Оригинальная статья / Original article

УДК 625.7:004.9

https://doi.org/10.21869/2223-1560-2025-29-4-70-92

Выбор разрядности компонентов нелинейного нейрона при реализации на ПЛИС

О.Г. Бондарь [1] ✉, Е.О. Брежнева [1], Д.А. Голубев [1]

[1] Юго-Западный государственный университет
ул. 50 лет Октября, д. 94, г. Курск 305040, Российская Федерация

✉ e-mail: b.og@mail.ru

Резюме

***Цель работы**: исследование зависимости между погрешностью данных на входе нейрона, предназначенного для применения в искусственной нейронной сети на ПЛИС, и погрешностью вычислений, а также разработка методики выбора разрядности компонентов нейрона, направленной на снижение аппаратных затрат при сохранении точности вычислений, адекватной точности исходных данных.*

***Методы.** В работе использовались методы проектирования цифровых устройств на основе языка описания VHDL, анализа погрешностей вычислений относительно эталонной модели с плавающей точкой, а также методы синтеза устройств и оценки используемых аппаратных ресурсов ПЛИС встроенные в Xilinx ISE. Для обработки результатов применялись методы математической статистики, включая построение регрессионных моделей зависимости точности и аппаратных затрат от разрядности исходных данных.*

***Результаты.** Предложен вариант оценки разрядности устройства обработки, позволяющий согласовать его разрядность с погрешностью исходных данных, исследовано влияние разрядности представления входных данных и весовых коэффициентов на точность вычислений и объём занимаемых нейроном аппаратных ресурсов, реализованном на ПЛИС. На основе VHDL-описания устройства создана параметризуемая модель, позволяющая согласованно изменять разрядность элементов нейрона при изменении разрядности входных сигналов. Для оценки влияния разрядности на точность вычислений использовалась эталонная модель на основе арифметики с плавающей точкой. Для каждого варианта разрядности проводились сравнительные вычисления выходного значения устройства, и рассчитывалась погрешность. Также анализировалось влияние разрядности на использование аппаратных ресурсов ПЛИС: количество LUT, регистров (FF). Апробация метода проводилась на базе ПЛИС Xilinx Spartan-3E XC3S500E (xc3s50e-4pq208), с использованием среды ISE Design Suite 14.7. Были реализованы несколько версий цифрового устройства с разрядностью входных данных от 4 до 12 бит (с учётом знакового разряда). Для каждого случая зафиксированы: тактовая частота работы, используемые ресурсы ПЛИС, точность измерений. На примере 12-битных исходных данных получена экспериментальная оценка объёма таблицы сигмоидальной функции (8192 ячеек), позволяющей достичь компромисса между точностью вычислений (максимальная приведенная погрешность – 0,12%) и объёмом аппаратных затрат (используется 1% аппаратных ресурсов ПЛИС).*

***Заключение.** В данной работе представлено описание схемы нейрона с сигмоидальной функцией активации, реализованной на языке описания аппаратуры VHDL, пригодной для интеграции в нейросетевые решения на программируемых логических интегральных схемах. Устройство принимает входные целочисленные значения фиксированной разрядности со знаком, осуществляет вычисление суммы взвешенных входных сигналов и смещения и формирует выход нейрона на основе таблицы поиска, хранящейся в блочной памяти (RAM). Приведено описание работы модуля, его масштабирование и оптимизация. Предложенный метод позволяет определить оптимальную разрядность устройства обработки, обеспечивающий согласованный с погрешностью исходных данных уровень погрешности при минимальных аппаратных затратах. Полученные зависимости могут быть использованы на этапе проектирования для выбора параметров цифровых модулей обработки информации в системах реального времени и встраиваемых устройствах.*

Ключевые слова: *цифровая обработка; искусственный нейрон; вычисления с фиксированной точкой; программируемая логическая интегральная схема (ПЛИС); VHDL; разрядность данных; аппаратная реализация; функция активации; таблица активации.*

Конфликт интересов: *Авторы декларируют отсутствие явных и потенциальных конфликтов интересов, связанных с публикацией настоящей статьи.*

Для цитирования: Бондарь О.Г., Брежнева Е.О., Голубев Д.А. Выбор разрядности компонентов нелинейного нейрона при реализации на ПЛИС // *И*звестия Юго-Западного государственного университета. 2025; 29(4): 70-92. https://doi.org/10.21869/2223-1560-2025-29-4-70-92.

Поступила в редакцию 09.09.2025 *Подписана в печать 03.10.2025* *Опубликована 22.12.2025*

Choice of component bit width for nonlinear neuron implementation on FPGA

Oleg G. Bondar [1] ✉, Ekaterina O. Brezhneva [1], Dmitry A. Golubev [1]

[1] Southwest State University
50 Let Oktyabrya str. 94, Kursk 305040, Russian Federation

✉ e-mail: b.og@mail.ru

Abstract

Purpose. *Investigation of the relationship between input data error of a neuron intended for use in an artificial neural network implemented on FPGA, and computational error, as well as development of a methodology for selecting the bit width of neuron components aimed at reducing hardware costs while maintaining computational accuracy consistent with the accuracy of the input data.*

Methods. *The study employed methods of digital circuit design based on the VHDL hardware description language, error analysis of computations relative to a floating-point reference model, as well as device synthesis and FPGA resource utilization estimation methods integrated into Xilinx ISE. Mathematical statistics techniques, including the construction of regression models describing the dependence of accuracy and hardware costs on input data bit width, were applied to process the experimental results.*

Results. *A method has been proposed for estimating the bit width of the processing unit, enabling its precision to be matched with the inherent error level of the input data. The impact of the bit width of input data and weight coefficients on computational accuracy and the amount of FPGA hardware resources consumed by the implemented neuron was investigated. Based on the VHDL description of the device, a parameterized model was developed that enables coordinated adjustment of the neuron's internal component bit widths as the bit width of input signals is varied. To assess the effect of bit width on computational accuracy, a floating-point-based reference model was used. For each bit-width configuration, comparative computations of the device's output were performed, and the resulting error was quantified. The influence of bit width on FPGA resource utilization — specifically the number of LUTs and flip-flops (FFs) — was also analyzed. The proposed methodology was validated on the Xilinx Spartan-3E XC3S500E (xc3s500e-4pq208) FPGA platform using the ISE Design Suite 14.7 environment. Multiple versions of the digital neuron were implemented, with input data bit widths ranging from 4 to 12 bits (including the sign bit). For each variant, the operating clock frequency, utilized FPGA resources, and computational accuracy were recorded.*

As a case study using 12-bit input data, an experimental evaluation determined that a sigmoid function lookup table with 8,192 entries achieves an optimal trade-off between computational accuracy (maximum relative error — 0.12%) and hardware cost (occupying only 1% of the FPGA's available resources).

Conclusion. *This paper presents a description of a neuron circuit with a sigmoid activation function, implemented in the VHDL hardware description language and suitable for integration into neural network solutions on Field-Programmable Gate Arrays (FPGAs). The device accepts signed integer input values of fixed bit width, computes the weighted sum of inputs and bias, and generates the neuron's output using a precomputed lookup table stored in block RAM. The operation, scaling, and optimization of the module are described in detail.*

The proposed method enables determination of the optimal bit width for the processing unit, ensuring that computational error remains consistent with the error level of the input data while minimizing hardware resource consumption. The obtained relationships can be utilized during the design phase to select parameters for digital processing modules in real-time systems and embedded devices.

Keywords: *digital processing;, artificial neuron; fixed-point arithmetic;, Field-Programmable Gate Array (FPGA); VHDL;, data bit width;, hardware implementation; activation function;, lookup table.*

Conflict of interest. *The Authors declare the absence of obvious and potential conflicts of interest related to the publication of this article.*

For citation: Bondar O. G., Brezhnev E. O., Golubev D. A. Choice of component bit width for nonlinear neuron implementation on FPGA. *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta = Proceedings of the South-west State University*. 2025; 29(4): 70-92 (In Russ.). https://doi.org/ 10.21869/2223-1560-2025-29-4-70-92.

Received 09.09.2025 *Accepted 03.10.2025* *Published 22.12.2025*

***

Введение

Использование программно-аппаратных ускорителей на базе программируемых логических интегральных схем (ПЛИС) становится всё более актуальным при решении задач машинного обучения, особенно в условиях ограниченных ресурсов и необходимости высокой производительности на единицу энергопотребления [1]. Одним из ключевых элементов таких систем является искусственный нейрон – базовая вычислительная единица, реализующая операции скалярного произведения и применения функции активации. Актуальность аппаратной реализации ИНС на ПЛИС подчеркивается конкретными применениями, например, эффективной и экономичной реализацией нейронной сети на ПЛИС для цифрового предыскажения (DPD) в многоканальных (MIMO) системах связи, где критически важны одновременно высокая пропускная способность, малое время отклика и рациональное использование аппаратных ресурсов [2]. Такая реализация позволяет решать практические инженерные задачи, такие как линеаризация усилителей мощности в базовых станциях сотовых сетей, что требует высокой скорости обработки и эффективного использования доступных ресурсов ПЛИС.

