УДК 621.382

РЕАЛИЗАЦИЯ НА ПЛИС МОДУЛЯ ИСКУССТВЕННОГО НЕЙРОНА ДЛЯ ПОСЛЕДОВАТЕЛЬНО-ПАРАЛЛЕЛЬНЫХ АРХИТЕКТУР НЕЙРОННЫХ СЕТЕЙ С ПРЯМОЙ СВЯЗЬЮ*

Ушенина И.В., *к.т.н., доцент кафедры «Программирование» Пензенского государственного технологического университета, e-mail: ivl23@yandex.ru*
Данилов Е.А., *к.т.н., доцент кафедры «Программирование» Пензенского государственного технологического университета, e-mail: evgenydplus@mail.ru*

FPGA IMPLEMENTATION OF THE ARTIFICAL NEURON MODULE FOR SERIES-PARALLEL ARCHITECTURES OF FEEDFORWARD NEURAL NETWORKS

Ushenina I.V., Danilov E.A.

In this paper, the authors consider the levels of parallelism used in feedforward neural networks and, taking this into account, evaluate the possibilities of implementing artificial neurons using FPGA resources. As a result, the artificial neuron module that supports neuron-level parallelism is implemented on FPGA and analyzed. The module has been designed so that when implementing a neural network layer by connecting several instances of the module into the chain, no external elements are necessary in the layer's circuit. This feature provides predictable timing characteristics for the neural network layer being implemented. In the module, the linear part of calculations and calculation of activation function are made by a DSP block and programmable logic blocks, respectively.

Key words: neuron, neuron-level parallelism, bit-level mapping method, FPGA, DSP block, timing analysis.

Ключевые слова: нейрон, параллелизм на уровне нейронов, метод поразрядного отображения, ПЛИС, блок ЦОС, временной анализ.

Исходя из уровней параллелизма вычислений в нейронных сетях с прямой связью, авторы оценивают возможности построения на ресурсах ПЛИС схем искусственных нейронов. В итоге реализован на ПЛИС и проанализирован модуль искусственного нейрона, поддерживающий параллелизм на уровне нейронов. Модуль содержит все необходимое, чтобы при формировании слоя нейронной сети из нужного количества его экземпляров требовалось только их объединение в цепочку без добавления к ней каких-либо внешних элементов. Это обеспечит предсказуемость временных характеристик реализуемой нейронной сети. Линейная часть вычислений в модуле выполняется блоком цифровой обработки сигналов ПЛИС, вычисление функции активации – блоками программируемой логики.

Введение

Аппаратная реализация нейронных сетей выполняется для ускорения их работы или обучения, а также, если требуется работа нейронной сети в реальном времени. Одним из требований к аппаратной платформе, на которой реализуется нейронная сеть, является возможность выполнения массивно-параллельных вычислений. Этому требованию удовлетворяют несколько видов платформ, в том числе графические процессоры (GPU), специализированные микросхемы и системы на кристалле (ASIC, SoC), ПЛИС (FPGA). Последние могут проигрывать другим платформам в производительности и энергоэффективности, но являются наиболее доступными [1-3]. Согласно обзору [3], около двух третей случаев аппаратной реализации алгоритмов искусственного интеллекта, в том числе нейронных сетей, приходятся на ПЛИС.

Поскольку нейронная сеть – это, по сути, массив соединенных между собой нужным образом одинаковых вычислительных блоков (искусственных нейронов), при реализации нейронной сети на ПЛИС логично оформить описание нейрона в виде модуля и использовать нужное количество его экземпляров. Однако в зависимости от вида параллелизма вычислений в нейронной сети схемы нейронов могут выглядеть по-разному и формировать нейронные сети, работающие с разной производительностью.

С этой точки зрения в работе проанализированы варианты реализации на ПЛИС модулей нейронов для нейронных сетей с прямой связью типа «многослойный персептрон». В итоге выбрана и реализована схема модуля с параллелизмом вычислений на уровне нейронов. При реализации слоя нейронной сети нужное количество экземпляров этого модуля объединяется без каких-либо дополнений получившейся схемы. Модуль реализован на ПЛИС GW5AST от Gowin Semiconductor [4]; оценены его производительность и ресурсоемкость. Для описания, реализации и временного анализа схемы модуля использована среда проектирования Gowin EDA.

