

Научная статья

DOI: 10.15593/2224-9397/2022.1.08

УДК 004.89

С.Ф. Тюрин^{1,2}, В.В. Бахтин^{1,2}, И.А. Подлесных^{1,3}¹Пермский национальный исследовательский политехнический университет,
Пермь, Россия²Пермский государственный национальный исследовательский университет,
Пермь, Россия³АО «Новомет-Пермь», Пермь, Россия

МЕТОД СИНТЕЗА УСТРОЙСТВ НЕЙРОСЕТЕВОГО РАСПОЗНАВАНИЯ НА ПРОГРАММИРУЕМОЙ ЛОГИКЕ ДЛЯ РЕАЛИЗАЦИИ РЕЖИМА FOG COMPUTING

Проблема реализации нейронных сетей на малопроизводительных устройствах рассматривалась во множестве работ различных авторов по всему миру. В этих работах приводятся методы и алгоритмы имплементации нейронных сетей на микроконтроллерах, ПЛИС и иных устройствах. **Целью исследования** является разработка метода синтеза устройств реализации искусственных нейронных сетей на программируемой логике, ориентированных на туманные вычисления. Основой для создания рассматриваемых устройств выступает искусственная нейронная сеть, которую потребуется разделить на несколько блоков. Каждый из этих вычислительных блоков исполняется на отдельном физическом устройстве. Связь между блоками осуществляется с помощью стандартных каналов и протоколов. **Методика исследования** базируется на математическом моделировании нейронной сети, схемотехническом моделировании каскада устройств нейросетевого распознавания, алгоритмизации и программировании декомпозиции и запуска блочной нейронной сети, анализе полученных результатов эксперимента. **В результате исследования** планируется получить метод синтеза устройств реализации искусственных нейронных сетей на программируемой логике, ориентированных на туманные вычисления. В статье рассмотрен сформулированный метод синтеза устройств. Реализованы электрические схемы функциональной работы блочных нейронных сетей на каскадах вычислительных устройств различной размерности. Разработаны программы, выполнявшиеся на микроконтроллерах для реализации нейросетевых вычислений, проведены конфигурация и настройка микроконтроллеров. Осуществлены экспериментальные запуски каскадов устройств и получены подтверждения работоспособности и эффективности метода синтеза устройств.

Ключевые слова: схемотехническое моделирование; метод синтеза; искусственная нейронная сеть; программируемая логика; туманные вычисления; AVR; устройство нейросетевого распознавания; микроконтроллер; Proteus.

S.F. Tyurin^{1,2}, V.V. Bakhtin¹, I.A. Podlesnykh^{1,3}

¹Perm National Research Polytechnic University, Perm, Russian Federation

²Perm State University, Perm, Russian Federation

³Novomet, Perm, Russian Federation

METHOD OF SYNTHESIS OF NEURAL NETWORK RECOGNITION DEVICES BASED ON PROGRAMMABLE LOGIC FOR THE IMPLEMENTATION OF FOG COMPUTING MODE

The problem of implementing neural networks on low-performance devices has been considered in many works by various authors around the world. These papers present methods and algorithms for the implementation of neural networks on microcontrollers, FPGAs and other devices. The aim of the study is to develop a method for synthesizing devices for implementing artificial neural networks based on programmable logic, focused on fog computing. The basis for the creation of the devices in question is an artificial neural network, which will need to be divided into several blocks. Each of these computing blocks is executed on a separate physical device. Communication between the blocks is carried out using standard channels and protocols. The research methodology is based on mathematical modeling of a neural network, circuit modeling of a cascade of neural network recognition devices, algorithmization and programming of decomposition and launch of a block neural network, analysis of the experimental results obtained. As a result of the research, it is planned to obtain a method for synthesizing devices for implementing artificial neural networks based on programmable logic, focused on fog computing. The article considers the formulated method of device synthesis. Electrical circuits of functional operation of block neural networks on cascades of computing devices of various dimensions are implemented. Programs have been developed, executed on microcontrollers for the implementation of neural network computing, configuration and configuration of microcontrollers have been carried out. Experimental launches of cascades of devices have been carried out and confirmations of the operability and effectiveness of the device synthesis method have been obtained.

Keywords: circuit modeling; synthesis method; artificial neural network; programmable logic; fog computing; AVR; neural network recognition device; microcontroller; Proteus.

Введение

Проблема реализации нейронных сетей на малопроизводительных устройствах рассматривалась во множестве работ различных авторов по всему миру. В этих работах приводятся методы и алгоритмы имплементации нейронных сетей на микроконтроллерах, ПЛИС и прочих устройствах. Как пример, данная задача рассматривается в статьях В.А. Руднева [1], Pierre-Emmanuel Novac и др. [2], Nicholas J. Cotton и др. [3], Tu Y и др. [4], а также в ряде прочих работ других современных ученых. Однако по-прежнему не решена задача синтеза сетей распределенных малопроизводительных устройств для размещения и использования

нейронных сетей на них. Это обстоятельство открывает широкий простор для работы в направлении создания новейших сетей распределенных вычислений для решения задач искусственного интеллекта и распознавания образов с использованием простых устройств с невысокой производительностью. В результате исследования планируется получить метод декомпозиции нейронной сети на блоки, которые будут работать на оконечных устройствах, также будет проведено испытание данного метода на тестовом каскаде вычислительных устройств.

Такие распределенные вычислительные системы, на которых производятся вычисления нейронных сетей, находят свое применение при решении множества задач и проблем современного мира, от простых бытовых до крайне нетривиальных вопросов науки и техники:

- Умный дом – интеллектуальная система управления жильем, которая может включать в себя распознавание голоса и команд от хозяина дома, автоматизированное управление климатом и освещением, уход за питомцами [5].

- Медицина – автоматическая оценка анализов пациента, позволяющая получить результаты вплоть до диагноза болезни удаленно и без участия врача.

- Безопасность – анализ толпы людей для выявления преступных элементов и потенциально опасных лиц с помощью поиска паттернов агрессивного поведения.

