

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

До захисту допускається

В.о. завідувача кафедри

_____ Рязанцев О.І.

«_____» _____ 2021 р.

МАГІСТЕРСЬКА РОБОТА

НА ТЕМУ:

Моделі і методи дослідження ефективності структурних методів підвищення продуктивності підсистеми генерації команд у ядрах сучасних процесорів

Освітньо-кваліфікаційний рівень “Магістр”

Спеціальність 122 – “Комп’ютерні науки”

Науковий керівник роботи:

Д.О. Недзельський

(підпис)

(ініціали, прізвище)

Консультант з охорони праці:

Я.О. Критська

(підпис)

(ініціали, прізвище)

Студент:

Р. Г. Костиря

(підпис)

(ініціали, прізвище)

Група:

КН-18дм

Севєродонецьк 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 "Комп'ютерні науки "

Спеціальність 122 – "Комп'ютерні науки"

ЗАТВЕРДЖУЮ:

Т.в.о. завідувача Сафонова С.О.
кафедри

« _____ » _____ 2019 р.

**ЗАВДАННЯ
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ**

Костирі Роману Геннадійовичу

1. Тема роботи «Моделі і методи дослідження ефективності структурних методів підвищення продуктивності підсистеми генерації команд у ядрах сучасних процесорів» керівник проекту (роботи) к.т.н., доц. Недзельський Д.О. затверджені наказом вищого навчального закладу від " 11 " 10 2019 року №135/15/15
2. Строк подання студентом проекту (роботи) 15.01.2021
3. Вихідні дані до проекту (роботи) - матеріали практики
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити).
 1. Огляд структур ядер сучасних процесорів.
 2. Дослідження структури ядра з універсальним функціональним пристроєм.
 3. Дослідження структури ядра з m спеціалізованими функціональними пристроями.
 4. Дослідження впливу перешкод від інформаційних залежностей та команд переходів.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслеників)

Презентація.

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Критська Яна Олександрівна, ст. викладач		

7. Дата видачі завдання 18.10.2019

Керівник Недзельський Д.О.

_____ (підпис)

Завдання прийняв до виконання
Костира Р.Г.

_____ (підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Найменування етапів дипломного проекту (роботи)	Термін виконання етапів проекту	Примітки
1	Аналіз стану питання в літературі. Вивчення структур ядер сучасних процесорів.	18.10.2020 – 16.11.2020	
2	Вивчення методів моделювання та вибір методу.	17.11.2020 – 22.11.2020	
3	Розробка моделі ядра з універсальним функціональним пристроєм та її дослідження.	23.11.2020 – 28.11.2020	
4	Розробка моделі ядра з m спеціалізованими функціональними пристроями та її дослідження.	29.11.2020 – 07.12.2020	
5	Дослідження впливу команд переходів на продуктивність підсистемі генерації команд.	08.12.2020 – 12.12.2020	
6	Порівняльний аналіз варіантів структури підсистемі генерації команд	13.12.2020 – 20.12.2020	
7	Розробка заходів з охорони праці.	21.12.2020 – 24.12.2020	
8	Оформлення пояснювальної записки і графічного матеріалу.	25.12.2020 – 12.01.2021	
9	Підготовка та подання магістерської роботи до захисту	13.01.2021 – 15.01.2021	

Студент

_____ (підпис)

Костира Р.Г.

_____ (прізвище та ініціали)

Науковий керівник

_____ (підпис)

Недзельський Д.О.

_____ (прізвище та ініціали)

АНОТАЦІЯ

Костиря Р.Г. Моделі і методи дослідження ефективності структурних методів підвищення продуктивності підсистеми генерації команд у ядрах сучасних процесорів

Розглянуті та проаналізовані структури ядер сучасних процесорів та методи забезпечення їх високої продуктивності. Розроблені моделі та досліджені впливи команд переходів на продуктивність підсистем виконання та підсистем генерації конвеєрних ядер процесорів. Проведений порівняльний аналіз ефективності структурних методів збільшення продуктивності підсистеми підготовки команд.

Практичне значення, галузь застосування роботи: Результати дослідження можуть бути корисні при оптимальному виборі структури ядра з урахуванням співвідношення як показників ефективності, так і апаратних витрат на реалізацію

Ключові слова: ядро, процесор, конвеєр, перешкоди, суперскалярність, ефективність, продуктивність.

АННОТАЦИЯ

Костыря Р.Г. Модели и методы исследования эффективности структурных методов повышения производительности подсистемы генерации команд в ядрах современных процессоров

Рассмотрены и проанализированы структуры ядер современных процессоров и методы обеспечения их высокой производительности. Разработанные модели и исследованы влияние команд переходов на производительность подсистем выполнения и подсистем генерации конвейерных ядер процессоров. Проведен сравнительный анализ эффективности структурных методов увеличения производительности подсистемы подготовки команд.

Практическое значение, область применения работы: Результаты исследования могут быть полезны при оптимальном выборе структуры ядра с учетом соотношения как показателей эффективности, так и аппаратных затрат на реализацию

Ключевые слова: ядро, процессор, конвейер, помехи, конвейер, суперскалярность, эффективность, производительность.

ABSTRACT

Kostyria RG. Models and Methods for Investigating the Effectiveness of Structural Methods for Increasing the Performance of a Command Generation Subsystem in Modern Processor Kernels

The structures of cores of modern processors and methods of ensuring their high performance are considered and analyzed. Models and investigations of the impact of transition teams on the performance of the performance subsystems and the generation systems of the conveyor cores of processors have been developed. A comparative analysis of the effectiveness of structural methods of increasing the productivity of the subsystem training teams.

Practical value, scope of work: The results of the study can be useful in the optimal choice of the structure of the kernel, taking into account the ratio of both performance indicators and hardware implementation costs.

Keywords: core, processor, pipeline, interference, conveyor, interference, supercollarity, efficiency, performance.

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1 ДОСЛІДЖЕННЯ СТРУКТУР ЯДЕР ТА МЕТОДІВ ЗАБЕЗПЕЧЕННЯ ВИСОКОЇ ПРОДУКТИВНОСТІ	12
1.1 Структура ядра сучасних процесорів	12
1.2 Структурні особливості процесорних ядер Skylake	12
1.3 Основні підсистеми структури ядра	14
1.3.1 Блок вибірки команд	14
1.3.2 Декодування і черга декодованих команд IDQ	14
1.3.3 Дешифратор	15
1.3.4 Кеш-пам'ять дешифрованих команд	16
1.3.5 Блок передбачення переходів	16
1.3.6 Буфер впорядкування та відставки	16
1.3.7 Планувальник	17
1.3.8 Регістри	17
1.3.9 Функціональні пристрої	18
1.4 Методи досягнення високої продуктивності ядра	19
1.4.1 Збільшення частоти	19
1.4.2 Конвеєрний принцип виконання команд	19
1.4.3 Багаторівнева підсистема кешування	19
1.4.4 Позачергове виконання команд	20
1.4.5 Суперскалярність	20
1.5 Про дослідження в літературі ефективності різних структурних методів збільшення продуктивності ядер	20
РОЗДІЛ 2 ПРО ВПЛИВ КОМАНД ПЕРЕХОДІВ НА ПРОДУКТИВНІСТЬ ПІДСИСТЕМ ВИКОНАННЯ КОНВЕЄРНИХ ЯДЕР ПРОЦЕСОРІВ	22
2.1 Структурні методи компенсації впливу залежностей з керування	22
2.2 Структура ядра процесора з блоком передбачення переходу і спеціальним буфером дешифрованих команд	22
2.3 Модель ядра процесора	23
2.4 Стани моделі та система рівнянь	24
2.5 Аналітичне розв'язання системи рівнянь	26
2.5.1 Коефіцієнт завантаження ФП	26
2.5.2 Коефіцієнт завантаження дешифратора	27

2.6 Висновки до розділу 2	27
РОЗДІЛ 3 ДОСЛІДЖЕННЯ ВПЛИВУ КОМАНД ПЕРЕХОДІВ НА ПРОДУКТИВНІСТЬ ПІДСИСТЕМИ ПІДГОТОВКИ КОМАНД КОНВЕЄРНИХ ЯДЕР ПРОЦЕСОРІВ	28
3.1 Модель структури підсистеми підготовки команд	28
3.2 Висновки до розділу 3	33
РОЗДІЛ 4 ПОРІВНЯЛЬНИЙ АНАЛІЗ ЕФЕКТИВНОСТІ СТРУКТУРНИХ МЕТОДІВ ЗБІЛЬШЕННЯ ПРОДУКТИВНОСТІ ПІДСИСТЕМИ ПІДГОТОВКИ КОМАНД	34
4.1 Модель ядра процесора	34
4.2 Досліджувані варіанти структур підсистеми генерації команд	36
4.3 Визначення продуктивностей варіантів підсистеми генерації команд	37
4.3.1 Варіант 1. Підсистема генерації команд без суперскалярності та без кеш-пам'яті дешифрованих команд ($S = 1$, немає КДК).	37
4.3.2 Варіант 2 Підсистема генерації команд без суперскалярності, але з кеш- пам'яттю дешифрованих команд ($S = 1$, є КДК).	37
4.3.3 Варіант 3. Підсистема генерації команд з суперскалярністю, але без кеш-пам'яті дешифрованих команд ($S > 1$, немає КДК).	38
4.3.4 Варіант 4. Підсистема генерації команд з суперскалярністю та з кеш- пам'яттю дешифрованих команд ($S > 1$, є КДК).	38
4.4 Діапазони зміни змінних у виразах для продуктивності	39
4.5 Порівняння варіантів	41

4.5.1 Порівняння варіантів 2 і 1	41
4.5.2 Порівняння варіантів 3 та 1	43
4.5.3 Порівняння варіантів 3 та 2	44
4.5.4 Порівняння варіантів 4 і 1	46
4.6 Висновки до розділу 4	48
РОЗДІЛ 5 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ. ЕКОЛОГІЯ	49
5.1 Загальні питання з охорони праці	49
5.1.1 Правові та організаційні основи охорони праці	49
5.1.2 Організаційно-технічні заходи з безпеки праці	50
5.2 Аналіз стану умов праці	50
5.2.1 Вимоги до приміщення	51
5.2.2 Вимоги до організації робочого місця	51
5.2.3 Навантаження та напруженість процесу праці	52
5.3 Виробнича санітарія	53
5.3.1 Аналіз небезпечних та шкідливих факторів при розробці виробу	53
5.3.2 Пожежна безпека	55
5.3.3 Електробезпека	55
5.4 Гігієнічні вимоги до параметрів виробничого середовища	56
5.4.1 Мікроклімат	56
5.4.2 Освітлення	57
5.4.2.1 Розрахунок освітлення	57
5.4.3 Вентилювання	58
5.5 Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій	58
5.5.1 Розрахунок захисного заземлення	60
5.5.1.1 Послідовність розрахунку:	60
5.6 Екологія та охорона навколишнього середовища	63
5.7 Висновки до розділу 5	63
ЗАКЛЮЧЕННЯ	65
ЛІТЕРАТУРА	66
Додаток А	68
Додаток Б	73

ВСТУП

Стрімкий розвиток науки і проникнення людської думки в усі нові області разом з рішенням поставлених раніше проблем постійно породжує потік питань і ставить нові, як правило складніші, завдання. За часів перших комп'ютерів здавалося, що збільшення їх швидкодії в 100 разів дозволить вирішити більшість проблем, проте продуктивність сучасних супер ЕОМ сьогодні є явно недостатньою для багатьох учених. Гідродинаміка, сейсмозвідка і прогноз погоди, моделювання хімічних сполук, дослідження віртуальної реальності - ось далеко не повний список областей науки, дослідники яких використовують кожен можливість прискорити виконання своїх програм.

Нарощування об'єму кеш-пам'яті і числа команд, що запускаються на виконання, не дають приросту продуктивності процесорів, пропорційного апаратним ресурсам, що витрачаються, і енергії. Сьогодні потрібно абсолютно інші структурні підходи.

Процесор, що виконує кожен команду за чотири такти (вибір команди, декодування і вибір операндів, виконання, запис результату), в принципі, забезпечує корисну роботу тільки в такті "виконання". Для збільшення такої низької (25%) ефективності використовується конвеєрна організація: такти послідовно обраних команд поєднуються в часі так, що в кожен період одна з команд обов'язково знаходиться на етапі "виконання". Для організації конвеєра потрібна додаткова апаратура, але виграти в продуктивності це виправдовує. Для виконання перетворень, що задаються командами процесора, служить сукупність функціональних пристроїв (ФП), таких як цілочисельний арифметико-логічний пристрій (АЛУ), АЛУ з плаваючою точкою і інші. Якщо тривалість перетворення у ФП перевищує один такт, з'являється можливість здійснення іншим ФП (чи тим же ФП - при конвеєрній реалізації) перетворення, що задається наступною командою. При цьому не треба чекати завершення вже виконуваних перетворень, а в результаті обчислення прискорюються.

Процесори, що реалізують цю можливість, називають "суперскалярними". У них кожного такту декодуються декілька команд. Кожна команда може бути виконана в своєму ФП.

Зменшення тривалості такту, розробка процесорів з багатотактними конвеєрами (у тому числі з сукупністю багатотактних ФП) потенційно забезпечують швидкодію, пропорційну сумарній продуктивності використовуваних ФП. Проте прості пристроїв через неможливість завантажити при виконанні програми обумовлюють зниження продуктивності. Ці прості пов'язані, принаймні, з трьома чинниками: очікування завершення обміну з істотно повільнішою пам'яттю, виконання команд умовних

переходів, залежності між командами - за даними і використовуваними ресурсами (наприклад, регістрами). Для боротьби з простоями, пов'язаними з повільною пам'яттю в обчислювальних системах використовується багаторівнева кеш-пам'ять, використовується розшарування пам'яті. Усе це дозволяє при забезпеченні тимчасової і просторової локалізації звернень до пам'яті створювати ефект швидкого доступу.

Для зменшення втрат продуктивності із-за простоїв ФП, обумовлених визначенням напрямів переходів у виконуваний програмі, служать схеми передбачення переходів і переупорядкування команд (виконання в іншому порядку). Останній метод дозволяє ослабити вплив на час простоїв ФП істинних залежностей між командами читання після запису (обчислення повинні зупинитися на команді, яка використовуватиме ще не отриманий результат іншої команди, - поступила на виконання, але не завершеною).

Сукупність цих і ряду інших прийомів забезпечила підвищення продуктивності сучасних комп'ютерів.

Актуальність теми. За часів перших комп'ютерів здавалося, що збільшення їх швидкодії в 100 разів дозволить вирішити більшість проблем, проте продуктивність сучасних комп'ютерів сьогодні є явно недостатньою для багатьох учених. Тому, дослідження існуючих структур і пошук засобів для підвищення їх ефективності є актуальним.

Об'єктом дослідження магістерської роботи є конвеєрне суперскалярне ядро процесора.

Предметом дослідження – аналітичні, чисельні, імітаційні моделі, необхідні для проведення досліджень комп'ютерних систем.

Метою є дослідження впливу структурних рішень, за допомогою яких забезпечується паралелізм при виконанні команд, на ефективність функціонування обчислювальної системи.

Для досягнення поставленої мети, в магістерській роботі сформульовані та вирішені такі **задачі**:

- проаналізовані структури ядер сучасних процесорів на прикладі процесорів фірми Intel та методи дослідження складних структур;
- розроблені аналітичні моделі та імітаційні моделі конвеєрних суперскалярних ядер процесорів при наявності «перешкод» роботі конвеєра. приведені результати досліджень.

Методи рішення поставлених задач базуються на використанні ідей систем масового обслуговування, сучасних технологій для створення аналітичних та імітаційних моделей.

