FPGA-Based Vit Inference Accelerator Optimization

Haoyu Lu *

Department of Electrical Engineering and Electronics, Liverpool University, Liverpool, United Kingdom

* Corresponding Author Email: sghlu10@liverpool.ac.uk

Abstract. In the AI era, the emergence of the transformer model contributes to a significant shift in the natural language processing field. Its derivative, the Vision Transformer (ViT), adapts these principles for image recognition and demonstrates superior performance over traditional Convolutional Neural Networks (CNNs). Despite its excellent performance, deploying these models on edge devices is impeded by their extensive computational demands and large memory requirements, which poses challenges for the limited resources and real-time processing needs at the edge. Hence, it is necessary to develop a new hardware accelerator to optimize ViT architecture. This paper reviews the development of Field-Programmable Gate Array (FPGA)-based ViT inference accelerators, focusing on their architectures and applications in solving the dilemmas of ViT model deployment. Additionally, it explores the optimization approaches on both algorithm and hardware and traces the advancements in deploying AI models at the edge using FPGAs. It summarizes the current trends in research on FPGA-based ViT accelerators and provides insights into future directions for innovations in hardware-accelerated AI. Generally, by tracing the related works for FPGA-based ViT inference accelerator optimization, this article presents a useful snapshot of current research in ViT hardware accelerators and contributes to clarifying future research directions in this area.

Keywords: FPGA; ViT; Hardware Accelerator.

1. Introduction

The first proposal of transformer architecture brings a groundbreaking shift in dealing with sequential data, primarily in the field of natural language processing [1]. This innovation method was further developed to computer vision with the Vision Transformer (ViT), which demonstrated that self-attention mechanisms have excellent performance in managing image-related tasks [2]. Compared to the traditional CNN model, the ViT model achieves higher accuracy under a comparable number of parameters and maintains competitive computational costs which contributes to highly effective model performance. However, due to the attention mechanism in the transformer encoder and the numerous matrix multiplications required in the feedforward neural network, the model showcases high computational complexity. Additionally, the substantial parameter of the transformer network illustrates the higher storage demands of the transformer model [3, 4]. These unique characteristics of transformer networks present challenges for their deployment and application on edge devices, especially in resource-limited edge computing environments such as autonomous driving and drone navigation. Several techniques on both algorithms and hardware like model compression method, FPGA and ASIC hardware accelerators for deep neural networks have achieved significant success in this field [5]. Nevertheless, the sequence data and image data required in NLP and computer vision have obvious differences, primarily in data structure and computational data flow. Image data presents more segmentation dimension and redundant information than sequence data. Furthermore, ViT only employs the Encoder module, thereby regularizing the distribution of the shortcut mechanism and mitigating the impact of path dependency [6]. Consequently, because of the unique data stream of ViT, designing a specialized efficient hardware architecture for it has emerged as a critical research topic.

Common hardware platforms include CPUs, GPUs, ASICs and FPGAs. CPUs are well-known as the general-purpose processor, but present adverse performances in both efficiency and energy consumption in deep learning computation. Additionally, although GPU is widely implemented in

neural network computation, its high consumption and throughput natures are prohibitive in the deployment of edge devices which limits the application in embedded scenarios. In terms of ASICs, the main metric is high energy efficiency, however, the deficiency in flexibility makes it inadequate to meet the rapid iteration of the models. Consequently, FPGA-based accelerator garners considerable emphasis for its balance of performance, consumption and adaptability. The choice of FPGA as the hardware platform for accelerator ViT models is mainly driven by its inherent architectural advantages which are highly compatible with the attention mechanism central to the transformer network. Besides, the high parallelization and reconfigurable logic block characteristics of FPGA allow the improvement of intensive matrix multiplications and attention calculations required by ViT inference.