- Лингвистика – автоматический перевод речи и текста в режиме реального времени [6–8]. Распознавание и выявление экстремистских высказываний.

- Управление – системы анализа данных и принятия решений на предприятиях.

- Экономика – анализ рыночной ситуации для прогнозирования поведения рынка недвижимости, ценных бумаг, драгоценных металлов и углеводородов [9].

- Искусство – нестандартное применение нейронных сетей для создания предметов искусства: музыки, картин, театральных сценариев и прочего.

Для начала необходимо дать несколько определений. Нейронная сеть – последовательность из нескольких слоев математических нейронов, соединенных друг с другом. Первый слой нейронов называется входным слоем, он анализирует данные об исследуемом объекте, и пе-

реводит это в вид коэффициентов, ограниченных определенным интервалом. Последний слой нейронов называется выходным слоем, именно он передает во внешнюю среду информацию о решении, принятом нейронной сетью [10, 11].

Нейронную сеть, все слои которой производят свои вычисления на одном и том же вычислительном устройстве, назовем **монолитной нейронной сетью (МНС)**. Если стоит задача получения нейронной сети, которая может проводить свои вычисления на нескольких связанных между собой устройствах, то нам потребуется разделить слои этой нейронной сети на блоки последовательных, идущих друг за другом слоев. Каждый из этих блоков будет выполнен на отдельном вычислительном устройстве, а промежуточные результаты будут переданы по сети между ними. Нейронную сеть, разбитую на набор подобных блоков, назовем **блочной нейронной сетью (БНС)** [12].

Исходя из представленных предположений, можно сформулировать **цель представленной работы**, – разработка метода синтеза устройств реализации искусственных нейронных сетей на программируемой логике, ориентированных на туманные вычисления.

Для достижения цели потребовалось решить следующие задачи:

1) создание **математической модели** распознавания объектов с помощью искусственных нейронных сетей устройствами **на программируемой логике**, ориентированными на туманные вычисления;

2) разработка **метода синтеза** устройств реализации искусственных нейронных сетей **на программируемой логике**, ориентированных на туманные вычисления;

3) реализация **алгоритма** преобразования классической нейронной сети в нейронную сеть, адаптированную для туманных вычислений в устройствах **на программируемой логике**;

4) оценка сложности нейросетевого распознавания с использованием нового метода;

5) тестирование нейросетевого распознавания на устройствах **на программируемой логике**, замеры результирующих параметров устройства.

В работе «Математическая модель искусственной нейронной сети для устройств на ПЛИС и микроконтроллерах, ориентированных на туманные вычисления» [13] была получена и представлена общая рекуррентная формула описания блочных нейронных сетей. Кроме того,

были описаны различные варианты разделения монолитной нейронной сети на блочные в зависимости от различных условий сети туманных (распределенных) вычислений. Таким образом, была решена первая из поставленных частных задач исследования.

В работе «Алгоритм разделения монолитной нейронной сети для реализации туманных вычислений в устройствах на программируемой логике» было дано детальное описание процесса разделения монолитной нейронной сети на блочные нейронные сети, а также процесса запуска и работы распределенной нейронной сети. Данное описание было представлено в виде кода на языке программирования высокого уровня JAVA. Дано описание нового специально разработанного формата файла, предназначенного для описания, хранения и передачи информации о нейронной сети как монолитной, так и блочной. В этой работе были представлены оценки сложности алгоритмов декомпозиции и функционирования БНС, т.е. в работе были решены третья и четвертая задачи исследования.

1. Моделирование работы каскадов устройств реализации искусственных нейронных сетей на программируемой логике в режиме туманных вычислений

Для тестирования предлагаемого метода было осуществлено схемотехническое моделирование различных конфигураций каскадов устройств нейросетевого распознавания в САПР Proteus [14–16]. Произведено моделирование работы нейронных сетей различных вариантов: монолитной нейронной сети на одном микроконтроллере, чтобы иметь эталонный вариант работающей модели, и представлены модели, в которых последовательно соединялись друг с другом три, пять и семь микроконтроллеров соответственно. Для каждого из случаев применения нескольких вычислителей в каскаде осуществлялась декомпозиция исходной монолитной нейронной сети в соответствии с количеством контроллеров.

Устройством, которое будет реализовывать нейросетевые вычисления, был выбран микроконтроллер ATmega32 фирмы ATMEL семейства AVR, который представляет собой 8-разрядный микроконтроллер общего назначения с RISC-ядром [17, 18]. Кристаллы AVR имеют в своем составе Flash-память программ, память данных EEPROM и SRAM, порты ввода-вывода и типовые для 8-битных микроконтрол-

леров периферийные устройства [19]. Немаловажно для нашей задачи, что микроконтроллер имеет программируемый USART и позволит нам передавать от одного устройства к другому результаты работы одного блока нейронной сети, которые станут исходными данными для следующего блока. Объем Flash-памяти 32 КБайта и будет предельным размером загружаемой для исполнения на микроконтроллере программы. Реализация программы для вычислительного устройства будет происходить на языке C, а именно его имплементации MicroC, созданной специально для реализации программ, которые будут исполняться на микроконтроллерах [20].

