An In-Memory Computing Accelerator with Reconfigurable Dataflow for Multi-Scale Vision Transformer with Hybrid Topology

Zhiyuan Chen[1], Yufei Ma[2,1,*], Keyi Li[1,2], Yifan Jia[1,2], Guoxiang Li[1,2], Meng Wu[1], Tianyu Jia[1], Le Ye[1,2], and Ru Huang[1,2]
[1]School of Integrated Circuits, Peking University, Beijing, China
[2]Institute for Artificial Intelligence, Peking University, Beijing, China
Email: {yufei.ma@pku.edu.cn}

Abstract

Transformer models equipped with multi-head attention (MHA) mechanism have demonstrated promise in computer vision (CV) tasks, i.e., vision transformers (ViTs). Nevertheless, the lack of inductive bias in ViTs leads to substantial computational and storage requirements, hindering their deployment on resource-constrained edge devices. To this end, multi-scale hybrid models are proposed to take the advantages of both transformers and convolutional neural networks (CNNs). However, existing domain-specific architectures focus on the optimization of either convolution or MHA at the expense of flexibility. In this work, an in-memory computing (IMC) accelerator is proposed to efficiently accelerate ViTs with hybrid MHA and convolution topology by introducing pipeline reordering. SRAM-based digital IMC macro is utilized to mitigate memory access bottleneck, while avoiding analog non-ideality. The reconfigurable processing engines and interconnections are investigated to enable the adaptable mapping of both convolution and MHA. Under typical workloads, experimental results exhibit that our proposed IMC architecture delivers 2.20× to 2.52× speedup and 40.6% to 74.8% energy reduction compared with the baseline design.

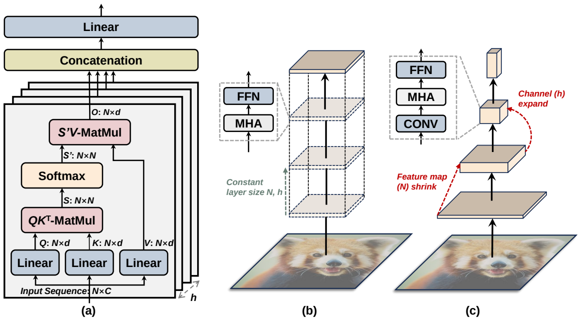


Figure 1: (a) Multi-head attention mechanism. (b) Standard vision transformer structure. (c) Multi-scale vision transformer structure.

1 Introduction

Transformer models equipped with multi-head attention (MHA) mechanism as shown in Figure 1(a) have significantly transformed the conventional paradigm of deep learning research. Initially implemented for machine translation, transformer models have surpassed traditional recurrent neural networks (RNNs) in most natural language processing (NLP) tasks [1]. Furthermore, recent works have also demonstrated their effectiveness in the realm of computer vision (CV), i.e., vision transformers (ViTs) [2], thereby challenging the dominance of convolutional neural networks (CNNs).

However, unlike CNNs, which inherently possess inductive bias, standard vision transformers [2] as shown in Figure 1(b) lack image-specific prior knowledge and require more redundant parameters and computation to learn the visual representations, limiting their application on resource constraint edge devices. The MHA mechanism, i.e., the bottleneck component of transformer models, involves matrix multiplication (MatMul) with quadratic complexity to the sequence length, exacerbating this problem. To remedy this issue, multi-scale models with hybrid network typologies such as CvT[3] and MobileViT[4] have been proposed to leverage the strengths of both transformer and CNNs as shown in Figure 1(c). These multi-scale hybrid models differ from standard ViTs with columnar structures, as they employ convolutional layers to progressively downsample the feature maps and increase the number of channels stage by stage. This results in variable layer sizes and reduces the operation proportion of MHA within the overall ViTs.

In-memory computing (IMC) is a promising design paradigm that has proven to be efficient for MatMul computations [5]. It mitigates the memory access overhead by incorporating arithmetic operations into the memory. Recently, several domain-specific architectures based on either IMC or conventional digital processing engines (PEs) have been proposed to accelerate transformer models [6–8]. However, these works typically focus on the optimization of MHA through operation fusion and pruning techniques without considering compatibility with convolutional layers. Furthermore, the efficiency and performance of previous works heavily rely on inflexible dataflow and parallelism schemes, which may perform well for standard ViTs with fixed sequence lengths and numbers of heads, but encounter substantial performance degradation when the layer size of the MHA mechanism varies across different models. Consequently, these approaches do not generalize to emerging ViT models that incorporate both MHA and convolutional layers in their hybrid network topology.

