HPDC 2025
July 20-23, 2025
Notre Dame, IN, USA

SpeedLLM: An FPGA Co-design of Large Language Model Inference Accelerator

Peipei Wang[1], Wu Guan[1], Liping Liang[1], Zhijun Wang[1], Hanqing Luo[1], Zhibin Zhang[2]
[1]Beijing University of Posts and Telecommunications,China [2]Institute of Computing Technology, Chinese Academy of Sciences,China

ABSTRACT

SpeedLLM, a neural network accelerator designed on the Xilinx Alevo U280 platform and optimized for the Tinyllama framework to enhance edge computing performance. Key innovations include ***data stream parallelism***, a ***memory reuse strategy***, and ***Llama2 operator fusion***, which collectively reduce latency and energy consumption. SpeedLLM's data pipeline architecture optimizes the read-compute-write cycle, while the memory strategy minimizes FPGA resource demands. The operator fusion boosts computational density and throughput. Results show SpeedLLM outperforms traditional Tinyllama implementations, achieving up to **4.8×** faster performance and **1.18×** lower energy consumption, offering improvements in edge devices.

INTRODUCTION

Background. The advancements in Artificial Intelligence have ushered in an era dominated by Large Language Models (LLMs) . When deployed on the edge scene, the architecture of Tinyllama,a compressed and optimized version of LLM, needs to be accelerated to reduce costs and energy consumption.

Challenges&Solutions. LLMs present significant challenges due to their enormous size and computational demands. Model compression techniques such as sparsification and quantization, although beneficial, often suffer from a lack of support by conventional hardware like GPUs, fails to translate into real-world performance gains.Field Programmable Gate Arrays (FPGAs) stand out as a particularly effective solution. The reconfigurability of FPGAs allows for the tuning of hardware algorithms to optimize both computational throughput and memory utilization.

SpeedLLM ARCHITECTURE

Overall architecture

SpeedLLM, an innovative acceleration solution implemented efficient inference of Tinyllama on the Xilinx Alveo U280 FPGA in Fig.1.

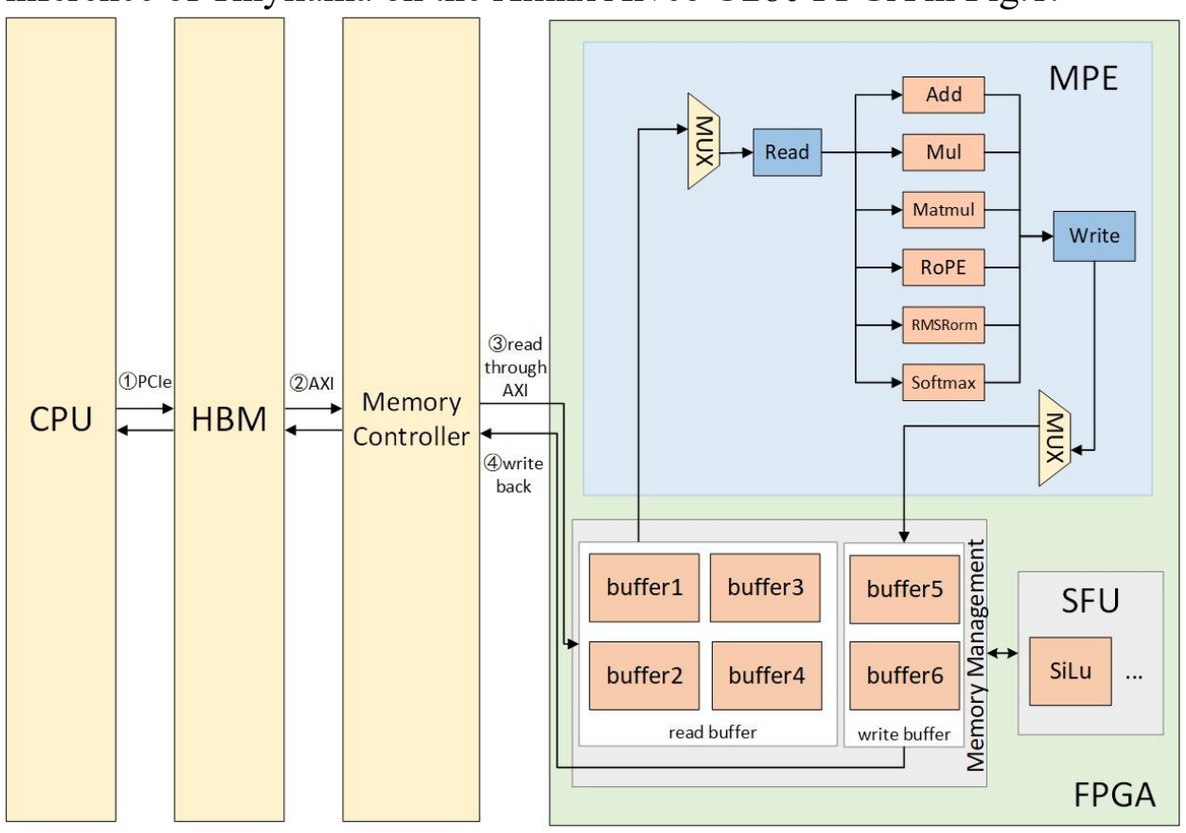


Figure 1:The overall architecture of SpeedLLM, including Matrix Processing Engine(MPE), Memory Management, and Special Function Unit(SFU).

