НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
сентябрь–октябрь 2026 Том 26 № 5 http://ntv.ifmo.ru/
SCIENTIFIC AND TECHNICAL JOURNAL OF INFORMATION TECHNOLOGIES, MECHANICS AND OPTICS
September–October 2026 Vol. 26 No 5 http://ntv.ifmo.ru/en/
ISSN 2226-1494 (print) ISSN 2500-0373 (online)

НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

doi: 10.17586/2226-1494-2026-26-5-971-980
УДК 004.31

Методы организации подсистемы хранения и передачи данных в аппаратном ускорителе сверточных нейронных сетей

Сергей Михайлович Табунщик[1]✉, Сергей Вячеславович Быковский[2], Иван Викторович Лукашов[3]

[1,2,3] Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация

[1] sergei_tabunshchik@itmo.ru✉, https://orcid.org/0009-0003-2262-0315
[2] sergei_bykovskii@itmo.ru, https://orcid.org/0000-0003-4163-9743
[3] lukashov@itmo.ru, https://orcid.org/0009-0001-8770-9789

Аннотация

Введение. В современных системах краевого искусственного интеллекта применяются специализированные аппаратные ускорители нейронных сетей для обработки большого объема данных непосредственно на пользовательских устройствах. Основными проблемами в организации подсистемы хранения и передачи данных в аппаратных ускорителях являются ограниченная пропускная способность шин, ограниченный объем памяти и высокие требования к скорости обработки и энергопотреблению. Растущие требования к производительности и энергоэффективности нейросетевых процессоров в условиях ограниченных ресурсов встраиваемых систем определяют актуальность исследования методов организации подсистемы хранения и передачи данных. **Метод.** Предложены варианты архитектурной организации подсистемы хранения и передачи данных: с полным управлением со стороны центрального процессора, с кэшированием в память типа Block Random Access Memory, с прямым доступом к памяти через общую шину AXI_GP и с выделенной высокоскоростной шиной AXI_HP. Для оценки эффективности использован метод симуляции инференса сверточного слоя нейронной сети YOLOv5s на модели нейросетевого процессора на уровне регистровых передач с расчетом пропускной способности и производительности для каждого варианта. **Основные результаты.** В результате исследования зависимости характеристик нейросетевого процессора от метода организации подсистемы хранения и передачи данных установлено, что разработанный нейросетевой процессор достигает максимальную производительность в 240 миллиардов целочисленных операций в секунду. Подтверждено преимущество использование интерфейса прямого доступа к оперативной памяти с пропускной способностью 1,2 ГБ/с, выделенного для передачи данных для вычислений. **Обсуждение.** Сравнение показало, что применение прямого доступа к памяти с выделенным высокоскоростным интерфейсом позволяет повысить производительность нейросетевого в среднем в 20 раз. Описанные методы организации подсистемы хранения и передачи данных позволяют достичь производительность в 1,19 раз выше, чем у аналогов, на задачах инференса сверточных нейронных сетей. Результаты демонстрируют потенциал применения разработанных методов при проектировании аппаратных ускорителей для систем краевого искусственного интеллекта.

Ключевые слова

нейросетевой процессор, подсистема передачи данных, сверточные нейронные сети, краевой искусственный интеллект, аппаратное ускорение, ПЛИС, систолический массив, контроллер прямого доступа к памяти, интерфейс AXI

Благодарности

Работа выполнена в рамках НИР 640097 Университета ИТМО.

Ссылка для цитирования: Табунщик С.М., Быковский С.В., Лукашов И.В. Методы организации подсистемы хранения и передачи данных в аппаратном ускорителе сверточных нейронных сетей // Научно-технический вестник информационных технологий, механики и оптики. 2026. Т. 26, № 5. С. 971–980. doi: 10.17586/2226-1494-2026-26-5-971-980

© Табунщик С.М., Быковский С.В., Лукашов И.В., 2026

Methods for organising the data storage and transmission subsystem in a hardware accelerator for convolutional neural networks

Sergei M. Tabunshchik[1]✉, Sergei V. Bykovskii[2], Ivan V. Lukashov[3]

[1,2,3] ITMO University, Saint Petersburg, 197101, Russian Federation

[1] sergei_tabunshchik@itmo.ru✉, https://orcid.org/0009-0003-2262-0315
[2] sergei_bykovskii@itmo.ru, https://orcid.org/0000-0003-4163-9743
[3] lukashov@itmo.ru, https://orcid.org/0009-0001-8770-9789

Abstract

Modern Edge AI systems use specialized hardware neural network accelerators to process large amounts of data directly on user devices. The main challenges in organizing the data storage and transfer subsystem in hardware accelerators are limited bus bandwidth, limited memory capacity, and high requirements for processing speed and energy consumption. The growing demands on the performance and energy efficiency of neural network processors in the context of limited resources in embedded systems make it relevant to study the methods of organizing the data storage and transfer subsystem. The article proposes options for the architectural organization of the data storage and transfer subsystem: with full control from the central processor, with caching in BRAM-type memory, with direct access to memory through the common AXI_GP bus, and with a dedicated high-speed AXI_HP bus. To evaluate the efficiency, the article uses the method of simulating the inference of the convolutional layer of the YOLOv5s neural network on a neural network processor model at the register transfer level, calculating the bandwidth and performance for each option. As a result of the study of the dependence of the characteristics of the neural network processor on the method of organizing the data storage and transmission subsystem, it was found that the developed neural network processor achieves a maximum performance of 240 GOPS. The advantage of using a direct access interface to RAM with a bandwidth of 1.2 GB/s, which is dedicated for data transmission for calculations, has been confirmed. The comparison showed that the use of direct memory access with a dedicated high-speed interface can increase the performance of inference of neural networks by an average of 20 times. The described methods for organizing the data storage and transfer subsystem achieve a performance that is 1.19 times higher than that of analogues on convolutional neural network inference tasks. The results demonstrate the potential for using these methods in the design of hardware accelerators for Edge AI systems.

Keywords

neural network processor, data transmission subsystem, convolutional neural networks, Edge AI, hardware acceleration, FPGA, systolic array, DMA controller, AXI interface

Acknowledgements

The work was carried out within the framework of Research and Development 640097 at ITMO University.

For citation: Tabunshchik S.M., Bykovskii S.V., Lukashov I.V. Methods for organising the data storage and transmission subsystem in a hardware accelerator for convolutional neural networks. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2026, vol. 26, no. 5, pp. 971–980 (in Russian). doi: 10.17586/2226-1494-2026-26-5-971-980

Введение

Современные приложения искусственного интеллекта (ИИ) отличаются необходимостью хранить, передавать и обрабатывать большой объем данных. Интенсивный рост сложности алгоритмов машинного обучения и расширение сфер применения ИИ приводят к экспоненциальному увеличению объемов обрабатываемых данных. Особую актуальность эта проблема приобретает в контексте краевого ИИ-подхода, при котором обработка данных выполняется не в централизованных облачных центрах, а непосредственно на периферийных устройствах или вблизи них [1, 2].

