

## ЗМІСТ

ВСТУП .....	7
РОЗДІЛ 1. ОГЛЯД ШЕСТИЯДЕРНИХ ПРОЦЕСОРІВ КОМПАНІЇ AMD	8
1.1. Історія розвитку багатоядерних процесорів компанії AMD .....	8
1.1.1. Процесор Athlon 64 X2 .....	8
1.1.2. Процесори Phenom.....	9
1.1.3. Процесори Phenom II .....	13
1.1.4. Архітектура Bulldozer.....	14
1.1.5. Архітектура Piledriver.....	15
1.1.6. Архітектура Steamroller.....	16
1.1.7. Архітектура Excavator .....	17
1.2. Огляд шестиядерного процесора Phenom II 1100T .....	17
1.2.1. Огляд архітектури.....	17
1.2.2. Технологія AMD Turbo Core.....	18
1.2.3. Чіпсет.....	19
1.2.4. Специфікації.....	19
1.2.5. Розгін процесора .....	21
1.3. Огляд шестиядерного процесора FX-6200 .....	21
1.3.1. Відмінності від Phenom II 1100T.....	21
1.3.2. Специфікації.....	22
1.3.3. Розгін процесора .....	23
1.4. Огляд шестиядерного процесора FX-6350 .....	24
1.4.1. Відмінності від FX-6200.....	24
1.4.2. Специфікації.....	25
1.4.3. Розгін процесора .....	26
1.5. Порівняння процесорів Phenom II 1100T, FX-6200, FX-6350 .....	27
1.5.1. Тести.....	27
1.6. Висновки до розділу 1 .....	29
РОЗДІЛ 2. РОЗРОБКА ПРОГРАМИ ПРГ1 ДЛЯ ПКС СП .....	31

2.1. Розробка паралельного математичного алгоритму .....	31
2.2. Розробка алгоритмів процесів .....	32
2.3. Розробка схеми взаємодії процесів .....	34
2.4. Розробка програми ПРГ1 .....	36
2.5. Тестування програми ПРГ 1 .....	37
2.6. Висновки до Розділу 2 .....	41
РОЗДІЛ 3. РОЗРОБКА ПРОГРАМИ ПРГ2 ДЛЯ ПКС ЛП .....	42
3.1. Розробка паралельного математичного алгоритму .....	42
3.2. Розробка алгоритмів процесів .....	42
3.3. Розробка схеми взаємодії процесів .....	47
3.4. Розробка програми ПРГ2 .....	50
3.5. Тестування програми ПРГ 2 .....	51
3.6. Висновки до Розділу 3 .....	54
ОСНОВНІ РЕЗУЛЬТАТИ І ВИСНОВКИ ПО РОБОТІ.....	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	57

## ВСТУП

Для розв'язання багатьох задач необхідна висока продуктивність та висока швидкість передачі інформації по каналах зв'язку, великі об'єми оперативної і постійної пам'яті, які не можуть забезпечити типові обчислювальні засоби. Одним з шляхів забезпечення таких вимог є організація паралельних та розподілених обчислень і відповідних технічних засобів їх реалізації.

Причому, ефективність паралельної обробки залежить як від продуктивності комп'ютерів, так і від розмірів і структури пам'яті, пропускної здатності каналів зв'язку, використаних мов програмування, компіляторів, операційних систем, чисельних методів та інших математичних досліджень. Такий широкий обсяг параметрів вимагає проведення досліджень на різних рівнях: на рівні розпаралелення алгоритмів, створення спеціальних мов програмування, компіляторів, багатопроцесорних систем, неоднорідних систем, кластерів і систем, що розподілені на великих територіях.

В даній курсовій роботі розглядаються шестиядерні процесори AMD та розроблюються алгоритми обчислення заданого математичного виразу в паралельній комп'ютерній системі з спільною пам'яттю та локальною пам'яттю.

Порівнюється прискорення виконання обчислень та ефективність двох паралельних алгоритмів.

## **РОЗДІЛ 1. ОГЛЯД ШЕСТИЯДЕРНИХ ПРОЦЕСОРІВ КОМПАНІЇ AMD**

Сучасні комп'ютерні процесори можна розглядати як комп'ютерну систему зі спільною пам'яттю, адже в кожному процесорі може бути 6 і більше ядер, які мають спільну кеш-пам'ять.[1] В наш час спостерігається швидкий розвиток процесорів для персональних комп'ютерів. Процесор – найшвидший пристрій в комп'ютері, тому потрібно розуміти архітектуру нових процесорів, щоб писати програми, які будуть максимально використовувати можливості процесора.

### **1.1. Історія розвитку багатоядерних процесорів компанії AMD**

Advanced Micro Devices, Inc. (AMD) — компанія виробник інтегрованої електроніки.[2] Це другий найбільший постачальник x86 сумісних процесорів і великий постачальник флеш-пам'яті. Через жорстку цінову конкуренцію з багаторічним суперником Intel фінансові показники не мають стабільності: благополучні періоди чергуються з періодами збитків. При цьому AMD, не економлячи, фінансує свої наукові дослідження (до 20% від обсягу продажу) та розширює свої виробничі потужності. У AMD одні з найсучасніших у галузі виробничі потужності, розташовані в США, Південно-Східній Азії. Що стосується азіатських виробництв AMD, то вони створені в Японії в рамках спільного з Fujitsu підприємства з виробництва модулів флеш-пам'яті на основі 0,35-мікронних технологій. Крім того, AMD має складальні і тестові майданчики у Сінгапурі і Таїланді. Стратегічними партнерами компанії AMD у виробництві персональних комп'ютерів є такі загальновідомі компанії, як Acer, Fujitsu/ICL, Hewlett-Packard і IBM.

#### **1.1.1. Процесор Athlon 64 X2**

Першим двоядерним мікропроцесором компанії AMD для персональних комп'ютерів став Athlon 64 X2. Цей процесор містить два ядра Athlon 64, розміщених на одному кристалі. Ядра мають у своєму розпорядженні загальний двоканальний контролер пам'яті, північний міст і додаткову логіку управління.

Початкові версії засновані на Athlon 64 степінга E і, залежно від моделі, мають 512 або 1024 КБ кешу другого рівня, а також 128 КБ кешу першого рівня на кожне ядро (рис. 1.1). Техпроцес виробництва зберігся з попередніх версій і становить 65 нм.

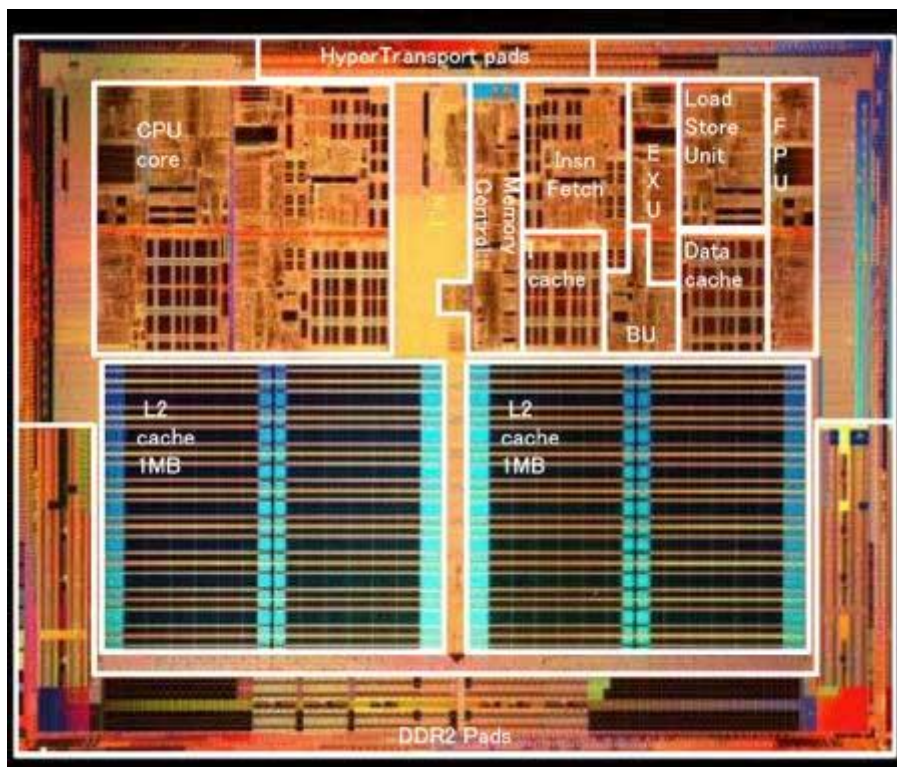


Рис. 1.1 Будова процесорів серії Athlon 64 X2

Athlon 64 X2 підтримує набір інструкцій SSE3 (які раніше підтримувалися тільки процесорами компанії Intel), що дозволило запускати з максимальною продуктивністю код, оптимізований для процесорів Intel.[3] Ці поліпшення не унікальні для Athlon 64 X2 і також є в релізах процесорів Athlon 64, побудованих на ядрах Venice, San Diego і Newark.

### 1.1.2. Процесори Phenom

У 2007 році компанія випустила серію процесорів Phenom для персональних комп'ютерів. AMD Phenom — багатоядерний центральний процесор від компанії AMD. Має: два, три або чотири ядра (Phenom X2, Phenom X3, Phenom X4 відповідно).[4] Базується на архітектурі K10. Трьохядерні версії (кодова назва Toliman) Phenom відносяться до серії 8000, а чотирьохядерні

(кодова назва Agena) до AMD Phenom X4 9000. В цій серії добавлено спільний кеш третього рівня, який дозволяє швидко обмінюватися інформацією між ядрами.

В AMD вважають, що чотирьохядерні Phenom-и були першими «істинними» чотирьохядерними процесорами, тому що саме вони, по суті, є монолітними багатоядерними (рис. 1.2), на відміну від серії процесорів Intel Core 2 Quad, які за дизайном є мультикристально-модульними процесорами.

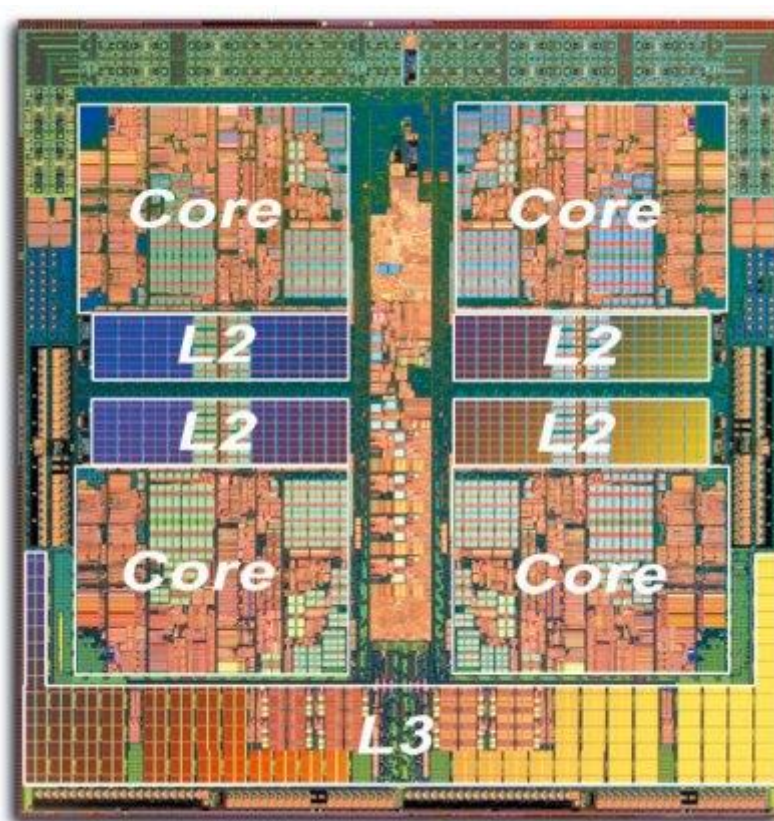


Рис. 1.2 Будова чотирьохядерних процесорів AMD Phenom

З погляду мікроархітектури K10 процесорів AMD, нові 4-ядерні Phenom для настільних ПК, виконані на базі архітектури Stars (ядро Agena) і подібні до нових 4-ядерних серверних процесорів AMD Opteron (ядро Barcelona).

Архітектура Stars має інтегрований в процесор двоканальний 128-бітний контролер пам'яті (2x64-біт, з можливістю спарених (ganged) або розпарених (unganged) режимів роботи) з підтримкою поки тільки DDR2, що працює на

частоті до 1066 МГц. Фізичний адресний простір при цьому збільшився до 48 біт, а підтримка пам'яті – до 256 Тб.

Кожне з чотирьох ядер процесора Phenom має 128 КБ власної кеш-пам'яті L1 (64 КБ для інструкцій і 64 КБ для даних), що в сумі становить 512 КБ кеш-пам'яті L1 на чотирьохядерний процесор. Сумарний об'єм кеш-пам'яті L2 становить 2 МБ, по 512 КБ на кожне ядро. Крім цього, архітектура Stars має 2 МБ кеш-пам'яті L3. На відміну від кеш-пам'яті рівнів L1 і L2, ексклюзивних для кожного ядра, кеш-пам'ять L3 розташована в інтегрованому контролері пам'яті і динамічно розподіляється між всіма ядрами.

Серед важливих характеристик, властивих новим 4-ядерним процесорам Phenom, слід зазначити наступні ключові функціональні можливості:

- Реалізацію 128-бітних (проти 64-бітних в AMD K8) виконавчих пристроїв з плаваючою крапкою (FP), кількість яких залишилася тією ж (3 блоки: FADD, FMUL і FSTORE);
- Наявність нового планувальника завдань для операцій з плаваючою комою, що тепер підтримує 36 нових 128-бітних операцій (SSE4A);
- Підтримку 128-бітних операцій SSE, що з'явилися на додаток до можливостей колишньої 64-бітної архітектури;
- Можливість обробки двох операцій SSE і одного SSE переносу за такт;
- Буфер модуля вибірки інструкцій став 32-байтним (раніше - 16 байт);
- Модуль передбачення розгалужень з 512-ходовим передбаченням непрямих розгалужень;
- Розширення шини L1-LSU (Load-Store Unit) до 2x128 біт (читання) і 2x64 біт (запис);
- Розширення шини L1-L2 кешу ядра процесора до 128 біт (при цьому тип її організації офіційно не розголошується);
- Реалізацію попередньої вибірки даних в L1-кеш процесора (що діє на всіх рівнях кешу процесора, а також оперативної пам'яті);

- Наявність об'єднаного кешу інструкцій/даних третього рівня (L3) спільного стосовно ядер процесора;
- Реалізацію шини HyperTransport 3.0, що дозволила збільшити швидкість обміну з системою до 20,8 Гб/с;
- Підтримку оновленого менеджера продуктивності і енергоспоживання, технології Cool'n'Quiet 2.0;
- Реалізацію технології AMD Virtualization Technology з функцією швидкої індексації Rapid Page Indexing.

