

УДК 004.42:378.147.227

Интерактивное средство поддержки изучения теории и практики работы с двоичным кодом

- Герасимов Максим** — студент бакалавриата 3-го курса направления 09.03.01 «Информатика и вычислительная техника». Научные интересы: интеллектуальные информационные системы, машинное обучение. E-mail: maxger60@gmail.com
- Зверькова Дарья Александровна** — студент бакалавриата 3-го курса направления 09.03.01 «Информатика и вычислительная техника». Научные интересы: двоичное кодирование, интеллектуальные информационные системы. E-mail: dzverkova50@gmail.com
- Карпич Эмилия Эдуардовна** — студент бакалавриата 3-го курса направления 09.03.01 «Информатика и вычислительная техника». Научные интересы: информационные системы, двоичное кодирование. E-mail: emiliya1104@gmail.com
- Гильванов Ринат Гаффанович** — канд. воен. наук, доцент кафедры «Информационные и вычислительные системы». Научные интересы: информационные системы, виртуальная и дополненная реальность, иммерсивные технологии. E-mail: gilvanov1950@mail.ru

Петербургский государственный университет путей сообщения Императора Александра I, Россия, 190031, Санкт-Петербург, Московский пр., 9

Для цитирования: Герасимов М., Зверькова Д. А., Карпич Э. Э., Гильванов Р. Г. Интерактивное средство поддержки изучения теории и практики работы с двоичным кодом // Интеллектуальные технологии на транспорте. 2026. № 1 (45). С. 51–58. DOI: 10.20295/2413-2527-2026-145-51-58

Аннотация. Современное информационное пространство тесно взаимосвязано с прогрессом цифровых технологий, базирующихся на двоичной системе счисления, что требует понимания принципов работы с бинарными кодами. Многие студенты сталкиваются со сложностями при усвоении этой темы из-за ее отвлеченности и большого объема теоретического материала. **Цель:** разработка интерактивного обучающего приложения, предназначенного для повышения эффективности изучения темы двоичного кодирования и закрепления практических навыков работы с бинарными представлениями чисел. **Методы:** использованы методы проектирования интерактивных обучающих систем, программная реализация выполнена на языке Python с использованием библиотеки PyQt5. Разработанная структура включает модуль теоретических материалов, калькулятор двоичных операций и практический тренажер с автоматической проверкой ответов. **Результаты:** создано полнофункциональное приложение, объединяющее теоретическую и практическую части изучения темы двоичного кода. В режиме калькулятора реализованы операции сложения, вычитания, умножения, деления и побитовых преобразований в прямом, обратном и дополнительном кодах. Практический модуль обеспечивает автоматическую генерацию заданий и формирует обратную связь, что способствует лучшему усвоению материала и развитию вычислительных навыков. **Практическая значимость:** разработанное приложение может использоваться в образовательных организациях при изучении дисциплин, связанных с цифровой логикой, вычислительной техникой и архитектурой ЭВМ. Программа повышает наглядность учебного процесса и обеспечивает интерактивную практику, способствующую закреплению теоретических знаний на практике.

Ключевые слова: двоичный код, разработка обучающего приложения, интерактивность, Python

2.3.5 — математическое и программное обеспечение вычислительных систем, комплексов и компьютерных сетей (технические науки)

Введение

Современная информационная сфера неразрывно связана с развитием цифровых технологий, основой которых является двоичная система счисления. Изучение устройства компьютера и цифровой техники, информатики и программирования предполагает владение принципами работы с двоичным кодом. Однако для многих обучающихся эта тема остается трудной из-за абстрактности и высокого объема теоретического материала.

Актуальной задачей, таким образом, является разработка обучающих программ, которые объединяют объяснение принципов работы с двоичным кодом и практику с автоматической обратной связью, способствующей закреплению знаний.

Цель работы — создать программный калькулятор двоичного кода и практический модуль, объединяющий теорию и автоматическую проверку результатов. Под программным калькулятором понимается интерактивное приложение с пользовательским интерфейсом, позволяющее выполнять вычисления над целыми числами в нескольких формах представления (прямой, обратный и дополнительный коды), выбирать базовую арифметическую операцию (сложение, вычитание, умножение, деление) и наблюдать пошаговое выполнение с учетом знакового бита, правил переноса и переполнения [1, 2].

