23.3 EdgeDiff: 418.4mJ/Inference Multi-Modal Few-Step Diffusion Model Accelerator with Mixed-Precision and Reordered Group Quantization

Sangjin Kim, Jungjun Oh, Jeonggyu So, Yuseon Choi, Sangyeob Kim, Dongseok Im, Gwangtae Park, Hoi-Jun Yoo

KAIST, Daejeon, Korea

The increasing demand for image generation on mobile devices [1] highlights the need for high-performing image-generative models, including the diffusion model (DM) [2, 3]. A conventional DM requires numerous UNet-based denoising timesteps (~50), leading to high computation and external memory access (EMA) costs. Recently, the Few-Step Diffusion Model (FSDM) [4] was introduced, as shown in Fig. 23.3.1, to reduce the denoising timesteps to 1-4 through knowledge distillation, while maintaining high image quality, reducing computations and EMA by 22.0× and 42.3×, respectively. However, prior diffusion-model architectures, which accelerated many steps of a DM [5, 6] through inter-timestep redundancy in the UNet, fail to speed up the few denoising steps of a FSDM due to the lack of redundancy between timesteps. Moreover, a multi-modal DM introduces additional computational costs for the encoder, and a FSDM shifts computational bottlenecks from the UNet to the encoder and decoder. Additionally, a FSDM becomes more sensitive to quantization due to increased precision demands with fewer denoising steps. To tackle these challenges, we exploit mixed-precision and group quantization [7] as a unified optimization scheme applicable to the encoder, UNet, and decoder in a FSDM, even without inter-timestep redundancy.

Accelerating a FSDM leveraging mixed-precision and group quantization presents three hardware challenges, as shown in Fig. 23.3.1. First, input activation (IA) channel saliency dynamically changes according to input conditioning, making preemptive handling of large-valued outliers [7-9] difficult and necessitating dynamic saliency handling for mixed precision and group quantization. Second, mixed-precision MAC requires both signed and unsigned operations, leading to hardware overhead such as bit-enlarged MAC units with higher power and area consumption [11, 12], and signed-only MAC units with reduced effective bitwidth [13, 14]. Finally, group quantization requires power-intensive FP accumulation and division operations due to the FP scaling factors (SF). To address these three challenges, EdgeDiff introduces three key features: 1) Condition-aware Reordered Group Mixed Precision (CRMP) with the Dynamic Grouping and Reordering Unit (DGRU) for high-performance group-quantized computations with mixed precision. 2) The Compress-and-Add (CAA) PE with bit shuffle tree (BST) for energy-efficient mixed-precision MAC operations. 3) The Tiered Accumulation Unit (TAU) and the Grid-based Quantization Unit (GQU) to minimize the hardware cost of the FP accumulation and division for group quantization.

Figure 23.3.2 shows the overall chip architecture of EdgeDiff. It comprises the DGRU, 8 Mixed Group Precision Cores (MGPC), 1.28MB global memory with a 1D SIMD core, and a top controller. In the DGRU, the Channel Grouping Unit (CGU) determines channel orders for group quantization and mixed precision, and the Reorder and Quantization Unit (RQU) reorders IA and W corresponding to channel orders from the CGU and quantizes them with a GQU. MGPC consists of a 16×16×32 Tensor PE and IA/W/PSUM/Scale memories. Each PE column in Tensor PE consists of a CAA with a BST for 32 4b-4b INT-MACs and TAUs for FP accumulation. Multiple PE columns can be combined in a row or column direction to support higher bit IA or W precision. For example, combining 2, 3, or 4 PE columns supports 8b, 12b, or 16b precision, respectively, with independent precision control over IA and W.

