

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ
ДЕРЖАВНИЙ ЗАКЛАД
„ЛУГАНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА”**

О.О. Смагіна, В.Ю. Козуб

«АРХІТЕКТУРА ОБЧИСЛЮВАЛЬНИХ СИСТЕМ»

**МЕТОДИЧНІ ВКАЗІВКИ ДО ЛАБОРАТОРНИХ РОБІТ
ДО ВИВЧЕННЯ ОСВІТНЬОГО КОМПОНЕНТА ДЛЯ ЗДОБУВАЧІВ ОСВІТИ СПЕЦІАЛЬНОСТІ
F3 – „Комп'ютерні науки”**

**Полтава
ДЗ „ЛНУ імені Тараса Шевченка”
2026**

Рецензенти:

Гоменюк С. І.

– доктор технічних наук, професор, декан математичного факультету Запорізького національного університету.

Могильний Г. А.

– кандидат технічних наук, доцент, директор навчально-наукового інституту математики та інформаційних технологій ДЗ „Луганський національний університет імені Тараса Шевченка”.

Методичні вказівки до лабораторних робіт з дисципліни «Архітектура обчислювальних систем» для здобувачів вищої освіти спеціальності F3 – „Комп'ютерні науки”/ Уклад. О.О. Смагіна, В.Ю. Козуб: – держ. закл. „Луган. нац. ун-т імені Тараса Шевченка”. – Полтава: ДЗ „ЛНУ імені Тараса Шевченка”, 2026. – 41 с.

Призначено для здобувачів з метою розвитку та закріплення практичних навичок з підготовки до лабораторних занять з дисципліни «Архітектура обчислювальних систем».

УДК 004.272:37

Рекомендовано до друку Вченою радою Луганського національного університету імені Тараса Шевченка (протокол №1 від 00 січня 2026 р.)

© Смагіна О.О., Козуб В.Ю., 2026
© ДЗ „ЛНУ імені Тараса Шевченка”, 2026

ПЕРЕДМОВА

Мета вивчення дисципліни: основною метою дисципліни «Архітектура обчислювальних систем» є формування у здобувачів сучасних уявлень про принципи побудови та функціонування апаратних компонентів комп'ютерних систем, оволодіння практичними навичками ефективного аналізу архітектурних рішень та оптимізації продуктивності обчислювальних систем в умовах вирішення реальних практичних завдань. Також до цілей освоєння дисципліни відносяться формування у здобувачів здібностей з реалізації оптимального підходу під час вибору апаратних компонентів, аналізу вузьких місць продуктивності та проєктування збалансованих обчислювальних архітектур для різних класів завдань.

В результаті освоєння курсу здобувач опанує: основи архітектурних принципів фон Неймана та Гарвардської архітектури; структуру та функціонування центрального процесора та його компонентів; систему команд та типи адресації пам'яті в архітектурах RISC та CISC; ієрархію пам'яті від регістрів до зовнішніх накопичувачів; принципи кеш-пам'яті та віртуальної пам'яті; механізми паралельної обробки та багатоядерних архітектур; конвеєрну обробку команд та суперскалярну архітектуру; архітектуру графічних процесорів та спеціалізованих обчислювачів; системні шини та сучасні інтерфейси введення-виведення; принципи енергоефективності мобільних архітектур; перспективні напрямки розвитку обчислювальних систем.

Вміти: аналізувати архітектурні особливості сучасних процесорів різних виробників; визначати оптимальну конфігурацію обчислювальної системи під конкретні завдання; оцінювати вплив ієрархії пам'яті на продуктивність додатків; аналізувати конфлікти в конвеєрі та методи їх усунення; порівнювати ефективність RISC та CISC архітектур для різних застосувань; визначати доцільність використання GPU для паралельних обчислень; оцінювати пропускну здатність системних шин та інтерфейсів;

аналізувати енергоефективність різних архітектурних рішень; підбирати оптимальні компоненти для збалансованої системи; інтерпретувати результати бенчмарків та тестів продуктивності.

Володіти: навичками роботи з технічною документацією процесорів та системних компонентів; методиками аналізу продуктивності обчислювальних систем; техніками профілювання додатків для виявлення вузьких місць; інструментами моніторингу апаратних ресурсів та їх завантаження; засобами тестування стабільності та продуктивності систем; практиками оптимізації використання кеш-пам'яті; методами порівняльного аналізу різних архітектурних рішень; техніками підбору компонентів з урахуванням сумісності та балансу; інструментами віртуалізації для тестування конфігурацій; навичками створення технічної документації апаратних рішень.

ЗМІСТ

ТЕМА 1. ОСНОВНІ ПРИНЦИПИ АРХІТЕКТУРИ КОМП'ЮТЕРА.....	6
ТЕМА 2 ЦЕНТРАЛЬНИЙ ПРОЦЕСОР (CPU) ТА ЙОГО КОМПОНЕНТИ.....	9
ТЕМА 3 СИСТЕМА КОМАНД ТА АДРЕСАЦІЯ.....	12
ТЕМА 4 ІЄРАРХІЯ ПАМ'ЯТІ В СУЧАСНИХ СИСТЕМАХ.....	15
ТЕМА 5 ПАРАЛЕЛІЗМ ТА БАГАТОЯДЕРНІ АРХІТЕКТУРИ.....	17
ТЕМА 6 КОНВЕЄРИЗАЦІЯ ТА ПРОДУКТИВНІСТЬ.....	19
ТЕМА 7 ГРАФІЧНІ ПРОЦЕСОРИ (GPU) ТА СПЕЦІАЛІЗОВАНІ ОБЧИСЛЮВАЧІ.....	22
ТЕМА 8 СИСТЕМНІ ШИНИ ТА ІНТЕРФЕЙСИ ВВЕДЕННЯ-ВИВЕДЕННЯ ЧАСТИНА 1.....	24
ТЕМА 8 СИСТЕМНІ ШИНИ ТА ІНТЕРФЕЙСИ ВВЕДЕННЯ-ВИВЕДЕННЯ ЧАСТИНА 2. ...	27
ТЕМА 9 ЕНЕРГОЕФЕКТИВНІСТЬ ТА МОБІЛЬНІ АРХІТЕКТУРИ ЧАСТИНА 1.. ..	30
ТЕМА 9 ЕНЕРГОЕФЕКТИВНІСТЬ ТА МОБІЛЬНІ АРХІТЕКТУРИ ЧАСТИНА	32
ТЕМА 10 МАЙБУТНЄ ОБЧИСЛЮВАЛЬНИХ СИСТЕМ ЧАСТИНА 1.....	35
ТЕМА 10 МАЙБУТНЄ ОБЧИСЛЮВАЛЬНИХ СИСТЕМ ЧАСТИНА 2.....	38

ТЕМА 1. ОСНОВНІ ПРИНЦИПИ АРХІТЕКТУРИ КОМП'ЮТЕРА

Мета роботи: набути практичних навичок аналізу фундаментальних принципів побудови комп'ютерних систем, освоїти відмінності між архітектурою фон Неймана та Гарвардською архітектурою, навчитися визначати базові компоненти обчислювальних систем та їх взаємодію, опанувати методики ідентифікації архітектурних особливостей сучасних пристроїв, розвинути розуміння еволюції комп'ютерних архітектур від класичних до сучасних гібридних рішень.

Практичне завдання

1. Необхідно провести порівняльний аналіз двох фундаментальних архітектур комп'ютерних систем - архітектури фон Неймана та Гарвардської архітектури. Для архітектури фон Неймана потрібно описати принцип збереження програм у тій самій пам'яті що й дані, проаналізувати як центральний процесор отримує інструкції та дані через одну спільну шину, визначити переваги такого підходу включаючи простоту реалізації та гнучкість використання пам'яті, виявити основний недолік що полягає у виникненні вузького місця фон Неймана коли процесор не може одночасно читати інструкцію та отримувати дані через одну шину.

