LLM on FPGA: Squeezing Language Models by Quantization and Multi-Query Attention and Its Efficient Hardware Architecture

Seoyoon Chae
Dept. of Electronic and Electrical Engineering
Sungkyunkwan University, Suwon, Korea
sychaeee@skku.edu

Taewook Kang
Dept. of Semiconductor Convergence Engineering
Sungkyunkwan University, Suwon, Korea
twkang@skku.edu

***Abstract*—We present an on-chip implementation of a compressed Transformer-based language model on a Xilinx Artix-7 FPGA. Our contributions include: (1) combining ultra-low-precision quantization (4 bits) and multi-query attention (MQA) to compress the KV cache by $8\times$, enabling sequence lengths up to 256 tokens; (2) a streaming hardware architecture in Verilog that implements pre-layernorm, attention, and feed-forward sublayers using block RAM (BRAM) and DSPs; and (3) post-synthesis results demonstrating real-time throughput (4.4 K tokens/s) with BRAM and DSP utilizations of 31.9% and 85%, respectively. The prototype supports generative inference entirely on-chip, paving the way for privacy-preserving, edge-scale LLMs. Code and scripts are available at `https://github.com/chae-sy/squeezing_lm`**

***Index Terms*—Large Language model, quantization, multi-query attention, Transformer accelerator, FPGA, on-device inference, hardware–software co-design**

I. Introduction

Transformer-based large language models (LLMs) deliver state-of-the-art results but impose heavy compute, memory, and energy demands [1]. Autoregressive generation caches intermediate key–value (KV) tensors to reduce attention complexity from $O(S^2d)$ to $O(Sd)$ per token. However, storing even 8-bit KV on-chip often caps the sequence length. For an Artix-7 device with approximately $608\,\mathrm{KiB}$ on-chip block random access memory (BRAM), the on-chip sequence capacity without external dynamic random access memory (DRAM) is roughly

$$\frac{\text{total on-chip BRAM}}{\text{number of layers} \times \text{KV cache depth} \times \text{hidden dimension}} = \frac{608\,\mathrm{KiB}}{12 \times 2 \times 768\,\mathrm{B}} \approx 33 \text{ tokens.} \tag{1}$$

To address these constraints, we combine sub-8-bit quantization with multi-query attention (MQA) to compress the KV cache by over $8\times$, enabling entirely on-chip inference for $S \leq 256$. Our key contributions are:

- **Ultra-low-bit KV cache:** Applying 4-bit quantization through SmoothQuant combined with LoRA(Low Rank Adaptation)-based Quantization Aware Training (QAT) significantly reduces cache size, with less than a 0.02 accuracy drop and even improved perplexity (PPL) at the 4-bit level.
- **Multi-query attention:** Sharing K/V across heads (MQA) slashes cache size and bandwidth while preserving quality after QAT.
- **FPGA co-design:** A streaming Verilog accelerator (pre-layernorm, attention, feed-forward networks (FFN)) achieving 4.4 K tokens/s at $250\,\mathrm{MHz}$ with 31.9% BRAM and 85% digital signal processor (DSP) utilization.

II. Related Work

[2] integrates multi-head attention (MHA) and FFN on systolic arrays for throughput. [3] presents a 5 nm ASIC combining per-vector-scaled 4-bit quantization. [4] proposes an FPGA framework with structural pruning. None of these simultaneously exploit KV caching with sub-8-bit quantization for long-form, generative tasks under tight on-chip memory budgets typical of cost-sensitive FPGAs.

III. Model Compression Methodology

A. SmoothQuant for Range Equalization

We applied SmoothQuant [5] to mitigate outliers prior to quantization. For W8A8, W6A8, W4A8, and W6A6 settings, the PPL remained around 60, and the accuracy stayed close to 0.6. However, when applying W4A4, the PPL spiked to 116.1 and accuracy dropped to 0.49. This indicates that SmoothQuant becomes unstable when applied to the W4A4 configuration.

B. LoRA-Based Quantization-Aware Training

We inserted rank-8 LoRA adapters [6] into the query, key, and value (Q/K/V) projections and fine-tuned the model on WikiText-2 for 10, 50, and 100 steps, with the backbone weights kept frozen. We observed a PPL of 45.5 after 100 training steps. When combined with SmoothQuant, the PPL further decreased to 22.2 and 55.6 for 100 and 10 training steps, respectively.

C. Multi-Query Attention

Standard MHA computes separate keys and values for each head, incurring high memory costs. MQA [7] reduces cache size by sharing keys and values across heads, while group-query attention (GQA) [8] strikes a balance by sharing them within groups. In our experiments, the group-3 MQA configuration achieved substantial memory savings but led to a 2.6-point perplexity increase compared to MHA.

Figure 1 summarizes accuracy and PPL results across different quantization schemes and MQA group configurations. When combining the W4A4 LoRA with SmoothQuant (10 training steps) and group-3 MQA, the PPL was 55.6 and accuracy reached 0.6, which is close to the baseline. Furthermore, increasing the training steps to 100 reduced the PPL further to 22.6, with accuracy remaining at 0.58.

