ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Програмно-апаратний комплекс для оцінки впливу архітектури комп'ютера на його продуктивність»

на здобуття освітнього ступеня бакалавра
зі спеціальності 123 Комп'ютерної інженерії
(код, найменування спеціальності)
освітньо-професійної програми Комп'ютерна інженерія
(назва)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ *Денис ВЕРЕСКУН*
(підпис) *Ім'я, ПРІЗВИЩЕ здобувача*

Виконав: здобувач вищої освіти гр. КІД-42
Денис ВЕРЕСКУН
Ім'я, ПРІЗВИЩЕ
Керівник: Юлія БОРОВА
науковий ступінь, вчене звання Ім'я, ПРІЗВИЩЕ
Рецензент: Сергій БОНДАР
науковий ступінь, вчене звання Ім'я, ПРІЗВИЩЕ

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра 123 Комп'ютерної інженерії

Ступінь вищої освіти бакалавр

Спеціальність 123 Комп'ютерної інженерії

Освітньо-професійна програма Комп'ютерна інженерія

ЗАТВЕРДЖУЮ
Завідувач кафедри
Комп'ютерної інженерії
_____ Наталія ЛАЩЕВСЬКА
_____
«_____» _____ 2025р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Верескун Денис Сергійович
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Програмно-апаратний комплекс для оцінки впливу архітектури комп'ютера на його продуктивність
керівник кваліфікаційної роботи Юлія БОРОВА,
(Ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «24» лютого 2025р. № 56
2. Строк подання кваліфікаційної роботи «3» червня 2024р.
3. Вихідні дані до кваліфікаційної роботи: науково-технічна література з архітектури комп'ютерів; характеристики процесорів Intel, AMD; результати тестування продуктивності (Cinebench, Geekbench, PassMark); дані про компоненти комп'ютера (CPU, RAM, GPU); публікації з аналізу ефективності комп'ютерних систем.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Поняття комп'ютерної архітектури. Типи процесорних архітектур (RISC, CISC, SIMD, MIMD)
2. Структура та компоненти сучасного комп'ютера. Програмні засоби для оцінки продуктивності
3. Оцінка впливу архітектури комп'ютера на його продуктивність
5. Перелік ілюстративного матеріалу: *презентація*
6. Дата видачі завдання «24» лютого 2025р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Збір даних	24.02–10.03.2025	Виконано
2	Обробка матеріалу	11.03–21.03.2025	Виконано
3	Розробка теоретичної частини	22.03–01.04.2025	Виконано
4	Написання 1-го та 2-го розділу	02.04–18.04.2025	Виконано
5	Написання 3-го розділу	19.04–30.04.2025	Виконано
6	Висновки за результатами аналізу	01.05–17.05.2025	Виконано
7	Розробка презентації	17.05–24.05.2025	Виконано
8	Передзахист	27.05–29.05.2025	Виконано
9	Здача в деканат	18.06–23.06.2025	Виконано

Здобувач вищої освіти _____ Денис ВЕРЕСКУН
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Керівник кваліфікаційної роботи _____ Юлія БОРОВА
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 44 стор., 5 рис., 3 табл., 30 джерел.

Мета роботи – оцінити вплив архітектури комп'ютера на його продуктивність за допомогою програмно-апаратного тестування.

Об'єкт дослідження – архітектура центрального процесора як ключовий фактор, що впливає на продуктивність комп'ютерної системи.

Предмет дослідження – методи та засоби оцінювання продуктивності комп'ютерної системи залежно від архітектурних особливостей процесора.

Короткий зміст роботи: У роботі досліджено вплив архітектури процесора на продуктивність комп'ютерної системи. Розглянуто основні типи архітектур, зокрема RISC і CISC, а також особливості реалізацій Intel та AMD. На базі фіксованої тестової платформи реалізовано програмно-апаратний підхід до порівняння процесорів із використанням бенчмарків Cinebench, Geekbench та PassMark. Отримані результати дозволили оцінити ефективність різних архітектур у типових сценаріях навантаження. У завершальній частині розглянуто сучасні тенденції розвитку архітектури, зокрема гетерогенність, чиплет-дизайн і нейроморфні системи.

КЛЮЧОВІ СЛОВА: АРХІТЕКТУРА КОМП'ЮТЕРА, ПРОЦЕСОР, ПРОДУКТИВНІСТЬ, RISC, CISC, INTEL, AMD, БАГАТОЯДЕРНІСТЬ, БЕНЧМАРК, CINEBENCH, GEEKBENCH, PASSMARK, CHIPLET, ЕНЕРГОЕФЕКТИВНІСТЬ.

ЗМІСТ

ВСТУП…… 8
1. ОГЛЯД АПАРАТНОЇ АРХІТЕКТУРИ КОМП'ЮТЕРІВ ……. 10
1.1 Поняття апаратної архітектури та її класифікація…… 10
1.2 Етапи розвитку архітектури комп'ютерів……. 12
1.3 Основні складові апаратної архітектури…… 13
1.4 Поширені архітектури процесорів: RISC, CISC, ARM……. 17
1.5 Особливості архітектур Intel, AMD і застосування у сучасних ПК…… 20
2. ВПЛИВ АПАРАТНОЇ АРХІТЕКТУРИ НА ЕФЕКТИВНІСТЬ РОБОТИ КОМП'ЮТЕРІВ…….. 23
2.1 Обмеження та виклики у порівнянні архітектур……... 23
2.2 Методологія оцінки ефективності комп'ютерів…… 25
2.3 Апаратна частина програмно-апаратного комплексу……... 27
2.4 Аналіз впливу процесорної архітектури…… 29
2.5 Тестування обчислювальної продуктивності за допомогою наявних програм…… 34
2.5.1 Тестування процесорів……. 35
2.6 Порівняння процесорів Intel та AMD за результатами бенчмарків…… 36
2.7 Особливості архітектур у контексті штучного інтелекту та хмарних обчислень…… 38
3. ПЕРСПЕКТИВИ РОЗВИТКУ АПАРАТНОЇ АРХІТЕКТУРИ……... 41
3.1 Тренди в архітектурі: багатоядерність, паралельність, енергоефективність…… 41
3.2 Інноваційні архітектури: MIMD, SIMD, нейроморфні, квантові системи…… 43
3.3 Вплив нових архітектур на майбутнє комп'ютерних систем…….. 45
3.4 Прогноз щодо розвитку архітектур з огляду на потреби IT-інфраструктури……... 47
ВИСНОВКИ…… 50
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ……... 52
Додаток А…… 54

ВСТУП

У світі, де цифрові технології визначають темп розвитку практично кожної сфери — від побуту до складної інженерії, питання продуктивності комп’ютерної системи постає особливо гостро. Вона напряму впливає на те, як швидко й ефективно можна виконати прикладні завдання, оптимізувати обчислення та раціонально використовувати наявні ресурси. З кожним роком апаратне забезпечення стає дедалі складнішим, а вибір — більш відповідальним: користувач стикається не просто з кількістю ядер чи частотою, а з цілим світом архітектурних рішень. Від класичних багатоядерних ЦП до складних гібридних систем з графічними прискорювачами, нейронними блоками, високошвидкісною пам’яттю — усе це потребує системного підходу до аналізу їх впливу на швидкодію та ефективність.

Дослідження архітектури комп’ютера — це вже давно не лише академічний інтерес. Це цілком прикладна потреба: для того, щоб зробити усвідомлений вибір, зекономити кошти, отримати максимальну віддачу від техніки, потрібно розуміти, як улаштовано систему «зсередини». І головне — вміти порівнювати не просто цифри у таблиці характеристик, а оцінювати реальні показники, які проявляються у задачах повсякденного чи професійного навантаження.

Саме тут на перший план виходять програмно-апаратні засоби — інструменти, що дозволяють протестувати систему в умовах, максимально наближених до реального використання. Вони не потребують складного проєктування з нуля: сучасний підхід передбачає використання вже перевірених рішень — бенчмарків, профілювальників, тестових стендів — для отримання об’єктивних і, що важливо, взаємно зіставних результатів.

Ця дипломна робота якраз і присвячена такому підходу. У центрі уваги — практичне дослідження, як конкретні елементи архітектури комп’ютера (кеш, підсистема пам’яті, типи шин, кількість і структура ядер) впливають на підсумкову

продуктивність. Для цього використано готові програмно-апаратні засоби, що дозволяє не відволікатися на розробку нових інструментів, а сконцентруватися на аналізі результатів та їх значенні в контексті ефективності обчислювальних платформ.

1. ОГЛЯД АПАРАТНОЇ АРХІТЕКТУРИ КОМП'ЮТЕРІВ

1.1 Поняття апаратної архітектури та її класифікація

Апаратна архітектура комп’ютера — це, по суті, «внутрішня будова» всієї системи. Вона визначає, як різні компоненти — процесор, пам’ять, відеокарта, накопичувачі, контролери — працюють разом як єдине ціле. Це не просто список частин, з яких складається комп’ютер, а спосіб їх організації: як вони з’єднані, як взаємодіють, як передають дані і розподіляють ресурси під час роботи. Від архітектури напряму залежить, наскільки швидко система справляється з поставленими завданнями, як вона реагує на навантаження, чи можна її вдосконалювати і модернізувати в майбутньому.

Сучасна архітектура комп’ютера побудована на кількох рівнях. Перший — структурний рівень — описує фізичне з’єднання компонентів: які використовуються шини, як вони передають дані між пристроями, яка логіка розміщення елементів на материнській платі. Це своєрідна інфраструктура, яка впливає на пропускну здатність і загальну стабільність роботи.

Функціональний рівень зосереджений на ролі, яку відіграє кожен модуль у процесі обчислення. Тут ідеться про розподіл задач між процесором, кеш-пам’яттю, контролерами пам’яті, відеокартою, блоком управління тощо. Цей рівень відповідає за логіку того, як запускаються програми, як зберігаються і обробляються дані, і як відбувається взаємодія з користувачем або зовнішніми пристроями.

На логічному рівні розглядається, як комп’ютер «розуміє» команди: як вони транслюються в мікрооперації, як працює конвеєр виконання інструкцій, як реалізована система командного набору. Цей рівень можна назвати найглибшим

— саме тут формується ефективність процесора, його здатність до паралельних обчислень, швидкість реакції на події тощо.

Існує кілька способів класифікації комп'ютерних архітектур. Один з них — за принципом організації логіки. У зосереджених системах вся логіка управління розташована в одному місці (найчастіше — у центральному процесорі). Інші частини просто виконують його команди. Це надійний підхід для більшості задач, але іноді він обмежує масштабування і швидкість у складних сценаріях. У розподілених архітектурах кожен компонент може мати власну логіку, що дозволяє працювати більш паралельно, швидше обробляти великі обсяги даних і краще адаптуватись до змішаних задач.

Крім цього, архітектури відрізняються за способом обміну даними: через спільну шину, через точка-точка інтерфейси або навіть через мережеву модель, як у багатопроцесорних системах. Ще один важливий параметр — ступінь інтеграції компонентів: класичні системи передбачають окремі чипи для всього, а сучасні SoC об'єднують кілька модулів (CPU, GPU, RAM-контролер тощо) на одному кристалі.

У наш час дедалі частіше зустрічаються гібридні архітектури. Це коли комп'ютер має різні типи обчислювальних блоків — наприклад, стандартні ядра процесора, графічний прискорювач і додатковий блок для обробки штучного інтелекту. Такий підхід дозволяє поєднати високу продуктивність із енергоефективністю та універсальністю. Зокрема, подібні архітектури використовуються в ноутбуках, смартфонах і навіть серверних системах.

Отже, архітектура комп'ютера — це не просто технічний термін, а ключ до розуміння того, як саме система працює, де її сильні та слабкі сторони, і що можна зробити для покращення її продуктивності або адаптації до нових завдань. Вона визначає не лише швидкість, але й стабільність, можливість апгрейду, сумісність з програмним забезпеченням і здатність відповідати вимогам часу.

1.2 Етапи розвитку архітектури комп’ютерів

Архітектура комп’ютерів — це не щось статичне. Вона змінювалась, вдосконалювалась і підлаштовувалась під нові потреби з того моменту, як з’явилися перші електронні машини. Як і будь-яка технологія, комп’ютери пройшли довгий шлях розвитку, і з кожним етапом цей шлях супроводжувався як новими можливостями, так і новими викликами. Зміни стосувалися не лише зовнішнього вигляду чи продуктивності, а й самих принципів побудови комп’ютерних систем — від обчислень на лампах до сучасних чиплетів із блоками штучного інтелекту.

