

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

---

---

« \_\_\_\_\_ » \_\_\_\_\_ 2021р.

**МАГІСТЕРСЬКА РОБОТА**

НА ТЕМУ:

**Моделі та методи для дослідження ефективності віртуальної  
багатопоточності в ядрах сучасних процесорів**

---

---

Освітньо-кваліфікаційний рівень “Магістр”  
Спеціальність 123 - “Комп’ютерна інженерія

Науковий керівник роботи:

\_\_\_\_\_

Д.О. Недзельський

\_\_\_\_\_

(ініціали, прізвище)

Консультант з охорони праці:

\_\_\_\_\_

Я.О. Критська

\_\_\_\_\_

(ініціали, прізвище)

Студент:

\_\_\_\_\_

А.В. Лісовін

\_\_\_\_\_

(ініціали, прізвище)

Група:

КІ-19дм

Сєвєродонецьк 2021

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки

Кафедра Комп'ютерних наук та інженерії

Освітньо-кваліфікаційний рівень магістр

Напрямок підготовки \_\_\_\_\_

(шифр і назва)

Спеціальність 123 - "Комп'ютерна інженерія"

(шифр і назва)

**ЗАТВЕРДЖУЮ:**

« \_\_\_\_\_ » \_\_\_\_\_ 2021р.

**З А В Д А Н Н Я  
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ**

Лісовіну Анатолію Валерійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Моделі та методи для дослідження ефективності  
віртуальної багатопоточності в ядрах сучасних процесорів

керівник проекту (роботи) Недзельський Дмитро Олександрович

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від "11" 10 2020 р. № 136/15.15

2. Строк подання студентом роботи 10.01.2021

3. Вихідні дані до

роботи Матеріали науково-дослідної практики,

наукова-методична література, дані інтернет-мережі

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Методи підвищення продуктивності ядер сучасних процесорів, дослідження програми «Множення матриць», дослідження реальної ефективності ядер процесорів при виконанні програми «Рішення др в приватних похідних», дослідження реальної ефективності ядер процесорів при виконанні програми «ШПФ», охрона праці, висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)  
Електронні плакати

## 6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці та безпека в надзвичайних ситуаціях	Критська Я.О. ст. викл. кафедри КНІ		

7. Дата видачі завдання 14.10.2020

Керівник

\_\_\_\_\_ (підпис)

Завдання прийняв до виконання

\_\_\_\_\_ (підпис)

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту ( роботи )	Примітка
1	Розробка технічного завдання	02.09.2020-15.09.2020	
2	Аналіз літературних джерел	16.09.2020-22.09.2020	
3	Аналіз технічних засобів	23.09.2020-25.09.2020	
4	Розробка методу	26.09.2020-10.10.2020	
4	Програмна реалізація	11.10.2020-06.11.2020	
5	Аналіз результатів дослідження	07.11.2020-25.11.2020	
6	Розробка частини проекту "Охорона праці та безпеки в надзвичайних ситуаціях"	26.11.2020-1.12.2020	
7	Оформлення пояснювальної записки, автореферату та презентації	2.12.2020-09.01.2021	

Студент

\_\_\_\_\_ (підпис)

Лісовін А.В.

\_\_\_\_\_ (прізвище та ініціали)

Науковий керівник

\_\_\_\_\_ (підпис)

Недзельський Д.О.

\_\_\_\_\_ (прізвище та ініціали)

## АНОТАЦІЯ

**Лісовін А.В. Моделі та методи для дослідження ефективності віртуальної багатопоточності в ядрах сучасних процесорів.**

Виконані огляди методів збільшення продуктивності ядер сучасних процесорів та літератури по дослідженню теоретичної і реальної ефективності ядер процесорів без та з врахуванням «перешкод». Розглянуто ефективності функціонування конвеєрних ядер процесора з універсальним функціональним пристроєм та з спеціалізованими функціональними пристроями без впливу «перешкод», та з врахуванням впливу «перешкод». Розроблено методику проведення досліджень комплексного впливу «перешкод» (інформаційних залежностей, операцій редукції, команд переходів) на ефективність ядер процесорів на прикладі широко поширених і програм, таких як «Множення матриць», «Рішення диференціальних рівнянь в приватних похідних методом сіток», «Швидке перетворення Фур'є». Досліджено функціонування розробленої моделі ядра процесора на прикладах цих програм. Отримані вирази для коефіцієнтів використання окремих підсистем в залежності від параметрів ядра і виконуваних програм.

**Ключові слова:** Віртуальна багатопоточність, ефективність, коефіцієнт використання, ядро процесора.

## ABSTRACT

**Lisovin AV Models and methods for studying the efficiency of virtual multithreading in the cores of modern processors.**

Reviews of methods for increasing the performance of modern processor cores and the literature on the study of theoretical and real efficiency of processor cores without and taking into account "interference". The efficiency of operation of processor cores of the processor with a universal functional device and with specialized functional devices without the influence of "interference" and taking into account the influence of "interference" is considered. A method for conducting research on the complex influence of "interference" (information dependencies, reduction operations, transition commands) on the efficiency of processor cores on the example of common and programs such as "Matrix multiplication", "Solution of differential equations in partial derivatives by grids", "Fast Fourier transform". The functioning of the developed model of the processor core is studied on the examples of these programs. The resulting expressions for the utilization factors of individual subsystems depending on the parameters of the kernel and executable programs.

**Keywords:** Virtual multithreading, efficiency, utilization factor, processor core.

## ЗМІСТ

<b>ВСТУП</b> .....	<b>6</b>
<b>1 МЕТОДИ ПІДВИЩЕННЯ ПРОДУКТИВНОСТІ ЯДЕР СУЧАСНИХ ПРОЦЕСОРІВ</b> .....	<b>7</b>
1.1 Суть та передумови віртуальної багатопоточності.....	7
1.1.1 Види структур багатопоточної віртуальності.....	7
1.2 Огляд літератури з дослідження реальної ефективності ядр процесорів з урахуванням «перешкод» .....	8
1.2.1 Види «перешкод» .....	8
1.2.2 Вплив оперативної пам'яті.....	8
1.2.3 Вирішення проблем зі структурними конфліктами, інформаційними залежностями, операціями редукції .....	9
1.3 Вплив конфлікту по керуванню .....	9
1.4 Ефективність функціонування комп'ютера з одноядерним процесором з універсальним функціональним пристроєм при відсутності перешкод .....	11
1.5 Ефективність функціонування конвеєрного ядра процесора зі спеціалізованими функціональними пристроями .....	11
1.6 Комплексний вплив «перешкод» у вигляді інформаційних залежностей, операцій редукції, команд переходів та звернення до ОП на ефективність ядр процесорів .....	12
1.7 Методика проведення дослідження .....	13
1.8 Модель ядра процесора при виконанні програми .....	14
1.9 Особливості підсистеми виконання команд, що складається з кеш-пам'яті даних першого рівня L1D, пристроїв Множення і Складання в ядрах сучасних процесорів... ..	17
<b>2 ДОСЛІДЖЕННЯ ВИКОНАННЯ ПРОГРАМИ «МНОЖЕННЯ МАТРИЦЬ»</b> .....	<b>18</b>
2.1 Про програму «Множення матриць».....	18
2.2 Особливості виконання ядра програми «Множення матриць».....	19
2.3 Множення матриць при одноразовому виконанні програми.....	20
2.3.1 Обидва масиви поміщаються в кеш-пам'ять даних першого рівня L1D.....	20
2.3.2 Обидва масиви поміщаються в кеш-пам'ять даних першого рівня L1D.....	22
2.3.3 Обидва масиви поміщаються в кеш-пам'ять даних другого рівня .....	23
2.3.4 Обидва масиви поміщаються в кеш-пам'ять даних другого рівня .....	26
2.3.5 Обидва масиви вміщуються в кеш-пам'ять третього рівня.....	27
2.3.6 Обидва масиви вміщуються в кеш-пам'ять третього рівня.....	29
2.3.7 Масиви не поміщаються в кеш-пам'ять третього рівня.....	30
2.3.8 Масиви не поміщаються в кеш-пам'ять третього рівня.....	32
2.4 Множення матриці при одночасному виконанні 2-х потоків.....	34
2.4.1 Обидва масиви поміщаються в кеш-пам'ять даних першого рівня L1D.....	34
2.4.2 Обидва масиви поміщаються в кеш-пам'ять даних першого рівня L1D.....	37
2.4.3 Обидва масиви поміщаються в кеш-пам'ять даних другого рівня.....	40
2.4.4 Обидва масиви поміщаються в кеш-пам'ять даних другого рівня.....	43
2.4.5 Обидва масиви вміщуються в кеш-пам'ять третього рівня.....	45
2.4.6 Обидва масиви вміщуються в кеш-пам'ять третього рівня.....	47
2.4.7 Масиви не поміщаються в кеш-пам'ять третього рівня. ....	49
2.4.8 Масиви не поміщаються в кеш-пам'ять третього рівня.....	52
2.5 Експериментальні дані про виконання програми .....	54
2.6 Висновки.....	56
<b>3 ДОСЛІДЖЕННЯ ВИКОНАННЯ ПРОГРАМИ «РІШЕННЯ ДР В ПРИВАТНИХ ПОХІДНИХ»</b> .....	<b>59</b>
3.1 Особливості програми.....	59
3.2 Один потік .....	61

3.2.1 Обидва масиви поміщаються в кеш-пам'ять даних першого рівня L1D.....	61
3.2.2 Обидва масиви поміщаються в кеш-пам'ять даних другого рівня .....	66
3.2.3 Обидва масиви вміщуються в кеш-пам'ять третього рівня.....	68
3.2.4 Масиви не поміщаються в кеш-пам'ять третього рівня.....	70
3.3 Два потоки .....	72
3.3.1 Обидва масиви вміщуються в кеш-пам'ять даних першого рівня L1D.....	72
3.3.2 Обидва масиви поміщаються в кеш-пам'ять даних другого рівня.....	75
3.3.3 Обидва масиви поміщаються в кеш-пам'ять даних третього рівня.....	77
3.3.4 Масиви не поміщаються в кеш-пам'ять третього рівня.....	79
3.4 Експериментальні дані про виконання програми «Рішення ДР у приватних похідних» .....	81
3.5 Висновки.....	82
4 ДОСЛІДЖЕННЯ ВИКОНАННЯ ПРОГРАМИ «ШПФ» .....	85
4.1 Про швидке перетворення Фур'є.....	85
4.2 Особливості виконання програми ШПФ .....	85
4.3 Перша ітерація .....	89
4.4 Друга і наступні ітерації обчислення 512 пар комплексних точок .....	90
4.5 Дослідження реалізацій двопоточності при виконанні програми ШПФ .....	93
4.6 Експериментальні дані про виконання програми «Швидке перетворення Фур'є»..	94
4.7 Висновки.....	95
5 ОХОРОНА ПРАЦІ.....	96
5.1 Загальні питання з охорони праці .....	96
5.2 Аналіз стану умов праці.....	97
5.2.1 Вимоги до приміщень.....	97
5.2.2 Вимоги до організації місця праці .....	98
5.2.3 Навантаження та напруженість процесу праці .....	99
5.3 Виробнича санітарія.....	100
5.3.1 Аналіз небезпечних та шкідливих факторів при виробництві (експлуатації) виробу .....	100
5.3.2 Пожежна безпека.....	101
5.3.3 Електробезпека .....	102
5.4 Гігієнічні вимоги до параметрів виробничого середовища .....	105
5.4.1 Освітлення .....	105
5.4.2 Вентилювання .....	106
5.5 Екологія .....	106
5.6 Висновки до п'ятого розділу .....	107
5.7 Перелік джерел посилань до розділу 5.....	107
ВИСНОВКИ .....	109
ЛІТЕРАТУРА.....	112
Додаток А.....	114
Додаток Б.....	120
Додаток В.....	125
Додаток Г.....	128

## ВСТУП

Комп'ютери відіграють одну з головних ролей у сучасному світі. Їх використовують скрізь від касових апаратів для спрощення грошових операцій, до передових суперкомп'ютерів для моделювання різних фізичних або хімічних процесів, наприклад пошук нових ліків або проектування термоядерних реакторів. Для вирішення складної задачі потрібна велика обчислювальна потужність, коли дослідники отримують достатню потужність вони вдосконалюють математичні моделі і коло замикається.

Але як отримати потрібну обчислювальну потужність? Існують, наприклад, такі шляхи розвитку процесорів: перший з них полягає в збільшенні частоти, а другий - в збільшенні кількості команд, які виконуються за такт (IPC). В ідеалі слід прагнути до використання обох підходів одночасно, тобто, продуктивність комп'ютера збільшувати як за рахунок збільшення частоти, так і збільшення кількостей команд виконуваних за один такт (IPC). IPC і максимально можлива частота комп'ютера взаємопов'язані.

Найбільш прямий шлях до збільшення продуктивності комп'ютера це збільшення частоти. Якби, наприклад, вдалося все затримки в комп'ютері скоротити в до раз, то це призвело б до збільшення швидкодії в таке ж число раз.

Аналізуючи як успіхи мікроелектронної промисловості, так і успіхи розробників комп'ютерів можна зробити такі висновки:

Подальше підвищення частоти роботи процесорів понад (3-4) ГГц стає недоцільним через проблеми з відведенням тепла.

У структур процесорів, реалізованих в одній мікросхемі, використані практично всі відомі методи збільшення продуктивності. При переході на нові технологічні процеси в одному процесорі вдається підвищити продуктивність на (10-15)% при збереженні частоти роботи і споживаної енергії.

В розпорядженні розробників з'являються реальні ресурси у вигляді подвійної кількості транзисторів приблизно кожні два роки, а ось ефективно їх використовувати і істотно збільшити продуктивність одного окремо взятого ядра процесора не вдається.

Подальше збільшення продуктивності можливо тільки за умови збільшення ступеня паралелізму на рівні ядер процесорів, та систем з багатьма ядрами.

Одним із методів підвищення продуктивності ядер процесорів є віртуальна багатопоточність. Дослідження можливостей її реалізації та умов її використання є метою магістерської роботи.

# 1 МЕТОДИ ПІДВИЩЕННЯ ПРОДУКТИВНОСТІ ЯДЕР СУЧАСНИХ ПРОЦЕСОРІВ

## 1.1 Суть та передумови віртуальної багатопоточності

Одна з проблем сучасних ядер - це неможливість ефективно використовувати наявні великі ресурси спеціалізованих функціональних виконавчих пристроїв підсистеми обробки команд. Причина цього не погано розроблені структури ядра, а властивості реальних програм. Найчастіше в програмах використовується обмежене число спеціалізованих функціональних пристроїв (СФУ). Невикористані СФУ простоюють.

При наявності інформаційних залежностей між командами навіть з урахуванням найсучасніших технологій позачергового виконання команд, перейменування регістрів неминучі простої спеціалізованих функціональних виконавців пристроїв підсистеми обробки.

Ще більші простої виконавчих пристроїв виникають при зверненнях до кеш-пам'яті останнього рівня або до оперативної пам'яті (десятки і навіть сотні тактів) при невірному передбачених переходах, довго виконуються командах і т.п.

При заторі в одній стадії інші стадії конвеєра чекають, поки потрібний ресурс стане доступним. Якщо з поточною програмою нічого зробити не можна, поки не розсмокчиться затор, - то може, переключитися на іншу програму?

Народилася ідея одночасної багатопоточності (simultaneous multithreading або SMT, за термінологією Інтел - гіперпоточність, hyper-threading / HT) або віртуальної багатоядерності. Кожна фізична ядро представляється операційній системі як кілька логічних ядер, хоча майже всі ресурси в них є розділяються (крім архітектурного реєстрового файлу (РФ), стека адрес повернення і деяких інших блоків пам'яті, унікальних для кожного потоку).

### 1.1.1 Види структур багатопоточної віртуальності

Багатопоточні ядра процесорів паралельно в єдиному конвеєрі виконують команди з різних потоків управління.

Існує два основних типи багатопоточних ядер процесорів: виконують за такт команди тільки з одного потоку і виконують за такт команди з кількох потоків.

Більшість сучасних суперскалярних ядер, що виконують команди зі зміною їх черговості, підтримують принцип одночасної багатопоточності, наприклад ядра процесорів фірм IBM, Intel, AMD Zen 2 і Zen 3.



Одночасна багатопоточність передбачає виконання декількох потоків в кожному ядрі - планувальник використання обчислювальних ресурсів (СФУ) ядра визначає незалежні готові до виконання команди з обох потоків і одночасно видає їх в різні СФУ, тим самим забезпечуючи підвищений коефіцієнт їх використання.

Ідея підвищення продуктивності ядер процесорів шляхом реалізації в них віртуальної багатопоточності приваблива.

Однак, а чи не позначиться зменшена в 2 рази інтенсивність генерації команд для кожного потоку (при віртуальній двопоточності) негативно на продуктивності при виконанні паралельно двох потоків? Адже сумарна теоретична інтенсивність генерації команд фізичного ядра залишилася колишньою ( $1/2\lambda_{теор} + 1/2\lambda_{теор} = \lambda_{теор}$ ). А якщо обидва потоки виконують однакові потоки? За рахунок яких чинників можна досягти збільшення продуктивності? Існують і інші питання спільної роботи двох віртуальних ядер в складі однієї фізичної ядра.

Перш ніж приймати будь-які рішення щодо структурних питань ядер процесорів (та й будь-яких підсистем комп'ютера) необхідно достатньо добре дослідити вплив різних параметрів (в тому числі і «перешкод») на ефективність досліджуваних об'єктів.

## **1.2 Огляд літератури з дослідження реальної ефективності ядер процесорів з урахуванням «перешкод»**

### **1.2.1 Види «перешкод»**

При виконанні реальних програм ідеальну роботу підсистеми підготовки команд ядра процесора заважає ряд факторів, які називаються «перешкодами», це:

- Недостатня продуктивність оперативної пам'яті.
- Структурні конфлікти.
- Інформаційні залежності.
- Операції редукації.
- Конфлікти з управління.

### **1.2.2 Вплив оперативної пам'яті**

Дослідженню впливу продуктивності оперативної пам'яті на продуктивність комп'ютерів присвячена обширна література.

При цьому правильно зазначається, що однією з причин істотного зменшення реальної продуктивності ядер процесорів (і, відповідно, всього комп'ютера) від їх теоретичної

продуктивності є вплив оперативної пам'яті, особливо при випадковому і значній кількості звернень до неї.

Проблема недостатньої продуктивності оперативної пам'яті в порівнянні з продуктивністю ядра процесора не вирішена досі. Та й стан справ тільки погіршується. Темпи підвищення продуктивності процесорів і раніше істотно перевищують темпи зростання продуктивності оперативної пам'яті, і розрив продовжує збільшуватися.

### **1.2.3 Вирішення проблем зі структурними конфліктами, інформаційними залежностями, операціями редукції**

Структурні конфлікти вирішуються досить просто - розмножуються загальні ресурси.

Інформаційні залежності (залежно за даними), якщо це можливо, усуваються за допомогою технологій перейменування регістрів, позачергового виконання команд.

Операції редукції усунути неможливо. Вони невід'ємна частина алгоритму розв'язання задачі.

### **1.3 Вплив конфліктів по керуванню**

Залежно з управління виникають при виявленні команд переходу, як безумовних, так і умовних.

При виконанні будь-якої команди переходу проводиться перезавантаження конвеєра підготовки команд для виконання. При виконанні команди умовного переходу необхідно мати ознаку переходу, і тільки після його аналізу проводитиме перезавантаження конвеєра, якщо це необхідно.

Очікування формування ознаки переходу (причому, чим більше ступенів у конвеєра, тим більше час очікування) і перезавантаження конвеєра підготовки команд для виконання призводять до простоїв дешифратора команд і підсистеми обробки, а отже, і до зниження продуктивності всієї системи.

Втрати часу на перезавантаження конвеєра підготовки команд для виконання можуть бути мінімізовані за рахунок реалізації швидкодіючих підсистем кеш-пам'яті, спеціальних буферів команд.

В роботі [1] аналізувалися деякі структурні методи компенсації впливу залежностей з управління на продуктивність ядер сучасних процесорів.

Були досліджені три варіанти структури ядра: без блоку передбачення напрямки переходу; з блоком передбачення напрямки переходу, але без спеціальних буферів команд; з блоком передбачення напрямки переходу і спеціальним буфером розшифрувати команд.

Проведено порівняння ефективності різних структурних рішень по компенсації негативного впливу команд переходів на продуктивність ядра.

Показано, що в високопродуктивних ядрах для компенсації негативного впливу команд переходу на продуктивність недостатньо наявності тільки блоку передбачення переходів навіть з хорошими показниками ймовірності «вдалих» передбачень переходів. Необхідні структурні рішення, які забезпечили б мінімальні часи перезавантаження конвеєра команд при «вдалих» прогнозах переходів близькі до часів генерації команд дешифратором (наприклад, реалізація буфера дешифрованих команд).

Відзначено, що при істотних значеннях ймовірностей залежності з управління коефіцієнт використання підсистеми виконання команд значно зменшується.

Однак в цій роботі не досліджувалася ефективність окремих підсистем ядра (його підсистеми підготовки команд та підсистеми виконання команд).

Зважаючи на складність проблеми оцінки ефективності ядер сучасних процесорів з урахуванням всього різноманіття складності їх структури і динамічних умов виконання програм в умовах «перешкод» її вирішують шляхом декомпозиції від простого до складного.

Наприклад:

1. Досліджують ефективність функціонування комп'ютера з одноядерним процесором з універсальним функціональним пристроєм при відсутності «перешкод».
2. Досліджують ефективність функціонування ядра процесора зі спеціалізованими функціональними пристроями при відсутності «перешкод».
3. Досліджують ефективність функціонування ядра процесора з урахуванням комплексного впливу «перешкод» у вигляді інформаційних залежностей, операцій редукції, команд переходів і команд звернення до ОП.

Тільки після дослідження ефективності функціонування ядра процесора з урахуванням комплексного впливу «перешкод» можна відповісти на численні запитання про доцільність реалізації віртуальної багатопоточності.

#### **1.4 Ефективність функціонування комп'ютера з одноядерним процесором з універсальним функціональним пристроєм при відсутності перешкод**

В роботі [2] досліджувалася ефективність функціонування конвеєрного ядра процесора.

Було показано, що для цілей дослідження загальної ефективності компонентів ядра процесора і спрощення моделі може бути здійснена заміна спеціалізованих ФУ одним універсальним ФУ з еквівалентної продуктивністю.

Були отримані інтегральні оцінки ефективності підсистеми підготовки команд (узагальненого пристрою управління) і підсистеми виконання команд у вигляді універсального функціонального пристрою в залежності від розміру буфера заявок і коефіцієнта навантаження універсального ФУ.

Розмір буфера заявок  $n = 32$  забезпечує близькі до граничних значення коефіцієнтів завантаження.

#### **1.5 Ефективність функціонування конвеєрного ядра процесора зі спеціалізованими функціональними пристроями**

У реальних високопродуктивних ядрах сучасних процесорів підсистема обробки складається з багатьох спеціалізованих функціональних пристроїв.

З огляду на це необхідно мати не тільки інтегральну (узагальнену) оцінку ефективності як підсистеми підготовки команд і всієї підсистеми виконання команд, а й оцінки ефективності для кожного спеціалізованого функціонального пристрою в підсистемі виконання команд ядра процесора при виконанні різних програм.

В роботі [2] досліджувалася ефективність функціонування конвеєрного ядра процесора за умови, що підсистема виконання команд (підсистема обробки) складається з спеціалізованих функціональних пристроїв (ФУ).

В роботі була запропонована методика визначення коефіцієнтів використання як УУ, так і спеціалізованих ФУ з урахуванням частот появи в програмі команд різних типів.

Використання запропонованої моделі та методики для оцінки ефективності як підсистеми підготовки команд (пристрої управління -УУ), так і спеціалізованих функціональних пристроїв (СФУ) ядер процесорів показало, що в ідеальному випадку при виконанні програм з кількістю типів команд в програмі менше числа в підсистемі виконання, при відсутності «перешкод» роботі конвеєра теоретичні коефіцієнти завантаження значно менше 1, а коефіцієнт використання підсистеми підготовки команд (УУ) дорівнює 1.

Це означає, що підсистеми виконання команд недовантажені і частина часу простоюють. У роботі [2] не враховувався комплексний вплив «перешкод».

### **1.6 Комплексний вплив «перешкод» у вигляді інформаційних залежностей, операцій редуції, команд переходів та звернення до ОП на ефективність ядр процесорів**

Дані про реальну ефективність підсистем ядра при наявності «перешкод» у програмах у вигляді інформаційних залежностей, операцій редуції, наявності значної частини команд звернення до ОП в літературі відсутні. У той же час відомо, наприклад, що інформаційні залежності між командами, операції редуції, в реальних програмах зустрічаються дуже часто.

В роботі [3] розглянуто вплив «перешкод» роботи конвєсера ядра у вигляді команд переходів, інформаційних залежностей між командами, операцій редуції на реальну продуктивність і ефективність ядра процесора на прикладах програм «Множення матриць» та «Рішення диференціальних рівнянь в приватних похідних методом сіток». Зроблено висновки про те, що реальна ефективність ядра процесора при виконанні програм в ідеальних умовах, коли всі дані розміщуються в кеш-пам'яті даних першого рівня L1D, складають:

програми «Множення матриць» - не більше 1 команди за такт, а ефективність - менше 0.25.

програми «Рішення диференціальних рівнянь в приватних похідних методом сіток» - близько 0.725 команд за такт, а ефективність менш 0.20.

У цій роботі були досліджені окремі випадки впливу «перешкод» на ефективність ядра процесора при виконанні тільки 2-х типів програм в ідеальних умовах.

В той же час не досліджувалися:

вплив продуктивності ОП;

розмірності оброблюваних даних;

ефективність завантаження (коефіцієнти використання) основних компонент ядра процесора;

можливість одночасного виконання декількох потоків команд в одному ядрі.

Висновки.

При огляді літератури за темою не вдалося знайти роботи, в яких в достатній мірі відбивалися порушені проблеми.

Тому метою даної магістерської роботи є дослідження комплексного впливу «перешкод» на ефективність ядра процесора при виконанні різних типів програм в близьких до реальних умов, а саме дослідження:

впливу продуктивності ОП;

розмірності оброблюваних даних;  
 ефективність завантаження основних компонент ядра процесора;  
 можливість і ефективність одночасного виконання декількох потоків команд в одному ядрі.

### 1.7 Методика проведення дослідження

Дослідження комплексного впливу «перешкод» (інформаційних залежностей, операцій редукції, команд переходів) на ефективність ядер процесорів буде проводитися на прикладі широко поширених і наочних програм, таких як «Множення матриць», «Рішення диференціальних рівнянь в приватних похідних методом сіток», «Швидке перетворення Фур'є».

В досліджуваних програмах:

- виділяється ядро - ділянка програми, що забезпечує основний внесок під час виконання програми;
- розробляється на умовному асемблері програма ядра;
- з'ясовуються інформаційно залежні команди і команди редукції, якщо вони є;
- формуються групи команд, що виконують інформаційно залежні ділянки команд ядра;
- розробляється модель ядра процесора, що виконує ядро програми;
- в моделі інформаційно залежні групи команд виконуються універсальним функціональним пристроєм послідовно згідно з алгоритмом виконання програми;

Далі визначаються:

- кількість типів груп команд;
- ймовірності появи кожної групи в програмі;
- часи виконання кожної групи;
- середній час виконання ядра програми;
- коефіцієнт використання універсального функціонального пристрою;
- коефіцієнт використання пристрою управління моделі в залежності від значення різних параметрів програми і ядра процесора;
- формуються висновки;
- експериментально перевіряється достовірність розроблених теоретичних моделей.

## 1.8 Модель ядра процесора при виконанні програми

Пропонується наступна спрощена модель ядра процесора. Пристрій управління (УУ) моделі ядра процесора (еквівалент так званої front end частини ядра процесора) читає програму, дешифрує команди і поміщає чергову групу команд в буфер розшифрованих груп команд підсистеми виконання. Імовірність генерації  $i$ -ї групи команд  $W_i$ .

Передбачається, що:

Процес генерації груп команд УУ найпростіший з показовим законом розподілу часу між згенерованими групами команд. Інтенсивність створення груп команд

$$\lambda_{УУ} = \frac{1}{t_{ген}}$$

де  $t_{ген}$  - математичне очікування часу генерації групи команд.

Час підготовки чергової групи команд УУ залежить тільки від архітектурних і структурних особливостей ядра процесора.

УУ переходить до генерації нової групи команд тільки після завершення генерації попередньої групи команд.

Якщо буфер груп команд заповнений, то УУ припиняє генерацію груп команд.

Підсистема виконання груп команд (еквівалент так званої back end частини ядра процесора) складається з універсального функціонального пристрою (ФУ), який може виконувати будь-яку групу команд.

Передбачається, що універсальний ФУ, виконує групи команд з інтенсивністю

$$\mu_{ФУ} = \frac{1}{t_{ФУ}}$$

де  $t_{ФУ} = \sum_{i=1}^n \omega_{iгрупи} * t_{iгрупи}$ .

Група команд, що складається з команд читання 2-х операндів з кеш-пам'яті даних першого рівня (L1D), команди Множення прочитаних операндів і команди Складання результатів Множення (тип групи L1; L1; Мнж; Скл) виконується з інтенсивністю

$$\mu_L = \frac{1}{t_{СКЖ}}$$

Група команд, що складається з команди читання операнду з кеш-пам'яті даних першого рівня (L1D), команди читання операнду з оперативної пам'яті, команди Множення прочитаних

операндів і Складання результатів множення (тип групи L1; ОП; Мнж; Скл) виконується з інтенсивністю

$$\mu_{ОП} = \frac{1}{t_{ОП}},$$

де  $t_{ОП}$  - час читання операнду з оперативної пам'яті.

Група команд, що складається з команди читання операндів з кеш-пам'яті даних першого рівня L1D, команди читання операндів з кеш-пам'яті другого рівня L2, команди Множення прочитаних операндів і команди Складання результатів множення (тип групи L1; L2; Мнж; Скл) виконується з інтенсивністю, яка визначається часом читання операнду з кеш-пам'яті другого рівня L2

$$\mu_{L2} = \frac{1}{t_{L2}}.$$

Група команд, що складається з команди читання операнду з кеш-пам'яті даних першого рівня L1D, команди читання операнду з кеш-пам'яті третього рівня L3, команди Множення прочитаних операндів і команди Складання результатів Множення (тип групи L1; L3; Мнж; Скл) виконується з інтенсивністю, яка визначається часом читання операнду з кеш-пам'яті третього рівня L3

$$\mu_{L3} = \frac{1}{t_{L3}}.$$

Передбачається також, що:

Якщо ФУ вільний, то чергова згенерувала група команд з УУ відразу надходить на виконання в ФУ.

Якщо ж ФУ зайнятий виконанням групи команд, то чергова згенерована група команд з УУ надходить в буфер груп команд.

ФУ вибирає групи команд з буфера груп команд згідно дисципліни FIFO.

У будь-який момент часу ФУ виконує тільки одну групу команд.

ФУ переходить до виконання нової групи команд тільки після завершення виконання попередньої групи команд.

Якщо ФУ вільний і в буфері груп команд немає заявок, то ФУ простоює.

Потік виконаних в ФУ груп команд найпростіший, закон розподілу показовий з інтенсивністю згідно до формули

$$\mu_{ФУ} = \frac{1}{t_{ФУ}},$$



Процес функціонування моделі описаного вище ядра процесора при виконанні ядра програми може бути представлений у вигляді двофазної системи масового обслуговування, що складається з УУ, ФУ і буфера груп команд на  $n$  заявок між ними.

Структурна схема моделі представлена на рис. 1.1.

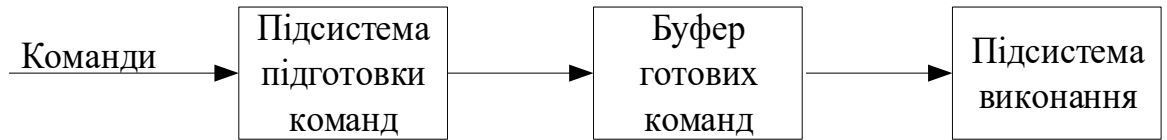


Рис. 1.1 – Структурна схема моделі

Склавши і вирішивши систему рівнянь для ймовірностей перебування моделі в різних станах можна отримати наступні вирази для коефіцієнтів використання компонентів моделі:

коефіцієнт використання універсального ФУ.

$$E = \frac{\rho^*(1-\rho^{n+1})}{1-\rho^{n+2}} \text{ для } \rho \neq 1; E = \frac{n+1}{n+2} \text{ для } \rho = 1$$

коефіцієнт використання УУ

$$H = \frac{1-\rho^{n+1}}{1-\rho^{n+2}} \text{ для } \rho \neq 1; H = \frac{n+1}{n+2} \text{ для } \rho = 1,$$

де:  $\rho$  – це коефіцієнт навантаження універсального ФУ, а  $n$  – розмір буфера.

При  $n \geq 64$  коефіцієнти завантаження компонентів моделі дуже близькі до граничних значень при  $n \rightarrow \infty$ , і рівні:

**коефіцієнт використання універсального ФУ:**

$$E = \rho - \text{при } \rho < 1; E = 1 - \text{при } \rho \geq 1;$$

**коефіцієнт використання УУ:**

$$H = \frac{1}{\rho} - \text{при } \rho > 1; H = 1 - \text{при } \rho \leq 1.$$

При  $\rho = 1$  про коефіцієнт використання ядра процесора можна судити за коефіцієнтом використання як УУ, так і еквівалентного ФУ.

У реальних сучасних ядрах процесорів підсистема генерації команд (УУ) більш продуктивна порівняно з підсистемою виконання команд (в моделі аналог підсистеми

виконання команд це еквівалентну ФУ). Відповідно коефіцієнт навантаження підсистеми виконання команд  $\rho > 1$  і про коефіцієнт використання ядра процесора необхідно судити за коефіцієнтом використання підсистеми генерації команд (УУ в розглянутій моделі), який дорівнює зворотній величині коефіцієнта навантаження ФУ.

Коефіцієнт навантаження еквівалентного ФУ (надалі ФУ) дорівнює

$$\rho_{\text{ФУ}} = \frac{\lambda_{\text{УУ}}}{\mu_{\text{ФУ}}} = \frac{t_{\text{ФУ}}}{t_{\text{УУ}}}. \quad (1.1)$$

Середній час виконання групи команд універсальним ФУ

$$t_{\text{ФУ}} = \sum_l^m w_{i_{\text{групи}}} * t_{i_{\text{групи}}}. \quad (1.2)$$

Ймовірності генерації груп команд визначаються конкретним видом виконуваної програми, а саме розмірами матриць і параметрами підсистеми пам'яті.

### **1.9 Особливості підсистеми виконання команд, що складається з кеш-пам'яті даних першого рівня L1D, пристроїв Множення і Складання в ядрах сучасних процесорів**

В ядрах сучасних процесорів кеш-пам'ять даних першого рівня L1D це конвеєрний пристрій з пропускною спроможністю 1 такт. Повний час читання з кеш-пам'яті даних першого рівня L1D становить (3-4) такти. Якщо операнди знаходяться в кеш-пам'яті даних першого рівня L1D, то два операнди можуть бути прочитані одночасно, причому читання може починатися кожний такт. Одночасно з операціями «Читання» з цієї кеш-пам'яті може виконуватися операція «Запис».