In this paper, an IMC accelerator with reconfigurable dataflow is proposed to meet the demand for flexibility of multi-scale ViTs with hybrid topology. The SRAM-based digital IMC macro is utilized to improve the energy efficiency for both convolution and MHA, which avoids the non-idealities of analog computing and costly conversions between analog and digital signals. To minimize redundant memory accesses and maximize utilization, we exploit reconfigurability at both the level of the IMC engine and the global interconnect. The main contributions of this paper are summarized as follows:

[*]Corresponding Author: Yufei Ma.

- The pipeline reordering of MHA is introduced to implement the softmax in a two-stage method, which reduces the interconnection complexity of IMC macros.
- A reconfigurable IMC-based engine is designed that incorporates hybrid computation patterns to favor both convolution and MHA. Especially, it leverages fused MatMul computation of MHA to eliminate redundant intermediate data transfer with quadratic complexity.
- Based on the reordered MHA pipeline, a flexible distributor network is proposed to orchestrate IMC engines and enable adaptable mapping for both convolutional layer and MHA mechanism, which maintains high utilization for varying layer sizes.
- Evaluation on TSMC 22nm demonstrates that the proposed accelerator achieves 2.20× to 2.52× speedup and 40.6% to 74.8% energy saving for MHA mechanism workloads and delivers 44.1% to 55.9% Energy-Delay-Product (EDP) reduction for typical multi-scale ViT models, compared with baseline IMC accelerator.

2 Background and Motivation

2.1 Vision Transformer Algorithms

Pioneer research has demonstrated that a transformer network can be applied to image recognition tasks with minimal reliance on image-specific prior knowledge [2]. The multi-head attention (MHA) mechanism in standard transformer block is illustrated in Figure 1(a). After linear projection, the query (Q), key (K), and value (V) matrices are generated from the input N-token sequence. Then, matrix Q is multiplied by the transpose of matrix K to compute the score matrix S. The row-wise softmax is performed to generate the matrix S'. Finally, matrix S' and matrix V are multiplied to compute the output matrix O. The computation and bandwidth requirements of MHA are quadratic to the sequence length N, which is usually considered as the performance bottleneck component of transformer execution.

However, compared to convolutional neural networks, which have dominated computer vision tasks for the last decade, standard vision transformer models suffer from a lack of spatial inductive bias and fail to efficiently extract information from local receptive fields. This not only increases the cost of training, but also makes standard ViTs tend to stack more parameters to learn corresponding visual representations than CNNs, which further hinder the deployment of ViTs on edge platforms with constrained power budget and hardware resources.

To mitigate this problem, inductive bias is introduced to construct efficient and lightweight multi-scale transformer backbone inspired by CNNs. Swin Transformer [9] replaces standard MHA with window-based attention mechanism, where only the correlation of tokens belong to the same window is considered. PVT [10] incorporates a pyramid structure to progressively shrink the scale of feature maps by patch embedding layers. CvT [3] employs convolutional projection instead of linear projection to reduce the amount of computation. MobileViT [4] proposes to replace local processing in convolutions with global processing using transformers to obtain higher accuracy than CNNs and standard ViT with a similar number of parameters.

In general, these emerging multi-scale ViT models achieve similar or even better recognition accuracy with smaller model sizes compared to standard ViTs [2] by the following innovations,

1) Hybrid network topology: unlike standard ViTs, which only use convolution as the first stem layer, most multi-scale ViTs integrates more convolutional layers, which may make MHA less computationally dominant. Thus, existing architectures that only accelerate MHA are not efficient enough for these models;

2) Varying hyper-parameters: standard ViTs follow a columnar structure with fixed sequence length N and head number h. In contrast, the hyper-parameters of multi-scale ViTs may differ from stage by stage, as shown in Figure 1(c). As a result, reconfigurable parallelism is required to support both MHA and convolution with highly flexibility.

2.2 Transformer Accelerator

Several accelerators based on conventional digital processing engines or in-memory computing techniques have been proposed to accelerate the transformer models. ELSA [6] adopts a novel approximation scheme to dynamically filter out unimportant computation of MHA. In TransCIM [8], bitline-transpose compute-in-memory macro is utilized to accelerate transformer models with structured sparsity. COSA [7] employs systolic arrays and hybrid data reuse pattern to fuse operations of MHA. RAWAtten [11] is tailored for window-based attention in Swin Transformer with hybrid near-memory computing and digital PE datapath.