Перші комп’ютери з’явилися у середині XX століття й були вражаючими за розмірами, споживанням енергії та складністю обслуговування. Вони працювали на електронних лампах і мали дуже просту — а насправді примітивну — архітектуру. Кожна команда виконувалась послідовно, керування здійснювалося прямо з машинного коду, а будь-які зміни вимагали втручання інженера. Ці системи були повільні, ненадійні, але саме вони поклали початок майбутній революції.

Коли на зміну лампам прийшли транзистори, архітектура комп’ютерів почала стрімко змінюватись. Стало можливим зменшити розміри систем, підвищити їхню надійність та значно зменшити споживання енергії. З’явилась концепція багаторівневої пам’яті, логіка управління стала складнішою, а разом із нею і можливості систем. У програмуванні з’явилися компілятори — і писати код стало набагато простіше та швидше. Це був важливий крок у бік масового використання комп’ютерів.

Наступний важливий крок — поява інтегральних схем. Це дозволило розміщувати все більше логіки на одному кристалі, що в підсумку призвело до створення мікропроцесорів. Тепер один чип міг виконувати функції цілого блоку обладнання. З’явилися багатоядерні процесори, конвеєрна обробка інструкцій,

кеш-пам’ять різних рівнів, підтримка багатозадачності. Архітектура комп’ютера стала справді гнучкою: її можна було адаптувати під різні задачі — від ігор до наукових розрахунків.

Окремим етапом у розвитку стала поява персональних комп’ютерів. Це вже були не машини для обчислювальних центрів, а пристрої для масового користування. Архітектура ПК стала універсальною: підтримка складних операційних систем, розширення оперативної пам’яті, робота з графікою, підключення різноманітних пристроїв. Комп’ютер перетворився з інструмента для обчислень у багатофункціональну платформу.

Сучасна архітектура комп’ютерів — це складний, багаторівневий конструктор. У ній використовуються не лише класичні процесорні ядра, а й графічні прискорювачі, блоки для обробки штучного інтелекту, шифрувальні модулі, контролери віртуалізації. Відбувається поступове зміщення акценту з «максимальної потужності» до «розумної оптимізації». Архітектура стає гібридною, здатною ефективно адаптуватись до нових типів навантажень: обробки відео, AI-алгоритмів, хмарної інфраструктури тощо.

Загалом, розвиток комп’ютерної архітектури — це історія про постійне вдосконалення. Кожен новий етап додавав гнучкості, продуктивності, енергоефективності. І цей процес триває дотепер: з появою нових задач — з’являються і нові рішення, нові архітектури, нові принципи взаємодії апаратного та програмного забезпечення. І, ймовірно, саме архітектурні зміни стануть основою для наступного прориву в комп’ютерній галузі.

1.3 Основні складові апаратної архітектури

Апаратна архітектура комп’ютера — це не просто набір заліза, а ціла система взаємопов’язаних компонентів, які разом забезпечують роботу всієї

обчислювальної платформи. Кожен з елементів виконує свою чітку функцію, але найголовніше — працює в тандемі з іншими. І саме від того, наскільки правильно підібрано ці компоненти та як добре вони узгоджуються між собою, залежить те, наскільки швидко, стабільно й ефективно працює вся система.

Одним із ключових вузлів є центральний процесор, або просто CPU — мозок комп'ютера. Він відповідає за виконання основних обчислювальних і логічних операцій, керує роботою інших компонентів. У сучасних процесорах все побудовано максимально продумано: кілька ядер, кеш-пам'ять кількох рівнів, модулі передбачення переходів, регістри, апаратне прискорення певних задач — усе це дозволяє обробляти команди швидко й ефективно. Багатоядерність стала стандартом: навіть бюджетні процесори здатні працювати з кількома потоками одночасно.

Ще один важливий елемент — оперативна пам'ять (RAM). Вона виконує роль «швидкого нотатника» для процесора: саме тут зберігаються всі тимчасові дані, до яких потрібен миттєвий доступ під час роботи програм. Чим більший обсяг і швидкість оперативної пам'яті — тим краще система справляється з багатозадачністю та важкими програмами.

Для зберігання даних використовуються накопичувачі: це можуть бути як звичні жорсткі диски (HDD), так і сучасні твердотільні накопичувачі (SSD). Перші — більш об'ємні, але повільні, другі — значно швидші, без рухомих частин, тому менш схильні до пошкоджень. У багатьох випадках саме заміна HDD на SSD дає користувачеві відчутний приріст у швидкості завантаження системи та роботи програм.

Материнська плата — це своєрідний «організатор» всієї системи. Вона об'єднує процесор, пам'ять, накопичувачі, відеокарту та всі інші пристрої, дозволяючи їм обмінюватися даними через системні шини та інтерфейси. На ній розміщені чипсети, контролери, живлення, інтерфейси вводу/виводу — усе, що забезпечує життєдіяльність комп'ютера.

Для кращого розуміння того, як саме взаємодіють між собою основні компоненти комп’ютера, доцільно звернутись до класичної моделі логічної організації системи, яка базується на поділі архітектури на північний (Northbridge) та південний (Southbridge) мости.

Північний міст відповідає за взаємодію високошвидкісних компонентів: центрального процесора (CPU), оперативної пам’яті (RAM) та графічного адаптера (GPU). Він з’єднується з процесором через системну шину (Front-side bus) та забезпечує високу пропускну здатність при передачі даних між основними вузлами.

Південний міст, у свою чергу, обслуговує периферійні пристрої, що не вимагають надвисокої швидкості обміну: накопичувачі (через PCI/PCIe), USB-порти, мережеві інтерфейси та аудіо. Зв'язок між північним і південним мостами забезпечує внутрішня шина (Internal bus).

Ця архітектура активно застосовувалась у попередніх поколіннях систем, однак залишається корисною для розуміння базових принципів взаємодії компонентів у сучасних ПК, зокрема з точки зору логіки маршрутизації даних.

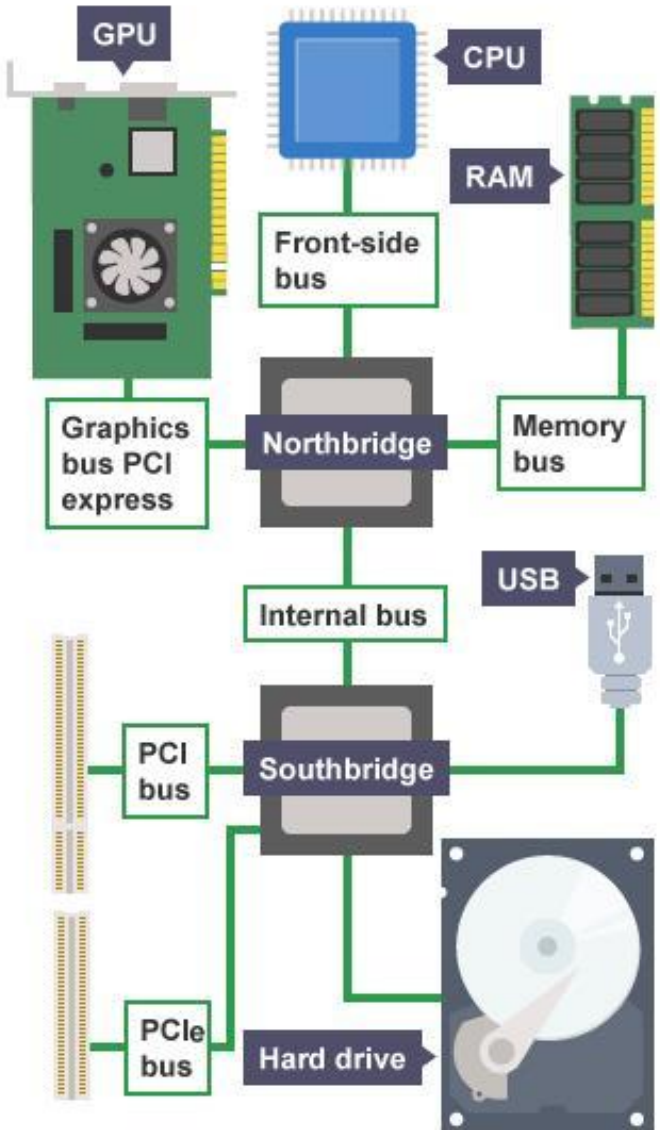


Рисунок 1.1 – Логічна структура архітектури комп’ютера з поділом на Northbridge і Southbridge.

Ще один критично важливий канал — системна шина. Через неї передається інформація між основними компонентами: процесором, пам’яттю, накопичувачами, графічними та іншими модулями. Виділяють кілька типів шин: даних, адресна та керуюча. Їхня ширина, частота та протоколи безпосередньо впливають на швидкість обміну даними.

У сучасних системах велику роль відіграє графічна підсистема, або GPU. Спершу відеокарти були потрібні виключно для виводу зображення на екран, але сьогодні вони активно застосовуються в 3D-графіці, відеомонтажі, симуляціях і

навіть у сфері штучного інтелекту. GPU відзначаються високим рівнем паралелізму — саме це робить їх незамінними для багатьох задач, які не під силу навіть потужному CPU.

До апаратної архітектури також належать пристрої введення та виведення — клавіатура, миша, монітор, принтери, вебкамери, мікрофони та інші засоби взаємодії з користувачем. Вони не лише забезпечують зручну роботу, а й визначають загальний комфорт і функціональність системи.

Не можна обійти увагою й систему охолодження. Це кулери, радіатори, термотрубки або рідинні рішення, які підтримують робочу температуру всіх основних компонентів. Без належного охолодження навіть найкращий процесор втратить у швидкості через тротлінг, а в гіршому разі — може вийти з ладу.

Завершує цей список блок живлення (PSU) — джерело енергії для всіх компонентів. Від його стабільності, потужності та якості залежить не тільки надійність системи, а й можливість її подальшої модернізації.

Усі ці елементи — від CPU до блоку живлення — працюють як одна злагоджена команда. Якщо один компонент слабкий або несумісний, він може стати вузьким місцем, яке гальмує всю систему. Саме тому правильний підбір і баланс між складовими — це основа стабільної, потужної та довговічної комп’ютерної архітектури.

1.4 Поширені архітектури процесорів: RISC, CISC, ARM

Архітектура процесора — це концептуальна модель, яка визначає принципи обробки інструкцій, формати команд, спосіб адресації пам’яті, набір доступних інструкцій та структуру внутрішніх обчислювальних блоків. Від

вибору архітектурного підходу залежить не лише швидкодія системи, а й її енергоефективність, універсальність, масштабованість і вартість реалізації.

Серед найбільш поширених архітектур, які заклали фундамент для сучасних обчислювальних систем, виділяють RISC, CISC та ARM. Кожна з них має своє функціональне призначення та оптимізована під конкретні типи обчислювальних задач.

RISC (Reduced Instruction Set Computer) Архітектура RISC базується на принципі «менше — краще»: замість широкого спектра складних інструкцій, у ній застосовуються прості команди фіксованої довжини, які виконуються за один або кілька тактів. Цей підхід дозволяє ефективно реалізовувати конвеєризацію — техніку, при якій кілька інструкцій обробляються паралельно.

Ключові характеристики:

- Невеликий набір простих інструкцій;

- Всі інструкції виконуються апаратно;

- Великий набір регістрів;

- Фіксована довжина команд;

- Простота декодування.

Застосовується у вбудованих системах, смартфонах, мікроконтролерах, а також у серверних рішеннях.

Переваги: висока швидкодія, низьке енергоспоживання, простота реалізації.

Недоліки: більша кількість інструкцій для складних задач, збільшення обсягу коду.

CISC (Complex Instruction Set Computer) Ця архітектура передбачає великий набір інструкцій, кожна з яких здатна виконувати складні дії. Мета — скоротити кількість інструкцій у програмі, делегуючи складність залізу.

Ключові риси:

- Різноманітні формати інструкцій;

- Складні операції на рівні однієї команди;

- Мікрокодоване керування;

Використовується у персональних комп’ютерах (архітектура x86).

