An 11.16μJ/token Edge SLM Decoder Accelerator with Scalable Ring-based Configuration for Token-level Pipelining in 16nm FinFET

Guanchen Tao[*1], Junyi Luo[*2], Shiwei Liu[1], Anhang Li[1], Gregory Kielian[3], Kauna Lei[1], Qirui Zhang[1], Dennis Sylvester[1] and Mehdi Saligane[2,3]

1. Department of Electrical & Computer Engineering, University of Michigan, Ann Arbor, 48109, MI, United States
2. Department of Engineering, Brown University, Providence, 02906, RI, United States
3. Google Research, Mountain View, 94043, CA, United States

mehdi_saligane@brown.edu

*Equally Credited Authors (ECAs)

Sub-billion-parameter language models (SLMs) enable practical on-device intelligence. However, edge deployment remains constrained by memory-bound decode stages and limited batch-level parallelism. Existing LLM accelerators, designed for cloud-scale workloads, suffer severe energy inefficiency in the decode stage when adapted to multi-request edge settings. We propose rDXE, a multi-chip, ring-based decoder engine that introduces token-level pipelining to sustain high utilization, low decode latency, and reduced memory traffic across diverse requests. Measured silicon results in 16nm demonstrate 5.26 TOPS/W and 11.16 μJ/token for SLM decode workloads. rDXE also integrates a vector recompute (VRC) unit that fuses Softmax and RMSNorm with GEMV, and an iWuR KV cache with hardware GQA support, achieving up to 4.66× latency reduction for non-linear layers and up to 1.31× end-to-end decode speedup.

Keywords—Transformers, Edge Accelerators, Language models, Token-level pipelining, On-chip memory

I. Introduction

Large language models (LLMs) have permeated many aspects of our daily lives, powering chatbots, summarization tools, and assistant applications. For many of these use cases, edge deployment is attractive due to improved privacy, reliability, latency, and operating cost [1]. To support this trend, sub-billion-parameter language models (SLMs) have been introduced, trading some accuracy for lower latency and energy. However, existing LLM accelerators [6–8], designed for cloud settings with abundant token- and batch-level parallelism, suffer degraded energy efficiency in the decode stage when moved to the edge. The autoregressive nature of decode stage makes it memory-bound: each iteration repeatedly incurring expensive external memory access (EMA) to reload weights and key-value states. In edge scenarios, limited batch sizes further reduce MAC utilization, imposing substantial strain on the power budget.

Notably, keeping all SLM weights on-chip is increasingly feasible. Current SRAM densities are unprecedented (38.1 Mb/mm² [2]), allowing ~250M INT8 parameters within a 50 mm² SRAM budget (roughly half a mobile SoC). This makes SLM practical to store in SRAM, eliminating EMA.

Furthermore, edge devices may need to handle concurrent requests from diverse applications, such as voice assistants, healthcare wearables, and smart keyboards. While batch-level parallelism can be exploited in most linear layers, the attention mechanism is constrained by request specific key-value caches and sequence lengths, which severely limits parallelism across concurrent user-requests.

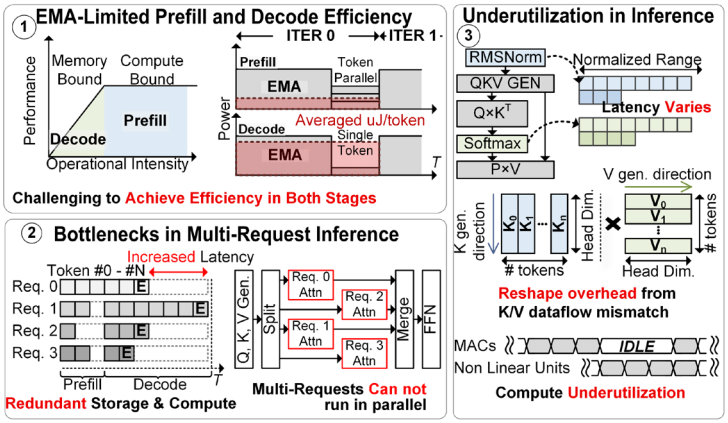


Fig. 1. System-level challenges of edge SLM inference showing prefill/decode asymmetry and multi-request padding overhead.

