DOI 10.31392/UDU-nc.series2.2024.23(30).10
УДК 37.091.33:004.942

Заєць Олександр Юрійович
аспірант IV року навчання спеціальності 011 Освітні / Педагогічні науки
Українського державного університету імені Михайла Драгоманова, м. Київ, Україна
ORCID ID: 0009-0006-1225-1677
o.yu.zaiets@npu.edu.ua

МЕТОДИЧНІ ПІДХОДИ ДО МОДЕЛЮВАННЯ 8-БІТНИХ МІКРОКОНТРОЛЕРНИХ ПРИСТРОЇВ У ПРОГРАМНОМУ СЕРЕДОВИЩІ PROTEUS

Анотація. На сьогоднішній день мікроконтролери – невід'ємна частина сучасного життя, оскільки їх застосування має широкий спектр пристроїв та систем: від побутової техніки до складних механізмів промисловості. Постійна необхідність мікроконтролерів вимагає ефективних підходів до їх розробки та моделювання. У статті висвітлено сутність методичних підходів до моделювання 8-бітних мікроконтролерних пристроїв у програмному середовищі Proteus у підготовці майбутніх ІТ-фахівців. Розглядаються етапи моделювання, включаючи розробку схеми, написання та налагодження програмного коду, а також верифікацію і тестування системи. Результати дослідження підтверджують високу ефективність програмного середовища Proteus для моделювання 8-бітних мікроконтролерних систем, що сприяє підвищенню якості навчання та розробки у галузі електроніки та мікропроцесорної техніки. Відзначено, що мікроконтролери продовжують відігравати ключову роль у розвитку технологій, покращуючи якість життя та сприяючи інноваціям у різних сферах. Актуальність даної теми обумовлена необхідністю підвищення ефективності розробки та тестування мікроконтролерних систем, зменшення витрат на прототипування та покращення якості кінцевих продуктів. Використання програмного середовища Proteus дозволяє досягти цих цілей, надаючи розробникам інструменти для точного моделювання і верифікації електронних систем. Таким чином метою даного дослідження є розроблення методичних підходів моделювання 8-бітних пристроїв, створення точних програмно-апаратних моделей мікроконтролерів, оптимізацію алгоритмів тестування та верифікації систем на основі симуляцій, а також підвищення швидкості та якості процесу проєктування. Це дасть можливість зменшити кількість помилок на етапі розроблення, скоротити час на налагодження прототипів і забезпечити високу надійність кінцевих електронних виробів.

Ключові слова: комп'ютерне моделювання, програмне середовище Proteus, AVR, симуляція апаратної частини, програмне забезпечення, розробка схем, налагодження коду, верифікація системи, прототипування, мікропроцесорна техніка, навчальний процес, інновації, технології.

Вступ. В наш час комп'ютерне моделювання, як метод пізнання об'єктів і явищ потужно використовується для розвитку навчальної діяльності студентів [1]. Проблема ефективного застосування комп'ютерного моделювання в навчальному процесі підготовки фахівців у ІТ-галузі є актуальною для педагогічної науки [2]. Дана стаття присвячена методичним підходам до моделювання 8-бітних мікроконтролерних пристроїв у програмному середовищі Proteus, що має значення, як для наукових досліджень, так і для практичної інженерної діяльності. Було розглянуто основні етапи моделювання, включаючи розробку схеми, написання та налагодження програмного коду, верифікацію та тестування системи. Особлива увага приділяється використанню Proteus в освітньому процесі, що сприяє підготовці висококваліфікованих фахівців у галузі електроніки та мікропроцесорної техніки.

Метою написання статті є розроблення методичних підходів до моделювання 8-бітних пристроїв, створення точних програмно-апаратних моделей мікроконтролерів, оптимізацію алгоритмів тестування та верифікації систем на основі симуляцій, а також підвищення швидкості та якості процесу проєктування.

Подання основного матеріалу дослідження. 8-бітні мікроконтролери представляють собою компактні, енергоефективні пристрої, що використовуються для управління різними електронними системами. Вони мають обмежену обчислювальну потужність у порівнянні з більш потужними мікроконтролерами, такими як 16-бітні або 32-бітні, але їх простота і низька вартість роблять їх ідеальними для багатьох застосувань.

Найпоширенішими 8-бітними мікроконтролерами є сімейства AVR від Atmel [3] (нині Microchip Technology), PIC від Microchip Technology, а також 8051 від Intel. Вони широко використовуються в закладах освіти для навчальних цілей, у побутовій техніці, автомобільних системах, медичних пристроях та багатьох інших сферах.