Проверка знаний реализуется в практическом режиме: обучающийся решает задачу и вводит ответ; приложение автоматически вычисляет правильный результат в выбранном коде представления, сравнивает его с ответом пользователя и предоставляет разбор — побитовые преобразования, указание мест ошибок и комментарии по переполнению (при наличии). Повторная генерация исходных данных обеспечивает тренировку и закрепление навыков.

В статье кратко изложены основы двоичной системы счисления и особенности выполнения базовых арифметических операций. Кроме того, представлен методический подход к построению проверочных заданий, включающий:

- поэтапность (выбор формата кода → выбор операции → самостоятельное решение → автоматическая проверка → разбор ошибок);

- визуализацию побитовых действий и роли знакового разряда;
- вариативность исходных данных для многократной тренировки;
- формирование обратной связи, ориентированной на коррекцию типичных ошибок и осмысленное применение правил.

Такой подход направлен на формирование устойчивых навыков работы с двоичным кодом и их перенос на задачи вычислительной техники.

Теоретические основы функционирования арифметико-логического устройства

Арифметико-логическое устройство (АЛУ) — один из ключевых компонентов центрального процессора, предназначенный для выполнения базовых арифметических и логических операций [3–5]. Оно работает с машинными словами — операндами, которые представляют числовую или символьную информацию. Длина машинного слова, как правило, соответствует разрядности архитектуры — чаще всего 32 или 64 бита.

Основной задачей АЛУ является обработка данных на уровне микропроцессорных операций [6]. Данные (операнды) [7], необходимые для выполнения вычислений, поступают в устройство из регистров или оперативной памяти, после чего АЛУ производит соответствующие операции и передает результат в нужный блок системы [8, 9]. Управление процессом осуществляет блок управления, формирующий управляющие сигналы на основе декодированных машинных команд.

Функциональность АЛУ охватывает широкий спектр задач, условно делящихся на следующие группы [4, 10]:

- арифметические операции включают сложение, вычитание, вычитание модулей (короткие операции), умножение, деление, а также операции с числами с плавающей точкой, десятичную арифметику и индексные вычисления;
- логические операции охватывают побитовые преобразования (И, ИЛИ, НЕ, XOR), сравнение значений, а также манипуляции с логическими структурами данных [2];

- специальные арифметические операции, такие как нормализация, арифметический сдвиг (сдвигаются только цифровые разряды, знаковый разряд остается на месте), логический сдвиг [1].

Корректность выполнения арифметических операций в цифровой системе во многом зависит от формата представления чисел. Для представления разных форматов чисел применяют разные коды:

- прямой код — наиболее простой способ представления чисел, где старший бит отвечает за знак, а остальные — за модуль числа. Его недостаток — наличие двух нулей (+0 и -0);

- обратный код сохраняет структуру прямого кода для положительных чисел, а отрицательные представляются через побитовую инверсию. Проблема двух нулей остается;

- в дополнительном коде для представления отрицательного числа используется инверсия всех битов положительного значения с последующим прибавлением единицы. Данный формат широко применяется в современных архитектурах благодаря удобству реализации арифметических операций [1, 2].

Разнообразие задач, стоящих перед АЛУ, предопределяет наличие различных их классификаций [3, 10].

1. По характеру обрабатываемых данных и их функциональному назначению:

- операции двоичной арифметики для чисел с фиксированной точкой применяются при выполнении базовых вычислений в целочисленном представлении;

- операции двоичной (или шестнадцатеричной) арифметики для чисел с плавающей точкой характерны для научных и инженерных задач, требующих расширенного диапазона представления;

- операции десятичной арифметики используются в финансовых приложениях и других областях, где важно точное представление десятичных дробей;

- операции индексной арифметики выполняются при работе с адресами памяти, включая модификацию адресов команд и данных;

- операции специальной арифметики включают нормализацию, округление, масштабирование и другие действия, специфичные для конкретных вычислительных задач;

- логические операции реализуются на уровне побитовых преобразований и служат основой для выполнения операций сравнения, маскирования и фильтрации данных;

- операции над алфавитно-цифровыми полями обеспечивают обработку символьной информации, что особенно важно при реализации функций ввода-вывода и взаимодействии с пользователем.

2. По способу представления чисел:

- для чисел с фиксированной точкой;

- для чисел с плавающей точкой.