Дальнейшее расширение областей применения аппаратных ускорителей на ПЛИС включает даже экстремальные условия, например, космические миссии, где ценятся такие характеристики ПЛИС, как радиационная стойкость, энергоэффективность и гибкость. Как показывает обзор, ПЛИС рассматриваются как перспективная платформа для реализации нейронных сетей, способных выполнять задачи автономных операций, анализа данных сенсоров и сжатия данных непосредственно на борту космических аппаратов [3]. Анализ существующих

подходов, представленный в работах, подобных обзору [4], подтверждает растущий интерес к оптимизации архитектур и методов реализации нейронных сетей на ПЛИС, направленных на достижение максимальной эффективности и энергоэкономичности для разнообразных приложений.

Реализация нейронных сетей и других нелинейных устройств цифровой обработки на ПЛИС является одной из актуальных задач в области встраиваемых систем и распределённых вычислений (эдж-компьютинг) [5]. Искусственный нейрон может быть реализован различными способами: от простых сумматоров и умножителей до сложных параллельных архитектур с конвейерной обработкой. Среди известных решений можно выделить работы, в которых используются VHDL/Verilog для проектирования параметризуемых модулей, поддерживающих различные типы активации и форматы представления чисел [6]. Анализ существующих подходов к аппаратной реализации, включая обзоры открытых инструментов [7], подчеркивает важность оптимизации базового вычислительного элемента – отдельного искусственного нейрона. Повышение точности вычислений на уровне отдельного нейрона, например, путем выбора подходящей функции активации, оптимизации разрядности представления данных или минимизации ошибок округления, является критически важным для обеспечения общей точности и надежности работы всей нейронной сети.

Функция активации является одним из самых ресурсоёмких этапов вычисления нейрона. Одним из эффективных способов её реализации является использование внешней таблицы (LUT), хранящейся в блочной памяти ПЛИС. В работах [1,8] представлено описание цифрового нейрона с использованием блочной памяти для реализации функции активации. Такой подход позволяет исключить дорогостоящие операции возведения в степень или деления, характерные для сигмоидальной функции, за счёт предварительного расчёта таблицы значений. Это делает возможным снижение аппаратных затрат и задержки вычисления функции активации.

Как отмечается в работе [9], реализация традиционных функций активации, таких как гиперболический тангенс и сигмоида, на ПЛИС может быть ресурсоемкой из-за необходимости выполнения сложных операций, таких как возведение в степень (exp) и деление (1/x), что может приводить к высокой задержке. В этой же работе рассматривается применение блочной памяти (LUT) для аппроксимации функции гиперболического тангенса. Такой подход позволяет обойти сложные вычисления, снижая аппаратные затраты и уменьшая вычислительную задержку.

Одним из ключевых направлений снижения аппаратных затрат является уменьшение разрядности используемых данных [10,11]. Многие исследования показывают, что переход от 32-битных чисел с плавающей точкой к 8- или да-

же 4-битным числам с фиксированной точкой позволяет значительно сократить использование DSP-блоков и регистров без существенной потери точности модели [12,13]. В [14] показано, что снижение разрядности может значительно уменьшить использование ресурсов ПЛИС (например, уменьшение LUT на более чем 40% при переходе с 8 бит до 4 бит) с минимальной потерей чувствительности. Актуальность точного и эффективного квантования подчеркивается необходимостью обеспечения гарантий точности, особенно в задачах регрессии и критически важных приложениях. В работе [15] представлена методология и программный инструментарий (Aster), которые позволяют автоматически определять оптимальное распределение разрядности (назначать смешанную точность) для представления чисел с фиксированной точкой в нейронной сети. Инструмент строго гарантирует, что совокупная ошибка округления на выходе сети не превысит заданную пользователем границу. Это особенно важно для применения нейронных сетей в системах управления и других задачах, где точность вычислений критична. Дальнейшее развитие методов квантования, включая разработку алгоритмов с настраиваемой точностью, таких как описанный в работе [16], где представлен алгоритм квантования с фиксированной точкой и регулируемой точностью для сверточных нейронных сетей, направлено на оптимизацию потока данных внутри сети и повышение эффективности использования ресурсов FPGA без ущерба для производительности.

Для количественной оценки влияния разрядности на точность вычислений применяются методы сравнения результатов работы с фиксированной точкой с эталонными значениями, полученными с помощью вычислений с плавающей точкой [13]. При этом проводится статистическая обработка ошибок – вычисление средней абсолютной и относительной погрешностей, дисперсии и доверительных интервалов.

Анализ существующих подходов к аппаратной реализации нейронных сетей на ПЛИС показывает, что важными особенностями ПЛИС являются параллелизм и конвейеризация [17]. Благодаря параллелизму можно многократно распределять и вычислять ресурсы, когда несколько модулей могут работать независимо, одновременно. Конвейеризация делает аппаратные ресурсы многоразовыми, что может значительно улучшить параллельную производительность. Однако, как отмечается в ряде работ, в условиях ограниченных ресурсов ПЛИС эти подходы могут быть недоступны или менее эффективны [8]. Например, в [18] для повышения вычислительной эффективности и минимизации использования ресурсов логических элементов (LUTs) и триггеров (flip-flops) применяются методы, такие как введение конвейерных регистров между промежуточными операциями и совместное использование арифметических операций (сложение, сдвиг) для разных вычислений нейронов.

Эффективность аппаратных ускорителей на ПЛИС подтверждается многочисленными исследованиями, демонстрирующими их превосходство в энергоэффективности и задержке по сравнению с традиционными процессорами и графическими ускорителями (GPU) для целого ряда задач ИНС, особенно в условиях ограниченных ресурсов [19]. Гибкость архитектуры ПЛИС позволяет адаптировать аппаратную реализацию под конкретную модель и сценарий использования, что делает их привлекательным выбором для разработки специализированных решений в области машинного обучения.

Таким образом, наряду с квантованием и оптимизацией функций активации, методы структурной оптимизации, такие как конвейеризация и параллелизм, играют важную роль в разработке эффективных аппаратных ускорителей на ПЛИС. Однако, как показывает практика [15, 16], в условиях ограниченных ресурсов поиск оптимального баланса между производительностью, точностью и аппаратной сложностью требует комплексного подхода, включающего все указанные методы.

В работах [20,21] предлагаются автоматизированные методы определения оптимальной разрядности, основанные на машинном обучении и статистическом анализе. В [20] описан метод автоматизированного гетерогенного квантования, который оптимизирует разрядность отдельных слоев или операций глубокой нейронной сети. Этот подход позволяет находить баланс между точностью модели и задержкой, используя методы анализа чувствительности и оптимизации. В работе [21] авторы охватывают широкий спектр методов квантования, включая адаптивные и автоматизированные подходы, направленные на поиск оптимальной конфигурации разрядности для минимизации потерь точности при снижении вычислительной сложности и объема памяти. Такие автоматизированные методы позволяют эффективно снижать аппаратные затраты без существенной потери качества модели.

Использованный авторами статьи подход основывается на использовании информации о погрешности исходных данных, а гибкая и параметризуемая модель нелинейного единичного нейрона позволяет: исследовать влияние разрядности устройств обработки данных на погрешность вычислений, оценить объём используемых аппаратных средств конкретной ПЛИС, использовать полученные данные для обоснованного выбора разрядности на этапе проектирования.

Точность обработки данных, объем аппаратных средств и производительность – конкурирующие характеристики при построении цифровых устройств обработки данных. Известным способом повышения быстродействия и упрощения устройств является применение целочисленной математики. При этом ограничение разрядности вычислителей оправданно ограниченной точностью исходных данных. В работе пока-

зана взаимосвязь разрядности компонентов искусственного нейрона образующих каскадную структуру. Демонстрация подхода осуществляется на примере ПЛИС Xilinx Spartan-3E XC3S500E, и иллюстрирует его эффективность в условиях жестких ограничений аппаратных ресурсов.

Приведены выражения, позволяющие произвести оценку разрядности ступеней вычислителя, результаты оценки аппаратных затрат и погрешности вычислений при разной разрядности исходных данных, подтверждённые моделированием.

Материалы и методы

Математическая основа метода базируется на формуле скалярного произведения с последующим добавлением смещения и нелинейным преобразованием через функцию активации:

$$y = f(\sum_{i=1}^{N} x_i w_i + b),$$

где x_i – входные значения; w_i – весовые коэффициенты; b – смещение; $f()$ – функция активации (в данном случае сигмоида), реализованная через таблицу, хранящуюся в блочной памяти ПЛИС.

В данной статье представлена VHDL-модель нелинейного нейрона, поддерживающая:

– обработку нескольких входных сигналов;

– учет весовых коэффициентов и смещения;

– нелинейное преобразование выхода с использованием табличного представления сигмоидальной функции;

– настройку разрядности входных сигналов, весовых коэффициентов и промежуточных результатов.

Структурная схема устройства представлена на рис. 1.