However, these works only focus on optimizing the MHA mechanism individually, without considering its compatibility with convolutional layers. In addition, even within the context of MHA, these works are usually designed with fixed dataflow and parallelism. This can lead to reduced resource utilization and efficiency when the number of heads h and sequence length N vary across different layers. As a result, they lack the flexibility needed to effectively support multi-scale ViTs involving hybrid topology and hierarchical features.

In this work, we propose an SRAM-based digital IMC accelerator for multi-scale ViTs, focusing on exploiting the reconfigurable dataflow from two perspectives. Firstly, a hybrid computation pattern is explored, which aims to efficiently support both convolutional layer and MHA mechanism, while simultaneously minimizing memory access requirements. Secondly, we employ a configurable interconnection that facilitates flexible mapping for both convolution and MHA, thereby enhancing resource utilization.

3 Pipeline Reordering of Attention

Due to the multi-scale characteristics, the layer size of both convolution and MHA differs from stage by stage. This can lead to a lower utilization rate when the parallelism of hardware is fixed. However, a reconfigurable distributor network with fat-tree interconnection topology has been shown to perform well in accelerating IMC-based CNN[12], because it allows for adaptable mapping of varying kernel sizes and channel numbers.

For MHA mechanism, a similar challenge emerges with variations in the number of heads h and sequence length N. Consequently, a flexible mapping strategy is necessary, where the number of IMC macros computing the same head can be modulated dynamically. Nevertheless, the softmax function within the original MHA execution pipeline entails global-reduction-and-broadcast operations. This leads to potentially complex interconnections and unnecessary data communication redundancy in its implementation. As depicted in Figure 2(a), the K and V matrices are split into multiple data segments that are stored in different IMC macros. The maximum value in each row of the score matrix S must be identified to maintain the precision of the exponential function. Moreover,

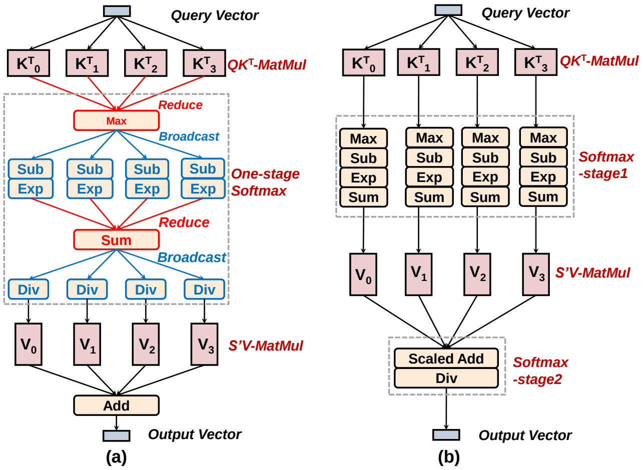


Figure 2: (a) Original Execution Pipeline and (b) Reordered Execution Pipeline of MHA.

the results of the exponential function should be summed up to establish the scale factor.

In order to tackle this problem, we employ a formula transformation akin to [13] to reorder the execution pipeline of MHA, as illustrated in Algorithm 1. Rather than attempting to fully complete the softmax immediately following the matrix multiplication of Q and K^{T}, we utilize a two-stage scheme to distribute computation between IMC macros and connection nodes within the distributor network. In the first stage between each pair of IMC macros, the segments of the score matrix S are streamed directly to subsequent processing without waiting for the computation of global maximum and summation. The second stage is conducted by connection nodes within the distributor network, which merge the results from IMC macros that are configured to compute the same attention head, as displayed in Figure 2(b). By adopting this approach, a unified fat-tree distributor network can be utilized for both convolution and MHA operations. The IMC macros are only required to process locally generated intermediate data, thereby reducing the overhead associated with interconnection communication.

