A 28nm 31.2 TOPS/W Diffusion Transformer Accelerator Using Memory-in-Compute Architecture

Soonha Hwang*, Kaining Zhou*, Nathan Chiang, Vignesh Sundaresha, and Naresh Shanbhag
University of Illinois at Urbana-Champaign, Urbana, IL, USA
{soonhah2, kainingz, chc10, vs49, shanbhag}@illinois.edu

***Abstract*—Diffusion transformers (DiTs) are effective for image and video generation, but their large model sizes and high computational requirements hinder their use on resource-constrained platforms. To address these major challenges, this work presents a DiT accelerator which enables efficient deployment on edge devices. The accelerator is co-designed with a state-of-the-art (SOTA) $2.2\,\mathrm{M}$ parameter count DiT algorithm achieving a Fréchet Inception Distance (FID) of 10.9. The accelerator adopts a custom instruction-set architecture (ISA) with an optimized vectorized processing execution model. A digital in-memory computing (DIMC) architecture is employed to speed-up the compute-heavy matrix-vector multiply (MVM) kernels in the DiT algorithm. A novel DIMC floorplan called memory-in-compute architecture (MICA) is proposed to alleviate the energy and latency overhead of DIMC's digital summer. MICA is further employed to efficiently generate independent Gaussian noise samples on-chip to support DiT's generative functionality. Measured results from a prototype IC fabricated in a $28\,\mathrm{nm}$ process indicates the MICA macro operates at a maximum frequency of $525\,\mathrm{MHz}$, achieves a SOTA compute density of $1.52\,\mathrm{TOPS/mm^2}$ at $(1\,\mathrm{V}, 525\,\mathrm{MHz})$ and a SOTA energy efficiency of $31.2\,\mathrm{TOPS/W}$ at $(0.52\,\mathrm{V}, 100\,\mathrm{MHz})$.**

***Index Terms*—In-Memory Computing, AI accelerators, Generative AI**

I. Introduction

Diffusion transformers (DiTs) have shown remarkable success due to their ability to generate high-quality images while leveraging the scalability and flexibility of transformer architectures [1]. These models operate by iteratively denoising random noise, which, while effective, leads to significant computational demands because of the numerous matrix-vector multiplications (MVMs) required at each denoising step. The large parameter sizes (e.g., $> 600\,\mathrm{M}$) and high computational complexity (e.g., $> 100\,\mathrm{GFLOPs}$ per step) challenge their deployment in resource-constrained edge devices for applications such as augmented and virtual reality (AR/VR).

To address this challenge, one-step diffusion models [2], [3] have been proposed recently, which consolidate the diffusion process into a single inference step to reduce both computational complexity and latency. However, this reduction is insufficient for the purposes of edge deployment, e.g., [3] reports a $610\,\mathrm{M}$ one-step DiT model. Furthermore, the MVM kernel remains the dominant source of computational complexity in DiTs. While digital in-memory computing (DIMC) [4]–[6] is a promising approach to realize energy-efficient MVM due to its robustness to noise and process scalability, the adder tree/digital summer dominates the overall energy, delay, and area [7].

While many transformer ICs have been recently reported [4], [8], [9], to the best of our knowledge, ours is the first to specifically target DiTs. We do so by addressing three key challenges (see Fig. 1): 1) *large parameter count*: we employ a holistic algorithm-architecture co-design methodology (Section III); 2) *high energy and latency overhead of the DIMC adder tree*: We propose an adder tree-centric DIMC floorplan referred to as memory-in-compute architecture (MICA) (Section IV); 3) *necessity of on-chip Gaussian noise sample generation*: We develop a Gaussian pseudo-random noise generator (GPRNG) leveraging MICA (Section V).

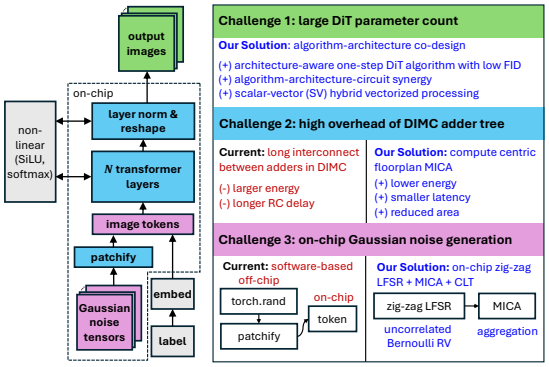


Fig. 1: Challenges in realizing a one-step DiT on-chip. This DiT chip processes all MVMs involved during inference.