8-бітні мікроконтролери мають такі основні характеристики:

- *Розрядність*. Опрацьовує дані довжиною 8 біт за одну операцію. Це містить обсяг даних, які можуть бути опрацьовані або передані за раз.
- *Архітектура*. Простота архітектури робить 8-бітні мікроконтролери легкими у вивченні та використанні. Вони часто використовують Гарвардську архітектуру, за якої розділяється пам'ять програм і даних, що дає можливість одночасного доступ до обох (програм та даних). Також використовується і архітектура фон Неймана, за якої пам'ять є єдиною для програм та даних.

У мікроконтролерах 8-бітної архітектури використовують два основних підходи до команди організації набору: RISC (Computer Instruction Set Computer) і CISC (Complex Instruction Set Computer).

RISC (Комп'ютер зі скороченим набором інструкцій) або RISC-архів, що орієнтований на зменшення кількості та складності команд, за допомогою якого забезпечується висока швидкість шляхом простого опрацювання інструкцій. Основні характеристики: прості команди (кожна команда виконується за один або кілька тактових циклів; команди мають однакову довжину, що спрощує декодування), орієнтування на швидкість (менше апаратної складності дозволяє підвищити швидкість виконання), використання реєстрів (велика кількість реєстрів (для зберігання проміжних даних і обчислень) мінімізує звернення. Тобто набір інструкцій мінімалістичний, містить тільки базові команди. Інструкції забезпечуються для одного або кількох тактів процесора. Оптимізація роботи ядра і підвищена продуктивність забезпечується шляхом простоти команд.

CISC – це структура мікроконтролерів і мікропроцесорів, яка використовує *складний набір інструкцій* для виконання завдань. Ключова ідея – зменшити кількість інструкцій у програмі шляхом реалізації більш складних операцій в одній команді. Особливості CISC полягають у *складному наборі команд* (можна виконати кілька дій за одну операцію (наприклад, завантаження, обчислення й збереження результату в пам'ять), де типовий набір інструкцій включає як базові, так і спеціалізовані команди*), великій довжині інструкцій* (команди можуть мати різну довжину (зазвичай більше 1 байта), що дозволяє реалізовувати більш складні функції) та *гнучкій адресації* (багатий вибір режимів адресації (прямий, непрямий, індексований тощо), що спрощує доступ до пам'яті).

- *Обмежена пам'ять*. Зазвичай, ці мікроконтролери мають до 64 кБ флеш-пам'яті для зберігання програм та до 4 кБ ОЗП для даних.
- *Енергоефективність*. Завдяки низькому енергоспоживанню, вони ідеально підходять для автономних пристроїв, що працюють від елементів живлення.
- *Інтерфейси введення/виведення*. 8-бітні мікроконтролери мають численні порти введення/виведення, що дає можливість їм управляти різними сенсорами, дисплеями, моторами та іншими пристроями.
- *Інтегровані периферійні пристрої*. Вони часто включають такі периферійні пристрої, як аналогово-цифрові перетворювачі (ADC), таймери, PWM-контролери, UART, SPI і I²C інтерфейси.

Переваги та недоліки 8-бітних мікроконтролерів у порівнянні з іншими типами:

Переваги:

1. *Низька вартість*. 8-бітні мікроконтролери є економічно вигідними для масового виробництва.

2. *Простота використання*. Вони мають простішу архітектуру і легше програмуються, що робить їх популярними для навчання та швидкого прототипування.

3. *Енергоефективність*. Завдяки низькому енергоспоживанню, вони ідеально підходять для автономних пристроїв, що працюють від батарейок.

4. *Компактність*. Малий розмір дозволяє використовувати їх у компактних пристроях з обмеженим простором.

Недоліки:

1. *Обмежена обчислювальна потужність*. Вони не підходять для складних обчислювальних задач, які потребують високої продуктивності.

2. *Малий обсяг пам'яті*. Обмежений обсяг пам'яті може бути недостатнім для складних програм.

3. *Обмежена функціональність*. Вони не мають таких розширених функцій, як більш потужні 16-бітні або 32-бітні мікроконтролери.

4. *Менша гнучкість*. 8-бітні мікроконтролери менш гнучкі у порівнянні з більш потужними моделями, що може обмежувати їх використання в деяких випадках.

Таким чином, 8-бітні мікроконтролери залишаються популярними завдяки своїм численним перевагам, незважаючи на деякі обмеження. Вони є вдалим рішенням для простих та енергоефективних пристроїв, що потребують невеликої кількості обчислювальної потужності.

Proteus є потужним програмним інструментом для моделювання електронних схем і систем, який надає розробникам широкий набір можливостей для створення і тестування електронних пристроїв [4], [5]. Однією з основних функцій є схемотехнічне моделювання (Schematic Capture), що дає можливість створювати схеми електронних пристроїв з використанням бібліотек компонентів. Ці бібліотеки містять широкий спектр електронних елементів, таких як резистори, конденсатори, транзистори, мікроконтролери, датчики тощо. Інша важлива функція – це симуляція, що забезпечує динамічну симуляцію роботи електронних схем. Це дає можливість розробникам перевіряти правильність роботи схем без фізичного створення прототипів.