Наукова новизна одержаних результатів. Одержали подальший розвиток моделі для досліджень ядер процесорів при наявності «перешкод». Розроблені моделі для досліджень ефективності ядер процесорів з різними структурними рішеннями, які враховують вплив «перешкод» роботі конвеєра ядра процесора. Отримано ряд аналітичних результатів для оцінки ефективності різних структурних рішень в ядрі процесора.

Практичне значення. Розроблені моделі дозволяють проводити дослідження в галузі комп'ютерних технологій, реально та обґрунтовано оцінювати доцільність нововведень, які пропонуються при розробці ядер процесорів.

Структура та обсяг магістерської роботи. Магістерська робота складається зі вступу, п'ятих розділів, висновків, списку використаних джерел та додатків. Робота викладена на 78 сторінках машинописного тексту, містить 37 сторінок основного тексту, 6 рисунків, 14 таблиць, 2 додатки на 11 сторінках. Бібліографічний список включає 12 найменувань.

РОЗДІЛ 1

ДОСЛІДЖЕННЯ СТРУКТУР ЯДЕР ТА МЕТОДІВ ЗАБЕЗПЕЧЕННЯ ЇХ ВИСОКОЇ ПРОДУКТИВНОСТІ

1.1 Структура ядра сучасних процесорів

Ядро процесора - це основна його частина, яка містить всі функціональні блоки виконує всі логічні і арифметичні операції.

1.2 Структурні особливості процесорних ядер Skylake

Базова структура в спрощеному вигляді наведена на рис. 1.1.

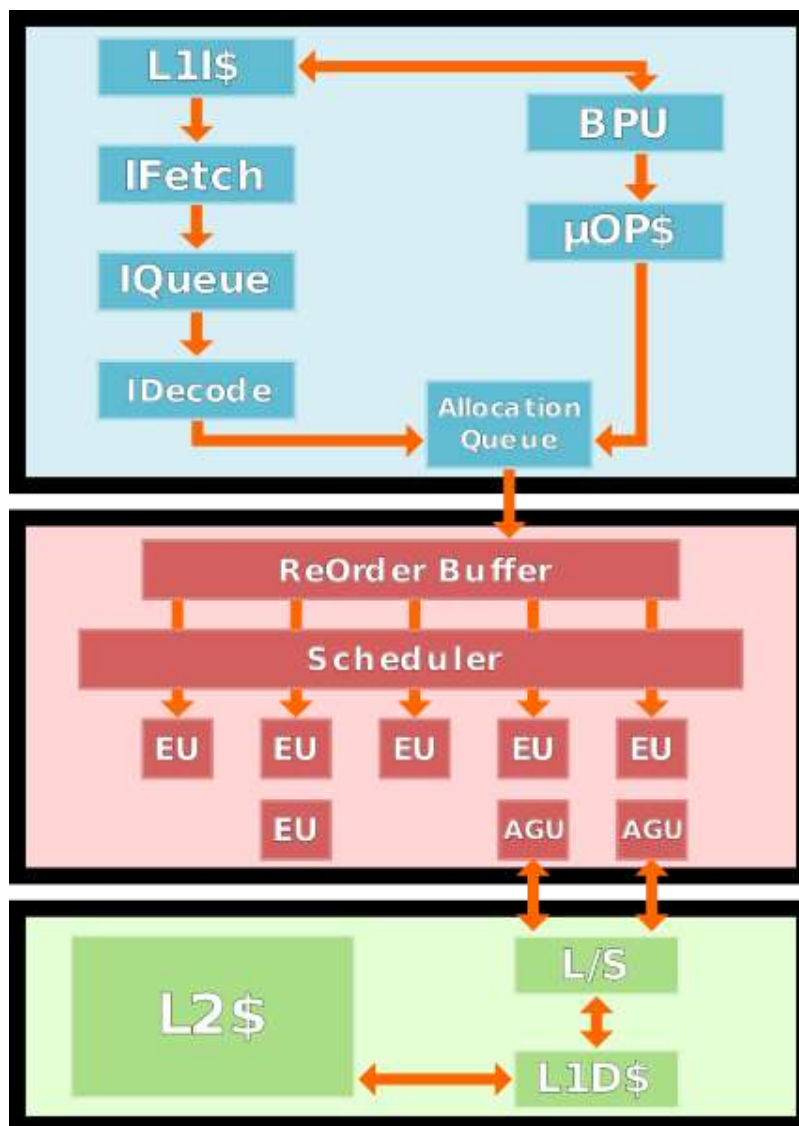


Рис. 1.1 - Структура ядра Skylake SP

Детальна структура ядра Skylake SP приведена на рис. 1.2.

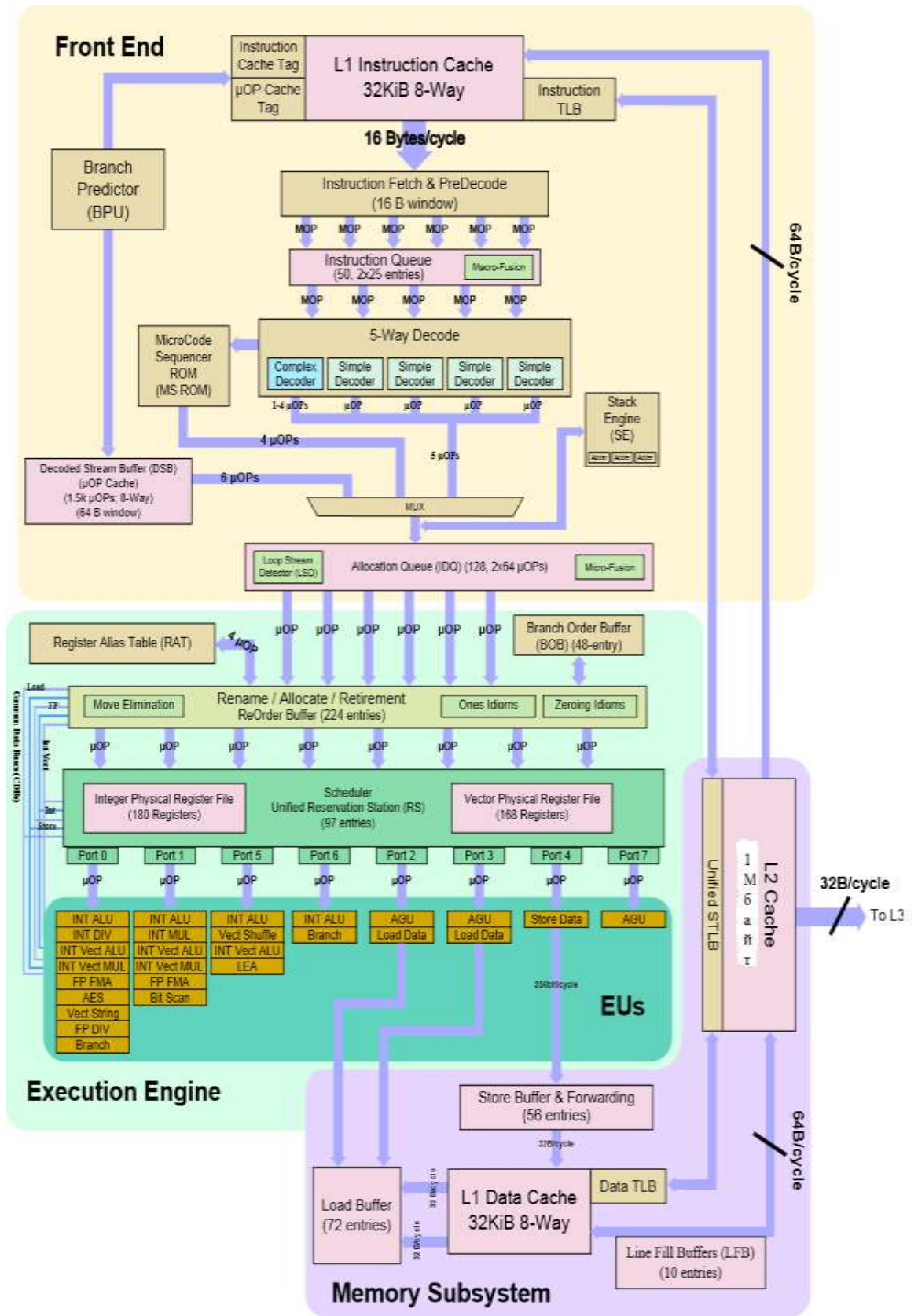


Рис. 1.2 - Детальна структура ядра Skylake-SP

1.3 Основні підсистеми структури ядра

1.3.1 Блок вибірки команд

Виконання будь-якої команди ядром процесора починається з вибірки команди з підсистеми пам'яті (перш за все з кеш-пам'яті команд L1K). РНК (регістр поточного номера команди) містить віртуальну адресу поточного блоку командної інформації. Ця віртуальна адреса видається в перетворювач віртуальних адрес в фізичні адреси (буфер TLBK для кеш-пам'яті команд L1K і буфер TLBD для кеш-пам'яті даних L1D)

Буфер TLB - це спеціальна кеш-пам'ять чисто асоціативного типу. На вхід цієї кеш-пам'яті подається номер віртуальної сторінки (старші розряди віртуальної адреси). Виконується порівняння цього віртуального номера з вмістом кожного осередку кеш-пам'яті. При вдалому пошуку з кеш-пам'яті витягується номер фізичної сторінки. До нього приєднується зсув в сторінці (молодші розряди віртуальної адреси), і фізична адреса готова.

Буфера TLB можуть бути однорівневими або дворівневими. В сучасних ядрах процесорів ємність буфера TLB становить десятки або навіть сотні рядків. Це означає, наприклад, що буде швидке перетворення віртуальної адреси (ВА) у фізичну адресу (ВА → ФА) для досить великого адресного простору.

У разі невдалого пошуку виникає переривання. Обробка цього переривання відбувається в привілейованому режимі при відключеній кеш-пам'яті L1K. Час обробки цього переривання дуже великий. Час вдалого перетворення віртуальної адреси в фізичну адресу становить (1 - 2) такти.

Фізична адреса блоку командної інформації подається на вхід кеш-пам'яті команд (L1K). Фізична адреса даних подається на вхід кеш-пам'яті даних (L1D).

Блок командної інформації (це 16 - 32) байти, з кеш-пам'яті L1K заноситься в буфер попереднього дешифрування.

Після попереднього дешифрування блоку команд (визначені межі команд в блоці, їх типи та інша необхідна інформація), він направляється на входи дешифраторів команд.

1.3.2 Декодування і черга декодованих команд IDQ

Передбачені наперед адреси виконуваних команд (по черзі для кожного потоку - при включеній технології Hyper-Threading) видають для перевірки їх наявності в кеш-пам'яті команд (L1K) і в кеш-пам'яті МОПів (L0m).

З кеш-пам'яті команд L1K порція (блок командної інформації) потрапляє в буфер декодера, а звідти - в сам декодер-довжиномір (ILD), що обробляє до 7 або 6 команд за такт (з і без макрозлиття) в залежності від їх сукупної довжини та складності.

Розмічені команди потрапляють в одну з двох черг декодера IQ, по одній на потік. Декодер читає команди з черги і переводить їх в МОПи. Він складається з:

- кількох (4 і більше) простих трансляторів, які переводять 1 команду в 1 МОП, а з макрозлиттям - 2 команди в 1 МОП;
- складного транслятора, який переводить 1 команду в (2-4) Мопи;
- мікросеквенсера для найскладніших команд, які потребують 5 і більше МОПів з «прошивки».

Результат декодування надходить в кеш-пам'ять МОПів і в дві черги МОПів (по одній на потік). Черги МОПів (офіційно іменуються IDQ - instruction decode queue, черга декодованих команд) мають можливість блокування циклу, якщо його виконувана частина там уміщається.

1.3.3 Дешифратор

Функціональні виконавчі блоки всіх ядер сучасних процесорів не працюють з командами архітектури IA-32/64. У всіх ядер кожного процесора є своя, «внутрішня» система команд, яка має мало спільного з тими командами, які описані в архітектурі. Команди, які виконуються ядром набагато простіші, «примітивніші», ніж команди архітектури.

Дешифратор перетворює команди архітектури IA-32/64 у «внутрішні» команди (МОПи), що виконуються ядром (при цьому досить часто одна вихідна команда перетвориться в кілька простіших «внутрішніх»). Дешифратор є важливою частиною ядра сучасного процесора: від його швидкодії залежить те, наскільки постійним буде потік команд, що надходять на виконавчі функціональні блоки. Чи будуть вони працювати або простоювати - багато в чому залежить від ефективності роботи дешифратора.

У високопродуктивних ядрах процесорів є не менше 4 дешифраторів. Якщо дешифраторів 4, то 3 дешифратора використовуються для дешифрування простих команд, а 1 дешифратор - для перетворення складних команд.

Прості команди архітектури IA-32/64 за дешифрування представляються за допомогою одного-двох МОПів, а складні команди - трьома і більше мопами. 3 дешифратора для дешифрування простих команди реалізовані апаратно, а один дешифратор для дешифрування складних команд - з використанням мікропрограм.

1.3.4 Кеш-пам'ять дешифрованих команд

Одним з найбільш важливих нововведень в структурах ядер, починаючи зі структури Sandy Bridge, з'явилася кеш-пам'ять дешифрованих мікрооперацій, або кеш-пам'ять команд L0m ємністю кілька тисяч мікрооперацій. Вона кешує на виході декодера всі попередньо дешифровані мікрооперації. Як тільки надходить на обробку нова команда, блок упереджувальної вибірки насамперед зв'язується з кеш-пам'яттю L0m, і в разі виявлення збігу, завантаження конвеєра по чотири мікрооперації за такт в обхід основних декодерів здійснюється вже з кеш-пам'яті L0m. Незадіяні і простоючі ланцюги декодерів (дешифраторів), до речі, вельми складні, і тому досить «енергоспоживаючі», в цей момент просто відключаються від живлення. В іншому випадку, коли кеш-пам'ять дешифрованих команд виявляється не використовується, триває звичайна робота по вибірці і декодуванню команд, а кеш-пам'ять дешифрованих команд переводиться в режим економії енергії.

Кеш-пам'ять L0m в якійсь мірі можна вважати частиною кеш-пам'яті першого рівня L1К, в яку вона, до речі, інтегрована, але окремою і дуже швидкою її частиною. За деякими даними, при роботі з багатьма додатками, ймовірність вдалого «попадання» в кеш-пам'ять дешифрованих команд дуже велика і може досягати 80%.

1.3.5 Блок передбачення переходів

Коли в потоці команд зустрічаються розгалуження, ядро процесора, щоб не припиняти генерацію команд, намагається вгадати подальшу поведінку програми і продовжити роботу з найбільш імовірною гілкою програми. У таких випадках вибірка чергового блоку команд проводиться з використанням механізму передбачення переходів.

1.3.6 Буфер впорядкування та відставки

Після перейменування регістрів МОПи записуються в буфер впорядкування та відставки.

З буфера впорядкування МОПи в міру готовності їх до виконання видаються в спеціалізовані функціональні виконавчі пристрої за умови, що ті готові до їх прийому.

В хронологічному порядку, кожна МОП може знаходитися в наступних станах:

1. МОП ще не готова до виконання - вона знаходиться в черзі планувальника.

2. МОП готова до виконання (всі аргументи для неї готові).
3. МОП виконується одним із спеціалізованих функціональних виконавчих пристроїв.
4. МОП виконана і чекає «відставки» або скасування «спекулятивної» гілки (знаходиться в буфері впорядкування та відставки).
5. МОП знаходиться в процесі «відставки».