This paper reports a DiT accelerator implemented in a $28\,\mathrm{nm}$ CMOS technology. Measured results show that the MICA macro operates at a maximum frequency of $525\,\mathrm{MHz}$ achieves a state-of-the-art (SOTA) compute density of $1.52\,\mathrm{TOPS/mm^2}$ at $(1\,\mathrm{V}, 525\,\mathrm{MHz})$. At $(0.52\,\mathrm{V}, 100\,\mathrm{MHz})$, the MICA macro achieves a SOTA energy efficiency of $31.2\,\mathrm{TOPS/W}$.

II. Algorithm-Architecture Co-Design

A SOTA $2.2\,\mathrm{M}$ parameter one-step DiT model [10] achieving a Fréchet Inception Distance (FID) of 10.9 was co-designed with the DiT accelerator architecture in Fig. 2. This was done by first quantizing the floating-point model in [10] to $8\,\mathrm{b}$ fixed-point with a uniform quantization group size of 32 for attention computation and 128 for all other MVMs. The fixed-point model achieves an FID of 11.3. Next, the accelerator's architectural parameters were chosen to match those of the DiT model. Shown in Fig. 2, the accelerator has an I/O bitwidth of 32 to achieve high-throughput data transfer; each MICA bank comprises 32×32 SRAM bitcells to support four $8\,\mathrm{b}$ weight 32-dimensional dot products (DPs) to generate four $32\,\mathrm{b}$ outputs. These dimensions were chosen to be compatible with the quantization group size and the processor bitwidth.

*contributed equally.

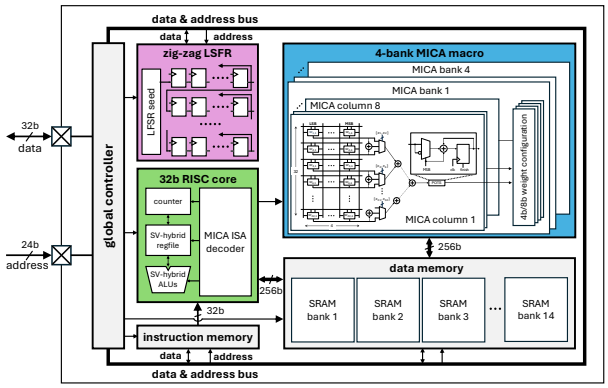


Fig. 2: The proposed MICA-based DiT accelerator.

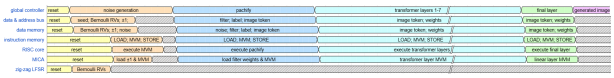


Fig. 3: System timing diagram.

III. Chip Architecture

We propose a DiT accelerator architecture featuring (see Fig. 2): 1) four 32×32 MICA banks as the core computational engine (Section IV); 2) a 32 b GPRNG block (Section V) comprising a zig-zag linear feedback shift register (LFSR) to generate 32 b independent Gaussian noise samples required by the diffusion algorithm; and 3) a custom ISA with vectorized processing realized in a RISC core. The system timing diagram in Fig. 3 shows the execution sequence of the inference steps shown in Fig. 1 in each architectural block of Fig. 2. The transformer layers account for more than 99% of the overall latency.

A. MICA ISA

We propose a custom ISA inspired by that of RISC-V (see Table I for a snapshot) to fully exploit the flexibility of our DiT accelerator. Following the format of RISC-V, we customize the opcode field (bit 6-0) and a function field (bit 14-12) to define instructions that support required functionalities while being matched to the MICA bank parameters. For example, the ISA provides the `LOAD ARRAY` and `MVM` instructions to support two primary MICA operations (write and DP), with additional parameters stored in the configuration registers for variations. Also, to support flexible computation, the ISA provides memory operations with variable operand lengths, e.g., `WORD` and `QUAD-WORD`.

