



<http://dx.doi.org/10.35596/1729-7648-XXXX-XX-X-XX-XX>

*Оригинальная статья*  
*Original paper*

УДК 004.021

## АППАРАТНАЯ РЕАЛИЗАЦИЯ НЕЙРОННОЙ СЕТИ ПРЯМОГО РАСПРОСТРАНЕНИЯ ДЛЯ РАСПОЗНАВАНИЯ РУКОПИСНЫХ ЦИФР НА БАЗЕ FPGA

Е.А. КРИВАЛЬЦЕВИЧ, М.И. ВАШКЕВИЧ

*Белорусский государственный университет информатики и радиоэлектроники  
(г. Минск, Республика Беларусь)*

*Поступила в редакцию (дату отмечает редакция)*

© Белорусский государственный университет информатики и радиоэлектроники, 2024

**Аннотация.** Представлена реализация нейронной сети для распознавания рукописных цифр на базе платформы Zynq-7000. Выполнен анализ влияния точности представления параметров НС на её производительность, а также на требуемые для реализации аппаратные ресурсы ПЛИС.

**Ключевые слова:** нейронная сеть, распознавание рукописных цифр, полносвязный слой, MNIST, FPGA, PYNQ, битовые плоскости.

**Конфликт интересов.** Авторы заявляют об отсутствии конфликта интересов.

**Для цитирования.** Кривальцевич Е.А., Вашкевич М.И. Аппаратная реализация нейронной сети прямого распространения для распознавания рукописных цифр на базе FPGA. *Цифровая трансформация*. 20\*\*;  
\*\*(\*): \*\*\*-\*\*\*.

## HARDWARE IMPLEMENTATION OF A DIRECT PROPAGATION NEURAL NETWORK FOR HANDWRITTEN DIGIT RECOGNITION BASED ON FPGA

EGOR A. KRIVALCEVICH, MAXIM I. VASHKEVICH

*Belarusian State University of Informatics and Radioelectronics (Minsk, Republic of Belarus)*

*Submitted*

© Belarusian State University of Informatics and Radioelectronics, 2024

**Abstract.** This paper presents the implementation of a neural network for recognizing handwritten digits based on the Zynq-7000 platform is presented. An analysis of the impact of the accuracy of the NN parameter representation on its performance, as well as on the FPGA hardware resources required for implementation, is performed.

**Keywords:** neural network, handwritten digit recognition, fully connected layer, MNIST, FPGA, PYNQ, bit planes.

**Conflict of interests.** The authors declare no conflict of interests.

**For citation.** Krivalcevic E.A., Vashkevich M.I. Hardware implementation of a direct propagation neural network for recognizing handwritten digits based on FPGA. *Digital transformation*. 20\*\*; \*\*(\*): \*\*\*-\*\*\*.

## Введение

Нейронные сети (НС) играют ключевую роль в развитии информационных технологий, особенно в таких областях, как компьютерное зрение и искусственный интеллект [\*\*]. Широкое распространение НС приводит к тому, что появляется необходимость создания специальных аппаратных акселераторов, позволяющих повысить производительность приложений, основанных на нейросетевых технологиях. Программируемые логические интегральные схемы (ПЛИС) типа FPGA (Field Programmable Gate Array) представляют собой реконфигурируемые вычислительные платформы, имеющие невысокое энергопотребление. По этой причине ПЛИС часто выбирают в качестве вычислительной среды для реализации НС, особенно в тех случаях, когда производительности процессоров общего назначения недостаточно, а высокое энергопотребление графических процессоров неприемлемо. Это особенно актуально в контексте разработки встраиваемых систем и роботизированных платформ [1-3].

К преимуществам аппаратной реализации НС на базе FPGA является возможность использовать для пользовательских типов данных, позволяющих контролировать точность представления параметров нейросетевой модели [1-3]. Причем выбор точности представления напрямую будет влиять на аппаратные затраты ПЛИС, необходимые для обеспечения выполнения операций над данными.

Целью данной работы является разработка аппаратной реализации однослойной НС прямого распространения для распознавания рукописных цифр, а также исследование влияния разрядности коэффициентов НС на точность распознавания и на аппаратные затраты ПЛИС.