МОПи продовжують зберігатися в буфері впорядкування до тих пір, поки не будуть виконані і відставлені все більш старі МОПи. Під час відставки проводиться запис остаточних значень в архітектурні реєстри і оперативну пам'ять. Відставка операцій, видалення інформації про них з буфера впорядкування і запис остаточних значень проводиться в програмному порядку, в якому МОПи надійшли в буфер впорядкування. Це необхідно для того, щоб в разі виключення або переривання скасувати результати всіх наступних операцій, виконаних у позачерговому порядку.

1.3.7 Планувальник

Планувальник (один або кілька) видачі МОПів аналізує стан всіх спеціалізованих функціональних виконавчих пристроїв (ФУі), а також готовність МОПів в буфері. Якщо в буфері для ФУі є хоча б одна готова до виконання МОП і ФУі вільний (звільнилася перший ступінь конвеєра - якщо ФУі конвеєрного типу, або завершено виконання операції - якщо ФУі комбінаційного типу), то планувальник видає МОП з буфера на вхід ФУі.

Заявки з буфера видаються на виконання не в тому порядку, в якому вони слідують у програмі, а по готовності. У конкретне ФУі готові до виконання МОПи видаються відповідно до дисципліни FIFO.

При заповненні буфера впорядкування дешифратору команд забороняється записувати нові МОПи.

За один такт планувальник може видати кілька МОПів, в залежності від структури ядра процесора (наприклад, в ядрах зі структурою Haswell теоретично можна в одному такті видати до 7 МОПів).

1.3.8 Регістри

У кожному ядрі є значна кількість реєстрів різного роду. це:

- архітектурні реєстри загального призначення;

- адресні реєстри;
- службові;
- керуючі;
- буферні реєстри.

Крім архітектурних реєстрів загального призначення (цілочисельних, реєстрів даних з плаваючою точкою, «векторних» реєстрів) в ядрі є кілька сот, так званих, фізичних реєстрів, які утворюють фізичний реєстровий цілочисельний файл і фізичний реєстровий файл даних з плаваючою точкою.

При перейменуванні реєстрів кожен архітектурний реєстр підміняється одним з фізичних реєстрів. У функціональних пристроях операції виконуються над вмістом фізичних реєстрів. На етапі «відставки», якщо при виконанні команди не відбулося винятків і всі попередні команди завершилися нормально, відбувається «відставка» команди і вміст результату з фізичного реєстру переписується в відповідний архітектурний реєстр.

Між ступенями конвеєра підготовки команд в ядрі є буфери. Ємність цих буферів залежить від виконуваних функцій і структури ядра.

1.3.9 Функціональні пристрої

Підсистема обробки даних сучасних структур високопродуктивних ядер процесорів складається з декількох (не менше 6) спеціалізованих функціональних пристроїв:

- пристрій для виконання операцій над числами;
- пристрою для формування адрес оперативної пам'яті (3 пристрої в найбільш продуктивних ядрах);
- кеш-пам'ять даних першого рівня L1D;
- кеш-пам'ять другого рівня L2.

Спеціалізовані пристрої можуть бути двох типів - конвеєрні або комбінаційні. Спеціалізовані пристрої конвеєрного типу можуть починати виконання нових операцій кожного такту. Спеціалізовані пристрої комбінаційного типу можуть починати виконання нових операцій тільки після завершення попередньої операції.

Сукупність перших етапів конвеєра ядра, які виконують функції з читання команд з підсистеми пам'яті, буфертування їх, «розмітку» (якщо це необхідно), попереднього і остаточного дешифрування утворює підсистему генерації команд. Наступні етапи конвеєра команд ядра виконують згенеровані команди.

1.4 Методи досягнення високої продуктивності ядра

1.4.1 Збільшення частоти

Найбільш простий шлях до збільшення продуктивності комп'ютера, це збільшення частоти. Якби, наприклад, вдалося всі затримки в комп'ютері скоротити в декілька разів, то це призвело б до збільшення швидкодії в таке ж число разів. В останні роки були досягнуті величезні успіхи в створенні швидкодіючої елементної бази та відповідних методів монтажу. Очікується подальший прогрес, заснований на використанні нових технологій і зниження розмірів пристроїв. Цей шлях, однак, має ряд обмежень, і в даний час частота процесорів стабілізувалася в районі 4-5 ГГц:

1.4.2 Конвеєрний принцип виконання команд

Конвеєрна обробка передбачає роздільне виконання деякої операції в кілька етапів (кілька ступенів) з передачею даних від одного етапу до наступного. Продуктивність при цьому зростає завдяки тому, що одночасно на різних етапах конвеєра виконуються декілька операцій. Конвеєризація ефективна тільки тоді, коли завантаження конвеєра близьке до повного, а швидкість подачі нових операндів відповідає максимальній продуктивності конвеєра. Якщо відбувається затримка, то паралельно буде виконуватися менше операцій і сумарна продуктивність знизиться.

Використання конвеєрного принципу та інших структурних методів дозволило істотно підвищити продуктивність ядер процесорів. Однак можливості традиційних структурних методів прискорення виконання команд в одному ядрі процесора на сучасному етапі практично вичерпані.

1.4.3 Багаторівнева підсистема кешування

Система кешування в переважній більшості сучасних процесорів є багаторівневою (трирівневою). Кеш першого рівня (самий «близький» до ядра) традиційно поділяється на дві (як правило, рівні) половини: кеш інструкцій (L1I) і кеш даних (L1D). Спочатку інформація шукається в L1, якщо її там немає - в L2, потім - в L3, і вже потім, якщо ні на одному рівні кеша вона не знайдена - йде звернення до основної пам'яті.

1.4.4 Позачергове виконання команд

Основна ідея позачергового виконання полягає в можливості уникнути простою ядра процесора в тих випадках, коли дані, необхідні для виконання чергової команди, недоступні.

Ядро з позачерговим виконанням виконує ті команди, операнди яких готові, незалежно від порядку проходження команд в програмі. Команди виконуються не в порядку їх слідування в програмі, а в порядку доступності даних в регістрах процесора.

Позачергове виконання команд тим ефективніше, чим довший конвеєр ядра процесора, і чим більша різниця між швидкістю роботи пам'яті (оперативної або кешів) і швидкістю роботи ядра процесора. У сучасних комп'ютерах ядра процесорів працюють набагато швидше пам'яті, тому під час очікування надходження даних процесор може виконати декілька команд, операнди яких доступні.

1.4.5 Суперскалярність

Суперскалярне ядро процесора - це ядро, яке здатне генерувати (і виконувати, відповідно) кілька команд.

При використанні конвеєра кількість вузлів не змінюється; збільшення продуктивності досягається за рахунок одночасної роботи вузлів, відповідальних за різні стадії обробки інструкцій одного потоку. При використанні суперскалярності збільшення продуктивності досягається за рахунок одночасної роботи великої кількості однакових вузлів, незалежно оброблених команди одного потоку (в тому числі, і більшої кількості конвеєрів).

1.5 Про дослідження в літературі ефективності різних структурних методів збільшення продуктивності ядер

Вплив деяких методів збільшення продуктивності ядер процесорів розглянуто в роботах [6], [7], [8].

Зокрема, в роботі [6] наведено методику та оцінки коефіцієнтів використання підсистеми генерації команд та підсистеми виконання команд в найсприятливіших умовах роботи конвеєрного ядра при відсутності «перешкод» для будь-яких значень: розміру буфера готових для виконання команд; коефіцієнтів навантаження функціональних

пристроїв; типів програм. Навіть при відсутності «перешкод» коефіцієнти використання спеціалізованих функціональних пристроїв далекі від граничних значень.

В роботі [7] досліджені ефективності деяких структурних рішень щодо зменшення негативного впливу команд умовних переходів.

В роботі [8] розроблена модель підсистеми генерації команд при виконанні програм, що складаються з обчислювальних команд і команд переходів різного типу.

Більш повні роботи з дослідження впливу на продуктивність підсистеми підготовки команд різних параметрів як структури підсистеми підготовки команд, так і характеристик виконуваних програм, а також ефективності різних структурних методів збільшення продуктивності ядер сучасних процесорів автору невідомі.

Мета магістерської роботи. Дослідження ефективності варіантів структури підсистеми генерації команд (з і без суперскалярності; з і без кеш-пам'яті дешифрованих команд; з і без блоку передбачення переходів) в залежності від параметрів як структури ядра, так і характеристик виконуваних програм.

Для досягнення поставленої мети в магістерській роботі розроблені моделі, методики та аналітичні вирази для оцінки ефективності різних варіантів структури підсистеми генерації команд.

РОЗДІЛ 2

ПРО ВПЛИВ КОМАНД ПЕРЕХОДІВ НА ПРОДУКТИВНІСТЬ ПІДСИСТЕМ ВИКОНАННЯ КОНВЕЄРНИХ ЯДЕР ПРОЦЕСОРІВ

2.1. Структурні методи компенсації впливу залежностей з керування

В сучасних високопродуктивних ядрах процесорів для компенсації впливу залежностей з керування використовуються блоки передбачення напрямку переходу, а також спеціальні буфери дешифрованих команд.

У найпростішому варіанті структури ядра може бути відсутнім як блок передбачення напрямку переходу, так і спеціальний буфер дешифрованих команд.

Існують структури ядер з блоком передбачення напрямку переходу і спеціальним буфером дешифрованих команд.

Доцільно розглянути варіант структури ядра з блоком передбачення напрямку переходу і спеціальним буфером дешифрованих команд.

2.2 Структура ядра процесора з блоком передбачення переходу і спеціальним буфером дешифрованих команд

В цьому варіанті дешифровані команди з дешифратора направляються як у буфер підсистеми обробки, так і в буфер дешифрованих команд. Наприклад, ємність буфера дешифрованих команд в ядрах процесорів фірми Intel зі структурою Sandy Bridge становила 1500 команд. При «вдалому» прогнозі напрямку переходу дешифровані команди нової гілки витягуються із спеціального буфера дешифрованих команд в буфер підсистеми обробки, минаючи дешифратор.

Час перезавантаження конвеєра підготовки команд при «вдалому» прогнозі напрямки переходу складе

$$T_{2.уд} = t_{Б.пред} + t_{чт.буф.ком} ,$$

де $t_{чт.буф.ком}$ - час читання першої команди нової гілки з швидкого буфера дешифрованих команд.

При «невдалому» прогнозі напрямку переходу час перезавантаження конвеєра підготовки команд складе

$$T_{2.неуд} = t_{ож} + t_{Б.пред} + t_{чит.пам} + t_{зап.кон}.$$

2.3 Модель ядра процесора

Структура моделі ядра з урахуванням припущень представлена на рис. 2.1.

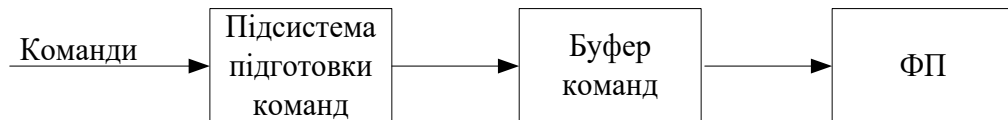


Рис.2.1. Структура моделі ядра процесора

Зроблені такі припущення:

перший етап (підсистема підготовки команд) генерує потік команд (заявок) з інтенсивністю λ . На її вхід по конвеєру надходить нескінченна послідовність команд програми наступних типів:

команди обробки і команди звернення до підсистеми пам'яті - з ймовірністю ω_0 ;

команди умовного переходу - з ймовірністю ω_n ($\omega_n + \omega_0 = 1$);

команди обробки, команди звернення до підсистеми пам'яті виконуються послідовно в другому ступені (в підсистемі виконання, яка представлена як універсальний функціональний пристрій - ФУ);

між етапами є буфер на n заявок;

потік заявок, що згенерував дешифратор, найпростіший з експоненціальним законом розподілу. Якщо буфер заявок заповнений, то дешифратор блокується, тобто зупиняється.

Підсистема виконання (ФП) виконує заявки (команди) з інтенсивністю μ . Закон розподілу часу виконання заявок експонентний. Дисципліна вибору заявок з буфера - FIFO. При відсутності підготовлених до виконання заявок ФП простоює. Якщо буфер порожній і дешифратор згенерував чергову заявку, то вона миттєво надходить в ФП.

При виявленні чергової команди переходу відбувається перезавантаження конвеєра підготовки команд з інтенсивностями μ_{n1} (при «вдалому» прогнозі напрямки переходу з ймовірністю q) або μ_{n2} (при «невдалому» прогнозі напрямку переходу з ймовірністю $1-q$). На час перезавантаження конвеєра підготовки команд дешифратор блокується

(простоює). Підсистема виконання (ФП) при наявності готових до виконання заявок (команд) виконує їх (працює).

2.4. Стани моделі та система рівнянь

Граф станів моделі системи наведено на рис. 2.2.

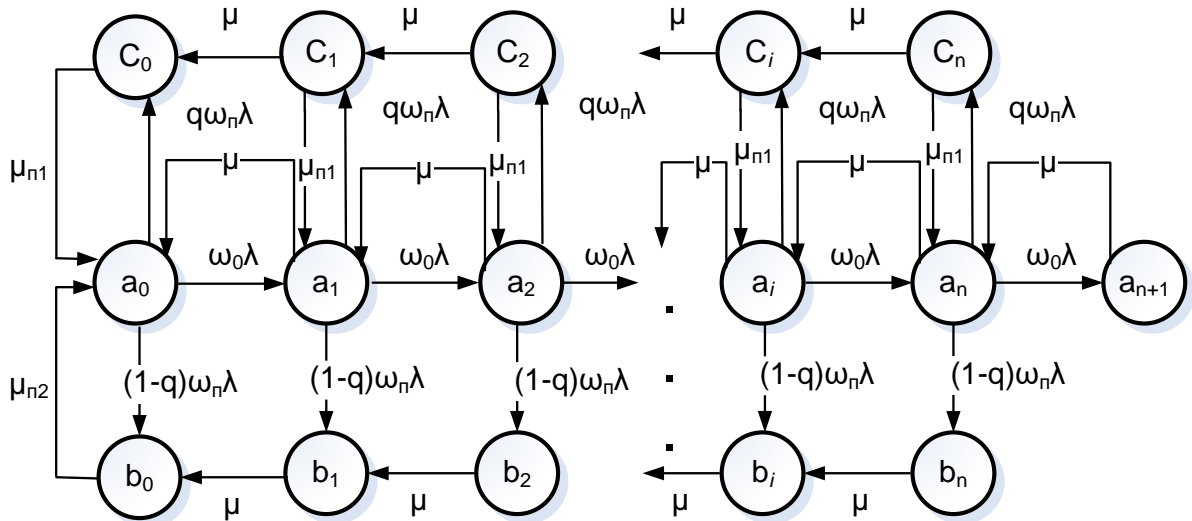


Рис. 2.2 - Граф станів моделі системи з блоком передбачень

Стани моделі ідентифікуються за допомогою індексу i ($0 \leq i \leq n+1$), що відображає кількість згенерованих першим етапом (дешифратором) заявок. Заявки можуть перебувати - одна на виконанні в ФП, $(i-1)$ - в буфері. Дешифратор команд і ФП можуть або працювати, або простоювати.

