

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

«На правах рукопису»

УДК _____

До захисту допущено:

Завідувач кафедри

_____ Сергій СТИПЕНКО

«___» _____ 20__ р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-професійною програмою «Комп'ютерні системи та мережі»

зі спеціальності 123 «Комп'ютерна інженерія»

**на тему: «Підвищення ефективності паралельних обчислень на
розрядному рівні у системах на кристалі»**

Виконав (-ла):

студент (-ка) II курсу, групи ІО-02мп

Князева Анжеліка Олександрівна _____

Керівник:

Професор кафедри ОТ, д.т.н., професор,

Жабін Валерій Іванович _____

Консультант з нормконтролю:

Професор кафедри ОТ, д.т.н., професор,

Кулаков Юрій Олексійович _____

Рецензент:

Декан факультету прикладної математики, д.т.н., професор,

Дичка Іван Андрійович _____

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних
посилань.

Студент (-ка) _____

Київ – 2021 року

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

Факультет (інститут) Інформатики та обчислювальної техніки
(повна назва)

Кафедра Обчислювальної техніки
(повна назва)

Освітньо-кваліфікаційний ступінь магістр
(назва ОКР)

Спеціальність 123. Комп'ютерна інженерія
(код і назва)

Спеціалізація 123. Комп'ютерні системи та мережі
(код і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри
Стіренко С.Г.
(підпис) (ініціали, прізвище)

« » _____ 2021 р.

**ЗАВДАННЯ
на магістерську дисертацію студенту**

Князєвій Анжеліці Олександрівні
(прізвище, ім'я, по батькові)

1. Тема дисертації «Підвищення ефективності паралельних обчислень на розрядному рівні у системах на кристалі»

Науковий керівник дисертації Жабін Валерій Іванович, д.т.н., проф.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «27» 10 2021 р. № 3587-с

2. Строк подання студентом дисертації _____

3. Об'єкт дослідження паралельні обчислення на розрядному рівні у системах на кристалі

4. Предмет дослідження методи підвищення ефективності виконання паралельних обчислень на розрядному рівні в системах на кристалі

5. Перелік завдань, які потрібно розробити: аналіз переваг та недоліків використання паралельних методів обчислення у системах, визначення критеріїв ефективності паралельних обчислень у системах на кристалі, розробка нових методів і засобів паралельних обчислень на розрядному рівні у системах на кристалі

6. Консультанти розділів дисертації:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормконтроль	Кулаков Юрій Олексійович		

7. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів дисертації	Примітка
1	Вивчення та аналіз завдання	10.09.2021	
2	Розробка архітектури та загальної структури систем	30.09.2021	
3	Ознайомлення з літературою та підготовка теоретичної частини роботи	10.10.2021	
4	Затвердження теми роботи	27.10.2021	
5	Розробка методу підвищення ефективності паралельних обчислень у системах на кристалі	04.11.2021	
6	Оформлення пояснювальної записки	17.11.2021	
7	Захист	23.12.2021	

Студент

(підпис)

Анжеліка КНЯЗЄВА

(ініціали, прізвище)

Науковий керівник дисертації

(підпис)

Валерій ЖАБІН

(ініціали, прізвище)

АНОТАЦІЯ

Князева А.О. Підвищення ефективності паралельних обчислень на розрядному рівні у системах на кристалі. КПІ ім. Ігоря Сікорського, Київ, 2021.

Проект містить 92 с. тексту, 12 рисунків 21 таблиць, 11 посилань на літературні джерела.

Ключові слова: паралельні обчислення, системи на кристалі, програмована логічна інтегральна схема (ПЛІС), продуктивність, ефективність, споживана потужність, енергоресурси.

Об'єктом дослідження є процес виконання паралельних обчислень на розрядному рівні в системах на кристалі.

Предметом дослідження є методи та засоби виконання паралельних обчислень на розрядному рівні в системах на кристалі.

Метою дипломного проекту є підвищення ефективності виконання паралельних обчислень на розрядному рівні у системах на кристалі.

У магістерській дипломній роботі розроблено метод виконання паралельних обчислень на розрядному рівні у системах на кристалі. Було розглянуто можливість зменшення споживаної потужності та енергоресурсів під час виконання паралельних обчислень, а також описано метод скорочення кількості виводів на ПЛІС. Розроблено стартап-проект на основі даної дипломної роботи.

Отримані результати можуть використовуватись в навчальних цілях та наукових дослідженнях аналізу виконання паралельних обчислень на розрядному рівні у системах на кристалі.

ABSTRACT

Knyazeva A.O. Improving the efficiency of parallel calculations at the bit level in systems on the chip. Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, 2021.

The project contains 92 pages of text, 12 figures, 21 tables, 11 references to literature sources.

Key words: parallel computing, chip systems, programmable logic integrated circuit (FPGA), productivity, efficiency, power consumption, energy resources.

The object of research is the process of performing parallel calculations at the bit level in systems on the chip.

The subject of research is methods and means of performing parallel calculations at the bit level in systems on the chip.

In the master's thesis the method of performing parallel calculations at the bit level in systems on the crystal is developed. The possibility of reducing power consumption and energy resources during parallel calculations was considered and proved, and the method of reducing the number of FPGA outputs was described. A startup project based on this thesis has been developed.

The obtained results can be used for educational purposes and research of the analysis of performance of parallel calculations at bit level in systems on crystal.

РЕФЕРАТ

на магістерську дисертацію

виконану на тему: Підвищення ефективності паралельних обчислень на розрядному рівні у системах на кристалі

студенткою: Князевою Анжелікою Олександрівною

Робота складається із вступу, чотирьох розділів та висновків. Загальний обсяг роботи: 92 аркуша основного тексту, 12 ілюстрацій, 21 таблиць. При підготовці використовувалася література з 11 різних джерел.

Актуальність. Постійний ріст складності обчислювальних задач, які вирішуються на цифрових обчислювальних машинах, надає високу актуальність для усіх питань пов'язаних із підвищенням продуктивності, швидкості та точності виконання обчислень в системах. Паралельні обчислювання є найкращим методом виконання складних та масштабних розрахунків у системах, але вони є також складнішими порівняно з послідовними обчисленнями тому потребують більшої та детальнішої уваги до їх реалізації.

Мета і завдання дослідження. Метою даної дипломної роботи є підвищення ефективності виконання паралельних обчислень на розрядному рівні у системах на кристалі.

Для досягнення поставленої мети у роботі вирішуються такі завдання:

- аналіз обчислювальних задач для яких актуальне та можливе використання паралельного методу обчислень у системах на кристалі;
- аналіз переваг та недоліків використання паралельних методів обчислення у системах, порівняно з використанням послідовних методів обчислення;
- аналіз існуючих методів паралельних обчислень у системах на кристалі, їх переваги та недоліки, види задач задля вирішення яких було застосовано дані методи та ефективність їх використання;

- визначення критеріїв ефективності паралельних обчислень у системах на кристалі;

- вдосконалення методів і засобів паралельних обчислень на розрядному рівні у системах на кристалі.

Об'єктом дослідження є процес виконання паралельних обчислень на розрядному рівні в системах на кристалі.

Предмет дослідження: методи та засоби виконання паралельних обчислень на розрядному рівні в системах на кристалі.

Методи досліджень. Для досягнення поставлених завдань та мети у магістерській роботі було використано методи виконання паралельних обчислень у системах.

Наукова новизна одержаних результатів полягає у вирішенні актуальної проблеми розв'язку складних високопродуктивних обчислювальних задач за допомогою ефективного методу паралельних обчислень у системах на кристалі.

Практична цінність. У даній дипломній роботі розглядається можливість зменшення споживаної потужності та енергоресурсів під час виконання паралельних обчислень, а також скоротити кількість виводів на ПЛІС, що значно скорочує час, виділений на вирішення поставленої задачі.

Особистий внесок здобувача. Магістерська робота є самостійно виконаною роботою, у якій було вивчено, визначено та проаналізовано з авторської точки зору існуючі методи виконання паралельних обчислень на розрядному рівні у системах на кристалі та запропоновано покращення цих методів певними конкретними діями та умовами, які доведено аналітичним шляхом призводять до підвищення ефективності вирішення певного ряду задач.

Формулювання мети та завдань дослідження проводилось спільно з науковим керівником.

Практична цінність. Одержані результати у даній роботі можуть використовуватися у майбутніх дослідженнях за напрямками:

- вдосконалення алгоритмів вирішення задач у системах на кристалі;
- під час вирішення складних задач розробниками за допомогою методу розпаралелювання;
- під час виконання наступних бакалаврських та магістерських дипломних робіт за темою паралельні обчислення у системах на кристалі

Ключові слова:

паралельні обчислення, системи на кристалі, програмована логічна інтегральна схема (ПЛІС), продуктивність, ефективність, споживана потужність, енергоресурси.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	3
ВСТУП.....	4
РОЗДІЛ 1.....	7
ПАРАЛЕЛЬНІ ОБЧИСЛЕННЯ У СИСТЕМАХ НА КРИСТАЛІ Й ЇХ ІСНУЮЧИ МЕТОДИ ВИКОНАННЯ	7
1.1. Паралельні обчислення	7
1.1.1. Переваги використання методів паралельних обчислень ..	8
1.1.2. Недоліки використання паралельних обчислень	10
1.1.3. Аналіз основних рівнів розперелівання обчислень	12
ВИСНОВОК ДО РОЗДІЛУ 1	15
РОЗДІЛ 2.....	16
МЕТОДИКА ПОБУДОВИ ПАРАЛЕЛЬНИХ ПОТОКОВИХ ОБЧИСЛЕНЬ НА БАЗІ ПРОГРАМОВАНИХ ІНТЕГРАЛЬНИХ СХЕМ.....	16
2.1. Системи на кристалі. Визначення та структура	16
2.2. Системи на кристалі, вимоги елементної бази	13
2.3. Методи паралельних обчислень на розрядному рівні кристалі	17
2.4. Аналіз ефективності онлайн обчислень у системі з надлишковим числом	23
2.5. Додаткові питання паралельної обробки даних	27
2.5.1. Інформаційний граф задачі	30
2.6. Схема виконання обчислень	33
2.7. Чисельний аналіз ефективності паралельних обчислень	35
2.8. Додаткові можливості прискорення системи	37
2.9. Розрахунок ефективності використання системи.....	38

2.9.1. Закон Амдала.....	38
ВИСНОВОК ДО РОЗДІЛУ 2	41
РОЗДІЛ 3	42
ПОБУДОВА АЛГОРИТМІВ ВИКОНАННЯ ОПЕРАЦІЙ	42
В НЕАВТОНОМНОМУ РЕЖИМІ	42
3.1. Методика виведення алгоритмів у неавтономному режимі.....	42
3.2. Приклад використання зміщеної надлишкової системи числення.....	43
3.3. Побудова систем з безпосередніми зв'язками між обчислювальними модулями	47
ВИСНОВОК ДО РОЗДІЛУ 3	55
РОЗРОБКА СТАРТАП ПРОЕКТУ.....	56
4.1.Опис ідеї стартапу.....	56
4.2.Технологічний аудит ідеї	59
4.3.Аналіз ринкових можливостей запуску стартап-проекту.....	61
4.4.Розроблення маркетингової програми стартап-проекту.....	73
ВИСНОВОК ДО РОЗДІЛУ 4.....	76
ВИСНОВКИ	77
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	78

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ГПД	—	графік потоку даних
ІГ	—	інформаційний граф
ІЧ	—	інтегрований чіп
ОБ	—	обчислювальний блок
ОМ	—	обчислювальний модуль
ОП	—	операційний пристрій
ПКВМ	—	програмована користувачем вентильна матриця
ПЛІС	—	програмовні логічні інтегральні схеми
САПР	—	систем автоматизованого проектування
СНК	—	системи на кристалі
ЦОМ	—	цифрові обчислювальні машини
ЯПФ	—	ярусно-паралельна форма

ВСТУП

Магістерську роботу присвячено напряму підвищення ефективності паралельних обчислень на розрядному рівні у системах на кристалі. Постійний ріст складності обчислювальних задач, які вирішуються на цифрових обчислювальних машинах, надає високу актуальність для усіх питань пов'язаних із підвищенням продуктивності, швидкості та точності обчислень в системах.

Розпаралелювання є одним із найефективніших методів пришвидшення вирішення та масштабування обчислювальних задач. Основні принципи паралелізму були впроваджені ще у перші експериментальні паралельні машини, які з'явилися в 60-70 роках минулого століття. Цей метод може бути вдало використаним при вирішенні багатьох задач на кристалі, які мають жорсткі часові обмеження на їх вирішення й вимагають досить великої точності результатів зроблених обчислень. Його головна ідея у тому аби використовувати декілька окремих пристроїв або процесорів одного пристрою для вирішення однієї великої задачі, тобто розбити задачу на декілька підзадач, де кожна з підзадач зможе обчислюватись паралельно, тобто одночасно, незалежним пристроєм чи процесором.

Найголовнішими перевагами при вирішенні задач, які можуть дати паралельні обчислювання на кристалі це: збільшення швидкості виконання обчислень та збільшення масштабів обчислень.

Використання декількох, навіть не великої потужності обчислювальних ресурсів може пришвидшити процес знаходження розв'язків складних задач, причому виграш від оперативності часто перевищує вартість використання додаткових ресурсів саме тому паралельні обчислення на сьогодні частіше є більш актуальними, при вирішенні великих за об'ємом та кількістю даних задач, ніж послідовні.

Актуальним напрямком підвищення ефективності паралельної обробки інформації є побудова комп'ютерних систем на програмовній елементній базі ПЛІС (програмована логічна інтегральна схема). Застосування технологій ПЛІС дозволяє реалізувати в одному кристалі мікросхеми обчислювальну систему досить високої складності з гнучкою архітектурою. Можливість перепрограмування ПЛІС на фізичному рівні забезпечує швидке налаштування обчислювальної структури для реалізації довільних алгоритмів обчислювальних задач. [1]

Отже, у даній дипломній роботі будуть розглянуті переваги та недоліки використання методів паралельних обчислень, порівняно з послідовними обчисленнями. Проаналізовано для яких видів обчислювальних задач можливо ефективно застосувати методи паралельних обчислень на розрядному рівні у кристалі, а також розглянуто види задач для вирішення яких не можливо застосування методів розпаралелювання й чому. Будуть розглянуті вже відомі методи використання паралельних обчислень. У висновку до магістерської дипломної роботи буде надано аналіз порівняння ефективності існуючих методів паралельних обчислень з новими запропонованими методами й отримано аналітичний розбір задля яких конкретних задач ефективніше використовувати той чи інший метод паралельних обчислень у системах на розрядному рівні на кристалі.

Однак, при реалізації деяких обчислювальних процесів використання декількох пристроїв переробки інформації, що реалізують традиційні методи виконання операцій, не зменшує час обчислень.

При вирішенні багатьох практичних завдань знаходять застосування чисельні методи обчислень, найважливішим колесом яких є ітераційні методи. Вони застосовуються під час вирішення систем лінійних і нелінійних рівнянь, під час вирішення систем звичайних диференціальних рівнянь, диференціальних рівнянь із приватними похідними, інтегральних рівнянь, під час вирішення крайових завдань, обчисленні функціональних

залежностей тощо.

Характерна риса ітераційних обчислень у тому, що у кожному кроці ітерації використовується результат, по крайнього заходу, одного попереднього кроку. Отже, при реалізації ітераційних процесів потрібно виконувати послідовності операцій, що не розпаровуються.

Часто під час вирішення завдань доводиться також обчислювати функціональні залежності між величинами, поставлені багатомісними висловлюваннями. Як правило, такі вирази обчислюються в ЦВМ шляхом виконання суперпозиції одно- та двомісних операцій (функцій), що визначаються складом машинних команд. У зв'язку з цим при обчисленні багатомісних виразів, як і при реалізації ітераційних процесів, потрібно виконувати послідовності операцій, що не розпаровуються.

Традиційні методи машинної арифметики неможливо виконувати послідовності нерозпаралелюваних операцій у режимі суміщення навіть у тому випадку, коли обчислювальна система має кілька операційних пристроїв, здатних працювати паралельно в часі.

Таким чином, підвищення швидкодії засобів обчислювальної техніки при реалізації ітераційних процесів та обчисленні функцій багатьох змінних є важливим завданням, вирішення якого вимагає розробки нових принципів організації обчислювальних процесів як на структурному рівні, так і на рівні машинної арифметики.