Додатково, Proteus пропонує вбудоване моделювання мікроконтролерів, включаючи моделі багатьох мікроконтролерів, таких як AVR, PIC, ARM та інші. Це дає можливість програмувати і тестувати прошивки безпосередньо в середовищі Proteus. Програмне середовище також включає величезну бібліотеку електронних компонентів з можливістю їх налаштування, де користувачі можуть додавати власні компоненти і моделі. Крім того, з використанням Proteus можна створювати друковані плати (PCB), що є важливим кроком у розробці електронних пристроїв.

Використання Proteus надає розробникам можливості для комплексного моделювання як апаратної, так і програмної частини систем. Моделювання апаратної частини включає створення детальних схем з усіма необхідними компонентами, симуляцію роботи схем для перевірки коректності роботи в реальному часі, а також аналіз параметрів, таких як напруга, струм, потужність тощо. Програмне середовище також підтримує моделювання сенсорів та актуаторів, що дає можливість створювати більш реалістичні моделі електронних систем.

Моделювання програмної частини в Proteus включає інтеграцію з популярними компіляторами для мікроконтролерів, такими як MPLAB та AVR-GCC, що дозволяє завантажувати і тестувати прошивки безпосередньо в середовищі Proteus. Інструменти налагодження допомагають знаходити і виправляти помилки в програмному коді, а симуляція взаємозв'язку апаратної і програмної частини системи в реальному часі дозволяє перевіряти їх сумісність і коректну роботу.

Однією з головних переваг використання Proteus для моделювання мікроконтролерів є інтегроване середовище розроблення, що дає можливість моделювати і тестувати як апаратну, так і програмну частину в єдиному середовищі, що значно спрощує процес розроблення. Завдяки можливості симуляції, розробники можуть виявляти і виправляти помилки на етапі проєктування, що знижує витрати на фізичне прототипування і час на розроблення. Наявність великої кількості готових моделей компонентів дозволяє швидко створювати і тестувати схеми, а підтримка симуляції багатьох популярних мікроконтролерів робить Proteus універсальним інструментом для розроблення різноманітних електронних пристроїв.

Етапи моделювання: вибір мікроконтролера, розробка схеми, написання та налагодження програмного коду

1. *Вибір мікроконтролера.* У середовищі Proteus доступні різні моделі 8-бітних мікроконтролерів, такі як AVR, PIC та інші. Вибір мікроконтролера залежить від вимог проєкту, таких як кількість входів/виходів, обсяг пам'яті та інші параметри (рис. 1).

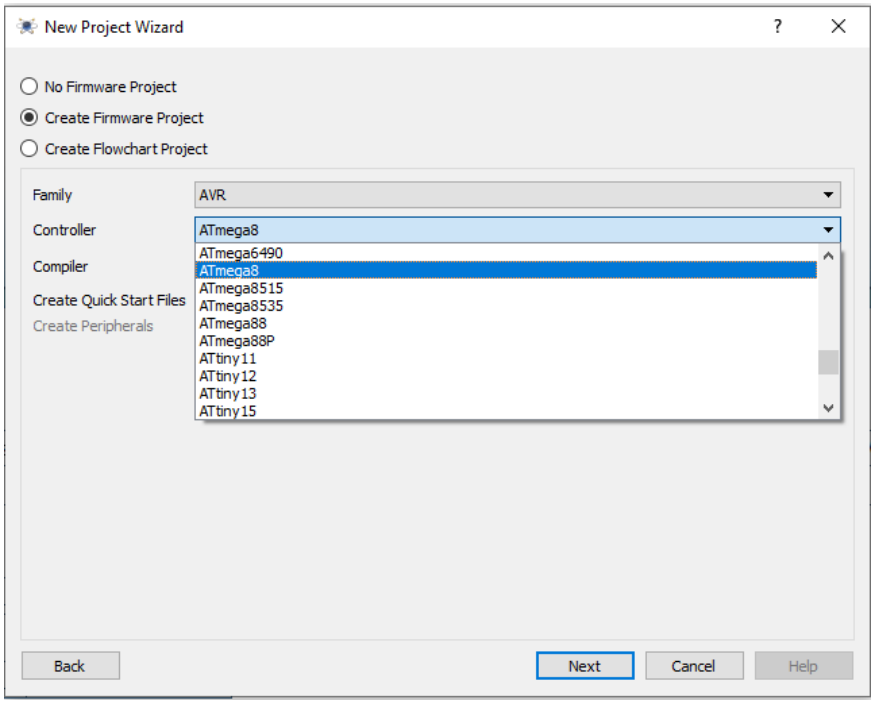

Рис. 1. Вікно створення симуляційного мікроконтролера

2. *Створення схеми*. На основі обраного мікроконтролера створюється схема, яка включає всі необхідні компоненти. Це можуть бути резистори, конденсатори, датчики, світлодіоди та інші елементи. Схема створюється у графічному середовищі ISIS [6], де компоненти розміщуються та з'єднуються між собою (рис. 2).

