Work-in-Progress: I-FlashAttention: Fully Integer Fused Attention for Efficient Vision Transformers

Sehyeon Oh
osehn@etri.re.kr
University of Science and Technology
Daejeon, Republic of Korea

Yongin Kwon
yongin.kwon@etri.re.kr
Electronics and Telecommunications Research Institute
Daejeon, Republic of Korea

Jemin Lee[*]
leejaymin@etri.re.kr
Electronics and Telecommunications Research Institute
Daejeon, Republic of Korea

Abstract

Transformer self-attention offers strong expressiveness, but its compute and memory cost grows rapidly with longer sequences. This results in frequent off-chip memory access, which becomes a major performance bottleneck. FlashAttention reduces this by dividing the sequence into tiles, computed entirely in on-chip memory. This avoids storing intermediate tensors off-chip and alleviates memory bandwidth issues. However, tile-wise online softmax requires floating-point operations for numerical stability using max-based scaling and accumulation. We propose I-FlashAttention, an integer-only version of FlashAttention. It uses shift-based exponential approximation and integer max-tracking to perform online softmax without floating point. All steps, from INT8 GEMM to output, are fused into a single Triton kernel. I-FlashAttention is 1.08× faster than FP16 FlashAttention and 7.10× faster than I-ViT.

CCS Concepts

• **Computing methodologies → Artificial intelligence**.

Keywords

Quantization, Attention, GPU Kernels

ACM Reference Format:
Sehyeon Oh, Yongin Kwon, and Jemin Lee. 2025. Work-in-Progress: I-FlashAttention: Fully Integer Fused Attention for Efficient Vision Transformers. In *International Conference on Compilers, Architecture, and Synthesis for Embedded Systems (CASES '25), September 28-October 3, 2025, Taipei, Taiwan.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3742872.3757072

1 Introduction

Transformer self-attention offers high expressiveness, but its compute and memory complexity scales as $O(N^2)$ with sequence length. This leads to significant bottlenecks caused by intermediate tensor transfers between on-chip and off-chip memory, particularly in large language models (LLMs) and Vision Transformers (ViTs). FlashAttention [2] mitigates this by tiling the sequence and performing computations entirely in on-chip memory. Only the final outputs are written off-chip, while intermediate tensors remain on-chip by accumulating tile-level results. However, FlashAttention and its successors cannot compute softmax in a single pass due to tiling. They process tiles sequentially, updating the maximum value for numerical stability and rescaling previous tile results accordingly. This approach demands high precision and inherently depends on floating-point operations. While prior works such as QAttn [3] and Int-FlashAttention [1] attempt to introduce integer arithmetic, they typically limit quantization to simulated integer GEMM, as shown in Figure 1. In this paper, we propose I-FlashAttention, a GPU kernel where all operations are performed in integer arithmetic. First, we implement an integer-based online softmax [5] using shift-based exponential approximation and tile-wise max tracking, enabling rescaling of prior tile outputs without floating-point dependency. This minimizes precision loss while preserving numerical stability. Second, we integrate the integer online softmax into the FlashAttention tiling structure, handling all stages from INT8 GEMM to output in a single Triton kernel [6]. This reduces off-chip memory access and improves GPU utilization through kernel fusion. Experiments show that I-FlashAttention achieves a 1.08× speedup over FP16 FlashAttention and 7.1× lower latency compared to I-ViT.

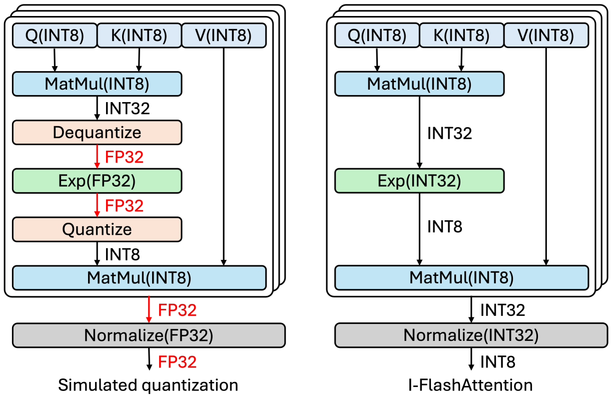

Figure 1: Comparison of simulated quantized attention and the proposed I-FlashAttention pipeline.

