

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
"КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ"

СЕРГІЄНКО АНАТОЛІЙ МИХАЙЛОВИЧ

УДК 004.383

**МОДЕЛІ, МЕТОДИ ТА ЗАСОБИ СИНТЕЗУ
ОБЧИСЛЮВАЛЬНИХ СИСТЕМ ДЛЯ ОБРОБКИ
ПОТОКІВ ДАНИХ**



Спеціальність 05.13.05 – Комп'ютерні системи та компоненти

АВТОРЕФЕРАТ

дисертації на здобуття наукового ступеня
доктора технічних наук

Київ – 2011

Дисертацією є рукопис.
Робота виконана в Національному технічному університеті України
"Київський політехнічний інститут", Міністерство освіти і науки,
молоді та спорту України, на кафедрі обчислювальної техніки

Науковий консультант:

доктор технічних наук, професор
Сімоненко Валерій Павлович,
Національний технічний університет України "КПІ",
професор кафедри обчислювальної техніки

Офіційні опоненти:

доктор технічних наук, старший науковий співробітник
Гамаюн Володимир Петрович,
Національний авіаційний університет,
завідувач кафедри прикладної інформатики

доктор технічних наук, професор
Теленік Сергій Федорович,
Національний технічний університет України "КПІ",
завідувач кафедри автоматики та управління в технічних системах

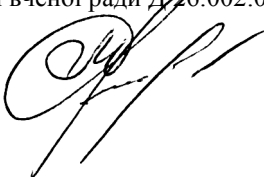
доктор технічних наук, старший науковий співробітник
Чеботарьов Анатолій Миколайович,
Інститут кібернетики ім. В.М.Глушкова НАН України,
в.о. завідувача відділу високопродуктивних комп'ютерних систем

Захист відбудеться " __ " _____ 2011 р. о _____ годині на засіданні
спеціалізованої вченої ради Д 26.002.02 у Національному технічному
університеті України "КПІ" (м. Київ, пр. Перемоги, 37, корп. 18, ауд. 516).

Відгуки на автореферат у двох примірниках, завірені печаткою установи,
просимо надсилати за адресою: 02056, м. Київ, пр. Перемоги, 37, Вченому
секретарю НТУУ "КПІ".

З дисертацією можна ознайомитись у бібліотеці Національного
технічного університету України "КПІ".

Автореферат розісланий " __ " _____ 2011 р.
Вчений секретар спеціалізованої вченої ради Д 26.002.02
канд. техн. наук, доцент



М.М.Орлова

ЗАГАЛЬНА ХАРАКТЕРИСТИКА РОБОТИ

Актуальність теми

В сучасних засобах обчислювальної техніки, таких як персональні комп'ютери, засоби мобільного зв'язку та різноманітні пристрої цифрової обробки сигналів (ЦОС), системах, в яких вирішуються задачі лінійної алгебри (ЛА), реалізовані циклічні алгоритми, що виконують обробку даних, які організовані в потоки. Це алгоритми функціонування модемів, компресії аудіо- та відеоданих, адаптивної фільтрації та розпізнавання образів, моделювання фізичних тіл і трьохвимірних сцен, сортування та пошуку в базах даних тощо. Постійне зростання складності таких алгоритмів і вимоги їхнього виконання в реальному часі призводять до необхідності безперервного зростання продуктивності сучасних обчислювальних систем (ОС). До ОС широкого вжитку, що реалізують ці алгоритми, зараз пред'являються такі протирічні вимоги, як висока швидкодія, суворо обмежене енергоспоживання, низька ціна. Причому постійне зростання складності нових проєктів ОС відбувається за умови необхідності прискорення їхнього проєктування.

Програмовані логічні інтегральні схеми (ПЛІС) використовують переваги як замовлених надвеликих інтегральних схем (НВІС) (швидкодіючі паралельні обчислення з мінімізованим енергоспоживанням), так і універсальних процесорів (можливість перепрограмування). Крім того, розробка нових НВІС супроводжується моделюванням проєктів на ПЛІС і передбачається, що частина майбутніх систем на кристалі буде реалізовуватись як програмоване логічне середовище. Але розробка і програмування проєктів ПЛІС є трудомісткою та відповідальною роботою, продуктивність якої є недостатньою для виконання нових складних проєктів.

Таким чином, задачі, які ставить сучасний етап розвитку обчислювальної техніки, не можуть бути виконані без розробки нових методів і засобів відображення алгоритмів обробки потоків даних у НВІС і ПЛІС, програмування багатопроцесорних ОС. Дослідження та розробка моделей, методів і засобів відображення алгоритмів обробки потоків даних у апаратні засоби, а також їх практична реалізація є актуальними задачами, мають теоретичний і практичний інтерес.

Зв'язок роботи з науковими програмами, планами, темами

У дисертаційну роботу включені основні результати, які одержані автором з 1991 по 2010 рр. на кафедрі обчислювальної техніки Національного технічного університету України "КПІ" при виконанні госпдоговірних робіт (теми "Розробка універсального оперативного течешукача підвищеної точності виміру для визначення місцезнаходження течі в трубопроводах транспорту води, пари, газу, нафтопродуктів", 2001 р., № ДР 0100U000088, "Розробка архітектури та побудова дослідного модуля спеціалізованого обчислювача на базі ПЛІС", 2008 р., № ДР 0107U012770) і держбюджетних робіт (теми "Теоретичне обґрунтування нової технології паралельних обчислень", 1997 р., № ДР 0196U01080, "Розробка основних елементів матрично-графової

теорії паралельних обчислень", 1999 р., № ДР 0198U000650, "Розробка методів і засобів відображення регулярних алгоритмів у програмоване обчислювальне середовище", 2001 р., № ДР 0100U000953, "Методи та засоби відображення періодичних алгоритмів в обчислювальні системи на кристалі", 2003 р., № ДР 0102U000595, "Теоретичне обґрунтування і дослідження методів підвищення ефективності використання апаратних засобів в системах цифрової обробки сигналів", 2005 р., № ДР 0104U000643, "Теоретичне обґрунтування та дослідження методів структурного проектування процесорів для обробки мультимедійних даних", 2008 р., № ДР 0106U002325, "Створення архітектурної концепції та методів структурного проектування мережевих процесорів з інтелектуальною обробкою пакетів даних», 2011р., № ДР 0109U002268). Більшість робіт завершено створенням діючих моделей і зразків обчислювальних модулів і процесорів.

Мета і завдання дослідження

Метою дисертаційної роботи є підвищення ефективності відображення алгоритмів обробки потоків даних у паралельні обчислювальні пристрої, які реалізуються як у ПЛІС, так і в багатопроцесорних ОС.

Для досягнення поставленої мети в дисертації вирішуються наступні задачі:

1. Проаналізувати задачі, алгоритми та структури пристроїв обробки потоків даних і сформулювати вимоги до елементної бази та засобів проектування ОС для обробки таких потоків, визначити критерій ефективності такої ОС.

2. Провести аналіз і класифікацію алгоритмічних моделей обробки потоків даних, методів та засобів їх відображення в паралельні ОС.

3. Теоретично обґрунтувати та розробити модель просторового графа синхронних потоків даних (ГСПД), яка дає змогу виконувати функціонально еквівалентні перетворення алгоритму та направлений пошук оптимального відображення алгоритму в паралельні ОС з суміщенням етапів вибору операційних ресурсів, складання розкладу виконання операцій і призначення операцій та змінних на ресурси.

5. Розробити методи відображення просторового ГСПД у конвеєрні ОС, які реалізуються в ПЛІС з використанням опису алгоритму і результуючої ОС мовою високого рівня, такою як VHDL, яка використовується в галузях ЦОС та для вирішення задач ЛА.

6. Розробити методи відображення алгоритмів обробки потоків даних у багатопроцесорні ОС для вирішення задач ЛА з мінімізацією простоїв їх процесорів.

7. Проаналізувати та удосконалити представлення даних в ОС для обробки потоків даних при ЦОС і для вирішення задач ЛА.

8. Створити засоби автоматизації відображення алгоритмів обробки потоків даних в ОС.

9. Перевірити ефективність розроблених методів при проектуванні ряду спеціалізованих ОС для вирішення задач ЦОС і лінійної алгебри.

Об'єктом дослідження в цій роботі є паралельні ОС для обробки потоків даних.

Предмет дослідження – математичні моделі алгоритмів обробки потоків даних і методи їх відображення в ОС з переважною реалізацією на основі ПЛІС.

Методи дослідження, які використані в роботі, базуються на теорії графів, теорії множин, теорії алгоритмів, теорії моделювання, матричній та лінійній алгебрі, методах комбінаторної оптимізації, а також теорем, тверджень і наслідків, які доведені в дисертації. Основні положення і теоретичні оцінки підтверджені результатами імітаційного моделювання на комп'ютері, а також випробуваннями ряду експериментальних і дослідних зразків спеціалізованих обчислювачів.

Експериментальна перевірка наукових положень, пропозицій і одержаних результатів виконувалась за допомогою проектування обчислювальних засобів розробленими методами з використанням їх опису стандартними мовами VHDL і Verilog з їх подальшим моделюванням в симуляторі, компілюванням у схему на рівні вентилів і конфігуруванням у ПЛІС фірми Xilinx.

Наукова новизна одержаних результатів.

На основі виконаних теоретичних та експериментальних досліджень створено єдиний математичний і методологічний підхід до проектування ОС для обробки потоків даних, суть якого полягає в представленні алгоритму обробки потоків даних на моделі просторового ГСПД і застосування ряду методів цілеспрямованого перетворення цього ГСПД, які призводять до оптимізованих структурних рішень або програм.

Отримані такі нові результати:

1. Критерій ефективності ОС, реалізованої в ПЛІС, який точніше враховує властивості як алгоритмів обробки потоків даних, так і елементної бази.

2. Модель просторового ГСПД, яка дає змогу покращити процес оптимізації ОС за рахунок сумісного виконання вибору ресурсів, призначення операцій алгоритму на них та складання розкладу операцій.

3. Метод синтезу ОС для обробки потоків даних шляхом перетворення моделі просторового ГСПД в структуру ОС та розклад виконання операцій в ній, який забезпечує при заданому періоді обчислень мінімізацію об'єму пам'яті, енергоспоживання, а також апаратних витрат, в тому числі кількості процесорних елементів, їхніх регістрів, мультиплексорів, з'єднань.

4. Метод синтезу конвеєрних обчислювальних пристроїв для ПЛІС, а також способи проектування таких пристроїв з використанням відображення періодичних алгоритмів з операторами керування, синтезу буферних схем, мови VHDL, які за рахунок мінімізації простоїв елементів ПЛІС та глибокої

конвейеризації обчислень дають змогу проектувати пристрої з екстремально малим періодом синхросерії та максимальним відношенням продуктивність – апаратні витрати.

5. Способи побудови функціональних схем фільтрів зі скінченною імпульсною характеристикою, рекурсивних фільтрів з кратними затримками, реалізованих на ПЛІС, що мають продуктивність та відношення продуктивність–апаратні витрати, які у 1,2 – 2 рази більше, ніж у фільтрів, розроблених традиційними способами.

6. Структурно-функціональні схеми конвеєрних обчислювачів для ЦОС, таких як процесори швидкого перетворення Фур'є (ШПФ), дискретного косинусного перетворення, цифрові фільтри, які за своїми характеристиками перевищують світові аналоги.

7. Спосіб представлення даних раціональними дробами, який при такій самій точності обчислень, що дає представлення з плаваючою комою, забезпечує обчислення ЦОС і вирішення задач ЛА зі зменшеними апаратними витратами на 25% та більш ніж у 1,2 рази збільшеною швидкодією.

8. Метод проектування паралельних ОС для вирішення задач ЛА, який використовує відображення просторового ГСПД і решітчастого графа алгоритму, що забезпечує більше завантаження процесорних елементів ОС, ніж існуючі методи.

9. Способи програмної конвеєризації багатопроцесорної ОС і програмування SIMD-процесора на основі відображення просторового ГСПД, які забезпечують збільшення продуктивності ОС за рахунок мінімізації простой процесорних елементів ОС.