Окремої уваги заслуговує технологія Cool'n'Quiet 2.0, в межах якої реалізована серія методів економії електроенергії. Так, на додаток до звичної можливості зниження напруги живлення і тактової частоти всього процесора, реалізована можливість динамічного керування тактовою частотою кожного ядра окремо. Крім цього, додаткове енергозбереження забезпечує технологія AMD CoolCore, що дозволяє відключати функціональні модулі ядра, які не використовуються. Маємо також технологію Dual Dynamic Power Management, що дозволяє задавати незалежні рівні напруги для ядра процесора і інтегрованого контролера пам'яті. Додаткова функція AMD Wideband Frequency Control забезпечує найбільш ефективний з погляду споживання енергії режим при зміні режимів продуктивності. Нарешті, технологія Multi-Point Thermal Control, що функціонує за рахунок інтегрованих у ядро термодатчиків, забезпечує термозахист чіпа, стежачи за рівнем температури і автоматично знижуючи продуктивність при досягненні температурної межі.

Серед інших енергозберігаючих технологій також слід зазначити появу режиму живлення C1E, що активується в момент переходу всіх ядер чіпу в стан очікування. При цьому відключається шина HyperTransport, пам'ять переходить у режим зниженого живлення і значно знижуються внутрішні тактові частоти процесора. Також цікава функція Processor Power Saving Indicator, що "повідомляє" регулятору напруги про зниження навантаження, а також забезпечує підтримку регуляторів напруги з PSI. Разом зі збереженням зворотної



сумісності процесорів AMD Phenom з VID-керуванням плат під роз'ємом AM2 (тобто заявлена можливість роботи процесорів AMD Phenom в «старих» материнських платах з Socket AM2), нова послідовна система VID-керування забезпечує 7-бітну точність регулювання напруги з кроком 12,5 мВ.

### 1.1.3. Процесори Phenom II

В 2010 році було представлено серію процесорів Phenom II, які виготовлені за 45 нм технологічним процесом, і базуються на AMD K10 мікроархітектурі, в свою чергу є наступниками процесорної лінійки Phenom.[5]

Головною особливістю нової архітектури є розширення модельного ряду новими шестиядерними процесорами (рис. 1.3).

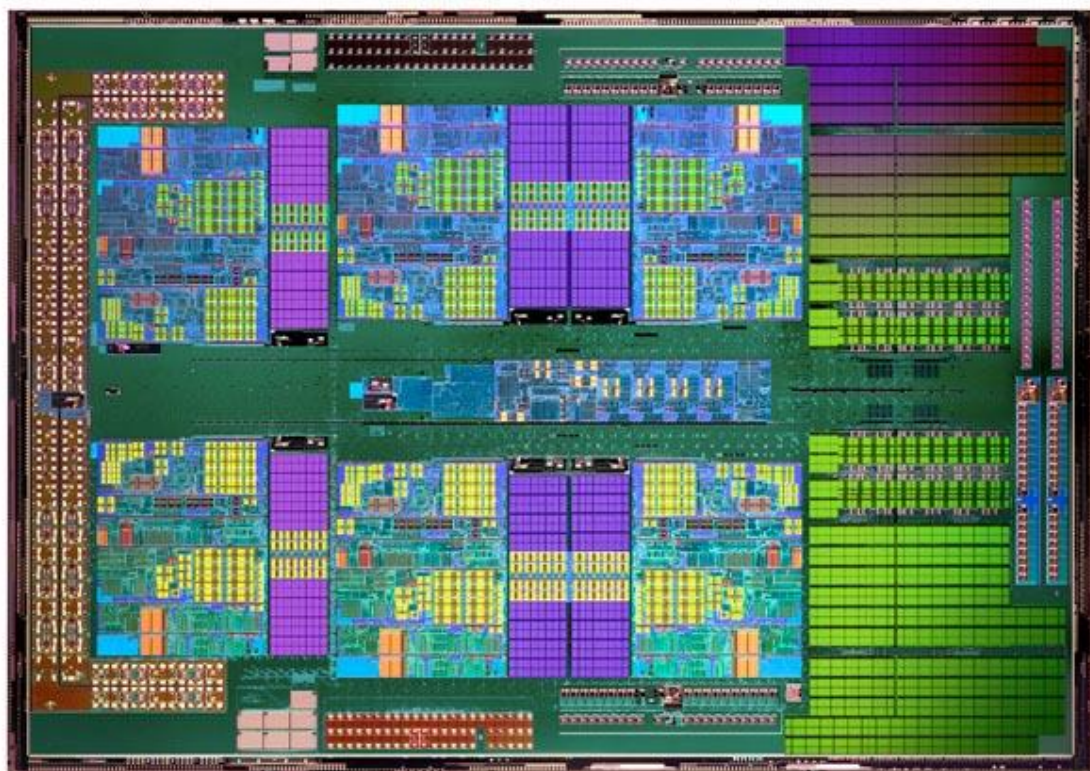


Рис. 1.3 Будова шестиядерних процесорів Phenom II X6

У процесорах Phenom II збільшили втричі загальний розмір кеш-пам'яті L3 від 2 МБ (як в лінії Phenom) до 6 МБ, завдяки цьому приріст продуктивності збільшився до 30%. Ще одна відмінність від попередніх Phenom це те, що технологія Cool'n'Quiet тепер застосовується для процесора в цілому, а не для

кожного окремого ядра. Це було зроблено з метою уникнення неправильної обробки потокових обчислень у Windows Vista, яка могла переводити однопотокові обрахунки на інші ядра, які працювали в режимі холостого ходу на частоті, зменшеній в два рази, в результаті продуктивність процесу також зменшувалася вдвічі.

Нові шестиядерні та чотирьохядерні процесори AMD Phenom II сумісні з материнськими платами з Socket AM3 і AM2+, хоча доведеться оновити BIOS.

#### **1.1.4. Архітектура Bulldozer**

В 2011 році AMD перейшла на нову архітектуру з кодовою назвою Bulldozer. Процесори покоління AMD K11 виготовляються за 32-нм технологією і призначені для серверів і високопродуктивних ПК.

Процесори Bulldozer, за запевненням представників AMD, мають повністю нову архітектуру в порівнянні з попередніми поколіннями AMD K8 і AMD K10. Тому компанія разом з процесорами презентувала новий сокет та чіпсет. Нові процесори сумісні з материнськими платами із Socket AM3+.[6]

Основною особливістю цієї архітектури є те, що вона модульна. Модуль складається з двох фізичних ядер зі спільною кеш-пам'яттю другого рівня. У нових процесорах для персональних комп'ютерів може бути 2, 3, або 4 модуля, тобто 4, 6, або 8 ядер відповідно. З цього зрозуміло, що AMD випустила перший 8-ядерний процесор для ринку ПК, оскільки в процесорів Intel максимальна кількість ядер рівна шести. Процесори Bulldozer вперше підтримують нові інструкції x86 (SSE4.1, SSE4.2, CVT16, AVX і XOP, в тому числі 4-операндний модуль FMAC). Кожне ядро має 128-бітний модуль FPU з підтримкою FMA, при чому ці модулі можуть об'єднуватися в один загальний 256-бітний FPU між двома ядрами. Цю конструкцію супроводжують два модуля обчислень над цілими числами (по одному на кожне ядро) з 4-ма лініями зв'язку і можливістю спільної вибірки та декодування. Таким чином, один модуль з двома ядрами еквівалентний двохядерному процесору при операціях з цілими числами та

однойдерному процесору при роботі над числами з плаваючою крапкою. Також кожен модуль має 2 МБ кешу другого рівня. А кеш третього рівня загальний для всіх модулів. Його розмір залежить від кількості модулів і може займати 4, 6, або 8 МБ.

У новій архітектурі введена підтримка нової версії технології AMD Direct Connect і чотирьох каналів HyperTransport 3.1 на кожен процесор. Можливість роботи з пам'яттю DDR3 і технологією розширення пам'яті AMD G3MX дозволить збільшити пропускну здатність процесора. Також покращено керування живленням.

Нові процесори отримують підтримку технології Turbo Core 2, яка дозволяє збільшити номінальну частоту всього процесора на 500МГц, або половину ядер на 1ГГц, і помітно підвищити продуктивність аналогічно технології Intel Turbo Boost.

### **1.1.5. Архітектура Piledriver**

В 2012 році компанія переходить на нову мікроархітектуру Piledriver. Piledriver – мікроархітектура, розроблена AMD в якості наступника Bulldozer. Piledriver використовує той же модульний дизайн. Одними з головних відмінностей в оновленій модульній архітектурі є поліпшені модуль передбачення розгалужень і планування використання модулів цілих чисел і чисел з плаваючою крапкою, поряд з переходом на новий тип тригерів з поліпшеними показниками енергоспоживання. На практиці це призвело до зросту частоти на 8-10 % та збільшення продуктивності приблизно на 15 % з аналогічним енергоспоживанням.

Спочатку в продажі вийшли процесори типу AMD Accelerated Processing Unit (APU) з кодовою назвою Trinity та серія мобільних продуктів.[7] Пізніше також розповсюдилися процесори FX-серії. Відмінністю APU від серії FX є наявність вбудованого графічного ядра (рис 1.4).

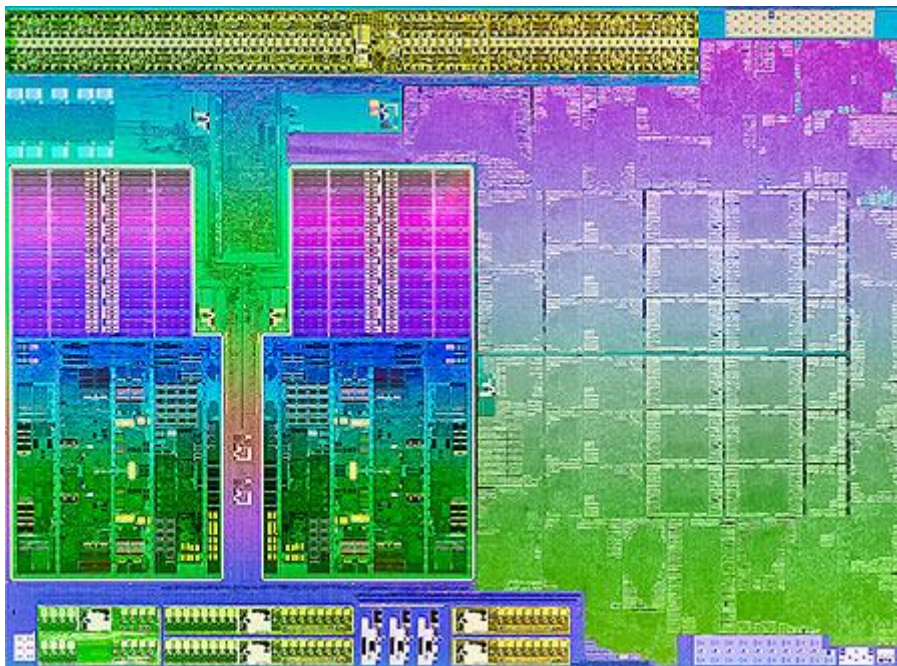


Рис. 1.4 Будова процесорів AMD APU

Його потужності вистачає для роботи з мультимедіа, а розташування центрального і графічного процесорів на одному кристалі дешевше, ніж два елементи окремо. Графічний процесор займає близько половини площі кристалу, що значно більше, ніж в аналогів від Intel. У ГП відсутня внутрішня пам'ять, тому він резервує частину оперативної пам'яті для власних потреб. Через відведення великої частини площі кристалу на ГП, довелося змінити архітектурі центрального процесору, а саме: відмовитися від використання кешу третього рівня ( в серії FX розподіл кешу залишився без змін).

З появою нової архітектури компанія AMD також націлилася на ринок мобільних комп'ютерів в низькому та середньому ціновому діапазоні. Тому були презентовані одно- та двомодульні APU з малим TDP (17-35 Вт).

Для встановлення процесорів FX-серії використовуються старий сокет AM3+, але для серії APU презентовано новий Socket FM2.

#### **1.1.6. Архітектура Steamroller**

На початку 2014 року компанія анонсувала нову архітектуру Steamroller. В Steamroller відсутня FX-серія, але APU продовжують використовувати модульні

структуру, як і їх попередники, одночасно спрямовані на досягнення вищого рівня паралелізму.

Відмінністю нових модулів є розділені декодери інструкцій для кожного ядра в модулі. Також на 25% збільшено шину розсилки на ядро, оновлені планувальники інструкцій, більші і гнучкіші кеші (кеш другого рівня може динамічно змінювати розмір), додана черга мікрооперацій, на кристалі розміщено більше регістрових ресурсів, покращений контролер пам'яті.

Згідно з оцінками з AMD, ці поліпшення збільшать кількість виконуваних інструкцій за такт до 30 % в порівнянні з ядром першого покоління Bulldozer при збереженні високих тактових частот Piledriver, але із зменшеним енергоспоживанням.[8] В середньому нові процесори швидші за попередників на 9 % в однопоточних програмах та на 18% в багатопоточних.

### **1.1.7. Архітектура Excavator**

В 2015 році запланований перехід на архітектуру Excavator. Зараз ця мікроархітектура на стадії розробки. Excavator APU будуть випускатися під кодовою назвою Carrizo. Очікується, підтримка нових інструкцій, таких як AVX2, BMI2 і RDRAND. Також очікується оновлення контролерів пам'яті для підтримки пам'яті стандартів DDR3 і DDR4.[9]

Excavator буде останньою архітектурою на базі архітектури Bulldozer.

## **1.2. Огляд шестиядерного процесора Phenom II 1100T**

### **1.2.1. Огляд архітектури**

У відповідь на представлений десктопний шестиядерний процесор Intel Core i7-980X Extreme компанія AMD анонсувала власні шестиядерні моделі процесорів AMD Phenom II X6 з кодовим ім'ям Thuban, які відрізняються в першу чергу більш прийнятною вартістю. На відміну від Intel Core i7-980X Extreme процесори AMD Phenom II X6 виробляються по 45 нм процесу ядра, як і попередні, Deneb.

Архітектура нових процесорів Thuban від Deneb відрізняється в основному наявністю ще двох процесорних ядер. Об'єм кеш-пам'яті 3-го рівня не змінився, він, як і раніше, складає 6 Мб. Кожне ядро процесора має 512 Кб власної кеш-пам'яті другого рівня, як і в Deneb.

Нові шестиядерні AMD Phenom II сумісні з материнськими платами з Socket AM3 і AM2+, після оновлення BIOS. Тому, перед покупкою, необхідно уточнити на сайті виробника чи додав він підтримку нового процесора в конкретній материнській платі.

