

Міністерство освіти і науки України
Національний університет “Львівська політехніка”

Мельник Віктор Анатолійович

УДК 004.056.55

**САМОКОНФІГУРОВНІ
ВИСОКОПРОДУКТИВНІ КОМП’ЮТЕРНІ СИСТЕМИ**

Спеціальність 05.13.05 – Комп’ютерні системи та компоненти

Автореферат

дисертації на здобуття наукового ступеня
доктора технічних наук

Львів – 2013

Дисертацією є рукопис

Робота виконана на кафедрі безпеки інформаційних технологій
Національного університету „Львівська політехніка”
Міністерства освіти і науки України

Науковий консультант: доктор технічних наук, професор
Стадник Богдан Іванович,
заслужений винахідник України,
Директор Інституту комп’ютерних технологій,
автоматики та метрології
Національного університету “Львівська політехніка”.

Офіційні опоненти: доктор технічних наук, професор
Тарасенко Володимир Петрович,
заслужений діяч науки і техніки України,
лауреат Державної премії в галузі науки і техніки,
завідувач кафедри спеціалізованих комп’ютерних систем
та системного програмування
Національного технічного університету України
«Київський політехнічний інститут»;

доктор технічних наук, старший науковий співробітник
Опанасенко Володимир Миколайович,
провідний науковий співробітник Інституту кібернетики
ім. В.М. Глушкова НАН України;

доктор технічних наук, професор
Дрозд Олександр Валентинович,
професор кафедри комп’ютерних
інтелектуальних систем і мереж
Одеського Національного політехнічного університету.

Захист відбудеться “ 21 ” червня 2013 р. о 10 год. 30 хв. на засіданні спеціалізованої вченої ради Д 35.052.08 у Національному університеті “Львівська політехніка” (79013, Львів-13, вул. С. Бандери, 12, ауд. 226 головного корпусу).

З дисертацією можна ознайомитися у бібліотеці Національного університету “Львівська політехніка” (79013, м. Львів, вул. Професорська,1).

Автореферат розісланий “ 21 ” травня 2013 р.

*Вчений секретар спеціалізованої вченої ради,
доктор технічних наук, професор*

Я. Т. Луцик

ЗАГАЛЬНА ХАРАКТЕРИСТИКА РОБОТИ

Актуальність теми.

Підвищення продуктивності комп'ютерних систем є визначальним для науки та для інженерних галузей, а також стало каталізатором появи низки нових сфер діяльності людини. Разом з тим, на початку 21-го століття почав сповільнюватися темп зростання продуктивності універсальних процесорів, що є базою персональних комп'ютерів та багатопроцесорних комп'ютерних систем. Причиною цього є фундаментальні обмеження в енергоефективності КМОН-технології, за якою сьогодні реалізують переважну більшість інтегральних схем, та обмеження, які накладає традиційний спосіб опрацювання інформації на архітектуру універсального процесора, зокрема, послідовний характер виконання команд програми та послідовний доступ до команд і даних в пам'яті.

Задача підвищення продуктивності комп'ютерних систем сьогодні в більшості випадків вирішується створенням багатоядерних процесорів та багатопроцесорних систем. Разом з тим, цей підхід також має принципові недоліки, головними з яких є низька реальна продуктивність таких систем, зумовлена невідповідністю їх структури структурі виконуваних алгоритмів, висока споживана потужність та низька ефективність використання обладнання. Для уникнення цих недоліків створюють комп'ютерні системи з спеціалізованими апаратними прискорювачами, однак вони є ефективними лише на вузьких класах алгоритмів. Тому одним з найперспективніших напрямів діяльності в сфері високопродуктивних обчислень сьогодні є створення реконфігурованих комп'ютерних систем (РККС), які позбавлені вказаних недоліків. РККС склали конкуренцію іншим типам високопродуктивних комп'ютерних систем завдяки високим технічним характеристикам сучасних кристалів програмованих логічних інтегральних схем (ПЛІС) – апаратній основі реконфігурованого середовища РККС, та досягненням у галузі технологій проектування апаратних засобів. Під'єднання синтезованих в реконфігурованому середовищі спеціалізованих процесорів зі структурою, яка в тій чи іншій мірі враховує особливості виконуваних алгоритмів, до комп'ютерних систем на основі універсальних процесорів, дозволяє на 2-3 порядки підняти їх продуктивність.

