

<https://doi.org/10.31891/2307-5732-2026-365-9>

УДК 004.27:004.421.2

ГРЕСЬКО СВИТЛАНА

Черкаський державний технологічний університет

<https://orcid.org/0000-0002-4049-4530>

e-mail: s.hresko@chdtu.edu.ua

ЛАВДАНСЬКА ОЛЬГА

Черкаський державний технологічний університет

<https://orcid.org/0000-0001-5571-2281>

e-mail: o.yehorova@chdtu.edu.ua

КОРОБЕЙНИК ЮРІЙ

Черкаський державний технологічний університет

<https://orcid.org/0009-0004-6096-5725>

e-mail: u.o.korobeinyk.asp24@chdtu.edu.ua

ГРЕСЬКО ЄВГЕНІЙ

Черкаський державний технологічний університет

<https://orcid.org/0009-0008-5119-0787>

e-mail: ye.i.hresko.asp23@chdtu.edu.ua

АНАЛІЗ ТА ДОСЛІДЖЕННЯ СУЧАСНИХ КОМП'ЮТЕРНИХ АРХІТЕКТУР

У статті проведено системний аналіз сучасних тенденцій розвитку комп'ютерних архітектур. Розглянуто основні напрями еволюції архітектур центральних процесорів, графічних прискорювачів, гетерогенних обчислювальних систем та спеціалізованих процесорів для штучного інтелекту. Визначено наукові проблеми, пов'язані з підвищенням енергоефективності, масштабованості та оптимізації паралельних обчислень. Представлено результати аналізу ефективності сучасних архітектурних підходів та окреслено перспективи подальших досліджень у цій сфері.

Аналіз висвітлює ключову проблему балансування високої обчислювальної продуктивності з енергоефективністю та масштабованістю, яка стає все більш актуальною через фізичні обмеження напівпровідникових технологій. Сучасні обчислювальні системи рухаються до гетерогенної інтеграції, поєднуючи центральні процесори, графічні процесори та прискорювачі на основі штучного інтелекту в рамках уніфікованих архітектур для досягнення оптимального розподілу завдань. У статті розглядаються сучасні наукові підходи до оптимізації обчислювальної продуктивності, зниження енергоспоживання та покращення взаємодії апаратного та програмного забезпечення в складних системах.

У дослідженні робиться висновок, що еволюція комп'ютерної архітектури характеризується переходом від універсальних до спеціалізованих та адаптивних обчислювальних систем. Очікується, що майбутній прогрес буде зумовлений розвитком відритих архітектур, таких як RISC-V, технології обчислень майже з пам'яттю та нейроморфні процесори, що імітують біологічні нейронні структури. Представлені результати сприяють розумінню сучасних архітектурних тенденцій та формують основу для подальших досліджень, спрямованих на розробку ефективних, масштабованих та інтелектуальних обчислювальних платформ.

Ключові слова: комп'ютерна архітектура, паралелізм, RISC, CISC, GPU, гетерогенність, штучний інтелект.

HRESKO SVITLANA, LAVDANSKA OLGA, KOROBAYNIK YURIY, HRESKO YEVHENII

Cherkassy State Technological University

The article presents a comprehensive analysis of modern computer architectures and their development trends in the context of contemporary computational challenges. The research explores the evolution of processor design principles from the classical von Neumann model to modern heterogeneous and AI-oriented systems. Particular attention is paid to the fundamental architectural paradigms, including RISC (Reduced Instruction Set Computing), CISC (Complex Instruction Set Computing), and various models of parallel computation such as SIMD (Single Instruction, Multiple Data), MIMD (Multiple Instruction, Multiple Data), and GPU-based architectures. The study also examines the emergence of specialized accelerators designed for artificial intelligence and deep learning tasks, such as Google TPU and Apple Neural Engine, which significantly improve performance and energy efficiency.

The analysis highlights the key problem of balancing high computational performance with energy efficiency and scalability, which has become increasingly relevant due to the physical limitations of semiconductor technologies. Modern computing systems are moving toward heterogeneous integration, combining CPUs, GPUs, and AI accelerators within unified architectures to achieve optimal task distribution. The paper reviews current scientific approaches to optimizing computational throughput, reducing energy consumption, and improving hardware-software interaction in complex systems.

The research concludes that the evolution of computer architecture is characterized by the transition from universal to specialized and adaptive computing systems. Future progress is expected to be driven by the development of open architectures such as RISC-V, near-memory computing technologies, and neuromorphic processors that imitate biological neural structures. The presented findings contribute to understanding current architectural trends and form the basis for further research aimed at designing efficient, scalable, and intelligent computing platforms.