У станах:

a_i ($1 \leq i \leq n$) - дешифратор і ФП працюють;

a_0 - дешифратор працює, а ФП простоює через відсутність заявок;

a_{n+1} - дешифратор команд простоює, так як буфер заповнений, ФП працює;

C_i ($1 \leq i \leq n$) - виконується перезавантаження конвеєра підготовки команд для дешифратора при «вдалому» прогнозі напрямку переходу, дешифратор команд блокований, ФП працює;

C_0 - виконується перезавантаження конвеєра підготовки команд для дешифратора при «вдалому» прогнозі переходу, дешифратор команд блокований, ФП не працює;

b_i ($1 \leq i \leq n$) - дешифратор блокований, так як він очікує завершення всіх раніше згенерованих заявок, ФП працює;

b_0 - виконується перезавантаження конвеєра підготовки команд для дешифратора при «невдалому» прогнозі напрямки переходу, дешифратор команд і ФП не працюють.

У кожному з перерахованих станів система може перебувати з певною ймовірністю. Рівняння балансу ймовірностей для станів наступні:

$$P_{a0}\lambda = P_{b0}\mu_{n2} + P_{a1}\mu + P_{c0}\mu_{n1}$$

$$P_{ai}(\lambda + \mu) = P_{ai-1}\omega_0\lambda + P_{a(i+1)}\mu + P_{ci}\mu_{n1}; (1 \leq i \leq n)$$

$$P_{an+1}\mu = P_{an}\omega_0\lambda$$

$$P_{b0}\mu_{n2} = P_{a0}(1-q)\omega_n\lambda + P_{b1}\mu$$

$$P_{bi}\mu = P_{ai}(1-q)\omega_n\lambda + P_{b(i+1)}\mu; (1 \leq i \leq n-1),$$

$$P_{bn}\mu = P_{an}(1-q)\omega_n\lambda$$

$$P_{c0}\mu_{n1} = P_{a0}q\omega_n\lambda + P_{c1}\mu$$

$$P_{ci}(\mu + \mu_{n1}) = P_{ai}q\omega_n\lambda + P_{c(i+1)}\mu; (1 \leq i \leq n-1),$$

$$P_{cn}(\mu + \mu_{n1}) = P_{an}q\omega_n\lambda.$$

В якості показників ефективності використовуються коефіцієнти завантаження дешифратора і функціонального пристрою.

2.5 Аналітичне розв'язання системи рівнянь

З одноманітного визначення ймовірностей станів

$P_{ai}; P_{bi}; P_{ci}$ ($2 \leq i \leq n$) випливає, що:

$$P_{ai} = RP_{ai-1}; P_{bi} = RP_{bi-1}; P_{ci} = RP_{ci-1},$$

але

$$P_{a1} \neq RP_{a0}; P_{b1} \neq RP_{b0}; P_{c1} \neq RP_{c0}.$$

З рівняння

$$P_{ai-1} * [R^2 - R(1 + \rho - \frac{A_1}{1 + A_1} \omega_n q \rho) + \omega_o \rho] = 0$$

коefficient R визначається наступним чином

$$R = \frac{D - \sqrt{D^2 - 4\omega_o \rho}}{2},$$

де $\rho = \frac{\lambda}{\mu}; A_1 = \rho * \rho_{n1}; \rho_{n1} = \frac{\mu_{n1}}{\lambda};$

$$D = 1 + \rho - \frac{A_1}{1 + A_1} q \omega_n \rho.$$

2.5.1 Коefіцієнт завантаження ФП

$$E_{\phi y} = 1 - (P_{a0} + P_{b0} + P_{c0})$$

$$E_{\phi y} = \frac{F * [(1 - R^{n+1}) + \omega_n \rho (1 - R^n)] \left\{ \frac{q}{1 + A_1} + \frac{1 - q}{1 - R} \right\}}{1 - R + F(1 - R^{n+1}) + \omega_n \rho \left\{ \frac{q}{A_1(1 + A_1)} [(1 + A_1 + F)(1 - R) + A_1 F(1 - R^n)] + \frac{(1 - q)}{A_2} [1 - R + F + \frac{A_2 F(1 - R^n)}{(1 - R)}] \right\}}$$

2.5.2 Коефіцієнт завантаження дешифратора

$$H_{\text{Деши}} = \sum_{i=0}^n P_{ai}$$

$$H_{\text{Деши}} = \frac{1 - R + F(1 - R^{n+1})}{1 - R + F(1 - R^{n+1}) + \omega_n \rho \left\{ \frac{q}{A_1(1 + A_1)} [(1 + A_1 + F)(1 - R) + A_1 F(1 - R^n)] + \frac{(1 - q)}{A_2} [1 - R + F + \frac{A_2 F(1 - R^n)}{(1 - R)}] \right\}}$$

$$\text{де } F = \frac{\omega_o * \rho}{1 + \omega_{\Pi} * \rho * \left(\frac{q}{1 + A_1} + \frac{1 - q}{1 - R} \right)}; \quad A_2 = \rho * \rho_{n2}; \quad \rho_{n2} = \frac{\mu_{n2}}{\lambda}$$

2.6 Висновки до розділу 2

1. Розроблена модель функціонування ядра при використанні одного з структурних методів компенсації втрат продуктивності.

2. Отримані аналітичні залежності показників ефективності (коефіцієнтів використання пристроїв) від різних параметрів (розміру буфера, коефіцієнтів перезавантаження конвеєра підготовки команд, частоти команд переходів, ймовірності «вдалих» передбачень напрямків переходів) структури з блоком передбачень та з буфером дешифрованих команд.

3. У високопродуктивних ядрах для компенсації негативного впливу команд переходу на продуктивність необхідний буфер дешифрованих команд.

РОЗДІЛ 3

ДОСЛІДЖЕННЯ ВПЛИВУ КОМАНД ПЕРЕХОДІВ НА ПРОДУКТИВНІСТЬ ПІДСИСТЕМИ ПІДГОТОВКИ КОМАНД КОНВЕЄРНИХ ЯДЕР ПРОЦЕСОРІВ

В цьому розділі досліджений вплив команд переходів на ефективність функціонування підсистеми підготовки команд з суперскалярністю в ідеальних умовах в залежності від ряду параметрів як структури ядра (q - ймовірності вдалих прогнозів переходів, A/B - відношення часу в тактах генерації першого пакету (команди) після переходу до часу в тактах генерації першого пакету (першої команди) при відсутності переходу), так і характеристик виконуваних програм (ω_{II} - частоти команд переходів).

3.1 Модель структури підсистеми підготовки команд

Для дослідження реальна структура ядра процесора представлена у вигляді моделі, яка складається з 2 фаз - підсистеми підготовки команд (ППК) і підсистеми виконання підготовлених команд (ПВК) з буфером між ними (рис. 3.1).

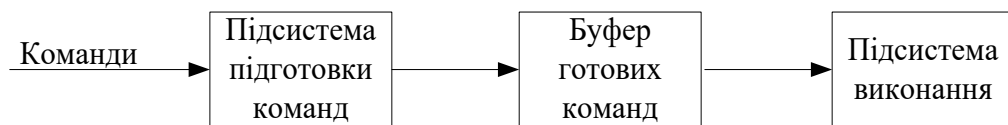


Рис.3.1 - Структура моделі ядра процесора

Ефективність функціонування підсистеми підготовки команд досліджується за таких умов:

на вхід ППК надходить нескінченна послідовність команд програми;

досліджувані програми виконуються в найбільш сприятливих з точки зору продуктивності умовах (зокрема, вся необхідна інформація знаходиться в кеш-пам'яті ядра);

розміри буферів між фазами ППК і буфера між ППК і ПВК необхідної величини;

з ймовірністю ω_{II} чергова команда програми це команда переходу будь-якого типу (чи то це команда безумовного переходу, то це команда умовного переходу);

з ймовірністю $1 - \omega_{II}$ чергова команда програми це команда, яка виконується в ПВК;

q - ймовірність вдалого передбачення переходу блоком передбачення переходів для команд переходу;

$1 - q$ - ймовірність невдалого передбачення переходу блоком передбачення переходів для команд переходу;

S - кількість команд, які генеруються ППК в одному пакеті (групі). Якщо $S = 1$, то в кожному такті генерується одна команда.

A - час в тактах генерації першого пакету (команди) після переходу;

B - час в тактах генерації першого пакету (першої команди) при відсутності переходу.

Визначимо середній час генерації одного пакета (першої команди) ППК наступним чином

$$t_{\text{ПАК}} = (1 - \omega_{\text{П}}) * B + \omega_{\text{П}} * [q * A + (1 - q) * B].$$

Після спрощення отримаємо

$$t_{\text{ПАК}} = B + \omega_{\text{П}} * q * (A - B).$$

Якщо в пакеті S команд, то середній час генерації однієї команди

$$t_{\text{КОМ}} = \frac{t_{\text{ПАК}}}{S} = \frac{B + \omega_{\text{П}} * q * (A - B)}{S}.$$

В ідеальному конвєєрі ППК

$$t_{\text{ПАК}}^{\text{ИДЕАЛ}} = B.$$

Продуктивність ППК це величина зворотна середньому часу підготовки однієї команди. Тоді продуктивність реальної ППК при відсутності суперскалярності

$$PP_{\text{РЕАЛ}} = \frac{1}{t_{\text{КОМ}}} = \frac{t_{\text{ПАК}}}{S} = \frac{S}{B + \omega_{\text{П}} * q * (A - B)}.$$

Продуктивність ідеальної ППК при відсутності суперскалярного

$$PP_{ИДЕАЛ} = \frac{B}{S}.$$

Часи підготовки команд виражені в тактах ядра процесора.

Визначимо ефективність функціонування ППК у вигляді відношення реальної продуктивності ППК до ідеальної продуктивності ППК

$$R = \frac{PP_{РЕАЛ}}{PP_{ИДЕАЛ}} = \frac{S * B}{S * [B + \omega_{II} * q * (A - B)]} = \frac{1}{1 + \omega_{II} * q * \left(\frac{A}{B} - 1\right)}.$$

У таблиці 3.1 наведені результати залежності ефективності функціонування ППК (коефіцієнта R) від параметрів ω_{II} , q , A/B .

Таблиця 3.1 - Залежності ефективності функціонування ППК (коефіцієнта R) від параметрів ω_{II} , q , A/B

ω_{II}	q	A/B	R
0.05	0.98	6	0.80321
		5	0.83612
		4	0.87184
		3	0.91075
		2	0.95329
		1	1
	0.95	6	0.80808
		5	0.84034
		4	0.87527
		3	0.91324
		2	0.95465
		1	1
0.10	0.98	6	0.67114
		5	0.71839
		4	0.77280
		3	0.83612
		2	0.91075
		1	1
	0.95	6	0.67797
		5	0.72464
		4	0.77821
		3	0.84033
		2	0.91324
		1	1

Продовження таблиці 3.1

ω_{II}	q	A/B	R
0.15	0.98	6	0.57639
		5	0.62972
		4	0.69396
		3	0.77280
		2	0.98551
		1	1
	0.95	6	0.58394
		5	0.63694
		4	0.70052
		3	0.77821
		2	0.87527
		1	1
0.20	0.98	6	0.50505
		5	0.56054
		4	0.62972
		3	0.71839
		2	0.83612
		1	1
	0.95	6	0.51282
		5	0.56818
		4	0.63694
		3	0.72464
		2	0.84034
		1	1

3.2 Висновки до розділу 3

1. При наявності в програмах команд переходів ($q \neq 0, A/B \neq 1$) ефективність ППК реальних ядер процесорів завжди менша 1.

2. При $A \rightarrow B$, тобто коли час перезавантаження конвеєра підсистеми підготовки команд при переході наближається до часу роботи етапу конвеєра при відсутності переходів, ефективність ППК прагне до граничного значення, рівного 1 незалежно від значення змінних ω_{II} і q .

3. При відсутності в програмах команд переходів ($\omega_{II} = 0$) ефективність ППК прагне до граничного значення, рівного 1.

4. При збільшенні ω_{II} ефективність ППК істотно зменшується.

5. Відношення A/B характеризує довжину конвеєра ППК. У ППК реальних ядер процесорів A і B відрізняються не менш ніж в 5-6 разів. Ось чому при розробці ППК ядер часи роботи окремих ступенів конвеєра ППК прагнуть зробити однаковими, а кількість ступенів мінімально можливим.

РОЗДІЛ 4

ПОРІВНЯЛЬНИЙ АНАЛІЗ ЕФЕКТИВНОСТІ СТРУКТУРНИХ МЕТОДІВ ЗБІЛЬШЕННЯ ПРОДУКТИВНОСТІ ПІДСИСТЕМИ ПІДГОТОВКИ КОМАНД

4.1 Модель ядра процесора

Для цілей дослідження реальна структура ядра процесора представлена у вигляді моделі, що складається з 2 фаз - підсистеми підготовки команд та підсистеми виконання підготовлених команд з буфером між ними (рис. 4.1). Буфер може бути як звичайним буфером, так і кеш-пам'яттю дешифрованих команд.

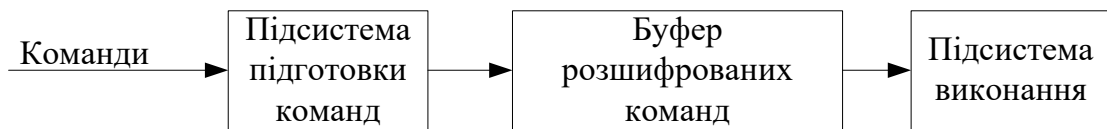


Рис. 4.1 - Структура моделі ядра процесора

У розділі досліджуються теоретичні (пікові) продуктивності чотирьох структур підсистеми генерації команд при виконанні наступних умов:

- в структурах ядер всіх варіантів є блок передбачення переходів;
- реалізований конвеєрний принцип;
- ширина і розміри всіх проміжних буферів між етапами підсистеми генерації команд достатньої величини;
- буфер між підсистемами моделі ядра необмеженої величини. Це дозволяє виключити з дослідження продуктивності підсистеми генерації команд структуру і продуктивність підсистеми виконання команд і її вплив на продуктивність підсистеми генерації команд.
- досліджувані програми виконуються в найбільш сприятливих з точки зору продуктивності умовах (зокрема, вся необхідна інформація знаходиться в кеш-пам'яті ядра);
- на вхід підсистеми генерації команд надходить нескінченна послідовність команд програми.

Параметри програм і досліджуваних структур підсистеми генерації команд ядра процесора наведені в таблиці 4.1.

Таблиця 4.1 - Параметри програм і досліджуваних структур підсистеми генерації

команд

Параметр	Опис
ω_{II}	ймовірність того, що чергова команда програми це команда переходу будь-якого типу (чи то це команда безумовного переходу, то це команда умовного переходу)
$1 - \omega_{II}$	ймовірність того, що чергова команда програми це команда, яка виконується в підсистемі виконання команд (обчислювальна команда)
q	ймовірність того, що перехід, визначений командою переходу, буде виконаний і буде здійснено перезавантаження конвеєра генерації команд
$1 - q$	ймовірність того, що перехід, визначений командою переходу, що не буде виконаний і перезавантаження конвеєра генерації команд не буде
S	кількість команд, які генеруються підсистемою генерації команд в одному такті. Якщо $S = 1$, то в кожному такті генерується одна команда
A	час (в тактах) перезавантаження конвеєра генерації команд в результаті виконання команди переходу при відсутності в структурі ядра кеш-пам'яті рошифрованих команд (КДК)
D	час (в тактах) перезавантаження конвеєра генерації команд в результаті виконання команди переходу при наявності в структурі ядра КДК
B	час (в тактах) генерації команд для структур ядра без КДК
B / S	B / S - час (в тактах) генерації однієї команди з пакета для структур ядра з ступенем суперскалярності $S > 1$ і без КДК
C	час (в тактах) генерації команди при її наявності в КДК
C / S	час (в тактах) генерації однієї команди з пакета при наявності команди в КДК для структур ядра зі ступенем суперскалярного $S > 1$.

