IEEE Author Portal
IEEE Transactions on Circuits and Systems II: Express Briefs

Sponsoring Society: CAS | IEEE

Special Issue on ISICAS 2026

A heterogeneous hybrid CNN-Transformer accelerator with multi-level Strassen MatMul

Submission ID 6729ac0d-e6f6-4f68-b4de-fb7f6d9d23bf

Submission Version Initial Submission

PDF Generation 31 May 2026 04:53:17 EST by Atypon ReX

Authors

Mr. Yuxiao Wen

Affiliations

- the State Key Laboratory of Semiconductor Physics and Chip Technologies, Institute of Semiconductors, Chinese Academy of Sciences, Beijing 100083, China
- the Beijing Key Laboratory of Semiconductor Intelligent Sensing and Computing, Institute of Semiconductors, Chinese Academy of Sciences, Beijing 100083, China
- the Center of Materials Science and Optoelectronics Engineering, University of Chinese Academy of Sciences, Beijing 100049, China

Mr. Wenyao Zhao

Affiliations

- the State Key Laboratory of Semiconductor Physics and Chip Technologies, Institute of Semiconductors, Chinese Academy of Sciences, Beijing 100083, China
- the Beijing Key Laboratory of Semiconductor Intelligent Sensing and Computing, Institute of Semiconductors, Chinese Academy of Sciences, Beijing 100083, China
- the Center of Materials Science and Optoelectronics Engineering, University of Chinese Academy of Sciences, Beijing 100049, China

Mr. Xuanzhe Xu

ORCiD
https://orcid.org/0009-0007-4927-4183

Affiliations

- the State Key Laboratory of Semiconductor Physics and Chip Technologies, Institute of Semiconductors, Chinese Academy of Sciences, Beijing 100083, China
- the Beijing Key Laboratory of Semiconductor Intelligent Sensing and Computing, Institute of Semiconductors, Chinese Academy of Sciences, Beijing 100083, China
- the Center of Materials Science and Optoelectronics Engineering, University of Chinese Academy of Sciences, Beijing 100049, China

Dr. Xu Yang

ORCiD
https://orcid.org/0000-0001-5944-2987

Affiliations

- the State Key Laboratory of Semiconductor Physics and Chip Technologies, Institute of Semiconductors, Chinese Academy of Sciences, Beijing 100083, China
- the Beijing Key Laboratory of Semiconductor Intelligent Sensing and Computing, Institute of Semiconductors, Chinese Academy of Sciences, Beijing 100083, China
- the Center of Materials Science and Optoelectronics Engineering, University of Chinese Academy of Sciences, Beijing 100049, China

Dr. Zhe Wang

ORCiD
https://orcid.org/0000-0001-8256-2848

Affiliations

- the State Key Laboratory of Semiconductor Physics and Chip Technologies, Institute of Semiconductors, Chinese Academy of Sciences, Beijing 100083, China
- the Beijing Key Laboratory of Semiconductor Intelligent Sensing and Computing, Institute of Semiconductors, Chinese Academy of Sciences, Beijing 100083, China
- the Center of Materials Science and Optoelectronics Engineering, University of Chinese Academy of Sciences, Beijing 100049, China

Prof. Zhao Zhang

ORCiD
https://orcid.org/0000-0002-9009-9045

Affiliations

- the State Key Laboratory of Semiconductor Physics and Chip Technologies, Institute of Semiconductors, Chinese Academy of Sciences, Beijing 100083, China
- the Beijing Key Laboratory of Semiconductor Intelligent Sensing and Computing, Institute of Semiconductors, Chinese Academy of Sciences, Beijing 100083, China
- the Center of Materials Science and Optoelectronics Engineering, University of Chinese Academy of Sciences, Beijing 100049, China

Prof. Jian Liu

ORCiD
https://orcid.org/0000-0001-8057-2444

Affiliations

- the State Key Laboratory of Semiconductor Physics and Chip Technologies, Institute of Semiconductors, Chinese Academy of Sciences, Beijing 100083, China
- the Beijing Key Laboratory of Semiconductor Intelligent Sensing and Computing, Institute of Semiconductors, Chinese Academy of Sciences, Beijing 100083, China
- the Center of Materials Science and Optoelectronics Engineering, University of Chinese Academy of Sciences, Beijing 100049, China

Prof. Nanjian Wu

Affiliations

- the State Key Laboratory of Semiconductor Physics and Chip Technologies, Institute of Semiconductors, Chinese Academy of Sciences, Beijing 100083, China
- the Beijing Key Laboratory of Semiconductor Intelligent Sensing and Computing, Institute of Semiconductors, Chinese Academy of Sciences, Beijing 100083, China
- the Center of Materials Science and Optoelectronics Engineering, University of Chinese Academy of Sciences, Beijing 100049, China