Пристрої Множення і Складання - це конвеєрні пристрої з пропускною спроможністю 1 такт і повним часом виконання (3-4) такти.

Для спрощення приймається, що повні часи читання з кеш-пам'яті даних першого рівня L1D, запису в кеш-пам'ять даних першого рівня L1D, множення і складання рівні 4-м тактам.

Якщо пристрій додавання виконує операцію редукції (нагромаджує складання), то він працює як комбінаційний пристрій з часом виконання 4 такти. З такою пропускною спроможністю змушені працювати і кеш-пам'ять даних першого рівня L1D, і пристрій Множення .

Час виконання групи команд визначається гіршим часом виконання або 2-х команд читання операндів з підсистеми пам'яті, або команди «Множення », або команди «Складання».

## 2 ДОСЛІДЖЕННЯ ВИКОНАННЯ ПРОГРАМИ «МНОЖЕННЯ МАТРИЦЬ»

### 2.1 Про програму «Множення матриць»

Нехай в ядрі процесора виконується програма «Множення матриць розмірністю  $n \times n$ ». Для отримання чергового елемента матриці результату виконується скалярний твір вектор-рядка матриці А на вектор-стовпець матриці В відповідно до алгоритму скалярного множення векторів

$$C_{ij} = \sum_{k=1}^n a_{ik} * b_{kj}.$$

Якщо розмір матриць досить великий, то для оцінки часу виконання ядра програми можна знехтувати часом виконання команд, які готують основний цикл.

Час виконання програми «Множення матриць» визначається специфікою виконуваної програми ядра процесора, а саме:

- Набором виконуваних команд.
- Наявністю інформаційних залежностей між командами.
- Наявністю або відсутністю будь-якої операції редукції в циклах.
- Часом виконання команд (перш за все команд звернення до оперативної пам'яті).
- Алгоритмом виконання команд записи в оперативну пам'ять (наскрізна або відкладений запис) і рядом інших особливостей.
- Кількістю спеціалізованих функціональних пристроїв в підсистемі виконання дешифрованих команд.
- Типом (комбінаційний або конвеєрний пристрій) і продуктивністю універсального функціонального пристрою.

Для оцінки ефективності ядра процесора при виконанні програми «Множення матриць» досліджується виконання ядра програми «Множення матриць», час виконання якого становить основну частину часу виконання програми.

Ядро програми «Множення матриць» це ділянка програми, яка виконує скалярний добуток рядка матриці А на стовпець матриці В. Умовною мовою асемблера ця ділянка програми (ядро) може мати такий вигляд (рис. 2.1):

1. **LOAD**  $a_{ik} \rightarrow R1$ , спосіб адресації при читанні - автоіндексація.
2. **LOAD**  $b_{kj} \rightarrow R2$ , спосіб адресації при читанні - автоіндексація.
3. **MPY**  $R1 * R2 \rightarrow R3$ .
4. **ADD**  $R0 + R3 \rightarrow R0$ .
5. **JMP** в початок циклу, якщо його треба продовжувати.
6. **STORE**  $R0 \rightarrow c_{ij}$ , спосіб адресації при записи - автоіндексація.

Рис. 2.1 -ядро програми визначення чергового результату

Цикл обчислення одного елементу результату повторюється  $n$  раз.

Виконання команди **JMP** поєднується з виконанням групи команд обчислення проміжного результату, що складається з перших 4-х команд ядра програми. Тому час виконання цієї команди не враховується при визначенні загального часу виконання ядра програми «Множення матриць».

Час виконання команди «Запис в ОП» після обчислення чергового елемента результату також не враховується, так як вона поєднується з виконанням наступного циклу ядра програми при обчисленні наступного елемента результуючої матриці.

## 2.2 Особливості виконання ядра програми «Множення матриць»

Підсистема підготовки команд ядра процесора генерує для підсистеми виконання ядра процесора групу з 4-х команд.

1. **LOAD**  $a_{ik} \rightarrow R1$ .
2. **LOAD**  $b_{kj} \rightarrow R2$ .
3. **MPY**  $R1 * R2 \rightarrow R3$ .
4. **ADD**  $R0 + R3 \rightarrow R0$ .

Перші 2 команди циклу ядра програми «Читання операндів» з підсистеми пам'яті незалежні і можуть виконуватися одночасно (паралельно).

3-я команда циклу «Множення» виконує множення операндів, які прочитані 1-ю і 2-ю командами циклу з підсистеми пам'яті. Ця команда інформаційно залежна від команд «Читання» операндів з підсистеми пам'яті і не може бути розпочата до тих пір, поки не будуть прочитані операнди з підсистеми пам'яті.

4-а команда циклу «Додавання» може додавати результат команди «Множення» і результат попередньої команди «Додавання». Ця команда інформаційно залежна від команди

«Множення» і попереднього результату команди «Додавання». Вона не може бути розпочата до тих пір, поки не будуть отримані обидва операнди, тобто поки не завершиться команда «Множення» і команда «Додавання» з попередніми операндами. Крім інформаційної залежності від команди «Множення» команда «Додавання» інформаційно залежить і від результату виконання попередньої команди «Додавання», тобто команда «Додавання» виконує операцію редуції.

## 2.3 Множення матриць при одноразовому виконанні програми

### 2.3.1 Обидва масиви поміщаються в кеш-пам'ять даних першого рівня L1D

В кеш-пам'яті даних першого рівня L1D поміщається матриця В і, принаймні, один поточний рядок матриці А.

У цьому випадку генеруються наступні групи команд:

- ОП; ОП; Мнж; Скл.
- L1; ОП; Мнж; Скл.
- L1; L1; Мнж; Скл.

Група команд ОП; ОП; Мнж; Скл замінюється еквівалентним збільшенням кількості звернень в оперативну пам'ять в групі команд L1; ОП; Мнж; Скл і в подальшому не враховується.

Отже, в циклі залишаються тільки такі групи команд:

- L1; ОП; Мнж; Скл.
- L1; L1; Мнж; Скл.

**Ймовірності груп команд при виконанні одного потоку з  $n^3$  циклів:**

$$P(L1; ОП; Мнж; СКЛ) = \frac{N(L1; ОП; Мнж; СКЛ)}{n^3} \cong \frac{2 * n^2}{16 * n^3} \cong \frac{1}{8 * n}$$

$$P(L1; L1; Мнж; СКЛ) = \frac{N(L1; L1; Мнж; СКЛ)}{n^3} \cong \frac{n^3 - 2 * n^2 / 16}{n^3} = \left(1 - \frac{1}{8 * n}\right)$$

**Час виконання груп команд при частоті ядра  $f=4$  ГГц:**

тип L1; L1; Мнж; Скл – 4 такти;

тип L1; ОП; Мнж; Скл – 256 тактів.

Середній час виконання універсальним ФУ групи команд(1.2):

$$t_{\Phi Y} = \frac{1}{8*n} * t_{OP} + \left(1 - \frac{1}{8*n}\right) * t_{CKL} \cong 4.492\tau.$$

$$\rho_{\Phi Y} = \frac{4.492\tau}{k\tau}.$$

При  $k=1$  (найбільш продуктивний варіант УУ)–  $\rho_{\Phi Y} = 4.492$ , а коефіцієнт використання УУ дорівнює 0.223 або 22.3%.

При  $k=2$  (менш продуктивний варіант УУ)–  $\rho_{\Phi Y} = 2.246$ , а коефіцієнт використання УУ дорівнює 0.446 або 44.6%.

### Загальний час виконання одного потоку

$$T_{ЗАГ} = T_{OP} + T_{OBЧ} \cong \frac{2n^2}{16} t_{OP} + n^3 t_{OBЧ} \cong \frac{n^2}{8} t_{OP} + n^2 t_{OP} \cong 1.125n^2 t_{OP}.$$

Так як  $n \leq 64$ , а  $nt_{CKL} \leq t_{OP}$ , то загальний час обчислення матриці результату визначається часом читання з ОП операндів і часом записи елементів результату.

### Загальний час зайнятості оперативної пам'яті

$$T_{ЗАГ}^{OP} = T_{OP}^{ЧТ} + T_{OP}^{ЗП} \cong \frac{2n^2}{16} t_{OP} + n^2 t_{OP} \cong 1.125n^2 t_{OP}$$

Якщо  $n = 64$  і 32-х розрядні елементи матриці, то за час обчислення чергового елемента результату паралельно виконується запис в ОП попереднього елемента. Тому час запису не включається в загальний час виконання програми. А в загальний час зайнятості ОП цей час включається.

### Коефіцієнт використання оперативної пам'яті

$$KKД_{OP} = \frac{T_{ЗАГ}^{OP}}{T_{ЗАГ}} = \frac{T_{OP}^{ЧТ} + T_{OP}^{ЗП}}{T_{ЗАГ}} \cong 1$$

### Висновок

При  $n \leq \frac{t_{OP}}{t_{СЛЖ}} \leq 64$  коефіцієнт використання (завантаження - ймовірність зайнятості) оперативної пам'яті дорівнює 1. Це означає, що запускати додатково потоки в цьому ядрі під час виконання програми Множення матриць з  $n = 64$  недоцільно.

Якщо  $n > 64$ , то  $KPD_{OP} < 1$  і можлива робота 2-х потоків.

**Коефіцієнт використання пропускної здатності підсистеми обчислення, що складається з кеш-пам'яті L1, пристроїв Множення і Складання**

Коефіцієнт корисного використання визначається як відношення теоретичного (пікового) значення пропускної здатності при виконанні однієї групи команд L1; L1; Мнж; Скл до реальної пропускної здатності при виконанні тієї ж однієї групи команд.

$$KKD_{OБЧ} = \frac{\mu_{TEOP}}{\mu_{PEAL}} = \frac{1}{4} = 0.25$$

Так як підсистема обчислення 3 такти з 4-х тактів простоює, то реальний коефіцієнт її використання складе  $KKD_{OБЧ} = 0.25$ .

### 2.3.2 Обидва масиву поміщаються в кеш-пам'ять даних першого рівня L1D

**Ймовірності груп команд при виконанні одного потоку з  $n^3$  циклів:**

$$P(L1; OP; Mнж; Cкл) = \frac{N(L1; OP; Mнж; Cкл)}{n^3} \cong \frac{2 * n^2}{8 * n^3} \cong \frac{1}{4 * n}$$

$$P(L1; L1; Mнж; Cкл) = \frac{N(L1; L1; Mнж; Cкл)}{n^3} \cong \frac{n^3 - 2 * n^2 / 8}{n^3} \cong \left(1 - \frac{1}{4 * n}\right)$$

Час виконання груп команд при частоті ядра  $f=4$  ГГц не змінюється порівняно з варіантом, коли елементи матриць це 32-х розрядні данні.

Середній час виконання універсальним ФУ груп команд

$$t_{\Phi Y} = \frac{1}{4 * n} * t_{OP} + \left(1 - \frac{1}{4 * n}\right) * t_{CKЛ} \cong 4.84\tau$$

$$\rho_{\Phi Y} = \frac{4.84\tau}{k\tau}$$

При  $k=1$  (більш продуктивний варіант УУ) –  $\rho_{\Phi Y} = 4.984$ , а **коефіцієнт використання УУ дорівнює 0.2066 або 20.66%**.

При  $k=2$  (менш продуктивний варіант УУ) –  $\rho_{\Phi Y} = 2.492$ , а коефіцієнт використання УУ дорівнює 0.4013 або 40.13%.

### Загальний час виконання одного потоку ядра програми

$$T_{ЗАГ} = T_{ОП} + T_{ОБЧ} \cong \frac{2n^2}{8} t_{ОП} + n^3 t_{ОБЧ} \cong \frac{2n^2}{8} t_{ОП} + n^2 t_{ОП} \cong 1.25n^2 t_{ОП}$$

### Загальний час зайнятості оперативної пам'яті при виконанні одного потоку

$$T_{ЗАГ}^{ОП} = T_{ОП}^{ЧТ} + T_{ОП}^{ЗП} = \frac{2n^2}{8} t_{ОП} + n^2 t_{ОП} = 1.25n^2 t_{ОП}$$

### Коефіцієнт використання оперативної пам'яті

$$ККД_{ОП} = \frac{T_{ЗАГ}^{ОП}}{T_{ЗАГ}} = \frac{T_{ОП}^{ЧТ} + T_{ОП}^{ЗП}}{T_{ЗАГ}} \cong 1$$

### Висновок

При  $n = 64$  при виконанні програми «Множення матриць» з 64-х розрядними даними (як і вразі з 32-розрядними даними) коефіцієнт завантаження (ймовірність зайнятості) оперативної пам'яті дорівнює 1. Це означає, що запускати додатково потоки в цьому ядрі під час виконання програми Множення матриць з  $n = 64$  недоцільно.

Якщо  $n > 64$ , то  $ККД_{ОП} < 1$  та можлива робота 2-х потоків.

### Коефіцієнт використання пропускну здатності підсистеми обчислення

Підсистема обчислення 3 такти з 4-х тактів простоює тому реальний коефіцієнт її використання складе  $ККД_{ОБЧ} = 0.25$ .

### 2.3.3 Обидва масиви поміщаються в кеш-пам'ять даних другого рівня

Матриця В і, принаймні, один поточний рядок матриці А поміщаються в кеш-пам'яті другого рівня L2.

У цьому прикладі генеруються наступні типи груп команд:

- ОП; ОП; Мнж; Скл.
- L1; ОП; Мнж; Скл.
- L1; L2; Мнж; Скл.
- L1; L1; Мнж; Скл.



Як і раніше, група команд ОП; ОП; Мнж; Скл замінюється еквівалентним збільшенням кількості груп команд L1; ОП; Мнж; Скл і в подальшому не враховується.

Разом, враховуються такі типи груп команд:

- L1; ОП; Мнж; Скл.
- L1; L2; Мнж; Скл.
- L1; L1; Мнж; Скл.

**Кількість груп команд кожного виду при роботі з 32-розрядними даними:**

$$N(L1; ОП; Мнж; Скл) = \frac{2n^2}{16}$$

$$N(L1; L2; Мнж; Скл) = \frac{n^2}{16} * (n-1) = \frac{n^3}{16} - \frac{n^2}{16} \cong \frac{n^3}{16}$$

$$N(L1; L1; Мнж; Скл) = n^3 - \frac{2n^2}{16} - \frac{n^2}{16} * (n-1) = \frac{15*n^3}{16} - \frac{n^2}{16} \cong \frac{15*n^3}{16}$$

**Ймовірності груп команд:**

$$P(L1; ОП; Мнж; Скл) = \frac{N(L1; ОП; Мнж; Скл)}{n^3} = \frac{2n^2}{16 * n^3} = \frac{1}{8 * n}$$

$$P(L1; L2; Мнж; Скл) = \frac{N(L1; L2; Мнж; Скл)}{n^3} = \frac{n^2 * (n-1)}{16 * n^3} \cong \frac{n-1}{16 * n} \cong \frac{1}{16}$$

$$P(L1; L1; Мнж; Скл) = \frac{N(L1; L1; Мнж; Скл)}{n^3} = \frac{15 * n^3}{16 * n^3} - \frac{n^2}{16 * n^3} = \frac{15}{16} - \frac{1}{16n} \cong \frac{15}{16}$$

**Час виконання груп команд при частоті ядра f=4 ГГц:**

- тип L1; L1; Мнж; Скл – 4 такти;
- тип L1; L2; Мнж; Скл – 12 тактів;
- тип L1; ОП; Мнж; Скл – 256 тактів.

Середній час виконання універсальним ФУ групи команд(1.2)

$$t_{\Phi Y} = \frac{1}{8*n} * t_{ОП} + \frac{1}{16} t_{L2} + \left(1 - \frac{1}{8*n} - \frac{1}{16}\right) * t_{СКЛ} \cong 4.75\tau.$$

$$\rho_{\Phi Y} = \frac{4.75\tau}{k\tau}.$$

При k=1 (більш продуктивний варіант УУ) –  $\rho_{\Phi Y} = 4.75$ , а коефіцієнт використання УУ дорівнює 0.2105 або 21.05%.

При  $k=2$  (менш продуктивний варіант УУ) –  $\rho_{\Phi Y} = 2.375$ , а коефіцієнт використання УУ дорівнює 0.4210 або 42.1%.

### Загальний час виконання одного потоку

$$T_{ЗАГ} = T_{ОП} + T_{L2} + T_{ОБЧ} \cong \frac{2n^2}{16} t_{ОП} + \frac{n^3}{16} t_{L2} + \frac{15}{16} n^3 t_{СКЛ}$$

### Загальний час зайнятості оперативної пам'яті

$$T_{ЗАГ}^{ОП} = T_{ОП}^{ЧТ} + T_{ОП}^{ЗП} = \frac{2n^2}{16} t_{ОП} + n^2 t_{ОП} = 1.125n^2 t_{ОП}$$

### Коефіцієнт використання оперативної пам'яті

$$ККД_{ОП} \cong \frac{\frac{2n^2}{16} t_{ОП} + n^2 t_{ОП}}{\frac{2n^2}{16} t_{ОП} + \frac{n^3}{16} t_{L2} + \frac{15}{16} n^3 t_{СКЛ}} \cong \frac{1}{1/9 + \frac{n}{18} \frac{t_{L2}}{t_{ОП}} + n \frac{15}{18} \times \frac{t_{СКЛ}}{t_{ОП}}} \cong 0.47$$

### Коефіцієнт використання підсистеми обчислення

Як і раніше, коефіцієнт корисного використання визначається як відношення теоретичної (пікового) значення пропускної здатності при виконанні однієї групи команд L1; L1; Мнж; Скл до реальної пропускної здатності при виконанні тієї ж однієї групи команд.

$$ККД_{ОБЧ} = \frac{\mu_{ТЕОР}}{\mu_{РЕАЛ}} = \frac{1}{4} = 0.25.$$

Підсистема обчислення 3 такти з 4-х тактів простоює, тому реальна пропускна здатність становить 1 такт.

### Коефіцієнт використання кеш-пам'яті L2

$$ККД_{ОП} = \frac{T_{ОП}^{ОБЩ}}{T_{ОБЩ}} \cong \frac{\frac{n^3}{16} t_{L2}}{\frac{2n^2}{16} t_{ОП} + \frac{n^3}{16} t_{L2} + \frac{15}{16} n^3 t_{СЛЖ}} \cong \frac{1}{\frac{2}{n} \frac{t_{ОП}}{t_{L2}} + 1 + 15 \frac{t_{СЛЖ}}{t_{L2}}} \cong 0.16.$$

### 2.3.4 Обидва масиви поміщаються в кеш-пам'ять даних другого рівня

**Ймовірності груп команд:**

$$P(\text{ОП}; L1; \text{Мнж}; \text{Скл}) = \frac{N(\text{ОП}; L1; \text{Мнж}; \text{Скл})}{n^3} \cong \frac{2 * n^2}{8 * n^3} \cong \frac{1}{4 * n}$$

$$P(L1; L1; \text{Мнж}; \text{Скл}) = \frac{N(L1; L1; \text{Мнж}; \text{Скл})}{n^3} \cong \frac{n^3 - 2 * n^2 / 8}{n^3} \cong \left(1 - \frac{1}{4 * n} - \frac{1}{8}\right) \cong \frac{7}{8}$$

$$P(L1; L2; \text{Мнж}; \text{Скл}) = \frac{N(L1; L2; \text{Мнж}; \text{Скл})}{n^3} = \frac{n^2 * (n - 1)}{8 * n^3} \cong \frac{n - 1}{8n} \cong \frac{1}{8}$$

Часи виконання груп команд не змінюються в порівнянні з варіантом, коли елементи матриць це 32-розрядні дані:

Середній час виконання універсальним ФУ групи команд (1.2)

$$t_{\text{ФУ}} = \frac{1}{4 * n} * t_{\text{ОП}} + \frac{1}{8} * t_{L2} + \left(1 - \frac{1}{4 * n} - \frac{1}{8}\right) * t_{\text{СКЛ}} \cong 5.5\tau.$$

$$\rho_{\text{ФУ}} = \frac{5.5\tau}{k\tau}.$$

При k=1 (більш продуктивний варіант УУ) –  $\rho_{\text{ФУ}} = 5.5$ , а коефіцієнт використання УУ дорівнює 0.1818 або 18.18%.

При k=2 (менш продуктивний варіант УУ) –  $\rho_{\text{ФУ}} = 2.75$ , а коефіцієнт використання УУ дорівнює 0.3636 або 36.36%.

**Загальний час виконання одного потоку**

$$T_{\text{ЗАГ}} = T_{\text{ОП}} + T_{L2} + T_{\text{ОБЧ}} = \frac{2n^2}{8} t_{\text{ОП}} + \frac{n^3}{8} t_{L2} + n^3 t_{\text{СКЛ}}$$

**Загальний час зайнятості оперативної пам'яті**

$$T_{\text{ЗАГ}}^{\text{ОП}} = T_{\text{ОП}}^{\text{ЧТ}} + T_{\text{ОП}}^{\text{ЗП}} = \frac{2n^2}{8} t_{\text{ОП}} + n^2 t_{\text{ОП}} = 1.25n^2 t_{\text{ОП}}$$

### Коефіцієнт використання оперативної пам'яті

$$KKD_{OP} = \frac{T_{OБЩ}^{OP}}{T_{OБЩ}} = \frac{T_{OP}^{ЧТ} + T_{OP}^{ЗП}}{T_{OБЩ}} = \frac{1.25n^2 t_{OP}}{\frac{n^2}{4} t_{OP} + \frac{n^3}{8} t_{L2} + \frac{7}{8} n^3 t_{СЛЖ}} = \frac{1}{\frac{1}{5} + \frac{n}{10} \frac{t_{L2}}{t_{OP}} + \frac{7}{10} n \frac{t_{СЛЖ}}{t_{OP}}} \cong 0.45.$$

Коефіцієнт завантаження (ймовірність зайнятості) оперативної пам'яті 0.45.

### Коефіцієнт використання кеш-пам'яті L2

$$KKD_{L2} = \frac{T_{3AG}^{L2}}{T_{3AG}} \cong \frac{\frac{n^3}{8} t_{L2}}{\frac{2n^2}{8} t_{OP} + \frac{n^3}{8} t_{L2} + \frac{7n^3}{8} t_{СКЛ}} \cong 0.27.$$

### 2.3.5 Обидва масиви вміщуються в кеш-пам'ять третього рівня

Матриця В і, принаймні, один поточний рядок матриці А поміщаються в кеш-пам'ять третього рівня L3.

У цьому прикладі генеруються наступні типи груп команд:

- ОП; ОП; Мнж; Скл.
- L1; ОП; Мнж; Скл.
- L1; L3; Мнж; Скл.
- L1; L1; Мнж; Скл.

Як і в попередніх прикладах група команд ОП; ОП; Мнж; Скл замінюється еквівалентним збільшенням кількості груп команд L1; ОП; Мнж; Скл і в подальшому не враховується.

Разом, враховуються такі типи груп команд:

- L1; ОП; Мнж; Скл.
- L1; L3; Мнж; Скл.
- L1; L1; Мнж; Скл.

**Кількість груп команд кожного виду при роботі з 32-розрядними даними:**

$$N(L1; ОП; Мнж; Скл) = \frac{2n^2}{16}$$

$$N(L1; L3; Мнж; Скл) = \frac{n^2}{16} * (n - 1) = \frac{n^3}{16} - \frac{n^2}{16}$$

$$N(L1; L1; Мнж; Скл) = n^3 - \frac{2n^2}{16} - \frac{n^2}{16} * (n - 1) \cong \frac{15 * n^3}{16} - \frac{n^2}{16}$$

**Ймовірності груп команд:**

$$P(L1; ОП; Мнж; Скл) = \frac{N(L1; ОП; Мнж; Скл)}{n^3} = \frac{2n^2}{16 * n^3} = \frac{1}{8 * n}$$

$$P(L1; L3; Мнж; Скл) = \frac{N(L1; L3; Мнж; Скл)}{n^3} = \frac{n^2 * (n - 1)}{16 * n^3} \cong \frac{n - 1}{16 * n} \cong \frac{1}{16}$$

$$P(L1; L1; Мнж; Скл) = \frac{N(L1; L1; Мнж; Скл)}{n^3} = \frac{15 * n^3}{16 * n^3} - \frac{n^2}{16 * n^3} = \frac{15}{16} - \frac{1}{16n} \cong \frac{15}{16}$$

**Час виконання груп команд при частоті ядра f=4 ГГц:**

- тип L1; L1; Мнж; Скл – 4 такти;
- тип L1; L3; Мнж; Скл – 36 тактів;
- тип L1; ОП; Мнж; Скл – 256 тактів.

Середній час виконання універсальним ФУ групи команд (1.2)

$$t_{\Phi Y} = \frac{1}{8 * n} * t_{ОП} + \frac{1}{16} t_{L3} + \left(1 - \frac{1}{8 * n} - \frac{1}{16}\right) * t_{СКЛ} \cong 6.06 \tau.$$

$$\rho_{\Phi Y} = \frac{6.06 \tau}{k \tau}.$$

При k=1 –  $\rho_{\Phi Y} = 6.06$ , а коефіцієнт використання УУ дорівнює 0.165 або 16.5%.

При k=2 –  $\rho_{\Phi Y} = 3.03$ , а коефіцієнт використання УУ дорівнює 0.33 або 33%.

**Загальний час виконання одного потоку**

$$T_{ЗАГ} = T_{ОП} + T_{L3} + T_{ОБЧ} \cong \frac{2n^2}{16} t_{ОП} + \frac{n^3}{16} t_{L3} + \frac{15}{16} n^3 t_{СКЛ}.$$

### Загальний час зайнятості оперативної пам'яті

$$T_{ЗАГ}^{ОП} = T_{ОП}^{ЧТ} + T_{ОП}^{ЗП} = \frac{2n^2}{16} t_{ОП} + n^2 t_{ОП} = 1.125n^2 t_{ОП}.$$

### Коефіцієнт використання оперативної пам'яті

$$ККД_{ОП} = \frac{T_{ЗАГ}^{ОП}}{T_{ЗАГ}} = \frac{T_{ОП}^{ЧТ} + T_{ОП}^{ЗП}}{T_{ЗАГ}} \cong \frac{\frac{2n^2}{16} t_{ОП} + n^2 t_{ОП}}{\frac{2n^2}{16} t_{ОП} + \frac{n^3}{16} t_{L3} + \frac{15}{16} n^3 t_{СКЛ}}.$$

Після спрощення виразу отримаємо

$$ККД_{ОП} \cong \frac{18}{2+n \frac{t_{L3}}{t_{ОП}} + 15n \frac{t_{СКЛ}}{t_{ОП}}} \cong 0.092$$

### Коефіцієнт використання кеш-пам'яті L3

$$ККД_{L3} = \frac{T_{ЗАГ}^{L3}}{T_{ЗАГ}} \cong \frac{\frac{n^3}{16} t_{L3}}{\frac{2n^2}{16} t_{ОП} + \frac{n^3}{16} t_{L3} + \frac{15}{16} n^3 t_{СКЛ}} \cong 0.37.$$

### 2.3.6 Обидва масиви вміщуються в кеш-пам'ять третього рівня

Ймовірності груп команд:

$$P(ОП; L1; Мнж; Скл) = \frac{N(ОП; L1; Мнж; Скл)}{n^3} \cong \frac{2 * n^2}{8 * n^3} \cong \frac{1}{4 * n}$$

$$P(L1; L1; Мнж; Скл) = \frac{N(L1; L1; Мнж; Скл)}{n^3} \cong \frac{n^3 - 2 * n^2 / 8}{n^3} \cong \left(1 - \frac{1}{4 * n} - \frac{1}{8}\right) \cong \frac{7}{8}$$

$$P(L1; L3; Мнж; Скл) = \frac{N(L1; L2; Мнж; Скл)}{n^3} = \frac{n^2 * (n - 1)}{8 * n^3} \cong \frac{n - 1}{8n} \cong \frac{1}{8}$$

Часи виконання груп команд не змінюються в порівнянні з варіантом, коли елементи матриць це 32-розрядні дані а частота ядра процесора  $f=4$  ГГц.

Середній час виконання універсальним ФУ групи команд (1.2)

$$t_{\Phi Y} = \frac{1}{4*n} * t_{OP} + \frac{1}{8} * t_{L3} + \left(1 - \frac{1}{4*n} - \frac{1}{8}\right) * t_{CKL} \cong 8.125\tau.$$

$$\rho_{\Phi Y} = \frac{8.125\tau}{k\tau}.$$

При  $k=1 - \rho_{\Phi Y} = 8.125$ , а коефіцієнт використання УУ дорівнює 0.123 або 12.3%.

При  $k=2 - \rho_{\Phi Y} = 4.0625$ , а коефіцієнт використання УУ дорівнює 0.246 або 24.60%.

**Загальний час виконання одного потоку**

$$T_{3AG} = T_{OP} + T_{L3} + T_{OBY} = \frac{2n^2}{8} t_{OP} + \frac{n^3}{8} t_{L3} + \frac{7}{8} n^3 t_{CKL} = \frac{n^2}{4} t_{OP} + \frac{n^3}{8} t_{L3} + \frac{7}{8} n^3 t_{CKL}$$

**Загальний час зайнятості оперативної пам'яті при виконанні одного потоку**

$$T_{3AG}^{OP} = T_{OP}^{CT} + T_{OP}^{3P} = \frac{2n^2}{8} t_{OP} + n^2 t_{OP} = \frac{n^2}{4} t_{OP} + n^2 t_{OP} = 1.25n^2 t_{OP}.$$

**Коефіцієнт використання оперативної пам'яті**

$$KKD_{OP} = \frac{T_{3AG}^{OP}}{T_{3AG}} = \frac{T_{OP}^{CT} + T_{OP}^{3P}}{T_{3AG}} \cong \frac{1.25n^2 t_{OP}}{\frac{n^2}{4} t_{OP} + \frac{n^3}{8} t_{L3} + \frac{7}{8} n^3 t_{CKL}} \cong 0.077$$

**Коефіцієнт використання кеш-пам'яті L3**

$$KKD_{L3} = \frac{\frac{n^3}{8} t_{L3}}{\frac{2n^2}{8} t_{OP} + \frac{n^3}{8} t_{L3} + \frac{7}{8} n^3 t_{CKL}} \cong \frac{1}{\frac{2}{n} \frac{t_{OP}}{t_{L3}} + 1 + 7 \frac{t_{CKL}}{t_{L3}}} \cong 0.55$$

### 2.3.7 Масиви не поміщаються в кеш-пам'ять третього рівня

У кеш-пам'яті даних третього рівня L3 не поміщається блок стовпців матриці В. Робота йде весь час з ОП.

У цьому прикладі генеруються наступні типи груп команд:

ОП; ОП; Мнж; Скл.

L1; ОП; Мнж; Скл.

**Кількість груп команд кожного виду при роботі з 32-розрядними даними:**

$$N(\text{ОП; ОП; Мнж; Скл}) = n^2 * \frac{n}{16} = \frac{n^3}{16}$$

$$N(L1; \text{ОП; Мнж; Скл}) = n^3 - \frac{n^3}{16} = \frac{15}{16}n^3$$

**Ймовірності груп команд:**

$$P(\text{ОП; ОП; Мнж; Скл}) = \frac{N(\text{ОП; ОП; Мнж; Скл})}{n^3} = \frac{\frac{n^3}{16}}{n^3} = \frac{1}{16}$$

$$P(L1; \text{ОП; Мнж; Скл}) = \frac{N(L1; \text{ОП; Мнж; Скл})}{n^3} = \frac{\frac{15}{16}n^3}{n^3} = \frac{15}{16}$$

**Часи виконання груп команд при частоті ядра процесора  $f=4$  ГГц:**

тип ОП; ОП; Мнж; –  $2*256 = 512$  тактів.

тип L1; ОП; Мнж; – 256 тактів.

**Середній час виконання універсальним ФУ групи команд (1.2)**

$$t_{\text{ФУ}} \cong \frac{1}{16} * 512\tau + \frac{15}{16} * 256\tau = \frac{17}{16} * 256\tau = 272\tau$$

$$\rho_{\text{ФУ}} = \frac{272\tau}{k\tau}$$

При  $k=1$  –  $\rho_{\text{ФУ}} = 272$ , а коефіцієнт використання УУ дорівнює 0.0038 або 0.38%.

При  $k=2$  –  $\rho_{\text{ФУ}} = 136$ , а коефіцієнт використання УУ дорівнює 0.0076 або 0.76%.

**Загальний час виконання одного потоку**

$$T_{\text{ЗАГ}} = T_{\text{ОП}} + T_{\text{ЗП}} \cong n^3 t_{\text{ОП}} + \frac{n^2}{16} t_{\text{ОП}} + \frac{15}{16} n^2 t_{L1D} + n^2 t_{\text{ОП}}$$



При великих розмірах матриць (наприклад,  $n > 4096$ ) з відносною похибкою значно менше 1% можна прийняти, що

$$T_{ЗАГ} = T_{ОП} + T_{ЗП} \cong n^3 t_{ОП} + n^2 t_{ОП} \cong (n+1) * n^2 t_{ОП}.$$

**Загальний час зайнятості оперативної пам'яті.**

$$T_{ЗАГ}^{ОП} = T_{ОП}^{ЧТ} + T_{ОП}^{ЗП} \cong (n+1) * n^2 t_{ОП}$$

**Коефіцієнт використання оперативної пам'яті**

$$КПД_{ОП} = \frac{T_{ЗАГ}^{ОП}}{T_{ЗАГ}} = \frac{T_{ОП}^{ЧТ} + T_{ОП}^{ЗП}}{T_{ЗАГ}} \cong 1$$

При виконанні одного потоку коефіцієнт завантаження (ймовірність зайнятості) оперативної пам'яті дорівнює 1.

### 2.3.8 Масиви не поміщаються в кеш-пам'ять третього рівня

**Кількість груп команд кожного виду**

$$N(\text{ОП; ОП; Мнж; Скл}) = n^2 * \frac{n}{16} = \frac{n^3}{16}$$

$$N(L1; \text{ОП; Мнж; Скл}) = n^3 - \frac{n^3}{16} = \frac{15}{16} n^3$$

**Ймовірності груп команд:**

$$P(\text{ОП; ОП; Мнж; Скл}) = \frac{N(\text{ОП; ОП; Мнж; Скл})}{n^3} = \frac{\frac{n^3}{16}}{n^3} = \frac{1}{16}$$

$$P(L1; \text{ОП; Мнж; Скл}) = \frac{N(L1; \text{ОП; Мнж; Скл})}{n^3} = \frac{\frac{15}{16} n^3}{n^3} = \frac{15}{16}$$

**Часи виконання груп команд при частоті ядра процесора  $f=4$  ГГц:**

тип ОП; ОП; Мнж; –  $2*256=512$  тактів.

тип L1; ОП; Мнж; – 256 тактів.

Середній час виконання універсальним ФУ групи команд (1.2)

$$t_{\Phi Y} \cong \frac{1}{16} * 512\tau + \frac{15}{16} * 256\tau = \frac{17}{16} * 256\tau = 272\tau$$

$$\rho_{\Phi Y} = \frac{272\tau}{k\tau}$$

При  $k=1$  (більш продуктивний варіант УУ) –  $\rho_{\Phi Y} = 272$ , а коефіцієнт використання УУ дорівнює 0.0038 або 0.38%.

При  $k=2$  –  $\rho_{\Phi Y} = 136$ , а коефіцієнт використання УУ дорівнює 0.0076 або 0.76%.