Разработка и исследование аппаратной реализации НС было разбито на несколько этапов. На первом этапе необходимо реализовать Python описание НС с использованием PyTorch, а также обучить сеть. Далее подготовить System Verilog описание НС. Затем осуществить отладку и аппаратно протестировать. На последнем этапе необходимо выполнить эксперимент и проанализировать результаты.

## Разработка программной модели НС

В работе рассматривается задача распознавания рукописных цифр по изображениям из набора данных MNIST. Используемый набор данных MNIST содержит 70 тыс. полутоновых изображений размера 28x28 пикселей рукописных цифр от 0 до 9 [4]. Набор разбит на две части: тренировочная выборка – 60 тыс. изображений и тестовая выборка – 10 тыс. изображений.

В работе рассматривается однослойная НС прямого распространения, состоящая из полносвязного слоя с выходной функцией активации softmax. Структура НС представлена на рис.1.

На входе  $784 = 28 \cdot 28$  нейрона, каждый подключен к одному из пикселей изображения. На выходе слой с десятью нейронами по одному на цифру. Соответственно, каждый из 10 выходов формируется как линейная комбинация 784 входов:

$$y_i = \sum_{j=0}^{783} w_{ij} \cdot x_j + b_i, \quad (1)$$

где  $w_{ij}$  – весовой коэффициент,  $x_j$  – текущий пиксель,  $b_i$  – смещение,  $i = 0, 9$ . Смещение для нейрона  $i$  – это постоянное значение, которое добавляется ко всем взвешенным суммам. Оно позволяет модели лучше обучаться, так как даёт гибкость для смещения активации в сторону от начала координат.

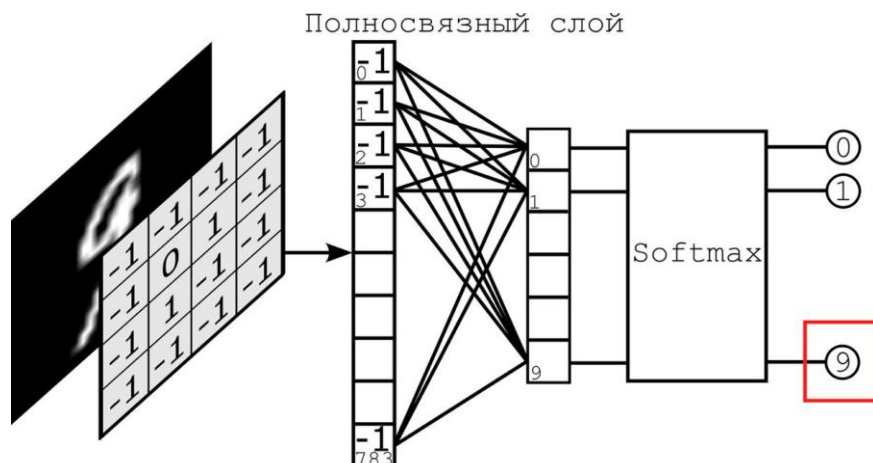


Рис. 1. Структура НС

Fig. 1. Structure of NN

Обучение НС выполнялось с использованием языка Python и библиотеки PyTorch. В процессе обучения НС использовался метод стохастического градиентного спуска (SGD). Этот метод выбирает случайные небольшие порции данных (батчи) на каждом шаге для обновления весов, что позволяет уменьшить количество вычислений по сравнению с использованием полного тренировочного набора данных на каждом шаге. Параметр скорости обучения был установлен равным  $\alpha = 0,003$ . Скорость обучения определяет, насколько сильно обновляются веса на каждом шаге. Введенный в алгоритм параметр импульса  $\gamma = 0,9$  позволяет ускорить сходимость и избежать застревания в локальных минимумах. Вместо того чтобы использовать только текущий градиент, импульс добавляет "инерцию" от предыдущих градиентов, что помогает сглаживать колебания и двигаться в направлении глобального минимума. Данные параметры описываются следующим выражением:

$$v_i = \gamma v_{i-1} + \alpha \nabla L(w_{i-1}), \quad (2)$$

где  $v_i$  – скорректированный градиент с учетом импульса,  $\nabla L(w_{i-1})$  – градиент функции потерь по весам,  $w_{i-1}$  – вес на предыдущем шаге.