Keywords: computer architecture, parallelism, RISC, CISC, GPU, heterogeneity, artificial intelligence.

Стаття надійшла до редакції / Received 21.02.2026

Прийнята до друку / Accepted 13.03.2026

Опубліковано / Published 28.05.2026



This is an Open Access article distributed under the terms of the [Creative Commons CC-BY 4.0](https://creativecommons.org/licenses/by/4.0/)

© Гресько Світлана, Лавданська Ольга, Коробейник Юрій, Гресько Євгеній

Постановка проблеми у загальному вигляді та її зв'язок із важливими науковими чи практичними завданнями

Швидкий розвиток інформаційних технологій, штучного інтелекту та інтернету речей зумовив зростання потреби у високопродуктивних і водночас енергоефективних обчислювальних системах. Традиційні архітектури центральних процесорів, засновані на класичній моделі фон Неймана, вичерпують свій потенціал через

обмеження у пропускній здатності між пам'яттю та процесором (так зване «вузьке місце фон Неймана») та неможливість ефективно масштабувати обчислення [1].

Проблема підвищення продуктивності не може більше вирішуватися лише шляхом збільшення тактової частоти або кількості транзисторів. Сучасна наука стоїть перед завданням пошуку нових архітектурних підходів, які забезпечать збалансованість між швидкодією, енергоефективністю та універсальністю. Це завдання є фундаментальним як для наукових досліджень у сфері комп'ютерних систем, так і для практичного впровадження нових технологій у промисловості, хмарних обчисленнях, телекомунікаціях і системах штучного інтелекту [2].

Аналіз досліджень і публікацій

Питання ефективності комп'ютерних архітектур досліджується в працях Дж. Хеннесі та Д. Паттерсона, які заклали теоретичні основи RISC-архітектур і сформулювали принципи кількісного підходу до оцінки продуктивності процесорів [1].

М. Флінн у своїй класифікації обчислювальних систем (SISD, SIMD, MISD, MIMD) [2] визначив фундаментальні моделі паралельної обробки даних, які стали базою для розвитку сучасних GPU-архітектур.

Подальші дослідження зосереджені на питаннях енергоефективності [3], відкритих архітектурних стандартів [4] та архітектур, орієнтованих на машинне навчання [5].

Проте, попри наявність значної кількості наукових і прикладних праць, залишається невирішеною проблема уніфікації підходів до побудови гетерогенних систем, які поєднують CPU, GPU, TPU та інші прискорювачі в єдину ефективну архітектурну модель. Також актуальними є питання динамічної оптимізації продуктивності та безпеки в багатопроцесорних середовищах [6;7].

Формулювання цілей статті

Метою даного дослідження є аналіз та узагальнення сучасних архітектурних рішень у комп'ютерних системах, а також виявлення ключових напрямів їх розвитку з позицій продуктивності, енергоефективності та адаптивності.

Для досягнення мети поставлено такі завдання:

- провести класифікацію сучасних комп'ютерних архітектур;
- проаналізувати принципи побудови RISC-, CISC-, GPU- та AI-орієнтованих архітектур;
- оцінити ефективність гетерогенних систем у контексті сучасних обчислювальних задач;
- визначити перспективи розвитку архітектур з урахуванням тенденцій штучного інтелекту та енергоефективності.

Виклад основного матеріалу

Сучасна комп'ютерна архітектура є результатом понад 70-річної еволюції обчислювальних систем. Класична архітектура фон Неймана, запропонована у 1945 році, передбачала послідовне виконання команд, спільне зберігання інструкцій і даних та наявність єдиної шини пам'яті [1]. Цей підхід став базою для більшості цифрових машин XX століття, однак із часом проявив свої обмеження: низьку пропускну здатність між пам'яттю та процесором, затримки при доступі до даних і неможливість ефективно паралельної обробки.

Подальший розвиток мікроелектроніки у 1970–1980-х роках привів до появи нових архітектурних концепцій — CISC (Complex Instruction Set Computing) і RISC (Reduced Instruction Set Computing) [6].

CISC-архітектури (x86, Intel, AMD) базуються на великому наборі складних інструкцій, що дозволяють виконувати багатокрокові операції за одну команду, зменшуючи кількість звернень до пам'яті. Проте складність апаратної реалізації призводить до збільшення енергоспоживання і затримок у конвеєрній обробці.

