

#### Список использованных источников

1. Сайт «www.sap.com» [Электронный ресурс] / Что такое интеграция приложений? Казаков, В. Е. Разработка back-end приложения «Учебно-методический отдел (учебные планы)» – Режим доступа: <https://www.sap.com/central-asia-caucasus/products/technology-platform/what-is-enterprise-integration/application-integration.html>. – Дата доступа: 05.05.2024.
2. Казаков, В. Е. Разработка back-end приложения «Учебно-методический отдел (учебные планы)» / М.С. Карнилов, В.Е. Казаков // Материалы докладов 55-й Международной научно-технической конференции преподавателей и студентов: в 2 т. / УО «ВГТУ». – Витебск, 2022. – Т. 2. – С. 5–7.
3. Плюснин, А. В. Основные принципы построения базы данных университета для полной интеграции СДО / А.В. Плюснин // Интерэкспо Гео-Сибирь. – 2020. – Т. 7. № 1. – С. 105–110.
4. Ilin, I. V. Integration of information and management technologies / I. V. Ilin Technoeconomics. – 2022. – Т. 1. № 1 (1). – С. 24–32.
5. Кузьмин, А. М. Модель изменений К. Левина / А. М. Кузьмин, Е. А. Высоковская // Методы менеджмента качества. – 2016. – № 7. – С. 21.
6. Rojas H., Arias K.A., Renteria R. 2021. Service-oriented architecture design for small and medium enterprises with infrastructure and cost optimization. Procedia Computer Science 179. – p. 488-497.
7. Измайлов, М. К. Проблемы внедрения цифровизационных инноваций в процессы производства / М. К. Измайлов // Экономика Северо-Запада: проблемы и перспективы развития. 2023. – № 2 (73). – С. 53–58.
8. Ходыкина, А. И. Управленческие инновации: понятие, сущность, проблемы внедрения в организациях / А. И. Ходыкина // Экономика и менеджмент инновационных технологий. – 2019. – № 9. – С.67–73.
9. Панько, Ю. В. Проблема сопротивления организационным изменениям в условиях внедрения инноваций на современном предприятии / Ю. В. Панько // Проблемы современной экономики: глобальный, национальный и региональный контекст. Сборник научных статей. Редакция: М. Е. Карпицкая (гл. ред.), С. Е. Витун (зам. гл. ред.) [и др.]. Гродно, 2021. – С. 135–141.
10. Смотрицкая, И. И. Организационные инновации в сфере государственного управления / И. И. Смотрицкая, С. И. Черных // Вестник Института экономики Российской академии наук. 2021. – № 1. – С. 9–25.

УДК 004.89

## МОДЕЛИРОВАНИЕ НЕЙРОННЫХ СЕТЕЙ НА ЯЗЫКЕ ОПИСАНИЯ АППАРАТУРЫ VHDL

*Добыш Д.С., студ., Черненко Д.В., ст. преп., Соколова А.С., ст. преп.  
Витебский государственный технологический университет,  
г. Витебск, Республика Беларусь*

*Реферат.* Искусственные нейронные сети часто выполняются в виде программного продукта. В статье предлагается альтернатива – представление нейронной сети на языке описания аппаратуры VHDL.

*Ключевые слова:* нейронная сеть, нейронный узел, модель нейрона, VHDL.

Модель нейронной сети состоит из узлов, объединённых в слои. Каждый из узлов одного слоя объединён со всеми узлами предыдущего слоя и получает от них сигналы. В рассматриваемой модели сигналы будут представлять собой целочисленные значения. Узел обрабатывает сигналы и передаёт их следующему слою. Обработка сигналов представляет собой вычисление суммы произведений каждого веса и соответствующего значения нейрона предыдущего слоя.

Каждый нейронный узел состоит из трех частей: компонента выполняющего функцию умножения; компонента накопления результатов умножения; регистров, в которых хранятся значения всех весов узла.

