

Міністерство освіти і науки України
Національний університет водного господарства та
природокористування
Навчально-науковий інститут кібернетики, інформаційних
технологій та інженерії
Кафедра обчислювальної техніки

04-04-267M

МЕТОДИЧНІ ВКАЗІВКИ

до виконання курсової роботи з навчальної дисципліни
«Архітектура комп'ютерів»

для здобувачів вищої освіти першого (бакалаврського) рівня
за освітньо-професійною програмою «Комп'ютерна
інженерія» спеціальності 123 «Комп'ютерна інженерія»
денної та заочної форм навчання

Рекомендовано науково-
методичною радою з якості
ННІКІТІ
Протокол № 4 від 24.02.2025 р.

Рівне – 2025

Методичні вказівки до виконання курсової роботи з навчальної дисципліни «Архітектура комп'ютерів» для здобувачів вищої освіти першого (бакалаврського) рівня за освітньо-професійною програмою «Комп'ютерна інженерія» спеціальності 123 «Комп'ютерна інженерія» денної та заочної форм навчання [Електронне видання] / Шатна А. В., Багнюк О. М. – Рівне : НУВГП, 2025 – 27 с.

Укладачі: Шатна А. В., старший викладач кафедри
обчислювальної техніки;

Багнюк О. М., старший викладач кафедри
обчислювальної техніки.

Відповідальний за випуск: Сидор А. І., к.т.н., доцент., в.о.
завідувача кафедри обчислювальної техніки.

Керівник (гарант) освітньої програми
«Комп'ютерна інженерія»
спеціальності 123 «Комп'ютерна інженерія» Сидор А. І.

© А. В. Шатна,
О. М. Багнюк, 2025
© НУВГП, 2025

ЗМІСТ

Вступ.....	4
1. Загальні положення.....	6
1.1. Мета і задачі курсової роботи.....	6
1.2. Тематика курсової роботи.....	6
1.3. Структура курсової роботи.....	6
2. Перелік та опис документів за результатами проектування.....	8
2.1. Титульний лист.....	8
2.2. Реферат.....	8
2.3. Технічне завдання.....	8
3. Основні вимоги до оформлення текстових документів...9	
3.1. Загальні вказівки.....	9
3.2. Посилання на нормативну літературу.....	8
3.3. Побудова тексту.....	10
3.4. Оформлення ілюстрацій.....	10
3.5. Побудова таблиць.....	11
3.6. Написання формул.....	12
3.7. Підготовка висновку.....	12
3.8. Оформлення списку джерел інформації.....	12
4. Індивідуальні завдання.....	13
Список джерел інформації	25
Додаток А Приклад оформлення та порядок слідування структурних елементів курсового проекту.....	26

ВСТУП

У сучасному навчанні фахівців у галузі комп'ютерної інженерії можна виділити три основних етапи, на кожному з яких розглядаються важливі аспекти цієї галузі:

1. Перший етап — вивчення основ обчислювальної математики, загальні принципи алгоритмізації, одна чи дві мови програмування, в тому числі програмування на асемблері, а також основи організації функціонування комп'ютерних систем.

2. Проміжний етап — включає вивчення архітектури комп'ютерних систем, системної інженерії, схемотехніки та системного програмування.

3. Завершальний етап — вивчення організації функціонування різних вузлів та модулів комп'ютерних систем, обчислювальних мереж, системного програмного забезпечення, САПР та експлуатації комп'ютерних систем і мереж.

Основна увага в курсі «Архітектура комп'ютерів» приділяється найбільш популярним на сьогодні комп'ютерам, зокрема IBM PC. Принципи викладу дисципліни орієнтовані на розгляд архітектурних рішень при побудові різних модулів комп'ютерів, а технічні характеристики, такі як швидкість, обсяг пам'яті, управляючі сигнали та інші, зазвичай не розглядаються детально, оскільки вони змінюються з часом, і актуальні дані можна знайти в довідковій літературі. Такий підхід дозволяє ефективно викладати великий обсяг матеріалу і залишатися актуальним у постійно змінюваному світі комп'ютерних технологій.

Крім того, архітектура окремих модулів комп'ютерів пов'язана з управлінням на низькому рівні — через регістри та адресний простір портів вводу/виводу. Знання принципів управління модулями на цьому рівні дає студентам глибоке розуміння архітектурних особливостей комп'ютера, що є необхідним для розробки ефективного програмного забезпечення.