**Загальний час виконання одного потоку**

$$T_{ЗАГ} = T_{ОП} + T_{ЗП} \cong n^3 t_{ОП} + \frac{n^2}{16} t_{ОП} + \frac{15}{16} n^2 t_{L1D} + n^2 t_{ОП}$$

При великих розмірах матриць (наприклад,  $n > 4096$ ) з відносною похибкою значно менше 1% можна прийняти, що

$$T_{ЗАГ} = T_{ОП} + T_{ЗП} \cong n^3 t_{ОП} + n^2 t_{ОП} \cong (n+1) * n^2 t_{ОП}$$

**Загальний час зайнятості оперативної пам'яті.**

$$T_{ЗАГ}^{ОП} = T_{ОП}^{ЧТ} + T_{ОП}^{ЗП} \cong (n+1) * n^2 t_{ОП}$$

**Коефіцієнт використання оперативної пам'яті**

$$КПД_{ОП} = \frac{T_{ЗАГ}^{ОП}}{T_{ЗАГ}} = \frac{T_{ОП}^{ЧТ} + T_{ОП}^{ЗП}}{T_{ЗАГ}} \cong 1$$

При виконанні одного потоку коефіцієнт завантаження (ймовірність зайнятості) оперативної пам'яті 1.

## 2.4 Множення матриці при одночасному виконанні 2-х потоків

### 2.4.1 Обидва масиви поміщаються в кеш-пам'ять даних першого рівня L1D

При одночасному запуску двох однакових потоків програми «Множення матриць» в двох віртуальних ядрах буде виконано:

$$\frac{2*n^2}{16} + \frac{2*n^2}{16} = \frac{n^2}{4}$$

звернень в ОП за читанням матриць-операндів.

Читання з ОП як і раніше виконується блоками по 64 байта. Так як матриці складаються з 32-х розрядних даних (4 байта), то в одному блоці відразу виходить 16 елементів даних.

Загальний час виконання груп команд зі зверненням в ОП

$$T_{ОП} = \frac{n^2}{4} t_{ОП}.$$

Частина команд звернення до кеш-пам'ять першого рівня L1D, Множення і Складання виконуються паралельно зі зверненням в ОП, тому в загальному часу виконання груп команд (L1; L1; Мнж; Скл) не враховуються.

**Загальна кількість груп команд (L1; L1; Мнж; Скл)**

$$N = (2*(n^3 - 2*n^2/16)) = 2(n^3 - n^2/8).$$

**Загальний час виконання груп команд (L1; L1; Мнж; Скл) 2-х потоків**

$$T_{ОБЧ} = 2*(n^3 - n^2/8) * t_{ОБЧ}.$$

Так як 2 групи команд (L1; ОП; Мнж; Скл) різних потоків виконуються паралельно із зсувом в 1 такт, то за час виконання операції додавання, рівне 4 тактів виконається 2 групи команд (L1; ОП; Мнж; Скл). Це означає, що пропускна здатність (інтенсивність) при виконанні однієї групи команд (L1; L1; Мнж; Скл) - 1 група команд за 2 такти.

Якщо у фізичному ядрі виконується тільки один потік програми «Множення матриць», то інтенсивність виконання однієї групи команд (L1; ОП; Мнж; Скл) буде дорівнювати 1 група команд (L1; L1; Мнж; Скл) за 4 такти.

Приймається, що  $t_{ОБЧ} = 2\tau$  при одночасному виконанні в підсистемі обчислення 2 груп команд ( L1; ОП; Мнж; Скл).

Якщо у фізичному ядрі виконується тільки один потік програми «Множення матриць», то інтенсивність виконання однієї групи команд (L1; L1; Мнж; Скл) буде дорівнювати 1 група команд за 4 такти, а  $t_{ОБЧ} = 4\tau$ .

### **$2n^2$ операцій записи елементів матриць результату.**

Якщо розмірність матриць, то запис одного  $i$ -го елемента результату може бути виконаний паралельно з обчисленням чергового ( $i+1$ )-го елемента результату. уже при  $n=64$  час обчислення чергового елемента матриці результату і час запису елемента в ОП близькі. Це означає, що при виконанні 2-х потоків запис в ОП буде поєднуватися з обчисленнями тільки для половини обчислених елементів результатів. Інша половина обчислених елементів результатів для обох потоків (а це становить  $n$  елементів) буде записана за час

$$T_{ЗП} = n^2 * t_{ОП}.$$

**Ймовірності груп команд при виконанні 2-х потоків такі ж, як і при виконанні одного потоку:**

$$P(L1; ОП; Мнж; Скл) = \frac{2 * N(L1; ОП; Мнж; Скл)}{2 * n^3} \cong \frac{2 * 2 * n^2}{16 * 2 * n^3} \cong \frac{1}{8 * n}$$

$$P(L1; L1; Мнж; Скл) = \frac{2 * N(L1; L1; Мнж; Скл)}{2 * n^3} \cong \frac{2 * (n^3 - 2 * n^2 / 16)}{2 * n^3} = \left(1 - \frac{1}{8 * n}\right)$$

**Часи виконання груп команд при частоті ядра процесора  $f = 4$  ГГц при виконанні 2-х потоків такі ж, як і виконанні одного потоку:**

тип L1; ОП; Мнж; Скл - 256 тактів;

тип L1; L1; Мнж; Скл - 4 такти.

За час виконання  $n$  циклів груп команд L1; L1; Мнж; Скл буде сформований один елемент результату матриці для одного потоку. Цей час  $T_{ЕЛ} = n * t_{СЛЖ}$ . Два потоки виконуються одночасно (паралельно). Апаратура фізичного ядра дозволяє виконувати одночасно 2 групи команд (L1; L1; Мнж; Скл) по одній групі команд (L1; L1; Мнж; Скл) для кожного потоку. У підсумку за час  $T_{ЕЛ} = n * t_{СКЛ}$  можуть бути сформовані 2 результати. Однак ці 2 результати треба записати в ОП. Час запису одного результату в ОП -  $T_{ОП}^{ЗП} = t_{ОП}$ . Якщо час обчислення чергової пари результатів для 2-х потоків більше  $2 * t_{ОП}$ , то запис пари результатів може поєднуватися з обчисленням чергової пари результатів і час запису можна не

враховувати. В іншому випадку час обчислення пари результатів буде визначатися часом їх запису в ОП.

При  $n = 64$  час обчислення пари результатів буде визначатися часом їх запису в ОП, тобто

$$T_{ОП}^{3П} = 2 * t_{ОП}$$

Так як 2 групи команд L1; L1; Мнж; Скл виконуються одночасно (в конвеєрі із зсувом на 1 такт), то пропускна здатність при виконанні однієї групи команд L1; L1; Мнж; Скл складе

$$\mu_{гр} = \frac{1}{t_{СКЛ}/2}$$

А час виконання однієї групи команд

$$t_{сп} = t_{СКЛ}/2$$

Середній час виконання універсальним ФУ групи команд (1.2)

$$t_{ФУ} = \frac{1}{g * n} * t_{ОП} + (1 - \frac{1}{g * n}) * t_{ОБЧ} \cong 2.5\tau$$

$$\rho_{ФУ} = \frac{\lambda_{УУ}}{\mu_{ФУ}} = \frac{t_{ФУ}}{t_{УУ}} = \frac{2.5\tau}{k\tau}$$

При  $k=1 - \rho_{ФУ} \cong 2.5$ , а коефіцієнт використання УУ дорівнює 0.4 або 40%.

При  $k=2 - \rho_{ФУ} = 1.25$ , а коефіцієнт використання УУ дорівнює 0.8 або 80%.

#### **Загальний час виконання 2-х потоків**

$$T_{ЗАГ} = 2T_{ОП} + T_{ОБЧ} = 2 * \frac{2n^2}{16} t_{ОП} + 2n^2 t_{ОП} = 2.25 * n^2 t_{ОП}$$

$$T_{ОБЧ} = 2n^3 t_{СКЛ}/2 = 2n^2 t_{ОП}$$

#### **Загальний час зайнятості оперативної пам'яті**

$$T_{ЗАГ}^{ОП} = T_{ОП}^{ЧТ} + T_{ОП}^{3П} = 2 \left( \frac{2n^2}{16} t_{ОП} + n^2 t_{ОП} \right) = 2.25n^2 t_{ОП}$$

де  $T_{ОП}^{3П} = n^2 * t_{ОП}$  - час запису в ОП одиночних елементів результатів однієї матриці.

$T_{ОБЧ}^{1Э} = n^3 t_{СКЛ}$  - час обчислення одиночних елементів результатів однієї матриці.

При  $n=64$  и  $t_{ОП} = 256\tau$  вони рівні.

При  $n = 64$  запис елементів матриці С виконується під час обчислення чергового елемента матриці С. А для другого потоку запис чергового елемента результату не поєднується з обчисленнями. Тому в загальний час виконання 2-х потоків додається час запису елементів результатів 2-го потоку.

#### **Коефіцієнт використання оперативної пам'яті**

$$ККД_{ОП} = \frac{T_{ЗАГ}^{ОП}}{T_{ЗАГ}} = \frac{T_{ОП}^{ЧТ} + T_{ОП}^{3П}}{T_{ЗАГ}} \cong 1$$

#### **Коефіцієнт прискорення при паралельному виконанні 2-х потоків**

$$S = \frac{T_{ЗАГ}^{ПОСЛ}}{T_{ЗАГ}^{ПАР}} = \frac{2\left(\frac{2n^2}{16} t_{ОП} + n^2 t_{ОП}\right)}{2 * \frac{2n^2}{16} t_{ОП} + 2n^2 * t_{ОП}} = 1$$

#### **2.4.2 Обидва масиву поміщаються в кеш-пам'ять даних першого рівня L1D.**

При одночасному запуску двох однакових потоків програми «Множення матриць» в двох віртуальних ядрах буде виконано:

$\left(\frac{2*n^2}{8} + \frac{2*n^2}{8}\right)$  звернень в ОП за читанням матриць-операндів.

Читання з ОП виконується блоками по 64 байти. Так як матриці складаються з 64-х розрядних даних (8 байт), то в одному блоці відразу зчитується 8 елементів даних.

Команди звернення в кеш-пам'ять першого рівня L1D, Множення і Складання виконується паралельно зі зверненням в ОП, тому в загальному часі виконання груп команд (L1; ОП; Мнж; Скл;) не враховуються.

#### **Загальний час виконання груп команд зі зверненням в ОП**

$$T_{ОП} = \frac{2 * n^2}{8} t_{ОП} + \frac{2 * n^2}{8} t_{ОП} = \frac{4 * n^2}{8} t_{ОП} = \frac{n^2}{2} t_{ОП}$$

### Загальна кількість груп команд (L1; L1; Мнж; Скл)

$$(n^3 - 2 * n^2 / 8) + (n^3 - 2 * n^2 / 8) = \\ = 2 * (n^3 - 2 * n^2 / 8) = 2 * (n^3 - n^2 / 4) = 2 * n^2 (n - 1/4) \cong 2 * n^3$$

### Час виконання груп команд (L1; L1; Мнж; Скл) 2-х потоків

2 групи команд (L1; L1; Мнж; Скл) різних потоків виконуються паралельно із зсувом в 1 такт. За час виконання операції додавання виконається паралельно 2 групи команд (L1; L1; Мнж; Скл). Це означає, що інтенсивність виконання однієї групи команд (L1; L1; Мнж; Скл) становить 1 група команд (L1; L1; Мнж; Скл) за 2 такти.

Якщо у фізичному ядрі виконується тільки один потік програми «Множення матриць», то інтенсивність виконання однієї групи команд (L1; L1; Мнж; Скл) буде дорівнює 1 група команд (L1; L1; Мнж; Скл) за 4 такти.

Загальна кількість операцій запису елементів двох матриць результату -  $2n^2$ .

**Загальний час виконання операцій запису елементів двох матриць результатів в ОП**

$$T_{3П} = 2n^2 t_{ОП}$$

**Ймовірності груп команд при виконанні 2-х потоків такі ж, як і при виконанні одного потоку:**

$$P(L1; ОП; Мнж; Скл) = \frac{2 * N(L1; ОП; Мнж; Скл)}{2 * n^3} \cong \frac{2 * 2 * n^2}{16 * 2 * n^3} \cong \frac{1}{8 * n} \\ P(L1; L1; Мнж; Скл) = \frac{2 * N(L1; L1; Мнж; Скл)}{2 * n^3} \cong \frac{2 * (n^3 - 2 * n^2 / 16)}{2 * n^3} = (1 - \frac{1}{8 * n})$$

**Часи виконання груп команд при частоті ядра процесора  $f = 4$  ГГц при виконанні 2-х потоків такі ж, як і виконанні одного потоку:**

тип L1; ОП; Мнж; Скл - 256 тактів;

тип L1; L1; Мнж; Скл - 4 такти.

За час виконання  $n$  циклів груп команд L1; L1; Мнж; Скл буде сформований один елемент результату матриці для одного потоку. Цей час  $T_{ЭЛ} = n * t_{СЛЖ}$ . Два потоки виконуються одночасно (паралельно). Апаратура фізичного ядра дозволяє виконувати одночасно 2 групи команд (L1; L1; Мнж; Скл) по одній групі команд (L1; L1; Мнж; Скл) для кожного потоку. У підсумку за час  $T_{ЕЛ} = n * t_{СКЛ}$  можуть бути сформовані 2 результати. Однак

ці 2 результати треба записати в ОП. Час запису одного результату в ОП -  $T_{ОП}^{3П} = t_{ОП}$ . Якщо час обчислення чергової пари результатів для 2-х потоків більше  $2 * t_{ОП}$ , то запис пари результатів може поєднуватися з обчисленням чергової пари результатів і час запису можна не враховувати. В іншому випадку час обчислення пари результатів буде визначатися часом їх запису в ОП.

При  $n = 64$  час обчислення пари результатів якраз і буде визначатися часом їх запису в ОП, тобто

$$T_{ОП}^{3П} = 2 * t_{ОП}$$

Так як 2 групи команд L1; L1; Мнж; Скл виконуються одночасно (в конвеєрі із зсувом на 1 такт), то пропускна здатність при виконанні однієї групи команд L1; L1; Мнж; Скл складе

$$\mu_{гр} = \frac{1}{t_{СКЛ}/2}$$

А час виконання однієї групи команд

$$t_{гр} = t_{СКЛ}/2$$

Середній час виконання універсальним ФУ групи команд(1.2)

$$t_{ФУ} = \frac{1}{g*n} * t_{ОП} + (1 - \frac{1}{g*n}) * t_{ОБЧ} \cong 2.5\tau$$

$$\rho_{ФУ} = \frac{2.5\tau}{k\tau}$$

При  $k=1 - \rho_{ФУ} \cong 2.5$ , а коефіцієнт використання УУ дорівнює 0.4 або 40%.

При  $k=2 - \rho_{ФУ} = 1.25$ , а коефіцієнт використання УУ дорівнює 0.8 або 80%.

**Загальний час виконання 2-х потоків**

$$T_{ЗАГ} = 2T_{ОП} + T_{ОБЧ} = 2 * \frac{2n^2}{16} t_{ОП} + 2n^2 t_{ОП} = \frac{n^2}{4} t_{ОП} + 2n^2 t_{ОП} = 2.25 * n^2 t_{ОП}$$

$$T_{ОБЧ} = 2n^3 t_{СЛЖ}/2 = 2n^2 t_{ОП}$$



### Загальний час зайнятості оперативної пам'яті

$$T_{ЗАГ}^{ОП} = T_{ОП}^{ЧТ} + T_{ОП}^{ЗП} = 2\left(\frac{2n^2}{16}t_{ОП} + n^2t_{ОП}\right) = 2.25n^2t_{ОП},$$

де  $T_{ОП}^{ЗП} = n^2 * t_{ОП}$  - час запису в ОП одиночних елементів результатів однієї матриці.

$T_{ОБЧ}^{АЕ} = n^3t_{СКЛ}$  - час обчислення одиночних елементів результатів однієї матриці.

При  $n=64$  и  $t_{ОП} = 256\tau$  вони рівні.

При  $n = 64$  запис елементів матриці С виконується під час обчислення чергового елемента матриці С. А для другого потоку запис чергового елемента результату не поєднується з обчисленнями. Тому в загальний час виконання 2-х потоків додається час запису елементів результатів 2-го потоку.

### Коефіцієнт використання оперативної пам'яті

$$ККД_{ОП} = \frac{T_{ЗАГ}^{ОП}}{T_{ЗАГ}} = \frac{T_{ОП}^{ЧТ} + T_{ОП}^{ЗП}}{T_{ЗАГ}} \cong 1$$

### Коефіцієнт прискорення при паралельному виконанні 2-х потоків

$$S = \frac{T_{ЗАГ}^{ПОСЛ}}{T_{ЗАГ}^{ПАР}} = \frac{2\left(\frac{2n^2}{16}t_{ОП} + n^2t_{ОП}\right)}{2 * \frac{2n^2}{16}t_{ОП} + 2n^2 * t_{ОП}} = 1$$

#### 2.4.3 Обидва масиви поміщаються в кеш-пам'ять даних другого рівня

Матриця В і, принаймні, один поточний рядок матриці А поміщаються в кеш-пам'яті другого рівня L2.

У цьому прикладі генеруються наступні типи груп команд:

- ОП; ОП; Мнж; Скл.
- L1; ОП; Мнж; Скл.
- L1; L2; Мнж; Скл.
- L1; L1; Мнж; Скл.

Як і раніше, група команд ОП; ОП; Мнж; Скл замінюється еквівалентним збільшенням кількості груп команд L1; ОП; Мнж; Скл і в подальшому не враховується.

Разом, враховуються такі типи груп команд:

- L1; ОП; Мнж; Скл.
- L1; L2; Мнж; Скл.
- L1; L1; Мнж; Скл.

**Кількість груп команд кожного виду:**

$$N(L1; ОП; Мнж; Скл) = 2 * \frac{2n^2}{16}$$

$$N(L1; L2; Мнж; Скл) = 2 * \frac{n^2}{16} * (n - 1) = 2(\frac{n^3}{16} - \frac{n^2}{16})$$

$$N(L1; L1; Мнж; Скл) = 2[n^3 - \frac{2n^2}{16} - \frac{n^2}{16} * (n - 1)] = 2[\frac{15 * n^3}{16} - \frac{n^2}{16}] \cong 2 * \frac{15 * n^3}{16}$$

**Ймовірності груп команд наступні:**

$$P(L1; ОП; Мнж; Скл) = \frac{N(L1; ОП; Мнж; Скл)}{2n^3} = \frac{2 * 2n^2}{16 * 2n^3} = \frac{1}{8 * n}$$

$$P(L1; L2; Мнж; Скл) = \frac{N(L1; L2; Мнж; Скл)}{2n^3} = \frac{2[n^2 * (n - 1)]}{16 * 2n^3} \cong \frac{n - 1}{16 * n} \cong \frac{1}{16}$$

$$P(L1; L1; Мнж; Скл) = \frac{N(L1; L1; Мнж; Скл)}{2n^3} = 2[\frac{15 * n^3}{16 * 2n^3} - \frac{n^2}{16 * 2n^3}] = \frac{15}{16} - \frac{1}{16n} \cong \frac{15}{16}$$

**Часи виконання груп команд при частоті ядра процесора f=4 ГГц:**

- тип L1; L1; Мнж; Скл – 4 такти;
- тип L1; L2; Мнж; Скл – 12 тактів;
- тип L1; ОП; Мнж; Скл – 256 тактів.

Середній час виконання універсальним ФУ групи команд (1.2)

$$t_{\Phi Y} = \frac{1}{8 * n} * t_{ОП} + \frac{1}{16} t_{L2} + (1 - \frac{1}{8 * n} - \frac{1}{16}) * t_{ОБЧ} \cong 2.875 \tau$$

$$\rho_{\Phi Y} = \frac{2.875 \tau}{k \tau}$$

При k=1 –  $\rho_{\Phi Y} = 2.875$ , а коефіцієнт використання УУ дорівнює **0.4 або 40%**.

При k=2 –  $\rho_{\Phi Y} = 2.375$ , а коефіцієнт використання УУ дорівнює 0.35 або 35 %.

### Загальний час послідовного виконання 2-х потоків

$$T_{ЗАГ}^{ПОСЛ} = 2(T_{ОП} + T_{L2} + T_{ОБЧ}) \cong 2\left(\frac{2n^2}{16}t_{ОП} + \frac{n^3}{16}t_{L2} + \frac{15}{16}n^3t_{ОБЧ}\right)$$

$$t_{ОБЧ} = t_{СКЛ}$$

### Загальний час паралельного виконання 2-х потоків

$$T_{ЗАГ}^{ПАР} = T_{ОП} + T_{L2} + T_{ОБЧ} \cong 2\left(\frac{2n^2}{16}t_{ОП} + \frac{n^3}{16}t_{L2} + \frac{15}{16}n^3t_{ОБЧ}\right)$$

$$t_{ОБЧ} = 0.5t_{СКЛ}$$

### Коефіцієнт прискорення при виконанні 2 потоків

$$S = \frac{T_{ЗАГ}^{ПОСЛ}}{T_{ЗАГ}^{ПАР}} = \frac{2\left(\frac{2n^2}{16}t_{ОП} + \frac{n^3}{16}t_{L2} + \frac{15}{16}n^3t_{СКЛ}\right)}{2\left(\frac{2n^2}{16}t_{ОП} + \frac{n^3}{16}t_{L2} + \frac{15}{16}n^3\frac{t_{СКЛ}}{2}\right)} \cong 1.65$$

### Загальний час зайнятості оперативної пам'яті при виконанні 2-х потоків

$$T_{ЗАГ}^{ОП} = T_{ОП}^{ЧТ} + T_{ОП}^{ЗП} = 2\left(\frac{2n^2}{16}t_{ОП} + n^2 * t_{ОП}\right)$$

### Коефіцієнт використання оперативної пам'яті при виконанні 2-х потоків

$$ККД_{ОП} = \frac{T_{ЗАГ}^{ОП}}{T_{ЗАГ}} = \frac{T_{ОП}^{ЧТ} + T_{ОП}^{ЗП}}{T_{ЗАГ}} \cong \frac{2\left(\frac{2n^2}{16}t_{ОП} + n^2t_{ОП}\right)}{2\left(\frac{2n^2}{16}t_{ОП} + \frac{n^3}{16}t_{L2} + \frac{15}{16}n^3t_{ОБЧ}\right)}$$

Після перетворення будемо мати

$$ККД_{ОП} \cong \frac{1}{1/9 + \frac{n}{18}\frac{t_{L2}}{t_{ОП}} + n\frac{15}{18}\frac{t_{ОБЧ}}{t_{ОП}}} \cong \frac{18}{2 + n\frac{t_{L2}}{t_{ОП}} + 15n\frac{t_{ОБЧ}}{t_{ОП}}} \cong 0.42$$

## Коефіцієнт використання кеш-пам'яті L2 при виконанні 2-х потоків

$$KK_{L2} = \frac{T_{3AG}^{L2}}{T_{3AG}} \cong \frac{2 \frac{n^3}{16} t_{L2}}{2 \left( \frac{2n^2}{16} t_{OP} + \frac{n^3}{16} t_{L2} + n^3 t_{СКЛ} \right)} \cong 0,15$$

Більше двох програм Множення матриць з  $n = 128$  запускати одночасно недоцільно через можливе перенавантаження оперативної пам'яті.

### 2.4.4 Обидва масиви поміщаються в кеш-пам'ять даних другого рівня

Ймовірності груп команд:

$$P(OP; L1; Mнж; СКЛ) = \frac{N(OP; L1; Mнж; СКЛ)}{2n^3} \cong 2 * \frac{2n^2}{8 * 2n^3} \cong \frac{1}{4n}$$

$$P(L1; L1; Mнж; СКЛ) = \frac{N(L1; L1; Mнж; СКЛ)}{2n^3} \cong \frac{2[n^3 - 2 * n^2/8]}{2n^3} \cong \left(1 - \frac{1}{4 * n} - \frac{1}{8}\right) \cong \frac{7}{8}$$

$$P(L1; L2; Mнж; СКЛ) = \frac{N(L1; L2; Mнж; СКЛ)}{2n^3} = \frac{2[n^2 * (n - 1)]}{8 * 2n^3} \cong \frac{n - 1}{8n} \cong \frac{1}{8}$$

Часи виконання груп команд не змінюються в порівнянні з варіантом, коли елементи матриць це 32-розрядні дані:

Середній час виконання універсальним ФУ групи команд (1.2)

$$t_{ФУ} = \frac{1}{4 * n} * t_{OP} + \frac{1}{8} * t_{L2} + \left(1 - \frac{1}{4 * n} - \frac{1}{8}\right) * t_{ОБЧ} \cong 3.75\tau$$

$$\rho_{ФУ} = \frac{3.75\tau}{k\tau}$$

При  $k=1$  –  $\rho_{ФУ} = 3.75$ , а коефіцієнт використання УУ дорівнює **0.27 або 27%**.

При  $k=2$  –  $\rho_{ФУ} = 1.875$ , а коефіцієнт використання УУ дорівнює 0.54 або 54 %.

### Загальний час послідовного виконання 2-х потоків

$$T_{3AG}^{ПОСЛ} = 2(T_{OP} + T_{L2} + T_{ОБЧ}) \cong 2 \left( \frac{2n^2}{8} t_{OP} + \frac{n^3}{8} t_{L2} + \frac{7}{8} n^3 t_{ОБЧ} \right)$$

$$t_{ОБЧ} = t_{СКЛ}$$

### Загальний час паралельного виконання 2-х потоків

$$T_{ЗАГ}^{ПАР} = T_{ОП} + T_{L2} + T_{ОБЧ} \cong 2 \left( \frac{2n^2}{8} t_{ОП} + \frac{n^3}{8} t_{L2} + \frac{7}{8} n^3 t_{ОБЧ} \right)$$

$$t_{ОБЧ} = 0.5 t_{СКЛ}$$

### Коефіцієнт прискорення при виконанні 2 потоків

$$S = \frac{T_{ЗАГ}^{ПОСЛ}}{T_{ЗАГ}^{ПАР}} \cong 1,46$$

### Загальний час зайнятості оперативної пам'яті при виконанні 2-х потоків

$$T_{ЗАГ}^{ОП} = T_{ОП}^{ЧТ} + T_{ОП}^{ЗП} = 2 \left( \frac{2n^2}{8} t_{ОП} + n^2 * t_{ОП} \right) \cong \frac{n^2}{2} t_{ОП} + 2n^2 t_{ОП} \cong 2.5n^2 t_{ОП}$$

### Коефіцієнт використання оперативної пам'яті

$$ККД_{ОП} = \frac{T_{ЗАГ}^{ОП}}{T_{ЗАГ}} = \frac{T_{ОП}^{ЧТ} + T_{ОП}^{ЗП}}{T_{ЗАГ}} \cong \frac{2.5n^2 t_{ОП}}{\frac{1}{4} (2n^2 t_{ОП} + n^3 t_{L2} + 7n^3 t_{ОБЧ})}$$

Після спрощень отримаємо

$$ККД_{ОП} \cong \frac{1}{0.2 + 0.1n \frac{t_{L2}}{t_{ОП}} + 0.7n \frac{t_{ОБЧ}}{t_{ОП}}} \cong 0.37.$$

### Коефіцієнт використання кеш-пам'яті L2 при виконанні 2-х потоків

$$ККД_{L2} = \frac{T_{ЗАГ}^{L2}}{T_{ЗАГ}} \cong \frac{2 \frac{n^3}{8} t_{L2}}{\frac{1}{4} (2n^2 t_{ОП} + n^3 t_{L2} + 7n^3 t_{ОБЧ})} \cong \frac{1}{\frac{2t_{ОП}}{t_{L2}n} + 1 + 7 \frac{t_{СКЛ}}{t_{L2}}} \cong 0.27.$$

### 2.4.5 Обидва масиви вміщуються в кеш-пам'ять третього рівня

Матриця В і, принаймні, один поточний рядок матриці А поміщаються в кеш-пам'ять третього рівня L3.

У цьому прикладі генеруються наступні типи груп команд:

- • ОП; ОП; Мнж; Скл.
- • L1; ОП; Мнж; Скл.
- • L1; L3; Мнж; Скл.
- • L1; L1; Мнж; Скл.

Як і в попередніх прикладах група команд ОП; ОП; Мнж; Скл замінюється еквівалентним збільшенням кількості груп команд L1; ОП; Мнж; Скл і в подальшому не враховується.

Разом, враховуються такі типи груп команд:

- L1; ОП; Мнж; Скл.
- L1; L3; Мнж; Скл.
- L1; L1; Мнж; Скл.

**Кількість груп команд кожного виду:**

$$N(L1; ОП; Мнж; Скл) = 2 \frac{2n^2}{16} = \frac{n^2}{4}$$

$$N(L1; L3; Мнж; Скл) = 2 * \left[ \frac{n^2}{16} * (n - 1) \right] = 2 * \left( \frac{n^3}{16} - \frac{n^2}{16} \right) \cong \frac{n^3}{8}$$

$$N(L1; L1; Мнж; Скл) = 2 * \left[ n^3 - \frac{2n^2}{16} - \frac{n^2}{16} * (n - 1) \right] \cong \frac{15n^3}{8}$$

**Ймовірності груп команд наступні:**

$$P(L1; ОП; Мнж; Скл) = \frac{N(L1; ОП; Мнж; Скл)}{2n^3} = \frac{2 * 2n^2}{16 * 2n^3} = \frac{1}{8 * n}$$

$$P(L1; L3; Мнж; Скл) = \frac{N(L1; L3; Мнж; Скл)}{2n^3} = \frac{2n^2 * (n - 1)}{16 * 2n^3} \cong \frac{n - 1}{16 * n} \cong \frac{1}{16}$$

$$P(L1; L1; Мнж; Скл) = \frac{N(L1; L1; Мнж; Скл)}{n^3} = \frac{2 * 15 * n^3}{16 * 2n^3} - \frac{2n^2}{16 * 2n^3} = \frac{15}{16} - \frac{1}{16n} \cong \frac{15}{16}$$

**Часи виконання груп команд при частоті ядра процесора f=4 ГГц:**

- тип L1; L1; Мнж; Скл – 4 такти;

- тип L1; L3; Мнж; Скл – 36 тактів;
- тип L1; ОП; Мнж; Скл – 256 тактів.

Середній час виконання універсальним ФУ групи команд (1.2)

$$t_{\Phi Y} = \frac{1}{8*n} * t_{OP} + \frac{1}{16} t_{L3} + (1 - \frac{1}{8*n} - \frac{1}{16}) * t_{OBY} \cong 4.19\tau$$

$$\rho_{\Phi Y} = \frac{4.19\tau}{k\tau}$$

При k=1 –  $\rho_{\Phi Y} = 4.19$ , а коефіцієнт використання УУ дорівнює **0.24 або 24%**.

При k=2 –  $\rho_{\Phi Y} = 2.095$ , а коефіцієнт використання УУ дорівнює **0.48 або 48 %**.

**Загальний час виконання одного потоку**

$$T_{OBY} = T_{OP} + T_{L3} + T_{OBY} \cong \frac{2n^2}{16} t_{OP} + \frac{n^3}{16} t_{L3} + \frac{15}{16} n^3 t_{СКЛ}$$

**Загальний час послідовного виконання 2-х потоків**

$$T_{ЗАГ}^{ПОСЛ} = 2T_{ЗАГ} \cong 2\left(\frac{2n^2}{16} t_{OP} + \frac{n^3}{16} t_{L3} + n^3 \frac{15}{16} t_{СКЛ}\right) \cong \frac{n^2}{4} t_{OP} + \frac{n^3}{8} t_{L3} + \frac{15}{8} n^3 t_{СКЛ}$$

**Загальний час паралельного виконання 2-х потоків**

$$T_{ЗАГ}^{ПАР} \cong 2\left(\frac{2n^2}{16} t_{OP} + \frac{n^3}{16} t_{L3} + \frac{15}{16} n^3 t_{СКЛ}/2\right) \cong \frac{n^2}{4} t_{OP} + \frac{n^3}{8} t_{L3} + \frac{15}{8} n^3 t_{СКЛ}/2$$

**Коефіцієнт прискорення при паралельному виконанні 2 потоків**

$$S = \frac{T_{ЗАГ}^{ПОСЛ}}{T_{ЗАГ}^{ПАР}} \cong 1,45$$

**Загальний час зайнятості оперативної пам'яті при виконанні одного потоку**

$$T_{ЗАГ}^{ОП} = T_{ОП}^{ЧТ} + T_{ОП}^{ЗП} = \frac{2n^2}{16} t_{OP} + n^2 t_{OP} = 1.125 t_{OP}$$

### Загальний час зайнятості оперативної пам'яті при виконанні 2-х потоків

$$T_{ОП}^{ПАР} \cong 2\left(\frac{2n^2}{8}t_{ОП} + n^2t_{ОП}\right) = 2.25t_{ОП}$$

### Загальний час паралельного виконання 2-х потоків

$$T_{ЗАГ}^{ПАР} = 2(T_{ОП} + T_{L3} + T_{ОБЧ}) = 2\left(\frac{2n^2}{16}t_{ОП} + \frac{n^3}{16}t_{L3} + \frac{15}{16}n^3t_{СКЛ}/2\right)$$

### Коефіцієнт використання оперативної пам'яті при виконанні 2-х потоків

$$ККД_{ОП} = \frac{T_{ЗАГ}^{ОП}}{T_{ЗАГ}^{ПАР}} = \frac{2.25n^2t_{ОП}}{\frac{n^2}{4}t_{ОП} + \frac{n^3}{8}t_{L3} + \frac{15}{8}n^3t_{СКЛ}/2} = \frac{18}{2 + n\frac{t_{L3}}{t_{ОП}} + 15n\frac{t_{СКЛ}}{t_{ОП}}} \cong 0.134$$

### Коефіцієнт використання кеш-пам'яті L3 при виконанні 2-х потоків

$$ККД_{L3} = \frac{T_{ЗАГ}^{L3}}{T_{ЗАГ}^{ПАР}} \cong \frac{\frac{n^3}{16}t_{L3}}{\frac{2n^2}{16}t_{ОП} + \frac{n^3}{16}t_{L3} + \frac{15}{16}n^3t_{СКЛ}} \cong 0.537$$

### 2.4.6 Обидва масиви вміщуються в кеш-пам'ять третього рівня

#### Ймовірності груп команд:

$$P(ОП; L1; Мнж; Скл) = \frac{2*N(ОП; L1; Мнж; Скл)}{2n^3} \cong 2 * \frac{2*n^2}{8*2n^3} \cong \frac{1}{4*n}$$

$$P(L1; L1; Мнж; Скл) = \frac{2*N(L1; L1; Мнж; Скл)}{2n^3} \cong \frac{2[n^3 - 2*n^2/8]}{2n^3} \cong \left(1 - \frac{1}{4*n} - \frac{1}{8}\right) \cong \frac{7}{8}$$

$$P(L1; L3; Мнж; Скл) = \frac{2*N(L1; L3; Мнж; Скл)}{2n^3} = 2 * \frac{[n^2*(n-1)]}{8*2n^3} \cong \frac{n-1}{8n} \cong \frac{1}{8}$$

Часи виконання груп команд не змінюються в порівнянні з варіантом, коли елементи матриць це 32-розрядні дані а частота ядра процесора  $f = 4$  ГГц.