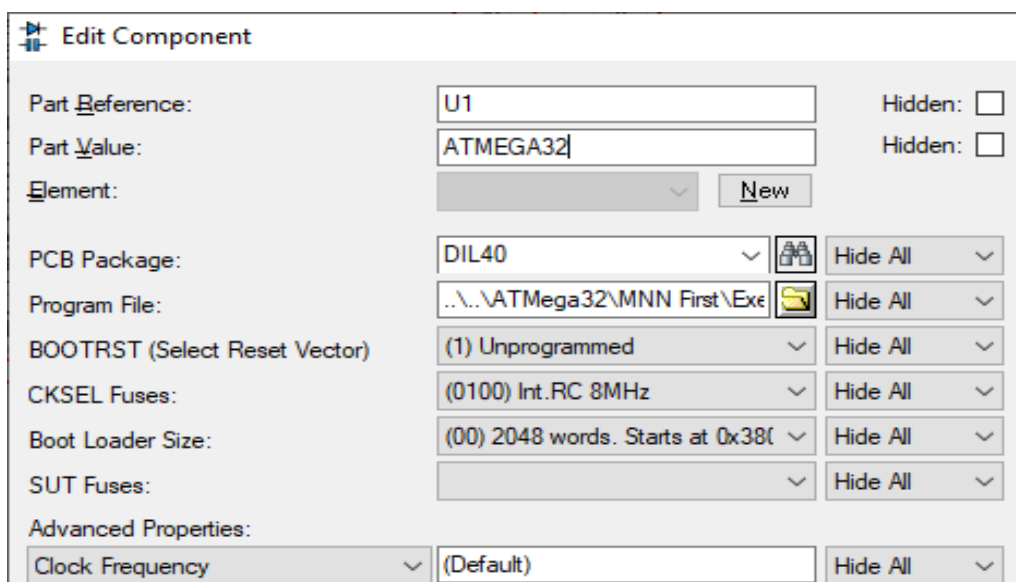


Рис. 1. Параметры запуска контроллера ATmega32

Настройка передачи с использованием UART начинается непосредственно с соединения пина PD1/TXD (трансммиттера UART) первого контроллера с пином PD0/RXD (ресивера UART) принимающего контроллера, но этого еще недостаточно для полноценной работы. Все настройки приемопередатчика хранятся в регистрах конфигурации UCSRA, UCSRB и UCSRC. Самое главное – осуществить конфигурацию UCSRB в битах 3 – TXEN и 4 – RXEN, их необходимо выставить в единицу для разрешения передачи и приема соответственно. Далее требуется установить в единицу флаги 6 – TXCIE и 7 – RXCIE, разрешающие прерывания передачи и приема. Нас интересуют в первую очередь прерывания приема, которые активно использовались для

получения данных из канала UART. У AVR есть регистр UDR, это UART Data Register. На самом деле это два разных регистра, но имеющие один адрес. Просто на запись он попадает в один (регистр передатчика), а на чтение берет из другого (регистр приемника). Настройки микроконтроллера, с которыми осуществлялось его использование, представлены на рис. 1.

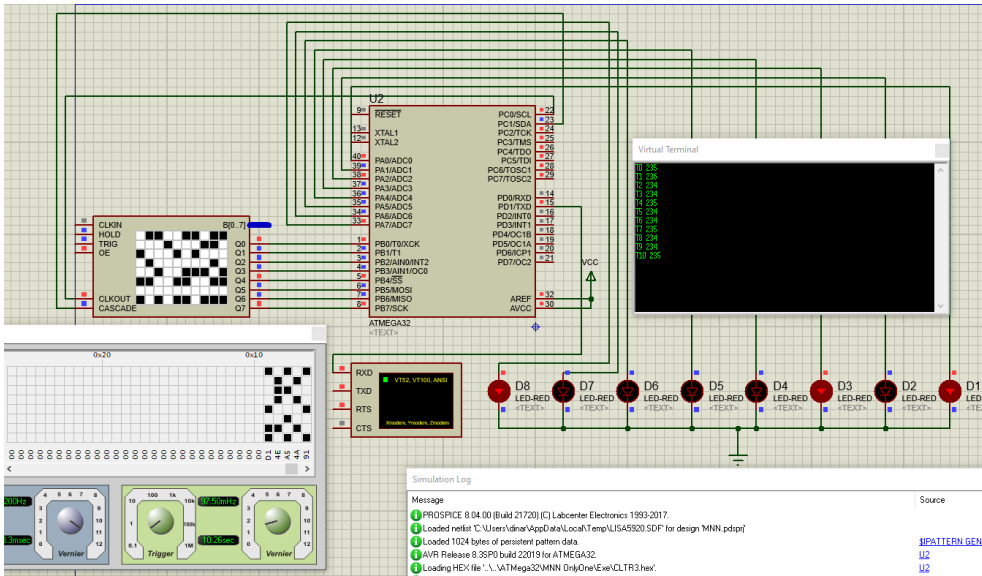


Рис. 2. Схема электрическая функциональная работы МНС на одном микроконтроллере ATМega32 в системе схемотехнического моделирования

Микроконтроллер в наших экспериментах работал на частоте 8 МГц. В случае с одним микроконтроллером используется генератор слов PATTERN GENERATOR, для генерации входных сигналов, который передавал сгенерированное слово на настроенные как входы пины PB1-8. Также к PC0 и PC1 были подключены выходы генератора слов CLKOUT и CASCADE, которые отвечали за генерацию сигнала при отправке нового слова и отправке последовательности с самого начала соответственно. Пины PA1-8 были настроены как выходы и передавали на LED-RED-светодиоды информацию о результате вычислений, произведенных монолитной нейронной сетью. Также результат передавался на Virtual Terminal, который подключался к PD1/TXD, т.е. передатчику UART, и получал результаты вычислений на табло для каждого из подаваемых примеров данных. Описанная схема подробно представлена на рис. 2.

В первом эксперименте МНС была разделена на три блока, каждый из которых исполнялся на отдельном микроконтроллере. Для этого функционал был частично разделен между контроллерами. Так, первый контроллер остался подключенным к генератору слов, получал от него вектор исходных данных и передавал контроллеру сигналы синхронизации. Третий, т.е. последний контроллер получил функции вывода результирующих данных как на массив светодиодов LED-RED1-8, так и на экран виртуального терминала, используя транмиттер UART, т.е. пин PD1/TXD.