На противагу цьому, RISC-архітектури (ARM, MIPS, RISC-V) [3;4] орієнтуються на мінімізацію набору інструкцій і виконання простої команди за один такт. Це дало змогу збільшити швидкодію, покращити конвеєризацію і знизити споживання енергії, що зробило RISC-процесори провідними у мобільних і вбудованих системах.

В умовах обмежень класичної архітектури фон Неймана ключовим напрямом стало впровадження паралелізму [2]. Паралельна обробка дозволяє виконувати одночасно кілька інструкцій або обробляти кілька потоків даних.

За класифікацією М. Флінна [2], існують чотири основні моделі:

- SISD (Single Instruction, Single Data) — традиційна послідовна архітектура;
- SIMD (Single Instruction, Multiple Data) — одночасне виконання однієї інструкції над кількома наборами даних (векторні процесори, GPU);
- MISD (Multiple Instruction, Single Data) — рідкісна модель, що використовується у спеціалізованих системах контролю;
- MIMD (Multiple Instruction, Multiple Data) — багатоінструкційна архітектура, що лежить в основі багатопроцесорних систем і суперкомп'ютерів.

Розвиток багатоядерних процесорів (Intel Core, AMD Ryzen, Apple M-серія) [6] став практичним втіленням моделі MIMD. Кожне ядро може незалежно виконувати свій набір інструкцій, а обмін даними відбувається через спільну кеш-пам'ять або між'ядерні шини. Водночас збільшення кількості ядер вимагає ефективного балансування навантаження, оптимізації кеш-ієрархії та зменшення міжпроцесорних затримок [8].

Сучасна тенденція розвитку комп'ютерної техніки полягає у створенні гетерогенних архітектур, у яких поєднуються різні типи процесорів для виконання спеціалізованих задач. У таких системах центральний процесор (CPU) відповідає за загальне керування, тоді як графічний процесор (GPU) або нейромережевий

прискорювач (TPU, NPU) — за масові паралельні обчислення. Прикладом є архітектура HSA (Heterogeneous System Architecture) компанії AMD [6], яка забезпечує спільний доступ до пам'яті для CPU та GPU без потреби в копіюванні даних. Подібний підхід реалізовано в архітектурі Apple M1/M2, де об'єднано CPU, GPU та Neural Engine в єдиній системі-на-кристалі (SoC).

Гетерогенність дозволяє значно підвищити енергоефективність і продуктивність на ват енергії, адже кожен тип процесора виконує лише ті обчислення, для яких він оптимізований [1;6]. Водночас це породжує науково-технічну проблему — розробку універсальних компіляторів і програмних моделей, здатних автоматично розподіляти задачі між різними обчислювальними блоками (наприклад, OpenCL, CUDA, ROCm).

Зі стрімким розвитком машинного навчання зростає потреба у створенні спеціалізованих процесорів для виконання тензорних і матричних операцій [9].

Прикладом є Tensor Processing Unit (TPU) компанії Google [5], оптимізований для нейронних обчислень із використанням низької точності (bfloat16). Архітектура TPU забезпечує прискорення до 30 разів порівняно з CPU і в кілька разів перевищує ефективність GPU у задачах навчання нейронних мереж.

Схожі рішення мають компанії Apple (Neural Engine) [7], Huawei (NPU) та NVIDIA (Tensor Cores).

Ключовими особливостями таких архітектур є:

- використання матриць обчислень (systolic arrays) для паралельної обробки даних;
- зменшення точності операцій (quantization) для економії енергії;
- апаратна підтримка глибоких нейронних мереж (DNN) і обчислень на основі графів (GNN).

Науковий інтерес у цій сфері полягає у визначенні оптимальної структури апаратних прискорювачів, здатних динамічно змінювати ресурси залежно від типу нейромережевої моделі [4;5].