2. Для Гарвардської архітектури треба описати принцип фізичного розділення пам'яті для програм та даних, проаналізувати наявність окремих шин для інструкцій та даних що дозволяє паралельний доступ, визначити переваги включаючи вищу продуктивність через відсутність конфліктів за доступ до пам'яті та можливість оптимізації кожного типу пам'яті під свої потреби, виявити недоліки такі як більша складність апаратної реалізації та менша гнучкість розподілу пам'яті. Необхідно навести сучасні приклади використання кожної архітектури, зокрема більшість настільних комп'ютерів та серверів використовують архітектуру фон Неймана, тоді як мікроконтролери Arduino та цифрові сигнальні процесори часто базуються на Гарвардській архітектурі або її модифікаціях.

3. Треба створити детальну структурну схему типового сучасного комп'ютера з ідентифікацією всіх базових компонентів включаючи центральний процесор який виконує арифметичні та логічні операції та керує роботою всієї системи, оперативну пам'ять RAM для зберігання поточних програм та даних під час роботи системи, постійну пам'ять ROM або флеш-пам'ять для зберігання мікропрограми BIOS або UEFI, системні шини для передачі даних адрес та сигналів управління між компонентами, контролери введення-виведення для взаємодії з периферійними пристроями такими як клавіатура миша монітор накопичувачі мережеві адаптери. На схемі необхідно показати напрямки потоків даних та управляючих сигналів між компонентами.

Контрольні питання

1. Опишіть детально принцип роботи архітектури фон Неймана включаючи цикл вибірки-виконання команд, поясніть що таке вузьке місце фон Неймана та як воно впливає на продуктивність системи, наведіть конкретні приклади ситуацій коли це вузьке місце стає критичним обмеженням, розгляньте які сучасні технології використовуються для пом'якшення цієї проблеми такі як кеш-пам'ять передбачення переходів та конвеєрна обробка.

2. Поясніть фундаментальні відмінності Гарвардської архітектури від архітектури фон Неймана з акцентом на організацію пам'яті та шин, опишіть переваги модифікованої Гарвардської архітектури що використовується в сучасних процесорах де на рівні кеш-пам'яті L1 застосовується розділення інструкцій та даних але нижчі рівні пам'яті є спільними, наведіть приклади конкретних процесорів або систем що використовують чисту Гарвардську архітектуру та поясніть чому саме для цих застосувань обрано такий підхід.

3. Опишіть роль та функції кожного базового компонента обчислювальної системи - процесора пам'яті системних шин контролерів введення-виведення, поясніть як ці компоненти взаємодіють між собою під час виконання типової програми наприклад веб-браузера, розгляньте

еволюцію кожного компонента від ранніх комп'ютерів до сучасних систем 2025 року, опишіть сучасні тенденції такі як інтеграція різнотипних обчислювальних блоків в одному чіпі системи на кристалі SoC та використання швидких послідовних інтерфейсів замість паралельних шин.

Література

1. Vasileios Leon, Muhammad Abdullah Hanif, Giorgos Armeniakos, Xun Jiao, Muhammad Shafique, Kiamal Pekmestzi, Dimitrios Soudris. Approximate Computing Survey, Part I: Terminology and Software & Hardware Approximation Techniques. 2023. URL: <https://arxiv.org/abs/2307.11124>
2. Y.-T. Liu, K. Wang, Y.-D. Liu, D.-S. Wang. A survey of universal quantum von Neumann architecture. 2023. URL: <https://arxiv.org/abs/2307.14219>

ТЕМА 2. ЦЕНТРАЛЬНИЙ ПРОЦЕСОР (CPU) ТА ЙОГО КОМПОНЕНТИ

Мета роботи: набути практичних навичок аналізу внутрішньої структури центрального процесора, освоїти принципи функціонування арифметико-логічного пристрою та системи регістрів, навчитися визначати архітектурні особливості сучасних процесорів різних виробників, опанувати методики порівняльного аналізу процесорних архітектур, розвинути розуміння циклу виконання команд та роботи блоку управління.

Практичне завдання

1. Необхідно провести детальний аналіз структури центрального процесора сучасного комп'ютера з ідентифікацією всіх ключових функціональних блоків. Потрібно описати арифметико-логічний пристрій ALU який виконує арифметичні операції такі як додавання віднімання множення ділення та логічні операції такі як AND OR NOT XOR над двійковими даними, визначити типові розрядності ALU в сучасних процесорах що становлять 64 біти для цілочисельних операцій та 256 або 512 для SIMD операцій з векторами. Описати блок операцій з рухомою комою FPU який спеціалізується на обчисленнях дійсних чисел за стандартом IEEE 754 та використовується для наукових розрахунків графіки та ігор.

2. Треба проаналізувати систему регістрів процесора яка включає регістри загального призначення для зберігання операндів та проміжних результатів обчислень, в архітектурі x86-64 наявні 16 регістрів загального призначення розрядністю 64 біти з іменами RAX RBX RCX RDX RSI RDI RBP RSP та R8 до R15, в архітектурі ARM64 наявні 31 регістр загального призначення X0 до X30 плюс спеціальні регістри. Потрібно описати спеціальні регістри включаючи лічильник команд PC або RIP який зберігає адресу наступної команди до виконання, регістр прапорців FLAGS який містить біти стану результату останньої операції такі як ознака нуля переповнення від'ємного результату переносу, показчик стека SP який вказує на вершину стека в пам'яті.

3. Необхідно описати блок управління який координує роботу всіх компонентів процесора та реалізує цикл виконання команд що складається з етапів вибірка інструкції з пам'яті за адресою в лічильнику команд, декодування інструкції для визначення типу операції та операндів, вибірка операндів з регістрів або пам'яті, виконання операції в ALU або іншому функціональному блоці, запис результату в регістр призначення або пам'ять, оновлення лічильника команд для переходу до наступної інструкції. Треба проілюструвати цей цикл на прикладі виконання простої команди додавання двох чисел з регістрів з записом результату в третій регістр.

Контрольні запитання

1. Опишіть детально структуру та функції арифметико-логічного пристрою ALU, поясніть які типи операцій він виконує та як результати операцій впливають на прапорці в регістрі стану, розгляньте як відбувається виконання складних операцій таких як множення та ділення які можуть вимагати декількох тактів або виконуватися в спеціалізованих блоках, наведіть приклади як ALU використовується в реальних обчисленнях програм.

2. Поясніть призначення різних типів регістрів в процесорі включаючи регістри загального призначення спеціальні регістри такі як лічильник команд показник стека регістр прапорців, опишіть чому обмежена кількість регістрів є архітектурним обмеженням та як компілятори та процесори впораються з ситуацією коли програмі потрібно більше змінних ніж є регістрів через техніку register spilling, розгляньте відмінності в кількості та використанні регістрів між архітектурами x86-64 та ARM64.

3. Опишіть повний цикл виконання команди від вибірки до запису результату з детальним поясненням що відбувається на кожному етапі, поясніть роль блоку управління в координації цього процесу, розгляньте як цей базовий цикл модифікується в сучасних процесорах через використання конвеєрної обробки та позачергового виконання, наведіть конкретні приклади виконання різних типів команд - арифметичної операції

завантаження з пам'яті умовного переходу, порівняйте архітектурні підходи Intel AMD та Apple до реалізації блоку управління.

