УДК 004.382.2:004.934.2

Масштабируемая гибридная потоковая архитектура для энергоэффективного вывода трансформерных нейронных сетей на ПЛИС

Зобов Олег Валерьевич zobovov@student.bmstu.ru
SPIN-код: 7699-7709

МГТУ им. Н.Э. Баумана, Москва, Россия

Аннотация. *Рассмотрена масштабируемая гибридная потоковая архитектура для энергоэффективного вывода трансформерных нейронных сетей на программируемых логических интегральных схемах. Основная идея состоит в том, что последовательные фрагменты вычислительного графа трансформера реализуются как крупнозернистые вычислительные блоки и соединяются в составные потоковые конвейеры; специализированный планировщик управляет порядком их активации и повторного использования во времени, при этом сохраняется высокая пропускная способность, характерная для пространственных архитектур. Предлагается мультипроцессорная организация, в которой отдельные ПЛИС выступают вычислительными узлами, связанными кольцевой межсоединительной сетью; накладные расходы синхронизации перекрываются текущими вычислениями в конвейере. Для открытой модели GPT Neo 355M при постобученном квантовании с 8 битными весами и 8 битными активациями получены задержки на токен 6,72 мс (1 узел), 3,90 мс (2 узла на одной ПЛИС) и 2,58 мс (4 узла на двух ПЛИС). По сравнению с современными ускорителями на ПЛИС временного и пространственного типов достигнуто снижение задержки в 2,05 и 1,63 раза соответственно при сопоставимых или меньших затратах ресурсов. Пропускная способность масштабируется до 388,5 токен/с (4 узла).*

Ключевые слова: *трансформерные нейронные сети, ПЛИС, аппаратные ускорители, структурно-фиксированные ускорители, архитектура вычислительных систем*

Scalable hybrid dataflow architecture for energy-efficient inference of transformer neural networks on FPGAs

Zobov Oleg Valerevich zobovov@student.bmstu.ru
SPIN-code: 7699-7709

BMSTU, Moscow, Russia

Abstract. *The paper presents a scalable hybrid dataflow architecture for energy efficient inference of transformer neural networks on FPGAs. Computation stages of the transformer are implemented as coarse grained compute blocks connected into composite dataflow pipelines; a dedicated scheduler governs the order of temporal activation and reuse of these blocks while preserving the high throughput typical of spatial designs. A multiprocessor organization is adopted, where individual FPGAs act as compute nodes connected via a ring interconnect; synchronization overhead is overlapped with ongoing pipeline computa-*

tion. With the open GPT Neo 355M model and post training quantization to 8 bit weights and 8 bit activations, token latencies of 6.72 ms (1 node), 3.90 ms (2 nodes on one FPGA), and 2.58 ms (4 nodes on two FPGAs) are obtained. Compared with state of the art temporal and spatial FPGA accelerators, token latency is reduced by 2.05 and 1.63 times, respectively, with comparable or lower resource usage. Throughput scales to 388.5 tokens/s (4 nodes).

Keywords: *Transformer neural networks, FPGA (field-programmable gate array), hardware accelerators, structurally fixed accelerators, architecture of computing systems*

Введение. Рассмотрена задача снижения задержки и повышения энергоэффективности вывода трансформерных моделей, применяемых в системах обработки естественного языка и генерации кода [1, 2]. Авторегрессионный вывод включает стадии предзаполнения контекста (формирование кэша ключей и значений) и по токенного декодирования; последняя обладает выраженной последовательностью и приводит к недоиспользованию параллельных ресурсов при классической организации вычислений. Существующие ПЛИС подходы либо опираются на временное повторное использование вычислительных блоков с частыми обращениями к внешней памяти, либо реализуют пространственные конвейеры, утилизация которых падает на по токенном декодировании. Настоящая работа предлагает архитектуру, объединяющую сильные стороны обоих подходов и сохраняющую масштабируемость при использовании нескольких кристаллов.

Материалы и методы. Архитектура строится вокруг трех типов крупнозернистых вычислительных блоков, каждый из которых охватывает логически связанную группу операций и включен в общий потоковый конвейер. Планировщик управляет последовательностью активации и повторного использования этих блоков по мере прохождения токена через стадии вычислительного графа трансформера.