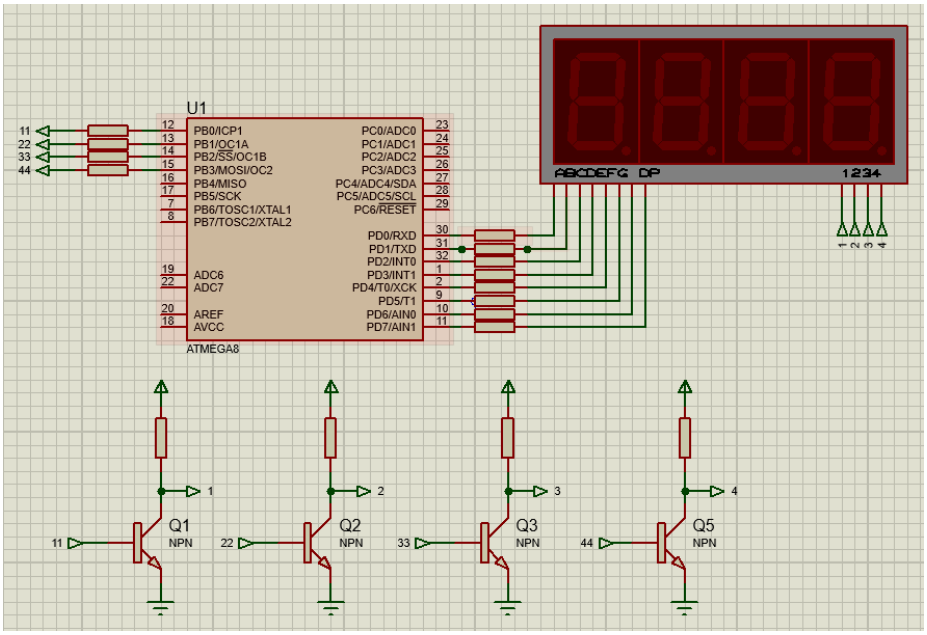

Рис. 2. Вигляд схеми підключення 7-сегментного індикатора в середовищі ISIS

3. *Налаштування параметрів*. Встановлюються параметри мікроконтролера, такі як тактова частота, напруга живлення та інші. Ці параметри визначають робочі умови мікроконтролера та впливають на його продуктивність (рис. 3).