To mitigate these issues, prior work [9] explored multi-user scenarios and introduced a software-based solution atop a multi-core system, but its scope remains tied to data-center environments. Likewise, [10] proposed a scalable multi-core architecture that primarily optimizes multi-batch scheduling, but does not target the decode stage where each batch contains only a single token's workload. In this paper, we present rDXE, a multi-chip scalable system with token-level pipelining that targets the decode stage of SLMs and efficiently supports multi-request, multi-token execution on edge devices.

II. Challenges and Contributions

To enable on-device processing, hardware must provide high energy efficiency, low memory requirements, and low latency. Edge SLMs have been proposed for efficient on-device inference. These models can be tuned to specific tasks, with MobileLLM 125M [3] achieving only an 8.3% drop on API calling tasks compared to ~56x larger 7B models, and MobileLLM 350M achieving a 29.1% gain on chat tasks while ~20x smaller. However, as shown in Fig. 1, several key challenges remain for efficient on-device inference:

Unbalanced Execution Pattern between Prefill and Decode: On-device SLM inference suffers from a fundamental stage asymmetry:

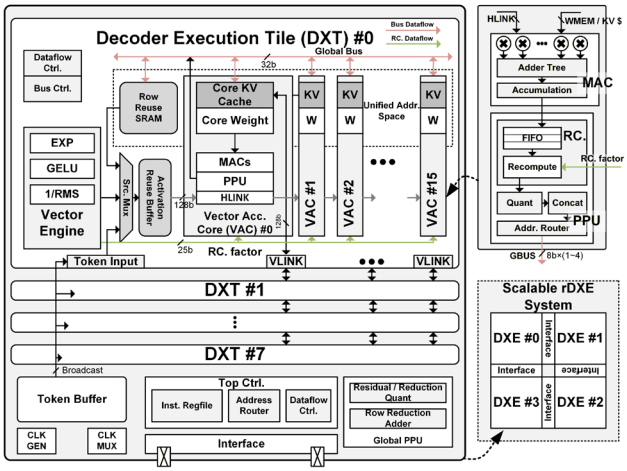


Fig. 2. Microarchitecture of a single decoder execution engine (DXE) with DXT, VAC, KV cache, and VRC paths.

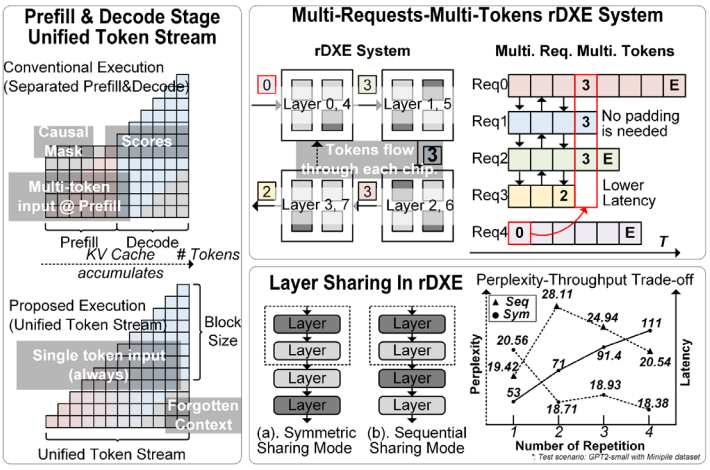


Fig. 3. Unified Token Stream and Multi-Requests–Multi-Tokens Execution Support in rDXE with Layer Sharing.

prefill benefits from high token parallelism, while decode remains memory-bound and restricted to batch-level parallelism. Each iteration triggers repeated EMA, creating a significant energy burden.