Література

1. Shuyao Cheng, Pengwei Jin, Qi Guo, Zidong Du, Rui Zhang, Yunhao Tian, Xing Hu, Yongwei Zhao, Yifan Hao, Xiangtao Guan, Husheng Han, Zhengyue Zhao, Ximing Liu, Ling Li, Xishan Zhang, Yuejie Chu, Weilong Mao, Tianshi Chen, Yunji Chen. Pushing the Limits of Machine Design: Automated CPU Design with AI. 2023. URL: <https://arxiv.org/pdf/2306.12456>
2. Ehsan Arianyan, Niloofar Gholipour, Davood Maleki, Neda Ghorbani, Abdollah Sepahvand, Pejman Goudarzi. A Systematic Review and Classification of HPC-Related Emerging Computing Technologies. 2025. URL: <https://www.mdpi.com/2079-9292/14/12/2476>

ТЕМА 3. СИСТЕМА КОМАНД ТА АДРЕСАЦІЯ

Мета роботи: набути практичних навичок аналізу системи команд процесорів різних архітектур, освоїти відмінності між філософіями RISC та CISC, навчитися визначати типи адресації пам'яті та їх застосування, опанувати методики порівняльного аналізу наборів інструкцій x86-64 ARM та RISC-V, розвинути розуміння впливу системи команд на продуктивність та енергоефективність, освоїти причини домінування різних архітектур у різних сегментах ринку.

Практичне завдання

1. Необхідно провести комплексний порівняльний аналіз двох фундаментальних підходів до проєктування системи команд процесора - CISC Complex Instruction Set Computer та RISC Reduced Instruction Set Computer. Для CISC архітектури потрібно описати філософію створення багатого набору складних інструкцій де одна команда може виконувати складну операцію еквівалентну декільком простим командам, проаналізувати команди змінної довжини від 1 до 15 байт в архітектурі x86, розглянути різноманітні режими адресації що дозволяють гнучко звертатися до пам'яті, визначити що декодування складних інструкцій вимагає багато транзисторів та енергії. Типовим представником є архітектура x86-64 від Intel та AMD що використовується в настільних комп'ютерах та серверах.

2. Для RISC архітектури треба описати філософію використання невеликого набору простих інструкцій де кожна команда виконується за один такт процесора, проаналізувати команди фіксованої довжини 32 біти що спрощує декодування та конвеєрну обробку, розглянути обмежений набір режимів адресації та принцип load-store де арифметичні операції виконуються тільки над даними в регістрах а доступ до пам'яті здійснюється окремими командами завантаження та збереження, визначити що проста структура команд дозволяє ефективнішу конвеєризацію та менше енергоспоживання. Типовими представниками є ARM що домінує в

мобільних пристроях та RISC-V як відкрита архітектура що набирає популярності.

Контрольні запитання

1. Опишіть фундаментальні відмінності між філософіями CISC та RISC з конкретними прикладами інструкцій з обох архітектур, поясніть історичний контекст виникнення RISC як реакції на зростаючу складність CISC процесорів, розгляньте чому сучасні CISC процесори x86 внутрішньо використовують RISC-подібні мікрооперації через декодування складних інструкцій в послідовності простих, обговоріть питання чи є чіткою межа між CISC та RISC в сучасних процесорах.

2. Поясніть детально кожен тип адресації пам'яті з прикладами команд на мові асемблера, опишіть в яких ситуаціях програмування кожен тип адресації є найбільш доцільним наприклад індексна адресація для роботи з масивами непряма для роботи з вказівниками та динамічними структурами даних, розгляньте як компілятори високорівневих мов програмування транслюють конструкції мови в команди з різними типами адресації, наведіть приклад доступу до елемента структури даних з використанням базово-індексної адресації.

3. Опишіть причини домінування архітектури ARM в мобільних пристроях станом на 2025 рік включаючи енергоефективність через RISC дизайн та big.LITTLE архітектуру гнучку ліцензійну модель що дозволяє виробникам створювати власні реалізації широку екосистему інструментів та програмного забезпечення, поясніть чому x86-64 залишається домінуючою в настільних комп'ютерах та серверах незважаючи на меншу енергоефективність через величезну базу сумісного програмного забезпечення та високу абсолютну продуктивність, розгляньте перспективи RISC-V як потенційного конкурента в майбутньому.

Література

1. Ahmed Heakl, Chaimaa Abi, Rania Hossam Abdulrahman Mahmoud. From CISC to RISC: Language-Model Guided Assembly Transpilation. 2024. URL: <https://arxiv.org/pdf/2411.16341>
2. Ahmed Heakl, Sarim Hashmi, Chaimaa Abi, Celine Lee, Abdulrahman Mahmoud. Guaranteed Guess: A Language Modeling Approach for CISC-to-RISC Transpilation with Testing Guarantees. 2025. URL: <https://arxiv.org/abs/2506.14606>

ТЕМА 4. ІЄРАРХІЯ ПАМ'ЯТІ В СУЧАСНИХ СИСТЕМАХ

Мета роботи: набути практичних навичок аналізу багаторівневої ієрархії пам'яті сучасних комп'ютерних систем, освоїти принципи роботи кеш-пам'яті та механізми підвищення її ефективності, навчитися розраховувати параметри продуктивності різних рівнів пам'яті, опанувати відмінності між технологіями оперативної пам'яті DDR4 та DDR5, розвинути розуміння організації віртуальної пам'яті та сторінкового механізму, освоїти порівняльний аналіз сучасних накопичувачів SSD та традиційних HDD.

Практичне завдання

1. Необхідно побудувати повну ієрархію пам'яті типового сучасного комп'ютера з вказанням для кожного рівня типового обсягу швидкості доступу та латентності. Ієрархія включає регістри процесора обсягом близько 1 кілобайта з часом доступу менше 1 наносекунди що є найшвидшою пам'яттю безпосередньо в ядрі процесора, кеш-пам'ять першого рівня L1 розділену на кеш інструкцій L1i та кеш даних L1d обсягом 32 або 64 кілобайти на ядро з латентністю 4-5 тактів або близько 1 наносекунди, кеш другого рівня L2 обсягом 256 кілобайт до 2 мб на ядро з латентністю 12-20 тактів або 3-6 наносекунд, кеш третього рівня L3 спільний між усіма ядрами обсягом 16-128 мб з латентністю 40-70 тактів або 10-20 наносекунд.
2. Далі в ієрархії йде оперативна пам'ять RAM типу DDR4 або DDR5 обсягом 8-128 гігабайт з латентністю 60-100 наносекунд та пропускною здатністю 25-100 гб на секунду залежно від стандарту та конфігурації, твердотільні накопичувачі SSD на базі флеш-пам'яті NAND обсягом 256 гб до 4 терабайт з латентністю читання 20-100 мікросекунд та послідовною швидкістю до 7 гігабайт на секунду для NVMe PCIe Gen4, жорсткі диски HDD обсягом 1-20 терабайт з латентністю 5-10 мілісекунд через механічне позиціонування головок та швидкістю до 200 мегабайт на секунду. Необхідно створити діаграму що візуалізує співвідношення обсягу та швидкості кожного рівня де видно що швидкість зменшується а обсяг зростає при русі вниз по ієрархії.

3. Потрібно детально розглянути принципи роботи кеш-пам'яті включаючи концепцію locality of reference - програми схильні звертатися до даних які використовувалися нещодавно temporal locality та до даних розташованих поблизу в пам'яті spatial locality.

Контрольні питання

1. Опишіть повну ієрархію пам'яті сучасного комп'ютера від регістрів до жорсткого диска з поясненням співвідношення обсягу швидкості та вартості кожного рівня, поясніть чому неможливо зробити весь комп'ютер з найшвидшої пам'яті через фізичні обмеження та економічну недоцільність, розгляньте принцип locality of reference та як він дозволяє ієрархії пам'яті бути ефективною, наведіть конкретні приклади як програми демонструють temporal та spatial locality.
2. Поясніть детально як працює кеш-пам'ять включаючи поняття cache hit коли дані знайдені в кеші та cache miss коли треба звертатися до наступного рівня, опишіть що таке лінія кешу та чому вона має розмір шістдесят чотири байти, розгляньте політики заміщення ліній кешу такі як LRU Least Recently Used коли кеш заповнений, поясніть проблему cache coherency в багатоядерних системах де різні ядра можуть мати копії одних даних у своїх кешах та як протокол MESI вирішує цю проблему.