2 Integer Algorithms for I-FlashAttention

I-FlashAttention is an integer-only implementation of the tile-based structure of FlashAttention, with its core components being Shift-Exp2 and the Int-Only Online Softmax. FlashAttention employs a tile-wise online softmax strategy, where the softmax is computed sequentially across tiles while accumulating and updating the running maximum and normalization denominator. To implement this

[*] Corresponding author. Email: leejaymin@etri.re.kr

CASES '25, Taipei, Taiwan

ACM ISBN 979-8-4007-1991-2/2025/09
https://doi.org/10.1145/3742872.3757072

in integer arithmetic, ShiftExp2 approximates the exponential function, and the Int-Only Online Softmax maintains the accumulated values using only integer operations. These two components are central to the design of I-FlashAttention.

2.1 ShiftExp2

Algorithm 1 is based on the ShiftExp operation proposed in I-ViT [4], and approximates the exponential function $\exp 2(x)$ using shift-based operations. The input $\hat{\mathbf{X}}^{(\text{int32})}$ is a tensor in int32 format, and s_X is a scale factor for integer computation. The output is $\hat{\mathbf{Y}}^{(\text{int32})}$, a tensor quantized in int32 format. The algorithm approximates $\exp 2(x)$ by dividing the input by the scale factor to obtain the quotient and remainder, which are then used in the computation.

Algorithm 1 ShiftExp2

```
function ShiftExp2($\hat{\mathbf{X}}^{(\text{int32})}, s_X$)
    $\hat{\mathbf{X}}^{(\text{quot})} \leftarrow \lfloor \hat{\mathbf{X}}^{(\text{int32})} / (-s_X) \rfloor$    ▷ quotient
    $\hat{\mathbf{X}}^{(\text{rem})} \leftarrow \lfloor -(\hat{\mathbf{X}}^{(\text{int32})} - \hat{\mathbf{X}}^{(\text{quot})} \cdot (-s_X)) \rfloor$    ▷ remainder
    $\hat{\mathbf{Y}}^{(\text{int32})} \leftarrow \left( (-\hat{\mathbf{X}}^{(\text{rem})} \gg 1) + s_X \right) \gg \hat{\mathbf{X}}^{(\text{quot})}$
    return $\hat{\mathbf{Y}}^{(\text{int32})}$
end function
```

2.2 Int-Only Online Softmax

Algorithm 2 implements the Int-Only Online Softmax, a core component of I-FlashAttention. This algorithm replaces the tile-based softmax computation in FlashAttention with integer operations, performing softmax sequentially across tiles while accumulating and updating the running maximum and normalization denominator. The exponential function $\exp 2(x)$ is approximated using ShiftExp2, allowing the entire softmax computation to be executed in the integer domain. The input $\hat{\mathbf{X}}^{(\text{int32})}$ is a tensor in int32 format, and s_X is a scale factor for integer arithmetic. The output $\hat{\mathbf{Y}}^{(\text{int8})}$ is a tensor quantized to int8 format.