3. По способу действия над операндами:

- последовательные — выполняют операции по битам (менее производительные, но проще в реализации);

- параллельные — обрабатывают все биты одновременно, обеспечивая высокую скорость вычислений.

4. По характеру использования элементов и узлов:

- блочные — каждая операция реализована в виде отдельного аппаратного узла, что обеспечивает быстрое действие, но требует значительных ресурсов;

- многофункциональные — переключаются между режимами, обеспечивая экономию ресурсов при умеренной скорости.

Примером простой, но важной операции является сложение чисел с фиксированной точкой в дополнительном коде [3]. Благодаря особенностям формата операция выполняется без дополнительной коррекции, что упрощает аппаратную реализацию и снижает вероятность ошибок.

Логические операции, в свою очередь, проводятся побитово и не зависят от формата представления чисел. Они активно применяются при реализации условных переходов, фильтрации данных и других задач логической обработки [3].

Разработка симулятора арифметико-логического устройства

Приложение реализовано на Python [11–13], графический интерфейс — на PyQt5 [14, 15]. Структура — модульная: вычислительное ядро (арифметика и логика в прямом, обратном и дополнительном

кодах), модуль визуализации пошаговых операций и тренажерный модуль.

Вычислительное ядро изолировано в модуле и оперирует фиксированной разрядностью 5 бит (1 знаковый + 4 модульных). Центральным элементом вычислительного ядра является функция `calculate_process`, выполняющая кодирование входных операндов в выбранном формате представления, выбор и запуск алгоритма операции и возврат пошагового журнала вычислений, сформированного соответствующими процедурами. Для арифметических операций функция обрабатывает признак переполнения и возвращает сообщение о выходе результата за допустимый диапазон. Дополнительно реализована обработка некорректных параметров (неизвестный тип кода/операция) и ошибок кодирования.

Код функции `calculate_process`:

```
def calculate_process(d1: int, d2: int, code_type: str,
operation: str):
    enc_map = {
        "Прямой код": (from_int_direct, to_int_direct),
        "Обратный код": (from_int_ones, to_int_ones),
        "Дополнительный код": (from_int_twos, to_
int_twos)
    }
    if code_type not in enc_map:
        return «», «Ошибка: неизвестный тип кода»
    encode, decode = enc_map[code_type]
    try:
        a_bits = encode(d1)
        b_bits = encode(d2)
    except ValueError as e:
        return «», f«Ошибка кодирования: {e}»
    if operation == "Сложение":
        final, log, over = add_binary_strings(a_bits, b_bits,
code_type)
        return log, "Переполнение" if over else f"{final}
({decode(final)})"
    if operation == "Умножение":
        final, log, over = multiply_binary_strings(a_bits,
b_bits, code_type)
        return log, "Переполнение" if over else f"{final}
({decode(final)})"
```

```
if operation == "Деление":
    final, log, over = divide_binary_strings(a_bits,
b_bits, code_type)
    return log, "Переполнение" if over else f"{final}
({decode(final)})"
if operation == "Побитовое И":
    final, log = bitwise_and_strings(a_bits, b_bits,
code_type)
    return log, f"{final} ({int(final, 2)})"
if operation == "Побитовое ИЛИ":
    final, log = bitwise_or_strings(a_bits, b_bits, code_
type)
    return log, f"{final} ({int(final, 2)})"
if operation == «Побитовое ИСКЛЮЧАЮЩЕЕ
ИЛИ»:
    final, log = bitwise_xor_strings(a_bits, b_bits,
code_type)
    return log, f"{final} ({int(final, 2)})"
return «», «Ошибка: неизвестная операция»
```

Поддерживаются три представления чисел: прямой, обратный и дополнительный коды. Рассмотрены парные функции кодирования/декодирования, валидация длины битовых строк и контроль диапазонов. Общая рабочая область значений ограничена интервалом $[-15; 15]$ для предотвращения переполнения.

Валидация включает проверки разрядности, корректности кодов, границ диапазонов, а также специальных ситуаций (деление на ноль). Все операции возвращают как машиночитаемый результат, так и развернутый текстовый журнал, что облегчает применение в учебных целях и разбор ошибок. Реализация симулятора АЛУ представлена в виде программного решения с пользовательским интерфейсом, обеспечивающим интерактивное обучение и самопроверку. Таким образом, одной из отличительных особенностей разработанного приложения является реализация практического режима, направленного на закрепление теоретических знаний студентов.