Вычисления в нейронах и ресурсы ПЛИС

На рисунке 1 представлено устройство искусственного нейрона. Нейрон под номером k, расположенный в i-м слое нейронной сети, содержащем K_i нейронов, принимает от K_{i-1} нейронов предыдущего слоя сигналы x_{k1}^i … $x_{kK_{i-1}}^i$ и умножает их на весовые коэффициенты w_{k1}^i … $w_{kK_{i-1}}^i$, а полученные произведения складывает и добав-

ляет к ним пороговое значение b_k^i:

$$a_k^i = b_k^i + \sum_{j=1}^{K_{i-1}} x_{kj}^i w_{kj}^j . \quad (1)$$

Полученная сумма (обозначим ее a_k^i) поступает на вычислитель функции активации (ФА), в качестве которой выступает сигмоида:

$$s(a_k^i) = \frac{1}{1+e^{-a_k^i}} . \quad (2)$$

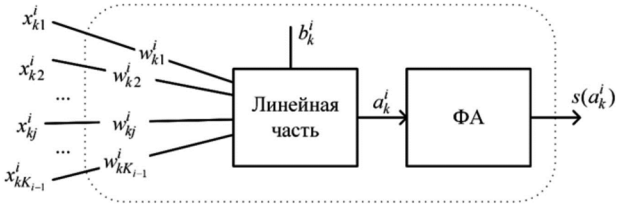


Рисунок 1 – Устройство искусственного нейрона

Умножение с накоплением, выполняемое в (1) – линейная операция, требующая сумматоров и умножителей. Так как она выполняется многократно, скорость ее выполнения оказывает большое влияние на производительность нейронной сети в целом. Очевидным решением для реализации вычислений по ней являются блоки цифровой обработки сигналов (ЦОС), которые присутствуют почти во всех современных ПЛИС хотя бы в количестве нескольких десятков. Типичный блок ЦОС содержит хотя бы один умножитель и сумматор, на базе которого можно реализовать аккумулятор. Блоки ЦОС можно объединять в цепочки [5].

В качестве ФА в нейронных сетях с прямой связью чаще всего используются сигмоида или гиперболический тангенс. Эти функции не только имеют подходящую форму, но и непрерывно дифференцируются, что необходимо для обучения нейронной сети по алгоритму обратного распространения ошибки. Обе функции – нелинейные, и для их непосредственного вычисления ресурс у ПЛИС отсутствует. В данной работе для вычисления сигмоиды (2) используется метод поразрядного отображения [6, 7]. Суть метода в том, что разряды двоичного представления $s(a_k^i)$ вычисляются отдельно друг от друга как булевы функции разрядов a_k^i. Для таких вычислений можно использовать блоки программируемой логики ПЛИС, сохранив менее многочисленные блоки ОЗУ и ЦОС для линейной части вычислений (1). В [7] авторы представили схемы вычисления отдельных разрядов $s(a_k^i)$. В данной работе в вычислителях ФА (рисунок 2) используется по два набора таких схем, принимающих значащие разряды аргумента функции сигмоиды, a_k^i. Первый набор вычисляет ФА, исходя из предположения, что $a_k^i < 0$. Полученный им результат обозначен на рисунке 2 как $sn(a_k^i)$. Второй набор схем предполагает, что $a_k^i \geq 0$, а полученный им результат обозначен как $sp(a_k^i)$. Далее нужное значение ФА выбирается мультиплексором в зависимости от знакового разряда a_k^i.

Таким образом, линейная часть вычислений в разрабатываемом модуле искусственного нейрона выполняется блоками ЦОС, а нелинейная – блоками программируемой логики. Коэффициенты для взвешивания входных сигналов нейронов хранятся в блоках ОЗУ ПЛИС.

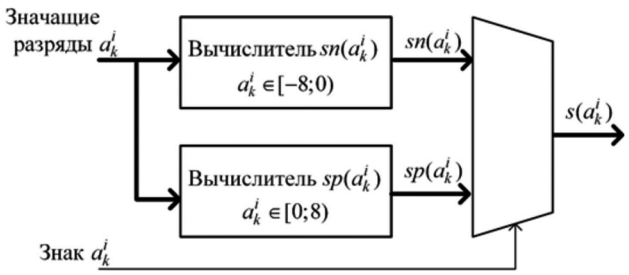


Рисунок 2 – Вычислитель функции активации

В [8] показано, что при аппаратной реализации обученной нейронной сети достаточно представления всех чисел, которыми она оперирует, в формате с фиксированной запятой с точностью 7-8 бит. Реализуемый здесь модуль искусственного нейрона работает с числами, значащая часть которых представлена 8 битами, а знаковый разряд, когда он нужен – дополнительным 9-м битом. Подробности представления чисел в модуле искусственного нейрона приведены в таблице 1.