Компонент умножения выполнен в последовательном виде для сокращения количества

необходимых логических элементов как показано на рисунке 1 а. Здесь показана четырехбитная схема умножения, но при увеличении разрядности такая схема становится в разы больше.

Компонент накопления результатов умножения состоит из сумматоров и регистра. Как и в элементе умножения, D-триггеры регистра ведут запись по спаду тактового импульса. Элемент накопления выделен в отдельный компонент, т. к. он требует отдельных управляющих сигналов. Схема такого решения приведена на рисунке 1 б.

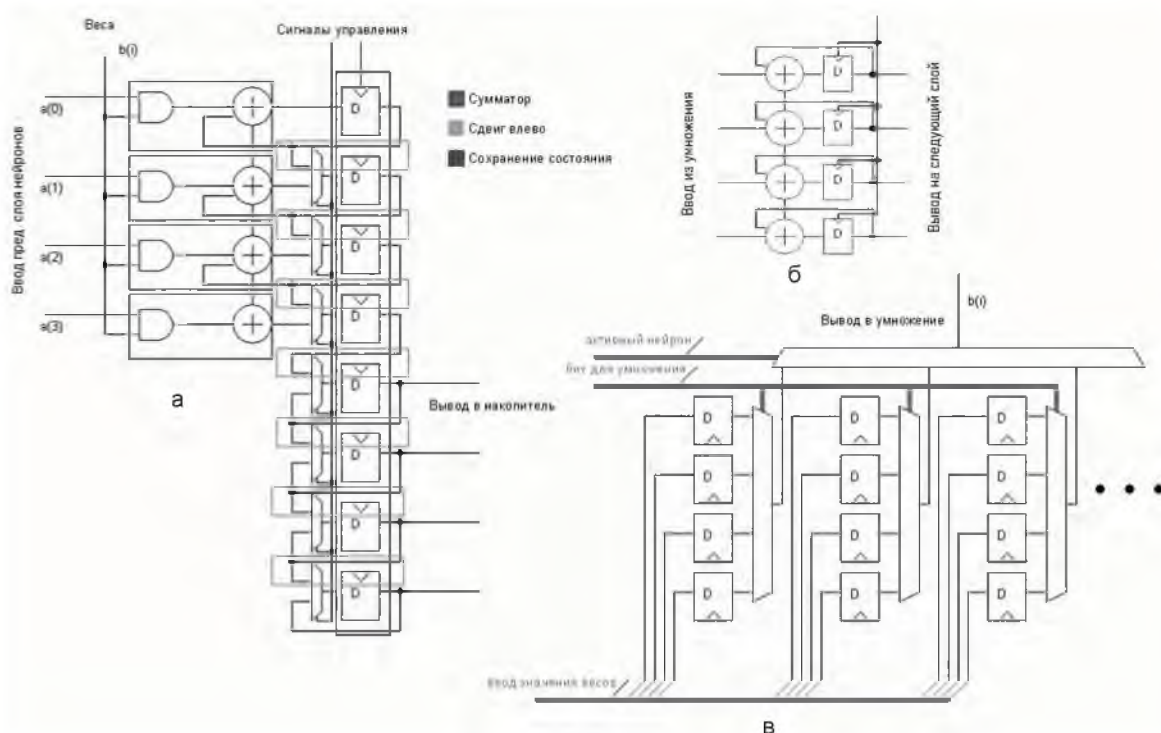


Рисунок 1 – Модели частей нейронного узла:  
 а – последовательный вид модели умножения; б – модель компонента накопления результатов умножения; в – модель массива регистров весов