Метою роботи є розробка та дослідження методів зменшення часу виконання послідовностей нерозпаралелюваних операцій та створення на їх основі швидкодіючих обчислювальних засобів для виконання послідовності залежних за даними операцій.

РОЗДІЛ 1

ПАРАЛЕЛЬНІ ОБЧИСЛЕННЯ У СИСТЕМАХ НА КРИСТАЛІ Й ЇХ ІСНУЮЧИ МЕТОДИ ВИКОНАННЯ

У цьому розділі буде надано визначення паралельним обчисленням та їх призначення. Описано основні принципи паралельних обчислень, переваги та недоліки їх використання порівняно з послідовними обчисленнями. Також буде надано перелік видів задач у яких ефективно використання саме паралельних видів обчислень у системах на кристалі та у яких випадках використання паралелізму і його принципів неможливе, чи не є ефективним методом для даних обчислень. Описано популярні існуючі методи паралельних обчислень та надано приклади їх використання.

1.1. Паралельні обчислення

Паралельні обчислення - це обчислення при яких частини однієї задачі вирішуються за допомогою декількох незалежних пристроїв об'єднаних у одну систему, чи процесорів в одному пристрої, одночасно. Їх головна ідея полягає у тому що, великі за розміром та складністю задачі можна розподілити на декілька менших задач, які називають підзадачами й вирішувати їх у один й той самий час на окремих пристроях переробки інформації, які об'єднані у одну систему.[2]

Головна ціль існування та розвитку паралельних обчислень - це зменшення часу витраченого на вирішення задачі. Багато необхідних обчислювальних задач потрібно вирішувати у реальному часі або для їх вирішення знадобиться великий об'єм обчислень та досить велика потужність характеристик обчислювального пристрою.

Використання паралельної обробки даних не єдиний шлях збільшення швидкості, точності та продуктивності виконання обчислень у системах.

Інший не менш популярний підхід - це збільшення потужності процесорних обчислень, але він має досить велику кількість обмежень:

1. Обмеженість швидкості перемикавання. Навіть при дуже швидких оптичних комунікаціях - швидкість перемикавання не може перевищувати швидкість світла, а часто навіть у надсучасних пристроях ця швидкість значно нижча за швидкість світла.
2. Обмеженість розмірів компонентів системи. Чим менший розмір компонентів пристрою - тим швидше пристрій може працювати. Однак існує фізичне обмеження на розмір компонентів системи, що пов'язано з їх молекулярним й атомним складом.
3. Економічні обмеження. Для збільшення швидкості та продуктивності процесора доводиться вирішувати інженерні й продовольчі проблеми. Ось чому кожне наступне покоління процесорів має більшу вартість ніж минуле.[2]

Отримати паралельний алгоритм вирішення задачі можливо шляхом розпаралелювання вже розробленого послідовного алгоритму або шляхом розробки нового паралельного алгоритму. Можливо для розпаралелювання алгоритму вирішення обчислювальної задачі доведеться замінити або модифікувати деякі його частини. Наприклад, прибрати або замінити деякі залежності між операціями.

1.1.1. Переваги використання методів паралельних обчислень

У наші дні принципи паралельних обчислень використовуються у більшості обчислювальних пристроїв, найбільш поширеними системами з паралелізмом є кластерні системи та багатоядерні комп'ютери, які сьогодні має майже кожна людина. Але, як й будь-яка технологія, паралельні обчислення мають свій розвиток, складність у проєктуванні та свої переваги й недоліки використання. Перевагами паралельних обчислень є:

- Можливість обчислення декількох задач в один й той самий момент. Ця перевага значно зменшує витрати часу розробника на вирішення декількох задач. Паралельне обчислення у системах дозволяє кількості задач, виконаних за заданий час, збільшуватись пропорційно до кількості процесорів, або окремих обчислювальних пристроїв об'єднаних в одну систему, згідно закону Густафсона. Але дана перевага використання методів паралельних обчислень не скорочує швидкість вирішення однієї конкретної задачі.

- Можливість швидше та ефективніше вирішити одну комплексну задачу, яку потрібно та можливо розбити на підзадачі й вирішувати на декількох обчислювальних пристроях об'єднаних у одну систему. Ця перевага значно збільшує швидкість та ефективність вирішення однієї задачі, але на жаль підходить не до усіх видів обчислювальних задач.

- Використання паралельних обчислень дозволяє також збільшити масштаби виконання обчислень. Масштабованість є дуже бажаною властивістю будь-якої комп'ютерної системи, мережі, або процесу. Це свідчить про здатність обробити більший обсяг завдань комп'ютерної системи або бути легко розширеною, коли до системи будуть додані додаткові обчислювальні ресурси. Також масштабованість вказує, що продуктивність обчислювальної системи прямо пропорційна кількості процесорів (ядер) що містяться в неї. Розрізняють усього два поняття масштабованості: слабка та сильна. Слабка масштабованість означає лінійний ріст розміру задачі з ростом числа процесорів при фіксованому часі виділеному на виконання алгоритму. Сильна масштабованість означає лінійний зріст прискорення з ростом прискорення числа процесорів при фіксованому розмірі задачі.

1.1.2. Недоліки використання паралельних обчислень

Кожна технологія, особливо та яка активно розвивається й є досить не простою має не тільки переваги у використанні, а ще й недоліки, так і паралельні обчислення також мають свої певні недоліки, які пов'язані, як й зі складністю самої технології так й зі стрімким розвитком. Тому паралельні обчислення мають такі недоліки їх використання:

- Не усі типи обчислювальних задач можливо вирішити за допомогою методу розпаралелювання. Вирішення не всіх задач підлягає розбиттю на підзадачі, обчислення яких може бути виконане за допомогою декількох пристроїв однієї системи. Це так звані ітераційні обчислення та обчислення які мають функціональні залежності між величинами. Характерна особливість ітераційних обчислень полягає у тому, що кожна наступна операція обчислення вимагає точного результату виконання, як мінімум однієї попередньої операції цього обчислення. Дуже часто при вирішенні задач доводиться також обчислювати функціональні залежності між величинами. Як правило, такі вирази обчислюються у цифрових обчислювальних машинах (ЦОМ) шляхом виконання суперпозиції одно- й двомісних операцій (функцій), визначених складом машинних команд. У зв'язку з цим при вирахуванні багатомісних виразів, як й при реалізації ітераційних процесів, вимагається виконувати послідовність нерозпаралелених операцій. Традиційні методи машинної арифметики не дозволяють виконувати послідовності нерозпаралелених операцій у режимі суміщення навіть у тому випадку, коли обчислювальна система має декілька операційних пристроїв, які здатні працювати паралельно у часі. Це обґрунтовується тим що у вказаному випадку виконання чергової операції не можна виконати до тих пір, поки не буде отримано повністю код результату попередньої операції.

- Часто вирішення задач із використанням методів паралелізму вимагає від розробників більше зусиль, ніж використання послідовного обчислення.

Це пов'язане з іншим типом мислення, який використовується під час архітектурного проєктування цієї задачі та розбиття її на підзадачі.

- Невизначеність пріоритету задачі чи підзадачі при виконанні паралельних обчислень. Під час даного обчислення інколи складно вирішити питання саме пріоритетності вирішення підзадач однієї високорівневої задачі. Тому що, рідко існують підзадачі від результату вирішення яких не залежить подальше вирішення будь-якої наступної підзадачі.

- Комунікація, та синхронізація процесів зазвичай одна з найбільших перешкод для досягнення гарної продуктивності використання паралельних обчислень.

- Нескінченна відстрочка. Одна чи декілька підзадач очікують зв'язку від іншої підзадачі до її виконання. Якщо такий зв'язок між підзадачами не налагоджено, то підзадачі нескінченно довго чекатимуть результату завершення одна від одної, що призведе до зупинки паралельного обчислення задачі.

- Взаємоблокування. Може статися при паралельно виконуваних підзадачах, що мають загальний доступ до одних даних. Можливо ситуація, при якій кожна із підзадач чекатиме поки інша не звільнить доступ до даних, що призведе до збільшення часу виконання обчислення чи взагалі до повної зупинки обчислення.

- Відмова обладнання. Якщо якійсь процесор чи пристрій системи дадуть збій, виникає проблема, при якій необхідно вирішити, як поводитимуться інші пристрої у системі, щоб не припиняти процес виконання обчислення.

- Зайвий паралелізм. Вираз «що більше процесорів, тим краще» часто не відповідає дійсності. Організація роботи та швидка синхронізація усіх пристроїв при їх дуже великій кількості може здатися для архітектора системи досить скрутною. Тому в кожному конкретному випадку необхідно використовувати оптимальну та необхідну кількість пристроїв у системі, щоб не зашкодити швидкодії її роботи.

1.1.3. Аналіз основних рівнів розперелівання обчислень

Можна виділити кілька основних рівнів розпаралелювання обчислень:

- рівень окремих завдань;
- рівень програмних модулів;
- рівень операцій;
- рівень розрядів операндів операцій.

Для організації паралельних обчислень існує безліч різних технологій, багато з них ґрунтуються на принципово різних підходах, принципах, мають різну сферу застосування та ступінь поширення. Наведено список засобів програмування для вирішення завдань на паралельних комп'ютерах, що включає понад сотню найменувань. Технології пов'язані з різними чинниками, основним у тому числі належить архітектура обчислювальних систем. Архітектурні особливості систем визначають спосіб обміну інформацією між процесорами (операційними модулями, обчислювачами). Залежно від способу обміну даними засоби розробки паралельних програм можна розділити на два типи:

- засоби, орієнтовані модель загальної пам'яті (мультипроцессори);
- кошти, зорієнтовані модель розподіленої пам'яті (мультикомп'ютери).

Існують і комбіновані системи (NUMA-system).

У розподілених обчислювальних системах (мультикомп'ютерних системах) обмін даними між комп'ютерами (процесорами) здійснюється за допомогою пакетів. При передачі пакетів провадиться пересилання додатково певного обсягу службової інформації. Звичайно, для дрібнозернистих алгоритмів такий похід не є ефективним через складну процедуру обміну даними.

Другий тип паралельних систем – це багатопроцесорні обчислювальні системи. Процесори у такій системі мають спільну пам'ять, через яку

здійснюється обмін інформацією словами певної довжини. Для алгоритмів з меншим зерном і обсягом даних мультипроцесори можуть забезпечити велику швидкість обчислень.

З урахуванням зернистості задачі та обсягу даних при розв'язанні задачі можна визначити болем прийнятний варіант архітектури системи.

Для перших двох зазначених рівнів розпаралелювання можуть бути використані спеціальні засоби розробки паралельних програм:

- примітиви мережевого транспортного рівня (сокети) [8];
- мови, засновані на моделі паралелізму за даними та управління [9];
- бібліотеки, що базуються на моделі передачі повідомлень;
- технології, засновані на моделі віддаленого виклику процедур: протокол OSF RPC [8], Glade у мові Ada, а також високорівневі технології, що базуються на об'єктно-орієнтованому підході з використанням інтерфейсів: RMI у мові Java [8], DCOM у ОС Windows, специфікація CORBA [8] протокол SOAP.
- технології, що базуються на моделі простору кортежів: Linda [7, 8], Jini (JavaSpace) [8];
- спеціалізовані бібліотеки паралельних процедур, орієнтованих на вирішення завдань у конкретних галузях, наприклад лінійної алгебри, методів Монте-Карло, генетичних алгоритмів;
- засоби автоматичного та напіваавтоматичного розпаралелювання послідовних програм.

Зазначені засоби можна поділити на засоби статичного та динамічного розпаралелювання. Однак для динамічного розпаралелювання обчислень на рівні операцій запропоновано технологію обчислень під управлінням потоку даних (data flow).

Потокові системи (data flow) були розроблені для динамічного розпаралелювання обчислювальних процесів на рівні команд. До перших

праць у цій галузі слід віднести роботи Д. Адамса та Дж. Родрігеса, які вплинули на подальші дослідження потокової моделі обчислень. Для поточкових систем відсутнє поняття програми як пов'язаного списку команд, що виконуються у певній послідовності під управлінням лічильника команд, що притаманне класичній моделі обчислень. Визначальним у разі є порядок виконання команд, а доступність даних для команди. Для машин такого класу програма є списком управляючих слів (акторів) і даних (токенів), які через пристрої введення надходять у середовище формування команд. З цих компонентів на апаратному чи мікропрограмному рівні формуються команди. Готові команди передаються до обчислювальних модулів, які працюють незалежно один від одного (асинхронно). За наявності вільного обчислювача команда в ідеальному випадку негайно виконується, генеруючи вихідні дані.

Розпаралелювання здійснюється автоматично в динаміці. Однак у таких системах також недоліком є втрата часу на пересилання даних між обчислювальними модулями.

Для розпаралелювання обчислень на рівні розрядів операндів запропоновані методи, які можуть бути використані для виконання послідовностей (ланцюжків) залежних за даними операцій. Ці методи дозволяють частково поєднувати одночасну обробку операнда для різних залежних команд. Розгляд таких методів обчислень присвячений наступний розділ.

ВИСНОВОК ДО РОЗДІЛУ 1

У даному розділі дисертації розглянуті методи паралельної обробки даних та основні відмінності обчислювальних систем при вирішенні задач із різною структурою. Показано, що більшість алгоритмів обробки даних у реальному часі в автоматичних управляючих системах мають дрібнозернисту структуру. Прискорити вирішення завдань можливо за рахунок суміщення у часі виконання паралельної обробки даних на рівні операцій та розрядів операндів залежних за даними операцій.

Підвищення ефективності обчислень на базі програмованих інтегральних мікросхем може бути реалізовано за рахунок зменшення внутрішніх ресурсів мікросхем, а саме, кількості виводів (пінів), а також їх функціональних та комутаційних засобів. Для реалізації дрібнозернистих алгоритмів доцільно використовувати методи паралельних обчислень на командному та розрядному рівні.

РОЗДІЛ 2.

МЕТОДИКА ПОБУДОВИ ПАРАЛЕЛЬНИХ ПОТОКОВИХ ОБЧИСЛЕНЬ НА БАЗІ ПРОГРАМОВАНИХ ІНТЕГРАЛЬНИХ СХЕМ

2.1. Системи на кристалі. Визначення та структура

Система на кристалі – це електронна схема, яка виконує функції цілого пристрою й розміщена на одній інтегральній схемі [3].

Система на кристалі поєднує необхідні електронні схеми різних комп'ютерних компонентів на одному інтегрованому чіпі (ІЧ). Система на кристалі – це повна електронна система підкладки, яка може містити аналогові, цифрові, змішані чи радіочастотні функції.

Залежно від призначення вона може оперувати як цифровими сигналами, так і аналоговими, аналого-цифровими, а також частотами радіодіапазону.

Якщо розмістити всі необхідні ланцюги на одному напівпровідниковому кристалі не вдається, застосовується схема з кількох кристалів, поміщених у єдиний корпус. СНК вважається більш вигідною конструкцією, тому що дозволяє збільшити відсоток придатних пристроїв при виготовленні, що зменшує ризик повної чи часткової зупинки обчислень при несправності одного з пристроїв системи, та спростити конструкцію корпусу.