TABLE I: A snapshot of MICA ISA

31 25	24 20	19 15	14 12	11 7	6 0	
imm[31:12]				rd	0110100	LUI
0000000	rs2	rs1	000	rd	0110000	ADD
imm[11:0]		rs1	000	rd	0010000	ADDI
0000000	rs2	rs1	000	rd	0001000	MVM
imm[11:0]		rs1	001	rd	0000000	LOAD ARRAY
imm[11:0]		rs1	010	rd	0000000	LOAD WORD
imm[11:0]		rs1	011	rd	0000000	LOAD QUAD-WORD
imm[11:5]	rs2	rs1	000	imm[4:0]	0100000	STORE BYTE
imm[11:5]	rs2	rs1	001	imm[4:0]	0100000	STORE WORD
imm[11:5]	rs2	rs1	010	imm[4:0]	0100000	STORE QUAD-WORD

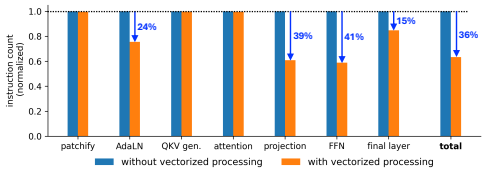


Fig. 4: Instruction count saving due to vectorized processing across all types of layers in our DiT model. "Total" measures the overall instruction count reduction including all layers.

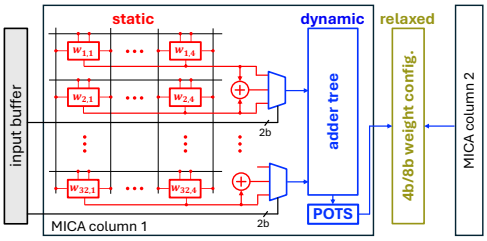


Fig. 5: A MICA column with LUT-based MAC (in blue).

B. Vectorized Processing

The accelerator employs vectorized processing for efficient instruction execution and reducing the number of necessary instructions. A hybrid register file with both scalar and vector registers is deployed. The vector register types include the vector length register, the vector stride register, and registers for vector operands. For example, during execution, if the MICA macro needs to load weights from data memory 32 times, the vector length register will be set to 32, and the vector stride register will be set according to the difference in the addresses of two consecutive weights. The resulting reduction in instruction count (see Fig. 4) ranges from 15%-to-41% when executing layers such as projection and FFN, which have a large number of weight loading instructions that can be vectorized. In contrast, when executing patchify, QKV generation, and attention layers, reduction in instruction count is minimal due to fewer weight loading, leading to fewer vectorization opportunities. In any case, a reduction of 36% in overall instruction count is observed in generating a single image.

IV. Memory-In-Compute Architecture (MICA)

MICA is based on a compute-centric floorplan designed to minimize interconnect delay and power consumption of the adder tree in DIMCs. It adopts the look-up table (LUT)-based multiply-accumulate (MAC) design in [6] to partition a column into three stages (see Fig. 5): 1) *static*; 2) *dynamic*; and 3) *relaxed*. The static stage precomputes the sum of two weights and does not toggle during inference. Therefore, this stage is implemented with relaxed timing and layout constraints, even permitting empty spaces, in exchange for higher density in the dynamic stage (see Fig. 6(b)). This asymmetry and reduced utilization increases write energy negligibly due to weight reuse. For example, in the DiT model [10], the weights are updated only once every 1024 input activations, resulting in weight writes contributing to 0.05% of the total power consumed in a MICA column.

The dynamic stage (in blue) comprising a 4:1 MUX, a $\log_2(N) - 1$-stage adder tree and a powers-of-two summation