Середній час виконання універсальним ФУ групи команд (1.2)

$$t_{\Phi Y} = \frac{1}{4*n} * t_{OП} + \frac{1}{8} * t_{L3} + \left(1 - \frac{1}{4*n} - \frac{1}{8}\right) * t_{OБЧ} \cong 8.125\tau$$

$$\rho_{\Phi Y} = \frac{8.125\tau}{k\tau}$$

При  $k=1$  –  $\rho_{\Phi Y} = 8.125$ , а коефіцієнт використання УУ дорівнює **0.123 або 12.3%**.

При  $k=2$  –  $\rho_{\Phi Y} = 4.0625$ , а коефіцієнт використання УУ дорівнює 0.246 або 24.60 %.

**Загальний час виконання одного потоку**

$$T_{ЗАГ} \cong T_{OП} + T_{L3} + T_{OБЧ} \cong \frac{2n^2}{8} t_{OП} + \frac{n^3}{8} t_{L3} + \frac{7}{8} n^3 t_{СКЛ} \cong$$

$$\cong \frac{n^2}{4} t_{OП} + \frac{n^3}{8} t_{L3} + \frac{7}{8} n^3 t_{СКЛ}$$

**Загальний час паралельного виконання 2-х потоків**

$$T_{ЗАГ}^{ПАР} = 2(T_{OП} + T_{L3} + T_{OБЧ}) = 2\left(\frac{2n^2}{8} t_{OП} + \frac{n^3}{8} t_{L3} + \frac{7}{8} n^3 t_{СКЛ}/2\right)$$

**Загальний час послідовного виконання 2-х потоків**

$$T_{ЗАГ}^{ПОСЛ} = 2(T_{OП} + T_{L3} + T_{OБЧ}) = 2\left(\frac{2n^2}{8} t_{OП} + \frac{n^3}{8} t_{L3} + \frac{7}{8} n^3 t_{СКЛ}\right)$$

**Коефіцієнт прискорення при паралельному виконанні 2-х потоків**

$$S = \frac{T_{ЗАГ}^{ПОСЛ}}{T_{ЗАГ}^{ПАР}} \cong \frac{2\left(\frac{n^2}{4} t_{OП} + \frac{n^3}{8} t_{L3} + \frac{7}{8} n^3 t_{СКЛ}\right)}{2\left(\frac{n^2}{4} t_{OП} + \frac{n^3}{8} t_{L3} + \frac{7}{8} n^3 t_{СКЛ}/2\right)} \cong \frac{2n^2 t_{OП} + n^3 t_{L3} + 7n^3 t_{СКЛ}}{2n^2 t_{OП} + n^3 t_{L3} + \frac{7n^3 t_{СКЛ}}{2}} \cong 1,27$$

### Загальний час зайнятості оперативної пам'яті при виконанні одного потоку

$$T_{ОП}^{1ППОТ} \cong \frac{2n^2}{8} t_{ОП} + n^2 t_{ОП} = 1.25t_{ОП}$$

### Загальний час зайнятості оперативної пам'яті при виконанні 2-х потоків

$$T_{ОП}^{ПАР} \cong 2\left(\frac{2n^2}{8} t_{ОП} + n^2 t_{ОП}\right) = 2.5t_{ОП}$$

### Загальний час паралельного виконання 2-х потоків

$$T_{ЗАГ}^{ПАР} = 2(T_{ОП} + T_{L3} + T_{ОБЧ}) = 2\left(\frac{2n^2}{8} t_{ОП} + \frac{n^3}{8} t_{L3} + \frac{7}{8} n^3 t_{СКЛ} / 2\right)$$

### Коефіцієнт використання оперативної пам'яті при паралельному виконанні 2-х потоків

$$ККД_{ОП} = \frac{T_{ЗАГ}^{ОП}}{T_{ЗАГ}^{ПАР}} = \frac{2.5n^2 t_{ОП}}{\frac{n^2}{2} t_{ОП} + \frac{n^3}{4} t_{L3} + \frac{7}{4} n^3 t_{СКЛ} / 2} = \frac{1}{0.2 + 0.1n \frac{t_{L3}}{t_{ОП}} + 0.7n \frac{t_{СКЛ}}{t_{ОП}}} \cong 0.098$$

### Коефіцієнт використання кеш-пам'яті L3 при виконанні 2-х потоків

$$ККД_{L3} = \frac{T_{ЗАГ}^{L3}}{T_{ЗАГ}^{ПАР}} \cong \frac{2 \frac{n^3}{8} t_{L3}}{2\left(\frac{2n^2}{8} t_{ОП} + \frac{n^3}{8} t_{L3} + \frac{7}{8} n^3 t_{СКЛ} / 2\right)} \cong \frac{1}{\frac{2 t_{ОП}}{n t_{L3}} + 1 + 7 \frac{t_{СКЛ}}{t_{L3}} / 2}$$

#### 2.4.7 Масиви не поміщаються в кеш-пам'ять третього рівня.

У кеш-пам'яті даних третього рівня L3 не поміщається блок стовпців матриці В. Робота йде весь час з ОП.

У цьому прикладі генеруються наступні типи груп команд:

ОП; ОП; Мнж; Скл.

L1; ОП; Мнж; Скл.

### Кількість груп команд кожного виду

$$N(\text{ОП}; \text{ОП}; \text{Мнж}; \text{Скл}) = n^2 * \frac{n}{16} = \frac{n^3}{16}$$

$$N(L1; \text{ОП}; \text{Мнж}; \text{Скл}) = n^3 - \frac{n^3}{16} = \frac{15}{16}n^3$$

### Ймовірності груп команд:

$$P(\text{ОП}; \text{ОП}; \text{Мнж}; \text{Скл}) = \frac{N(\text{ОП}; \text{ОП}; \text{Мнж}; \text{Скл})}{n^3} = \frac{\frac{n^3}{16}}{n^3} = \frac{1}{16}$$

$$P(L1; \text{ОП}; \text{Мнж}; \text{Скл}) = \frac{N(L1; \text{ОП}; \text{Мнж}; \text{Скл})}{n^3} = \frac{\frac{15}{16}n^3}{n^3} = \frac{15}{16}$$

### Часи виконання груп команд при частоті ядра процесора $f=4$ ГГц:

тип ОП; ОП; Мнж; –  $2*256 = 512$  тактів.

тип L1; ОП; Мнж; – 256 тактів.

Середній час виконання універсальним ФУ групи команд (1.2)

$$t_{\text{ФУ}} \cong \frac{1}{16} * 512\tau + \frac{15}{16} * 256\tau = \frac{17}{16} * 256\tau = 272\tau$$

$$t_{\text{ФУ}} \cong t_{\text{ОП}}$$

$$\rho_{\text{ФУ}} = \frac{\lambda_{\text{УУ}}}{\mu_{\text{ФУ}}} = \frac{t_{\text{ФУ}}}{t_{\text{УУ}}} = \frac{272\tau}{k\tau}$$

При  $k=1$  –  $\rho_{\text{ФУ}} = 272$ , а коефіцієнт використання УУ дорівнює **0.0038 або 0.38%**.

При  $k=2$  –  $\rho_{\text{ФУ}} = 136$ , а коефіцієнт використання УУ дорівнює 0.0076 або 0.76 %.

### Загальний час виконання одного потоку

$$T_{\text{ЗАГ}} = T_{\text{ОП}} + T_{\text{ЗП}} \cong n^3 t_{\text{ОП}} + \frac{n^2}{16} t_{\text{ОП}} + \frac{15}{16} n^2 t_{L1D} + n^2 t_{\text{ОП}}$$

### Загальний час зайнятості оперативної пам'яті

$$T_{\text{ЗАГ}} = T_{\text{ОП}} + T_{\text{ЗП}} \cong n^3 t_{\text{ОП}} + n^2 t_{\text{ОП}} \cong (n + 1) * n^2 t_{\text{ОП}}$$

### Коефіцієнт використання оперативної пам'яті

$$KKД_{ОП} = \frac{T_{ЗАГ}^{ОП}}{T_{ЗАГ}} = \frac{T_{ОП}^{ЧТ} + T_{ОП}^{ЗП}}{T_{ЗАГ}} \cong 1$$

### Загальний час зайнятості ОП при послідовному виконанні 2-х потоків

$$T_{ЗАГ}^{ПОСЛ} = 2(T_{ОП} + T_{ЗП}) \cong 2n^3 t_{ОП}$$

### Загальний час зайнятості оперативної пам'яті при паралельному виконанні 2-х потоків

$$T_{ЗАГ}^{ПАР} = 2(T_{ОП} + T_{ЗП}) \cong 2n^3 t_{ОП}$$

### Загальний час при паралельному виконанні 2-х потоків

$$T_{ЗАГ}^{ПАР} = 2(T_{ОП} + T_{ЗП}) \cong 2n^3 t_{ОП}$$

У кожній групі команд послідовно виконуються команди (спочатку читання операнда з ОП, потім Множення і додавання). Команди читання з ОП виконуються послідовно. На їх фоні виконується Множення і Додавання. Кеш-пам'ять даних першого рівня повторно практично не використовується. Тому час виконання кожної групи команд і вей програми визначається часом роботи з ОП.

### Коефіцієнт використання оперативної пам'яті при паралельному виконанні 2-х потоків

$$KKД_{ОП} = \frac{T_{ЗАГ}^{ПАР}}{T_{ЗАГ}} = \frac{2(T_{ОП}^{ЧТ} + T_{ОП}^{ЗП})}{T_{ЗАГ}} \cong \frac{2n^3 t_{ОП}}{2n^3 t_{ОП}} \cong 1$$

При виконанні одного потоку коефіцієнт завантаження (ймовірність зайнятості) оперативної пам'яті теж дорівнює 1.

### Коефіцієнт прискорення при паралельному виконанні 2-х потоків

$$S = \frac{2T_{ПОСЛ}}{T_{ПАР}} \cong \frac{2n^3 t_{ОП}}{2n^3 t_{ОП}} \cong 1$$

### 2.4.8 Масиви не поміщаються в кеш-пам'ять третього рівня

#### Кількість груп команд кожного виду при виконанні одного потоку

У кеш-пам'яті даних третього рівня L3 не поміщається блок стовпців матриці В. Робота йде весь час з ОП.

У цьому прикладі генеруються наступні типи груп команд:

ОП; ОП; Мнж; Скл.

L1; ОП; Мнж; Скл.

#### Кількість груп команд кожного виду

$$N(\text{ОП}; \text{ОП}; \text{Мнж}; \text{Скл}) = 2n^2 * \frac{n}{16} = \frac{2n^3}{16}$$

$$N(\text{L1}; \text{ОП}; \text{Мнж}; \text{Скл}) = 2n^3 - \frac{n * 2^3}{16} = \frac{15}{16} 2n^3$$

#### Ймовірності груп команд:

$$P(\text{ОП}; \text{ОП}; \text{Мнж}; \text{Скл}) = \frac{N(\text{ОП}; \text{ОП}; \text{Мнж}; \text{Скл})}{2n^3} = \frac{\frac{2n^3}{16}}{n^3} = \frac{1}{16}$$

$$P(\text{L1}; \text{ОП}; \text{Мнж}; \text{Скл}) = \frac{N(\text{L1}; \text{ОП}; \text{Мнж}; \text{Скл})}{2n^3} = \frac{\frac{15}{16} 2n^3}{n^3} = \frac{15}{16}$$

#### Часи виконання груп команд при частоті ядра процесора f=4 ГГц:

тип ОП; ОП; Мнж; – 2\*256 = 512 тактів.

тип L1; ОП; Мнж; – 256 тактів.

Середній час виконання універсальним ФУ групи команд

$$t_{\text{ФУ}} \cong \frac{1}{16} * 512\tau + \frac{15}{16} * 256\tau = \frac{17}{16} * 256\tau = 272\tau$$

$$t_{\text{ФУ}} \cong t_{\text{ОП}}$$

$$\rho_{\text{ФУ}} = \frac{\lambda_{\text{УУ}}}{\mu_{\text{ФУ}}} = \frac{t_{\text{ФУ}}}{t_{\text{УУ}}} = \frac{272\tau}{k\tau}$$

При k=1 –  $\rho_{\text{ФУ}} = 272$ , а коефіцієнт використання УУ дорівнює 0.0038 або 0.38%.

При  $k=2 - \rho_{\Phi Y} = 136$ , а коефіцієнт використання УУ дорівнює 0.0076 або 0.76 %.

**Загальний час виконання одного потоку**

$$T_{ЗАГ} = T_{ОП} + T_{ЗП} \cong n^3 t_{ОП} + \frac{n^2}{16} t_{ОП} + \frac{15}{16} n^2 t_{L1D} + n^2 t_{ОП}$$

При великих розмірах матриць (наприклад,  $n > 4096$ ) з відносною похибкою значно менше 1% можна прийняти, що

$$T_{ЗАГ} = T_{ОП} + T_{ЗП} \cong n^3 t_{ОП} + n^2 t_{ОП} \cong (n + 1) * n^2 t_{ОП}$$

**Загальний час зайнятості оперативної пам'яті**

$$T_{ЗАГ}^{ОП} = T_{ОП}^{ЧТ} + T_{ОП}^{ЗП} \cong (n + 1) * n^2 t_{ОП}$$

**Коефіцієнт використання оперативної пам'яті**

$$ККД_{ОП} = \frac{T_{ЗАГ}^{ОП}}{T_{ЗАГ}} = \frac{T_{ОП}^{ЧТ} + T_{ОП}^{ЗП}}{T_{ЗАГ}} \cong 1$$

**Загальний час зайнятості ОП при виконанні одного потоку**

$$T_{ЗАГ}^{ПОСЛ} = T_{ОП} + T_{ЗП} \cong (n + 1) * n^2 t_{ОП} \cong n^3 t_{ОП}$$

**Загальний час зайнятості ОП при послідовному виконанні 2-х потоків**

$$T_{ЗАГ}^{ПОСЛ} = 2(T_{ОП} + T_{ЗП}) \cong 2n^3 t_{ОП}$$

**Загальний час зайнятості оперативної пам'яті при паралельному виконанні 2-х потоків**

$$T_{ЗАГ}^{ПАР} = 2(T_{ОП} + T_{ЗП}) \cong 2n^3 t_{ОП}$$

### Загальний час при паралельному виконанні 2-х потоків

$$T_{ЗАГ}^{ПАР} = 2(T_{ОП} + T_{ЗП}) \cong 2n^3 t_{ОП}$$

### Загальний час послідовного виконання 2-х потоків

$$T_{ЗАГ}^{ПОСЛ} = 2(T_{ОП} + T_{ЗП}) \cong 2n^3 t_{ОП} + n^2 t_{ОП} \cong 2n^3 t_{ОП}$$

### Загальний час паралельного виконання 2-х потоків

$$T_{ЗАГ}^{ПАР} = 2(T_{ОП} + T_{ЗП}) \cong 2n^3 t_{ОП}$$

### Коефіцієнт прискорення при паралельному виконанні 2-х потоків

$$S = \frac{2T_{ПОСЛ}}{T_{ПАР}} \cong \frac{2n^3 t_{ОП}}{2n^3 t_{ОП}} \cong 1.$$

## 2.5 Експериментальні дані про виконання програми

Точні значення модельних (теоретичні) і експериментальних даних не збігаються, що і слід було очікувати.

Однак тенденції результатів однозначно однакові.

Чому відсутня прискорення при паралельному виконанні 2-х потоків при малих ( $n < 80$  і дуже великих розмірах матриць ( $n > 2048$ ) розмірах?

Відсутність прискорення при малих розмірах матриць пояснюється насамперед особливостями використаного алгоритму множення матриць.

Вихідний код програми приведено у Додатку А.

Результати експериментів наведені в таблиці 2.1 і на рис. 2.2.

Таблиця 2.1 Результати дослідження програми «Множення матриць»

Розмір	Прискорення теоретичне	Прискорення реальне
Плаваюча точка – 32 розряди		
64	1	1
128	1,65	1,69
512	1,45	1,37
$n > 2^{14}$	1	1,04
Плаваюча точка – 64 розряди		
64	1	1
128	1,46	1,6
256	1,27	1,36
$n > 2^{14}$	1	1,01

У чергового результаті обчислення він відразу записується в ОП минаючи кеш-пам'яті всіх рівнів. Запис в ОП виконується окремими елементами - 32-х або 64-х розрядними порціями кожен в залежності від форматів даних в матрицях. Запис одного елемента результату досить тривалий процес - (60 - 70) нс, що еквівалентно приблизно 256 тактів при частоті близько 4 ГГц.

При розмірності матриць  $n = 64$  елемента час обчислення чергового результату в ядрі процесора становить величину (близько 256 тактів) меншу тривалістю запису чергового результату в ОП.

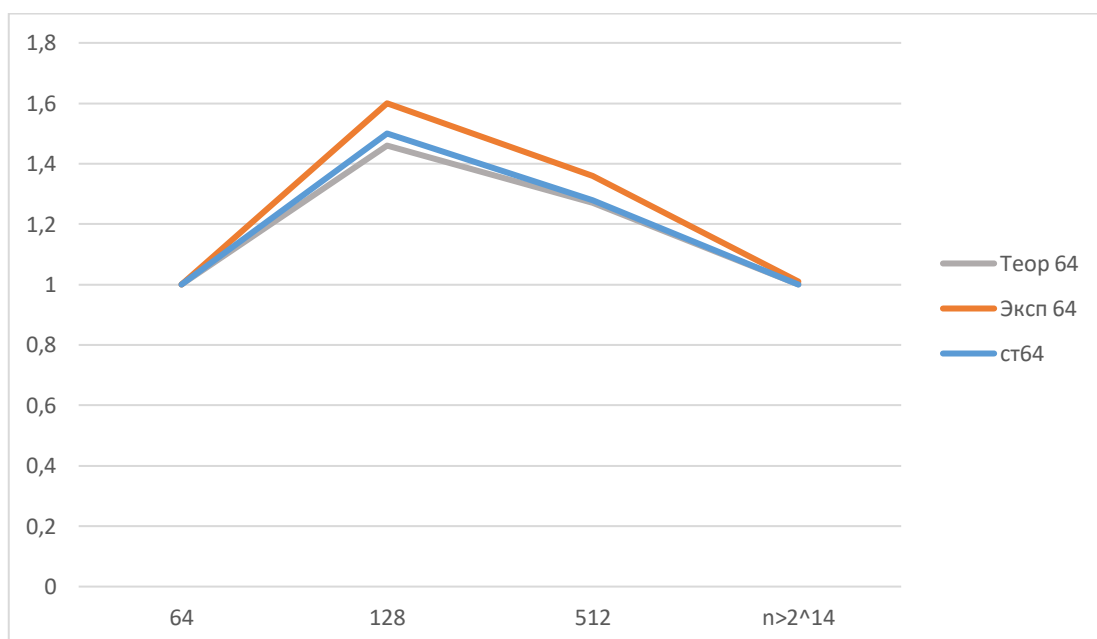


Рис. 2.2 Графік зміни прискорення



У підсумку повний час обчислення одного результату повністю визначається часом звернень в ОП. Коефіцієнт використання ОП при цьому дорівнює 1.

Такий результат підтверджує висновок про те, що немає сенсу запускати в одному фізичному ядрі 2 програми, які в значній мірі одночасно використовують одні й ті ж критично важливі ресурси.

Аналогічна ситуація виникає і при множенні матриць дуже великого розміру, коли не спрацьовують кеш-пам'яті всіх рівнів.

При  $n > 2048$  одна матриця з 32-х розрядними даними займає 16 Мбайт, а один вектор-стовпець матриці вимагає 128 Кбайт простору.

У цьому випадку програма змушена безперервно працювати з ОП при читанні елементів матриці В і коефіцієнт використання ОП дорівнює 1.

## 2.6 Висновки

У таблиці 2.2 наведені підсумкові теоретичні результати виконання програми «Множення матриць» для всіх розглянутих варіантів.

Таблиця 2.2 - Підсумкові теоретичні результати виконання програми «Множення матриць» для всіх розглянутих варіантів.

Розмір матриць	Розрядність даних	Кіл. потоків	ККД <sub>уу</sub> %	ККД <sub>оп</sub> %	ККД <sub>обч</sub> %	ККД <sub>L2</sub> %	ККД <sub>L3</sub> %	Коеф. прискорення
64	32	1	22.3	100	25	-	-	-
	64	1	20.7	100	25	-	-	-
	32	2	40	100	50	-	-	1
	64	2	33	100	50	-	-	1
128	32	1	21	47	25	16	-	-
	64	1	18.2	45	25	27	-	-
	32	2	40	42	50	15	-	1.65
	64	2	27	37	50	27	-	1.46
512	32	1	16.5	9.9	25	-	36	-
	64	1	12.3	7.7	25	-	55	-
	32	2	24	13.4	50	-	53.7	1.45
	64	2	12.3	9.8	50	-	70.5	1.27
2 <sup>14</sup>	32	1	0.4	100	25	-	-	-
	64	1	0.4	100	25	-	-	-
	32	2	0.4	100	25	-	-	1
	64	2	0.4	100	25	-	-	1

1. Коефіцієнт використання УУ (і всього ядра процесора) при використанні моделі ядра процесора дорівнює величині зворотної коефіцієнту навантаження універсального ФУ.

2. Коефіцієнт використання УУ (і всього ядра процесора), визначений з моделі ядра процесора, є верхньою межею ефективності ядра процесора.

3. Максимальний зафіксований показник використання УУ (і всього ядра процесора) змінюється в досить широкому діапазоні - від 1% до 22.3% в залежності від розмірності матриць і форматів даних.

4. Якщо параметри УУ гірші (коефіцієнт  $k$ , що визначає перезавантаження конвеєра команд через команд переходу дорівнює 2) коефіцієнт використання УУ (і всього ядра процесора) буде в 2 рази більший. Однак, навіть в гіршому випадку він не буде перевищувати 50%.

5. Коефіцієнт використання пропускної здатності обчислювальної компоненти підсистеми виконання команд, що складається з кеш-пам'яті даних першого рівня L1D, пристроїв Множення і Додавання не залежить від розмірності матриць і форматів даних та дорівнює:

0.25 - при виконанні одного потоку.

0.5 - при виконанні 2-х потоків.

Це означає, що обчислювальна компонента підсистеми виконання команд, що складається з кеш-пам'яті даних першого рівня L1D, пристроїв Множення і Додавання може бути використана при виконанні інших потоків.

6. При реальному виконанні однопоточкового варіанту програми «Множення матриць» високопродуктивне ядро процесора буде використовуватися не більше ніж на 25%, а обчислювальний компонент підсистеми виконання команд понад 75% всього часу буде простоювати.

7. При виконанні однопоточкового варіанту програми не всі ресурси процесора можуть бути задіяні. Ресурси продуктивності як підсистеми підготовки команд (УУ), та й підсистеми виконання команд достатні для виконання як мінімум ще одного потоку команд.

8. Коефіцієнти прискорення при виконанні 2-х поточного варіанту програми «Множення матриць» істотно залежать від розмірності матриць і не значно залежать від розрядності даних.

9. При малих розмірах матриць, коли дані повністю розміщуються в кеш-пам'яті даних першого рівня L1D ( $n < 64$ ) прискорення відсутнє. Це пояснюється специфікою програми «Множення матриць», яка виражається в тому, що час запису елемента результату безпосередньо в ОП більше часу обчислення цього елемента результату.

10. При розмірах матриць, коли для роботи з даними використовуються кеш-пам'яті другого і третього рівнів ( $n$  змінюється в межах від 64 до 512) прискорення роботи 2-х потоків існує і змінюється в межах від 1.27 до 1.65 в залежності від розрядності даних.

11. При дуже великих розмірах матриць, коли в кеш-пам'ять третього рівня L3 не поміщається блок з 64-ма байтами даних одного стовпчика матриці (розмір матриць 214 і більше) вся робота з даними здійснюється через ОП і двопоточність не має сенсу. Коефіцієнт завантаження пристрою управління ядра (його підсистеми підготовки команд) дуже маленький.

### 3 ДОСЛІДЖЕННЯ ВИКОНАННЯ ПРОГРАМИ «РІШЕННЯ ДР В ПРИВАТНИХ ПОХІДНИХ»

#### 3.1 Особливості програми

Диференціальні рівняння в приватних похідних являють собою широко застосовуваний математичний апарат при розробці моделей в самих різних областях науки і техніки.

Одним з найбільш поширених підходів чисельного рішення диференціальних рівнянь є метод кінцевих різниць (метод сіток). Слідуючи цьому підходу, область вирішення  $D$  представляється у вигляді дискретного (як правило, рівномірного) набору (сітки) точок (вузлів).

Використовуючи п'яти-точковий шаблон для обчислення значень похідних, можна уявити рівняння Пуассона в кінцево-різницевої формі

$$\frac{u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{ij}}{h^2} = f_{ij}.$$

Дане рівняння може бути дозволено щодо  $u_{ij}$ :

$$u_{ij} = 0.25(u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - h^2 f_{ij}).$$

Основна ділянка програми визначення чергової ітерації (ядро цієї програми) на умовній мові асемблера може мати такий вигляд (рис. 3.1):

1. **LOAD**  $U_{i-1,j}^{k-1} \rightarrow R1$ , читання верхнього рядка з ОП
2. **LOAD**  $U_{i,j-1}^{k-1} \rightarrow R2$ , читання центрального рядка (лівий) з ОП
3. **ADD**  $R1 + R2 \rightarrow R3$  (Верхній + лівий центральний)
4. **LOAD**  $U_{i+1,j}^{k-1} \rightarrow R4$  читання нижнього рядка з ОП
5. **LOAD**  $U_{i,j+1}^{k-1} \rightarrow R5$ , читання правого елемента центр. рядка з **L1D**
6. **ADD**  $R4 + R5 \rightarrow R6$  (Нижній + правий центральний)
7. **ADD**  $R3 + R6 \rightarrow R7$
8. **LOAD**  $h^2 * f_{i,j-1}^{k-1} \rightarrow R8$ , читання з ОП
9. **SUB**  $R7 - R8 \rightarrow R9$
10. **MPY**  $0,25 * R9 \rightarrow R10$
11. **STORE**  $R10 \rightarrow U_{ij}^k$ , запис до **L1D**
12. **LOAD**  $U_{ij}^{k-1} \rightarrow R11$ , читання з **L1D** шукана точка попер. ітерації
13. **SUB**  $R10 - R11 \rightarrow R12$
14. **ADD**  $R12 + R13 \rightarrow R13$  (в R13 накопичується похибка ітерації)
15. **JMP** початок циклу

Рис. 3.1 – Приклад ядра програми розрахунку чергового значення шуканої функції

Всього в наведеному спрощеному прикладі ядра програми обчислення чергового наближення значення шуканої функції 15 команд.

Умови виконання ядра програми в підсистемі виконання:

1. Масив значень нульового наближення і масив знаходяться в оперативній пам'яті.
2. Кеш-пам'ять даних першого рівня L1D це конвеєрний пристрій з пропускнуою спроможністю 1 такт і може кожен такт здійснювати 2 читання і 1 запис. Повний час звернення до кеш-пам'яті L1D - 4такти.
3. Функціональні пристрої Множення і Складання конвеєрні з пропускнуою спроможністю 1 такт. Повний час виконання операцій 4 такти.
4. У ядрі програми є команда редукції (накопичення похибки чергової ітерації). Час виконання операції редукції - 4 такти.

В ядрі програми можна виділити 7 випадків інформаційної залежності.

#### 1-й випадок

3-я команда  $\text{ADD } R1 + R2 \rightarrow R3$  інформаційно залежна від 1-й і 2-й команд  $\text{LOAD } U_{i-1,j}^{k-1} \rightarrow R1, \text{LOAD } U_{i+1,j}^{k-1} \rightarrow R2$  і може починатися тільки після їх завершення.

#### 2-й випадок

6-а команда  $\text{ADD } R4 + R5 \rightarrow R6$  інформаційно залежна від 4-ї команди  $\text{LOAD } U_{i,j-1}^{k-1} \rightarrow R4$  і 5-ї команди  $\text{LOAD } U_{i,j+1}^{k-1} \rightarrow R5$ , і може починатися тільки після їх завершення.

#### 3-й випадок

7-а команда  $\text{ADD } R3 + R6 \rightarrow R7$  інформаційно залежна від 3-й команди  $\text{ADD } R1 + R2 \rightarrow R3$  та від 6-й команди  $\text{ADD } R4 + R5 \rightarrow R6$  і може починатися тільки після їх завершення.

#### 4-й випадок

9-а команда  $\text{SUB } R7 - R8 \rightarrow R9$  інформаційно залежна від 7-ї та 8-ї команд  $\text{LOAD } h^2 * f_{i,j-1}^{k-1} \rightarrow R8$  и  $\text{ADDR3} + R6 \rightarrow R7$  і може починатися тільки після їх завершення.

#### 5-й випадок

10-а команда  $\text{MPY } 0,25 * R9 \rightarrow R10$  інформаційно залежна від 9-ї команди  $\text{SUB } R7 - R8 \rightarrow R9$  і може починатися тільки після її завершення.

#### 6-й випадок

11-а команда  $\text{STORE } R10 \rightarrow U_{i,j}^k$  інформаційно залежна від 10-ї команди  $\text{MPY } 0,25 * R9 \rightarrow R10$  і може починатися тільки після її завершення.

13-а команда **SUB R10 – R11 → R12** інформаційно залежна від 12-ї команди **LOAD  $U_{ij}^{k-1} \rightarrow R11$**  та 10-ї команди **MPY 0,25 \* R9 → R10** і може починатися тільки після їх завершення (обчислення похибки в  $i, j$  точці).

#### 7-й випадок

14-а команда **ADD R12 + R13 → R13** інформаційно залежна від 13-ї команди **SUBR10 – R11 → R12** и 14-й команди **ADD R12 + R13 → R13** (яка обчислює сумарну похибка в попередній точці) і може починатися тільки після їх завершення. 14-а команда **ADD R12 + R13 → R13** виконує операцію редукції.

12-а команда **LOAD  $U_{ij}^{k-1} \rightarrow R11$**  інформаційно незалежна і може починатися в будь-який момент, коли вільна підсистема пам'яті.

Усі команди **LOAD** (читання операндів з підсистеми пам'яті в регістри) інформаційно незалежні.

Приймається, що підсистема виконання команд це універсальне ФУ, продуктивність якого еквівалентна пікової продуктивності сучасних ядер процесорів. Це універсальне ФУ має тривірневу кеш-пам'ять і може одночасно читати 2 операнди з кеш-пам'яті даних першого рівня і записувати дані.

Процес обчислення значень функції ітераційний.

У початковому стані всі масиви даних розташовані в оперативній пам'яті. Читання з оперативної пам'яті (ОП) здійснюється блоками по 64 байти.

Запис в оперативну пам'ять здійснюється в залежності від того, чи є дані в кеш-пам'яті. Якщо їх немає, то виконується запис в ОП. Якщо дані є в кеш-пам'яті, то запис виконується в кеш-пам'ять.

Розмір початкового масиву нульового наближення і масиву значень функції  $h^2 * f_{ij}$  -  $m * m$ .

При виконанні першої ітерації з ОП зчитується  $2m^2$  елементів ( $m^2$  елементів початкових наближень і  $m^2$  значень функції  $h^2 * f_{ij}$ ). Якщо чергове значення в точці присутнє в кеш-пам'яті першого рівня, то воно записується в кеш-пам'ять першого рівня, інакше воно записується в оперативну пам'ять.

## 3.2 Один потік

### 3.2.1 Обидва масиви поміщаються в кеш-пам'ять даних першого рівня L1D

Процес рішення диференціального рівняння в приватних похідних ітераційний. Кількість ітерацій  $k$  залежить від значень початкового наближення і необхідної точності.

При виконанні кожної ітерації обчислюється  $m^2$  точок відповідно до рівняння (3.1).

### Перша ітерація

Всі дані розташовані в оперативній пам'яті (ОП).

Всього при виконанні 1-й ітерації з ОП зчитується  $2m^2$  значень -  $m^2$  точок чергових наближень и  $m^2$  точок значень функції  $h^2 * f_{ij}$ .

При кожному читанні з ОП отримується блок даних, що складається з 16 елементів - якщо дані 32-х розрядні.

При виконанні першої ітерації над усім масивом наближень розрізняють наступні випадки:

1. Обчислення  $\frac{m}{16}$  точок наближень в верхньому ряду масиву з  $m$  точок. При цьому виконується 4 звернення в ОП і необхідні обчислення з часом  $t_{ОБЧ}^1$ .

Загальний час обчислення  $\frac{m}{16}$  точок наближень в верхньому ряду масиву з  $m$  точок складе

$$T^1 = \frac{m}{16} (4t_{ОП} + t_{ОБЧ}^1)$$

2. Обчислення  $15 * \frac{m}{16}$  точок наближень в верхньому ряду масиву з  $m$  точок. При цьому дані вже знаходяться в кеш-пам'яті даних першого рівня L1D і виконуються всі необхідні обчислення без звернення в оперативну пам'ять з часом  $t_{ОБЧ}^2$ .

Загальний час обчислення  $15 * \frac{m}{16}$  точок наближень в верхньому ряду масиву з  $m$  точок складе

$$T^2 = 15 * \frac{m}{16} t_{ОБЧ}^2$$

3. Обчислення  $(m - 1) * \frac{m}{16}$  точок наближень, починаючи з 2-ї точки в стовпці, в  $\frac{m}{16}$  вертикальних стовбцях масиву з  $m$  точок. При цьому виконується 2 звернення в ОП і необхідні обчислення з часом  $t_{ВБЧ}^1$ . Загальний час обчислення  $(m - 1) * \frac{m}{16}$  точок наближень в вертикальних шпальтах масиву з  $m$  точок складе

$$T^3 = (m - 1) * \frac{m}{16} (2t_{ОП} + t_{ВБЧ}^1).$$

4. Обчислення  $15 * (m - 1)$  точок в кожному з прочитаних  $\frac{m}{16}$  стовпців точок наближень, починаючи з 2-ї точки в стовбці. При цьому всі необхідні дані вже знаходяться в кеш-пам'яті даних першого рівня L1D. Виконуються всі необхідні звернення до кеш-пам'ять даних першого рівня L1D і необхідні обчислення з часом  $t_{ОБЧ}^2$  для кожної точки.

Загальний час обчислення

$$15(m-1)\frac{m}{16}=(m-1)\frac{15}{16}m$$

точок наближень в вертикальних шпальтах масиву з  $m$  точок складе

$$T^4=(m-1)\frac{15m}{16}t_{\text{ОБЧ}}^2$$

**Час виконання першої ітерації над  $m^2$  точок**

$$T_{\text{ИТЕР}}^1 \cong N_1 * T_1 + N_2 * T_2 + N_3 * T_3 + N_4 * T_4$$

$$T_{\text{ИТЕР}}^1 \cong \frac{m}{16} (4t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + \frac{15m}{16} t_{\text{ОБЧ}}^2 + \frac{m(m-1)}{16} (2t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + \frac{15m(m-1)}{16} * t_{\text{ОБЧ}}^2$$

Після приведення подібних членів

$$T_{\text{ИТЕР}}^1 \cong \frac{m^2}{16} (2t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + \frac{15}{16} m^2 * t_{\text{ОБЧ}}^2 + \frac{m}{16} 2t_{\text{ОП}}$$

**Наближене значення часу першої ітерації, прийняте для всіх подальших обчислень**

$$T_{\text{ИТЕР}}^1 \cong \frac{2m^2}{16} (t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + \frac{15}{16} m^2 * t_{\text{ОБЧ}}^2$$

Відносна похибка при використанні наближеного часу виконання першої ітерації складе не більше 0.6% при  $m = 64$ ,  $t_{\text{ОП}} = 256$ ,  $t_{\text{ОБЧ}}^1 = 12\tau$ ,  $t_{\text{ОБЧ}}^2 = 24\tau$ .