Уровни параллелизма вычислений в нейронных сетях

При ускорении нейронных сетей или организации их работы в реальном времени важнейшей характеристикой реализуемых схем является производительность, которая во многом зависит от степени параллелизма выполнения вычислений. Выделяют [9-11] несколько

Таблица 1 – Форматы представления чисел в модуле искусственного нейрона

Вид переменной	Обозначение	Диапазон значений	Знаковый разряд, бит	Целая часть, бит	Дробная часть, бит
j-й входной сигнал *k*-го нейрона в i-м слое	x_{kj}^i	[0;1)	0	0	8
Выходной сигнал *k*-го нейрона в *i*-м слое	y_k^i	[0;1)	0	0	8
Весовой коэффициент *j*-го входного сигнала, поступающего на *k*-й нейрон *i*-го слоя	w_{kj}^i	[-8;8)	1	3	5
Аргумент ФА *k*-го нейрона в *i*-м слое	a_k^i	[-8;8)	1	3	5
Значение ФА *k*-го нейрона в *i*-м слое	$s(a_k^i)$	[0;1)	0	0	8

уровней параллелизма вычислений в нейронных сетях, среди которых: 1) уровень слоев, когда каждый слой нейронной сети реализован на отдельных ресурсах, что позволяет организовать конвейерные вычисления; 2) уровень нейронов, когда каждый нейрон занимает отдельный ресурс, и вычисления во всех нейронах как минимум в пределах одного слоя нейронной сети выполняются параллельно; 3) уровень синапсов, то есть поступающих на нейрон входных сигналов, при котором все они умножаются на весовые коэффициенты одновременно, а последующее сложение полученных произведений по возможности распараллеливается.

Нейронные сети могут сочетать несколько уровней параллелизма. Крайними случаями являются полное отсутствие параллелизма, когда вычисления, требуемые от каждого нейрона в каждом слое, выполняются последовательно единственным вычислительным блоком [12], и параллелизм на всех трех уровнях [11]. Тем не менее, при полном отсутствии параллелизма возможности ПЛИС не используются в полной мере, а параллелизм на всех уровнях реализуем на большинстве ПЛИС только при малых размерах сети.

Если ПЛИС содержит небольшое (относительно размеров сети) количество блоков ЦОС, вероятно, придется отказаться от параллелизма на уровне слоев и выбрать один из уровней параллелизма – нейронов, как, например, в [13-15], или синапсов, как в [16]. Сравним эти два варианта.

При параллелизме на уровне нейронов каждый нейрон в слое нейронной сети автономен, так как реализуется на отдельных ресурсах ПЛИС (рисунок 3). На все нейроны слоя в каждый момент времени поступает сигнал только от одного из нейронов предыдущего слоя. То есть, на каждый k-й нейрон i-го слоя поступают с каждым тактом сигналы x_{k1}^i, x_{k2}^i, …, x_{kj}^i, …, x_{kK-1}^i. Весовые коэффициенты w_{k1}^i, w_{k2}^i, …, w_{kj}^i, …, w_{kK-1}^i с каждым тактом считываются из блоков памяти (ОЗУ), подключенных к умножителям. Согласно (1), x_{kj}^i и w_{kj}^i перемножаются, и их произведение поступает в аккумулятор (его образуют сумматор и следующий за ним регистр). Перед началом цикла K_{i-1} операций умножения с накоплением в регистр аккумулятора помещается пороговое значение b_k^i. Результат вычислений (1), a_k^i, поступает на блок вычисления ФА.

При отсутствии параллелизма на уровне слоев выходные сигналы нейронов, y_k^i, должны быть снова по очереди поданы на их входы для выполнения вычислений следующего, i + 1-го слоя сети. Сигналы y_k^i принимает мультиплексор, выход которого синхронизируется регистром 1. Вместо мультиплексора для поочередной подачи сигналов y_k^i на входы нейронов может использоваться регистр сдвига [13, 14].

Структура, представленная на рисунке 3, может использоваться для реализации слоев нейронных сетей, начиная со второго. Нейроны первого слоя не обрабатывают, а только перераспределяют поступающие сигналы. Для выбора между сигналами от первого слоя, x_{kj}^2, и последующих слоев, x_{kj}^i, перед умножителями установлен еще один мультиплексор.

Обозначив T_{clk} период тактового сигнала, поступающего на регистры и ОЗУ схемы на рисунке 3, получим, что на все вычисления в i-м слое нейронной сети требуется время $t_н$:

$$t_н = T_{clk} \cdot (K_{i-1} + 2). \quad (3)$$

Определения параллелизма на уровне синапсов, данные в [9-11], предписывают нейронам выполнять все операции умножения x_{kj}^i на w_{kj}^i одновременно и по возможности распараллеливать последующие операции суммирования полученных произведений. Таким образом, нейрон должен содержать K_{i-1} блоков ЦОС для выполнения умножений.

Для распараллеливания суммирования произведений в [9] предлагается использовать в нейроне «дерево» сумматоров. Тогда выходной сигнал нейрона, y_k^i, будет вычислен за время, пропорциональное $\log_2 K_{i-1}$. Но древовидная организация сумматоров на ПЛИС невыгодна: она требует или повышенного расхода блоков ЦОС, или реализации части сумматоров на блоках программируемой логики с потерей производительности.