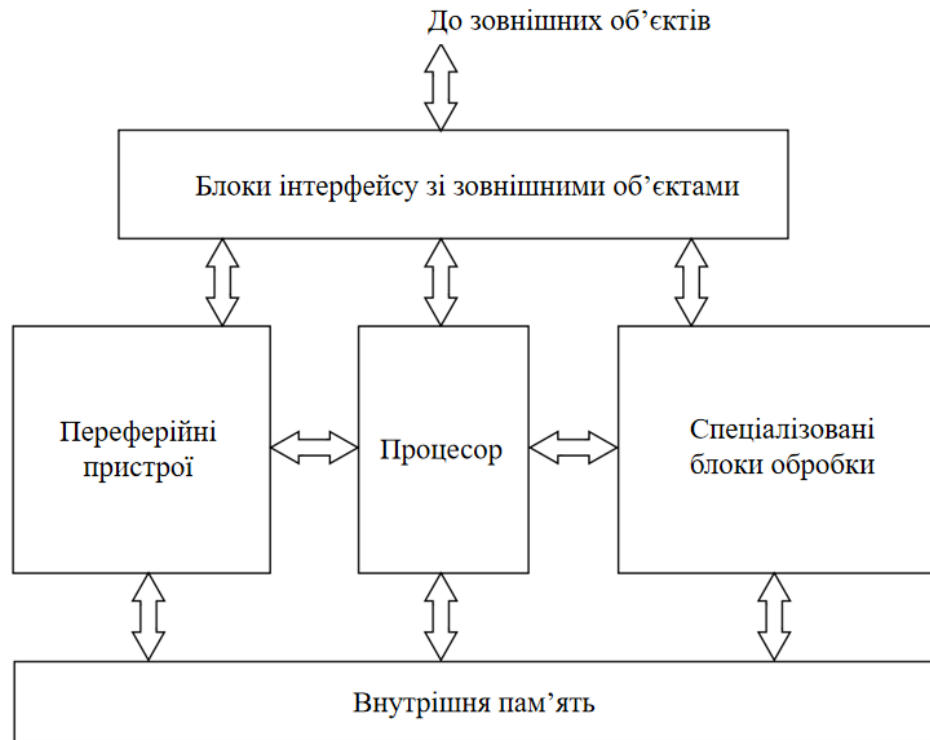


Рис 2.1. Приклад структури системи на кристалі

Основним цифровим блоком у системах на кристалі зазвичай є процесор, який виконує програмну обробку цифрових даних. Спеціалізовані блоки обробки забезпечують апаратне виконання функцій, специфічних для цієї системи. Це може бути, наприклад, блоки цифрової обробки сигналів, аналогові схеми, перетворювачі потоків даних та ін. Різні типи модулів пам'яті (SRAM, DRAM, ROM, EEPROM, Flash) можуть входити до складу СНК або підключатися до неї, як зовнішні блоки. Таймери, АЦП та ЦАП, широтно-імпульсні модулятори та інші цифрові пристрої можуть інтегруватися до складу СНК, як периферійні пристрої. Інтерфейс із зовнішніми пристроями забезпечується за допомогою паралельних та послідовних портів, різних шинних та комунікаційних контролерів та інших інтерфейсних блоків. Організація зв'язків між блоками системи також може бути різною: можливе використання різних стандартизованих шин (типу AMBA) або спеціалізованих локальних інтерфейсів.

2.2. Системи на кристалі, вимоги елементної бази

Сучасна реалізація систем на кристалі має наступні варіанти:

- у вигляді замовленої HBIC (ASIC);
- на базі ПЛІС високої інтеграції (FPGA) [4].

При реалізації СНК у вигляді ASIC використовується традиційний маршрут проектування ASIC з використанням апаратно реалізованих СФ-блоків, інтегрованих у структуру HBIC, та синтезованих СФ-блоків, які виробник транслює у фізичну структуру за допомогою власних бібліотек функціональних елементів. Використовуючи засоби систем автоматизованого проектування (САПР), набір необхідних СФ-блоків і сучасні технології, можливо реалізувати у вигляді ASIC більшість електронних пристроїв, що монтуються в даний час на друкованих платах. Таким чином, є можливість заміни систем реалізованих на платі системами на кристалі [4].

Абревіатура FPGA означає Field Programmable Gate Array - програмована користувачем вентильна матриця. Програмована користувачем вентильна матриця, ПКВМ - напівпровідниковий пристрій, що може бути налаштований виробником або розробником після виготовлення; звідси назва: -програмується користувачем. ПКВМ програмується шляхом зміни логіки роботи принципової схеми, наприклад, за допомогою вихідного коду мовою проектування (типу VHDL), на якому можна описати цю логіку роботи мікросхеми. ПКВМ є однією з архітектурних різновидів програмованих логічних інтегральних схем (ПЛІС).

Переваги використання систем заснованих на платі:

- використання перевірених серійних компонентів;
- простіший процес тестування та налагодження системи;
- можливість заміни несправних компонентів;

- низька вартість створення дослідних зразків та малих серій.[4]

Переваги використання систем на кристалі:

- можливість отримання більш високих технічних показників (вища продуктивність виконання обчислень, енергоспоживання);
- нижча вартість при великому випуску [4].

Переваги реалізації СНК на базі FPGA (ПЛІС):

- малі витрати на розробку та створення зразків;
- можливість корекції проекту;
- використання перевірених серійних виробів;
- простіший процес тестування та налагодження (можливість реалізації та налагодження «по частинах»).

Таким чином, системи на кристалі (СНК) на базі FPGA мають практично ті ж переваги, що й системи на платі, але відрізняються кращими технічними характеристиками - нижчим енергоспоживанням, меншими габаритами та масою. Із усього вищесказаного, можливо зробити висновок, що СНК на базі FPGA (ПЛІС) будуть поступово витісняти системи на платі. При цьому замість мікропроцесорів і мікроконтролерів у цих СНК будуть використовуватися різні варіанти процесорних СФ блоків.

Важливою умовою ефективного використання програмованих мікросхем є можливість швидкого завантаження даних в систему. Для цього треба забезпечити достатню кількість виводів (пінів), інакше основною перешкодою високої швидкості обчислень буде проблема доступу до даних

Сучасні ПЛІС мають мільони функціональних компонентів, що дає можливість розмістити в корпусі мікросхеми багато апаратури. Однак, не завжди можна забезпечити паралельне функціонування компонентів системи або декількох систем. Причиною цього є обмеження на кількість виводів для

введення та виведення даних.

При використанні НВІС (замовних або програмованих) обчислювальні модулі можуть розміщуватися всередині однієї мікросхеми. Можливо, для побудови системи знадобиться кілька мікросхем (рис. 2.1). Це визначається як розмірністю завдань, і складністю обчислювальних модулів, що від реалізованих ними алгоритмів виконання операцій. Таблиця сімейства ПЛІС (табл. 2.1) ілюструє функціональні можливості мікросхем.

Таблиця 2.1

Мікросхеми сімейства STRATIX

	ПЛІС	ALM	Еквив. ЛЭ	Regs	М9К Блоки	М144К Блоки	Встроенная память (Кбит)	MLAB (Кбит)	Макс. кол-во умножителей 18х18
Stratix III Logic	EP3SL50	19,000	47,500	38,000	108	6	1,836	594	216
	EP3SL70	27,000	67,500	54,000	150	6	2,214	844	288
	EP3SL110	42,600	106,500	85,200	275	12	4,203	1,331	288
	EP3SL150	56,800	142,000	113,600	355	16	5,499	1,775	384
	EP3SL200	79,560	198,900	159,120	468	24	7,668	2,486	576
	EP3SE260	101,760	254,400	203,520	864	48	14,688	3,180	768
	EP3SL340	135,200	338,000	270,400	1,144	48	17,208	4,225	576
Stratix III Enhanced	EP3SE50	19,000	47,500	38,000	400	12	5,328	594	384
	EP3SE80	32,000	80,000	64,000	495	12	6,183	1,000	672
	EP3SE110	42,600	106,500	85,200	639	16	8,055	1,331	896
	EP3SE260	101,760	254,400	203,520	864	48	14,688	3,180	768

Рис.2.2. Характеристики мікросхем сімейства STRATIX III

Використання декількох мікросхем створює додаткові проблеми. Недоліком такої реалізації є те, що мікросхеми можуть не мати необхідну кількість виводів для задоволення потреб у зовнішніх входах і виходах схеми, а також для зв'язку між різними блоками пристрою. Ще одна проблема

пов'язана з тим, що частина логічних ресурсів залишається не задіяною, і в результаті для реалізації необхідної функціональності системи буде потрібно багато мікросхем.

Зважаючи на важливість цих проблем, на початку 90-х років компанія Virtual Machine Works оприлюднила свою нову технологію під назвою VirtualWire (віртуальні з'єднання), представлену як технологія виробництва великих пристроїв, для реалізації яких доводиться використовувати кілька мікросхем [1].

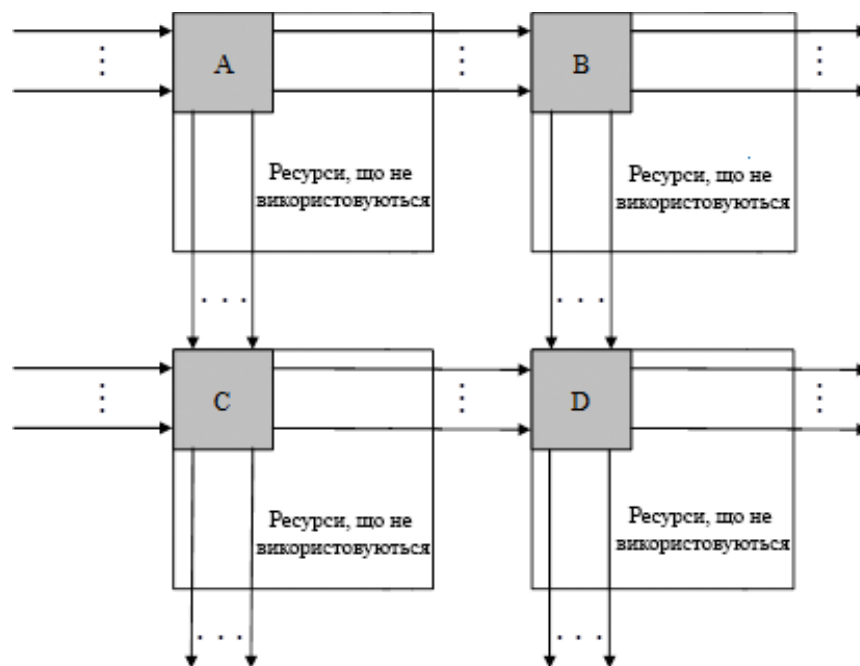


Рис. 2.3. Структурна декомпозиція системи на HBIC з урахуванням числа виводів мікросхем: A, B, C, D – частини системи

Ідея, закладена в основу технології Virtual Wire, ось у чому. Оскільки певна кількість внутрішніх ресурсів пристрою витрачається марно, частина їх можна використовувати для реалізації спеціальних ланцюгів. Ці ланцюги дозволяють перемикає входи мікросхеми між декількома регістрами, кожен з яких підключений до певного логічного блоку. Виходи цих логічних блоків можуть бути мультиплексовані, а вихідні дані збережені у регістрах. Для цього всередині кожної ПЛІС має бути реалізований автомат, який використовується для управління регістрів, мультиплексорів та інших схем.

Ця технологія, хоч і може вирішити проблему нестачі виводів мікросхем, але не дозволяє повною мірою використовувати незадіяні ресурси мікросхем. Крім того, мультиплексування створює великі тимчасові затримки просування потоків даних, що суперечить ідеї потокової моделі обчислень.

Одним із підходів до вирішення проблеми нестачі висновків мікросхем є використання квазіпаралельної арифметики, що дозволяє поєднувати процеси порозрядного введення операндів та порозрядного формування результатів. Це дозволяє зменшити необхідну кількість зв'язків між обчислювальними модулями, а також забезпечити виконання послідовностей залежних операцій у неавтономному режимі (у режимі суміщення). Це дає потенційну можливість прискорення обробки даних у поточкових системах, побудованих на базі квазіпаралельних обчислювальних модулів.

2.3. Методи паралельних обчислень на розрядному рівні кристалі

На сьогодні застосування методів паралельних обчислень на кристалі вже набуло певної популярності й привернуло увагу до себе вчених та розробників, які розробили та описали деякі методи розпаралелювання. У цьому підрозділі ознайомимося з ними детальніше.

Послідовність нерозпаралелювальних операцій можна представити у вигляді графа потоку даних (ГПД), який подано на рис.2.4.[5].

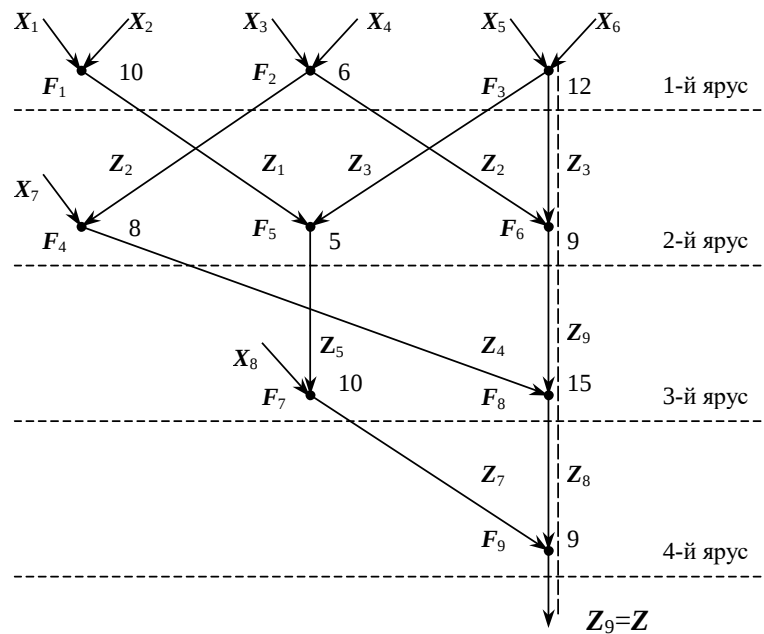


Рис. 2.4. ГПД послідовно-паралельного алгоритму

На ГПД F_j - операції; $X_i^{(j)}$ - вихідні дані; Z_j - внутрішні проміжні результати, що формуються при виконанні операції; $Y_i^{(j)}$ - зовнішні проміжні результати. За наявності зовнішніх проміжних результатів такий граф відбиває послідовність нерозпаралелюваних операцій, що лежать на критичному шляху графа алгоритму.

Операції, які є не розпаралелюваними, можна виконувати у режимі часткового суміщення. На кожному кроці в операційному пристрої (ОП) вводиться по одному розряду операндів й таким чином формується один розряд результату. Якщо таким же чином виконуватимуться всі операції F_j , то розряд проміжного результату, отриманий на i -му кроці при виконанні i -ї операції, може бути використаний на $(i+1)$ -му кроці при виконанні $(j+1)$ -ї операції. У зазначеному випадку виконання наступної операції буде починатися відразу після отримання першого розряду результату цієї операції.

Припустимо, що ОП, який виконує операцію F_j починає формувати розряди результату із затримкою на P_j кроків. У цьому випадку число кроків, необхідне для одержання n розрядів остаточного результату при виконанні K операцій, становить

$$M = n - 1 + \sum_{j=1}^K (P_j + 1). \quad (2.1)$$

Тривалість кроку обчислень має відповідати такому умові [5]:

$$T_{\mathcal{U}} \geq \max_j T_{\mathcal{U}j} \quad (2.2)$$

де $T_{\mathcal{U}j}$ – є тривалість циклу формування розряду результату при виконанні j -ї операції. А, отже, час виконання до нерозпаралелювальних операцій буде визначатися саме так:

$$T = \left[n - 1 + \sum_{j=1}^K (P_j + 1) \right] T_{\mathcal{U}} \quad (2.3)$$

Час виконання операції з глибиною K , $(T_0(K))$ визначається у припущенні, що операція входить у послідовність, що містить тільки такі операції. Підставляючи у формулу (1.4) $P_j = P$ і $T_{\mathcal{U}j} = T_{\mathcal{U}}$, отримаємо наступний вираз:

$$T_0(K) = [n - 1 + K(p + 1)] T_{\mathcal{U}}, \quad (2.4)$$

де $T_{\mathcal{U}}$ є тривалістю циклу формування одного розряду результату під час виконання цієї операції [5].

Будемо використовувати граничний середній час T_0^* , як ще одну характеристику виконання операції, що входить у послідовність. До операцій:

$$T_0^* = \lim_{K \rightarrow \infty} \frac{T_0(K)}{K} = (p + 1) T_{\mathcal{U}} \quad (2.5)$$

Ця характеристика дозволяє оцінити теоретично мінімальний середній час виконання операції, що входить до послідовності [5].

З виразів (2.4) і (2.5) випливає, що для підвищення ефективності необхідно прагнути до зменшення значень P_i і $T_{\mathcal{U}j}$.

Слід зауважити, що суміщення при виконанні операцій, що не розпаровуються, може бути досягнуто і в тому випадку, коли на кожному кроці вводиться кілька розрядів операндів і формується кілька розрядів результату [5].

Зменшення часу реалізації дрібнозернистих алгоритмів може бути досягнуто методом розпаралелювання обчислень на рівні обробки машинних слів, тобто за рахунок паралельного виконання операцій у різних обчислювальних вузлах.

Однак у деяких випадках, при обчисленні багатомісних виразів шляхом суперпозиції функцій з меншим числом аргументів виникає необхідність виконання послідовностей операцій, залежних за даними. Розпаралелювання ланцюжків залежних операцій тільки на рівні обробки машинних слів на разі не є можливим.