При  $m = 128$  похибка зменшиться майже в 2 рази.

Надалі буде використовуватися наближена оцінка часу виконання першої ітерації

**Час виконання  $i$ -ї ( $i > 1$ ) ітерації, коли всі дані вже знаходяться в кеш-пам'яті даних першого рівня  $t_{L1D}$**

$$T_{\text{ИТЕР}}^i \cong m^2 * t_{\text{ОБЧ}}^2$$



### Час виконання (k-1) ітерацій

$$T_{ITER}^{k-1} \cong (k-1) * T_{ITER}^i \cong (k-1) * m^2 * t_{OБЧ}^2$$

Загальний час виконання всіх k ітерацій з урахуванням операцій записи результатів в ОП

$$T_{ЗАГ} = T_{ITER}^1 + T_{ITER}^{k-1} + T_{ЗАПИСУ}$$

$$T_{ЗАГ} \cong \frac{m^2}{16} (2t_{ОП} + t_{OБЧ}^1) + \frac{15}{16} m^2 * t_{OБЧ}^2 + (k-1)m^2 * t_{OБЧ}^2 + \frac{m^2}{16} t_{ОП}$$

### Середній час обчислення наближення в одній точці

$$t_{ТОЧКИ}^{1cp} \cong \frac{T_{ЗАГ}}{km^2} = \frac{3t_{ОП} + t_{OБЧ}^1 + (16k-1) * t_{OБЧ}^2}{16k} \cong t_{OБЧ}^2 + \frac{3t_{ОП} + t_{OБЧ}^1 - t_{OБЧ}^2}{16k}$$

$$\cong 24.37\tau$$

### Коефіцієнт використання УУ (і всього ядра процесора)

$$ККД_{УУ} = \frac{1}{\rho_{ФУ}}$$

де  $\rho_{ФУ}$  - коефіцієнт навантаження ФУ (1.1).

Кожна група складається з 15 архітектурних команд. При ступені суперскалярності S=4 архітектурних команд буде згенеровано за 15 / S тактів за умови, що S команд видаються кожен такт.

Час генерації однієї групи команд з 15 архітектурних команд

$$t_{УУ} = \frac{15_{команд}}{LS}$$

де L - кількість тактів, яке УУ витрачає на генерацію однієї групи, S кількість команд за такт.

При  $t_{ОП} = 256\tau$ ;  $m = 64$ ; ;  $t_{OБЧ}^1 = 12\tau$   $t_{OБЧ}^2 = 24\tau$ ;  $L = 1$ ;  $S = 4$ :

$$t_{yy} = \frac{15 \text{команд}}{4}$$

$$t_{\text{ТОЧКИ}}^{1\text{ср}} \cong t_{\text{ОБЧ}}^2 + \frac{3t_{\text{ОП}} - (t_{\text{ОБЧ}}^2 - t_{\text{ОБЧ}}^1)}{16k}$$

$$\rho_{\text{ФУ}} = \frac{t_{\text{групи}}^{\text{ср}}}{L * S \tau} \cong \frac{t_{\text{ОБЧ}}^2 + \frac{3t_{\text{ОП}} - (t_{\text{ОБЧ}}^2 - t_{\text{ОБЧ}}^1)}{16k}}{4\tau}$$

$$KKД_{yy} = \frac{1}{\rho_{yy}} \cong \frac{1}{\frac{t_{\text{ОБЧ}}^2}{4} + \frac{2t_{\text{ОП}} - (t_{\text{ОБЧ}}^2 - t_{\text{ОБЧ}}^1)}{16k * 4}} \cong \frac{1}{6 + \frac{500}{16k * 4}} \cong \frac{1}{6 + \frac{7.8125}{k}}$$

Максимальне значення коефіцієнта використання УУ досягається при  $k \rightarrow \infty$  і воно рівне  $1/6=0.167$  або 16.7%.

При  $k=1$   $KKД_{yy} = 0.0724$ .

При  $k=10$   $KKД_{yy} = 0.148$ .

При  $k=128$   $KKД_{yy} = 0.164$ .

**Якщо  $L = 2$  і УУ видає кожні  $S$  команд за 2 такту, то при**

$$KKД_{yy} = \frac{1}{\rho_{yy}} \cong 0.33$$

### Коефіцієнт використання оперативної пам'яті

$$KKД_{\text{ОП}} \cong \frac{T_{\text{ОП}}}{T_{\text{ЗАГ}}} \cong \frac{\frac{m^2}{16} 3t_{\text{ОП}}}{\frac{m^2}{16} (3t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + \frac{(16k-1)}{16} m^2 * t_{\text{ОБЧ}}^2} \cong 0,0154$$

### Коефіцієнт використання обчислювальної підсистеми

$$KKД_{\text{ОБЧ}} \cong \frac{T_{\text{ОБЧ}}}{T_{\text{ЗАГ}}} \cong \frac{\frac{m^2}{16} t_{\text{ОБЧ}}^1 + \frac{15}{16} m^2 * t_{\text{ОБЧ}}^2 + (k-1)m^2 * t_{\text{ОБЧ}}^2}{\frac{m^2}{16} (3t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + \frac{15}{16} m^2 * t_{\text{ОБЧ}}^2 + (k-1)m^2 * t_{\text{ОБЧ}}^2} \cong 0,985$$

Реальний становить 0.25 від обчисленого значення, так як всі компоненти обчислювальної підсистеми це конвеєрні пристрої, коефіцієнт використання визначався як для комбінаційних пристроїв.

**Підсумок коефіцієнти завантаження підсистем ядра рівні:**

$$\text{ККД}_{\text{уу}} \cong 16.4\%$$

$$\text{ККД}_{\text{оп}} \cong 1.54\%$$

$$\text{ККД}_{\text{обч}} \cong 98.5\%$$

### 3.2.2 Обидва масиви поміщаються в кеш-пам'ять даних другого рівня

**Час виконання першої ітерації над  $m^2$  точок**, коли всі дані розташовані в оперативній пам'яті (ОП).

$$T_{\text{ІТЕР}}^1 \cong \frac{m^2}{16} (2t_{\text{оп}} + t_{\text{обч}}^1) + \frac{15}{16} m^2 * t_{\text{обч}}^2$$

**Час виконання  $i$ -ї ( $i > 1$ ) ітерації**, коли всі дані вже знаходяться в кеш-пам'яті даних другого рівня  $t_{L2}$

$$T_{\text{ІТЕР}}^i \cong \frac{m^2}{16} (2t_{L2} + t_{\text{обч}}^1) + \frac{15}{16} m^2 * t_{\text{обч}}^2$$

**Час виконання  $(k-1)$  ітерацій**

$$T_{\text{ІТЕР}}^{k-1} \cong (k-1) * T_{\text{ІТЕР}}^i \cong (k-1) * \left[ \frac{m^2}{16} (2t_{L2} + t_{\text{обч}}^1) + \frac{15}{16} m^2 * t_{\text{обч}}^2 \right]$$

**Загальний час виконання всіх  $k$  ітерацій**

$$T_{\text{ЗАГ}} = T_{\text{ІТЕР}}^1 + T_{\text{ІТЕР}}^{k-1} + T_{\text{ЗП}} \cong \frac{m^2}{16} (3t_{\text{оп}} + t_{\text{обч}}^1) + (k-1) * \frac{m^2}{16} (2t_{L2} + t_{\text{обч}}^1) + k * \frac{15}{16} m^2 * t_{\text{обч}}^2$$

**Середній час обчислення однієї точки при виконанні всіх  $k$  ітерацій**

$$\begin{aligned} t_{\text{ТОЧКИ}}^{\text{ср}} &\cong \frac{T_{\text{ЗАГ}}}{k * m^2} = \frac{\frac{m^2}{16} (3t_{\text{оп}} + t_{\text{обч}}^1) + (k-1) * \frac{m^2}{16} (2t_{L2} + t_{\text{обч}}^1) + k * \frac{15}{16} m^2 * t_{\text{обч}}^2}{k * m^2} \cong \\ &\cong \frac{3t_{\text{оп}} + k(t_{\text{обч}}^1 + 15 * t_{\text{обч}}^2) + (k-1) * 2t_{L2}}{16k} \\ \rho_{\text{ФУ}} &= \frac{t_{\text{ср}}}{L * S \tau} \cong \frac{3t_{\text{оп}} + k(t_{\text{обч}}^1 + 15 * t_{\text{обч}}^2) + (k-1) * 2t_{L2}}{16k} \frac{1}{L * S \tau} \end{aligned}$$

$$\begin{aligned} \text{ККД}_{\text{УУ}} &= \frac{1}{\rho_{\text{УУ}}} \cong \frac{LSt}{3t_{\text{ОП}} + k(t_{\text{ОБЧ}}^1 + 15 * t_{\text{ОБЧ}}^2) + (k-1) * 2t_{L2}} \cong \\ &\cong \frac{16k * 4\tau}{3t_{\text{ОП}} - 2t_{L2} + k(t_{\text{ВЫЧ}}^1 + 15 * t_{\text{ВЫЧ}}^2 + 2t_{L2})} \cong \frac{1}{6.188 + \frac{7.625}{k}} \end{aligned}$$

Максимальне значення коефіцієнта використання УУ досягається при  $k \rightarrow \infty$  і воно рівно  $1/6=0.167$  або 16.7%.

При  $k=1$   $\text{ККД}_{\text{УУ}} = 0.0723$  або 7.23%.

При  $k=10$   $\text{ККД}_{\text{УУ}} = 0.144$  або 14.4%.

При  $k=128$   $\text{ККД}_{\text{УУ}} = 0.159$  або 15.9%.

**Якщо  $L = 2$  і УУ видає кожні  $S$  команд за 2 такту, то при**

$\text{ККД}_{\text{УУ}} = 0.318$  або 31.8%.

**Коефіцієнт використання оперативної пам'яті**

$$\begin{aligned} \text{ККД}_{\text{ОП}} &\cong \frac{T_{\text{ОП}}}{T_{\text{ЗАГ}}} \cong \\ &\cong \frac{\frac{m^2}{16} 3t_{\text{ОП}}}{\left[\frac{m^2}{16} (3t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + \frac{15}{16} m^2 * t_{\text{ОБЧ}}^2\right] + (k-1) * \left[\frac{m^2}{16} (2t_{L2} + t_{\text{ОБЧ}}^1) + \frac{15}{16} m^2 * t_{\text{ОБЧ}}^2\right]} \cong \\ &\cong \frac{3t_{\text{ОП}}}{3t_{\text{ОП}} + k(t_{\text{ОБЧ}}^1 + 15 * t_{\text{ОБЧ}}^2) + (k-1) * 2t_{L2}} \cong 0,0149 \end{aligned}$$

**Коефіцієнт використання обчислювальної підсистеми**

$$\text{ККД}_{\text{ОБЧ}} \cong \frac{T_{\text{ОБЧ}}}{T_{\text{ЗАГ}}} \cong \frac{\frac{m^2}{16} t_{\text{ОБЧ}}^1 + k * \left[\frac{m^2}{16} t_{\text{ОБЧ}}^1 + \frac{15}{16} m^2 * t_{\text{ОБЧ}}^2\right]}{\frac{m^2}{16} (3t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + \frac{15}{16} m^2 * t_{\text{ОБЧ}}^2 + (k-1) * \left[\frac{m^2}{16} (2t_{L2} + t_{\text{ОБЧ}}^1) + \frac{15}{16} m^2 * t_{\text{ОБЧ}}^2\right]} \cong 0,926$$

Так як L1D, пристрій Множення та пристрій складання це конвеєрні пристрої з пропускнуою спроможністю 1 операція за такт, то реальний коефіцієнт використання кожного пристрою буде істотно менший.

### Коефіцієнт використання кеш-пам'яті другого рівня L2

$$KKД_{L2} \cong \frac{T_{L2}}{T_{3AG}} \cong \frac{(k-1) * \frac{m^2}{16} 2t_{L2}}{\frac{m^2}{16} (3t_{OP} + t_{OBЧ}^1) + \frac{15}{16} m^2 * t_{OBЧ}^2 + (k-1) * [\frac{m^2}{16} (2t_{L2} + t_{OBЧ}^1) + \frac{15}{16} m^2 * t_{OBЧ}^2]} \cong 0,0593$$

**Підсумок коефіцієнти завантаження підсистем ядра рівні:**

$$KKД_{YV} = 15.9\%$$

$$KKД_{OP} = 1.49\%$$

$$KKД_{OBЧ} = 92.6\%$$

$$KKД_{L2} = 5.93\%$$

### 3.2.3 Обидва масиви вміщуються в кеш-пам'ять третього рівня

При виконанні першої ітерації, коли всі дані розташовані в оперативній пам'яті (ОП).

При  $t_{OP} = 256\tau$ ;  $m = 512$ ;  $t_{OBЧ}^1 = 12\tau$ ;  $t_{OBЧ}^2 = 24\tau$ ;  $t_{L3} = 36$ ;  $L = 1$ ;  $S = 4$

**Час виконання першої ітерації над  $m^2$  точок**

$$T_{ITER}^1 \cong \frac{m^2}{16} (2t_{OP} + t_{OBЧ}^1) + \frac{15}{16} m^2 * t_{OBЧ}^2$$

**Час виконання  $i$ -ї ( $i > 1$ ) ітерації, коли всі дані вже знаходяться в кеш-пам'яті даних третього рівня  $t_{L3}$**

$$T_{ITER}^i \cong \frac{m^2}{16} (2t_{L3} + t_{OBЧ}^1) + \frac{15}{16} m^2 * t_{OBЧ}^2 .$$

**Час виконання  $(k-1)$  ітерацій**

$$T_{ITER}^{k-1} \cong (k-1) * T_{ITER}^i \cong (k-1) * [\frac{m^2}{16} (2t_{L3} + t_{OBЧ}^1) + \frac{15}{16} m^2 * t_{OBЧ}^2].$$

**Загальний час виконання всіх  $k$  ітерацій**

$$T_{3AG} = T_{ITER}^1 + T_{ITER}^{k-1} + T_{3II} \cong \frac{m^2}{16} (3t_{OP} + t_{OBЧ}^1) + (k-1) * \frac{m^2}{16} (2t_{L3} + t_{OBЧ}^1) + k \frac{15}{16} m^2 * t_{OBЧ}^2$$

### Середній час обчислення однієї точки при виконанні всіх $k$ ітерацій

$$\begin{aligned}
 t_{\text{ТОЧКИ}}^{\text{ср}} &\cong \frac{\frac{m^2}{16}(3t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + (k-1) * \frac{m^2}{16}(2t_{L3} + t_{\text{ОБЧ}}^1) + k \frac{15}{16} m^2 * t_{\text{ОБЧ}}^2}{k * m^2} \cong \\
 &\cong \frac{3t_{\text{ОП}} + t_{\text{ОБЧ}}^1 + (k-1) * [2t_{L3} + t_{\text{ОБЧ}}^1] + k * 15 * t_{\text{ОБЧ}}^2}{16k} \cong \\
 &\cong \frac{3t_{\text{ОП}} + k(t_{\text{ОБЧ}}^1 + 15 * t_{\text{ОБЧ}}^2) + (k-1) * 2t_{L3}}{16k} \\
 \rho_{\text{ФУ}} &= \frac{t_{\text{групи}}^{\text{ср}}}{L * S\tau} \cong \frac{\frac{3t_{\text{ОП}} + k(t_{\text{ОБЧ}}^1 + 15 * t_{\text{ОБЧ}}^2) + (k-1) * 2t_{L3}}{16k}}{LS\tau} \\
 \text{ККД}_{\text{УУ}} &= \frac{1}{\rho_{\text{УУ}}} \cong \frac{LS\tau}{\frac{3t_{\text{ОП}} + k(t_{\text{ОБЧ}}^1 + 15 * t_{\text{ОБЧ}}^2) + (k-1) * 2t_{L3}}{16k}} \\
 \text{ККД}_{\text{УУ}} &\cong \frac{1\tau}{\frac{t_{\text{ОБЧ}}^1 + 15 * t_{\text{ОБЧ}}^2 + 2t_{L3}}{64} + \frac{3t_{\text{ОП}} - 2t_{L3}}{64k}} \cong \frac{1}{6.388 + \frac{10.875}{k}}
 \end{aligned}$$

Максимальне значення коефіцієнта використання УУ досягається при  $k \rightarrow \infty$  і воно дорівнює 0.156 або 15.6%.

При  $k=1$   $\text{ККД}_{\text{УУ}} = 0.0581$  або 5.81%.

При  $k=10$   $\text{ККД}_{\text{УУ}} = 0.134$  або 13.4%.

При  $k=128$   $\text{ККД}_{\text{УУ}} = 0.154$  або 15.4%.

**Якщо  $L = 2$  і УУ видає кожні  $S$  команд за  $2$  такти, то**

$\text{ККД}_{\text{УУ}} = 0.308$  або 30.8%.

### Коефіцієнт використання оперативної пам'яті

$$\begin{aligned}
 \text{ККД}_{\text{ОП}} &\cong \frac{T_{\text{ОП}}}{T_{\text{ЗАГ}}} \cong \\
 &\cong \frac{\frac{m^2}{16} 2t_{\text{ОП}} + \frac{m^2}{16} t_{\text{ОП}} = \frac{m^2}{16} 3t_{\text{ОП}}}{\frac{m^2}{16}(3t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + \frac{15}{16} m^2 * t_{\text{ОБЧ}}^2 + (k-1) * \left[ \frac{m^2}{16}(2t_{L3} + t_{\text{ОБЧ}}^1) + \frac{15}{16} m^2 * t_{\text{ОБЧ}}^2 \right]} \cong \\
 &\cong \frac{3t_{\text{ОП}}}{3t_{\text{ОП}} + k(t_{\text{ОБЧ}}^1 + 15 * t_{\text{ОБЧ}}^2) + (k-1) * 2t_{L3}}
 \end{aligned}$$

Коефіцієнт корисного використання (завантаження) оперативної пам'яті з урахуванням записи результатів в пам'ять

$\text{ККД}_{\text{ОП}} \cong 0.0134$ .

### Коефіцієнт використання обчислювальної підсистеми

$$\begin{aligned}
 \text{ККД}_{\text{ОБЧ}} &\cong \frac{T_{\text{ОБЧ}}}{T_{\text{ЗАГ}}} \cong \\
 &\cong \frac{\frac{m^2}{16} t_{\text{ОБЧ}}^1 + k * \left[ \frac{m^2}{16} t_{\text{ОБЧ}}^1 + \frac{15}{16} m^2 * t_{\text{ОБЧ}}^2 \right]}{\frac{m^2}{16} (3t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + \frac{15}{16} m^2 * t_{\text{ОБЧ}}^2 + (k-1) * \left[ \frac{m^2}{16} (2t_{L3} + t_{\text{ОБЧ}}^1) + \frac{15}{16} m^2 * t_{\text{ОБЧ}}^2 \right]} \cong \\
 &\cong \frac{k * [t_{\text{ОБЧ}}^1 + 15 * t_{\text{ОБЧ}}^2]}{3t_{\text{ОП}} + k * (t_{\text{ОБЧ}}^1 + 15 * t_{\text{ОБЧ}}^2) + (k-1) * 2t_{L3}} = 0.828
 \end{aligned}$$

### Коефіцієнт використання кеш-пам'яті третього рівня L3

$$\begin{aligned}
 \text{ККД}_{L3} &\cong \frac{T_{L3}}{T_{\text{ОБЩ}}} \cong \\
 &\cong \frac{(k-1) * \frac{m^2}{16} 2t_{L3}}{\frac{m^2}{16} (3t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + \frac{15}{16} m^2 * t_{\text{ОБЧ}}^2 + (k-1) * \left[ \frac{m^2}{16} (2t_{L3} + t_{\text{ОБЧ}}^1) + \frac{15}{16} m^2 * t_{\text{ОБЧ}}^2 \right]} \cong \\
 &\cong \frac{(k-1) * 2t_{L3}}{3t_{\text{ОП}} + k * (t_{\text{ОБЧ}}^1 + 15 * t_{\text{ОБЧ}}^2) + (k-1) * 2t_{L3}} \cong 0.159.
 \end{aligned}$$

### Підсумок коефіцієнти використання підсистем ядра рівні:

$$\text{ККД}_{\text{УУ}} = 15.4\%$$

$$\text{ККД}_{\text{ОП}} = 1.34\%$$

$$\text{ККД}_{\text{ОБЧ}} = 82.8\%$$

$$\text{ККД}_{L3} = 15.9\%$$

### 3.2.4 Масиви не поміщаються в кеш-пам'ять третього рівня

При виконанні кожної ітерації, коли всі дані розташовані в оперативній пам'яті (ОП).

**Час виконання однієї ітерації над  $m^2$  точок**

$$T_{\text{ІТЕР}}^1 \cong \frac{m^2}{16} (2t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + \frac{15}{16} m^2 * t_{\text{ОБЧ}}^2 + \frac{m^2}{16} t_{\text{ОП}} + \frac{m}{16} 2t_{\text{ОП}}$$

$$T_{\text{ІТЕР}}^1 \cong \frac{m^2}{16} (3t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + \frac{15}{16} m^2 * t_{\text{ОБЧ}}^2 + \frac{m}{16} 2t_{\text{ОП}}$$

### Загальний час виконання всіх k ітерацій

$$T_{\text{ІТЕР}}^k \cong k * T_{\text{ІТЕР}}^1 \cong k \left[ \frac{m^2}{16} (3t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + \frac{15}{16} m^2 * t_{\text{ОБЧ}}^2 + \frac{m}{16} 2t_{\text{ОП}} \right]$$

### Середній час обчислення наближення в одній точці

$$t_{\text{ТОЧКИ}}^{\text{1ср}} \cong \frac{T_{\text{ІТЕР}}^k}{k * m^2} = \frac{k \left[ \frac{m^2}{16} (3t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + \frac{15}{16} m^2 * t_{\text{ОБЧ}}^2 + \frac{m}{16} 2t_{\text{ОП}} \right]}{k * m^2}$$

$$t_{\text{ТОЧКИ}}^{\text{1ср}} \cong \frac{3t_{\text{ОП}} + t_{\text{ОБЧ}}^1 + 15 * t_{\text{ОБЧ}}^2 + \frac{2t_{\text{ОП}}}{m}}{16}$$

$$\rho_{\text{ФУ}} = \frac{t_{\text{ТОЧКИ}}^{\text{1ср}}}{L * S\tau} \cong \frac{\frac{3t_{\text{ОП}} + t_{\text{ОБЧ}}^1 + 15 * t_{\text{ОБЧ}}^2 + \frac{2t_{\text{ОП}}}{m}}{16}}{LS\tau}$$

$$\rho_{\text{ФУ}} \cong \frac{3t_{\text{ОП}} + t_{\text{ОБЧ}}^1 + 15 * t_{\text{ОБЧ}}^2 + \frac{2t_{\text{ОП}}}{m}}{16LS\tau}$$

$$\text{ККД}_{\text{УУ}} = \frac{1}{\rho_{\text{УУ}}} \cong \frac{16LS\tau}{3t_{\text{ОП}} + t_{\text{ОБЧ}}^1 + 15 * t_{\text{ОБЧ}}^2 + \frac{2t_{\text{ОП}}}{m}} \cong 0.0561$$

Якщо  $L = 2$  і УУ видає кожні  $S$  команд за 2 такту, то

$$\text{ККД}_{\text{УУ}} = 0.112$$

### Коефіцієнт використання оперативної пам'яті

$$\begin{aligned} \text{ККД}_{\text{ОП}} &\cong \frac{T_{\text{ОП}}}{T_{\text{ЗАГ}}} \cong \frac{k * \left[ \frac{m^2}{16} * 3t_{\text{ОП}} + \frac{2t_{\text{ОП}}}{m} \right]}{k * \left[ \frac{m^2}{16} (3t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + \frac{15m^2}{16} t_{\text{ОБЧ}}^2 + \frac{2t_{\text{ОП}}}{m} \right]} \cong \\ &\cong \frac{3t_{\text{ОП}}}{3t_{\text{ОП}} + t_{\text{ОБЧ}}^1 + 15 * t_{\text{ОБЧ}}^2} \cong 0.674 \end{aligned}$$

### Коефіцієнт використання обчислювальної підсистеми

$$\text{ККД}_{\text{ОБЧ}} \cong \frac{T_{\text{ОБЧ}}}{T_{\text{ЗАГ}}} \cong \frac{k * \frac{m^2}{16} * t_{\text{ОБЧ}}^1 + k * \frac{15m^2}{16} t_{\text{ОБЧ}}^2}{k * \left[ \frac{m^2}{16} (2t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + \frac{15m^2}{16} t_{\text{ОБЧ}}^2 + \frac{2t_{\text{ОП}}}{m} \right]} \cong$$



$$\cong \frac{t_{\text{ОБЧ}}^1 + 15t_{\text{ОБЧ}}^2}{2t_{\text{ОП}} + t_{\text{ОБЧ}}^1 + 15t_{\text{ОБЧ}}^2 + \frac{32t_{\text{ОП}}}{m^3}} \cong \frac{t_{\text{ОБЧ}}^1 + 15t_{\text{ОБЧ}}^2}{2t_{\text{ОП}} + t_{\text{ОБЧ}}^1 + 15t_{\text{ОБЧ}}^2} \cong 0.326$$

**Коефіцієнти використання підсистем ядра і прискорення такі:**

$$\text{ККД}_{\text{ДУ}} = 5.61\%$$

$$\text{ККД}_{\text{ОП}} = 67.4\%$$

$$\text{ККД}_{\text{ОБЧ}} = 32.6\%.$$

### 3.3 Два потоки

#### 3.3.1 Обидва масиви вміщуються в кеш-пам'ять даних першого рівня L1D

Процес рішення диференціального рівняння в приватних похідних ітераційний. Кількість ітерацій  $k$  залежить від значень початкового наближення і необхідної точності.

При виконанні кожної ітерації обчислюється  $m^2$  точок відповідно до рівняння (1).

Команди звернення до ОП виконуються послідовно комбінаційною пристроєм ОП.

Так як всі дані для обох потоків розміщуються в кеш-пам'яті даних першого рівня L1D, то загальна кількість звернень в ОП складе  $2 * \frac{2m^2}{16}$ .

Під час виконання звернень в ОП або в кеш-пам'яті другого або третього рівнів відсутні суміщення груп команд, що виконують виконані команд ОБЧ<sup>1</sup> для паралельного потоку.

Під час виконання груп команд, що виконують ОБЧ<sup>2</sup> для першого потоку, можуть бути виконані групи команд ОБЧ<sup>2</sup> для другого потоку, і під час виконання команд, що виконують групи команд ОБЧ<sup>2</sup> для другого потоку, можуть бути виконані групи команди ОБЧ<sup>2</sup> для першого потоку.

**Тоді загальний час паралельного виконання 2-х потоків однакових ядер програми «Рішення ДУ в приватних похідних» складе**

$$\begin{aligned} T_{\text{ЗАГ}}^{2\text{П}} &\cong 2 * \left[ \frac{m^2}{16} (2t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + \frac{m^2}{16} t_{\text{ОП}} \right] + \frac{15m^2}{16} * t_{\text{ОБЧ}}^2 + (k - 1) * m^2 * t_{\text{ОБЧ}}^2 \cong \\ &\cong 2 * \left[ \frac{m^2}{16} (3t_{\text{ОП}} + t_{\text{ОБЧ}}^1) \right] + \frac{(16k - 1)}{16} * m^2 * t_{\text{ОБЧ}}^2 \end{aligned}$$

**Середній час обчислення чергового наближення в одній точці при паралельному виконанні 2-х потоків**

$$t_{\text{точки}}^{\text{ср}} \cong \frac{T_{3\text{АГ}}^{2\text{П}}}{2m^2k} = \frac{2 * \left[ \frac{m^2}{16} (3t_{\text{ОП}} + t_{\text{ОБЧ}}^1) \right] + \frac{15m^2}{16} * t_{\text{ОБЧ}}^2 + (k-1) * m^2 * t_{\text{ОБЧ}}^2}{2m^2k} \cong$$

$$\cong \frac{2 * (3t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + (16k-1) * t_{\text{ОБЧ}}^2}{32 * k} \cong 12,38$$

**Коефіцієнт використання УУ (і всього ядра процесора)**

$$\rho_{\text{УУ}} = \frac{t_{\text{групи}}^{\text{ср}}}{L * S\tau} \cong \frac{2 * (3t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + (16k-1) * t_{\text{ОБЧ}}^2}{32 * k} \cdot \frac{1}{L * S\tau}$$

$$\text{ККД}_{\text{УУ}} = \frac{1}{\rho_{\text{УУ}}} \cong \frac{L * S\tau}{2 * (3t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + (16k-1) * t_{\text{ОБЧ}}^2}$$

$$\cong \frac{32 * k * 4\tau}{2 * (3t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + (16k-1) * t_{\text{ОБЧ}}^2}$$

$$\cong \frac{1}{\frac{t_{\text{ОБЧ}}^2}{8} + \frac{6t_{\text{ОП}} - (t_{\text{ОБЧ}}^2 - t_{\text{ОБЧ}}^1)}{128k}} \cong \frac{1}{3 + \frac{1524}{128k}} \cong \frac{1}{3 + \frac{11.9}{k}}$$

Максимальне значення коефіцієнта використання УУ досягається при  $k \rightarrow \infty$  і воно дорівнює  $1/3 = 0.333$  або 33.3%.

При  $k=1$   $\text{ККД}_{\text{УУ}} = 0.067$  або 6.7%.

При  $k=10$   $\text{ККД}_{\text{УУ}} = 0.239$  або 23.9%.

При  $k=128$   $\text{ККД}_{\text{УУ}} = 0.323$  або 32.3%.

**Якщо УУ видає кожні S команд кожні 2 такти, то**

$$\rho_{\text{УУ}}^{2\text{П}} \cong \frac{12.38}{8} = 1.5475$$

$$\text{ККД}_{\text{УУ}}^{2\text{П}} = \frac{1}{\rho_{\text{УУ}}^{2\text{П}}} \cong \frac{1}{1.5475} = 0.646$$

**Коефіцієнт використання оперативної пам'яті при паралельному виконанні 2-х потоків**

$$\text{ККД}_{\text{ОП}}^{2\text{П}} \cong \frac{T_{\text{ОП}}^{2\text{П}}}{T_{\text{ПАР}}^{2\text{П}}} \cong$$

$$\cong \frac{2 * \left[ \frac{m^2}{16} 2t_{\text{оп}} + \frac{m^2}{16} t_{\text{оп}} \right]}{2 * \left[ \frac{m^2}{16} (2t_{\text{оп}} + t_{\text{обч}}^1) \right] + \frac{15m^2}{16} * t_{\text{обч}}^2 + (k-1) * m^2 * t_{\text{обч}}^2 + \frac{m^2}{16} 2t_{\text{оп}}} \cong$$

$$\cong \frac{3t_{\text{оп}}}{2t_{\text{оп}} + t_{\text{обч}}^1 + (8k-0.5) * t_{\text{обч}}^2 + t_{\text{оп}}} \cong \frac{3t_{\text{оп}}}{3t_{\text{оп}} + t_{\text{обч}}^1 + (8k-0.5) * t_{\text{обч}}^2} \cong 0.0303$$

### Коефіцієнт використання обчислювальної підсистеми

$$KKD_{\text{обч}}^{2\text{П}} \cong \frac{T_{\text{обч}}^{2\text{П}}}{T_{\text{пар}}^{2\text{П}}} \cong \frac{2 * \frac{m^2}{16} t_{\text{обч}}^1 + \frac{15m^2}{16} * t_{\text{обч}}^2 + (k-1) * m^2 * t_{\text{обч}}^2}{2 * \left[ \frac{m^2}{16} (2t_{\text{оп}} + t_{\text{обч}}^1) + \frac{m^2}{16} t_{\text{оп}} \right] + \frac{15m^2}{16} * t_{\text{обч}}^2 + (k-1) * m^2 * t_{\text{обч}}^2} \cong$$

$$\cong \frac{2 * t_{\text{обч}}^1 + (16k-1) * t_{\text{обч}}^2}{2[(3t_{\text{оп}} + t_{\text{обч}}^1)] + (16k-1) * t_{\text{обч}}^2} \cong 0.97$$

### Коефіцієнт прискорення

Коефіцієнт прискорення визначається як відношення часу послідовного виконання 2-х потоків до часу паралельного виконання 2-х потоків.

$$S \cong \frac{T_{\text{посл}}^{2\text{П}}}{T_{\text{пар}}^{2\text{П}}} \cong \frac{2 \left[ \frac{m^2}{16} (2t_{\text{оп}} + t_{\text{обч}}^1) + \frac{15m^2}{16} * t_{\text{обч}}^2 + (k-1) * m^2 * t_{\text{обч}}^2 + \frac{m^2}{16} t_{\text{оп}} \right]}{2 * \left[ \frac{m^2}{16} (2t_{\text{оп}} + t_{\text{обч}}^1) \right] + \frac{15m^2}{16} * t_{\text{обч}}^2 + (k-1) * m^2 * t_{\text{обч}}^2 + \frac{m^2}{16} 2t_{\text{оп}}} \cong$$

$$\cong \frac{2[2t_{\text{оп}} + t_{\text{обч}}^1 + (16k-1) * t_{\text{обч}}^2 + t_{\text{оп}}]}{2 * (2t_{\text{оп}} + t_{\text{обч}}^1) + (16k-1) * t_{\text{обч}}^2 + 2t_{\text{оп}}} \cong 1.94$$

### Підсумок коефіцієнти використання підсистем ядра рівні:

$$KKD_{\text{у}}^{2\text{П}} = 32.3\%.$$

$$KKD_{\text{оп}}^{2\text{П}} = 3.03\%.$$

$$KKD_{\text{обч}}^{2\text{П}} = 97\%.$$

**Коефіцієнт прискорення = 1.94.**

### 3.3.2 Обидва масиви поміщаються в кеш-пам'ять даних другого рівня

**Загальний час виконання одного потоку з k ітерацій:**

$$T_{\text{ЗАГ}} \cong \frac{m^2}{16} (2t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + \frac{15m^2}{16} * t_{\text{ОБЧ}}^2 + (k-1) * \left[ \frac{m^2}{16} (2t_{L2} + t_{\text{ОБЧ}}^1) + \frac{15m^2}{16} * t_{\text{ОБЧ}}^2 \right] + \frac{m^2}{16} t_{\text{ОП}} \cong \frac{m^2}{16} 3t_{\text{ОП}} + (k-1) \frac{m^2}{16} 2t_{L2} + k \frac{m^2}{16} (t_{\text{ОБЧ}}^1 + t_{\text{ОБЧ}}^2)$$

**Загальний час послідовного виконання 2-х потоків:**

$$T_{\text{ЗАГ}} \cong 2 \left\{ \frac{m^2}{16} 3t_{\text{ОП}} + (k-1) \frac{m^2}{16} 2t_{L2} + k \frac{m^2}{16} (t_{\text{ОБЧ}}^1 + t_{\text{ОБЧ}}^2) \right\}$$

**Загальний час паралельного виконання 2-х потоків:**

$$T_{\text{ПАР}}^{2\text{П}} \cong 2 \left[ \frac{m^2}{16} (2t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + (k-1) * \left[ 2 \frac{m^2}{16} (2t_{L2} + t_{\text{ОБЧ}}^1) \right] + 2 \frac{m^2}{16} t_{\text{ОП}} + k \frac{15m^2}{16} * t_{\text{ОБЧ}}^2 \right] \cong \frac{m^2}{16} \{ 6t_{\text{ОП}} + (k-1) * 4t_{L2} + k * (2t_{\text{ОБЧ}}^1 + 15t_{\text{ОБЧ}}^2) \}$$