Компонент умножения и накопления можно представить в виде схемы, но при использовании шестнадцатибитовых значений, схемы станут слишком громоздкими. Альтернативой является моделирование с использованием язык VHDL, где модель будет описана лишь несколькими строками кода. Например, код описания сумматоров компонента умножения для размера 16 бит выглядит следующим образом:

```

adder_array:
for i in 15 downto 0 generate
    mul_ready(i) <= w_in and neural_in(i);
    fst_addr:
    if i=0 generate
        mul_add : h_a port map (
            a => mul_ready(i), b => acc_out(i),
            co => add_carry(i), s => add_out(i)
        );
    end generate fst_addr;
    other_addr:
    if i>0 generate
        mul_add : f_a port map (
            a => mul_ready(i), b => acc_out(i), ci => add_carry(i-1),
            co => add_carry(i), s => add_out(i)
        );
    end generate other_addr;
end generate adder_array;
    
```

Для массива регистров хранения весов требуется особый подход. Количество регистров в массиве, в отличие от размерности чисел, будет разное для каждого нейрона одного слоя. Для описания такой структуры в VHDL существует оператор generic:

```

entity weights_g is
  generic (
    neuron_count : integer;
    addr_space : integer
  );
  port (
    act_neuron : in bit_vector(addr_space-1 downto 0);
    req_wbit : in bit_vector(3 downto 0);
    w_in : in bit_vector(15 downto 0);
    clk, w_en : in bit;
    w_out : out bit
  );
end weights_g;

```

Для управления всей системой используется отдельный компонент управления нейронным слоем один на всю сеть. Элемент управления нейронным слоем включает в себя: управление умножением, распределение битов весов и выборку активного нейрона для подсчёта.

Этот компонент выбирает нейроны из предыдущего слоя один за одним, указывает для каждого нейрона текущего слоя верный регистр из массива регистров весов и выполняет умножение. После выполнения умножения выбирается следующий нейрон и так далее пока не будут выбраны все нейроны предыдущего слоя. Так как нейроны текущего слоя делят источник информации, эта операция выполняется параллельно для целого слоя нейронов. Сам компонент управления

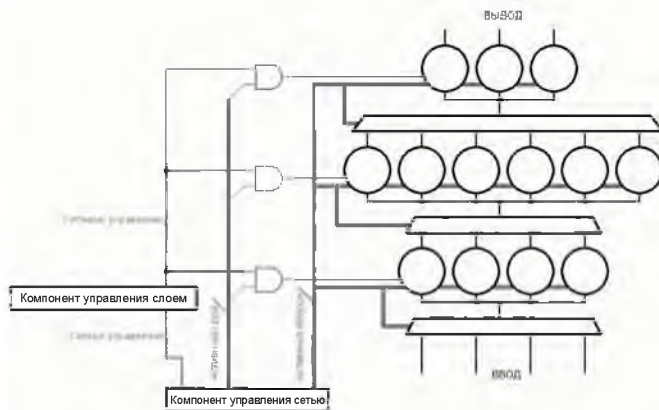


Рисунок 2 – Общая модель нейронной сети

слоя представлен в виде счётчика, который считает каждый такт с начала активации сети и на основе этого такта выдаёт сигналы управления нейронам. Как только подсчёт всех нейронов в слое завершается, управление передаётся следующему слою и так пока не будет достигнут результат.

Между слоями располагается мультиплексор, выбирающий по заданному адресу активный нейрон, к которому будет применяться операция умножения на следующем слое. Этот же адрес идёт в выборку регистра в массиве регистров следующего слоя.

Общая модель нейронной сети представлена на рисунке 2.