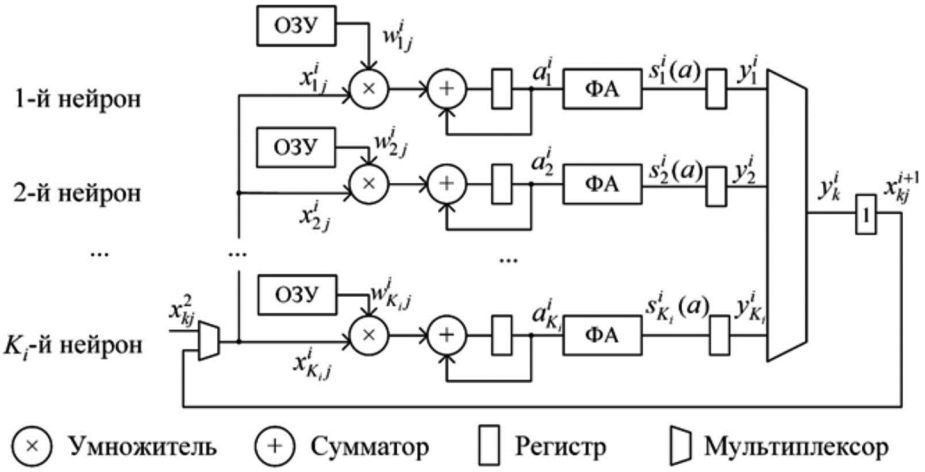


Рисунок 3 – Слой нейронной сети с параллелизмом на уровне нейронов (параллелизм на уровнях синапсов и слоев отсутствует; вместо мультиплексора сигналов y_k^i может использоваться регистр сдвига)

Для ПЛИС больше подходит лестничная организация сумматоров в нейроне, при которой сумматоры соединены последовательно, образуя цепочку. То есть, в каждом из K_{i-1} блоков ЦОС нейрона выполняется взвешивание одного из его входных сигналов и передача произведения на один из входов сумматора. На другой вход сумматора поступает сумма произведений x_{kj}^i и w_{kj}^i, вычисленная предыдущими сумматорами цепочки. На первый сумматор в цепочке поступает, кроме произведения $x_{k1}^i w_{k1}^i$, пороговое значение b_k^i. Значение y_k^i при лестничной организации сумматоров будет вычисляться за время, пропорциональное K_{i-1}.

При отсутствии параллелизма на уровне нейронов и строгом следовании определению параллелизма на уровне синапсов схема с лестничной организацией сумматоров выполняет вычисления для нейронов *i*-го слоя по очереди и проигрывает в производительности схеме с параллелизмом на уровне нейронов (3) примерно в K_i раз.

Если отклониться от определения параллелизма на уровне синапсов и допустить конвейеризацию вычислений в умножителях и сумматорах в пределах слоя нейронной сети, можем получить схему, представленную на рис. 4, где лестничная организация сумматоров совмещена с конвейеризацией вычислений. То есть, часть умножителей и сумматоров, уже выполнивших расчеты для *k*-го нейрона, начинают обслуживать (*k*+1)-й нейрон, (*k*+2)-й и т.д. Входные сигналы $x_{k1}^i \dots x_{kK-1}^i$. поступают на эту схему одновременно, и для любого k это будет один и тот же набор сигналов. При каждом *j*-м ($j \in [1; K_{i-1}]$) блоке ЦОС находится блок памяти (ОЗУ), хранящий набор коэффициентов $w_{1j}^i \dots w_{K_i j}^i$, одним из которых будет взвешиваться *j*-й входной сигнал реализуемого в данный момент времени *k*-го нейрона. По окончании расчетов, соответствующих *k*-у нейрону, формируется выходной сигнал y_k^i, который поступает на цепочку регистров. По окончании расчетов, соответствующих K_i-у нейрону, на выходах регистров этой цепочки содержится набор сигналов $y_1^i \dots y_{K_i}^i$, которые поступают на мультиплексоры. На мультиплексоры также поступают сигналы от первого слоя сети – $x_{k1}^2 \dots x_{kK_1}^2$. При переходе к обслуживанию следующего, *i* + 1-го, слоя нейронной сети выходы мультиплексоров синхронизируются и поступают на умножители.

При использовании схемы на рисунке 4 все вычисления в *i*-м слое нейронной сети займут время t_c

$t_c = T_{clk} \cdot (K_{i-1} + K_i + 1)$.

Для схем на рисунке 3 и рисунке 4 минимальное значение T_{clk} будет определяться временем одной итерации вычислений по (1), выполняемых блоком ЦОС, или временем вычисления ФА. Таким образом, конвейеризация вычислений в пределах одного слоя в схеме с параллелизмом на уровне синапсов и лестничной организацией сумматоров сокращает ее отставание в производительности от схемы с параллелизмом на уровне нейронов.