Figure 23.3.3 illustrates CRMP operations and an implementation of the DGRU. In a DM, output activation (OA) values generated by a MGPC typically have similar scales within channels but differ significantly across channels. The channel saliency also fluctuates depending on conditions (prompt or reference image), making it hard to handle outliers, leading to significant quantization errors with differently scaled data in a group. To address this, the RGMP reorders IA and W through the DGRU according to the OA values of the previous layer. This reordering reduces scale differences within groups during group quantization and allows more efficient handling of outliers with mixed precision for dense matrix multiplication in a MGPC without the need for fine-grained precision control. The DGRU operates in two main stages. First, the CGU sorts the 320 saliency values to divide them into 10 groups with 32 channels each. The CGU recursively partitions the array by comparing it with the pivot value (the first value). The CGU checks the upper 4b of each partition's start and end indices for each recursion iteration to employ group-bypassed sorting that skips partitions within the same group, reducing grouping latency by 29.8%. Second, the RQU performs reordering and quantization in a streaming manner while the MGPC loads IA from global memory. The grouping results from the CGU are used for address translation, enabling reordering during the read, and after quantization in the GQU, the data is sent to the IMEM of the MGPC. CRMP improves energy efficiency by 1.49× to 2.72× across the encoder, UNet, and decoder modules of FSDM. CRMP enhances the speed of the few-step diffusion model by 1.61× in text-to-image (T2I) and 1.67× in image-to-image (I2I) benchmarks. While the overall speed improvements reach 15.7× for T2I and 13.1× for I2I compared to conventional massive-step DMs, CRMP plays a key role in further boosting the few-step model's performance.

Figure 23.3.4 shows the concept and implementation of the CAA PE and BST. The mixed precision MAC requires both signed (S) and unsigned (U) operations for W and IA, necessitating four sign-mode configurations (SS, US, SU, UU) by changing the sign ($S_{i,j}$) applied to each bit-wise product. Unlike multiplication with an adder-tree (MAT) PE, CAA mitigates this configuration overhead by simply changing the accumulation order: performing inter-channel addition first, followed by inter-bit addition. The CAA implements a MAC operation with two stages: 1) Inter-channel addition stage accumulation using 16 bit-wise compressors (BCs). Each BC handles one of the 16 possible bit-wise products formed from the 4b IA and 4b W. 2) Inter-bit adder (IBA) performs partial-sum accumulation. Each BC stage consists of 4 8-to-4 compressors that sum the bit-wise product (AND) results across 32 channels. Following this, the shift-add logic in the IBA accumulates the bit-wise partial sums, where the partial sum for the most significant bit of IA and W is either added or subtracted, depending on the sign mode. In contrast to MAT-based designs, which require sign reconfiguration of the multiplier, the CAA architecture introduces minimal area overhead of less than 1.5% by simply using ADD/SUB instead of ADD in the IBA stage. The BST reduces the toggle rate of the compressor logic, which consumes 71.4% of CAA power, by aligning the compressor's input while maintaining the total number of 1 bits. The BST is a three-level tree structure of Unit Shuffling Logic, each composed of an OR gate and an AND gate, and bits are aligned as they pass through the BST. Before shuffling ($\text{Shuffle}_{\text{IN}}$), the bit-wise product (AND) results in randomly distributed 0s and 1s with a probability of 1/4. After shuffling ($\text{Shuffle}_{\text{Out}}$), the 1 bits are aligned, reducing the toggle rate by 1.72×. Combining CAA and BST reduces INT MAC power by 36.6% compared to a MAT-based PE without introducing area overhead.

Figure 23.3.5 shows the detailed implementation of the TAU and GQU. The TAU stacks 2 accumulators: an integer accumulator (I-AC) and a conditionally activated floating-point accumulator (F-AC). Instead of directly accumulating the scaled sum (product of the PSUM from the INT MAC and $\text{SF}_{\text{IA,Man}}$, S_{W}) in the F-AC, the Out-of-range Detection Unit (ORDU) in the I-AC first checks the required range for accumulation. The ORDU detects the used range of the scaled sum, adds it to $\text{SF}_{\text{IA,Exp}}$, and then checks if it fits within the I-AC's range. If the condition is met, the accumulation is performed in the I-AC. However, if the ORDU detects that the scaled sum exceeds the range or an overflow occurs in the I-AC, the scaled sum or accumulation result is redirected to the F-AC. Incorporating a 24b I-AC in the TAU reduces the accumulation energy by 76.2% with only a 3.4% PE area overhead, leading to 80.3% core energy efficiency improvement for 4b operations. The GQU implements the quantizer by adopting 16 comparators with a scale factor grid (SFG) instead of the power-intensive floating-point division logic. The Grid Generate Circuit (GGC) finds the maximum value among the 32 OAs within a group and multiplies it by a bit scale to determine the SF. The SF_{Man} is multiplied by grid values (-7.5, -6.5, -5.5, …, 7.5) to create an SFG broadcasted to 32 Level-Detector Circuits (LDC). Each LDC arithmetically shifts the OA with the difference of SF_{Exp} and OA_{Exp} and compares it against the SFG using 16 comparators. The comparison results in a unary code that provides the 4b quantization result of the OA divided by SF. Additionally, by iteratively subtracting the product of the quantized result and SF_{Man} from the OA, 8b, 12b, and 16b quantization results can be obtained. Overall, the GQU consumes 5.4-to-21.7× less energy and requires 66.7% less area than implementations with FP dividers.