2. FPGA-based ViT Inference accelerator

2.1. Basic theory of FPGA

Field-Programmable Gate Arrays (FPGAs) are a type of configurable integrated circuit that can be programmed or reprogrammed after manufacturing. As a border category known as Programmable Logic Devices (PLDs), FPGAs consist of an array of programmable logic blocks and interconnects which allows them to be configured to conduct diverse digital functions. It is generally employed in applications that require flexibility, speed and parallel processing capabilities, such as telecommunications, automotive and industry fields. In FPGAs, the logic blocks can be programmed to carry out complex combinational functions or work as basic logic gates such as AND and XOR. In addition, these elements involve simple flip-flops or more complete memory block structures. This re-programmability allows FPGAs to perform flexible, reconfigurable computations similar to those conducted by computer software.

FPGA is also deemed as an important role in the development of embedded systems because it allows parallel initiation of hardware and system software development, which is beneficial to the early-stage systematic performance simulation, and various system tests and design iterations before finalizing the system architecture.

Moreover, with the development of AI, a notable application of FPGA is hardware acceleration, where it enhances specific algorithmic segments and distributes computational loads between themselves and general processors. Since 2014, Bing has utilized FPGA-based accelerators to improve the response speed of its search algorithms, significantly reducing latency and facilitating query handling [7]. By 2018, the use of FPGA had been further expanded, particularly with Microsoft's "Project Catapult". This project focuses on employing FPGAs to accelerate artificial neural networks which significantly improve performance in machine learning applications [8, 9]. It is obvious that the implementation of FPGA in the AI field has become a tendency and proved to be successful. Therefore, research on FPGA-based accelerators is a promising direction, not only due to the unique characteristics of FPGAs—such as their reconfigurability and efficiency in parallel processing—but also because these features align exceptionally well with the computational demands of ViT models.

2.2. FPGA-based ViT inference accelerator architecture

Two predominant architectural designs are employed in the realm of FPGA-based ViT inference accelerator, namely the Neural Processing Engine (NPE) and LTrans-OPU architecture. As the overlay architecture illustrated in Figure 1, the NPE structure integrates the memory management and processing unit in a pipeline, enabling efficient latency handling and maximizing unit concurrency. This working model of NPE seems tailored approach for attention-centric computations of ViT and satisfies the complex data flow and its intensive computational needs [10]. The NPE architecture is shown in the Figure 1.

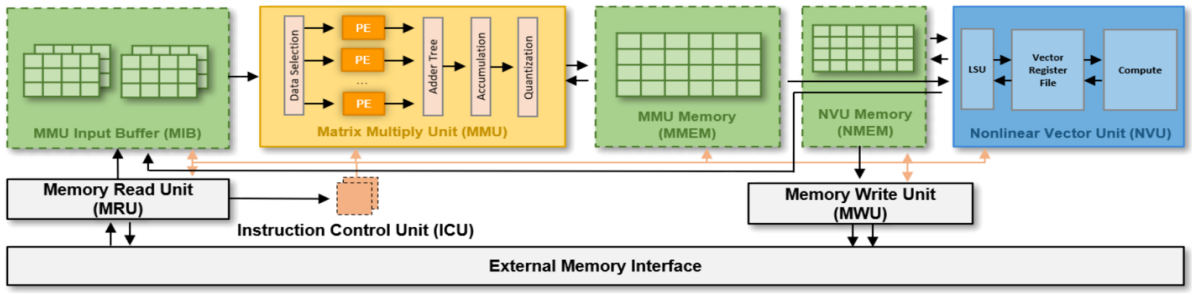

Fig 1. NPE architecture [10]

Conversely, the LTrans-OPU framework separates the operation into four critical modules, managing the tasks from memory access to post-processing. Additionally, the configurable computational units, which can adapt to multiple transformer model dimensions, optimize the inherent complex non-linear functions, including SoftMax and Layernorm [11]. The LTrans-OPU overlay architecture is shown in the Figure 2.