Algorithm 1: Reordered Attention Pipeline

```
Input: Sequence Length $N$, Embedding Dimension $d$,
       Segment Size $M$, Query Matrix $Q_{N\times d}$, Key Matrix
       $K_{N\times d}$, Value Matrix $V_{N\times d}$
Output: Output Matrix $O_{N\times d}$
for $i \leftarrow 0$ to $N$ do
    $\vec{q} \leftarrow Q[i,:]$;
    for $j \leftarrow 0$ to CEIL($N/M$) do
        /* $QK^{\mathrm{T}}$-MatMul */
        $K_j \leftarrow K[(j+1)M-1 : jM,:]$;
        $\vec{s} \leftarrow \vec{q} \cdot K_j^{\mathrm{T}}$;
        /* First stage of softmax */
        $max_j \leftarrow$ MAX($\vec{s}$);
        $\overrightarrow{exp_j} \leftarrow$ EXP($\vec{s} - max_j$);
        $sum_j \leftarrow$ SUM($\overrightarrow{exp_j}$);
        /* $S'V$-MatMul */
        $V_j \leftarrow V[(j+1)M-1 : jM,:]$;
        $\vec{o}'_j \leftarrow \overrightarrow{exp_j} \cdot V_j$;
    end
    /* Second stage of softmax */
    $max \leftarrow$ -inf;
    $sum \leftarrow 0$;
    $\vec{o}' \leftarrow \vec{0}$;
    for $j \leftarrow 0$ to CEIL($N/M$) do
        $max\_temp \leftarrow max$;
        $max \leftarrow$ MAX($max, max_j$);
        $scale0 \leftarrow$ EXP($max\_temp - max$);
        $scale1 \leftarrow$ EXP($max_j - max$);
        $sum \leftarrow sum \cdot scale0 + sum_j \cdot scale1$;
        $\vec{o}' \leftarrow \vec{o}' \cdot scale0 + \vec{o}'_j \cdot scale1$;
    end
    $O[i,:] \leftarrow \vec{o}'/sum$;
end
```

4 Proposed Architecture

4.1 Architecture Overview

The overall architecture of the proposed IMC accelerator is depicted in Figure 3. This design features customized coarse-grained data transfer and tensor computation instructions to facilitate tightly coupled communication between the host CPU and the accelerator. The command scheduler accepts, decodes, and dispatches streaming commands from the host. Two lightweight Direct Memory Access (DMA) engines are incorporated into the accelerator to move data between off-chip DRAM and on-chip SRAM buffers. During the inference process, the input matrix Q and output tiles are stored separately in two unified buffers.

The digital IMC macros are implemented and organized in pairs, i.e., IMC engines. These engines are used for both convolutional and linear layers (which can be treated as 1×1 convolutions), where they store weight tiles and operate in parallel mode. To reduce intermediate data movement, the pipeline mode is employed to fuse the MatMul operations of MHA. In this scenario, the data segments of matrices K and V are separately stored in two distinct IMC macros.

The IMC engines are interconnected via a distributor network with a fat-tree topology. Connection nodes within this distributor network allow for adaptable mapping and reconfigurable parallelism, accommodating both convolution and MHA operators. The input fetcher loads, reuses, and aligns input activations from one of the unified buffers, converting them into serial bit streams using shift registers. For convolutions, an im2col operation is conducted, flattening the sliding window into a vector format. For MHA, each row of matrix Q is fetched only once and stored in a DFF (D Flip-Flop)-based reuse buffer. This reuse buffer also eliminates redundant SRAM access during the continuous slide of convolutional layers. The distributor network receives these serial input bits and distributes them to each IMC engine. The accumulated computation results are then gathered and transmitted to the post-processing unit. In this unit, partial sum accumulation, requantization, and ReLU operations are performed before writing the results back to the unified buffer.

