23.6 MEGA.mini: A Universal Generative AI Processor with a New Big/Little Core Architecture for NPU

Donghyeon Han[1,2], Anantha P. Chandrakasan[1]

[1]Massachusetts Institute of Technology, Cambridge, MA
[2]Chung-Ang University, Seoul, Korea

The global AI market is growing explosively with the rise of generative AI applications, such as image manipulation and text-to-text/image/video creation. AI was primarily expected to automate only simple tasks like classification and data analysis. However, the advent of generative AI has transformed it into a creativity assistant, helping people think more creatively by offering new perspectives through deep neural networks (DNNs). As shown in Fig. 23.6.1, there are various types of DNNs used in generative AI, which can be categorized into two groups: non-diffusion models (NDMs) and diffusion models (DMs). NDMs, such as variational autoencoders (VAEs), generative adversarial networks (GANs), and transformers, are still predominantly used. DMs require 25-100× more operations due to their need for iterative inference (INF). Consequently, generative AI processors need to efficiently accelerate both NDMs and DMs.

[1-5] aim to achieve energy-efficient NDM or DM acceleration, but such works have not solved 3 key problems. First, generative AI demands higher bit-precision for input activations (IAs) than traditional discriminative AI. For example, in DMs, using dynamic fixed-point (DFXP) representation [6] even distorts the output because the IA range varies significantly across different time steps. Additionally, generative AI is vulnerable to quantization errors caused by incorrect integer lengths during the FXP-based IA representation, and the conventional IA quantization lacks a strategy for accuracy recovery to mitigate this issue. Secondly, while [4, 5] attempted to tackle the high-precision IA problem in DMs by leveraging inter-time-step similarity (ITSS) which involves saving inout tensors from a previous iteration and restoring them in the current iteration to minimize computational overhead or skipping operations for similar inputs. This approach is difficult to apply in modern DMs due to the time embedding layer. This layer helps to enhance the output quality of generative AI, but adds random noise into the tensor, reducing the average reusability to 38.1%. Furthermore, ITSS significantly increases external memory access (EMA) because it stores all tensors in external DRAM, resulting in energy consumption due to EMA becoming 20.4× higher than that of on-chip operations. Lastly, modern generative AI models no longer use batch normalization; thus, it is impossible to skip calculating the mean and standard deviation of tensors. Additionally, they mostly employ non-ReLU functions as non-linear activation (NLA), thus, most processors rely on external special function units for lossless and complex NLA operations, introducing extra latency and energy consumption.

The proposed processor solves these problems through the following features: 1) a new Big/Little core architecture [7], MEGA.mini, to support fixed-point (FXP) and floating point (FP) hybrid representations for IAs, 2) an output synchronizer with a task snatching unit to maximize core utilization, 3) a cross-shaped memory architecture to increase chip scalability while minimizing memory access, and 4) a unified tensor streaming core (UTSC) to support efficient normalization and NLA.

Figure 23.6.2 shows the overall architecture of the proposed processor, which consists of 4 MEGA.mini cores, 2 global inout memories (IOMEMs), 2 global weight memories (WMEMs), and a RISC-based top controller. Each MEGA.mini core accelerates DNN by distributing computations to two different heterogeneous cores. The MEGA core consists of a 32×32 processing element (PE) array that maximizes data reuse by broadcasting weights to each row and IAs to each column. The MEGA core does not contain local memory, thus, it loads IAs and weights from the adjacent global IOMEM and WMEM. In addition, the MEGA core employs multiply-accumulate units (MAC) that support FXP-IAs with sign-magnitude (SM) representation, which enhances efficiency by reducing bit-toggling. Conversely, the mini core designed with 32 FP PEs shows lower data reusability but supports the skipping of zero operations appearing in IAs. It reads a single nonzero (NZ) IA from the local IOMEM and loads corresponding weights from the adjacent global WMEM. After operations of MEGA or mini core, the partial-sums (PS) generated by each core are merged by an aggregation core. The results are then stored in temporal memory (TMEM), or sent to UTSC for normalization and NLA. Finally, the final output tensor is either transferred to the global IOMEM or the encoding memory (ENCMEM), depending on the IA data type required for the next layer.