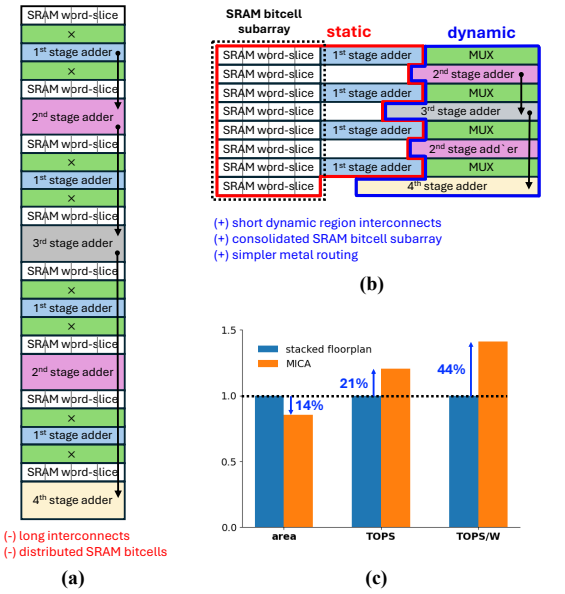


Fig. 6: Floorplan comparison: (a) DIMC's symmetric floorplan, (b) MICA's compute-centric floorplan – symmetry is enforced only in the adder tree, and (c) normalized gains of MICA w.r.t. DIMC in area, TOPS, TOPS/W.

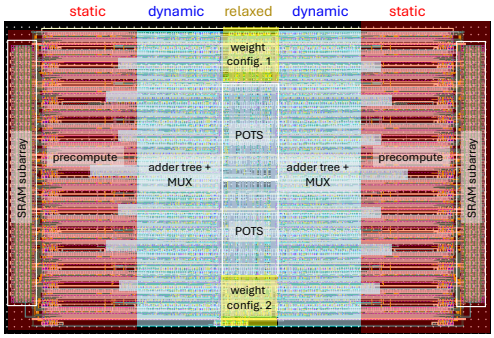


Fig. 7: The layout of two 32×4 MICA columns.

(POTS), is the most timing critical and power hungry ($> 95\%$ of the MICA column) of the three stages. Therefore, the dynamic stage has a dense floorplan to minimize interconnect length (see Fig. 6(b)). As a result, compared to DIMC's [5], [6] (see Fig. 6(a)) symmetric floorplan, MICA's compute-centric floorplan (Fig. 6(b)) achieves a $5.4\times$ reduction in interconnect length in the adder tree.

The relaxed stage comprises a $4\,\mathrm{b}/8\,\mathrm{b}$ configuration unit in which two $4\,\mathrm{b}$ MICA columns can be dynamically reconfigured as a single $8\,\mathrm{b}$ column. This flexibility enables the mapping of both $4\,\mathrm{b}$ or $8\,\mathrm{b}$ quantized DiT models. This stage incurs minimal energy overhead due to its simple arithmetic logic (~1.7%). The layout of the relaxed stage is split and distributed to occupy two unused regions (see Fig. 7) resulting from a mismatch between the layout dimensions of the POTS and adder tree in the dynamic region. This placement strategy improves overall density by using empty spaces that would otherwise remain unused.

Post layout simulations with RC extraction (Fig. 6(c)) show

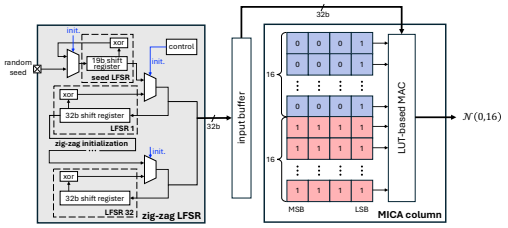


Fig. 8: GPRNG utilizes a zig-zag LFSR (left) to generate Bernoulli bits which are summed via MICA (right) to generate near independent Gaussian samples. The 16 MICA column weights are initialized to $W = +1$ (first 16) and $W = -1$ (next 16) to achieve zero mean. A total of 64 Bernoulli bits are summed to obtain a sample approximating the $\mathcal{N}(0, 16)$ distribution.

that MICA yields a 44% improvement in energy efficiency (TOPS/W) and a 21% increase in throughput (TOPS) for a DP dimension $N = 32$ at $(1\,\mathrm{V}, 525\,\mathrm{MHz})$. Note: these benefits are expected to increase with N.