Режим калькулятора представляет собой инструмент для проведения вычислений двух целых чисел в трех системах кодирования. Режим практики представлен в виде интуитивно понятного

графического интерфейса (рис. 1), который позволяет пользователю взаимодействовать с заданием.

В режиме калькулятора поддерживается выбор наиболее распространенных форм представления чисел в цифровых устройствах: прямой код, обратный код, дополнительный код. Каждый из перечисленных форматов имеет свою особенность обращения с положительными и отрицательными числами, особенно при выполнении операций сложения и умножения.

Пользователь имеет возможность выбрать тип арифметической операции. В текущей версии приложения поддерживаются пять арифметических операций: сложение, умножение, побитовые И, ИЛИ, ИСКЛЮЧАЮЩЕЕ ИЛИ.

Для примера работы калькулятора будет произведено суммирование чисел 2 и 3 в прямом коде в режиме калькулятора.

В поле ввода «Регистр 1» вводим цифру 3, в поле «Регистр 2» вводим цифру 2. Далее необходимо выбрать кодировку и операцию между регистрами.

При нажатии «Получить ответ» пользователь может получить подробный результат выполнения

операции. Программа автоматически генерирует пошаговое решение с итоговым ответом (рис. 2).

В случае затруднений пользователю доступна пошаговая инструкция к использованию калькулятора при нажатии кнопки «Помощь».

В практическом режиме пользователю необходимо решить 6 заданий, связанных с двоичным кодированием, за отведенное время, где условия заданий генерируются программой автоматически (значения регистров, операция, кодировка). После выполнения всех заданий выводится итоговое окно с результатами практики (рис. 3).

В случае неверного ответа пользователю будет дана возможность просмотреть верное решение задания.

Заключение

Разработка интерактивного обучающего приложения для изучения двоичного кодирования представляет собой перспективное средство поддержки учебного процесса. Программа сочетает в себе теоретический материал, программный калькулятор, а также модуль практики с возможностью проверки