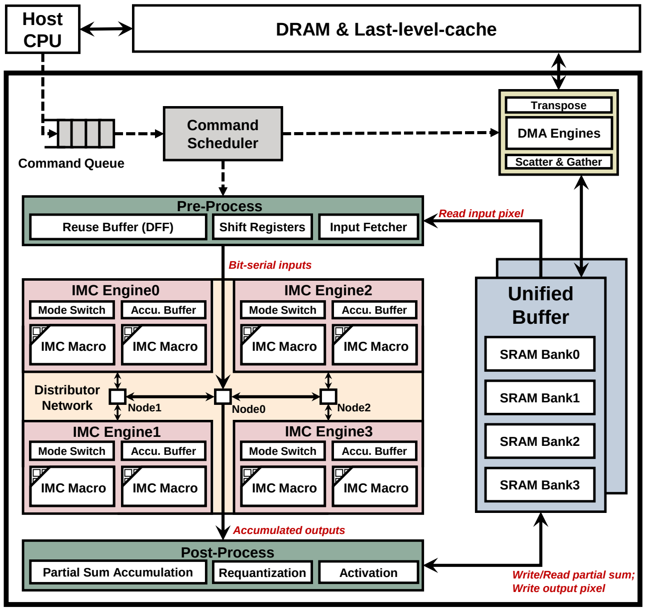


Figure 3: Overall Architecture

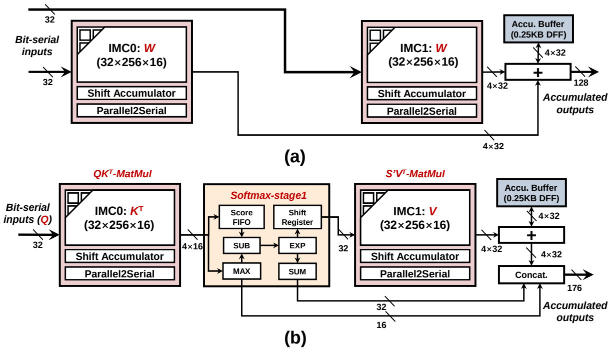


Figure 4: The in-memory computing (IMC) engine configured in (a) the parallel mode and (b) the pipeline mode.

4.2 Reconfigurable IMC Engine

The structure of the IMC engine is illustrated in Figure 4. Each IMC engine is made up of two IMC macros (IMC0 and IMC1) along with their associated peripheral logic. A 32×256 digital IMC macro is implemented, which can perform 32×32 1×8-bit vector-matrix multiplications per cycle. The storage-to-computation ratio is set at 16. Thus, each IMC engine is equivalent to 256 INT8 Multiply-and-Accumulation (MAC) units with 32KB storage. Shift accumulators are employed to accumulate the outputs from the macros.

Each IMC engine supports two modes specifically designed for the convolutional layer and the MHA mechanism, respectively. The parallel mode is leveraged for convolutional and linear layers. In this mode, the results of both IMC macros are summed up to provide sufficient parallelism for the dimension of inner product, as depicted in Figure 4(a).

The pipeline mode, as illustrated in Figure 4(b), is tailored for the MHA mechanism. Considering the quadratic spatial complexity of the score matrix S, the QK^{T}-MatMul and $S'V$-MatMul operations of MHA are fused, eliminating redundant memory access. IMC0 and IMC1 are assigned to store segments of the matrices K^{T} and V, respectively. During the computation of MHA, matrix Q in the unified buffer gets fetched row by row and is fed into each IMC engine via the distributor network. Each row of matrix Q is first multiplied with the matrix K^{T} in IMC0 to compute one row of the score matrix S, which is then pushed into the Score FIFO. Simultaneously, the Max unit searches for the maximum element of the current row of matrix S and sends it to the exponential function (Exp) unit, which is implemented using the methodology in [14]. Next, the Exp unit starts fetching data from the Score FIFO and completes the first stage of the softmax operation, in collaboration with the Sum unit. The results from the Exp unit are streamed to IMC1 and multiplied by a segment of matrix V. The outputs of the Max unit, Sum unit, and IMC1 are returned to the distributor network for the second stage computation of the softmax operation.

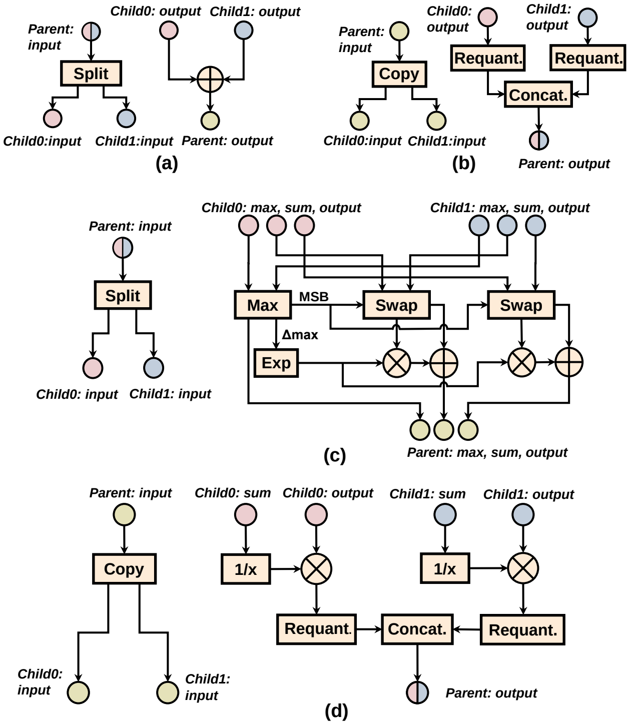


Figure 5: The connection node that configured into (a) SM mode for convolution, (b) CC mode for convolution, (c) SM mode for MHA, and (d) CC mode for MHA.