Именно краевой ИИ становятся основой для множества современных приложений: автономных транспортных средств, роботов и дронов, умных камер видеонаблюдения и носимых медицинских устройств. Для ускорения расчета задач краевого ИИ в перечисленных сценариях нагрузка вычислений перекладывается на специализированные аппаратные ускорители нейронных сетей. Ускорители выполняют обработку большого объема данных в реальном времени во встраиваемых системах. Это накладывает ограничения на подсистемы хранения и передачи данных в специализированных ускорителях, интегрированных непосредственно в конечное оборудование. Ограниченная пропускная способность внутренних шин данных приводит к возникновению «узкого места» при передаче нескольких потоков данных между памятью, датчиками и вычислительной подсистемой. К этому добавляются высокие требования к скорости обработки: задержки в передаче и обработке данных могут нарушить работу систем реального времени, где критически важна быстрая реакция на входные сигналы. Существенную проблему представляет энергопотребление: операции по передаче и обработке больших объемов данных создают значительную нагрузку на энергосистему устройства, что особенно критично для автономных и мобильных встраиваемых решений. Кроме того, объем доступной памяти в таких системах строго ограничен, а хранение промежуточных и итоговых данных требует значительный объем локальной памяти. Для сокращения влияния описанных ограничений требуется применение специализированных методов организации подсистемы хранения и передачи данных [3, 4].

Аппаратные ускорители сверточных нейронных сетей «Eyeriss» и «Eyeriss v2» обеспечивают высокую точность при минимальном потреблении энергии в режиме реального времени за счет применения иерархической организации памяти, включающей внешнюю

память, общий буфер кэширования данных и локальные буферы в вычислительных элементах, а также за счет принципа организации вычислений с упором на построчное размещение данных и широковещательное распространение данных по кристаллу. Такая организация подсистемы хранения и передачи данных позволяет размещать данные максимально близко к вычислительным элементам, сокращая число обращений к внешней памяти и снижая энергозатраты на перемещение данных. В «Eyeriss v2» дополнительно применяется сжатое представление данных и адаптивная маршрутизация, что особенно важно для энергоэффективных ускорителей нейронных сетей [5–7].

В ускорителе «Envision» организация подсистемы хранения и передачи данных построена на принципе максимального локального повторного использования данных и минимизации обращений к внешней памяти. Архитектура подсистемы хранения и передачи данных включает внешнюю память, буфер Static Random-Access Memory для промежуточного хранения 65 536 слов данных для параллельной подачи в вычислительную подсистему и контроллер прямого доступа к памяти (ПДП) для ускоренной подгрузки входных данных из внешнего интерфейса ввода/вывода. «Envision» также поддерживает параллельную обработку подслов, что позволяет гибко выбирать разрядность вычислений в зависимости от требований конкретного слоя и тем самым сокращать объем передаваемой информации. Дополнительное применение динамического управления напряжением, точностью и частотой дает возможность адаптировать работу системы к текущей нагрузке и снижать избыточные затраты на хранение и передачу данных [8, 9].

В процессорах «TPUv2» и «TPUv3» для промежуточного хранения и потоковой подачи данных в систолический массив применяется память с высокой пропускной способностью (High Bandwidth Memory, HBM) объемом 8–16 ГБ. Высокая пропускная способность HBM достигается за счет высокой степени параллельности передачи данных с помощью использования 32 128-битных шин, соединяющих модули Block Random Access Memory (BRAM) небольшого объема с вычислительным ядром. Подсистема высокоскоростной межъядерной передачи данных организована с помощью сети ICI с параллельными каналами с пропускной способностью до 656 Гбит/с и обеспечивает построение двумерного тора с двунаправленной пропускной способностью 15,9 Тбит/с. В «TPUv4» подсистема хранения данных расширяется за счет дополнительной распределенной памятью с HBM для быстрого доступа к общим данным из обоих тензорных ядер на чипе. Межъядерная сеть Inter-Carrier Interconnect (ICI) с трехмерной тороидальной структурой реализует систему с неоднородным доступ к памяти (Non-Uniform Memory Access), повышая пропускную способность и отказоустойчивость при масштабе 4096 чипов [10, 11].

Для реализации высокопроизводительного нейросетевого процессора на базе программируемой логической интегральной схемы (ПЛИС) с существующими ограничениями, требуется реализовать подсистему хранения и передачи данных с высокой пропускной способностью. Целью работы является анализ методов организации подсистемы хранения и передачи данных в аппаратных ускорителях нейронных сетей для повышения производительности нейросетевого процессора с авторской архитектурой.

Описание разработанной архитектуры ускорителя сверточных нейронных сетей

В настоящей работе рассматривается метод организации подсистемы хранения и передачи данных нейросетевого процессора, архитектура которого предложена в работе [12]. Изучается зависимость характеристик проектируемого нейросетевого процессора от метода организации подсистемы хранения и передачи данных, позволяющего добиться максимальной производительности вычислений. Исследование позволит сократить задержки при передаче данных, повысить производительность нейросетевого процессора, расширить объем памяти для хранения обрабатываемых данных и сократить объем данных, необходимых для передачи между компонентами нейросетевого процессора.

Выполнен сравнительный анализ методов организации подсистемы хранения и передачи данных нейросетевого процессора. Критериями сравнения выступают пропускная способность и производительность предложенного нейросетевого процессора. Методы организации подсистемы хранения и передачи данных нейросетевого процессора рассматриваются в рамках реализации процессора на базе системы на кристалле ZYNQ-7000.

Центральным элементом нейросетевого процессора является вычислительный блок, позволяющий аппаратно ускорять расчет сверточных слоев нейронных сетей за счет конвейерно-параллельной организации. Так как вычислительный блок является обработчиком данных с высокой пропускной способностью HBM, то подсистемы управления вычислениями и хранения и передачи данных должны обеспечивать высокоскоростную подачу большого объема входных и выходных данных для достижения максимальной производительности.