**Середній час обчислення чергового наближення в точці при паралельному виконанні 2-х потоків:**

$$t_{\text{ТОЧКИ}}^{\text{ср}} \cong \frac{T_{\text{ПАР}}^{2\text{П}}}{k * 2m^2} \cong \frac{\frac{m^2}{16} \{ 6t_{\text{ОП}} + (k-1) * 4t_{L2} + k * (2t_{\text{ОБЧ}}^1 + 15t_{\text{ОБЧ}}^2) \}}{k * 2m^2} \cong \frac{3t_{\text{ОП}} + (k-1) * 2t_{L2} + k * (t_{\text{ОБЧ}}^1 + 7.5 * t_{\text{ОБЧ}}^2)}{16 * k} \cong 13,74$$

**Коефіцієнт використання оперативної пам'яті:**

$$\text{ККД}_{\text{ОП}} = \frac{T_{\text{ПАР}}^{\text{ОП}}}{T_{\text{ОБЩ}}^{\text{ПАР}}} \cong \frac{6 \frac{m^2}{16} t_{\text{ОП}}}{\frac{m^2}{16} \{ 6t_{\text{ОП}} + (k-1) * 4t_{L2} + k * (2t_{\text{ОБЧ}}^1 + 15t_{\text{ОБЧ}}^2) \}} \cong \frac{3t_{\text{ОП}}}{3t_{\text{ОП}} + (k-1) * 2t_{L2} + k * (t_{\text{ОБЧ}}^1 + 7.5 * t_{\text{ОБЧ}}^2)} \cong 0,027$$

### Коефіцієнт використання обчислювальної підсистеми

$$\begin{aligned} \text{ККД}_{\text{ОБЧ}} &= \frac{T_{\text{ПАР}}^{\text{ОБЧ}}}{T_{\text{ЗАГ}}^{\text{ПАР}}} \cong \\ &\cong \frac{2 \frac{m^2}{16} t_{\text{ОБЧ}}^1 + \frac{15m^2}{16} * t_{\text{ОБЧ}}^2 + (k-1) * [2 \frac{m^2}{16} t_{\text{ОБЧ}}^1 + \frac{15m^2}{16} * t_{\text{ОБЧ}}^2]}{2[(\frac{m^2}{16} (2t_{\text{ОП}} + t_{\text{ОБЧ}}^1)] + \frac{15m^2}{16} * t_{\text{ОБЧ}}^2 + (k-1) * [2 \frac{m^2}{16} (2t_{L2} + t_{\text{ОБЧ}}^1) + \frac{15m^2}{16} * t_{\text{ОБЧ}}^2] + 2 \frac{m^2}{16} t_{\text{ОП}}} \cong \\ &\cong \frac{t_{\text{ОБЧ}}^1 + 7.5 * t_{\text{ОБЧ}}^2}{\frac{3t_{\text{ОП}}}{k} + \frac{k-1}{k} * 2t_{L2} + t_{\text{ОБЧ}}^1 + 7.5 * t_{\text{ОБЧ}}^2} \cong 0,866 \end{aligned}$$

### Коефіцієнт використання (завантаження) кеш-пам'яті другого рівня L2

$$\begin{aligned} \text{ККД}_{L2} &\cong \frac{T_{L2}}{T_{\text{ЗАГ}}} \cong \\ &\cong \frac{2 * (k-1) * \frac{m^2}{16} 2t_{L2}}{2(\frac{m^2}{16} (2t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + 2 * (k-1) * \frac{m^2}{16} (2t_{L2} + t_{\text{ОБЧ}}^1) + k \frac{15m^2}{16} * t_{\text{ОБЧ}}^2 + 2 \frac{m^2}{16} t_{\text{ОП}})} \cong \\ &\cong \frac{(k-1) * 2t_{L2}}{3t_{\text{ОП}} + (k-1) * 2t_{L2} + k * 0.5 * t_{\text{ОБЧ}}^1 + k * 7.5 * t_{\text{ОБЧ}}^2} \cong 0,107 \end{aligned}$$

### Коефіцієнт прискорення

$$\begin{aligned} S &\cong \frac{T_{\text{ПОСЛ}}^{2\text{П}}}{T_{\text{ПАР}}^{2\text{П}}} \cong \\ &\cong \frac{2\{\frac{m^2}{16} (2t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + \frac{15m^2}{16} * t_{\text{ОБЧ}}^2 + (k-1) * [\frac{m^2}{16} (2t_{L2} + t_{\text{ОБЧ}}^1) + \frac{15m^2}{16} * t_{\text{ОБЧ}}^2 + \frac{m^2}{16} t_{\text{ОП}}]\}}{2 \frac{m^2}{16} (2t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + \frac{15m^2}{16} * t_{\text{ОБЧ}}^2 + 2(k-1) * \frac{m^2}{16} (2t_{L2} + t_{\text{ОБЧ}}^1) + (k-1) \frac{15m^2}{16} * t_{\text{ОБЧ}}^2 + 2 \frac{m^2}{16} t_{\text{ОП}}} \cong \\ &\cong \frac{3t_{\text{ОП}} + (k-1) * 2t_{L2} + k(t_{\text{ОБЧ}}^1 + 15 * t_{\text{ОБЧ}}^2)}{3t_{\text{ОП}} + (k-1) * 2t_{L2} + k(t_{\text{ОБЧ}}^1 + 7.5 * t_{\text{ОБЧ}}^2)} \cong 1,81 \end{aligned}$$

### Коефіцієнти використання підсистем ядра і прискорення такі:

ККД<sub>УУ</sub> ≈ 24.6%.

ККД<sub>ОП</sub> ≈ 2.7%.

ККД<sub>ОБЧ</sub> ≈ 86.6%.

ККД<sub>L2</sub> ≈ 10.7%.

**Коефіцієнт прискорення = 1.81.**

### 3.3.3 Обидва масиви поміщаються в кеш-пам'ять даних третього рівня

**Загальний час виконання одного потоку з k ітерацій:**

$$T_{\text{ЗАГ}} \cong \frac{m^2}{16} (2t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + \frac{15m^2}{16} * t_{\text{ОБЧ}}^2 + (k-1) * \left[ \frac{m^2}{16} (2t_{L3} + t_{\text{ОБЧ}}^1) + \frac{15m^2}{16} * t_{\text{ОБЧ}}^2 \right] + \frac{m^2}{16} t_{\text{ОП}} \cong \frac{m^2}{16} 3t_{\text{ОП}} + (k-1) \frac{m^2}{16} 2t_{L3} + k \frac{m^2}{16} (t_{\text{ОБЧ}}^1 + t_{\text{ОБЧ}}^2)$$

**Загальний час послідовного виконання 2-х потоків:**

$$T_{\text{ЗАГ}} \cong 2 \left\{ \frac{m^2}{16} 3t_{\text{ОП}} + (k-1) \frac{m^2}{16} 2t_{L3} + k \frac{m^2}{16} (t_{\text{ОБЧ}}^1 + t_{\text{ОБЧ}}^2) \right\}$$

**Загальний час паралельного виконання 2-х потоків:**

$$T_{\text{ПАР}}^{2\text{П}} \cong 2 \left[ \left( \frac{m^2}{16} (2t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + (k-1) * \left[ 2 \frac{m^2}{16} (2t_{L3} + t_{\text{ОБЧ}}^1) \right] \right) + 2 \frac{m^2}{16} t_{\text{ОП}} + k \frac{15m^2}{16} * t_{\text{ОБЧ}}^2 \right] \cong \frac{m^2}{16} \{ 6t_{\text{ОП}} + (k-1) * 4t_{L3} + k * (2t_{\text{ОБЧ}}^1 + 15t_{\text{ОБЧ}}^2) \}$$

**Середній час обчислення чергового наближення в точці при паралельному виконанні 2-х потоків:**

$$t_{\text{ТОЧКИ}}^{\text{ср}} \cong \frac{T_{\text{ПАР}}^{2\text{П}}}{k * 2m^2} \cong \frac{\frac{m^2}{16} \{ 6t_{\text{ОП}} + (k-1) * 4t_{L3} + k * (2t_{\text{ОБЧ}}^1 + 15t_{\text{ОБЧ}}^2) \}}{k * 2m^2} \cong \frac{3t_{\text{ОП}} + (k-1) * 2t_{L3} + k * (t_{\text{ОБЧ}}^1 + 7.5 * t_{\text{ОБЧ}}^2)}{16 * k} \cong 16,84$$

**Коефіцієнт корисного використання УУ (і всього ядра процесора)**

$$t_{\text{УУ}} = \frac{15 \text{команд}}{S = 4} \cong 4\tau$$

$$\rho_{\text{УУ}} = \frac{16.84}{4} = 4.21$$

$$\text{ККД}_{\text{УУ}} = \frac{1}{4,21} \cong 0.238$$

Якщо УУ видає кожні S команд кожні 2 такти, то

$$ККД_{УУ} = \frac{1}{2.1} \cong 0.476$$

**Коефіцієнт використання оперативної пам'яті**

$$\begin{aligned} ККД_{ОП} &= \frac{T_{ПАР}^{ОП}}{T_{ЗАГ}^{ПАР}} \cong \frac{6 \frac{m^2}{16} t_{ОП}}{\frac{m^2}{16} \{6t_{ОП} + (k-1) * 4t_{L3} + k * (2t_{ОБЧ}^1 + 15t_{ОБЧ}^2)\}} \cong \\ &\cong \frac{3t_{ОП}}{3t_{ОП} + (k-1) * 2t_{L3} + k * (t_{ОБЧ}^1 + 7.5 * t_{ОБЧ}^2)} \cong 0,022 \end{aligned}$$

**Коефіцієнт використання обчислювальної підсистеми**

$$\begin{aligned} ККД_{ОБЧ} &= \frac{T_{ПАР}^{ОБЧ}}{T_{ЗАГ}^{ПАР}} \cong \\ &\cong \frac{2 \frac{m^2}{16} t_{ОБЧ}^1 + \frac{15m^2}{16} * t_{ОБЧ}^2 + (k-1) * [2 \frac{m^2}{16} t_{ОБЧ}^1 + \frac{15m^2}{16} * t_{ОБЧ}^2]}{2[(\frac{m^2}{16} (2t_{ОП} + t_{ОБЧ}^1)] + \frac{15m^2}{16} * t_{ОБЧ}^2 + (k-1) * [2 \frac{m^2}{16} (2t_{L3} + t_{ОБЧ}^1) + \frac{15m^2}{16} * t_{ОБЧ}^2] + 2 \frac{m^2}{16} t_{ОП}} \cong \\ &\cong \frac{t_{ОБЧ}^1 + 7.5 * t_{ОБЧ}^2}{\frac{3t_{ОП}}{k} + \frac{k-1}{k} * 2t_{L3} + t_{ОБЧ}^1 + 7.5 * t_{ОБЧ}^2} \cong 0,707 \end{aligned}$$

**Коефіцієнт використання кеш-пам'яті третього рівня L3**

$$\begin{aligned} ККД_{L3} &\cong \frac{T_{l2}}{T_{ЗАГ}} \cong \\ &\cong \frac{2 * (k-1) * \frac{m^2}{16} 2t_{l2}}{2(\frac{m^2}{16} (2t_{ОП} + t_{ОБЧ}^1) + 2 * (k-1) * \frac{m^2}{16} (2t_{L3} + t_{ОБЧ}^1) + k \frac{15m^2}{16} * t_{ОБЧ}^2 + 2 \frac{m^2}{16} t_{ОП}} \cong \\ &\cong \frac{(k-1) * 2t_{l2}}{3t_{ОП} + (k-1) * 2t_{L3} + k * 0.5 * t_{ОБЧ}^1 + k * 7.5 * t_{ОБЧ}^2} \cong 0,265 \end{aligned}$$

### Коефіцієнт прискорення

$$S \cong \frac{T_{\text{ПОСЛ}}^{2\Pi}}{T_{\text{ПАР}}^{2\Pi}} \cong$$

$$\cong \frac{2\left\{\frac{m^2}{16}(2t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + \frac{15m^2}{16} * t_{\text{ОБЧ}}^2 + (k-1) * \left[\frac{m^2}{16}(2t_{L3} + t_{\text{ОБЧ}}^1) + \frac{15m^2}{16} * t_{\text{ОБЧ}}^2 + \frac{m^2}{16}t_{\text{ОП}}\right]\right\}}{2\frac{m^2}{16}(2t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + \frac{15m^2}{16} * t_{\text{ОБЧ}}^2 + 2(k-1) * \frac{m^2}{16}(2t_{L3} + t_{\text{ОБЧ}}^1) + (k-1)\frac{15m^2}{16} * t_{\text{ОБЧ}}^2 + 2\frac{m^2}{16}t_{\text{ОП}}} \cong$$

$$\cong \frac{3t_{\text{ОП}} + (k-1) * 2t_{L3} + k(t_{\text{ОБЧ}}^1 + 15 * t_{\text{ОБЧ}}^2)}{3t_{\text{ОП}} + (k-1) * 2t_{L3} + k(t_{\text{ОБЧ}}^1 + 7.5 * t_{\text{ОБЧ}}^2)} \cong 1,67$$

### Коефіцієнти використання підсистем ядра і прискорення

ККД<sub>уу</sub>=24.6%.

ККД<sub>ОП</sub> =2.2%.

ККД<sub>ОБЧ</sub>=70.7%.

ККД<sub>L3</sub>=26.5%.

**Коефіцієнт прискорення = 1.67.**

### 3.3.4 Масиви не поміщаються в кеш-пам'ять третього рівня

#### Час виконання одного потоку

$$T_{\text{ЗАГ}}^{1\Pi} \cong k\left[\frac{m^2}{16}(2t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + \frac{15}{16}m^2 * t_{\text{ОБЧ}}^2 + \frac{m}{16}2t_{\text{ОП}} + \frac{m^2}{16}t_{\text{ОП}}\right]$$

#### Загальний час послідовного виконання 2-х потоків

$$T_{\text{ПОСЛ}}^{2\Pi} \cong 2k\left[\frac{m^2}{16}(2t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + \frac{15}{16}m^2 * t_{\text{ОБЧ}}^2 + \frac{m}{16}2t_{\text{ОП}} + \frac{m^2}{16}t_{\text{ОП}}\right]$$

#### Загальний час паралельного виконання 2-х потоків

$$T_{\text{ПАР}}^{2\Pi} \cong 2k\left[\frac{m^2}{16}(2t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + \frac{m}{16}2t_{\text{ОП}} + \frac{m^2}{16}t_{\text{ОП}}\right] + k\frac{15}{16}m^2 * t_{\text{ОБЧ}}^2 \cong$$

$$\cong 2k\left[\frac{m^2}{16}(3t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + \frac{m}{16}2t_{\text{ОП}}\right] + k\frac{15}{16}m^2 * t_{\text{ОБЧ}}^2$$



**Середній час обчислення чергового наближення в точці при паралельному виконанні 2-х потоків**

$$t_{\text{Точки}}^{\text{ср}} \cong \frac{T_{\text{ПАР}}^{2\text{П}}}{k * 2m^2} \cong \frac{2k[\frac{m^2}{16}(2t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + \frac{m}{16}2t_{\text{ОП}} + \frac{m^2}{16}t_{\text{ОП}}] + k\frac{15}{16}m^2 * t_{\text{ОБЧ}}^2}{k * 2m^2} \cong$$

$$\cong \frac{3t_{\text{ОП}} + t_{\text{ОБЧ}}^1 + \frac{2t_{\text{ОП}}}{m} + 7.5 * t_{\text{ОБЧ}}^2}{16} \cong 60$$

**Коефіцієнт використання УУ (і всього ядра процесора)**

$$\rho_{\text{УУ}} = \frac{60}{4} = 15$$

$$\text{ККД}_{\text{УУ}} \cong \frac{1}{15} = 0.067$$

**Якщо УУ видає кожні S команд кожні 2 такту, то**

$$\text{ККД}_{\text{УУ}} \cong \frac{8}{60} = 0.133$$

**Коефіцієнт використання оперативної пам'яті при паралельному виконанні 2-х потоків**

$$\text{ККД}_{\text{ОП}} = \frac{T_{\text{ОП}}}{T_{\text{ЗАГ}}} \cong \frac{2k[\frac{m^2}{16}3t_{\text{ОП}} + \frac{m}{16}2t_{\text{ОП}}]}{2k[\frac{m^2}{16}(3t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + \frac{m}{16}2t_{\text{ОП}}] + k\frac{15}{16}m^2 * t_{\text{ОБЧ}}^2} \cong$$

$$\cong \frac{3t_{\text{ОП}} + \frac{2t_{\text{ОП}}}{m}}{3t_{\text{ОП}} + t_{\text{ОБЧ}}^1 + \frac{2t_{\text{ОП}}}{m} + 7.5 * t_{\text{ОБЧ}}^2} \cong 0,8$$

**Коефіцієнт використання обчислювальної підсистеми**

$$\text{ККД}_{\text{ОБЧ}} = \frac{T_{\text{ОБЧ}}}{T_{\text{ЗАГ}}} \cong \frac{t_{\text{ОБЧ}}^1 + 7.5 * t_{\text{ОБЧ}}^2}{3t_{\text{ОП}} + \frac{2t_{\text{ОП}}}{m} + t_{\text{ОБЧ}}^1 + 7.5 * t_{\text{ОБЧ}}^2} \cong 0,2$$

### Загальний час послідовного виконання 2-х потоків

$$T_{\text{ПОСЛ}}^{2\text{П}} \cong 2k \left[ \frac{m^2}{16} (2t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + \frac{15}{16} m^2 * t_{\text{ОБЧ}}^2 + \frac{m}{16} 2t_{\text{ОП}} + \frac{m^2}{16} t_{\text{ОП}} \right]$$

### Загальний час паралельного виконання 2-х потоків

$$\begin{aligned} T_{\text{ПАР}}^{2\text{П}} &\cong 2k \left[ \frac{m^2}{16} (2t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + \frac{m}{16} 2t_{\text{ОП}} + \frac{m^2}{16} t_{\text{ОП}} \right] + k \frac{15}{16} m^2 * t_{\text{ОБЧ}}^2 \cong \\ &\cong 2k \left[ \frac{m^2}{16} (3t_{\text{ОП}} + t_{\text{ОБЧ}}^1) + \frac{m}{16} 2t_{\text{ОП}} \right] + k \frac{15}{16} m^2 * t_{\text{ОБЧ}}^2 \end{aligned}$$

### Коефіцієнт прискорення

$$S \cong \frac{T_{\text{ПОСЛ}}^{2\text{П}}}{T_{\text{ПАР}}^{2\text{П}}} \cong \frac{3t_{\text{ОП}} + t_{\text{ОБЧ}}^1 + 15 * t_{\text{ОБЧ}}^2 + \frac{2t_{\text{ОП}}}{m}}{3t_{\text{ОП}} + t_{\text{ОБЧ}}^1 + 7.5t_{\text{ОБЧ}}^2 + \frac{2t_{\text{ОП}}}{m}} \cong 1,19$$

### Коефіцієнти використання підсистем ядра і прискорення такі:

ККД<sub>уу</sub> = 6.7%.

ККД<sub>ОП</sub> = 80%.

ККД<sub>ОБЧ</sub> = 20%.

**Коефіцієнт прискорення = 1.19.**

## 3.4 Експериментальні дані про виконання програми «Рішення ДР у приватних похідних»

Вихідний код програми приведено у Додатку Б.

Експериментальні дані про виконання програми «Рішення ДР у приватних похідних» наведені в таблиці 3.1 та на рис. 3.1.

Точні значення модельних (теоретичні) і експериментальних даних не збігаються, що і слід було очікувати.

Однак тенденції результатів однозначно однакові.

Таблиця 3.1 - Експериментальні дані про виконання програми «Рішення ДР у приватних похідних»

Розмір	Теоретичне прискорення	Реальне прискорення
64	1,94	1,42
128	1,81	1,45
512	1,67	1,37
$n > 2^{14}$	1,19	1,30

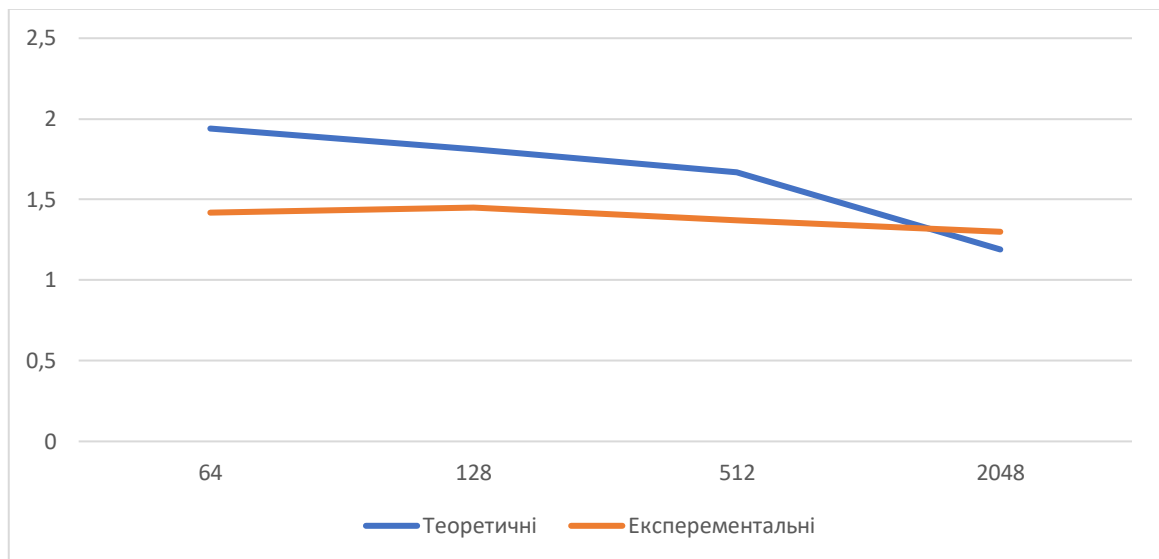


Рис. 3.1 Графік залежності прискорення від розміру матриці

### 3.5 Висновки

У таблиці 3.2 наведено зведені дані про виконання однопоточних і двопоточних програми «Рішення ДР в приватних похідних». Всі дані 32-х розрядні. Кількість ітерацій - 128.

Таблиця 3.2 Результати програми «Рішення ДУ в приватних похідних».

Розмір сітки	Кіл. потоків	ККД <sub>уу</sub> %	ККД <sub>оп</sub> %	ККД <sub>обч</sub> %	ККД <sub>L2</sub> %	ККД <sub>L3</sub> %	Коеф. приск.
64	1	16.4	1.54	98.5	-	-	-
	2	32.3	3	97	-	-	1.94
128	1	15.9	1.49	92.6	5.93	-	-
	2	24.6	2.7	86.6	10.7	-	1.81
512	1	15.4	1.34	82.8	-	15.9	-
	2	24.6	2.2	70.7	-	26.5	1.67
$n > 2^{14}$	1	5.6	67.4	32.6	-	-	-
	2	7.7	80	20	-	100	1.19

1. Як і при дослідженні програми «Множення матриць» коефіцієнт використання УУ (і всього ядра процесора), визначений з моделі ядра процесора, є верхньою межею ефективності ядра процесора.

2. Максимальний зафіксований показник використання УУ (і всього ядра процесора) при виконанні одного потоку далеко від межі і змінюється в досить широкому діапазоні - від 6% до 17% в залежності від розмірності сітки при максимальній продуктивності УУ.

3. Якщо параметри УУ гірші (коефіцієнта  $L$ , що визначає перезавантаження конвеєра команд через команди переходу, дорівнює 2) коефіцієнт використання УУ (і всього ядра процесора) при виконанні одного потоку буде в 2 рази більше. Однак, навіть в гіршому випадку він не буде перевищувати 50%.

4. Те, що коефіцієнт використання УУ значно менше 50% означає, що можливості УУ використовуються неефективно, а значить можливо в одному фізичному ядрі реалізувати кілька потоків одночасно.

5. При виконанні одного потоку коефіцієнти використання компонентів підсистеми виконання команд ядра (обчислювальної компоненти, що складається з кеш-пам'яті даних першого рівня L1D, пристрої Множення і Додавання, кеш-пам'яті другого і третього рівнів), оперативної пам'яті значно залежать від розмірності сітки і далекі від граничних значень. Це означає, що ядро і оперативна пам'ять можуть бути використані для виконання декількох потоків.

6. Реальні коефіцієнти використання підсистеми виконання команд ядра (обчислювальної компоненти, що складається з кеш-пам'яті даних першого рівня L1D, пристрої Множення і додавання) складають 25% від значень, які наведені в таблиці. Це пояснюється тим, що всі перераховані компоненти є конвеєрними пристроями з пропускною спроможністю в 1 такт. При визначенні ж коефіцієнтів використання в таблиці враховувався повний час виконання операцій в кожному компоненті, тобто пристрої розглядались, як комбінаційні пристрої з повним часом виконання операцій 4 такти.

7. Коефіцієнти прискорення при виконанні 2-х поточного варіанту програми «Рішення диференціальних рівнянь в приватних похідних» істотно залежать від розмірності сітки.

8. При малих розмірах сітки, коли дані повністю розміщуються в кеш-пам'яті даних першого рівня L1D ( $m < 64$ ) прискорення максимальне. Це пояснюється тим, що переважна частина загального часу виконання ядра програми визначається роботою через кеш-пам'ять даних першого рівня.

9. При розмірах сітки, коли для роботи з даними використовуються кеш-пам'яті другого і третього рівнів ( $m$  змінюється в межах від 64 до 512) прискорення при виконанні 2-х потоків існує і змінюється в межах від 1.81 до 1.67.

10. При дуже великих розмірах сітки, коли в кеш-пам'ять третього рівня L3 не поміщається сітка наближень і масив значень функції (розмір сітки  $2^{14}$  і більше) без огляду на те, що вся робота з даними здійснюється через ОП, 2-х поточність забезпечує коефіцієнт прискорення 1.19.

### **Загальний висновок**

При реальному виконанні однопоточного варіанту програми «Рішення ДР у приватних похідних» високопродуктивне ядро процесора буде використовуватися не більше ніж на 25%, а обчислювальні компоненти підсистеми виконання команд понад 75% всього часу буде простоювати.

Ресурси продуктивності як підсистеми підготовки команд (УУ), та й підсистеми виконання команд достатні для виконання як мінімум ще одного потоку команд.

## 4 ДОСЛІДЖЕННЯ ВИКОНАННЯ ПРОГРАМИ «ШПФ»

### 4.1 Про швидке перетворення Фур'є

Швидке перетворення Фур'є (ШПФ) часто використовується в різних застосуваннях, зокрема в спектральному аналізі.

Дослідження виконання швидкого перетворення Фур'є (БПФ2) буде проводитися при  $N = 512$  комплексних точок.

Програма обчислення ШПФ по основі 2 при  $N = 1024$  точки складається з 10 ітерацій. У кожній ітерації обчислюється 512 комплексні точки - виконується 512 циклів. Цикли всередині кожної ітерації інформаційно незалежні один від одного. Усередині кожного циклу існують інформаційні залежності між командами.

Кожна наступна ітерація інформаційно залежить від результатів попередньої ітерації, тобто ітерації виконуються строго послідовно.

Кількість ітерацій  $k = \lg_2 N$ . При  $N=1024$   $k = 10$ .

У загальному випадку всі дані, які необхідні для обчислення ШПФ, а також проміжні і підсумкові результати, є комплексними числами. Розрядність даних -32. Формат представлення чисел - плаваюча точка.

Вхідний масив даних - 512 комплексних числа. Масив повертаючих коефіцієнтів - 512 комплексних чисел. Вихідний масив даних - 512 комплексних числа.

Масиви проміжних результатів чергової ітерації - 512 комплексних числа.

### 4.2 Особливості виконання програми ШПФ

При виконанні першої ітерації програми вхідні дані і повертаючі коефіцієнти знаходяться в оперативній пам'яті.

При виконанні всіх наступних ітерацій всі дані знаходяться в кеш-пам'яті даних першого рівня L1D.

У програмі обчислені проміжні дані записуються на місце даних попередньої ітерації. Це означає, що всі записи проміжних результатів виконуються в кеш-пам'ять даних першого рівня L1D.

Ядро процесора має наступні параметри:

1. Кеш-пам'ять даних першого рівня L1D це конвеєрний пристрій з пропускнуою спроможністю 1 такт і може кожен такт здійснювати 2 читання і 1 запис. Повний час звернення до кеш-пам'ять L1D 4такти.

2. Читання з оперативної пам'яті (ОП) здійснюється блоками по 64 байти.

3. Запис в оперативну пам'ять здійснюється в залежності від того, чи є дані в кеш-пам'яті. Якщо дані є в кеш-пам'яті, то запис виконується в кеш-пам'ять. Якщо даних немає в кеш-пам'яті, то виконується запис в ОП 32-розрядними словами (або 64-розрядними подвійними словами) в залежності від форматів оброблюваних даних.

4. Функціональні пристрої Множення і Складання конвеєрні з пропускнуою спроможністю 1 такт. Повний час виконання операцій 4такти.

5. Одночасно підсистема виконання може виконувати не менше 4-х команд.

6. Підсистема генерації команд здатна генерувати не менше 4-х команд кожен такт.

Основна ділянка програми визначення пари комплексних точок чергової ітерації (ядро цієї програми) на умовній мові асемблера може мати такий вигляд (рис. 4.1).

1. **LOAD**  $Re\ X\ 2 \rightarrow R1$ , читання речової складової X2
2. **LOAD**  $Im\ X\ 2 \rightarrow R2$ , читання уявної складової X2
3. **LOAD**  $Re\ W \rightarrow R3$ , читання речової складової W
4. **LOAD**  $Im\ W \rightarrow R4$ , читання уявної складової W
5. **MPY**  $R1 * R3 \rightarrow R5$  - Множення речових складових
6. **MPY**  $R2 * R4 \rightarrow R6$  Множення уявних складових
7. **MPY**  $R1 * R3 \rightarrow R7$  Множення речової на уявну
8. **MPY**  $R4 * R2 \rightarrow R8$  Множення уявної на речову
9. **LOAD**  $Re\ X\ 1 \rightarrow R9$ , читання речової складової X1
10. **LOAD**  $Im\ X\ 1 \rightarrow R10$  читання уявної складової X1
11. **SUB**  $R5 - R6 \rightarrow R11$  речова складова  $X2 * W$
12. **ADD**  $R7 + R8 \rightarrow R12$  уявна складова  $X2 * W$
13. **ADD**  $R9 + R11 \rightarrow R5$   $Re\ X\ 1 + Re\ X\ 2 * W$
14. **ADD**  $R10 + R12 \rightarrow R6$   $Im\ X\ 1 + Im\ X\ 2 * W$
15. **SUB**  $R1 - R11 \rightarrow R7$   $Re\ X\ 2 - Re\ X\ 2 * W$
16. **SUB**  $R2 - R12 \rightarrow R8$   $Im\ X\ 2 - Im\ X\ 2 * W$
17. **STORE**  $R5$   $Re\ X\ 1 + Re\ X\ 2 * W$
18. **STORE**  $R6$   $Im\ X\ 1 + Im\ X\ 2 * W$
19. **STORE**  $R7$   $Re\ X\ 2 - Re\ X\ 2 * W$
20. **STORE**  $R8$   $Im\ X\ 2 - Im\ X\ 2 * W$
21. **JMP** початок циклу

Рис. 4.1 – Приклад ядра програми розрахунку чергового значення шуканої функції

Всього в наведеному спрощеному прикладі ядра програми обчислення чергової пари комплексних точок черговий ітерації (2-а - 10-а ітерації) 21 команда.

В ядрі програми відсутні команди редукації.

Все команди LOAD (читання операндів з підсистеми пам'яті в регістри) інформаційно незалежні і можуть починатися навіть одночасно.

В ядрі програми можна виділити 13 випадків інформаційної залежності.

#### **1-й випадок**

5-а команда **MPY R1 – R3 → R5** - (Множення дійсних складових) інформаційно залежна від 1-ї та 2-ї команд LOAD, і може починатися тільки після їх завершення.

#### **2-й випадок**

6-а команда **MPY R2 \* R4 → R6** Множення уявних складових інформаційно залежна від 2-ї та 4-ї команд LOAD і може починатися тільки після їх завершення.

#### **3-й випадок**

7-а команда **MPY R1 \* R3 → R7** ( Множення дійсної компоненти на уявну) інформаційно залежна від 1-ї та 3-ї команд LOAD і може починатися тільки після їх завершення.

#### **4-й випадок**

8-а команда **MPY R4 \* R2 → R8** (Множення уявної компоненти на дійсну) інформаційно залежна від 2-ї та 4-ї команд LOAD і може починатися тільки після їх завершення.

#### **5-й випадок**

11-а команда **SUB R5 – R6 → R11**, яка обчислює дійсну складову **X2 \* W**, інформаційно залежна від 5-ї та 6-ї команд MPY і може починатися тільки після їх завершення.

#### **6-й випадок**

12-а команда ADD інформаційно залежна від 7-ї та 8-ї команд MPY і може починатися тільки після їх завершення.

13-а команда ADD інформаційно залежна від 9-ї LOAD та 11-ї SUB команд і може починатися тільки після їх завершення.

#### **7-й випадок**

14-а команда ADD інформаційно залежна від 10-ї LOAD команди і 12-й команди ADD і може починатися тільки після їх завершення.

#### **8-й випадок**

15-а команда SUB інформаційно залежна від 1-ї LOAD команди і 11-ї команди SUB і може починатися тільки після їх завершення.



**9-й випадок**

16-а команда SUB інформаційно залежна від 2-ї LOAD команди і 12-ї команди ADD і може починатися тільки після їх завершення.

**10-й випадок**

17-а команда STORE інформаційно залежна від 13-ї команди ADD і може починатися тільки після її завершення.

**11-й випадок**

18-а команда STORE інформаційно залежна від 14-ї команди ADD і може починатися тільки після її завершення.

**12-й випадок**

19-а команда STORE інформаційно залежна від 15-ї команди ADD і може починатися тільки після її завершення.

**13-й випадок**

20-а команда STORE інформаційно залежна від 16-ї команди ADD і може починатися тільки після її завершення.

**В ядрі процесора із зсувом на один такт одночасно можуть починатися:**

1-а і 2-а команди LOAD, що забезпечують операнди для 5-ї команди MPY.

3-тя і 4-а команди LOAD, що забезпечують операнди для 6-ї команди MPY.

9-а і 10-а команди LOAD, що забезпечують операнди для 13-ї та 14-ї команд ADD, відповідно.

**Після завершення:**

1-ї і 2-ї команд LOAD може починатися 5-а команда MPY.

4-ї і 5-ї команд LOAD може починатися 6-а команда MPY.

7-ї і 8-ї команди MPY може починатися навіть одночасно з командами 5 і 6.

**Усі команди MPY формують проміжні результати.**

11-а команда SUB може починатися після завершення 5-ї і 6-ї команд MPY.

12-а команда ADD може починатися після завершення 7-ї і 8-ї команд MPY.

13-а команда ADD може починатися після завершення 11-ї команди SUB і 9-ї команди LOAD.