Prof. Runjiang Dou
Corresponding Author
Submitting Author

ORCiD
https://orcid.org/0000-0002-8674-1285

Affiliations

- the State Key Laboratory of Semiconductor Physics and Chip Technologies, Institute of Semiconductors, Chinese Academy of Sciences, Beijing 100083, China
- the Beijing Key Laboratory of Semiconductor Intelligent Sensing and Computing, Institute of Semiconductors, Chinese Academy of Sciences, Beijing 100083, China
- the Center of Materials Science and Optoelectronics Engineering, University of Chinese Academy of Sciences, Beijing 100049, China

Prof. Liyuan Liu

ORCiD
https://orcid.org/0000-0003-2585-323X

Affiliations

- the State Key Laboratory of Semiconductor Physics and Chip Technologies, Institute of Semiconductors, Chinese Academy of Sciences, Beijing 100083, China
- the Beijing Key Laboratory of Semiconductor Intelligent Sensing and Computing, Institute of Semiconductors, Chinese Academy of Sciences, Beijing 100083, China
- the Center of Materials Science and Optoelectronics Engineering, University of Chinese Academy of Sciences, Beijing 100049, China

Additional Information

EDICS

DCS Digital Circuits and Systems (and VLSI)

DCS Digital Circuits and Systems (and VLSI) / DCS280 Digital accelerators

DCS280 Digital accelerators

TCAS-II Subject Category

Digital Circuits and Systems and VLSI

Files for peer review

All files submitted by the author for peer review are listed below. Files that could not be converted to PDF are indicated; reviewers are able to access them online.

Name	Type of File	Size	Page
A heterogeneous hybrid CNN-Transformer accelerator with multi-level Strassen MatMul.pdf	Main Document - PDF	544.4 KB	Page 6

A heterogeneous hybrid CNN-Transformer accelerator with multi-level Strassen MatMul

Yuxiao Wen, Wenyao Zhao, Xuanzhe Xu, Xu Yang, *Member, IEEE*, Zhe Wang, *Member, IEEE*, Zhao Zhang, *Member, IEEE*, Jian Liu, *Member, IEEE*, Nanjian Wu, *Senior Member, IEEE*, Runjiang Dou, *Member, IEEE*, and Liyuan Liu, *Member, IEEE*

***Abstract*—Hybrid convolutional neural network (CNN)–Transformer networks combine the local feature learning efficiency of CNNs with the global representation capability of Transformers, making them highly suitable for edge vision tasks. However, the distinct computation patterns of self-attention and convolution, together with large-scale matrix multiplications (MatMul), still pose challenges for edge deployment. To overcome these limitations, we propose a heterogeneous hybrid CNN–Transformer accelerator with multi-level Strassen MatMul. First, a multi-core architecture comprising a systolic array, a vector processing element (VPE) array, and a reduction core is adopted, the three cores work in parallel, pipelining fixed-point quantized nonlinear operators together with convolution and MatMul. Second, the systolic array is reconfigured into four modes that respectively implement standard convolution, pointwise convolution, depthwise convolution, and MatMul, a multi-level Strassen MatMul is realized through 2×2 block-parallel computation and VPE array. The accelerator is validated on FPGA and deployed with multiple CNN–Transformer networks, achieving a 1.58× speedup over standard MatMul, 363.2 fps on EfficientViT-B1, and 277.1 fps on MobileViT-xs.**

***Index Terms*—Hybrid CNN–Transformer, accelerator, FPGA, MobileViT, EfficientViT.**

I. INTRODUCTION

THE vision transformers (ViTs) enables networks to capture global image information [1], [2]. However, compared with convolutional neural networks (CNNs), ViTs incur significantly higher computation and parameter counts, making it unsuitable for edge tasks. Consequently, several researchers have proposed hybrid CNN–Transformer (HybridViTs) networks that capture both local and global features, their lightweight design and competitive accuracy have made them the preferred choice for edge vision tasks [3], [4], [5].

For efficient low-latency inference, many works have developed FPGA or ASIC accelerators for standalone CNNs or Transformers [6], [7], [8]. However, HybridViTs involve both convolution and self-attention paradigms as well as nonlinear operators, whose distinct computation patterns make efficient deployment more challenging. Trio-ViT [9] implements an accelerator for a Softmax-free Efficient ViT. However, removing Softmax weakens the attention selectivity of the network and limits architectural flexibility. FLASH [10] processes CNNs and Transformers separately via a two-stage pipeline, assigning dedicated hardware to each stage; this wastes resources and lacks the flexibility to support other HybridViTs.