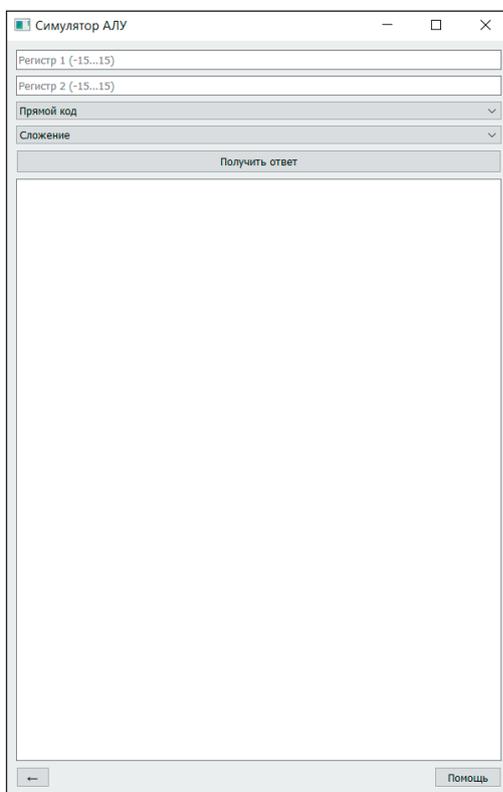


Рис. 1. Окно калькулятора

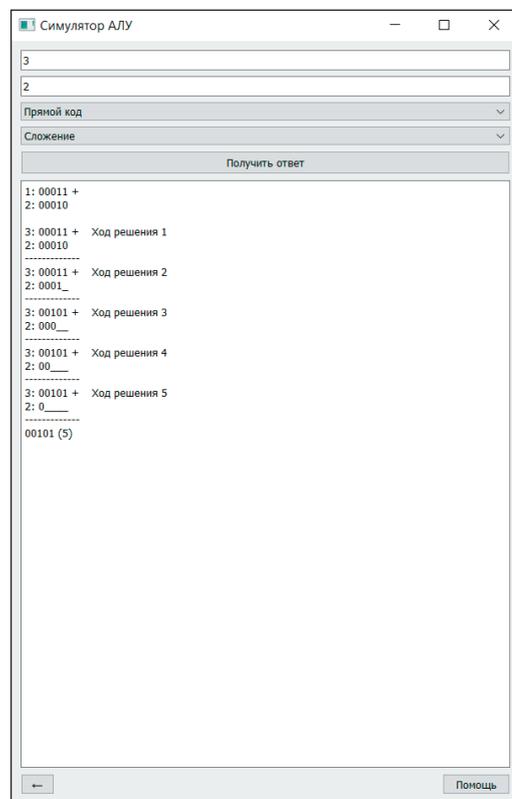


Рис. 2. Пример полученного ответа

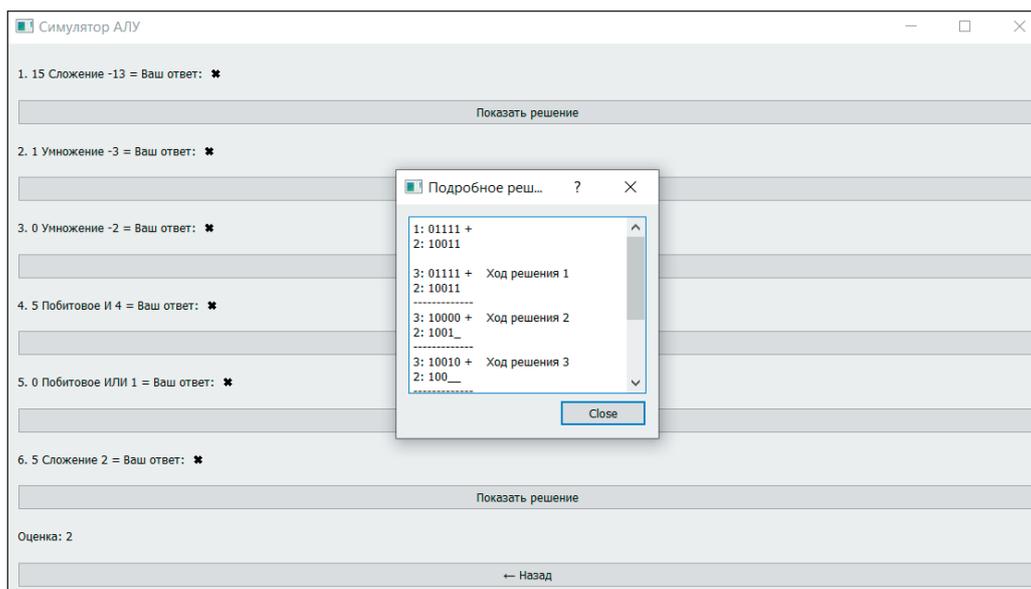


Рис. 3. Пример вывода окна с итогами практики

ответов. Комплексный подход ориентирован на развитие у студентов не только теоретических знаний, но и практических умений, что особенно важно для формирования прочных и полных представлений о принципах работы цифровых устройств.

Ожидается, что использование приложения в учебной среде будет способствовать улучшению

усвоения материала и формированию качественных навыков работы с двоичным кодом. В будущем планируются расширение возможностей, проведение эксперимента и интеграция программного продукта в учебные курсы для подтверждения его эффективности и получения методических рекомендаций по его использованию.

СПИСОК ИСТОЧНИКОВ

1. Кодирование информации (двоичные коды): справочник / под ред. Н. Т. Березюка. Харьков: Вища школа, 1978. 252 с.
2. Чернецова Е. А. Теория информации и кодирования. Практикум. СПб.: Российский гос. гидрометеорологический ун-т, 2021. 172 с.
3. Каган Б. М. Электронные вычислительные машины и системы: учебное пособие для вузов. 3-е изд., перераб. и доп. М.: Энергоатомиздат, 1991. 592 с.
4. Арифметико-логическое устройство // Желтов К. Ю. Проектирование процессоров RISC-V в среде Logisim и на языке Verilog в среде Quartus. URL: <http://riscv-alliance.ru/material/proektirovanie-procprocessorov-risc-v-v-srede-logisim-i-na-yazyke-verilog-v-srede-quartus> (дата обращения: 05.04.2025).
5. Максимов Н. В., Партыка Т. Л., Попов И. И. Архитектура ЭВМ и вычислительных систем: учебник. 5-е изд., перераб. и доп. М.: Форум, 2013. 510 с.
6. Луцик Ю. А., Лукьянова И. В. Арифметические и логические основы вычислительной техники: учебное пособие. Минск: Белорусский гос. ун-т информатики и радиоэлектроники, 2014. 174 с.
7. Ситников С. Ю., Ситников Ю. К. Электронные вычислительные машины: Арифметика. Логика. Элементарная база: учебное пособие. Казань: Казанский гос. энергетический ун-т, 2015. 168 с.
8. Карпов А. В., Калабанов С. А., Ишмуратов Р. А. Основы цифровой электроники: учебное пособие. Казань: Изд-во Казанского ун-та, 2019. 75 с.
9. Харрис С. Л., Харрис Д. М. Цифровая схемотехника и архитектура компьютера: RISC-V = Digital Design and Computer Architecture: RISC-V Edition / пер. с англ. под ред. А. Ю. Романова. М.: ДМК Пресс, 2021. 810 с.
10. Одинец А. И. Цифровые устройства: конспект лекций. Омск: Омский гос. техн. ун-т, 2009. 64 с.