Latency Bottlenecks and Padding Overhead in Multi-Request Scenarios: In multi-request settings, overall latency is limited by the request with the longest decode sequence. Additionally, requests with differing sequence lengths must be padded to a uniform length, wasting both energy and processing cycles. Prior selective batching approaches [4] cannot resolve the lack of batch-level parallelism in the attention stage especially under small SLMs and strict edge power budgets.

Non-MAC Operations Lead to Underutilization of MAC: Non-MAC operations such as Softmax and RMSNorm disrupt the steady dataflow of matrix multiplication, reducing MAC unit utilization and complicating scheduling. The key–value cache (KV$) further introduces substantial memory requirements and bandwidth demands, and its asymmetric access patterns for keys and values lead to low SRAM utilization and unnecessary stall cycles.

Motivated by these challenges we propose rDXE, a ring-based decoder execution engine designed for efficient on-device SLM inference. Our main contributions are:

1. **Multi-Requests-Multi-Tokens processing**: rDXE employs a token-level pipeline where each DXE handles one layer or sub-layer and forwards compact intermediate results to the next. This token-level pipelining provides seamless dataflow across both prefill and generation stages, sustaining 77.8% utilization across diverse workloads while supporting multiple concurrent requests.
2. **Vector recompute (VRC):** Hardware support for VRC is proposed, which fuses non-linear Softmax and RMSNorm with linear MAC operations, achieving up to 1.31× latency saving.
3. **Optimization of On-chip Buffers**: An irregular-write, uniform-read (iWuR) scheme mitigates underutilization of KV$ bandwidth and achieves further latency savings up to 1.54×. We also demonstrate the first hardware implementation of group query attention (GQA) [5] through vertical links (VLINK), which can significantly reduce KV$ capacity demand. Furthermore, adaptive dataflow—TB broadcast to all DXTs and rr-SRAM unicast within DXT—enhances data reuse and reduces memory traffic.

III. Proposed Transformer Decoder Accelerator

Fig. 2 illustrates the zoom-in microarchitecture of a single DXE, which can be extended to an rDXE for larger model support. It consists of an I/O interface, a token buffer (TB), a top-level controller, a global post-processing unit (GPPU), and eight decoder execution tiles (DXT). The TB receives input tokens from the I/O interface and flexibly delivers them via multicast to all DXTs. Each DXT integrates 16 vector–accumulate cores (VACs) connected through a 32-bit global bus (GBUS), along with two local data-reuse buffers: a row-reuse SRAM (rr-SRAM) for reusing recent outputs and an activation-reuse buffer for repeated activation access. A dedicated vector engine (VE) within each DXT computes the RC factors, which will be broadcast to the recompute unit (RCU) of each VAC, for VRC, and generates GELU activation outputs. Inside each DXT, VACs are linked via horizontal links (HLINK), forming a systolic input datapath. Combined with the local reuse buffers, this structure maximizes both temporal and spatial reuse of activations. Each VAC contains a multiply-and-accumulate (MAC) unit of 16 multipliers, a local post-processing unit (PPU) for quantization and GBUS communication, and an RCU for VRC support. To reduce wiring delay and power in this memory-centric design, every VAC is equipped with local KV$ and WMEM.

A. Multi-Requests-Multi-Tokens processing in rDXE

To maintain high hardware utilization across both stages, each DXE is designed to process a single token, thereby enabling a unified data flow. DXEs are organized in a ring (rDXE), with each DXE assigned to process one or more layers for a single token from a request at each time step. For example, as illustrated on the left of Fig. 3, a configuration with four DXEs (each containing weights for two layers) handles tokens from Users #0 to #3 in the pipeline. When User #3's request is fully served, their computation completes early, immediately freeing resources for User #4's request. This approach eliminates redundant computation and storage, such as zero padding (to align token lengths in prefill) or waiting for the slowest batch in decode, allowing users to finish early or join late. Relative to a baseline supporting 8-way and 16-way simultaneous vector dot-product execution, rDXE delivers 32.7% and 52.4% higher utilization, respectively. The I/O interface supports 16-channel quad-SPI with a bandwidth of 800 Mb/s. The interface transfers only one token during the computation of the previous token, thereby minimizing overhead