On the other hand, matrix multiplication (MatMul) dominates neural network computation, constraining inference speed and hardware efficiency on edge devices. The Strassen MatMul algorithm offers a classic approach to reducing the complexity of MatMul [11]. Its Winograd form further reduces the number of additions compared with the original algorithm, but at the cost of increased multiplier bit-width [12]. Given these trade-offs, the Strassen MatMul is more suitable for hardware implementation. However, implementing the Strassen MatMul on chip introduces challenges such as complex data scheduling and extra addition operations. The work in [13] employs seven parallel acceleration cores and pipelines the addition and multiplication operations to realize the Strassen MatMul. Nevertheless, this hardware designs implement the algorithm in a nested recursive fashion, which incurs unnecessary addition overhead and restricts the flexibility of the systolic array.

In this brief, we propose a multi-core heterogeneous accelerator based on multi-level Strassen MatMul, which efficiently executes various HybridViTs. The main contributions of this work are as follows:

1) First, we propose a parallel architecture that integrates a systolic array (SA), a vector processing element (VPE) array, and a reduction core (RC). The SA handles all MatMuls and convolutions, while the VPE array and RC together execute fixed-point Softmax, LayerNorm, and data rearrangement. This unified design handles all key HybridViTs operations without per-operator hardware, enabling efficient and flexible inference.

2) Second, we present a reconfigurable systolic array that supports four modes: standard convolution, pointwise

This work was supported in part by the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant XDB0460000, and in part by the National Natural Science Foundation of China under Grant 62334008, Grant U25A20489 and Grant 62404218, and in part by the Beijing Natural Science Foundation under Grant Z220005 and Grant L254053. (Corresponding author: Runjiang Dou).

The authors are with the State Key Laboratory of Semiconductor Physics and Chip Technologies, Institute of Semiconductors, Chinese Academy of Sciences, Beijing 100083, China, the Beijing Key Laboratory of Semiconductor Intelligent Sensing and Computing, Institute of Semiconductors, Chinese Academy of Sciences, Beijing 100083, China, and also with the Center of Materials Science and Optoelectronics Engineering, University of Chinese Academy of Sciences, Beijing 100049, China.

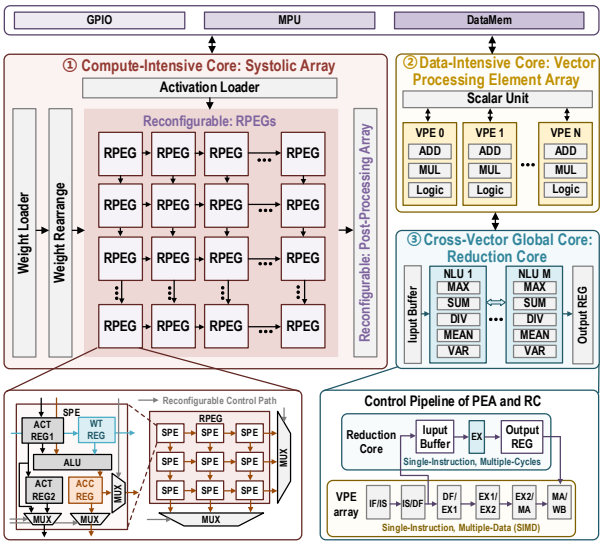


Fig. 1. The overall architecture of multi-core hardware.

convolution, depthwise convolution, and multi-level Strassen MatMul. A 2×2 block-parallel computation strategy works in parallel with the VPE array to address the high addition overhead and complex scheduling inherent to multi-level Strassen. Transposed MatMul is further enabled through weight rearrangement.

3) Finally, the accelerator is validated on FPGA, synthesized in 55 nm technology, and deployed with multiple HybridViTs, achieving 1.58× speedup over standard MatMul. Quantization of nonlinear operators results in only 0.69% accuracy loss, with final inference speeds of 363.2 FPS on EfficientViT-B1 and 277.1 FPS on MobileViT-xs.

II. Multi-Core Hardware Architecture

HybridViTs operators can be classified into three categories: (a) compute-intensive, namely MatMul and convolution; (b) data-intensive, such as post-processing and image rearrangement; (c) cross-vector global, including computing the maximum or summation of a vector group. Based on this classification, MatMul and convolution are handled by the compute-intensive core together with the data-intensive core, whereas nonlinear operators are executed by the data-intensive core together with the cross-vector global core. Hence, we adopt a three-core architecture. As shown in Fig. 1, the accelerator integrates a systolic array, a vector processing element array, and a reduction core, complemented by control and memory modules.

