parameters.

are high, which completes a one-bit multiplication. The read access transistor discharges the RBL when the multiplication result is '1', facilitating accumulation in the Z direction. The partial sum is then converted into the digital domain using a 2-bit ADC and sense amplifier. The accumulation phase can also be digitally designed using a set of digital adder trees to access the storage node, but this would demand using complementary P and N devices-based logic using IGZO BEOL, which is challenging to realize. Multi-bit functionality can be achieved by integrating a shift-and-add unit for each Z column. This bit-serial approach allows multipliers to be flexibly defined by digital control, easing mapping to the machine learning library.

III. 3D eDRAM TENSOR CORE ARCHITECTURE

Fig. 4 depicts the high-level architecture of the 3D eDRAM TensorCore. The complete architecture comprises 4x4 3D TensorCubes, with each cube featuring a 64x64x4 3D eDRAM computation array. A single cube is capable of performing a matrix multiplication of dimensions 64x4 by 4x64. Consequently, the entire architecture can process matrix multiplications with a tile size of 256x4 by 4x256. To present the architecture design comprehensively, we will proceed in a bottom-up manner, initially explaining the dataflow to support int8 and bfloat16, then elucidating the accumulation pattern within one matrix multiplication tile and the matrix tiling approach for large-scale matrix multiplication on the entire architecture.

A. Bit stationary dataflow

The 3D eDRAM TensorCore is capable of performing 1-bit matrix multiplication, as demonstrated in section II. To facilitate multi-bit input for both multipliers, we propose a bit-stationary dataflow for int8 and bfloat16 matrix multiplication. This proposed dataflow establishes a temporal mapping for multi-bit multiplication, breaking it down into multiple 1-bit matrix multiplications. As a result, the 3D eDRAM TensorCore becomes capable of executing multi-bit computations effectively.

1) Support for INT8: Fig. 5 illustrates the detailed bit-stationary dataflow employed in the proposed 3D eDRAM TensorCore for performing signed int8 multiply-add operations along the Z direction. To facilitate analysis, we focus on a single element in the result matrix and illustrate the computation dataflow accordingly. In Fig. 5 (a), each Z-directional layer receives two 8-bit signed integer numbers as multipliers, with the signed multiplication results accumulating into the partial sum. Fig. 5 (b) provides a temporal breakdown of the bit-stationary dataflow. The computation begins with the least significant bit (LSB) of input b, while the bit of input a is initially stationed at the LSB. As the processing of input b bits finishes, we shift the bit of input a to the next position and repeat the iteration for all bits of input b. The iteration continues until we reach the most significant bit (MSB) for both input a and b, at which point the computation finishes.

The shift-add logic requires modification to support signed multi-bit multiplication and addition. To achieve this, we incorporate the Baugh-Wooley algorithm, which efficiently handles sign bits for single multiplication, into our shift-add logic. This integration enables our logic to handle signed multiplication and addition. Fig. 5 (c) illustrates the implementation of a 4-layer signed accumulation logic for bit-stationary dataflow. The accumulated result is initially read out by an ADC and further processed based on the current bit for input a and b. If both of the current input a bit and the current input b bit are MSB (Most Significant Bit) or neither of them is MSB, we take the obtained result and pass it to the next shift logic. Otherwise, we pass $4 -$ result to the shift logic. The shift logic performs a left shift on the passed value by the sum of the current a bit and b bit, and the shifted value is then accumulated into the partial sum. Upon completion of the last accumulation for the output result, we add the final value by 1024 and flip the MSB bit to obtain the correct final accumulation result.

2) Support for BF16: To implement the in-memory accumulation of floating-point numbers, the exponents of multiplication results must be identical. We propose a matrix pre-alignment technique that aligns the input from one of the input matrices to ensure a consistent exponent for each vertical

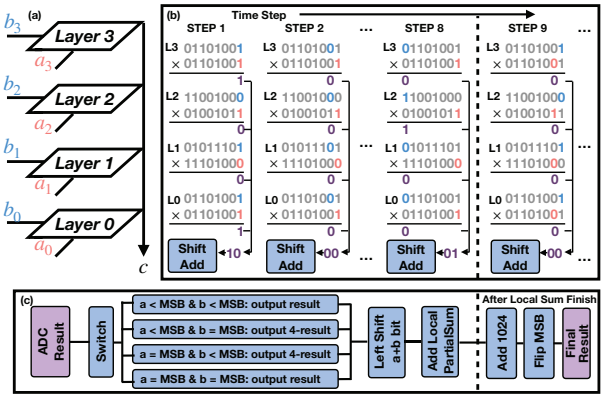

Fig. 5. Multi-bit multiplication and accumulation on proposed architecture.