В предлагаемом варианте реализации нейросетевого процессора главным элементом управления является процессор общего назначения (центральный процессор (ЦП)). Его основными задачами являются конфигурация вычислительной подсистемы под расчет конкретной прикладной задачи, контроль процесса вычисления и организация потока данных между внешними источниками данных через интерфейсы ввода/вывода, Double Data Rate (DDR) памятью и аппаратным ускорителем на ПЛИС (тензорное ядро). Для организации потоков управления и данных в аппаратном ускорителе реализованы коммутатор данных, контроллер управляющих регистров, блок управления и блок памяти. Для взаимодействия с тензорным ядром используется следующая цепочка элементов: интерфейс BRAM с 32-битными линиями данных является входным интерфейсом тензорного ядра, подключение к коммутатору с интерфейсом Advanced eXtensible Interface (AXI) выполнено с помощью AXI-BRAM контроллера и интерфейса ведомого устройства AXI с 32-битными линиями данных, а к коммутатору AXI подключен ЦП с помощью интерфейса ведущего устройства AXI_GP (General Purpose)

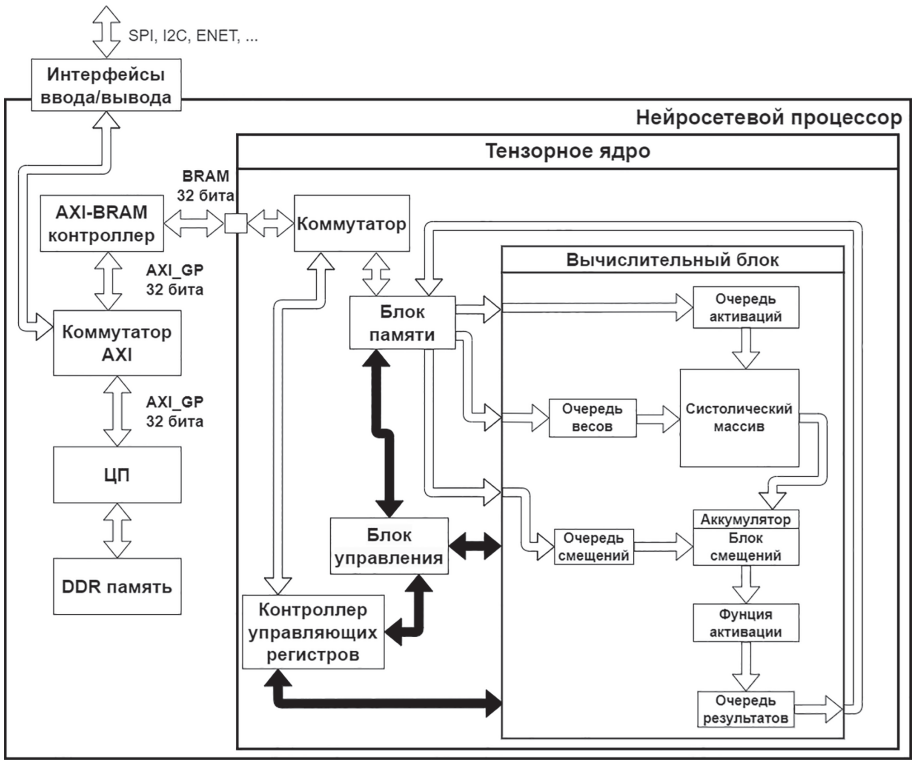


Рис. 1. Структурная схема нейросетевого процессора.
Шины данных (белые стрелки); шины управляющих сигналов (черные стрелки)

Fig. 1. Structural diagram of a neural network processor.
White arrows are data buses, black arrows are control signal buses

с 32-битными линиями данных. Структурная схема нейросетевого процессора представлена на рис. 1.

Коммутатор позволяет направлять данные в блок назначения в зависимости от адреса. Контроллер управляющих регистров хранит конфигурацию вычислительной подсистемы, информацию о текущем состоянии процесса вычисления и предоставляет интерфейс для изменения управляющих сигналов, подключенных к вычислительному блоку. Блок памяти используется для коммутации и кеширования разных типов входных и выходных данных из вычислительного блока. Блок управления используется для синхронизации стадий вычисления с загрузкой и выгрузкой данных из вычислительного блока с помощью управляющих сигналов. Для промежуточного хранения входных активаций, весов, смещений и выходных активаций в вычислительном блоке используются очереди типа First In First Out. Очереди позволяют организовать асинхронную передачу данных между блоком памяти и вычислительной подсистемой.

Описание предложенных методов организации подсистемы хранения и передачи данных

Вариант 1 реализации подсистемы хранения и передачи данных заключается в создании блока памяти в виде коммутатора входных активаций, весов, смещений и выходных активаций между ЦП и вычислительным блоком (рис. 2). Это позволяет полностью делегировать ЦП задачи управления вычислением, подачей входных и выгрузкой выходных данных из вычислительной подсистемы.

Вариант 2 отличается добавлением в блок памяти модуля памяти типа BRAM и логики коммутации данных между локальной памятью и вычислительным блоком (рис. 3). Это позволяет использовать высокоскоростную память для промежуточного кеширования входных активаций, весов, смещений и выходных активаций.

Вариант 3 заключается в разработке дополнительного внешнего интерфейса BRAM, подключенного к блоку памяти и к общей шине AXI через AXI-BRAM контроллер и контроллер ПДП (рис. 4). Этот способ позволяет осуществить подгрузку данных напрямую из внешней DDR-памяти в обход ЦП.

Вариант 4 заключается в выделении отдельного высокоскоростного канала данных AXI_HP (High Performance) с 64-битными линиями данных для подключения контроллера ПДП к внешней DDR-памяти (рис. 5). Этот способ позволяет избежать разделения по времени общей шины для передачи управляющих пакетов и пакетов с данными между ЦП, внешней памятью и аппаратным ускорителем.

Экспериментальное исследование зависимости характеристик нейросетевого процессора от метода организации подсистемы хранения и передачи данных

Для оценки пропускной способности элементов подсистемы хранения и передачи данных и вычис-