4.3 Reconfigurable Distributor Network

The connection nodes form the distributor network that governs IMC engines under unified scheduling. Each node can be configured in two modes to cater for both convolution and MHA operations: the split-merge (SM) mode and the copy-concatenate (CC) mode.

As depicted in Figure 5, for convolution, each node in SM mode partitions the input bit-serial data from the parent node and distributes them to the child nodes. Concurrently, the partial sum results from the child nodes are summed up and forwarded to the parent node. When configured in CC mode, the connection node duplicates the input bit-serial data from the parent node for both child nodes. This improves data reuse. The computed results from the child nodes are then combined and transferred to the parent node, where requantization is performed if necessary. Thus, the

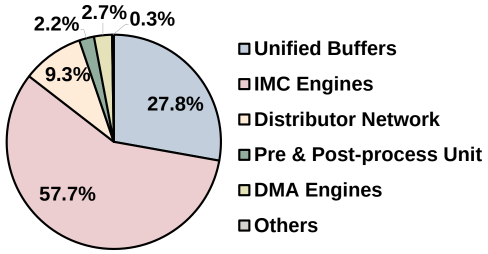


Figure 6: Area breakdown of the proposed accelerator

distributor network facilitates a flexible pattern for input reuse and output accumulation in convolutional layers.

For the MatMul fusion of MHA, connection nodes serve to perform the second stage of the softmax function, as opposed to just simple addition or concatenation as in convolution. IMC engines that compute the same head of MHA are connected by nodes in SM mode, which execute a weighted summation for computation results from child nodes based on Algorithm 1. Connection nodes in CC mode scale the results with the sum of the exponential function and concatenate the final results belonging to different heads.

Non-linear units are implemented following the same methodology as Exp units in IMC engines. These can be configured to conduct either an exponential function or division, according to the mode of the corresponding connection nodes. In this way, the distributor network can dynamically adjust parallelism along the dimension of sequence length N and the number of heads h, thereby accommodating MHA layers with varying layer shapes.

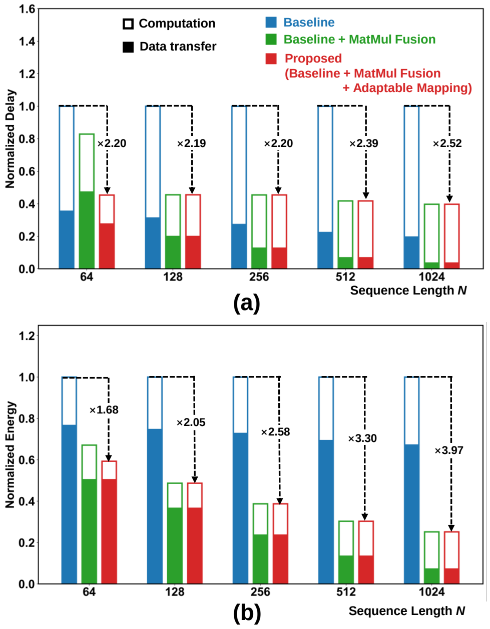
5 Evaluation

5.1 Experimental Setup

The circuit of digital IMC macro is designed with Cadence Virtuoso with transistor level simulation for power estimation and layout placement for area evaluation. Other parts of the proposed accelerator is implemented in RTL Verilog HDL with SRAM generated by Memory Compiler. The accelerator is synthesized by Synopsys Design Compiler under TSMC 22nm technology. The clock frequency is set to 500MHz. Typical design variables are selected, where the number of IMC engines is 4 and the storage of each unified buffer is set to 128KB.

A performance model has been developed using Python, which estimates the delay and energy consumption with a heuristics mapping search scheme and incorporated power information of the IMC macro and SRAM. The dynamic power of off-chip DRAM access are modeled by CACTI7[15], using the typical settings of DDR3-1600.

To study the experimental results in detail and avoid unfair comparison, an IMC-based accelerator is set as the baseline. It follows a similar architecture setting, but only support rigid convolution dataflow without MHA-oriented optimizations, where all the connection nodes are fixed to the SM mode. The MatMul operations of MHA will be treated as two separate 1×1 convolutions in baseline. Three ViT models, e.g., DeiT-Tiny[16], PVT-Tiny[10], and CvT-Small[3], are used as the benchmark for evaluation. Among them, DeiT follows a standard ViT structure while PVT and CvT are multi-scale ViT models with hybrid topology. To provide a comprehensive benchmark, we varied the input image resolution of DeiT and PVT.