Figure 23.6.3 provides more details about the MEGA.mini core. The MEGA.mini core can support hybrid IAs, where most inlier data (95-97%) is represented using FXP, and only minor outlier data (3-5%) is represented using FP to minimize accuracy degradation. The bulky MEGA core processes dense FXP-IA with in-order execution, while the small mini core quickly processes sparse FP-IA by skipping zero inputs. To further improve the efficiency of the MEGA core, its SM MAC identifies the sign of the multiplication results in advance, directing the results to one of two registers for accumulation. Moreover, the MSB accumulation parts of the MAC are replaced with an asynchronous counter (CNT), which reduces dynamic power consumption by eliminating the clock-tree, register, and full-adder (FA). The proposed MAC offers 15.6% lower area and 36.3% lower power consumption than a conventional 2's complement (2S) MAC design. The efficiency of the mini core is also improved further by adopting dynamic FP12 (DFP12), which adjusts exponent bias based on the integer length used for FXP-IA to maintain a broad data range. Since the mini core only processes outlier data, the FP MAC is simplified by removing the denormalization circuit. The aggregation core gathers accumulation results from both MEGA and mini cores, converting them to FP16 for final aggregation. The MEGA core transfers two accumulation results stored in its dual registers, and the aggregation core normalizes them using the integer length data stored in the layerwise INT LUT. This LUT reflects the results of long-term updates for NDMs and short-term updates for DMs to keep the FP-IA ratio at 3-5%. Overall, the MEGA.mini core achieves 80.5% lower power consumption than a core made with FP16 PEs, and 77.5% lower energy consumption of EMA by avoiding ITSS.

As described in Fig. 23.6.4, the MEGA.mini core supports five different task allocation modes according to the IA data type and layer configuration. Although both MEGA and mini cores load weights from the same global WMEM, they avoid bank conflicts by dividing input channels into two groups and adopting time multiplexing. In MEGA+mini mode (hybrid IAs), core underutilization can occur due to task imbalance and simultaneous access to TMEM during aggregation. To address this problem, the design includes task pre-loading and an enhanced aggregation core with an output synchronizer. The output synchronizer snatches long-latency tasks from the mini core, reordering upcoming computations to prevent simultaneous TMEM access with the MEGA core. The output synchronizer, along with automatic INT LUT update, reduces relative processing time by 53.1-66.9%. Furthermore, 4 MEGA.mini cores are organized into a cluster with a cross-shaped memory architecture that supports both unicasting and broadcasting modes. The unicasting mode ensures each core operates independently by allocating different input batches and output channels. In contrast, the broadcasting mode sacrifices programmability but allows adjacent cores to share the same input batch or output channel. The broadcasting mode can reduce power consumption by 10.4-34.7% and eliminate duplicate EMA by sharing global memory between two adjacent cores.

Figure 23.6.5 shows detailed circuits for the UTSC adopting delayed statistics and conditional polynomial approximation. The UTSC performs post-processing of streaming output tensors by pre-calculating new hyperparameters for scaling and shifting operations. Traditional normalization can calculate these new hyperparameters only after completing the statistical analysis, causing additional latency and memory access. The UTSC reuses statistics from the previous iteration (in DM) or previous frame (in NDM) for the scaling pre-calculation while simultaneously analyzing new statistics. If the delayed statistics result in large errors, a recovery circuit detects these errors and re-calculates normalization using the new statistics. In the NLA processing, the UTSC distinguishes between inlier and outlier streaming tensors and applies different polynomial approximations based on the standard deviation loaded from delayed statistics. Both normalization and NLA approximation are performed using a shared PE, achieving 52% lower area and 36% higher energy efficiency compared with the conventional special function unit.

Figure 23.6.6 shows measurement results and a comparison table. Unlike conventional processors [1-5], which only focused on either NDM or DM, the proposed processor supports all generative AI model types while minimizing accuracy loss through hybrid IA. The processor shows 1.84-to-8.07× higher energy efficiency during NDM acceleration. Compared with the previous DM accelerators [4, 5], it achieves 75.4% lower energy consumption even considering the external DRAM access. The processor is fabricated in 28nm CMOS technology. Its power consumption and energy efficiency vary according to MEGA.mini operating scenario and IA sparsity, successfully demonstrating not only NDMs, but also DMs, while maintaining high computing efficiency without accuracy degradation. In summary, this work introduces MEGA.mini, a new Big/Little NPU architecture realizing not only energy-efficient AI computing, but also universal NPU solutions for on-device generative AI.