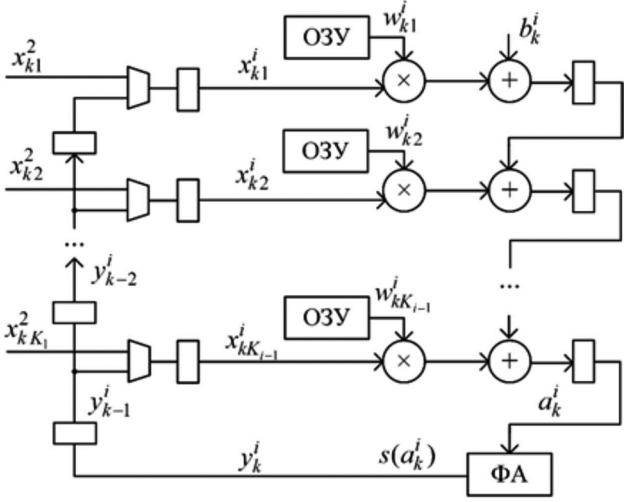


Рисунок 4 – Слой нейронной сети с параллелизмом на уровне синапсов и конвейеризацией вычислений в пределах слоя (параллелизм на уровнях нейронов и слоев отсутствует)

Это отставание можно сократить еще больше при дальнейшей конвейеризации схемы с параллелизмом на уровне синапсов и выполнении в каждый момент времени вычислений, относящихся не только к разным нейронам, но и к разным слоям. Однако по сравнению со схемой с параллелизмом на уровне нейронов (рисунок 3) у полученной схемы будет ряд недостатков.

Во-первых, построение нейронной сети на основе схемы с параллелизмом на уровне синапсов – это не соединение нужного количества экземпляров модуля искусственного нейрона, а настройка размеров одного модуля в зависимости от количества нейронов в предыдущем слое. Это может сделать комплекс мер для сохранения высокой производительности нейронной сети и их эффективность зависимыми от размеров сети.

Во-вторых, при отсутствии параллелизма на уровне слоев размеры схемы (количество в ней блоков ЦОС) определяются размерами слоя с максимальным количеством нейронов. При обслуживании слоев с меньшим количеством нейронов промежуточные суммы произведений проходят более длинные, чем необходимо, цепочки сумматоров, что замедляет работу нейронной сети.

В-третьих, в отличие от параллелизма на уровне нейронов, где все нейроны слоя могут управляться одним и тем же набором сигналов, схема с параллелизмом на уровне синапсов нуждается в индивидуальном управлении каждым блоком ЦОС. Это усложняет управляющий автомат и может увеличить нагрузку на трассировочные ресурсы ПЛИС.

Таким образом, для модуля искусственного нейрона, реализуемого на ПЛИС, предпочтительнее оказывается поддержка параллелизма на уровне нейронов.

Схема модуля искусственного нейрона

Представим схему модуля искусственного нейрона, поддерживающую параллелизм на уровне нейронов и предназначенную для реализации на ПЛИС GW5AST.

При разработке модуля предполагалось его исполь-

зование в нейронных сетях как с параллелизмом на уровне слоев, так и без него. От модуля также требовалось, чтобы его схема не зависела от количества нейронов в слоях сети, а также, чтобы при формировании из его экземпляров нейронной сети дополнительные внешние элементы были не нужны, и сеть представляла бы собой массив одинаковых вычислительных блоков. Все это обеспечивает предсказуемость временных характеристик как отдельного модуля, так и составленной из его экземпляров нейронной сети, а также облегчает описание нейронных сетей различных размеров.

С этой точки зрения рассмотрены два варианта обработки массива выходных сигналов параллельно работающих нейронов, y_k^i, которые при переходе к вычислениям следующего слоя должны по очереди снова поступать на входы нейронов. Первый вариант, показанный на рисунке 3, – использование мультиплексора с K_i входами. Так как ПЛИС не располагают мультиплексорами с большим количеством входов, на практике это будет «дерево» из многих мультиплексоров с намного меньшим, чем K_i, количеством входов у каждого из них. Внизу «дерева» будет находиться несколько мультиплексоров с суммарным количеством входов, равным K_i. На вершине «дерева» будет находиться один мультиплексор, формирующий выходной сигнал. Сколько мультиплексоров пройдут выходные сигналы нейронов и за какое время, зависит от величины K_i и от архитектуры ПЛИС. Мультиплексоры «дерева» не получится равномерно распределить по модулям отдельных нейронов, то есть «дерево» придется оформлять отдельным модулем и настраивать в зависимости от размеров слоя сети, что противоречит выдвинутым выше требованиям.