Figure 23.3.6 shows the measurement results of EdgeDiff and the comparison table. EdgeDiff is evaluated on the MS-COCO dataset [15] using the SDXL-turbo model [4]. For T2I generation, EdgeDiff achieves 1.7-to-1.9× faster generation speed with CRMP and shows only <0.5 FID loss compared to the baseline. Additionally, CAA with BST, TAU, and GQU reduce overall power consumption by 41.6%. Previous DM processors [5, 6] rely on inter-timestep redundancy to accelerate only the UNet, overlooking acceleration of the encoder and decoder. In contrast, EdgeDiff end-to-end optimizes energy consumption across the Encoder, UNet, and Decoder using CRMP, resulting in 3.3-to-6.8× lower energy consumption for a T2I task vs. prior work [5, 6]. Additionally, EdgeDiff supports multi-modal benchmarks by incorporating multiple encoder networks (Image and Text) and demonstrates I2I generation performance with >30dB PSNR. EdgeDiff is fabricated in 28nm CMOS technology and occupies a 20.25mm² die area, as shown in Fig. 23.3.7. In summary, this work proposes EdgeDiff, a 418.4mJ/inference diffusion model accelerator with multi-modal few-step denoising for mobile devices applications.

2025 IEEE International Solid-State Circuits Conference (ISSCC) | 979-8-3315-4101-9/25/$31.00 ©2025 IEEE | DOI: 10.1109/ISSCC49661.2025.10904594