Acknowledgement:
This work was supported by the MIT/MTL Samsung Semiconductor Research Fund.

2025 IEEE International Solid-State Circuits Conference (ISSCC) | 979-8-3315-4101-9/25/$31.00 ©2025 IEEE | DOI: 10.1109/ISSCC49661.2025.10904514

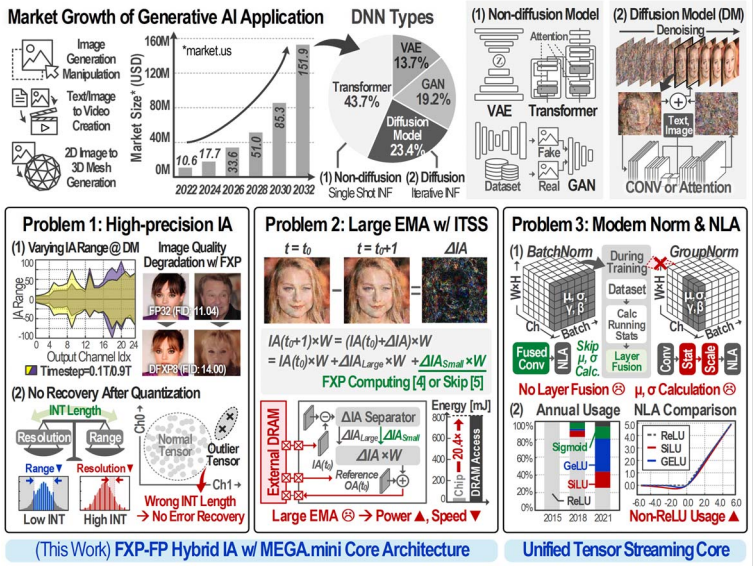


Figure 23.6.1: DNN model types of generative AI and 3 problems of conventional solutions.

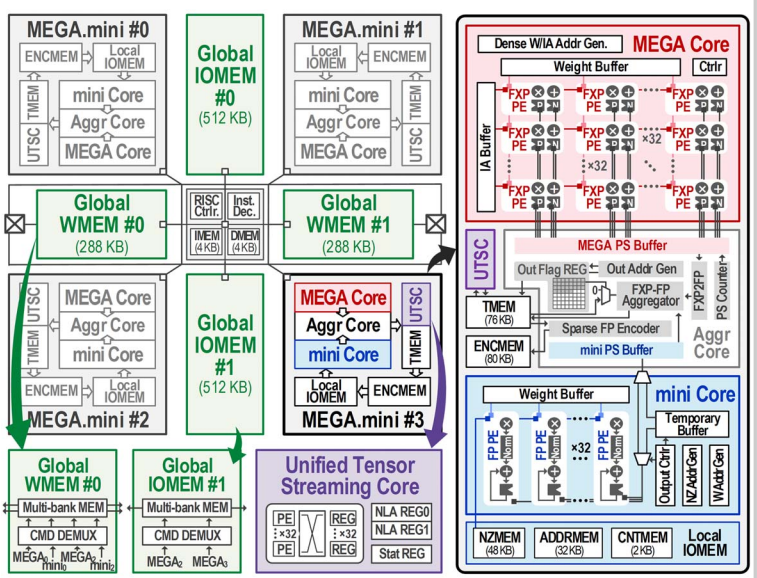


Figure 23.6.2: Overall chip architecture.