Методичні вказівки для курсової роботи з дисципліни «Архітектура комп'ютерів» спрямовані на застосування теоретичних знань для отримання практичних навичок у створенні програмного забезпечення, використовуючи можливості управління модулями комп'ютерів на рівні портів вводу/виводу. Курсовий проєкт також дає можливість студентам самостійно вирішувати професійні завдання у цій сфері.

Курсова робота є першим етапом практичної підготовки студентів у рамках бакалаврської програми за спеціальністю «Комп'ютерна інженерія» і здійснюється в 4-му семестрі.

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1. Мета і задачі курсової роботи

Курсова робота з дисципліни «Архітектура комп'ютерів» виконується студентами 4-го семестру бакалаврської програми за спеціальністю 123 «Комп'ютерна інженерія». Метою курсової роботи є застосування теоретичних знань, отриманих під час вивчення курсу, для набуття практичного досвіду у створенні ефективного програмного забезпечення. Завдання курсової роботи полягає в самостійному вирішенні конкретної професійної проблеми в межах обраної тематики та розробки відповідного програмного рішення, використовуючи сучасні методи та технології програмування.

1.2. Тематика курсової роботи

Тематики курсових проектів представлені в Розділі 4. Окрім того, за погодженням з керівником проекту можуть бути запропоновані інші теми, що відповідають актуальним задачам розробки ефективного програмного забезпечення в різних прикладних сферах, таких як медицина, фінанси, промисловість, освіта, та телекомунікації, з використанням можливостей управління модулями комп'ютерів на рівні портів вводу/виводу.

1.3. Структура курсової роботи

Структура та зміст курсового проекту мають відповідати затвердженій темі та отриманому завданню.

Розробка змісту дипломного проекту здійснюється студентом у співпраці з керівником.

2 ПЕРЕЛІК ТА ОПИС ДОКУМЕНТІВ ЗА РЕЗУЛЬТАТАМИ ПРОЕКТУВАННЯ

2.1. Титульний лист

Приклад оформлення титульного листа наведено в Додатку А.

Примітка: У Додатку А сполучення КІ-ХХ означає групу, в якій навчається студент, а сполучення ХХХХ – варіант відповідно до списку групи.

2.2. Реферат

Реферат, що є частиною комплексу документів, містить короткий опис завдань, які були вирішені в процесі проектування. Реферат обов'язково складається українською та англійською мовами. За необхідності, або на вимогу керівника, реферат може бути підготовлена також іншими мовами, наприклад, німецькою. Приклад оформлення реферату наведено в Додатку А.

2.3. Технічне завдання

Технічне завдання розробляється відповідно до вимог ЄСПД ДСТУ ГОСТ 19.201-78. Під час курсового проектування деякі розділи та підрозділи технічного завдання не включаються, відповідно до вимог ЄСПД до документів. Приклад оформлення технічного завдання для курсового та дипломного проектування наводиться в Додатку А.

3 ОСНОВНІ ВИМОГИ ДО ОФОРМЛЕННЯ ТЕКСТОВИХ ДОКУМЕНТІВ

3.1. Загальні вказівки

Текстові документи (зокрема програмні документи та інші) повинні бути оформлені на аркушах формату А4 (210 x 297 мм). Якщо необхідно включити зображення схем, таблиць або малюнків, допускається використання аркушів формату А3 (297 x 420 мм), складених до формату А4. Документи повинні бути набрані за допомогою комп'ютера шрифтом Times New Roman розміром 14 пунктів з полуторним міжрядковим інтервалом у текстовому редакторі Word. Вони виконуються на одній стороні аркуша білого паперу українською мовою. Розміри полів: верхнє — 20 мм, ліве — 25 мм, нижнє — 20 мм, праве — 15 мм. Рисунки (включаючи схеми алгоритмів) та таблиці можуть бути виконані за допомогою комп'ютера, олівця або чорною пастою.

3.2. Посилання на нормативну літературу

Посилання на нормативні документи (стандарти, правила, технічні умови, інструкції) та інші джерела (книги, статті, патенти, авторські свідоцтва на винаходи) повинні бути надані в тексті пояснювальної записки, якщо вони містять інформацію, яка підтверджує обґрунтованість прийнятих рішень, методик розрахунків, вибору формул, коефіцієнтів або нормативних значень. При цьому вказується порядковий номер відповідного джерела в списку літератури, що наведений наприкінці записки, у квадратних дужках, наприклад [1], [3,5,7].