Таким чином, якщо для виконання операцій використовуються тільки традиційні методи машинної арифметики, то реалізацію послідовності операцій не можна здійснити за час, менший ніж сумарний час виконання всіх операцій. Якщо послідовність містить k операцій, час виконання послідовності операцій у разі не може бути менше за:

$$T = \sum_{j=1}^k t_j, \quad (2.6)$$

де j – номер операції, t_j – час виконання j -ї операції.

У загальному випадку цей час не є повним часом реалізації алгоритму, оскільки тут не враховується час обміну інформацією між пристроями системи, що суттєво залежить від її топології.

Критичним шляхом зваженого ярусно-паралельного графа називають такий шлях, якому відповідає максимальна сума терезів вершин, що лежать на цьому шляху. На рис.2.4 такий шлях відзначений пунктирною лінією. У цьому випадку $T=t_3+t_6+t_8+t_9$ що становить 45 умовних одиниць часу.

Зауважимо, що цей час не є загальним часом реалізації алгоритму, тому що тут не враховується час обміну інформацією між процесорами, що суттєво залежить від топології системи.

Підвищення швидкодії обчислювальних систем під час виконання ланцюжків залежних операцій можливо досягнути при застосуванні методів машинної арифметики, які дозволяють виконувати залежні за даними операції у режимі часткового суміщення за допомогою надлишкових систем числення. Вимоги до алгоритмів виконання залежних операцій:

На підставі вищевикладеного можна сформулювати вимоги, що дозволяють виконувати нерозпаралелювані операції в режимі суміщення.

1. Порозрядне введення операндів та порозрядне формування результату.
2. Розряди результату, формовані кожному кроці, не змінюють своєї достовірності (не коригуються) надалі процесі обчислення.
3. Затримка при формуванні розрядів результату, виражена в числі кроків, менша за розрядність операндів (ідеальним є випадок, коли розряди результату формуються без затримки).
4. Однакова форма представлення інформації на виході ОП, формує результат, і вході ОП, використовує цей результат як операнда.

Розглянемо можливість застосування традиційних методів виконання арифметичних операцій під час виконання ланцюжка залежних операцій.

Найбільшого поширення в цифровій обчислювальній техніці отримали однорідні позиційні системи числення з натуральною основою $k \geq 2$ і цифрами, що належать множині $\{0, 1, \dots, k-1\}$. Операції складання та віднімання у таких системах виконуються, починаючи з молодших розрядів. При складанні (відніманні) двох молодших розрядів операндів формується перенесення (позиція) у старший розряд і молодший розряд результату, який надалі не коригується. При додаванні наступних розрядів враховується перенесення з молодшого розряду та формується наступний достовірний розряд результату, представлений у тій же системі числення, що й операнди.

Отже, у канонічних системах числення можна виконати всі чотири викладені вище вимоги при реалізації операцій складання та віднімання.

Відомі чотири основні способи множення чисел у позиційних системах числення [2]. Множення можна виконувати або зі старших або з молодших розрядів множника. У кожному разі можна здійснювати зсув або множимого, або суми часткових творів. Всі чотири способи вимагають, однак, щоб перед початком множення, принаймні, один із співмножників був представлений усіма розрядами. У зв'язку з цим не може бути виконано третю із зазначених вище вимог.

Якщо ж множимо представлено паралельним кодом, то при множенні з молодших розрядів введення множника і формування результату можна поєднувати в часі. Такий спосіб множення застосовується, наприклад, зменшення часу обчислення многочленів [1], а також при реалізації чисельних методів вирішення диференціальних рівнянь [2]. Існують також методи множення в канонічних системах числення, які дозволяють будувати ОУ, що задовольняють усім чотирма вимогами [2]. Проте третя вимога задовольняється лише у разі, коли потрібно отримати твір, представлений усіма розрядами, тобто подвоєної довжини стосовно операндам. У разі введення співмножників може бути поєднаний з процесом формування n молодших розрядів твори. Якщо ж результат повинен бути представлений n старшими розрядами, то до введення всіх розрядів співмножників жоден із цих розрядів не може бути отриманий, тобто третя вимога не виконуватиметься. Зауважимо також, що при виконанні послідовності операцій завдання округлення проміжних результатів є одним з першорядних.

При використанні методів поділу з відновленням і без відновлення залишку в канонічних системах числення [4] цифра частки кожному кроці визначається результаті порівняння величини дільника з величиною поточного залишку (першому кроці поділеного). Якщо коди поділеного і дільника представлені не всіма розрядами, то визначити їх співвідношення на першому кроці в загальному випадку неможливо і, отже, не можна

отримати достовірний розряд приватного.

Послідовність залежних операцій виконується у неавтономному (on-line) режимі за допомогою ланцюжка обчислювальних модулів (ОМ), які дозволяють поєднувати виконання залежних операцій на рівні обробки розрядів слів, наступним чином. На кожному кроці у ОМ вводиться по одному розряду операндів і формується один розряд результату, починаючи зі старших розрядів. При цьому розряд проміжного результату, отриманий на i -му кроці в одному ОМ при виконанні j -ї операції, може бути використаний на $(i+1)$ -му кроці в іншому ОМ при виконанні $(j+1)$ -ї операції. При такому режимі обчислень виконання наступної операції буде починатися відразу після отримання першого розряду результату цієї операції. Кожен ОМ починає формувати розряди операнда з деякою затримкою на p_j кроків після отримання перших розрядів операнда.

Режим роботи таких ОМ називають неавтономним (on-line), оскільки для виконання послідовності операцій необхідно кілька ОМ, які синхронно обмінюються інформацією у процесі роботи.

Завдяки розрядній передачі даних зменшується необхідна кількість зв'язків між компонентами системи порівняно з використанням паралельних пристроїв. Це особливо важливо при реалізації систем в інтегральному виконанні, оскільки зменшення числа внутрішніх та зовнішніх з'єднань підвищує надійність систем та набагато економніше використовує ресурси інтегральних схем.

2.4. Аналіз ефективності онлайн обчислень у системі з надлишковим числом

У даному аналізі було досліджено можливість підвищення ефективності обчислень у системах на кристалі за рахунок перекриття супутніх операцій та необхідного скорочення апаратних ресурсів. Розглядалися обчислювальні системи, де обчислювальні блоки безпосередньо пов'язані відповідно до

графіка потоку даних. Вузли графіка відповідають операціям, а дуги — даним. Пряма передача даних між блоками дозволяє скоротити час, що витрачається на складні процедури обміну даними.[6]

Було запропоновано алгоритм та відповідну структуру обчислювального блоку, що дає змогу виконувати дво-, три- та чотиризначні операції у режимі онлайн шляхом розрядної обробки даних, представлених у надлишковій симетричній системі числення. Показано, що використання обчислювальних блоків, що обробляють операнди порозрядно, дає змогу виконувати послідовності операцій, пов'язаних за допомогою даних, у паралельному режимі на рівні обробки розрядів операндів. У порівнянні з використанням паралельних блоків, кількість необхідних внутрішніх комутаційних схем і розпинівок мікросхеми зменшується за рахунок послідовної передачі даних. Проте швидкість обробки даних все ще висока.

Час виконання обчислення залежить не тільки від часу виконання алгоритму паралельних розгалужень, але й від часу, витраченого на обмін інформацією між обчислювальними блоками паралельної системи (ОБ). [6]

У ПЗС виходи деяких ОБ з'єднані з входами інших ОБ відповідно до графіка потоку даних (ГПД). У цьому випадку зменшується витрата часу, оскільки немає необхідності використовувати складні процедури для обміну даними.

До перших робіт, присвячених розробці онлайн методів виконання основних арифметичних операцій, що дозволяють синхронізувати процеси розрядного введення операндів та їх обробки з розрядів старшого порядку, можна віднести роботи, де показано, що розрядна обробка даних можлива за використанням резервних систем нумерації, в яких кількість дозволених значень цифрових символів перевищує базу системи нумерації. Прикладами можуть служити системи з базовими $k=2$ та розрядними символами $g \in \{-1, 0, 1\}$ (симетрична система) або $g \in \{0, 1, 2\}$ (зміщена система) [6].

Під час вирішення цієї проблеми, на кожному кроці розрахунку один операнд цифра вводиться і одна цифра результату формується. При цьому одна цифра проміжного результату, отримані в ході j -го виконання операції по i -й крок в одному ТЗ може бути використана під час $(j + 1)$ -го виконання операції на $(i+1)$ -го кроку в іншому ОБ. У цьому режимі обчислення, виконання наступної операції буде розпочато відразу після отримання першої цифри результату операції. Для узгодження діапазону зміни значень розрядів операндів на входах і виходах ОБ передбачена затримка початку формування розрядів високого порядку результату (за кількістю тактів). Наприклад, $Z^*=f(n)$ функція обчислюється із затримкою p циклів у вигляді $Z=2^{-p}f(X)$. Режим роботи таких ОБ називається режимом on-line, оскільки для виконання послідовності операцій, пов'язаних за допомогою даних, необхідно мати декілька ОБ, які будуть працювати одночасно, змінюючи цифри операндів у процесі виконання.

Час виконання послідовності операцій, пов'язаних за допомогою даних, визначається так:

$$T = \left[n - 1 + \sum_{j=1}^K (p_j + 1) \right] t_c \quad (2.7)$$

де n – кількість розрядів результату; p_j – затримка формування розрядів результату j -ї операції; t_c – час виконання розрахункового циклу [6].

Використовуючи паралельні ОБ, ті самі розрахунки можна було б виконати протягом наступного періоду часу

$$T = T_o + \sum_{i=1}^K t_i, \quad (2.8)$$

де t_i – i -й час виконання операції; T_o – додаткові витрати часу, наприклад, на введення та обмін даними.

Таким чином, якщо у ОС для виконання операцій використовуються традиційні методи машинної арифметики, то реалізацію послідовних алгоритмів не можна здійснити за час, менший за сумарний час виконання всіх операцій, а реалізацію послідовно-паралельних алгоритмів - операцій, що лежать на критичному шляху ярусно-паралельного графа алгоритму.

Підвищення швидкодії ОС при реалізації послідовних і послідовно-паралельних алгоритмів може бути досягнуто за рахунок застосування методів машинної арифметики, що дозволяють виконувати послідовності алгоритмічно нерозпаролених операцій в режимі суміщення.

Послідовність виконання операцій у неавтономному режимі ілюструється діаграмою на рис. 2.5, з якої видно, як визначається час обчислень при розпаралелюванні на рівні обробки розрядів операндів.

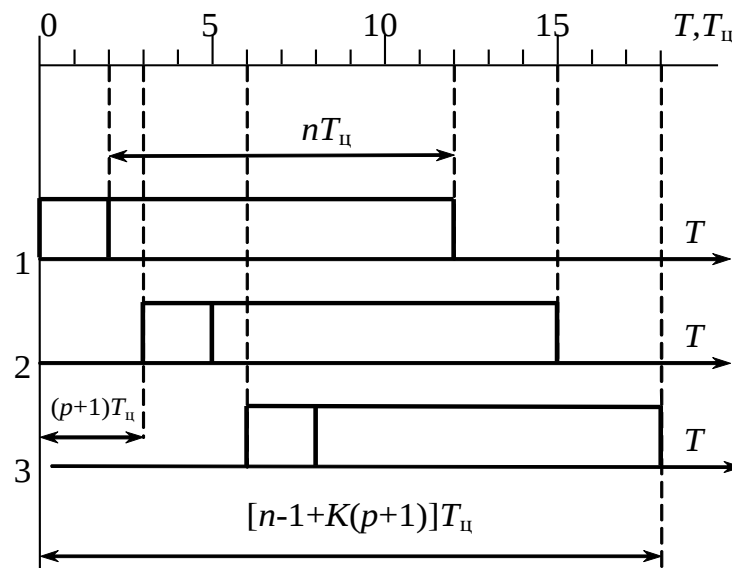


Рис. 2.5. Часова діаграма залежних операцій у 3-х ОМ

При реалізації систем із порозрядною обробкою даних спроектованих на базі квазіпаралельних обчислювальних модулів витрачається менше ресурсів ПЛІС. Проглядається економія як елементів внутрішнього ресурсу (функціональні і комунікаційні модулі, вбудовані пристрої), так й саме

виводи ПЛІС.

Завдяки цьому методу збільшується ймовірність розміщення всієї системи на одній мікросхемі. Саме проектування й реалізація системи заснована на одній мікросхемі забезпечує підвищення ефективності виконання паралельних обчислень, а саме збільшення надійності системи, зменшення енергоспоживання та габаритів обчислювальної системи.

Обмін інформаційними даними між обчислювальними модулями всередині однієї мікросхеми виконується швидше, ніж між окремими компонентами системи, які реалізовані на різних мікросхемах. Саме це дає можливість збільшити частоту тактування, що призводить до прискорення обробки даних та інформації.

2.5 Додаткові питання паралельної обробки даних

Зі зростанням продуктивності обчислювальних засобів зростає обчислювальна складність алгоритмів й вимоги щодо його швидкодії. Не дивлячись на дуже велику різноманітність архітектур побудови виконання паралельних обчислень, все більшої зацікавленості набуває проектування пристроїв на базі популярної та сучасної технології, FPGA (field-programmable gate array). Чому ж саме FPGA технологія є на сьогодні найпривабливішою порівняно з іншими архітектурами? Регулярна структура програмованої логічної інтегральної схеми (ПЛІС) дозволяє будувати на основі всього лиш однієї мікросхеми комплекси з декількома навіть незалежними пристроями.

Існує лише тільки два основних способи паралельної обробки даних на яких базуються принципи реалізації паралельних обчислень на базі ПЛІС - це паралелізм та конвеєрність, котрі відрізняються лише порядком обробки даних.

Перший принцип реалізації паралельних обчислень полягає у можливості одночасного, тобто паралельного виконання декількох

однотипних дій. Другий принцип реалізації дозволяє розбити виконання складної задачі на низку простих послідовних підзадач з поєднанням їх виконання у один й той самий час. Розглянемо детально ці принципи реалізації.[7]

Конвеєрними системами називають такі обчислювальні системи у яких декілька спеціальних обчислювальних блоків (ОБ) одночасно працюють над виконанням частин одного єдиного потоку команд. Тобто конвеєрні системи, ніби послідовно виконують підзадачі однієї задачі переходячи від одного блоку до іншого й отримуючи результати виконання кожної підзадачі для виконання наступної. Конвеєрній обробці даних властива така організація одночасних обчислень, за якої вхідні дані операції, виконується в i -тий момент часу на j -му пристрої, містять вихідні дані операції, що виконувалася в $(i-1)$ -тий момент часу на $(j-1)$ -му пристрої. Тобто після завершення обробки поточної команди на поточному етапі виконання команди переходить на наступний етап, а на даний етап обробки надходить наступна команда.

Припустимо, що кожен ступінь конвеєра виконує обробку за один такт, тоді після завантаження конвеєра (5 тактів) виконання команд буде проводитися з періодом в один такт. Вважаючи, що просте (не конвеєрне) виконання команди займає 5 тактів, маємо прискорення виконання команд у 5 разів. Даний спосіб реалізації паралельних обчислень є простішим порівняно з використанням паралелізму, бо він не вимагає спеціальних методів розпаралелювання цілої задачі на окремі гілки для їх виконання й також не потребує надання пріоритетності для кожної підзадачі бо їх виконання проходить послідовно, але при цьому кожен блок системи застосовано під розрахунок й він не є бездіяльним майже у кожен проміжок часу. Ще однією неоцінимою перевагою використання саме конвеєрного методу проектування системи є зменшення затримки перемикання між обчислювальними блоками системи за рахунок послідовного виконання підзадач й відсутності пріоритетів.

Паралельні обчислювальні системи - це системи у яких множина підзадач однієї задачі, одночасно виконується множиною різних процесорів чи пристроїв об'єднаних в одну систему.

Паралелізм - це спосіб одночасного (паралельного) виконання більш ніж однієї арифметико-логічної операції або окремої обчислювальної гілки. Можливість паралельного виконання цих операцій визначається правилом Рассела. Розглянемо детальніше це правило.[7]

Об'єкти, що програмуються, А та В є незалежними один від одного та можуть виконуватися паралельно, тобто в один й той самий час на різних пристроях чи процесорах, тільки якщо виконується наступна умова:

$$(InB \wedge OutA) \vee (InA \wedge OutB) \wedge (OutA \wedge OutB) = \emptyset, \quad (2.1.)$$

де $In(A)$ - набір вхідних змінних, а $Out(A)$ - набір вихідних змінних об'єкта А. Якщо раптом умова (2.1) не виконується у системі при виконанні паралельних обчислень, то між А та В існує пряма залежність і ці програмні об'єкти не можуть виконуватися паралельно.[7]

Наведемо приклад, у якому порушується умова (2.1) у першому своєму термі:

$$A: R = R1 + R2$$

$$B: Z = R + C$$

У цьому випадку операції А та В не можуть виконуватись паралельно, бо результат виконання обчислень А має бути використаним у якості операнда у виразі В. У разі порушення умову (2.1) у другому термі така залежність має назву - обернена:

$$A: R = R1 + R2$$

$$B: R1 = C1 + C2$$

Тут операції А та В не можуть виконуватись одночасно тобто

паралельно, бо виконання операції В викликає зміну одного з операндів в А.