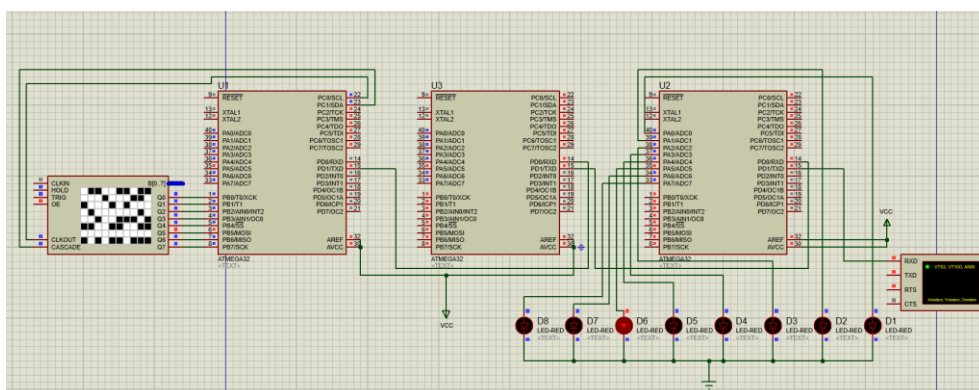


Рис. 3. Схема электрическая функциональная работы БНС на трех микроконтроллерах ATmega32 в системе схемотехнического моделирования

Соединение же первого и второго микроконтроллеров осуществлялось с помощью UART, а именно транмиттер первого микроконтроллера TXD соединялся с ресивером второго контроллера RXD. Подробно схема представлена на рис. 3.

Похожим образом собирается схема на пять микроконтроллеров, первый микроконтроллер получает входные данные, осуществляет вычисления и передает результат по UART второму. Следующий принимает данные, обрабатывает их и передают контроллеру номер три. Аналогично отработывают третий и четвертый вычислительные узлы, необходимо четко следить, чтобы на промежуточных контроллерах биты управления передачей и приемом, а также биты, разрешающие прерывания по отправке и получению, выставлялись в единицы. Последнее вычислительное устройство завершает обработку и выводит результаты на массив светодиодов и в терминал. С результатами схемотехнического моделирования можно ознакомиться на рис. 4. Схожим образом, но

с добавлением еще двух промежуточных вычислительных блоков осуществляется моделирование каскада из семи устройств нейросетевого распознавания.

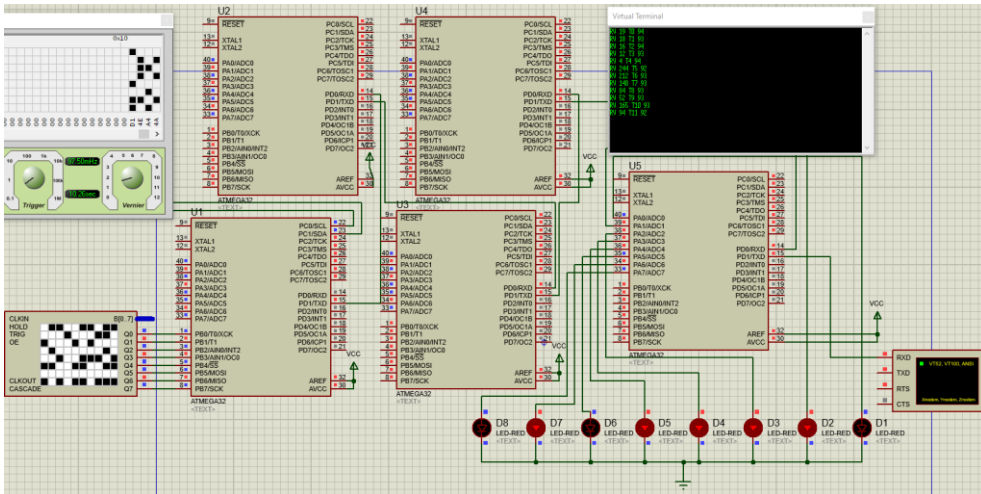


Рис. 4. Схема электрическая функциональная работы БНС на трех микроконтроллерах ATmega32 в системе схемотехнического моделирования

2. Реализация программы контроллера и конфигурирование устройства нейросетевого распознавания

Перейдем к рассмотрению непосредственно метода синтеза устройств реализации искусственных нейронных сетей на программируемой логике [21–23], ориентированных на туманные вычисления [24, 25]. Для этого проследим путь от исходной монолитной нейронной сети до устройства нейросетевого распознавания, реализующего конкретный вычислительный блок распределенной блочной нейронной сети.

Для работы с нейронными сетями в унифицированном виде на устройствах с программируемой логикой потребовалось создать легкий формат хранения нейронной сети в файле. Формат файлов получил расширение .ann – сокращенное от artificial neural network. МНС в данном формате подается на вход алгоритму декомпозиции нейронной сети, также ему на вход подается число вычислительных устройств D, на которые требуется декомпозировать имеющуюся сеть. В нашем эксперименте оставшимися параметрами были условие разделение по слоям и разделение на примерно равные части, т.е. случай вычислительных устройств одинаковой производительности. В результате де-

композиции получаем D файлов в формате *.app, в каждом из которых находится один из блоков распределенной нейронной сети. В файле для разделения числа слоев и описания каждого из слоев используется символ перевода каретки на новую строку, но в дальнейшем полученное представление нейронной сети будет использовано в программе для микроконтроллера, поэтому разделитель верхнего уровня был заменен на символ «:».

Для разработки и компиляции прошивки для микроконтроллера ATmega32 было использовано программное обеспечение CodeVisionAVR – интегрированная среда разработки программного обеспечения для микроконтроллеров семейства AVR фирмы Atmel. Для реализации программы для вычислительного узла использовался язык программирования C. При разработке программы для контроллера создается проект, при создании выбирается модель конкретного микроконтроллера, для которого создается прошивка, в нашем случае – ATmega32. Также при создании указываются стартовые параметры, т.е. значения регистров конфигураций, отвечающие за настройку функционала контроллера, при необходимости существует возможность поменять эти значения уже непосредственно в программном коде разрабатываемого проекта. Нейронная сеть устанавливается в прошивку копированием содержимого файла *.app в исходный код программы в виде строки `char nnetworkString[]`. Пример интеграции описания нейронной сети в код программы представлен на рис. 5.

```
69 unsigned long delay = 0;
70 unsigned int inputVector = 0;
71 unsigned int outputVector = 0;
72 int time = 0;
73 int fo=0;
74 char nnetworkString[] = "30:0,8,0;12;1;1;0,1;12;1;1;1,2;12;1;1;2,4;12;1;1;0,5;
```