4.2 Досліджувані варіанти структур підсистеми генерації команд

Варіант 1. Підсистема генерації команд без суперскалярності та без кеш-пам'яті дешифрованих команд ($S = 1$, без КДК).

Це найпростіший варіант структури підсистеми генерації команд. У підсистемі генерації команд є тільки один дешифратор команд. Чергова команда читається з підсистеми пам'яті (як правило з кешу команд L1К). Продешифрована команда заноситься в буфер підсистеми виконання команд.

Варіант 2. Підсистема генерації команд без суперскалярності, але з кеш-пам'яттю дешифрованих команд ($S = 1$, є КДК).

В ядрі процесора є кеш-пам'ять дешифрованих команд і тільки один дешифратор команд. Чергова порція командної інформації читається з підсистеми пам'яті (як правило, з кешу команд L1К). Продешифрована команда поміщається в КДК (кеш дешифрованих команд).

Варіант 3. Підсистема генерації команд з суперскалярністю, але без кеш-пам'яті дешифрованих команд ($S > 1$, немає КДК).

В ядрі процесора немає кеша дешифрованих команд і є S дешифраторів команд. Чергова порція командної інформації (пакет) читається з підсистеми пам'яті (як правило, з кешу команд L1К) і потім дешифрується необхідною кількістю дешифраторів, і пакет готових команд заноситься в буфер готових до виконання команд.

Варіант 4. Підсистема генерації команд з суперскалярністю і з кеш-пам'яттю розшифрувати команд ($S > 1$, є КДК).

Це найбільш складний варіант структури ядра. В ядрі процесора є кеш-пам'ять дешифрованих команд і S дешифраторів команд. Чергова порція командною інформації (пакет) читається з підсистеми пам'яті (як правило з кешу команд L1К) і потім дешифрується $S > 1$ дешифраторами. Продешифрований пакет команд заноситься в КДК.

4.3 Визначення продуктивностей варіантів підсистеми генерації команд

Під продуктивністю підсистеми генерації команд буде розумітися величина зворотна середньому часу генерації однієї команди.

4.3.1 Варіант 1. Підсистема генерації команд без суперскалярності та без кеш-пам'яті дешифрованих команд ($S = 1$, немає КДК).

Середній час генерації команди дорівнює

$$t_{КОМ} = (1 - \omega_{\Pi}) * B + \omega_{\Pi} * [q * (A + B) + (1 - q) * B].$$

Після спрощення

$$t_{КОМ} = B + \omega_{\Pi} * q * A.$$

Продуктивність першого варіанту структури підсистеми генерації команд

$$PP_{ВАР1} = \frac{1}{t_{КОМ}} = \frac{1}{B + \omega_{\Pi} * q * A}.$$

4.3.2 Варіант 2 Підсистема генерації команд без суперскалярності, але з кеш-пам'яттю дешифрованих команд ($S = 1$, є КДК).

Середній час генерації команди дорівнює

$$t_{КОМ} = (1 - \omega_{\Pi}) * [(1 - P_{КДК}) * B + P_{КДК} * C] + \omega_{\Pi} * \{ [q * [(1 - P_{КДК}) * (A + B) + P_{КДК} * D] + (1 - q) * [(1 - P_{КДК}) * B + P_{КДК} * C]] \}$$

Після спрощення

$$t_{КОМ} = P_{КДК} * C + (1 - P_{КДК}) * (B + \omega_{\Pi} * q * A).$$

Продуктивність другого варіанту структури підсистеми генерації команд

$$PP_{\text{ВАР2}} = \frac{1}{t_{\text{КОМ}}} = \frac{1}{P_{\text{КДК}} * C + (1 - P_{\text{КДК}}) * (B + \omega_{\text{П}} * q * A)}.$$

4.3.3 Варіант 3. Підсистема генерації команд з суперскалярністю, але без кеш-пам'яті дешифрованих команд ($S > 1$, немає КДК).

Середній час генерації команди дорівнює

$$t_{\text{КОМ}} = (1 - \omega_{\text{П}}) * B / S + \omega_{\text{П}} * [q * (A + B / S) + (1 - q) * B / S].$$

Після перетворень

$$t_{\text{КОМ}} = (1 - \omega_{\text{П}}) * B / S + \omega_{\text{П}} * (q * A + B / S),$$

$$t_{\text{КОМ}} = B / S + \omega_{\text{П}} * q * A$$

Продуктивність третього варіанту структури підсистеми генерації команд

$$PP_{\text{ВАР3}} = \frac{1}{t_{\text{КОМ}}} = \frac{1}{B / S + \omega_{\text{П}} * q * A}.$$

4.3.4 Варіант 4. Підсистема генерації команд з суперскалярністю та з кеш-пам'яттю дешифрованих команд ($S > 1$, є КДК).

Середній час генерації команди дорівнює

$$t_{\text{КОМ}} = (1 - \omega_{\text{П}}) * [(1 - P_{\text{КДК}}) B / S + P_{\text{КДК}} * C / S] + \omega_{\text{П}} * \{ [q * [(1 - P_{\text{КДК}}) * (A + B / S) + P_{\text{КДК}} * (D + C / S)] + (1 - q) * [(1 - P_{\text{КДК}}) * B / S + P_{\text{КДК}} * C / S] \}$$

Після перетворень

$$t_{\text{КОМ}} = P_{\text{КДК}} * C / S + (1 - P_{\text{КДК}}) * B / S + \omega_{\text{П}} * q * [(1 - P_{\text{КДК}}) * A + P_{\text{КДК}} * D].$$

Продуктивність цього варіанту структури підсистеми генерації команд

$$PP_{\text{ВАР4}} = \frac{1}{t_{\text{КОМ4}}} = \frac{1}{P_{\text{КДК}} * C / S + (1 - P_{\text{КДК}}) * B / S + \omega_{\text{П}} * q * [(1 - P_{\text{КДК}}) * A + P_{\text{КДК}} * D]}$$

Або

$$PP_{\text{ВАР4}} = \frac{S}{P_{\text{КДК}} * C + (1 - P_{\text{КДК}}) * B + S * \omega_{\text{П}} * q * [(1 - P_{\text{КДК}}) * A + P_{\text{КДК}} * D]}$$

4.4 Діапазони зміни змінних у виразах для продуктивності

Частота команд переходу $\omega_{\text{П}}$ визначається виключно типом виконуваної програми. Однією з поширених команд переходів є команда перевірки завершення циклу і перехід на початок циклу в переважній більшості випадків.

Довжина циклу (кількість команд в циклі) може змінюватися в широких межах - від одиниць до декількох десятків і більше.

Для дослідження обрано діапазон зміни від 0.05 до 0.20, що еквівалентно кількості команд в циклі від 20 до 5.

Ймовірність наявності дешифрованих команд в кеш-пам'яті дешифрованих команд

по визначенню ймовірності дорівнює
$$P_{\text{КДК}} = \frac{N_{\text{вдалих_звернень}}}{N_{\text{звернень}}},$$

де $N_{\text{вдалих_звернень}}$ - кількість звернень, коли роздешифровані команди є в кеш-пам'яті роздешифрованих команд;

$N_{\text{звернень}}$ - загальна кількість звернень в кеш-пам'ять роздешифрованих команд.

Якщо в циклі є k команд, а сам цикл виконується n раз, то

$$P_{\text{КДК}} = \frac{(n-1) * k}{n * k} = \frac{n-1}{n}.$$

При досить великих значеннях n $P_{КДК} \rightarrow 1$, і чим більше n , тим менше похибка. Для дослідження обрано діапазон зміни від 0.99 до 0.90, що еквівалентно кількості циклів від 10 до 100.

Ймовірність виконання переходу командою розгалуження по визначенню ймовірності дорівнює

$$q = \frac{N_{\text{виконаних_переходів}}}{N_{\text{переходів}}},$$

де $N_{\text{виконаних_переходів}}$ - кількість виконаних переходів в початок циклу;

$N_{\text{переходів}}$ - загальна кількість команд переходів в циклі.

Якщо цикл виконується n раз, то

$$q = \frac{(n-1)}{n}.$$

При досить великих значеннях n $q \rightarrow 1$, і чим більше n , тим менше похибка.

Так як

$$P_{КДК} = \frac{n-1}{n} \quad \text{и} \quad q = \frac{(n-1)}{n},$$

то для дослідження обрані діапазони зміни $P_{КДК}$ и q від 0.99 до 0.90, що еквівалентно кількості циклів від 10 до 100.

Діапазон зміни змінної A (це час перезавантаження конвеєра генерації команд в результаті виконання команди переходу) - від 6 до 4 тактів в залежності від структурних особливостей конвеєра генерації команд.

Діапазон зміни змінної B (це час генерації команд дешифратором при відсутності кеш-пам'яті роздешифрованих команд або відсутності роздешифрованої команди в кеш-пам'яті роздешифрованих команд) обраний 1 або 2 такти в залежності від структурних особливостей конвеєра генерації команд.

Діапазон зміни змінної C - 1 такт.

Діапазон зміни коефіцієнта суперскалярності $S = 1, 2, 3, 4$.

Діапазон зміни змінної D – 1, 2 такти.

4.5 Порівняння варіантів

Для порівняння різних варіантів структур підсистеми генерації команд використано відношення їх продуктивностей

$$R_{ij} = \frac{PP_i}{PP_j}.$$

4.5.1 Порівняння варіантів 2 і 1

Відношення продуктивностей варіантів структури з кеш-пам'яттю роздешифрованих команд і без неї при відсутності суперскалярності дає формулу

$$R_{21} = \frac{PP_2}{PP_1} = \frac{B + \omega_{\Pi} * q * A}{P_{КДК} * C + (1 - P_{КДК}) * (B + \omega_{\Pi} * q * A)}.$$

Після перетворень

$$R_{21} = \frac{B / C + \omega_{\Pi} * q * A / C}{P_{КДК} + (1 - P_{КДК}) * (B / C + \omega_{\Pi} * q * A / C)}.$$

Точні і наближені значення R_{21} в залежності від значень змінних B / C ; A / C ; $P_{КДК}$; ω_{Π} ; q наведені в таблиці 4.2.

Таблиця 4.2 - Залежності R_{21} від параметрів B/C ; A/C ; $P_{КДК}$; ω_{II} ; q

ω_{II}	q	$P_{КДК}$	A/C	B/C	Точне R_{21}	Наближене R_{21}	Похибка (%)
0.05	0.99	0.99	6	1	1.303	1.297	0.47
	0.95	0.95	6	1	1.328	1.285	3.24
	0.90	0.90	6	1	1.340	1.270	5.22
	0.99	0.99	4	1	1.196	1.198	0.17
	0.95	0.95	4	1	1.186	1.190	3.37
	0.90	0.90	4	1	1.175	1.180	4.26
0.10	0.99	0.99	6	1	1.585	1.594	0.57
	0.95	0.95	6	1	1.526	1.570	2.88
	0.90	0.90	6	1	1.461	1.540	5.41
	0.99	0.99	4	2	2.364	2.396	1.35
	0.95	0.95	4	2	2.226	2.380	6.92
	0.90	0.90	4	2	2.078	2.360	13.58
0.20	0.99	0.99	6	1	2.162	2.188	1.20
	0.95	0.95	6	1	2.125	2.140	7.06
	0.90	0.90	6	1	1.877	2.080	10.81
	0.99	0.99	4	2	2.743	2.792	1.79
	0.95	0.95	4	2	2.537	2.760	9.18
	0.90	0.90	4	2	2.321	2.720	17.2

Висновки

1. $R_{21} > 1$ при будь-яких значеннях змінних. Це означає, що варіант структури без суперскалярності, але з кеш-пам'яттю роздешифрованих команд більш продуктивний, що підтверджує інтуїтивні припущення про те, що ускладнення структури має забезпечувати збільшення продуктивності.

2. Для порівняння варіантів 2 і 1 для варіантів програм з $P_{КДК}$ близьким до 1 в першому наближенні можна використовувати наближене вираження $R_{21} \approx B/C + \omega_{\Pi} * q * A/C$.

3. Значення показника ефективності варіанту 2 в порівнянні з варіантом 1 залежить головним чином від співвідношення змінних B і C . Змінні A , ω_{Π} , q впливають значно менше.

4.5.2 Порівняння варіантів 3 та 1

Якщо порівняти продуктивності варіантів структури з і без суперскалярного, але без кеш-пам'яті роздешифрованих команд, отримаємо формулу

$$R_{31} = \frac{ПП_3}{ПП_1} = \frac{B + \omega_{\Pi} * q * A}{B/S + \omega_{\Pi} * q * A} = \frac{1 + \omega_{\Pi} * q * A/B}{1/S + \omega_{\Pi} * q * A/B}$$

Граничне значення показника ефективності R_{31} при прагненні показника суперскалярності S до ∞ дорівнює

$$\lim_{S \rightarrow \infty} R_{31} = \frac{B + \omega_{\Pi} * q * A}{B/S + \omega_{\Pi} * q * A} = 1 + \frac{1}{\omega_{\Pi} * q * A/B}$$

Значення R_{31} , які залежать від змінних A/B ; ω_{Π} ; q ; S , наведені в таблиці 4.3.

Таблиця 4.3 - Залежності коефіцієнта R_{31} від параметрів A/B ; ω_{Π} ; q ; S .

ω_{Π}	q	S	A/B	R_{31}
0.05	0.99	2	6	1.627
		3	6	2.058
		4	6	2.371
		2	4	1.716
		3	4	2.255
		4	4	2.674

Висновки

1. Так як $S > 1$, то чисельник R_{31} завжди більше 1. Це означає, що реалізація суперскалярності навіть в мінімальному вигляді збільшує теоретичну продуктивність підсистеми генерації команд.

2. Збільшення коефіцієнта суперскалярності незначно збільшує теоретичну продуктивність підсистеми генерації команд без кеш-пам'яті роздешифрованих команд.

3. Граничне значення показника ефективності R_{31} при прагненні показника суперскалярності S до ∞ визначається в основному значеннями змінних A/B ; ω_{Π} ; q .

$$\lim_{S \rightarrow \infty} R_{31} = 1 + \frac{1}{\omega_{\Pi} * q * A / B}.$$

4.5.3 Порівняння варіантів 3 та 2

Відношення продуктивностей варіантів структури з суперскалярністю і без кеш-пам'яті роздешифрованих команд і без суперскалярності, але з кеш-пам'яттю роздешифрованих команд дає такий вираз

$$R_{32} = \frac{PP_3}{PP_2} = \frac{P_{КДК} * C + (1 - P_{КДК}) * (B + \omega_{\Pi} * q * A)}{B / S + \omega_{\Pi} * q * A}.$$

Після перетворення

$$R_{32} = \frac{P_{КДК} + (1 - P_{КДК}) * (B / C + \omega_{\Pi} * q * A / C)}{B / (C * S) + \omega_{\Pi} * q * A / C}.$$

Для порівняння варіантів 3 та 2 при виконанні програм з $P_{КДК}$ близьким до 1 можна використовувати наближене вираження

$$R_{32} \approx \frac{S}{B / C + S * \omega_{\Pi} * q * A / C}.$$

Висновки

1. Варіант структури з суперскалярністю і без кеш-пам'яті роздешифрованих команд продуктивніший ніж варіанти структури без суперскалярності, але з кеш-пам'яттю роздешифрованих команд.

2. R_{32} монотонно зростаюча функція від мінімального значення, меншого 1 при $S = 1$, до максимального значення при $S = \infty$, рівного

$$\lim_{S \rightarrow \infty} R_{32} = \frac{P_{КДК} + (1 - P_{КДК}) * (B / C + \omega_{\Pi} * q * A / C)}{\omega_{\Pi} * q * A / C}.$$

Максимальне значення R_{32} не залежить від значення коефіцієнта суперскалярного S .