A. Vector Processing Element Array and Reduction Core

The VPE array comprises a scalar unit and 256 VPEs. The scalar unit controls the array, while the VPEs execute SIMD operations, including addition, multiplication, and logic operations. The reduction core consists of 32 nonlinear units (NLU), which can access cross-unit data and, through a pipeline, produce a cross-vector global value such as the maximum, summation, etc. The lower right of Fig. 1 shows the control pipeline between the VPE array and the reduction core. Reduction core instructions execute in a single-instruction,

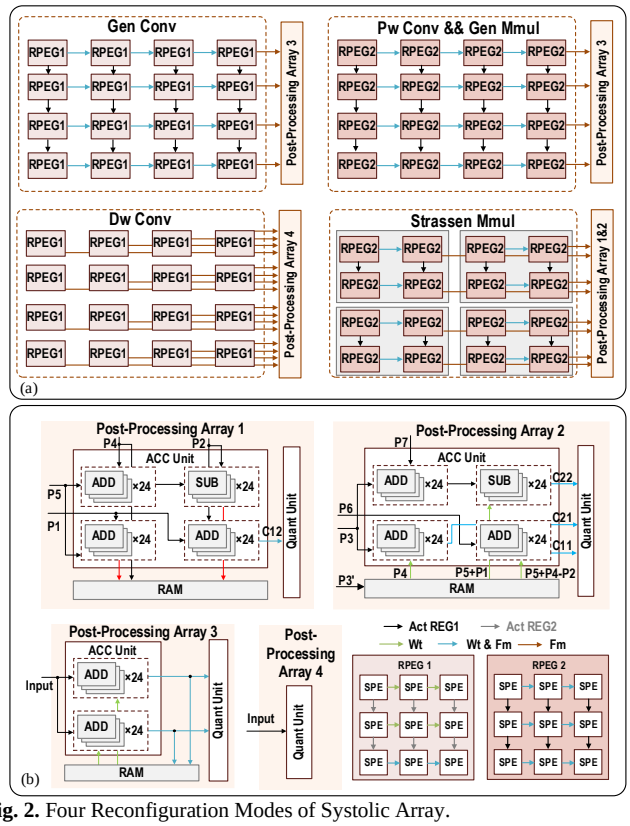


Fig. 2. Four Reconfiguration Modes of Systolic Array.

multiple-cycle manner and can run in parallel with the VPE array. The VPE array decodes the reduction core configuration, writes the configuration together with the values from VPE registers into input buffer. The reduction core then iteratively computes and writes results to output register, finally the VPE array retrieves these results.

B. Reconfigurable Systolic Array

The systolic array consists of a weight loader unit, a weight rearrange unit, an activation loader unit, 16×16 array of reconfigurable processing element groups (RPEGs), and a post-processing array. The weight rearrange unit enables automatic matrix transposition. The RPEGs can be set to four dataflow configurations supporting standard convolution, pointwise convolution, depthwise convolution [14], and one-level Strassen MatMul. Each RPEG contains a 3×3 array of systolic processing elements (SPEs), and each SPE includes two activation registers, one weight register, one accumulation register, and an 8-bit MAC ALU.

To efficiently support the compute-intensive operators, the systolic array is reconfigured as shown in Fig. 2. The post-processing array contains 72 addition units and 24 subtraction units, which are organized into four operating modes that handle the addition/subtraction operations in the Strassen MatMul or convolution. Fig. 2(a) depicts the four reconfigurable modes of the systolic array core, each tailored to the distinct characteristics of standard convolution, pointwise convolution, depthwise convolution, and the Strassen MatMul, thereby maximizing hardware utilization.

For standard convolution, the RPEG1 configuration is

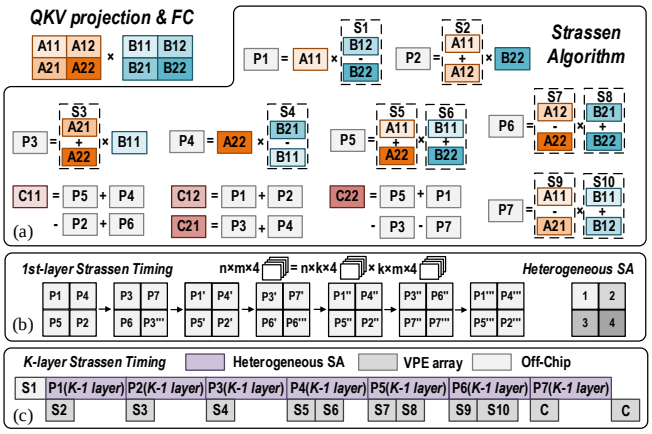


Fig. 3. Dataflow of the multi-level Strassen MatMul.