Она включает следующие основные компоненты:

• [k*INPUT_WIDTH] – вектор входных сигналов фиксированной разрядности;

• [k*WEIGHT_WIDTH]–- вектор весовых коэффициентов;

• Multiplier Array – массив умножителей для вычисления произведений входных сигналов и весов и форматирования результата (отсечение младших разрядов и конкатенация для исключения последующего переполнения);

• Summation Unit – сумматор произведений и смещения;

• Address Generator – преобразует сумму в адрес для таблицы активации;

• Activation Table – хранит предварительно вычисленные значения сигмоидальной функции активации для положительных значений аргумента;

• Output value Generator – формирует значение сигмоидальной функции и преобразование беззнакового целого в знаковое.

Модель нейрона параметризуема и может быть адаптирована под различные задачи за счет изменения параметров общего назначения: количества входов нейрона (INPUT_SIZE), разрядности входных сигналов (INPUT_WIDTH), разрядности весовых коэффициентов (WEIGHT_WIDTH).

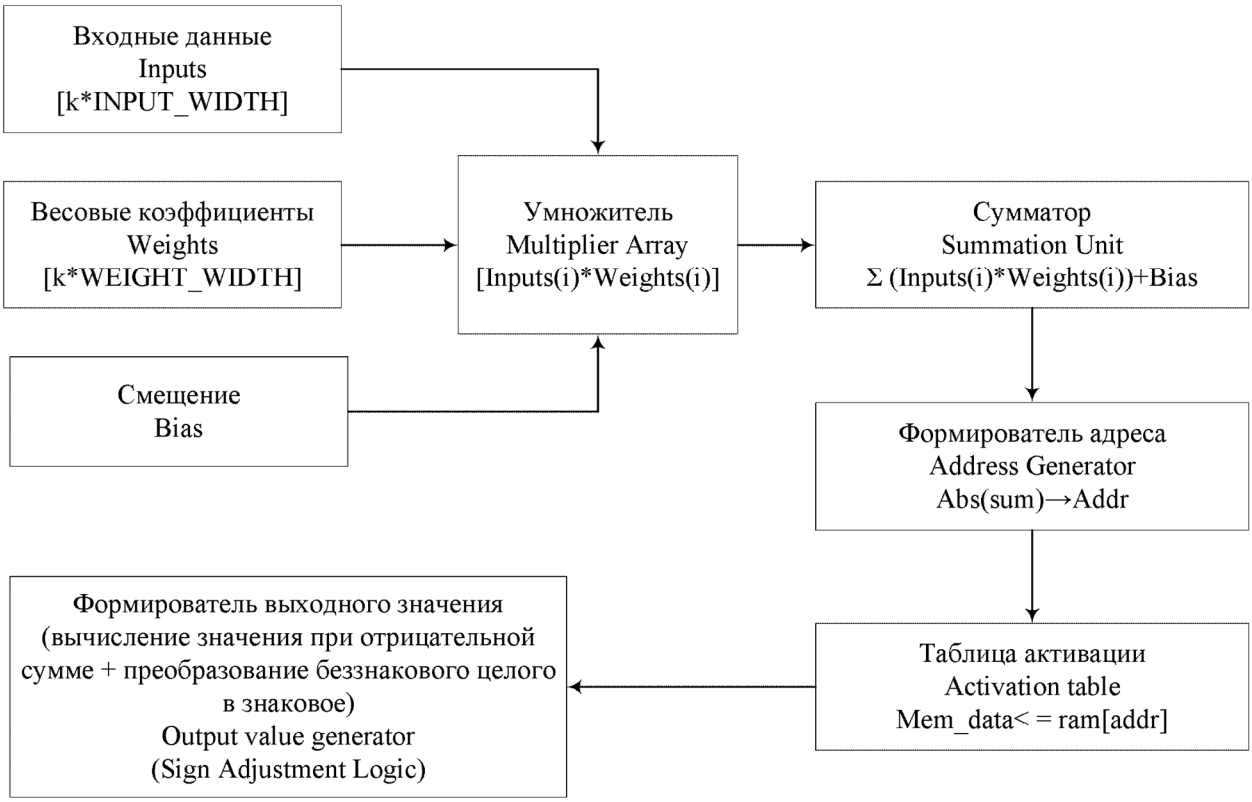

Рис. 1. Структурная схема нелинейного нейрона на ПЛИС

Fig. 1. Structural diagram of a nonlinear neuron implemented on an FPGA

Функционирование устройства:

1. Умножение входов на веса. Для каждого входа выполняется операция умножения соответствующего значения на связанный весовой коэффициент. Результат умножения хранится во временном массиве. Для снижения разрядности результата умножения используется усечение младших разрядов произведения с сохранением знака.

2. Суммирование результатов. После умножения все произведения суммируются с учетом знака и переполнения, смещение предварительно нормируется.

3. Формирование адреса для таблицы активации. В качестве адреса RAM, хранящей таблицу функции активации, принимается абсолютное значение суммы. При этом используется свойство симметрии сигмоидальной функции, что уменьшает объём памяти, необходимой для хранения таблицы в два раза.

Значения функции активации для положительных значений аргумента вычисляются заранее и из внешнего файла загружаются в блочную память ПЛИС. Это позволяет избежать сложных вычислений в реальном времени и повысить производительность системы.

4. Верификация и оптимизация. Для целей анализа и тестирования предусмотрены отладочные сигналы.

Для реализации максимального быстродействия и с учётом ограниченных ресурсов ПЛИС все вычисления осуществляются в формате с фиксированной точкой.

Разработанная VHDL- модель может быть адаптирована под параллельную обработку с использованием конвейера для ускорения выполнения операций, допускает простую замену функций активации изменением содержимого таблицы в блочной RAM и заменой формирователя выходного значения.

В коде предусмотрены отладочные сигналы, которые позволяют наблюдать промежуточные значения (например, произведения входных данных на весовые коэффициенты), что облегчает тестирование и поиск ошибок.

Для каждого варианта разрядности входных данных и весов (3-7-9-11), соответствующих дифференциальным режимам распространённых 8-10-12 разрядных АЦП, производился синтез VHDL-описания устройства в среде ISE Design Suite 14.7. Фиксировались метрики использования аппаратных ресурсов ПЛИС: количество LUT (программируемая таблица истинности), количество регистров (FF), определялась тактовая частота после размещения конфигурируемых логических блоков и трассировки схемы. Таблицы LUT и регистры FF — два базовых ресурса ПЛИС, которые используются для реализации любой цифровой логики. Именно поэтому они считаются ключевыми метриками при сравнении аппаратных затрат. Эти параметры позволяют оценить зависимость «разрядность – точность – аппаратные затраты» [2].

Набор входных сигналов, весов и смещений генерировался случайным образом в диапазоне 0-1. Для сгенерированных значений одновременно вычислялись рефренсные значения выходных сигналов с использованием арифметики с плавающей запятой. Для ПЛИС входные значения преобразовывались к целочисленным, требуемой разрядности предварительным масштабированием (умножением исходного числа на 2n-1, где n – разрядность данных без знака) и переводом в двоичную систему счисления.

Выходной сигнал VHDL- модели нейрона делился на 2^l-1, где l – разрядность значений функции активации нейрона. Полученные значения (y_{VHDL}) сопоставлялись с референсными значениями (y_{float}). Рассчитывалась погрешность, приведенная к максимальному значению:

$$\delta = \frac{y_{VHDL} - y_{float}}{y_{\max}} \cdot 100\% .$$

Для моделирования и отладки устройства использовался интегрированный с ISE Design Suite симулятор ISim. На основе полученных данных строились линейные регрессионные зависимости между разрядностью и приведенной погрешностью вычислений.

Методика определения разрядности компонентов нейрона. Предлагаемый вариант оценки разрядности компонентов модели нейрона ориентирован на обработку измерительной информации искусственной нейронной сетью и базируется на том, что разрешающая способность измерительных устройств согласуется с их погрешностью. Исходя из этого, целочисленный нормирующий

множитель для преобразования предварительно нормированных входных данных, представленных в диапазоне [0, 1], выбирается так, чтобы единица младшего разряда соответствовала приведённой погрешности измерений. Сигмоидальная функция активации рассчитывается по следующей формуле:

$$y = \frac{1}{1+e^{-s}},$$

где $S = \sum_1^k x_i \cdot w_i + bias$; k – число входных параметров нейрона.