Практичне значення одержаних результатів полягає в тому, що застосування запропонованих методів проектування паралельних ОС забезпечує наступні можливості: 1) зниження трудомісткості та скорочення строків одержання множини оптимізованих альтернативних структурних рішень при проектуванні конвеєрних ОС, що реалізуються як у ПЛІС, НВІС, так і в одно- та багатопроцесорних ОС на базі мікропроцесорів загального призначення; 2) підвищення завантаженості обчислювальних ресурсів і пам'яті ОС; 3) розробку високоефективних ОС обробки сигналів з параметрами, які є близькими до наперед заданих.

Методи проектування ОС для обробки потоків даних, по-перше, дозволяють описувати за допомогою ГСПД задачу в цілому, а також її складові підзадачі, по-друге, дають розробнику можливість не тільки синтезу множини ОС з заданими характеристиками, але й конструювання алгоритмів з необхідними структурними властивостями. Тому вони можуть бути розвинуті в методи ітеративного проектування алгоритмів і паралельних ОС та бути основою відповідних САПР. Запропоновані методи спрощують розробку паралельних програм для багатопроцесорних ОС, завдяки зниженню трудомісткості етапів складання розкладу виконання операцій та

організації керування взаємодією процесорів та можуть бути використані як основа компілятора для паралельних ОС.

Ряд проектів ОС для вирішення задач ЦОС та лінійної алгебри, що одержані в процесі підготовки дисертації та перевірки нових методів, є обчислювальними модулями, які описані на мовах VHDL і Verilog. Вони мають високі технічні характеристики та можуть бути впроваджені у різних нових розробках з мінімальними додатковими часовими та фінансовими витратами.

Запропонований ряд методів проектування ОС приймає до уваги більшість прогресивних тенденцій і підходів до проектування ОС, особливості сучасної елементної бази, дає змогу класифікувати моделі паралельних алгоритмів, підійти з єдиної позиції до процесу проектування спеціалізованих ОС. Тому він може слугувати основою курсів навчання проектуванню таких ОС для студентів, аспірантів та спеціалістів.

Методологічні й наукові аспекти дисертаційних досліджень знайшли свою реалізацію в навчально-методичному забезпеченні лекційних курсів "Комп'ютерна схемотехніка", "Архітектура комп'ютерів", "Апаратно-орієнтоване програмування", які читаються автором студентам факультетів інформатики і обчислювальної техніки, інституту спеціального зв'язку та захисту інформації Національного технічного університету України "КПІ".

Особистий внесок здобувача

Основний зміст дисертаційної роботи та її результати повністю відображені в опублікованих наукових роботах автора. Основний зміст дисертації представлено 62 наукових роботах, з них в 13 одноосібних роботах [1, 4, 8, 9, 12-15, 18, 21-23, 58] і 49 роботах, що опубліковані у співавторстві. В роботах [3, 7, 29, 31, 56], які надруковані у співавторстві, автору належать алгоритми, які реалізовано в описаних системах, в роботах [2, 6, 10, 13, 27, 28, 30, 32-37, 47, 49] автором запропоновано методи і методики відображення ГСПД в структуру ОС, в [5, 41, 42, 51, 52, 55] автору належить метод відображення алгоритму в конвеєрну структуру і одержані структури ОС. В роботах [11, 25, 40, 43-46, 54, 57, 60, 61] автором запропоновані спосіб представлення даних та синтезовані структури пристроїв, в роботі [50] запропонована класифікація моделей алгоритмів обробки потоків даних, а в роботах [16, 17, 24, 26, 39, 45, 46, 53] автором узагальнені й уточнені одержані ним методи проектування конвеєрних ОС і досліджені результати їхнього практичного застосування. Прикладні аспекти застосування результатів розглянуті в роботах [20, 38, 48, 59, 62].

Сформульовані в дисертаційній роботі наукові результати, висновки і рекомендації належать особисто автору та є його науковим доробком. Дисертація є одноосібно виконаною науковою працею.

Апробація результатів дисертації

Основні результати дисертаційної роботи були представлені, обговорювались і були схвалені на 33 міжнародних і регіональних наукових

конференціях і семінарах:

I – III та V – VIII міжнародних конференціях "Parallel Processing and Applied Mathematics" (PPAM'94, Czestochova, 1994, PPAM'97, Zakopane, 1997, PPAM'99, Kazimierz Dolny, 1999, PPAM'2003, Czestochova, 2003, PPAM'05, Poznan, 2005, PPAM'07, Gdansk, 2007, PPAM'09, Wroclaw, 2009, Poland), міжнародному семінарі "Parallel Numerics'94" (Smolenice, Slovakia, 1994), міжнародній конференції, присвяченій пам'яті акад. С.А. Чумихіна (Київ, 1995), I та VII міжнародних конференціях "Автоматизація проектування дискретних систем" (CADDD'95, 1995 і CADDD'07, 2007, Мінськ, Республіка Беларусь), міжнародній конференції "International Conference on Signal Processing, Application and Technology" (ICSPAT'95, Boston, USA), 7-й міжнародній конференції "Parallel Architectures and Cellular Automata" (Parcella'96, Berlin, Germany, 1996), I та III міжнародних конференціях "Parallel Computing in Electrical Engineering" (PARELEC'98, Bialystok, 1998, PARELEC'2002, Warsaw, 2002, Poland), міжнародному симпозиумі "Комп'ютери у Європі. Минуле, сучасне і майбутнє" (Київ, 1998), 24-й міжнародній конференції "Euromicro Conference on Parallel and Distributed Processing" (Euromicro'98, Vasteras, Sweden, 1998), II – IV і XI регіональних конференціях "Reprogramowalne układy cyfrowe" (RUC'99, 1999, RUC'2000, 2000, RUC'2001, 2001, RUC'2008, 2008, Szczecin, Poland), міжнародній конференції "International Active-VHDL Conference" (Харків, 2001), міжнародній конференції "Масштабируемые системы и компьютерные сети" (SCALNET'2004, Кременчуг, 2004), міжнародній конференції "Информационные технологии в управлении энергетическими системами" (Київ, 2005), I, II та III міжнародних конференціях "Моделювання" (Київ, 2006, 2008, 2010), 9-й та 10-й міжнародних конференціях "Досвід розробки і застосування САПР в мікроелектроніці" (CADSM'2007, CADSM'2009, Львів, Поляна, 2007, 2009), 14-й міжнародній конференції "Mixed Design of Integrated Circuits and Systems" (MIXDES'2007, Ciechocinek, Poland, 2007), міжнародній конференції (CODATA-21, Київ, 2008), 2-й міжнародній конференції "East-West Design & Test" (EWDTS'08, Львів, 2008), ювілейній міжнародній науково-практичній конференції 50-річчя створення кафедри ОТ (Київ, 2010).

Публікації. Основні положення дисертації викладені у 62 роботах. Серед них 1 монографія, 25 статей у провідних фахових часописах і збірниках (10 без співавторів), 25 статей у матеріалах міжнародних конференцій, 3 тези та 8 статей в інших фахових часописах і збірниках.

Структура та обсяг дисертації

Дисертаційна робота складається зі вступу, шести розділів, висновків, списку використаних джерел з 310 найменувань, містить 27 таблиць та 110 ілюстрацій. Робота містить 286 сторінок основного тексту, 59 сторінок з таблицями та ілюстраціями і 20 сторінок використаних джерел. Повний обсяг роботи 373 сторінки.

Основний зміст роботи

У вступі обґрунтовується актуальність теми дисертаційної роботи, формулюється мета і задачі дослідження, основні положення, що виносяться на захист.

У першому розділі розглянуті основні задачі, алгоритми та пристрої обробки потоків даних, вимоги до елементної бази та засобів проектування відповідних ОС.

Для задання алгоритмів обробки потоків даних традиційно використовують різні графові моделі потоків даних. В такій моделі вершини – актори – представляють собою оператори, а дуги – потоки даних. Під потоком мається на увазі засіб передачі даних між елементарними обчислювачами, якими є актори. Два потоки вважаються синхронними, якщо встановлена стала відповідність між індексами даних (токенів) в них. Граф потоків даних вважається синхронним, якщо всі потоки є синхронними. Актор миттєво виконує відповідну функцію над вхідними даними і видає результат на свій вихід, якщо стверджується задана умова його запуску.

В результаті аналізу існуючих алгоритмічних моделей обробки потоків даних була складена їх класифікація. Основна класифікаційна ознака – статичність розкладу виконання акторів, яка дозволяє подальшу апаратну реалізацію алгоритму. За цією класифікацією вибрано граф синхронних потоків даних (ГСПД, synchronous dataflow graph, SDF) як основну модель для задання алгоритмів. Направлена дуга ГСПД передає дане миттєво або з затримкою у своєму буфері типу FIFO. В ГСПД кожен актор запускається при наявності даних на його входах, протягом циклу моделювання генерує в інцидентні дуги і споживає з них дані, число яких є сталим в кожному циклі. Причому конфігурація станів потоків ГСПД (множини даних в дугах з відповідними індексами) на початку кожного циклу є сталою. Однорідний ГСПД відрізняється тим, що усі актори протягом одного циклу спрацьовують по одному разу. Таким чином, ГСПД представляє собою граф однієї ітерації алгоритму з доданою інформацією про міжітераційні зв'язки. При цьому дуги, які замикають ациклічний ГСПД в циклічний граф, мають буфери FIFO та відповідають міжітераційним залежностям. Сигнальний граф, який застосовується в проектуванні ОС для ЦОС, є аналогом однорідного ГСПД.

Були проаналізовані методи та засоби високорівневого синтезу ОС. Згідно із загальноприйнятою методологією, синтез ОС виконується за три етапи: етап формування множини апаратних ресурсів, етап складання розкладу виконання операцій та етап призначення операцій ресурсам. Методи синтезу ОС розрізняються між собою порядком застосування підетапів: призначення змінних регістрам або шинам, призначення операцій процесорним елементам (ПЕ), складання розкладу виконання операцій, пересилок змінних тощо, а також алгоритмами їх виконання. Найбільш відповідальний і складний етап –

це складання розкладу, тому що від нього залежить продуктивність ОС та завантаженість її ресурсів, і він представляє собою вирішення складної, в загальному випадку, *np*-повної задачі комбінаторної оптимізації.

Згідно з поширеним підходом, розклад спрацьовування операцій шукають для одного циклу виконання ГСПД з плануванням обчислень для його ациклічної частини. Тут використовують алгоритми спискового планування, силового планування, розфарбування графа, метод з вирівнюванням положення операторів за лівою границею простору ресурси – час (метод лівої границі – *left edge scheduling*). Але їх безпосереднє застосування дає низьке завантаження ресурсів на початку та у кінці циклу. Для урахування циклічності алгоритму аналізують граф конфліктів ресурсів і операцій, а також граф інтервалів існування змінних.

Оптимальність синтезованої ОС залежить від усіх етапів синтезу, які, як правило, виконуються окремо один від одного. Кожен з них має локальну мету, що протирічить меті інших етапів. Так, мінімізація ресурсів на першому етапі призводить до збільшення тривалості розкладу. Тому існуючі методи системного синтезу часто не забезпечують високу якість ОС.

Ідея методів, що пропонуються в дисертації, полягає в тому, щоб виконувати три етапи синтезу ОС за один або два кроки. Завдяки цьому досягається більш глибокий рівень оптимізації структурних рішень. Крім того, складання розкладу виконання операцій, яке виконується з побудовою структурного рішення, представляється як чітка задача комбінаторної оптимізації, що може бути зведена до задачі, яка вирішується стандартними методами, такими як метод гілок і границь, метод лівої границі, генетичний метод, метод модельованого відпалювання тощо. Детальне врахування властивостей неоднорідного ГСПД дає змогу виконувати направлений пошук оптимального циклічного розкладу виконання операцій.

На основі аналізу методів та засобів високорівневого синтезу сформульовані вимоги до елементної бази, методів і засобів проектування ОС для реалізації алгоритмів обробки потоків даних.

Вперше була запропонована методика моделювання ГСПД з використанням мови VHDL. При цьому було встановлено, що за допомогою VHDL можна як моделювати алгоритм обробки синхронних потоків даних на існуючих стимуляторах, так і реалізувати його апаратно з періодом виконання $L = 1$ такт, наприклад, в ПЛІС, завдяки чому прискорюється проектування ОС. Також була показано, що при орієнтації на апаратну реалізацію алгоритму, слід використовувати однорідний ГСПД, так як відображення неоднорідного (*multirate*) ГСПД є значно складнішим і потребує перевірки відсутності блокувань.