adopted, loading convolution kernel parameters into each RPEG and computing convolution in a weight-stationary, activation-stream manner, which processes 16 input channels and 16 output channels at a time. Depthwise convolution requires no accumulation across input channels; therefore, the result of each RPEG is directly output, enabling concurrent computation of 256 channels. Pointwise convolution resembles MatMul, so we use the RPEG2 configuration, computing convolution or MatMul for 48 input channels and 48 output channels per operation.

For the Strassen MatMul, the systolic array is reconfigured into a 2×2 grid of sub-arrays, and modes 1 and 2 of the post-processing array are adopted, enabling the additions, subtractions, and multiplications of the Strassen MatMul to be computed in a pipelined manner. The detailed computation flow is described in section III.

III. Multi-Level Strassen for MatMul

The computational flow of the Strassen MatMul is illustrated in Fig. 3(a). Standard MatMul requires 8 multiplications and 4 additions. The Strassen MatMul, in contrast, performs 7 multiplications and 18 additions/subtractions. Deeper recursion levels offer greater speedups but also introduce additional additions and more complex data scheduling. The work in [13] recursively implements one- and two-level Strassen MatMul in hardware; however, this recursive approach introduces excessive adders and restricts the flexibility of the systolic array. It can be observed that lower levels involve more additions relative to multiplications, while higher levels have the opposite. Therefore, we place the first-level additions in the post-processing array and the subsequent additions in the VPE array, preventing excessive adders and enabling flexibility. Using this strategy, the multi-level Strassen MatMul is implemented in the accelerator.

For the first-level Strassen MatMul: we aim to keep the first-level blocks small to support various matrix sizes. Based on the feature map sizes in our networks, a 24×24 block size is chosen. When the systolic array is reconfigured, configuring it as 1×7 blocks would waste SPEs. Observing that the MatMul in the algorithm is typically multi-channel, we group four channels into one computation. The timing of the first-level Strassen MatMul is shown in Fig. 3(b). the values P5+P4−P2, P1+P2, and P5+P1 are obtained; in mode 2, all C (combination)

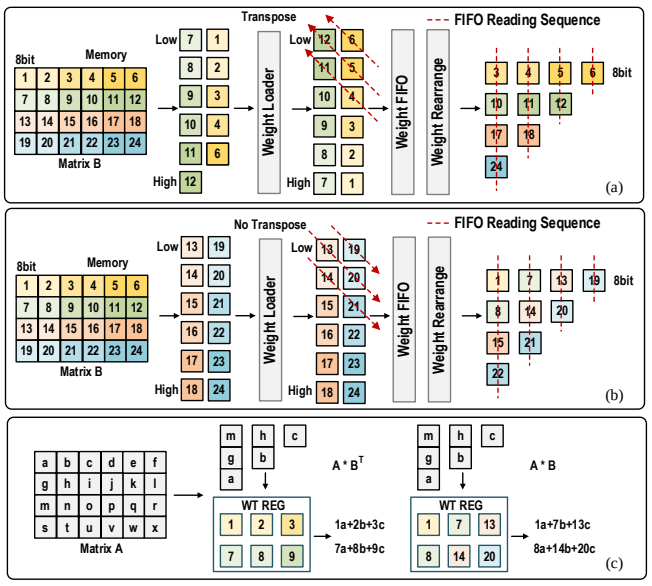


Fig. 4. Data loading scheme for transposed MatMul.

are produced and P (multiplication) for the next level is buffered. Seven such cycles complete one four-channel MatMul.

For k-level Strassen MatMul: the results from the k-1 level must undergo another iteration of S (addition/subtraction), P, and C. This process cannot reuse the adders from the first level. As shown in Fig. 3(c), we eliminate the extra cycles of additions through parallel computation between the VPE array and the SA. The P computations run on the systolic array using the k - 1 level Strassen approach, while the S and C computations are handled by VPE array.

Matrix Transposition: transformer contains transposed MatMul. We adopt a specialized data loading scheme that eliminates the need for explicit matrix transposition. Fig. 4(a) illustrates the parameter loading flow when transposition is required: the data are flipped through the weight loader and then fed into weight FIFO; finally, the weight rearrange unit reorders the output and feeds it to the systolic array. Fig. 4(b) shows the loading flow when no transposition is needed. Fig. 4(c) provides the systolic array computing process with and without transposition, demonstrating that this loading scheme can realize transposed MatMul through a simple data loading procedure.