Второй вариант – использование регистра сдвига с параллельной загрузкой, количество звеньев которого равно K_i. Модуль искусственного нейрона сможет содержать одно звено этого регистра, соединяемое со звеньями других модулей при формировании слоя сети. Чтобы все разряды y_k^i сдвигались за один такт, каждый разряд y_k^i передается по отдельной сдвигающей цепочке триггеров. Таким образом, приходим к схеме регистра сдвига, представленной на рисунке 5. Каждое звено этого регистра представлено набором триггеров с мультиплексорами перед входами данных. На один из входов мультиплексоры принимают разряды выходного сигнала y_k^i «своего» нейрона, на другой – разряды выходного сигнала звена регистра сдвига предыдущего нейрона.

На рисунке 6 представлена схема модуля искусственного нейрона, реализованная на ПЛИС GW5AST. Блоки ЦОС этой ПЛИС содержат по два умножителя, а значит, вычисление выходного сигнала идет почти в два раза быстрее, чем представлено в (3). Весовые коэффициенты для обоих входных сигналов, x_{kj}^i и x_{kj+1}^i, хранятся в общем блоке памяти (ОЗУ). Ячейки ОЗУ разрядностью 18 бит хранят по два весовых коэффициента, считываемых одновременно.

Умножители модуля могут принимать как сигналы от первого слоя нейронной сети, то есть x_{kj}^2 и x_{kj+1}^2, так и сигналы x_{kj}^i и x_{kj+1}^i, рассчитанные ранее этим и другими модулями и поступающие от регистра сдвига. Для выбора нужных входных сигналов перед умножителями установлены мультиплексоры. В отличие от схемы на рисунке 3, входные сигналы умножителей синхронизированы регистрами 5-8.

В связи с тем, что в блоках ЦОС содержится по два умножителя, все модули, составляющие слой нейронной сети, разбиваются на две равные группы с отдельными друг от друга регистрами сдвига выходных сигналов y_k^i.

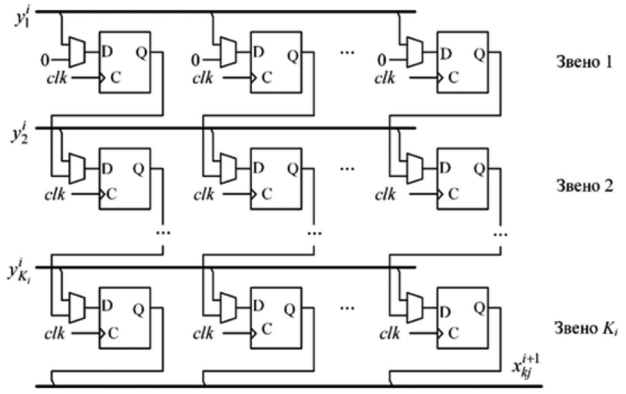


Рисунок 5 – Регистр сдвига со звеньями, распределяемыми по модулям искусственного нейрона

Таким образом, на все вычисления в слое нейронной сети, состоящем из представленных на рисунке 6 модулей, потребуется время $t_н$:

$$t_н = T_{clk} \cdot (\frac{K_{i-1}}{2} + 4) .$$

Единственное дополнение, которое может потребоваться схеме слоя нейронной сети, состоящей из K_i экземпляров представленного модуля, – это регистры, расставленные на путях следования сигналов от регистров сдвига к умножителям. Эти регистры нужны для сохранения производительности (тактовой частоты) на уровне, достигнутом внутри модулей.

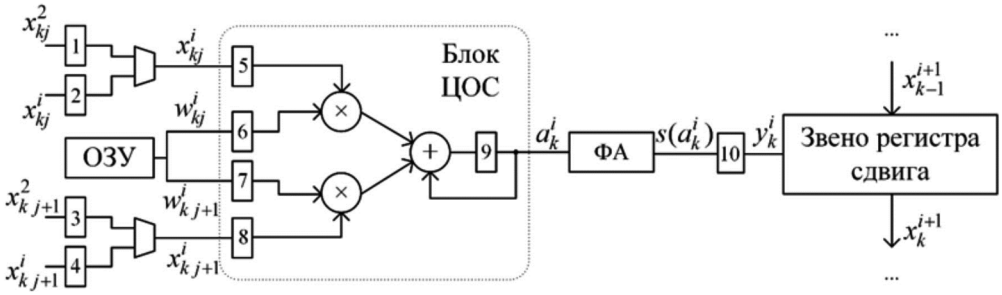


Рисунок 6 – Схема модуля искусственного нейрона, реализованная на ПЛИС GW5AST

Результаты реализации модуля искусственного нейрона

Результаты реализации модуля искусственного нейрона на ПЛИС GW5AST представлены в таблицах 2 и 3. В таблице 2 приведены потребности модуля в различных видах ресурсов ПЛИС. На реализацию модуля требуется один блок ЦОС, один блок ОЗУ и 27 блоков программируемой логики.