У другому розділі сформульовані вимоги до методів та засобів відображення алгоритмів, критерії оптимізації ОС, побудована модель просторового ГСПД у вигляді конфігурації алгоритму (КА). Досліджені властивості КА,

включаючи відображення в конфігурації структури і передування, на основі чого розроблено метод синтезу конвеєрних ОС.

Початкові дані для проектування ОС – це алгоритм, представлений ГСПД і критерії оптимізації. Розглядаються ГСПД, в яких число груп змінних, які споживаються актором, дорівнює числу результатів, що видаються цим актором протягом одного циклу, тобто однорідні ГСПД.

Передбачається, що структура ОС складається з простих ПЕ, які зв'язані між собою згідно з графом структури ОС. Такий ПЕ включає в себе арифметико-логічний пристрій певного типу (блок множення, суматор) з регістром результату на його виході (або з буфером FIFO) і мультиплексо-рами вхідних даних на його входах. ПЕ виконує операцію над вхідними даними з записом результату в регістр не більше, ніж за один такт. Якщо ПЕ не має регістра, то вважається, що операція виконується з дельта-затримкою. Така модель ПЕ з певними обмеженнями також може бути поширена на ПЕ, які реалізовані на основі мікропроцесора.

За критерії оптимізації приймаються період виконання алгоритму $Q_T = L t_C$ і сума зважених кількостей ПЕ різного типу: $Q_S = \sum C_p q_p$, де L – число тактів періоду, t_C – тривалість такту, яка дорівнює довжині критичного шляху проходження сигналу від одного регістра до іншого, C_p – апаратна складність ПЕ p -го типу, q_p – кількість ПЕ p -го типу. Інтегральним критерієм є відношення продуктивність–апаратні витрати $Q = 1/(Q_T Q_S)$.

Через великі затримки у лініях зв'язку, в ПЛІС важливо досягати конвеєрної обробки даних з мінімальною довжиною критичного шляху. При відображенні ГСПД у ПЛІС з $L > 1$ в затримку критичного шляху додаються затримки мультиплексорів. У ПЛІС апаратні витрати мультиплексорів можуть суттєво перевищувати витрати регістрів і суматорів. Тому доцільно мінімізувати як число ПЕ, так і кількість входів їх мультиплексорів, яка крім того, пропорційна складності ліній зв'язку. Результуючі критерії оптимізації для більшості проектів ЦОС на ПЛІС фірми Xilinx визначені як наступні:

$$\begin{aligned} Q_S &= n_R + n_A + 20n_M + (0,28 \dots 0,57)n_X; \\ Q_T &= L(n'_A + (1,9 \dots 2,3)n'_M + (0,5 \dots 0,6)n'_X), \end{aligned}$$

де n_R – число регістрів в ОС, n_A – число суматорів; n_M – число блоків множення; n_X – число входів мультиплексорів; n'_A , n'_M , n'_X – числа суматорів, блоків множення, мультиплексорів, відповідно, які стоять у критичному шляху; діапазон коефіцієнтів залежить від серії ПЛІС.

Для того, щоб мати можливість цілеспрямовано виконувати одночасний пошук як структури ОС, так і розкладу виконання в ній операторів, була запропонована модель алгоритму у вигляді просторового ГСПД.

Будь-який граф алгоритму G_A , в тому числі ГСПД, граф залежності даних, який йому відповідає, задається матрицею інцидентності $A = ||a_{ij}||_{N \times M}$, де $a_{ij} = 1$, якщо дуга j графу G_A кінцем інцидентна i -й вершині або $a_{ij} = -1$,

якщо вона інцидентна i -й вершині оператора і $a_{i,j} = 0$ – при відсутності інцидентності, N і M – число вершин і дуг графу G_A .

У загальному випадку, просторовий ГСПД представляється в n -вимірному цілочисельному просторі $\mathbf{Z}^n$. Вектори $\mathbf{K}_i \in \mathbf{Z}^n$ ототожнюються з вершинами графу G_A , тобто вони є тегами цих вершин. Координати векторів $\mathbf{K}_i$ можуть мати різну інтерпретацію. Частина координат векторів $\mathbf{K}_i$ представляють координати $\mathbf{K}_{S_i}$ ПЕ, в якому виконується i -й оператор, а інша частина $\mathbf{K}_{T_i}$ – кодує момент виконання цього оператора $\mathbf{K}_i = \begin{pmatrix} \mathbf{K}_{S_i} \\ \mathbf{K}_{T_i} \end{pmatrix}$. В іншій інтерпретації, наприклад, при відображенні гнізда циклів алгоритму, у ядрі якого виконується цей оператор, вектор $\mathbf{K}_i$ дорівнює набору індексів змінних.

За іншими координатами векторів $\mathbf{K}_i$ може вестись відлік, наприклад, рівнів ієрархії алгоритму і структури, типу оператора та ПЕ, затримки виконання оператора, кодуватись інформація про особливості операторів і ПЕ, таких, як необхідність конвеєрної реалізації, вказівка напрямку оптимізації і т.і. Далі розглядатиметься розмірність простору $n = 3$. Тоді i -й оператор просторового ГСПД ототожнюється з вектором $\mathbf{K}_i = (s, p, t)^T$, ($i = 1 \dots N$), де p – тип операції, s – просторова координата, t – часова координата.

Вершини ГСПД, як правило, генерують мітки, що відповідають окремим змінним x_j , причому на всі вихідні дуги однієї вершини передається одна й та сама змінна x_j . Якщо вершина видає змінні x_i, x_j у вихідні дуги, то необхідно розщепити її на дві вершини, одна з яких генерує x_i , а інша – x_j .

Аналогічно відповідності вершин операторів, кожна змінна x_j алгоритму ототожнюється з n -вимірним вектором $\mathbf{D}_j \in \mathbf{Z}^3$, ($j = 1 \dots M$). Згідно з визначенням матриці A , яка має елементи $a_{i,j}, a_{i,j}$, виконується рівність

$$\mathbf{D}_j = \mathbf{K}_l - \mathbf{K}_i = a_{l,j} \mathbf{K}_l + a_{i,j} \mathbf{K}_i, \quad (1)$$

де $\mathbf{K}_i$ відповідає оператору з результатом x_j , а $\mathbf{K}_l$ – оператору, для якого x_j – вхідний операнд, тобто $\mathbf{K}_i$ безпосередньо передує $\mathbf{K}_l$.

Множини векторів $K = \{\mathbf{K}_i\}$ і $D = \{\mathbf{D}_j\}$ взаємозв'язані так само, як множини вершин і дуг графа алгоритму. Таким чином, представлення ГСПД в багатовимірному просторі означає його кодування матрицями векторів $\mathbf{K}_i, \mathbf{D}_j$, а також матрицею A на основі (1). Щоб точно задати багатовимірний ГСПД, введено наступні визначення:

Конфігурацією алгоритму (КА) є трійка $C_A = (K, D, A)$, де $K = (\mathbf{K}_1, \dots, \mathbf{K}_N)$ – матриця векторів-вершин $\mathbf{K}_i = (s_i, p_i, t_i)^T \in \mathbf{Z}^3$; $D = (\mathbf{D}_1, \dots, \mathbf{D}_{M+1})$ – матриця векторів-дуг $\mathbf{D}_j = (s_j, p_j, t_j)^T \in \mathbf{Z}^3$; $A = ||a_{i,j}||_{N \times (M+1)}$ – матриця інцидентності графа G_A . Ця матриця доповнена $M+1$ -м стовпцем з елементом $a_{p, M+1} = 1$, що відповідає базисному вектору $\mathbf{D}_{M+1} = \mathbf{D}_B$, що з'єднує початок системи координат простору $\mathbf{Z}^3$ з довільним вектором-вершиною $\mathbf{D}_B = \mathbf{K}_p \in K$. У загальному випадку, кількість таких стовпців дорівнює числу компонент зв'язності G_A .

Якщо G_A – ГСПД, то КА представляє собою *просторовий ГСПД*. На відміну від графа залежностей даних, у такого графа є вектори-дуги D_{Dj} , які відповідають залежностям між i -ю та $i - n$ -ю ітераціями алгоритму. Вони навантажені затримками FIFO глибиною n і належать до замкнених циклів графа. Тобто D_{Dj} відповідає змінним $x_{i-1}, \dots, x_{i-n}$, які зберігаються в буфері FIFO дуги.

Розглянемо алгоритм рекурсивного фільтра другого порядку з формулою: $y[i] = x[i] + ay[i-2] + by[i-1]$. Ця формула обчислюється програмним циклом:

```
for i = 1, N do
    St1: y1[i] = a*y[i-2];      St2: y2[i] = b*y[i-1];
    St3: y3[i] = x[i] + y1[i];  St4: y[i] = y2[i] + y3[i];
end.
```

В ньому операторам St1,...,St4 відповідають вектори-вершини $K_i = (s, p, q, t)^T$, ($i=1, \dots, 4$) просторового ГСПД, де s – просторова координата, p – тип операції, q – часова координата відліку ітерацій алгоритму, t – часова координата відліку тактів в ітерації. Змінним $y[i], \dots, y3[i]$ відповідають вектори-дуги D_j , причому змінні $y[i-1]$, $y[i-2]$ ототожнюються з векторами-дугами D_{D1} , D_{D2} , які мають затримки на 1 та 2 ітерації.

Конфігурацією структури (КС) є трійка $C_S = (K_S, D_S, A)$, де $K_S = (K_{S1}, \dots, K_{SN})$ і $D_S = (D_{S1}, \dots, D_{S(M+1)})$ – матриці векторів-вершин $K_{Si} = (k_{Si,1}, \dots, k_{Si,m})^T \in \mathbb{Z}^m$ і векторів-дуг $D_{Sj} = (d_{Sj,1}, \dots, d_{Sj,m}) \in \mathbb{Z}^m$, що дорівнюють координатам ПЕ, в яких виконується i -й оператор та відносним координатам лінії зв'язку, по якій передається j -а змінна, відповідно; A – матриця інцидентності ГСПД. У простому випадку, $K_{Si} = (s_i, p_i)^T \in \mathbb{Z}^2$.

Конфігурацією передування (КП) є трійка $C_T = (K_T, D_T, A)$, де $K_T = (K_{T1}, \dots, K_{TN})$ – матриця векторів-вершин $K_{Ti} = (k_{Ti,1}, \dots, k_{Ti,(n-m)})^T \in \mathbb{Z}^{n-m}$, які означають моменти виконання i -х операторів і $D_T = (D_{T1}, \dots, D_{T(M+1)})$ – матриця векторів-дуг $D_{Tj} = (d_{Tj,1}, \dots, d_{Tj,(n-m)})^T \in \mathbb{Z}^{n-m}$, які відповідають затримкам передачі j -х змінних. У найпростішому випадку $K_{Ti} = (t_i) \in \mathbb{Z}$. Тоді матриці K_T відповідає *вектор часової розгортки* $T = (t_1, \dots, t_N)^T$, який вказує такти виконання операторних вершин.

Для просторового ГСПД і відповідних йому конфігурацій доведено наступні твердження.

Матриці КА зв'язані між собою лінійними залежностями, які впливають з (1):

$$D = KA; \quad K = D_O A^{-1}_O, \quad (2)$$

де A^{-1}_O – інверсна матриця остова ГСПД. Таким чином, маючи наперед задані координати векторів-дуг $D_{Oj} \in D_O$ остова графу, які характеризують

затримки виконання операторів, за формулами (2) знаходяться шукані координати ПЕ, задані матрицею K .

КА є *коректною*, якщо в матриці K відсутні однакові вектори, тобто

$$\forall K_i, K_j (K_i \neq K_j, i \neq j). \quad (3)$$

Якщо в ГСПД є цикли, то повинна бути нульовою сума векторів-дуг D_j , які входять у будь-який цикл графа, тобто для i -го циклу

$$\sum b_{i,j} D_j = 0, \quad (4)$$

де $b_{i,j}$ – елемент i -ї строки цикломатичної матриці ГСПД.