14-а команда ADD може починатися після завершення 12-ї команди ADD і 10-ї команди LOAD.

15-а команда ADD може починатися після завершення 11-ї команди SUB і 1-ї команди LOAD.

16-а команда SUB може починатися після завершення 12-ї команди ADD і 2-ї команди LOAD.

17-а команда STORE може починатися після завершення 13-ї команди ADD.

18-а команда STORE може починатися після завершення 14-ї команди ADD.

19-а команда STORE може починатися після завершення 15-ї команди SUB.

20-а команда STORE може починатися після завершення 16-ї команди SUB.

4 команди STORE можуть бути суміщені з виконанням команд LOAD наступної пари комплексних точок (в наступному циклі). Тому часи виконання цих команд STORE можна не враховувати в часі виконання одного циклу обчислення чергової пари комплексних точок.

### 4.3 Перша ітерація

Всі дані розташовані в оперативній пам'яті (ОП).

З оперативної пам'яті дані зчитуються блоками, що складаються з 64 байтів. У кожному блоці міститься 16 32-х розрядних елементів.

Так як запис обчислених даних здійснюється на місце прочитаних даних, то всі записи виконуються в кеш-пам'ять даних першого рівня L1D.

Розрізняють такі групи послідовностей команд:

1-я група (обчислення першої пари комплексних точок в першому стовпці чергового блоку 64-х байтних даних з ОП - 16 32-х розрядних елементів).

В оперативну пам'ять (ОП) звертаються 6 команд LOAD. Команди звернення до ОП виконуються послідовно.

У кеш-пам'ять даних першого рівня L1D звертаються 4 команди STORE при обчисленні наступної пари комплексних точок.

Для БПФ2 з 512 комплексних точок (512 дійсних і 512 уявних складових) кількість повторень команд першої групи (1 перший елемент в першому ряду колонки \* кількість колонок по 16 елементів)

$$n_1 = 1 * \frac{N}{2 * 16} = \frac{N}{2 * 16} = \frac{1024}{2 * 16} 32$$

#### Час виконання 1-ї групи команд

$$T_1 \cong 6t_{\text{ОП}} + 4t_{\text{СКЛ}} \cong 6 * 256\tau + 4 * 4\tau = 1552\tau$$

Часи виконання 4-х команд STORE при обчисленні наступної пари комплексних точок не враховуються, так як вони поєднуються з основними обчисленнями.

2-а група (обчислюються 15 пар комплексних точок у кожному стовпчику).

Всі звернення в підсистему пам'яті виконуються в кеш-пам'ять даних першого рівня L1D.

Час обчислення однієї пари комплексних точок визначається за умови, що:

Повні часи виконання операцій  $t_{L1D} = 4\tau$ ;  $t_{УМН} = 4\tau$ ;  $t_{СЛЖ} = 4\tau$ .

Операції виконуються в конвеєрному режимі;

Кеш-пам'ять даних першого рівня L1D може кожен такт виконувати 2 операції читання і одну операцію запис.

Запис даних в кеш-пам'ять даних першого рівня L1D поєднується з читанням даних.

**Час виконання 2-ї групи команд**

$$T_2 \cong t_{L1D} + 13\tau \cong 4\tau + 13\tau \cong 17\tau$$

**Кількість повторень команд другої групи (15 раз в колонці \* 32 колонки)**

$$n_2 = 15 * 32 = 480$$

**Час виконання першої ітерації (обчислення 512 пар комплексних точок):**

$$T_{ИТЕР}^1 \cong n_1 * T_1 + n_2 * T_2 \cong 32 * 1552\tau + 480 * 17\tau \cong 57824\tau$$

**Середній час обчислення однієї пари комплексних точок при виконанні 1-ї ітерації**

$$T_{ИТЕР}^{1\text{сред}} \cong \frac{T_{ИТЕР}^1}{512} = \frac{57824\tau}{512} = 112.94\tau$$

#### **4.4 Друга і наступні ітерації обчислення 512 пар комплексних точок**

Всі дані знаходяться в кеш-пам'яті даних першого рівня L1D і, відповідно, всі звернення в підсистему пам'яті виробляються в кеш-пам'ять даних першого рівня L1D.

Час обчислення однієї пари комплексних точок визначається за умови, що:

Повні часи виконання операцій  $t_{L1D} = 4\tau$ ;  $t_{МНЖ} = 4\tau$ ;  $t_{СКЛ} = 4\tau$ .

Операції виконуються в конвеєрному режимі;

Кеш-пам'ять даних першого рівня L1D може кожен такт виконувати 2 операції читання і одну операцію запис.

Запис даних в кеш-пам'ять даних першого рівня L1D поєднується з читанням даних.

**Час обчислення однієї пари комплексних точок в і-й ітерації**

$$T^i \cong t_{L1D} + 13\tau \cong 4\tau + 13\tau \cong 17\tau$$

**Час виконання 512 пар комплексних точок в і-й ітерації**

$$T_{\text{ИТЕР}}^i = 512 * T^i = 512 * 17\tau = 8704\tau$$

**Час виконання 9 ітерацій**

$$T_{\text{ИТЕР}}^9 = 9 * T_{\text{ИТЕР}}^i = 9 * 8704\tau = 78336\tau$$

**Загальний час виконання всіх 10 ітерацій**

$$T_{\text{ЗАГ}} = T_{\text{ИТЕР}}^1 + 9 * T_{\text{ИТЕР}}^i \cong 57824\tau + 78336\tau = 136160\tau$$

**Середній час обчислення однієї пари комплексних точок після всіх ітерацій**

$$T_{\text{ЗАГ}}^{\text{ср}} \cong \frac{136160\tau}{5120} = 26.6\tau$$

**Показники ефективності**

Як і раніше коефіцієнт використання УУ

$$\text{ККД}_{\text{УУ}} = \frac{1}{\rho_{\text{ФУ}}},$$

де  $\rho_{\text{ФУ}}$ - коефіцієнт навантаження ФУ формула (1.1).

Кожна група складається з 20 архітектурних команд. При ступеня суперскалярності S, 20 архітектурних команд буде згенеровано за 20/S тактів за умови, що S команд видаються кожен такт.

Час генерації однієї групи команд з 20 архітектурних команд

$$t_{\text{УУ}} = S * \frac{20 \text{команд}}{S = 4} \cong S * 5\tau,$$

де  $S$  - кількість тактів, за які УУ генерує  $S$  команд.

При  $S=1$ ,  $t_{уу} \cong 5\tau$ .

$$t_{\text{пари}}^{\text{ср}} \cong 26.6\tau$$

$$\rho_{уу} \cong \frac{26.6}{5} \cong 5.32$$

$$\text{ККД}_{уу} \cong \frac{1}{5.32} \cong 0.188$$

Якщо УУ видає  $S$  команд кожні 2 такти, то

$$\text{ККД}_{уу} \cong \frac{10}{29.7} \cong 0.376$$

### **Коефіцієнт використання ОП**

$$\text{ККД}_{\text{ОП}} = \frac{T_{\text{ОП}}}{T_{\text{ЗАГ}}} \cong \frac{32 * 6t_{\text{ОП}}}{136160} \cong \frac{49162}{136160} \cong 0.361$$

### **Коефіцієнт використання кеш-пам'яті даних першого рівня L1D**

$$\text{ККД}_{L1D} = \frac{T_{L1D}}{T_{\text{ОБЩ}}} \cong \frac{5120 * 7\tau}{136160\tau} \cong \frac{35840}{136160} \cong 0.263$$

### **Коефіцієнт використання пристрою Множення**

$$\text{ККД}_{\text{МНЖ}} = \frac{T_{\text{МНЖ}}}{T_{\text{ЗАГ}}} \cong \frac{5120 * 4\tau}{136160\tau} \cong \frac{20480}{136160} \cong 0.15$$

### **Коефіцієнт використання пристрою складання**

$$\text{ККД}_{\text{СКЛ}} = \frac{T_{\text{СКД}}}{T_{\text{ЗАГ}}} \cong \frac{5120 * 6\tau}{136160\tau} \cong \frac{30720}{136160} \cong 0.226$$

Аналіз коефіцієнтів використання підсистем ядра показує, що вони задіяні малу частину (не більше ніж по 25%) загального часу виконання програми, а значить є всі необхідні передумови реалізації віртуальної багатоядерності.

#### 4.5 Дослідження реалізацій двопоточності при виконанні програми ШПФ

**Час виконання всіх 10 ітерацій одного потоку**

$$T_{ЗАГ}^{1ПОТОКУ} \cong 57824\tau + 78336\tau = 136160\tau$$

**Час послідовного виконання всіх 10 ітерацій двох потоків**

$$T_{ПОСЛ}^{2ПОТОКІВ} \cong 2 * T_{ЗАГ}^{1ПОТОКУ} \cong 272320\tau$$

**При паралельному виконанні 2-х потоків:**

послідовно виконуються всі дії, пов'язані зі зверненнями в ОП;

паралельно виконуються (поєднуються) всі дії без звернень в ОП;

**Час паралельного виконання всіх 10 ітерацій двох потоків складе**

$$T_{ПАР}^{2ПОТОКІВ} \cong 2 * T_{ОП} + (5120 - 32) * t_{ОБЧ} \cong 2 * 32 * 1552\tau + 5088 * 20\tau \cong 99328\tau + 101760\tau \cong 201088\tau$$

**Коефіцієнт прискорення дорівнює**

$$\frac{T_{ПОСЛ}^{2ПОТОКІВ}}{T_{ПАР}^{2ПОТОКІВ}} \cong \frac{272320\tau}{201088\tau} = 1.35$$

**Середній час обчислення однієї пари комплексних точок при паралельному виконанні 2-х потоків**

$$T_{ЗАГ}^{ср} \cong \frac{201088\tau}{2 * 5120} = 19.64\tau$$

**Коефіцієнт використання УУ (і всього ядра процесора)**

При  $S=1$ ,  $t_{уу} \cong 5\tau$ .

$$t_{пари}^{ср} \cong 19.64\tau$$

$$\rho_{уу} \cong \frac{19.64}{5} \cong 3.93$$

$$ККД_{уу} \cong \frac{1}{3.93} \cong 0.255$$

Якщо УУ видає S команд кожні 2 такти, то

$$KKД \cong \frac{10}{19.64} \cong 0.51$$

**Коефіцієнт використання ОП**

$$KKД_{ОП} = \frac{T_{ОП}}{T_{ЗАГ}} \cong \frac{2*32*6t_{ОП}}{201088\tau} \cong \frac{98304}{201088\tau} \cong 0.489$$

**Коефіцієнт використання кеш-пам'яті даних першого рівня L1D**

$$KKД_{L1D} = \frac{T_{L1D}}{T_{ЗАГ}} \cong \frac{5120*14\tau}{201088\tau} \cong 0.356$$

**Коефіцієнт корисного використання пристрою Множення**

$$KKД_{МНЖ} = \frac{T_{МНЖ}}{T_{ЗАГ}} \cong \frac{5120*8\tau}{201088\tau} \cong 0.204$$

**Коефіцієнт корисного використання пристрою Складання**

$$KKД_{СКЛ} = \frac{T_{СКЛ}}{T_{ЗАГ}} \cong \frac{5120*12\tau}{201088\tau} \cong 0.306$$

#### **4.6 Експериментальні дані про виконання програми «Швидке перетворення Фур'є»**

Вихідний код програми приведено у Додатку В.

Експериментальні дані про виконання програми «БПФ» приведені в таблиці 4.1 и на рис.

4.1.

Таблиця 4.1 - Експериментальні дані про виконання програми «БПФ»

Розмір, комп. точок	Теоретичне прискорення	Реальне прискорення
512	1,34	1,33

#### 4.7 Висновки

1. Коефіцієнт використання компонентів ядра процесора не перевищує 25%.
2. Причому це значення є верхньою межею коефіцієнта використання ядра процесора.
3. Реальне значення коефіцієнта використання ядра процесора буде менше внаслідок того, що при дослідженні програми не враховані організуючі команди, а також перепорядкування вихідного масиву даних.
4. Основною причиною низького коефіцієнта використання ядра процесора є інформаційні залежності між обчислювальними командами. Через це не вдається використовувати потенціал конвеєрних обчислювальних пристроїв Множення і Складання і одночасної їх роботи.
5. Програма обчислення ШПФ є прикладом того, що підсистема пам'яті не завжди є гальмом обчислювального процесу, так як переважна кількість звернень в підсистему пам'яті виконується в швидку кеш-пам'ять даних першого рівня L1D з максимальним поєднанням.
6. Співвідношення команд звернення до підсистему пам'яті і команд обробки даних 1: 1, тобто 10 команд звернення до підсистему пам'яті і 10 команд обробки даних.
7. При паралельному виконанні 2-х програм в одному фізичному ядрі коефіцієнт прискорення може досягати значення 1.34, а коефіцієнти використання компонентів ядра не перевищують 50%.



## 5 ОХОРОНА ПРАЦІ

### 5.1 Загальні питання з охорони праці

Умови праці на робочому місці, безпека технологічних процесів, машин, механізмів, устаткування та інших засобів виробництва, стан засобів колективного та індивідуального захисту, що використовуються працівником, а також санітарно-побутові умови повинні відповідати вимогам нормативних актів про охорону праці. В законі України «Про охорону праці» [1] визначається, що охорона праці - це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності.

При роботі з обчислювальною технікою змінюються фізичні і хімічні фактори навколишнього середовища: виникає статична електрика, електромагнітне випромінювання, змінюється температура і вологість, рівень вміст кисню і озону в повітрі. Повітря забруднюється шкідливими хімічними речовинами антропогенного походження за рахунок деструкції полімерних матеріалів, які використовуються для обробки приміщень та обладнання. Неправильна організація робочого місця сприяє загальному і локальній напрузі м'язів ший, тулуба, верхніх кінцівок, викривлення хребта і розвитку остеохондрозу. На всіх підприємствах, в установах, організаціях повинні створюватися безпечні і нешкідливі умови праці. Забезпечення цих умов покладається на власника або уповноважений ним орган (далі роботодавець). Умови праці на робочому місці, безпека технологічних процесів, машин, механізмів, устаткування та інших засобів виробництва, стан засобів колективного та індивідуального захисту, що використовуються працівником, а також санітарно-побутові умови повинні відповідати вимогам нормативних актів про охорону праці. Роботодавець повинен впроваджувати сучасні засоби техніки безпеки, які запобігають виробничому травматизмові, і забезпечувати санітарно-гігієнічні умови, що запобігають виникненню професійних захворювань працівників. Він не має права вимагати від працівника виконання роботи, поєднаної з явною небезпекою для життя, а також в умовах, що не відповідають законодавству про охорону праці. Працівник має право відмовитися від дорученої роботи, якщо створилася виробнича ситуація, небезпечна для його життя чи здоров'я або людей, які його оточують, і навколишнього середовища.

Державна політика в галузі охорони праці визначається відповідно до Конституції України Верховною Радою України і спрямована на створення належних, безпечних і здорових умов праці, запобігання нещасним випадкам та професійним захворюванням. Відповідно до

статті 3 Закону України «Про охорону праці» [1] (далі – Закону) законодавство про охорону праці складається з Закону, Кодексу законів про працю України [2], Закону України "Про загальнообов'язкове державне соціальне страхування від нещасного випадку на виробництві та професійного захворювання, які спричинили втрату працездатності" [3] та прийнятих відповідно до них нормативно-правових актів, норм міжнародного договору (ратифіковані Конвенції і Рекомендації МОТ, директиви Європейської Ради).

Користувачі персональних комп'ютерів, для яких ця робота є головною, підлягають медичним оглядам: попереднім — під час влаштування на роботу і періодичним — протягом професійної діяльності раз на два роки. Жінок з часу встановлення вагітності та в період годування дитини грудьми до роботи з ПК не допускають.

Наявні трудові відносини між працівниками і роботодавцями в Україні за темою дипломного проекту регулюються Кодексом законів про працю (КЗпП) України, відповідно до якого права працюючої людини на охорону праці охороняються всебічно та норми охорони праці неухильно інтегровані до правил внутрішнього розпорядку організації/підприємства.

## **5.2 Аналіз стану умов праці**

У підрозділі надають аналіз умов праці на робочому місці, зокрема, стан та характеристики приміщень, вимоги до організації робочого місця, навантаження на напруженість процесу праці.

Робота над дослідженням ефективності віртуальної багатопоточності в ядрах сучасних процесорів проходитиме в побутовому приміщенні. Для даної роботи достатньо однієї людини, для якої надано робоче місце зі стаціонарним комп'ютером.

### **5.2.1 Вимоги до приміщень**

Геометричні розміри приміщення зазначені у таблиці 4.1. Для зручності спільної роботи з іншими працівниками (обговорення ідей, з'ясування проблем і т.д.) в кімнаті є диван і журнальний стіл. Задля дотримання визначеного рівня мікроклімату в будівлі встановлено систему опалення та кондиціонування.

Для забезпечення потрібного рівня освітленості кімната має вікно та систему загального рівномірного освітлення, що встановлена на стелі. Для дотримання вимог пожежної безпеки встановлено порошковий вогнегасник та систему автоматичної пожежної сигналізації.

Таблиця 4.1 – Розміри робочого приміщення

Найменування	Значення
Довжина, м	6
Ширина, м	4
Висота, м	2,5
Площа, м <sup>2</sup>	24
Об'єм, м <sup>3</sup>	60

Згідно до санітарних норм мікроклімату виробничих приміщень згідно ДСН 3.3.6.042-99 [6] розмір площі для одного робочого місця оператора персонального комп'ютера має бути не менше 6 кв.м, а об'єм — не менше 20 куб. м. Отже, дане приміщення цілком відповідає зазначеним нормам.

### 5.2.2 Вимоги до організації місця праці

Робоче місце – це зона простору, що оснащена необхідним устаткуванням, де відбувається трудова діяльність одного працівника чи групи працівників [5].

Проведені дослідження показують, що при раціональній організації робочих місць продуктивність праці зростає знати на 15-25 %.

Основні ергономічні вимоги до проектування робочого місця в системі «людина – техніка – виробниче середовище» приведені у таблиці 4.2:

Таблиця 4.2 - Характеристики робочого місця

Найменування параметра	Фактичне значення	Нормативне значення
Висота робочої поверхні, мм	750	680 ÷ 800
Висота простору для ніг, мм	730	не менше 600
Ширина простору для ніг, мм	660	не менше 500
Глибина простору для ніг, мм	700	не менше 650
Висота поверхні сидіння, мм	470	400 ÷ 500
Ширина сидіння, мм	400	не менше 400
Глибина сидіння, мм	400	не менше 400
Висота поверхні спинки, мм	600	не менше 300
Ширина опорної поверхні спинки, мм	500	не менше 380
Радіус кривини спинки в горизонтальній площині, мм	400	400
Відстань від очей до екрану дисплея, мм	800	700 ÷ 800

Гігієнічні вимоги визначають умови життєдіяльності і працездатності людини у процесі взаємодії з технікою і середовищем; показниками є рівень освітлення, температура, вологість, шум, вібрація, токсичність, загазованість тощо згідно ДСН 3.3.6.042-99 [6].

Антропометричні вимоги визначають відповідність конструкцій техніки антропометричним характеристикам людини (зріст, розміри тіла та окремі рухові ланки). Показниками є раціональна робоча поза, оптимальні зони досягнення, раціональні трудові рухи.

### 5.2.3 Навантаження та напруженість процесу праці

За фізичним навантаженням робота відноситься до категорії легкі роботи (Ia), її виконують сидячи з періодичним ходінням. Щодо характеру організування виконання дипломної роботи, то він підпадає під нав'язаний режим, оскільки певні розділи роботи необхідно виконати у встановлені конкретні терміни. За ступенем нервово-психічної напруги виконання роботи можна віднести до II – III ступеня і кваліфікувати як помірно напружений – напружений за умови успішного виконання поставлених завдань.

Під час виконання робіт використовують ПК та периферійні пристрої (лазерні та струменеві), що призводить до навантаження на окремі системи організму. Такі перекося у напруженні різних систем організму, що трапляються під час роботи з ПК, зокрема, значна напруженість зорового аналізатора і довготривале малорухоме положення перед екраном, не тільки не зменшують загального напруження, а навпаки, призводять до його посилення і появи стресових реакцій.

Найбільшому ризику виникнення різноманітних порушень піддаються: органи зору, м'язово скелетна система, нервово-психічна діяльність, репродуктивна функція у жінок.

Тобто наявні психофізіологічні небезпечні та шкідливі фактори:

а) фізичного перевантаження:

- статичного;
- динамічного.

б) нервово-психічного перевантаження:

- розумового перенапруження;
- монотонності праці;
- перенапруження аналізаторів;
- емоційних перевантажень.

Рекомендовано застосування екранних фільтрів, локальних світлофільтрів (засобів індивідуального захисту очей) та інших засобів захисту, а також інші профілактичні заходи [9].

Роботу за дипломним проектом визнано, таку, що займає 50% часу робочого дня та за восьмигодинної робочої зміни рекомендовано встановити додаткові регламентовані перерви тривалістю 15 хв через кожну годину роботи.

### 5.3 Виробнича санітарія

На підставі аналізу небезпечних та шкідливих факторів при виробництві (експлуатації), пожежної безпеки можуть бути надалі вирішені питання необхідності забезпечення працюючих достатньою кількістю освітлення, вентиляції повітря, організації заземлення, тощо.

#### 5.3.1 Аналіз небезпечних та шкідливих факторів при виробництві (експлуатації) виробу

Роботу, пов'язану з ЕОП з ВДТ, у тому числі на тих, які мають робочі місця, обладнані ЕОМ з ВДТ і ПП, виконують із забезпеченням виконання НПАОП 0.00-7.15-18 [9] «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», які встановлюють вимоги безпеки до обладнання робочих місць, до роботи із застосуванням ЕОМ з ВДТ і ПП. Переважно роботи за проектами виконують у кабінетах чи інших приміщеннях, де використовують різноманітне електрообладнання, зокрема персональні комп'ютери (ПК) та периферійні пристрої.

Основними робочими характеристиками персонального комп'ютера є наступні:

- робоча напруга  $U = +220\text{В} \pm 5\%$ ;
- робочий струм  $I = 2\text{А}$ ;
- споживана потужність  $P = 350\text{Вт}$ .

Робоче місце має відповідати вимогам Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин, затверджених постановою Головного державного санітарного лікаря України від 10.12.98 N 7 [5].

Аналіз небезпечних та шкідливих виробничих факторів виконується у табличній формі (табл. 4.3).

Таблиця 4.3 – Аналіз небезпечних і шкідливих виробничих факторів

Небезпечні і шкідливі виробничі фактори	Джерела факторів (види робіт)	Кількісна оцінка	Нормативні документи
<b>Фізичні</b>			
- підвищений рівень напруги електричної мережі, замикання якої може відбутися через тіло людини	-//-	4	[7]
- недостатність природного світла	порушення умов праці (вимог до приміщень)	2	[8]
- недостатнє освітлення робочої зони	порушення гігієнічних параметрів виробничого середовища	3	[8]
<b>Психофізіологічні:</b>			
- нервово-психічна перевантаження (розумове, перенапруження аналізаторів-зорових)	- пошук інформації для постановки теми; - пошук та аналіз аналогів і літератури; - пошук наявних технологій, моделювання та аналіз алгоритмів; - виконання роботи за темою диплома, тестування; - оформлення роботи	4	[9] [5]
- фізичні (статичне – сидіння)	порушення умов праці (організації місця праці-сидіння користувача, ) та організації робочого часу - безперервна робота)	2	[9] [5]

### 5.3.2 Пожежна безпека

Небезпека розвитку пожежі на обчислювальному центрі обумовлюється застосуванням розгалужених систем електроживлення ЕОМ, вентиляції і кондиціонування. Небезпека загоряння пов'язана з особливістю комп'ютерів – із значною кількістю щільно розташованих на монтажній платі і блоках електронних вузлів і схем, електричних і комутаційних кабелів, резисторів, конденсаторів, напівпровідникових діодів і транзисторів. Надійна робота окремих елементів і мікросхем в цілому забезпечується тільки в певних інтервалах температури, вологості і при заданих електричних параметрах. При відхиленні реальних умов експлуатації від розрахункових можуть виникнути пожежонебезпечні ситуації.

Для гасіння пожеж в офісному приміщенні пропонується використовувати порошкові або вуглекислотні вогнегасники, так як вони є універсальними.

Простори усередині приміщень в межах, яких можуть утворюватися або знаходитися пожежонебезпечні речовини і матеріали відповідно до ДСТУ Б В.1.1-36:2016 [10] відносяться до пожежонебезпечної зони класу П-Па. Це обумовлено тим, що в приміщенні знаходяться тверді горючі та важкозаймісті речовини та матеріали. Приміщенню, у якому розташоване робоче місце, присвоюється II ступень вогнестійкості.

Потенційними джерелами запалювання можуть бути:

- Іскри і дуги короткого замикання;
- електрична іскра при замиканні і розмиканні ланцюгів;
- перегрів від тривалого перевантаження,
- відкритий вогонь і продукти горіння,
- наявність речовин, нагрітих вище за температуру самозаймання,
- розрядна статична електрика.

### 5.3.3 Електробезпека

Згідно з класифікацією приміщень за ступенем небезпеки ураження електричним струмом [7], приміщення в якому проводяться всі роботи відносяться до першого класу (без підвищеної небезпеки). Під час роботи використовуються електроустановки з напругою живлення 36 В, 220 В, та 360 В. Опір контуру заземлення повинен мати не більше 4 Ом.

Розрахунок проводять за допомогою методу коефіцієнта використання (екранування) електродів. Коефіцієнт використання групового заземлювача  $\eta$  – це відношення діючої провідності цього заземлювача до найбільш можливої його провідності за нескінченно великих відстаней між його електродами. Коефіцієнт використання вертикальних заземлювачів  $\eta_v$  в залежності від розміщення заземлювачів та їх кількості знаходиться в межах 0,4...0,99. Взаємну екрануючу дію горизонтального заземлювача (з'єднувальної смуги) враховують за допомогою коефіцієнта використання горизонтального заземлювача  $\eta_c$ .

Послідовність розрахунку:

- 1) Визначається необхідний опір штучних заземлювачів  $R_{шт.з.}$ :

$$R_{шт.з.} = \frac{R_d \cdot R_{пр.з.}}{R_{пр.з.} - R_d}, \quad (4.1)$$

де  $R_{пр.з.}$  – опір природних заземлювачів;

$R_d$  – допустимий опір заземлення. Якщо природні заземлювачі відсутні, то  $R_{шт.з.} = R_d$ .

Підставивши числові значення у формулу (4.1), отримуємо:

$$R_{\text{шт.з.}} = \frac{4 \cdot 40}{40 - 4} \approx 4 \text{ Ом}.$$

2) Опір заземлення в значній мірі залежить від питомого опору ґрунту  $\rho$ , Ом·м. Приблизне значення питомого опору глини приймаємо  $\rho = 40$  Ом·м (табличне значення).

3) Розрахунковий питомий опір ґрунту,  $\rho_{\text{розр.}}$ , Ом·м, визначається відповідно для вертикальних заземлювачів  $\rho_{\text{розр.в}}$ , і горизонтальних  $\rho_{\text{розр.г}}$ , Ом·м за формулою:

$$\rho_{\text{розр.}} = \Psi \cdot \rho, \quad (4.2)$$

де  $\Psi$  – коефіцієнт сезонності для вертикальних заземлювачів і кліматичної зони з нормальною вологістю землі, приймається для вертикальних заземлювачів  $\rho_{\text{розр.в}} = 1,7$  і горизонтальних  $\rho_{\text{розр.г}} = 5,5$  Ом·м.

$$\rho_{\text{розр.в}} = 1,7 \cdot 40 = 68 \text{ Ом} \cdot \text{м};$$

$$\rho_{\text{розр.г}} = 5,5 \cdot 40 = 220 \text{ Ом} \cdot \text{м}.$$

4) Розраховується опір розтікання струму вертикального заземлювача  $R_{\text{в}}$ , Ом, за (4.3).

$$R_{\text{в}} = \frac{\rho_{\text{розр.в}}}{2 \cdot \pi \cdot l_{\text{в}}} \cdot \left( \ln \frac{2 \cdot l_{\text{в}}}{d_{\text{ст}}} + \frac{1}{2} \cdot \ln \frac{4 \cdot t + l_{\text{в}}}{4 \cdot t - l_{\text{в}}} \right), \quad (4.3)$$

де  $l_{\text{в}}$  – довжина вертикального заземлювача (для труб – 2 – 3 м;  $l_{\text{в}} = 3$  м);

$d_{\text{ст}}$  – діаметр стержня (для труб – 0,03 – 0,05 м;  $d_{\text{ст}} = 0,05$  м);

$t$  – відстань від поверхні землі до середини заземлювача, яка визначається за ф. (4.4):

$$t = h_{\text{в}} + \frac{l_{\text{в}}}{2}, \quad (4.4)$$

де  $h_{\text{в}}$  – глибина закладання вертикальних заземлювачів (0,8 м); тоді

$$t = 0,8 + \frac{3}{2} = 2,3 \text{ м};$$

$$R_{\text{в}} = \frac{68}{2 \cdot \pi \cdot 3} \cdot \left( \ln \frac{2 \cdot 3}{0,05} + \frac{1}{2} \cdot \ln \frac{4 \cdot 2,3 + 3}{4 \cdot 2,3 - 3} \right) = 18,5 \text{ Ом}.$$



5) Визначається теоретична кількість вертикальних заземлювачів  $n$  штук, без урахування коефіцієнта використання  $\eta_v$ :

$$n = \frac{2R_E}{R_D} = \frac{2 \times 18,5}{4} = 9,25. \quad (4.5)$$

І визначається коефіцієнт використання вертикальних електродів групового заземлювача без врахування впливу з'єднувальної стрічки  $\eta_v = 0,57$  (табличне значення).

б) Визначається необхідна кількість вертикальних заземлювачів з урахуванням коефіцієнта використання  $\eta_v$ , шт:

$$n = \frac{2 \cdot R_E}{R_D \cdot \eta_v} = \frac{2 \cdot 18,5}{4 \cdot 0,57} \approx 16 \quad (4.6)$$

7) Визначається довжина з'єднувальної стрічки горизонтального заземлювача  $l_c$ , м:

$$l_c = 1,05 \cdot L_B \cdot (n_B - 1), \quad (4.7)$$

де  $L_B$  – відстань між вертикальними заземлювачами, (прийняти за  $L_B = 3$  м);

$n_B$  – необхідна кількість вертикальних заземлювачів.

$$l_c = 1,05 \cdot 3 \cdot (16 - 1) \approx 48 \text{ м.}$$

Визначається опір розтіканню струму горизонтального заземлювача (з'єднувальної стрічки)  $R_r$ , Ом:

$$R_r = \frac{\rho_{\text{розр.г}}}{2 \cdot \pi \cdot l_c} \cdot \ln \frac{2 \cdot l_c^2}{d_{\text{см}} \cdot h_r}, \quad (4.8)$$

де  $d_{\text{см}}$  – еквівалентний діаметр смуги шириною  $b$ ,  $d_{\text{см}} = 0,95b$ ,  $b = 0,15$  м;

$h_r$  – глибина закладання горизонтальних заземлювачів (0,5 м);

$l_c$  – довжина з'єднувальної стрічки горизонтального заземлювача  $l_c$ , м

$$R_r = \frac{220}{2 \cdot \pi \cdot 48} \cdot \ln \frac{2 \cdot 48^2}{0,95 \cdot 0,15 \cdot 0,5} = 8,1 \text{ Ом.}$$

8) Визначається коефіцієнт використання горизонтального заземлювача  $\eta_c$ , відповідно

до необхідної кількості вертикальних заземлювачів  $n_b$ .

Коефіцієнт використання з'єднувальної смуги  $\eta_c = 0,3$ .

Розраховується результуючий опір заземлювального електроду з урахуванням з'єднувальної смуги:

$$R_{\text{заг.}} = \frac{R_E \cdot R_r}{R_E \cdot \eta_c + R_r \cdot \eta_E \cdot \eta_E} \leq R_d \quad (4.9)$$

Висновок: дане захисне заземлення буде забезпечувати електробезпеку будівлі, так як виконується умова:  $R_{\text{заг.}} < 4$  Ом, а саме:

$$R_{\text{заг.}} = \frac{18,5 \cdot 8,1}{18,5 \cdot 0,3 + 8,1 \cdot 16 \cdot 0,57} = 1,9 \leq R_d .$$

## 5.4 Гігієнічні вимоги до параметрів виробничого середовища

### 5.4.1 Освітлення

Згідно з [8] для виробничих та адміністративних приміщень світловий коефіцієнт приймається не менше  $1/8$ , в побутових –  $1/10$ :

$$S_b = \left( \frac{1}{5} \div \frac{1}{10} \right) \times S_n, \quad (4.10)$$

де  $S_b$  – площа віконних прорізів,  $\text{м}^2$ ;

$S_n$  – площа підлоги,  $\text{м}^2$ .

$$S_n = a \cdot b = 6 \cdot 4 = 24 \text{ м}^2,$$

$$S = 1/10 \cdot 24 = 2,4 \text{ м}^2.$$

Приймаємо 1 вікно площею  $S=2,4 \text{ м}^2$ .

Світильники загального освітлення розташовуються над робочими поверхнями в рівномірно-прямокутному порядку. Для організації освітлення в темний час доби передбачається обладнати приміщення, довжина якого складає 6 м, ширина 4 м, світильниками ЛПО2П, оснащеними лампами типу ЛБ (дві по 80 Вт) з світловим потоком 5200 лм кожна.

Розрахунок штучного освітлення виробляється по коефіцієнтах використання світлового потоку, яким визначається потік, необхідний для створення заданої освітленості при загальному рівномірному освітленні. Розрахунок кількості світильників  $n$  виробляється по формулі (4.11):

$$n = \frac{E \times S \times Z \times K}{F \times U \times M}, \quad (4.11)$$

де  $E$  – нормована освітленість робочої поверхні, визначається нормами – 300 лк;

$S$  – освітлювана площа,  $m^2$ ;  $S = 24 m^2$ ;

$Z$  – поправочний коефіцієнт світильника (1,1 для люмінесцентних ламп);

$K$  – коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації – 1,5;

$U$  – коефіцієнт використання, залежний від типу світильника, показника індексу приміщення і т.п. – 0,575

$M$  – число люмінесцентних ламп в світильнику – 2;

$F$  – світловий потік лампи – 5200лм.

Підставивши числові значення у формулу (4.11), отримуємо:

$$n = \frac{300 \times 24 \times 1.1 \times 1.5}{5200 \times 0.575 \times 2} \approx 2.$$

Приймаємо освітлювальну установку, яка складається з 3-х світильників, які складаються з 2-х люмінесцентних ламп загальною потужністю 80 Вт, напругою – 220 В.

#### 5.4.2 Вентилювання

Здійснюється провітрювання приміщення, в залежності від погодних умов, тривалість повинна бути не менше 10 хв. Найкращий обмін повітря здійснюється при наскрізному провітрюванні.

#### 5.5 Екологія

Діяльність за темою магістерської роботи, а саме: дослідження ефективності віртуальної багатопоточності в ядрах сучасних процесорів, в процесі її виконання впливає на навколишнє природне середовище і регламентується нормами діючого законодавства: Законом України «Про охорону навколишнього природного середовища» [11], Законом України «Про

забезпечення санітарного та епідемічного благополуччя населення» [12], Законом України «Про відходи» [13].