Рис. 5. Пример переноса нейронной сети в код программы для контроллера

Существуют три типа программ, различных по функционалу, для первого контроллера – с обработкой входных данных, для промежуточных вычислителей – с получением, обработкой и отправкой данных, а также для последнего контроллера – с получением, обработкой и выводом результатов. Рассмотрим немного подробнее этот вариант прошивки, его можно увидеть на рис. 6. Флаг `fo` выставляется в 1 в рамках прерывания получения данных `interrupt [USART_RXC] void usart_rx_isr(void)`, которое вызывается при получении определенного

количества бит, здесь же полученные данные помещаются во входной вектор $inputVector = data$. В основной же части в переменную $time$ помещается значение счетного регистра TCNT2, содержащего число тиков таймера на текущий момент, по сути – локальное время выполнения. Далее осуществляется вызов непосредственно функции работы нейронной сети, встроенной в данную программу. В переменную $delay$ помещается вычисленная разница между временем после окончания вычисления результатов работы сети и переменной $time$. Полученный вектор и задержка в тиках выводятся на экран виртуального терминала. Полученный вектор отправляется на PORTA для демонстрации на сигналах светодиодов.

```
515 while (1)
516     {
517         if (fo==1)
518             {
519                 fo=0;
520                 time = TCNT2;
521                 outputVector = computingNN(inputVector);
522                 delay = countDelay(TCNT2, time);
523                 printf("RV %u T%i %u%c%c", outputVector, j,delay,10,13);
524                 PORTA=outputVector;
525                 j++;
```

Рис. 6. Основной код программы финального контроллера

3. Результаты эксперимента

Было проведено несколько запусков различных схмотехнических моделей. Первым было осуществлено моделирование исполнения МНС на одном микроконтроллере. Далее осуществлена декомпозиция нейронной сети на блоки для исполнения на трех, пяти и семи контроллерах. На вход каждой модели были последовательно представлены одинаковые исходные данные, созданные в генераторе слов. Результаты всех описанных итераций моделирования представлены в таблице.

Полученные результаты подтверждают, что предложенный метод синтеза устройств нейросетевого распознавания для реализации режима туманных вычислений работает верно, ведь независимо от того, выполняется нейронная сеть на одном устройстве или вычисления происходят в каскаде с различным числом вычислительных устройств, результаты во всех перечисленных случаях получаются одинаковыми.

Результаты работы каждой из схем на одинаковых входных параметрах

1 АТМega32(МНС)	3 АТМega32(БНС)	5 АТМega32(БНС)	7 АТМega32(БНС)
00010011	00010011	00010011	00010011
00010010	00010010	00010010	00010010
00010000	00010000	00010000	00010000
00001100	00001100	00001100	00001100
00000100	00000100	00000100	00000100
11110100	11110100	11110100	11110100
11010100	11010100	11010100	11010100
10010100	10010100	10010100	10010100
01010100	01010100	01010100	01010100
00110100	00110100	00110100	00110100
00100100	00100100	00100100	00100100

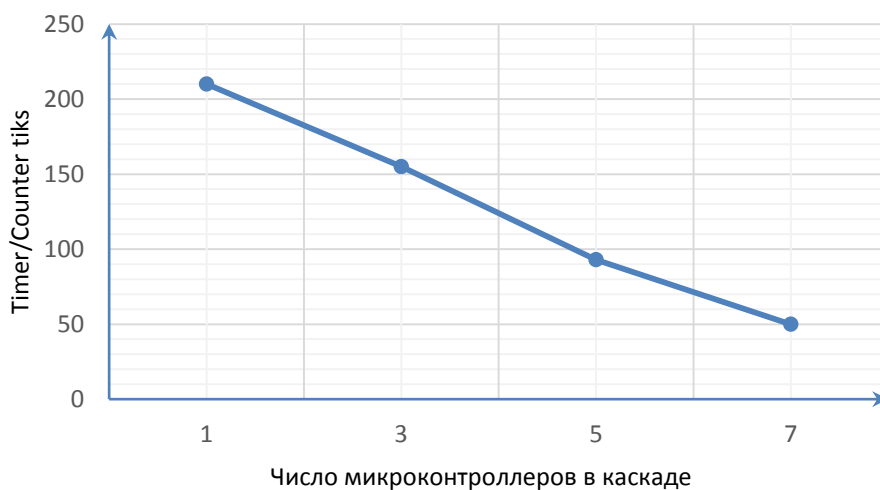


Рис. 7. Зависимость затраченного одним контроллером числа тиков внутреннего таймера от числа микроконтроллеров в схемотехническом моделировании

Метод синтеза позволяет получить каскад блоков блочной НС, который дает те же самые результаты, что и исходная монолитная НС. Далее были осуществлены замеры времени выполнения вычислений на одном микроконтроллере при каждой декомпозиции, измерялось время выполнения на последнем в каскаде вычислительном узле. Для

корректной работы таймера в контроллерах ATmega32 необходимо выставить значение предделителя, т.е. числа пропускаемых сигналов внутреннего тактового генератора, который был настроен на частоту 8 МГц. Предделитель устанавливается в регистре управления TCCR2, в битах CS20, CS21 и CS22, в наших экспериментах они были выставлены в единицы, что соответствует предделителю в 1024. Таким образом, при данном предделителе и установленной частоте один тик микроконтроллера будет составлять примерно 92 мкс. Средние результаты числа тиков таймера представлены на рис. 7.

Полученные результаты свидетельствуют о том, что нагрузка на конкретный вычислительный узел сокращается, но не пропорционально проценту оставшихся слоев НС для вычисления, а по графику, близкому к линейной зависимости, на это могло повлиять различное число нейронов на каждом из слоев нейронной сети, которая участвовала в экспериментах. Однако полученные результаты подтверждают эффективность предложенного метода.