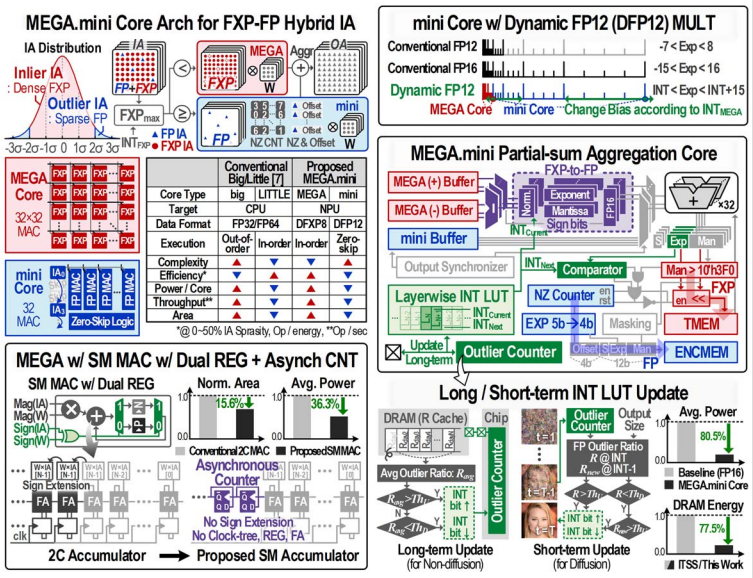


Figure 23.6.3: MEGA.mini core architecture with FXP-FP hybrid IA representation.

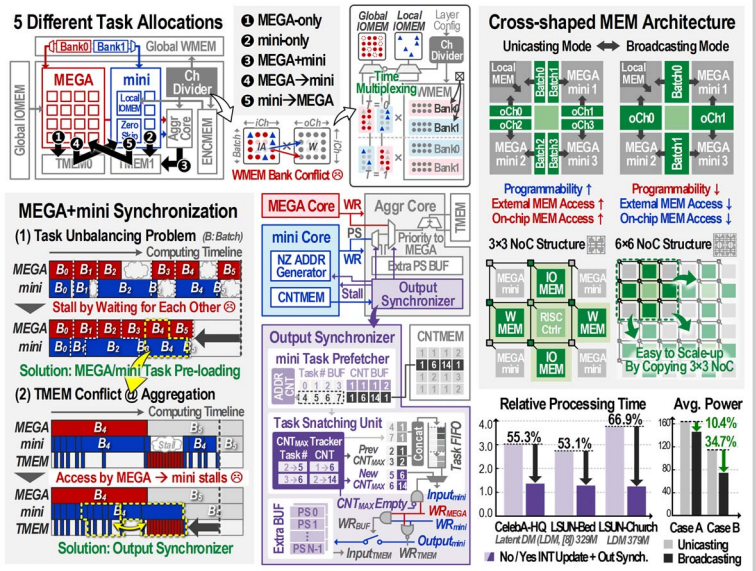


Figure 23.6.4: Five different task allocations of MEGA.mini with output synchronizer and cross-shaped memory architecture.

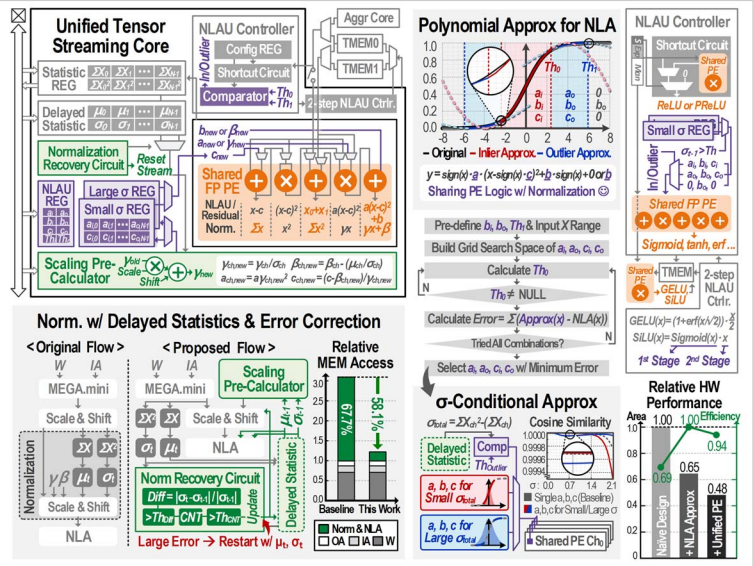
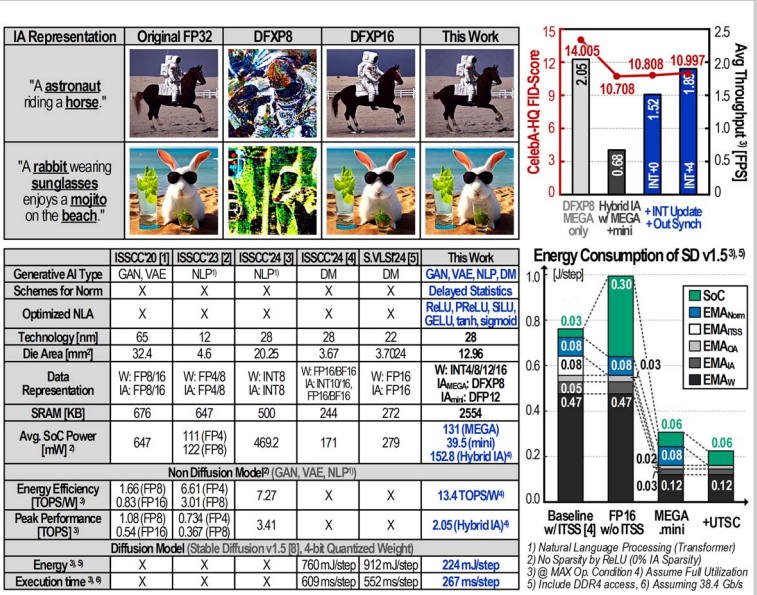