3.3. Побудова тексту

Написання текстових документів повинно відповідати вимогам ДСТУ 3008–95. Текстовий документ може включати як суцільний текст, так і текст, поділений на графи (наприклад, специфікації, таблиці, відомості тощо). Використання прикрас у тексті, таких як зміна шрифтів, підкреслення слів, фраз або назв елементів рубрикації, а також застосування кольорових букв і подібних ефектів, не допускається.

Основними структурними елементами тексту є розділи, підрозділи, пункти та підпункти. Структурні елементи, такі як «ЗМІСТ», «ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ», «ВСТУП», «ВИСНОВОК», «СПИСОК ДЖЕРЕЛ ІНФОРМАЦІЇ», не нумеруються, а їхні назви виконують функцію заголовків для відповідних структурних частин.

Кожен розділ має починатися з нової сторінки і бути пронумерований арабськими цифрами без крапки, наприклад: 1, 2, 3 і так далі. Нумерація підрозділів складається з номера розділу та порядкового номера підрозділу, розділених крапкою. Після номера підрозділу крапка не ставиться, наприклад: 1.1, 1.2 і так далі.

3.4. Оформлення ілюстрацій

Ілюстрації, що включаються в пояснювальну записку, повинні відповідати вимогам стандартів ЄСКД та ЄСПД. Їх кількість має бути достатньою для наочного пояснення викладеного матеріалу. Ілюстрації можна розміщувати безпосередньо після посилання на них у тексті або на наступних сторінках. Також вони можуть бути розміщені в додатках.

Ілюстрації можуть бути виконані на білому або міліметровому папері за допомогою олівця або чорної пасти. Усі ілюстрації, незалежно від способу виконання, позначаються словом «Рис.» і нумеруються арабськими цифрами в межах усього тексту або кожного розділу окремо (якщо ілюстрацій багато). Якщо ілюстрації нумеруються в межах розділу, їх номер складається з номера розділу та порядкового номера ілюстрації, розділених крапкою. Наприклад, ілюстрація першого розділу матиме номер «Рис. 1», а ілюстрація другого підрозділу другого розділу – «Рис. 2.1».

При посиланні на ілюстрацію в тексті вказується її повний номер, наприклад «Рис. 1» або «Рис. 1.5». Повторні посилання на ілюстрації надаються зі скороченням: «див. Рис. 4» або «див. Рис. 1.2».

3.5. Побудова таблиць

Таблицю слід розмішувати після першого згадування про неї в тексті або на наступній сторінці. У пояснювальній записці таблиці повинні відповідати вимогам ДСТУ 3008-95. Якщо в документі кілька таблиць, їх нумерують арабськими цифрами по всьому тексту. Напис типу «Таблиця 2» або «Таблиця 1.2» (друга таблиця першого розділу) розташовують ліворуч над таблицею без лапок і крапки.

На всі таблиці повинно бути посилання в тексті, наприклад, (табл. 2) або (табл. 3.1). У разі повторного посилання використовують форму «див. табл. 2» або «див. табл. 1.4». Якщо в документі лише одна таблиця, їй присвоюється номер згідно з наведеними вимогами.

3.6. Написання формул

Порядкові номери всіх формул вказуються в круглих дужках у крайньому правому куті на рівні формули. Якщо формула є дробом з горизонтальною рисою, номер розташовується по центру основної лінії. Якщо формула складається з кількох рядків, номер ставиться проти останнього рядка. У разі нумерації групи формул, номер поміщається праворуч від центру дужки, на правому краю сторінки.

3.7. Підготовка висновку

Висновки та пропозиції є підсумковим етапом курсового проекту. У них потрібно оцінити результати виконаної роботи, обґрунтувати обрані підходи для досягнення поставленої мети, а також подати основні технічні характеристики розробленого об'єкта з вказівкою ключових показників. Особливо важливо зазначити, що розроблений об'єкт (система, пристрій, програма, база даних, модель) повністю відповідає вимогам дипломного завдання. У висновках також можна запропонувати напрямки подальшого удосконалення проекту, спрямовані на підвищення його техніко-економічних показників. Обсяг висновків не повинен перевищувати однієї-двох сторінок пояснювальної записки, з наскрізною нумерацією кожного висновку.