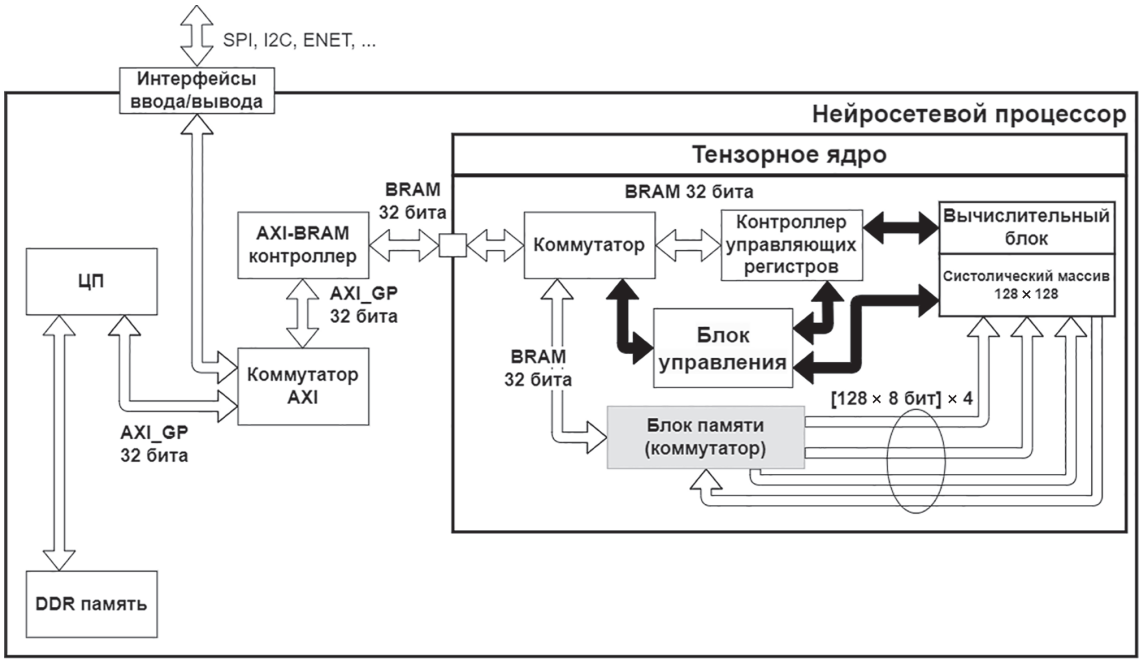


Рис. 2. Коммутируемый поток данных. Блок памяти подключен к вычислительному блоку четырьмя шинами данных по 128 линий каждая шириной 8 бит

Fig. 2. Switched data flow. The memory unit is connected to the computing unit by 4 data buses with 128 8-bit data lines

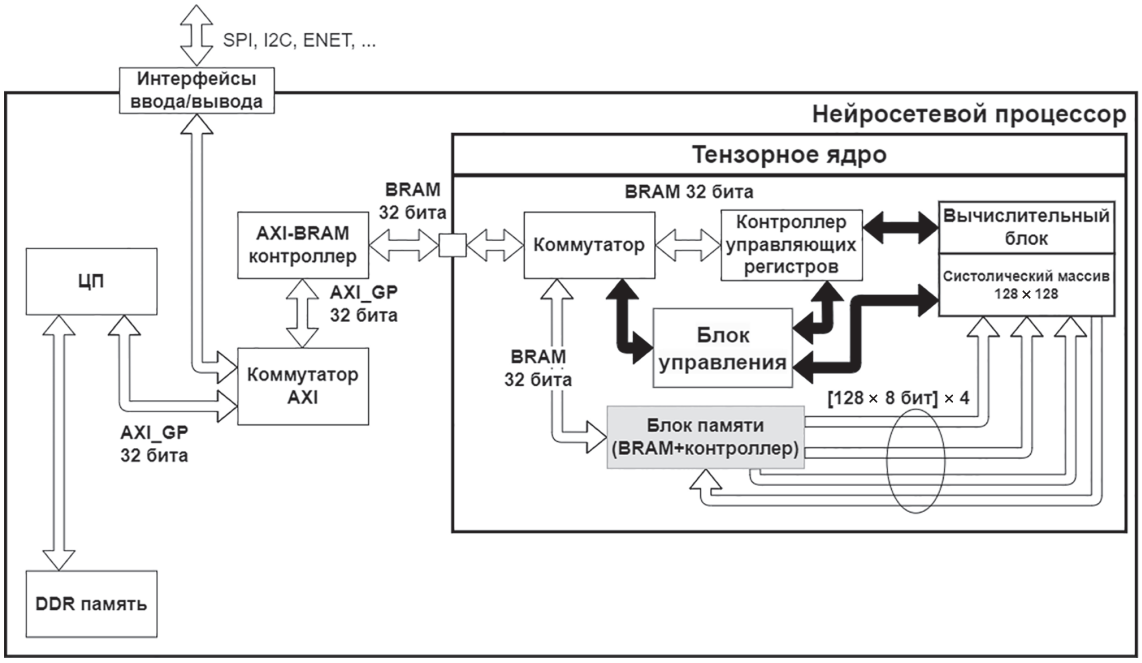


Рис. 3. Поток данных с кешированием

Fig. 3. Cached data flow

лительной подсистемы и производительности нейросетевого процессора применено симуляционное моделирование расчета прикладной задачи на уровне регистровых передач. В качестве прикладной задачи использован расчет первого сверточного слоя (conv2d) нейронной сети YOLOv5s с параметрами: размеры входной матрицы активаций: $\mathbf{W}_{in} = 640$, $\mathbf{H}_{in} = 640$, $\mathbf{C}_{in} = 3$; размер ядра свертки $K = 6$; количество выходных каналов $\mathbf{C}_{out} = 64$; шаг смещения ядра свертки (stride) $\mathbf{s}_{conv} = 2$; размер дополнения границ входной матрицы (padding) $\mathbf{p}_{conv} = 2$; коэффициент разрежения входной матрицы (dilation) $\mathbf{d}_{conv} = 1$ [13]. В результате расчета сверточного слоя с представленными параметрами формируется матрица результирующих активаций размером: $\mathbf{W}_{out} = 320$, $\mathbf{H}_{out} = 320$, $\mathbf{C}_{out} = 64$.

Для оценки зависимости характеристик проектируемого нейросетевого процессора от метода организации подсистемы хранения и передачи данных использова-