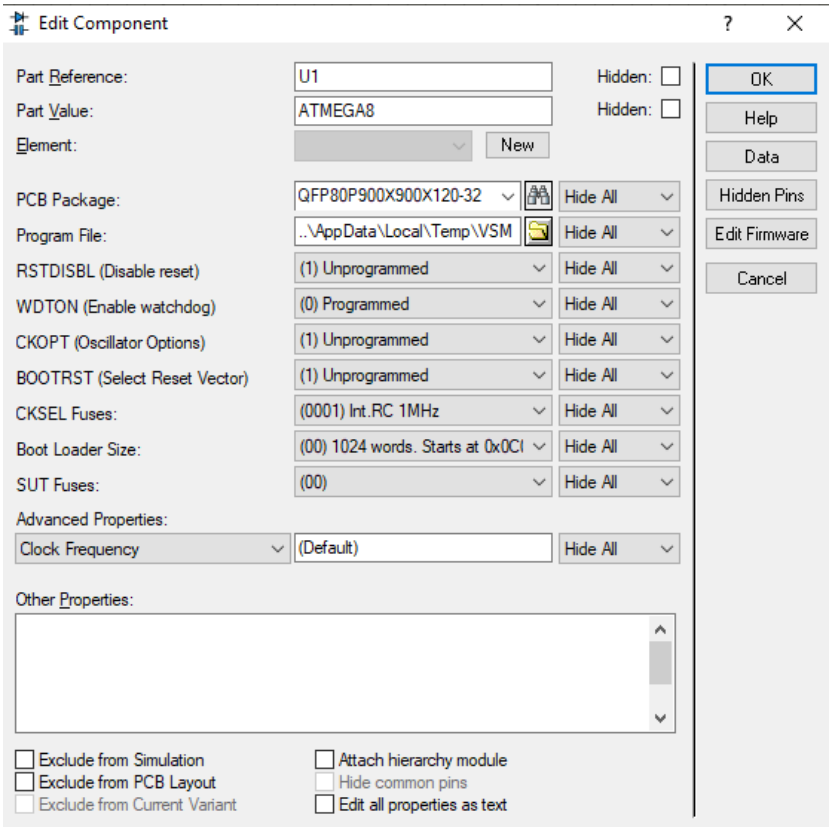

Рис. 3. Вікно налаштування параметрів мікроконтролера ATMega8

main.c

```c
#define F_CPU 1000000L
#include <avr/io.h>
#include <util/delay.h>

#define NUMBER PORTD
#define DIGIT PORTB

unsigned int numbers[10] = {
    0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x07, 0x7f, 0x6f
};

// Просте реалізація лінійного конгруентного генератора
unsigned int lcg_seed = 12345;

unsigned int lcg_rand() {
    lcg_seed = (1103515245 * lcg_seed + 12345) % 32768;
    return lcg_seed;
}

void split_number(unsigned int input_number, unsigned int *digit1, unsigned int *digit2, unsigned int *digit3, unsigned int *digit4) {
    *digit1 = input_number / 1000;       // тисячі
    *digit2 = input_number % 1000 / 100; // сотні
    *digit3 = input_number % 100 / 10;   // десятки
    *digit4 = input_number % 10;         // одиниці
}

int main(void) {
    DDRB = 0b00001111; // Встановлюємо перші 4 пини порту B як виходи
    DDRD = 0b11111111; // Встановлюємо всі пини порту D як виходи

    unsigned int digit1, digit2, digit3, digit4;
    unsigned int random_number;

    while (1) {
        random_number = lcg_rand() % 10000; // Генеруємо випадкове число від 0 до 9999
        split_number(random_number, &digit1, &digit2, &digit3, &digit4);

        // Відображаємо цифри на індикаторі
        DIGIT = 0b00000001; // Включаємо 1-й розряд
        NUMBER = numbers[digit1]; // Відображаємо 1-шу цифру
        _delay_ms(300);

        DIGIT = 0b00000010; // Включаємо 2-й розряд
        NUMBER = numbers[digit2]; // Відображаємо 2-гу цифру
        _delay_ms(300);

        DIGIT = 0b00000100; // Включаємо 3-й розряд
        NUMBER = numbers[digit3]; // Відображаємо 3-ю цифру
        _delay_ms(300);

        DIGIT = 0b00001000; // Включаємо 4-й розряд
        NUMBER = numbers[digit4]; // Відображаємо 4-ту цифру
        _delay_ms(300);
    }
}
```

Рис. 4. Вікно для введення програмного коду

4. *Написання коду та компіляція*. Розробляється програмний код для мікроконтролера, який завантажується у середовище Proteus. Код може бути написаний на мовах програмування, таких як C або Assembly [7]. Він визначає логіку роботи мікроконтролера та взаємозв'язок з іншими компонентами схеми.

У даному прикладі реалізовано використання лінійного конгруентного генератора (ЛКГ) в Proteus – це процес створення та симуляції алгоритму генерації псевдовипадкових чисел за допомогою лінійного конгруентного методу у програмному середовищі Proteus.

Формула ЛКГ, яка використовується для генерації псевдовипадкових чисел, виглядає так:

$$X_{n+1} = (a \cdot X_n + c) \bmod m,$$

де X_{n+1} – наступне псевдовипадкове число,

X_n – поточне псевдовипадкове число (або початкове значення X_0, яке називається *зерном*),

(a) – множник (параметр генератора) обирається для забезпечення гарної дисперсії псевдовипадкових чисел,

(c) – приріст (зміщення, або інкремент),

m – модуль (обмежує значення, X_n знаходиться в межах від 0 до m-1).

Якщо $c = 0$, генератор називається *мультиплікативним*. Якщо $c \neq 0$, то генератор є повним ЛКГ.

Розроблений програмний код для мікроконтролера завантажується в джерело коду програмного середовища Proteus. Проводиться його компіляція для отримання виконуваного файлу. *Компіляція* – це процес перетворення вихідного коду, написаного на мові програмування високого рівня, у машинний код, який може виконуватися мікроконтролером [8].

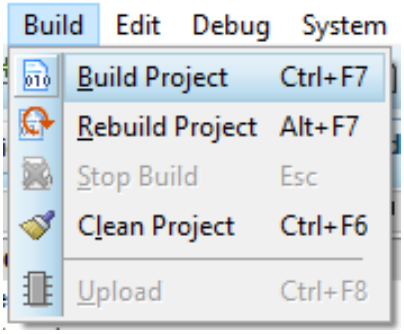

Рис. 5. Вікно меню компіляції коду в програмному середовищі Proteus

5. *Симуляція*. Після завершення завантаження коду необхідно активувати симуляцію, натиснувши кнопку "**Play**". Це дозволяє спостерігати за виконанням програми та перевіряти її відповідність запланованим функціональним характеристикам. Результати симуляції аналізуються, і за необхідності вносяться корективи до коду. Проведення симуляції на ранніх етапах розроблення дає змогу виявляти та усувати помилки, що, своєю чергою, знижує час та витрати на подальше створення реального пристрою.