### **1.2.2. Технологія AMD Turbo Core**

Спеціально для додатків, не оптимізованих під багатопоточне виконання, у нових процесорах AMD Phenom II X6 була реалізована технологія AMD Turbo Core за аналогією Intel Turbo Boost.

Технологія AMD Turbo Core значно відрізняється від технології конкурента Intel Turbo Boost. По-перше, AMD Turbo Core збільшує частоту не в декілька щаблів залежно від кількості задіяних ядер, а лише однократно збільшуючи частоту на 500 МГц вище номінальної. По-друге, технологія AMD Turbo Core підвищує частоту одразу трьох ядер, за умови, що три інші ядра не діють. При цьому частота непрацюючих ядер знижується до 800 МГц. Остання умова необхідна для того, щоб в режимі прискорення процесор не виходив за рамки свого теплового пакета, тому що напруга на процесорі примусово збільшується.

Кажучи коротко, технологія AMD Turbo Core викликає наступний ефект:

- Частота трьох процесорних ядер, що простоюють, знижується до 800 МГц;
- Збільшується напруга на всіх процесорних ядрах;
- Частота трьох активних ядер процесора збільшується на 500 МГц вище штатної.



### 1.2.3. Чіпсет

Разом з новими процесорами був представлений і новий «топовий» чіпсет – AMD 890FX.

Між умовно старшою системною логікою AMD 790FX і новою AMD 890FX дуже мало відмінностей. Фактично вся різниця криється в появі нового південного моста SB850, який тепер став підтримувати інтерфейс SATA 3.0. Чіпсет AMD 890FX, як і його попередник AMD 790FX, виконаний по 65 нм техпроцесу, тому і тепловий пакет у нього залишився попереднім, рівним 19,6 Вт. А ось набір системної логіки AMD 890GX виконується по 55 нм техпроцесу, але тепловий пакет у нього вище - 25 Вт, через наявність інтегрованого відеоприскорювача Radeon HD 4290. Основна особливість чіпсету AMD 890FX полягає в наявності 44-х ліній PCI Express 2.0, що дозволяє поєднувати за допомогою технології CrossFireX дві дискретні відеокарти в конфігурації 16x+16x або чотири відеокарти в конфігурації x8+x8+x8+x8.

### 1.2.4. Специфікації

Важливими характеристиками процесора є кількість ядер, тактова частота, об'єм кеш-пам'яті, сокет і тепловий пакет. Якщо процесор буде використовуватися для багатопоточних програм, то чим більша кількість ядер – тим краще. Також при обчисленнях велику роль відіграє об'єм кешу. Великий об'єм кешу зменшує кількість кеш-промахів, що позитивно впливає на загальну швидкодію.

Основні специфікації процесора Phenom II 1100T зібрані в таблиці 1.1.

Таблиця 1.1 Специфікації процесора Phenom II 1100T

Модель	AMD Phenom II X6 1100T
Маркування	HDE00ZFBK6DGR
Процесорний роз'єм	Socket AM3, AM2+
Тактова частота, МГц	3300

Множник	16,5
Частота шини HT, МГц	2000
Об'єм кеш-пам'яті L1, КБ	128 x 6
Об'єм кеш-пам'яті L2, КБ	512 x 6
Об'єм кеш-пам'яті L3, КБ	6144
Ядро	Thuban
Кількість ядер	6
Напруга живлення, В	1,470
Тепловий пакет, Вт	125
Тактова частота в режимі AMD Turbo Core, МГц	до 3800
Критична температура, °C	62
Техпроцес, нм	45
Підтримка технологій	AMD Turbo Core Cool'n'Quiet 3.0 CoolCore Technology Dual Dynamic Power Management Enhanced Virus Protection Virtualization Technology Core C1 and C1E states Package S0, S1, S3, S4 and S5 states
Вбудований контролер пам'яті	
Типи пам'яті	DDR2-800/1066 DDR3-800/1066/1333/1600
Кількість каналів пам'яті	2
Максимальний об'єм пам'яті, ГБ	16
Максимальна пропускна здатність, ГБ/с	21,3
Підтримка ECC	є



### **1.2.5. Розгін процесора**

Оскільки даний процесор належить до серії Black Edition, його розгін проводився шляхом звичайного підвищення множника. При піднятті множника до 20 процесор успішно пройшов тестування стабільності на частоті 4000 МГц.

Для процесорів AMD характерна ситуація, коли найбільший приріст продуктивності одержують моделі з більш низькою номінальною тактовою частотою, тому що моделі процесорів, засновані на тому самому ядрі, мають загальну межу тактової частоти для повітряного охолодження. Як флагманська модель, що працює на частоті 3300 МГц, процесор AMD Phenom II X6 1100T показує найнижчий приріст продуктивності серед моделей на ядрі Thuban – близько 11 %. Проте, шість ядер, кожне з яких стабільно працює на частоті 4000 МГц, все ж показали непоганий результат. В тестах приріст продуктивності коливався в межах від 0,17% до 21,09%.

## **1.3. Огляд шестиядерного процесора FX-6200**

### **1.3.1. Відмінності від Phenom II 1100T**

Модель AMD FX-6200, анонсована наприкінці лютого 2012 року разом з AMD FX-4170. Даний ЦП виготовлений по 32 нм техпроцесу і складається з трьох обчислювальних модулів, кожний з яких містить по два ядра, що разом дає нам шість ядер. Процесор також відрізняється підвищеною до 3,8 ГГц, але термопакет залишився без змін - 125 Вт.

Корпус процесора такий же, як і у інших представників сімейства AMD FX. Та і від більш старих Phenom II для Socket AM3 мало чим відрізняється. Масивна металева кришка захищає кремнієву основу процесора від пошкоджень при встановленні системи охолодження і одночасно забезпечує рівномірний розподіл тепла, запобігаючи перегріву окремих ділянок кристала. Маркування на кришці дає користувачу максимум інформації про процесор. Для цієї моделі вона наступна - FD6200FRW6KGU:

- F – процесор належить до сімейства AMD FX;

- D – сфера застосування даного процесора – робочі станції;
- 6200 – модельним номер;
- FR – тепловий пакет процесора 125 W;
- W – впакований процесор у корпус 938-pin Socket AM3+;
- 6 – загальна кількість активних ядер;
- K – об'єм кеш-пам'яті L2 2 МБ на кожний модуль і 8 МБ кеш-пам'яті L3;
- GU - ядро процесора степінга OR-B2

### 1.3.2. Специфікації

Дана модель є найпотужнішим шестиядерним процесором на архітектурі Bulldozer. Детально ознайомитися із її характеристиками можна в таблиці 1.2.

Таблиця 1.2 Специфікації процесора AMD FX-6200

Модель	<b>AMD FX-6200</b>
Маркування	FD6100WMW6KGU
Процесорний роз'єм	Socket AM3+
Тактова частота (номінальна), МГц	3800
Максимальна тактова частота з ТС 2.0, МГц (для всіх ядер)	4100
Множник (номінал)	16,5
Частота шини HT, МГц	2200
Об'єм кеш-пам'яті L1, КБ	3 x 64 (інструкції) 6 x 16 (дані)
Об'єм кеш-пам'яті L2, МБ	3 x 2
Об'єм кеш-пам'яті L3, МБ	8
Ядро	Zambezi
Кількість ядер	6
Напруга живлення, В	0,75-1,40

Тепловий пакет, Вт	125
Критична температура, °C	75
Техпроцес, нм	32
Підтримка технологій	Multiple low-power states Enhanced Virus Protection Advanced Power Management Virtualization Technology Hardware Thermal Control Core C0, C1, C1E, C6, CC6, states Package S0, S1, S3, S4 and S5 states AMD Turbo CORE technology 2.0
Вбудований контролер пам'яті	
Типи пам'яті	DDR3-1066/1333/1600/1866
Кількість каналів пам'яті	2
Максимальний об'єм пам'яті, ГБ	16
Максимальна пропускна здатність, ГБ/с	21,3
Підтримка ECC	є

### 1.3.3. Розгін процесора

Однією з основних переваг лінійки AMD FX виробник вказує розблокований множник процесора, про що додатково нагадує маркування Black Edition, розміщене на упаковці. Розблокований множник дає можливість розігнати AMD FX-6200 вручну до частот недоступних при використанні Turbo Core. І головне що при цьому не обов'язково використовувати рідинне охолодження і дорогі оверклокерські материнські плати.

При збільшенні множника, для забезпечення стабільної роботи системи, довелося також підвищити напругу живлення, що подається на процесор. Максимальна частота, при якій тестова система працювала без помилок, досягла

4786 МГц. При цьому напругу довелося підвищити до 1,5 В, а процесорний множник був встановлений на x23,5. Цікаво, що молодший процесор AMD FX-6100 вдалося розігнати трохи більше, але для цього довелося піднімати частоту системної шини на 15 МГц, а із AMD FX-6200 системну шину довелося підняти тільки на 3 МГц.

Середній приріст продуктивності за рахунок розгону склав 16,39%. Показник досить непоганий, і відчутти вигравш можна практично в будь-якому додатку. Головне не варто забувати, що при роботі на підвищених частотах процесор працює на межі своїх можливостей, тому, не подбавши про гарне охолодження, його можна спалити.

## **1.4. Огляд шестиядерного процесора FX-6350**

### **1.4.1. Відмінності від FX-6200**

AMD FX-6350 є представником високопродуктивної лінійки компанії AMD. Він не обладнаний графічним прискорювачем, але при цьому має досить високу швидкодію та шість обчислювальних ядер. У лінійці AMD FX є два шестиядерні процесори, які основані на мікроархітектурі AMD Piledriver і входять у сімейство AMD Vishera.

Процесор AMD FX-6350 має класичний корпус, який за формою не відрізняється від інших моделей компанії AMD. За написом на лицьовій стороні можна довідатися, що сам кристал вирощений у Німеччині, а остаточна збірка проведена в Малайзії. Що стосується маркування CPU, то при його розшифруванні можна скласти досить детальну картину про даний пристрій. У нашому випадку FD6350FRW6KHK має таке позначення:

- F – сімейство AMD FX;
- D – сегмент робочих станцій;
- 6350 – номер моделі;
- FR – тепловий пакет 125 Вт;
- W – процесорний роз'єм Socket AM3+;

- 6 – загальна кількість активних ядер;
- К – обсяг кеш-пам'яті L2 – 2 МБ на кожний двоядерний модуль і 8 МБ загальної кеш-пам'яті L3;
- НК - ядро процесора степінга OR-C0.

#### 1.4.2. Специфікації

Дана модель є найпотужнішим шестиядерним процесором на архітектурі Bulldozer. Відмінністю від специфікацій FX-6200 є змінена частота, додаткові технології та відсутність критичної температури. Детально ознайомитися із її характеристиками можна в таблиці 1.2.

Таблиця 1.2 Специфікації процесора AMD FX-6350

Модель	<b>AMD FX-6350</b>
Маркування	FD6100WMW6KGU
Процесорний роз'єм	Socket AM3+
Тактова частота (номінальна), МГц	3900
Максимальна тактова частота з ТС 2.0, МГц (для всіх ядер)	4200
Множник (номінал)	16,5
Частота шини HT, МГц	2200
Об'єм кеш-пам'яті L1, КБ	3 x 64 (інструкції) 6 x 16 (дані)
Об'єм кеш-пам'яті L2, МБ	3 x 2
Об'єм кеш-пам'яті L3, МБ	8
Ядро	Zambezi
Кількість ядер	6
Напруга живлення, В	0,888-1,404
Тепловий пакет, Вт	125

Критична температура, °C	-
Техпроцес, нм	32
Підтримка технологій	Multiple low-power states Enhanced Virus Protection Advanced Power Management Virtualization Technology Hardware Thermal Control Core C0, C1, C1E, C6, CC6, states Package S0, S1, S3, S4 and S5 states AMD Turbo CORE technology 2.0
Вбудований контролер пам'яті	
Типи пам'яті	DDR3-1066/1333/1600/1866
Кількість каналів пам'яті	2
Максимальний об'єм пам'яті, ГБ	16
Максимальна пропускна здатність, ГБ/с	21,3
Підтримка ECC	є

### 1.4.3. Розгін процесора

Розблокований множник даного процесора значно підвищує його розгінний потенціал. При оверклокінгу була відключена функція AMD Turbo Core 3.0, а значення множника зафіксоване на позначці «x23,5». Також була підвищена напруга до 1,160 В і проведений повний тест стабільності та нагрівання, при якому температура CPU не перевищувала 59°C (з використанням стендового кулера замість боксової версії). Після всіх перерахованих операцій була досягнута динамічна тактова частота в 4716 МГц, що є прекрасним результатом.

У підсумку середній приріст продуктивності склав 14,30%, що є хорошим результатом. Це дозволило наблизитися до показників більш дорогого AMD FX-8350. Але також варто враховувати, що й так не найнижче енергоспоживання всієї системи (199 Вт при максимальному навантаженні) після розгону ще більше зросте (281 Вт при максимальному навантаженні), тому дані дії варто проводити тільки після заміни штатної системи охолодження на більш потужну.

## 1.5. Порівняння процесорів Phenom II 1100T, FX-6200, FX-6350

### 1.5.1. Тести

Для оцінки продуктивності процесорів використовують спеціалізоване програмне забезпечення. Серед найвідоміших найбільше підходять такі, як GeekBench, 3D Mark 11 (Physics), Passmark.

Програма GeekBench симулює сценарії поведінки середньостатистичного користувача, а також має режим тестування на максимальній продуктивності. Загальні бали з різних тестів додаються і виводиться результат. Тестування проходить в чотирьох можливих режимах: виконання програми в багатопоточному режимі або однопоточному режимі, в 32-бітному середовищі, або в 64-бітному. В даному тестуванні використовувалася третя версія програми в 32-бітному середовищі (рис. 1.5).