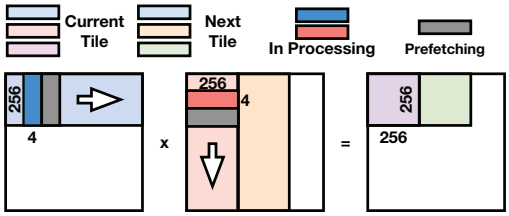

Fig. 6. Tiling pattern for large-scale matrix multiplication.

plane.

Under this scheme, during the data preparation stage, an exponent target will be identified for each vertical plane. This is usually the maximum exponent for all multiplication results within the plane. Following this, every element of one of the matrices will be subjected to a left shift, based on the target and the corresponding exponent of the element from the other matrix. Once accumulation has been completed, the result is returned to the standard floating-point format. In the case of BF 16, the exponent comprises 8 bits, with the sign and mantissa taking up the remaining 8 bits. This is compatible with the INT8 architecture that was proposed previously.

B. Tiling for large scale computation

To support large-scale matrix multiplication, which is a common operation in machine learning applications, an efficient tiling method is essential. Fig. 6 illustrates the tiling pattern employed in the 3D eDRAM TensorCore architecture. Our approach involves partitioning the computation based on an output stationary pattern. Each tile focuses on completing the computation for a single 256x256 output block. Within each tile, a subtitling technique in an outer-product style is utilized to maximize the utilization of multiplication and accumulation components while minimizing bandwidth and data-preparation overhead. During the computation in the 3D TensorCore, current subtiles are stored in the cache, while the next subtiles are pre-fetched into a separate ping-pong cache. Leveraging the bit-stationary dataflow and bit-level data reuse, the memory and cache bandwidth requirements are significantly reduced. Notably, the subtiles stored in the cache can be maintained for 64 accumulation steps before switching, allowing for fully overlapped cache fetching and yielding higher compute performance.

IV. Evaluation

Our design is assessed using Hspice simulation, under the TSMC 40nm technology. We chose three distinct benchmarks to evaluate our design effectively: a 4096x4096 general matrix multiplication (GEMM 4096), the NeRF model, and the LLaMA-7B model. In the case of the machine learning models, our primary focus is on deconstructing the layers and conducting a detailed layer-wise assessment, with exclusive attention given to the matrix-multiplication and convolutional layers. The on-chip cache latency/energy and the off-chip DRAM latency are included in the simulation based on the Cacti simulator.

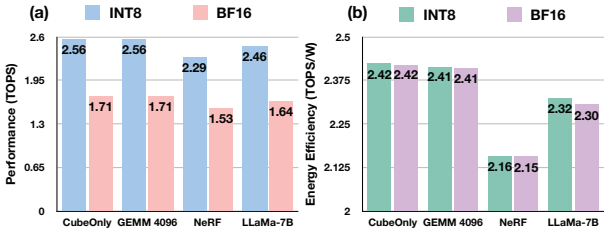

Fig. 7. (a) Simulation results of performance (TOPS). (b) Energy efficiency (TOPS/W) for the proposed design. The CubeOnly column means the maximum performance/energy efficiency for 16 cubes.

A. Architecture simulation

Fig. 7 shows the simulation results on three benchmarks for both INT8 and BF16 datatype. The CubeOnly column indicates the upper bound of performance and energy efficiency, which is determined by executing all 16 TensorCubes simultaneously and measuring the corresponding performance and efficiency. This value serves as an upper bound for this architecture, as it accounts for the full utilization of computation units without considering any cache overhead.

Based on the results shown in Fig. 7, it is evident that this architecture demonstrates exceptional proficiency in executing extensive matrix multiplications with near-optimal performance and energy efficiency. Specifically, it achieves a throughput of 2.56 (1.71) TOPS and 2.41 (2.41) TOPS/W for INT8 (BF16) precision. Furthermore, this architecture shows its ability to efficiently handle large-scale machine learning models such as the NeRF model for mixed reality and the LLaMA-7B model for language processing.