4. Результаты исследования

В результате проделанной работы был разработан и сформулирован метод синтеза устройств реализации искусственных нейронных сетей на программируемой логике, ориентированных на туманные вычисления. Главная особенность этого метода – учет входных параметров, т.е. его можно настраивать под различные задачи.

Вторым важным результатом является тестирование нейросетевого распознавания на устройствах **на программируемой логике** и замеры результирующих параметров устройства. Они показывают, что метод работоспособен и эффективно решает задачу.

Таким образом, все частные задачи исследования решены и можно заключить, что цель исследования достигнута.

Существуют архитектуры ИНС и варианты использования блоков, которые совпадут в каскадах различных монолитных ИНС после декомпозиции, в которых последовательность блочных ИНС будет превращаться в граф вычислений, потенциально возможны даже асинхронные реализации подобных графов нейронных сетей.

В будущем возможно продолжить исследования в данном направлении, модифицировав предложенный метод синтеза таким образом,

чтобы на каждом из вычислительных узлов можно было получать дополнительные параметры и добавлять их к исходным данным для проведения нейросетевых вычислений.

Заключение

Целью исследования являлась разработка метода синтеза устройств реализации искусственных нейронных сетей на программируемой логике, ориентированных на туманные вычисления. В **результате исследования** разработан и сформулирован метода синтеза устройств реализации искусственных нейронных сетей на программируемой логике, ориентированных на туманные вычисления. Также реализована математическая модель распознавания объектов с помощью искусственных нейронных сетей устройствами на ПЛИС и микроконтроллерах, ориентированными на туманные вычисления, сформулированы необходимые алгоритмы декомпозиции НС и работы распределенной НС.

Научная новизна в представленных результатах состоит том, что разработанный метод позволяет сформировать каскад устройств нейросетевого распознавания на программируемой логике и запускать блоки нейронной сети в каскаде на физически распределенных устройствах.

Получены результаты схемотехнического моделирования каскадов устройств нейросетевого распознавания на программируемой логике. Результаты подтверждают работоспособность метода и эффективное снижение нагрузки на вычислительные узлы.

Библиографический список

1. Руднев В.А. Применение микроконтроллеров для реализации нейронных сетей // Вестник Южно-Уральского государственного университета. Сер.: Компьютерные технологии, управление, радиоэлектроника. – 2012. – № 23. – С. 181–183.
2. Quantization and deployment of deep neural networks on microcontrollers / P.E. Novac [et al.] // Sensors. – 2021. – Vol. 21, № 9. – P. 2984. DOI: 10.3390/s21092984
3. Cotton N.J., Wilamowski B.M., Dundar G. A neural network implementation on an inexpensive eight bit microcontroller // 2008 International Conference on Intelligent Engineering Systems. – 2008. – P. 109–114. DOI: 10.1109/INES.2008.4481278

4. A power efficient neural network implementation on heterogeneous FPGA and GPU devices / Y. Tu [et al.] // 2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI). – 2019. – P. 193–199. DOI: 10.1109/IRI.2019.00040

5. An application placement technique for concurrent IoT applications in edge and fog computing environments / M. Goudarzi [et al.] // IEEE Transactions on Mobile Computing. – 2020. – Vol. 20, № 4. – P. 1298–1311. DOI: 10.1109/TMC.2020.2967041

6. Бахтин В.В. Модификация алгоритма идентификации и категоризации научных терминов с использованием нейронной сети // Нейрокомпьютеры: разработка, применение. – 2019. – Т. 21, № 3. – С. 14–19.

7. Bakhtin V.V., Isaeva E.V. New TSBuilder: shifting towards cognition // 2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus). – 2019. – P. 179–181. DOI: 10.1109/EIConRus.2019.8656917

8. Bakhtin V.V., Isaeva E.V., Tararkov A.V. TSMiner: from TSBuilder to ecosystem // 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus). – 2021. – P. 221–224. DOI: 10.1109/EIConRus51938.2021.9396569

9. Yasnitsky L.N., Yasnitsky V.L. Technique of design for integrated economic and mathematical model for mass appraisal of real estate property. Study case of Yekaterinburg housing market // Journal of Applied Economic Sciences. – 2016. – Vol. 11, № 8. – P. 1519–1530.

10. Deep learning in the fog / A. Sobecki [et al.] // International Journal of Distributed Sensor Networks. – 2019. – Vol. 15, № 8. DOI: 10.1177/1550147719867072

11. Akhmetzyanov K.R., Yuzhakov A.A., Kokoulin A.N. Neural network development based on knowledge about environmental influence // 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus). – 2020. – P. 218–221. DOI: 10.1109/EIConRus49466.2020.9039226

12. Бахтин В.В., Подлесных И.А. Алгоритм построения графа совместной работы каскадов устройств нейросетевого распознавания, реализующих блочные нейронные сети // Сборник материалов IX Междунар. науч. конф., посв. 85-лет. проф. В.И. Потапова. – Омск, 2021. – С. 277–278.

13. Бахтин В.В. Математическая модель искусственной нейронной сети для устройств на ПЛИС и микроконтроллерах, ориентирован-

ных на туманные вычисления // Вестник Пермского национального исследовательского политехнического университета. Электротехника, информационные технологии, системы управления. – 2021. – № 40. – P. 109–129.