Приведем пример использования модели для описания сети:

```

layer0 : layer generic map (addr_space => 2, neuron_count => 4, prev_neuron_count => 4) port map (
  act_neuron => act_neuron, prev_layer => prev_layer, clk => clk,
  -- control signals
  wen => wen, ren => ren, sen > sen, gen => gen, mul_reset => mul_reset,
  req_wbit => req_wbit,
  write_in => write_in, weight_write => weight_write, nacc_en => nacc_en, nacc_reset =>
  nacc_reset
);
layer1 : layer generic map (addr_space => 3, neuron_count => 6, prev_neuron_count => 4) port map (
  act_neuron => act_neuron, prev_layer => prev_layer, clk => clk,
  -- control signals
  wen => wen, ren => ren, sen > sen, gen => gen, mul_reset => mul_reset,
  req_wbit => req_wbit,
  write_in => write_in, weight_write => weight_write, nacc_en => nacc_en, nacc_reset =>
  nacc_reset
);
layer2 : layer generic map (addr_space => 2, neuron_count => 3, prev_neuron_count => 6) port map (
  act_neuron => act_neuron, prev_layer => prev_layer, clk => clk,
  -- control signals
  wen => wen, ren => ren, sen > sen, gen => gen, mul_reset => mul_reset,
  req_wbit => req_wbit,
  write_in => write_in, weight_write => weight_write, nacc_en => nacc_en, nacc_reset =>
  nacc_reset
);

```

Достоинством предложенной модели является высокая производительность. Используя последовательный алгоритм, данная модель избегает излишней сложности реализации. Недостатком последовательного подхода является необходимость отдельных компонентов управления.

УДК 004.89

## **СИСТЕМА РАЗВЕРТЫВАНИЯ СЕРВИСА РАБОЧЕГО МЕСТА СОТРУДНИКА ПРИЕМНОЙ КОМИССИИ**

**Бизюк А.Н., ст. преп.**

*Витебский государственный технологический университет,  
г. Витебск, Республика Беларусь*

Реферат. В современном образовательном пространстве эффективное управление приемной кампанией в университете является ключевым для обеспечения высокого качества образования и привлечения лучших студентов. Предлагается инновационный подход, основанный на современных информационных технологиях: виртуализации, контейнеризации, проксировании и автоматизации с GitLab. Этот подход обеспечивает эффективное управление процессами приема, упрощает работу персонала и повышает удобство для абитуриентов, совмещая в себе высокую производительность, масштабируемость и безопасность.

Ключевые слова: приемная кампания, информационная система, университет, автоматизация, Proxmox, Docker, Caddy, безопасность данных, масштабируемость, гибкость, эффективность.

В современном образовательном пространстве эффективное управление приемной кампанией в университете является ключевым фактором для обеспечения высокого качества образования и привлечения лучших студентов. С учетом динамичного изменения правил и требований к приему абитуриентов, необходимость в современных информационных системах, способных автоматизировать и оптимизировать процессы приема, становится все более актуальной.

Целью данного исследования является предложение эффективного и инновационного подхода к управлению приемной кампанией в университете, способного улучшить процессы приема абитуриентов, повысить прозрачность и эффективность системы, а также обеспечить надежность и безопасность данных.

Информационная система для приема документов от абитуриентов во время приемной кампании в университет построена на основе базы данных и специализированного программного обеспечения. Вот общая структура такой системы:

1. База данных абитуриентов содержит информацию о каждом абитуриенте, включая персональные данные (имя, контактная информация), предыдущее образование, список предметов и оценки по ним, а также приоритеты специальностей.

2. Автоматическое формирование заявления. На основе предоставленных данных абитуриентом, система автоматически генерирует заявление для участия в конкурсе.

3. Механизм распределения абитуриентов по специальностям. Система анализирует предоставленные абитуриентами приоритеты специальностей и их суммарные баллы. На основе этих данных производится автоматическое распределение абитуриентов по специальностям с учетом предпочтений и доступных мест.

4. Учет вступительных испытаний и перераспределение. В процессе сдачи вступительных испытаний система ведет учет оценок и производит перераспределение абитуриентов в случае необходимости, основываясь на новых данных об оценках.

5. Интерфейс для администраторов и сотрудников приемной комиссии. Для управления и мониторинга приемной кампании система предоставляет удобный интерфейс для администраторов и сотрудников приемной комиссии. В этом интерфейсе можно просматривать данные абитуриентов, управлять процессом приема документов, принимать решения о распределении и прочее.