Для обучения и тестирования входные данные необходимо привести к требуемому виду. Входные данные приводятся к диапазону  $[-1, 1]$  и устанавливается их среднеквадратическое отклонение (СКО) равным 0.5, что улучшает стабильность и скорость обучения нейронной сети. Это позволяет избежать проблем с затухающими и резко возрастающими градиентами, ускоряет сходимость и помогает модели лучше различать признаки, что в итоге повышает её точность и обобщающую способность. На рис. 2 представлен пример изображения в данном формате.

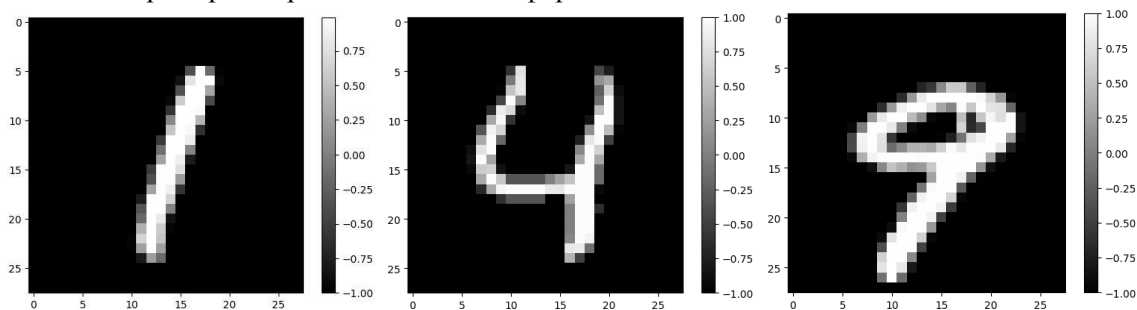


Рис. 2. Пример представления данных

Fig. 2. Example of data presentation

Также для тестирования модели необходимо подготовить набор тестовых изображений. Для этого происходит считывание каждого изображения, приведение к требуемому виду, разложение в строку и сохранение в результирующий массив NumPy. В результате полученный массив сохраняется и может быть использован для тестирования модели.

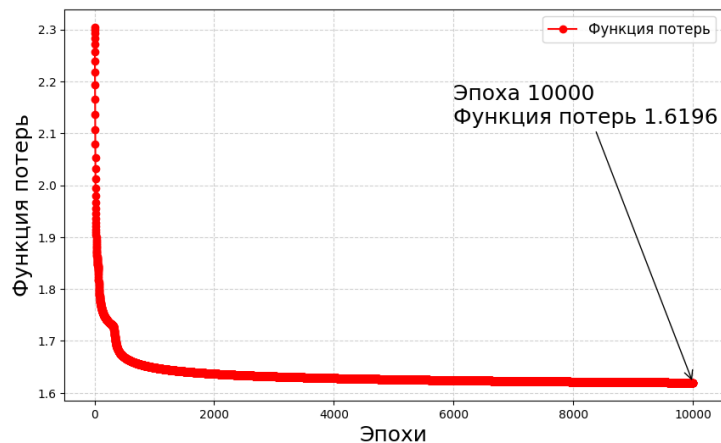


Рис. 3. Результаты обучения НС

Fig. 3. NN training results

Обучение выполнялось на 10 тыс. эпох, что позволило модели достичь высокой точности на тренировочном наборе данных. В результате обучения была получена матрица весовых коэффициентов размером  $10 \times 784$ .

### Реализация НС на FPGA

Для этого нужно просто отправить любую из 10 тыс. Строк этого массива на нейтронную сеть, позволяет с использованием языка Python взаимодействовать с аппаратными блоками FPGA, реализованными в виде IP-ядер, что делает процесс тестирования и разработки более гибким и удобным. Структура разработанной системы для распознавания рукописных цифр на базе платформы Zynq представлена на рис. 4. Для подачи изображения в НС оно предварительно считывается и передаётся в IP-блок по интерфейсу AXI-Lite. IP-блок, реализующий НС, описан на языке SystemVerilog.