Организация данных и память. Весовые коэффициенты линейных преобразований и кеш ключей/значений размещаются в памяти с высокой пропускной способностью (HBM). Доступ к банкам HBM осуществляют параллельные контроллеры прямого доступа к памяти (ПДП), закрепленные за соответствующими каналами. Для повышения эффективности обмена применяется пакетирование 8 битных данных и пакетные передачи; внутри узла обмен между блоками организуется через общие буферы и очереди FIFO. Такое построение разводит по времени операции чтения/записи и вычисления, упрощает размещение и трассировку и позволяет повысить рабочую частоту.

Функциональные вычислительные блоки.

Ядро матричной обработки. Выполняет блочно матричные умножения для всех линейных преобразований (проекции запросов, ключей, значений и выходов, а также полносвязные слои). Архитектурно представляет собой массив умножителей накопителей, сгруппированных по числу каналов HBM; каждый набор связан со своим контроллером ПДП. Частичные суммы накап-

ливаются и передаются в модуль смещения и масштабирования для последующего квантования результатов.

Ядро многоголового внимания. Содержит две независимые секции умножения накопления (для вычисления оценок внимания по ключам и для смешивания по значениям), а также узлы маски причинности и функции softmax. Применяется по головная конвейеризация: вычисление softmax для предыдущей головы перекрывается формированием оценок внимания для текущей.

Ядро нормализации по слоям и остаточных связей. Размещено на критическом пути между линейными преобразованиями и узлом внимания и выполняется в перекрытии с матрической обработкой, тем самым сокращает долю неконвейеризуемых участков.

Планировщик реализован как конечный автомат, который, в зависимости от текущей стадии обработки, активирует соответствующее ядро и организует повторное задействование крупных вычислительных блоков. В отличие от мелкозернистого каскада пространственных операторов, такой подход повышает пиковую утилизацию логики при каждом срабатывании и уменьшает число циклов «чтение — обработка — запись» через память с высокой пропускной способностью, характерных для чисто временных схем.

Мультипроцессорная организация и межсоединения. Система реализуется как мультипроцессорная архитектура, в которой отдельные ПЛИС выступают вычислительными узлами, соединенными кольцевой межсоединительной сетью. Применяется модельный параллелизм: разбиение весов линейных слоев по выходному измерению (каждый узел формирует свою долю выходного вектора), а кеша ключей/значений — по головам внимания. Хост подает на все узлы одинаковые входные эмбеддинги; сборка полного результата выполняется посредством обмена подпакетами по кольцу. Передачи по сети перекрываются текущими вычислениями: в блочно-матричных умножениях обмен результатами предыдущего блока скрывается под расчетом текущего блока, а явные накладные расходы проявляются только на границах крупных задач. Для сопоставления с имеющимися ПЛИС реализациями учитывались как «временные», так и «пространственные» архитектуры.

Квантование. Применяется постобученное квантование с 8-битными весами и 8-битными активациями (масштабы и смещения подбираются по статистике активаций) без дополнительного обучения модели. Такой подход снижает нагрузку на подсистему памяти и умножители накопители при контролируемом снижении точности [3].

Аппаратная реализация. Развязка стадий через очереди FIFO, локализация обменов и привязка контроллеров ПДП к каналам памяти с высокой пропускной способностью позволили достичь рабочей частоты порядка 285 МГц на плате AMD Xilinx Alveo U50 (две суперлогические области; по одному вычислительному узлу на SLR). Проектирование выполнено в Vitis HLS с последующим синтезом в Vivado; модель — GPT Neo 355M (открытая реализация) [2, 4, 5].

Результаты. Оценка проводилась цикл точным моделированием с учетом полосы каналов памяти с высокой пропускной способностью и пропускной способности межсоединений. Для модели GPT Neo 355M с постобученным квантованием (8-битные веса и 8-битные активации) получены: — задержка генерации токена: 6,72 мс (1 узел), 3,90 мс (2 узла на одной ПЛИС), 2,58 мс (4 узла на двух ПЛИС); — пропускная способность: 149,3 токен/с (1 узел), 257,1 токен/с (2 узла; ускорение в 1,72 раза относительно 1 узла), 388,5 токен/с (4 узла; ускорение в 1,51 раза относительно 2 узлов).