11. Сечина Г.П. Микропроцессорные средства: учебно-методическое пособие. Нижнекамск: Нижнекамский химико-технологический институт, 2012. 60 с.
12. Python Documentation. URL: <http://docs.python.org> (дата обращения: 15.04.2025).
13. PyCharm Quick start guide. URL: <http://www.jetbrains.com/help/pycharm/quick-start-guide.html> (дата обращения: 15.04.2025).
14. Шаманов А.П. Системы счисления и представление чисел в ЭВМ: учебное пособие. Екатеринбург: Изд-во Уральского ун-та, 2016. 52 с.
15. Qt for Python Documentation. URL: <http://doc.qt.io/archives/qtforpython-5> (дата обращения: 15.04.2025).

Дата поступления: 28.10.2025

Решение о публикации: 16.02.2026

An Interactive Tool to Support the Study of the Theory and Practice of Working with Binary Code

- Maksim Gerasimov** — 3th year Bachelor's Degree in 09.03.01 Informatics and Computer Technology. Research interests: intellectual information systems, machine learning. E-mail: maxger60@gmail.com
- Darya A. Zverkova** — 3th year Bachelor's Degree in 09.03.01 Informatics and Computer Technology. Research interests: binary coding, intellectual information systems. E-mail: dzverkova50@gmail.com
- Emiliya E. Karpich** — 3th year Bachelor's Degree in 09.03.01 Informatics and Computer Technology. Research interests: information systems, binary coding. E-mail: emiliya1104@gmail.com
- Rinat G. Gilvanov** — PhD in Military Sciences, Associate Professor of the Information and Computing Systems Department. Research interests: information systems, virtual and augmented reality, immersive technologies. E-mail: gilvanov1950@mail.ru

Emperor Alexander I St. Petersburg State Transport University, 9, Moskovsky ave., Saint Petersburg, 190031, Russia

For citation: Gerasimov M., Zverkova D.A., Karpich E. E., Gilvanov R. G. An Interactive Tool to Support the Study of the Theory and Practice of Working with Binary Code, *Intellectual Technologies on Transport*, 2026, no. 1 (45), pp. 51–58. DOI: 10.20295/2413-2527-2026-145-51-58. (In Russian)

Abstract. *The modern information environment is tightly linked to the advancement of digital technologies based on the binary numeral system, thereby requiring a solid understanding of binary coding principles. Many learners find this subject challenging because of its abstract character and the substantial theoretical content it entails. **Purpose:** the present work seeks to create an interactive educational application aimed at improving the effectiveness of instruction on binary coding and at strengthening students' practical abilities in handling binary representations of numbers. **Methods:** the research employs methods of interactive educational system design, with the software implemented in Python using the PyQt5 framework. The resulting architecture comprises a theoretical reference module, a binary operation calculator, and a practical training mode with automated answer verification. **Results:** a fully functional application has been created, integrating both theoretical and practical aspects of studying binary coding. The calculator mode implements operations for addition, subtraction, multiplication, division, and bitwise transformations in direct, inverse, and complementary codes. The practical module ensures the automatic generation of tasks and provides formative feedback, which enhances*

material retention and the development of computational skills. **Practical significance:** the developed application can be used in educational institutions for teaching disciplines related to digital logic, computing technology, and computer architecture. The programme enhances the visibility of the educational process and provides interactive practice that aids in the reinforcement of theoretical knowledge through practical application.