Література

1. Chongnan Ye, Meng Chen, Qisheng Jiang, Chundong Wang. Enabling Atomic Durability for Persistent Memory with Transiently Persistent CPU Cache. 2022. URL: <https://arxiv.org/pdf/2210.17377>
2. Ezinam Bertrand Talaki, Olivier Savry, Mathieu Bouvier Des Noes, David Hely. A Memory Hierarchy Protected against Side-Channel Attacks. 2022. URL: <https://www.mdpi.com/2410-387X/6/2/19>

ТЕМА 5. ПАРАЛЕЛІЗМ ТА БАГАТОЯДЕРНІ АРХІТЕКТУРИ

Мета роботи: набути практичних навичок аналізу різних форм паралелізму в сучасних процесорах, освоїти відмінності між багатоядерними та багатопроеесорними системами, навчитися визначати ефективність технологій Hyper-Threading та Simultaneous Multithreading, опанувати концепцію паралелізму на рівні інструкцій ILP, розвинути розуміння застосування багатоядерності в реальних додатках станом на 2025 рік, освоїти методики визначення оптимальної конфігурації ядер для різних навантажень.

Практичне завдання

1. Необхідно провести порівняльний аналіз основних підходів до організації паралельних обчислень в сучасних процесорах. Потрібно розглянути багатопроеесорну систему SMP Symmetric Multiprocessing де декілька окремих фізичних процесорів встановлені на материнській платі та мають спільний доступ до оперативної пам'яті через системну шину або між'єднання такі як Intel UPI Ultra Path Interconnect або AMD Infinity Fabric, така конфігурація типова для серверів де можуть бути 2, 4 або 8 процесорних сокетів що дозволяє масштабувати систему до 100 ядер та терабайтів пам'яті але має складнощі з латентністю доступу до віддаленої пам'яті NUMA Non-Uniform Memory Access.

2. Розглянути багатоядерну архітектуру де декілька процесорних ядер інтегровані в один кристал процесора, що є стандартом для всіх сучасних процесорів від смартфонів до серверів. Для настільних процесорів станом на 2025 рік типова конфігурація включає від 4 до 16 ядер для Intel Core та AMD Ryzen, до 24 ядер для високопродуктивних десктопних процесорів Intel Core i9 та AMD Ryzen Threadripper, серверні процесори можуть мати до 96 ядер в одному процесорі AMD EPYC або 64 ядра в Intel Xeon. Необхідно проаналізувати гібридну архітектуру Intel що поєднує Performance cores P-cores оптимізовані для високої однопотокової продуктивності та Efficient

cores E-cores оптимізовані для енергоефективності та багатопотокових навантажень, наприклад Intel Core i9-14900K має 8 P-cores та 16 E-cores для загально 24 ядра.

Контрольні питання

1. Опишіть фундаментальні відмінності між багатопроцесорними та багатоядерними системами з точки зору апаратної реалізації спільного використання ресурсів та латентності комунікації, поясніть поняття NUMA в багатопроцесорних системах та як воно впливає на продуктивність, розгляньте переваги та недоліки інтеграції декількох ядер на одному кристалі порівняно з окремими процесорами, опишіть сучасні тренди в кількості ядер для різних сегментів від мобільних пристроїв до серверів станом на 2025 рік.

2. Поясніть детально як працює технологія Hyper-Threading або SMT включаючи які ресурси дублюються а які використовуються спільно двома потоками, опишіть в яких типах навантажень SMT дає максимальний приріст продуктивності а в яких може бути неефективним або навіть шкідливим, розгляньте проблеми безпеки пов'язані з SMT такі як атаки через side-channel коли один потік може витягувати інформацію про дії іншого потоку через спільне використання кешу, наведіть приклади реальних додатків де SMT суттєво покращує продуктивність.

3. Опишіть концепцію паралелізму на рівні інструкцій ILP та як суперскалярні процесори з позачерговим виконанням намагаються його використати.

Література

1. Temitayo Adefemi. What Every Computer Scientist Needs to Know About Parallelization. 2025. URL: <https://arxiv.org/pdf/2504.03647>
2. Elie Kfoury, Samia Choueiri, Ali Mazloun, Ali AlSabeh, Jose Gomez, Jorge Crichigno. A Comprehensive Survey on SmartNICs: Architectures, Development Models, Applications, and Research Directions. 2024. URL: <https://arxiv.org/abs/2405.09499>

ТЕМА 6. КОНВЕЄРИЗАЦІЯ ТА ПРОДУКТИВНІСТЬ

Мета роботи: набути практичних навичок аналізу конвеєрної обробки команд у сучасних процесорах, освоїти принципи виявлення та вирішення конфліктів у конвеєрі, навчитися розраховувати теоретичну та реальну продуктивність конвеєрних архітектур, опанувати концепцію суперскалярної архітектури та її відмінності від простої конвеєризації, розвинути розуміння впливу конвеєра на продуктивність реальних додатків, освоїти методики оптимізації коду для ефективного використання конвеєрної обробки.

Практичне завдання

1. Необхідно проаналізувати базову п'ятиступеневу конвеєрну архітектуру процесора яка складається з етапів IF Instruction Fetch вибірка інструкції з пам'яті або кешу інструкцій, ID Instruction Decode декодування інструкції та читання операндів з регістрів, EX Execute виконання операції в арифметико-логічному пристрої або обчислення адреси для доступу до пам'яті, MEM Memory Access звернення до пам'яті для операцій завантаження або збереження даних, WB Write Back запис результату в регістр призначення.

2. Потрібно побудувати часову діаграму виконання послідовності з п'яти незалежних інструкцій показуючи як в ідеальному випадку без конфліктів кожен такт процесора завершується виконання однієї інструкції після заповнення конвеєра що дає теоретичне п'ятикратне прискорення порівняно з послідовним виконанням.

3. Розглянути три основні типи конфліктів у конвеєрі які знижують його ефективність. Структурні конфлікти structural hazards виникають коли два етапи конвеєра потребують одного апаратного ресурсу одночасно наприклад якщо кеш інструкцій та кеш даних не розділені то етап IF та MEM будуть конфліктувати за доступ до пам'яті, вирішення полягає в дублюванні ресурсів або призупиненні конвеєра stall на один такт. Конфлікти за даними data hazards виникають коли інструкція потребує результату попередньої

інструкції яка ще не завершила запис результату, типи включають RAW Read After Write коли інструкція читає регістр в який попередня ще не записала результат, вирішення через forwarding або bypassing коли результат передається безпосередньо з виходу виконавчого блоку на вхід наступної інструкції без очікування запису в регістр.

Контрольні питання

1. Опишіть детально принцип роботи конвеєрної обробки команд з поясненням кожного етапу п'ятиступеневого конвеєра IF ID EX MEM WB, побудуйте часову діаграму показуючи як виконуються п'ять послідовних інструкцій в конвеєрному процесорі порівняно з неконвеєрним, розрахуйте теоретичне прискорення та поясніть чому практичне прискорення менше через наявність конфліктів та необхідність призупинень конвеєра, опишіть як глибина конвеєра впливає на тактову частоту та штрафи за неправильні передбачення.

2. Поясніть три типи конфліктів у конвеєрі - структурні конфлікти за даними конфлікти за управлінням - з конкретними прикладами коли кожен виникає, опишіть методи вирішення кожного типу конфліктів включаючи дублювання ресурсів для структурних forwarding для конфліктів за даними передбачення переходів для конфліктів за управлінням, розгляньте що таке pipeline stall та bubble в конвеєрі та як вони знижують ефективність, наведіть приклад послідовності інструкцій де виникає RAW конфлікт та покажіть як forwarding його вирішує.