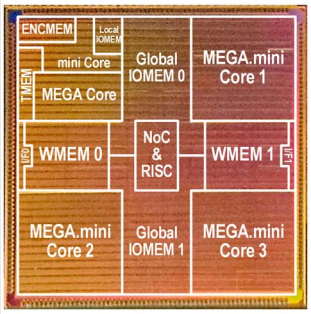
Figure 23.6.5: Unified tensor streaming core (UTSC) with delayed statistics for normalization and conditional polynomial approximation for non-linear activation (NLA).

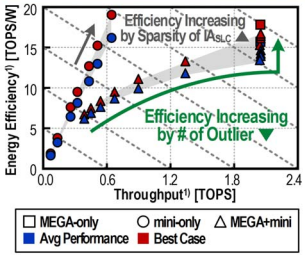


	ISSCC'20 [1]	ISSCC'23 [2]	ISSCC'24 [3]	ISSCC'24 [4]	S.VLSI'24 [5]	This Work
Generative AI Type	GAN, VAE	NLP[6]	NLP[6]	DM	DM	GAN, VAE, NLP, DM
Schemes for Norm	X	X	X	X	X	Delayed Statistics
Optimized NLA	X	X	X	X	X	ReLU, PReLU, SILU, GELU, tanh, sigmoid
Technology [nm]	65	12	28	28	22	28
Die Area [mm²]	32.4	4.6	20.25	3.67	3.7024	12.96
Data Representation	W: FP8/16 IA: FP8/16	W: FP4/8 IA: FP4/8	W: INT8 IA: INT8	W: FP16/8F16 IA: INT10/16, FP16/8F16	W: FP16 IA: FP16	W: INT4/8/12/16 IA$_{MEGA}$: DFXP8 IA$_{mini}$: DFP12
SRAM [KB]	676	647	500	244	272	2554
Avg. SoC Power [mW] [2]	647	111 (FP4) 122 (FP8)	469.2	171	279	131 (MEGA) 39.5 (mini) 152.8 (Hybrid IA)[4]
Non Diffusion Model[2] (GAN, VAE, NLP[1])						
Energy Efficiency [TOPS/W] [3]	1.66 (FP8) 0.83 (FP16)	6.61 (FP4) 3.01 (FP8)	7.27	X	X	13.4 TOPS/W[4]
Peak Performance [TOPS] [3]	1.08 (FP8) 0.54 (FP16)	0.734 (FP4) 0.367 (FP8)	3.41	X	X	2.05 (Hybrid IA)[4]
Diffusion Model (Stable Diffusion v1.5 [8], 4-bit Quantized Weight)						
Energy [3, 5]	X	X	X	760 mJ/step	912 mJ/step	224 mJ/step
Execution time [3, 6]	X	X	X	609 ms/step	552 ms/step	267 ms/step

1) Natural Language Processing (Transformer)
2) No Sparsity by ReLU (0% IA Sparsity)
3) @ MAX Op. Condition 4) Assume Full Utilization
5) Include DDR4 access, 6) Assuming 38.4 Gb/s

Figure 23.6.6: Measurement results and performance comparison table.