Нарешті, якщо ж умова (2.1) порушується у своєму третьому термі, або частині така залежність називається конкурентною:

$$A: R = R1 + R2$$

$$B: R = C1 + C2$$

У цьому порушенні третьої частини умови (2.1) паралельне виконання операцій А та В дає невизначений результат.[7]

Збільшення розпаралелювання будь-якої системи полягає у пошуку та усуненні зазначених вище залежностей.

Найбільш спільною формою представлення цих залежностей є саме інформаційний граф задачі (ІГ).

2.5.1. Інформаційний граф задачі

Під інформаційним графом задачі розуміємо таке уявлення про структуру обчислювальної системи, при якому кожному оператору, який здійснює перетворення інформації, відповідає кожна операційна вершина інформаційного графа. Наявність або відсутність зв'язків між вершинами, які прийнято називати дугами, визначається за допомогою інформаційної залежності між операторами цієї задачі.

На стадії архітектурного проектування паралельних обчислень, може бути представлений метод виконання цих обчислень, у виді графа "підзавдання – повідомлення". Ця модель проектування дозволяє розробнику визначити такі необхідні умови для ефективного виконання паралельних обчислень, як:

– розбити цілу обчислювальну задачу на підзадачі, які обов'язково будуть мати однакову обчислювальну складність. Інколи у складних високорівневих задачах розбиття на підзадачі є досить складним завданням;

– забезпечити досить низький рівень інформаційної залежності між підзадачами. Це означає, що результат виконання кожної підзадачі має містити мінімальну кількість зв'язків між іншими підзадачами, тобто результат виконання однієї підзадачі має залежати від виконання якомога меншої кількості підзадач. Високий рівень інформаційної залежності між підзадачами призводить до збільшення часу виконання обчислень цілої задачі тим самим набагато зменшуючи ефективність виконання паралельних обчислень.[7]

На стадії виконання опису паралельних обчислень може бути використана наступна модель, яка має вигляд графа, який називається "процеси – канали", в якій замість підзадач використовують поняття процесів, а замість інформаційних залежностей – канали передачі. Дана модель допомагає архітектору паралельних обчислень виконати наступні дії, для більшої ефективності виконання паралельних обчислень:

- здійснити найбільш оптимальний розподіл підзадач по процесорам чи окремим пристроям обчислювальної системи;
- виконати детальний аналіз ефективності розробленого паралельного методу обчислення;
- забезпечити можливість контролю та управління процесом виконання паралельних обчислень.

Приклад ІГ для конкретної задачі подано на Рисунку 2.1.(а), а порядок виконання операцій у задачі подано на Рисунку 2.1.(б)

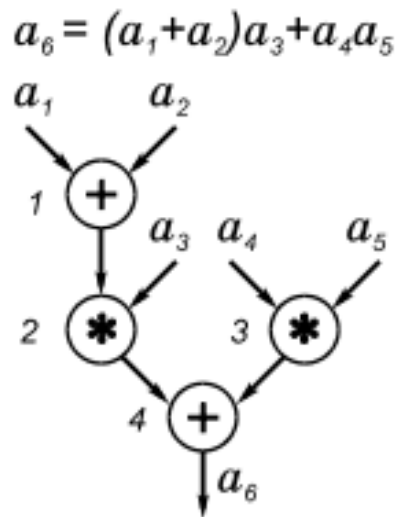


Рис 2.1.(а) - Інформаційний граф математичного виразу

1	+	<u>a_1</u>	<u>a_2</u>	2, 2
2	*	<u> </u>	<u>a_3</u>	4, 2
3	*	<u>a_4</u>	<u>a_5</u>	4, 2
4	+	<u> </u>	<u> </u>	a_6 , 2

Рис 2.1.(б) - Порядок виконання операцій у математичному виразі

Більш визначеною формою уявлення методу паралелізму є так звана ярусно-паралельна форма (ЯПФ): алгоритм обчислень подається у вигляді ярусів, причому у нульовий ярус входять тільки підзадачі (гілки), результат виконання яких не залежить один від одного, на перший ярус - підзадачі обчислення, які залежать тільки від результатів виконання операцій на нульовому ярусу, другий - від першого ярусу і т.д.[7]

Для ЯПФ характерні параметри, що тією чи іншою мірою відображають ступінь паралелізму методу обчислень:

b_i - ширина і-го ярусу;

B - ширина графа ЯПФ (максимальна ширина ярусу, тобто максимум з b_i , $i = 1, 2, \dots$);

l_i - довжина ярусу (час операцій);

L довжина графа;

ε - коефіцієнт заповнення ярусів;

θ - коефіцієнт розкиду зазначених параметрів і т.д.

2.6. Схема виконання обчислень

Поетапне виконання алгоритму побудови архітектури для виконання паралельних обчислень надано на Рисунку 2.2.[8]

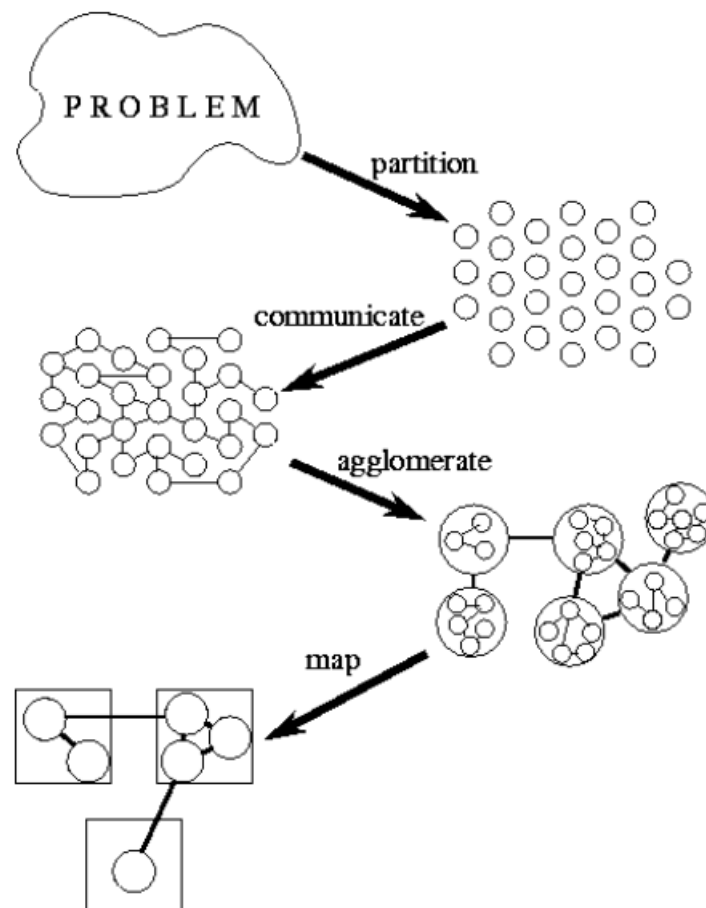


Рис. 2.2. - Спільна схема виконання паралельних обчислень

На першому етапі проєктування, який увесь показано на Рисунку 2.2., виконання паралельних обчислень починає виконуватись саме з розподілу цілої обчислювальної задачі на рівні, маючи низький рівень залежності частини, тобто підзадачі. Рівень декомпозиції задач, розподіл задач на підзадачі буває різним й має свої переваги та недоліки. Існує три типи декомпозиції обчислень, а саме:

- формування максимально можливої кількості підзадач, тобто розбиття цілої обчислювальної задачі на найменші її частини. Цей тип декомпозиції забезпечує використання гранично допустимого паралелізму задачі та ускладнює аналіз паралельних обчислень.[8]

- поділ задачі на малу кількість підзадач, але великого розміру. Приводить до чіткої схеми паралельних обчислень. Досить мала кількість підзадач вимагає надання їм певної пріоритетності, але зробити це значно простіше ніж при великій кількості малих за розміром підзадач. Недоліком цього типу проєктування є ускладнення ефективного використання великої кількості процесорів чи пристроїв. Не усі обчислювальні пристрої системи можуть бути зайнятими при досить малій кількості підзадач, тобто утворюються зайві процесори у системі, а так як кожна підзадача має великий розмір то час її виконання також буде більший у порівнянні з часом виконання меншої за розміром підзадачі.

- проміжний рівень - використання у якості елементів декомпозиції лише тільки тих підзадач, для яких методи паралельних обчислень вже відомі. Обирані такі підзадачі при такому підході називаються базовими, вони можуть бути елементарними (не допускаються до подальшого розподілу) або складовими.

На другому етапі реалізації паралельних обчислень виконується визначення зв'язків між підзадачами. Цей етап є важливим під час проєктування паралельних обчислень, тому що кожна підзадача, задля ефективного використання методу паралелізму, повинна мати якомога меншу кількість підзадач, виконання яких залежить від результату цієї підзадачі.

Тому чим менше зв'язність однієї підзадачі з іншими, тим більша продуктивність виконання паралельних обчислень.[8]

На третьому етапі проєктування паралельних обчислень виконується масштабування підзадач. Масштабування проводиться тільки тоді, коли кількість наявних підзадач, у складі цілої задачі, не збігається з числом доступних процесорів чи пристроїв у системі. Для того, щоб скоротити кількість підзадач, необхідно виконати укрупнення (агрегацію) обчислень, за наступними умовами:

- підзадачі повинні мати однакову обчислювальну складність, а обсяг та інтенсивність інформаційних взаємодій між ними повинні бути мінімально можливими, тобто інформаційний граф цілої задачі повинен мати низький рівень залежності між підзадачами;

- першими пріоритет на об'єднання повинні мати підзавдання з високим ступенем інформаційної взаємозалежності, аби ці підзадачі було легше розподілити між процесорами системи й уникнути великої часової затримки за рахунок очікування виконання залежних підзадач.

У випадку недостатньої кількості підзадач для завантаження всіх доступних процесорів чи пристроїв у системі, необхідно виконати декомпозицію обчислень.[8]

На четвертому, й останньому етапі проєктування обчислень виконується розподіл задач на вузли обчислювальної системи. На цьому етапі завершується проєктування обчислення задачі й переходимо до аналізу ефективності виконаних обчислень, та детального опису їх характеристик.

2.7. Чисельний аналіз ефективності паралельних обчислень

Принциповий момент розробки паралельних алгоритмів - це аналіз ефективності використання методу паралелізму у кожній вирішуваній задачі:

- оцінка ефективності виконання конкретних вибраних методів виконання розпаралелювання обчислень;

– оцінка найбільш можливого прискорення процесу вирішення розглянутої задачі (аналіз усіх способів виконання обчислень.

Вводимо необхідні визначення й припущення щодо оцінки

Оцінка загальної обчислювальної потужності системи з використанням методів паралелізму називається піковою продуктивністю системи. Вона визначається, як максимальна кількість операцій, яка може бути виконана системою за одну одиницю часу, при відсутності втрат часу на встановлення зв'язку та передачі інформації між ОБ. Пікова продуктивність багатопроцесорної системи визначається, як кількість функціональних пристроїв, призначених для виконання операцій з плаваючою точкою, помножене на частоту роботи процесора і число процесорів.[2]

Пікова продуктивність ще має назву теоретична продуктивність. Така назва підкреслює, що у реальній обчислювальній системі продуктивність не тільки не перевищить, а й ніколи не досягне цього порога.[2]

Наприклад, для комп'ютера з двома пристроями з плаваючою точкою та частотою 500 МГц пікова продуктивність дорівнює 1000 Mflops (1 Gflops). Ефективність використання інших функціональних пристроїв (цілочисленна арифметика, звернення до пам'яті та ін.) виявляється шляхом порівняння реально досяжною на тести продуктивності з піковою.

Реальна продуктивність – це кількість операцій, яка дійсно виконується у середньому за одиницю часу. Реальна продуктивність зазвичай сильно менше ніж пікова продуктивність. Перевищення пікової продуктивності у порівнянні з реальною пояснює, наскільки ж дана архітектура пристосована до вирішення конкретної задачі.[2]

З визначень випливає, що реальна та пікова продуктивності системи є суми відповідно до реальних та пікових продуктивностей, що становлять систему процесорів. Пікова продуктивність обчислюється як добуток $n \times f \times k$, де n – кількість операцій з плаваючою комою, що виконуються за один такт, f

– тактова частота процесора, k – кількість ядер у процесорі.

2.8. Додаткові можливості прискорення системи

Дуже важливою характеристикою оцінки ефективності виконання паралельних обчислень у системах на кристалі є прискорення.

Прискорення реалізації алгоритму на обчислювальній системі, що складається з однакових процесорів - відношення часу виконання алгоритму на одному процесорі (на одному ядрі процесора) до часу паралельного виконання.[8]

$$S_p(n) = T_1(n)/T_p(n), \quad (2.2)$$

де T_1 – час розв'язання задачі з використанням однопроцесорної системи, а T_p – час вирішення тієї ж задачі з використанням p кількості пристроїв чи процесорів у системі.

Прискорення залежить від вибору архітектором проєктування системи паралельного алгоритму та ще й від того, наскільки ж цей алгоритм пристосований до архітектури обчислювальної системи.

Ідеальним прискоренням у системі є, коли

$$S_p(n) = p, \quad (2.3)$$

тому що прискорення задачі в ідеальному випадку має дорівнювати кількості процесів чи окремих пристроїв, на які була розподілена ціла задача.

Також при наданні оцінки ефективності обчислення можемо зустріти надлінійне прискорення:

$$S_p(n) > p. \quad (2.4)$$

Воно може мати місце у характеристиках оцінки ефективності використання паралельних обчислень у системах з наступних причин:

– нелінійний характер залежності складності розв'язання задачі від обсягу оброблюваних даних;

– відмінність обчислювальних схем послідовного і паралельного методів[8]

2.9. Розрахунок ефективності використання системи

Однією з важливих характеристик оцінки ефективності виконання паралельних обчислень у системах на кристалі є ефективність використання системи обчислення. Ефективність реалізації алгоритму обчислення у системі, яка складається з (s) кількості однакових, за апаратними можливостями, процесорів - це відношення прискорення системи $(S_p(n))$ до кількості цих процесорів (s) .

$$E_p(n) = T_1(n) / (P * T_p(n)) = S_p(n) / P \quad (2.5)$$

Також у першій частині цієї рівності показано, що ефективність у системі це також визначення середньої частки виконання паралельного алгоритму, протягом якого процесори чи пристрої системи реально використовуються для вирішення задачі.

Чисельні характеристики якості паралельних обчислень часто є суперечливими: підвищення якості щодо прискорення може призвести до погіршення якості ефективності та навпаки.

2.9.1. Закон Амдала

Досягненню максимального прискорення у системі може перешкоджати існування послідовних розрахунків, які не піддаються розпаралелюванню.

Нехай f є частка послідовних обчислень у алгоритмі обробки даних, що застосовується.

Прискорення процесу виконання обчислень при використанні p процесорів обмежується величиною:

$$S_p \leq 1 / (f + (1 - f) / P) \leq S^* = 1 / f \quad (2.6)$$

Частка послідовних обчислень може бути істотно знижена при виборі придатних для розпаралелювання методів.[8]

Для великого ряду завдань частка послідовних обчислень $f = f(n)$ є спадною функцією від n , і в цьому випадку прискорення для фіксованого числа процесорів може бути збільшено за рахунок збільшення обчислювальної складності розв'язуваної задачі.

У цьому випадку прискорення $S_p = S_p(n)$ є зростаючою функцією від параметра n .