КП S_T для ГСПД є *коректною* відносно часової функції R , якщо

$$K_T - K_{Ti} = D_{Tj} \in D_T \Rightarrow \begin{cases} R(K_{Tr}) \geq R(K_{Ti}) & \text{– для ненавантажених дуг;} \\ R(K_{Tr}) < R(K_{Ti}) & \text{– для дуг із затримками } D_{Dj}; \\ i, r = 1, \dots, N; \quad j = 1, \dots, M+1; \end{cases} \quad (5)$$

тобто, коли дві вершини зв'язані дугою, то момент виконання оператора вершини, що передує, завжди не більше моменту виконання оператора наступної вершини.

Розклад виконання операторів для просторового ГСПД, що виконується з періодом L тактів, є коректним, якщо оператори, які відображаються в один ПЕ, виконуються в різних тактах, тобто

$$\forall K_p, K_j \in K(k_{i,r} = k_{j,r}; r = 1, \dots, n-1) \Rightarrow \begin{cases} k_{i,n} \equiv k_{j,n} \pmod{L}, \text{ де } D_{Dj} = K_i - K_j \text{ – дуга з затримкою;} \\ k_{i,n} \not\equiv k_{j,n} \pmod{L} \text{ – у решті випадків.} \end{cases} \quad (6)$$

КА лінійним відображенням або розщепленням своїх компонентів перетворюється в КС і КП так, що:

$$K = \begin{pmatrix} K_s \\ K_t \end{pmatrix}; \quad D = \begin{pmatrix} D_s \\ D_t \end{pmatrix}. \quad (7)$$

За умови, що КА, КП та розклад виконання операторів є коректними, тобто задовольняють умовам (3) – (6), одержана за умовою (7) модель ОС буде виконувати заданий алгоритм у конвексному режимі з періодом L тактів.

Однотипні оператори слід відображати в ПЕ такого самого типу, тобто

$$K_p, K_j \in K_{p,q} (k_i = k_j = p, s_i = s_j = q), |K_{p,q}| \leq L, \quad (8)$$

де $K_{p,q}$ – множина векторів-вершин операторів p -го типу, які відображаються в q -й ПЕ p -го типу ($q=1, 2, \dots, q_{p\max}$).

На рис. 1,а показаний приклад ГСПД у двовимірному просторі, а на рис.1,б – відповідний йому граф залежностей даних (ГЗД).

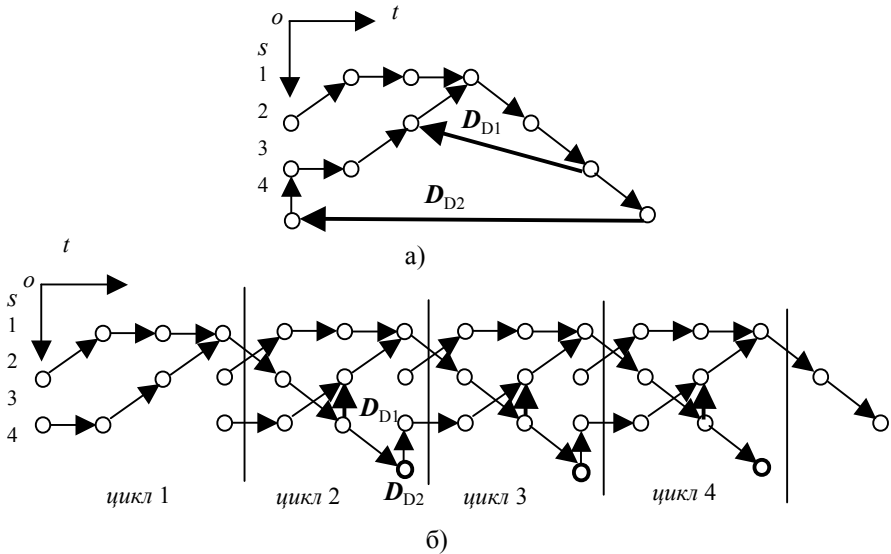


Рис. 1. Приклад ГСПД а) і відповідного йому ГЗД б).

При цьому, згідно з умовами (3) – (6), КА і КП цього ГСПД, а також розклад, представлений ГЗД, є коректними при $L = 3$. Цей приклад показує наступне:

- є можливість скласти розклад виконання операторів періодичного алгоритму (ГЗД на рис.1 є періодичним), коли завантаженість ПЕ наближається до 100%;

- протягом одного періоду виконання алгоритму ПЕ можуть обчислювати до трьох сусідніх ітерацій, причому в довільному порядку (другий ПЕ – в порядку 1-2-3, третій ПЕ – в порядку 2-1-3);

- вершини, які з'єднані векторами ітераційної залежності (D_{D1} , D_{D2}), виконуються в одному такті, тобто, вони коректно забезпечують передачу даних між ітераціями.

Метод синтезу ОС для обробки потоків даних шляхом трансформації моделі просторового ГСПД полягає в пошуці ефективної КА. Причому різні КА, що відрізняються своїми матрицями K , представляють різні структурні рішення, тобто матриці K відповідають точкам у просторі можливих рішень. Ефективна КА, яка має мінімізовані простоти своїх ПЕ, шукається у два етапи.

На першому етапі виконання методу вершини і дуги ГСПД розміщуються в тривимірному просторі $\mathbb{Z}^3$ як множини векторів K_i і D_j з урахуванням умов (2)–(6), (8), тобто формується початкова КА. Мінімізується число ПЕ в структурі шляхом виконання вимоги: $|K_{p,q}| \rightarrow L$, коли число вершин, які

відображаються в один ПЕ, прямує до L . Завдяки цьому, мінімізується число ПЕ в структурі ОС, яке прямує до мінімально можливого значення для заданого алгоритму і параметру L .

Слід відзначити, що перестановки вершин відносно осі часу ot , які відбуваються на цьому етапі, еквівалентні оптимізації алгоритму відомим способом ресинхронізації (retiming). Таким чином, вирішення *np*-складної задачі ресинхронізації значно спрощується завдяки відкиданню заздалегідь не вигідних рішень.

На другому етапі просторовий ГСПД урівноважується. Для цього розглядається ациклічний підграф ГСПД без дуг D_{Dj} із затримками. Під час врівноваження у всі дуги підграфу додаються проміжні вершини операторів затримки (регістрів). У результатуючій врівноваженій КА всі вектори-дуги, крім дуг D_{Dj} , дорівнюють $D_j = (a_j, b_j, 1)^T$ або $D_j = (a_j, b_j, 0)^T$. В такому ГСПД вершини-оператори формують яруси, відстань між якими за координатою часу ot дорівнює 1 такт. Врівноважена КА оптимізується через взаємні перестановки векторів-вершин з одного ярусу. При цьому досягається як мінімізація числа регістрів, так і числа входів мультиплексорів. Застосовуються такі стратегії, як алгоритм лівої границі, стратегія прямування завантаженості ПЕ до повної, тобто $|K_{p,q}| \rightarrow L$. Результуючі структура ОС і розклад виконання операторів алгоритму одержуються згідно з (7).

Запропонований метод дає змогу цілеспрямовано виконувати пошук як структури ОС, так і розкладу виконання операторів алгоритму в ній. Для цього використовуються критерії складності Q_S і Q_T , які обчислюються з матриць K і D . Так, число ПЕ в ОС дорівнює кількості множин однакових стовпців у матриці K_S , сумарна кількість ліній зв'язку в ОС дорівнює

$$M_S = M - M_0 - \sum_v \sum_i (M_{vi} - 1),$$

де M – загальна кількість векторів D_j в ГСПД, M_0 – число векторів $D_{ij} = (0, 0, t_{ij})^T$, $M_{v,i}$ – число векторів D_{ij} , які відображаються в одну лінію зв'язку.

Для ефективного представлення інформації про ГСПД і прискорення оптимізації на першому етапі синтезу ОС, запропоновано використовувати досконалий остов алгоритму. *Досконалий остов* – це остов ГСПД з матрицею A_O , в якому: відсутні стягуючі дуги і дуги D_{Dj} із затримками, тобто дуги між ітераційних залежностей; дуги належать до найбільш довгих маршрутів; відмічені вершини, в які заходять критичні поперечні дуги. Наявність таких вершин може призвести до некоректного розкладу виконання операторів. Запропоновано алгоритм побудови такого остову.

Пошук часової складової відображення ГСПД полягає в підстановці мінімальних допустимих значень часових координат у матрицю D_{TO} досконалиго остову, в обчисленні шуканих координат векторів-вершин K_{Ti} з матриці K_T за правим рівнянням (2) і знаходженні координат решти векторів D_{Ti} матриці D_T за лівим рівнянням (2) з перевіркою умов (4) і (6). Доведено, що розклад, одержаний таким чином, відповідає розкладу, складеному за прин-

ципом найшвидшого призначення (ASAP). Також доведено, що мінімізація об'єму пам'яті ПЕ досягається мінімізацією координат векторів $\mathbf{D}_{Tj}$.

Таким чином, пошук ефективного структурного рішення ОС виконується без побудови власне структури ОС при одночасному слідкуванні за одержанням розкладу і призначенням операторів та змінних ресурсам. Це дає змогу виконувати більш глибоку оптимізацію, застосовувати відомі методи оптимізації, такі, як методи гілок і меж, покрокового конструювання, моделюваного відпалювання або генетичний метод. В останньому випадку матриця K представляє собою хромосому, яка кодує варіант структурного рішення.

Також розроблені способи мінімізації числа ліній зв'язку, мінімізації об'єму пам'яті, відображення вершин затримки в запам'ятовуючий пристрій з довільною вибіркою або в буфер FIFO, мінімізації мультиплексорів, метод синтезу ОС, що виконує умовні оператори.

У третьому розділі представлені метод і способи проектування конвеєрних обчислювачів, орієнтовані на реалізацію в ПЛІС. Запропоновано високорівневий опис періодичного алгоритму мовою VHDL. Розроблено метод відображення просторового ГСПД в структуру конвеєрного обчислювального пристрою (ОП), який забезпечує максимальне врахування властивостей елементної бази ПЛІС та проектування ОП на рівні регістрових передач.

На першому етапі синтезу формується просторовий ГСПД згідно з умовами (2)–(6), (8) так само, як це виконується за методом синтезу ОС для обробки потоків даних шляхом трансформації моделі просторового ГСПД, описаному в другому розділі. При цьому мінімізується кількість таких ресурсів, як суматори і особливо – блоки множення.

На другому етапі виконується урівноважування КА з мінімізацією числа регістрів та входів мультиплексорів. В результаті формуються L множин операторів, які відповідають умові (8), причому m -а множина K_m вміщує оператори, які відповідають векторам-вершинам, для яких

$$\forall \mathbf{K}_i = (s_p, p_p, t_i)^T, m \equiv t_i \bmod L, m = 0, 1, \dots, L-1.$$

Для цього в результатуючій КА виділяються яруси, які відстають один від одного на L тактів вздовж осі ot ; вершини цих ярусів формують множини K_m . Також формуються множини $K_{p,q}$ векторів-вершин операторів p -го типу, які відображаються в q -й ПЕ. Кожна з цих множин позначається унікальним іменем, наприклад, urq . Таким чином, кожному з операторів, які позначені векторами $\mathbf{K}_i$, ставиться у відповідність єдина пара: $\langle urq, t \rangle$, де urq означає ім'я змінної (сигналу), яка зберігається у відповідному регістрі ПЕ, а t – номер такту, в якому виконується присвоювання цій змінній, який обчислюється за модулем L .

При виконанні першого та другого етапів ефективно використовувати розроблений програмний засіб Paredit, який дає змогу автоматично

генерувати урівноважену КА з урахуванням періодичності обчислень та з застосуванням оптимізації методом лівого краю і потім вручну покращувати рішення, використовуючи еволюційний підхід.

На третьому етапі одержується структурно рішення ОП як опис оптимізованого просторового ГСПД мовою VHDL або іншою високорівневою мовою поведінкового опису апаратури. Це рішення представляє собою опис архітектури VHDL, який вміщує оператор процесу синхронізації та оператор процесу з описом функціонування ОП.

Спочатку описується оператор процесу синхронізації, який генерує фази алгоритму, наприклад, $\text{cycle} = 0, 1, \dots, L-1$. Потім описується оператор процесу функціонування ОП наступним чином. В ньому окремими операторами описується присвоювання змінним, які відповідають векторам $\mathbf{D}_{Dj}$ із затримками.