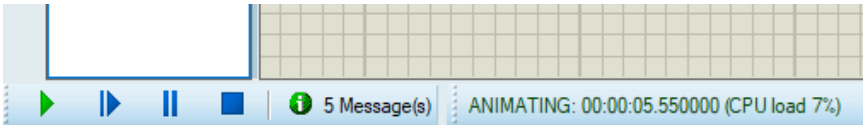

Рис. 6. Кнопка запуску симуляції

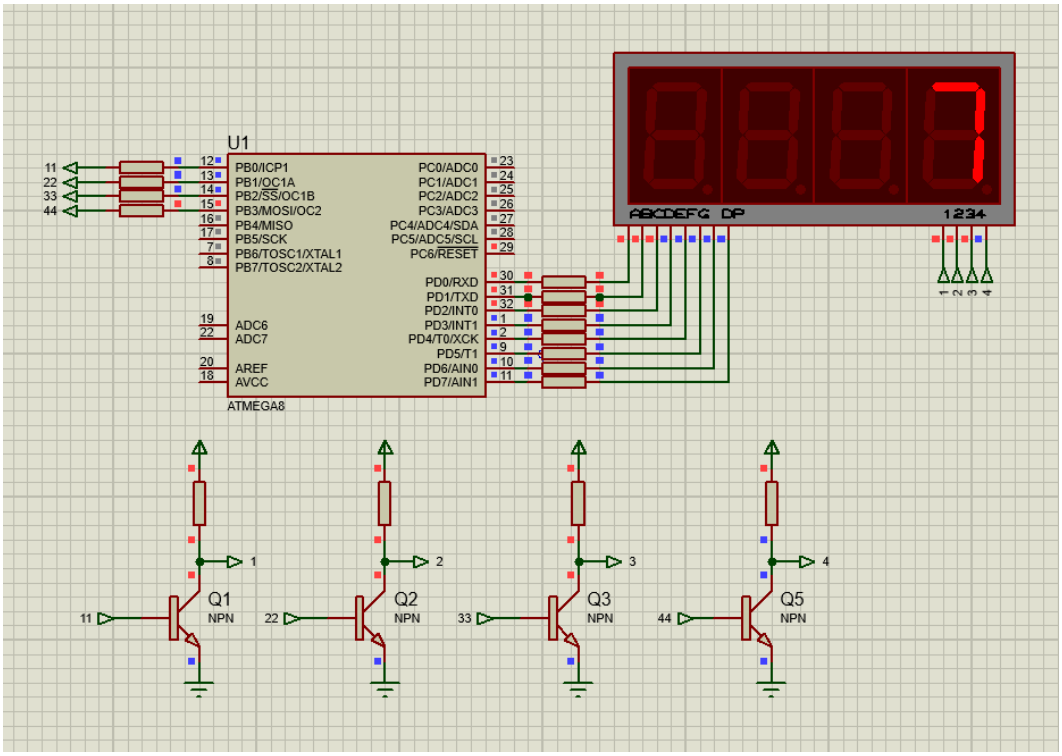

Рис. 7. Запущена симуляція використання лінійного конгруентного генератора

Верифікація та тестування системи є важливими етапами моделювання, що дозволяють переконатися у правильності роботи схеми та програмного коду. Верифікація включає перевірку схеми на наявність логічних помилок і забезпечення відповідності проєктним вимогам. За допомогою Proteus розробник може виконувати віртуальні вимірювання напруги, струму та інших параметрів, перевіряти правильність підключень та функціонування компонентів.

Тестування системи проводиться шляхом запуску симуляції. Це дозволяє побачити, як система працює в динаміці, перевірити реакцію на різні вхідні сигнали, виявити потенційні проблеми і недоліки. Програмне середовище Proteus дозволяє детально аналізувати роботу схеми, фіксувати і усувати проблеми ще на етапі проєктування, що значно знижує ризики при виготовленні фізичного пристрою.

Приклади використання моделювання в освітньому процесі. Моделювання у середовищі Proteus широко використовується в навчальному процесі в УДУ імені Михайла Драгоманова для підготовки студентів у галузі 122 Комп'ютерні науки, зокрема при вивченні курсу дисципліни «Комп'ютерна схемотехніка». Завдяки можливості створювати та симулювати електронні схеми, студенти отримують практичні навички роботи з мікроконтролерами без необхідності використання фізичних компонентів [9].

Одним із прикладів використання моделювання є розроблення лабораторних робіт, у процесі виконання яких студенти створюють конкретні схеми, пишуть код для мікроконтролерів, здійснюють моделювання та тестування своїх рішень у віртуальному середовищі. Це дозволяє їм зрозуміти основи електроніки, принципи роботи мікроконтролерів, навчитися налагоджувати і верифікувати свої проєкти.