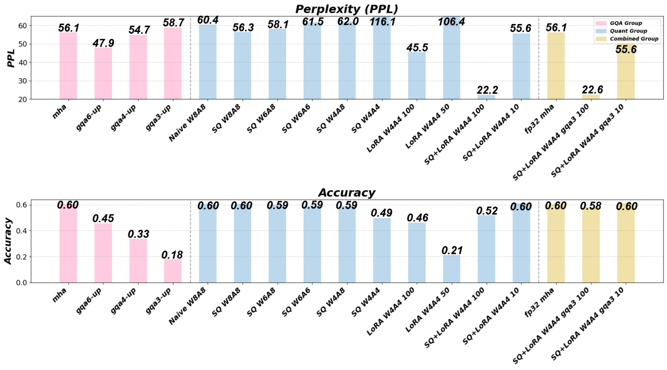

Fig. 1. Perplexity and accuracy across quantization and attention settings.

IV. Hardware Architecture

Fig. 2 depicts the top-level architecture with pre-layernorm, attention, and FFN units. The layernorm computes mean/variance over the channel dimension and applies a look-up table (LUT)-based reciprocal square root to approximate $1/\sqrt{\sigma^2+\epsilon}$, followed by 4-bit affine parameters (γ, β).

Fig. 3 details the attention and FFN datapaths. The attention engine performs a 1×1 convolution to form concatenated Q/K/V, stores compressed 4-bit K/V in shared BRAM, computes Q–K dot products on a processing elements (PE) array, applies a masked softmax via exponential/reciprocal LUTs, and forms the context vector via a weighted sum over V. The FFN expands $C \rightarrow 4C$, applies LUT-based GELU, and contracts $4C \rightarrow C$, each stage implemented as streaming mat-vec using DSPs and BRAM.

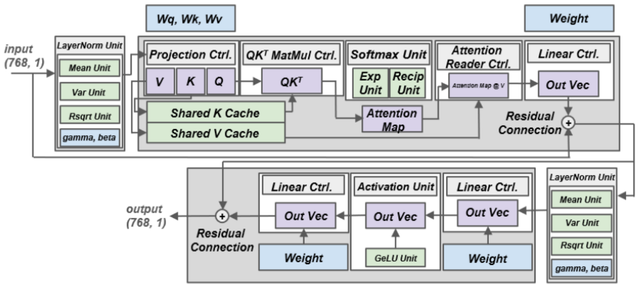

Fig. 2. Top-level FPGA accelerator dataflow.

TABLE I
Resource utilization of the proposed accelerator.

Resource	Used	Available	Utilization (%)
DSP slices	204	240	85.0
BRAM	193.5 KB	608 KB	31.9
LUTs	10,000	63,400	15.7

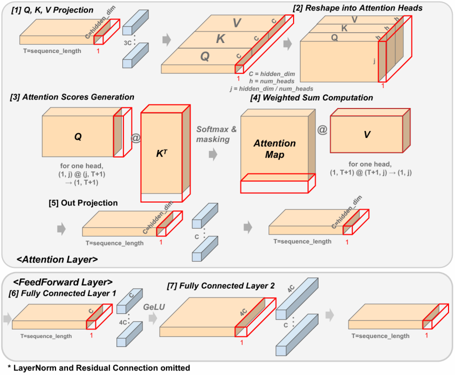

Fig. 3. Computation flow of the Transformer block.

V. Evaluation

We synthesize the design on an Artix-7 100T (CSG324) using Vivado 2018.3. Operating at 250 MHz, the pipeline sustains one token/cycle after fill, achieving 4.4 K tokens/s. On-chip sequence length support extends to 256 tokens. Resource utilization is summarized in Table I.

VI. Conclusion and Future Work

By co-designing 4-bit quantization, MQA, and a streaming FPGA architecture, we demonstrate fully on-chip LLM inference for 256-token sequences on Artix-7. Future work includes scaling to larger models, mixed-precision strategies, and energy-efficiency optimizations.

References

[1] A. Vaswani *et al.*, “Attention Is All You Need,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 5998–6008.

[2] S. Lu, M. Wang, S. Liang, J. Lin, and Z. Wang, “Hardware Accelerator for Multi-Head Attention and Position-Wise Feed-Forward in the Transformer,” in *Proc. IEEE Int. System-on-Chip Conf. (SOCC)*, Las Vegas, NV, USA, 2020, pp. 84–89.

[3] B. Keller *et al.*, “A 95.6-TOPS/W Deep Learning Inference Accelerator With Per-Vector Scaled 4-bit Quantization in 5 nm,” *IEEE J. Solid-State Circuits*, vol. 58, no. 4, pp. 1129–1141, Apr. 2023.

[4] X. Zhang, Y. Li, Y. Chen, and C. Wu, “Algorithm-Hardware Co-Design of Attention Mechanism on FPGA Devices,” *ACM Trans. Embed. Comput. Syst.*, vol. 20, no. 5s, pp. 1–24, Sep. 2021.

[5] G. Xiao *et al.*, “SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models,” in *Proc. Int. Conf. on Machine Learning (ICML)*, 2023, pp. 38087–38099.

[6] E. J. Hu *et al.*, “LoRA: Low-Rank Adaptation of Large Language Models,” in *Proc. Int. Conf. on Learning Representations (ICLR)*, 2022.

[7] N. Shazeer, “Fast Transformer Decoding: One Write-Head is All You Need,” *arXiv:1911.02150*, 2019.

[8] J. Ainslie, J. Lee-Thorp, M. De Jong, Y. Zemlyanskiy, F. Lebrón, and S. Sanghai, “GQA: Training generalized multi-query transformer models from multi-head checkpoints,” *arXiv:2305.13245*, 2023.