Таблиця 1

Основні типи архітектур та їх застосування

Архітектура	Ключові особливості і переваги	Типові сфери застосування	Приклади реалізацій
RISC (Reduced ISA)	Простий інструкційний набір, висока продуктивність на такт, енергоефективність. Переваги: простота масштабування, багато регістрів, легка конвеєризація. Недоліки: більше команд у програмі, потребує оптимального компілятора.	Вбудовані системи, мобільні пристрої, мікроконтролери, сервери (де важлива ефективність).	ARM, MIPS, SPARC, RISC-V, PowerPC.
CISC (Complex ISA)	Розгорнутий набір команд, орієнтація на мінімізацію коду і апаратну підтримку складних операцій. Переваги: компактний код, простіше програмування на асемблері, багата функціональність. Недоліки: складна апаратура, потенційно нижча швидкодія на такт.	Персональні комп'ютери, сервери, мейнфрейми; області зі значною спадщиною ПЗ.	x86-64 (Intel, AMD), IBM System/370 (згодом Z), VAX, Motorola 68k.
ARM (RISC-похідна)	RISC-архітектура з акцентом на низьке споживання. Thumb-режими для компактного коду, NEON SIMD для мультимедіа, безліч варіантів ядер. Ліцензійна модель IP.	Мобільні пристрої (смартфони, планшети), IoT, побутова електроніка, мережеве обладнання; дедалі більше – ноутбуки і хмарні сервери.	ARM Cortex-A (процесори смартфонів, Apple M1/M2), Cortex-M (мікроконтролери), Qualcomm Snapdragon, Samsung Exynos.
GPU (графічний процесор)	Масивно-паралельна SIMD/SIMT архітектура, тисячі ядер, висока пропускна здатність пам'яті. Переваги: величезна продуктивність на паралельних задачах (графіка, нейронні мережі). Недоліки: складність програмування, низька ефективність на послідовних або розгалужених алгоритмах.	3D графіка і рендеринг, GPGPU обчислення (наукові симуляції, машинне навчання), обробка відео, криптографія (в минулому).	NVIDIA GeForce/Quadro/Tesla, AMD Radeon/Instinct, Intel ARC, спеціалізовані AI GPU (Google TPU – не GPU, але аналогічна роль).
SIMD (одна інстр. – багато даних)	Паралелізм на рівні даних. Є підхід всередині CPU (векторні розширення) і окремий клас машин (векторні процесори). Дозволяє значно прискорити обробку масивів.	Векторні суперкомп'ютери (напр. Cray X-MP), мультимедійні та наукові програми на звичайних CPU (через SSE/AVX), графічні шейдери в GPU.	SSE, AVX у x86; NEON у ARM; Vector Engine у NEC SX-Aurora; старі Cray, Illiac IV тощо.

Архітектура	Ключові особливості і переваги	Типові сфери застосування	Приклади реалізацій
MIMD (багато інстр. – багато даних)	Паралелізм на рівні потоків/задач. Багато незалежних ядер виконує власні програми. Масштабування через багатоядерність і кластеризацію.	Більшість сучасних систем: багатоядерні CPU, сервери з кількома сокетами, кластери і хмарні системи, розподілені обчислення.	Будь-які multi-core процесори (Intel/AMD, ARM, RISC-V чипи з 2/4/8/... ядрами); HPC-кластери (на базі CPU+GPU вузлів); хмарні сервери.
VLIW (дуже довге слово)	Статично (компілятором) спланований паралелізм на рівні команд. Кілька ALU виконують пакет інструкцій одночасно. Плюси: проста логіка, висока теоретична продуктивність на специфічних навантаженнях. Мінуси: залежність від компілятора, «бульбашки» при розгалуженнях, можливе роздування коду.	Цифрова обробка сигналів, мультимедіа; області, де критична швидкість на типових шаблонах і доступний хороший компілятор. Менш підходить для загального призначення.	Intel Itanium (EPIC), HP PA-WideWord (прототип), Transmeta Crusoe, DSP Texas Instruments C6000, старі GPU AMD (TeraScale).
FPGA / ПЛІС	Програмована на рівні логічних вентилів архітектура. Дозволяє створити кастомний паралельний прискорювач для завдання. Вельми гнучка, може переконфігуруватися. Недоліки: низькі частоти, складність розробки.	Прототипування цифрових схем; апаратні прискорювачі в датацентрах (пошук, AI); телеком обладнання (адаптація до стандартів); спеціалізовані пристрої з малими серіями (наукові прилади, космос).	Xilinx Virtex, Xilinx Zynq (містить ARM+FPGA), Intel (Altera) Stratix, Lattice iCE40 (мали FPGA).
DSP (сигнальний процесор)	Процесор для обробки сигналів: спеціалізований набір команд (MAC, фільтрація), часто гарвардська пам'ять, апаратні цикли, фіксована кома, можливий VLIW. Оптимізований під детерміновану швидкість.	Цифрова обробка звуку, відео, радіосигналів; кодеки, модеми, управління і автоматика; будь-де, де потрібна обробка потоку даних у реальному часі з прогнозованою затримкою.	Analog Devices SHARC, TI TMS320 (C54x, C55x, C6000), Motorola 56K, Qualcomm Hexagon (DSP в мобільних SoC).
ASIC / спеціалізовані	Не універсальна архітектура, а «заточена» під задачу апаратна логіка. Максимальна продуктивність/енергоефективність на одній задачі, мінімум зайвої гнучкості.	Нейромережі (TPU, NPU для AI), мережеві пакети (SmartNIC, DPU), криптографія (SHA, майнінг ASIC), зберігання даних (Apsacer), ігрові консолі (спеціальні чипи), ін. – коли серійний обсяг виправдовує розробку ASIC.	Google TPU, Huawei Ascend, Nvidia Jetson NVDLA (відкритий дизайн для AI), Intel Movidius (відео/AI), спеціалізовані контролери (NVMe, RAID), та ін.
Neuromorphic	Архітектура, що імітує нейронні мережі мозку: багато простих обчислювачів-«нейронів», подієва асинхронна обробка. Вкрай низьке енергоспоживання на певних задачах AI.	Дослідницькі проекти, експерименти в AI (розпізнавання образів, сенсорні системи). Поки не використовується широко, але перспективно для енергоефективного AI.	IBM TrueNorth (4096 ядер-нейронів), Intel Loihi, Qualcomm Zeroth, університетські прототипи (SpiNNaker).
Квантові	Обчислення на квантових кубітах, принципово інша парадигма (квантові вентиля, суперпозиція). Непридатні для класичних програм, але можуть вирішувати певні задачі значно швидше.	RSA-факторизація, пошук в базах (алг. Гровера), моделювання квантових систем, оптимізація – поки наукові демонстрації.	D-Wave (відпалювання), IBM Quantum (надпровідні кубіти ~127 кубітів), Google Sycamore (53 кубіти), IonQ (іонні пастки).