(a)

(b)

Figure 7: (a)Delay and (b)Energy analysis for MHA mechanism with diverse sequence length.

5.2 Implementation Results

The synthesis results exhibit that the overall area of the proposed IMC accelerator is 2.18 mm^2. The area breakdown is depicted in Figure 6. The IMC macros and SRAM buffers constitute the largest portions of the area, accounting for 57.7% and 27.8%, respectively. To reorder the pipeline of MHA execution, additional logic units are integrated into the connection nodes of the distributor network to merge and scale the outputs from IMC engines. These include multipliers, comparators, and non-linear units. However, it only takes up 9.3% of the area, and thereby doesn't significantly burden the whole system.

5.3 Efficiency Evaluation for MHA Mechanism

We incrementally increase the sequence length N from 64 to 1024, placing emphasis on the performance enhancement for MHA. The results are presented in Figure 7. Leveraging the MatMul fusion in the IMC engine reduces energy consumption across all instances, primarily by eliminating unnecessary off-chip transfers. This approach proves particularly effective for MHA operations with extended sequence lengths N, resulting in energy savings as high as 74.9% and speed increases up to 2.5×. However, for cases with N not exceeding 64, there is no significant performance improvement. This suggests that MHA operations with shorter sequence lengths may experience severe utilization degradation. To manage this issue effectively, we introduce a reconfigurable distributor network that enables adaptable mapping for any sequence length N. This ensures over 2× speedups across all scenarios, thus enhancing overall system performance.

5.4 Efficiency Evaluation for ViT models

As illustrated in Figure 8, our proposed IMC engines effectively fuse the MatMul operations of MHA to eliminate off-chip DRAM accesses, resulting in significant savings. When compared with multi-scale ViTs such as PVT and CvT, our MatMul fusion approach proves particularly beneficial for DeiT, yielding energy savings

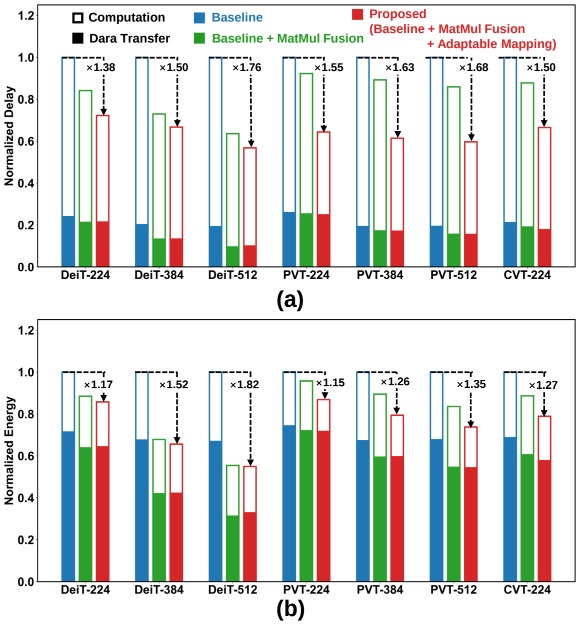


Figure 8: (a)Delay and (b)Energy analysis for overall ViTs.

Table 1: Comparison with other work

Work	SpAtten[17]	Proposed
Technology	40nm	22nm
Frequency(MHz)	1000	500
Area(mm^2)	1.55	2.18
Throughput(GOPS)	360	828.04
Area Effi.(GOPS/mm^2)	238	379.8
Engine	Digital PE	Digital IMC
Target	NLP Transformer	Multi-scale ViT
Sparsity	Yes	No

ranging from 11.5% to 44.5% and speed increases between 1.19× and 1.57×. This result occurs because DeiT adheres to a standard ViT structure where the MHA mechanism plays a pivotal role. Multi-scale ViTs, on the other hand, involve more convolution and linear operations. Despite this, the fixed MatMul fusion dataflow fails to benefit all the MHA operations due to the diverse layer sizes. Fortunately, this issue can be mitigated by our proposed flexible distributor network that enables adaptable mapping.