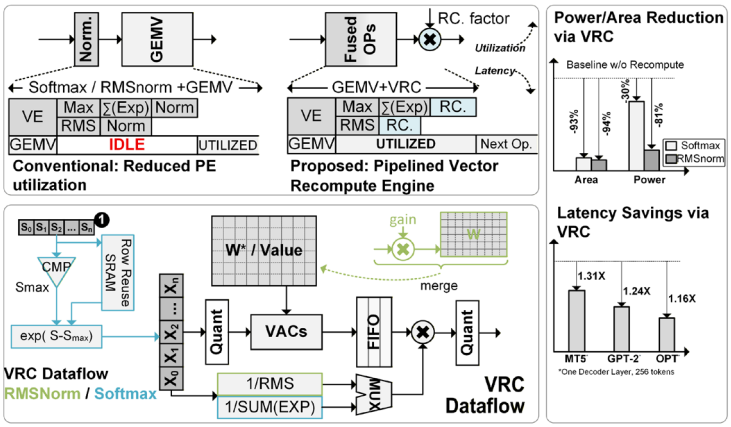


Fig. 4. VRC fuses the GEMV with Non-linear operation, reducing latency without compromising precision.

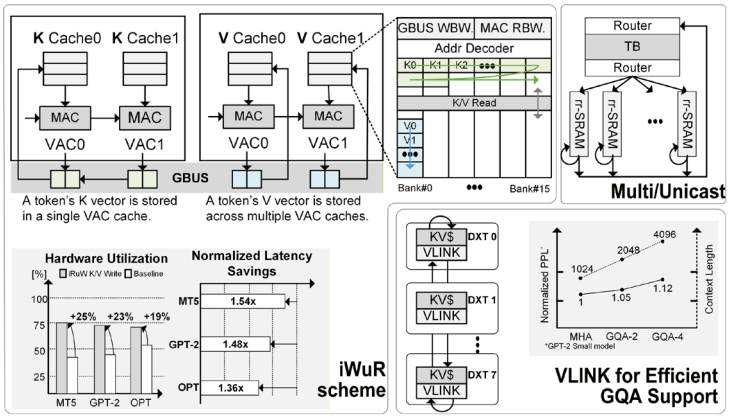


Fig. 5. iWuR, GQA KV sharing, and broadcast/unicast flow for on-chip dataflow optimation.

and hiding I/O latency. This enables the rDXE to support token-level pipelining, ensuring scalable Edge SLM execution under concurrent multi-request workloads. Additionally, the reconfigurable interconnect between DXEs enables flexible layer-sharing patterns. Unlike [1], which had to use immediate block-wise sharing to avoid GPU cache movement, rDXE's ring topology keeps weights stationary and circulates tokens. This supports symmetric and sequential sharing, reducing perplexity by 25% without increasing on-chip memory.