V. Gaussian Pseudo Random Number Generator

Aided by MICA's efficient summing capability, the GPRNG generates near-independent zero mean Gaussian noise samples by leveraging the Central Limit Theorem. The GPRNG (see Fig. 8) comprises 32 $32\,\mathrm{b}$ LFSRs connected in a serial zig-zag pattern, and initialized using a $19\,\mathrm{b}$ seed LFSR. Ideally, the 32 LFSRs need to generate uncorrelated Bernoulli random bits which will then be summed up using MICA to obtain uncorrelated, hence independent, Gaussian samples. The zig-zag topology ensures distinct seed values in each of the 32 LFSRs, while minimizing the need for off-chip random seed generation. The non-zero lag correlation coefficients of GPRNG-generated Gaussian samples obtained by summing 64 Bernoulli bits generated by the LFSRs were found to be < 0.0042, indicating their near independence.

The generated Bernoulli random variables are aggregated using a 32×4 MICA column. MICA's energy-efficient adder tree enables this accumulation at low latency and power, without requiring off-chip or software-based noise generation. During patchify, the filters are scaled by 0.25 to generate image token outputs with unit variance. The images generated by the DiT algorithm [10] using noise samples from GPRNG was found to result in FID scores close to those obtained using `torch.randn`.

VI. Measured Results

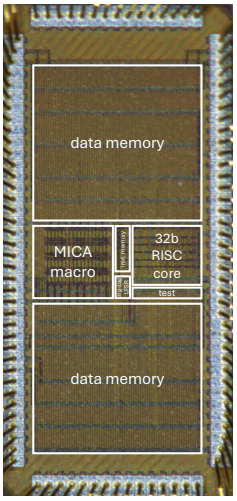
Fig. 9 shows the chip summary. Fabricated in a $28\,\mathrm{nm}$ CMOS process, the MICA macro operates at a maximum frequency of $525\,\mathrm{MHz}$ at VDD of $1\,\mathrm{V}$, where it achieves $0.06\,\mathrm{TOPS}$ and $1.52\,\mathrm{TOPS/mm^2}$ for $8\,\mathrm{b}$ input and $8\,\mathrm{b}$ weight (see Fig. 10(b)). When operating at $100\,\mathrm{MHz}$, our MICA macro operates at $0.52\,\mathrm{V}$ where peak energy efficiency is measured at $31.2\,\mathrm{TOPS/W}$ under the checkerboard test (the worst case). Table II shows that our design achieves SOTA energy and area efficiency when compared to recent transformer processor ICs.

Based on the measured results, images are estimated to be generated at a maximum speed of $0.8\,\mathrm{s/image}$ at $500\,\mathrm{MHz}$ and best energy efficiency of $58.5\,\mathrm{mJ/image}$ at $100\,\mathrm{MHz}$, $0.52\,\mathrm{V}$.

TABLE II: Comparison table

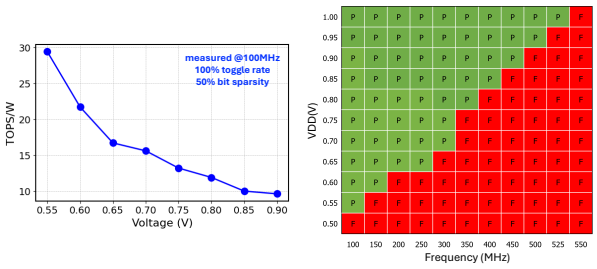
	ISSCC'21 [11]	ISSCC'22 [5]	VLSI'22 [6]	JSSC'23 [4]	CICC'24 [8]	This Work
Technology	22 nm	5 nm	12 nm	28 nm	28 nm	28 nm
Complexity	Macro	Macro	Macro	**Processor**	**Processor**	**Processor**
Target workload	CNN	CNN	CNN	Sparse transformer	General	**DiT** + General
Weight precision	4/8/12/16	4	4/8	8/16	8	4/8
Input precision	1-8	4	4-8	8/16	8	1-8
Supply voltage (V)	0.72	0.5-0.9	0.72	0.6-1.0	0.76-0.92	0.52-1.0
Frequency (MHz)	500	360-1440	800	240	75-152	100-525
Peak performance (TOPS) [1]	0.102	0.735 [2]	0.336	1.48	-	0.06 [3]
Area efficiency ($\mathrm{TOPS/mm^2}$) [1]	1.01	13.8 [2]	2.60	0.221	-	**1.52** [3]
Energy efficiency (TOPS/W) [1]	24.7	63 [2]	30.3	20.5	4.52	**31.2** [4]

[1] 1 operation refers to an (8 b input, 8 b weight) addition or multiplication; [2] Estimated by [5]; [3] Measured at (1 V, 525 MHz);
[4] Measured at (0.52 V, 100 MHz); checkerboard test with weight bit sparsity of 50%, and a worst-case input toggle rate of 100%.