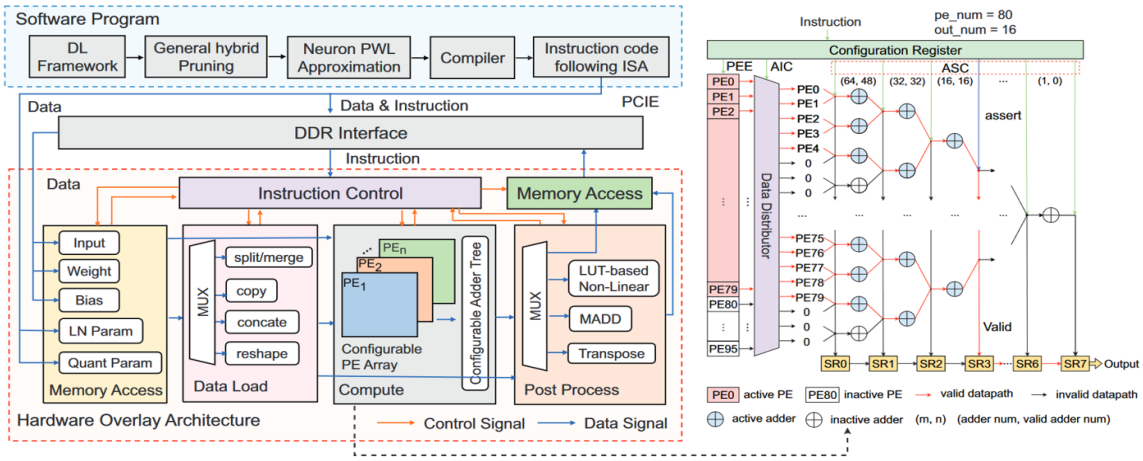

Fig 2. LTrans-OPU overlay architecture [11]

These architectures emphasize the differences between the stream pattern and the overlap pattern. The stream pattern employs the optimized hardware unit to specific algorithm tasks which allows high efficiency but scarifies the flexibility. Instead, the overlay pattern provides a large and reusable processing engine. Although it can flexibly adapt to the shift of models, this pattern may be unable to reach the peak efficiency of tailored hardware.

Based on the previous discussion, it is obvious that the overlay pattern is specialized for flexibility whereas the stream pattern is dedicated to acceleration through optimizing the computational units for a certain model. Considering the multiple functional modules of ViT, the stream pattern is regarded as the most suitable architecture for building the FPGA-based accelerator which aligns with the workflow of ViT and ensures a streamlined execution of the attention-centric processes.

3. Algorithmic optimization

Algorithmic optimization is vital to deploying the transformer-based neural network, especially under the consideration of the application on diverse hardware devices, such as smartphones and IoT devices. Quantization is regarded as one of the most useful approaches to compressing neural networks which switch floating-point numbers to lower-bandwidth integers, thereby reducing the memory consumption and computational cost. Additionally, quantization is dedicated to minimizing the bit-width required to represent network weights, and effectively reducing the memory storage and computational overhead for ViT.

Two types of methods are generally employed for quantization, namely Quantization-Aware Training (QAT) and Post-Training Quantization (PTQ). Despite minimal accuracy degradation

produced by quantized networks, the QAT requires extensive optimization time, training dataset and hyperparameter turning, which showcase that QAT is impractical when the dataset is unavailable or needs quick deployment. Conversely, PTQ performs advantages on rapid deployment and quantization by employing unlabeled calibration images after training in the quantization processing. Furthermore, there is another noteworthy difference between QAT and PTQ: a pre-trained model is quantized and then fine-tuned with training data to adjust parameters and recover from any accuracy loss in QAT. Whereas in PTQ, calibration data is used to calibrate the pre-trained model, determining clipping ranges and scale factors, followed by model quantization based on calibration results. Notably, the calibration process generally occurs in parallel with the fine-tuning process in QAT [12, 13].

The typical approaches to deploying quantized neural network models are known as simulated quantization and pure integer quantization. The quantized parameters in the simulated quantization are stored at low precision whereas the operations such as convolution and matrix multiplication are conducted by using floating-point arithmetic. As the flowchart in the middle of Figure 3 illustrates, this method requires dequantizing the quantized parameters before floating-point operations, which indicates that it is difficult to fully use the benefits of rapid and efficient low-precision logic. Nevertheless, as shown on the right side of Figure 3, the pure integer quantization allows the entire inference to use integer arithmetic and conducts all the operations by using low-precision integer arithmetic. These quantization strategies not only render ViT models more compatible with FPGA environments but also improve the potential for efficient inference on resource-limited edge devices. The Comparison between full-precision inference, inference with simulated quantization, and inference with integer-only quantization is shown in Figure 3.