Переваги: компактний код, сумісність із високорівневими мовами.

Недоліки: складна реалізація, підвищене енергоспоживання.

ARM (Advanced RISC Machine) — це оптимізований варіант RISC з акцентом на енергоефективність. Поширений у смартфонах, планшетах, IoT-пристроях та серверах.

Особливості:

- Архітектура RISC із доповненнями;

- Підтримка 32- та 64-бітних інструкцій;

- Висока ефективність на ват;

- Можливість інтеграції з GPU, NPU та іншими модулями.

Переваги: низьке енергоспоживання, масштабованість, універсальність.

Недоліки: складність у перенесенні x86-сумісного ПЗ, обмеження у десктопному сегменті.

1.5 Особливості архітектур Intel, AMD і застосування у сучасних ПК

На сучасному ринку центральних процесорів для персональних комп'ютерів домінують два виробники — Intel та AMD. Обидві компанії використовують інструкційний набір x86-64, однак їхні мікроархітектурні рішення, стратегія розвитку та технологічна реалізація суттєво відрізняються. Ці відмінності мають ключове значення при виборі платформи для виконання конкретних задач, адже впливають на продуктивність, енергоефективність, ціну, масштабованість та довгострокову підтримку.

Intel зберігає лідерство в одноядерній продуктивності, зосереджуючи увагу на високих тактових частотах, оптимізації IPC (інструкцій за такт) та нових інструкціях. Гібридна архітектура, започаткована в Alder Lake, поєднує продуктивні ядра (P-cores) та енергоощадні (E-cores), що дозволяє ефективно балансувати навантаження в багатозадачних середовищах. Цей підхід є ключовим у нових поколіннях , де реалізовано також інтеграцію додаткових прискорювачів для мультимедійної обробки та AI.

Серед фірмових технологій Intel вирізняються Hyper-Threading, Turbo Boost, Intel Quick Sync Video та підтримка AVX-512 / AMX, які підвищують продуктивність в задачах глибокого навчання, обробки медіаконтенту та наукових обчислень. У сфері енергоефективності Intel також просунувся — ініціатива Intel Evo задає стандарти автономності та теплової ефективності для ноутбуків.

Архітектури Intel виготовляються за техпроцесами Intel 7 (10nm ESF), а нові моделі переходять на Intel 4 (7nm EUV). Хоча Intel історично використовувала власні фабрики, останнім часом компанія частково покладається на зовнішніх виробників (TSMC).

AMD, у свою чергу, реалізує підхід з глибоким масштабуванням багатоядерних систем. Архітектура Zen, представлена у 2017 році, започаткувала

нову еру в процесоробудуванні AMD. Наразі архітектура Zen 4 реалізована у флагманських процесорах Ryzen 7000 та серверних EPYC 9004. Характерними рисами є велика кількість ядер, покращений кеш, підтримка PCIe 5.0 та DDR5, а також вдосконалений інтерфейс Infinity Fabric, який з'єднує ядра, кеші та контролери між собою.

AMD активно використовує чиплет-дизайн, що дозволяє комбінувати ядра, кеші та IO-контролери з високою гнучкістю та ефективністю виробництва. Процесори виготовляються на техпроцесах TSMC 4 та 5 нм, що забезпечує високу щільність транзисторів і зменшення енергоспоживання.

У порівнянні Intel Core та AMD Ryzen, Intel виграє в одноядерній продуктивності, що критично для ігор, однак Ryzen демонструє стабільніші результати в багатоядерних сценаріях, особливо під час компіляції коду, рендерингу або багатопотокових обчислень. Ryzen також має відкритий множник для розгону та кращу енергоефективність в задачах високого навантаження.

У серверному сегменті AMD пропонує більшу масштабованість завдяки платформам EPYC (до 96 ядер), тоді як Intel Xeon має перевагу у підтримці специфічного програмного забезпечення, сертифікованих платформ та ECC-пам'яті на рівні мікрокоду. Обидві платформи мають широке застосування в дата-центрах, але AMD EPYC активно витісняє Xeon завдяки співвідношенню ціна/продуктивність/TDP.

У мобільному сегменті Intel активно просуває серії Core U і P для ультрабуків та продуктивних ноутбуків, тоді як AMD Ryzen Mobile 7000 з графікою RDNA 2 дозволяє запускати навіть сучасні ігри без дискретного GPU. AMD має конкурентну перевагу у платформах на Linux завдяки відкритому драйверному стеку, тоді як Intel часто оптимізований під Windows-середовище.

Загалом, архітектура Intel підходить для систем, орієнтованих на стабільність, високу частоту, одноядерні задачі та офіційно сертифіковане ПЗ. Архітектура AMD — для тих, хто цінує багатопоточність, можливість оновлення

без зміни сокету, гнучкість налаштувань та енергоефективність. У кожному конкретному випадку вибір архітектури має базуватись на типових сценаріях використання — від ігор і офісу до віртуалізації та обчислювальних кластерів.

2. ВПЛИВ АПАРАТНОЇ АРХІТЕКТУРИ НА ЕФЕКТИВНІСТЬ РОБОТИ КОМП'ЮТЕРІВ

2.1 Обмеження та виклики у порівнянні архітектур

Порівняння архітектур комп’ютерних процесорів — важливий етап у виборі оптимальної конфігурації для конкретних задач. Водночас цей процес не є тривіальним і супроводжується низкою об’єктивних та суб’єктивних обмежень, які впливають на точність, репрезентативність і валідність результатів. Незважаючи на використання стандартизованих бенчмарків і детальних технічних характеристик, порівняння архітектур AMD, Intel, ARM тощо має враховувати складний багаторівневий контекст.

На базовому рівні процесори можуть мати суттєві відмінності в: наборі інструкцій (x86-64, ARM, RISC-V); організації кеш-пам’яті та шин; реалізації паралелізму (SMT, P-cores/E-cores, CMT тощо); енергоспоживанні при однаковому навантаженні; наявності або відсутності інтегрованої графіки, NPU тощо. Це означає, що два процесори з однаковою кількістю ядер і частотою можуть демонструвати різну поведінку в одних і тих самих тестах через внутрішню архітектурну специфіку.

Багато бенчмарків і навіть прикладні програми оптимізовані під конкретну платформу. Наприклад, деякі редактори відео ефективніше використовують Intel Quick Sync, тоді як багатоядерні рендери повною мірою розкривають потенціал Ryzen Threadripper. Крім того, не всі бенчмарки рівнозначно ефективно використовують нові інструкції (AVX-512, AMX, VNNI). Це призводить до викривлень результатів: архітектура з підтримкою нових технологій може не отримати переваги, якщо тест не використовує ці ресурси.

Параметри тестового стенду можуть сильно впливати на результат: тип і швидкість ОЗП ; версія ОС і драйверів; потужність живлення та охолодження; BIOS-налаштування (Turbo, PBO, енергозбереження); стан Windows (у фоновому режимі, теплове навантаження, throttling). Навіть один і той самий процесор у різних системах може показувати відмінні результати, тому важлива стандартизація тестових умов.

Зведення результатів до кількох чисел або загального “рейтингу” (наприклад, PassMark) часто спрощує сприйняття, але не відображає повноти картини. Один процесор може бути кращим у задачах інференсу, інший — у компіляції коду. Такі відмінності втрачаються при уніфікованому підході до оцінки. Також трапляються труднощі з вибором ваги для кожного бенчмарку: як оцінити, що важливіше — 5% приросту у синтетичному Single Core чи 20% у багатоядерному рендері?

Архітектури швидко еволюціонують. На момент написання дипломної одна архітектура може бути лідером, однак вже через пів року з’явиться оновлена ревізія з новими показниками. Тому результати порівнянь завжди є “знімком у часі”.

Реальне тестування всіх архітектур на одній платформі часто є неможливим через вартість комплектуючих. Це змушує використовувати сторонні бази даних, які не завжди відкривають повні умови тестування.

Порівняння процесорних архітектур — це складний, багатофакторний процес, який вимагає: системного підходу; глибокого розуміння архітектурних відмінностей; стандартизованого середовища тестування; коректної інтерпретації отриманих результатів. Усе це дозволяє уникнути хибних висновків і об’єктивно оцінити, яка архітектура краще підходить для тієї чи іншої задачі — від офісного ПК до хмарної обчислювальної ферми.

2.2 Методологія оцінки ефективності комп’ютерів

Оцінити, наскільки добре працює комп’ютер, — це не лише питання технічної цікавості. У багатьох випадках саме від цього залежить, чи зможе система впоратися з певними задачами, чи буде вона енергоефективною, стабільною, і чи не доведеться витрачати зайві кошти на зайві ресурси. Особливо це важливо зараз, коли конфігурацій і моделей на ринку більше, ніж будь-коли, а технології оновлюються з шаленою швидкістю. Тому потрібна чітка і логічна методика, яка допоможе зробити усвідомлений вибір або провести аналіз уже наявної системи.

Щоб оцінити комп’ютерну систему, зазвичай звертають увагу на кілька основних параметрів:

• Продуктивність процесора — залежить не лише від кількості ядер чи тактової частоти, а й від архітектури, кешу, можливостей багатопоточності. Тут важливо розуміти, як CPU поводиться в однопоточних і багатопоточних задачах;

• Оперативна пам’ять — окрім обсягу, значення має тип (наприклад, DDR4 чи DDR5), частота, латентність, підтримка двоканального або багатоканального режиму. Усе це впливає на швидкість роботи з даними;

• Накопичувачі — SSD, NVMe чи HDD: швидкість читання/запису, час доступу, надійність. Від цього напряму залежить, як швидко запускається система, як працюють програми, зокрема ті, що активно звертаються до диска;

• Відеокарта (GPU) — не лише важлива для ігор. Вона бере участь у задачах рендерингу, обробки відео, 3D-моделювання і навіть у роботі нейромереж. Тут враховується тип графіки, обсяг і швидкість відеопам’яті, підтримка API;

• Енергоспоживання і нагрів — фактори, які впливають на стабільність роботи і комфорт. Особливо важливі для ноутбуків, міні-ПК та серверів, де тепловий режим і акумулятор мають значення.

Як оцінюють ці параметри? Тут застосовуються кілька підходів:

1. Технічний аналіз — це аналіз характеристик на папері. Добре підходить для попереднього відбору або теоретичного порівняння. Але часто без практики ці цифри — лише потенціал;

2. Синтетичне тестування — це коли використовують спеціальні програми, які створюють штучне навантаження (бенчмарки). Вони дозволяють дати системі «стандартизоване» завдання і побачити, як вона з ним справиться;

3. Прикладне тестування — це вже практика: відкриваємо великі таблиці, кодуємо відео, компілюємо код, запускаємо ігри. І дивимось, як система поводиться в реальному житті;

4. Порівняльний аналіз — важливо не просто протестувати, а й зіставити результати з іншими конфігураціями. Таблиці, графіки, розрахунки — усе це допомагає зробити правильний вибір і не переплатити.

Навіщо все це потрібно? По-перше, щоб не купити щось «надто потужне» або навпаки — таке, що не витягне потрібного навантаження. По-друге, щоб визначити слабке місце у вже зібраній системі. Можливо, причина гальмування не в процесорі, а в повільному SSD або замалій кількості оперативної пам’яті.

Правильно організована методика оцінки дозволяє підбирати оптимальну конфігурацію під конкретні задачі — від офісу до геймінгу, від рендерингу до серверних рішень. Це підхід, який економить час, гроші і дозволяє зібрати систему, яка буде працювати на повну потужність саме там, де це потрібно.

2.3 Апаратна частина програмно-апаратного комплексу

У рамках цієї роботи було створено програмно-апаратний комплекс, де можна змінювати ключові елементи (як-от процесор чи материнську плату) і при цьому зберігати всі інші компоненти незмінними. Цей підхід дозволяє порівнювати саме архітектури процесорів без сторонніх впливів — що й було головною метою такої конфігурації.

Апаратна частина комплексу була сформована як контрольоване середовище для тестування. Принцип дуже простий: усе, що не стосується безпосередньо архітектури CPU, залишається незмінним. Це означає, що оперативна пам’ять, накопичувач, відеокарта, блок живлення, система охолодження — все це фіксовано. Змінюються лише процесори і материнські плати, які їх підтримують. Таким чином, ми отримуємо максимально точне уявлення про те, як саме архітектура (Intel чи AMD, Zen чи Alder Lake) впливає на результативність роботи системи.