Закон Амдала визначає принципово важливі для паралельних обчислень положення:

- Чисельна характеристика прискорення залежить від потенційного паралелізму завдання (величина a) та параметрів апаратури (числа процесорів n)

- Граничне прискорення у системі на кристалі визначається властивостями задачі. Нехай, наприклад, величина a дорівнює 0,2, тоді у цьому випадку прискорення у системі не може перевищувати 5 при будь-якій кількості процесорів, тобто максимальне прискорення визначається потенційним паралелізмом завдання.[8]

Якщо система має декілька архітектурних рівнів проектування з різними формами паралелізму, то якісне загальне прискорення буде дорівнювати:

$$S_p = s_1 \times s_2 \times s_3 \quad (2.7)$$

де s_i - прискорення деякого архітектурного рівня системи.

Наведемо реальний приклад використання закону Амдала на обчислюванні числа π через інтеграл (чисельне інтегрування методом прямокутників)[8]

$$\pi = 4 \int_0^1 \frac{1}{1+x^2} \quad (2.8)$$

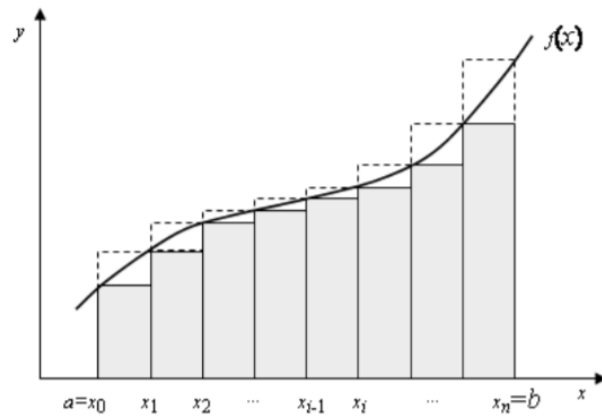


Рис. 2.3. - Чисельне інтегрування методом прямокутників

Аналіз ефективності обчислення методом прямокутників:

- n – кількість розбиття відрізка $[0;1]$
- обчислювальна складність завдання: $W = 6n$
- кількість вузлів у сітці $m = \text{round}(n/P) \leq n/P + 1$
- обсяг обчислень на одному процесорі $W_p = 6m = 6 * n / P + 6$
- час паралельного вирішення задачі (обчислення елементів + додавання) $T_p = 6 * n / P + 6 + \log_2 P$
- прискорення $S_p = T_1 / T_p = 6n / (6 * n / P + 6 + \log_2 P)$
- ефективність $E_p = S_p / P = 6n / (6 * n + 6P + P \log_2 P)$

ВИСНОВОК ДО РОЗДІЛУ 2

У розділі показана можливість виконання послідовності залежних операцій у режимі часткового суміщення в часі. Для цього необхідно використовувати неавтономні методи виконання арифметичних операцій, які забезпечують суміщення у часі порозрядного введення операндів, обробки та виведення. Такий режим роботи арифметичних пристроїв відбувається в надлишкових системах числення

За рахунок порозрядної передачі даних між операційними модулями зменшується необхідна кількість вивидів мікросхем, що, в свою чергу, дає можливість зменшення внутрішніх ресурсів мікросхем.

РОЗДІЛ 3

ПОБУДОВА АЛГОРИТМІВ ВИКОНАННЯ ОПЕРАЦІЙ В НЕАВТОНОМНОМУ РЕЖИМІ

3.1. Методика виведення алгоритмів у неавтономному режимі

Розглянемо алгоритм обчислення функції в двійковій надлишковій симетричній системі числення з цифрами $\{-1, 0, 1\}$ в неавтономному режимі.

Нехай функція має вигляд $Z = 2^{-p}F(X)$, де $X=\{X_i\}$ – множина аргументів (операндів), причому, $i=1,2,\dots,n$.

Ми визначимо коди операндів та результату, які складаються лише з i цифр після крапки через X_i, Z_i , цифри цих кодів у позиції, зі значенням ваги 2^{-i} , позначем як x_i та z_i .

Якщо на i -м кроці цифра z_i вибирається з урахуванням наступних умов

$$Z_i - 2^{-i} \leq 2^{-p}F(X_i) < Z_i + 2^{-i}, \quad (3.1)$$

то після введення n цифр ми отримаємо значення $Z_n = Z$ з мірою неточності, абсолютне значення якого не перевищує 2^{-n} . Значення p – затримка отримання разрядів результату у кількості кроків.

Визначення R_i як $R_i = 2^i[2^{-p}F(X_i)]$ умова (3.1) може бути представлена таким чином

$$Z_i - 2^{-i} \leq R_i < Z_i + 2^{-i} \quad (3.2)$$

Більше того, R_i може бути визначено за формулою

$$R_i = H_i - D, \quad (3.3)$$

Де D – прирощення функції від аргументів на i -му кроці.

Якщо цифри результату залежать від трьох діапазонів значення H_i , то правило, згідно з яким на i -му кроці можна вибрати цифру результату, формулюється так

$$z_i = \begin{cases} -1, & \text{if } H_i < -2^{-2}; \\ 0, & \text{if } -2^{-2} \leq H_i < 2^{-2}; \\ 1, & \text{if } H_i \geq 2^{-2}. \end{cases} \quad (3.4)$$

Одержана формула забезпечує формування цифр результату у множенні $\{-1, 0, 1\}$ тобто в надлишковій симетричній системі числення.

Система числення може бути симетричною або зміщеною, наприклад з цифрами $\{0, 1, 2\}$

3.2. Приклад використання зміщеної надлишкової системи числення

Розглянемо одержання алгоритму обчислення функції, яка використовується для перетворення координат у режимі реального часу. Для дрібнозернистих алгоритмів ефективним може бути перетворення інформації на низькому рівні паралелізму, наприклад, на рівні обробки машинних слів та розрядів операндів.

Однак, швидкість обробки інформації залежить не тільки від часу виконання операцій в паралельних гілках, але і від витрат часу на обмін інформацією між гілками, тобто між обчислювальними модулями (ОМ) паралельної системи. Зменшити витрати часу на обмін даними дозволяє використання паралельних обчислювальних систем з безпосередніми зв'язками між обчислювальними модулями (ПСБЗ) [3, 4]. У таких системах виходи одних ОМ підключаються до входів інших ОМ відповідно графу потоку даних (ГПД). В процесі обчислень дані пересилаються безпосередньо від одних ОМ до інших, перетворюючись на кожному кроці відповідно операціям, заданим вершинами ГПД. В такому випадку відсутні складні процедури пересилки паралельних кодів даних між ВМ, тобто зменшуються витрати часу на обмін даними між ними.

Для скорочення апаратних витрат в ПСБЗ доцільно використовувати ОМ, що дозволяють поєднувати процеси порозрядного введення операндів, їх обробки і порозрядного формування результату зі старших розрядів. Такий режим можна забезпечити з використанням симетричної або зміщеної надлишкової системи числення.

У роботі [3] показано перевагу зміщеної системи, пов'язаної зі зменшенням необхідного обладнання при реалізації функцій, які мають тільки додатні приватні похідні.

Нижче розглядається метод обчислення функції квадратного кореня із суми квадратів з використанням зміщеної системи числення. Метод можна використовувати, наприклад, в процесі перетворення декартових координат в полярні.

Запишемо функцію у вигляді $Z = 2^{-p} \sqrt{X^2 + Y^2}$. Тут X і Y є додатними правильними дробами виду $X = \sum_1^n x_i k^{-i}$ та $Y = \sum_1^n y_i k^{-i}$, де $k = 2$ – основа системи числення; $x_i, y_i \in \{0, 1, 2\}$ – цифри операндів у двійковій зміщеній системі числення. Коди чисел X, Y та Z , що містять тільки i старших розрядів позначимо так: X_i, Y_i та Z_i . Параметр p показує кількість циклів затримки формування першого розряду результату z_1 після надходження перших розрядів операндів x_1 и y_1 . Для одержання результату з урахуванням затримки необхідно виконати $m = n + p$ циклів. Результат можна записати у вигляді $Z = \sum_1^m z_i k^{-i}$, де $z_i \in \{0, 1, 2\}$. Використовуємо розглянуту вище методику.

Якщо в кожному i -му ($i = \overline{1, m}$) циклі цифру результату вибирати таким чином, щоб виконувалась нерівність

$$Z_i^2 \leq 2^{-2p} (X_i^2 + Y_i^2) < (Z_i + 2^{-i})^2, \quad (3.5)$$

то після m циклів буде одержано значення функції $Z_m = Z$.

Співвідношення (3.5) можна привести до вигляду

$$0 \leq (2^{-2p} (X_i^2 + Y_i^2) - Z_i^2) 2^{i-1} < Z_i + 2^{-i-1}. \quad (3.6)$$

Введемо позначення

$$R_i = (2^{-2p}(X_i^2 + Y_i^2) - Z_i^2)2^{i-1} \quad (3.7)$$

та подамо (2) з урахуванням (3) у вигляді:

$$0 \leq R_i < Z_i + 2^{-i-1}. \quad (3.8)$$

Перед початком обчислень, коли $i=0$ та $R_0=0$, нерівність (3.9) виконується. Нехай вона виконується і на $(i-1)$ -му кроці, тобто:

$$0 \leq R_{i-1} < Z_{i-1} + 2^{-i}. \quad (3.10)$$

Враховуючи, що $X_i = X_{i-1} + x_i 2^{-i}$, $Y_i = Y_{i-1} + y_i 2^{-i}$ та $Z_i = Z_{i-1} + z_i 2^{-i}$, із виразу (3.7) знайдемо

$$R_i = 2R_{i-1} + 2^{-2p}(x_i X_{i-1} + y_i Y_{i-1}) + 2^{-2p-i-1}(x_i^2 + y_i^2) - Z_{i-1}z_i - z_i^2 2^{-i-1}. \quad (3.11)$$

Позначивши

$$H_i = 2R_{i-1} + 2^{-2p}(x_i X_{i-1} + y_i Y_{i-1}) + 2^{-2p-i-1}(x_i^2 + y_i^2), \quad (3.12)$$

отримаємо

$$R_i = H_i - Z_{i-1}z_i - z_i^2 2^{-i-1}. \quad (3.13)$$

Тоді вираз (3.7) з урахуванням (3.8) можна подати у вигляді

$$Z_{i-1}z_i + z_i^2 2^{-i-1} \leq H_i < Z_{i-1}z_i + Z_{i-1} + z_i 2^{-i} + z_i^2 2^{-i-1} + 2^{-i-1}. \quad (3.14)$$

Виходячи з формули (3.14), визначимо правило вибору z_i в i -му циклі ізомножини значень $\{0, 1, 2\}$:

$$z_i = \begin{cases} 0, & \text{якщо } 0 \leq H_i < Z_{i-1} + 2^{-i-1}; \\ 1, & \text{якщо } Z_{i-1} + 2^{-i-1} \leq H_i < 2Z_{i-1} + 2^{-i+1}; \\ 2, & \text{якщо } 2Z_{i-1} + 2^{-i+1} \leq H_i < 3Z_{i-1} + 2^{-i+2} + 2^{-i-1}. \end{cases} \quad (3.15)$$

Для отримання правильного результату необхідно утримувати значення H_i в інтервалі

$$0 \leq H_i < 3Z_{i-1} + 2^{-i+2} + 2^{-i-1}. \quad (3.16)$$

Оскільки значення H_i залежать від p , що слідує з (3.7), вибором значення p можна забезпечити виконання співвідношення (3.12) для заданого діапазону зміни операндів.

В кожному i -му циклі визначається чергова цифра z_i результату наступним чином.

За формулою (3.12) знаходимо значення H_i , виходячи з співвідношень (3.15) визначаємо цифру результату, а значення R_i для наступного циклу знаходимо за формулою (3.13).

Для 8-розрядних операндів $X = 0.10020211$ та $Y = 0.10000220$ стани перемінних в циклах зведено в табл. 3.1. Після виконання $m = n + p = 8 + 3 = 11$ циклів одержали результат $Z = 0,00011011101$. Дійсне значення функції $Z^* = \sqrt{X^2 + Y^2} = 0,11011101$ формується із запізнюванням на 3 циклу. Процес ілюструється табл. 3.1.

Таблиця 3.1

i	x_i	y_i	H_i	Умова, що виконується	z_i	R_i
1	1	1	00.0000001000000000	$0 \leq H_1 < Z_0 + 2^{-2}$	0	00.0000001000000000
2	0	0	00.0000010000000000	$0 \leq H_2 < Z_1 + 2^{-3}$	0	00.0000010000000000
3	0	0	00.0000100000000000	$0 \leq H_3 < Z_2 + 2^{-4}$	0	00.0000100000000000
4	2	0	00.0001010010000000	$Z_3 + 2^{-5} \leq H_4 < 2Z_3 + 2^{-3}$	1	00.0000110010000000
5	0	0	00.0001100100000000	$Z_4 + 2^{-6} \leq H_5 < 2Z_4 + 2^{-4}$	1	00.0000010100000000
6	2	2	00.0001001101000000	$0 \leq H_6 < Z_5 + 2^{-7}$	0	00.0001001101000000
7	1	2	00.001011010111010	$Z_6 + 2^{-8} \leq H_7 < 2Z_6 + 2^{-6}$	1	00.000101000111010
8	1	0	00.001010111001001	$Z_7 + 2^{-9} \leq H_8 < 2Z_7 + 2^{-7}$	1	00.000100010001001
9	-	-	00.001000100010010	$Z_7 + 2^{-10} \leq H_9 < 2Z_8 + 2^{-8}$	1	00.000001101110010
10	-	-	00.000011011100100	$0 \leq H_{10} < Z_9 + 2^{-11}$	0	00.000011011100100
11	-	-	00.000110111001000	$Z_{10} + 2^{-12} \leq H_{11} < 2Z_{10} + 2^{-10}$	1	00.0000000000000000

У системах з порозрядною передачею даних між обчислювальними модулями потрібен менший ресурс ПЛІС в порівнянні з системами, в яких передача даних між модулями проводиться паралельним кодом. Економляться як функціональні елементи всередині ПЛІС, так і зовнішні виводи (піни) мікросхем. Це дає можливість розмістити на одній мікросхемі більший апаратний ресурс [5]. Реалізація обчислювальної системи на одній мікросхемі забезпечує підвищення її надійності, зменшення енергоспоживання та габаритів. Таким чином, одержані результати підтверджують ефективність методів порозрядної обробки інформації зі старших розрядів у системах на базі ПЛІС.

3.3. Побудова систем з безпосередніми зв'язками між обчислювальними модулями

Узагальнену функціональну модель систем класу ПСБЗ (паралельної системи з безпосередніми зв'язками між модулями) можна подати у вигляді

$$SYS = \langle DI, DO, M, F, C \rangle,$$

де DI – множина вхідних даних; DO – множина результатів; M – множина ОМ; F – система функцій перетворення даних; C – характеристика засобів комутації.

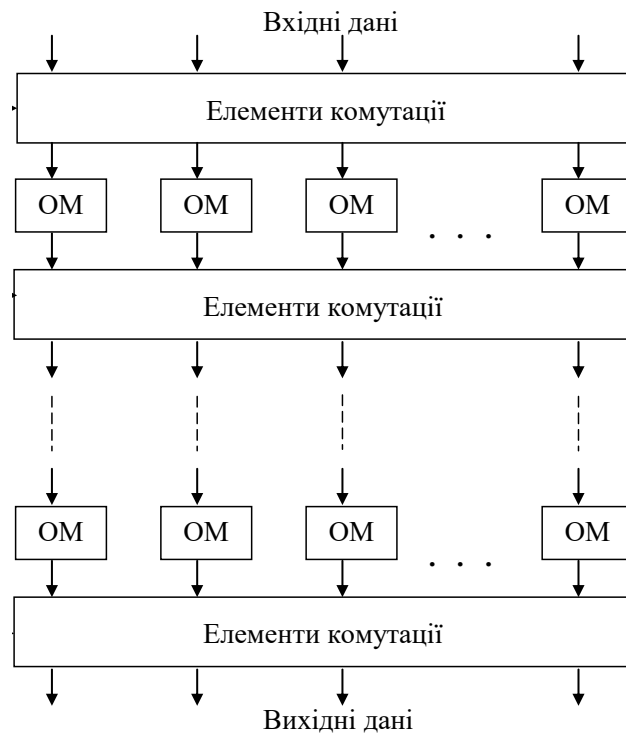


Рис. 3.1. Структура системи класу ПСБЗ

Система функцій (F) представлена у вигляді алгоритмів виконання операцій в обчислювальних модулях. Характеристика (C) визначає можливості взаємодії між ОМ у просторі системи.