ПЛИС GW5AST содержит 298 блоков ЦОС, поэтому при отсутствии параллелизма на уровне слоев на ней можно реализовывать нейронные сети, содержащие в слоях до 298 нейронов.

Общее количество нейронов сети, реализуемой на ПЛИС, ограничивается также ее запасами блочной памяти; в данном случае ПЛИС содержит 340 блоков ОЗУ по 18 кбит. На хранение весовых коэффициентов для сигналов, поступающих от одного слоя из 298 нейронов, требуется 2,62 кбит памяти. Таким образом, максимальный размер нейронной сети, который может быть реализован на выбранной ПЛИС с помощью представленного модуля – 6 слоев по 298 нейронов в каждом слое.

На реализацию модуля также тратится 27 блоков программируемой логики, что составляет менее 0,16 % от их общего запаса в ПЛИС. При реализации нейронной сети с использованием всех имеющихся блоков ЦОС будет занято около 47 % от всего запаса блоков программируемой логики. То есть, блоки программируемой логики в данном случае не являются критическим ресурсом для реализации нейронной сети.

Если синхронизирующие регистры в модуле расположены, как показано на рисунке 6, получаются пять ступеней конвейера, которым соответствуют пять видов операций в слое нейронной сети. В таблице 3 приведены составляющие этих операций и время их выполнения. Видно, что больше всего времени (6,761 нс) занимает вычисление ФА. Это время и будет ограничивать максимальную тактовую частоту модуля $F_{\max}$. С учетом этого времени, качества и условий распространения тактового сигнала и времени установления синхронных элементов временной анализ модуля показал, что его $F_{\max}$ составляет 128 МГц.

При переходе от одного модуля к состоящей из его экземпляров нейронной сети без параллелизма на уровне слоев $F_{\max}$ снижается из-за большой задержки распространения сигналов от регистров сдвига до регистров 2 и 4 каждого нейрона (рисунок 6). При реализации слоя нейронной сети на ПЛИС GW5AST с использованием предложенного модуля эта задержка может составить до 11,67 нс. Чтобы вернуть $F_{\max}$ к уровню, достигнутому для отдельно взятого модуля, можно разместить по два регистра на путях следования x_{kj}^i и x_{kj+1}^i.

Таблица 2 – Ресурсоемкость модуля искусственного нейрона

Ресурсы ПЛИС и реализуемые на них элементы модуля	Требуемый объем ресурсов
Блоки ЦОС (умножители, сумматор, регистры 5-9)	1 блок из 298
Блоки памяти (ОЗУ)	1 блок из 340
Блоки программируемой логики (ФА)	21 блок из 17280
Блоки программируемой логики (звено регистра сдвига, регистры 1-4, регистр 10, мультиплексоры)	6 из 17280

Таблица 3 – Временные характеристики модуля искусственного нейрона

Вид операции	Составляющие операций	Время выполнения, нс
1. Доставка x_{kj}^i и w_{kj}^i к регистрам 5-8	Чтение ОЗУ	1,608
	Распространение сигналов от ОЗУ до регистров 6 и 7	до 1,371
	Запись регистров 1-4	0,306
	Распространение сигналов от регистров 1-4 до регистров 5 и 8 Из них время работы мультиплексоров	до 1,917 0,463
2. Одна итерация в блоке ЦОС	Умножение с накоплением согласно формуле (1)	3,338
3. Вычисление ФА	Запись регистра 9	0,306
	Распространение сигналов от регистра 9 до вычислителя ФА	до 2,922
	Вычисление ФА	3,531
	Распространение сигналов от вычислителя ФА до регистра 10	до 0,002
4. Доставка y_k^i к триггерам регистра сдвига	Запись регистра 10	0,306
	Распространение сигналов от регистра 10 до триггеров звена регистра сдвига (см. рисунок 5) Из них время работы мультиплексоров звена регистра сдвига	до 1,902 0,463
5. Прохождение звена регистра сдвига	Запись триггеров звена регистра сдвига	0,306
	Распространение сигналов от триггеров звена регистра сдвига в составе модуля до триггеров звена регистра сдвига в следующем модуле ((k+1)-й нейрон) Из них время работы мультиплексоров звена регистра сдвига в следующем модуле	до 0,632 0,463

Заключение

Реализован на ПЛИС и проанализирован модуль искусственного нейрона, предназначенный для нейронных сетей с прямой связью и поддерживающий параллелизм вычислений на уровне нейронов. За счет этого время вычислений в слое нейронной сети не зависит от количества нейронов в нем, а за счет наличия в модуле двух умножителей время вычислений в слое пропорционально половине количества нейронов в предыдущем слое. Независимо от размеров нейронной сети, состоящей из экземпляров модуля, ее производительность может быть сохранена на уровне производительности отдельных нейронов.