14. The construction of single-chip microcomputer virtual experiment platform based on proteus / W. Xinhuan [et al.] // 2010 5th International Conference on Computer Science & Education. – 2010. – P. 609–611. DOI: 10.1109/ICCSE.2010.5593538

15. Su B., Wang L. Application of Proteus virtual system modeling (VSM) in teaching of microcontroller // 2010 International Conference on E-Health Networking Digital Ecosystems and Technologies (EDT). – 2010. – Vol. 2. – P. 375–378. DOI: 10.1109/EDT.2010.5496343

16. Chen J., Yang S. Application of Proteus software in MCU teaching // 2011 Second International Conference on Mechanic Automation and Control Engineering. – 2011. – P. 6359–6362. DOI: 10.1109/MACE.2011.5988496

17. Mostafa G. Design of a single chip Digital Weighing Machine using ATmega32 microcontroller architecture // 2015 International Conference on Advances in Electrical Engineering (ICAEE). – 2015. – P. 93–96. DOI: 10.1109/ICAEE.2015.7506804

18. Design and realization of a single-phase inverter with numerical control based on an Atmega32 / M. El Ouariachi [et al.] // 2017 14th International Multi-Conference on Systems, Signals & Devices (SSD). – 2017. – P. 239–244. DOI: 10.1109/SSD.2017.8166982

19. PID-Neural controller based on AVR Atmega128 / X.K. Pham [et al.] // 2008 10th International Conference on Control, Automation, Robotics and Vision. – 2008. – P. 1573–1576. DOI: 10.1109/ICARCV.2008.4795759

20. Fryza T. Basic C code implementations for AVR microcontrollers // 2007 14th International Workshop on Systems, Signals and Image Processing and 6th EURASIP Conference focused on Speech and Image Processing, Multimedia Communications and Services. – 2007. – P. 434–437, DOI: 10.1109/IWSSIP.2007.4381134

21. A convolutional neural network fully implemented on fpga for embedded platforms / M. Bettoni [et al.] // 2017 New Generation of CAS (NGCAS). – 2017. – P. 49–52. DOI: 10.1109/NGCAS.2017.16

22. Huynh T.V. Deep neural network accelerator based on FPGA // 2017 4th NAFOSTED Conference on Information and Computer Science. – 2017. – P. 254–257. DOI: 10.1109/NAFOSTED.2017.8108073

23. Faggin F. Neural network hardware // IJCNN International Joint Conference on Neural Networks, 1992. – Vol. 1. – P. 153. DOI: 10.1109/IJCNN.1992.287238

24. Hong H.J. From cloud computing to fog computing: unleash the power of edge and end devices // 2017 IEEE international conference on cloud computing technology and science (CloudCom). – 2017. – P. 331–334. DOI: 10.1109/CloudCom.2017.53

25. Wadhwa H., Aron R. Fog computing with the integration of internet of things: Architecture, applications and future directions // IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCLOUD/SocialCom/SustainCom). – 2018. – P. 987–994. DOI: 10.1109/BDCLOUD.2018.00144

References

1. Rudnev V.A. Primenenie mikrokontrollerov dlia realizatsii neironnykh setei [The use of microcontrollers for the implementation of neural networks]. *Vestnik Iuzhno-Ural'skogo gosudarstvennogo universiteta. Komp'yuternye tekhnologii, upravlenie, radioelektronika*, 2012, no. 23, pp. 181-183.

2. Novac P.E. et al. Quantization and deployment of deep neural networks on microcontrollers. *Sensors*, 2021, vol. 21, no. 9, 2984 p. DOI: 10.3390/s21092984

3. Cotton N.J., Wilamowski B.M., Dundar G. A neural network implementation on an inexpensive eight bit microcontroller. *2008 International Conference on Intelligent Engineering Systems*, 2008, pp. 109-114. DOI: 10.1109/INES.2008.4481278

4. Tu Y. et al. A power efficient neural network implementation on heterogeneous FPGA and GPU devices. *2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI)*, 2019, pp. 193-199. DOI: 10.1109/IRI.2019.00040

5. Goudarzi M. et al. An application placement technique for concurrent IoT applications in edge and fog computing environments. *IEEE Trans-*

actions on Mobile Computing, 2020, vol. 20, no. 4, pp. 1298-1311. DOI: 10.1109/TMC.2020.2967041

6. Bakhtin V.V. Modifikatsiia algoritma identifikatsii i kategorizatsii nauchnykh terminov s ispol'zovaniem neironnoi seti [Modification of the algorithm for identification and categorization of scientific terms using a neural network]. *Neirokomp'iutery: razrabotka, primenenie*, 2019, vol. 21, no. 3, pp. 14-19.

7. Bakhtin V.V., Isaeva E.V. New TSBuilder: shifting towards cognition. *2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, 2019, pp. 179-181. DOI: 10.1109/EIConRus.2019.8656917

8. Bakhtin V.V., Isaeva E.V., Tararkov A.V. TSMiner: from TSBuilder to Ecosystem. *2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, 2021, pp. 221-224. DOI: 10.1109/EIConRus51938.2021.9396569

9. Yasnitsky L.N., Yasnitsky V.L. Technique of design for integrated economic and mathematical model for mass appraisal of real estate property. Study case of Yekaterinburg housing market. *Journal of Applied Economic Sciences*, 2016, vol. 11, no. 8, pp. 1519-1530.

10. Sobecki A. et al. Deep learning in the fog. *International Journal of Distributed Sensor Networks*, 2019, vol. 15, no. 8. DOI: 10.1177/1550147719867072

11. Akhmetzyanov K.R., Yuzhakov A.A., Kokoulin A.N. Neural Network Development Based on Knowledge about Environmental Influence. *2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, 2020, pp. 218-221. DOI: 10.1109/EIConRus49466.2020.9039226

12. Bakhtin V.V., Podlesnykh I.A. Algoritm postroeniia grafa sovместnoi raboty kaskadov ustroystv neirosetevogo raspoznaniia, realizuiushchikh blochnye neironnye seti [Algorithm for constructing a coworking cascades graph of neural network recognition devices implementing block neural networks]. *Sbornik materialov IX Mezhdunarodnoi nauchnoi konferentsii, posviashchennoi 85-letiiu professora V.I. Potapova*, Omsk, 2021, pp. 277-278.

13. Bakhtin V.V. Matematicheskaiia model' iskusstvennoi neironnoi seti dlia ustroystv na PLIS i mikrokontrollerakh, orientirovannykh na tumannye vychisleniia [Mathematical model of an artificial neural network

for FPGA devices and microcontrollers focused on fog computing]. *Vestnik Permskogo natsional'nogo issledovatel'skogo politekhnicheskogo universiteta. Elektrotehnika, informatsionnye tekhnologii, sistemy upravleniia*, 2021, no. 40, pp. 109-129.