Як видно з таблиці, вибір архітектури визначається вимогами задачі. Загального «кращого» рішення не існує – кожна архітектура має свої сильні сторони. Сучасні комп'ютерні системи часто поєднують різні архітектурні елементи: наприклад, типовий смартфон містить багатоядерний ARM CPU (RISC), GPU для графіки і AI, DSP для обробки сигналів зв'язку, і можливо NPU-акселератор для нейронних мереж – все на одному чипі (SoC).

Однією з найважливіших характеристик сучасних архітектур є співвідношення «продуктивність/споживана енергія». Для подолання цієї проблеми активно досліджуються енергоефективні технології, серед яких:

- динамічне масштабування напруги та частоти (DVFS) [3];
- використання багаторівневої кеш-пам'яті;
- обчислення поблизу пам'яті (Near-Memory Computing);
- нейроморфні архітектури, які імітують роботу біологічного мозку (IBM TrueNorth, Intel Loihi).

Такі системи демонструють у десятки разів менше енергоспоживання при збереженні прийнятної швидкодії, що відкриває перспективи для використання їх у мобільних і вбудованих пристроях майбутнього.

Висновки з даного дослідження

і перспективи подальших розвідок у даному напрямку

У результаті проведеного аналізу встановлено, що сучасні комп'ютерні архітектури перебувають на етапі переходу від традиційних універсальних до спеціалізованих, гетерогенних систем. RISC-підхід демонструє вищу енергоефективність у порівнянні з CISC при збереженні достатнього рівня універсальності [1;3;4]. GPU-архітектури є найбільш оптимальними для масових паралельних обчислень [5;6], тоді як CPU зберігають роль координатора потоків. Гетерогенні системи є основою сучасних високопродуктивних платформ [6; 7]. Використання спеціалізованих прискорювачів (TPU, NPU) дозволяє досягати приросту продуктивності у десятки разів у задачах штучного інтелекту.

Основними напрямками розвитку виступають:

- підвищення енергоефективності та продуктивності;
- інтеграція апаратних прискорювачів для задач штучного інтелекту;
- використання відкритих архітектур (RISC-V) для розширення гнучкості систем;
- розвиток концепцій «обчислень поблизу пам'яті» та «нейроморфних архітектур».

Подальші дослідження мають бути спрямовані на створення уніфікованих архітектурних моделей, здатних адаптуватися до типу задачі в реальному часі, а також на розробку нових алгоритмів управління енергоспоживанням і балансуванням навантаження в гетерогенних середовищах.