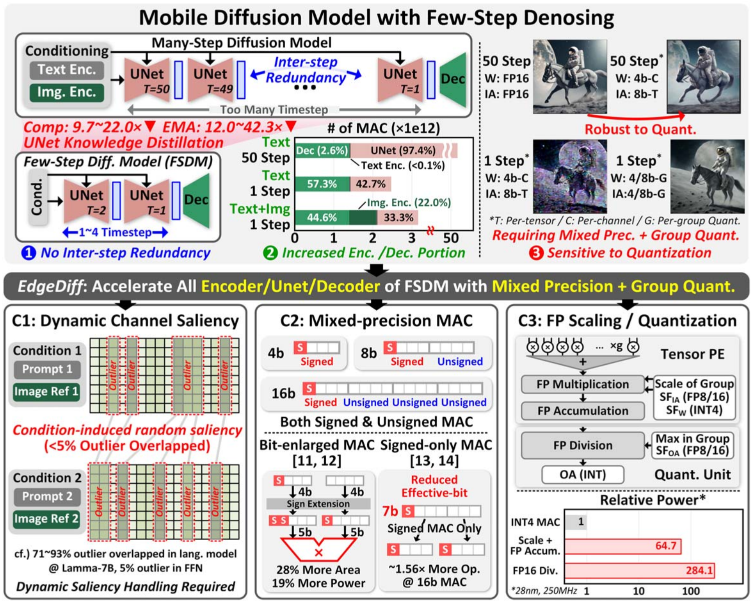

Mobile Diffusion Model with Few-Step Denosing
Conditioning
Text Enc.
Img. Enc.
Many-Step Diffusion Model
UNet T=50
UNet T=49
Inter-step Redundancy
UNet T=1
Dec
Too Many Timestep
Comp: 9.7~22.0×▼ EMA: 12.0~42.3×▼
UNet Knowledge Distillation
Few-Step Diff. Model (FSDM)
Cond.
UNet T=2
UNet T=1
1~4 Timestep
❶ No Inter-step Redundancy
# of MAC (×1e12)
Text 50 Step
Dec (2.6%)
UNet (97.4%)
Text Enc. (<0.1%)
Text 1 Step
57.3%
42.7%
Text+Img 1 Step
Img. Enc. (22.0%)
44.6%
33.3%
0 1 2 3 50
❷ Increased Enc. /Dec. Portion
50 Step W: FP16 IA: FP16
50 Step* W: 4b-C IA: 8b-T
Robust to Quant.
1 Step* W: 4b-C IA: 8b-T
1 Step* W: 4/8b-G IA:4/8b-G
*T: Per-tensor / C: Per-channel / G: Per-group Quant.
Requiring Mixed Prec. + Group Quant.
❸ Sensitive to Quantization
EdgeDiff: Accelerate All Encoder/Unet/Decoder of FSDM with Mixed Precision + Group Quant.
C1: Dynamic Channel Saliency
Condition 1
Prompt 1
Image Ref 1
Outlier
Condition-induced random saliency (<5% Outlier Overlapped)
Condition 2
Prompt 2
Image Ref 2
cf.) 71~93% outlier overlapped in lang. model @ Lamma-7B, 5% outlier in FFN
Dynamic Saliency Handling Required
C2: Mixed-precision MAC
4b S Signed
8b S Signed Unsigned
16b S Signed Unsigned Unsigned Unsigned
Both Signed & Unsigned MAC
Bit-enlarged MAC [11, 12]
Signed-only MAC [13, 14]
4b 4b
Sign Extension
5b 5b
Reduced Effective-bit
7b S
Signed MAC Only
28% More Area 19% More Power
~1.56× More Op. @ 16b MAC
C3: FP Scaling / Quantization
×g
Tensor PE
FP Multiplication
Scale of Group SF_IA (FP8/16) SF_W (INT4)
FP Accumulation
FP Division
Max in Group SF_OA (FP8/16)
OA (INT)
Quant. Unit
Relative Power*
INT4 MAC 1
Scale + FP Accum. 64.7
FP16 Div. 284.1
*28nm, 250MHz 1 10 100

Figure 23.3.1: Characteristics and challenges of few-step diffusion model.

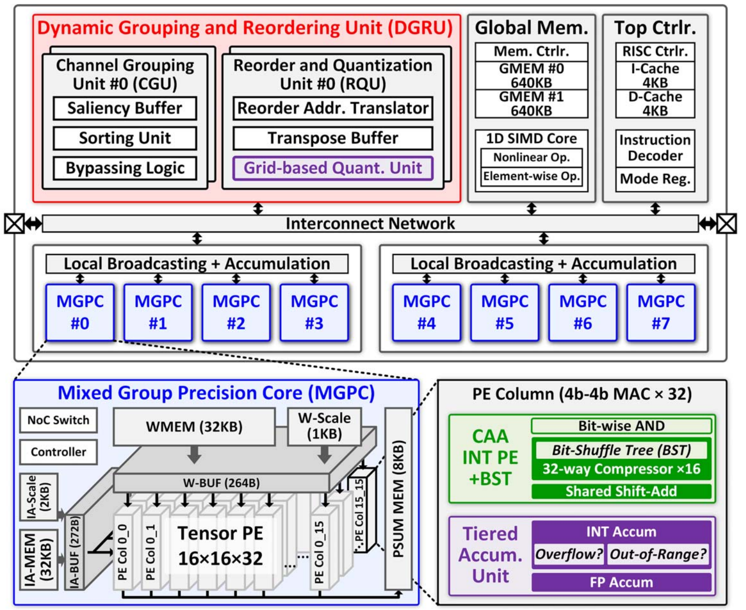


Figure 23.3.2: Overall chip architecture of proposed EdgeDiff.