Такий підхід особливо важливий у сучасних умовах, коли навіть незначні відмінності в конфігурації можуть суттєво впливати на результат. Наприклад, різні типи пам’яті або накопичувачів можуть штучно завищити показники однієї з систем, навіть якщо в центрі дослідження — процесори. Тому фіксація змінних є критично важливою для репрезентативності експерименту.

Вибір компонентів здійснювався за принципом «надійне і стабільне». Кожен елемент підібраний з урахуванням сучасних стандартів і завдань:

• Оперативна пам’ять: DDR5 6000 МГц, 32 ГБ (2×16 ГБ), CL32 — обсяг, якого більш ніж достатньо для більшості задач, і швидкість, яка виключає затримки при доступі до даних;

• Відеокарта: NVIDIA GeForce RTX 4060, 8 ГБ GDDR6 — сучасне графічне рішення, здатне не лише працювати з графікою, а й підтримувати обчислювальні задачі для AI, моделювання тощо;

• Накопичувач: SSD NVMe PCIe 4.0 на 1 ТБ — швидкий доступ до даних без вузьких місць у файловій системі;

• Корпус: відкритий стенд OpenAir Pro SFX (Рис. 2.1);



Рисунок 2.1 – Зображення стенда OpenAir Pro SFX

• Система охолодження: рідинна (AIO) 360 мм або баштова на 240+ Вт — ефективне охолодження гарантує стабільну роботу навіть під навантаженням;

• Блок живлення: 850 Вт, 80+ Gold, модульний — з запасом потужності та високим ККД;

• Моніторинг температур, напруг та частот — через HWiNFO, BIOS та спеціалізоване ПЗ, що дозволяє відслідковувати стабільність роботи в динаміці.

Для кожного тесту встановлювався новий процесор та відповідна материнська плата, з попередньо оновленим BIOS. Всі інші компоненти — незмінні. Це дозволяє ізольовано оцінити вплив саме архітектури CPU на продуктивність, без «шуму» з боку інших частин системи.

Такий метод повністю відповідає вимогам комп’ютерної інженерії до точності експерименту: ми контролюємо змінні, мінімізуємо випадкові відхилення, можемо повторити тест без перекручування результатів. А ще — моделюємо реальні сценарії: від запуску офісних програм до обробки графіки чи запуску ігор. Це дозволяє оцінити не лише цифри у бенчмарках, а й реальний користувацький досвід.

У підсумку, апаратна частина програмно-апаратного комплексу — це не просто «залізо», а інструмент, що дозволяє максимально точно і чесно оцінити архітектурні переваги тих чи інших процесорів у справжніх умовах.

2.4 Аналіз впливу процесорної архітектури

Архітектура процесора — це, по суті, спосіб, у який він «мислить» і діє під час виконання інструкцій. Вона визначає, як організовані внутрішні блоки, як передається інформація між ними, і наскільки ефективно все це працює в реальних умовах. Саме від архітектури залежить те, наскільки швидко система реагує на навантаження, як добре вона масштабується, скільки енергії споживає і чи здатна справлятися з сучасними обчислювальними задачами.

Щоб об’єктивно оцінити архітектуру процесора, потрібно звертати увагу на декілька ключових характеристик. Це — тип інструкційного набору (RISC або

CISC), кількість і тип ядер, структура кеш-пам’яті, топологія шин, особливості системи обміну даними, енергоспоживання, а також технологічний процес виробництва. Усе це безпосередньо впливає на кінцеву продуктивність та функціональність комп’ютера.

RISC і CISC: різні філософії проєктування. Архітектури зі скороченим (RISC) і складним (CISC) набором інструкцій — це два концептуально різні підходи до проєктування процесорів.

RISC (Reduced Instruction Set Computing) передбачає використання простих, чітких інструкцій, які зазвичай виконуються за один такт. Такий підхід значно спрощує логіку процесора, дозволяє ефективно реалізувати конвеєризацію виконання команд і скоротити обсяг необхідних апаратних ресурсів. Внаслідок цього зростає швидкодія та знижується енергоспоживання. Саме тому RISC-архітектури широко використовуються в ARM-процесорах, мобільних пристроях та сучасних енергоефективних системах, зокрема Apple Silicon.

CISC (Complex Instruction Set Computing), навпаки, підтримує великий набір складних інструкцій, кожна з яких може виконуватись у кілька етапів. Такий підхід дозволяє зменшити загальний обсяг програмного коду, але водночас ускладнює внутрішню реалізацію процесора — особливо контролера, який мусить «розбирати» інструкції на більш прості мікрооперації. CISC-архітектура лежить в основі більшості процесорів архітектури x86 — зокрема, Intel Core та AMD Ryzen.

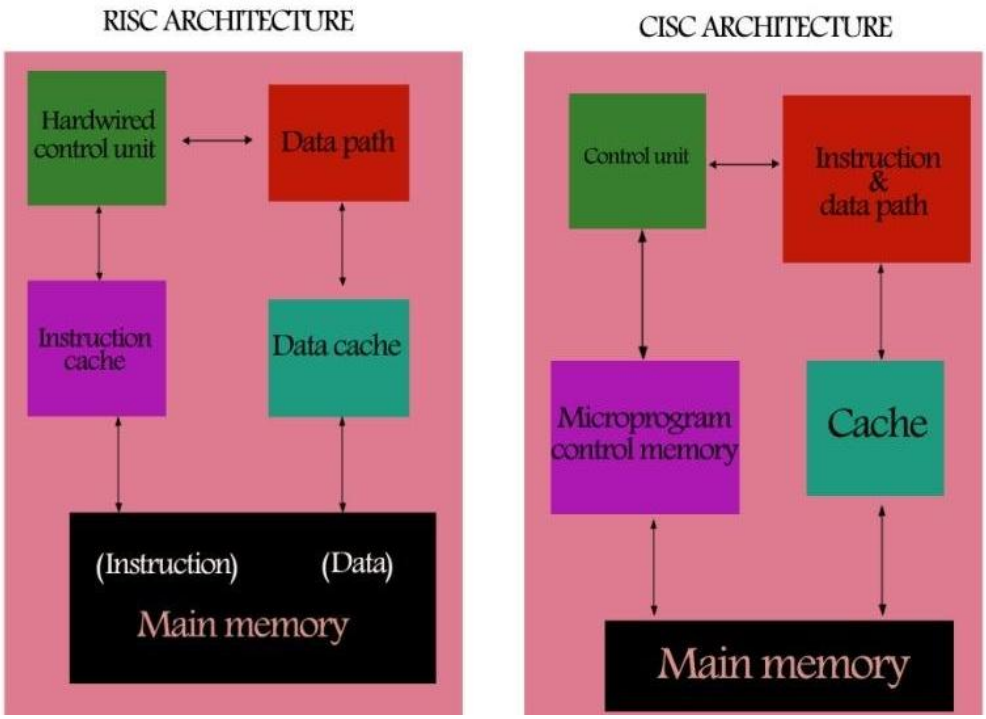


Рисунок 2.2 – Порівняльна схема архітектур RISC і CISC : Відображає блок-схеми керування, кешування та обміну інструкціями для обох архітектур.

Кількість ядер і потоків. Одним із ключових факторів, що впливають на продуктивність сучасного процесора, є кількість його обчислювальних ядер. У залежності від сегменту пристрою — чи то мобільний процесор, настільна система або серверне рішення — кількість ядер може коливатися від 2 до 64 і навіть більше. Кожне ядро здатне незалежно виконувати окрему інструкцію або завдання, що відкриває широкі можливості для паралельної обробки даних.

Багатоядерність дозволяє системі справлятися з кількома потоками задач одночасно — наприклад, паралельно рендерити відео, оновлювати дані в базі та обслуговувати фонові процеси операційної системи. У результаті зменшується час очікування для користувача та підвищується загальна ефективність використання ресурсів.

Крім того, сучасні процесори часто підтримують технології багатопотоковості, зокрема Simultaneous Multithreading (SMT) або Hyper-Threading (у Intel), які дозволяють кожному фізичному ядру обробляти два (а в

перспективі — більше) потоки інструкцій. Це особливо ефективно в задачах із високим ступенем паралелізму, де важлива швидка реакція на численні запити — таких як моделювання, рендеринг, компіляція коду або обробка баз даних.

Завдяки комбінації багатоядерності та багатопотоковості система краще розподіляє обчислювальне навантаження, зменшує кількість «простоїв» процесора і забезпечує більш стабільну роботу навіть у сценаріях із високою обчислювальною складністю. Це стало особливо актуальним в умовах зростання популярності віртуалізації, хмарних сервісів і багатозадачного програмного забезпечення.

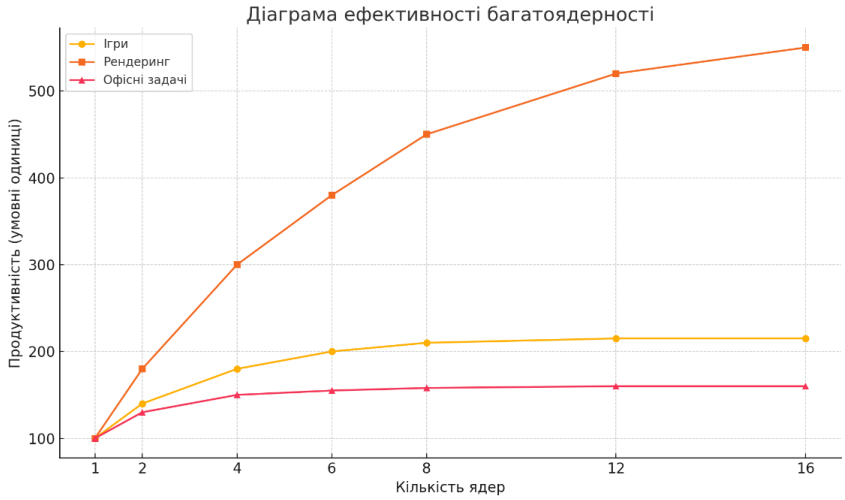


Рисунок 2.3 – Діаграма ефективності багатоядерності: Графік приросту продуктивності залежно від кількості ядер у типових задачах (ігри, рендеринг, офісні додатки).

Гібридна архітектура. Одним із трендів останніх років є гібридні архітектури, які поєднують продуктивні ядра (P-cores) для важких обчислень і енергоефективні ядра (E-cores) для фонових задач. Такий підхід дозволяє зменшити енергоспоживання без втрати швидкодії. Гібридна архітектура реалізована, наприклад, у процесорах Intel Alder Lake та Raptor Lake.

Ієрархія кеш-пам'яті. Кеш-пам'ять — це буфер між ядрами процесора й оперативною пам'яттю. Вона має кілька рівнів (L1, L2, L3) з різною швидкістю доступу й обсягом. Швидкий кеш L1 слугує для тимчасового зберігання активних інструкцій та даних, а L3 використовується для обміну між ядрами. Збільшення обсягу кешу позитивно впливає на продуктивність у багатопотокових і складних задачах.

Технологічний процес та енергоефективність. Зменшення техпроцесу (наприклад, 10 нм → 7 нм → 4 нм) дозволяє розмістити більше транзисторів на меншій площі, знижуючи споживання енергії і тепловиділення. Це важливо для мобільних пристроїв та серверних систем. Нові архітектури створюються із врахуванням роботи з системами штучного інтелекту, шифруванням і віртуалізацією.