В провідних наукових установах світу, зокрема в Університеті м. Единбург (Великобританія), Університеті м. Вашингтон та Каліфорнійському Університеті Берклі (США), Таганрозькому технологічному інституті Південного федерального університету (Росія), а також України – в Національному університеті «Львівська політехніка», Інституті кібернетики НАН України, Національному технічному університеті України «Київський політехнічний інститут», створено наукові школи, які займаються проблематикою реконфігурованих обчислень. Значна кількість всесвітньо відомих компаній пропонують на ринок як РККС (*SRC Computers, Stone Ridge Technology* та ін.), так і апаратні та програмні засоби для їх побудови – ПЛІС та інтегровані середовища для проектування та синтезу в них спеціалізованих процесорів (*Xilinx, Altera, Microsemi* та ін.), реконфігуровні прискорювачі та програмні засоби для роботи з ними (*Nallatech, Clearspeed* та ін.).

Разом з тим, задачі проектування та синтезу спеціалізованих процесорів в реконфігурованому середовищі, зміни конфігурації реконфігурованого середовища, а перед цим розподілу обчислювального навантаження між універсальними та

спеціалізованими процесорами РККС, потребують залучення значних людських ресурсів, що суттєво знижує ефективність РККС. Постає задача пошуку шляхів підвищення їх ефективності. В роботі ця задача вирішується шляхом створення нового класу комп'ютерних засобів – самоконфігуровних високопродуктивних комп'ютерних систем. В самоконфігуровних комп'ютерних системах (СККС) виконання зазначених вище трудо- та часомістких процесів перекладено з користувача на комп'ютерні засоби, що дозволило використати всі потенційні можливості, які надаються властивістю зміни конфігурації реконфігуровного середовища, і забезпечило їм одне з чільних місць серед найперспективніших засобів високопродуктивних обчислень. Однак для реалізації цих можливостей є необхідним створення теорії побудови СККС. Для цього необхідно провести розроблення та дослідження невивчених перспективних моделей та методів обчислень, нових способів опрацювання інформації та нових комп'ютерних засобів, щоб перекласти трудомісткі та часомісткі процеси з користувача на ці засоби.

Такі дослідження мають привести до істотних зрушень у підвищенні ефективності комп'ютерних систем, від яких залежить прогрес у оборонній та космічній техніці, інформаційно-комунікаційних технологіях, вимірювальній техніці та метрології, електроніці, медицині та інших галузях, чим і зумовлена актуальність даної роботи.

Зв'язок роботи з науковими програмами, планами, темами. Тема дисертаційної роботи відповідає науковому напряму кафедри безпеки інформаційних технологій Національного університету “Львівська політехніка”. Дисертація виконана в межах науково-дослідних робіт “Створення програмних засобів високопродуктивних комп'ютерних систем на основі універсальних комп'ютерів”, номер державної реєстрації 0109U007349, виконаною в 2009-2010 роках, та “Розроблення теорії побудови багатопортової пам'яті комп'ютера на принципах паралельного доступу до даних”, номер державної реєстрації 0112U001213, виконаною в 2012-2013 роках в рамках пріоритетного тематичного напряму Національного університету “Львівська політехніка” “Нові інтелектуальні, комп'ютерні, радіоелектронні, інфокомунікаційні вимірювальні технології, системи, пристрої та бортові системи космічних апаратів”.

Мета і задачі дослідження. Метою дисертаційної роботи є підвищення ефективності комп'ютерних систем з реконфігуровною логікою шляхом побудови самоконфігуровних високопродуктивних комп'ютерних систем і їхніх компонентів, їх дослідження та реалізація.