3. Опишіть механізми передбачення переходів в сучасних процесорах від простого статичного передбачення до складних динамічних предикторів, поясніть як працює Branch History Table.

Література

1. Rinu Joseph. A Survey of Deep Learning Techniques for Dynamic Branch Prediction. 2021. URL: <https://arxiv.org/abs/2112.14911>
2. Nathan Gober, Gino Chacon, Lei Wang, Paul V. Gratz, Daniel A. Jiménez, Elvira Teran, Seth Pugsley, Jinchun Kim. The Championship Simulator:

Architectural Simulation for Education and Competition. 2022. URL:
<https://arxiv.org/pdf/2210.14324>

ТЕМА 7. ГРАФІЧНІ ПРОЦЕСОРИ (GPU) ТА СПЕЦІАЛІЗОВАНІ ОБЧИСЛЮВАЧІ

Мета роботи: набути практичних навичок аналізу архітектурних відмінностей між центральними та графічними процесорами, освоїти принципи масивно-паралельної обробки в GPU, навчитися визначати області застосування GPU обчислень та спеціалізованих процесорів, опанувати структуру CUDA cores та Stream processors, розвинути розуміння ролі нейронних процесорів NPU в сучасних системах станом на 2025 рік, освоїти методики вибору між CPU та GPU для різних типів обчислювальних задач.

Практичне завдання

1. Необхідно провести детальний порівняльний аналіз архітектурних принципів центрального процесора CPU та графічного процесора GPU. Для CPU описати що він оптимізований для виконання послідовних задач з низькою латентністю де кожна інструкція виконується якомога швидше, має невелику кількість потужних ядер від 4 до 32 в настільних системах кожне з складною логікою передбачення переходів позачергового виконання великими багаторівневими кешами для мінімізації латентності доступу до пам'яті, підтримує складну систему команд з можливістю обробки різноманітних типів задач. CPU використовує значну частину площі кристала для контрольної логіки та кеш-пам'яті а не для обчислювальних блоків.

2. Для GPU описати що він оптимізований для виконання масивно-паралельних задач з високою пропускнуою здатністю де важлива кількість операцій в секунду а не латентність окремої операції, має тисячі простих обчислювальних ядер наприклад NVIDIA GeForce RTX 4090 має 16384 CUDA cores AMD Radeon RX 7900, XTX має 12988 Stream Processors кожне ядро значно простіше за CPU ядро без складної логіки передбачення та невеликими кешами. GPU використовує архітектуру SIMT Single Instruction Multiple Threads де велика група потоків виконує одну інструкцію над різними даними одночасно що ідеально підходить для графічних операцій де

той самий шейдерний код застосовується до мільйонів пікселів або вершин паралельно.

Контрольні питання

1. Опишіть фундаментальні архітектурні відмінності між CPU та GPU з поясненням чому CPU має мало складних ядер а GPU тисячі простих ядер, поясніть поняття латентність-орієнтована архітектура CPU проти пропускна здатність-орієнтована архітектура GPU, розгляньте як розподіляється площа кристала між обчислювальними блоками контрольною логікою та кеш-пам'яттю в CPU та GPU, наведіть приклади типів задач де CPU буде ефективнішим та де GPU має перевагу.

2. Поясніть що таке CUDA cores в термінології NVIDIA та Stream Processors в термінології AMD та як вони відрізняються від ядер CPU, опишіть архітектуру SIMT та як GPU виконує тисячі потоків паралельно організованих в блоки та сітки, розгляньте роль спеціалізованих блоків в сучасних GPU таких як Tensor Cores для AI обчислень RT Cores для ray tracing, поясніть ієрархію пам'яті GPU від регістрів та shared memory до глобальної відеопам'яті GDDR6 або HBM та як програмісту треба оптимізувати доступ до пам'яті для максимальної продуктивності.

3. Опишіть що таке нейронні процесори NPU та для яких задач вони призначені, поясніть чому NPU є більш енергоефективними для AI задач порівняно з CPU або GPU через спеціалізовану архітектуру та низькорозрядну арифметику.

Література

1. Christoforos Kachris. A Survey on Hardware Accelerators for Large Language Models. 2024. URL: <https://arxiv.org/abs/2401.09890>

2. Robert Chab, Fei Li, Sanjeev Setia. Algorithmic Techniques for GPU Scheduling: A Comprehensive Survey. 2025. URL: <https://www.mdpi.com/1999-4893/18/7/385>

ТЕМА 8. СИСТЕМНІ ШИНИ ТА ІНТЕРФЕЙСИ ВВЕДЕННЯ-ВИВЕДЕННЯ ЧАСТИНА 1

Мета роботи: набути практичних навичок аналізу архітектури системних шин та їх впливу на продуктивність обчислювальної системи, освоїти принципи функціонування сучасних інтерфейсів введення-виведення, навчитися розраховувати пропускну здатність та затримки шин, опанувати методики порівняльного аналізу різних стандартів підключення периферійних пристроїв, розвинути розуміння еволюції інтерфейсів PCIe USB та NVMe, освоїти інструменти діагностики та моніторингу роботи системних шин.

Практичне завдання

1. Необхідно провести комплексний аналіз архітектури системних шин сучасного персонального комп'ютера на базі процесора Intel Core чотирнадцятого покоління або AMD Ryzen серії вісім тисяч. Завдання включає дослідження структури системної шини що з'єднує процесор з чіпсетом, визначення кількості ліній PCIe що виділяються безпосередньо від процесора для відеокарти та накопичувачів NVMe, аналіз розподілу ліній PCIe від чіпсета для інших пристроїв включаючи додаткові слоти розширення мережеві адаптери звукові карти контролери USB та SATA, дослідження шини DMI Direct Media Interface в системах Intel або Infinity Fabric в системах AMD що з'єднує процесор та чіпсет.

2. Необхідно створити структурну схему що відображає всі основні компоненти системи процесор чіпсет оперативну пам'ять відеокарту накопичувачі периферійні пристрої та шини що їх з'єднують з вказанням кількості ліній та пропускну здатності кожної шини.

3. Для аналізу продуктивності інтерфейсу PCIe необхідно розрахувати теоретичну пропускну здатність для різних конфігурацій враховуючи що PCIe третього покоління забезпечує швидкість один гігабайт на секунду на одну лінію в одному напрямку або два гігабайти на секунду в обох напрямках разом, PCIe четвертого покоління забезпечує два гігабайти на

секунду на лінію в одному напрямку або чотири гігабайти на секунду двонаправлено, PCIe п'ятого покоління забезпечує чотири гігабайти на секунду на лінію в одному напрямку або вісім гігабайт на секунду двонаправлено. Потрібно розрахувати максимальну швидкість передачі даних для відеокарти підключеної через PCIe четвертого покоління шістнадцять ліній що дає тридцять два гігабайти на секунду в одному напрямку, для накопичувача NVMe SSD підключеного через PCIe четвертого покоління чотири лінії що дає вісім гігабайт на секунду двонаправлено, для накопичувача NVMe підключеного через PCIe п'ятого покоління чотири лінії що дає шістнадцять гігабайт на секунду двонаправлено, для мережевого адаптера 10 гігабіт Ethernet підключеного через PCIe третього покоління чотири лінії.

Контрольні запитання

1. Опишіть архітектуру системних шин сучасного комп'ютера включаючи як процесор з'єднаний з оперативною пам'яттю чіпсетом відеокартою та накопичувачами, поясніть що таке лінії PCIe та як вони розподіляються між пристроями з обмеженого пулу ліній доступних від процесора та чіпсета, розгляньте різницю між підключенням пристрою безпосередньо до процесора з низькою латентністю та через чіпсет з вищою латентністю через проміжну шину DMI або Infinity Fabric, наведіть типову конфігурацію розподілу PCIe ліній в системі з процесором Intel Core чотирнадцятого покоління або AMD Ryzen вісім тисяч.