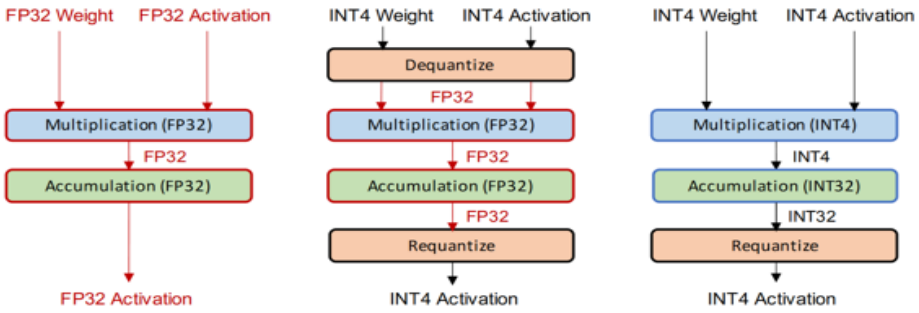

Fig 3. Comparison between full-precision inference (Left), inference with simulated quantization (Middle), and inference with integer-only quantization (Right) [13]

In general, algorithmic optimization aims to achieve high performance and resource-efficient utilization. Through these optimization techniques, the models can maintain their accuracy even under constrained bit widths, thereby promoting the application and development of ViT hardware accelerators.

4. Hardware optimization

4.1. High-end FPGA Device implementation of ViT hardware accelerator

As previously mentioned, the main differences between image data and sequential data required in CV and NLP fields are seldom considered in existing accelerator designs. To design a transformer model that can efficiently conduct ViT inference, high-end FPGA devices emerge as a promising solution. The work introduces ViA, a novel ViT accelerator architecture, that aims at executing transformer applications efficiently while mitigating the cost implications associated with data locality and path dependence issues inherent in the existing related accelerator designs [6].

The hardware architecture and data flow pattern of ViA accelerator are designed to align with the computational characteristics of Vision Transformers, particularly focusing on enhancing data throughput and minimizing latency for each stage of the ViT computation. In ViA architecture, the

Norm Self-attention Module (NSA) is compromised with several small units, including the Feature Value Generator (FVG) and Matrix Multiply Unit (MMU), which can conduct layer normalization and multi-head self-attention at the same time. Additionally, the design of the Norm MLP Module (NMP) allows it to cooperate with the NSA and ensure the efficiency of the feed-forward network's data processing by parallel operations [14].

Furthermore, the data flow in ViA architecture is also tailored to align with the ViT computational characteristics. Data is generated to streams that flow through the processing elements (PEs) in a pipelined manner. This framework is critical to the self-attention mechanism because of its continuous processing of query, key, and value matrices. It is also noteworthy that the PEs are structured to run alternatively: as one set of PEs is actively processing one stream of data, the other set prepares to engage the next data stream. This working mode ensures FPGA always has active computation and guarantees a highly efficient data processing cycle.

The ViA accelerator demonstrates excellent performance compared with previous FPGA works. Compared to TEC, there is a significant improvement in data precision and model compression which exerts positive impacts on both performance and computational efficiency as well [15]. Compared with other designs, ViA achieves a 4 to 10-fold increase in throughput, and computational density and energy efficiency are improved by 7 to 15 and 2.6 to 10.9, respectively [6].

4.2. Edge (low-end) FPGA Device implementation of ViT hardware accelerator

To address the challenges of ViT model deployment on edge devices, a novel accelerator architecture named ViTA is designed. ViTa is primarily aimed at resource-constrained FPGA devices (Zynq ZC7020 MPSoC, 53200LUT, 220 DSP 630KB BRAM), and supports a variety of popular ViT models.