Keywords: binary code, learning application development, interactivity, Python

REFERENCES

1. Berezyuk N. T. (ed.) Kodirovanie informatsii (dvoichnye kody): spravochnik [Encoding Information (Binary Codes): A Reference Guide]. Kharkov, Vishcha Shkola Publishing House, 1978, 252 p. (In Russian)
2. Chernetsova E. A. Teoriya informatsii i kodirovaniya. Praktikum [Information and Coding Theory. Practical Exercises]. Saint Petersburg, Russian State Hydrometeorological University, 2021, 172 p. (In Russian)
3. Kagan B. M. Elektronnye vychislitelnye mashiny i sistemy: uchebnoe posobie dlya vuzov [Electronic Computing Machines and Systems: Educational Manual for Higher Education Institutions]. Moscow, Energoatomizdat Publishing House, 1991, 592 p. (In Russian)
4. Arifmetiko-logicheskoe ustroystvo [Arithmetic-Logic Unit]. In: Zheltov K. Yu. *Proektirovanie protsessorov RISC-V v srede Logisim i na yazyke Verilog v srede Quartus [Designing RISC-V Processors in the Logisim Environment and in Verilog Using Quartus]*. Available at: <http://riscv-alliance.ru/material/proektirovanie-procprocessorov-risc-v-v-srede-logisim-i-na-yazyke-verilog-v-srede-quartus> (accessed: April 15, 2025). (In Russian)
5. Maksimov N. V., Partyka T. L., Popov I. I. Arkhitektura EVM i vychislitelnykh sistem: uchebnyk [Architecture of Electronic Computing Machines and Systems: Textbook]. Moscow, Forum Publishing House, 2013, 510 p. (In Russian)
6. Lutsik Yu. A., Lukyanova I. V. Arifmeticheskie i logicheskie osnovy vychislitelnoy tekhniki: uchebnoe posobie [Arithmetic and Logical Foundations of Computer Engineering: Educational Manual]. Minsk, Belarusian State University of Informatics and Radioelectronics, 2014, 174 p. (In Russian)
7. Sitnikov S. Yu., Sitnikov Yu. K. Elektronnye vychislitelnye mashiny: Arifmetika. Logika. Elementnaya baza: uchebnoe posobie [Electronic Computing Machines: Arithmetic, Logic, and Element Basis: Educational Manual]. Kazan, Kazan State Power Engineering University, 2015, 168 p. (In Russian)
8. Karpov A. V., Kalabanov S. A., Ishmuratov R. A. Osnovy tsifrovoy elektroniki: uchebnoe posobie [Fundamentals of Digital Electronics: Educational Manual]. Kazan, Publishing House of the Kazan Federal University, 2019, 75 p. (In Russian)
9. Harris S. L., Harris D. M. Tsifrovaya skhemotekhnika i arkhitektura kompyutera: RISC-V [Digital Design and Computer Architecture: RISC-V Edition], Moscow, DMK Press Publishing House, 2021, 810 p. (In Russian)
10. Odinets A. I. Tsifrovye ustroystva: konspekt lektsiy [Digital Devices: Lecture Notes]. Omsk: Omsk State Technical University, 2009, 64 p. (In Russian)
11. Sechina G. P. Mikroprotsessornye sredstva: uchebno-metodicheskoe posobie [Microprocessor Tools: Educational and Methodological Manual]. Nizhnekamsk, Nizhnekamsk Chemical-Technological Institute, 2012, 60 p. (In Russian)
12. Python Documentation. Available at: <http://docs.python.org> (accessed: April 15, 2025).
13. PyCharm Quick start guide. Available at: <http://www.jetbrains.com/help/pycharm/quick-start-guide.html> (accessed: April 15, 2025).
14. Shamanov A. P. Sistemy schisleniya i predstavlenie chisel v EVM: uchebnoe posobie [Number Systems and Representation of Numbers in Electronic Computing Machines: Educational and Methodological Manual]. Yekaterinburg, Publishing House of the Ural Federal University, 2016, 52 p. (In Russian)
15. Qt for Python Documentation. Available at: <http://doc.qt.io/archives/qtforpython-5> (accessed: April 15, 2025).

Received: 28.10.2025

Accepted: 16.02.2026