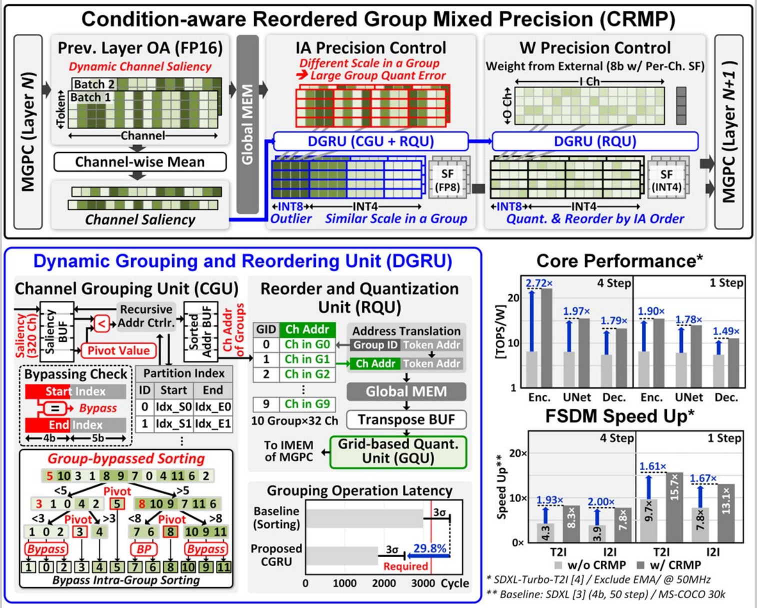


Figure 23.3.3: Operation of CRMP and detailed architecture of CGU and RQU.

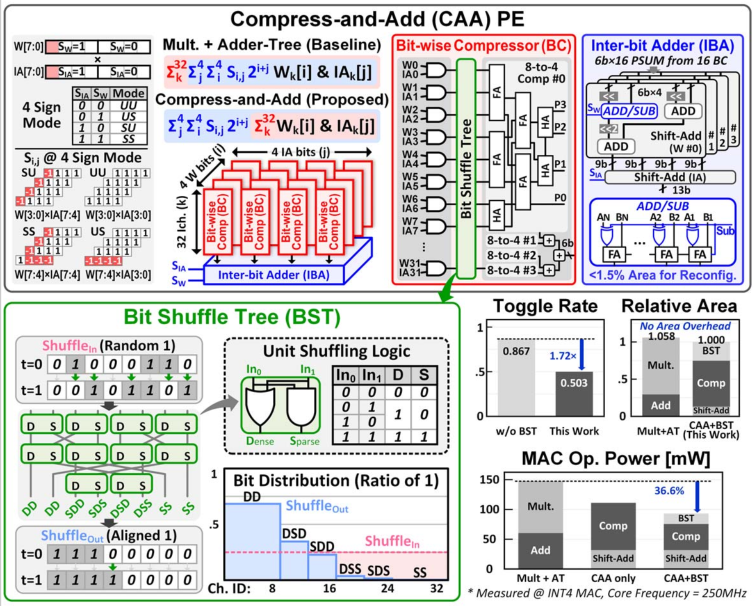


Figure 23.3.4: Concept and implementation of CAA PE with BST.