Далі в цьому операторі процесу у межах дії умовного оператора, який описує зміну станів за фронтом синхросерії, ставиться оператор `case cycle`, в якому розміщено L альтернатив від 0 до $L-1$. Причому в t -й альтернативі ставляться оператори присвоювання сигналу `urq` результату відповідної операції, яка відмічена парою $\langle \text{urq}, t \rangle$.

Результуюча VHDL – модель ОП може бути протестована та скомпільована стандартним компілятором-синтезатором звичайним порядком. Таким чином, за запропонованим методом формально одержується VHDL-опис результуючого проекту ОП, який придатний для синтезу ОП на ПЛІС або замовлений НВІС і який має мінімізовані апаратні витрати та тривалість тактового інтервалу. Причому власне сама структура фільтра не будується. Завдання побудови структури ОС перекладається на компілятор-синтезатор, наприклад, Synplify фірми Synopsys, який розпізнає семантику операторів мови VHDL, що кодують розділення ресурсів за різними тактами виконання.

Також розроблено способи синтезу буферних схем, які дають змогу проектувати буферні схеми з ефективним використанням бібліотечних компонентів ПЛІС, таких як блоки пам'яті, буфери FIFO з відводами.

ГСПД з квазістатичним розкладом виконання операторів називаються такий ГСПД, для якого момент запуску деякого актора знаходиться динамічно, тому що він залежить від даних. Умовний ГСПД з квазістатичним розкладом виконання операторів взаємно однозначно відповідає певній VHDL-програмі, в якій та чи інша гілка алгоритму вибирається за оператором `if-then-else` або `case`. При цьому можна наполягти, щоб у гілках операторів були розглянуті всі без винятку логічні альтернативи. При такій умові при синтезі ОП з опису мовою VHDL оператори `if-then-else` або `case` можуть відображатися з розділенням ресурсів.

Доведено твердження: якщо в умовному ГСПД підграфи умовних розгалужень відповідають операторам `if-then-else` або `case`, в гілках яких розглянуті всі без винятку логічні альтернативи, що відображаються в єдиний апа-

ратний ресурс, то для такого ГСПД можна скласти статичний розклад. Це твердження також забезпечує мінімізацію апаратних витрат в ОП.

На основі цього твердження розроблено метод відображення циклічних алгоритмів з операторами керування в структуру ОП, який дає змогу відображати як звичайні алгоритми з умовними операторами, так і ієрархічні алгоритми. Розглянуто також застосування відомих способів оптимізації до оптимізації просторового ГСПД, таких, як ресинхронізація, згортання та розкручування ГСПД, організація передуючих обчислень.

Приведено опис програмного засобу Paredit, який призначений для графічного задання, оптимізації та відображення просторового ГСПД в структуру конвеєрного ОП. У ньому впроваджена автоматична мінімізація числа регістрів конвеєрного ОП за методом лівої границі.

У четвертому розділі досліджені питання проектування конвеєрних обчислювачів для ЦОС. Вперше були спроектовані та випробувані конвеєрні процесори ШПФ з основами 8 і 16, які відрізняються від процесорів з основами 2 і 4 меншими кількістю блоків множення та об'ємом пам'яті, а також більшою точністю обчислень. Розроблені процесори двовимірного дискретного косинусного перетворення показали високі параметри швидкодії при низьких апаратних витратах у порівнянні з рядом кращих аналогів.

Метод синтезу конвеєрних ОП впроваджено в програмі генератора фільтрів зі скінченною імпульсною характеристикою, яка генерує моделі фільтрів на VHDL з апаратними витратами, що в середньому, на 25% менші за витрати аналогів та з тактовою частотою, яка в 1,15–1,5 разів вища, ніж у аналогічних фільтрів і досягає 550 МГц в ПЛІС Xilinx Virtex-5. Програма генератора впроваджена у засобі проектування Aldec ActiveHDL.

Досвід проектування ряду цифрових фільтрів, у тому числі й тестового хвильового фільтру 5-го порядку, показав, що метод синтезу конвеєрних ОП на основі ПЛІС дає змогу синтезувати рекурсивні фільтри з періодом L і мінімізованими тактовим періодом і апаратними витратами. Причому, структури фільтрів з $L=2$ та 3 при тій самій продуктивності, що й фільтри з $L=1$, які розроблені традиційним способом, мають до 2-х разів менші апаратні витрати. Це досягнуто завдяки тому, що на відміну від традиційних методів синтезу структур, метод відображення просторового ГСПД дає змогу навіть для алгоритмів з циклічними залежностями, до яких відноситься рекурсивний фільтр, максимально конвеєризувати обчислення і тим самим одержати як мінімальний тактовий період, так і мінімізовані апаратні витрати.

Цифровий фільтр прийнято описувати передаточною характеристикою $H_0(Z)$ від комплексної змінної Z . Члену Z^k в формулі $H_0(Z)$ відповідає затримка на k ітерацій або ланцюжок з k регістрів затримки. Якщо в фільтрі збільшити кількість таких регістрів в n разів, то одержується фільтр з кратними затримками з характеристикою $H_n(Z) = H_0(Z^n)$. Застосовуючи спосіб маскування амплітудо-частотних характеристик послідовно з'єднаних ступенів

фільтрів з кратними затримками, одержують високоякісні цифрові фільтри. ГСПД такого фільтра має вектори-дуги D_{Dj} із затримками величиною n , що дає змогу одержувати на його основі структури фільтрів з максимальною конвеєризацією.

Розроблено спосіб структурного синтезу рекурсивних фільтрів із кратними затримками, який забезпечує мінімізацію тактового інтервалу і апаратних витрат, включаючи мультиплексори при заданому L . Завдяки цьому способу вперше синтезовані рекурсивні цифрові фільтри з параметрами, що динамічно змінюються в широкому діапазоні. У випробуваному в ПЛІС рекурсивному фільтрі нижніх частот частота зрізу змінюється плавно в межах 0,015 – 0,4 від частоти дискретизації, рівень згасання складає не менше 75–80 дБ, а нахил АЧХ у перехідній смузі – не менше 100 дБ/октава. Апаратні витрати складають 706 CLB slices і 3 блоки множення при $L=8$ в ПЛІС Xilinx Virtex2P.

Запропоновано для мінімізації похибок обчислень в ЦОС, а також для вирішення задач лінійної алгебри (ЛА) застосувати представлення даних у вигляді раціональних дробів. Раціональний дріб a/b – це числовий об'єкт, який складається з цілочисельних чисельника a і знаменника b . Цей дріб може з досить високою точністю наближатись до заданого числа $x \approx a/b$. Раціональні дроби мають нескладний апарат арифметичних дій. Так, множення a/b на c/d дорівнює $ac/(bd)$, а додавання цих дробів – $(ad+bc)/(bd)$. При цьому власне ділення цілих чисел не виконується.

Так як розрядність чисельників і знаменників приблизно вдвічі менша, ніж розрядність, яка необхідна для операцій з цілими числами при такій самій точності представлення чисел, то складність апаратних суматорів дробів наближається до складності блоків множення цілих чисел, а складність блоків множення дробів приблизно вдвічі менша, ніж блоків множення цілих чисел. Для збереження широкого динамічного діапазону після арифметичних операцій чисельник і знаменник нормалізуються через зсув вліво на однакову кількість розрядів, яка не перевищує половини їх розрядності.

Застосування такого представлення в процесорі ковзного дискретного перетворення Фур'є і в процесорі для авторегресивного аналізу показало, що арифметика раціональних дробів забезпечує як високу точність обчислень, так і малі апаратні витрати і високу тактову частоту при конфігуруванні процесорів у ПЛІС в порівнянні з аналогічними пристроями з плаваючою комою.

У п'ятому розділі метод синтезу конвеєрних ОС адаптовано до проектування обчислювачів для вирішення задач ЛА.

Запропоновано метод проектування паралельних ОС на основі комплексного використання відображення просторового ГСПД і решітчастого графа алгоритму. На першому етапі синтезу за цим методом базова функція f (чи ряд таких функцій) початкового алгоритму відображається у внутрішню структуру конвеєрного ПЕ. При цьому використовується метод синтезу

конвеєрних ОС, який описано у другому розділі. Результатом етапу є розклад виконання операторів алгоритму обчислення функції f , причому обчислення виконуються в конвеєрному режимі з циклом T_C за час $T_D = pT_C$ тактів, де $p=2,3,\dots$ – число функцій f , які виконуються в ПЕ паралельно. При синтезі структури ПЕ мінімізуються апаратні витрати і період тактового інтервалу.

На другому етапі решітчастий граф алгоритму відображається в структуру процесорної матриці за допомогою відомого методу синтезу систолічних процесорів. Потім до p суміжних вершин ПЕ одержаного графа структури систолічного процесора склеюються в одну вершину, яка виконує функцію f в конвеєрному режимі.

На відміну від відомих методів синтезу процесорних матриць для вирішення задач ЛА запропонований метод забезпечує те, що функції f різного типу з ядра циклів алгоритму можуть виконуватись за довільне число тактів та починати своє виконання неодноразово, що збільшує завантаженість ПЕ системи. Завдяки конвеєризації ПЕ, затримка виконання операторів може бути більшою за один такт, але тривалість такту суттєво скорочується, і тому ПЕ мають більшу продуктивність. Також через розділення ресурсів, коли апаратні блоки виконують схожі оператори ядра циклів у різні проміжки часу, зменшуються апаратні витрати і збільшується завантаженість ОС.

Завдяки використанню методу одержані нові обчислювальні схеми для виконання алгоритмів Гівенса, Холецкого, методу спряжених градієнтів, обернення тріоплицевих матриць та інших, які мають оптимізовану завантаженість своїх ПЕ при високій продуктивності ОС.

Моделюванням встановлено, що вирішення задач ЛА на ПЛІС є ефективним при використанні арифметики раціональних дробів, яка, на відміну від обчислень з плаваючою комою, дає змогу вирішувати задачі ЛА з меншими на 25% апаратними витратами і з більшою на 25% тактовою частотою при такій самій точності обчислень.

Встановлено, що похибка розкладання Холецкого при реалізації в арифметиці раціональних n -разрядних дробів оцінюється як $2^{-n}MN^{1/3}$, де N – розмірність матриці, M – математичне сподівання вхідних даних. Моделюванням встановлено, що ОС з арифметикою дробів при $n = 32$ і $M = (0,01 - 1000)$ має таку саму або кращу точність обчислень, що і ОС з плаваючою комою одинарної точності. Крім того, при реалізації ОС на основі ПЛІС досягається баланс витрат в числі блоків множення і логічних схем. Якщо початкові дані представляють цілочисельну зумовлену задачу ЛА, то при достатній розрядності раціональних дробів, в такій ОС одержуються точні результати.

Після побудови ряду проектів спеціалізованих процесорів для вирішення задач ЛА встановлено, що ці задачі ефективно вирішуються на ПЛІС, і такі ОС конкурують з ОС на мікропроцесорах загального призначення завдяки

тому, що в них у повній мірі реалізується дрібно- та середньозернистий паралелізм задачі.

У шостому розділі показано застосування відображення просторового ГСПД при розробці математичного забезпечення для багатопроцесорних ОС. Представлено спосіб програмної конвеєризації, який одержано адаптацією методу проектування паралельних ОС, що описаний в п'ятому розділі. За цим способом синтезовані процесорна матриця для реалізації алгоритму Гівенса, яка реалізована на сигнальних мікропроцесорах Motorola DSP96002 і програмне забезпечення для її ПЕ мовою асемблера. На відміну від відомого способу програмної конвеєризації, в цьому способі враховується не тільки власне конвеєризація етапів виконання програми, але й конвеєрне виконання команд і міжпроцесорних пересилок даних, завдяки чому зменшуються простої конвеєрних обробляючих блоків ПЕ і міжпроцесорних лінків.

Запропоновано спосіб відображення ГСПД в структуру мультимедійного SIMD-процесора, який реалізований в засобі для трансляції ГСПД в асемблерну програму з командами Intel MMX. Метод оснований на трансформації просторового ГСПД таким чином, щоб він максимальною мірою покривався шаблонами команд з системи команд MMX. Випробування згенерованих програм, таких як програма хвильового фільтру 5-го порядку, показало, що в процесорі PentiumPro середня кількість виконуваних команд протягом одного такту (рівень суперскалярного паралелізму) складає 1,56, а сумарні простої процесора через заповнення буферу перестановки даних, непопадання в кеш тощо мінімізовані до 10%, що свідчить про високу ефективність методу.