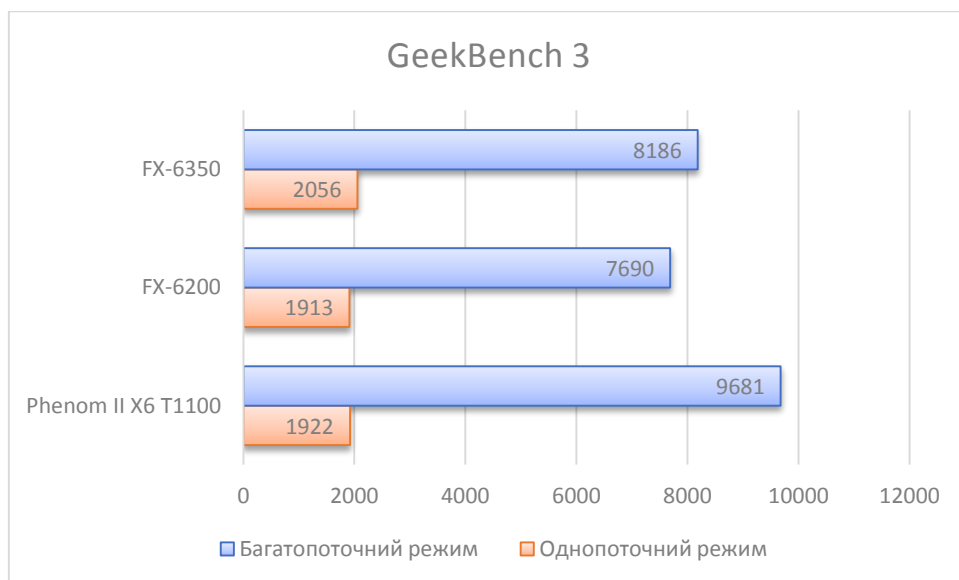


Рис. 1.5 Результати тестів програми GeekBench 3

3D Mark 11 (Physics) – ще один відомий засіб для тестування процесора. В цій програмі процесор проводить різні розрахунки з моделювання фізичних явищ, а також емулює роботу штучного інтелекту. Обидві операції є дуже ресурсоємними і гарно піддаються розпаралелюванню. Отримані результати проілюстровані на рис. 1.6.

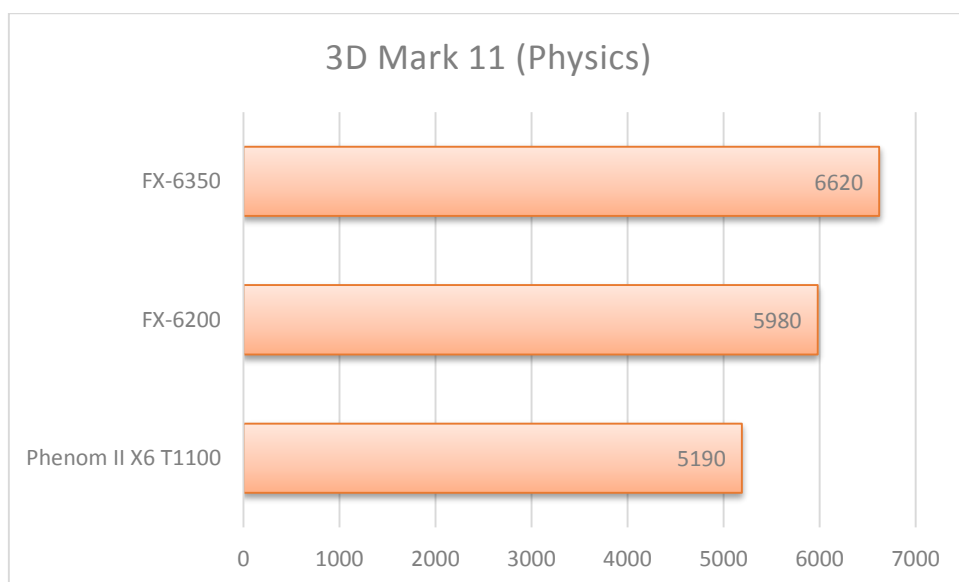


Рис. 1.6 Результати тестів програми 3D Mark 11 (Physics)

Passmark – комплексна програма для тестування центрального процесора. До неї входять 8 різних тестів, для перевірки різних елементів процесора, щоб дати адекватну оцінку його швидкодії. Отримані результати проілюстровані на рис. 1.7



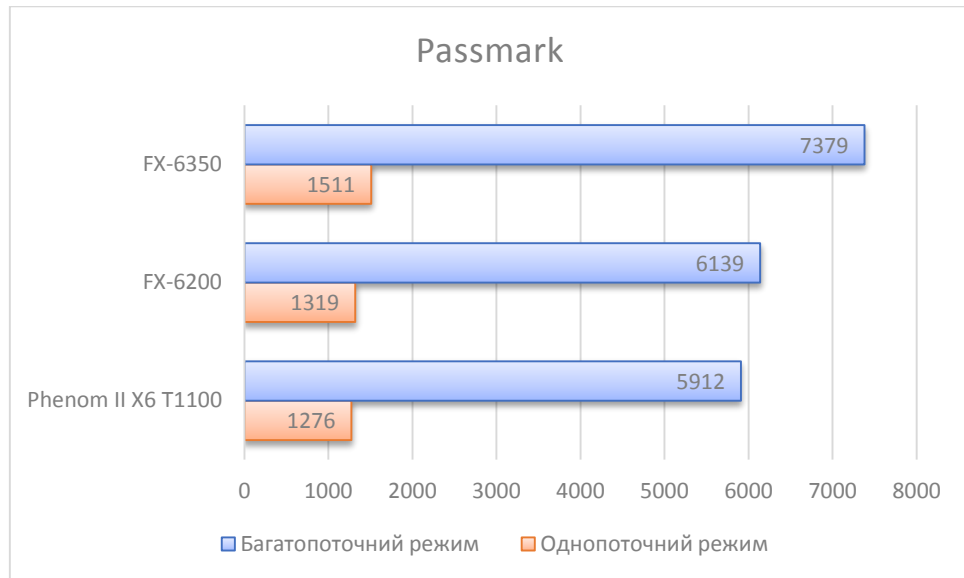


Рис. 1.7 Результати тестів програми Passmark

### 1.6. Висновки до розділу 1

1. Компанія AMD за останні дев'ять років випустила 6 поколінь багатоядерних процесорів. Починаючи з двоядерних і закінчуючи восьмиядерними процесорами для ринку персональних комп'ютерів.
2. З точки зору архітектури ці покоління можна поділити на 2 частини. В першу входять Athlon 64 X2, Phenom та Phenom II. В другу – Bulldozer, Piledriver, Srteamroller. Останні відрізняються від перших наявністю модульної архітектури. Тобто, процесор поділяється не на ядра, а на модулі, кожен з яких містить по два ядра.
3. Процесори з архітектурами Phenom та Phenom II відрізняються від Athlon 64 X2 тим, що у них є кеш-пам'ять третього рівня. Ця пам'ять спільна для всіх ядер і дозволяє швидко обмінюватися даними.
4. Процесори з архітектурами Piledriver і Srteamroller, на відміну від попередніх поколінь, мають серії процесорів з вбудованим графічним прискорювачем.
5. Огляд шестиядерних процесорів показав, що з кожним поколінням зростає частота процесора і на архітектурі Srteamroller перетнула межу в 4ГГц. В цілому приріст в частотах між першим і поточним поколінням

шестиядерних процесорів становить 18%. А також 33% приросту кеш-пам'яті третього рівня.

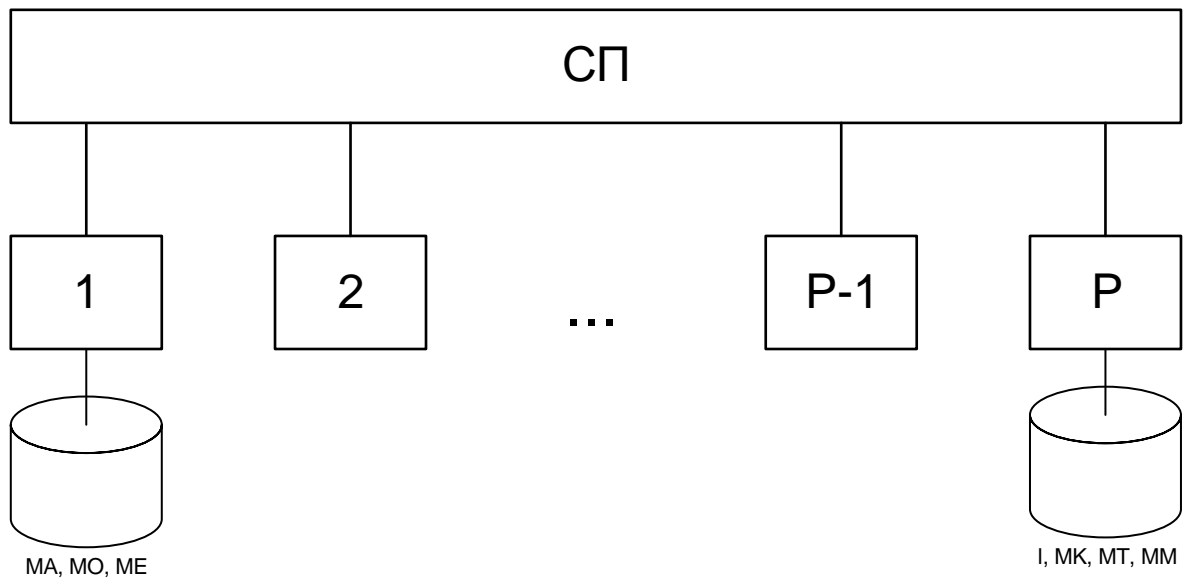
6. Тести в GeekBench показали перевагу процесора Phenom II 1100T в багатопоточному режимі на 22% над FX-6350 і на 27% над FX-6200. Це пояснюється тим, що кожне ядро цього процесора має власний FPU, а в інших двох процесорів FPU збільшений вдвічі, але один на два ядра. Зате в однопоточному режимі вже не було такої переваги. Результати всіх процесорів майже однакові. І в цілому залежать від частоти процесора, тому перевагу отримав FX-6350.
7. В тесті 3D Mark 11 (Physics) найкраще себе показав процесор FX-6350, обігнавши процесор FX-6200 на 11% і процесор Phenom II 1100T на 28%. Завдяки високим частотам на ядрах та великим розміром кеш-пам'яті третього рівня.
8. Результати програми Passmark показали, що в багатопоточному режимі процесори FX-6350 і FX-6200 отримали більше балів, ніж процесор Phenom II 1100T, на 25% і на 3% відповідно. Оскільки цей тест має більше обчислень з цілими числами, то на результат впливає найбільше частота процесора, і трохи менше – розмір кешу. І в однопоточному режимі також лідером є FX-6350. В даному випадку це заслуга лише вищої частоти та оптимізованої архітектури.
9. Оскільки TDP всіх трьох процесорів становить 125 Вт, то найдоцільнішим вибором з точки зору економії електроенергії, відводу тепла і швидкодії є процесор FX-6350.

## РОЗДІЛ 2. РОЗРОБКА ПРОГРАМИ ПРГ1 ДЛЯ ПКС СП

Математична задача:

$$MA = \min(MO) * MK + l * ME * (MT * MM)$$

Структура ПКС СП зображена на рис. 2.1. Від і вивід даних відбувається в крайніх процесах.



### 2.1. Розробка паралельного математичного алгоритму

Згідно з технічним завданням необхідно розробити паралельний алгоритм.

Його можна розділити на наступні етапи:

1)  $e_i = \min(MO_H), i = 1..P.$

2)  $e = \min(e, e_i), \forall P: e.$

3)  $MA_H = e * MK_H + l * ME * (MT * MM_H), \forall P: e, ME, MT.$

Пояснення до використовуваних констант:

- $N$  – розмірність векторів і матриць;
- $P$  – кількість ядер;
- $H = \frac{N}{P}.$

Оцінка прискорення і ефективність розробленого алгоритму, спираючись на теорему Мунро-Петерсона:

$$t_P = \lceil \log_2(m + 1) \rceil$$

де  $m$  – кількість бінарних операцій;

$t_P$  – час розв'язання задачі на  $P$  вузлах.

1. Операція множення двох матриць розмірністю  $N * N$  потребує  $N$  операцій множення і  $N - 1$  операцій додавання для формування одного елемента матриці МА. Отже,  $m_1 = 2N - 1$ ,  $t_{P1} = \lceil \log_2(m_1 + 1) \rceil = \lceil \log_2(2N) \rceil = 1 + \log_2(N)$ .
2. Для виконання пошуку мінімуму в одному рядку матриці розмірністю  $N * N$  необхідно  $N$  порівнянь. Якщо вже знайдені мінімуми в рядках, то для пошуку мінімуму в матриці знадобиться ще  $N$  порівнянь. Отже,  $m_2 = 2N$ ,  $t_{P2} = \lceil \log_2(m_2 + 1) \rceil = \lceil \log_2(2N + 1) \rceil = 2 + \log_2(N)$ .
3. Операція додавання виконується за один такт  $t_{P3} = 1$ .

Отже, сумарний час виконання операції:

$$t_P = t_{P1} + t_{P2} + t_{P3} = 1 + \log_2(N) + 2 + \log_2(N) + 1 = 4 + 2 * \log_2(N)$$

Якщо  $N = 1000$ , то  $t_P = 22$

## 2.2. Розробка алгоритмів процесів

### Задача $T_1$ :

- 1) Введення  $ME, MO$ .
- 2) Сигнал  $T_j$  ( $j = 2..P$ ) про закінчення вводу.  $S_{j;1.1} (j = 2..P)$
- 3) Чекати закінчення введення в  $T_P$ .  $W_{P;1.P}$
- 4) Обчислення1  

$$e_1 = \min(MO_H)$$
- 5) Обчислення2  $КД$   

$$e = \min(e, e_1)$$

- 6) Сигнал  $T_j$  ( $j = 2..P$ ) про закінчення Обчислення2.  $S_{j;2.1}(j = 2..P)$   
 7) Чекати закінчення Обчислення2 в  $T_j$  ( $j = 2..P$ ).  $W_{j;2.j}(j = 2..P)$   
 8) Копії КД

- $e_1 = e$
- $l_1 = l$
- $ME_1 = ME$
- $MT_1 = MT$

- 9) Обчислення3

$$MA_H = e_1 * MK_H + l_1 * ME_1 * (MT_1 * MM_H)$$

- 10) Чекати закінчення обчислення  $MA_H$  в  $T_j$  ( $j = 2..P$ ).  $W_{j;3.j}(j = 2..P)$   
 11) Виведення  $MA$ .