IV. Parallel Execution of Nonlinear Operations with VPE Array and RC

A. Integer-Only Softmax

Softmax requires computing exponentials. Many researchers adopt lookup table (LUT) methods, which, although effective, consume additional hardware resources. We quantize the Softmax operation [15] and realize a fixed-point Softmax through parallel computation using the reduction core and the VPE array.

First, the maximum input value is computed and subtracted from all inputs, constraining the exponent to a smaller range and preventing overflow. Then, the exponential is computed via the method in Algorithm 1. When the exponent input is

Algorithm 1 Integer-only Exponential and Softmax

Input: 8bit: x_i, 16bit: S
Output: 8bit: Softmax(x_i)

function I-Exp(x, S)
  a, b, c ← 0.3585, 1.353, 0.344
  F ← (x × log2e × S) [31:16]
  I ← (x × log2e × S) [15:0]
  Exp (I) ≈ (I × ln2 / S + b / S) ^ 2 + c / (a × S ^ 2)
return (Exp (I) × 2^{-F}) [7:0]

function I- Softmax(x_i, S)
  ex_i ←I-Exp(x_i - max(x_i))
  $sumex_i$ ← sum(ex_i)
return ex_i / $sumex_i$

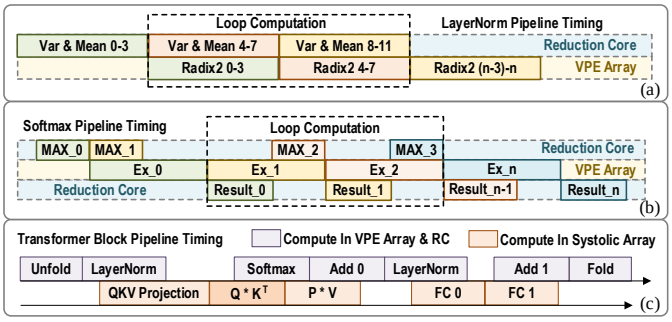


Fig. 5. Pipeline timing of the nonlinear operations and the Transformer block.

restricted to (−ln2, 0], the exponential can be approximated as:

$$Exp(I) = 0.3585 \times (I + 1.353)^2 + 0.344 \qquad (1)$$

Accordingly, the input is decomposed into the form $Exp(x) = Exp(I) \times 2^{-F}$, and the quantization parameter S is fused at the input stage, thereby completing the Softmax computation. All intermediate values are kept in 16-bit fixed-point format throughout the computation to minimize the quantization error.

B. Integer-Only LayerNorm

LayerNorm involves division, square root, and variance computation. For square root and division, many researchers adopt the CORDIC method [16], [17], but these two operations cannot be computed independently, resulting in considerable latency. Some have proposed methods that compute division and square root simultaneously using a radix-4 algorithm [18]; however, although this approach reduces the computational workload, the large number of conditional decisions is hardware-unfriendly. Therefore, the radix-2 square root calculation shown in Algorithm 2 is adopted.

First, the mean and variance are computed as described in Algorithm 2, then iteratively compute $x_i - mean(x_i)/\sqrt{var(x_i)}$ using the I-radix2-sqrt algorithm, and finally perform inverse quantization. This method employs 16-bit quantization, is more hardware-friendly, and allows reuse of the VPE array and the reduction core.

C. VPE array and Reduction Core Parallel Execution

All the above operators can be decomposed into data-intensive operations and cross-vector global operations. Parallel logic and addition operations are classified as data-intensive and executed in the VPE array, while summation, maximum, and other cross-vector global operations are handled by RC. Moreover, Softmax and LayerNorm can time-share the same hardware, effectively improving resource utilization.

Fig. 5(a) and (b) show the pipeline timing of Softmax and LayerNorm. These operators are executed in parallel by the VPE array and RC, and the VPE array to RC size ratio is set to 8 to balance their computation cycles, attaining high parallelism. Fig. 5(c) illustrates the pipeline timing of the entire Transformer block, where the three cores execute in a pipelined parallel manner, significantly boosting computation speed. Experimental results show that with fixed-point quantization, the overall accuracy degradation of the entire algorithm is about 0.69%.