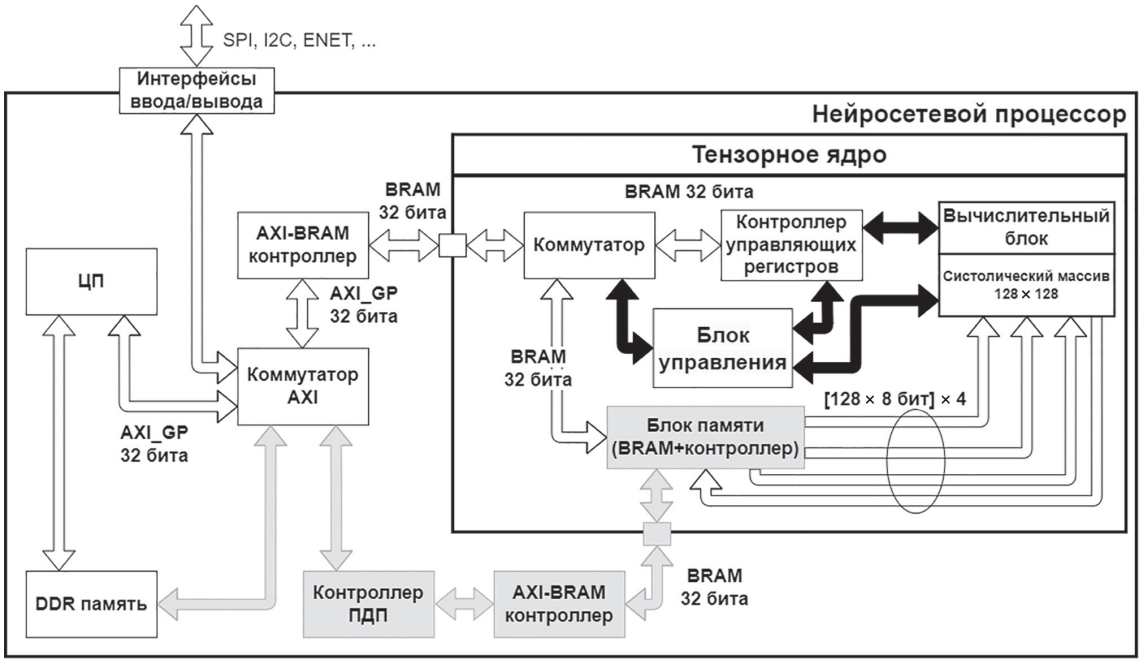


Рис. 4. Поток данных с прямого доступа к памяти через интерфейс AXI общего назначения (AXI general purpose, AXI_GP)

Fig. 4. Data flow with DMA via AXI_GP (general purpose)

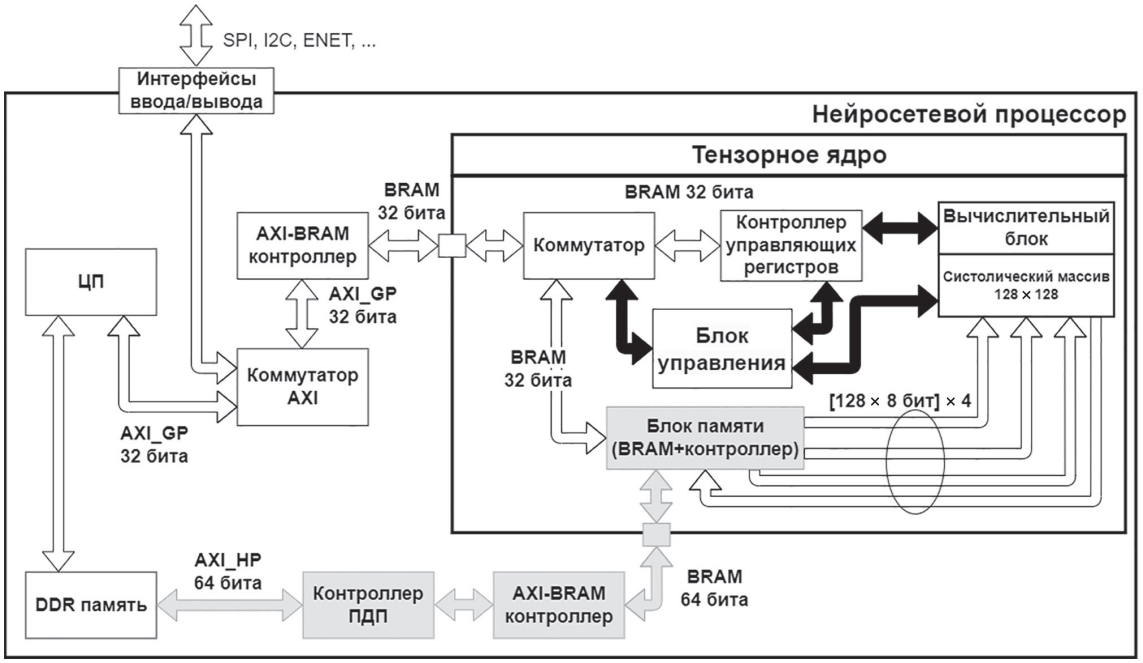


Рис. 5. Поток данных с прямого доступа к памяти через высокопроизводительный интерфейс AXI (AXI high performance, AXI_HP)

Fig. 5. Data flow with DMA via AXI_HP (high performance)

ны следующие микроархитектурные параметры: размер пакета данных (четыре значения по 8 бит); размер пакета управления (32-битное значение управляющего регистра); размер систолического массива [14] (128 × 128 вычислительных элементов); размер шин передачи входных активаций, весов, смещений и выходных активаций (128 параллельных значений по 8 бит); частота тактового сигнала ЦП, интерфейса AXI_GP и вычислительного ядра (tensor_core) (100 МГц); частота тактового сигнала интерфейса AXI_HP и контроллера ПДП (150 МГц).

Пропускная способность элементов нейросетевого процессора рассчитана как максимальный объем данных в гигабайтах, который может быть обработан или передан через данный элемент за одну секунду.

Для расчета итоговой производительности нейросетевого процессора на описанной задаче учитывается время, затраченное на передачу входных данных, вре-

мя, затраченное на вычисление результатов в вычислительном конвейере, время, затраченное на остановку конвейера для ожидания подгрузки новых данных в очереди или выгрузки результатов из очередей. В качестве метрики производительности использовано количество миллиардов целочисленных операций совмещенного умножения-сложения (Multiply-Accumulate, MAC) с 8-битными аргументами, которые могут быть вычислены систолическим массивом за одну секунду (GOPS).

В сверточном слое с микроархитектурными параметрами K, $\mathbf{C}_{out}$, $\mathbf{s}_{conv}$, $\mathbf{p}_{conv}$, $\mathbf{d}_{conv}$ операция свертки выполняется с использованием входных матриц активаций $\mathbf{A}$ и весов $\mathbf{B}$ по формуле;

$$\mathbf{R}_{\mathbf{W}_{out}\times\mathbf{H}_{out}\times\mathbf{C}_{out}} = \mathbf{A}_{\mathbf{W}_{in}\times\mathbf{H}_{in}\times\mathbf{C}_{in}} \cdot \mathbf{B}_{\mathbf{K}\times\mathbf{K}\times\mathbf{C}_{in}\times\mathbf{C}_{out}},$$

где $\mathbf{R}$ — выходная матрица активаций; $\mathbf{W}_{in}/\mathbf{W}_{out}$ — ширина входной/выходной матриц активаций; $\mathbf{H}_{in}/\mathbf{H}_{out}$ — высота входной/выходной матриц активаций; $\mathbf{C}_{in}/\mathbf{C}_{out}$ — количество каналов во входной/выходной матрицах активаций.