Оценим максимально допустимый шаг табличного представления сигмоидальной функции, обеспечивающий разрешение функции, соответствующее разрешению входных данных. Максимальное значение производной от сигмоидальной функции достигается при значении аргумента $s = 0$ и равно:

$$\left(\frac{dy}{ds}\right)_{\max} = \left.\frac{e^{-s}}{\left(1+e^{-s}\right)^2}\right|_{s=0} = \frac{1}{4}.$$

При этом допустимое абсолютное приращение функции при максимальном шаге аргумента $\Delta S_{\max}$:

$$\Delta y_{\max} = \frac{\Delta s_{\max}}{4} = \frac{1}{2^n},$$

где n – разрядность нормированных исходных данных в двоичной системе счисления. Отсюда можно определить максимально допустимый шаг аргумента при табулировании функции:

$$\Delta s_{\max} = 4\Delta y_{\max} = \frac{4}{2^n} = \frac{1}{2^{n-2}}.$$

Выразим переменную s в уравнении сигмоидальной функции через целочисленные значения нового аргумента, при этом новый аргумент интерпретируется как номер строки, табулированной функции:

$$s = \frac{x}{2^{n-2}}.$$

Объём таблицы следует ограничить максимальным значением аргумента, при котором значение функции отличается от 1 меньше, чем на величину абсолютной погрешности (которая соответствует разрядности исходного представления входных данных):

$$y_{\max} = \frac{1}{1+e^{-\frac{x_{\max}}{2^{n-2}}}} \geq 1 - \frac{1}{2^n}.$$

Отсюда следует:

$$e^{-\frac{x_{\max}}{2^{n-2}}} < \frac{1}{2^n},$$

а $x_{\max}$ определяется из:

$$x_{\max} > 2^{n-2}\, n \ln 2 = 0{,}693147 \cdot 2^{n-2} \cdot n, \qquad (1)$$

где $x_{\max}$ – это максимальное значение алгебраической суммы взвешенных входных данных и смещения, которому ставится в соответствие максимальное значение аргумента, и при котором значение $y_{\max}$ отличается от единицы не более чем на $1/2^n$.

Альтернативно максимальное значение аргумента может выбираться в соответствии с выражением

$$s_{\max} = \frac{x_{\max}}{2^{n-2}} \approx 8 \div 9;$$

$$x_{\max} = (8 \div 9) \cdot 2^{n-2}. \qquad (2)$$

Следует учитывать, что максимальное значение аргумента сигмоидальной функции должно быть достижимо, потому максимальное значение взвешенных сумм и смещения должно быть гарантировано равно, или больше $x_{\max}$.

Максимальное значение $x_{\max}$ определяется разрядностью входных данных n и их количеством:

$$x_{\max} = (2^n - 1)(k+1) \approx 2^{n-2} 2^{2+\log_2(k+1)},$$

где k – количество входных параметров, а единица учитывает смещение.

Оно достигается при $k > 1$. При значениях $x_{\max}$, превышающих условия (1) или (2), значение функции принимается равным максимальному.

Поскольку количество входов сумматора определяется количеством параметров, влияющих на значение функции, то для исключения переполнения разрядной сетки сумматор должен иметь количество дополнительных разрядов

$$m = \lceil \log_2(k+1) \rceil.$$

Таблица сигмоидальной функции заполняется целочисленными масштабированными значениями.

$$Y(|x|) = (2^n - 1) y(|x|).$$

В силу симметрии сигмоидной функции относительно точки (0, 0.5) её значения при отрицательном значении аргумента определяются как:

$$y(-s) = 1 - y(|x|).$$

Поэтому

$$Y(-x) = Y\max - Y(|x|) = (2^n - 1) - Y(|x|). \quad (3)$$

Предлагаемый вариант оценки оптимального объема памяти (разрядности адреса) для табличной реализации сигмоидальной функции с учетом разрядности входных данных исследован экспериментально.

Результаты и их обсуждение

В табл. 1 представлены результаты исследования зависимости приведённой погрешности (δ) модели нелинейного нейрона и аппаратных затрат в виде таблиц поиска (LUT) и триггеров (FF) от разрядности исходных данных. Следует отметить, что при анализе аппаратных затрат учитывалось наличие дополнительных отладочных сигналов в коде, что приводит к повышенному расходу ресурсов, но не влияет на закономерность зависимости ресурсопотребления от разрядности данных. При максимальной разрядности исходных данных процент затрачиваемого оборудования от общих ресурсов ПЛИС составляет всего 1%, что особенно важно для встраиваемых устройств и встроенных вычислительных систем с ограниченными возможностями.

Таблица 1. Зависимость погрешности и аппаратных ресурсов от разрядности входных данных

Table 1. Dependence of error and hardware resources on input data bit-width

Разрядность / Bit depth	LUT / (%)	FF / (%)	δ, %	Тактовая частота, МГц / Clock frequency, MHz
4	55	86	12,02	138,274
8	70	129	1,04	92
10	79	143	0,34	85
12	81 (1%)	147 (1%)	0,1	130,174

При ссылке на разрядность далее учитывается и дополнительный знаковый разряд.

На рис. 2-9 представлены гистограммы распределений абсолютных ошибок, представляющие собой зависимость частостей (*f*) от абсолютной погрешности (Δ), и графики, демонстрирующие разброс приведенной погрешности (δ) для разрядности исходных данных 4-8-10-12 (M – порядковый номер измерения).

На рис. 10 представлен график зависимости приведенной погрешности от разрядности исходных данных.

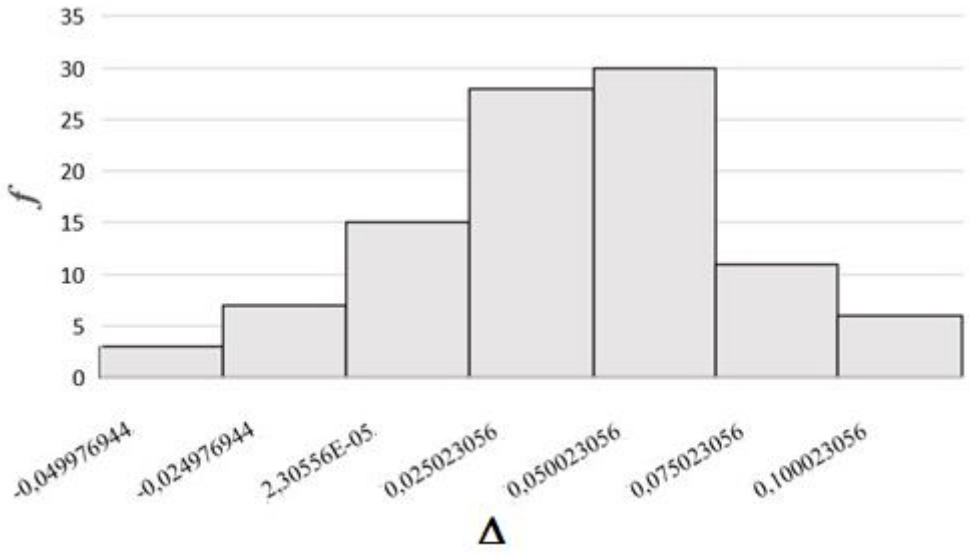

Рис. 2. Гистограмма распределений абсолютных ошибок (*w* =4)

Fig. 2. Histogram of absolute error distributions (w = 4)

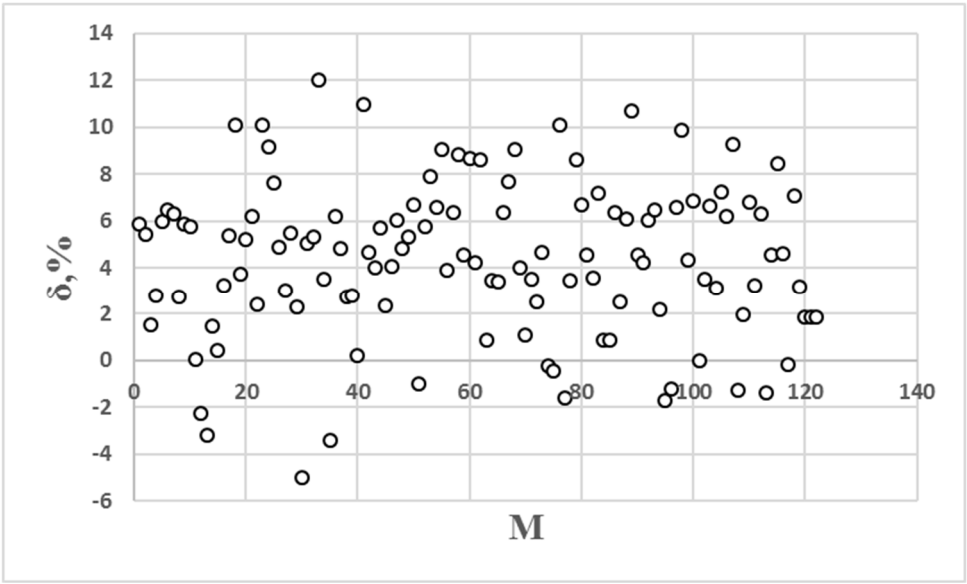

Рис. 3. Разброс приведенных погрешностей (*w* =4, δmax – 12,02%)

Fig. 3. Scatter plot of relative errors (w = 4, δmax = 12.02%)