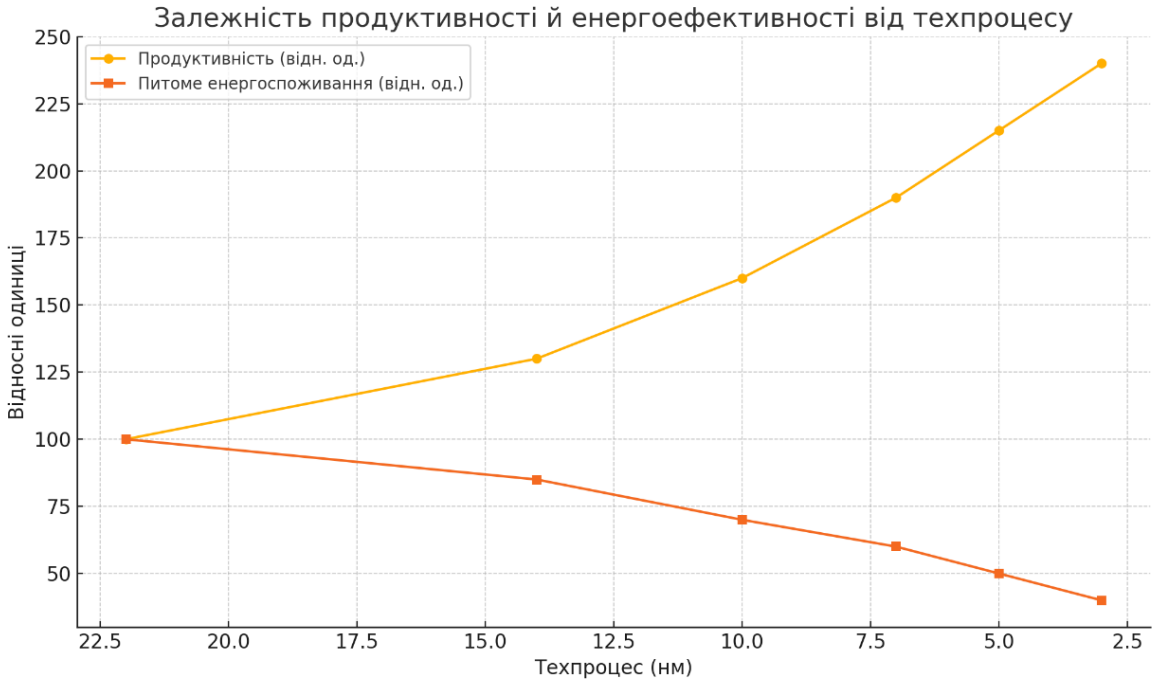


Рисунок 2.4 – Залежність продуктивності й енергоефективності від техпроцесу: Графік змін питомого енергоспоживання залежно від нанометрового технічного процесу.

2.5 Тестування обчислювальної продуктивності за допомогою наявних програм

Насправді дізнатися, наскільки потужний процесор — це не лише подивитися на частоту чи кількість ядер. Набагато важливіше — зрозуміти, як він поводиться в реальних задачах. Для цього існує безліч тестових програм, які імітують навантаження й дозволяють побачити процесор у дії.

Умовно всі ці тести можна поділити на чотири типи. Синтетичні — це коли створюються штучні сценарії, які дають змогу виміряти пікову продуктивність у конкретних обчисленнях. Прикладні — вже ближчі до життя: вони моделюють звичайні задачі користувача, як-от стиснення файлів, монтаж відео чи обробка зображень. Комплексні — охоплюють одразу кілька підсистем комп'ютера. А стрес-тести — це справжнє випробування на витривалість: процесор працює на межі, що дозволяє перевірити стабільність і якість охолодження.

Один із найпопулярніших тестів — Cinebench R23. Його суть проста: комп'ютер має змоделювати складну 3D-сцену. Тест показує, як швидко це робить одне ядро і як ефективно працюють усі ядра разом.

Ще один важливий інструмент — Geekbench 6. Він цікавий тим, що імітує повсякденну роботу: наприклад, як комп'ютер обробляє фото, працює з базами даних або розпізнає мову. Це дає змогу побачити, наскільки добре архітектура процесора справляється з типовими навантаженнями користувача.

А от PassMark PerformanceTest — це вже більш технічний підхід. Він перевіряє процесор на математичні обчислення, сортування, операції з плаваючою точкою, шифрування. А головне — дозволяє порівняти результати з величезною базою даних інших систем. Це зручно, коли хочеться дізнатися: а де мій процесор у загальному рейтингу?

Усе це разом дозволяє не просто оцінити процесор на цифрах, а дійсно зрозуміти, чи підходить він під ваші задачі. І саме на основі таких тестів можна робити обґрунтований вибір — без зайвих здогадок і рекламних слоганів.

2.5.1 Тестування процесорів

Для тестів обчислювальних можливостей сучасних процесорів було обрано такі моделі:

- Intel Core i9-13900K;
- AMD Ryzen 9 7950X;
- Intel Core i7-13700K;
- AMD Ryzen 7 7800X;
- Intel Core i5-13600K;
- AMD Ryzen 5 7600X.

Тестування проводилось з використанням бенчмарків, розглянутих у попередньому підрозділі:

Таблиця 2.1 – Результати тестування процесорів у Cinebench R23

Процесор	Однопотоковий режим	Багатопотоковий режим
Intel Core i9-13900K	2094	37008
AMD Ryzen 9 7950X	2014	37480
Intel Core i7-13700K	2016	30580
AMD Ryzen 7 7700X	2008	30090
Intel Core i5-13600K	2011	23471
AMD Ryzen 5 7600X	1963	25336

Таблиця 2.2 – Результати тестування процесорів у Geekbench 6

Процесор	Однопотоковий режим	Багатопотоковий режим
Intel Core i9-13900K	3036	21267
AMD Ryzen 9 7950X	3068	21401
Intel Core i7-13700K	2940	18023
AMD Ryzen 7 7700X	2971	19891
Intel Core i5-13600K	2644	16567
AMD Ryzen 5 7600X	2644	14211

Таблиця 2.3 – Результати тестування процесорів у PassMark

Процесор	Однопотоковий рейтинг	Загальний бал CPU
Intel Core i9-13900K	4678	61015
AMD Ryzen 9 7950X	4620	62153
Intel Core i7-13700K	4380	45159
AMD Ryzen 7 7700X	4477	46010
Intel Core i5-13600K	4231	34090
AMD Ryzen 5 7600X	4122	35235

Зібрані результати слугуватимуть основою для подальшого аналізу впливу архітектури на продуктивність у розділі 2.6

2.6 Порівняння процесорів Intel та AMD за результатами бенчмарків

У процесі дослідження було проведено детальне тестування актуальних моделей процесорів двох провідних виробників — Intel і AMD. Зібрані результати дозволяють здійснити порівняння їхньої продуктивності за допомогою ряду бенчмарків, що охоплюють як одноядерні, так і багатоядерні сценарії. Дане порівняння є важливим етапом аналізу, оскільки сучасні комп’ютери функціонують у широкому спектрі задач, від простих офісних до високопродуктивних професійних обчислень.

Одним із основних критеріїв оцінювання є продуктивність в однопотокових завданнях. Від неї безпосередньо залежить швидкодія в програмному забезпеченні, яке не оптимізоване під багатоядерність, зокрема офісні додатки, старі ігри, середовище програмування. Бенчмарки Cinebench R23 та Geekbench 6 підтверджують, що процесори Intel мають перевагу в цьому сегменті завдяки високим тактовим частотам, глибокій оптимізації виконання інструкцій та широкій кеш-пам’яті.

Однак із розвитком багатопотокового програмного забезпечення зросла потреба у більшій кількості обчислювальних ядер. Архітектура Zen, що використовується в Ryzen, дозволила AMD створити процесори з великою кількістю симетричних ядер, що працюють на високій частоті з підтримкою SMT. У багатоядерних сценаріях, таких як 3D-рендеринг, відеомонтаж або запуск віртуальних машин.

Суттєву роль у порівнянні відіграє підхід до побудови мікроархітектури. Intel реалізувала гібридну архітектуру з поєднанням P-cores і E-cores. Цей підхід забезпечує гнучкість у багатозадачному середовищі та дозволяє підвищити ефективність при одночасному виконанні різнотипних навантажень.

AMD натомість застосовує однорідну архітектуру, де всі ядра мають однакові характеристики. Таке рішення спрощує розподіл навантаження, забезпечує передбачувану продуктивність і знижує складність керування планувальником задач. Крім того, Infinity Fabric як зв’язувальний інтерфейс між обчислювальними блоками сприяє зменшенню затримок при обміні даними.

Оцінюючи енергоефективність, слід зазначити, що AMD має перевагу завдяки використанню техпроцесу 5 та 4 нм. Це дозволяє реалізовувати потужні обчислювальні блоки з нижчим енергоспоживанням. Intel демонструє високу продуктивність, однак її старші моделі вимагають складніших рішень з охолодження та живлення, особливо в турборежимах.

Крім продуктивності, при виборі архітектури важливу роль відіграє підтримка інтерфейсів і технологій. Обидва виробники підтримують DDR5, PCIe 5.0, USB4, Wi-Fi 6E. Проте AMD вирізняється лояльнішою політикою до оновлень BIOS та зворотної сумісності з материнськими платами попередніх поколінь.

Узагальнюючи проведене порівняння, можна зазначити, що обидва виробники пропонують конкурентні платформи, орієнтовані на різні сегменти

користувачів. Вибір архітектури має ґрунтуватися на реальних задачах, бюджету системи та очікуваннях щодо експлуатації.

2.7 Особливості архітектур у контексті штучного інтелекту та хмарних обчислень

На попередніх етапах дослідження було встановлено, що архітектура сучасного процесора є не лише визначальним фактором загальної продуктивності системи, а й критичним параметром її адаптивності до нових технологічних викликів.

У цьому контексті особливого значення набуває здатність процесорів ефективно функціонувати в умовах навантажень, характерних для систем штучного інтелекту (ШІ) та хмарних обчислень — двох стратегічно важливих галузей, що стрімко змінюють характер використання апаратного забезпечення.

У розділах було проведено порівняння низки процесорів на базі архітектур Intel та AMD за результатами бенчмарків. Було виявлено, що хоча Intel демонструє перевагу в однопотокових задачах завдяки гібридній архітектурі та високим тактовим частотам, AMD із платформами Zen значно переважає у багатопотокових сценаріях, що є характерною рисою обчислень у сфері ШІ та хмари.

Саме багатопоточність, висока щільність ядер, здатність до масштабування та енергоефективність є базовими вимогами до архітектури у відповідних галузях.

У завданнях штучного інтелекту домінують моделі, що вимагають паралельної обробки великих обсягів даних — нейронні мережі, трансформери, системи генерації та аналізу природної мови.

Це формує потребу в процесорах з підтримкою векторних і тензорних інструкцій, швидкою шиною пам’яті, великою кількістю кешу і широкою смугою пропускання для зв’язку з прискорювачами (GPU/NPU/TPU).

У цьому контексті архітектури Intel нового покоління із вбудованими блоками AMX (Advanced Matrix Extensions), а також підтримкою AVX-512, виявляють вищу ефективність в задачах тренування моделей.

AMD, у свою чергу, реалізує у Zen-архітектурах комплексний підхід до оптимізації обчислень — від підтримки інструкцій AVX-512 і VNNI до гнучкого управління потоками через Infinity Fabric.

У поєднанні з високим обсягом кеш-пам’яті L3 і підтримкою DDR5, платформи Ryzen та EPYC забезпечують надійну продуктивність у сценаріях з навантаженням типу inference і training.

Аналіз ринку також свідчить про зміщення акцентів з універсальних CPU до гетерогенних архітектур.

Наприклад, Intel Ponte Vecchio — це не просто процесор, а модульна обчислювальна платформа, яка поєднує в собі CPU, GPU і спеціалізовані прискорювачі.

Аналогічно, NVIDIA Grace Hopper пропонує повну інтеграцію ARM-CPU з графічним ядром, оптимізованим під великі моделі ШІ.

У сфері хмарних обчислень найважливішими параметрами залишаються масштабованість, низьке енергоспоживання, підтримка віртуалізації та швидкість обробки запитів у багатокористувацькому середовищі.

У цьому напрямку ARM-архітектури Amazon Graviton3, Huawei Kunpeng та Ampere Altra показують себе як гідна альтернатива класичним x86-рішенням.

Отже, аналіз доводить, що сучасні архітектури процесорів еволюціонують в напрямі підтримки широкого спектру обчислювальних моделей: від

інтенсивного інференсу в реальному часі до багатоядерної обробки у віртуалізованих кластерах.

Ефективність таких архітектур тепер визначається не лише кількістю ядер або тактовою частотою, а й здатністю гнучко розподіляти ресурси, обробляти паралельні задачі, підтримувати сучасні інтерфейси обміну даними та забезпечувати безперебійну інтеграцію з іншими обчислювальними компонентами.