3. При ймовірності $P_{КДК}$ близькою до 1 для порівняння варіантів 3 та 2 можна використовувати наближений вираз

$$R_{32} \approx \frac{S}{B / C + S * \omega_{\Pi} * q * A / C}$$

та його граничне значення

$$\lim_{S \rightarrow \infty} R_{32} = \frac{1}{\omega_{\Pi} * q * A / C}.$$

Наприклад, при $B / C = 1$, $A / C = 6$, $\omega_{\Pi} = 0.05$ і $q = 1$

$$R_{32} = \frac{1}{1 / S + 0.3}$$

А якщо:

$S = 1$ - $R_{32} = 0.769$; $S = 2$ - $R_{32} = 1.250$; $S = 3$ - $R_{32} = 1.579$; $S = 4$ - $R_{32} = 1.818$; $S = \infty$ - $R_{32} = 3.333$.

4. При реалізації варіанту структури з суперскалярністю, але без кеш-пам'яті роздешифрованих команд слід обережно вибирати коефіцієнт суперскалярності.

Збільшення S понад оптимального значення не забезпечує істотного приросту продуктивності.

4.5.4 Порівняння варіантів 4 і 1

Відношення продуктивностей варіантів структури з суперскалярністю і кеш-пам'яттю роздешифрованих команд і без суперскалярності та без кеш-пам'яті роздешифрованих команд дає такий вираз

$$R_{41} = \frac{PP_4}{PP_1} = \frac{S * (B + \omega_{\Pi} * q * A)}{P_{КДК} * C + (1 - P_{КДК}) * B + S * \omega_{\Pi} * q * [(1 - P_{КДК}) * A + P_{КДК} * D]}$$

Після перетворень

$$R_{41} = \frac{S * (B / C + \omega_{\Pi} * q * A / C)}{P_{КДК} * (1 + S * \omega_{\Pi} * q * D / C) + (1 - P_{КДК}) * (B / C + S * \omega_{\Pi} * q * A / C)} .$$

При ймовірності $P_{КДК}$ близькою до 1 (при виконанні програм з великою кількістю циклів) для порівняння варіантів 4 і 1 можна використовувати наближений вираз

$$R_{41} \leq \frac{B / C + \omega_{\Pi} * q * A / C}{1 / S + \omega_{\Pi} * q * D / C} .$$

Граничне значення відношення R_{41} дорівнюватиме

$$\lim_{S \rightarrow \infty} R_{41} = \frac{B / C + \omega_{\Pi} * q * A / C}{\omega_{\Pi} * q * D / C} .$$

У виразі для R_{41} чітко видно вклади в підвищення продуктивності коефіцієнта суперскалярності S (чим більший коефіцієнт S , тим більший коефіцієнт R_{41}) і кеш-пам'яті роздешифрованих команд (чим більше відношення B/C та A/C , тим більше вплив кеш-пам'яті роздешифрованих команд).

Деякі значення R_{41} в залежності від значень змінних наведені в таблиці 4.4.

Таблиця 4.4 - Значення R_{41} в залежності від значень змінних B/C ; A/C ; D/C ; S ; ω_{II} ; q

ω_{II}	q	S	B/C	A/C	D/C	Точне R_{41}	Наближене R_{41}	Похибка R_{41} (%)
0.05	0.99	1	1	6	1	1.233	1.236	0.24
		2	1	6	1	2.350	2.360	0.43
		3	1	6	1	3.366	3.386	0.59
		4	1	6	1	4.294	4.331	0.86
		1	1	4	1	1.140	1.142	0.17
		2	1	4	1	2.174	2.180	0.28
		3	1	4	1	3.117	3.129	0.39
		4	1	4	1	3.980	4.000	0.50

Висновки

1. Реалізація суперскалярності в поєднанні з кеш-пам'яттю дешифрованих команд істотно збільшує теоретичну продуктивність підсистеми генерації команд в порівнянні з найпростішим варіантом структури без суперскалярності та без кеш-пам'яті дешифрованих команд. $R_{41} > 1$ при будь-яких значеннях змінних.

2. Ефективність структури підсистеми підготовки команд з суперскалярністю в поєднанні з кеш-пам'яттю дешифрованих команд монотонно зростає від мінімального значення при $S = 1$ до граничного значення

$$\lim_{S \rightarrow \infty} R_{41} = \frac{B/C + \omega_{II} * q * A/C}{\omega_{II} * q * D/C}$$

Наприклад, при $B/C = 1$, $A/C = 6$, $\omega_{II} = 0.05$ і $q = 1$ $\lim_{S \rightarrow \infty} R_{41} = 26$.

3. Граничне значення показника ефективності R_{41} при прагненні показника суперскалярності S до ∞ визначається в основному значеннями змінних ω_{II} , q , A/D , B/D .

4. При ймовірностях $P_{КДК}$ та q близьких до 1 (особливо при виконанні циклічних ділянок програм з великою кількістю циклів), для порівняння варіантів 4 і 1 можна використовувати наближене вираження

$$R_{41} \leq \frac{B / C + \omega_{\Pi} * q * A / C}{1 / S + \omega_{\Pi} * q * D / C} .$$

4.6 Висновки до розділу 4

1. Варіант структури без суперскалярності, але з кеш-пам'яттю дешифрованих команд більш продуктивний, що підтверджує інтуїтивні припущення про те, що ускладнення структури має забезпечувати збільшення продуктивності.

2. Збільшення коефіцієнта суперскалярності незначно збільшує теоретичну продуктивність підсистеми генерації команд без кеш-пам'яті дешифрованих команд.

3. Варіант структури з суперскалярністю і без кеш-пам'яті дешифрованих команд продуктивніший ніж варіанти структури без суперскалярності, але з кеш-пам'яттю дешифрованих команд.

4. При реалізації варіанту структури з суперскалярністю, але без кеш-пам'яті дешифрованих команд слід обережно вибирати коефіцієнт суперскалярності. Збільшення S понад оптимального значення не забезпечує істотного приросту продуктивності.

5. Реалізація суперскалярності в поєднанні з кеш-пам'яттю дешифрованих команд істотно збільшує теоретичну продуктивність підсистеми генерації команд в порівнянні з найпростішим варіантом структури без суперскалярності та без кеш-пам'яті роздешифрованих команд.

РОЗДІЛ 5

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ. ЕКОЛОГІЯ

В даному розділі проведено аналіз потенційних небезпечних та шкідливих виробничих факторів, причин пожеж. Розглянуті заходи, які дозволяють забезпечити гігієну праці і виробничу санітарію. На підставі аналізу розроблені заходи з техніки безпеки та рекомендації з пожежної профілактики.

Завданням даного проекту бакалавра було «Дослідження ефективності ядер процесорів з спеціалізованими функціональними пристроями». Так як в процесі проектування виконувалось у домашніх умовах, то аналіз потенційно небезпечних і шкідливих виробничих чинників виконується для персонального комп'ютера на якому буде виконуватись дослідження.

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»[0] 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.

5.1.1 Правові та організаційні основи охорони праці

Основним організаційним напрямом у здійсненні управління в сфері охорони праці є усвідомлення пріоритету безпеки праці і підвищення соціальної відповідальності держави, і особистої відповідальності працівників.

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»[0] (далі – Закону) законодавство про охорону праці складається з Закону, Кодексу законів про працю

України, Закону України "Про загальнообов'язкове державне соціальне страхування від нещасного випадку на виробництві та професійного захворювання, які спричинили втрату працездатності" та прийнятих відповідно до них нормативно-правових актів, норм міжнародного договору (ратифіковані Конвенції і Рекомендації МОТ, директиви Європейської Ради).

5.1.2 Організаційно-технічні заходи з безпеки праці

В організації/підприємстві проводиться навчання і перевірка знань з питань охорони праці відповідно до вимог Типового положення про порядок проведення навчання і перевірки знань з питань охорони праці, затвердженого наказом Держнаглядохоронпраці України від 26.01.2005 N 15, зареєстрованого в Міністерстві юстиції України 15.02.2005 за N 231/10511 [0].

Обов'язковими вимогами враховане наступне:

– не слід допускати до роботи осіб, що в установленому порядку не пройшли навчання, інструктаж та перевірку знань з охорони праці, пожежної безпеки та цих Правил.

– на підприємстві/організації, де експлуатуються ЕОМ з відео дисплейними терміналами (ВДТ) і периферійними пристроями (ПП), розробляється інструкція з охорони праці відповідно до Положення про розробку інструкцій з охорони праці, затвердженого наказом Держнаглядохоронпраці від 29.01.98 N 9, зареєстрованого в Міністерстві юстиції України 07.04.98 за N 226/2666 [0].

– ознайомлення з правилами безпеки праці, одержання відповідних інструктажів засвідчується у журналі інструктажів.

5.2 Аналіз стану умов праці

Робота над створенням локальної комп'ютерної мережі проходить в побутовому приміщенні. Для даної роботи достатньо однієї людини, для якої надано робоче місце зі стаціонарним комп'ютером.

5.2.1 Вимоги до приміщення

Геометричні розміри приміщення зазначені у таблиці 5.1.

Таблиця 5.1 – розміри робочого місця

Параметр	Значення
Довжина, м	3
Ширина, м	5
Висота, м	2,5
Площа, м ²	15
Об'єм, м ³	37,5

Згідно до санітарних норм мікроклімату виробничих приміщень [0] 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 робочого місця

При порівнянні відповідності характеристик робочого місця нормативним основні вимоги до організації робочого місця [0] (табл. 5.2) і відповідними фактичними значеннями для робочого місця, констатуємо повну відповідність.

Таблиця 5.2 – Характеристика 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

Продовження таблиці 5.2

Найменування параметра	Фактичне значення	Нормативне значення
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

У кабінеті є електрична мережа з напругою 220 В, яка створює небезпеку ураження електричним струмом. ПК та периферійні пристрої можуть бути джерелами електромагнітних випромінювань, аерозолів та шкідливих речовин (часток тонеру, оксидів нітрогену та озону).

За ступенем пожежної безпеки приміщення належить до категорії В. Кабінет оснащений переносним вуглекислотним вогнегасником ВВК-5 .

Наявна аптечка для надання долікарської допомоги, а також у кабінеті роблять вологе прибирання та щоденно провітрюють приміщення.

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.

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 розробці виробу

Аналіз небезпечних та шкідливих виробничих факторів виконується у табличній формі (табл. 5.3). 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 , 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:

- робоча напруга $U = +220\text{В} \pm 5\%$;
- робочий струм $I = 2\text{А}$;
- споживана потужність $P = 600\text{ Вт}$.

Таблиця 5.3 – Аналіз небезпечних і шкідливих виробничих факторів

Небезпечні і шкідливі виробничі фактори	Джерела факторів (види робіт)	Нормативні документи
Фізичні		
- підвищена температура поверхонь обладнання	експлуатація ЕОМ, серверного обладнання для роботи	[0.]
- підвищена або знижена вологість повітря	-//-	[0]
- підвищена або знижена рухливість повітря	-//-	[0]
- підвищений рівень напруги електричної мережі	-//-	[0] [0]
- підвищений рівень статичної електрики	-//-	[0]
- підвищена напруженість електромагнітного поля	-//-	[0]
Небезпечні і шкідливі виробничі фактори	Джерела факторів (види робіт)	Нормативні документи
- недостатність природного світла	порушення умов праці (вимог до приміщень)	[0]
- недостатнє освітлення робочої зони	порушення гігієнічних параметрів виробничого середовища	[0]
Психофізіологічні		
-нервово-психічна перевантаження	Розумова робота над проектом	[0] [0]
- фізичні (статичне – сидіння)	порушення умов праці та організації робочого часу	[0]

Робочі місця мають 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 [0]. За умов роботи з ПК виникають наступні небезпечні та шкідливі чинники: несприятливі мікрокліматичні умови, освітлення, електромагнітні випромінювання, забруднення повітря шкідливими речовинами (джерелом, яких можуть бути: принтер, сканер та інші джерела виділення багатьох хімічних речовин - напр., озону, оксидів азоту та аерозолів високодисперсних частинок тонера), шум, вібрація, електричний струм, електростатичне поле, напруженість трудового процесу та інше.

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.

5.3.3 Електробезпека

На робочому місці виконуються наступні вимоги електробезпеки: ПК, периферійні пристрої та устаткування для обслуговування, електропроводи і кабелі за виконанням та ступенем захисту відповідають класу зони за ПУЕ (правила улаштування електроустановок), мають апаратуру захисту від струму короткого замикання та інших аварійних режимів. Лінія електромережі для живлення ПК, периферійних пристроїв і устаткування для обслуговування, виконана як окрема групова три- провідна мережа, шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Штепсельні з'єднання та електророзетки крім контактів фазового та

нульового робочого провідників мають спеціальні контакти для підключення нульового захисного провідника. Електромережа штепсельних розеток для живлення персональних ПК, укладено по підлозі поруч зі стінами відповідно до затвердженого плану розміщення обладнання та технічних характеристик обладнання. Металеві труби та гнучкі металеві рукави заземлені. Захисне заземлення включає в себе заземлюючих пристроїв і провідник, який з'єднує заземлюючий пристрій з обладнанням, яке заземлюється - заземлюючий провідник.

5.4 Гігієнічні вимоги до параметрів виробничого середовища

5.4.1 Мікроклімат

Мікроклімат робочих приміщень – це клімат внутрішнього середовища цих приміщень, що визначається діючої на організм людини з'єднанням температури, вологості, швидкості переміщення повітря. В даному приміщенні проводяться роботи, що виконуються сидячи і не потребують динамічного фізичного напруження, то для нього відповідає категорія робіт Іа. Отже оптимальні значення для температури, відносної вологості й рухливості повітря для зазначеного робочого місця відповідають нормам [0] і наведені в табл. 5.4.

Таблиця 5.4 – Норми мікроклімату робочої зони об'єкту

Період року	Категорія робіт	Температура С°	Відносна вологість %	Швидкість руху повітря, м/с
Холодна	легка-1 а	22 - 24	40 – 60	0,1
Тепла	легка-1 а	23 - 25	40 – 60	0,1

Дане приміщення обладнане системами опалення, кондиціонування повітря або припливно-витяжною вентиляцією. У приміщенні на робочому місці забезпечуються оптимальні значення параметрів мікроклімату: температури, відносної вологості й рухливості повітря у відповідності до [0]. Рівні позитивних і негативних іонів у повітрі мають відповідати [0]. Для забезпечення оптимальних параметрів мікроклімату в приміщенні проводяться перерви в роботі співробітників, з метою його провітрювання.

Існують спеціальні системи кондиціонування, які забезпечують підтримання в приміщенні балансу оптимальних параметрів мікроклімату.

Контроль параметрів мікроклімату в холодний і теплий період року здійснюється не менше 3-х разів на зміну (на початку, середині, в кінці).

5.4.2 Освітлення

У проекті, що розробляється, передбачається використовувати суміщене освітлення. У світлий час доби використовуватиметься природне освітлення приміщення через віконні отвори, в решту часу використовуватиметься штучне освітлення. Штучне освітлення створюється газорозрядними лампами.

5.4.2.1 Розрахунок освітлення

Для виробничих та адміністративних приміщень світловий коефіцієнт приймається не менше $1/8$, в побутових – $1/10$:

$$S_b = \left(\frac{1}{5} \div \frac{1}{10} \right) \cdot S_n \quad (5.1)$$

де S_b – площа віконних прорізів, m^2 ;

S_n – площа підлоги, m^2 .

$$S_n = a \cdot b = 5 \cdot 3 = 15 \text{ м}^2,$$

$$S = 1/10 \cdot 15 = 1,5 \text{ м}^2.$$

Приймаємо 1 вікно площею $S=1,5 \text{ м}^2$.