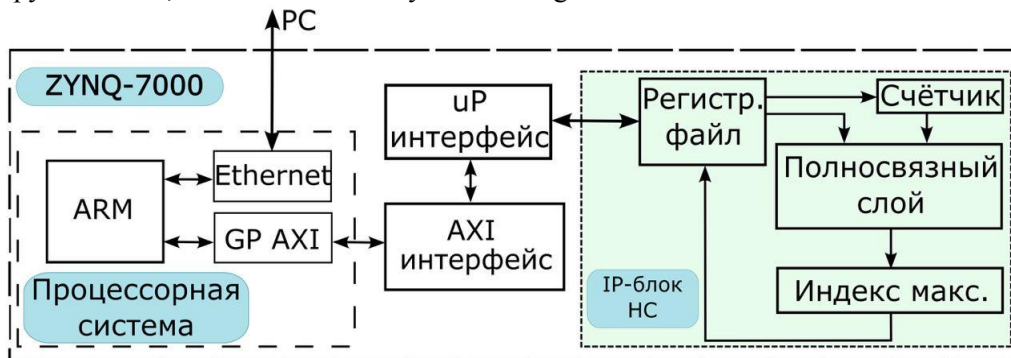


Рис. 4. Структурная реализация проекта

Fig. 4. Structural implementation of the project

Блок процессорной системы состоит из: двухъядерного процессора ARM Cortex-A9 для выполнения вычислений, Ethernet для связи с компьютером и GP AXI для соединения по AXI интерфейсам с другими блоками. Для соединения процессорной системы с самописным IP блоком используется AXI4-Lite интерфейс и uP-интерфейс. Так через AXI происходит подключение к процессорной системе, а интерфейс uP используется в качестве переходник между IP-блоком и AXI интерфейсом.

IP-блок состоит из нескольких компонент. Регистровый файл используется в качестве временного хранилища данных. При получении изображения или результата детектирования данные проходят через данный блок, который передает и получает данные из процессорной системы через uP и AXI4-Lite интерфейсы. Счётчик используется для передачи номера текущего пикселя в полносвязный слой. Полносвязный слой реализуется на базе десяти MAC-ядер. Каждое ядро производит 784 операции умножения значения пикселя на соответствующий

весовой коэффициент, хранящийся в памяти устройства. Таким образом формируется обработка изображения в соответствии со всеми весовыми классами. В результате получается массив из десяти элементов, представляющий выходные данные слоя. Далее полученный массив из 10 элементов поступает на вход блока поиска индекса максимального элемента. В данном блоке происходит сравнение всех входных значений и осуществляется выбор наибольшего элемента массива, индекс которого передаётся на выход в качестве определенного значения. Найденное число передаётся обратно в процессорную систему, проходя последовательно через регистровый файл, uP и AXI4-Lite интерфейсы. Структура данного блока показана на рис.5.

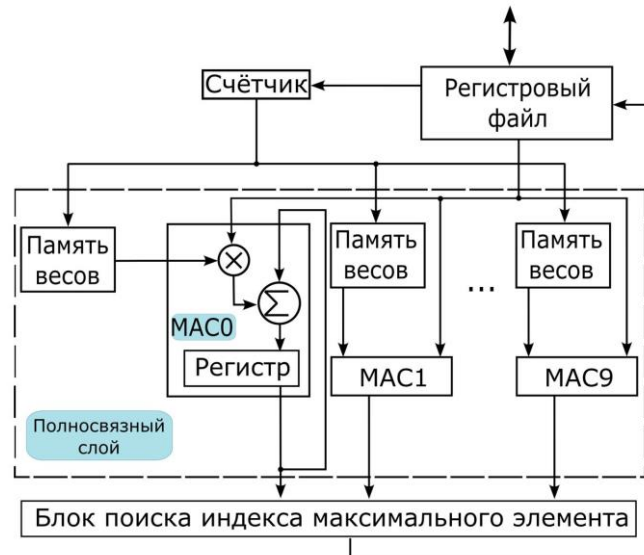


Рис. 5. Структура полносвязного слоя  
Fig. 5. The structure of a fully connected layer

### Экспериментальные исследования и выводы

На этапе тестирования исследовалось влияние разрядности весовых коэффициентов на точность распознавания цифр, а также на аппаратные затраты FPGA. Разрядность коэффициентов НС изменялась от 2 до 16 бит. Для каждой разрядности производилась подача на НС всех 10 тыс. тестовых изображений базы MNSIT. Для анализа полученных результатов выполнялось построение матрицы спутывания, которая показывает в процентном соотношении точность определения цифр. На рис. 6 представлен пример матрицы спутывания.