Algorithm 2 Int-Only Online Softmax

```
function I-ONLINE-SOFTMAX($\hat{\mathbf{X}}^{(\text{int32})}, s_X$)
    $\hat{m}_0 \leftarrow -2^{31},\ \hat{d}_0 \leftarrow 1$
    for $j \leftarrow 1, V$ do
        $\hat{m}_j \leftarrow \max(\hat{m}_{j-1}, \hat{\mathbf{X}}_j^{(\text{int32})})$
        $\hat{d}_j \leftarrow \left\lfloor \hat{d}_{j-1} \cdot \frac{\text{ShiftExp2}(\hat{m}_{j-1} - \hat{m}_j, s_X)}{s_X} \right\rfloor + \text{ShiftExp2}(\hat{\mathbf{X}}_j^{(\text{int32})} - \hat{m}_j, s_X)$
    end for
    for $i \leftarrow 1, V$ do
        $\hat{\mathbf{Y}}_i^{(\text{int8})} \leftarrow \left\lfloor 127 \cdot \frac{\text{ShiftExp2}(\hat{\mathbf{X}}_i^{(\text{int32})} - \hat{m}_V, s_X)}{\hat{d}_V} \right\rfloor$
    end for
    return $\hat{\mathbf{Y}}^{(\text{int8})}$
end function
```

3 Experiments

Figure 2 presents the throughput comparison measured on an RTX 2080Ti with two different input sizes. I-FlashAttention achieves a speedup of 7.10× over I-ViT and 1.08× over FP16 FlashAttention. Its throughput is lower than that of QAttn and INT-FlashAttention, as integer-based softmax does not offer performance benefits on GPU. Nevertheless, I-FlashAttention employs a fully integerized architecture including softmax, making it suitable for integer-only hardware. It also maintains a comparable level of accuracy to QAttn, with an SQNR of 33.30 and MSE of 1.23×10^3.

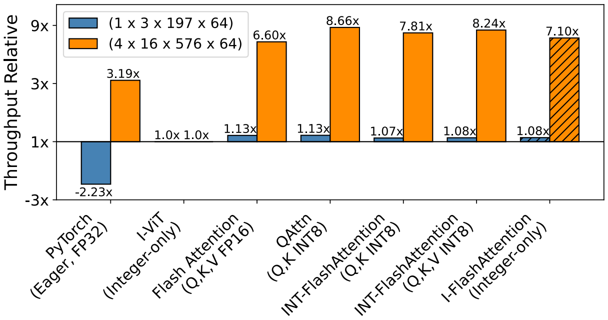

Figure 2: Normalized throughput comparison across different attention methods using DeiT-Tiny and ViT-L/16 input sizes.

4 Conclusion

In this paper, we propose I-FlashAttention to address both the dependency on floating-point operations in self-attention and the bottleneck caused by off-chip memory accesses. By employing a shift-based exponential approximation and integer online softmax, intermediate results are processed entirely using integer operations. These components are integrated into the FlashAttention tiling structure, effectively minimizing off-chip memory accesses. As a result, I-FlashAttention achieves both high computational efficiency and numerical accuracy in a fully integer-only environment.

Acknowledgments

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP)grant funded by the Korea government (MSIT) (No.RS-2024-00459797, No.RS-2023-00277060, No.RS-2025-02214497)

References

[1] Shimao Chen, Zirui Liu, Zhiying Wu, Ce Zheng, Peizhuang Cong, Zihan Jiang, Yuhan Wu, Lei Su, and Tong Yang. 2024. Int-flashattention: Enabling flash attention for int8 quantization. *arXiv preprint arXiv:2409.16997* (2024).

[2] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems* 35 (2022), 16344–16359.

[3] Piotr Kluska, Adrián Castelló, Florian Scheidegger, A Cristiano I Malossi, and Enrique S Quintana-Ortí. 2024. Qattn: Efficient gpu kernels for mixed-precision vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3648–3657.

[4] Zhikai Li and Qingyi Gu. 2023. I-vit: Integer-only quantization for efficient vision transformer inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 17065–17075.

[5] Maxim Milakov and Natalia Gimelshein. 2018. Online normalizer calculation for softmax. *arXiv preprint arXiv:1805.02867* (2018).

[6] Philippe Tillet, Hsiang-Tsung Kung, and David Cox. 2019. Triton: an intermediate language and compiler for tiled neural network computations. In *Proceedings of the 3rd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages*. 10–19.