3.8. Оформлення списку джерел інформації

Список використаної літератури оформлюється згідно ДСТУ 8302:2015 «Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання» з урахуванням правок (код УКНД 01.140.40).

4 ІНДИВІДУАЛЬНІ ЗАВДАННЯ

Номер завдання відповідає номеру студента в журналі. Кожне завдання включає теоретичне питання та одне практичне завдання. Відповіді на теоретичні питання подається у вигляді реферату, де описуються архітектурні особливості відповідного модуля, режими його роботи та можливості управління на рівні портів вводу/виводу. Відповідь на практичне завдання оформлюється у вигляді схеми алгоритму та програми, написаної на одній з мов високого рівня програмування (C++, C#). Загальний обсяг відповідей на індивідуальне завдання (основна частина пояснювальної записки) має бути не менше 15 сторінок тексту, підготовленого в редакторі Word і роздрукованого на папері.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ №1

1. Спеціальна конструкція процесора RISC-V
 - Опис: розробка та реалізація спеціального процесора RISC-V з використанням мови опису обладнання (HDL), наприклад Verilog або VHDL. Зосередьтеся на оптимізації для конкретних програм або енергоефективності.
 - Основні компоненти: АЛУ, блок керування, реєстровий файл, інтерфейс пам'яті.
2. Розробити програму, яка забороняє на 100 тиків роботу з клавіатурою (у цей час генерація звуку «вибуху»).

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ №2

1. Прискорювач нейронної мережі на основі FPGA

- Опис: створіть прискорювач на основі FPGA для алгоритмів глибокого навчання, оптимізуючи швидкість і енергоефективність.

- Ключові компоненти: реалізації рівнів (наприклад, згорткові, повністю підключені), архітектури потоків даних і порівняльний аналіз продуктивності.

2. Розробити програму, яка заблоковує доступ до миші на 50 тиків та виводить анімацію блимання екрану під час блокування.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ №3

1. Реалізація протоколу когерентності кешу

- Опис: реалізуйте та оцінюйте різні протоколи когерентності кешу (наприклад, MOESI) у моделюванні багатоядерної системи.

- Ключові компоненти: моделювання кількох процесорів, ієрархія пам'яті та показники продуктивності.

2. Розробити програму, яка аналізує структуру директорій FAT, визначає максимальну кількість файлів у підкаталозі та їх середній розмір.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ №4

1. Аналіз продуктивності різних архітектур пам'яті

- Опис: Порівняйте та проаналізуйте продуктивність різних архітектур пам'яті (наприклад, SRAM, DRAM, Flash) щодо швидкості, енергоспоживання та затримки.

- Ключові компоненти: імітаційні моделі, програми порівняльного аналізу та аналіз результатів.

2. Розробити програму, яка створює ефект «плаваючого вікна», де текстовий курсор буде автоматично переміщатися між різними вікнами на екрані, залежно від часу.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ №5

1. Енергоефективний дизайн апаратного забезпечення для пристроїв IoT

- Опис: розробка мікроконтролера з низьким енергоспоживанням для додатків Інтернету речей, зосереджуючись на енергоефективних обчислювальних техніках.

- Ключові компоненти: режими сну, периферійні пристрої з низьким енергоспоживанням і методи збору енергії.

2. Розробити програму, яка генерує випадкові геометричні фігури на екрані, кожен раз з новим кольором, і дозволяє їх «перетягувати» мишею по екрану.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ №6

1. Апаратне прискорення для криптографічних алгоритмів

- Опис: Створіть апаратний прискорювач для криптографічних алгоритмів (наприклад, AES або RSA) за допомогою FPGA або ASIC.

- Ключові компоненти: шляхи даних, логіка керування та оцінка продуктивності порівняно з програмними реалізаціями.

2. Розробити програму, яка імітує анімацію руху автомобіля по екрану, причому «дорога» складається з чергуючих горизонтальних ліній, і автомобіль повинен слідувати по цих лініях.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ №7

1. Віртуалізація та впровадження гіпервізора
 - Опис. Реалізуйте легкий гіпервізор для керування віртуальними машинами на певній архітектурі, зосереджуючись на продуктивності та безпеці.
 - Ключові компоненти: керування пам'яттю, планування процесора та віртуалізація вводу/виводу.
2. Розробити програму, яка змінює колір фону кожної другої секунди та виводить вниз екрану поточний час у вигляді цифрового годинника.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ №8