Література

1. Hennessy J. L., Patterson D. A. Computer architecture: a quantitative approach. – Morgan Kaufmann, 2023. – 936 p. – Режим доступу: <https://www.elsevier.com/books/computer-architecture-a-quantitative-approach/hennessy/978-0-12-820109-1>
2. Flynn M. J. Some computer organizations and their effectiveness // *IEEE Transactions on Computers*. – 1972. – Vol. C-21, No. 12. – P. 948–960. – DOI: <https://doi.org/10.1109/TC.1972.5009071>
3. ARM architecture reference manual / ARM Ltd. – Cambridge : ARM Developer, 2024. – 754 p. – Режим доступу: <https://developer.arm.com/documentation>
4. Patterson D. A., Waterman A. The RISC-V reader: an open architecture atlas. – Strawberry Canyon, 2021. – 128 p. – Режим доступу: <https://riscv.org/exchange/the-risc-v-reader/>
5. Jouppi N. P., Young C., Patil N. та ін. In-datacenter performance analysis of a tensor processing unit // *Proceedings of the 44th Annual International Symposium on Computer Architecture (ISCA)*. – 2017. – P. 1–12. – DOI: <https://doi.org/10.1145/3079856.3080246>
6. Heterogeneous system architecture (HSA) foundation white paper / AMD Inc. – Sunnyvale, 2022. – 42 p. – Режим доступу: <https://www.amd.com/en/technologies/hsa>
7. Apple silicon architecture overview / Apple Inc. – Cupertino, 2023. – 37 p. – Режим доступу: <https://developer.apple.com/documentation/apple-silicon>
8. Франчук Н. Основні аспекти класифікацій архітектур комп'ютерних систем // *Modern Information Technologies and Innovation Methodologies of Education in Professional Training: Methodology, Theory, Experience, Problems*. – 2019. – Вип. 54. – С. 140–144. – Режим доступу: <http://journals.uran.ua/index.php/2307-4914>
9. Хіміч О. М. та ін. Інтелектуальний паралельний комп'ютер на процесорах Intel Xeon Phi нового покоління // *Наука та інновації*. – 2018. – Т. 14, № 6. – С. 66–79. – DOI: <https://doi.org/10.15407/scin14.06.066>

References

1. Hennessy J. L., Patterson D. A. Computer architecture: a quantitative approach. – Morgan Kaufmann, 2023. – 936 p. – Rezhym dostupu: <https://www.elsevier.com/books/computer-architecture-a-quantitative-approach/hennessy/978-0-12-820109-1>
2. Flynn M. J. Some computer organizations and their effectiveness // *IEEE Transactions on Computers*. – 1972. – Vol. C-21, No. 12. – P. 948–960. – DOI: <https://doi.org/10.1109/TC.1972.5009071>
3. ARM architecture reference manual / ARM Ltd. – Cambridge : ARM Developer, 2024. – 754 p. – Rezhym dostupu: <https://developer.arm.com/documentation>
4. Patterson D. A., Waterman A. The RISC-V reader: an open architecture atlas. – Strawberry Canyon, 2021. – 128 p. – Rezhym dostupu: <https://riscv.org/exchange/the-risc-v-reader/>
5. Jouppi N. P., Young C., Patil N. та ін. In-datacenter performance analysis of a tensor processing unit // *Proceedings of the 44th Annual International Symposium on Computer Architecture (ISCA)*. – 2017. – P. 1–12. – DOI: <https://doi.org/10.1145/3079856.3080246>
6. Heterogeneous system architecture (HSA) foundation white paper / AMD Inc. – Sunnyvale, 2022. – 42 p. – Rezhym dostupu: <https://www.amd.com/en/technologies/hsa>
7. Apple silicon architecture overview / Apple Inc. – Cupertino, 2023. – 37 p. – Rezhym dostupu: <https://developer.apple.com/documentation/apple-silicon>
8. Franchuk N. Osnovni aspekty klasyfikatsii arkhitektur kompiuternykh system // *Modern Information Technologies and Innovation Methodologies of Education in Professional Training: Methodology, Theory, Experience, Problems*. – 2019. – Vyp. 54. – S. 140–144. – Rezhym dostupu: <http://journals.uran.ua/index.php/2307-4914>
9. Khimich O. M. та ін. Intel'ektualnyi paralelnyi kompiuter na protsesorakh Intel Xeon Phi novoho pokolinnia // *Nauka ta innovatsii*. – 2018. – T. 14, № 6. – S. 66–79. – DOI: <https://doi.org/10.15407/scin14.06.066>