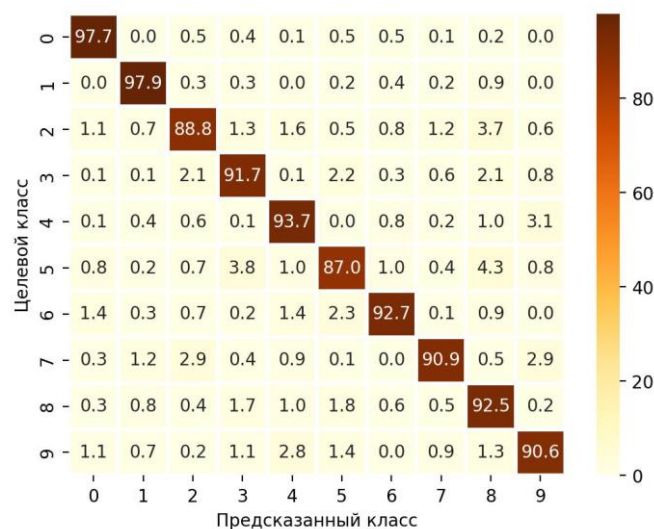
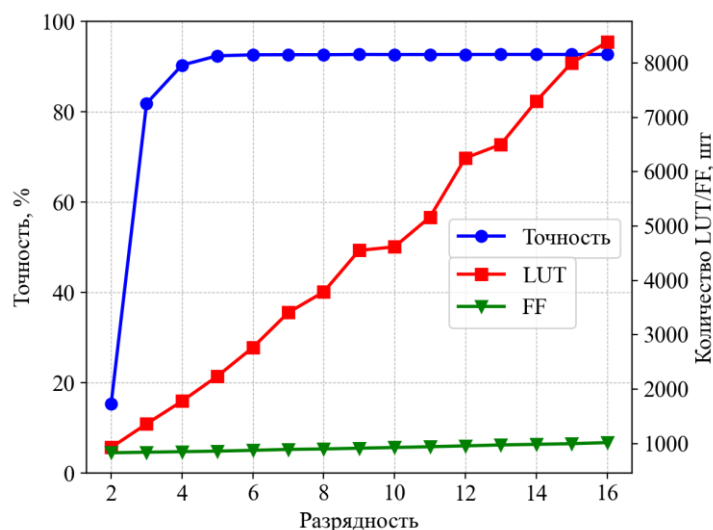


Рис. 6. Матрица спутывания для 5-разрядного представления весов НС  
Fig. 6. Confusion matrix for 5-bit representation of NN weights

Исследование аппаратных затрат осуществлялось на основе отчётов о размещении дизайна на FPGA, полученные в среде Xilinx Vivado. Анализ аппаратных затрат при различной разрядности весовых коэффициентов НС показал, что при уменьшении разрядности уменьшается число требуемых для реализации НС блоков LUT и FF (триггеров). Полученные результаты экспериментов представлены на рис. 7, где на одном графике совмещены точность распознавания и количество использованных элементов LUT и FF в зависимости от разрядности коэффициентов НС.



**Рис. 7.** Точность и аппаратные затраты на реализацию НС  
**Fig. 7.** Accuracy and hardware cost for implementing NN

Изменения остальных параметров нейронной сети не было замечено. Поэтому можно представить общие затраты для платы ZYBO Z7 из семейства FPGA Xilinx Zynq-7000 и 5-разрядного представления весов НС:

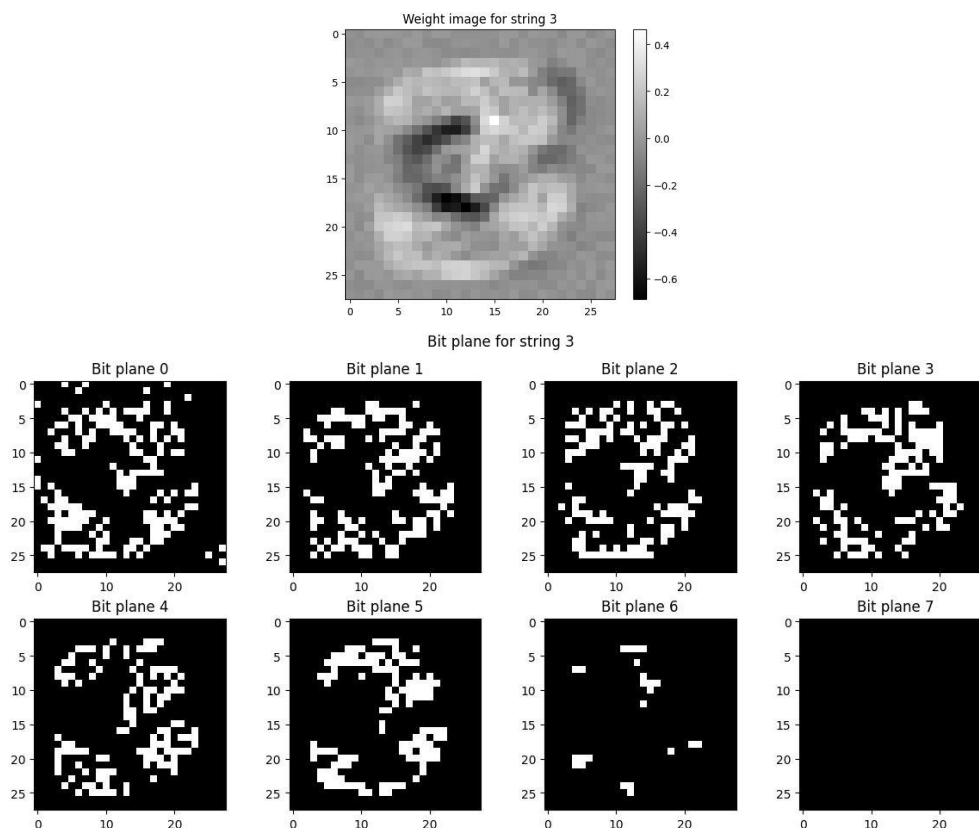
**Таблица 1.** Аппаратные затраты на реализацию НС на FPGA ZYBO Z7

**Table 1.** Hardware costs for implementation a NN on FPGA ZYBO Z7

Вариант блока Block variant	Количество Number of blocks	Доступно Available	В процентах Utility, %
LUT как логика / LUT as logic	2180	17600	12,39
LUT как память / LUT as memory	60	6000	1
Триггеры / Flip Flop	862	35200	2,45
Блочная память / BRAM18	10	120	8,33

Данный эксперимент можно подробнее проанализировать используя разложение на битовые плоскости. Для этого необходимо взять каждую строку из весовой матрицы. Собрать из неё изображение 28 на 28 и разложить его на плоскости. В результате разложения видно, что основная информация об изображении находится в 6 первых битовых плоскостях, что говорит нам о том, что более высокая точность весовых коэффициентов будет являться избыточной так как она несёт минимальный объем информации. На рис. 8 представлено разложение на битовые плоскости весового класса для цифры 3, а также её общее представление.





**Рис. 8.** Разложение весового ряда для цифры 3 на битовые плоскости  
**Fig. 8.** Decomposition of the weight series for the number 3 into bit planes

В результате мы получаем, что вся информация находится в первых 7 битовых плоскостях. Но так как точность определения цифр изменяется минимально для 5 и 6 битового представления весовых коэффициентов, что показано на рис. 6, следовательно, можно сделать вывод, что наиболее оптимальной разрядность будет 5 бит, которая позволяет с высокой вероятностью правильно распознать цифры на изображении и не использовать избыточные аппаратные ресурсы FPGA.