1. Динамічне масштабування напруги та частоти (DVFS)
 - Опис: Впровадження системи DVFS на процесорі для оптимізації продуктивності та споживання енергії залежно від робочого навантаження.
 - Ключові компоненти: вимірювання робочого навантаження, алгоритми масштабування та тестування продуктивності.
2. Розробити програму, яка створює графік синусоїдальної хвилі та дає можливість змінювати частоту хвилі за допомогою повзунка.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ №9

1. Розробка та впровадження 3D-чипів
 - Опис. Дослідіть проблеми проектування та переваги 3D-інтегральних схем (IC) порівняно з традиційними 2D-проектами.
 - Ключові компоненти: управління температурою, конструкція з'єднання та оцінка продуктивності.

2. Розробити програму, яка здійснює автоматичне масштабування зображень, виводячи їх з лінійним інтерполюванням для збільшення та зменшення.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ №10

1. Оптимізація компілятора для конкретних архітектур

- Опис: розробіть компілятор, який оптимізує код для конкретної архітектури, зосереджуючись на плануванні інструкцій, розподілі реєстрів і розгортанні циклу.

- Ключові компоненти: бекенд компілятора, методи оптимізації та аналіз продуктивності.

2. Розробити програму, яка ініціює «систему сигналізації», де при натисканні на клавішу виконується серія звукових сигналів, а на екрані з'являється анімація спалаху червоного кольору.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ №11

1. Система порівняльного аналізу та оцінки ефективності

- Опис: створіть систему порівняльного аналізу для оцінки продуктивності різних архітектур або конфігурацій за допомогою набору програм.

- Ключові компоненти: показники ефективності, візуалізація результатів і порівняльний аналіз.

2. Розробити програму, яка автоматично відображає на екрані серію випадкових геометричних фігур, змінюючи їхній колір і розмір кожні 3 секунди.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ №12

1. Проектування захищеної архітектури процесора

- Опис: досліджуйте та впроваджуйте функції безпеки в архітектурі процесора, такі як безпечне завантаження, довірені середовища виконання або апаратне шифрування.

- Ключові компоненти: протоколи безпеки, компроміси продуктивності та моделювання загроз.

2. Розробити програму, яка виводить на екран часову шкалу з 24 годинам, де кожні 5 хвилин додається нова мітка, що показує поточний час.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ №13

1. Реалізація апаратного прискорювача для алгоритмів машинного навчання

Опис: Створіть апаратний прискорювач для виконання операцій машинного навчання (наприклад, для алгоритмів класифікації або регресії) на FPGA. Оптимізуйте використання пам'яті та швидкість виконання операцій.

Ключові компоненти: Паралельна обробка даних, оптимізація шляху даних, вибір алгоритмів для апаратної реалізації, енергоефективність.

2. Розробити програму, яка відображає рухомий об'єкт, який рухається по екрану в напрямку, вказаному користувачем за допомогою миші.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ №14

1. Розробка адаптивного механізму управління кешем для багатоядерної архітектури

Опис: Реалізуйте адаптивний механізм управління кешем для багатоядерної системи, який динамічно налаштовує стратегію кешування в залежності від навантаження на процесори.

Ключові компоненти: Механізми кешування, багатоядерні процесори, стратегія управління кешем, алгоритми оптимізації доступу до пам'яті.

2. Розробити програму для виведення статистики по введеним користувачем текстам (наприклад, кількість слів, символів, унікальних слів).

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ №15

1. Проектування та оптимізація пам'яті з підтримкою невизначеного доступу в реальному часі

Опис: Розробіть архітектуру пам'яті, що підтримує невизначений доступ (наприклад, з системами реального часу), з урахуванням жорстких обмежень на затримки і продуктивність.

Ключові компоненти: Архітектура пам'яті, механізми управління доступом, системи реального часу, обмеження затримок.

2. Розробити програму, яка реалізує ефект «каруселі» зображень, що автоматично змінюються через певний інтервал часу.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ №16

1. Розробка апаратного прискорювача для обробки зображень на FPGA

Опис: Реалізуйте апаратний прискорювач для виконання основних операцій обробки зображень (наприклад, фільтрація, детекція країв, морфологічні операції) на FPGA.

Ключові компоненти: Швидкість обробки зображень, використання апаратних блоків для математичних операцій, оптимізація для реального часу.