Although the performance and efficiency of DeiT models are not significantly enhanced through the distributor network, the delay and energy dissipation of PVT and CvT models are further decreased by 24.2% to 31.2% and 9.3% to 11.6%, respectively. These results underscore that our proposed accelerator is tailored to efficiently handle ViT models featuring multi-scale attributes. Taking into account all the proposed optimizations, the Energy-Delay-Product (EDP) is reduced by 38.0% to 68.8% for DeiT, 44.1% to 55.9% for PVT, and 47.5% for CvT, respectively.

5.5 Comparison with Existing Work

Most existing transformer accelerators feature the extreme optimization of MHA with sparsity. We compare the proposed design with one of them in Table 1. Without the benefits from sparsity, the proposed accelerator still outperform SpAtten[17] in area efficiency, thanks to the outstanding computation density of digital IMC. Besides, our design targets at multi-scale and lightweight ViT backbones, which is more suitable for vision tasks on edge.

6 Conclusion

In this paper, we propose an IMC architecture featuring a reconfigurable dataflow, designed explicitly to expedite multi-scale vision transformers. These ViTs employ a hybrid network topology that combines convolutional layers with MHA mechanism. Based on reordered MHA pipeline, reconfigurable IMC engines and distributor network are devised to eliminate redundant data movement and allow for adaptable mapping to favor both convolution and MHA. Experimental results demonstrate that our proposed design achieves a speedup ranging from 2.20× to 2.52× and energy savings between 40.6% to 74.8% for the MHA workloads. Additionally, it offers a reduction in EDP of 44.1% to 55.9% for typical multi-scale ViTs when compared with the baseline IMC accelerator.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China under Grant 2023YFB4404603 and 2020YFB2206002, in part by National Natural Science Foundation of China under Grant 92164301, 62204003, 62225401, and 61927901, in part by Beijing Municipal Natural Science Foundation No.4232067.

References

[1] Ashish Vaswani et al. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[2] Alexey Dosovitskiy et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[3] Haiping Wu et al. Cvt: Introducing convolutions to vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22–31, 2021.

[4] Sachin Mehta and Mohammad Rastegari. Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer. In *International Conference on Learning Representations*, 2022.

[5] Yu-Der Chih et al. 16.4 an 89tops/w and 16.3tops/mm2 all-digital sram-based full-precision compute-in memory macro in 22nm for machine-learning edge applications. In *2021 IEEE International Solid-State Circuits Conference (ISSCC)*, volume 64, pages 252–254, 2021.

[6] Tae Jun Ham et al. Elsa: Hardware-software co-design for efficient, lightweight self-attention mechanism in neural networks. In *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*, pages 692–705, 2021.

[7] Zhican Wang et al. Cosa:co-operative systolic arrays for multi-head attention mechanism in neural network using hybrid data reuse and fusion methodologies. In *2023 60th ACM/IEEE Design Automation Conference (DAC)*, pages 1–6, 2023.

[8] Fengbin Tu et al. Trancim: Full-digital bitline-transpose cim-based sparse transformer accelerator with pipeline/parallel reconfigurable modes. *IEEE Journal of Solid-State Circuits*, 58(6):1798–1809, 2023.

[9] Ze Liu et al. Swin transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9992–10002, 2021.

[10] Wenhai Wang et al. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 548–558, 2021.

[11] Wantong Li et al. Rawatten: Reconfigurable accelerator for window attention in hierarchical vision transformers. In *2023 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 1–6, 2023.

[12] Zhenhua Zhu et al. Mnsim 2.0: A behavior-level modeling tool for processing-in-memory architectures. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 42(11):4112–4125, 2023.

[13] Tri Dao et al. Flashattention: Fast and memory-efficient exact attention with io-awareness. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 16344–16359. Curran Associates, Inc., 2022.

[14] Joonsang Yu et al. Nn-lut: Neural approximation of non-linear operations for efficient transformer inference. In *Proceedings of the 59th ACM/IEEE Design Automation Conference*, DAC '22, page 577–582, New York, NY, USA, 2022.

[15] Rajeev Balasubramonian et al. Cacti 7: New tools for interconnect exploration in innovative off-chip memories. *ACM Trans. Archit. Code Optim.*, 14(2), jun 2017.

[16] Hugo Touvron et al. Training data-efficient image transformers amp; distillation through attention. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 10347–10357, 18–24 Jul 2021.

[17] Hanrui Wang et al. Spatten: Efficient sparse attention architecture with cascade token and head pruning. In *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pages 97–110, 2021.