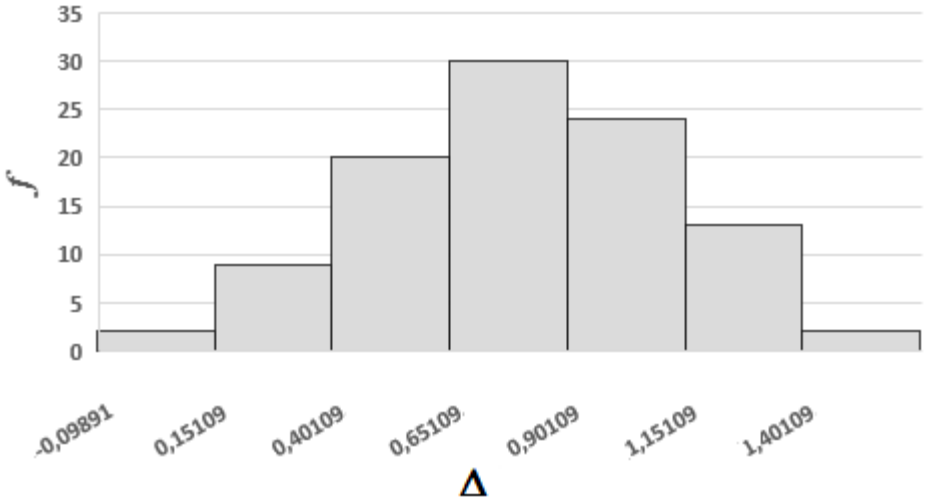

Рис. 4. Гистограмма распределений абсолютных ошибок (*w* =8)

Fig. 4. Histogram of absolute error distributions (w = 8)

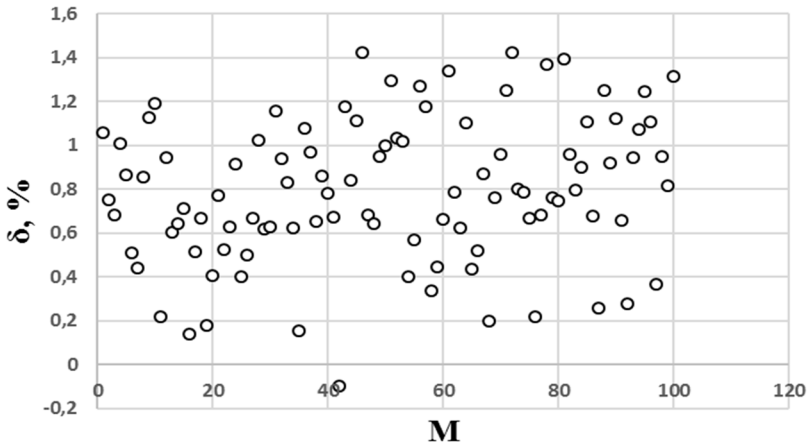

Рис. 5. Разброс приведенных погрешностей (*w* =8, δmax – 1,4%)

Fig. 5. Scatter plot of relative errors (w = 8, δmax = 1.4%);

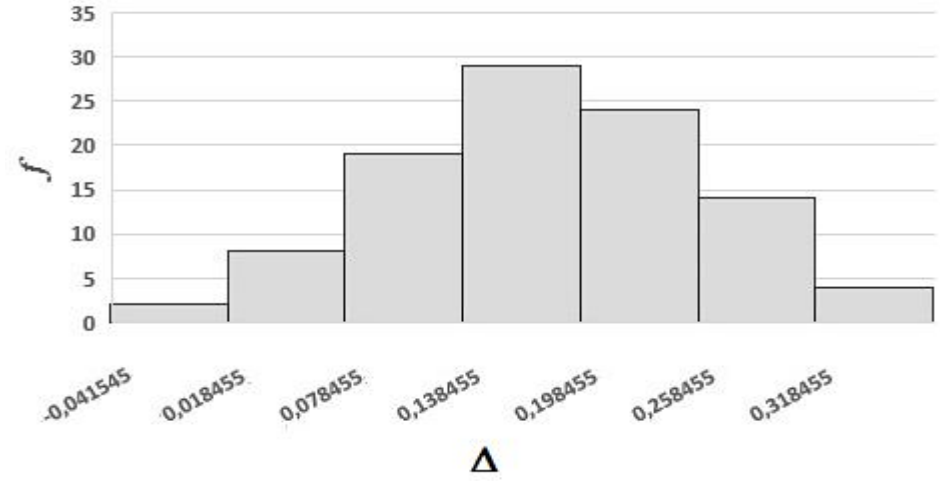

Рис. 6. Гистограмма распределений абсолютных ошибок (*w* =10)

Fig. 6. Histogram of absolute error distributions (w = 10)

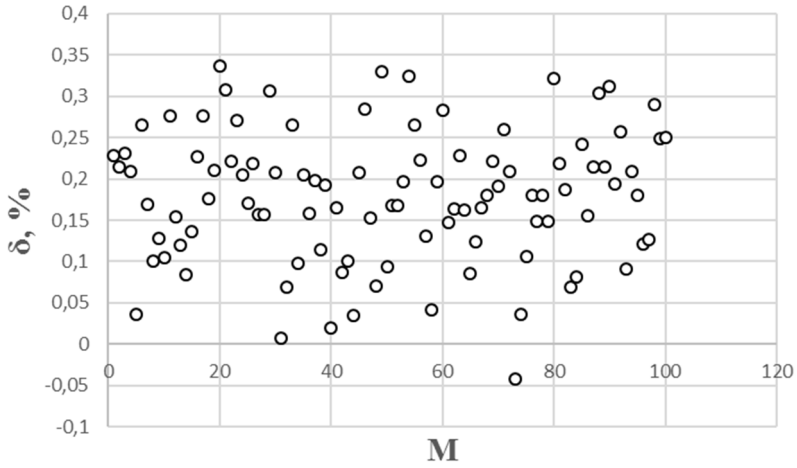

Рис. 7. Разброс приведенных погрешностей (*w* =10, δmax – 0,34%)

Fig. 7. Scatter plot of relative errors (w = 10, δmax = 0.34%)

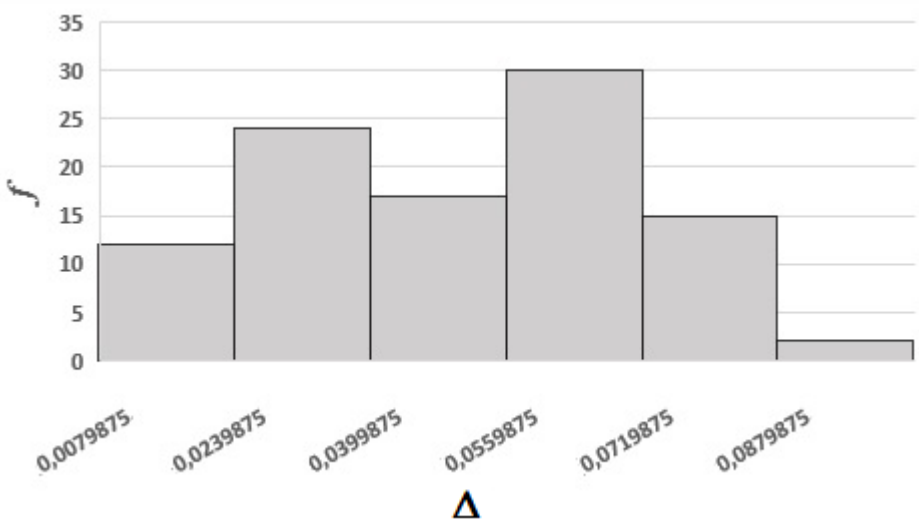

Рис. 8. Гистограмма распределений абсолютных ошибок (*w* =12)

Fig. 8. Histogram of absolute error distributions (w = 12)

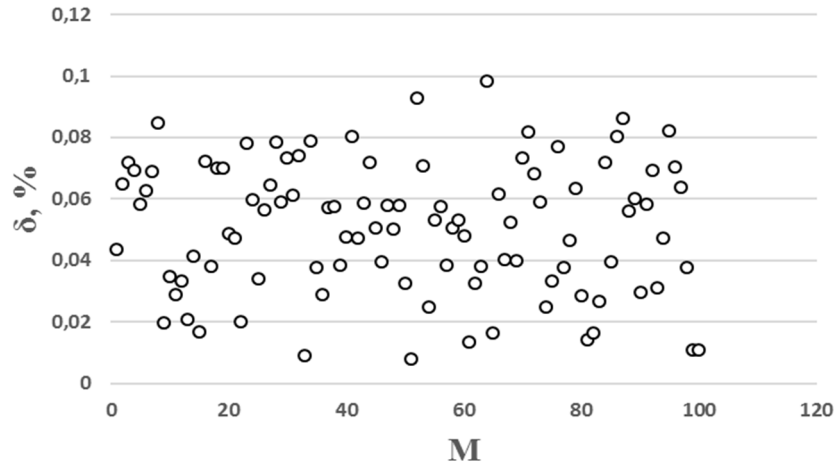

Рис. 9. Разброс приведенных погрешностей (*w* =12, δmax – 0,10%)

Fig. 9. Scatter plot of relative errors (w = 12, δmax = 0.10%)