2. Поясніть принцип роботи шини PCIe включаючи що таке лінія lane як послідовний інтерфейс point-to-point топологію пакетну передачу даних, розрахуйте пропускну здатність для різних конфігурацій PCIe Gen три Gen чотири Gen п'ять з різною кількістю ліній один чотири вісім шістнадцять, опишіть зворотну сумісність PCIe де пристрій нового покоління може працювати в слоті старого покоління але на нижчій швидкості, розгляньте як PCIe використовується не тільки для слотів розширення але й для

підключення чіпсета накопичувачів мережесих адаптерів та інших вбудованих пристроїв.

3. Опишіть еволюцію стандарту USB від версії два точка нуль до USB чотири з зазначенням швидкості кожного покоління та типових застосувань, поясніть плутанину в назвах USB три точка нуль USB три точка один Gen один Gen два USB три точка два та як їх відрізнити, розгляньте роз'єм USB Type-C та його переваги включаючи реверсивність підключення підтримку альтернативних режимів таких як DisplayPort та Thunderbolt доставку живлення до 240 ват в новому стандарті USB PD три точка один, порівняйте NVMe та SATA інтерфейси для накопичувачів з точки зору швидкості латентності черги команд та поясніть чому NVMe є значно швидшим через використання PCIe та оптимізований протокол для флеш-пам'яті.

Література

1. Yipeng Pan, Zhichuan Guo, Mengting Zhang. Design of a Fast and Scalable FPGA-Based Bitmap for RDMA Networks. 2024. URL: https://www.researchgate.net/publication/387003885_Design_of_a_Fast_and_Scalable_FPGA-Based_Bitmap_for_RDMA_Networks

2. Daniel Bittman, Peter Alvaro, Pankaj Mehra, Darrell D. E. Long, Ethan L. Miller, Benjamin Reed. Twizzler: a Data-Centric OS for Non-Volatile Memory. 2020. URL: https://www.researchgate.net/publication/367479852_Twizzler_a_Data-Centric_OS_for_Non-Volatile_Memory

ТЕМА 8. СИСТЕМНІ ШИНИ ТА ІНТЕРФЕЙСИ ВВЕДЕННЯ-ВИВЕДЕННЯ ЧАСТИНА 2

Мета роботи: набути практичних навичок практичного тестування та діагностики продуктивності системних шин та інтерфейсів, освоїти методики виявлення вузьких місць в підсистемі введення-виведення, навчитися інтерпретувати результати бенчмарків накопичувачів та мережевих інтерфейсів, опанувати інструменти моніторингу використання пропускної здатності шин, розвинути розуміння впливу конфігурації PCIe ліній на продуктивність системи, освоїти оптимізацію розподілу пристроїв по слотах розширення.

Практичне завдання

1. Необхідно провести практичне тестування продуктивності накопичувачів різних типів з використанням спеціалізованих бенчмарків. Для жорсткого диска HDD виміряти швидкість послідовного читання та запису великих файлів яка зазвичай становить 100-200 мегабайт на секунду, швидкість випадкового доступу random читання та запису невеликих блоків 4 кілобайти яка буде значно нижчою через механічне позиціонування головок зазвичай менше 1 мегабайта на секунду з сотнями операцій введення-виведення на секунду IOPS, виміряти латентність доступу яка становить 5-10 мілісекунд через час обертання диска та позиціонування.

2. Для SATA SSD виміряти послідовну швидкість близько 500-550 мегабайт на секунду що близько до теоретичного максимуму SATA три, випадкову швидкість яка буде значно вищою за HDD досягаючи десятків тисяч IOPS завдяки відсутності механічних частин.

3. Для NVMe SSD через PCIe четвертого покоління чотири лінії виміряти послідовну швидкість читання яка може досягати 5-7 гігабайт на секунду залежно від моделі, послідовний запис 3-5 гігабайт на секунду який часто нижчий за читання через особливості флеш-пам'яті, випадкову продуктивність яка може досягати 500 тис. до 1 млн. IOPS для читання дрібних блоків демонструючи величезну перевагу над SATA, латентність яка

становить десятки мікросекунд що в сотні разів краще за HDD. Створити порівняльну таблицю результатів для HDD SATA SSD NVMe PCIe Gen три NVMe PCIe Gen чотири з висновками про відносну продуктивність та доцільність використання кожного типу для різних завдань таких як операційна система та програми сховище великих медіафайлів архів документів.

Контрольні запитання

1. Опишіть методики тестування продуктивності накопичувачів включаючи різницю між послідовним та випадковим доступом чому випадкова продуктивність важлива для операційної системи та баз даних, поясніть що таке IOPS та чому SSD мають на порядки вищі IOPS порівняно з HDD, розгляньте параметри які впливають на продуктивність SSD такі як тип флеш-пам'яті SLC MLC TLC QLC наявність DRAM кешу алгоритми контролера ефект заповнення диска на швидкість запису через garbage collection, наведіть приклади результатів бенчмарків для типових представників HDD SATA SSD NVMe Gen три NVMe Gen чотири.

2. Поясніть як кількість ліній PCIe впливає на продуктивність різних типів пристроїв відеокарт накопичувачів мережевих адаптерів, опишіть ситуації коли зменшення кількості ліній дає відчутний вплив на продуктивність та коли вплив мінімальний, розгляньте типові конфігурації материнських плат де слоти можуть працювати в режимах x16 x8 x4 залежно від того які слоти зайняті та як процесор і чіпсет розподіляють доступні лінії, наведіть рекомендації по оптимальному розміщенню пристроїв в слотах PCIe для максимальної продуктивності системи.

3. Опишіть фактори які впливають на реальну продуктивність мережевих інтерфейсів включаючи накладні витрати протоколів пропускну здатність проміжних мереж.

Література

1. Yipeng Pan, Zhichuan Guo, Mengting Zhang. Design of a Fast and Scalable FPGA-Based Bitmap for RDMA Networks. 2024. URL:

https://www.researchgate.net/publication/387003885_Design_of_a_Fast_and_Scalable_FPGA-Based_Bitmap_for_RDMA_Networks

2. Daniel Bittman, Peter Alvaro, Pankaj Mehra, Darrell D. E. Long, Ethan L. Miller, Benjamin Reed. Twizzler: a Data-Centric OS for Non-Volatile Memory. 2020. URL:

https://www.researchgate.net/publication/367479852_Twizzler_a_Data-Centric_OS_for_Non-Volatile_Memory

ТЕМА 9. ЕНЕРГОЕФЕКТИВНІСТЬ ТА МОБІЛЬНІ АРХІТЕКТУРИ

ЧАСТИНА 1

Мета роботи: набути практичних навичок аналізу енергоефективності різних процесорних архітектур, освоїти принципи динамічного управління частотою та напругою, навчитися визначати особливості мобільних архітектур оптимізованих для ноутбуків та смартфонів, опанувати концепцію big LITTLE архітектури ARM, розвинути розуміння балансу між продуктивністю та енергоспоживанням, освоїти методики вимірювання та оптимізації енергоефективності обчислювальних систем.

Практичне завдання

1. Необхідно провести комплексний аналіз енергоефективності різних процесорних архітектур через порівняння показника продуктивність на ват. Розглянути настільні процесори високої продуктивності такі як Intel Core i9-14900K з TDP Thermal Design Power 125 ват базове та 253 вати максимальне споживання або AMD Ryzen 9 7970 X з TDP 170 ват які забезпечують найвищу абсолютну продуктивність але споживають багато енергії. Порівняти з енергоефективними настільними процесорами Intel Core i5 або AMD Ryzen 5 серії з TDP 65 ват які забезпечують добрий баланс продуктивності та споживання для типових задач офісної роботи веб-браузингу мультимедіа.

2. Розглянути ноутбучні процесори Intel Core серії U з TDP 15-28 ват або серії P з TDP 28-45 ват які оптимізовані для автономної роботи від батареї.