Використання Proteus в освітньому процесі сприяє розвитку критичного мислення, навичок розв'язування технічних задач, а також дозволяє студентам більш ефективно засвоювати матеріал і готуватися до роботи у сфері електроніки та мікропроцесорної техніки.

Висновки. У цьому дослідженні було розглянуто методичні підходи до моделювання 8-бітних мікроконтролерних пристроїв у програмному середовищі Proteus. Аналіз показав, що Proteus є потужним і ефективним інструментом для розроблення, тестування та оптимізації електронних схем, що значно спрощує процес проєктування і знижує витрати на

фізичне прототипування. Завдяки високої точності симуляції і наявності великої бібліотеки компонентів можна створювати складні системи і перевіряти їх роботу у віртуальному середовищі. Приклади використання Proteus в освітньому процесі демонструють його ефективність у підготовці фахівців у галузі електроніки та мікропроцесорної техніки.

Програмне середовище Proteus має широкі перспективи для використання у майбутніх дослідженнях і розробленнях. Завдяки своїм функціям і характеристикам, у Proteus можна зосередитися на інноваціях та оптимізації процесів розроблення електронних пристроїв. Подальше вдосконалення програмного середовища і розширення його функціональності може зробити його ще більш потужним інструментом для інженерів і дослідників. Використання Proteus у поєднанні з іншими сучасними інструментами і технологіями може значно прискорити процес розроблення і впровадження нових електронних систем. Мікроконтролери, особливо 8-бітні, продовжують відігравати ключову роль у розвитку технологій та інновацій. Вони знаходять застосування у широкому спектрі пристроїв і систем, від побутової техніки до складних промислових механізмів. Простота використання, низька вартість і висока енергоефективність роблять їх незамінними у багатьох сферах. Розвиток і вдосконалення мікроконтролерних технологій сприяє створенню нових інноваційних продуктів і рішень, що покращують якість життя та сприяють технологічному прогресу. Завдяки використанню мікроконтролерів, інженери можуть розробляти складні системи автоматизації, управління і моніторингу, що відкриває нові можливості для різних галузей промисловості та наукових досліджень.

Таким чином, програмне середовище Proteus та мікроконтролери є потужними інструментами для розвитку сучасних технологій, забезпечуючи ефективність, надійність і інноваційність розробок у галузі електроніки та мікропроцесорної техніки.

Подальші дослідження можуть бути спрямовані на подальшу оптимізацію методів моделювання та тестування 8-бітних мікроконтролерів у середовищі Proteus. Це може включати розроблення нових моделей компонентів, за допомогою яких можна ще точніше симулювати реальні умови роботи пристроїв, а також створення бібліотек з додатковими можливостями для аналізу та автоматизації тестування. Також перспективним є інтеграція Proteus з іншими програмними інструментами для комплексного моделювання систем, що включають не лише мікроконтролери, але й інші типи електронних пристроїв. Іншим напрямком може стати дослідження використання Proteus у контексті проєктування більш сучасних мікроконтролерних систем, наприклад, 16- або 32-бітних архітектур.

Список використаних джерел:

[1] Буров Є. В. Концептуальне моделювання інтелектуальних програмних систем: монографія. Нац. ун-т «Львівська політехніка». Львів, 2012. 429 с.

[2] Шамоня В.Г., Семеніхіна О.В., Друшляк М.Г. Використання віртуального середовища “Proteus” для підготовки майбутніх фахівців у галузі інформаційних технологій. Освітній вимір. 2019. Том 1. С. 187-193.

[3] Офіційна web-сторінка ATMEL (Microchip Technology). URL: https://www.microchip.com/en-us/products/microcontrollers-and-microprocessors/8-bit-mcus/avr-mcus (дата звернення: 25.09.2024).

[4] Офіційна web-сторінка програмного середовища Proteus. URL: https://www.labcenter.com (дата звернення: 25.09.2024).

[5] Вовк П.Б. Технологія проектування цифрових систем керування на базі AVR-мікроконтролерів. *International Scientific and Practical Conference of Young Scientists and Students "Actual Problems of Automation and Control"*. 2018. №6. С. 76-82. URL: http://av.lntu.edu.ua/csv/konf_2018.pdf

[6] Цирульник С. М. Задорожний В. К. Застосування програми ISIS пакету Proteus VSM при вивченні курсу «Мікропроцесорна техніка»: матеріали XIII міжнародної конференції з автоматики управління (Автоматика 2006). Вінниця: Універсум-Вінниця, 2007. С. 526–530.