14. Xinhuan W. et al. The construction of single-chip microcomputer virtual experiment platform based on proteus. *2010 5th International Conference on Computer Science & Education*, 2010, pp. 609-611. DOI: 10.1109/ICCSE.2010.5593538

15. Su B., Wang L. Application of Proteus virtual system modeling (VSM) in teaching of microcontroller. *2010 International Conference on E-Health Networking Digital Ecosystems and Technologies (EDT)*, 2010, vol. 2, pp. 375-378. DOI: 10.1109/EDT.2010.5496343

16. Chen J., Yang S. Application of Proteus software in MCU teaching. *2011 Second International Conference on Mechanic Automation and Control Engineering*, 2011, pp. 6359-6362. DOI: 10.1109/MACE.2011.5988496

17. Mostafa G. Design of a single chip Digital Weighing Machine using ATmega32 microcontroller architecture. *2015 International Conference on Advances in Electrical Engineering (ICAEE)*, 2015, pp. 93-96. DOI: 10.1109/ICAEE.2015.7506804

18. El Ouariachi M. et al. Design and realization of a single-phase inverter with numerical control based on an Atmega32. *2017 14th International Multi-Conference on Systems, Signals & Devices (SSD)*, 2017, pp. 239-244. DOI: 10.1109/SSD.2017.8166982

19. Pham X.K. et al. PID-Neural controller based on AVR Atmega128. *2008 10th International Conference on Control, Automation, Robotics and Vision*, 2008, pp. 1573-1576. DOI: 10.1109/ICARCV.2008.4795759

20. Fryza T. Basic C code implementations for AVR microcontrollers. *2007 14th International Workshop on Systems, Signals and Image Processing and 6th EURASIP Conference focused on Speech and Image Processing, Multimedia Communications and Services*. 2007. pp. 434-437, DOI: 10.1109/IWSSIP.2007.4381134

21. Bettoni M. et al. A convolutional neural network fully implemented on fpga for embedded platforms. *2017 New Generation of CAS (NGCAS)*, 2017, pp. 49-52. DOI: 10.1109/NGCAS.2017.16

22. Huynh T.V. Deep neural network accelerator based on FPGA. *2017 4th NAFOSTED Conference on Information and Computer Science*, 2017, pp. 254-257. DOI: 10.1109/NAFOSTED.2017.8108073

23. Faggin F. Neural network hardware. *IJCNN International Joint Conference on Neural Networks*, 1992, vol. 1, 153 p. DOI: 10.1109/IJCNN.1992.287238

24. Hong H.J. From cloud computing to fog computing: unleash the power of edge and end devices. *2017 IEEE international conference on cloud computing technology and science (CloudCom)*, 2017, pp. 331-334. DOI: 10.1109/CloudCom.2017.53

25. Wadhwa H., Aron R. Fog computing with the integration of internet of things: Architecture, applications and future directions. *IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom)*, 2018, pp. 987-994. DOI: 10.1109/BDCloud.2018.00144

Сведения об авторах

Тюрин Сергей Феофентович (Пермь, Россия) – заслуженный изобретатель Российской Федерации, доктор технических наук, профессор, профессор кафедры «Автоматика и телемеханика» Пермского национального исследовательского политехнического университета (614990, Пермь, Комсомольский пр., 29, e-mail: tyurinsergfeo@yandex.ru); профессор кафедры «Математическое обеспечение вычислительных систем» Пермского государственного национального исследовательского университета (614990, Пермь, ул. Букирева, 15).

Бахтин Вадим Вячеславович (Пермь, Россия) – аспирант, младший научный сотрудник кафедры «Автоматика и телемеханика» Пермского национального исследовательского политехнического университета (614990, Пермь, Комсомольский пр., 29, e-mail: bakhtin_94@bk.ru), старший преподаватель кафедры «Информационная безопасность и системы связи» Пермского государственного национального исследовательского университета (614990, Пермь, ул. Букирева, 15).

Подлесных Иван Александрович (Пермь, Россия) – аспирант кафедры «Автоматика и телемеханика» Пермского национального исследовательского политехнического университета (614990, Пермь, Комсомольский пр., 29, e-mail: podlesnihwork@gmail.com); математик ДИР ИТЦ АО «Новомет-Пермь» (614065, Пермь, ш. Космонавтов, 395).

About the authors

Sergey F. Tyurin (Perm, Russian Federation) – Honored Inventor of the Russian Federation, Doctor of Technical Sciences, Professor, Professor at the Department of Automation and Telemechanics Perm National Research Polytechnic University (614990, Perm, 29, Komsomolsky pr., e-mail: tyurinsergfeo@yandex.ru), Professor at the Department of Software Computing Systems Perm State University (614990, Perm, 15, Bukireva str.).

Vadim V. Bakhtin (Perm, Russian Federation) – Graduate Student, junior researcher of the Department of Automation and Telemechanics Perm National Research Polytechnic University (614990, Perm, 29, Komsomolsky pr., e-mail: bakhtin_94@bk.ru), Senior lecturer at the Department of Information Security and Communication Systems Perm State University (614990, Perm, 15, Bukireva str.).

Ivan A. Podlesnykh (Perm, Russian Federation) – Graduate Student of the Department of Automation and Telemechanics Perm National Research Polytechnic University (614990, Perm, 29, Komsomolsky pr., e-mail: podlesnihwork@gmail.com), Mathematician at Novomet (614065, Perm, 395, shosse Kosmonavtov).

Поступила 15.03.2022

Одобрена 28.03.2022

Принята к публикации 20.06.2022

Финансирование. Исследование проводится при поддержке РФФИ на средства гранта № 20-37-90036 Аспиранты «Метод синтеза устройств нейросетевого распознавания для реализации режима Fog computing».

Конфликт интересов. Конфликт интересов по отношению к статье отсутствует.

Вклад авторов. Все авторы внесли равный вклад в написание статьи.