Для досягнення мети необхідно було виконати наступні задачі:

1. Провести аналіз принципів побудови, технічних характеристик і технологій проектування високопродуктивних комп'ютерних систем з використанням пристроїв реконфігуровної логіки, визначити проблемні питання їх побудови та обґрунтувати актуальність вирішення цих питань.
2. Розробити основи організації функціонування РККС, виявити вузькі місця і запропонувати вдосконалені послідовності виконання обчислювальних процесів.
3. Удосконалити підходи до організації роботи апаратних та програмних засобів, які використовуються для виконання обчислювальних процесів в РККС,

звернувши особливу увагу на проблеми взаємодії універсального процесора з реконфігуровним середовищем.

4. Запропонувати вимоги та дослідити варіанти архітектури і методи побудови спеціалізованих процесорів, які реалізуються в реконфігуровному середовищі комп'ютерних систем.
5. Дослідити підходи до реалізації засобів створення програмних моделей спеціалізованих процесорів і вибрати з них ефективні для застосування в комп'ютерних системах з реконфігуровною логікою.
6. Вдосконалити спосіб опрацювання інформації в комп'ютерних системах з реконфігуровною логікою та розробити засоби для його реалізації.
7. Розробити та дослідити зразки базових компонентів СККС і показати їх ефективність шляхом проведення аналітичних оцінювань та експериментальних досліджень.

Об'єкт дослідження – процеси опрацювання інформації в самоконфігуровних високопродуктивних комп'ютерних системах.

Предмет дослідження – методи та засоби побудови та проектування самоконфігуровних високопродуктивних комп'ютерних систем.

Методи досліджень, які використані в роботі, базуються на теорії проектування комп'ютерних систем, теорії проектування надвеликих інтегральних схем, обчислювальній та дискретній математиці. У проведених дослідженнях використано та розвинуто теорію побудови реконфігуровних комп'ютерних систем та теорію високорівневого проектування програмних моделей спеціалізованих процесорів, а також використано апарат теорії алгоритмів, множин, матриць, графів, комбінаторику, цифрову схемотехніку, моделювання алгоритмів та апаратних засобів комп'ютера, експериментальні дослідження, програмування мовами опису апаратних засобів, програмування мовами високого рівня.

Наукова новизна роботи.

1. Отримала подальший розвиток теорія побудови РККС, а саме:
 - 1.1. Здійснено класифікацію та проведено дослідження базових структур і відповідних їм типів архітектури РККС. Розвинуто теорію їх проектування з виокремленням проблем, що перешкоджають підвищенню ефективності цих систем, і знайдено шляхи та розроблено методи їх вирішення.
 - 1.2. Вперше досліджено ступінь впливу технічних характеристик засобів взаємодії компонентів РККС на її продуктивність, що забезпечує можливість розрахунку тривалості етапів опрацювання інформації в системі та, відповідно до запропонованої методики, дозволяє вибрати ефективніший для виконання алгоритму заданої обчислювальної складності тип архітектури РККС.
2. Розроблено основи теорії побудови СККС, які полягають в наступному:
 - 2.1. Вперше запропоновано концепцію побудови СККС та метод самоконфігурування, який, на відміну від методу конфігурування РККС, передбачає автоматичне виконання розподілу програми між універсальним комп'ютером та реконфігуровним прискорювачем, створення файлу конфігурації реконфігуровного середовища та автоматичне створення в цьому середовищі спеціалізованого процесора, що дозволило скоротити

час та зменшити складність опрацювання інформації, а також зняти обмеження, які накладає спеціалізація процесора, і забезпечити ефективне використання реконфігуровної логіки для виконання довільних задач.