### Задача $T_P$ :

- 1) Введення  $l, MK, MT, MM$ .  
 2) Сигнал  $T_j$  ( $j = 1..P - 1$ ) про закінчення вводу.  $S_{j;1.2}(j = 1..P - 1)$   
 3) Чекати закінчення введення в  $T_1$ .  $W_{1;1.1}$   
 4) Обчислення1  

$$e_p = \min(MO_H)$$
  
 5) Обчислення2 КД  

$$e = \min(e, e_1)$$
  
 6) Сигнал  $T_j$  ( $j = 1..P - 1$ ) про закінчення Обчислення2.  $S_{j;2.2}(j = 1..P - 1)$   
 7) Чекати закінчення Обчислення2 в  $T_j$  ( $j = 1..P - 1$ ).  $W_{j;2.j}(j = 1..P - 1)$   
 8) Копії КД
- $e_p = e$
  - $l_p = l$
  - $ME_p = ME$

- $MT_p = MT$

9) Обчислення3

$$MA_H = e_p * MK_H + l_p * ME_p * (MT_p * MM_H)$$

10) Сигнал  $T_1$  про закінчення обчислення  $MA_H$ .  $S_{1;3.2}$

**Задача  $T_i$  ( $i = 2..P - 1$ ):**

1) Чекати закінчення введення в  $T_1, T_P$ .  $W_{1,P;1.1,1.P}$

2) Обчислення1

$$e_j = \min(MO_H)$$

3) Обчислення2 КД

$$e = \min(e, e_j)$$

4) Сигнал  $T_j$  ( $j = 2..P - 1, j \neq i$ ) про закінчення

Обчислення2.  $S_{j;2.i}$  ( $j = 2..P - 1, j \neq i$ )

5) Чекати закінчення Обчислення2

в  $T_j$  ( $j = 2..P - 1, j \neq i$ ).  $W_{j;2.j}$  ( $j = 2..P - 1, j \neq i$ )

6) Копії КД

- $e_j = e$
- $l_j = l$
- $ME_j = ME$
- $MT_j = MT$

7) Обчислення3

$$MA_H = e_j * MK_H + l_j * ME_j * (MT_j * MM_H)$$

8) Сигнал  $T_1$  про закінчення обчислення  $MA_H$ .  $S_{1;3.i}$

### 2.3. Розробка схеми взаємодії процесів

На основі алгоритму процесів розроблені структурні схеми моніторів (рис. 2.2, 2.3), що реалізовані з допомогою захищеного модуля мови програмування Ада.

Для взаємодії процесів використовується два захищені модулі:

- 1) *ResourceMonitor* – призначений для вирішення задачі взаємного виключення.
- 2) *SynchronizeMonitor* – призначений для вирішення задачі синхронізації.

Захищений модуль *ResourceMonitor* має захищені функції, що дозволяють лише копіювати спільний ресурс:

- *CopyE* – копіювання спільного ресурсу – мінімального елемента матриці *MO*.
- *CopyL* – копіювання спільного ресурсу – числа *L*.
- *CopyME* – копіювання спільного ресурсу – матриці *ME*.
- *CopyMT* – копіювання спільного ресурсу – матриці *MT*.

Захищені процедури *ResourceMonitor* призначені для зміни спільного ресурсу:

- *WriteME* – запис матриці *ME* в монітор.
- *WriteMT* – запис матриці *MT* в монітор.
- *WriteL* – запис числа *L* в монітор.
- *CalculateE* – виконання операції  $e = \min(e, e_j)$ .

Захищені процедури монітора *SynchronizeMonitor* призначені для посилання сигналу:

- *SignalInput* – сигнал про закінчення вводу.
- *SignalCalculate2* – сигнал про закінчення обчислення  $e$ .
- *SignalCalculate3* – сигнал про закінчення обчислення  $MA_H$ .

Захищені входи монітора *SynchronizeMonitor* призначені для очікування на завершення подій:

- *WaitInput* – очікування завершення вводу.
- *WaitCalculate2* – очікування завершення обчислення  $e$ .
- *WaitCalculate3* – очікування завершення обчислення  $MA_H$ .

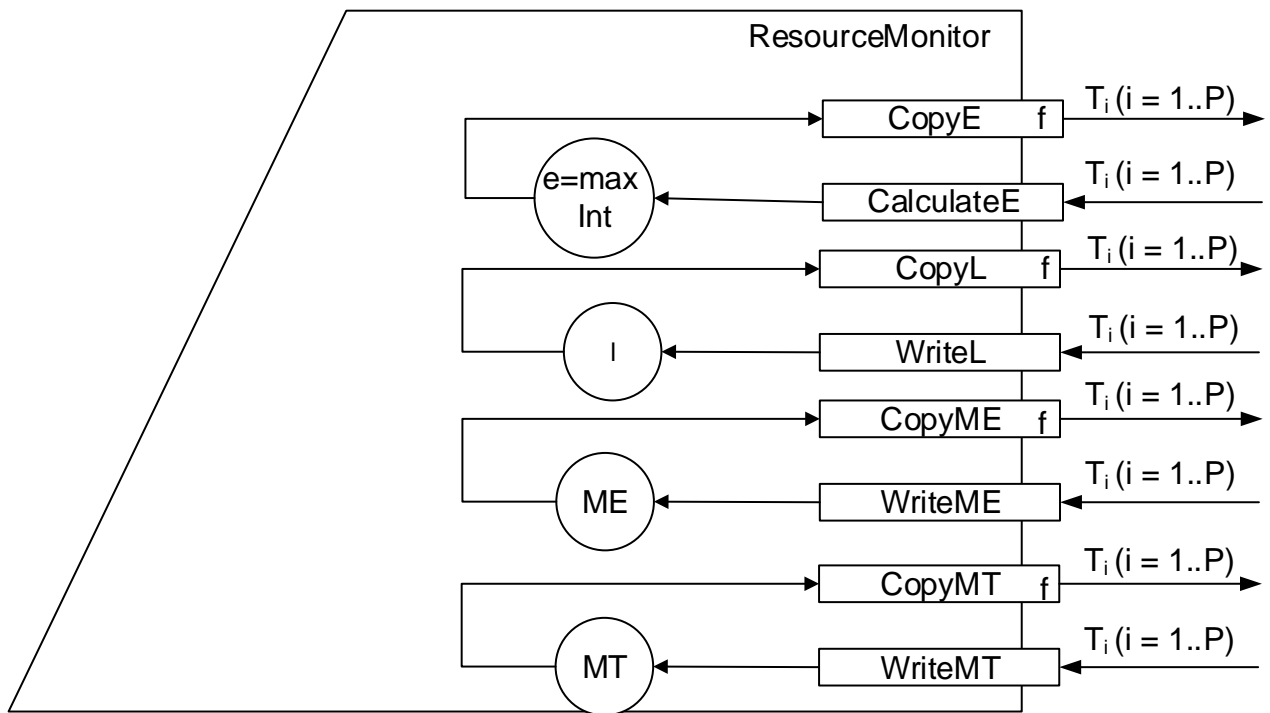


Рис. 2.2. Структурна схема монітора для вирішення задачі взаємного виключення для ПРГ1

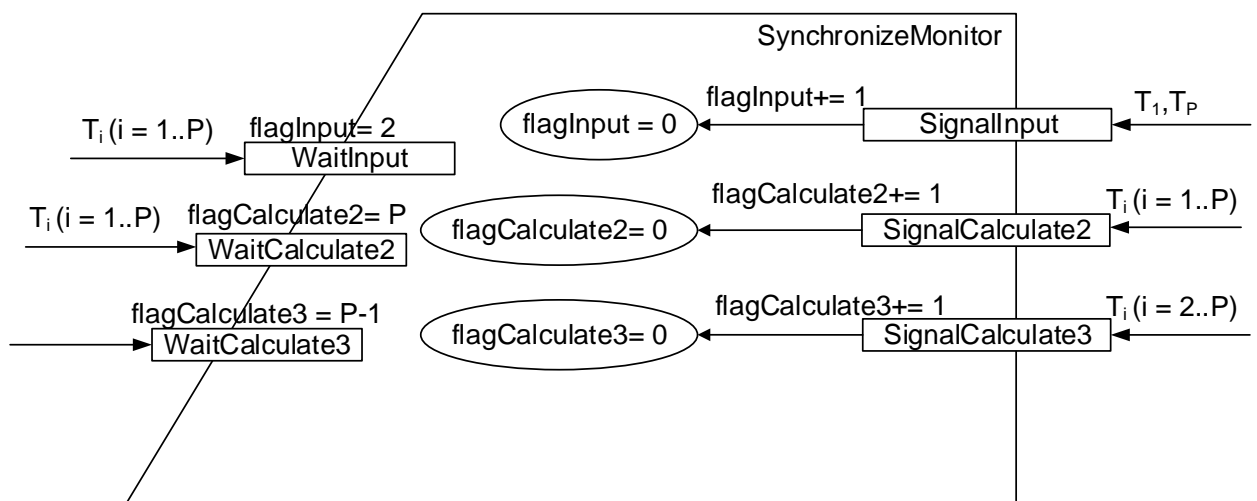


Рис. 2.3. Структурна схема монітора для вирішення задачі синхронізації для ПРГ1

## 2.4. Розробка програми ПРГ1

Програма ПРГ1 згідно технічного завдання розроблена на мові програмування Ада. Для взаємодії процесів використовується концепція моніторів (захищених модулів).



Програма складається з головної процедури Prg1, яка складається з:

- Визначення констант  $N$ ,  $P$ ,  $H$ .
- Визначення типу матриці Matrix.
- Процедури введення/виведення матриці.
- Специфікації та реалізації моніторів ResourceMonitor та SynchronizeMonitor.
- Специфікацію та реалізацію задачного типу CalculateTask, що має дискримінант taskNumber – ідентифікатор (номер) процесу.
- Тип вказівника на задачу для динамічної ініціалізації задач з допомогою дискримінанта.
- Тип масиву вказівників CalculateTaskPointer на задачі для збереження всіх задач.

Лістинг розробленої програми наведено у додатку А.

Алгоритм роботи задач наведено у додатку В.

## **2.5. Тестування програми ПРГ 1**

Метою проведення тестування є оцінка коефіцієнтів прискорення і коефіцієнтів ефективності для розроблених програм при їх виконанні на реальній паралельній обчислювальній системі. Для визначення вищевказаних коефіцієнтів проведений ряд експериментів із різними розмірностями операндів ( $N = 1200, 1500, 1800, 2100$ ) і різною кількістю працюючих ядер ( $P = 1, 2, 3, 4$ ).

Для виміру часу використовується стандартна процедура Clock мови програмування Ада, що повертає поточний час системи. Різниця між часом системи після завершення обчислення і часом при запуску програми дає час виконання обчислення, що вимірюється в секундах.

Для оцінки ефективності програми використовуються коефіцієнти прискорення та ефективності.

Коефіцієнт прискорення  $K_n$  показує скорочення часу виконання паралельної програми в паралельній системі з  $P$  процесорами  $T_p$  в порівнянні з часом виконання послідовної програми в однопроцесорній системі  $T_1$ :

$$K_n = T_1 / T_p$$

Коефіцієнт ефективності  $K_e$  застосування комп'ютерної системи показує ступінь використання  $P$  процесорів системи:

$$K_e = T_1 / (T_p \cdot P) \cdot 100\% = K_n / P \cdot 100\%$$

Результати тестування і проведених досліджень ефективності розробленої програми наведено в табл. 2.1-2.3.

Таблиця 2.1. Час виконання програми ПРГ1

N	T1	T2	T3	T4
1200	30	15	12	8
1500	59	31	24	16
1800	102	52	40	28
2100	168	89	68	49

На основі даних із табл. 2.1 виконано розрахунок значень коефіцієнтів прискорення, які наведені в табл. 2.2.

Таблиця 2.2. Коефіцієнти прискорення для програми ПРГ1

N	P			
	1	2	3	4
1200	1	2	2,5	3,75
1500	1	1,903226	2,458333	3,6875
1800	1	1,961538	2,55	3,642857
2100	1	1,88764	2,470588	3,428571

Коефіцієнти ефективності табл. 2.3 обчислено за даними табл. 2.2.

Таблиця 2.3. Коефіцієнти ефективності для програми ПРГ1

N	P			
	1	2	3	4
1200	100	100	83,33333	93,75
1500	100	95,16129	81,94444	92,1875
1800	100	98,07692	85	91,07143
2100	100	94,38202	82,35294	85,71429

Використовуючи табл. 2.1-2.3 побудовано графіки зміни часу виконання обчислень та коефіцієнтів прискорення і ефективності в залежності від  $N$  і  $P$  (рис. 2.4 – 2.6).

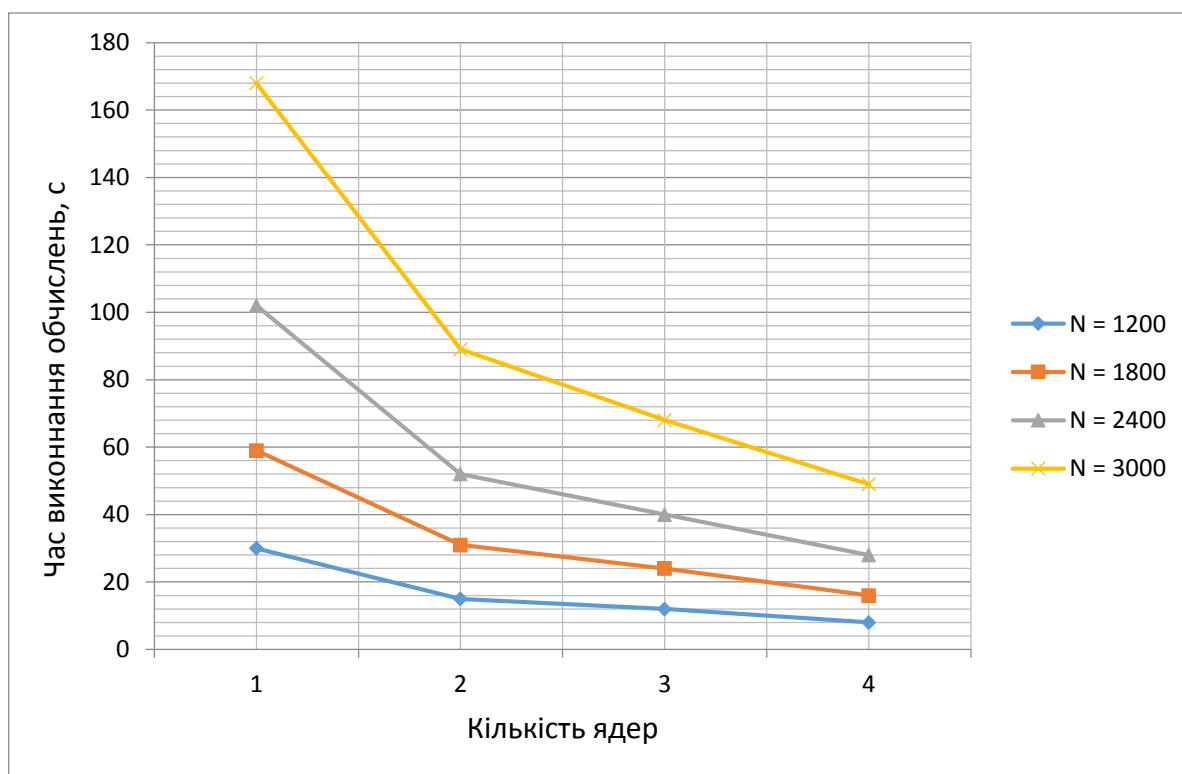


Рис. 2.4. Графік зміни часу виконання обчислень програми ПРГ1 в залежності від кількості ядер