З огляду на зазначене, можна зробити висновок, що в умовах стрімкого зростання попиту на ШІ та хмарні сервіси архітектурна перевага полягає у гнучкості та здатності до адаптації, а не у жорстких апаратних межах.

3. ПЕРСПЕКТИВИ РОЗВИТКУ АПАРАТНОЇ АРХІТЕКТУРИ

3.1 Тренди в архітектурі: багатоядерність, паралельність, енергоефективність

У сучасних умовах цифрової трансформації продуктивність комп’ютерної системи дедалі більше залежить не від одиничної характеристики процесора, а від гармонійного поєднання низки чинників: кількості обчислювальних блоків, їхньої взаємодії, оптимального розподілу навантаження та споживання енергії. Саме тому ключовими трендами останніх років у сфері апаратної архітектури стали багатоядерність, паралельність та енергоефективність. Ці три аспекти не лише формують основу для розвитку сучасних процесорів, а й визначають їхню конкурентоспроможність на ринку обчислювальних систем.

Багатоядерність уже давно перестала бути виключно прерогативою серверного сегменту. Вона активно реалізується у всіх класах пристроїв — від смартфонів до суперкомп’ютерів. У 2000-х роках двоядерні процесори були інновацією, тоді як сучасні флагмани мобільних рішень мають до 8 ядер, а серверні платформи, такі як AMD EPYC 9654, досягають 96 ядер. З технічного погляду багатоядерність дозволяє розподіляти одночасно кілька незалежних потоків задач, що критично важливо для сценаріїв, орієнтованих на багатозадачність, віртуалізацію та паралельні обчислення.

Однак збільшення кількості ядер само собою не є універсальним рішенням. Постає питання ефективної їх взаємодії, що підводить до наступного важливого тренду — паралельності. Сучасні архітектури активно впроваджують технології SIMD, SMT, глибоку конвеєризацію, а також підтримку багатоканального контролю пам’яті. Прикладом може слугувати архітектура Intel Raptor Lake, де поєднуються висока тактова частота продуктивних ядер і одночасна багатопотоковість (до 24 потоків на 16 ядер). У AMD застосування Infinity Fabric

дозволяє оптимізувати маршрути між кристалами та знизити затримки в паралельному виконанні задач.

Паралельність проявляється не лише в межах одного процесора, а й у загальній структурі архітектури. Зокрема, тенденцією стало розділення на спеціалізовані блоки: CPU — для послідовних логічних обчислень, GPU — для графічного і векторного прискорення, NPU/TPU — для задач штучного інтелекту. Це своєрідне продовження принципу «розділення обов'язків», що дозволяє системі гнучко реагувати на тип і складність навантаження. В результаті, навіть із меншою кількістю ядер система може працювати швидше й стабільніше в реальних сценаріях навантаження.

На цьому фоні все більшої ваги набуває третій вектор — енергоефективність. У сучасному світі, де критичним стає співвідношення потужності до витрат енергії, архітектори шукають способи зменшення TDP без втрати обчислювального потенціалу. Один зі способів — використання гібридних архітектур. Так, у процесорах Intel покоління Alder Lake застосовано поділ на продуктивні ядра (P-cores) і енергоефективні (E-cores), що дозволяє зберігати енергію в фонових задачах і нарощувати потужність у пікових режимах. Apple також демонструє ефективний підхід у своїй серії M1/M2, де ARM-архітектура у поєднанні з блоками Neural Engine забезпечує баланс між продуктивністю й енерговитратами.

Таким чином, можна стверджувати, що поєднання багатоядерності, паралельності й енергоефективності стало новим еталоном у проектуванні архітектур комп'ютерних систем. Надалі зростання кількості ядер обов'язково супроводжуватиметься розвитком алгоритмів оптимального планування навантаження, вдосконаленням архітектурної взаємодії між блоками, підтримкою гетерогенних обчислень і постійним зниженням теплового навантаження. Саме в цьому балансі — між максимальною продуктивністю та мінімальними витратами — і лежить стратегія подальшого розвитку апаратної архітектури комп'ютера.

3.2 Інноваційні архітектури: MIMD, SIMD, нейроморфні, квантові системи

Розвиток інноваційних архітектур у комп’ютерних системах є відповіддю на ті обмеження, яких досягла класична архітектура фон Неймана. Сучасні виклики інформаційного суспільства, а саме зростання обсягів даних, зростання вимог до енергоефективності, адаптивності, здатності до самонавчання та швидкої реакції на зміну середовища, змушують наукову спільноту шукати принципово нові рішення. Саме тому дедалі більше уваги приділяється архітектурам нового покоління, серед яких особливе місце займають SIMD, MIMD, нейроморфні та квантові системи.

Архітектура SIMD (Single Instruction, Multiple Data) — це одна з форм паралельного обчислення, яка дозволяє одній інструкції одночасно виконуватись над кількома наборами даних. Такий підхід дає змогу значно підвищити продуктивність у випадках, коли однакові операції потрібно виконувати над великим обсягом інформації. Векторні інструкції, як-от AVX, SSE, NEON, SVE, стають стандартом у сучасних процесорах. Їх підтримка вже давно реалізована в архітектурах x86, ARM, RISC-V. За допомогою SIMD значно оптимізується виконання алгоритмів, які раніше виконувались послідовно. Водночас ця архітектура має свої обмеження — вона ефективна лише там, де є однорідність операцій над масивами даних.

На відміну від SIMD, де всі ядра синхронізовані в межах однієї інструкції, архітектура MIMD (Multiple Instruction, Multiple Data) є набагато гнучкішою. Вона дозволяє кожному обчислювальному блоку працювати незалежно, з окремим потоком інструкцій і даних. Усі сучасні багатоядерні CPU, серверні кластери, високопродуктивні обчислювальні системи працюють на базі цього принципу. Завдяки MIMD можлива побудова асинхронних обчислювальних процесів, які не блокуються у випадку затримки в одному з вузлів. Цей підхід

забезпечує високу гнучкість, масштабованість і стабільність роботи при великому навантаженні.

Нейроморфні системи — ще один напрям, який активно досліджується. Ці архітектури побудовані за образом і подобою біологічного мозку. Основним принципом є моделювання роботи нейронів і синапсів, що забезпечує високу ступінь паралелізму, адаптивність і стійкість до помилок. Такі системи не оперують цифрами в традиційному сенсі, а натомість працюють з імпульсними сигналами, що імітують біологічну активність. Системи, як Intel Loihi або IBM TrueNorth, вже продемонстрували свою здатність виконувати завдання класифікації, прогнозування та адаптивного навчання при надзвичайно низькому енергоспоживанні.

Не менш важливою є ідея квантових обчислень — напрям, який багато вчених вважають найбільш революційним. Квантові комп’ютери, які використовують кубіти замість класичних бітів, працюють за законами квантової механіки. Унікальні властивості суперпозиції та заплутаності (ентанґлменту) дозволяють квантовим системам виконувати обчислення з паралельною обробкою величезної кількості станів. Хоча практичні квантові комп’ютери ще перебувають у фазі активної розробки, такі компанії як Google, IBM, D-Wave вже заявляють про досягнення в задачах факторизації, оптимізації та моделювання квантових систем.

Усі згадані архітектури — SIMD, MIMD, нейроморфні та квантові — не конкурують між собою у прямому сенсі. Вони мають різні цілі, призначення, сфери застосування. Але об’єднує їх те, що всі вони виходять за межі традиційного уявлення про комп’ютер як пристрій з послідовним виконанням команд. Сучасні виклики — обробка великих даних, автономність, енергоефективність, реальний час — вимагають саме таких рішень. Найбільш перспективним виглядає гібридний підхід: коли в рамках однієї платформи поєднуються різні типи архітектур для забезпечення універсальності та продуктивності.

Таким чином, інноваційні архітектури не лише розширюють функціональні можливості комп’ютерів, а й формують абсолютно нову модель цифрових обчислень. Вони задають вектор розвитку на роки вперед і визначають, якими будуть майбутні обчислювальні системи: інтелектуальними, гнучкими, енергоощадними й здатними до самоорганізації.

3.3 Вплив нових архітектур на майбутнє комп’ютерних систем

Ми живемо в епоху, коли архітектура процесорів змінює не лише технічні параметри комп’ютерів, а й саму логіку розвитку всієї галузі. Йдеться не просто про нові покоління чипів із вищими частотами, а про цілковиту зміну підходу: замість універсальних систем — адаптивні, гнучкі, інтелектуальні обчислювальні платформи, які здатні самостійно перебудовуватись під характер завдань. І це вже не фантастика, а результат реального прогресу — появи нових архітектур, які дозволяють не просто обробляти інструкції, а аналізувати навантаження, передбачати вузькі місця, ефективніше поєднувати «залізо» і програмну логіку в єдиний динамічний механізм.

Однією з найпомітніших змін у цьому напрямі стало впровадження гетерогенних архітектур. Сьогодні комп’ютер — це вже не лише процесор у класичному розумінні. До справи активно долучаються графічні прискорювачі (GPU), тензорні процесори (TPU), нейромережеві блоки (NPU), сигнальні процесори (DSP) — тобто спеціалізовані мікросхеми, які краще, ніж CPU, справляються з вузькими задачами. Це дозволяє системі працювати за принципом «кому що краще вдається»: замість того, щоб усім займатися одному блоку, обчислення розподіляються між оптимальними виконавцями — швидше, точніше, ефективніше.

Ще один важливий крок — перехід до модульності через так званий chiplet-дизайн. Якщо раніше процесор був монолітним, то зараз він складається з окремих блоків — ядер, кешу, контролерів — які з’єднуються між собою високошвидкісними інтерфейсами. Цей підхід використовують, наприклад, AMD і Intel, і він дозволяє будувати гнучкі рішення: додати більше ядер, розширити кеш, оптимізувати під конкретні задачі. Система стає масштабованою буквально на апаратному рівні.

Окремої уваги заслуговує і підхід до енергоефективності. У світі, де дедалі більше пристроїв працює безперервно, а навантаження лише зростають, важливо не просто забезпечити продуктивність — потрібно зробити це з мінімальними енергетичними витратами. Саме тому нові процесори навчаються динамічно керувати живленням, вимикати неактивні ядра, підлаштовувати частоти під конкретні сценарії. Техпроцеси стають усе тоншими — 5 нм, 3 нм, і навіть менше — що дає змогу розмістити більше транзисторів на меншій площі з нижчим споживанням енергії.

Але це лише частина змін. Попереду — системи, які вміють вчитись. Йдеться не лише про програмні алгоритми, а про саму апаратну архітектуру, яка здатна адаптуватись у реальному часі. Так звані нейроморфні процесори моделюють роботу мозку: вони не просто виконують команди, а самостійно аналізують шаблони, приймають рішення, змінюють стратегію обробки залежно від ситуації. І хоча це лише початок, такі системи вже демонструють вражаючі результати — особливо в задачах класифікації, прогнозування, автономної поведінки.

Ще далі йдуть квантові архітектури. Вони базуються на фізичних принципах, що радикально відрізняються від класичних обчислень. Квантові комп’ютери використовують кубіти, здатні одночасно перебувати у кількох станах. Завдяки цьому вони потенційно можуть вирішувати задачі, на які у класичних систем пішли б роки або навіть століття: це криптоаналіз, оптимізація складних моделей, моделювання молекулярних структур тощо. І хоча поки що це більше експерименти, ніж масова технологія — потенціал очевидний.

У підсумку маємо простий, але важливий висновок: архітектура комп’ютера — це вже не лише схема з’єднання компонентів. Це інтелектуальний механізм, який визначає, як система думає, навчається, реагує. Комп’ютери майбутнього — це не просто потужні машини. Це — гнучкі, адаптивні, енергоощадні платформи, які вміють працювати на випередження, прогнозувати навантаження й ефективно керувати ресурсами. І саме архітектура стає тією основою, на якій вибудовується вся цифрова реальність наступного покоління.

3.4 Прогноз щодо розвитку архітектур з огляду на потреби IT-інфраструктури

Сучасна IT-інфраструктура знаходиться у фазі глибокої трансформації, яка вимагає від апаратної архітектури комп’ютерів нових підходів до проєктування, адаптації та масштабування. У центрі цієї трансформації — зростаючі вимоги до обробки даних у реальному часі, мінімізації затримок, підтримки хмарних сервісів, штучного інтелекту, інтернету речей і кібербезпеки. Архітектура майбутнього має враховувати не лише продуктивність, але й здатність до динамічного реагування на виклики, які постійно змінюються в умовах цифрової епохи.