[7] Цирульник С. М., Азаров О. Д., Крупельницький Л. В., Трояновська Т. І. Програмування мікроконтролерів AVR: навчальний посібник. Вінниця: ВНТУ, 2018. С. 44-54.

[8] Цирульник С. М., Лисенко Г. Л. Проектування мікропроцесорних систем: навчальний посібник. Вінниця: ВНТУ, 2012. 91 с.

[9] Заєць О.Ю. Методичні підходи до моделювання пристроїв на 8-бітних мікроконтролерах в програмному середовищі Proteus. *Інформаційно-комунікаційні технології в освіті*, (12). (2024). URL: https://e-journals.udu.edu.ua/index.php/ikt/article/view/1437

METHODICAL APPROACHES TO MODELING 8-BIT MICROCONTROLLER DEVICES IN THE PROTEUS SOFTWARE ENVIRONMENT

Oleksandr Zaiets

Abstract. Nowadays, microcontrollers are an integral part of modern life since their application has a wide range of devices and systems, from household appliances to complex industrial mechanisms. The constant need for microcontrollers requires practical approaches to their development and modeling. The article highlights the essence of methodological approaches to modeling 8-bit microcontroller devices in the Proteus software environment in the training of future IT specialists. The modeling stages are considered, including circuit development, program code writing and debugging, and system verification and testing. The results of the study confirm the high efficiency of the Proteus software environment for modeling 8-bit microcontroller systems, which contributes to improving the quality of education and development in the field of electronics and microprocessor technology. It is noted that microcontrollers continue to play a key role in the development of technology, improving the quality of life and promoting innovation in various fields. The relevance of this topic is due to the need to increase the efficiency of the development and testing of microcontroller systems, reduce prototyping costs, and improve the quality of final products. Using the Proteus software environment allows you to achieve these goals by providing developers with tools for accurately modeling and verifying electronic systems. Thus, this research aims to develop methodological approaches to modeling 8-bit devices, create accurate software and hardware models of microcontrollers, optimize algorithms for testing and verification of systems based on simulations, and increase the speed and quality of the design process. This will reduce the number of errors at the development stage, reduce the time for debugging prototypes, and ensure the high reliability of final electronic products.

Keywords: computer modeling, Proteus software environment, AVR, hardware simulation, software, circuit design, code debugging, system verification, prototyping, microprocessor technology, educational process, innovation, technology.

References (translated and transliterated)

[1] Ye. V. Burov, *Kontseptualne modeliuvannia intelektualnykh prohramnykh system: monohrafiia* [Conceptual Modeling of Intelligent Software Systems: Monograph], Ministry of Education and Science of Ukraine, National University "Lvivska Politekhnika". Lviv, 2012, 429 p. (in Ukrainian).

[2] V.G. Shamonia, O.V. Semenikhina, M.G. Drushlyak (2019) Using the virtual environment “Proteus” for training future specialists in the field of information technologies. Educational Dimension. Volume 1. Pp. 187-193. (in Ukrainian).

[3] The official web page of ATMEL (Microchip Technology) [Online]. Available: https://www.microchip.com/en-us/products/microcontrollers-and-microprocessors/8-bit-mcus/avr-mcus. (in English).

[4] The official web page of Proteus Software [Online]. Available: https://www.labcenter.com. (in English).

[5] P.B. Vovk Technology of designing digital control systems based on AVR microcontrollers. International Scientific and Practical Conference of Young Scientists and Students "Actual Problems of Automation and Control". 2018. 6. P. 76-82. [Online]. Available: http://av.lntu.edu.ua/csv/konf_2018.pdf (in Ukrainian).

[6] S. M. Tsyrulnyk and V. K. Zadorozhnyi, "Application of the ISIS Program of the Proteus VSM Package in the Study of the Course 'Microprocessor Technology'," in *Proceedings of the XIII International Conference on Automation Control (Avtomatyka 2006)*, Vinnytsia: Universum-Vinnytsia, 2007, pp. 526-530. (in Ukrainian).

[7] S. M. Tsyrulnyk, O. D. Azarov, L. V. Krupelnytskyi, and T. I. Troyanovska, *Prohramuvannia mikrokontrolleriv AVR* [Programming AVR Microcontrollers: Textbook], Vinnytsia: VNTU, 2018, 44-54 p. (in Ukrainian).

[8] S. M. Tsyrulnyk and H. L. Lysenko, *Proektuvannia mikroprotsesornykh system* [Designing Microprocessor Systems: Textbook], Vinnytsia: VNTU, 2012, 91 p. (in Ukrainian).

[9] O.Yu. Zayets “Methodical approaches to modeling devices on 8-bit microcontrollers in the Proteus software environment.” *Information and Communication Technologies in Education*, vol. 12, May 2024. [Online]. Available: https://e-journals.udu.edu.ua/index.php/ikt/article/view/1437. (in Ukrainian).