2. Розробити програму для переміщення зображення по екрану за допомогою клавіш стрілок.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ №17

1. Апаратне прискорення для алгоритмів пошуку в графах

Опис: Розробіть апаратний прискорювач для виконання алгоритмів пошуку в графах, таких як пошук в глибину (DFS) або пошук в ширину (BFS), на FPGA або ASIC.

Ключові компоненти: Алгоритми пошуку в графах, паралельна обробка, обробка великих обсягів даних, оптимізація використання пам'яті.

2. Розробити програму, яка генерує випадкову послідовність чисел від 1 до 100 та виводить їх на екран, сортує за допомогою одного з алгоритмів сортування та відображає процес сортування.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ №18

1. Інтерфейс високошвидкісного вводу/виводу для спеціалізованих обчислень

Опис: Розробіть інтерфейс вводу/виводу з високою пропускну здатністю для спеціалізованого

обчислювального пристрою, що працює з великими даними (наприклад, для наукових або фінансових обчислень).

Ключові компоненти: Інтерфейс вводу/виводу, висока пропускну здатність, обробка великих обсягів даних, оптимізація латентності.

2. Розробити програму для створення анімації «падаючих часток» на екрані (як в ефекті дощу або снігу).

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ №19

1. Реалізація апаратного модуля для підтримки операцій з великими цілими числами (BigInt)

Опис: Створіть апаратний модуль для швидких операцій з великими цілими числами, що використовуються в криптографії, розв'язках математичних задач або обчисленнях з високою точністю.

Ключові компоненти: Апаратне забезпечення для обробки великих цілих чисел, підтримка операцій множення, ділення, порівняння.

2. Розробити програму, яка аналізує текстовий файл і підраховує кількість повторень кожного слова в ньому, виводячи результат у вигляді таблиці.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ №20

1. Оптимізація архітектури мікроконтролера для автономних роботизованих систем

Опис: Розробіть мікроконтролер, оптимізований для автономних роботизованих систем з обмеженими ресурсами енергії. Оцініть вплив різних режимів роботи на продуктивність та енергоспоживання.

Ключові компоненти: Енергоефективні режими роботи, периферійні пристрої, оптимізація обчислювальних ресурсів, автономія системи.

2. Розробити програму для автоматичного створення мозаїки з зображень, де програма об'єднує кілька зображень в одну картину.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ №21

1. Реалізація алгоритмів стиснення даних на FPGA

Опис: Розробіть апаратний прискорювач для алгоритмів стиснення даних (наприклад, Huffman або LZ77) на FPGA з орієнтацією на швидкість та ефективність.

Ключові компоненти: Алгоритми стиснення, апаратне прискорення, швидкість обробки, енергоефективність.

2. Розробити програму для генерації випадкових текстових паролів із заданою довжиною та складністю (використання великих/малих літер, цифр та спеціальних символів).

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ №22

1. Проектування та оптимізація векторних обчислень на GPU

Опис: Розробіть апаратний блок для виконання векторних обчислень на графічному процесорі (GPU), оптимізуючи обчислювальні ресурси та мінімізуючи енергоспоживання.

Ключові компоненти: Векторні обчислення, використання GPU для паралельної обробки, оптимізація використання обчислювальних одиниць.

2. Розробити програму для підрахунку кількості символів, слів і абзаців у текстовому файлі.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ №23

1. Розробка апаратного прискорювача для алгоритмів машинного зору на FPGA

Опис: Створіть апаратний прискорювач для алгоритмів машинного зору (наприклад, для виявлення об'єктів або класифікації зображень) на FPGA з орієнтацією на високу швидкість обробки.

Ключові компоненти: Паралельна обробка зображень, використання векторних операцій, оптимізація використання пам'яті, швидкість обробки в реальному часі.

2. Розробити програму для створення анімації падіння тексту по екрану, де текст падає вниз та розсіюється, імітуючи ефект дощу.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ №24

1. Розробка апаратного прискорювача для обчислення хеш-функцій

Опис: Створіть апаратний прискорювач для обчислення хеш-функцій (наприклад, SHA-256 або MD5) за допомогою FPGA або ASIC для високопродуктивних застосувань (наприклад, блокчейн).

Ключові компоненти: Логіка хешування, апаратне прискорення, оптимізація потоків даних, порівняння швидкості та енергоефективності з програмними рішеннями.