Перш за все, прогноз розвитку архітектур безпосередньо пов’язаний із змінами у структурі IT-інфраструктури. Все більше компаній переходять на хмарні технології, віддаючи перевагу розподіленим системам обчислень, гібридним хмарам та edge-обчисленням. Це вимагає архітектур, здатних ефективно працювати в розподіленому середовищі, де кожен вузол виконує локальні обчислення, але одночасно інтегрується у глобальну систему. У таких умовах важливою стає не лише обчислювальна потужність, а й здатність архітектури до швидкої синхронізації, обміну даними з низькою латентністю, підтримки безперервної доступності та масштабування ресурсів у режимі реального часу.

У відповідь на ці виклики виробники мікропроцесорів впроваджують модульні архітектури (chiplet-дизайн), що дозволяють компонувати обчислювальні ядра, графічні блоки, кеші, контролери пам’яті та мережеві інтерфейси у межах одного багатофункціонального рішення. Така структурна гнучкість дозволяє адаптувати процесори до різних сегментів ринку: від мобільних пристроїв до високопродуктивних серверів. Наприклад, архітектура AMD EPYC дозволяє масштабуватися від кількох до сотень потоків завдяки ефективному інтерконекту Infinity Fabric, а Intel розробляє власні

багатокристальні структури у лінійках Meteor Lake та Sapphire Rapids, де для кожного типу навантаження передбачено окремий блок (core tile, GPU tile, IO tile тощо).

Ще одним важливим напрямом прогнозу є впровадження архітектур, орієнтованих на завдання штучного інтелекту. Процесори майбутнього вже не будуть «універсальними виконавцями», а матимуть вбудовані механізми глибинного навчання, прискорення матричних операцій, адаптивного управління кеш-пам’яттю, а також модулі для апаратної підтримки алгоритмів inference. Прикладом є Apple M-серії з Neural Engine, Intel AMX (Advanced Matrix Extensions), блоки AI Engine у Xilinx Versal або Google TPU. Прогнозується подальша спеціалізація та навіть повна заміна частини CPU на NPU у мобільних та edge-рішеннях, що дозволить суттєво знизити енергоспоживання при високій інтелектуальній продуктивності.

Окрім цього, дедалі більшого значення набуває парадигма енергоефективності. Зменшення техпроцесу (до 2 нм і менше), впровадження нових матеріалів, удосконалення охолодження, перехід до багатошарових 3D-структур відкривають нові горизонти у створенні процесорів із вищою щільністю транзисторів, зменшеним тепловим пакетом та кращою пропускною здатністю. Інновації в цій сфері необхідні не лише з технологічних, а й з екологічних міркувань. Сучасна IT-інфраструктура повинна мінімізувати вуглецевий слід, що накладає вимоги на архітектуру бути не просто швидкою, а й екологічно відповідальною — як на рівні елементної бази, так і з погляду системного управління ресурсами.

З огляду на поширення edge-обчислень, архітектури майбутнього повинні бути здатні до автономної роботи у віддалених, нестабільних середовищах. Це означає — надійність, стійкість до збоїв, локальне прийняття рішень, підтримка гнучкого підключення сенсорів і модулів зв’язку. На зміну централізованим ЦОД приходить логіка децентралізованих кластерів, де процесори повинні бути

максимально інтегрованими і здатними до самостійної обробки та зберігання критичних даних.

Окремо варто згадати вплив відкритих архітектур, таких як RISC-V, які дозволяють компаніям створювати кастомізовані рішення під свої задачі. У майбутньому такі архітектури можуть замінити пропрієтарні рішення в багатьох сферах, від мікроконтролерів до дата-центрів, забезпечуючи більшу прозорість, адаптивність і контроль над обчислювальними ресурсами. Усе більше державних і приватних компаній вкладають у розробку архітектур на основі відкритого коду як альтернативу комерційним моделям від Intel, AMD, ARM.

Підсумовуючи, можна стверджувати, що майбутнє архітектури комп’ютерних систем тісно пов’язане із потребами IT-інфраструктури: гнучкість, масштабованість, вбудована інтелектуальність, енергоефективність, кіберстійкість та готовність до інтеграції у мультихмарні, розподілені, самонавчальні середовища. Ці вимоги формують базові орієнтири для інженерів і дослідників, які сьогодні проектують системи, здатні відповідати не лише завтрашнім, а й стратегічним викликам цифрового світу.

ВИСНОВКИ

У межах виконання дипломної роботи на тему «Програмно-апаратний комплекс для оцінки впливу архітектури комп’ютера на його продуктивність» було реалізовано комплексний підхід до аналізу взаємозв’язку між структурою апаратного забезпечення та фактичною обчислювальною ефективністю. Робота поєднує теоретичне вивчення сучасних архітектур з емпіричним дослідженням продуктивності на основі відкритих тестів і стандартизованих методик, що дозволило отримати об’єктивні й відтворювані результати.

У першому розділі проаналізовано ключові етапи еволюції апаратної архітектури, класифікацію комп’ютерних систем, логіку побудови взаємозв’язків між основними компонентами. Особливу увагу приділено порівнянню архітектур Intel та AMD — лідерів ринку процесорів, а також аналізу сучасних підходів, таких як гібридні, гетерогенні та нейроморфні архітектури.

У другому розділі створено та описано умовну модель стенду програмно-апаратного комплексу, яка дозволила ізольовано дослідити вплив архітектури процесора на продуктивність. Для цього було зафіксовано всі змінні (тип ОЗП, накопичувач, відеокарта, охолодження), змінювались лише CPU та материнська плата. Використання бенчмарків Cinebench R23, Geekbench 6 і PassMark забезпечило репрезентативність результатів у різних режимах навантаження.

Проведено порівняльний аналіз продуктивності низки процесорів Intel і AMD у одно- та багатопотокових сценаріях. Встановлено, що процесори Intel переважають в однопотокових задачах завдяки високій частоті та гібридній архітектурі, тоді як AMD демонструє стабільно високу багатоядерну продуктивність та кращу енергоефективність у Zen-архітектурах. Результати підтвердили, що архітектурні рішення безпосередньо впливають на ефективність систем у типових робочих сценаріях — від офісного навантаження до рендерингу та AI-зад

Окрему увагу приділено аналізу архітектур у контексті штучного інтелекту та хмарних обчислень. Визначено, що сучасні CPU активно інтегруються з NPU, GPU, TPU-блоками та підтримують прискорювачі на рівні інструкцій (AVX-512, AMX, VNNI). Виявлено, що здатність архітектури до масштабування, багатопотоковості та енергоощадності є визначальними у середовищах AI та хмари.

У третьому розділі здійснено огляд перспектив розвитку архітектур з урахуванням потреб IT-інфраструктури. Показано, що майбутнє лежить у напрямку модульних і відкритих архітектур (chiplet-дизайн, RISC-V), гібридних систем і когнітивних платформ, які здатні адаптуватися до задач і змін середовища в режимі реального часу.

Загалом, результати дипломної роботи підтвердили практичну доцільність створення програмно-апаратного комплексу для оцінки архітектурних особливостей комп’ютерних систем. Такий підхід дозволяє здійснювати обґрунтований вибір процесора, оптимізувати конфігурації під цільові задачі та закласти основу для подальших досліджень у сфері комп’ютерної інженерії.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Мельник А. О. Архітектура комп'ютера: підруч. для студ. вищ. навч. закл. / А. О. Мельник . — Луцьк: Волинська обласна друкарня, 2008. — 470с.
2. Тарарака В.Д. Архітектура комп’ютерних систем : навч. посіб. / В.Д. Тарарака. — Житомир : ЖДТУ, 2018. — 383 с.
3. Порівняння топових процесорів INTEL і AMD | Основні відмінності | Hotline.ua. *Hotline.ua*.URL: https://hotline.ua/ua/guides/porvnyannya-procesorv-intel-ta-amd/
4. NanoReview. CPU Benchmark Comparison https://nanoreview.net/en/cpu-compare
5. GECID.com - огляди і новини світу IT. *GECID.com - огляди і новини світу IT*. URL: https://ua.gecid.com/
6. Intel проти AMD: які процесори кращі у 2024 році // Tom’s Hardware. https://www.tomshardware.com/features/amd-vs-intel-cpus
7. ITC.ua. Нові тенденції: прогнози щодо майбутніх комп’ютерних компонентів // https://itc.ua/ua/blogs/novi-tendentsiyi-prognozy-shhodo-majbutnih-komp-yuternyh-komponentiv/
8. Overclockers.ua. Порівняння процесорів https://www.overclockers.ua/ua/cpu/info/compare/
9. Порівняння процесорів // Notebookcheck.net. — Режим доступу: https://www.notebookcheck.net/Benchmarks-Tech.123.0.html
10. Архітектура комп'ютерів. *Google Drive:* URL: https://sites.google.com/view/vovkpetro/головна?authuser=0
11. Посібник по комп’ютерних процесорах для початківців. *Acer Corner*. URL: https://blog.acer.com/ua/discussion/1943/posibnik-po-komp-yuternih-procesorah-dlya-pochatkivciv
12. Amdahl, G.M. Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities // *AFIPS Conference Proceedings*. – 1967.
13. Hennessy, J., Patterson, D. The Future of Computer Architecture // *Communications of the ACM*. – 2019.
14. Думас, Дж. Д. Computer Architecture: Fundamentals and Principles of Computer Design / Джозеф Д. Думас II. – Бока Ратон : CRC Press, 2005. – 406 с.
15. Мельник А. О. Архітектура комп’ютера. – Луцьк : Волинська обласна друкарня, 2008. – 470 с.
16. Мюллер С. Модернизация и ремонт ПК. – 19-е изд. – Київ : Вільямс, 2011. – 1072 с.
17. Харріс Д., Харріс С. Цифрова схемотехніка і архітектура комп’ютера. 2017. – 772 с.
18. Ґолдстайн Г. Комп’ютер: від Паскаля до фон Неймана / Г. Ґолдстайн. – Прінстон : Princeton University Press, 1980. – 365 с.
19. Данилов Ю. А. Джон фон Нейман / Ю. А. Данилов. 1981. – 64 с.

20. What is Computer Architecture? [Електронний ресурс] // Spiceworks. – 2023. – Режим доступу: https://www.spiceworks.com/tech/tech-general/articles/what-is-computer-architecture/
21. What is Computer Architecture? Definition, Types, Structure | EM360Tech. *EM360Tech | B2B Enterprise Tech News, Podcasts, Insights*. URL: https://em360tech.com/tech-articles/what-computer-architecture-definition-types-structure
22. BBC Bitesize. The architecture of the Central Processing Unit (CPU) - Systems architecture - OCR - GCSE Computer Science Revision - OCR - BBC Bitesize. *BBC Bitesize*. URL: https://www.bbc.co.uk/bitesize/guides/zbfny4j/revision/1
23. Jacob Sorber. What is computer architecture?, 2025. *YouTube*. URL: https://www.youtube.com/watch?v=_f1hE7LZx08
24. GeeksforGeeks. Central Processing Unit (CPU) - URL: https://www.geeksforgeeks.org/central-processing-unit-cpu/ (date of access: 12.06.2025).
25. Understanding CPU Architecture. *Tutorials on Technical and Non Technical Subjects*. URL: https://www.tutorialspoint.com/digital-electronics/cpu-architecture.htm
26. Computer Organization - Von Neumann architecture - GeeksforGeeks. *GeeksforGeeks*. URL: https://www.geeksforgeeks.org/computer-organization-von-neumann-architecture/
27. Матвієнко М.П., Розен В.П., Закладний О.М. Архітектура комп'ю-терів. Київ: Видавництво Ліра-К. 2013. 264 с.
28. Архітектура комп'ютера. Частина 1 / Кравченко Ю.В., Левченко О.О. та ін. – Київ, 2022, Новий світ-2000 – 220с.
29. Будова та основні характеристики процесора. *Chaynikam.Info | Главная страница*. URL: https://www.chaynikam.info/ukr/stat_cpu.html
30. Як вибрати процесор?. *ITbox - інтернет-магазин електроніки та комп'ютерної техніки*. URL: https://www.itbox.ua/ua/blog/Yak-vibrati-procesor/?srsltid=AfmBOor_C-yMLkywDVbIu5699x2ViHiy4Q0gFOwg4EBbD5L1kwpWPOLP

Додаток А

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

(Презентація)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ



Навчально-науковий інститут Інформаційних технологій
Кафедра Комп’ютерної інженерії

КВАЛІФІКАЦІЙНА РОБОТА НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА

на тему:«Програмно-апаратний комплекс для оцінки впливу архітектури комп’ютера на його продуктивність**»**

Виконав студент групи КІД-42
Верескун Денис Сергійович

Керівник кваліфікаційної роботи:
Борова Юлія Олександрівна,асистент кафедри

Київ – 2025

Мета роботи – оцінити вплив архітектури комп’ютера на його продуктивність за допомогою програмно-апаратного тестування.