Значения элементов матрицы результирующих активаций $\mathbf{R}$ получим в виде:

$$r_{xyz} = \sum_{k=0}^{\mathbf{C}_{in}} \sum_{j=0}^{K} \sum_{i=0}^{K} a_{(x+i)(y+j)k} b_{ijkz},$$

где r — элемент выходной матрицы активаций $\mathbf{R}$; a — элемент входной матрицы активаций $\mathbf{A}$; b — элемент матрицы весов $\mathbf{B}$; $x \in (0; \mathbf{W}_{out}]$; $y \in (0; \mathbf{H}_{out}]$; $z \in (0; \mathbf{C}_{out}]$ [15].

Ширина $\mathbf{W}_{out}$ и высота $\mathbf{H}_{out}$ матрицы результирующих активаций $\mathbf{R}$ рассчитываются по формулам:

$$\mathbf{W}_{out} = \left\lfloor \frac{\mathbf{W}_{in} + 2\mathbf{p}_{conv} - \mathbf{d}_{conv}(K-1) - 1}{\mathbf{s}_{conv}} + 1 \right\rfloor,$$

$$\mathbf{H}_{out} = \left\lfloor \frac{\mathbf{H}_{in} + 2\mathbf{p}_{conv} - \mathbf{d}_{conv}(K-1) - 1}{\mathbf{s}_{conv}} + 1 \right\rfloor.$$

Для вычисления операции свертки в двумерном систолическом массиве со значениями микроархитектурных параметров K, $\mathbf{C}_{out}$, $\mathbf{s}_{conv}$, $\mathbf{p}_{conv}$, $\mathbf{d}_{conv}$ необходимо использовать двумерные матрицы активаций $\mathbf{A}_{2d}$ и весов $\mathbf{B}_{2d}$. Следовательно, в двумерном представлении операции свертки имеют вид

$$\mathbf{R}_{2d_{(\mathbf{W}_{out}\mathbf{H}_{out})\mathbf{C}_{out}}} = \mathbf{A}_{2d\ (\mathbf{W}_{out}\mathbf{H}_{out})(K\cdot K\cdot\mathbf{C}_{in})} \cdot \mathbf{B}_{2d_{(K\cdot K\cdot\mathbf{C}_{in})\mathbf{C}_{out}}}.$$

Количество операций MAC $\mathbf{op}_{task}$, необходимых для вычисления произведения матриц $\mathbf{A}_{2d}$ и $\mathbf{B}_{2d}$, рассчитывается по формуле

$$\mathbf{op}_{task} = (\mathbf{W}_{out}\mathbf{H}_{out})(K\cdot K\cdot\mathbf{C}_{in})\mathbf{C}_{out},$$

где $\mathbf{W}_{out} = 320$, $\mathbf{H}_{out} = 320$, $K = 6$, $\mathbf{C}_{in} = 3$, $\mathbf{C}_{out} = 64$.

В результате подстановки значений получим, что для выполнения операции умножения матриц $\mathbf{A}_{2d}$ на $\mathbf{B}_{2d}$ в выбранной прикладной задаче, требуется произвести $\mathbf{op}_{task} = 707\,788\,800$ операций MAC.

Для расчета производительности P количество целочисленных операций MAC с 8-битными аргументами в секунду применим формулу

$$P = \frac{\mathbf{op}_{task}}{t_{task}},$$

где t_{task} — время, затраченное на выполнение задачи.

В рамках проведения оценки характеристик нейросетевого процессора определена пропускная способность элементов нейросетевого процессора в следующих случаях: максимальная расчетная пропускная способность вычислительного модуля; пропускная способность основных интерфейсов, необходимая для достижения максимальной расчетной пропускной способности вычислительного модуля; максимально возможная пропускная способность вычислительного модуля во всех вариантах реализации подсистемы хранения и передачи данных; максимально возможная пропускная способность основных интерфейсов. Основными рассматриваемыми интерфейсам являются: между блоком памяти в тензорном ядре и вычислительным модулем; между коммутатором и блоком памяти в тензорном ядре; между коммутатором и контроллером управляющих регистров; между блоком памяти в тензорном ядре и контроллером ПДП, AXI_GP, AXI_HP. Расчетные оценки и значения пропускной способности элементов нейросетевого процессора в предложенных вариантах реализации подсистемы хранения и передачи данных представлены в табл. 1. Расчетные оценки измеряемых характеристик отображают максимальную производительность основных интерфейсов, необходимую для достижения максимальной пропускной способности вычислительного модуля. Для расчета пропускной способности вычислительного модуля и его интерфейсов в вариантах с использованием локальной памяти для кэширования учитывается частота промаха кэша P_{miss}. Это позволяет учитывать ситуации, в которых необходимые для вычислений данные отсутствуют в кэше и требуется больше времени на подгрузку данных из основной памяти.

На основе рассчитанных оценок пропускной способности вычислительного блока получены оценки производительности нейросетевого процессора на описанной прикладной задаче. Расчетные оценки и значения производительности нейросетевого процессора в предложенных вариантах реализации подсистемы хранения и передачи данных представлены в табл. 2. Расчетные оценки измеряемых характеристик отображают максимальную производительность и пропускную способность, которые может достичь вычислительный модуль при максимальной заполняемости данными.

Полное делегирование ЦП задач управления вычислением, подачей входных и выгрузкой выходных данных из вычислительной подсистемы в варианте 1 реализации подсистемы хранения и передачи данных приводит к необходимости постоянного контроля состояния процесса вычисления с помощью операций чтения контрольных регистров. Это необходимо для корректного принятия решения о загрузке и выгрузки

Таблица 1. Пропускная способность элементов нейросетевого процессора в разных вариантах реализации подсистемы хранения и передачи данных, ГБ/с

Table 1. Bandwidth of neural network processor elements in different implementations of the data storage and transmission subsystem, GB/s

Элемент процессора	Расчетные оценки	Вариант реализации			
		1	2	3	4
Вычислительный блок	12	0,1	$4(1 - P_{miss}) + 0{,}1P_{miss}$	$4(1 - P_{miss}) + 0{,}14P_{miss}$	$4(1 - P_{miss}) + 0{,}3P_{miss}$
Шина между блоком памяти и вычислительным блоком	12	0,1	$4(1 - P_{miss}) + 0{,}1P_{miss}$	$4(1 - P_{miss}) + 0{,}14P_{miss}$	$4(1 - P_{miss}) + 0{,}3P_{miss}$
Шина между коммутатором и блоком памяти	36	0,3	0,3	0,4	0,4
Шина между коммутатором и контроллером управляющих регистров	12	0,1	0,1	0,4	0,4
Шина AXI_GP	48	0,4	0,4	0,4	0,4
Шина между блоком памяти и контроллером ПДП	36	—	—	0,4	1,2
Шина AXI_HP	36	—	—	—	1,2