- 2.2. Вперше, на основі методу самоконфігурування, розроблено спосіб опрацювання інформації в СККС для варіантів їх одно- та багатопроцесорної реалізації, в якому всі дії, включаючи розподіл програми на підпрограми універсального комп'ютера та реконфігуровного середовища, створення на мові опису апаратних засобів програмної моделі спеціалізованого процесора для виконання підпрограми реконфігуровного середовища, виконання її логічного синтезу та завантаження до реконфігуровного середовища отриманих в результаті логічного синтезу файлів конфігурації, виконуються автоматично, і який є теоретичною основою проектування СККС.
- 2.3. Вперше розроблено принципи структурної організації та функціонування СККС для варіантів їх одно- та багатопроцесорної реалізації.
- 2.4. Вперше розроблено методику розрахунку характеристик тривалості процесів опрацювання інформації в СККС, методику оцінювання часу виконання в них програми і забезпечуваного прискорення, що дало можливість довести принципові переваги СККС над РККС і підтвердило ефективність запропонованої концепції, розроблених методу самоконфігурування та способу опрацювання інформації.
- 2.5. Вперше розроблено концептуальні основи побудови системи розподілу обчислювального навантаження, принципи її структурної організації та метод розподілу його між комп'ютером і реконфігуровним середовищем, що дало можливість здійснити проектування та реалізацію її програмних засобів.
- 2.6. Проведено дослідження та сформовано вимоги до засобів взаємодії між комп'ютером і реконфігуровним прискорювачем, а також розроблено базову модель і архітектуру програмно-апаратної системи взаємодії, що дало можливість визначити функції та здійснити проектування і реалізацію драйвера реконфігуровного прискорювача, бібліотеки рівня користувача та системних утиліт для роботи з реконфігуровним прискорювачем.
- 2.7. Запропоновано метод організації віддаленого використання реконфігуровних прискорювачів, який полягає у наданні реконфігуровного прискорювача як сервісу через комп'ютерну мережу, розроблено структуру, принципи організації функціонування та методику налаштування системи віддаленого доступу до реконфігуровних прискорювачів, а також схему та методи взаємодії користувачів з реконфігуровним прискорювачем в межах системи віддаленого доступу.
3. Розроблено основи теорії побудови паралельних спеціалізованих процесорів для виконання алгоритмів з інваріантною до даних структурою в реконфігуровному середовищі СККС, які полягають в наступному:
 - 3.1. Вперше розроблено спосіб опрацювання інформації з впорядкованими компонентами програми, який, на відміну від відомих способів опрацювання інформації, враховує в програмі просторові та часозалежні

характеристики виконуваного алгоритму та забезпечує паралельне виконання алгоритмів з інваріантною структурою.

- 3.2. Розроблено нову архітектуру паралельних спеціалізованих процесорів для виконання в реконфігурованому середовищі СККС алгоритмів з інваріантною до даних структурою, яка базується на способі опрацювання інформації з впорядкованими компонентами програми.
- 3.3. Вперше розроблено базові структури паралельних спеціалізованих процесорів для виконання алгоритмів з інваріантною до даних структурою, що дозволяє вибирати з них доцільну для реалізації за значенням продуктивності або затрат обладнання. Створено методикау розрахунку і отримано вирази для оцінювання тривалості виконання ними програми, досліджено їхні технічні характеристики. Розроблено теоретичні основи їх проектування, що дало можливість сформулювати теоретичну базу для побудови систем генерування програмних моделей спеціалізованих процесорів цього типу в СККС.
4. Узагальнено і удосконалено теорію високорівневого проектування програмних моделей спеціалізованих процесорів (ПМСП), а саме:
 - 4.1. Вперше запропоновано класифікацію методів проектування ПМСП для реконфігурованих прискорювачів та розроблено відповідні їм методи побудови в СККС засобів генерування ПМСП: 1) з використанням бібліотек ПМСП, 2) з використанням бібліотек компонентів ПМСП, 3) з використанням конфігурованих ПМСП, та 4) з описів алгоритмів мовою високого рівня. Розроблено концептуальні основи побудови системи генерування ПМСП.
 - 4.2. Проведено аналіз відповідності розроблених методів побудови засобів генерування ПМСП вимогам до їх застосування в СККС в частині функціональної повноти системи генерування та в частині технічних характеристик і архітектури генерованих нею ПМСП, а також розроблено структури відповідних цим методам засобів генерування, в результаті чого встановлено, що четвертий метод, зважаючи на його функціональну повноту, може бути застосований як основний для побудови системи генерування, а інші – як допоміжні у випадку використання СККС для виконання наперед визначеного класу алгоритмів.
 - 4.3. Сформульовано теоретичні основи організації функціонування засобів генерування ПМСП з описів алгоритмів мовою високого рівня та запропоновано таку їх класифікацію за способом генерування: 1) переведенням опису алгоритму в модель апаратно-програмної системи, 2) переведенням опису алгоритму в логічні вентиля ПЛІС, та 3) конфігуруванням базової конфігурованої ПМСП під заданий алгоритм.
 - 4.4. Вперше отримано вирази для визначення тривалості генерування ПМСП засобами генерування, з використанням яких проведено порівняльний аналіз та оцінювання доцільності застосування розроблених методів побудови цих засобів в СККС, що дало можливість розробити структуру і основи організації функціонування системи генерування ПМСП та розробити генератор на основі бібліотеки компонент ПМСП та