2. Розробити програму для створення анімованого слайдшоу з набору зображень, де кожне зображення змінюється через певний інтервал часу.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ №25

1. Реалізація високопродуктивного апаратного контролера для роботи з мережею 5G

Опис: Створіть апаратний контролер для підтримки протоколів зв'язку в мережах 5G з високою пропускнуою здатністю та низькою затримкою.

Ключові компоненти: Протоколи 5G, апаратне прискорення обробки пакетів, оптимізація використання пропускнуої здатності каналу, контроль за затримкою.

2. Розробити програму для підсвічування синтаксису програмного коду у текстовому файлі.

СПИСОК ДЖЕРЕЛ ІНФОРМАЦІЇ

1. Поворознюк А. І. Архітектура комп'ютерів. Архітектура мікропроцесорного ядра та системних пристроїв : навчальний посібник. Ч.1. Харків : "Торнадо", 2019. 355 с.

2. Поворознюк А. І. архітектура компютерів. Архітектура зовнішньої пам'яті відеосистеми та зовнішніх інтерфейсів: Навчальний посібник. Ч.2. Харків : "Торнадо", 2019. 296 с.

3. СТЗВО-ХПІ-3.01-2018. Текстові документи у сфері навчального процесу. Загальні вимоги до виконання. Чинний від 28.09.2018.

4. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. Чинний від 2017-07-01. Київ : ДП "УкрНДНЦ", 2016. 26 с.

5. ДСТУ ГОСТ 7.1-2006. Система стандартів з інформації, бібліотечної та видавничої справи. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання. ГОСТ 7.1–2003, IDT. Чинний з 2008-04-01. Київ : Держспоживстандарт України, 2007. III, III, 47 с.

6. ДСТУ 3966-2009. Термінологічна робота. Засади і правила розроблення стандартів на терміни та визначення понять. Чинний від 01.07.2010.

7. СТЗВО-ХПІ-2.01-2018. Дипломні проекти та дипломні роботи. Загальні вимоги до виконання. Чинний від 28.09.2018.

Додаток А

Приклад оформлення та порядок слідування структурних
елементів курсової роботи

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВОДНОГО
ГОСПОДАРСТВА ТА ПРИРОДОКОРИСТУВАННЯ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ КІБЕРНЕТИКИ,
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ІНЖЕНЕРІЇ
КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

КУРСОВА РОБОТА

з навчальної дисципліни: «Архітектура комп'ютерів»

Виконав студент 2 курсу, групи КІ-ХХ:

_____/М.В. Бойчура/
(підпис, прізвище та ініціали)

Варіант ХХХХ

Керівник
_____/А.В. Шатна/
(підпис, прізвище та ініціали)

Рівне 2024

РЕФЕРАТ

Звіт про виконання КР: 15с., 12 рис., 3 табл, 5 інформативних джерел, 1 додаток.

Ключові слова: КЕШ-ПАМ'ЯТЬ, КОНТРОЛЕР ПЕРЕРИВАНЬ, АЛГОРИТМ ЧИТАННЯ ФАЙЛІВ, АЛГОРИТМ БРЕЗЕНХЕЙМА, МОДИФІКАЦІЯ, ПРОГРАМА, РЕГІСТРИ ПАЛІТРИ.

Викладено принципи роботи та особливості керування кеш-пам'яттю, а також можливості програмування контролера переривань, із зазначенням ролі OCW та ICW. Описано алгоритм читання файлів з використанням структури ROOT і FAT. Проведена модифікація алгоритму Брезенхейма для створення товстих ліній. Розроблено програму, що завантажує дві відеосторінки з файлу та дозволяє приховувати одну з чвертей екрана при натисканні клавіші 1, 2, 3 або 4 (з використанням програмування регістрів палітри).

ABSTRACT

Report on the Completion of the Coursework: 15 pages, 12 figures, 3 tables, 5 informational sources, 1 appendix.

Keywords: CACHE MEMORY, INTERRUPT CONTROLLER, FILE READING ALGORITHM, BRESENHAM'S ALGORITHM, MODIFICATION, PROGRAM, PALETTE REGISTERS.

The principles of cache memory operation and control management are outlined, along with the capabilities of programming the interrupt controller, highlighting the roles of OCW and ICW. The algorithm for reading files using the ROOT and FAT structures is described. A modification of Bresenham's algorithm for generating thick lines is presented. A program is developed that loads two video pages from a file and allows hiding one-quarter of the screen by pressing keys 1, 2, 3, or 4 (using palette register programming).