В процесі діяльності з виконанням дипломного проектуванням виникають процеси поводження з відходами ІТ галузі. Нижче надано перелік відходів, що утворюються в процесі роботи:

- Відпрацьовані люмінесцентні лампи - I клас безпеки.
- Змінні носії інформації - IV клас безпеки.
- Відпрацьовані вогнегасники - IV клас безпеки.
- Макулатура - IV клас безпеки.

## 5.6 Висновки до п'ятого розділу

В розділі чотири зроблено аналіз умов праці, шкідливих та небезпечних чинників, з якими стикається робітник. Було визначено параметри і певні характеристики приміщення для роботи над запропонованим проектом написаному в кваліфікаційній роботі, описано, які заходи потрібно зробити для того, щоб дане приміщення відповідало необхідним нормам і було комфортним і безпечним для робітника.

Приведені рекомендації щодо організації робочого місця, а також важливу інформацію щодо пожежної та електробезпеки. Також були наведені інструкції з охорони праці, техніки безпеки при роботі на комп'ютері.

На підставі аналізу розроблені заходи з техніки безпеки, організації вимог освітлення робочого місця та рекомендації з пожежної профілактики.

## 5.7 Перелік джерел посилань до розділу 5

1. Закон України "Про охорону праці". Вводиться в дію Постановою ВР № 2695-ХІІ від 14.10.92, ВВР, 1992, № 49, ст.669. - Режим доступу: [www. URL: https://zakon.rada.gov.ua/laws/show/2694-12](http://www.URL:https://zakon.rada.gov.ua/laws/show/2694-12)
2. Кодекс законів про працю України. Затверджується Законом № 322-VIII від 10.12.71 ВВР, 1971. Режим доступу: [www. URL: https://zakon.rada.gov.ua/laws/show/322-08](http://www.URL:https://zakon.rada.gov.ua/laws/show/322-08)
3. Закон України "Про загальнообов'язкове державне соціальне страхування від нещасного випадку на виробництві та професійного захворювання, які спричинили втрату працездатності". Наказ від 21 грудня 2000 року N 2180-III. Режим доступу: [www. URL: https://dnaop.com/html/2065/doc-zakon-ukrajini-pro-zagalynoobovjzskove-derzhavne-socialyne-strahuvannya-vid-neshhasnogo-vipadku-na-virobnictvi-ta-profesijnogo-z](http://www.URL:https://dnaop.com/html/2065/doc-zakon-ukrajini-pro-zagalynoobovjzskove-derzhavne-socialyne-strahuvannya-vid-neshhasnogo-vipadku-na-virobnictvi-ta-profesijnogo-z)

4. Про затвердження Типового положення про порядок проведення навчання і перевірки знань з питань охорони праці НПАОП 0.00-4.12-05. Наказ від 26.01.2005 №15. Режим доступу: [www. URL: https://zakon.rada.gov.ua/laws/show/z0231-05](https://zakon.rada.gov.ua/laws/show/z0231-05)

5. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПІН 3.3.2.007-98. Затверджено Постановою Головного державного санітарного лікаря України 10 грудня 1998 р. N 7. Режим доступу: [www. URL: https://zakon.rada.gov.ua/rada/show/v0007282-98](https://zakon.rada.gov.ua/rada/show/v0007282-98)

6. Санітарні норми мікроклімату виробничих приміщень ДСН 3.3.6.042-99. Постанова N 42 від 01.12.99. Режим доступу: [www. URL: https://zakon.rada.gov.ua/rada/show/va042282-99](https://zakon.rada.gov.ua/rada/show/va042282-99)

7. Електробезпека в будівлях і спорудах. Вимоги до захисних заходів від ураження електричним струмом. Наказ від 1 липня 2016 року N 204. Режим доступу: [www. URL: http://epicentre.co.ua/dstu/doc28522.html](http://epicentre.co.ua/dstu/doc28522.html)

8. ДБН В.2.5-28:2018 «Природне і штучне освітлення». Режим доступу: [www. URL: http://www.minregion.gov.ua/wp-content/uploads/2018/12/V2528-1.pdf](http://www.minregion.gov.ua/wp-content/uploads/2018/12/V2528-1.pdf)

9. НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». Зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за № 508/31960. Режим доступу: [www. URL: https://zakon.rada.gov.ua/laws/show/z0508-18](https://zakon.rada.gov.ua/laws/show/z0508-18)

10. ДСТУ Б В.1.1-36:2016 «Визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою». Наказ від 15.06.2016 №158. Режим доступу: [www. URL: https://zakon.rada.gov.ua/rada/show/v0158858-16](https://zakon.rada.gov.ua/rada/show/v0158858-16)

11. Закон України «Про охорону навколишнього природного середовища» . Вводиться в дію Постановою ВР № 1268-ХІІ від 26.06.91, ВВР, 1991, № 41, ст.547. Режим доступу: [www. URL: https://zakon.rada.gov.ua/laws/show/1264-12](https://zakon.rada.gov.ua/laws/show/1264-12)

12. Закони України «Про охорону навколишнього природного середовища». Вводиться в дію Постановою ВР № 4005-ХІІ від 24.02.94, ВВР, 1994, № 27, ст.219. Режим доступу: [www. URL: https://zakon.rada.gov.ua/laws/show/4004-12](https://zakon.rada.gov.ua/laws/show/4004-12)

13. Закон України «Про відходи». Відомості Верховної Ради України (ВВР), 1998, № 36-37, ст.242. Режим доступу: [www. URL: https://zakon.rada.gov.ua/laws/show/187/98-вр](https://zakon.rada.gov.ua/laws/show/187/98-вр)

## ВИСНОВКИ

У розділі 1 проведено огляд літератури про методи збільшення продуктивності ядер сучасних процесорів, зокрема:

Розглянуто оцінки продуктивності комп'ютера (ядра процесора)

- збільшення частоти
- Інші способи збільшення продуктивності одного ядра процесора
- Суть і передумови віртуальної багатоядерності
- Варіанти структур багатопотокової віртуальності.

Розглянуто ефективності функціонування конвеєрних ядер процесора з універсальним функціональним пристроєм і зі спеціалізованими функціональними пристроями без урахування впливу «перешкод».

Проведено огляд літератури з дослідження реальної ефективності ядер процесорів з урахуванням «перешкод»

Розроблено методику проведення досліджень комплексного впливу «перешкод» (інформаційних залежностей, операцій редукції, команд переходів) на ефективність ядер процесорів на прикладі широко поширених і наочних програм, таких як «Множення матриць», «Рішення диференціальних рівнянь в приватних похідних методом сіток», « швидке перетворення Фур'є ».

У досліджуваних програмах:

- виділяється ядро - ділянка програми, що забезпечує основний внесок під час виконання програми;
- розробляється на умовному асемблері програма ядра;
- з'ясовуються інформаційно залежні команди і команди редукції, якщо вони є;
- формуються групи команд, що виконують інформаційно залежні ділянки команд ядра;
- розробляється модель ядра процесора, що виконує ядро програми;
- в моделі інформаційно залежні групи команд виконуються універсальним функціональним пристроєм послідовно згідно з алгоритмом виконання програми.

Далі визначаються:

- кількість типів груп команд;
- ймовірності появи кожної групи в програмі;
- часи виконання кожної групи;
- середній час виконання ядра програми;
- коефіцієнт використання універсального функціонального пристрою;

– коефіцієнт використання пристрою управління моделі в залежності від значення різних параметрів програми і ядра процесора.

У розділах 2, 3 і 4 досліджені ефективності функціонування розробленої моделі ядра процесора на прикладах програм «Множення матриць», «Рішення диференціальних рівнянь в приватних похідних методом сіток», «Швидке перетворення Фур'є».

Результати досліджень наведені в таблицях 5.1 - .5.3.

Таблиця 5.1 - результати дослідження програми «Множення матриць»

Розмір матриць	Розрядність даних	Кіл. потоків	ККД <sub>уу</sub> %	ККД <sub>оп</sub> %	ККД <sub>обч</sub> %	ККД <sub>L2</sub> %	ККД <sub>L3</sub> %	Коеф. прискорення
64	32	1	22.3	100	0.25	-	-	-
	64	1	20.7	100	0.25	-	-	-
	32	2	40	100	0.5	-	-	1
	64	2	33	100	0.5	-	-	1
128	32	1	21	47	0.25	16	-	-
	64	1	18.2	45	0.25	27	-	-
	32	2	40	42	0.5	15	-	1.65
	64	2	27	37	0.5	27	-	1.46
512	32	1	16.5	9.9	0.25	-	36	-
	64	1	12.3	7.7	0.25	-	55	-
	32	2	24	13.4	0.5	-	53.7	1.45
	64	2	12.3	9.8	0.5	-	70.5	1.27
n>2 <sup>14</sup>	32	1	0.4	100	0.25	-	-	-
	64	1	0.4	100	0.25	-	-	-
	32	2	0.4	100	0.25	-	-	1
	64	2	0.4	100	0.25	-	-	1

Таблиця 5.2 - Результати дослідження програми «Рішення диференційних рівнянь у приватних похідних»

Розмір	Теоретичне прискорення	Реальне прискорення
Плаваюча точка – 32 розряди		
64	1	1
128	1,65	1,69
512	1,45	1,37
n>2 <sup>14</sup>	1	1,04
Плаваюча точка – 64 розряди		
64	1	1
128	1,46	1,6
256	1,27	1,36
n>2 <sup>14</sup>	1	1,01

Таблиця 5.3 - Результати дослідження програми «ШПФ»

Розмір, комп. точок	Теоретичне прискорення	Реальне прискорення
512	1,34	1,33

### Зроблено наступні висновки

Коефіцієнт використання УУ (і всього ядра процесора), визначений з моделі ядра процесора, є верхньою межею ефективності ядра процесора.

Максимальне значення коефіцієнта використання УУ (і всього ядра процесора) змінюється в досить широкому діапазоні - від 1% до 22.3% в залежності від типу програми, розмірності масивів і форматів даних.

У кращому випадку, коли УУ в одному такті генерує  $S = 4$  команди коефіцієнт використання УУ (і всього ядра процесора) не перевищуватиме 25%.

Пропускна здатність обчислювальної компоненти підсистеми виконання команд, що складається з кеш-пам'яті даних першого рівня L1D, пристроїв Множення і Додавання не залежить від розмірності масивів і форматів даних та дорівнює: 0.25 - при виконанні одного потоку; 0.5 - при виконанні 2-х потоків. Це означає, що обчислювальна компонента підсистеми виконання команд, що складається з кеш-пам'яті даних першого рівня L1D, пристроїв Множення і Додавання може бути використана для виконання інших потоків.

Коефіцієнти прискорення при виконанні 2-х поточних варіантів більше 1, що доводить доцільність введення режиму двопоточності.

Якщо при виконанні однопоточного варіанту програми коефіцієнт використання критично важливого ресурсу комп'ютера (наприклад оперативної пам'яті) близький 100%, то використання двупоточного режиму не дасть ефекту.

Експериментальні дані досить добре корелюють з теоретичними даними. Точні значення модельних (теоретичні) і експериментальна даних не збігаються, що і слід було очікувати. Однак тенденції результатів однозначно однакові.

Програма обчислення ШПФ (співвідношення команд звернення до підсистеми пам'яті та команд обробки даних 1: 1, тобто 10 команд звернення до підсистеми пам'яті та 10 команд обробки даних) є прикладом того, що підсистема пам'яті не завжди є гальмом обчислювального процесу, якщо переважна частина звернень в підсистему пам'яті виконується в швидку кеш-пам'ять даних першого рівня L1D з максимальним поєднанням.



## ЛІТЕРАТУРА

1. Недзельський Д.А. Дослідження і аналіз ефективності структурних методів компенсації впливу команд переходу на продуктивність конвексних ядер процесорів. Луганськ: Вісник Східноукраїнського національного університету ім. В. Даля, 2013. - №16 (205), Частина 2. - С.174-181.
2. Недзельський Д.А. Дослідження ефективності одноядерних суперскалярні обчислювальних систем. Луганськ. Вісник Східноукраїнського національного університету ім. В. Даля. - 2011. - №15 (169) Ч. 2. - с. 133-140.
3. Недзельський Д.О., Сафонова С.О. Про реальну продуктивність та ефективність ядер СУЧАСНИХ процесорів. Наукові вісті Далівського університету. Електронне видання. 2019. № 17. <http://dspace.snu.edu.ua:8080/jspui/handle/123456789/3206>.
4. Недзельський Д.А. Дослідження ефективності підсистеми генерації команд в ядрах сучасних процесорів. Луганськ: Вісник Східноукраїнського національного університету ім. В. Даля, 2017. - №8 (238), - С.64-66.
5. Є.О. Шквар. Ефективність використання технології віртуальної багатопоточності при паралельних розв'язанні Рівняння Пуассона. Вісник НАУ, Київ. 2012 N3 (101), том 2, С.138-141.
6. Буділовський С. Simultaneous Multithreading (SMT) в топовом AMD Ryzen 7 2700X: тестування в синтетичі і іграх. Постійний URL: [https://ru.gecid.com/cpu/simultaneous\\_multithreading\\_smt\\_v\\_amd\\_ryzen\\_7\\_2700x](https://ru.gecid.com/cpu/simultaneous_multithreading_smt_v_amd_ryzen_7_2700x)
7. Михайло Кузьмінський. Nehalem: структура та продуктивність. Відкриті системи, №8 2009.
8. Андрій Созінов. Intel Sunny Cove: мікроархітектура процесорів наступного покоління. 12.12.2018. Постійний URL: <https://3dnews.ru/979520>.
9. Ілля Гавриченко. Мікроархітектура Zen 2. 19.06.2019. Постійний URL: <https://3dnews.ru/989344/mikroarhitektura-zen-2/page-2.html>.
10. Nedzelskyi D . M. Derkach, Ye. Tatarchenko, S. Safonova, L. Shumova, V. Kardashuk. Research of Efficiency of Multi-Core Computers with Shared Memory. 2019 IEEE 7th International Conference on Future Internet of Things and Cloud (FiCloud).
11. Nedzelskyi D. About number of effectively performed streams in multi-core computers with shared memory. Вісник Східноукраїнського національного університету імені Володимира Даля. Науковий журнал. Сєверодонецьк: СНУ ім.В.Даля. 2019. № 5 (253). С. 74-77.
12. Nedzelskyi D. Efficient Streaming on Multi-Core of Shared-Memory Computing. In: Theoretical and Applied Computer Science and Information Technology: Proceedings of the III

International Conference TACSIT-2019, May 8-9 2019, Severodonetsk: Volodymyr Dahl East Ukrainian National University, 2019, pp. 17-18.

13. Tullsen D. Multithreading Architecture / D. Tullsen, M. Nemirovsky., 2012. – 109 с.
14. Хеннессі Джон Л., Паттерсон Девід А. Комп'ютерна архітектура. Кількісний підхід. Видання 5-е М.: ТЕХНОСФЕРА, 2016. - 936 с., ISBN 978-5-94836-413-1.
15. Основи паралельного програмування з використанням технологій MPI та OpenMP: навчальний посібник / Р.В. Жалнин, Е.Н. Панюшкіна, Е. Е. Пескова, П.А. Шаманалісь. - Саранськ: Вид-во СВМО, 2013. - 78 с.
16. Кудін А.В. Архітектура і операційні системи паралельних обчислювальних систем / Кудін А.В., Ліньова О.В. – Нижній Новгород, 2007. – 73 с.
17. William Stallings, Computer Organization and Architecture: Designing and Performance, 5th Ed., Prentice Hall, 2000
18. Воєводін В.В., Воєводін Вл.В., Паралельні обчислення, СПб: БХВ-Петербург, 2004
19. Tullsen D.M., Eggers S.J., Levy H.M., Simultaneous multithreading: Maximizing on-chip parallelism, Proc. 22nd International Symposium on Computer Architecture, 1995
20. Tullsen D.M., Exploiting choice: Instruction fetch and issue on an implementable simultaneous multithreading processor, Proc. 23rd International Symposium on Computer Architecture, 1996
21. Семеренко, В. П. Технології паралельних обчислень : навчальний посібник / Семеренко В. П. – Вінниця : ВНТУ, 2018. – 104 с

## Додаток А

```
#include <stdio.h>
#include <stdlib.h>
#include <omp.h>
#include <fstream>
#include <iostream>
#include <conio.h>
#include <ctime>

using namespace std;

void transpose(double* A, double* B, int n) {
    int i, j;
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            B[j * n + i] = A[i * n + j];
        }
    }
}

int inputM(string nameFile, double* matrix, int n) {
    int r = (n * n) + 1, size = 0;
    ifstream fin(nameFile, ios::in);
    if (fin.is_open())
    {
        int count = 0;
        char temp;
        cout << "open file" << '\n';
        while (!fin.eof())// пробегаем пока не встретим конец файла eof
        {
```

```

        fin >> temp;//в пустоту считываем из файла числа
        count++;
    }
    size = count;
    fin.close();
}
else cout << "file not open";
fin.open(nameFile, ios::in);
int a = 0,i =0;
char temp, temp_pre='0';
fin.seekg(0,ios::beg);
while (!fin.eof())// конверт в дабл
{
    fin >> temp;
    char* red = &temp;
    if(temp_pre=='\n')
    {
        matrix[i] = atof(red);
        i++;
    }
    temp_pre = temp;
}
/*cout << "buff full" << '\n' << "i=" << i<<'\n';
for (int i = 0; i < n * n; i++)
{

    if (i % 10 == 0)
        cout << matrix[i]<<' ';

}*/

```

```

    return 0;
}

void gemm(double* A, double* B, double* C, int n)
{
    int i, j, k;
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            double dot = 0;
            for (k = 0; k < n; k++) {
                dot += A[i * n + k] * B[k * n + j];
            }
            C[i * n + j] = dot;
        }
    }
}

void gemm_omp(double* A, double* B, double* C, int n)
{
    omp_set_num_threads(2);
#pragma omp parallel
    {
        int i, j, k;
#pragma omp for private (i,j,k)
        for (i = 0; i < n; i++) {
            for (j = 0; j < n; j++) {
                double dot = 0;
#pragma omp parallel reduction(+:dot)
                for (k = 0; k < n; k++) {
                    dot += A[i * n + k] * B[k * n + j];
                }
            }
        }
    }
}

```

```

        }
        C[i * n + j] = dot;
    }
}

void gemmT(double* A, double* B, double* C, int n)
{
    int i, j, k;
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            double dot = 0;
            for (k = 0; k < n; k++) {
                dot += A[i * n + k] * B[j * n + k];
            }
            C[i * n + j] = dot;
        }
    }
}

/*void gemmT_omp(double* A, double* B, double* C, int n)
{
    double* B2;
    B2 = (double*)malloc(sizeof(double) * n * n);
    transpose(B, B2, n);
    omp_set_num_threads(2);
#pragma omp parallel shared(A,B2,n,C)
    {
        int i, j, k;

```

```

#pragma omp for private(i,j,k)
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            double dot = 0;
#pragma omp parallel reduction(+:dot)
                for (k = 0; k < n; k++) {
                    dot += A[i * n + k] * B2[j * n + k];
                }
            C[i * n + j] = dot;
        }
    }

}

free(B2);
}*/

int main() {
    int m, X = 2;
    double start, finish, duration;
    const int numberName = 8;
    string fileName[numberName] = { "A1.txt", "A2.txt", "A3.txt", "A4.txt",
                                    "B1.txt", "B2.txt", "B3.txt", "B4.txt" };
    int const n[4] = { 64, 128, 512, 2048 };
    double* A, * B, * C, * B2;
    A = (double*)malloc(sizeof(double) * n[X] * n[X]);
    B = (double*)malloc(sizeof(double) * n[X] * n[X]);
    C = (double*)malloc(sizeof(double) * n[X] * n[X]);
    B2 = (double*)malloc(sizeof(double) * n[X] * n[X]);
    inputM(fileName[X], A, n[X]);
    inputM(fileName[X + 4], B, n[X]);
}

```

```
transpose(B, B2, 256);  
transpose(A, C, 256);  
m = 256;  
start = clock();  
        gemm(C, B2, A, m);  
        gemm(C, B2, A, m);  
finish = clock();  
cout << "N = " << m << '\n';  
duration = double(finish - start)/CLOCKS_PER_SEC;  
cout << duration << "s";  
_getch();  
return 0;}
```



## Додаток Б

```

#include <iostream>
#include <windows.h>
#include <stdio.h>
#include <omp.h>
#include <conio.h>
#include <time.h>
#include <math.h>

using namespace std;
// Function for the Gauss-Seidel algorithm
void SerialResultCalculation(float* pMatrix, int Size, float& Eps, int& Iterations) {
    float dm, dmax, temp; int i, j; // Loop variables
    Iterations = 0;
do {
    dmax = 0;
    for (i = 1; i < Size - 1; i++)
        for (j = 1; j < Size - 1; j++) {
            temp = pMatrix[Size * i + j];
            pMatrix[Size * i + j] = 0.25 * (pMatrix[Size * i + j + 1] +
                pMatrix[Size * i + j - 1] +
                pMatrix[Size * (i + 1) + j] +
                pMatrix[Size * (i - 1) + j]);
            dm = fabs(pMatrix[Size * i + j] - temp);
            if (dmax < dm) dmax = dm;
        } Iterations++;
    } while (dmax > Eps);
}

void ParallelResultCalculation(float* pMatrix, int Size, float& Eps, int& Iterations) {

```

```

float d, dmax, temp, dm; int i, j; // Loop variables
Iterations = 0;
omp_lock_t dmax_lock;
omp_init_lock(&dmax_lock);
do {
    dmax = 0; // максимальное изменение значений
#pragma omp parallel for shared(pMatrix, Size, dmax) private(i, temp, dm, d)
    for (i = 1; i < Size - 1; i++)
    {
        dm = 0;
        // #pragma omp parallel for shared(pMatrix, Size, dmax) private(j, temp, d)
        for (j = 1; j < Size - 1; j++) {
            temp = pMatrix[Size * i + j];
            pMatrix[Size * i + j] = 0.25 * (pMatrix[Size * i + j + 1] +
                pMatrix[Size * i + j - 1] +
                pMatrix[Size * (i + 1) + j] +
                pMatrix[Size * (i - 1) + j]);
            d = fabs(pMatrix[Size * i + j] - temp);
            if (dm < d) dm = d;
            omp_set_lock(&dmax_lock);
            if (dmax < d) dmax = dm;
            omp_unset_lock(&dmax_lock);
        }
    }
    Iterations++;
} while (dmax > Eps);
}

// Function for computational process termination
void ProcessTermination(float* pMatrix) { delete[] pMatrix; }

// Function for formatted matrix output

```

```

void PrintMatrix(float* pMatrix, int RowCount, int ColCount) {
    int i, j; // Loop variables
    for (i = 0; i < RowCount; i++) {
        for (j = 0; j < ColCount; j++)
            printf("%7.4f ", pMatrix[i * ColCount + j]);
        printf("\n");
    }
}

// Function for simple setting the grid node values
void DummyDataInitialization(float* pMatrix, int Size) {
    int i, j; // Loop variables
    float h = 1.0 / (Size - 1);
    // Setting the grid node values
    for (i = 0; i < Size; i++) {
        for (j = 0; j < Size; j++)
            if ((i == 0) || (i == Size - 1) || (j == 0) || (j == Size - 1))
                pMatrix[i * Size + j] = 100;
            else
                pMatrix[i * Size + j] = 0;
    }
}

} // Function for memory allocation and initialization of grid nodes
void ProcessInitialization(float*& pMatrix, int& Size, float& Eps) {
    //// Setting the grid size
    //do {
    //    printf("\nВведите размер сетки: ");
    //    scanf_s("%d", &Size);
    //    if (Size <= 2)
    //        printf("\nРазмер сетки должен быть больше 2!\n");
    //} while (Size <= 2);
    //// Setting the required accuracy

```

```

//do {
//    printf("\nВведите необходимую точность: ");
//    scanf_s("%lf", &Eps);
//    if (Eps <= 2)
//        printf("\nТочность должна быть больше 0!\n");
//} while (Eps <= 0);
// Memory allocation
pMatrix = new float[Size * Size];
// Setting the grid node values
DummyDataInitialization(pMatrix, Size);
}
void main() {
    setlocale(LC_ALL, "Ru");
    float* pMatrix, *pMatrix2;
    // Matrix of the grid nodes
    int Size = 512; // Matrix size
    float Eps = 0.3; // Required accuracy
    int Iterations; // Iteration number
    clock_t start, finish, duration;
    printf("Алгоритм Гаусса - Зейделя!\n");
    // Process initialization
    ProcessInitialization(pMatrix, Size, Eps);
    ProcessInitialization(pMatrix2, Size, Eps);
    // Matrix output
    //printf("Инициализированная матрица: \n");
    //PrintMatrix(pMatrix, Size, Size);
    // The Gauss-Seidel method
    start = clock();
    //SerialResultCalculation(pMatrix2, Size, Eps, Iterations);
    SerialResultCalculation(pMatrix, Size, Eps, Iterations);
}

```

```
//ParallelResultCalculation(pMatrix, Size, Eps, Iterations);  
finish = clock();  
duration = (finish - start);  
// The Gauss-Seidel method  
//printf(" Результирующая матрица: \n");  
// PrintMatrix(pMatrix, Size, Size);  
cout << " ВРЕМЯ ВЫПОЛНЕНИЯ = " << duration << endl;  
printf(" Количество итераций - %d\n", Iterations);  
// Computational process termination  
ProcessTermination(pMatrix);  
ProcessTermination(pMatrix2);  
_getch();  
}
```

## Додаток В

```

#include <iostream>

#include <ctime>

// AVal - массив анализируемых данных, Nvl - длина массива должна быть кратна степени 2.
// FTvl - массив полученных значений, Nft - длина массива должна быть равна Nvl.

const double TwoPi = 6.283185307179586;

void FFTAnalysis(double* AVal, double* FTvl, int Nvl, int Nft) {
    int i, j, n, m, Mmax, lstp;
    double Tmpr, Tmpi, Wtmp, Theta;
    double Wpr, Wpi, Wr, Wi;
    double* Tmvl;
    n = Nvl * 2; Tmvl = new double[n];
    for (i = 0; i < n; i += 2) {
        Tmvl[i] = 0;
        Tmvl[i + 1] = AVal[i / 2];
    }
    i = 1; j = 1;
    while (i < n) {
        if (j > i) {
            Tmpr = Tmvl[i]; Tmvl[i] = Tmvl[j]; Tmvl[j] = Tmpr;
            Tmpr = Tmvl[i + 1]; Tmvl[i + 1] = Tmvl[j + 1]; Tmvl[j + 1] = Tmpr;
        }
        i = i + 2; m = Nvl;
        while ((m >= 2) && (j > m)) {
            j = j - m; m = m >> 1;
        }
        j = j + m;
    }
    Mmax = 2;
    while (n > Mmax) {

```

```

Theta = -TwoPi / Mmax; Wpi = sin(Theta);
Wtmp = sin(Theta / 2); Wpr = Wtmp * Wtmp * 2;
Istp = Mmax * 2; Wr = 1; Wi = 0; m = 1;
while (m < Mmax) {
    i = m; m = m + 2; Tmpr = Wr; Tmpi = Wi;
    Wr = Wr - Tmpr * Wpr - Tmpi * Wpi;
    Wi = Wi + Tmpr * Wpi - Tmpi * Wpr;
    while (i < n) {
        j = i + Mmax;
        Tmpr = Wr * Tmvl[j] - Wi * Tmvl[j - 1];
        Tmpi = Wi * Tmvl[j] + Wr * Tmvl[j - 1];
        Tmvl[j] = Tmvl[i] - Tmpr; Tmvl[j - 1] = Tmvl[i - 1] - Tmpi;
        Tmvl[i] = Tmvl[i] + Tmpr; Tmvl[i - 1] = Tmvl[i - 1] + Tmpi;
        i = i + Istp;
    }
}
Mmax = Istp;
}
for (i = 0; i < Nft; i++) {
    j = i * 2; FTvl[i] = 2 * sqrt(pow(Tmvl[j], 2) + pow(Tmvl[j + 1], 2)) / Nvl;
}
delete[]Tmvl;
}
void MatrixGeneration(double* matrix, const int n)
{
    for (int i = 0; i < n*n ; i++)
        {matrix[i] = double(rand() % 9 - 1);
}
int main()
{
    double* AVal, * FTvl;
    int n = 128, Nvl = n*n, Nft = n*n;

```

```
double start, finish;

AVal = (double*)malloc(sizeof(double) * Nvl);
FTvl = (double*)malloc(sizeof(double) * Nft);
MatrixGeneration(AVal,n);
start = clock();
FFTAnalysis(AVal, FTvl, Nvl, Nft);
finish = clock();
//std::cout << start << ' ' << finish;
double runtime = (double)(finish - start) / CLOCKS_PER_SEC;
std::cout << runtime << "s";
system("pause");
}
```



## Додаток Г.

---

*Міністерство освіти і науки України  
Східноукраїнський національний університет ім. В. Даля  
Кафедра комп'ютерних наук та інженерії*

### *Магістерська робота*

*«Моделі та методи для дослідження ефективності  
віртуальної багатопоточності в ядрах сучасних процесорів.»*

---

*ст.гр. КІ-19дм Лісовін А.В.*

*Керівник: Недзельський Д.О.*

---

## Актуальність роботи

---

**Об'єктом дослідження** магістерської роботи є технологія віртуальної багатопоточності в ядрі процесора.

**Метою дослідження** є визначення умов доцільності та ефективності реалізації віртуальної багатопоточності в ядрах сучасних процесорів .

**Методи рішення поставлених задач** базуються на використанні ідей систем масового обслуговування, сучасних технологій для створення аналітичних та імітаційних моделей.

**Наукова новизна** полягає в тому, що в роботі розроблений метод оцінки ефективності віртуальної багатопоточності з урахуванням параметрів програм, кількості потоків та параметрів структури ядра.

---

## Методика дослідження

---

- виділяється ядро - ділянка програми, яка забезпечує основний внесок під час виконання програми;
- розробляється на умовному асемблері програма ядра;
- з'ясовуються інформаційно залежні команди і команди редукції, якщо вони є;
- формуються групи команд, які виконують інформаційно залежні ділянки команд ядра програми;
- розробляється модель ядра процесора, яке виконує ядро програми;
- в моделі інформаційно залежні групи команд виконуються універсальним функціональним пристроєм послідовно згідно з алгоритмом виконання програми;

---

## Модель ядра процесора

---

У дослідженні використовується спрощена модель ядра приведена на рисунку

Підсистема підготовки команд(еквівалент front end частини ядра процесора)

Підсистема виконання груп команд (еквівалент back end)

Між ними буфер на n заявок

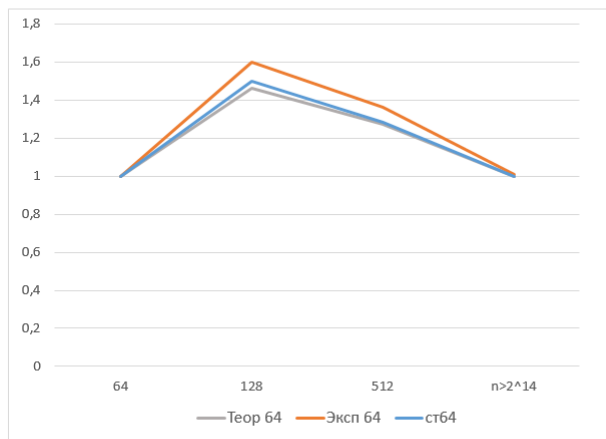


## Умови виконання ядра програми

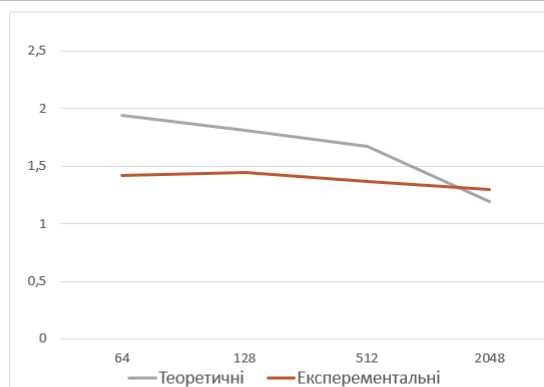
1. Масив значень нульового наближення і масив знаходяться в оперативній пам'яті.
2. Кеш-пам'ять даних першого рівня L1D це конвеєрний пристрій з пропускною спроможністю 1 такт і може кожен такт здійснювати 2 читання і 1 запис. Повний час звернення до кеш-пам'яті L1D - 4такти.
3. Функціональні пристрої Множення і Складання конвеєрні з пропускною спроможністю 1 такт. Повний час виконання операцій 4 такти.
4. У ядрі програми є команда редуції (накопичення похибки чергової ітерації). Час виконання операції редуції - 4 такти.

## Результати програми «Множення матриць»

Розмір	Теоретичне прискорення	Реальне прискорення
Плаваюча крапка – 32 розряди		
64	1	1
128	1,65	1,69
512	1,45	1,37
$n > 2^{14}$	1	1,04
Плаваюча крапка – 64 розряди		
64	1	1
128	1,46	1,6
256	1,27	1,36
$n > 2^{14}$	1	1,01



## Результати програми «Рішення ДР»



Розмір	Теоретичне прискорення	Реальне прискорення
64	1,94	1,42
128	1,81	1,45
512	1,67	1,37
$n > 2^{14}$	1,19	1,30

## Результати програми «ШПФ»

Програма обчислення ШПФ є прикладом того, що підсистема пам'яті не завжди є гальмом обчислювального процесу, так як переважна кількість звернень в підсистему пам'яті виконується в швидку кеш-пам'ять даних першого рівня L1D з максимальним поєднанням.

Розмір	Теоретичне прискорення	Реальне прискорення
1024	1,34	1,33

Співвідношення команд звернення до підсистему пам'яті і команд обробки даних 1: 1, тобто 10 команд звернення до підсистему пам'яті і 10 команд обробки даних.

# ВИСНОВКИ

При дослідженні виконання програми «Множення матриць» отримані такі результати.

Коефіцієнт використання УУ (і всього ядра процесора), визначений з еквівалентної спрощеної моделі ядра процесора, є верхньою межею ефективності ядра процесора.

Максимальне значення коефіцієнта використання УУ (і всього ядра процесора) змінюється в досить широкому діапазоні - від 1% до 22.3% в залежності від типу програми, розмірності масивів і форматів даних.

У кращому випадку коефіцієнт використання УУ не перевищує 25%.

Пропускна здатність обчислювальної компоненти підсистеми виконання команд не залежить від розмірності масивів і форматів даних та дорівнює:

- 0.25 - при виконанні одного потоку;
- 0.5 - при виконанні 2-х потоків.

Коефіцієнти прискорення при виконанні 2-х поточних варіантів більше 1, що доводить доцільність введення режиму двопоточності.

Якщо при виконанні однопоточного варіанту програми коефіцієнт використання критично важливого ресурсу комп'ютера близький 100%, то використання двопоточного режиму не дає ефекту.

Експериментальне дані досить добре корелюють з теоретичними даними.

Аналогічні результати отримані і при дослідженні програмі «Рішення диференційних рівнянь з приватними похідними»

Програма обчислення ШПФ (співвідношення команд звернення до підсистему пам'яті і команд обробки даних 1: 1), є прикладом того, що підсистема пам'яті не завжди є гальмом обчислювального процесу, якщо переважна число звернень в підсистему пам'яті виконується в швидку кеш-пам'ять даних першого рівня L1D з максимальним поєднанням.