** Исследование выполнено за счет гранта Российского научного фонда и Пензенской области № 24-21-20100, https://rscf.ru/project/24-21-20100/*

Литература

1. Nurvitadhi E. et al. Can FPGAs beat GPUs in accelerating next-generation deep neural networks? Proceedings of the 2017 ACM/SIGDA international symposium on field-programmable gate arrays (FPGA'17). 2017, pp. 5-14.

2. Шипицин С.П., Ямаев М.И. Развитие аппаратно-ориентированных нейронных сетей на FPGA и ASIC. Вестник Пермского национального исследовательского политехнического университета. Электротехника, информационные технологии, системы управления. 2019. № 31. С. 177-192.

3. Talib M.A. et al. A systematic literature review on hardware implementation of artificial intelligence algorithms. The Journal of Supercomputing. 2021. vol. 77. no. 2. pp. 1897-1938.

4. GW5AST series of FPGA Products. URL: https://cdn.gowinsemi.com.cn/DS1104E.pdf (Дата обращения: 25.10.24).

5. Arora V Digital Signal Processing (DSP) User Guide. URL: https://www.gowinsemi.com/upload/database_doc/2927/document/6671c51a10e8b.pdf (Дата обращения: 25.10.24).

6. Tommiska M.T. Efficient digital implementation of the sigmoid function for reprogrammable logic. IEEE Proceedings-Computers and Digital Techniques. 2003, vol. 150, no. 6, pp. 403-411.

7. Ushenina I.V., Danilov E.A. Implementation of the Sigmoid Function on Current FPGAs Using the Bit-Level Mapping Method. Proceedings of the 2024 International Russian Automation Conference (RusAutoCon). 2024. pp. 295-300.

8. Holt J.L., Hwang J.N. Finite precision error analysis of neural network hardware implementations. IEEE Transactions on Computers. 1993, vol. 42, no. 3, pp. 281-290.

9. Omondi A.R., Rajapakse J.C. FPGA implementations of neural networks. New York: Springer, 2006. 360 p.

10. Nordström T., Svensson B. Using and designing massively parallel computers for artificial neural networks. Journal of parallel and distributed computing. 1992, vol. 14, no. 3, pp. 260-285.

11. Savich, A., Moussa, M., Areibi S. A scalable pipelined architecture for real-time computation of MLP-BP neural networks. Microprocessors and Microsystems. 2012, vol. 36, no. 2, pp. 138-150.

12. Gorbounov Y., Chen H. Optimized FPGA Implementation of an Artificial Neural Network Using a Single Neuron. Proceedings of International Conference on Computer Science and Education in Computer Science. 2023, pp. 233-246.

13. Medus L.D., Iakymchuk T., Frances-Villora J.V., Bataller-Mompeán M., Rosado-Munoz A. A novel systolic parallel hardware architecture for the FPGA acceleration of feedforward neural networks. IEEE Access. 2019, vol. 7, pp. 76084-76103.

14. Thasnimol V.S., George M.A. Hardware Accelerator Implementation of Multilayer Perceptron. Proceedings of Congress on Intelligent Systems (CIS 2020). 2021, vol. 1, pp. 107-119.

15. Vu H.M., Thang H.V. A Customized Hardware Architecture for Multi-layer Artificial Neural Networks on FPGA. Proceedings of Fourth International Conference on Information Systems Design and Intelligent Applications. 2018. pp. 637-644.

16. Gomperts A., Ukil A., Zurfluh F. Development and implementation of parameterized FPGA-based general purpose neural networks for online applications. IEEE Transactions on Industrial Informatics. 2010. vol. 7, no. 1, pp. 78-89.

НОВЫЕ КНИГИ



Гаврилов К.Ю., Каменский И.В., Кирдяшкин В.В., Линников О.Н.

Моделирование и обработка радиолокационных сигналов в MATLAB: Учебное пособие. М.: Изд-во Радиотехника, 2020 г. 264 с.: ил.

Рассмотрены методы моделирования радиолокационных сигналов при отражении от сложных целей, принципы моделирования аналоговых и цифровых устройств обработки сигналов, включающие формирование двумерной матрицы цифровых отсчетов, методы согласованной фильтрации, обнаружения и обработки сигналов в импульсно-доплеровских радиолокационных системах.

Показаны примеры обработки наиболее распространенных видов радиолокационных сигналов – импульсных, с линейной частотной модуляцией и фазо-кодоманипулированных сигналов. Приведены программы моделирования и обработки сигналов в среде MATLAB.

Для студентов, аспирантов и инженеров, изучающих и использующих теорию радиолокации и методы моделирования и обработки радиолокационных сигналов. Будет полезна научным работникам и разработчикам радиолокационных систем.