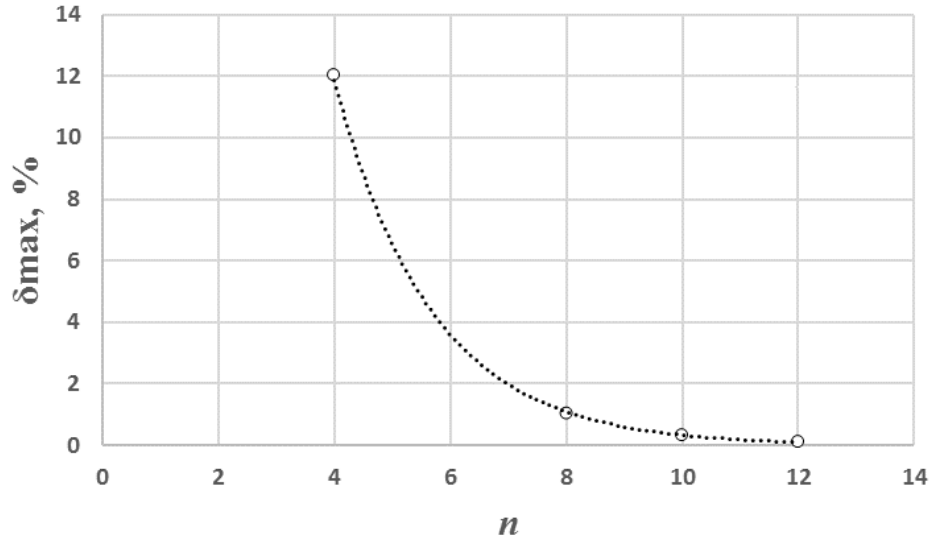

Рис. 10. Зависимости приведенной погрешности от разрядности исходных данных

Fig. 10. Dependence of relative error on input data bit-width

Проведено усреднение результатов многократных измерений при различных наборах входных данных и весов, что позволило повысить достоверность полученных зависимостей. Величина максимально приведённой погрешности уменьшается экспоненциально, что объясняется фундаментальными свойствами квантования данных и соответствует результатам, представленным в [1-4]. Объём используемых аппаратных средств (LUT и FF) растёт с увеличением разрядности, но скорость роста замедляется, что связано с более эффективной оптимизацией при высокой разрядности. Частота снижается с ростом разрядности, но затем снова возрастает, так как при высокой разрядности (12 бит) синтезатор эффективнее оптимизирует логику, что приводит к возобновлению роста частоты. При средней разрядности (10 бит) наблюдается максимальное снижение частоты из-за увеличения глубины комбинационных путей.

Для демонстрации влияния объема памяти (M) табличной реализации сигмоидальной функции на погрешность выходных данных (δ, %) представлены результаты экспериментальных исследований при фиксированной разрядности беззнакового целого равного 11 разрядам (табл. 2).

Таблица 2. Влияние объёма памяти на погрешность вычисления выходного значения

Table 2. Influence of memory size on output value computation error

Количество ячеек памяти (M) / Number of memory cells (M)	δ, %
16384	0,1
8192	0,12
4096	0,2
2048	0,34

График зависимости погрешности вычисления выходного значения нелинейного нейрона от количества ячеек памяти (M) при фиксированном значении разрядности исходных данных представ-

лен на рис. 11. При расчётном значении шага дискретизации 1/512, что соответствует объёму таблицы 4096 значений, уменьшение шага в 4 раза приводит к уменьшению максимального значения погрешности вычислений лишь в 2 раза. Сокращение объёма таблицы сигмоидной функции активации достигается как за счёт её симметрии, так и за счёт выбора оптимального шага дискретизации. При этом восстановление значений функции при отрицательных значениях аргумента требует дополнительных операций в соответствии с выражением (3). Для этого необходим переход от беззнакового целого к целому со знаком, вычитание и коммутация на выход значения функции соответствующего знаку аргумента. Помимо этого, в зависимости от знака аргумента, следует с помощью коммутатора выбрать одно из двух значений. Ускорение и сокращение оборудования при выполнении этих операций достигается за счёт того, что значение функции активации при отрицательном значении аргумента может быть получено поразрядной инверсией табличного значения

$$Y(-x) = \sim Y\left(|x|\right),$$

что реализуется с помощью логических элементов сложения по модулю 2, на первые входы которых подаются разряды табличного значения, а на объединённые вторые входы знак аргумента.

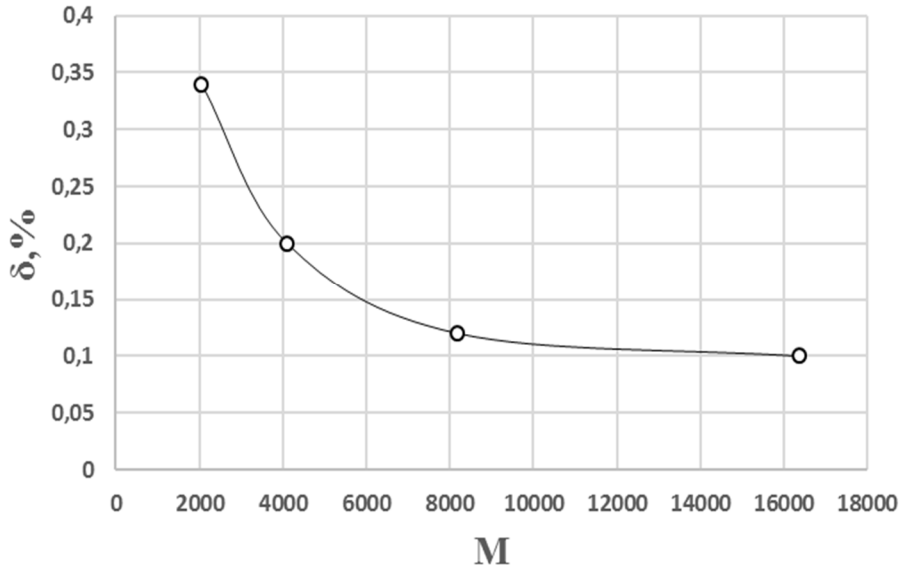

Рис. 11. Зависимость погрешности вычисления выходного значения от количества ячеек памяти (M)

Fig. 11. Dependence of output value computation error on the number of memory cells (M)

При погрешности исходных данных, соответствующей 11-битному разрешению, оптимальным является использование 8192 ячеек памяти, позволяющее достигнуть компромисс между точностью вычислений и затрачиваемыми аппаратными ресурсами.

Анализ существующих исследований в области квантования нейронных сетей для аппаратной реализации на

ПЛИС показывает, что основное внимание уделяется оценке влияния пониженной точности представления данных (весов, активаций) на общую точность работы сетей в целом. Многие работы, такие как [5, 10, 21] и исследования, посвященные бинарным/низкобитным сетям [10, 22] демонстрируют, что переход от 32-битной арифметики с плавающей точкой к 8-, 4-, 2- или даже 1-битной целочисленной арифметике позволяет значительно сократить использование аппаратных ресурсов и энергопотребление с минимальной потерей точности на прикладных задачах, таких как классификация изображений или обработка сигналов. Например, работы по бинаризации [10,13] показывают, что сети могут сохранять высокую эффективность (например, точность классификации >90% на MNIST) при использовании всего 1-2 бит для представления весов и активаций. Методы, такие как FINN – для синтеза с низкобитным квантованием, или Aster – неравномерного квантования, фокусируются на автоматизации процесса квантования для сетей, обеспечивая либо высокую скорость вывода, либо гарантии точности в пределах допустимой погрешности для конкретной задачи. Однако эти исследования, в основном, оценивают точность и эффективность на уровне всей сети или слоев, применяя квантование как часть оптимизации модели или архитектуры сети для конкретных приложений.

В настоящей работе акцент сделан на оптимизацию квантования на уровне отдельного нейрона как базового вычислительного элемента нейронной сети, приводятся теоретические обоснования и результаты экспериментальных исследований влияния квантования на вычислительную точность VHDL-модели нейрона. Подход, предложенный в статье [23], где исследуются различные модели нейронов на основе аналоговых решений с пакетной нормализацией, косвенно подчеркивает значимость точности самого вычислительного блока. Аналогично, работа [14] показывает, что разрядность данных (включая весовые коэффициенты и функцию активации, которые обрабатываются внутри нейронов) напрямую влияет на ресурсы ПЛИС и производительность системы (например, снижение разрядности весов с 8 до 4 бит уменьшило использование LUT более чем на 40% с минимальной потерей чувствительности).

Строгое квантование с гарантиями ошибки, как в Aster [15], также подразумевает анализ ошибок на уровне элементарных операций, происходящих в нейроне (Aster оптимизирует разрядность до 4-16 бит, гарантируя общую ошибку меньше заданного порога).

Таким образом, оптимизация самого нейрона, как элементарной ячейки вычислений, представляется важным и логически обоснованным шагом. Повышение точности вычислений на этом уровне может кумулятивно улучшить точность всей сети, особенно в глубо-

ких архитектурах, где ошибки могут накапливаться. Фокусировка на одиночном нейроне позволяет детально исследовать и минимизировать вносимую ошибку, что может привести к более предсказуемому и надежному поведению сетей, реализованных на ресурсно-ограниченных устройствах.