На Рис.3.1. показано приклад організації потокових систем із безпосередніми зв'язками для реалізації ярусно-паралельних алгоритмів, детальний опис яких надано у 2-гому розділі даної роботи. ОМ налагоджуються для виконання спеціальних операцій, а певні елементи комутації слугують для забезпечення якісного зв'язку між обчислювальними модулями.[9]

При використанні у системах квазіпаралельних ОМ відбувається суміщення процесів порозрядного введення операндів, їх обробки та виведення результату. Це надає потенційну можливість системі виконувати залежні операції (результат однієї операції є операндом для іншої операції) у режимі часткового суміщення в часі.

На кожному кроці виконання обчислень у системах на кристалі в ОМ вводиться по одному розряду операндів, починаючи зі старших, і формується один розряд результату. При цьому розряд проміжного результату, отриманий на i -му кроці в одному обчислювальному модулі під час виконання j -ї операції, може бути використаний ще на $(i+1)$ -му кроці в іншому обчислювальному модулі під час виконання $(j+1)$ -ї операції. При писаному вище режимі виконання обчислень, виконання кожної наступної операції буде починатися відразу після отримання саме першого розряду результату цієї операції, а не після завершення попередньої операції. Для виконання послідовності цієї операцій є необхідною умовою, сумісна робота декількох ОМ, які обмінюються інформаційними даними у процесі роботи. [9]

ПСБЗ містить однотипні обчислювальні модулі. Для обчислення степені чисел використовується метод бінарного дерева [9].

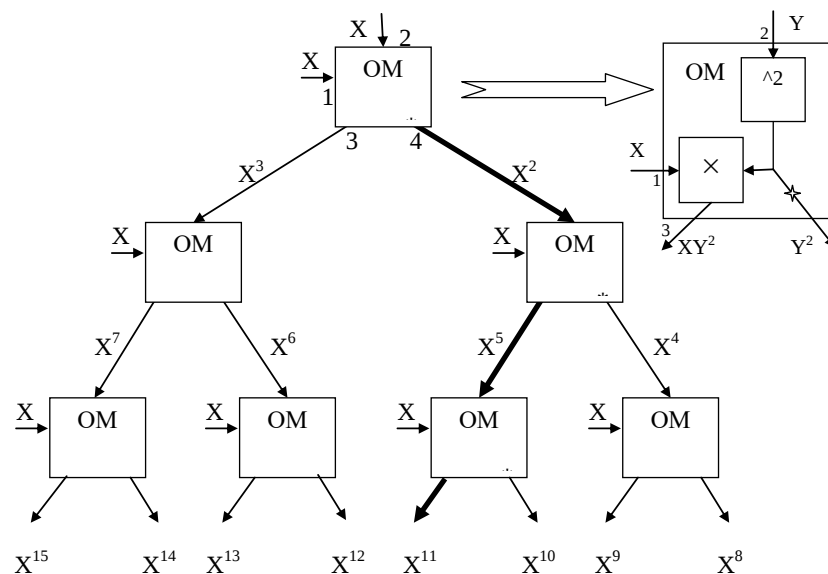


Рис.3.2. - Дерево однотипних модулів

Виконання операцій у неавтономному режимі починаючи саме зі старших розрядів можливе, як у зміщених, так й симетричних надлишкових системах числення. Усі розряди обробляються однаково, що дає можливість прискорити виконання операції у системах на кристалі. Якщо операнди й

можуть мати різні знаки, то доцільно виконувати операції у симетричних системах числення.

Обчислювальні модулі ПСБМ повинні обмінюватись даними в процесі обробки інформації. Система будується на основі потокового графа, вершини якого відповідають операціям, а дуги – даним. ОМ зв'язані між собою безпосередньо, тобто дані з одного ОМ передаються в другий ОМ без затримки, наприклад, у загальній пам'яті. Формальні методи побудови структури ПСБС відомі.

Як було показано в розділі 1 час обчислень залежить від тривалості циклів та затримки разрядів результату. Для прискорення обчислень доцільно використовувати ОМ, що реалізують багатомісні операції []. Такий ОМ дає меншу затримку формування розрядів результату.

Розглянемо це на прикладі ОМ для обчислення раціональної функції $Z = 2^{-p}(AB + C)/D$ для симетричної системи числення з цифрами $\{-1, 0, 1\}$ і значеннями операндів: $|A| \leq 1, |B| \leq 1, |C| \leq 1$ та $2^{-1} \leq D \leq 1$.

Аналогічно попередньому прикладу (3.1), побажаємо, щоб похибка визначалась співвідношенням

$$Z_i - \delta_i \leq 2^{-p}(A_i B_i + C_i)/D_i < Z_i + \delta_i,$$

где $2^{-(i+1)} \leq \delta_i < 2^{-i}$. Тоді для визначення прирощення функції в кожному циклі одержимо формулу

$$H_i = 2H_{i-1} - 2D_{i-1}z_{i-1} + 2^{-p}(A_{i-1}b_i + B_{i-1}a_i + a_i b_i 2^{-i} + c_i) - Z_{i-2}d_i - z_{i-1}d_i 2^{-i+1} \quad (3.16)$$

Спрощена структура ОМ показана на рис. 3.2.

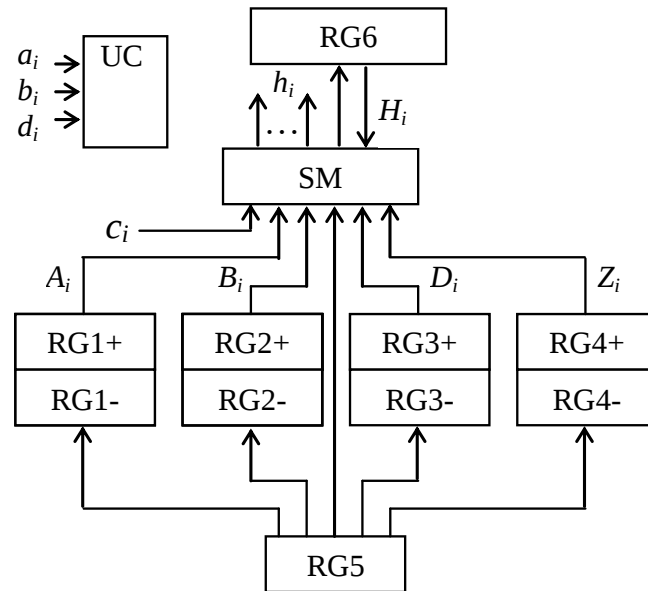


Рис. 3.1. Обчислювальний модуль

На кожному кроці обчислення вводиться одна цифра кожного операнда і формується одна цифра результату z^i . Для цього використовуються формули (3.16), (3.4) і (3.3).

Кожний ОМ можна налаштувати на обчислення різних функцій (рис. 3.2).

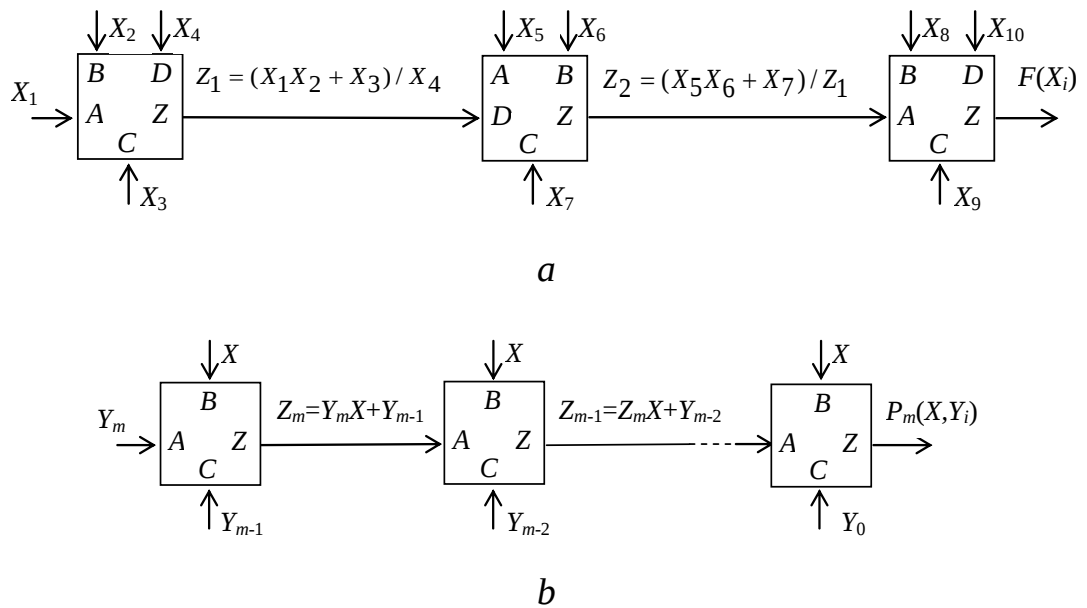


Рис. 3.2. Варіанти з'єднання ОМ в системі для реалізації різних функцій: а — для раціональної функції; б — для обчислення полінома

В ОМ на рис.3.1 операнди записуються у регістри з відповідною назвою, УС – пристрій управління. Регістр Rg5 формує цифру з вагою 2^{-i} в i -му циклі, щоб отримати значення кодів у відповідних регістрах відповідно до формул (3.4). Формування коду в регістрах управляється сигналами пристрою управління. Так, наприклад, A_i накопичується в RG1 + i RG1-. Цифри z_i визначаються за допомогою верхніх цифр аналізу коду на виході суматора СМ. Код H_i формується на СМ і зберігається в RG6. На СУ входах і виходах цифр -1, 0 і 1 кодуються двома бітами по двох провідниках.

Як видно з рис. 3.2, з'єднавши певним чином входи і виходи деяких ОБ, можна обчислити різні раціональні функції багатьох аргументів. Наприклад, схема на рис.3.2а обчислює функцію

$$F(X_i) = \frac{X_4 X_5 X_6 X_8 + X_4 X_7 X_8 + X_1 X_2 X_9 + X_3 X_9}{X_1 X_2 X_{10} + X_3 X_{10}}.$$

Для обчислення функції $Z=2^{-p}(AB+C)/D$ використовується один ОБ замість послідовності з трьох для множення, додавання та ділення відповідно. Для вищезгаданих ОБ значення $p \in 2, 2 \text{ і } 3$ відповідно. ОМ на рис. 3.2 має $p=5$, тобто дає вигравш в швидкості обчислень.

На рис. 3.2б б показана схема підключення ОБ для обчислення поліному

$$P_m(X, Y_i) = Y_m X^m + Y_{m-1} X^{m-1} + \dots + Y_1 X + Y_0$$

за методом Горнера першого порядку.

Ця схема дозволяє обчислювати різні функції методом поліноміального наближення. У цьому випадку кожен ОБ виконує одну ітерацію за методом Горнера, тобто обчислює функцію $Z = 2^{-p}(AB+C)$. Для даної тримісної операції затримку формування результату можна звести до $p=3$ циклів. Якщо немає необхідності переналаштувати

схему поліноміального обчислювача для виконання іншої операції, то можна виключити регістри збереження коду Z_i та D_i зі схеми, наведеної на рис. 3.2. Обчислення поліному виконується за час

$$T = [m(p + 1) + n - 1]t_c \cong (4m + n - 1)t_c,$$

де m – степінь поліномуа, n – розрядність.

У розглянутому випадку затримка $p=3$ для кожного каскаду. Наприклад, для обчислення полінома 5-го ступеня з 64-значними операндами нам знадобиться 83 цикли. Загалом тривалість одного циклу залежить від часу H_i обчислення змінної за спрощеною формулою (3.4) з урахуванням $D=1$. Визначається здебільшого шляхом додавання 5 доданків з використанням дерева суматорів. (з сучасними апаратними компонентами це займає наносекунди) [6].

Якщо для виконання послідовності операцій, пов'язаних за допомогою даних, ми використовуємо паралельний ОБ, який працює з класичною двійковою системою числення, час виконання буде визначатися за формулою (1.9).

Використовуючи квазіпаралельний ОБ, можна одночасно реалізувати процес розрядного введення операндів та їх обробки. Це дозволяє прискорити обчислення в реальному часі при введенні операндів порозрядно.

У цьому аналізі розглянуто можливість підвищення ефективності обробки даних за рахунок перекриття операцій в режимі online з розрядним обміном даними між ОБ.

Розглянуто алгоритм і структура ОБ, які можуть бути налаштовані на виконання операцій з різною кількістю операндів. Використання запропонованого ОБ дозволяє прискорити обчислення в режимі онлайн.

Всередині однієї мікросхеми обмін даними здійснюється швидше, ніж між системними блоками, побудованими на різних мікросхемах. Це дає нам можливість збільшити тактову частоту і, отже, прискорити обробку даних.

Тому область застосування систем може бути розширена в режимі реального часу, коли на обробку даних накладаються тимчасові обмеження.

Слід зазначити, що використання ОБ, що працюють в режимі online в циклі автоматичного керування об'єктами та процесами, дає додатковий позитивний ефект завдяки онлайн арифметиці. Завдяки отриманню результату розрахунку, починаючи з розрядів старшого порядку, ми маємо потенційну можливість почати генерувати керуючі дії для об'єкта під час обчислення першого результату старших розрядів, а потім отримати точне значення інтенсивності керуючої дії під час кожного наступного розряду результату. У деяких випадках це дозволяє підвищити якість управління об'єктом.

Таким чином. у цьому підрозділі було розглянуто паралельні обчислення, їх визначення, переваги та недоліки використання порівняно з послідовними обчисленнями. А також системи на кристалі їх визначення, призначення, проєктування та розробка.

Далі буде описано критерії ефективності паралельних обчислень, проаналізовано ефективність методів паралельних обчислень, принципи яких було описано у цьому розділі, визначено чисельно їх переваги та недоліки.

ВИСНОВОК ДО РОЗДІЛУ 3

У цьому розділі магістерської дисертації було наведено метод скорочення необхідного ресурсу ПЛІС для більш ефективного виконання паралельних обчислень у системах на кристалі. Метод дозволяє за рахунок зменшення функціональних та комутаційних ресурсів, що необхідні для побудови обчислювальної системи, зменшення споживаної потужності обчислювальної системи, підвищити надійність та дає потенційну можливість прискорити реалізацію дрібнозернистих обчислювальних алгоритмів у системах на кристалі.

РОЗРОБКА СТАРТАП ПРОЕКТУ

4.1.Опис ідеї стартапу

Паралельні обчислення у системах на кристалі є не новим методом обчислення високопродуктивних задач, але як більшість сучасних необхідних технологій для обчислення вимагають вдосконалення, тобто підвищення ефективності їх виконання. У другому розділі даної дипломної роботи було описано характеристики за якими визначається ефективність зроблених обчислень та висунуто проблеми та недоліки використання, які є можливість покращити. У третьому розділі було надано вирішення декількох недоліків паралельних обчислень у системах на кристалі, які допоможуть значно підвищити ефективність виконання паралельних обчислень. На даному етапі проектування паралельних обчислень у системах на кристалі, навіть з усуненими деякими недоліками обчислювальної системи, не може виступати як повноцінний продукт, однак, за умови подальшої розробки можна спланувати ідеї, скласти стратегію розвитку та дій для подальшого просування та створення конкурентно-спроможного продукту.

У даному розділі магістерської дисертації буде описано розробку стартапу для методу підвищення ефективності паралельних обчислень на розрядному рівні у системах на кристалі, який можна використовувати та реалізовувати на елементній базі ПЛІС. У Таблиці 4.1 показано зміст запропонованої ідеї, напрямки для застосування, переваги та вигоди у цих напрямках та різниця цієї ідеї від існуючих рішень.

Зміст та ідея розроблюваного стартап-проекту

Зміст та ідея стартапу	Можливості та напрямки застосування	Переваги для користувача
Методи підвищення ефективності паралельних обчислень на розрядному рівні у системах на кристалі	Використання розробленого методу підвищення ефективності у проектуванні паралельних обчислень у системах на кристалі	Можливість швидше та точніше розрахувати високорівневу та складну задачу
	Використання отриманої інформації у наступних наукових роботах, які пов'язані з використанням та проектуванням паралельних обчислень у системах	Можливість для користувача та розробника підвищити чи покращити ефективність використання паралельних обчислень у системах
	Зменшення споживаної потужності та енергоресурсів під час обчислень у системі	Зменшується температура системи тим самим утворюючи запобігання високого нагрівання у системи, яке часто є причиною збою у роботі системи

Аналіз потенційних техніко-економічних переваг розробленого автором даної роботи методу (чим відрізняється від вже існуючих методів використання паралельних обчислень у системах на кристалі).