	Specifications		
Technology	TSMC 28nm HPC+ 1P10M		
Die Area	3.6 mm × 3.6 mm (12.96 mm²)		
Supply Voltage	0.57 V – 0.9 V		
Operating Frequency	10 MHz – 250 MHz		
SRAM [KB]	MEGA.mini	Global IOMEM	WMEM
	946	1024	576
Data Representation	Weight	Input Activation	
	FXP4/8/12/16	DFXP8 (MEGA), DFP12 (mini)	
# of MAC / Core	MEGA Core	mini Core	
	1024	32	

MEGA.mini Operating Scenario		MEGA-only[5]	mini-only	MEGA+mini[3] Hybrid IA
Power Consumption [mW] [7]	BP[1]	75.8 – 148	22.9 – 43.3	88.8 – 172
	BE[2]	4.56 – 13.5	1.38 – 3.94	5.35 – 15.9
Peak Throughput [GOPS] [4,7]	BP[1]	2048	64[5] – 640[6]	~2048
	BE[2]	204.8	6.4[5] – 64[6]	~204.8
Peak Energy Efficiency [TOPS/W] [7]	BP[1]	13.8 – 27.0	1.48[5] – 27.9[6]	11.9 – 23.0
	BE[2]	15.2 – 44.9	1.62[5] – 46.4[6]	12.9 – 38.1

Model Name	BitNet [8]	SD v1.5 [9]	LDM-329M	LDM-SR	DDGAN[10]
Task	Language Model	Text-to-Image Generation	Unconditional Image Gen.	Super Resolution	Unconditional Image Gen.
Avg. TOPS/W [2,7]	19.30	14.61	15.57	14.23	13.46
Avg. TOPS [1,7]	2.048	1.986	1.950	1.836	1.525

1) Best Performance, 0.9V, 250MHz, 2) Best Efficiency, 0.57V, 25MHz, 3) Full Utilization 4) 1MAC = 2Ops, 5) 0% IA Sparsity, 6) 90% IA Sparsity, 7) Exclude EMA

Figure 23.6.7: Chip photograph and performance summary.

References:

[1] S. Kang et al., "GANPU: A 135TFLOPS/W Multi-DNN Training Processor for GANs with Speculative Dual-Sparsity Exploitation," *ISSCC*, pp. 140-142, 2020.

[2] T. Tambe et al., "A 12nm 18.1TFLOPs/W Sparse Transformer Processor with Entropy-Based Early Exit, Mixed-Precision Predication and Fine-Grained Power Management," *ISSCC*, pp. 342-344, 2023.

[3] S. Kim et al., "C-Transformer: A 2.6-18.1μJ/Token Homogeneous DNN-Transformer/Spiking-Transformer Processor with Big-Little Network and Implicit Weight Generation for Large Language Models," *ISSCC*, pp. 368-370, 2024.

[4] R. Guo et al., "A 28nm 74.34TFLOPS/W BF16 Heterogenous CIM-Based Accelerator Exploiting Denoising-Similarity for Diffusion Models," *ISSCC*, pp. 362-364, 2024.

[5] Y. Qin et al., "A 52.01TFLOPS/W Diffusion Model Processor with Inter-Time-Step Convolution-Attention-Redundancy Elimination and Bipolar Floating-Point Multiplication," *IEEE Symp. VLSI Circuits*, pp. C304-C305, 2024.

[6] D. Shin et al., "DNPU: An 8.1TOPS/W reconfigurable CNN-RNN processor for general-purpose deep neural networks," *ISSCC*, pp. 240-241, 2017.

[7] ARM, https://armkeil.blob.core.windows.net/developer/Files/pdf/white-paper/big-little-technology-the-future-of-mobile.pdf, Accessed: Sep. 2024.

[8] H. Wang et al., "Bitnet: Scaling 1-bit transformers for large language models," arXiv:2310.11453, 2023.

[9] R. Rombach et al., "High-Resolution Image Synthesis with Latent Diffusion Models," *CVPR*, 2022.

[10] Z. Xiao et al., "Tackling the Generative Learning Trilemma with Denoising Diffusion GANs," *ICLR*, 2022.

979-8-3315-4101-9/25/$31.00 ©2025 IEEE