Выводы

В отличие от большинства работ, в которых оптимизация проводится либо по точности, либо по ресурсам, в данной работе предложен интегрированный подход, позволяющий учитывать оба параметра одновременно. Это даёт возможность находить оптимальное значение разрядности, при котором погрешность вычислений остаётся в допустимых пределах, а аппаратные затраты минимальны.

В работе получена экспоненциальная регрессионная модель, позволяющая прогнозировать уровень погрешности вычисления выходного значения устройства цифровой обработки в зависимости от используемой разрядности. Это делает возможным автоматизированный выбор параметров вычислителя на этапе проектирования. Погрешность вычислений одиночного нейрона для 12-разрядных исходных данных находится на уровне 0,1%.

Все исследования проводились с использованием реального синтеза на ПЛИС Xilinx Spartan 3E XC3S500E, что обеспечивает высокую достоверность результатов и возможность их прямого применения в практике проектирования.

В ходе исследования был разработан и применён метод оценки влияния разрядности представления входных данных и весовых коэффициентов на точность вычислений и объём занимаемых аппаратных ресурсов в цифровом устройстве обработки, реализованном на ПЛИС, применяемом при реализации искусственного нейрона. На основе VHDL-описания нейрона была создана параметризуемая модель, позволяющая изменять разрядность входных сигналов и весовых коэффициентов в требуемом диапазоне. Экспериментально подтверждена предложенная авторами аналитическая зависимость, позволяющая определить оптимальный объем памяти (разрядность адреса) для табличной реализации сигмоидальной функции с учетом требуемой погрешности вычислений и разрядности входных данных. Оптимальным является использование 8192 ячеек памяти, позволяющее достигнуть компромисс между точностью вычислений (δ_{max}=0,12%) и затрачиваемыми аппаратными ресурсами.

В результате исследований получена базовая модель для дальнейшей автоматизации и оптимизации нелинейного цифрового устройства на ПЛИС, идентичная одиночному нейрону, особенно в условиях ограниченных ресурсов и необходимости минимизации энергопотребления и площади кристалла.

Предложенная реализация, основанная на последовательной обработке данных и использовании знаковых чисел, ориентирована на минимизацию

ресурсов, что делает её особенно актуальной для бюджетных ПЛИС, где параллелизм и конвейеризация недоступны, модель также может быть использована в системах, при реализации спецвычислителей с фиксированной точкой, где критична экономия ресурсов и энергии.

В отличие от подавляющего большинства работ, где автоматизация реализована на уровне системного проектирования, предложенная модель позволяет автоматизировать выбор разрядности на уровне отдельного нейрона, что упрощает масштабирование и интеграцию в более сложные системы и может быть использована во встраиваемых устройствах.

Список литературы

1. Accelerating FPGA Implementation of Neural Network Controllers via 32-bit Fixed-Point Design for Real-Time Control / C. Hingu, X. Fu, R. Challoo, J. Lu, X. Yang, L. Qingge // 2023 IEEE 14th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON). 2023; 952-959. https://doi.org/10.1109/ UEMCON59035.2023.10316098

2. Neural Network on the Edge: Efficient and Low Cost FPGA Implementation of Digital Predistortion in MIMO Systems / Y. Jiang, A. Vaicaitis, M. Leeser, J. Dooley // Design, Automation & Test in Europe Conference & Exhibition (DATE). Antwerp, Belgium, 2023. P. 1–2. https://doi.org/10.23919/DATE56975.2023.10137251

3. Antunes P., Podobas A. FPGA-Based Neural Network Accelerators for Space Applications: A Survey. arXiv 2025, arXiv:2504.16173v2. https://doi.org/10.48550/ arXiv.2504.16173

4. Prashanth B.U.V., Ahmed M.R. Design and Implementation of Reconfigurable Neuro-Inspired Computing Model on a FPGA // Adv. Sci. Technol. Eng. Syst. J. 2020. Vol. 5 (5). P. 331–338. https://doi.org/10.25046/aj050541

5. CBin-NN: An Inference Engine for Binarized Neural Networks / F. Sakr, R. Berta, J. Doyle, A. Capello, A. Dabbous, L. Lazzaroni, Bellotti F. // Electronics. 2024. 13. P. 1624. https://doi.org/10.3390/electronics13091624

6. Kumari B.A.S., Kulkarni S.P., Sinchana C.G. FPGA Implementation of Neural Nets // Int. J. Electron. Telecommun. 2023. Vol. 69(3). P. 599–604. https://doi.org/10.24425/ijet.2023.146513

7. Лебедев М.С., Белецкий П.Н. Реализация искусственных нейронных сетей на ПЛИС с помощью открытых инструментов // Труды ИСП РАН. 2021. 33 (6). С. 175–192. https://doi.org/10.15514/ISPRAS-2021-33(6)-12

8. Acharya R.Y., Le Jeune L., Mentens N., Ganji F., Forte D. Quantization-aware Neural Architectural Search for Intrusion Detection. arXiv 2024. arXiv:2311.04194v2. https://doi.org/10.48550/arXiv.2311.04194

9. Efficient Neural Networks on the Edge with FPGAs by Optimizing an Adaptive Activation Function / Y. Jiang, A. Vaicaitis, J. Dooley, M. Leeser // Sensors. 2024. 24(6). P.1829. https://doi.org/10.3390/s24061829

10. FPGA-QNN: Quantized Neural Network Hardware Acceleration on FPGAs / M. Tasci, A. Istanbullu, V. Tumen, S. Kosunalp // Appl. Sci. 2025. 15. P. 688. https://doi.org/10.3390/app15020688

11. Solovyev R., Kustov A., Telpukhov D., Rukhlov V., Kalinin A. Fixed-Point Convolutional Neural Network for Real-Time Video Processing in FPGA. arXiv 2018, arXiv:1808.09945v2. https://doi.org/10.48550/arXiv.1808.09945

12. Wu H., Zheng L., Zhao G., Xu G., Xu M., Liu X., Lin D. Integer Quantization for Deep Learning Inference: Principles and Empirical Evaluation. arXiv 2020, arXiv:2004.09602v3. https://doi.org/10.48550/arXiv.2004.09602

13. Compressing deep neural networks on FPGAs to binary and ternary precision with hls4ml / Ngadiuba J., Loncar V., Pierini M., Summers S., Di Guglielmo G., Duarte J., Harris P., Rankin D., Jindariani S., Liu M., Pedro K., Tran N., Kreinar E., Sagear S., Wu Z., Hoang D. // Mach. Learn.: Sci. Technol. 2021. 2. 015001. https://doi.org/10.1088/2632-2153/aba042

14. Fixed-Point Analysis and FPGA Implementation of Deep Neural Network Based Equalizers for High-Speed PON / N. Kaneda, C.-Y. Chuang, Z. Zhu, A. Mahadevan, B. Farah, K. Bergman, D. Van Veen, V. J. Houtsma // Lightwave Technol. 2022. 40 (7). P. 1972–1980. https://doi.org/10.1109/JLT.2021.3133723

15. Sound Mixed Fixed-Point Quantization of Neural Networks / D. Lohar, C. Jeangoudoux, A. Volkova, E. Darulova // ACM Trans. Embedd. Comput. Syst. 2023. 22 (5s), 136:1–136:26. https://doi.org/10.1145/3609118

16. Jia H., Chen X., Dong D. FPGA-Based Implementation and Quantization of Convolutional Neural Networks // Proceedings of the 2025 3rd International Conference on Communication Networks and Machine Learning (CNML 2025). Nanjing, China, February 21–23, 2025. ACM, New York, NY, USA, 2025. 5 pages. https://doi.org/10.1145/3728199.3728263

17. Pipelined Architecture for a Semantic Segmentation Neural Network on FPGA / H. Le Blevec, M. Léonardon, H. Tessier, M. Arzel // Proceedings of the 2023 30th IEEE International Conference on Electronics, Circuits and Systems (ICECS). Istanbul, Turkey, 2023. P. 1–4. https://doi.org/10.1109/ICECS58634.2023.10382715

18. FPGA implementation of a complete digital spiking silicon neuron for circuit design and network approach / X. Miao, X. Ji, H. Chen, A.M. Mayet, G. Zhang, C. Wang, J. Sun // Sci Rep. 2025. 15. P. 8491. https://doi.org/10.1038/s41598-025-92570-z

19. Wang C.; Luo Z. A Review of the Optimal Design of Neural Networks Based on FPGA // Appl. Sci. 2022. 12. P. 10771. https://doi.org/10.3390/app122110771

20. Claudionor N. Coelho, Jr., Kuusela A., Li S. et al. Automatic heterogeneous quantization of deep neural networks for low-latency inference on the edge for particle detectors // Nat Mach Intell. 2021. 3. P. 675–686. https://doi.org/10.1038/s42256-021-00356-5