алгоритмічні основи функціонування програми генерування файлів верхнього рівня.

5. Створено експериментальні зразки базових компонентів СККС і отримано дані про їх характеристики, які підтверджують основні положення теорії побудови СККС.

Практичне значення одержаних результатів.

У дисертації отримано результати, які мають як теоретичне, так і прикладне значення.

Результати, що мають теоретичне значення:

1. Узагальнено і удосконалено теорію високорівневого проектування програмних моделей спеціалізованих процесорів.
2. Розроблено основи теорії побудови паралельних спеціалізованих процесорів для виконання алгоритмів з інваріантною до даних структурою.
3. Розроблено концепцію побудови, метод самоконфігурування та способи опрацювання інформації в СККС.

Результати, що мають прикладне значення:

Розроблено новий тип комп'ютерних засобів – самоконфігуровні комп'ютерні системи, які характеризуються малим часом і відносною простотою опрацювання інформації та відсутністю обмежень, які накладає спеціалізація процесора, а також ефективним використанням реконфігуровної логіки для виконання довільних задач, що забезпечує їм чільне місце серед найперспективніших засобів високопродуктивних обчислень.

Отримані в дисертації результати наукових досліджень знайшли використання в ряді науково-дослідних робіт та проектів, зокрема:

1. Розроблену архітектуру паралельних спеціалізованих процесорів для виконання алгоритмів з інваріантною до даних структурою використано для реалізації процесора швидкого перетворення Фур'є, створеного на науково-виробничому підприємстві «Інтрон», м. Львів, і може бути покладено в основу побудови високопродуктивних спеціалізованих процесорів для виконання інших алгоритмів. Зазначений процесор, реалізований в ПЛІС *XC4085XLABG560* фірми *Xilinx*, порівняно з існуючими аналогами характеризується майже в 2 рази нижчими затратами обладнання при підвищенні швидкодії в 1,5 рази.
2. Розроблені пристрої паралельної пам'яті з впорядкованим доступом використано при синтезі названого вище процесора швидкого перетворення Фур'є та в рамках науково-дослідного проекту з розроблення теорії побудови багатопортової пам'яті комп'ютера на принципах паралельного доступу до даних, виконаного у Національному університеті «Львівська політехніка». В рамках зазначеного проекту використано результати реалізації чотирьох типів паралельної пам'яті з впорядкованим доступом: з попереднім налаштуванням, з одночасним надходженням даних та індексів, з змінним впорядкованим доступом та з фіксованим впорядкованим доступом. В результаті реалізації розроблених типів пам'яті в ПЛІС *bvcsx75tff484-2* фірми *Xilinx* отримано значення тактової частоти від 802 до 1012 МГц, тобто ця пам'ять може працювати на максимальній частоті ПЛІС, що, з врахуванням надання нею

паралельного доступу до даних, забезпечує їй продуктивність, недосяжну для інших типів пам'яті.