B. VRC: Fusing Nonlinear Normalization with Linear GEMV

Conventional accelerators execute nonlinear operations (e.g., Softmax, RMSNorm) sequentially with linear MAC processing, which lowers MAC utilization, and increases overall latency. Prior work [10] applies polynomial approximations to nonlinear layers and additionally uses delayed statistics with rollback to recover accuracy when approximation errors occur. This approach introduces detection circuitry and incurs latency penalties whenever rollbacks are triggered. Another design [6] also relies on polynomial approximation and reuses the MAC units for Softmax, but it does not reduce the overall latency. In both cases, the use of polynomial approximations leads to degraded numerical precision. We propose VRC, which fuses these nonlinear stages directly into the GEMV pipeline without compromising precision. As shown in Fig. 4, the input token is simultaneously fed to both the MAC unit and the vector engine (VE). For RMSNorm, the MAC computes the dot product using weights pre-merged offline with the RMSNorm gain factor, while the VE computes the reciprocal RMS (RC factor). The MAC's outputs are recombined with the quantized RC factor (25b) broadcast to the RCU of each VAC. InvSqrt operation for RC computation is implemented with a fast Newton–Raphson method. For Softmax, obtaining the maximum value of Q×K is performed during the write back of its result previous to P×V stage. During P×V stage, The VE simultaneously computes the reciprocal exponential sum (RC factor) while MAC executes dot products with V. Both VRC designs reduce power and area by fully pipelining the non-linear functions, eliminating large on-chip buffers for feature storage and reusing quantization units from VACs. This fusion achieves 93%/94% area reduction and 30%/81% power reduction for Softmax/RMSNorm. It achieves up to 4.66× latency reduction for the non-linear operations alone and improves end-to-end latency by up to 1.31× on a single OPT-350M layer with only a slight accuracy loss from quantization.

C. On-Chip Databuffer Optimization

We show that by carefully accounting for dataflow, we can reduce on-chip buffer (KV$, rr-SRAM, TB) footprint and access energy, while increasing local data reuse—features that are especially favorable for edge applications. As shown in Fig. 5, our GBUS and VAC caches support the iWuR scheme: during K writeback, each K output from a VAC is sent to other VACs. Inside each VAC, the KV$ supports byte-level writes with K stored row-by-row. V values are written back only to the local VAC, stored column-by-column. This organization enables uniform, low-latency reads of K and V. In the MT5-small model, iWuR improves utilization by up to 25% over baseline and yields 1.36×–1.54× latency reduction across workloads. Furthermore, VLINK enables GQA by reconfiguring KV$ across DXTs, allowing KV sharing across attention heads. On GPT2-small, sharing KV across 3–6 heads increase validation loss by only 0.041% – 2.5%, while reducing KV$ required capacity by up to 83%. TB supports broadcast mode within each DXT, while the rr-SRAM in each DXT enables unicast mode. During attention layer, heads are split along the output-channel dimension, aligning naturally with broadcast mode; later projection layer acts on all heads together, aligning with unicast mode. Same kind of pattern also applies to MLP layer in our dataflow. This organization increases data reuse and reduces traffic between TB and rr-SRAM.

IV. Results

A. Chip Level Evaluation

We prototyped a single DXE in 16 nm FinFET technology with a total area of 14.21 mm² as shown in Fig. 6. The DXE operates from 12.5 MHz to 200 MHz across a supply range of 0.56 V–1.0 V, consuming 18.95 mW–281.01 mW.

Reported energies are normalized using the highest reported efficiencies: 189.8 TOPS/W [8], 88.36 TOPS/W [6], and 47.8 TOPS/W [7]. For a MT5 model's decoder layer (d_model = 512, n_head = 6, d_ffn = 1024), the #Ops include QKV, Q×K, P×V, projection, and FFN layers. The EMA reduction reported in prior works is 87.5%, 75.7%, and 73%. Leveraging a scalable multi-chip configuration, our design places all weights on-chip, requiring only a single weight load at initialization. In contrast, even assuming prior works can amortize one initial weight load across all input tokens, their decode stage still requires reloading weights once per generated token. Under these settings, as illustrated in Fig. 7, rDXE achieves 11.6 µJ/token and 5.26