Algorithm 2 Integer-only LayerNorm

Input: 8bit: x_i, 16bit: mul, sft
Output: 8bit: LayerNorm(x_i)

function I-radix2-sqrt(x0, z0)
  for (int i =0; i <n; i ++)
    sp, sz, sx ← z0 × 2^{-i}, z0 ×2^{-2i}, x0 ×2^{-i}
    if (z < 1) $c, c2$ ← 1, 1
    else if (z > 1) $c, c2$ ← -1, 1
    else $c, c2$ ← 0, 0
    z ← $z + c2 \times sz / 4 + c \times p \times 2^{-i}$
    p ← $p + c \times sp$ / 2; x ← $x + c \times sx$ / 2
return x

function I- LayerNorm(x_i, mul, sft)
  $radix2_i$ ← I-radix2-sqrt($x_i - mean(x_i), var(x_i)$)
return $radix2_i$ × mul × 2^{-sft}

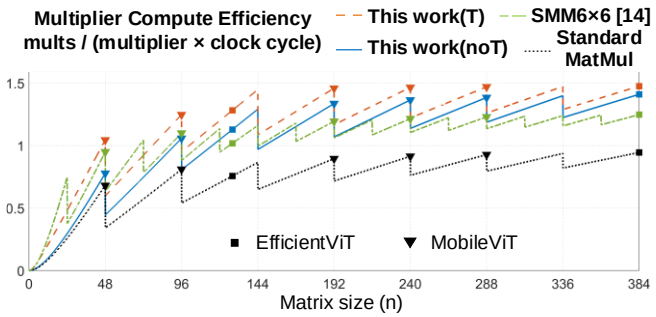


Fig. 6. Maximum achievable multiplier compute efficiency comparison across different architectures. Triangles denote typical input matrix sizes for MobileViT, and squares denote those for EfficientViT.

V. Implementation and Result

The multiplier compute efficiency (MCE) is defined as mults / (multiplier × clock cycle) and measures multiplier utilization, with clock cycles including matrix transposition overhead. Fig. 6 compares the maximum MCE across architectures. The orange curve shows the proposed design with transposed MatMul optimization, and the blue curve the version without. At typical EfficientViT and MobileViT input sizes, the proposed architecture achieves the highest utilization. It applies one-level Strassen MatMul for sizes 0–48, two-level for 48–96, and three-level beyond 96, delivering a 1.58× speedup over standard MatMul.

Table I presents the quantized algorithm and hardware inference results. The evaluation covers MobileNetv2, MobileViT-xs, and EfficientViT-B1 on ImageNet-1k [19], using 8-bit quantization for convolutions and MatMuls, 16-bit for nonlinear operations. The results show that MobileNetv2 achieves the highest frame rate of 498.2 FPS, MobileViT-xs requires the smallest storage of 2.32 MB, and EfficientViT-B1 delivers the best accuracy of 77.95%. Supporting multiple HybridViTs allows edge hardware to flexibly choose a network based on storage, accuracy, and real-time demands.

TABLE I
QUANTIZED ALGORITHM AND HARDWARE INFERENCE RESULTS

Model	Precision	Top-1	Params	Frame Rate
MobileNetv2	8 (A, W) 16(Nonlinear)	71.05 (-0.44)	3.47 M	**498.2**
MobileViT-xs		72.77 (-0.54)	**2.32 M**	277.1
EfficientViT-B1		**77.95** (-1.10)	9.97 M	363.2

TABLE II
COMPARISON WITH RELATED AND COMPARABLE WORKS

	TCAS-I'23[7]	**TVLSI '26[8]**	**TCAS-I'25[9]**	**TNNLS '26[10]**	***This work***	
Technology	16 nm (FPGA)	16 nm (FPGA)	16 nm (FPGA)	16 nm (FPGA)	16 nm (FPGA) 55 nm (ASIC)	
Model	ViT-Tiny	ViT-Tiny	EFF ViT1[1]	MBViT	MB ViT	EFF ViT2[2]
Model Type	Trans.	Trans.	CNN + Trans.	CNN + Trans.	CNN + Trans.	
Frequency	300 M	300 M	200 M	100 M	300 M	
BRAM	-	383.0	162.0	1141.0	404.0	
LUT	144.0 K	-	130.6 K	350.9 K	256.0 K	
DSP	1268	1754	1024	1180	778	
Throughput (GOPS)	616.1	455.3	796	-	609.6	508.5
Frame Rate (FPS)	245.3	182.1	190	99.8	277.1	**363.2**
GOPS / DSP	0.486	0.260	0.777	-	**0.783**	0.654

[1]EFFViT1 adopts the network from [3]. [2]EFFViT2 adopts the network from [5].