### Заключение

В работе исследовано влияние разрядности весовых коэффициентов в НС прямого распространения для распознавания рукописных цифр на точность детектирования цифр по средствам аппаратной реализации НС на FPGA и последующего её тестирования. А также использовано разложение весовых классов на битовые плоскости, чтобы наглядно продемонстрировать объемы информации хранящиеся на каждом слое. Соответствующий анализ позволяет сделать вывод о том, что 7 и последующие битовые плоскости не несут полезной информации, а на 6 плоскости объем данных минимален, что не существенно влияет на точность. Дана оценка эффективности реализации НС на FPGA. С этой целью проведены исследования аппаратных затрат. Для этого программным образом на языке Python выполнено построение графиков с затратами основных блоков FPGA на основе отчётов из среды Xilinx Vivado, а также приведена таблица затрат основных функциональных блоков. Согласно результатам экспериментов, предлагается использовать 5 бит для представления весовых коэффициентов НС при реализации на базе FPGA.

### Список литературы

1. Mittal S. A survey of FPGA-based accelerators for convolutional neural networks // Neural computing and applications. – 2020. – Т. 32. – No. 4. – P. 1109-1139

2. Ahmad A., Pasha M. A. FFConv: an FPGA-based accelerator for fast convolution layers in convolutional neural networks // ACM Transactions on Embedded Computing Systems (TECS). – 2020. – Т. 19. – No. 2. – P. 1-24.
3. Giardino D. et al. FPGA implementation of hand- written number recognition based on CNN // International Journal on Advanced Science, Engineering and Information Technology. – 2019. – Т. 9. – No. 1. – P. 167-171.
4. The MNIST database of handwritten digits [Электронный ресурс] – Электронные данные – Режим доступа: <https://yann.lecun.com/exdb/mnist/>

### References

1. S. A survey of FPGA-based accelerators for convolutional neural networks // Neural computing and applications. – 2020. – Т. 32. – No. 4. – P. 1109-1139
2. Ahmad A., Pasha M. A. FFConv: an FPGA-based accelerator for fast convolution layers in convolutional neural networks // ACM Transactions on Embedded Computing Systems (TECS). – 2020. – Т. 19. – No. 2. – P. 1-24.
3. Giardino D. et al. FPGA implementation of hand- written number recognition based on CNN // International Journal on Advanced Science, Engineering and Information Technology. – 2019. – Т. 9. – No. 1. – P. 167-171.
4. MNIST Handwritten Numbers database [Electronic resource] – Electronic data – Access mode: <https://yann.lecun.com/exdb/mnist/Mittal>

### Вклад авторов

Кривальцевич Е.А. реализовал и обучил НС, аппаратно реализовал структуру НС, а также провел экспериментальные исследования.

Вашкевич М.И. определил задачи, которые необходимо было решить в ходе проведения исследований, принимал участие в аппаратной реализации НС и тестировании на FPGA, участвовал в проведении экспериментальных исследований и интерпретации результатов эксперимента.

### Authors contribution

Krivaltsevich E.A. implemented and trained the NN, implemented the NN structure in hardware, and conducted experimental studies.

Vashkevich M.I. defined the tasks that needed to be solved during the research, participated in the hardware implementation of the NN and testing on FPGA, participated in conducting experimental studies and interpreting the experimental results.

### Сведения об авторах

Кривальцевич Е.А., студент 4 курса по специальности электронные вычислительных средства, кафедра электронных вычислительных средств, Белорусский государственный университет информатики и радиоэлектроники.

Вашкевич М.И. доктор техн. наук, профессор кафедры электронных вычислительных средств, Белорусский государственный университет информатики и радиоэлектроники.

### Information about the authors

Krivaltsevich E.A., 4th year student, specializing in electronic computing, Department of Electronic Computing, Belarusian State University of Informatics and Radioelectronics.

Vashkevich M.I. Dr. of Sci. (Tech.), Professor at the Electronic Computing Facilities Department, Belarusian State University of Informatics and Radioelectronics.

### Адрес для корреспонденции

### Address for correspondence



220013, Республика Беларусь,  
г. Минск, ул. П. Бровки, 6  
Белорусский государственный университет  
информатики и радиоэлектроники  
Тел.: +375 (17) 293-84-20  
E-mail: [porhun@bsuir.by](mailto:porhun@bsuir.by)  
Порхун Максим Игоревич

220013, Republic of Belarus,  
Minsk, P. Brovki St., 6  
Belarusian State University  
of Informatics and Radioelectronics  
Tel.: +375 (17) 293-84-20  
E-mail: [porhun@bsuir.by](mailto:porhun@bsuir.by)  
Porhun Maxim Igorevich