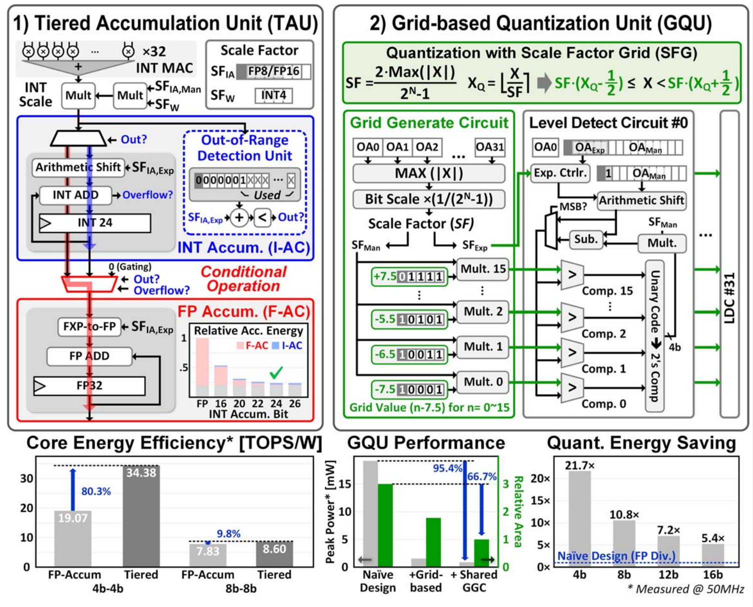
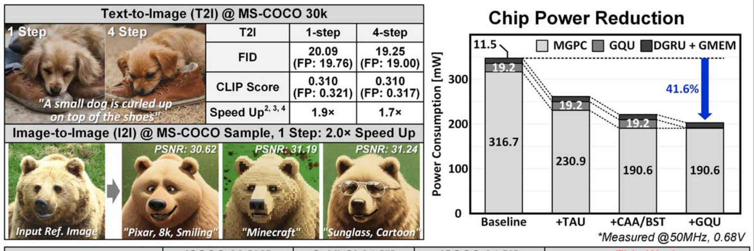


Figure 23.3.5: Operation and detailed implementation of TAU and GQU.



Text-to-Image (T2I) @ MS-COCO 30k

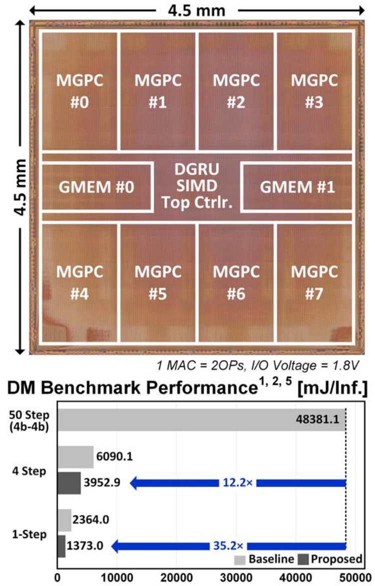
T2I	1-step	4-step
FID	20.09 (FP: 19.76)	19.25 (FP: 19.00)
CLIP Score	0.310 (FP: 0.321)	0.310 (FP: 0.317)
Speed Up$^{2, 3, 4}$	1.9×	1.7×

Image-to-Image (I2I) @ MS-COCO Sample, 1 Step: 2.0× Speed Up

		ISSCC 22 [10]	S. VLSI 24 [5]	ISSCC 24 [6]	This Work
Technology (nm)		28	22	28	**28**
Supply Voltage (V)		0.56-1.1	0.6-1.0	0.6-1.0	**0.68-1.0**
Frequency (MHz)		50-510	120-540	50-540	**50-250**
Die Area (mm^2)		6.82	3.70	3.67	**20.25**
Function		Vision Transformer	Diffusion Model	Diffusion Model	**Few-step Diffusion Model**
Diffusion Model Support	Full DM	X	X (UNet Only)	X (UNet Only)	**O (Encoder + UNet + Decoder)**
	Denoising	X	Multi-step Only	Multi-step Only	**+ Few-step Model**
	Modality	X	Unconditional	Unconditional	**Conditional: Text, Image**
Precision		INT12	Approx. FP16	INT+FP16/BF16	**INT 4/8/12/16 + Group Quant.**
Performance [TOPS]		0.52-4.07 (INT12)	0.47-12.02 (FP16)	6.636 (BF16) 4.424 (FP16)	**31.3 (INT4) / 7.8 (INT8) 3.05 (INT 12) / 1.95 (INT16)^1**
Energy Efficiency [TOPS/W]	Peak	1.91-27.56 (INT 12)	1.67-52.01 (FP16)	73.34 (BF16) 67.89 (FP16)	**39.9 (INT4) / 34.4 (INT4-GQ) 10.0 (INT8) / 8.6 (INT8-GQ)^2**
	Benchmark	8.2 (ImageNet ViT-B)	-	58.01-67.89 (50 Step)	**14.2-15.5 (1 Step - 4 Step)$^{2, 3}$**
Generation Energy [mJ/Inf.]$^{3, 4}$		-	>1476^6 (1 Step) >2847^5 (50 Step)	>1400^5 (50 Step, Unet Only)	**418.4 (1 Step) / 786.3 (4 Step)^2**

1) @250MHz, 1.0V 2) @50MHz, 0.68V 3) Measured @ SDXL-Trubo-T2I, 512×512 Image Size 4) EMA is excluded Energy estimated using reported maximum efficiency 5) w/ inter-timestep redundancy 6) w/o inter-timestep redundancy

Figure 23.3.6: Measurement results and performance comparison table.