The hardware architecture of ViTA minimizes unnecessary interim result storage by leveraging coarse-grained head-level pipelining and inter-layer optimization for multi-layer perceptron (MLP), significantly reducing redundant off-chip memory access. Also, there is a column-based double-buffer mechanism designed to read the weights, and a calculation module that can process data in a row-granular pipeline. This accelerator is based on a configurable processing element array and uses a corresponding scheduling scheme to maximize the efficiency of using the limited resources available on the target edge FPGA devices. By employing these methods, ViTA achieves nearly 90% hardware utilization efficiency and manages to operate on a low power budget of 0.88W at 150MHz clock frequency, providing reasonable frame rates suitable for edge applications [15].

5. Conclusion

The future research direction of the FPGA-based ViT hardware accelerator can be divided into two sections, namely algorithmic research, and hardware research. In the realm of algorithms, future research may focus on diminishing the dependency on extensive bandwidth and memory capacity. This objective may be attained through the implementation of low-bit quantization strategies, such as One-Bit LLM. Then it is required to be coupled with algorithmic pruning and employ the mixed expert modules to handle the multi-tasks. At the hardware level, the primary goal remains the enhancement of computational energy efficiency. This process includes not only efficient execution and approximate calculation of nonlinear functions, such as GELU and Softmax, but also demands the improvement of attention mechanisms and the efficient operation of linear layers.

In summary, the hardware design of accelerators is in service of algorithms. Developing hardware-friendly algorithms with a co-design approach of software and hardware. On this basis, designing efficient hardware computational architectures will become the cornerstone of driving the development of hardware accelerators.

References

[1] Ashish V. Attention is all you need. Advances in neural information processing systems, 2017, 30: I..

[2] Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020.

[3] Liu Y, Zhang Y, Wang Y, et al. A survey of visual transformers. IEEE Transactions on Neural Networks and Learning Systems, 2023.

[4] Liu Z, Lin Y, Cao Y, et al. Swin transformer: Hierarchical vision transformer using shifted windows. Proceedings of the IEEE/CVF international conference on computer vision. 2021: 10012-10022.

[5] Shawahna A, Sait S M, El-Maleh A. FPGA-based accelerators of deep learning networks for learning and classification: A review. ieee Access, 2018, 7: 7823-7859..

[6] Wang T, Gong L, Wang C, et al. Via: A novel vision-transformer accelerator based on fpga. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2022, 41(11): 4088-4099.

[7] Frank B H. Microsoft's Brainwave makes Bing's AI over 10 times faster. 2018. https://venturebeat.com/ai/microsofts-brainwave-makes-bings-ai-over-10-times-faster/.

[8] Chung E, Fowers J, Ovtcharov K, et al. Serving dnns in real time at datacenter scale with project brainwave. iEEE Micro, 2018, 38(2): 8-20.

[9] Microsoft Azure. Accelerating AI on the intelligent edge: Microsoft and Qualcomm create vision AI developer kit. 2018. https://azure.microsoft.com/en-us/blog/accelerating-ai-on-the-intelligent-edge-microsoft-and-qualcomm-create-vision-ai-developer-kit/.

[10] Khan H, Khan A, Khan Z, et al. NPE: An FPGA-based overlay processor for natural language processing. arXiv preprint arXiv:2104.06535, 2021.

[11] Bai Y, Zhou H, Zhao K, et al. LTrans-OPU: A Low-Latency FPGA-Based Overlay Processor for Transformer Networks. 2023 33rd International Conference on Field-Programmable Logic and Applications (FPL). IEEE, 2023: 283-287.

[12] Yu Y, Wu C, Zhao T, et al. OPU: An FPGA-based overlay processor for convolutional neural networks. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2019, 28(1): 35-47.

[13] Yuan Z, Xue C, Chen Y, et al. Ptq4vit: Post-training quantization for vision transformers with twin uniform quantization. European conference on computer vision. Cham: Springer Nature Switzerland, 2022: 191-207.

[14] Zhang X, Wu Y, Zhou P, et al. Algorithm-hardware co-design of attention mechanism on fpga devices. ACM Transactions on Embedded Computing Systems (TECS), 2021, 20(5s): 1-24.

[15] Nag S, Datta G, Kundu S, et al. ViTA: A vision transformer inference accelerator for edge applications. 2023 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE, 2023: 1-5.