Таблица 2. Оценки производительности нейросетевого процессора и пропускной способности вычислительного модуля

Table 2. Estimates of the performance of the neural network processor and the throughput of the computing module

Характеристика	Расчетные оценки	Вариант реализации			
		1	2	3	4
Пропускная способность computing_unit, ГБ/с	12	0,1	0,1–4	0,14–4	0,3–4
Производительность процессора *P*, GOPS	720	6	6–240	8,4–240	18–240

данных. В свою очередь данный контроль приводит к двойному количеству операций чтения и записи по AXI-BRAM интерфейсу по отношению к количеству операций передачи полезных данных для вычислений. В варианте 1 ограничением является связка ЦП и внешней DDR-памяти, так как для организации коммутации данных между памятью и вычислительным модулем требуется выполнять большое количество дорогих по времени операций чтения и записи во внешнюю память. К дополнительному замедлению вычислений приводит необходимость тратить процессорное время ЦП на взаимодействия с внешними интерфейсами для получения новых данных и отправки результатов обработки.

Использование высокоскоростной памяти типа BRAM в варианте 2 для промежуточного кеширования входных активаций, весов, смещений и выходных активаций позволяет сократить количество операций чтения и записи во внешнюю память за счет переиспользования данных, временно расположенных в кэш-памяти. Применение логики коммутации данных между локальной памятью и вычислительным блоком позволяет сократить количество операций чтения контрольных регистров для контроля загрузки данных в вычислительную подсистему и перейти к загрузке данных по прерыванию. Также логика коммутации позволяет сократить объем памяти необходимый для хранения данных за счет отсутствия необходимости хранить нулевые и незначащие значения разряженных матриц. Однако недостатком варианта 2 является объем кэш-памяти, который может быть реализован на ограниченных ресурсах ПЛИС и не позволяет разместить весь объем данных необходимых для расчета прикладной задачи. В результате частый пропуск кэша приводит к необходимости подгружать большой объем данных через медленный ЦП.

Применение контроллера ПДП в варианте 3 реализации позволяет сократить процессорное время ЦП, необходимое для подгрузки новых данных в кэш-память. Однако использование общей шины для ПДП, передачи пакетов с информацией о состоянии вычисления и взаимодействия с внешними интерфейсами для получения новых данных и отправки результатов обработки является проблемой и не позволяет организовать высокоскоростной доступ к внешней памяти с данными.

Организация отдельного выделенного AXI_HP интерфейса с повышенной пропускной способностью для ПДП в варианте 4 реализации подсистемы хранения и передачи данных позволяет сократить время простоя системы при возникновении промаха кэша и необходимости подгрузки новых в кэш-память.

Заключение

Проведенное исследование позволило выявить существенное влияние методов организации подсистемы хранения и передачи данных на производительность нейросетевого процессора. Описанные методы организации подсистемы хранения и передачи данных позволяют достичь производительность в 240 GOPS на

задаче инференса сверточных нейронных сетей, что в 1,19 раз выше, чем у аналогов: «Eyeriss» — 20 GOPS и «Eyeriss v2» — 201 GOPS. При этом теоретически возможная производительность выше в 3,58 раз, чем у аналогов, что показывает возможность достижения более высокой скорости на задачах расчета вывода сверточных нейронных сетей. С учетом ограничений пропускной способности подсистемы хранения и передачи данных спроектированный нейросетевой процессор позволяет достичь производительность соизмеримую с эффективностью аппаратных ускорителей сверточных нейронных сетей «Eyeriss» и «Eyeriss v2». Разработанные методы могут быть применены при проектировании аппаратных ускорителей для систем краевого искусственного интеллекта. Перспективными направлениями дальнейших исследований являются разработка адаптивных алгоритмов управления кэшем и оптимизация механизмов прямого доступа к памяти.

Литература

1. Nakhle F. Shrinking the giants: Paving the way for TinyAI // Device. 2024. V. 2. N 8. P. 100411. doi: 10.1016/j.device.2024.100411
2. Gill S.S., Golec M., Hu J., Xu M., Du J., Wu H., et al. Edge AI: A taxonomy, systematic review and future directions // Cluster Computing. 2025. V. 28. N 1. P. 18. doi: 10.1007/s10586-024-04686-y
3. Chenna D. Evolution of convolutional neural network (CNN): compute vs memory bandwidth for edge AI // arXiv. 2023. arXiv:2311.12816. 2023. doi: 10.48550/arXiv.2311.12816
4. Montgomerie-Corcoran A., Bouganis C.S. POMMEL: Exploring off-chip memory energy & power consumption in convolutional neural network accelerators // Proc. of the 24th Euromicro Conference on Digital System Design (DSD). 2021. P. 442–448. doi: 10.1109/dsd53832.2021.00073
5. Chen Y.-H., Krishna T., Emer J.S., Sze V. Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks // IEEE Journal of Solid-State Circuits. 2017. V. 52. N 1. P. 127–138. doi: 10.1109/jssc.2016.2616357
6. Sze V., Chen Y.-H., Yang T.-J., Emer J.S. Efficient processing of deep neural networks: A tutorial and survey // Proceedings of the IEEE. 2017. V. 105. N 12. P. 2295–2329. doi: 10.1109/jproc.2017.2761740
7. Chen Y.H., Yang T.-J., Emer J., Sze V. Eyeriss v2: A flexible accelerator for emerging deep neural networks on mobile devices // IEEE Journal on Emerging and Selected Topics in Circuits and Systems. 2019. V. 9. N 2. P. 292–308. doi: 10.1109/jetcas.2019.2910232
8. Moons B., Uytterhoeven R., Dehaene W., Verhelst M. 14.5 envision: A 0.26-to-10tops/w subword-parallel dynamic-voltage-accuracy-frequency-scalable convolutional neural network processor in 28 nm fdsoi // Proc. of the IEEE International Solid-State Circuits Conference (ISSCC). 2017. P. 246–247. doi: 10.1109/ISSCC.2017.7870353
9. Yin S., Ouyang P., Tang S., Tu F., Li X., Zheng S., et al. A high energy efficient reconfigurable hybrid neural network processor for deep learning applications // IEEE Journal of Solid-State Circuits. 2017. V. 53. N 4. P. 968–982. doi: 10.1109/jssc.2017.2778281
10. Jouppi N.P., Yoon D.H., Kurian G., Li S., Patil N., Laudon J., et al. A domain-specific supercomputer for training deep neural networks // Communications of the ACM. 2020. V. 63. N 7. P. 67–78. doi: 10.1145/3360307
11. Jouppi N., Kurian G., Li S., Ma P., Nagarajan R., Nai R., et al. TPU v4: An optically reconfigurable supercomputer for machine learning with hardware support for embeddings // Proc. of the 50th Annual International Symposium on Computer Architecture. 2023. P. 1–14. doi: 10.1145/3579371.3589350
12. Tabunshchik S., Bykovskii S., Kustarev P. Neural processor microarchitecture design for CNN processing acceleration // Proceedings of SPIE. 2024. V. 13239. P. 132390L. doi: 10.1117/12.3036453
13. Yu K., Tang G., Chen W., Hu S., Li Y., Gong H. MobileNet-YOLO v5s: An improved lightweight method for real-time detection of sugarcane stem nodes in complex natural environments // IEEE Access. 2023. V. 11. P. 104070–104083. doi: 10.1109/access.2023.3317951
14. Peddawad C., Goel A. Matrix-Matrix Multiplication Using Systolic Array Architecture in Bluespec. Project report. 2015.
15. Khan A., Sohail A., Zahoora U., Qureshi A.S. A survey of the recent architectures of deep convolutional neural networks // Artificial Intelligence Review. 2020. V. 53. N 8. P. 5455–5516. doi: 10.1007/s10462-020-09825-6