У висновках зроблено підсумок результатів дисертаційної роботи.

ОСНОВНІ РЕЗУЛЬТАТИ ТА ВИСНОВКИ

Робота присвячена побудові методології проектування апаратних і апаратно-програмних ОС для обробки потоків даних, які адаптовані для реалізації в ПЛІС, НВІС або багатопроцесорних ОС. В роботі отримані такі теоретичні та практичні результати:

1. Створена розширена класифікація моделей представлення паралельних алгоритмів для обробки потоків даних, на основі якої встановлено, що для синтезу апаратних пристроїв обробки таких потоків найбільш придатна модель графа синхронних потоків даних (ГСПД). Причому слід вибирати однорідний ГСПД через відсутність в ньому блокувань, можливість еквівалентних перетворень в нього інших типів ГСПД, простоту його опису мовою VHDL і подальшого відображення в апаратні ресурси, включаючи відображення в ПЛІС та мікропроцесорні системи.

2. Вперше запропонована модель просторового ГСПД для представлення алгоритмів обробки потоків даних, яка на відміну від інших моделей,

дає змогу сумісно виконувати етапи вибору ресурсів, складання розкладу виконання операторів і призначення операторів ресурсам, забезпечує ефективний спосіб кодування рішення задачі синтезу ОС. Модель наочно представляє рішення в графічному вигляді, сприяє більш ефективній оптимізації з використанням як відомих методів оптимізації, так і ручного удосконалення проекту ОС. Сформульовані і доведені умови коректності та еквівалентності просторового ГСПД, а також вимоги до еквівалентних і функціонально-еквівалентних перетворень цієї моделі. Розроблений алгоритм перетворення просторового ГСПД у відповідний граф залежності по даних.

3. Вперше розроблено метод відображення просторового ГСПД в структуру ОС і розклад виконання операторів, який на відміну від більшості методів синтезу ОС, дає змогу суміщувати етапи вибору ресурсів, складання розкладу і призначення операторів і даних ресурсам, завдяки чому спрощується пошук оптимального рішення.

4. Розроблено удосконалений метод синтезу конвеєрних обчислювачів для ПЛІС на рівні регістрових передач, який є формалізованим і відрізняється тим, що синтезовані пристрої мають мінімізовані апаратні витрати та тривалість тактового інтервалу при заданому циклі обчислень L , причому структурне рішення одержується без побудови власне структури ОС, завдяки чому оптимізація ОС виконується направлено і має зменшену трудомісткість. Створено версію програмного засобу Paredit для автоматизованого синтезу спеціалізованих ОС на основі відображення просторового ГСПД.

5. Вперше запропоновано метод відображення ГСПД з операторами керування в структуру ОС, який дає змогу реалізувати в ПЛІС як прості, так і ієрархічні алгоритми.

6. За допомогою методу синтезу конвеєрних обчислювальних пристроїв для ПЛІС вперше були спроектовані конвеєрні процесори ШПФ з основами 8 і 16, які відрізняються від інших процесорів ШПФ меншими апаратними витратами, а також більшою точністю обчислень. Розроблені за цим методом процесори прямого і зворотнього дискретного косинусного перетворення показали високу швидкодію при низьких апаратних витратах в порівнянні з рядом кращих аналогів.

7. На основі методу синтезу конвеєрних обчислювальних пристроїв для ПЛІС розроблено удосконалений спосіб побудови функціональних схем фільтрів зі скінченною імпульсною характеристикою та вперше створено методiku проектування рекурсивних фільтрів з кратними затримками, реалізованих на ПЛІС, що мають характеристики продуктивності та відношення продуктивність-апаратні витрати, які у 1,2 – 2 рази більше, ніж у фільтрів, спроектованих традиційними способами. У результаті синтезу ряду практичних і тестових структур цифрових фільтрів встановлено, що метод синтезу конвеєрних обчислювальних пристроїв для ПЛІС дає змогу проектувати фільтри з заданим періодом L та мінімальними апаратними витратами. Причому структури рекурсивних фільтрів з $L = 2$ або 3 мають до

двох разів менші апаратні витрати, ніж фільтри з $L = 1$, які одержані традиційним методом і мають таку саму пропускну спроможність.

8. Вперше запропоновано метод проектування паралельних ОС на основі сумісного використання відображення просторового ГСПД і решітчастого графу алгоритму, який, на відміну від методів синтезу систолічних структур, дає змогу підвищити тактову частоту ОС завдяки конвеєризації процесорних елементів ОС і мінімізувати апаратні витрати через розділення ресурсів. Застосування цього методу дало змогу побудувати нові структурні рішення ОС для виконання алгоритмів Гівенса, Холецького, методу спряжених градієнтів, обернення теплицевих матриць та інших, які мають максимальну завантаженість своїх процесорів (до 80% від повної при виконанні алгоритму Гівенса) при високій продуктивності ОС. При реалізації одержаних структур ОС в ПЛІС в них, на відміну від більшості паралельних ОС, в повній мірі реалізовано дрібнозернистий паралелізм і тому досягається максимальна продуктивність та мінімальні апаратні простоти.

9. Для вирішення задач ЛА і ЦОС вперше запропоновано представлення даних у вигляді раціональних дробів. Його застосування в процесорі для ковзного перетворення Фур'є, процесорі для авторегресійного аналізу сигналів дало змогу одержати як високу точність обчислень, так і невеликі апаратні витрати та високу тактову частоту. Застосування такого представлення для побудови спецпроцесорів на основі ПЛІС, призначених для обернення теплицевих матриць, виконання алгоритмів Дарбіна, Холецького та інших показало, що в ОС зменшуються апаратні витрати принаймні на 25% і збільшується продуктивність на 25% у порівнянні з ОС, в якій використовується плаваюча кома при такій самій точності обчислень.

10. Запропоновано удосконалений метод програмної конвеєризації багатопроесорної ОС на основі відображення просторового ГСПД, який мінімізує простоти процесорів завдяки як програмній конвеєризації, так і організації систолічних обчислень. На відміну від традиційної програмної конвеєризації, новий метод в повній мірі враховує затримки конвеєрів команд, доступу до оперативної пам'яті і міжпроцесорних пересилок.

11. Розроблено метод відображення просторового ГСПД в структуру SIMD-процесора, який, завдяки властивостям відображення ГСПД в багатопроесорну ОС, вперше дав змогу формально складати високоефективні асемблерні програми обробки потоків даних з мінімізованими простоями як конвеєрів команд, так і пам'яті процесора. Розроблена на основі методу відображення просторового ГСПД версія транслятора виконує генерацію асемблерних кодів для мультимедійного сопроцесора, які забезпечують мінімізацію простоїв такого процесора, як Pentium-Pro, до рівня 10%.

СПИСОК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

1. Сергиенко А.М. VHDL для проектирования вычислительных устройств. –К.: –"ДиаСофт". –2003. –210 с.
2. Сергиенко А.М. Спецпроцессоры для авторегрессионного анализа сигналов // Электронное моделирование. – 2010. –Т.32, №2. –С. 87-96.
3. Сергиенко А.М. Пространственный граф синхронных потоков данных// Вісник НТУУ «КПІ»: “Інформатика, управління та обчислювальна техніка”: зб. наук. праць. –2010. –№51. –С. 40-46.
4. Сергиенко А.М., Динамически перестраиваемые цифровые фильтры на ПЛИС / А.М. Сергиенко, Т.М. Лесик // Электронное моделирование. –2010. –Т.32, №6. –С. 47-56. – *Запропоновано структури фільтрів, які перестроюються динамічно і синтезуються відображенням просторового ГСПД.*
5. Сергиенко А.М. Реализация алгоритма Холецкого на ПЛИС/ А.М. Сергиенко, В.Л.Лепеха, Т.М. Лесик. // Вісник НТУУ «КПІ»: “Інформатика, управління та обчислювальна техніка”: зб. наук. праць. –2010. –Т.52, –С. 40-45. – *Автором запропоновано представлення даних і структуру ОС.*
6. Сергиенко А.М. Реализация перестраиваемых рекурсивных цифровых фильтров на ПЛИС / А.М. Сергиенко, Т.М. Лесик // Вісник НТУУ «КПІ»: “Інформатика, управління та обчислювальна техніка”: зб. наук. праць. –2010. –№52. –С. 47-51. – *Автором запропоновано спосіб реалізації фільтрів з кратними затримками.*
7. Сергієнко А.М. Деякі особливості проектування мікроконтролерів для СНК / А.М.Сергієнко, В.Л.Лепеха // Вісник НТУУ «КПІ»: “Інформатика, управління та обчислювальна техніка”: зб. наук. праць. –2009. –№50. –С. 70-73. – *Проаналізована будова ПЛІС і вироблено критерій ефективності структури, які в ній реалізовані.*
8. Сергиенко А.М. Применение рациональных дробей в специализированных вычислителях // Вісник НТУУ «КПІ»: “Інформатика, управління та обчислювальна техніка”: зб. наук. праць. –2009. –№50. –С. 74-77. –*Запропоновано представлення даних в спеціалізованих обчислювачах.*
9. Сергиенко А.М. Спосіб демодуляції сигналів з багаточастотною модуляцією / А.М.Сергієнко, В.М. Мелковська // Вісник НТУУ “КПІ”: “Інформатика, управління та обчислювальна техніка”: зб. наук. праць. –2008, –№48. –С. 82–84. –*Запропоновано алгоритм кожного дискретного перетворення Фур’є в арифметиці дробів і виконано синтез схеми демодулятора.*
10. Сергієнко А.М. Проектування обчислювачів з регістровими затримками // Вісник НТУУ “КПІ”: “Інформатика, управління та обчислювальна техніка”: зб. наук. праць. –2008, –№49. –С. 73–76.
11. Сергиенко А.М. Алгоритмические модели обработки потоков данных / А.М. Сергиенко, В.П. Симоненко // Электронное моделирование. –2008. –Т.30, №6. –С. 49–60. – *Автору належать критичний огляд моделей для задання алгоритмів обробки потоків даних і класифікація таких моделей.*
12. Сергиенко А.М. Отображение периодических алгоритмов в программируемые логические интегральные схемы / А.М. Сергиенко, В.П. Симо-

ненко // Электронное моделирование. –2007. –Т.29. –№2. –С. 49–61. –*Розроблено метод відображення ГСПД в структуру ПЛІС, а також метод проектування структур цифрових фільтрів з багатократними затримками.*

13. Сергієнко А.М. Досконалий кістяк графа алгоритма // Вісник НТУУ “КПІ”: “Інформатика і обчислювальна техніка”: зб. наук. праць. – 2007. –№46. –С. 62–67.

14. Сергієнко А.М. Синтез структур для виконання періодичних алгоритмів з операторами керування // Вісник НТУУ “КПІ”, сер. Інформатика, управління та обчислювальна техніка. – 2007. –№ 47. –С. 45-48.

15. Сергієнко А.М. Спецпроцесори для двовимірного дискретного косинусного перетворення / А.М. Сергієнко, В.Л. Лепеха, Т.М. Лесик // Вісник НТУУ “КПІ”: “Інформатика і обчислювальна техніка”: зб. наук. праць. –2007. –№ 47. –С. 49-52. – *Автором виконано проектування процесорів методом відображення ГСПД.*

16. Сергиенко А.М. Применение арифметики рациональных дробей для реализации метода сопряженных градиентов // Электронное моделирование. –2006. –Т.28, № 1. – С.33–41.

17. Сергієнко А.М. Конфігурована обчислювальна система для вирішення задач лінійної алгебри / О.М. Клименко, А.М. Сергієнко, Ю.В. Шевченко, С.Г. Овраменко // Электронное моделирование. –2005. –Т.27, №1. –С. 109–114. –*Запропоновано метод відображення ГСПД в конвеєрну ОС.*

18. Сергиенко А.М. Отображение алгоритма QR-разложения Гивенса в многопроцессорную систему // Электронное моделирование. –2004. –Т.26, №5. – С. 43–53.

19. Сергиенко А.М. Отображение алгоритмов обработки изображений в программируемых логических интегральных схемах / В.Л. Лепеха, А.М.Сергиенко, Ю.В. Шевченко // Электронное моделирование. –2004. –Т.26, –№4. –С. 75–82. –*Автором запропоновано структуру процесорної матриці і алгоритми, які реалізовано в ній методом відображення ГСПД.*

20. Сергиенко А.М. Особенности VHDL как языка параллельного программирования // Электронное моделирование. –2003. –Т 25, № 3.–с.115-123.