3. Проаналізувати мобільні процесори для смартфонів та планшетів на архітектурі ARM такі як Qualcomm Snapdragon 8 Gen 3 Apple A 17 Pro MediaTek Dimensity 9300 які мають типове споживання від 2 до 5 ват під навантаженням та мінімальне споживання в режимі очікування завдяки агресивному управлінню живленням. Розрахувати показник продуктивність на ват для кожної категорії процесорів використовуючи результати багатопотокових бенчмарків таких як Cinebench Multi-Core та поділивши на

TDP або виміряне споживання, результати покажуть що мобільні ARM процесори мають в три-п'ять разів кращу енергоефективність порівняно з настільними x86 процесорами хоча абсолютна продуктивність настільних залишається вищою.

Контрольні запитання

1. Опишіть поняття енергоефективності обчислювальних систем та показник продуктивність на ват як метрику для порівняння різних архітектур, поясніть чому енергоефективність стала критично важливою в сучасних системах особливо для мобільних пристроїв працюючих від батареї та для дата-центрів де витрати на електроенергію та охолодження є значними, розгляньте trade-off між абсолютною продуктивністю та енергоефективністю.

2. Де найшвидші процесори зазвичай не є найефективнішими, наведіть конкретні приклади порівняння споживання та продуктивності настільних ноутбучних та мобільних процесорів станом на 2025 рік.

3. Поясніть принципи роботи технологій динамічного управління частотою таких як Intel Turbo Boost та AMD Precision Boost, опишіть як процесор визначає чи може підвищити частоту на основі факторів включаючи поточне споживання енергії температуру кристала кількість активних ядер тип навантаження, розгляньте як напруга живлення масштабується разом з частотою через залежність динамічного споживання від квадрата напруги що робить зниження напруги дуже ефективним для економії енергії, поясніть чому процесори не працюють на максимальній частоті постійно та як агресивне зниження частоти в режимі очікування дозволяє економити енергію.

Література

1. Stefan Huber, Tobias Lorey, Michael Felderer. Techniques for Improving the Energy Efficiency of Mobile Apps: A Taxonomy and Systematic Literature Review. 2023. URL: <https://arxiv.org/pdf/2308.08292>

2. Dian Shi, Liang Li, Rui Chen, Pavana Prakash, Miao Pan, Yuguang Fang.
Towards Energy Efficient Federated Learning over 5G+ Mobile Devices. 2021.
URL: <https://arxiv.org/abs/2101.04866>

ТЕМА 9. ЕНЕРГОЕФЕКТИВНІСТЬ ТА МОБІЛЬНІ АРХІТЕКТУРИ

ЧАСТИНА 2

Мета роботи: набути практичних навичок аналізу спеціалізованих архітектур для мобільних пристроїв, освоїти принципи big LITTLE архітектури ARM та гібридних архітектур Intel, навчитися визначати оптимальний баланс продуктивності та часу автономної роботи, опанувати методики планування задач на гетерогенних процесорах, розвинути розуміння впливу архітектурних рішень на тривалість роботи від батареї, освоїти оптимізацію енергоспоживання на рівні системи та програмного забезпечення.

Практичне завдання

1. Необхідно провести детальний аналіз big LITTLE архітектури ARM яка є ключовою технологією для енергоефективності мобільних процесорів. Архітектура поєднує в одному чіпі два типи ядер - високопродуктивні big cores або Performance cores базовані на мікроархітектурах таких як ARM Cortex-X чотири або власних розробках виробників які оптимізовані для максимальної однопотокової продуктивності та використовуються для вимогливих задач таких як ігри відеообробка складні обчислення, та енергоефективні LITTLE cores або Efficiency cores базовані на мікроархітектурах таких як ARM Cortex-A520 які оптимізовані для мінімального енергоспоживання та використовуються для фонових задач веб-браузингу відтворення музики синхронізації даних. Типова конфігурація сучасного мобільного процесора може включати 1-2 найпотужніших X-cores 2-4 середніх A-cores 4 LITTLE cores загалом 8 ядер різної продуктивності.

2. Розглянути конкретний приклад Qualcomm Snapdragon вісім Gen три який має архітектуру 1+5+2 з одним Cortex-X 4 ядром на частоті 3 з половиною гГц для найвищої пікової продуктивності, 5 Cortex-A720 ядрами на частоті три з чвертю гігагерца для балансу продуктивності та ефективності, 2 Cortex-A520 ядрами на частоті два з чвертю гігагерца для фонових задач з мінімальним споживанням. Проаналізувати як операційна

система Android через планувальник задач HMP Heterogeneous Multi-Processing або Energy Aware Scheduling приймає рішення про розміщення потоків на відповідних ядрах на основі вимог до продуктивності пріоритету задачі поточного рівня заряду батареї термального стану процесора. Легкі задачі та фонові процеси виконуються на LITTLE ядрах споживаючи мінімум енергії, інтерактивні додатки використовують середні ядра, тільки найвимогливіші задачі активують потужні big ядра на короткі проміжки часу.

Контрольні запитання

1. Опишіть детально концепцію big LITTLE архітектури ARM включаючи мотивацію її створення для оптимізації енергоспоживання мобільних пристроїв, поясніть відмінності між Performance та Efficiency ядрами з точки зору мікроархітектури розміру кристала частоти споживання енергії, розгляньте типові конфігурації ядер в сучасних мобільних процесорах Qualcomm Snapdragon MediaTek Dimensity Samsung Exynos станом на 2025 рік, наведіть конкретні приклади продуктивності та споживання різних ядер та їх застосування для різних типів задач.

2. Поясніть як операційна система приймає рішення про розміщення задач на відповідних ядрах в гетерогенній системі, опишіть механізми Energy Aware Scheduling в Linux та Android та Intel Thread Director в Windows для x86 гібридних процесорів, розгляньте які метрики використовуються для класифікації задач включаючи використання CPU затримки wakeup latency тип інструкцій, поясніть trade-off між міграцією потоку на більш відповідне ядро та накладними витратами на саму міграцію включаючи втрату теплих кешів, наведіть приклади як різні типи додатків та процесів розподіляються між типами ядер.

Література

1. Stefan Huber, Tobias Lorey, Michael Felderer. Techniques for Improving the Energy Efficiency of Mobile Apps: A Taxonomy and Systematic Literature Review. 2023. URL: <https://arxiv.org/pdf/2308.08292>

2. Dian Shi, Liang Li, Rui Chen, Pavana Prakash, Miao Pan, Yuguang Fang.
Towards Energy Efficient Federated Learning over 5G+ Mobile Devices. 2021.
URL: <https://arxiv.org/abs/2101.04866>

ТЕМА 10. МАЙБУТНЄ ОБЧИСЛЮВАЛЬНИХ СИСТЕМ ЧАСТИНА 1

Мета роботи: набути практичних навичок аналізу перспективних технологій обчислювальних систем, освоїти базові принципи квантових обчислень та їх відмінності від класичних комп'ютерів, навчитися визначати області застосування нейроморфних процесорів, опанувати концепції фотонних обчислень як альтернативи електронним системам, розвинути розуміння технологічних обмежень класичних обчислювальних архітектур, освоїти прогнозування розвитку комп'ютерних технологій на найближче десятиліття.

Практичне завдання

1. Необхідно провести аналіз фундаментальних обмежень сучасних класичних обчислювальних архітектур які стимулюють пошук альтернативних підходів. Розглянути завершення дії закону Мура де кількість транзисторів на кристалі подвоювалася приблизно кожні два роки протягом десятиліть але сповільнилася через фізичні обмеження наближення до атомарних розмірів де сучасні техпроцеси чотири-три нанометри мають транзистори розміром в десятки атомів та квантові ефекти тунелювання стають проблемою, економічні обмеження де вартість нових фабрик для виробництва передових чіпів досягла десятків мільярдів доларів.