We deploy the hardware on VCU118 FPGA board and synthesize it in 55 nm technology at 300 MHz. Table II presents the comparison with related and comparable works. Compared with [7], [8], proposed accelerator supports HybridViTs. In contrast to Trio-ViT [9], proposed accelerator executes Softmax in parallel and accommodates a wider range of HybridViTs. Compared with FLASH [10], proposed accelerator employs reconfigurable hardware to compute nonlinear operators, MatMuls, and convolutions, achieving higher hardware utilization. Operating at 300 MHz on the FPGA, EfficientViT-B1 achieves 609.6 GOPS and the highest frame rate of 363.2 FPS, while MobileViT-xs achieves 508.5 GOPS and the highest DSP efficiency of 0.783 GOPS/DSP. Proposed accelerator meets the real-time and resource-efficiency requirements of edge.

VI. CONCLUSION

This brief proposes a multi-core heterogeneous accelerator for HybridViTs, combining systolic array, VPE array, and RC to execute MatMuls, convolutions, and nonlinear operators. For MatMul, a 2×2 block-parallel systolic array cooperates with VPE array to implement a multi-level Strassen algorithm, achieving 1.58× speedup over standard MatMul. The accelerator supports multiple HybridViTs, enabling edge hardware to adapt to a broader range of application scenarios.

REFERENCES

[1] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.
[2] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10012–10022.
[3] H. Cai, C. Gan, and S. Han, "EfficientViT: Enhanced linear attention for high-resolution low-computation visual recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Jun. 2023, pp. 17302–17313.
[4] S. Mehta and M. Rastegari, "MobileViT: Light-weight, General-purpose, and Mobile-friendly Vision Transformer," 2022, *arXiv:2110.02178*.
[5] H. Cai, J. Li, M. Hu, C. Gan, and S. Han, "EfficientViT: Multi-Scale Linear Attention for High-Resolution Dense Prediction," 2024, *arXiv:2205.14756*.
[6] F. Wang et al., "A Conv-GEMM reconfigurable accelerator with WS-RS dataflow for high throughput processing," *Electronics Letters*, vol. 60, no. 3, p. e13125, Feb. 2024.
[7] M. Huang, J. Luo, C. Ding, Z. Wei, S. Huang, and H. Yu, "An Integer-Only and Group-Vector Systolic Accelerator for Efficiently Mapping Vision Transformer on Edge," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 70, no. 12, pp. 5289–5301, Dec. 2023.
[8] J. Dong et al., "MoE-Sched: Enabling Efficient FPGA Deployment of Mixture-of-Experts Vision Transformers via Coordinated Scheduling," *IEEE Trans. VLSI Syst.*, vol. 34, no. 1, pp. 104–117, Jan. 2026.
[9] H. Shi, H. Shao, W. Mao, and Z. Wang, "Trio-ViT: Post-Training Quantization and Acceleration for Softmax-Free Efficient Vision Transformer," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 72, no. 3, pp. 1296–1307, Mar. 2025.
[10] N. J. Kim et al., "FLASH: Energy-Efficient FPGA Acceleration via Linear Approximation and Streamlined Two-Stage Pipeline Architectures for Quantized CNN–Transformer Hybrid Networks," *IEEE Trans. Neural Netw. Learning Syst.*, pp. 1–14, 2026.
[11] V. Strassen, "Gaussian elimination is not optimal," Numerische Math., vol. 13, no. 4, pp. 354–356, Aug. 1969.
[12] S. Winograd, "On multiplication of 2× 2 matrices," Linear Algebra Appl., vol. 4, no. 4, pp. 381–388, 1971.
[13] T. E. Pogue and N. Nicolici, "Strassen Multisystolic Array Hardware Architectures," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 33, no. 5, pp. 1323–1333, May 2025.
[14] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 4510–4520.
[15] S. Kim, A. Gholami, Z. Yao, M. W. Mahoney, and K. Keutzer, "I-BERT: Integer-only BERT Quantization," 2021, *arXiv:2101.01321*.
[16] Z. Li and Q. Gu, "I-ViT: Integer-only Quantization for Efficient Vision Transformer Inference," 2023, *arXiv:2207.01405*.
[17] S. Moon, M. Li, G. K. Chen, P. C. Knag, R. K. Krishnamurthy, and M. Seok, "T-REX: A 68-to-567μs/Token 0.41-to-3.95μJ/Token Transformer Accelerator with Reduced External Memory Access and Enhanced Hardware Utilization in 16nm FinFET," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, 2025, pp. 406–408.
[18] D. Tian et al., "A Low-Latency Power Series Approximate Computing and Architecture for Co-Calculation of Division and Square Root," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 71, no. 6, pp. 2723–2734, Jun. 2024.
[19] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.