		Specifications		
Technology		Samsung 28nm 1P8M CMOS		
Chip Area		4.5 mm × 4.5 mm (20.25 mm^2)		
SRAM		Core	Global	Others
		600 KB	1280 KB	72KB
Supply Voltage		0.68-1.0 V		
Frequency		50-250MHz		
Data Type		GEMM	Scale Factor	Accum.
		INT 4/8/12/16	IA: FP8/16 W: INT4	FP32
Quantization		Per-group (32)	Per-channel	
Energy Efficiency[1] [TOPS/W]	INT4	34.4	39.9	
	INT8	8.6	10.0	
Diffusion Mode Performance				
Task[2]		Text-to-Image Generation	Image-to-Image Generation	
Energy[1,3] [mJ]	1 Step	418.4	506.5	
	4 Step	786.3	846.0	
Generation Time[4, 5] [s]	1 Step	2.12	2.31	
	4 Step	7.33	7.52	

1) @50MHz, 0.68V 2) @ SDXL-Trubo, 512×512 Image
3) Excluding EMA , 4) @250MHz, 1.0V
5) Including EMA (Estimated w/ DDR3 SDRAM)

Figure 23.3.7: Chip micrograph and performance summary.

References:

[1] Y. Li et al., "Snapfusion: Text-to-image Diffusion Model on Mobile Devices Within Two Seconds," *NeurIPS*, 2024.
[2] J. Ho et al., "Denoising Diffusion Probabilistic Models," *NeurIPS*, 2020.
[3] D. Podell et al., "SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis," *ICLR*, 2024.
[4] A. Sauer et al., "Adversarial Diffusion Distillation," Arxiv Preprint, 2023.
[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*, 2024.
[6] R. Guo et al., "A 28nm 74.34TFLOPS/W BF16 Heterogenous CIM-Based Accelerator Exploiting Denoising-Similarity for Diffusion Models," *ISSCC*, 2024.
[7] Y. Zhao et al., "Atom: Low-Bit Quantization for Efficient and Accurate LLM Serving," *Machine Learning and Systems*, pp. 196-209, 2024.
[8] J. Lin et al., "AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration," *Machine Learning and Systems*, pp. 87-100, 2024.
[9] G. Xiao et al., "Smoothquant: Accurate and Efficient Post-Training Quantization for Large Language Models," *ICML*, 2023.
[10] Y. Wang et al., "A 28nm 27.5TOPS/W Approximate-Computing-Based Transformer Processor with Asymptotic Sparsity Speculating and Out-of-Order Computing," *ISSCC*, 2022.
[11] S. Ryu et al., "Bitblade: Area and Energy-Efficient Precision-Scalable Neural Network Accelerator with Bitwise Summation" *IEEE/ACM DAC*, 2019.
[12] D. Han "HNPU: An Adaptive DNN Training Processor Utilizing Stochastic Dynamic Fixed-Point and Active Bit-Precision Searching," *IEEE JSSC*, vol. 56, no. 9, pp. 2858-2869, 2021.
[13] D. Im et al., "Sibia: Signed Bit-Slice Architecture for Dense DNN Acceleration with Slice-Level Sparsity Exploitation," *IEEE HPCA*, 2023.
[14] D. Im et al., "LUTein: Dense-Sparse Bit-Slice Architecture With Radix-4 LUT-Based Slice-Tensor Processing Units," *IEEE HPCA*, 2024.
[15] T.-Y. Lin et al., "Microsoft COCO: Common Objects in Context," *ECCV*, 2014.