Сравнение с ускорителями на ПЛИС. Для сопоставления использованы представитель «временной» архитектуры с командно управляемыми вычислительными ядрами и межкристальной кооперацией и представитель «пространственной» архитектуры на основе систолических массивов [6, 7]. В нормированных метриках предлагаемая архитектура обеспечивает: — 2 узла (1 ПЛИС): уменьшение задержки на токен в 1,41 раза относительно «временной» архитектурой и в 1,08 раза относительно «пространственной»; — 4 узла (2 ПЛИС): уменьшение задержки на токен в 2,05 и в 1,63 раза соответственно, при умеренном потреблении аппаратных ресурсов.

Обсуждение полученных результатов. Выигрыш над «временными» реализациями объясняется уменьшением числа обращений к памяти с высокой пропускной способностью за счет укрупнения стадий в крупные вычислительные блоки и их повторного использования под управлением планировщика, а также перекрытием нормализации и остаточных связей с матрическими операциями. Преимущество над «пространственными» архитектурами проявляется на по токенном декодировании: вместо простоя множества конвейерных стадий выполняется повторное задействование насыщенного блока матричной обработки, а по головная конвейеризация внимания скрывает часть затрат функции softmax и доступа к кешу значений. Сублинейная масштабируемость (переход от одного к двум узлам — ускорение в 1,72 раза; от двух к четырем — в 1,51 раза) обусловлена долей операций, ограниченно кооперируемых между узлами, и «проступанием» затрат на квантование и синхронизацию при малой гранулярности блоков; эти факторы уменьшаются при увеличении нагрузки на узел и масштабе модели [3, 8].

Заключение. Представлена гибридная потоковая архитектура, в которой крупнозернистые вычислительные блоки в составе конвейера и мультипроцессорная организация с узлами на отдельных ПЛИС обеспечивают уменьшение задержки по токенного декодирования и высокую энергоэффективность вывода трансформеров. Для открытой модели GPT Neo 355M достигнуты задержки 2,58–6,72 мс в диапазоне 1–4 узлов и получены выигрыши в 2,05 и 1,63 раза относительно «временных» и «пространственных» ПЛИС ускорителей. Перспективы включают поддержку более крупных языковых моделей, автоматизацию планирования конвейера и оптимизацию межсоединений в кольцевой сети.

Список источников

[1] Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser Ł., Polosukhin I. Attention is all you need. *Advances in neural information processing systems*, 2017, vol. 30, pp. 5998–6008.

[2] Black S., Gao L., Wang P., Leahy C., Biderman S., Hallahan E., Anthony Q., Phang J., Gao L., Chen X. *GPT Neo: Large scale autoregressive language modeling with mesh tensorflow.* URL: https://github.com/EleutherAI/gpt-neo (дата обращения 08.09.2025).

[3] Yao Z., Aminabadi R. Y., Zhang M., Wu X., Li C., He Y. ZeroQuant: Efficient and affordable post training quantization for large scale transformers. *Preprint. arXiv:2206.01861*, 2022, 15 p.

[4] *Alveo U50 Data Center Accelerator Card: Product Brief.* URL: https://www.amd.com/en/products/accelerators/alveo/u50.html (дата обращения 08.09.2025).

[5] *Vitis High Level Synthesis User Guide; Vivado Design Suite Documentation.* URL: https://docs.xilinx.com/ (дата обращения 08.09.2025).

[6] Hong S., Moon S., Kim J., Lee S., Kim M., Lee D., Kim J.Y. DFX: A Low latency Multi FPGA Appliance for accelerating transformer based text generation. *Preprint. arXiv:2209.10797*, 2022, 14 p.

[7] Chen Y., Li T., Chen X., Cai Z., Su T. High Frequency Systolic Array based transformer accelerator on field programmable gate arrays. *Electronics*, 2023, vol. 12, no. 4, art. no. 822. https://doi.org/10.3390/electronics12040822

[8] Wang Z., Huang H., Zhang J., Shuhai A.G. Benchmarking high bandwidth memory on FPGAs. *IEEE 28th Annual International Symposium on Field Programmable Custom Computing Machines (FCCM)*, 2020, pp. 111–119. https://doi.org/10.1109/FCCM48280.2020.00024