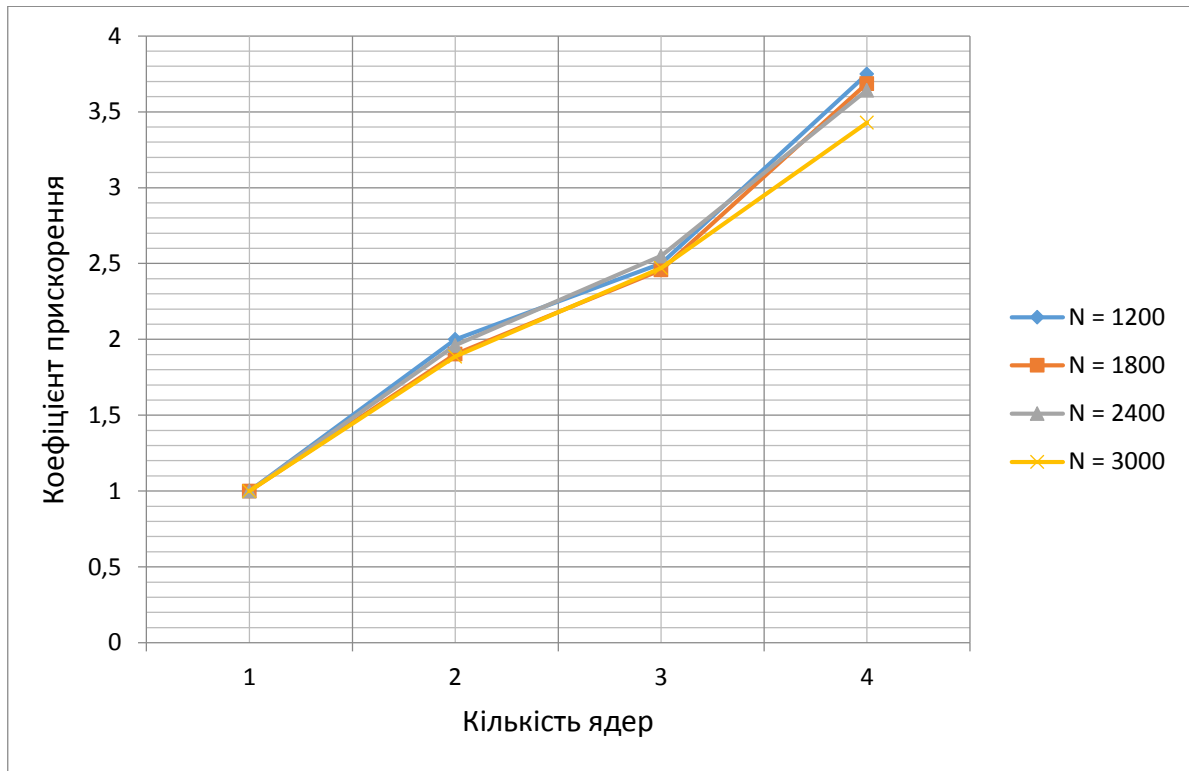


Рис. 2.5. Графік зміни коефіцієнту прискорення програми ПРГ1 в залежності від кількості ядер

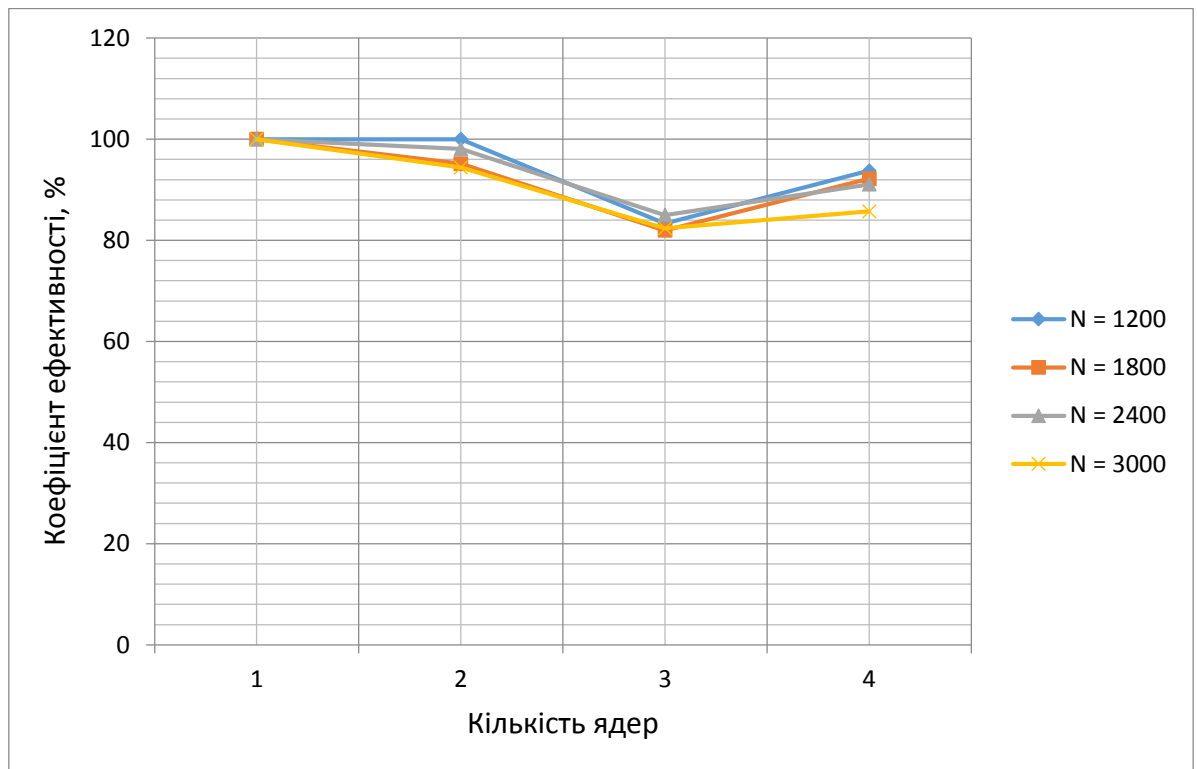


Рис. 2.6. Графік зміни коефіцієнту ефективності програми ПРГ1 в залежності від кількості ядер

## 2.6. Висновки до Розділу 2

Виконано розробку програми ПРГ1 для ПКС із СП. При розробці використовувалась мова програмування Ада. Для взаємодії процесів використовувався механізм монітора.

На основі даних тестування програми ПРГ1 заповнено таблиці 2.1-2.3 та побудовано відповідні графіки на яких показано наступні результати:

- Використання багатоядерної ПКС забезпечує скорочення часу обчислення заданої математичної задачі.
- Максимальне значення  $K_n = 3,75$  забезпечує ПКС з  $P = 4$  та  $N=1200$ .
- Мінімальне значення  $K_n = 1,88764$  забезпечує ПКС з  $P = 2$  та  $N = 2100$ .
- По графіку з рис. 2.4 видно, що коефіцієнт прискорення збільшується пропорційно до кількості використовуваних ядер.
- Коефіцієнти ефективності коливаються в межах від 81,9-100%, це свідчить про високу ефективність розробленого паралельного алгоритму.
- Максимальне значення  $K_e = 100\%$  для ПКС з  $P = 2$  та  $N = 1200$ ;
- Мінімальне значення  $K_e = 81,9\%$  для ПКС з  $P = 3$  та  $N = 1500$ .
- Нерівномірність коефіцієнтів ефективності на графіку з рис. 2.5 пояснюється похибкою вимірювання часу. Так, похибка виміру часу для  $N = 1200$  найбільша, тому графік коефіцієнта ефективності при такій розмірності може бути відносно неточним.

Оцінивши значення  $K_n$  та  $K_e$  можна зробити висновок про доцільність використання ресурсів ПКС для розв'язання задач такого типу.

### РОЗДІЛ 3. РОЗРОБКА ПРОГРАМИ ПРГ2 ДЛЯ ПКС ЛП

Математична задача така ж, як і в Розділі 2.

$$MA = \min(MO) * MK + l * ME * (MT * MM)$$

Структура ПКС ЛП зображена на рис. 3.1. Ввід і видів даних відбувається в процесах з номерами 1 і Q.

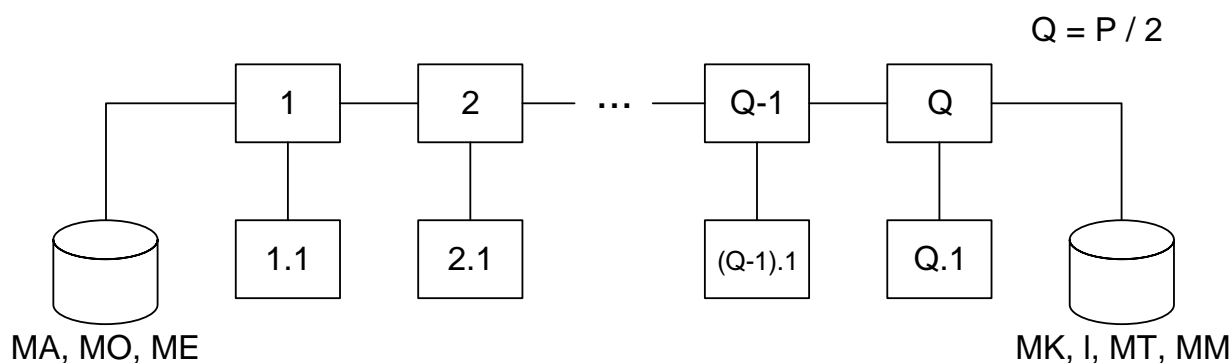


Рис. 3.1 Структура ПКС із локальною пам'яттю

#### 3.1. Розробка паралельного математичного алгоритму

Відповідно до технічного завдання необхідно виконати розробку паралельного алгоритму. Паралельний математичний алгоритм можна подати у вигляді наступних етапів:

1)  $e_i = \min(MO_H), i = 1..Q; 1.1..Q.1.$

2)  $e = \min(e, e_i), 3P: e.$

3)  $MA_H = e * MK_H + l * ME * (MT * MM_H), 3P: e, ME, MT.$

Пояснення до використовуваних констант:

- $N$  – розмірність векторів і матриць;
- $P$  – кількість ядер;
- $H = \frac{N}{P}$ ;
- $Q = \frac{P}{2}$ .

#### 3.2. Розробка алгоритмів процесів

Задача  $T_1$ :

- 1) Введення  $ME, MO$ .

- 2) Передати ME, MO<sub>N-2H</sub> в T<sub>2</sub>.
- 3) Прийняти l, MK<sub>2H</sub>, MM<sub>2H</sub>, ME, MT від T<sub>2</sub>.
- 4) Передати l, MK<sub>H</sub>, MM<sub>H</sub>, ME, MT, ME, MO<sub>H</sub> в T<sub>1.1</sub>.
- 5) Обчислення1

$$e_1 = \min(MO_H)$$

- 6) Прийняти e<sub>1.1</sub> від T<sub>1.1</sub>.
- 7) Обчислення2

$$e_1 = \min(e_{1.1}, e_1)$$

- 8) Передати e<sub>1</sub> в T<sub>2</sub>.
- 9) Прийняти e<sub>1</sub> від T<sub>2</sub>.
- 10) Передати e<sub>1</sub> в T<sub>1.1</sub>.
- 11) Обчислення3

$$MA_H = e_1 * MK_H + l_1 * ME_1 * (MT_1 * MM_H)$$

- 12) Прийняти MA<sub>H</sub> від T<sub>1.1</sub>.
- 13) Прийняти MA<sub>N-2H</sub> від T<sub>2</sub>.
- 14) Виведення MA.

**Задача T<sub>Q</sub>, Q = (P/4)+1, Q ≠ P/2:**

- 1) Прийняти l, MK<sub>2HQ</sub>, MT, MM<sub>2HQ</sub> від T<sub>Q+1</sub>.
- 2) Передати l, MK<sub>2H(Q-1)</sub>, MT, MM<sub>2H(Q-1)</sub> в T<sub>Q-1</sub>.
- 3) Прийняти ME, MO<sub>(N-2Q)H</sub> від T<sub>Q-1</sub>.
- 4) Передати ME, MO<sub>(N-2Q-2)H</sub> в T<sub>Q+1</sub>.
- 5) Передати l, MK<sub>H</sub>, MM<sub>H</sub>, ME, MT, ME, MO<sub>H</sub> в T<sub>Q.1</sub>.
- 6) Обчислення1

$$e_Q = \min(MO_H)$$

- 7) Прийняти e<sub>Q.1</sub> від T<sub>Q.1</sub>.
- 8) Прийняти e<sub>Q+1</sub> від T<sub>Q+1</sub>.
- 9) Прийняти e<sub>Q-1</sub> від T<sub>Q-1</sub>.
- 10) Обчислення2

$$e_Q = \min(e_Q, e_{Q.1}, e_{Q+1}, e_{Q-1})$$

- 11) Передати  $e_1$  в  $T_{Q+1}$ .
- 12) Передати  $e_1$  в  $T_{Q-1}$ .
- 13) Передати  $e_1$  в  $T_{Q,1}$ .
- 14) Обчислення3

$$MA_H = e_Q * MK_H + l_Q * ME_Q * (MT_Q * MM_H)$$

- 15) Прийняти  $MA_H$  від  $T_{Q,1}$ .
- 16) Прийняти  $MA_{2H(Q-1)}$  від  $T_{Q+1}$ .
- 17) Передати  $MA_{2HQ}$  в  $T_{Q-1}$ .

**Задача  $T_Q, Q = P/2, Q \neq 2$ :**

- 1) Введення  $l, MK, MT, MM$ .
- 2) Передати  $l, MK_{N-2H}, MT, MM_{N-2H}$  в  $T_{Q-1}$ .
- 3) Прийняти  $ME, MO_{2H}$  від  $T_{Q-1}$ .
- 4) Передати  $l, MK_H, MM_H, ME, MT, ME, MO_H$  в  $T_{Q,1}$ .
- 5) Обчислення1

$$e_Q = \min(MO_H)$$

- 6) Прийняти  $e_{Q,1}$  від  $T_{Q,1}$ .
- 7) Обчислення2

$$e_Q = \min(e_Q, e_{1,1})$$

- 8) Передати  $e_Q$  в  $T_{Q-1}$ .
- 9) Прийняти  $e_Q$  від  $T_{Q-1}$ .
- 10) Передати  $e_Q$  в  $T_{Q,1}$ .
- 11) Обчислення3

$$MA_H = e_Q * MK_H + l_Q * ME_Q * (MT_Q * MM_H)$$

- 12) Прийняти  $MA_H$  від  $T_{Q,1}$ .
- 13) Передати  $MA_{2H}$  в  $T_{Q-1}$ .