21. Сергиенко А.М. Методика проектирования цифровых фильтров с помощью VHDL // Моделирования та інформаційні технології: зб. наук. праць. Ін-т проблем моделювання в енергетиці, НАН України. –2002. –Вип. 12. –С. 99–107.

22. Сергієнко А.М. Генератор модулів фільтрів для ПЛІС / В.І. Василенко, О.М. Клименко, Л.М. Логінова, А.М. Сергієнко // Моделирования та інформаційні технології: зб. наук. праць. Ін-т проблем моделювання в енергетиці, НАН України. –2002. –Вип.12. –С. 13–18. –*Автором запропоновано методику автоматичного синтезу структур фільтрів.*

23. Сергиенко А.М. Отображение регулярных алгоритмов в структуры специализированных процессоров / Ю.С. Каневский, С.Г. Овраменко, А.М. Сергиенко // Электронное моделирование. –2002. –Т.24, №2. –С. 46–59. –*Автору належить метод відображення ГСПД в конвеєрну структуру.*

24. Сергиенко А.М. Последовательная и параллельная реализация метода верхней релаксации при решении задач математической физики / Каневский Ю.С., Лепеха В.Л., Сергиенко А.М. // Электронное моделирование. –2000. –Т.22, №6. –С. 3–12. –*Автором запропоновано структуру процесорної матриці і алгоритми, які реалізовано в ній.*

25. Сергиенко А.М. Последовательная и параллельная реализация метода минимальных невязок при решении задач математической физики / Р.Выжиковски, В.Л. Лепеха, А.М. Сергиенко // Электронное моделирование. –1999. –Т.21, №6. –С. 35–47. –*Автором запропоновано структуру процесорної матриці і алгоритми, які реалізовано в ній.*

26. Сергиенко А.М. Структурное проектирование рекурсивных цифровых фильтров / Ю.С. Каневский, Л.М. Логинова, А.М. Сергиенко // Электронное моделирование. –1995, – Т.17, № 3. –С. 18–22. –*Автором запропоновано метод відображення ГСПД в структуру фільтра .*

27. Sergiyenko A. Application specific processors for the autorgressive signal analysis / O. Maslennikow, A. Sergiyenko, N. Maslennikowa, P. Ratushnyak, M. Wozniak // 8-th Int. Conf. “Parallel Processing and Applied Mathematic”, PPAW’2009. Part I: “Lecture Notes in Computer Science”. –Berlin: Springer, –2010, –V. 6068. –Р. 80–86. –*Запропоновано виконання алгоритмів в арифметиці раціональних дробів, які реалізовані в конвеєрному ПЕ.*

28. Сергиенко А.М. Вычислительные модели параллельных алгоритмов, реализуемых аппаратно / А.М. Сергиенко, В.Л. Лепеха, Т.М. Лесик // 36. праць ювілейної міжн. наук.-практ. конф. 50-річчя створення каф. ОТ. –Київ, 6-8 квітня 2010. –Київ: НТУУ «КПІ». –2010. –С. 125-127. – *Автором зроблено критичний огляд методів структурного синтезу ОС.*

29. Сергиенко А.М. Динамически перестраиваемые цифровые фильтры на ПЛИС / А.М. Сергиенко, Т.М. Лесик // 36. матеріалів міжн. наук. конф. «Моделювання-2010». –Київ: ІПМЕ НАН України. –2010. –Т.3. –С. 161-166. –*Запропоновано структури цифрових фільтрів, які перестроюються динамічно і які синтезуються відображенням просторового ГСПД.*

30. Sergiyenko A.M. Tensor approach to the application specific processor design / A.M. Sergiyenko, O. Maslennikow, Y. Vinogradow // Proc.10th Int. Conf. “The Experience of Designing and Application of CAD Systems in Microelectronics”, CADSM’2009, 24-28 Feb. 2009. –IEEE Library. –2009. –Р. 146-149. –*Зроблена критична оцінка тензорної методології і запропоновано метод відображення ГСПД в структуру спецпроцесора.*

31. Sergiyenko A. Parallel Implementation of Cholesky LL^T Algorithm in FPGA-Based Processor / A. Sergiyenko, O. Maslennikow, V. Lepekha, A. Tomas, R. Wyrzykowski // “Lecture Notes in Computer Science”. –Berlin: Springer. –2008. –V. 4967. –Р. 137-147. –*Запропоновано виконання алгоритму Холецкого, який реалізований з застосуванням методу відображення ГСПД.*

32. Сергиенко А.М. Сперцпроцессоры для авторегрессионного анализа сигналов / А.М. Сергиенко, В.Л. Лепеха, О.В. Масленников // Тр. межд. конф. Моделирование–2008. –14-16 мая 2008. –Киев: ИПМЭ НАН Украины. –2008.

–T2. –С. 499-503. –*Автором синтезовано структури спецпроцесорів методами відображення ГСПД.*

33. Sergiyenko A. Generator opisow VHDL blokow operacyjnych dzialajacych w arytmetyce ulamkowej / A. Sergiyenko, O. Maslennikov, P. Ratuszniak // Prace XI Konferencji Naukowej „Reprogramowalne układy cyfrowe”, RUC’2008. – Szczecin, Poland. –2008. –Р. 167-172. –*Автором запропоновано спосіб проектування цифрових фільтрів з кінечною імпульсною характеристикою.*

34. Sergiyenko A. Mapping DSP Algorithms into FPGA / A.Sergiyenko, T. Lesyk, O. Maslennikov // Proc. of IEEE East-West Design & Test Symp., EWDTS’08. –Lviv, October 9-13, 2008. –Kharkov: KNURE. –2008. –Р. 343-348. –*Запропоновано метод відображення алгоритмів ЦОС в ПЛІС.*

35. Сергиенко А.М., Лепеха В.Л. Масленников О.В. Сперцпроцессоры для авторегрессионного анализа сигналов / А.М. Сергиенко, В.Л. Лепеха, О.В. Масленников // Тр. межд. конф. “Моделирование–2008”. –14-16 мая, 2008. – Киев: ИПМЭ НАН Украины. –2008. –Т2. –С. 499–503. –*Автором синтезовано структури спецпроцесорів методами відображення ГСПД в ПЛІС.*

36. Сергиенко А.М. Структурное проектирование рекурсивных цифровых фильтров / А.М. Сергиенко, О.В. Масленников // Тр. межд. конф. “Автоматизация проектирования дискретных систем”, CADDD’07. –Минск. –2007. –С.92–99. –*Запропоновано метод відображення ГСПД алгоритму рекурсивної фільтрації в конвеєрну структуру .*

37. Sergiyenko A. Implementation of Linear Algebra Algorithms in FPGA-based Fractional Arithmetic Units / A. Sergiyenko, O. Maslennikov, P. Ratuszniak // Proc. 9-th Int. Conf. “The Experience of Designing and Application of CAD Systems in Microelectronics”, CADSM’2007. –Lviv, Polyana, –20–24 Feb., 2007. –2007. –р.228–234. –*Автором запропоновано виконання чисельних алгоритмів з застосуванням арифметики раціональних дробів.*

38. Sergiyenko A. Implementation of Cholesky LL^T -decomposition algorithm in FPGA-based rational fraction parallel processor / A. Sergiyenko, O. Maslennikov, P. Ratuszniak // Proc. 14-th Int. Conf. "Mixed Design of Integrated Circuits and Systems", MIXDES’2007. – Ciechocinek, Poland. –21–23 June, 2007. – Technical University of Łódź. – 2007. –Р. 287–292. –*Запропоновано виконання чисельних алгоритмів з застосуванням арифметики раціональних дробів.*

39. Sergiyenko A. FPGA Implementation of the Conjugate Gradient Method / Maslennikov O., Lepekha V., Sergiyenko A. // Lecture Notes in Computer Science. – Berlin: Springer. –2006. –V. 3911. –Р. 526–533. –*Автором виконано реалізацію алгоритму методом відображення ГСПД в багатопроцесорну конвеєрну структуру.*

40. Sergiyenko A. Mapping DSP algorithms into FPGA / O. Maslennikov, A. Sergiyenko // Proc. IEEE Int. Symposium on Parallel Computing in Electrical Engineering, PARELEC’06. –Poland: Bialystok, 13-17 Sept., 2006. –2006. –Р. 208-213. –*Автором запропоновано метод відображення ГСПД алгоритму ЦОС в конвеєрну структуру .*

41. Sergienko A. Sparse matrix solving on FPGA. / Sergienko A. Klimenko A. // Тр. межд. конф. “Информационные технологии в управлении энергетическими системами”. – Киев, 18-19 окт., 2005. – Киев: ИПМЭ НАН Украины. – 2005. – С. 47–48. – *Запропоновано виконання алгоритму вирішення тьоплицевої системи рівнянь з застосуванням арифметики раціональних дробів.*
42. Сергиенко А.М. Что делать с одним миллиардом транзисторов. Argc& Argv. Программирование/информационные технологии/стандарты. – 2004. – Т. 57, № 6. – С. 9-19.
43. Sergiyenko A. Configurable Microprocessor Array for DSP Applications / O. Maslennikov, Ju. Shevtshenko, A. Sergiyenko // Lecture Notes in Computer Science. Berlin: Springer. – V. 3019. – 2004. – P. 36–41. – *Автором запропоновано структуру процесорної матриці і алгоритми, які реалізовано в ній.*
44. Sergiyenko A.M. Configurable encryption processor / V.I. Vasylenko, V.L. Lepekha, A.M. Sergiyenko // Proc. Int. Conf. SCALNET'2004. – in Bulletin of Kremenchuk State Polytechn. Univ. “Transactions of SCALNET'2004”. – 2004. – p. 121–123. – *Автором синтезовано конвеєрну схему, що реалізує алгоритм Монгмері відображенням ГСПД.*
45. Sergiyenko A. Optymizacja architektury mikrokontrolerow przeznaczonych do realizacji w układach FPGA / Maslennikow O., Sergiyenko A., Lacki M. // Prace VI Konferencji „Reprogramowalne układy cyfrowe”, RUC'2003, Szczecin, 2003. – 2003. – S. 87-92. – *Досліджено елементну базу ПЛІС і запропоновано критерії ефективності ОС, що реалізована в ПЛІС.*
46. Sergiyenko A. Configurable microcontroller array / Maslennikov O., Shevtshenko Ju., Sergiyenko A. // Proc. of the 3-rd Int. Conf. on Parallel Computing in Electrical Engineering. PARELEC'2002. – Warsaw, Poland. – 22–25 Sept. – 2002. – P. 47–49. – *Запропоновано структуру процесорної матриці і алгоритми, які реалізовано в ній.*
47. Sergiyenko A. Implementation of Givens QR Decomposition in FPGA / A. Sergiyenko, O. Maslennikov // Lecture Notes in Computer Science. – Berlin: Springer. – 2002. – V. 2328. – P. 453–459. – *Автором запропоновано нові версії алгоритма Гівенса, які відображаються методом програмної конвеєризації.*
48. Sergiyenko A. FIR filter soft core generator / A. Sergiyenko, V. Vasylenko, O. Maslennikow // Prace IV Konferencji Krajowej „Reprogramowalne układy cyfrowe”, RUC'2001. – Szczecin, Poland. – 2001. – P. 167-172. – *Запропоновано методику синтезу структур фільтрів та їхніх блоків множення.*
49. Sergiyenko A. A Method for Mapping DSP Algorithms into Pentium MMX™ Architecture. / Sergiyenko A., Kaniewski J., Arefjev A., Kortshev D. // Proc 3-d Int. Conf. On Parallel Processing and Applied Mathematics, PPA'M'99. – Kazimierz Dolny, Poland. – Sept. 14-17., 1999. – 1999. – P. 348–356. – *Запропоновано метод програмування SIMD-систем шляхом програмної конвеєризації.*
50. Sergiyenko A. Implementation of IIR Digital Filters in FPGA / A. Sergiyenko, V. Lepekcha, J. Kaniewski, P. Sołtan // Pracy II Konferencji Krajowej