	This Work	VLSI25 [9]	JSSC25 [11]	ISSCC25 [8]	ISSCC24 [7]
LLM Acceleration	**Yes**	Yes	No	Yes	Yes
Multi-Requests Support	**Yes**	Yes	No	No	No
Edge Application	**Yes**	No	Yes	No	No
Tech. (nm)	**16**	4	40	28	28
Supply Voltage (V)	**0.56-1.0**	0.65-0.9	0.9	0.58-1.0	0.7-1.1
Frequency (MHz)	**12.5-200**	25-1000	50-100	25-200	50-200
Die Area (mm2)	**14.21**	5.28	65.6	20.25	20.25
Precision	**INT8**	FP16	Posit8	A:INT4/8/16 W:INT1-16/1.58	INT8
On-Chip Memory (kB)	**4096**	2160	12000	500	500
Power (mW)	**18.95-281.01**	11-735	N/A	4.69-82.07	47.6-469.2
Performance(1) (TOPS or TFLOPS)	**0.81-1.63**	8.19	0.0256-0.0512	4.92-13.1	3.41
Energy Efficiency(2) (TOPS/W or TFLOPS/W)	**2.51-6.47**	25.3	0.43-0.50	77.6-189.8	22.9-47.8
Max EMA Reduction Percentage (%)	**100%**	N/A	100%	80.25%	73%
Normalized Energy Consumption(2)(3) (uJ/token)	**11.16**	273.89	119.53	17.28	37.89
Normalized Energy Efficiency(2)(3) (TOPS/W or TFLOPS/W)	**5.26**	0.21	0.49	3.40	1.55

(1)Numbers at the fastest voltage and frequency. (2)Numbers at the most energy-efficient voltage and frequency.
(3) Estimated with LPDDR3 SDRAM and including EMA of 85pJ/B [1], the MT5 model was evaluated with 960 input tokens and 64 output tokens.

Fig. 6. Measurement results and comparison table

TOPS/W, with even greater benefits expected as the output-to-input token ratio increases.

B. System Level rDXE Evaluation

We further assessed system-level inference throughput in Fig. 8. With 7 DXEs yielding 99.47 mm^2 in 16nm, it is affordable in mobile devices, given mobile SoCs today exceed 100mm^2 even in more advanced technologies (sub-5nm). rDXE reaches 4412 tokens/s on the GPT-2 model, demonstrating its effectiveness for decoding workloads in edge SLM deployment.

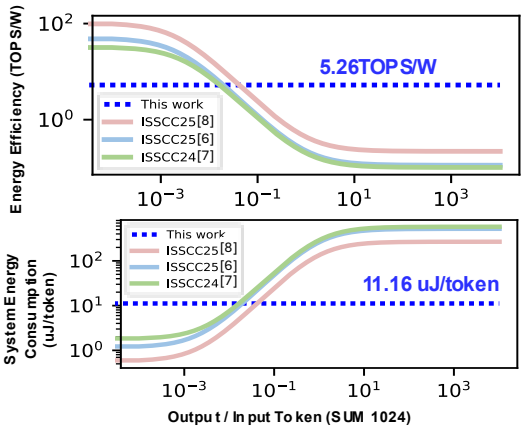
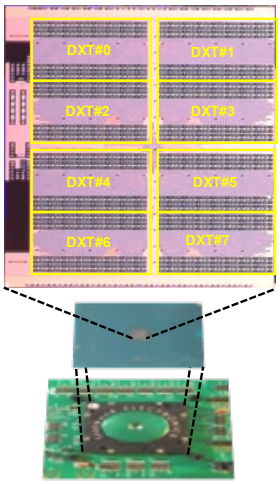


Fig. 7. Energy efficiency and system energy consumption degrade as output/inputtoken ratio increases.

V. Conclusions

This work presented rDXE, a scalable decoder engine tailored for edge SLM inference. By introducing token-level pipelining, rDXE sustains high utilization across multi-request, multi-token workloads. Our VRC mechanism fuses nonlinear operations with MAC compute to reduce latency, while the iWuR KV$ and hardware-efficient GQA design lower memory pressure and mitigate stalls. Together, these techniques deliver a highly efficient decode pipeline that overcomes the memory-bound nature of edge inference, enabling fast, low-energy on-device processing.