**Задача  $T_Q, Q = P/2, Q = 2$ :**

- 1) Введення  $l, MK, MT, MM$ .
- 2) Передати  $l, MK_{N-2H}, MT, MM_{N-2H}$  в  $T_{Q-1}$ .



- 3) Прийняти  $ME, MO_{2H}$  від  $T_{Q-1}$ .
- 4) Передати  $l, MK_H, MM_H, ME, MT, ME, MO_H$  в  $T_{Q+1}$ .
- 5) Обчислення1

$$e_Q = \min(MO_H)$$

- 6) Прийняти  $e_{Q+1}$  від  $T_{Q+1}$ .
- 7) Прийняти  $e_{Q-1}$  від  $T_{Q-1}$ .
- 8) Обчислення2

$$e_Q = \min(e_Q, e_{Q+1}, e_{Q-1})$$

- 9) Передати  $e_Q$  в  $T_{Q-1}$ .
- 10) Передати  $e_Q$  в  $T_{Q+1}$ .
- 11) Обчислення3

$$MA_H = e_Q * MK_H + l_Q * ME_Q * (MT_Q * MM_H)$$

- 12) Прийняти  $MA_H$  від  $T_{Q+1}$ .
- 13) Передати  $MA_{2H}$  в  $T_{Q-1}$ .

**Задача  $T_Q, Q < (P/4)+1, Q \neq 1$ :**

- 1) Прийняти  $ME, MO_{(N-2Q)H}$  від  $T_{Q-1}$ .
- 2) Передати  $ME, MO_{(N-2Q-2)H}$  в  $T_{Q+1}$ .
- 3) Прийняти  $l, MK_{2HQ}, MT, MM_{2HQ}$  від  $T_{Q+1}$ .
- 4) Передати  $l, MK_{2H(Q-1)}, MT, MM_{2H(Q-1)}$  в  $T_{Q-1}$ .
- 5) Передати  $l, MK_H, MM_H, ME, MT, ME, MO_H$  в  $T_{Q+1}$ .
- 6) Обчислення1

$$e_Q = \min(MO_H)$$

- 7) Прийняти  $e_{Q+1}$  від  $T_{Q+1}$ .
- 8) Прийняти  $e_{Q-1}$  від  $T_{Q-1}$ .
- 9) Обчислення2

$$e_Q = \min(e_Q, e_{Q+1}, e_{Q-1})$$

- 10) Передати  $e_Q$  в  $T_{Q+1}$ .
- 11) Прийняти  $e_Q$  від  $T_{Q+1}$ .

- 12) Передати  $e_Q$  в  $T_{Q-1}$ .
- 13) Передати  $e_Q$  в  $T_{Q+1}$ .
- 14) Обчислення3

$$MA_H = e_Q * MK_H + l_Q * ME_Q * (MT_Q * MM_H)$$

- 15) Прийняти  $MA_H$  від  $T_{Q+1}$ .
- 16) Прийняти  $MA_{2H(Q-1)}$  від  $T_{Q+1}$ .
- 17) Передати  $MA_{2HQ}$  в  $T_{Q-1}$ .

**Задача  $T_Q, Q > (P/4)+1, Q \neq P/2$ :**

- 1) Прийняти  $l, MK_{2HQ}, MT, MM_{2HQ}$  від  $T_{Q+1}$ .
- 2) Передати  $l, MK_{2H(Q-1)}, MT, MM_{2H(Q-1)}$  в  $T_{Q-1}$ .
- 3) Прийняти  $ME, MO_{(N-2Q)H}$  від  $T_{Q-1}$ .
- 4) Передати  $ME, MO_{(N-2Q-2)H}$  в  $T_{Q+1}$ .
- 5) Передати  $l, MK_H, MM_H, ME, MT, ME, MO_H$  в  $T_{Q+1}$ .
- 6) Обчислення1

$$e_Q = \min(MO_H)$$

- 7) Прийняти  $e_{Q+1}$  від  $T_{Q+1}$ .
- 8) Прийняти  $e_{Q-1}$  від  $T_{Q+1}$ .
- 9) Обчислення2

$$e_Q = \min(e_Q, e_{Q-1}, e_{Q+1})$$

- 10) Передати  $e_Q$  в  $T_{Q-1}$ .
- 11) Прийняти  $e_Q$  від  $T_{Q-1}$ .
- 12) Передати  $e_Q$  в  $T_{Q+1}$ .
- 13) Передати  $e_Q$  в  $T_{Q+1}$ .
- 14) Обчислення3

$$MA_H = e_Q * MK_H + l_Q * ME_Q * (MT_Q * MM_H)$$

- 15) Прийняти  $MA_H$  від  $T_{Q+1}$ .
- 16) Прийняти  $MA_{2H(Q-1)}$  від  $T_{Q+1}$ .
- 17) Передати  $MA_{2HQ}$  в  $T_{Q-1}$ .

### Задача $T_{Q,1}$ , $Q = 1..P/2$ :

- 1) Прийняти  $l$ ,  $MK_H$ ,  $MM_H$ ,  $ME$ ,  $MT$ ,  $ME$ ,  $MO_H$  від  $T_Q$ .
- 2) Обчислення1

$$e_Q = \min(MO_H)$$

- 3) Передати  $e_Q$  в  $T_Q$ .
- 4) Прийняти  $e_Q$  від  $T_Q$ .
- 5) Обчислення3

$$MA_H = e_Q * MK_H + l_Q * ME_Q * (MT_Q * MM_H)$$

- 6) Передати  $MA_H$  в  $T_Q$ .

### 3.3. Розробка схеми взаємодії процесів

На основі алгоритму для всіх типів задач розроблено структурні схеми взаємодії задач (рис. 3.2). За допомогою цих схем можна наочно спостерігати як саме відбувається пересилка даних. Таке графічне зображення дозволяє простіше зрозуміти механізм пересилки повідомлень та взаємодії процесів.

Усі задачі логічно розбиваються на декілька типів:

- 1) Перша задача  $T_1$  («крайня зліва»), в якій вводяться дані ( $MA$ ,  $ME$ ,  $MO$ ).
- 2) Остання задача  $T_Q$  («крайня зправа»), в якій вводяться дані ( $B$ ,  $MB$ ,  $MO$ ).
- 3) Середня «проміжна» задача  $T_Q$ ,  $Q = (P/4)+1$ , в якій збирається результат обчислення  $a, b$  і яка слугує посередником при початковій пересилці даних (щоб не виникало взаємного блокування при рандеву).
- 4) «Проміжні» задачі ліворуч від середньої -  $T_Q$ ,  $Q < (P/4)+1$ ,  $Q \neq 1$ .
- 5) «Проміжні» задачі праворуч від середньої -  $T_Q$ ,  $Q > (P/4)+1$ ,  $Q \neq P/2$ .
- 6) «Нижні» задачі  $T_{Q,1}$ ,  $Q = 1..P/2$ .

Таке логічне розбиття на типи гарантує, що пересилка даних та збір і розповсюдження проміжних результатів буде виконуватися за кількість тактів, що дорівнює діаметру системи. Тобто, найбільш оптимально.

Передача повідомлень між процесами та синхронізація відбувається за допомогою викликів таких функцій:

MPI\_Recv – функція блокує процес поки не будуть отриманні всі данні.

MPI\_Send – функція блокує процес поки не будуть передані всі дані.

MPI\_Isend – функція викликає передачу даних без блокування процесу.

MPI\_Wait – очікування виконання умови (наприклад, очікування завершення передачі даних).

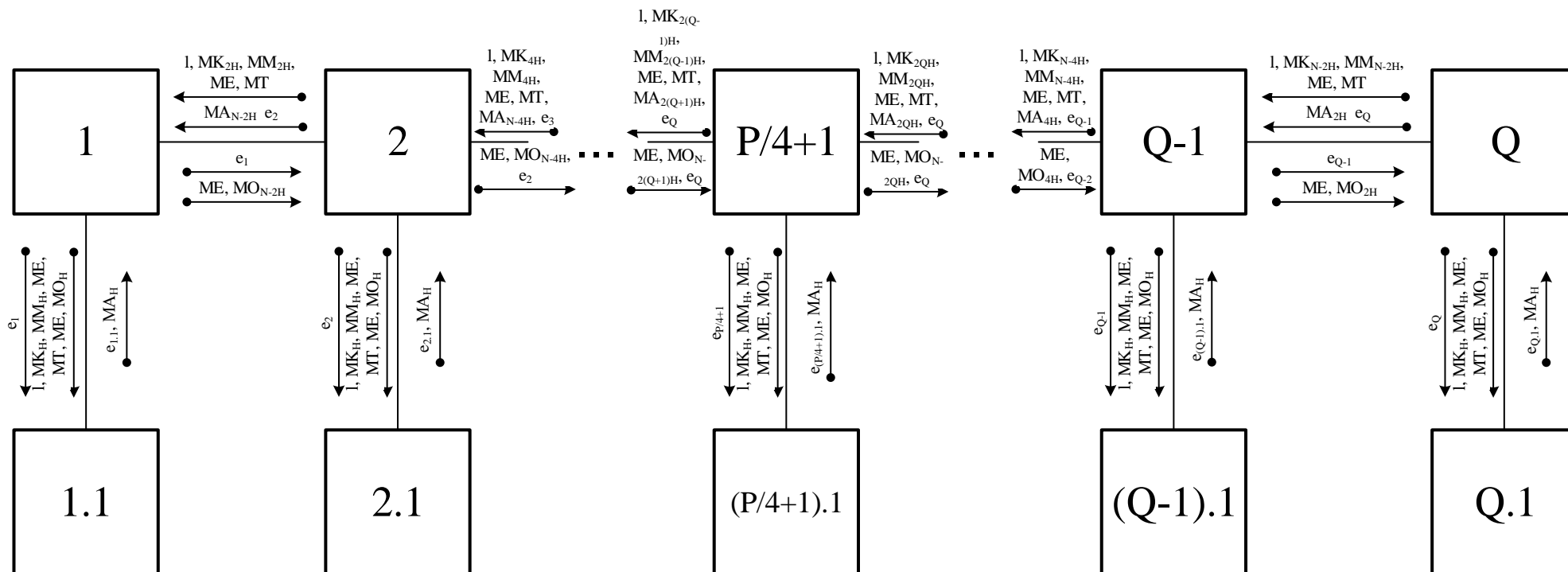


Рис. 3.2. Структурна схема взаємодії процесів ПРГ2 для ПКС ЛП

### 3.4. Розробка програми ПРГ2

Програма ПРГ2 згідно технічного завдання розроблена на мові програмування C++ із використанням засобів бібліотеки MPI.

Програма складається з файлів Matrix.h, Matrix.cpp, CourseWork(mpi).cpp. У файлі Matrix.h розміщується об'явлення класу Matrix та його конструкторів і функцій. У файлі Matrix.cpp розміщується визначення конструкторів і функцій класу Matrix:

getRows() – повертає кількість рядків у матриці;

getColumns() – повертає кількість стовпців у матриці;

getPtrToArray() – вказівник на перший елемент матриці;

input() – заповнення елементів матриці значенням константи FILLER;

copy(Matrix& copyMatrix) – копіює матрицю;

mult(Matrix& res, Matrix& multMatr) – множення матриць;

getMin() – пошук мінімуму по всій матриці;

getMin(int from, int to) – пошук мінімуму в заданому участку;

delExcessive(int from, int to) – видалення усіх елементів, окрім діапазону, вказаному в аргументах. Це необхідно для зменшення використання пам'яті при пересилках матриць. Після закінчення передачі матриці, виконується видалення зайвих елементів.

merge(Matrix& MAh) – об'єднання двох матриць.

У класі CourseWork(mpi).cpp визначений метод main(), в якому запускаються процеси, а також константи  $N$ ,  $P$ ,  $H$ , допоміжні функції та процеси.

Лістинг розробленої програми наведено у додатку Б.

Алгоритм роботи задач наведено у додатку В.

### 3.5. Тестування програми ПРГ 2

Метою проведення тестування є оцінка коефіцієнтів прискорення і коефіцієнтів ефективності для розроблених програм при їх виконанні на реальній паралельній обчислювальній системі. Для визначення вищевказаних коефіцієнтів проведений ряд експериментів із різними розмірностями операндів ( $N = 1200, 1500, 1800, 2100$ ) і різною кількістю працюючих ядер ( $P = 1, 2, 3, 4$ ).

Для виміру часу використовується процедура `clock()` бібліотеки `time.h` мови програмування C++, що повертає поточний час системи. Різниця між часом системи після завершення обчислення і часом при запуску програми дає час виконання обчислення, що вимірюється в мілісекундах.

Для оцінки ефективності програми використовуються коефіцієнти прискорення та ефективності.

Коефіцієнт прискорення  $K_n$  показує скорочення часу виконання паралельної програми в паралельній системі з  $P$  процесорами  $T_p$  в порівнянні з часом виконання послідовної програми в однопроцесорній системі  $T_1$ :

$$K_n = T_1 / T_p$$

Коефіцієнт ефективності  $K_e$  застосування комп'ютерної системи показує ступінь використання  $P$  процесорів системи:

$$K_e = T_1 / (T_p \cdot P) \cdot 100\% = K_n / P \cdot 100\%$$

Результати тестування і проведених досліджень ефективності розробленої програми наведено в табл. 3.1-3.3.

Таблиця 3.1. Час виконання програми ПРГ 2

N	T1	T2	T3	T4
1200	15	8	6	4
1500	29	15	11	9
1800	50	26	19	15
2100	85	44	32	24

На основі даних із табл. 3.1 виконано розрахунок значень коефіцієнтів прискорення, які наведені в табл. 3.2.

Таблиця 3.2. Коефіцієнти прискорення для програми ПРГ2

N	P			
	1	2	3	4
1200	1	1,875	2,5	3,75
1500	1	1,933333	2,636364	3,222222
1800	1	1,923077	2,631579	3,333333
2100	1	1,931818	2,65625	3,541667

Коефіцієнти ефективності табл. 3.3 обчислено за даними табл. 3.2.

Таблиця 3.3. Коефіцієнти ефективності для програми ПРГ2

N	P			
	1	2	3	4
1200	100	93,75	83,33333	93,75
1500	100	96,66667	87,87879	80,55556
1800	100	96,15385	87,7193	83,33333
2100	100	96,59091	88,54167	88,54167

Використовуючи табл. 3.1-3.3 побудовано графіки зміни часу виконання обчислень та коефіцієнтів прискорення і ефективності в залежності від  $N$  і  $P$  (рис. 3.3 – 3.5).