Існуючі методи детально описані у першому розділі даної роботи, їх три й вони саме такі: метод виконання послідовності нерозпаралелюваних операцій у режимі суміщення (Метод №1), методи та засоби поєднання операцій на рівні паралельної обробки розрядів операндів (Метод №2), аналіз ефективності онлайн обчислень у системі з надлишковим числом (Метод №3).

Таблиця 4.2.

Визначення слабких, сильних та нейтральних характеристик ідеї

№	Технічні та економічні характеристики розроб. ідеї-проекту	Розроб. проект	Існуючі Методи		
			1	2	3
1	Зменшена споживча потужність та затрати енергоресурсів системи	+	-	-	-
2	Використання методу паралелізму	+	-	+	+
3	Використання графу послідовно паралельної обробки	+	+	-	-

Таблиця 4.2. (продовження)

№	Технічні та економічні характеристики розроб. ідеї-проекту	Розроб. проект	Існуючі Методи		
			1	2	3
4	Використання режиму часткового суміщення	+	+	-	-
5	Розпаралелювання обчислень на рівні обробки машинних слів	+	-	+	+
6	Зменшені апаратні ресурси ПЛІС	+	-	-	+

У таблиці 4.2 було надано слабкі, сильні та нейтральні характеристики методів паралельних обчислень у системах на кристалі. Як видно із таблиці, основною перевагою розробленого методу є можливість зменшення апаратних ресурсів системи, тобто ПЛІС й зменшення споживаної потужності системи.

4.2. Технологічний аудит ідеї

У таблиці 4.3 показано обрані технології для реалізації паралельних обчислень на розрядному рівні у системах на кристалі. Проектування та розробка паралельних обчислень у системах на кристалі оснований на елементній базі ПЛІС.

Можливість реалізації стартап проекту

№	Головна ідея для стартап проекту	Необхідні технології	Наявність	Доступність
1	Метод підвищення ефективності паралельних обчислень на розрядному рівні у системах на кристалі	Використання ресурсів BRAM і DSP48E1	Є в наявності	Доступна безкоштовно
2		Програмована логічно-інтегрована схема (ПЛІС)	Є в наявності	За ринковою ціною
3		Алгоритм побудови та розробки паралельних обчислень	Є в наявності	Доступна безкоштовно

4.3.Аналіз ринкових можливостей запуску стартап-проекту

У цьому пункті буде визначено ринкові можливості, які можна використати під час ринкового впровадження даного проекту, а також ринкові загрози, що можуть виникнути під час реалізації проекту. Цей аналіз дозволяє спланувати напрямки розвитку проекту враховуючи стан ринкового середовища та потреб клієнтів. Таблиця 4.4. показує стан потенційного ринку для розроблюваного стартапу.

Таблиця 4.4.

Характеристика існуючого ринку стартап-проекту

№	Показник	Характеристика
1	Кількість головних гравців, од	< 10
2	Динаміка ринку (якісна оцінка)	У сфері паралельних обчислень у системах на кристалі на етапі стрімкого зросту
3	Наявність обмежень для входу (вказати характер обмежень)	Обмежень майже немає, окрім складності даної технології
4	Специфічні вимоги до стандартизації та сертифікації	Відсутні

Основна характеристика потенційних клієнтів

№	Потреба у стартап- проекті	Потенційні клієнти	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Зацікавленість у більшій швидкості виконання обчислень	Користувачі, що зацікавлені в покращенні ефективності використанн я обчислень у системах на кристалі	Вид та складність задачі й апаратні можливості використаног о обладнання	Швидкість та зручність розпаралелюванн я обчислень у системі
2	Необхідність у зменшенні енергоресурсі в та споживаної потужності	Клієнти із проблемами великого нагрівання систем під час обчислення	Вид та складність задачі й апаратні можливості використаног о обладнання	Менше споживана потужність й енергетичні ресурси порівняно із послідовними

Основні фактори загроз при виході на ринок

№	Фактор	Зміст	Реакція
1	Висока ціна обладнання	Ціна на ринку на обладнання для даного методу досить велика	Закупівля обладнання для більших підприємств, чи використання коли отриманий гонорар за вирішені задачі на цьому обладнанні перевищать його собівартість
2	Існуючі методи виконання паралельних обчислень у системах на кристалі	Ринок має цінні напрацювання у цій сфері, й актуальність та популярність паралельних обчислень на розрядному рівні на кристалі зумовлює досить велику конкуренцію новим розробленим методам	Складний спосіб подачі та реалізації методів підвищення ефективності паралельних обчислень на розрядному рівні

Основні фактори можливостей для розроблюваного стартап-проекту

№	Фактор	Зміст	Реакція
1	Залучення клієнтів	Склавши чітку маркетингову кампанію, залучити клієнтів	Спланувати маркетингову кампанію, провести презентацію методу та можливостей, що він має
2	Залучення інвесторів	Створення бізнес-плану допоможе залучити інвестиції у проект, що дозволить найняти розробників для реалізації цього методу у промислових масштабах	Підвищення ефективності паралельних обчислень на розрядному рівні у системах на кристалі, збільшать об'єми та швидкість обчислювальних задач

Аналіз конкуренції на ринку

Основні властивості конкурентів	Проявлення властивості	Вплив
Схожість методу з іншими аналогами	На ринку вже існують методи паралельних обчислень у системах на кристалі й вони мають свою цінність й унікальність	Виконувати експерименти зі створенням нових методів
Доступність до технологій, що використовуються у проекті	Майже кожен із розробників має доступ до необхідного обладнання для реалізації даного методу	Реагувати на вихід нових методів паралельних обчислень у системах на кристалі на ринок, та їх функціонал

Основні фактори конкурентоспроможності проекту

№	Фактор конкурентоспроможності	Обґрунтування
1	Простота та зручність розробленого методу	Враховуючи недоліки існуючих методів та досвіду розробників та архітекторів паралельних обчислень у системах на кристалі, можна створити більш простий та зручний метод використання паралельних обчислень
2	Бета-тестування	Впровадження бета-тестування для виявлення найбільш важливих нових можливостей аби максимального адаптувати продукт до потреб кінцевого споживача
3	Унікальне зменшення споживаної потужності й енергоресурсів	Часто системи на кристалі під час виконання обчислень нагріваються дуже сильно, що призводить до вимкнення компонентів системи й збою виконання обчислень. Даний метод усунув цю проблему.

SWOT-аналіз стартап-проекту

Сильні сторони: - Зручний та простий у використанні метод; - Врахування досвіду розробників попередніх методів паралельних обчислень у системах на кристалі; - Бета-тестування для виявлення недоліків та побажань клієнтів	Слабкі сторони: - Метод має конкуренцію порівняно із попередніми методами розпаралелювання обчислень - Наявність більших можливостей методу в порівнянні з методами конкурентів - Фінансова обмеженість, що може затримати вихід на ринок
Можливості: - Отримати можливість продати метод підвищення ефективності паралельних обчислень у системах на кристалі	Загрози: - Відсутні

Альтернативні сценарії для впровадження розроблюваного стартап-проекту

№	Можлива альтернатива	Ймовірність отримання ресурсів	Термін реалізації
1	Публікація даного методу на наукових відкритих ресурсах	Дуже низька. Основними ресурсами будуть пожертви від вдячних користувачів цього методу	Миттєво
2	Представлення методу на наукових конференціях та презентаціях	Висока. Представленням на конференціях можна поширити інформацію про розроблений метод та залучити платоспроможних інвесторів	Від півроку до 5 років

Переваги та недоліки розроблюваного стартап-проекту у порівнянні з конкурентами

№	Профіль цільової групи	Бажання та готовність сприймати продукт	Попит у аудиторії	Конкуренція на ринку	Складн. виходу на ринок
1	Приватні компанії	Не всі готові прийняти	Не для всіх є необхідним	Доволі низька конкуренція	Складна
2	Промислові організації	Не всі готові прийняти одразу	Необхідний вже	Доволі низька конкуренція	Середня
3	Дослідницькі програми та університети	Часто готові прийняти	Дуже необхідний	Доволі низька конкуренція	Легка
Висновок: надано перевагу у роботі із дослідницькими програмами та університетами, однак розглядаються пропозиції від усіх цільових груп					

Базова стратегія розвитку для стартап-проекту

№	Альтернатива	Обраний спосіб для розвитку та впровадження стартапу	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Просування додатку за допомогою реєстрації на дослідницьких подіях та участі у конференціях, презентаціях, тощо	Пошук сучасних наукових конференцій та дослідницьких подій	Безкоштовна чи мало-вартісна реєстрація на участь у наукових подіях	Впровадження нового функціоналу та можливостей розробленого методу згідно відгуків розробників й архітекторів, які спробували даний метод на практиці

Визначення базової стратегії конкурентної поведінки

№	Проект-першопрохідець	Тип розвитку	Схожість проекту з конкурентами	Стратегія поведінки на конкурентному ринку
1	Ні	Змішаний тип розвитку: пошук нових споживачів та залучення споживачів конкурентів	Методи конкурентів схожі, тому компанія буде копіювати основні характеристик и конкурентів	Оновлення функціоналу та можливостей згідно побажань розробників та архітекторів

Стратегія для позиціонування продукту

№	Вимоги до продукту від аудиторії	Базова стратегія розвитку	Ключові конкуренто-спроможні позиції стартап-проекту	Асоціації, які формують сприйняття проекту на ринку
1	Впровадження нових функцій, часті оновлення	Збільшити кількість персоналу, досліджувати поведінку користувачів конкурентів, їх побажання	Швидка реакція на недоліки, їх виправлення	Підрахунок кроків, здоровий спосіб життя, моніторинг активності

4.4.Розроблення маркетингової програми стартап-проекту

Таблиця 4.16.

Концепція продукту

№	Потреба	Вигода	Існуюча перевага та дії для створення цієї переваги
1	Підвищення швидкості виконання обчислень у системах на кристалі	Збільшення швидкості виконання обчислень призводить до продуктивності усього підприємства	Можливість швидко вирішувати одну складну задачу, або декілька легких задач одночасно
2	Зменшення витрат енергоресурсів та споживчої потужності	Запобігає нагріванню системи до високих температур	Сильно зменшує вірогідність збоїв у роботі системи причини яких висока температура системи
3	Скорочення виводів ПЛІСа	Підвищується ймовірність реалізації системи на одній мікросхемі.	Дає можливість підвищення надійності систем, зменшення енергоспоживання та габаритів.

Таблиця 4.17.

Межі ціни для стартап-проекту

№ п/п	Рівень цін на методи- замінники	Рівень цін на методи- аналоги	Рівень доходів цільової групи користувачів	Верхня та нижня межі становлення ціни на метод/послугу
1	0\$ - 5\$	0\$ - 10\$	> 5000\$/місяць	0,99\$-4,99\$

Таблиця 4.18.

Система для формування ринку збуту продукту

№ п/ п	Закупівельна поведінка	Функції збуту постачальник а	Глибина каналу збуту	Оптимальн а система збуту
1	Продаж методу на наукових конференціях, які утримують комісію у розмірі 15%	Просування та забезпечення продажу методу, залучення нових користувачів	Однорівневий (продаж через спеціалізовані місця й події)	Вертикальна (власність на метод залишається у розробника)

Таблиця 4.19.

Визначення маркетингових комунікацій для стартап-проекту

№	Специфіка клієнтів	Канали зв'язку клієнтів	Позиціонування	Мета рекламного повідомл.	Концепція рекламного повідомл.
1	Клієнти – розробники обчислювальних систем на кристалі	Конференції, виступи, презентації на спеціалізованих заходах	Надання послуг у комерційних та дослідницьких цілях	Демонстр. можливост. методу	Пояснення цільовій аудиторії переваг методу та його можливост.

ВИСНОВОК ДО РОЗДІЛУ 4

У цьому розділі було проведено розробку стартап-проекту для методу підвищення ефективності паралельних обчислень на розрядному рівні у системах на кристалі, опис якого було представлено у попередніх розділах. У рамках цього розділу було описано ідею для стартап проекту, проведено порівняльний аналіз можливостей конкурентів, їх переваг та недоліків і вирішено наступні завдання:

1. Описано ідею та її мінімальний функціонал для виходу на ринок. Проаналізовано переваги та недоліки конкурентів, визначено переваги розробленого методу та його можливий розвиток;
2. Проведено розгорнутий технічний аудит проекту;
3. Проведено аналіз ринку та можливості для поширення та продажу методу на цьому ринку. Розроблено ринкову стратегію розвитку та план розвитку в умовах наявного ринку. Визначено цільові групи та способи просування проекту та методу;
4. Описано план з маркетингового просування проекту на ринку та сформовано основний канал для продажу продукту, а також визначено цільові групи користувачів.

Отже, у цьому розділі було проведено аналіз можливостей запуску додатку у якості стартап-проекту, оцінено ризики та можливості його запуску.

ВИСНОВКИ

При виконанні даної магістерської дипломної роботи було проаналізовано предметне середовище виконання паралельних обчислень на розрядному рівні у системах на кристалі. Розглянуто види задач для яких ефективно використання паралельних методів обчислення. Детально проаналізовано переваги та недоліки використання паралельних обчислень у системах на кристалі. Надано існуючі методи та способи реалізації систем на кристалі та чисельно описано найважливіші характеристики виконання обчислень задля визначення їх ефективності. Розв'язане важливе науково-прикладне завдання –підвищення ефективності використання паралельних обчислень на розрядному рівні у системах на кристалі.

Розроблений метод виконання паралельних обчислень на розрядному рівні у системах на кристалі, допомагає зменшити споживану потужність усієї системи та суттєво зекономити енергоресурси під час виконання паралельних обчислень, а також запобігти нагріванню компонентів системи до високих температур, що зменшує ризики збою повноцінної роботи через причину вимкнення пристроїв системи через перегрів. Також було розроблено метод порозрядної обробки операндів у режимі реального часу, що призводить до зменшення виводів ПЛІС, пришвидшення передачі інформації у системі між обчислювальними блоками системи, зменшення споживаної потужності системи та її габаритів.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Воеводин В.В. Параллельные вычисления / В.В.Воеводин, Вл.В.Воеводин. – СПб.: БХВ-Петербург, 2002. – 608 с
2. Основні поняття паралельних обчислень [Електронний ресурс] Режим доступу: <https://studfile.net/preview/6060346/>
3. Система на кристалле [Електронний ресурс] Режим доступу: <https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0%D0%BD%D0%B0%D0%BA%D1%80%D0%B8%D1%81%D1%82%D0%B0%D0%BB%D0%BB%D0%B5>
4. Системы на кристалле. Особенности реализации и перспективы применения. [Електронний ресурс] Режим доступу: <https://russianelectronics.ru/sistemy-na-kristalle-osobennosti-realizaczii-i-perspektivy-primeneniya-2/>
5. Жабін В.І. Выполнение последовательностей зависимых операций а режиме совмещения / Жабін В.І., д-р техн. наук К. : НТУУ «КПІ», 9 с.
6. Жабін В.І. Analysis of On-Line Computation Effectiveness in Redundant Number System/ Valerii Zhabin, DcS, Prof.dept. of Computer Engineering, Ivan Dychka, DcS, Prof.dept. of Computer Systems Software, Valentina Zhabina, PhD, Assoc. Prof.dept. of Computer Systems Software, К. : НТУУ «КПІ», 2018р, 7 с.
7. Шпаковский Г.І. Реалізація паралельних обчислень:MPI, OpenMP, кластеры, грид, многоядерные процессоры, графические процессоры, квантовые компьютеры/Г.І.Шпаковский, Минск, 2011р., 176 с.
8. Кулаков К. О. Методы и алгоритмы параллельных вычислений. Проектирование параллельных алгоритмов/ Кулаков К. О., Петрозаводск, 2016р. 33с.
9. Жабін В.І. Скорочення необхідного ресурсу ПЛІС для обчислювальних систем на кристалі./ Валерій Жабін, д.т.н., проф., Валентина Жабіна, к.т.н., доцент, 5 с.

10. Zhabin V., Zhabina V. Methods of On-Line Computation Acceleration in Systems with Direct Connection between Units. 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies DESSERT, Ukraine, Kyiv, May 14-18. 2020. – Kyiv: Proceedings Dessert 2020, P. 356-362. (Scopus)
11. Zhabin, V.I., Korneichuk, V.I., Tarasenko, V.P., “Design of High-Speed Specialized Computers for the Realization of Many-Place Expressions”, Automatic Control and Computer Sciences, 1981, vol. 15, no 6, pp. 15-18.