Об`єкт дослідження – архітектура центрального процесора як ключовий фактор, що впливає на продуктивність комп’ютерної системи.

Предмет дослідження – методи та засоби оцінювання продуктивності комп’ютерної системи залежно від архітектурних особливостей процесора.

2

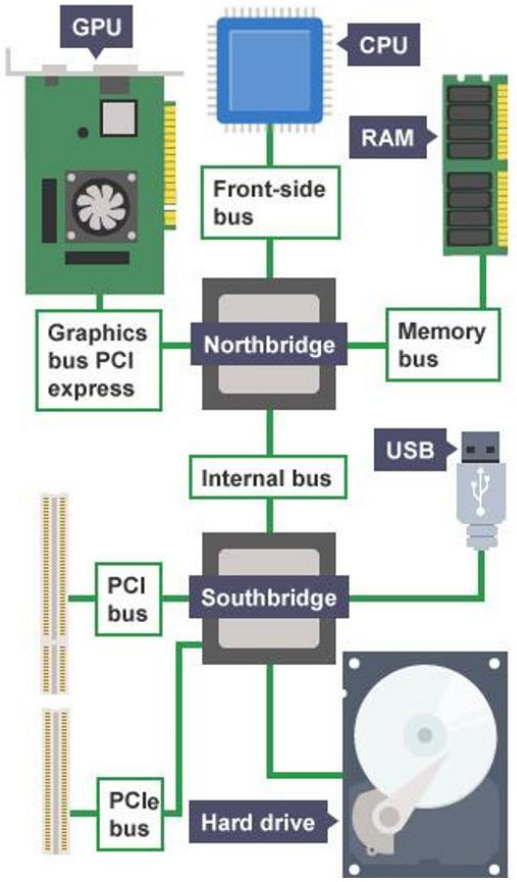
Актуальність обраної теми

Сьогодні від швидкості й ефективності комп’ютера залежить усе: як ми працюємо, навчаємось, створюємо нові технології. Вибір правильної архітектури впливає не лише на “сухі” цифри в тестах, а й на те, чи буде система реально справлятись із нашими завданнями — від офісу до складних розрахунків чи ігор. Саме тому важливо знати, як різні архітектури працюють “під капотом” і що це дає на практиці.

3

Огляд апаратної архітектури комп’ютера:

Апаратна архітектура комп’ютера визначає, як основні компоненти — процесор, пам’ять, відеокарта, накопичувачі та периферія — взаємодіють між собою через материнську плату. Всі елементи з’єднані системними шинами та чипсетом, що відповідає за обмін даними. CPU напряму працює з RAM і GPU, а накопичувачі та USB-пристрої — через контролери. Така організація забезпечує швидкодію, масштабованість і стабільність системи.



4

Класифікація архітектур процесорів:

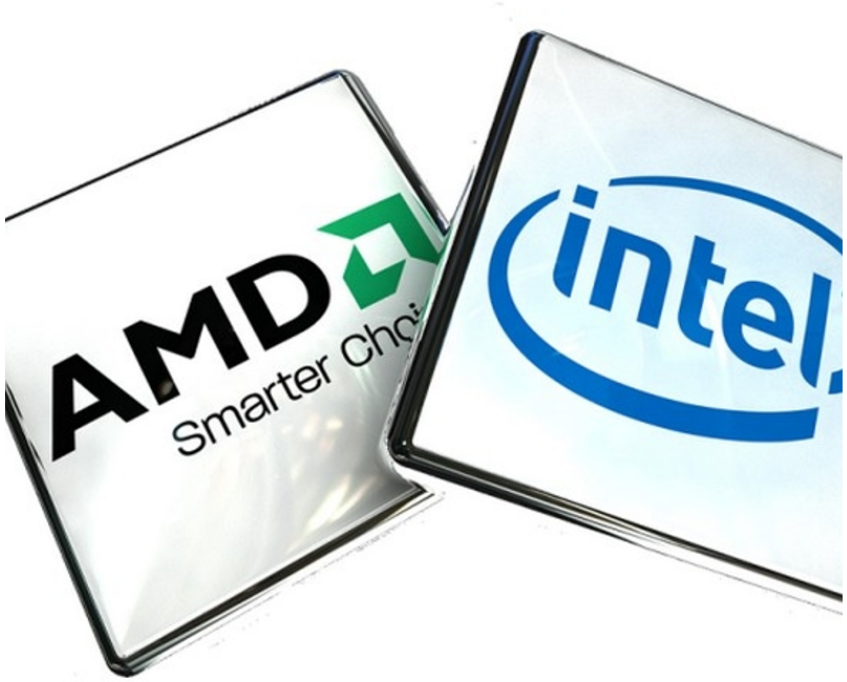
Сучасні процесори використовують три основні архітектури: RISC, CISC і ARM.

Архітектура	Переваги	Використання
RISC	Швидкість, простота	ARM, Apple M-серія
CISC	Сумісність, гнучкість	Intel, AMD (x86)
ARM	Енергоефективність	Мобільні, вбудовані, ПК

5

Особливості архітектур Intel та AMD

Intel робить ставку на гібридну архітектуру, яка поєднує продуктивні й енергоефективні ядра, що дозволяє краще розподіляти навантаження. Це забезпечує високу однопотокову продуктивність, що важливо для ігор та офісних програм. AMD, у свою чергу, використовує архітектуру Zen та чиплет-дизайн, який дозволяє легко масштабувати кількість ядер. Це забезпечує перевагу в багатоядерних задачах, таких як рендеринг, компіляція, віртуалізація, а також кращу енергоефективність при високих навантаженнях.



09

Стенд: ідея та методика дослідження

Для порівняння архітектур процесорів в ідентичних умовах було використано відкритий стенд, на якому фіксувались усі компоненти, крім CPU та материнської плати. Такий підхід дозволив виключити вплив інших змінних, зокрема відеокарти, пам’яті, накопичувачів або охолодження. Усі тести проводились при стабільному живленні, однакових версіях BIOS, драйверах і налаштуваннях. Завдяки відкритій конструкції стенду, легко контролювались температури, споживання та поведінка компонентів під навантаженням.



Приклад корпусу стенду для тестування процесора та материнської плати.

7

Методика тестування продуктивності

Для об’єктивного порівняння процесорів було обрано синтетичні бенчмарки, що широко застосовуються в галузі: **Cinebench R23**, **Geekbench 6** та **PassMark**. Кожен із них дозволяє оцінити різні аспекти продуктивності:





•**Cinebench R23** — використовується для оцінки швидкодії в рендерингу, базується на рушії Cinema 4D. Показує як одноядерну, так і багатоядерну продуктивність.

•**Geekbench 6** — кросплатформенний тест, що моделює типові повсякденні задачі: шифрування, редагування фото, штучний інтелект. Також оцінює як single-core, так і multi-core навантаження.

•**PassMark** — комплексний тест, що включає CPU, RAM, графіку та дискову підсистему. Дозволяє отримати загальний рейтинг системи та окремі оцінки для кожного модуля.

PASSMARK SOFTWARE

8

Тестування процессорів

Процесор	Однопотоковий режим	Багатопотоковий режим
Intel Core i9-13900K	2094	37008
AMD Ryzen 9 7950X	2014	37480
Intel Core i7-13700K	2016	30580
AMD Ryzen 7 7700X	2008	30090
Intel Core i5-13600K	2011	23471
AMD Ryzen 5 7600X	1963	25336

Результати тестування процесорів у Cinebench R23

Процесор	Однопотоковий режим	Багатопотоковий режим
Intel Core i9-13900K	3036	21267
AMD Ryzen 9 7950X	3068	21401
Intel Core i7-13700K	2940	18023
AMD Ryzen 7 7700X	2971	19891
Intel Core i5-13600K	2644	16567
AMD Ryzen 5 7600X	2644	14211

Результати тестування процесорів у Geekbench 6

Процесор	Однопотоковий рейтинг	Загальний бал CPU
Intel Core i9-13900K	4678	61015
AMD Ryzen 9 7950X	4620	62153
Intel Core i7-13700K	4380	45159
AMD Ryzen 7 7700X	4477	46010
Intel Core i5-13600K	4231	34090
AMD Ryzen 5 7600X	4122	35235

Результати тестування процесорів у PassMark

9

Результати тестування

У результаті тестування процесорів Intel і AMD за допомогою Cinebench R23, Geekbench 6 і PassMark було виявлено важливі закономірності.

Процесори **Intel** стабільно демонструють **вищу продуктивність в однопотокових задачах**, що критично для ігор, офісного ПЗ і програмування.

AMD мають **кращу продуктивність у багатоядерних навантаженнях**, особливо в рендерингу, компіляції, AI-розрахунках.



10

Висновки щодо архітектур Intel та AMD

Архітектура процесора є визначальним фактором його продуктивності та енергоефективності. Процесори **Intel**, починаючи з покоління Alder Lake, побудовані за **гібридним принципом**, який поєднує продуктивні ядра (P-ядра) та енергоефективні ядра (E-ядра). Це забезпечує гнучкий розподіл навантаження та високу швидкодію в однопотокових задачах — іграх, браузингу, офісних і інтерактивних сценаріях.

Натомість **AMD** використовує **архітектуру Zen та чиплет-дизайн**, що дозволяє розміщувати більше повноцінних ядер в одному процесорі. Завдяки цьому AMD має перевагу в багатоядерних задачах, таких як рендеринг, кодування відео, обчислення наукових моделей, віртуалізація та розробка.

Також варто враховувати **енергоефективність та тепловий пакет (TDP)**: AMD, особливо у моделях на базі 4 та 5-нм техпроцесу, показує нижче енергоспоживання при аналогічному навантаженні, що робить її вигідною для робочих станцій і серверів.

Таким чином, **вибір архітектури повинен базуватись на конкретних завданнях**:

•Для загального використання, ігор, навчання — оптимальним буде Intel.

•Для професійного застосування, що вимагає багатопотокової обробки — краще обрати AMD.

11

Перспективи розвитку архітектур ПК

Сучасні тенденції свідчать, що архітектура комп’ютера відіграє вирішальну роль у забезпеченні продуктивності, енергоефективності та стабільності системи. Подальший розвиток спрямований на покращення саме цих характеристик.

Одним із головних напрямів є **гібридні архітектури**, де поєднуються ядра різної потужності — це дозволяє краще адаптуватися до типу навантаження (як у Intel). У той же час **AMD вдосконалює чиплет-дизайн**, що забезпечує високу багатоядерну продуктивність і масштабованість.

Також зростає інтерес до **відкритих архітектур, як-от RISC-V**, що дають більше гнучкості при розробці систем під конкретні задачі. Паралельно розвивається інтеграція **AI-модулів** безпосередньо в процесори — це ключ до подальшого розвитку інтелектуальних обчислень.

12

ВИСНОВКИ

У результаті дослідження було створено програмно-апаратний стенд, який дозволив уніфіковано тестувати процесори з різною архітектурою в однакових умовах. Такий підхід забезпечив чистоту експерименту та дозволив отримати достовірні й репрезентативні дані.

Аналіз результатів тестування показав, що архітектура процесора напряму впливає на продуктивність у різних типах задач. Intel краще показав себе в однопотокових навантаженнях, тоді як AMD переважає в багатопотокових сценаріях та демонструє кращу енергоефективність.

Робота має практичне значення для користувачів, які обирають оптимальну конфігурацію ПК залежно від своїх задач, а також для виробників, яким важливо враховувати баланс між архітектурою, продуктивністю та енерговитратами.