Світильники загального освітлення розташовуються над робочими поверхнями в рівномірно-прямокутному порядку. Для організації освітлення в темний час доби передбачається обладнати приміщення, довжина якого складає 5 м, ширина 5 м, світильниками ЛПО2П, оснащеними лампами типа ЛБ (дві по 80 Вт) з світловим потоком 3200 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 (5.2):

$$n = \frac{E \cdot S \cdot Z \cdot K}{F \cdot U \cdot M}, \quad (5.2)$$

де E – нормована освітленість робочої поверхні, визначається нормами – 300 лк;

S – освітлювана площа, m^2 ; $S = 15 m^2$;

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 – 3200лм (для ЛБ-40-2).

Підставивши числові значення у формулу (5.2), отримуємо:

$$n = \frac{300 \cdot 15 \cdot 1,1 \cdot 1,5}{3200 \cdot 0,575 \cdot 2} = 2,018$$

Приймаємо освітлювальну установку, яка складається з 2-х світильників, які складаються з 2-х люмінесцентних ламп загальною потужністю 40 Вт, напругою – 220 В.

5.4.3 Вентилювання

У приміщенні, де знаходяться ЕОМ, повітрообмін реалізується за допомогою природної організованої вентиляції. Цей метод забезпечує приток потрібної кількості свіжого повітря, що визначається в СНІП.

Також має здійснюватися провітрювання приміщення, в залежності від погодних умов, тривалість повинна бути не менше 10 хв. Найкращий обмін повітря здійснюється при наскрізному провітрюванні.

5.5 Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій

Відповідно до санітарно-гігієнічних нормативів та правил експлуатації обладнання наводимо приклади деяких заходів безпеки.

1) Заходи безпеки під час експлуатації персонального комп'ютера та периферійних пристроїв передбачають:

- правильне організування місця праці та дотримання оптимальних режимів праці та відпочинку під час роботи з ПК;

- експлуатацію сертифікованого обладнання;

- дотримання заходів електробезпеки;

- забезпечення оптимальних параметрів мікроклімату;

- забезпечення раціонального освітлення місця праці (освітленість робочого місця не перевищувала $2/3$ нормальної освітленості приміщення);

- облаштовуючи приміщення для роботи з ПК, потрібно передбачити припливно-витяжну вентиляцію або кондиціювання повітря:

- а) якщо об'єм приміщення 20 м^3 , то потрібно подати не менш як $30 \text{ м}^3/\text{год}$ повітря;

- б) якщо об'єм приміщення у межах від 20 до 40 м^3 , то потрібно подати не менш як $20 \text{ м}^3/\text{год}$ повітря;

- в) якщо об'єм приміщення становить понад 40 м^3 , допускається природна вентиляція, у випадку, коли немає виділення шкідливих речовин.

- зниження рівня шуму та вібрації:

- а) у джерелі виникнення, шляхом застосування раціональних конструкцій, нових матеріалів і технологічних процесів;

- б) звукоізолювання устаткування за допомогою глушників, резонаторів, кожухів, захисних конструкцій, оздоблення стін, стелі, підлоги тощо;

- в) використання засобів індивідуального захисту).

2) Заходи безпеки під час експлуатації інших електричних приладів передбачають дотримання таких правил:

- постійно стежити за справним станом електромережі, розподільних щитків, вимикачів, штепсельних розеток, лампових патронів, а також мережевих кабелів живлення, за допомогою яких електроприлади під'єднують до електромережі;

- постійно стежити за справністю ізоляції електромережі та мережевих кабелів, не допускаючи їхньої експлуатації з пошкодженою ізоляцією;

- не тягнути за мережевий кабель, щоб витягти вилку з розетки;

- не закривати меблями, різноманітним інвентарем вимикачі, штепсельні розетки;

- не підключати одночасно декілька потужних електропристроїв до однієї розетки, що може викликати надмірне нагрівання провідників, руйнування їхньої ізоляції, розплавлення і загоряння полімерних матеріалів;

- не залишати включені електроприлади без нагляду;

- не допускати потрапляння всередину електроприладів крізь вентиляційні отвори рідин або металевих предметів, а також не закривати їх та підтримувати в належній чистоті, щоб уникнути перегрівання та займання приладу;

- не ставити на електроприлади матеріали, які можуть під дією теплоти, що виділяється, загорітися (канцелярські товари, сувенірну продукцію тощо).

Від ураження струмом застосовують різні електричні захисні засоби:

а) Ізолюючі – ізолюють людини від струмоведучих або заземлених частин, а також від землі. Вони діляться на основні та додаткові.

б) Основні – володіють ізоляцією, здатної довго витримувати робоче напругу електроустановки і тому ними дозволяється стосуватися струмоведучих частин, знаходячи-трудящих під напругою.

в) Запобіжні – володіють ізоляцією нездатною витримати робоча напруга електроустановки, і тому вони не можуть самостійно захищати людину від ураження струмом під цим напругою. Їх значення - посилити захисні дії основних і ізолюючих засобів, разом з якими вони повинні застосовуватися, при чому при використанні основних захисних засобів достатньо застосування одного запобіжного захисного засобу.

5.5.1 Розрахунок захисного заземлення

Згідно з класифікацією приміщень за ступенем небезпеки ураження електричним струмом № [0], приміщення в якому проводяться всі роботи відноситься 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.

5.5.1.1 Послідовність розрахунку:

1) Визначається необхідний опір штучних заземлювачів $R_{шт.з.}$:

$$R_{шт.з.} = \frac{R_d \cdot R_{пр.з.}}{R_{пр.з.} - R_d}; \quad (5.3)$$

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 (5.3), 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 ρ , 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:

$$\rho_{\text{розр.}} = \psi \cdot \rho \quad (5.4)$$

де ψ – 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_B , Ом, за (5.).

$$R_B = \frac{\rho_{\text{розр.в}}}{2 \cdot \pi \cdot l_B} \cdot \left(\ln \frac{2 \cdot l_B}{d_{\text{ст}}} + \frac{1}{2} \cdot \ln \frac{4 \cdot t + l_B}{4 \cdot t - l_B} \right), \quad (5.5)$$

де l_B – довжина вертикального заземлювача (для труб – 2 – 3 м; $l_B = 3$ 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. (5.6):

$$t = h_E + \frac{l_E}{2}, \quad (5.6)$$

де h_E – глибина закладання вертикальних заземлювачів (0,8 м); тоді

$$t = 0,8 + \frac{3}{2} = 2,3 \text{ м};$$

$$R_B = \frac{68}{2 \cdot \pi \cdot 3} \cdot \left(\ln \frac{2 \cdot 3}{0,05} + \frac{1}{2} \cdot \ln \frac{4 \cdot 2,3 + 3}{4 \cdot 2,3 - 3} \right) = 18,5 \text{ Ом}$$

1) Визначається теоретична кількість вертикальних заземлювачів n штук, без урахування коефіцієнта використання η_B :

$$n = \frac{2R_E}{R_D} = \frac{2 \cdot 18,5}{4} = 9,25, \quad (5.7)$$

І визначається коефіцієнт використання вертикальних електродів групового заземлювача без врахування впливу з'єднувальної стрічки $\eta_B = 0,57$ (табличне значення).

- 2) Визначається необхідна кількість вертикальних заземлювачів з урахуванням коефіцієнта використання η_B , шт:

$$n = \frac{2 \cdot R_E}{R_D \cdot \eta_E} = \frac{2 \cdot 18,5}{4 \cdot 0,57} \approx 16, \quad (5.8)$$

- 3) Визначається довжина з'єднувальної стрічки горизонтального заземлювача l_C , м:

$$l_C = 1,05 \cdot L_B \cdot (n_B - 1), \quad (5.9)$$

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_T , Ом:

$$R_T = \frac{\rho_{\text{розр.г}}}{2 \cdot \pi \cdot l_C} \cdot \ln \frac{2 \cdot l_C^2}{d_{\text{см}} \cdot h_T}, \quad (5.10)$$

де $d_{\text{см}}$ – еквівалентний діаметр смуги шириною b , $d_{\text{см}} = 0,95b$, $b = 0,15$ м;

h_T – глибина закладання горизонтальних заземлювачів (0,5 м);

l_C – довжина з'єднувальної стрічки горизонтального заземлювача l_C , м

$$R_T = \frac{220}{2 \cdot \pi \cdot 48} \cdot \ln \frac{2 \cdot 48^2}{0,95 \cdot 0,15 \cdot 0,5} = 8,1 \text{ Ом}$$

- 4) Визначається коефіцієнт використання горизонтального заземлювача η_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:

$$R_{\text{заг.}} = \frac{R_E \cdot R_T}{R_E \cdot \eta_C + R_T \cdot n_E \cdot \eta_E} \leq R_D, \quad (5.11)$$

Висновок: дане захисне заземлення буде забезпечувати електробезпеку будівлі, так як виконується умова: $R_{\text{заг.}} < 4$ Ом, а саме:

$$R_{\text{заг.}} = \frac{18,5 \cdot 8,1}{18,5 \cdot 0,3 + 8,1 \cdot 16 \cdot 0,57} = 1,9 \leq R_D$$

При виникненню пожеж при роботі на ПЕОМ від таких можливими джерел запалювання як:

- іскри і дуги коротких замикань;
- перегрів провідників, резисторів та інших радіодеталей ПЕОМ, від тривалої перевантаження та наявності перехідного опору;
- іскри при розмиканні і розмиканні ланцюгів;
- розряди статичної електрики;
- необережному поводженню з вогнем, а також вибухи газо-повітряних і паро-повітряних сумішей.

5.6 Екологія та охорона навколишнього середовища

Діяльність за темою магістерської роботи, а саме: Дослідження ефективності ядер процесорів з спеціалізованими функціональними пристроями в процесі її виконання впливає на навколишнє природне середовище і регламентується нормами діючого законодавства: Законом України «Про охорону навколишнього природного середовища», Законом України «Про забезпечення санітарного та епідемічного благополуччя населення», Законом України «Про відходи», Законом України «Про охорону атмосферного повітря», Законом України «Про захист населення і територій від надзвичайних ситуацій техногенного та природного характеру», Водний кодекс України.

В процесі діяльності дослідження за допомогою ПК виникають процеси поводження з відходами ІТ галузі. Нижче надано перелік відходів, що утворюються в процесі роботи:

- Відпрацьовані люмінесцентні лампи - I клас безпеки
- Акумулятор для джерел безперебійного живлення – III клас безпеки
- Змінні носії інформації - IV клас безпеки
- Макулатура - IV клас безпеки
- Побутові відходи - IV клас безпеки

Зберігання відходів та їх утилізація виконується згідно до вимог Державних санітарних правил і норм ДСанПіН 2.2.7.029 [0].

5.7 Висновки до розділу 5

В результаті проведеної роботи було 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. Були наведені розміри приміщення та наведено значення температури, вологості й рухливості повітря, необхідна кількість і потужність ламп та інші параметри, значення яких впливає на умови праці робітника, а також – наведені інструкції з охорони праці, техніки безпеки при роботі на комп'ютері.

ЗАКЛЮЧЕННЯ

В розділі 1 розглянуті структури ядер сучасних процесорів та методи досягнення їх високої продуктивності.

В розділі 2 розглянуті перешкоди в вигляді команд переходів та їх вплив на продуктивність ядра. Розроблена модель та отримані аналітичні результати залежності ефективності ядра процесора від різних параметрів як структури, так і програм.

В розділі 3 розроблена модель та отримані аналітичні результати залежності ефективності підсистеми підготовки команд ядра процесора як від різних параметрів підсистеми підготовки команд, так і програм.

В розділі 4:

розроблена модель функціонування ядра при використанні різних структурних методів компенсації втрат продуктивності;

отримані аналітичні залежності показників ефективності від різних параметрів наступних структур ядра: структура без блоку передбачення; структура з блоком передбачень та без буфера дешифрованих команд; структура з блоком передбачень та з буфером дешифрованих команд.

проведено порівняння ефективності різних структурних рішень по підвищенню продуктивності підсистеми підготовки ядра.

Результати дослідження можуть бути корисні при оптимальному виборі структури ядра з урахуванням співвідношення як показників ефективності, так і апаратних витрат на реалізацію.

В розділі 5 розглянуті питання охорони праці та основні напрямки їх дотримання.

ЛІТЕРАТУРА

1. M. Milenkovic, A. Milenkovic, J. Kulick. Demystifying Intel Branch Predictors. Proceedings of the Workshop on Duplicating, Deconstructing and Debunking, 2002.
2. IA-32 Intel Architecture Optimization Reference Manual. Intel, 2006.
3. С. Гарматюк. Современные десктопные процессоры архитектуры IA-32/64: общие принципы работы. iXBT.com, 2006.
4. В. Inkley. Inside the Intel Core Microarchitecture. Intel Developer Forum, 2006.
5. Михаил Кузьминский. Nehalem: структура и производительность. Открытые системы, №8, 2009.
6. Недзельский Д.А. Исследование и анализ эффективности структурных методов компенсации влияния команд переходов на производительность конвейерных ядер процессоров. Луганськ: Вісник Східноукраїнського національного університету ім. В. Даля, 2013. - №16 (205), Частина 2. - С.174-181.
7. Недзельский Д.А. Исследование эффективности подсистемы генерации команд в ядрах современных процессоров. Луганськ: Вісник Східноукраїнського національного університету ім. В. Даля, 2017. - №8 (238), - С.64-66.
8. Недзельський Д.О., Костиця Р.Г. Дослідження ефективності суперскалярності та кеш-пам'яті продешифрованих команд сучасних процесорів. Сєверодонецьк: Вісник Східноукраїнського національного університету ім. В. Даля, 2018. - №6 (247), - С.110-117.
9. Орлов С.А., Цилькер Б.Я. Организация ЭВМ и систем. 2-е изд. – СПб.: Питер, 2011. – 688 с.
10. Брайант Р., О'Халларон Д. Компьютерные системы: архитектура и программирование. Пер. с англ. – СПб.: БХВ-Петербург, 2005. – 1104 с.: ил.
11. Клейнрок Л. Вычислительные системы с очередями. -М.: Мир, 1979. - 600с.
12. В. Inkley. Inside the Intel Core Microarchitecture. Intel Developer Forum, 2006.
13. Закон України «Про охорону праці».
14. НПАОП 0.00.-1.28-10 «Правил охорони праці під час експлуатації електронно-обчислювальних машин».
15. НПАОП 0.00-4.12-05 «Типове положення про порядок проведення навчання і перевірки знань з питань охорони праці».
16. НПАОП 0.00-4.15-98 «Положення про розробку інструкцій з охорони праці».
17. НПАОП 40.1-1.01-97 «Правила безпечної експлуатації електроустановок».

18. НАПБ Б.02.005-2003 Типове положення про інструктажі, спеціальне навчання та перевірку знань з питань пожежної безпеки на підприємствах, в установах та організаціях України.
19. ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень» .
ДСанПіН 3.3.2.007-98 «Правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин».
20. ДСанПіН 2.2.7.029 Гігієнічні вимоги щодо поводження з промисловими відходами та визначення їх класу небезпеки для здоров'я населення
- 21.ГОСТ 12.1.044-89 «ССБТ. Вогнестійкість. Номенклатура показників і методи їх визначення».
22. ГОСТ 12.1.030-81 «Електробезпека. Захисне заземлення, занулення».
23. ГОСТ 12.1.006-84 «ССБТ. Електромагнітні поля радіочастот».
24. ГОСТ 13109-97 «Електрична енергія. Сумісність технічних засобів. Норми якості електричної енергії в системах електропостачання загального призначення».
25. ДБН В.2.5-28:2015 «Державні Будівельні Норми України. Природне і штучне освітлення».
26. НАПБ Б.03.002-2007 «Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною безпекою».
27. НАПБ Б.03.002-2007 «Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною безпекою».