Key contributions

Customized data pipeline: We propose a multi-level read-compute-write iteration that minimizes the iterative and time-consuming cycles, obtaining an increase in the throughput and a reduction in the execution time by ensuring that compute units are constantly fed with data, avoiding idle times.

Memory Allocation Reuse Strategy: This strategy implements a cyclic or loop-back use of memory where each segment is reused after data processing is complete, without waiting for all processing to conclude. This cyclic reuse is managed through efficient scheduling algorithms that track memory usage patterns and predict availability, thus facilitating a more continuous and seamless data feed into the processor.

Operators Fusion of Llama2: Fusing operations into a single, composite operator minimizes the intermediate data writes/read between operations, reducing processing time and memory usage.

EXPERIMENTS AND RESULTS

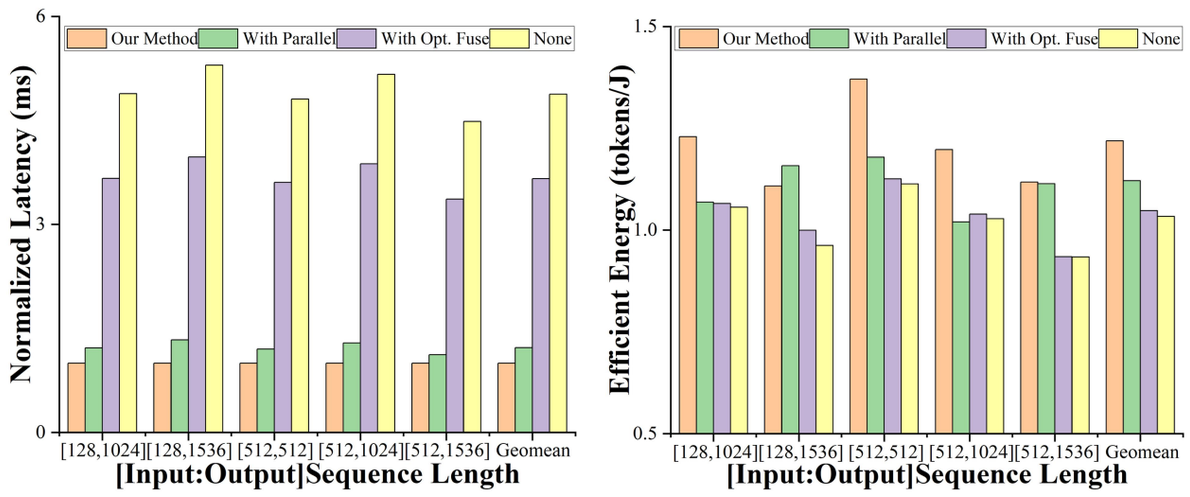
Evaluation Setup

We use a Llama2 architecture model series trained on the TinyStories dataset. We use the stories 15M dataset in Tinyllama and implement the accelerator on U280 FPGAs, verified with RTL emulation using Vitis 2021.1.

Evaluation Results

Latency & Thoughput. Latency measures the total time taken for complete inference by the timing function in the host program, while throughput quantifies the decoding speed by calculating the ratio of output tokens to the duration of the decode stage. Fig.2(a) shows our accelerator significantly surpasses the unoptimized one, delivering a latency speedup of up to 4.8 times.

Energy efficiency. Fig.2(b) shows the energy efficiency of our accelerator. Compared to no fuse accelerator, our method achieves 1.01× energy efficiency, mainly due to reduced redundant off-chip memory communications through the llama model. With higher throughput and comparable poweruse, ours achieves 1.18× better energy efficiency than an unoptimized accelerator. This enhanced performance mainly stems from the use of specially tailored high-performance kernels and their effective integration within our accelerator.



(a) Normalized Latency **(b) Effective energy**

Figure 2: The performance of SpeedLLM

CONCLUSION

SpeedLLM builds upon and extends existing research landscape by integrating several proven optimization strategies into a single coherent system that functions efficiently on the U280 FPGA. By implementing effective methods on LLMs and hardware design, our accelerator significantly enhances the performance capabilities of computing devices, driving forward for real-world applications of deep learning in resource-constrained environments.

REFERENCES

[1] Hongzheng Chen, Jiahao Zhang, Yixiao Du, Shaojie Xiang, Zichao Yue, Niansong Zhang, Yaohui Cai, and Zhiru Zhang. 2024. Understanding the potential of fpga-based spatial acceleration for large language model inference. ACM Transactionson Reconfigurable Technology and Systems 18, 1 (2024), 1–29.
[2] Nazanin Farahpour, Zhenman Fang, and Glenn Reinman. 2020. Fpga-based near data processing platform selection using fast performance modeling (wip paper).In The 21st ACM SIGPLAN/SIGBED Conference on Languages, Compilers, and Tools for Embedded Systems. 151–155.
[3] Shulin Zeng, Jun Liu, Guohao Dai, Xinhao Yang, Tianyu Fu, Hongyi Wang, Wen-heng Ma, Hanbo Sun, Shiyao Li, Zixiao Huang, et al. 2024. Flightllm: Efficient large language model inference with a complete mapping flow on fpgas. In Pro-ceedings of the 2024 ACM/SIGDA International Symposium on Field Programmable Gate Arrays. 223–234.