References

1. Nakhle F. Shrinking the giants: Paving the way for TinyAI. *Device*, 2024, vol. 2, no. 8, pp. 100411. doi: 10.1016/j.device.2024.100411
2. Gill S.S., Golec M., Hu J., Xu M., Du J., Wu H., et al. Edge AI: A taxonomy, systematic review and future directions. *Cluster Computing*, 2025, vol. 28, no. 1, pp. 18. doi: 10.1007/s10586-024-04686-y
3. Chenna D. Evolution of convolutional neural network (CNN): compute vs memory bandwidth for edge AI. *arXiv*, 2023. arXiv:2311.12816. 2023. doi: 10.48550/arXiv.2311.12816
4. Montgomerie-Corcoran A., Bouganis C.S. POMMEL: Exploring off-chip memory energy & power consumption in convolutional neural network accelerators. *Proc. of the 24th Euromicro Conference on Digital System Design (DSD)*, 2021, pp. 442–448. doi: 10.1109/dsd53832.2021.00073
5. Chen Y.-H., Krishna T., Emer J.S., Sze V. Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks. *IEEE Journal of Solid-State Circuits*, 2017, vol. 52, no. 1, pp. 127–138. doi: 10.1109/jssc.2016.2616357
6. Sze V., Chen Y.-H., Yang T.-J., Emer J.S. Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, 2017, vol. 105, no. 12, pp. 2295–2329. doi: 10.1109/jproc.2017.2761740
7. Chen Y.H., Yang T.-J., Emer J., Sze V. Eyeriss v2: A flexible accelerator for emerging deep neural networks on mobile devices. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2019, vol. 9, no. 2, pp. 292–308. doi: 10.1109/jetcas.2019.2910232
8. Moons B., Uytterhoeven R., Dehaene W., Verhelst M. 14.5 envision: A 0.26-to-10tops/w subword-parallel dynamic-voltage-accuracy-frequency-scalable convolutional neural network processor in 28 nm fdsoi. *Proc. of the IEEE International Solid-State Circuits Conference (ISSCC)*, 2017, pp. 246–247. doi: 10.1109/ISSCC.2017.7870353
9. Yin S., Ouyang P., Tang S., Tu F., Li X., Zheng S., et al. A high energy efficient reconfigurable hybrid neural network processor for deep learning applications. *IEEE Journal of Solid-State Circuits*, 2017., vol. 53, no. 4, pp. 968–982. doi: 10.1109/jssc.2017.2778281
10. Jouppi N.P., Yoon D.H., Kurian G., Li S., Patil N., Laudon J., et al. A domain-specific supercomputer for training deep neural networks. *Communications of the ACM*, 2020, vol. 63, no. 7, pp. 67–78. doi: 10.1145/3360307
11. Jouppi N., Kurian G., Li S., Ma P., Nagarajan R., Nai R., et al. TPU v4: An optically reconfigurable supercomputer for machine learning with hardware support for embeddings. *Proc. of the 50th Annual International Symposium on Computer Architecture*, 2023, pp. 1–14. doi: 10.1145/3579371.3589350
12. Tabunshchik S., Bykovskii S., Kustarev P. Neural processor microarchitecture design for CNN processing acceleration. *Proceedings of SPIE*, 2024, vol. 13239, pp. 132390L. doi: 10.1117/12.3036453
13. Yu K., Tang G., Chen W., Hu S., Li Y., Gong H. MobileNet-YOLO v5s: An improved lightweight method for real-time detection of sugarcane stem nodes in complex natural environments. *IEEE Access*, 2023, vol. 11, pp. 104070–104083. doi: 10.1109/access.2023.3317951
14. Peddawad C., Goel A. *Matrix-Matrix Multiplication Using Systolic Array Architecture in Bluespec*. Project report. 2015.
15. Khan A., Sohail A., Zahoora U., Qureshi A.S. A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, 2020, vol. 53, no. 8, pp. 5455–5516. doi: 10.1007/s10462-020-09825-6

Авторы

Табунщик Сергей Михайлович — аспирант, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, SC 59508353800, https://orcid.org/0009-0003-2262-0315, sergei_tabunshchik@itmo.ru
Быковский Сергей Вячеславович — кандидат технических наук, доцент, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, SC 57216469537, https://orcid.org/0000-0003-4163-9743, sergei_bykovskii@itmo.ru
Лукашов Иван Викторович — кандидат технических наук, младший научный сотрудник, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, https://orcid.org/0009-0001-8770-9789, lukashov@itmo.ru

Authors

Sergei M. Tabunshchik — PhD Student, ITMO University, Saint Petersburg, 197101, Russian Federation, SC 59508353800, https://orcid.org/0009-0003-2262-0315, sergei_tabunshchik@itmo.ru
Sergei V. Bykovskii — PhD, Associate Professor, ITMO University, Saint Petersburg, 197101, Russian Federation, SC 57216469537, https://orcid.org/0000-0003-4163-9743, sergei_bykovskii@itmo.ru

Ivan V. Lukashov — PhD, Junior Researcher, ITMO University, Saint Petersburg, 197101, Russian Federation, https://orcid.org/0009-0001-8770-9789, lukashov@itmo.ru

Статья поступила в редакцию 20.05.2026
Одобрена после рецензирования 16.08.2026
Принята к печати 22.09.2026

Received 20.05.2026
Approved after reviewing 16.08.2026
Accepted 22.09.2026

Работа доступна по лицензии
Creative Commons
«Attribution-NonCommercial»