21. Gholami A., Kim S., Dong Z., Yao Z., Mahoney M.W., Keutzer K. A Survey of Quantization Methods for Efficient Neural Network Inference. CRC: Boca Raton, FL, USA, 2021. https://doi.org/10.48550/arXiv.2103.13630

22. Courbariaux M., Hubara I., Soudry D., El-Yaniv R., Bengio Y. Binarized Neural Networks: Training Deep Neural Networks with Weights and Activations Constrained to +1 or −1. arXiv 2016, arXiv:1602.02830. https://doi.org/10.48550/arXiv.1602.02830

23. Kavitha S., Kumar C., Alwabli A. A low-power, high accuracy digital design of batch normalized non-linear neuron models: Synthetic experiments and FPGA evaluation // Ain Shams Eng. J. 2025. 16 (8). P. 103469. https://doi.org/10.1016/j.asej.2025.103469

References

1. Hingu C., Fu X., Challoo R., Lu J., Yang X., Qingge L. Accelerating FPGA Implementation of Neural Network Controllers via 32-bit Fixed-Point Design for Real-Time Control. In: *2023 IEEE 14th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. 2023. P. 952-959. https://doi.org/10.1109/ UEMCON59035.2023.10316098

2. Jiang Y., Vaicaitis A., Leeser M., Dooley J. Neural Network on the Edge: Efficient and Low Cost FPGA Implementation of Digital Predistortion in MIMO Systems. In: *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. Antwerp, Belgium; 2023. P. 1–2. https://doi.org/10.23919/DATE56975.2023.10137251

3. Antunes P., Podobas A. FPGA-Based Neural Network Accelerators for Space Applications: A Survey. arXiv 2025, arXiv:2504.16173v2 https://doi.org/10.48550/arXiv.2504.16173

4. Prashanth B.U.V., Ahmed M.R. Design and Implementation of Reconfigurable Neuro-Inspired Computing Model on a FPGA. *Adv. Sci. Technol. Eng. Syst. J.* 2020; 5 (5): 331–338. https://doi.org/10.25046/aj050541

5. Sakr F., Berta R., Doyle J., Capello A., Dabbous A., Lazzaroni L., Bellotti F. CBin-NN: An Inference Engine for Binarized Neural Networks. *Electronics*. 2024; 13: 1624. https://doi.org/10.3390/electronics13091624

6. Kumari B.A.S., Kulkarni S.P., Sinchana C.G. FPGA Implementation of Neural Nets. *Int. J. Electron. Telecommun*. 2023; 69(3): 599–604. https://doi.org/10.24425/ijet.2023.146513

7. Lebedev M.S., Belecky P.N. Artificial Neural Network Inference on FPGAs Using Open-Source Tools. *Trudy ISP RAN = Proceedings of the Institute for System Programming of the RAS (Proceedings of ISP RAS)*. 2021;33(6):175-192. (In Russ.). https://doi.org/10.15514/ ISPRAS-2021-33(6)-12

8. Acharya R.Y., Le Jeune L., Mentens N., Ganji F., Forte D. Quantization-aware Neural Architectural Search for Intrusion Detection. arXiv 2024, arXiv:2311.04194v2. https://doi.org/10.48550/arXiv.2311.04194

9. Jiang Y., Vaicaitis A., Dooley J., Leeser M. Efficient Neural Networks on the Edge with FPGAs by Optimizing an Adaptive Activation Function. *Sensors*. 2024; 24(6):1829. https://doi.org/10.3390/s24061829

10. Tasci M., Istanbullu A., Tumen V., Kosunalp S. FPGA-QNN: Quantized Neural Network Hardware Acceleration on FPGAs. *Appl. Sci.* 2025; 15: 688. https://doi.org/10.3390/app15020688

11. Solovyev R., Kustov A., Telpukhov D., Rukhlov V., Kalinin A. Fixed-Point Convolutional Neural Network for Real-Time Video Processing in FPGA. arXiv 2018, arXiv:1808.09945v2. https://doi.org/10.48550/arXiv.1808.09945

12. Wu H., Zheng L., Zhao G., Xu G., Xu M., Liu X., Lin D. Integer Quantization for Deep Learning Inference: Principles and Empirical Evaluation. arXiv 2020, arXiv:2004.09602v3. https://doi.org/10.48550/arXiv.2004.09602

13. Ngadiuba J., Loncar V., Pierini M., Summers S., Di Guglielmo G., Duarte J., Harris P., Rankin D., Jindariani S., Liu M., Pedro K., Tran N., Kreinar E., Sagear S., Wu Z., Hoang D. Compressing deep neural networks on FPGAs to binary and ternary precision with hls4ml. *Mach. Learn.: Sci. Technol*. 2021. 2: 015001. https://doi.org/10.1088/2632-2153/aba042

14. Kaneda N., Chuang C.-Y., Zhu Z., Mahadevan A., Farah B., Bergman K., Van Veen D., Houtsma V. Fixed-Point Analysis and FPGA Implementation of Deep Neural Network Based Equalizers for High-Speed PON. *J. Lightwave Technol*. 2022; 40 (7): 1972–1980. https://doi.org/10.1109/JLT.2021.3133723

15. Lohar D., Jeangoudoux C., Volkova A., Darulova E. Sound Mixed Fixed-Point Quantization of Neural Networks. *ACM Trans. Embedd. Comput. Syst*. 2023; 22 (5s): 136:1–136:26. https://doi.org/10.1145/3609118

16. Jia H., Chen X., Dong D. FPGA-Based Implementation and Quantization of Convolutional Neural Networks. In: *Proceedings of the 2025 3rd International Conference on Communication Networks and Machine Learning (CNML 2025)*, Nanjing, China, February 21–23, 2025. ACM, New York, NY, USA; 2025. 5 pages. https://doi.org/10.1145/3728199.3728263

17. Le Blevec H., Léonardon M., Tessier H., Arzel M. Pipelined Architecture for a Semantic Segmentation Neural Network on FPGA. In: *Proceedings of the 2023 30th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, Istanbul, Turkey; 2023. P. 1–4. https://doi.org/10.1109/ICECS58634.2023.10382715

18. Miao X., Ji X., Chen H., Mayet A.M., Zhang G., Wang C., Sun J. FPGA implementation of a complete digital spiking silicon neuron for circuit design and network approach. *Sci Rep*. 2025; 15: 8491. https://doi.org/10.1038/s41598-025-92570-z

19. Wang C., Luo Z. A Review of the Optimal Design of Neural Networks Based on FPGA. *Appl. Sci.* 2022; 12: 10771. https://doi.org/10.3390/app122110771

20. Claudionor N. Coelho Jr., Kuusela A., Li S. *et al.* Automatic heterogeneous quantization of deep neural networks for low-latency inference on the edge for particle detectors. *Nat Mach Intell.* 2021; 3: 675–686. https://doi.org/10.1038/s42256-021-00356-5

21. Gholami A., Kim S., Dong Z., Yao Z., Mahoney M.W., Keutzer K. A Survey of Quantization Methods for Efficient Neural Network Inference; CRC: Boca Raton, FL, USA; 2021. https://doi.org/10.48550/arXiv.2103.13630

22. Courbariaux M., Hubara I., Soudry D., El-Yaniv R., Bengio Y. Binarized Neural Networks: Training Deep Neural Networks with Weights and Activations Constrained to +1 or −1. arXiv 2016, arXiv:1602.02830. https://doi.org/10.48550/arXiv.1602.02830

23. Kavitha S., Kumar C., Alwabli A. A low-power, high accuracy digital design of batch normalized non-linear neuron models: Synthetic experiments and FPGA evaluation. *Ain Shams Eng. J.* 2025: 16 (8): 103469. https://doi.org/10.1016/j.asej.2025.103469

Информация об авторах / Information about the Authors

Бондарь Олег Григорьевич, кандидат технических наук, доцент, доцент кафедры космического приборостроения и систем связи, Юго-Западный государственный университет, г. Курск, Российская Федерация, e-mail: b.og@mail.ru

Oleg G. Bondar, Cand. of Sci. (Engineering), Associate Professor, Associate Professor of Space Instrumentation and Communication Systems Department, Southwest State University, Kursk, Russian Federation, e-mail: b.og@mail.ru

Брежнева Екатерина Олеговна, кандидат технических наук, доцент кафедры космического приборостроения и систем связи, Юго-Западный государственный университет, г. Курск, Российская Федерация, e-mail: bregnevaeo@mail.ru

Ekaterina O. Brezhneva, Cand. of Sci. (Engineering), Associate Professor of Space Instrumentation and Communication Systems Department, Southwest State University, Kursk, Russian Federation, e-mail: bregnevaeo@mail.ru

Голубев Дмитрий Александрович, студент кафедры космического приборостроения и систем связи, Юго-Западный государственный университет, г. Курск, Российская Федерация, e-mail: golubew.2019@mail.ru

Dmitry A. Golubev, Student of Space Instrumentation and Communication Systems Department, Southwest State University, Kursk, Russian Federation, e-mail: golubew.2019@mail.ru