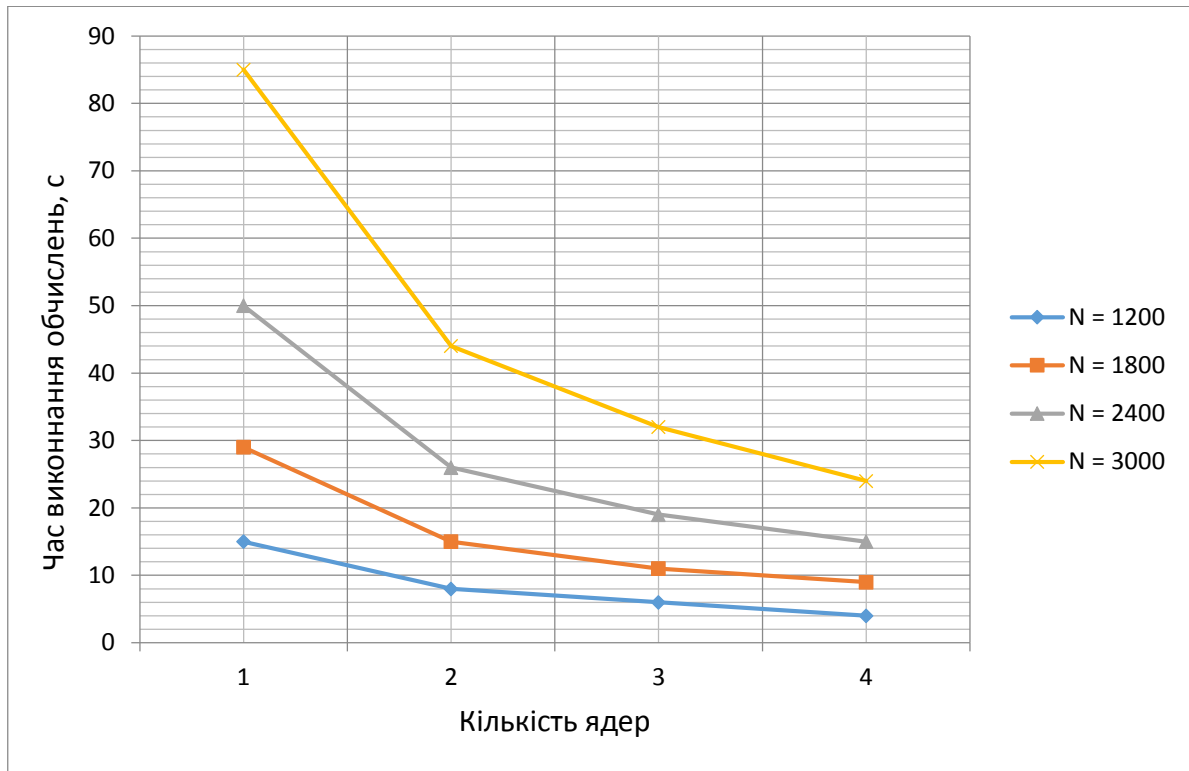


Рис. 3.3. Графік зміни часу виконання обчислень програми ПРГ1 в залежності від кількості ядер

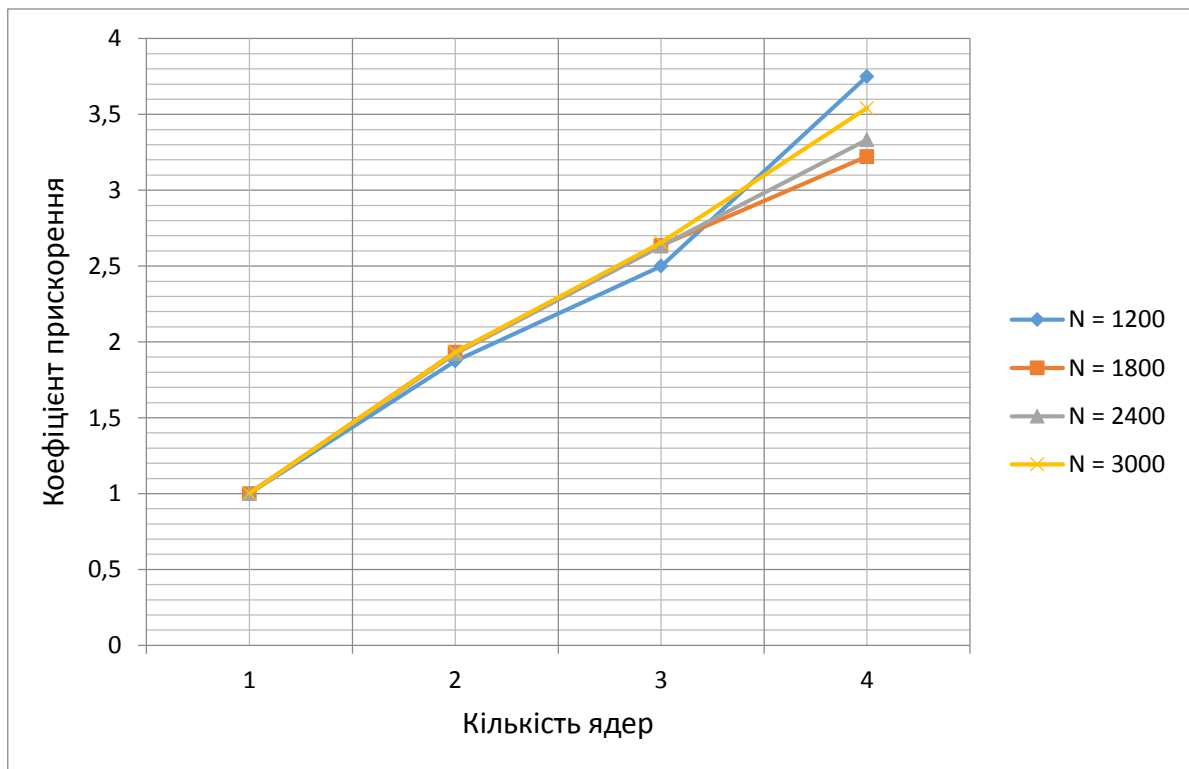


Рис. 3.4. Графік зміни коефіцієнту прискорення програми ПРГ1 в залежності від кількості ядер

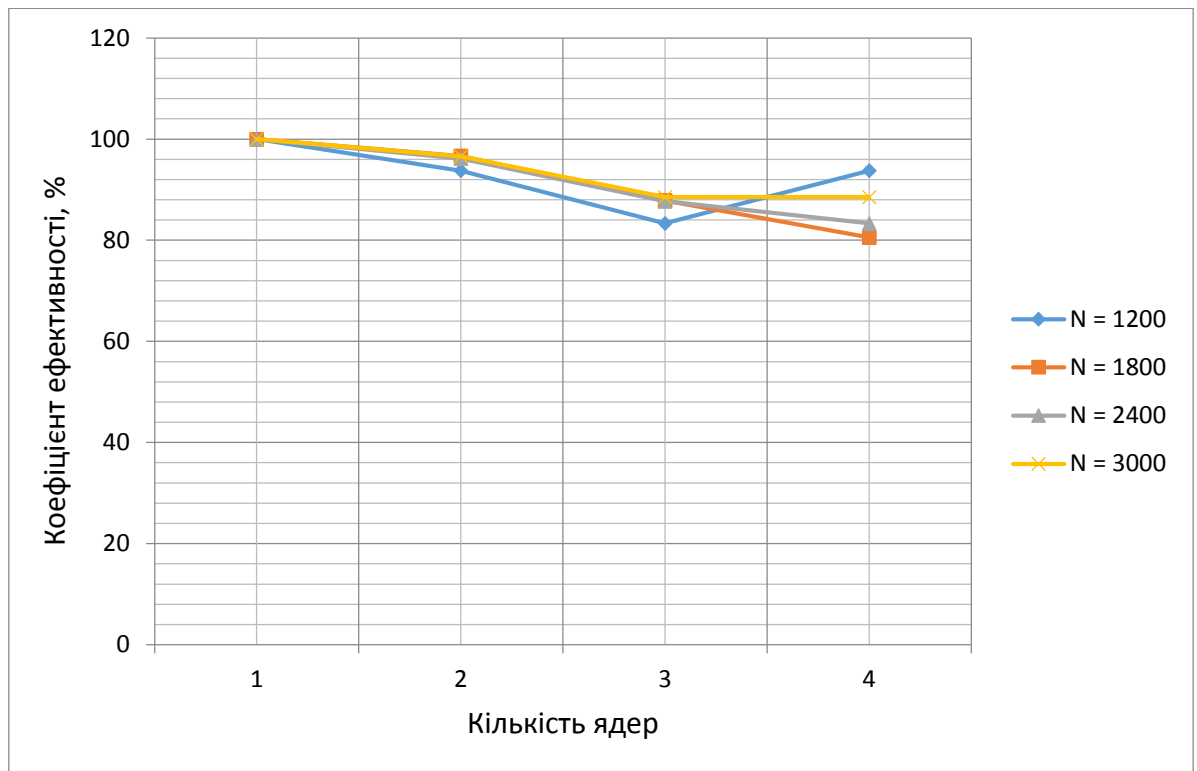


Рис. 3.5. Графік зміни коефіцієнту ефективності програми ПРГ1 в залежності від кількості ядер

### 3.6. Висновки до Розділу 3

Виконано розробку програми ПРГ2 для ПКС із ЛП. При розробці використовувалась мова програмування C++. Для передачі повідомлень між процесами використовувалася бібліотека MPI.

На основі даних тестування програми ПРГ2 заповнено таблиці 3.1-3.3 та побудовано відповідні графіки, на яких показано наступні результати:

- Використання багатоядерної ПКС забезпечує скорочення часу обчислення заданої математичної задачі.
- Максимальне значення  $K_n = 3,75$  забезпечує ПКС з  $P = 4$  та  $N=1200$ .
- Мінімальне значення  $K_n = 1,875$  забезпечує ПКС з  $P = 2$  та  $N=1200$ .
- По графіку з рис. 3.5 видно, що коефіцієнт прискорення збільшується пропорційно до кількості використовуваних ядер.
- Коефіцієнти ефективності коливаються в межах від 80-100%, це свідчить про ефективність розробленого паралельного алгоритму.
- Максимальне значення  $K_e = 96,66667\%$  для ПКС з  $P = 2$  та  $N = 1500$ ;

- Мінімальне значення  $K_e = 80,55556\%$  для ПКС з  $P = 4$  та  $N = 1500$ .

Не зважаючи на те, що пересилка даних між процесами – досить затратна операція, реалізація задачі з допомогою бібліотеки MPI демонструє досить ефективне використання ядер.

Оцінивши значення  $K_n$  та  $K_e$  можна зробити висновок про доцільність використання бібліотеки MPI для розв'язання задач такого типу.

## ОСНОВНІ РЕЗУЛЬТАТИ І ВИСНОВКИ ПО РОБОТІ

1. Проведено огляд шестиядерних процесорів компанії AMD. Розглянуто 6 поколінь архітектур багатоядерних процесорів. Серед основних закономірностей спостерігаються збільшення частоти ядер, збільшення кількості ядер, збільшення кеш-пам'яті, зниження енергоспоживання. Починаючи з покоління Bulldozer компанія AMD почала випускати процесори з модульною архітектурою (по 2 ядра в модулі). Починаючи з архітектури Piledriver з'явилася нова серія процесорів APU. В них відсутня кеш-пам'ять третього рівня, зате присутній графічний прискорювач, який займає приблизно половину площі процесора. З анонсом архітектури Piledriver також була оновлена серія процесорів APU для мобільних комп'ютерів зі зменшеними в рази (до 17 Вт) показниками енергоспоживання. Це збільшить час роботи від акумулятора та зменшить виділення тепла.

2. Розроблені алгоритми програм для багатоядерних обчислювальних систем, що вирішують поставлені математичні задачі. Вирішення задачі на ПКС із СП з допомогою механізму монітора показало кращі показники щодо прискорення та ефективності, ніж при вирішенні на ПКС із ЛП з допомогою бібліотеки MPI. Це пояснюється тим, що пересилка даних – затратна операція, оскільки передаються великі об'єми даних (при  $N = 3000$  розмір однієї матриці  $N * N$  буде рівним 36 МБ).

3. Перевагою системи зі спільною пам'яттю над системою з локальною пам'яттю є спільні ресурси, які не копіюються для кожного процесу. При виконанні програма, яка реалізує задачу в системі зі СП буде займати менше пам'яті, ніж програма, яка реалізує таку саму задачу в системі з ЛП.

4. Ще одною перевагою системи зі СП над системою з ЛП є зрозуміліший програмний код, що забезпечує більшу швидкість програмування та якісну підтримку програмного забезпечення.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Жуков І.А., Корочкін О.В. Паралельні та розподілені обчислення: Навч. посібник. – К.: Корнійчук, 2005. – 226 с. – ISBN 996-7599-36-1.
2. Our History [Електронний ресурс] .— Режим доступу:  
<http://www.amd.com/en-us/who-we-are/corporate-information/history> — дата звернення: 20.05.2014.— Назва з екрану.
3. Athlon 64 X2 [Електронний ресурс] .— Режим доступу:  
[http://en.wikipedia.org/wiki/Athlon\\_X2#Athlon\\_X2](http://en.wikipedia.org/wiki/Athlon_X2#Athlon_X2) — дата звернення: 20.05.2014.— Назва з екрану.
4. AMD Phenom [Електронний ресурс] .— Режим доступу:  
[http://en.wikipedia.org/wiki/Phenom\\_\(processor\)](http://en.wikipedia.org/wiki/Phenom_(processor)) — дата звернення: 20.05.2014.— Назва з екрану.
5. Phenom II [Електронний ресурс] .— Режим доступу:  
[http://en.wikipedia.org/wiki/Phenom\\_II](http://en.wikipedia.org/wiki/Phenom_II) — дата звернення: 20.05.2014.— Назва з екрану.
6. Bulldozer (microarchitecture) [Електронний ресурс] .— Режим доступу:  
[http://en.wikipedia.org/wiki/Bulldozer\\_\(microarchitecture\)](http://en.wikipedia.org/wiki/Bulldozer_(microarchitecture)) — дата звернення: 20.05.2014.— Назва з екрану.
7. Piledriver (microarchitecture) [Електронний ресурс] .— Режим доступу:  
[http://en.wikipedia.org/wiki/Piledriver\\_\(microarchitecture\)](http://en.wikipedia.org/wiki/Piledriver_(microarchitecture)) — дата звернення: 20.05.2014.— Назва з екрану.
8. Steamroller (microarchitecture) [Електронний ресурс] .— Режим доступу:  
[http://en.wikipedia.org/wiki/Steamroller\\_\(microarchitecture\)](http://en.wikipedia.org/wiki/Steamroller_(microarchitecture)) — дата звернення: 20.05.2014.— Назва з екрану.
9. Excavator (microarchitecture) [Електронний ресурс] .— Режим доступу:  
[http://en.wikipedia.org/wiki/Excavator\\_\(microarchitecture\)](http://en.wikipedia.org/wiki/Excavator_(microarchitecture)) — дата звернення: 20.05.2014.— Назва з екрану.