Додаток А

Код програми:

```
#include <iostream>
# Include < math . h >
# Include < stdlib . h >
# Include < time . h >
using namespace std ;
int main ()
{
    int i, j, n, m, h = 0, z = 0, s = 0, r = 0, e;
    long double s1 = 0, s2 = 0, A1, A2, Wn, Wo, P, Q, w;
    // Введення ключових змінних і створення масиву
    cout << "A1:";
    cin >> A1;
    cout << "A2:";
    cin >> A2;
    cout << "Wn:";
    cin >> Wn;
    cout << "P:";
    cin >> P;
    cout << "Q:";
    cin >> Q;
    cout << "Розмір буфера:";
    cin >> n;
    Wo = 1-Wn;
    m = n;
    n = (n + 2) + ((n + 1) * 2);
    e = n + 1;
    float ** matrix = new float * [n];
    for (i = 0; i <n; i ++ )
    matrix [i] = new float [e];
    // Заповнення матриці
    for (i = 0; i <n; i ++ ) {
        for (j = 0; j <e; j ++ ) {
```

```

matrix [i] [j] = 0;
}
}

// Заповнити матрицю правильно
for (i = 0; i <n; i++)
{
    for (j = 0; j <e; j++)
    {
        if (i == (n-1)) {
            matrix [i] [j] = 1;
        } Else {
            if (i == 0) {
                // Заповнення рядка A0
                matrix [0] [0] = P;
                matrix [0] [1] = -1;
                matrix [0] [34] = -A2;
                matrix [0] [67] = -A1;
            }
        }
        if (i > 0 && i <33) {
            // Заповнення сегмента Ai
            if (i == (h + 1)) {
                h = i;
                matrix [h] [z] = (-Wo * P);
                matrix [h] [z + 1] = 1 + P;
                matrix [h] [z + 2] = -1;
                matrix [h] [z + 35] = -A2;
                matrix [h] [z + 67] = -A1;
                z++;
            }
        } Else if (i == 33) {
            // Кінцеве заповнення An + 1
            z = 0;
            matrix [i] [z + 32] = -Wo * P;
            matrix [i] [z + 33] = 1;
        }
    }
}

```

```

} Else if (i == 34) {
    // Заповнення рядка B0
    matrix [34] [0] = - ((1-Q) * Wn * P);
    matrix [34] [35] = A2;
    matrix [34] [36] = -1;
} Else if (i > 34 && i < 66) {
    // Заповнення сегмента Bi
    if (i > h) {
        h = i;
        matrix [h] [s + 1] = - ((1-Q) * Wn * P);
        matrix [h] [s + 35] = 1;
        matrix [h] [s + 36] = -1;
        s ++;
    }
} Else if (i == 66) {
    // Кінцеве заповнення Bn
    s = 0;
    matrix [i] [s + 32] = - ((1-Q) * Wn * P);
    matrix [i] [s + 66] = 1;
} Else if (i == 67) {
    // Заповнення рядка C0
    matrix [67] [0] = - (Q * Wn * P);
    matrix [67] [68] = -1;
    matrix [67] [67] = A1;
} Else if (i > 67 && i < n) {
    // Заповнення сегмента Ci
    if (i > h) {
        h = i;
        matrix [h] [r + 1] = - (Q * Wn * P);
        matrix [h] [r + 69] = -1;
        matrix [h] [r + 68] = 1 + A1;
        r ++;
    }
}
}
}
}

```

```

        }
// Вивід матриці
cout << "Вихідний масив:" << endl;
    for (i = 0; i <n; i++)
    {
    for (j = 0; j <e; j++)
    cout << matrix [i] [j] << "";
    cout << endl;
    }
cout << endl;
// Метод Гаусса
// Прямий хід, приведення до верхнетреугольному виду
    float tmp;
    int k;
        float * xx = new float [e];
    for (i = 0; i <n; i++)
    {
    tmp = matrix [i] [i];
    for (j = n; j >= i; j--)
    matrix [i] [j] /= tmp;
    for (j = i + 1; j <n; j++)
    {
    tmp = matrix [j] [i];
    for (k = n; k >= i; k--)
    matrix [j] [k] -= tmp * matrix [i] [k];
    }
    }
//Зворотній хід
xx [n-1] = matrix [n-1] [n];
    for (i = n-2; i >= 0; i--)
    {
    xx [i] = matrix [i] [n];
    for (j = i + 1; j <n; j++) xx [i] -= matrix [i] [j] * xx [j];
    }
// Виводимо результати

```

```
for (i = 0; i < n; i++)
{
if (i == 0 || i == 34 || i == 67) {
    s1 = s1 + xx [i];
    w = 1-s1;
}
if (i <= m) {
    s2 = s2 + xx [i];
}
}
cout << "\n Коефіцієнт завантаження ФП :" << w;
cout << "\n Коефіцієнт завантаження ПК :" << s2;
cout << endl;
delete [] matrix;
system ( "pause");
return 0;
}
```


Міністерство освіти і науки, молоді та спорту
України. Технологічний інститут СНУ ім. В.Даля (м
Сєвєродонецьк)

Моделі і методи дослідження ефективності структурних методів підвищення продуктивності підсистеми генерації команд у ядрах сучасних процесорів

Студент гр. КН-18дм

Костыря Р.Г.

Керівник проекту

Недзельский Д.О.

Припущення зроблені при розробці моделі:

перший етап (підсистема підготовки команд) генерує потік команд (заявок) з інтенсивністю λ . На її вхід по конвеєру надходить нескінченна послідовність команд програми наступних типів:

команди обробки і команди звернення до підсистеми пам'яті - з ймовірністю ω_0 ;

команди умовного переходу - з ймовірністю ω_n ($\omega_n + \omega_0 = 1$);

команди обробки, команди звернення до підсистеми пам'яті виконуються послідовно в другому ступені (в підсистемі виконання, яка представлена як універсальний функціональний пристрій - ФУ);

між етапами є буфер на n заявок;

потік заявок, що згенерував дешифратор, найпростіший з експоненціальним законом розподілу. Якщо буфер заявок заповнений, то дешифратор блокується, тобто зупиняється.

Структура моделі ядра з урахуванням припущень



3

Досліджувані варіанти структур підсистеми генерації команд

Варіант 1. Підсистема генерації команд без суперскалярності та без кеш-пам'яті розшифрованих команд ($S=1$, без КДК).

Це найпростіший варіант структури підсистеми генерації команд. У підсистемі генерації команд є тільки один дешифратор команд. Чергова команда читається з підсистеми пам'яті (як правило з кешу команд L1K). Пролешифрована команда заноситься в буфер підсистеми виконання команд.

Варіант 2. Підсистема генерації команд без суперскалярності, але з кеш-пам'яттю розшифрованих команд ($S=1$, є КДК).

В ядрі процесора є кеш-пам'ять розшифрованих команд і тільки один дешифратор команд. Чергова порція командної інформації читається з підсистеми пам'яті (як правило, з кешу команд L1K). Пролешифрована команда поміщається в КДК (кеш розшифрованих команд).

Варіант 3. Підсистема генерації команд з суперскалярністю, але без кеш-пам'яті розшифрованих команд ($S>1$, немає КДК).

В ядрі процесора немає кеша розшифрованих команд і є S дешифраторів команд. Чергова порція командної інформації (пакет) читається з підсистеми пам'яті (як правило, з кешу команд L1K) і потім дешифрується необхідною кількістю дешифраторів, і пакет готових команд заноситься в буфер готових до виконання команд.

Варіант 4. Підсистема генерації команд з суперскалярністю і з кеш-пам'яттю розшифрувати команд ($S>1$, є КДК).

Це найбільш складний варіант структури ядра. В ядрі процесора є кеш-пам'ять розшифрованих команд і S дешифраторів команд. Чергова порція командною інформації (пакет) читається з підсистеми пам'яті (як правило з кешу команд L1K) і потім дешифрується $S>1$ дешифраторами. Пролешифований пакет команд заноситься в КДК.

4

Порівняння варіантів 2 і 1

ω_{Π}	q	$P_{КДК}$	A/C	B/C	Точне R_{21}	Наближене R_{21}	Похибка (%)
0.05	0.99	0.99	6	1	1.303	1.297	0.47
	0.95	0.95	6	1	1.328	1.285	3.24
	0.90	0.90	6	1	1.340	1.270	5.22
	0.99	0.99	4	1	1.196	1.198	0.17
	0.95	0.95	4	1	1.186	1.190	3.37
	0.90	0.90	4	1	1.175	1.180	4.26
0.10	0.99	0.99	6	1	1.585	1.594	0.57
	0.95	0.95	6	1	1.526	1.570	2.88
	0.90	0.90	6	1	1.461	1.540	5.41
	0.99	0.99	4	2	2.364	2.396	1.35
	0.95	0.95	4	2	2.226	2.380	6.92
	0.90	0.90	4	2	2.078	2.360	13.58
0.20	0.99	0.99	6	1	2.162	2.188	1.20
	0.95	0.95	6	1	2.125	2.140	7.06
	0.90	0.90	6	1	1.877	2.080	10.81
	0.99	0.99	4	2	2.743	2.792	1.79
	0.95	0.95	4	2	2.537	2.760	9.18
	0.90	0.90	4	2	2.321	2.720	17.2

5

Порівняння варіантів 2 і 1

$R_{21} > 1$ при будь-яких значеннях змінних. Це означає, що варіант структури без суперскалярності, але з кеш-пам'яттю роздешифрованих команд більш продуктивний, що підтверджує інтуїтивні припущення про те, що ускладнення структури має забезпечувати збільшення продуктивності.

Для порівняння варіантів 2 і 1 для варіантів програм з $P_{КДК}$ близьким до 1 в першому наближенні можна використовувати наближене вираження $R_{21} \approx B/C + \omega_{\Pi} * q * A/C$.

Значення показника ефективності варіанту 2 в порівнянні з варіантом 1 залежить головним чином від співвідношення змінних B і C . Змінні A , ω_{Π} , q впливають значно менше.

6

Порівняння варіантів 3 та 1

ω_{Π}	q	S	A/B	R_{31}
0.05	0.99	2	6	1.627
		3	6	2.058
		4	6	2.371
		2	4	1.716
		3	4	2.255
		4	4	2.674

Так як $S > 1$, то чисельник R_{31} завжди більше 1. Це означає, що реалізація суперскалярності навіть в мінімальному вигляді збільшує теоретичну продуктивність підсистеми генерації команд.

Збільшення коефіцієнта суперскалярності незначно збільшує теоретичну продуктивність підсистеми генерації команд без кеш-пам'яті роздешифрованих команд.

Граничне значення показника ефективності R_{31} при прагненні показника суперскалярності S до ∞ визначається в основному значеннями змінних A/B ; ω_{Π} ; q .

7

Порівняння варіантів 3 та 2

Відношення продуктивностей варіантів структури з суперскалярністю і без кеш-пам'яті роздешифрованих команд і без суперскалярності, але з кеш-пам'яттю роздешифрованих команд дає такий вираз

$$R_{32} = \frac{PP_3}{PP_2} = \frac{P_{КДК} * C + (1 - P_{КДК}) * (B + \omega_{\Pi} * q * A)}{B / S + \omega_{\Pi} * q * A}$$

Після перетворення

$$R_{32} = \frac{P_{КДК} + (1 - P_{КДК}) * (B / C + \omega_{\Pi} * q * A / C)}{B / (C * S) + \omega_{\Pi} * q * A / C}$$

Для порівняння варіантів 3 та 2 при виконанні програм з $P_{КДК}$ близьким до 1 можна використовувати наближене вираження

$$R_{32} \approx \frac{S}{B / C + S * \omega_{\Pi} * q * A / C}$$

8

Порівняння варіантів 3 та 2

Варіант структури з суперскалярністю і без кеш-пам'яті роздешифрованих команд продуктивніший ніж варіанти структури без суперскалярності, але з кеш-пам'яттю роздешифрованих команд.

R_{32} монотонно зростаюча функція від мінімального значення, меншого 1 при $S = 1$, до максимального значення при $S = \infty$, рівного

$$\lim_{S \rightarrow \infty} R_{32} = \frac{P_{КДК} + (1 - P_{КДК}) * (B / C + \omega_{\Pi} * q * A / C)}{\omega_{\Pi} * q * A / C}.$$

При реалізації варіанту структури з суперскалярністю, але без кеш-пам'яті роздешифрованих команд слід обережно вибирати коефіцієнт суперскалярності. Збільшення S понад оптимального значення не забезпечує істотного приросту продуктивності.

9

Порівняння варіантів 4 і 1

ω_{Π}	q	S	B / C	A / C	D / C	Точне R_{41}	Наближене R_{41}	Похибка (%)
0.05	0.99	1	1	6	1	1.233	1.236	0.24
		2	1	6	1	2.350	2.360	0.43
		3	1	6	1	3.366	3.386	0.59
		4	1	6	1	4.294	4.331	0.86
		1	1	4	1	1.140	1.142	0.17
		2	1	4	1	2.174	2.180	0.28
		3	1	4	1	3.117	3.129	0.39
		4	1	4	1	3.980	4.000	0.50

10

Порівняння варіантів 4 і 1

Реалізація суперскалярності в поєднанні з кеш-пам'яттю розшифрованих команд істотно збільшує теоретичну продуктивність підсистеми генерації команд в порівнянні з найпростішим варіантом структури без суперскалярності та без кеш-пам'яті розшифрованих команд. $R_{41} > 1$ при будь-яких значеннях змінних.

Ефективність структури підсистеми підготовки команд з суперскалярністю в поєднанні з кеш-пам'яттю розшифрованих команд монотонно зростає від мінімального значення при $S=1$ до граничного значення

$$\lim_{S \rightarrow \infty} R_{41} = \frac{B/C + \omega_{II} * q * A/C}{\omega_{II} * q * D/C}$$

Наприклад, при $B/C=1$, $A/C=6$, $\omega_{II}=0.05$ і $q=1$ $\lim_{S \rightarrow \infty} R_{41} = 26$.

Граничне значення показника ефективності R_{41} при прагненні показника суперскалярності S до ∞ визначається в основному значеннями змінних ω_{II} , q , A/D , B/D .

11

Висновки

Варіант структури без суперскалярності, але з кеш-пам'яттю розшифрованих команд більш продуктивний, що підтверджує інтуїтивні припущення про те, що ускладнення структури має забезпечувати збільшення продуктивності.

Збільшення коефіцієнта суперскалярності незначно збільшує теоретичну продуктивність підсистеми генерації команд без кеш-пам'яті розшифрованих команд.

Варіант структури з суперскалярністю і без кеш-пам'яті розшифрованих команд продуктивніший ніж варіанти структури без суперскалярності, але з кеш-пам'яттю розшифрованих команд.

При реалізації варіанту структури з суперскалярністю, але без кеш-пам'яті розшифрованих команд слід обережно вибирати коефіцієнт суперскалярності. Збільшення понад оптимального значення не забезпечує істотного приросту продуктивності.

Реалізація суперскалярності в поєднанні з кеш-пам'яттю розшифрованих команд істотно збільшує теоретичну продуктивність підсистеми генерації команд в порівнянні з найпростішим варіантом структури без суперскалярності та без кеш-пам'яті розшифрованих команд.

12