Chip summary	
Technology	28nm CMOS
Chip size (mm×mm)	2.0×0.97
MICA bank area (µm×µm)	258×38
MICA memory (kb)	4
Data memory size (Mb)	3.7
Supply voltage (V)	0.52-1.0
Input precision (bit)	1-8
Weight precision (bit)	4/8
Output precision (bit)	16/32

Fig. 9: Chip micrograph and summary.



(a) **(b)**

Fig. 10: (a) Energy efficiency vs. supply, and (b) Shmoo plot.

VII. Conclusion

This paper presents the first diffusion transformer accelerator IC. By leveraging algorithm-architecture co-design methodology, a custom ISA, an optimized vectorized execution model, and the novel digital in-memory computing floorplan (MICA), the fabricated MICA macro demonstrates SOTA area ($1.52\,\mathrm{TOPS/mm^2}$) and energy ($31.2\,\mathrm{TOPS/W}$) efficiency. Our work paves the way for IC implementations of energy-efficient, real-time, and high-quality generative AI based on diffusion models.

Acknowledgment

This work was supported by the Semiconductor Research Corporation (SRC) and the Defense Advanced Research Projects Agency (DARPA) under the JUMP 2.0 Center for the Co-Design of Cognitive Systems (CoCoSys).

References

[1] W. Peebles and S. Xie, "Scalable diffusion models with transformers," in *ICCV*, October 2023, pp. 4195–4205.

[2] T. Yin et al., "One-step diffusion with distribution matching distillation," in *CVPR*, 2024, pp. 6613–6623.

[3] W. Luo et al., "One-step diffusion distillation through score implicit matching," in *Advances in Neural Information Processing Systems*, vol. 37, 2024, pp. 115 377–115 408.

[4] F. Tu et al., "TranCIM: full-digital bitline-transpose CIM-based sparse transformer accelerator with pipeline/parallel reconfigurable modes," *IEEE JSSC*, vol. 58, no. 6, pp. 1798–1809, 2023.

[5] H. Fujiwara et al., "A 5-nm 254-TOPS/W 221-TOPS/$\mathrm{mm^2}$ fully-digital computing-in-memory macro supporting wide-range dynamic-voltage-frequency scaling and simultaneous MAC and write operations," in *ISSCC*, vol. 65, 2022, pp. 1–3.

[6] C.-F. Lee et al., "A 12nm 121-TOPS/W 41.6-TOPS/$\mathrm{mm^2}$ all digital full precision SRAM-based compute-in-memory with configurable bit-width for AI edge applications," in *VLSI Technology and Circuits*, 2022, pp. 24–25.

[7] C.-T. Lin et al., "DIMCA: an area-efficient digital in-memory computing macro featuring approximate arithmetic hardware in 28 nm," *IEEE JSSC*, vol. 59, no. 3, pp. 960–971, 2024.

[8] Y. Qiu, "Quartet: a 22nm 0.09mJ/Inference digital compute-in-memory versatile AI accelerator with heterogeneous tensor engines and off-chip-less dataflow," in *CICC*, 2024, pp. 1–2.

[9] S. Liu et al., "A 28nm 53.8TOPS/W 8b sparse transformer accelerator with in-memory butterfly zero skipper for unstructured-pruned NN and CIM-based local-attention-reusable engine," in *ISSCC*, 2023, pp. 250–252.

[10] V. Sundaresha, "Designing parameter and compute efficient diffusion transformers using distillation," in *ICLR 2025 Workshop*, 2025. [Online]. Available: https://openreview.net/forum?id=x1xp9gmszo

[11] Y.-D. Chih et al., "An 89TOPS/W and 16.3TOPS/$\mathrm{mm^2}$ all-digital SRAM-based full-precision compute-in memory macro in 22nm for machine-learning edge applications," in *ISSCC*, vol. 64, 2021, pp. 252–254.