B. Dataflow comparison: inner-product and outer-product

Fig. 8 illustrates the comparison between inner-product style dataflow and outer-product style dataflow in our architecture design. The energy breakdown in Fig. 8(a)(b) reveals that the 3D eDRAM array itself exhibits a considerably higher energy consumption when operating in inner-product style.

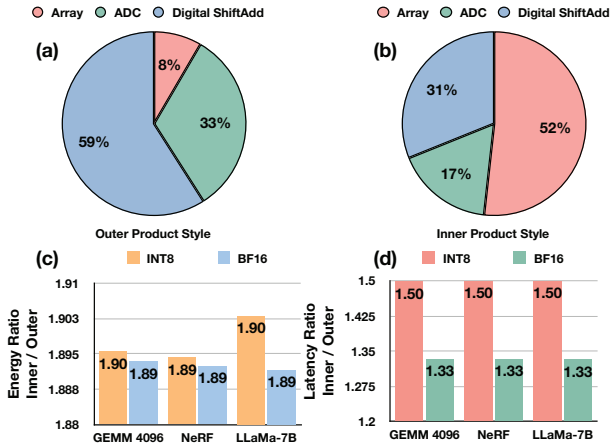

Fig. 8. Comparison results for inner-product style dataflow and outer-product style dataflow. (a) Energy breakdown for GEMM 4096 in inner-product style dataflow. (b) Energy breakdown for GEMM 4096 in outer-product style dataflow. (c) Energy comparison for three evaluated models. (d) Latency comparison for three evaluated models.

	This work	Intel Xeon	NVIDIA H100	Ara (RISCV vector coprocessor)
Technology	TSMC 40	Intel 10	TSMC 4nm	TSMC 65
Precision	INT8/BF16	INT/FP	INT/FP	INT/FP
Model	LLAMA, NeRF	LLAMA, NeRF	LLAMA, NeRF	Matrix multiplication (INT8)
Throughput (TOPS)	2.56(INT8)/ 1.71(BF16)	79.87(INT8)/ 39.94(BF16)	1600(INT8)/ 800(BF16)	0.627 (lane=4)
Average energy efficiency (TOPS/W)	2.41(INT8)/ 2.41(BF16)	0.0014(INT8)/ 0.0071(BF16)	0.0457(INT8)/ 0.0229(BF16)	2.5 (lane = 4)
Area efficiency (TOPS/mm2)	0.118(INT8)/ 0.079(BF16)	0.05(INT8)/ 0.025(BF16)	0.0179(INT8)/ 0.0098(BF16)	0.105

Fig. 9. Throughput, energy efficiency, and area efficiency comparison with CPU/GPU/Vector processor (Normalised to 40nm).

Specifically, the 3D eDRAM array accounts for 52% of the total energy consumption in inner-product style dataflow, while it only consumes 8% in outer-product style dataflow.

Fig. 8(c)(d) shows the energy/latency overhead for inner-product style dataflow vs outer-product style dataflow. Results show that inner-product style dataflow could consume up to 1.90x more energy and 1.49x more latency compared with outer-product style dataflow.

C. Compare with hardware

Fig.9 compares our result with CPU, GPU, RISCV vector coprocessor [13]. We scale technology and normalize it to 40nm to compare the energy/area efficiency. Our proposed design achieves decent energy efficiency for 2.41 TOPS/W BF16 precision and great area efficiency at 118(79) GOPS/mm^2 for BF16(INT8) precision.

V. Related Works

For previous 3D architecture works, Para-Net [14] introduces a method for improving convolutional neural network (CNN) performance by leveraging data-level parallelism on a 3D processing-in-memory (PIM) architecture. It addresses the main challenge of data movement in CNNs, resulting in improved processing time and cache efficiency. M3D-LIME [15] integrates Si-based CMOS logic, resistive random-access memory (RRAM)-based computing-in-memory (CIM), and ternary content-addressable memory (TCAM) layers in a monolithic 3D structure, and demonstrates significant energy efficiency in one-shot learning tasks. 3D-stacked Logic-in-Memory (LiM) Accelerator [16] introduced a 3D-stacked Logic-in-Memory Accelerator for sparse matrix multiplication by applying customized content addressable memory (CAM) hardware structure to exploit the inherent sparse data patterns and model the LiM based hardware accelerator layers that are stacked in between DRAM dies for the efficient sparse matrix operations.