„Reprogramowalne układy cyfrowe”, RUC'99, Szczecin, 1999. –1999. –Р. 233-239. –*Автором запропоновано проектування рекурсивних фільтрів за допомогою відображення ГСПД.*

51. Сергиенко А.М. Локально связанные вычисления - состояние и перспективы / Ю.С. Каневский, А.М. Сергиенко // *Праці міжн. симпозіума "Комп'ютери у Європі. Минуле, сучасне та майбутнє". –Київ: ІК НАН України. –1998. –с.460–467. –Зроблено критичний огляд методів синтезу ОС і систолічних процесорів.*

52. Sergienko A. A method for mapping DSP algorithms into application specific structures / Ju. Kanevski, A. Sergienko, O. Maslennikov, R. Wyrzykowski // *Proc. 24-th Euromicro Conf. on Parallel and Distributed Processing, Euromicro'98, Sweden, Vasteras. –IEEE Comp. Soc. Press. –1998. –V. 1. –P.365-371. –Запропоновано метод відображення ГСПД алгоритму ЦОС в конвеєрну структуру.*

53. Sergiyenko A. Mapping Regular Algorithms into Processor Arrays Using Software Pipelining / A. Sergiyenko, J. Kaniewski, O. Maslennikow, R. Wyrzykowski // *Proc.Int. Conf. On Parallel Computing in Electrical Engineering, PARELEC'98. – Poland, Bialystoc. –Sept. 2-5, 1998. –1998. –P.197–200. –Запропоновано метод програмної конвеєризації на основі відображення ГСПД.*

54. Sergienko A.M. A Method for Mapping Unimodular Loops into Application Specific Parallel Structures / Kanevski Ju.S., Sergienko A.M., Guzinski A. // *Proc. of the 2-nd Intern. Conf. "Parallel Proc. and Appl. Math. PPAM'97" - Zakopane, Poland, 1997. –1997. –P. 362-371. –Запропоновано задання гнізда циклів за допомогою ГСПД і метод його відображення.*

55. Sergienko A.M. Processor Array for Signal Computing and Numerical Applications / Kanevski Ju.S., Sergienko A.M., Maslennikov O.V. // *Proc. 7-th Int. Conf. "Parcella'96". –Berlin: Akademie Verlag, Mathematical Research. –1996. –V. 96. –P. 47-58. –Запропоновано алгоритми, які реалізовано в процесорній матриці методом програмної конвеєризації.*

56. Sergienko A.M. Structured Design of Recursive Digital Filters / Ju.S. Kanevski, L.M. Loginova, A.M. Sergienko // *Engineering Simulation. –1996. –V.13. –р. 381–389. –Автором запропоновано метод синтезу рекурсивних фільтрів відображенням ГСПД.*

57. Sergienko A.M. Processor Array for Signal Computing and Numerical Applications / Ju.S. Kanevski, A.M. Sergienko, O.V. Maslennikov // *Proc. of the Int. Conf. "ICSPAT'95" –Boston. –1995. –P. 1563–1567. –Запропоновано алгоритми, що реалізовані в ОС методом програмної конвеєризації.*

58. Sergienko A.M. A Method for the Structural Synthesis of Pipelined Array Processors. / Ju.S. Kanevski, A.M. Sergienko, H. Piech // *Proc. of the First Intern. Conf. "Parallel Proc. and Appl. Math. PPAM'94". – Poland, Czestochova. –1994. –P. 100–109. –Автором запропоновано метод відображення ГСПД в конвеєрну структуру.*

59. Sergienko A.M. Mapping Numerical Algorithms into Multipipelined Processors / Ju.S. Kanevski, A.M. Sergienko // Proc. of the Int. Workshop "Parallel Numerics'94", Smolenice. –1994. –P.192–202. –*Запропоновано метод відображення ГСПД в багатопроцесорну структуру з конвеєрними ПЕ.*

60. Sergiyenko A. Real time autoregressive processors / A. Sergiyenko, V. Simonenko, V. Lepekha // Abstracts of Int. Conf. "CODATA-21". –Kyiv: NTUU "KPI". –2008. –P.229-230. –*Запропоновано виконання чисельних алгоритмів з застосуванням арифметики раціональних дробів і методу відображення ГСПД.*

61. Sergiyenko A. Implementation of linear algebra algorithms in the FPGA-based rational fraction arithmetic processor / A. Sergiyenko, O. Maslennikow // Abstracts of Int. Conf. "CODATA-21". –Kyiv: NTUU "KPI". –2008. –P. 224. –*Синтезовано структури спецпроцесорів методами відображення ГСПД.*

62. Сергиенко А.М. САПР конвейерных вычислительных структур. / Ю.С. Каневский, Л.М. Логинова, А.М. Сергиенко // Тез. докл. межд. конф. "Автоматизация проектирования дискретных систем", CADDD'95. –Минск. –1995. –С. 143. –*Запропоновано метод відображення ГСПД в конвеєрну структуру.*

АНОТАЦІЇ

Сергієнко А.М. Моделі, методи і засоби синтезу обчислювальних систем для обробки потоків даних. –Рукопис.

Дисертація на здобуття наукового ступеня доктора технічних наук за спеціальністю 05.13.05 – Комп'ютерні системи та компоненти. – Національний технічний університет України "Київський політехнічний інститут", м. Київ, 2011 р.

Розроблено ряд методів синтезу конвеєрних обчислювальних систем (ОС) через відображення просторового графу синхронних потоків даних в структуру ОС та розклад виконання операцій в ній, який забезпечує при заданому періоді виконання алгоритму в L тактів мінімізацію тривалості тактового інтервалу, об'єму пам'яті, а також апаратних витрат, включаючи кількість процесорних елементів, їх регістрів, мультиплексорів, з'єднань. Розроблені методи синтезу конвеєрних ОС для програмованих логічних інтегральних схем (ПЛІС), проектування ОС з використанням мови VHDL, відображення алгоритмів обробки синхронних потоків даних з операторами керування, спосіб синтезу буферних схем.

Методи перевірені при проектуванні ОС для цифрової обробки сигналів і вирішення задач лінійної алгебри, які реалізовані в ПЛІС, параметри яких переважають або не гірше параметрів кращих зразків. Розроблені також методи програмної конвеєризації і програмування багатопроцесорних ОС.

Ключові слова: граф потоків даних, ПЛІС, VHDL, відображення алгоритму, цифрова обробка сигналів.

Сергиенко А.М. Модели, методы и средства синтеза вычислительных систем для обработки потоков данных. –Рукопись.

Диссертация на соискание ученой степени доктора технических наук по специальности 05.13.05 – Компьютерные системы и компоненты. – Национальный технический университет Украины "Киевский политехнический институт", г. Киев, 2011г.

Объектом исследований в диссертационной работе являются параллельные вычислительные системы (ВС) для обработки потоков данных и дискретных сигналов. Предметом исследований являются математические модели алгоритмов обработки потоков данных и методы их отображения в ВС с ориентацией на реализацию в программируемых логических интегральных схемах (ПЛИС). Целью работы является улучшение эффективности указанных методов отображения, в частности, минимизации аппаратных затрат при заданном периоде выполнения алгоритма L тактов.

Предложены классификация вычислительных моделей для задания и отображения итерационных алгоритмов и их представление на модели графа синхронных потоков данных (ГСПД) с помощью языка VHDL. Выработан критерий эффективности ВС, сконфигурированной в ПЛИС, учитывающий свойства как периодических алгоритмов, так и элементной базы. Предложена модель пространственного ГСПД, позволяющая улучшить оптимизацию ВС.

Разработан ряд методов синтеза конвейерных ВС путем отображения пространственного ГСПД в структуру ВС и ее расписание, который обеспечивает при заданном L минимизацию длительности тактового интервала, аппаратных затрат, включая количество процессорных элементов, их регистров, мультиплексоров, межсоединений, объем памяти а также энергопотребление. Методы основаны на том, что ГСПД представляется в многомерном целочисленном пространстве в виде конфигурации алгоритма (КА), которая является пространственным ГСПД, состоящей из матрицы векторов-вершин K , соответствующих операторам алгоритма, матрицы векторов-дуг, отвечающих непосредственным информационным связям между операторами или переменным и матрицы инцидентности ГСПД.

Поиск оптимального структурного решения ВС заключается в нахождении такой матрицы K , которая минимизирует заданный критерий качества и удовлетворяет ряд требований к КА, которые учитывают как временные ограничения, так и периодичность алгоритма с параметром L . На первом этапе синтеза ВС ГСПД размещается в многомерном пространстве по координатам которого отсчитываются номер процессорного элемента, где исполняется оператор, его тип и такт его исполнения с учетом определенных ограничений. При этом число вершин, отображаемых в один ПЭ должно стремиться к L , что обеспечивает минимизацию аппаратных затрат.

На втором этапе КА уравнивается вставкой в дуги ГСПД промежуточных вершин задержки, в результате чего вершины образуют ярусы, рас-

стояние между которыми по координате времени равно 1. После уравнивания КА выполняется ее оптимизация путем взаимных перестановок вершин из одного яруса с минимизацией числа регистров и числа входов мультиплексоров в результирующей структуре. Полученная КА отображается в граф структуры ВС или описывается на VHDL как искомая модель ВС.

Разработаны также метод синтеза конвейерных ВС для ПЛИС, включая метод проектирования с использованием языка VHDL, метод отображения алгоритмов с операторами управления, а также способы синтеза буферных схем, отображения иерархических условных алгоритмов.

Методы адаптированы к синтезу ВС для цифровой обработки сигналов. При этом построены способ синтеза функциональных схем фильтров с конечной импульсной характеристикой, реализуемых в ПЛИС и программа генератора таких фильтров. Предложен способ структурного синтеза рекурсивных фильтров с кратными задержками. Методы проверены при синтезе ряда структурно-функциональных схем конвейерных ВС для цифровой обработки сигналов, таких как процессоры быстрого преобразования Фурье, дискретного косинусного преобразования, цифровые фильтры, параметры которых превосходят или не хуже параметров лучших образцов.

Разработан метод проектирования параллельных ВС и ряд структур ВС, полученных с его помощью для решения задач линейной алгебры на основе комбинированного отображения пространственного ГСПД и решетчатого графа алгоритма. Предложено представление данных в виде рациональных дробей, благодаря чему осуществляется реализация ВС для решения задач линейной алгебры на базе ПЛИС с минимальными аппаратными затратами, высокой производительностью и требуемой точностью вычислений.

Разработан метод программной конвейеризации на основе пространственного ГСПД, который позволяет составлять эффективные программы обработки потоков данных в многопроцессорных ВС. Также предложен метод отображения пространственного ГСПД в структуру SIMD-процессора.

Методы проектирования параллельных ВС обеспечивают возможности: снижение трудоемкости и сокращение сроков получения множества оптимизированных альтернативных структурных решений при проектировании конвейерных ВС, реализуемых как в ПЛИС, СБИС, так и в одно- и многопроцессорных ВС; повышение загруженности ресурсов и памяти ВС; разработка ВС цифровой обработки сигналов с заданными параметрами. Методы могут служить основой курса обучения проектированию специализированных ВС для студентов, аспирантов и специалистов.

Ключевые слова: граф потоков данных, ПЛИС, VHDL, отображение алгоритма, цифровая обработка сигналов.

Sergiyenko A.M. Models, methods, and tools for the computer system synthesis intended for data flow computations. –Manuscript.

Thesis for a doctoral degree in engineering sciences, speciality: 05.13.05 – Computer systems and components. – National Technical University of Ukraine "Kiev Polytechnic Institute", Kiev, 2011.

A set of methods for pipelined computer systems is developed. The methods are based on mapping the spatial synchronous dataflow graph (SDF) into the computer structure, and its schedule. Methods provide minimizing clock period, hardware volume, register number, multiplexor complexity, and energy consumption by the fixed calculation period of L clock cycles. The methods for the pipelined datapath synthesis are developed, which are intended for FPGA, and widely use VHDL as the working language. The methods for mapping Boolean SDF, for buffer synthesis, for hierarchical SDF mapping are proposed as well.

The methods are proven by the design of a set of IP core modules for digital signal processing and linear algebra problem solving, which are configured in FPGA. The module parameters are equal or higher than ones for the best analogous modules. Methods of the software pipelining and SIMD system programming are proposed too.

Keywords: data flow graph, FPGA, VHDL, algorithm mapping, digital signal processing.