2. Розглянути проблему Dennard scaling де раніше зменшення розміру транзисторів автоматично знижувало енергоспоживання на транзистор дозволяючи збільшувати тактову частоту але це припинилося близько 2025 року через проблеми витoku струму leakage та теплового виділення що призвело до переходу на багатоядерність замість зростання частоти.

3. Потрібно описати базові принципи квантових обчислень які використовують квантові біти або кубіти замість класичних бітів. Класичний біт може бути в стані нуль або один в кожен момент часу, кубіт може бути в суперпозиції обох станів одночасно що описується як альфа помножити на нуль плюс бета помножити на один де альфа та бета є комплексними числами з умовою нормалізації, це дозволяє кубіту представляти обидва стани паралельно. Система з N кубітів може бути в суперпозиції всіх два в

степені N можливих станів одночасно що дає експоненційне зростання обчислювального простору, наприклад триста кубітів можуть представляти більше станів ніж атомів у Всесвіті. Розглянути явище квантової заплутаності entanglement де стан одного кубіта нерозривно пов'язаний зі станом іншого навіть на відстані що дозволяє кореляції неможливі в класичних системах та є ресурсом для квантових алгоритмів.

Контрольні запитання

1. Опишіть фундаментальні обмеження сучасних класичних обчислювальних архітектур включаючи завершення закону Мура та Dennard scaling, поясніть фізичні причини цих обмежень пов'язані з атомарними розмірами транзисторів квантовим тунелюванням витоком струму тепловиділенням, розгляньте економічні обмеження зростаючої вартості передових фабрик та зменшення віддачі від інвестицій в зменшення техпроцесу, опишіть як індустрія адаптується до цих обмежень через багатоядерність спеціалізовані акселератори chiplet архітектури трьохвимірне компонування чіпів, обговоріть чи означає це кінець прогресу в обчислювальній потужності чи просто зміну напрямків розвитку.

2. Поясніть базові принципи квантових обчислень включаючи що таке кубіт суперпозиція заплутаність та як вони відрізняються від класичних бітів, опишіть чому квантові комп'ютери мають потенційну експоненційну перевагу для певних класів задач через можливість дослідження експоненційного простору станів паралельно, розгляньте основні виклики створення масштабованих квантових комп'ютерів включаючи декогеренцію шум помилки в квантових вентилях необхідність квантового виправлення помилок що вимагає тисячі фізичних кубітів для одного логічного кубіта, наведіть приклади сучасних квантових платформ та їх параметри станом на 2025 рік.

3. Опишіть для яких типів задач квантові комп'ютери теоретично мають перевагу над класичними та для яких не мають, поясніть алгоритм Шора для факторизації та його потенційний вплив на криптографію

включаючи перехід до постквантової криптографії, розгляньте застосування квантових комп'ютерів для симуляції квантових систем оптимізації машинного навчання, обговоріть реалістичні очікування щодо часових рамок коли квантові комп'ютери стануть практично корисними для реальних задач та чи замінять вони класичні комп'ютери чи будуть спеціалізованими доповненнями, наведіть приклади компаній та організацій що інвестують в квантові обчислення та їх прогрес.

Література

1. Mehdi Hassanpour, Marc Riera, Antonio González. A Survey of Near-Data Processing Architectures for Neural Networks. 2021. URL: <https://arxiv.org/abs/2112.12630>
2. Philip Döbler, Manpreet Singh Jattana. A Survey on Integrating Quantum Computers into High Performance Computing Systems. 2025. URL: <https://arxiv.org/pdf/2507.03540>

ТЕМА 10. МАЙБУТНЄ ОБЧИСЛЮВАЛЬНИХ СИСТЕМ ЧАСТИНА 2

Мета роботи: набути практичних навичок аналізу альтернативних парадигм обчислень включаючи нейроморфні та фотонні системи, освоїти розуміння біоінспірованих архітектур що імітують роботу мозку, навчитися визначати перспективні напрямки розвитку комп'ютерних технологій на період 2025 - 2030 рр., опанувати методики прогнозування технологічних трендів, розвинути критичне мислення щодо реалістичності обіцянок нових технологій, освоїти підготовку до майбутньої професійної діяльності в умовах швидких технологічних змін.

Практичне завдання

1. Необхідно провести детальний аналіз нейроморфних процесорів які імітують архітектуру та принципи роботи біологічного мозку для ефективної обробки задач розпізнавання образів та навчання. На відміну від традиційної архітектури фон Неймана з розділенням обчислень та пам'яті нейроморфні чіпи інтегрують пам'ять та обчислення в штучних нейронах та синапсах подібно до біологічних систем.

2. Розглянути приклади включаючи Intel Loihi 2 який містить 1 мільйон штучних нейронів та 120 мільйонів синапсів з можливістю навчання on-chip споживаючи менше одного вата енергії що на порядки ефективніше за традиційні процесори для задач spike-based обчислень, IBM TrueNorth з 1 мільйоном нейронів та 256 мільйонами синапсів споживаючи лише 70 міліват в активному режимі.

3. Описати принципи роботи нейроморфних систем включаючи спайкові нейронні мережі Spiking Neural Networks де інформація кодується не абсолютними значеннями а часом та частотою спайків імпульсів подібно до біологічних нейронів що дозволяє асинхронну подієво-керовану обробку event-driven computing де енергія споживається тільки коли відбуваються події на відміну від синхронної тактової обробки традиційних процесорів. Розглянути пластичність синапсів можливість змінювати ваги з'єднань між нейронами для навчання безпосередньо в чіпі через алгоритми такі як STDP

Spike-Timing-Dependent Plasticity без необхідності зовнішнього тренування на GPU. Проаналізувати області застосування нейроморфних процесорів включаючи розпізнавання образів та мови в реал-тайм з низькою латентністю для автономних роботів дронів носимих пристроїв, обробка сенсорних даних від event-based камер які генерують дані тільки при зміні в сцені замість повних кадрів з фіксованою частотою що різко знижує обсяг даних та споживання.

Контрольні запитання

1. Опишіть принципи роботи нейроморфних процесорів та їх фундаментальні відмінності від традиційної архітектури фон Неймана, поясніть що таке спайкові нейронні мережі та подієво-керована обробка та чому вони є енергоефективними для певних типів задач, розгляньте приклади сучасних нейроморфних чіпів Intel Loihi IBM TrueNorth та їх характеристики, опишіть області застосування нейроморфних систем де вони мають переваги над традиційними процесорами та GPU включаючи розпізнавання образів обробка сенсорних даних автономні системи, обговоріть виклики масового впровадження нейроморфних систем включаючи необхідність нових алгоритмів та програмних фреймворків обмежену точність спайкових обчислень складність програмування.

2. Поясніть концепцію фотонних обчислень включаючи переваги використання фотонів замість електронів для передачі та обробки інформації такі як вища швидкість нижче енергоспоживання відсутність електромагнітних перешкод можливість wavelength division multiplexing, опишіть сучасний стан фотонних технологій включаючи фотонні інтегральні схеми оптичні інтерконекти в дата-центрах фотонні акселератори для AI, розгляньте виклики створення повністю фотонних комп'ютерів включаючи складність створення оптичних аналогів транзисторів пам'яті логічних елементів необхідність перетворення між оптичними та електронними сигналами, обговоріть реалістичні терміни коли фотонні обчислення можуть

стати комерційно доступними та чи замінять вони повністю електронні системи чи будуть гібридні підходи.

Література

1. Mehdi Hassanpour, Marc Riera, Antonio González. A Survey of Near-Data Processing Architectures for Neural Networks. 2021. URL: <https://arxiv.org/abs/2112.12630>
2. Philip Döbler, Manpreet Singh Jattana. A Survey on Integrating Quantum Computers into High Performance Computing Systems. 2025. URL: <https://arxiv.org/pdf/2507.03540>