	Specifications			
Technology	16nm			
Die Area	14.21 mm^2			
Voltage	0.56V-1.0 V			
Frequency	12.5 MHz-200 MHz			
SRAM	4MB			
Data Precision	VAC: INT8 VE:BF16;			
Power	18.95 mW - 281.01 mW			
Peak Performance	1.63 TOPS **(1)**			
Peak Energy Efficiency	6.47 TOPS/W **(2)**			
Model Type	MT5	Mobile LLM	GPT-2	GPT-2
Dataset	Minipile	PIQA	Wikitext	Wikitext
Parameters	350M	125M	125M	125M
Accuracy	39.83 (BLEU)	60 (ACC)	18.38 (PPL)	20.56 (PPL)
Layer Sharing	1	4	1	4
Required #DXEs in rDXE	2	8	27	7
Inference Throughput (Token/s) **(4)**	5103 **(3)**	3921	14577	4412

(1) @ 1.0 V, 200MHz (2) @ 0.67V, 25MHz
(3) Only the decoder of MT5 is evaluated
(4) Estimated with LPDDR3 SDRAM and including EMA of 85pJ/B [1]

Fig. 8. Die photo of the DXE (left) and system level specification table (right)

Acknowledgment

We thank Intel Corp. for the silicon donation and Google Research for support. The corresponding authors are Mehdi Saligane and Shiwei Liu.

References

[1] Yang, Lita, et al. "Enabling On-Device Large Language Models with 3D-Stacked Memory." NeurIPS 2024 Workshop Machine Learning with new Compute Paradigms.

[2] Chang, Tsung-Yung Jonathan, et al. "A 38.1 Mb/mm 2 SRAM in a 2nm-CMOS-Nanosheet Technology for High-Density and Energy-Efficient Compute." 2025 IEEE International Solid-State Circuits Conference (ISSCC). Vol. 68. IEEE, 2025.

[3] Liu, Zechun, et al. "Mobilellm: Optimizing sub-billion parameter language models for on-device use cases." Forty-first International Conference on Machine Learning. 2024.

[4] Yu, Gyeong-In, et al. "Orca: A distributed serving system for Transformer-Based generative models." 16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22). 2022

[5] Ainslie, Joshua, et al. "Gqa: Training generalized multi-query transformer models from multi-head checkpoints." arXiv preprint arXiv:2305.13245 (2023).

[6] Qin, Yubin, et al. "23.8 An 88.36 TOPS/W Bit-Level-Weight-Compressed Large-Language-Model Accelerator with Cluster-Aligned INT-FP-GEMM and Bi-Dimensional Workflow Reformulation." 2025 IEEE International Solid-State Circuits Conference (ISSCC). Vol. 68. IEEE, 2025.

[7] Kim, Sangyeob, et al. "20.5 C-transformer: A 2.6-18.1 μJ/token homogeneous DNN-transformer/spiking-transformer processor with big-little network and implicit weight generation for large language models." 2024 IEEE International Solid-State Circuits Conference (ISSCC). Vol. 67. IEEE, 2024.

[8] Kim, Sangyeob, Jungwan Lee, and Hoi-Jun Yoo. "Slim-Llama: A 4.69 mW Large-Language-Model Processor with Binary/Ternary Weights for Billion-Parameter Llama Model." 2025 IEEE International Solid-State Circuits Conference (ISSCC). Vol. 68. IEEE, 2025.

[9] Kim, Jung-Hoon, et al. "Adelia: A 4nm LLM Accelerator with Streamlined Dataflow and Dual-Mode Parallelization for Efficient Generative AI Inference." 2025 Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits). IEEE, 2025.

[10] Han, Donghyeon, and Anantha P. Chandrakasan. "MEGA. mini: A Universal Generative AI Processor with a New Big/Little Core Architecture for NPU." 2025 IEEE International Solid-State Circuits Conference (ISSCC). Vol. 68. IEEE, 2025.

[11] Prabhu, Kartik, et al. "MINOTAUR: A Posit-Based 0.42–0.50-TOPS/W Edge Transformer Inference and Training Accelerator." IEEE Journal of Solid-State Circuits (2025)