For eDRAM-based accelerator design, eDRAM-CIM [17] illustrates the practicability for eDRAM-based CIM design by proposing the matrix multiplication accelerator with 1T1C bit cells. The 4T2C eDRAM CIM design [18] proposed a matrix-vector multiplication engine with ternary weight support and improved retention time. Gain-Cell CIM [19] presented a leakage and read bitline swing aware eDRAM CIM design based on 2T1C eDRAM bit cells by using the intrinsic RBL capacitors to perform CIM computations within the limited available RBL swing in a 2T1C eDRAM.

VI. Conclusion

We explored architecture design with recent advancements in Back-End-Of-Line (BEOL) transistors and monolithic 3D integration techniques. A high-performance matrix multiplication unit based on ultra-high density monolithic 3D eDRAM is proposed and co-designed with analog CIM circuits. By optimizing with outer-product style bit-stationary dataflow, the simulation results show that the proposed architecture achieves significant energy efficiency, performance, and area efficiency.

Acknowledgement

We thank the UT iMAGINE consortium for supporting this research project.

References

[1] H. Touvron *et al.*, "Llama: Open and efficient foundation language models," *arXiv:2302.13971*, 2023.

[2] B. Mildenhall *et al.*, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, 2021.

[3] T. Brown *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, 2020.

[4] J. H. Kim *et al.*, "Aquabolt-xl: Samsung hbm2-pim with in-memory processing for ml accelerators and beyond," in *HCS*. IEEE, 2021.

[5] M. He *et al.*, "Newton: A dram-maker's accelerator-in-memory (aim) architecture for machine learning," in *MICRO*, 2020.

[6] Y. Yu and N. K. Jha, "Energy-Efficient Monolithic Three-Dimensional On-Chip Memory Architectures," *TNANO*, 2018.

[7] A. Keshavarzi *et al.*, "FerroElectronics for Edge Intelligence," *IEEE Micro*, 2020.

[8] A. Belmonte *et al.*, "Capacitor-less, long-retention (¿400s) dram cell paving the way towards low-power and high-density monolithic 3d dram," in *IEEE IEDM*, 2020.

[9] M. Oota *et al.*, "3D-Stacked CAAC-In-Ga-Zn Oxide FETs with Gate Length of 72nm," in *IEDM*, 2019.

[10] A. Belmonte *et al.*, "Tailoring igzo-tft architecture for capacitorless dram, demonstrating $> 10^3$s retention, $> 10^{11}$ cycles endurance and lg scalability down to 14nm," in *IEEE IEDM*, 2021.

[11] S. Raman *et al.*, "Igzo cim: Enabling in-memory computations using multilevel capacitorless indium–gallium–zinc–oxide-based embedded dram technology," *IEEE JXCDC*, 2022.

[12] Z. Chen *et al.*, "15.3 a 65nm 3t dynamic analog ram-based computing-in-memory macro and cnn accelerator with retention enhancement, adaptive analog sparsity and 44tops/w system energy efficiency," in *ISSCC*. IEEE, 2021.

[13] M. Cavalcante *et al.*, "Ara: A 1-ghz+ scalable and energy-efficient risc-v vector processor with multiprecision floating-point support in 22-nm fd-soi," *TVLSI*, 2019.

[14] Y. Wang *et al.*, "Exploiting parallelism for cnn applications on 3d stacked processing-in-memory architecture," *TPDS*, 2018.

[15] Y. Li *et al.*, "Monolithic 3d integration of logic, memory and computing-in-memory for one-shot learning," in *IEDM*. IEEE, 2021.

[16] Q. Zhu *et al.*, "Accelerating sparse matrix-matrix multiplication with 3d-stacked logic-in-memory hardware," in *HPEC*, 2013.

[17] S. Xie *et al.*, "16.2 edram-cim: Compute-in-memory design with reconfigurable embedded-dynamic-memory array realizing adaptive data converters and charge-domain computing," in *ISSCC*, 2021.

[18] C. Yu *et al.*, "A logic-compatible edram compute-in-memory with embedded adcs for processing neural networks," *IEEE TCAS-I*, 2021.

[19] S. Xie *et al.*, "Gain-cell cim: Leakage and bitline swing aware 2t1c gain-cell edram compute in memory design with bitline precharge dacs and compact schmitt trigger adcs," in *VLSI Symposium*. IEEE, 2022.