3. Розроблені формалізована модель алгоритму та структура алгоритму використані при виконанні за підтримки Міністерства Науки та Вищої Освіти Республіки Польща на факультеті електротехніки, автоматики та інформатики Університету Технологій м. Кельце (*Kielce University of Technology*), Польща, науково-дослідної роботи з розвитку теорії алгоритмів і можуть бути застосовані для створення методологічних основ автоматичного проектування спеціалізованих процесорів для паралельного виконання алгоритмів з інваріантною до даних структурою.
4. Розроблені концептуальні основи побудови системи генерування програмних моделей спеціалізованих процесорів та програму генерування файлів верхнього рівня мовою *VHDL* використано при виконанні за підтримки Міністерства Науки та Вищої Освіти Республіки Польща на факультеті електротехніки, автоматики та інформатики Університету Технологій м. Кельце (*Kielce University of Technology*), Польща, науково-дослідної роботи з розвитку теорії побудови конфігурованих моделей обчислювальних пристроїв, і надалі були застосовані для створення генераторів програмних моделей спеціалізованих процесорів на основі бібліотеки їх компонент.
5. Розроблені засоби для організації взаємодії компонентів СККС та для проведення експериментальних досліджень і оцінювання характеристик РККС залежно від складності задачі використано при виконанні науково-дослідного проекту зі створення програмних засобів високопродуктивних комп'ютерних систем на основі універсальних комп'ютерів за договором з Міністерством освіти і науки України у Відкритому міжнародному університеті розвитку людини «Україна» для організації взаємодії та для виконання низки експериментальних досліджень на РККС з прискорювачем *XPRESSFX100-11* компанії *PLDA*. Розроблені засоби організації взаємодії компонентів СККС можуть бути використані для організації взаємодії реконфігурованих прискорювачів в довільних комп'ютерних системах, у тому числі й з віддаленим доступом.
6. Розроблені концептуальні основи побудови системи розподілу обчислювального навантаження використано в рамках науково-дослідного проекту, виконаного на науково-виробничому підприємстві «Інтрон» при реалізації програмних засобів системи розподілу обчислювального навантаження між універсальним комп'ютером і реконфігурованим прискорювачем, і надалі можуть бути використані для створення високопродуктивних комп'ютерних систем не тільки з реконфігурованими, але й з спеціалізованими апаратними прискорювачами.

Результати дисертаційної роботи в частині розроблення засобів організації взаємодії компонентів самоконфігурованої комп'ютерної системи також використано:

1. Під час виконання на фірмі *Duolog Technologies Ltd.*, м. Голуей (*Galway*), Ірландія, наукових робіт “Розробка та дослідження архітектури системи захисту інформації для мереж IEEE 802.15.4b” та “Апаратний модуль захисту

інформації за алгоритмом *AES-CCM** в мережах IEEE 802.15.4b” в рамках науково-дослідного проекту “Малопотужні бездротові мережі з малим радіусом дії” за підтримки Європейської Комісії через Програму Трансферу Знань Асоціації Марії Кюрі. Зокрема, впроваджені в рамках цих робіт наукові результати зі створення малопотужних мереж в подальшому були розвинуті до розроблення методу організації віддаленого доступу до реконфігурованих прискорювачів.

2. Під час виконання на науково-виробничому підприємстві «Інтрон» науково-технічного проекту «Система оперативно-технологічного зв'язку Укрзалізниці» в її частинах з розроблення плати з'єднання сервера з периферійними пристроями, пристрою арбітражу шини даних інтерфейсу фізичного рівня та контролера інтерфейсу фізичного рівня. Зокрема, впроваджені в рамках цих проектів наукові результати зі створення зазначених частин в подальшому були розвинуті до розроблення засобів організації взаємодії комп'ютера з реконфігурованим прискорювачем.

Методологічні й наукові результати дисертаційної роботи отримали використання в навчально-методичному забезпеченні лекційних курсів з дисциплін «Архітектура комп'ютерних систем» та «Комп'ютерні методи високорівневого проектування пристроїв захисту», які читаються автором спеціалістам і магістрам спеціальності «Безпека інформаційних і комунікаційних систем» на кафедрі безпеки інформаційних технологій Національного університету «Львівська політехніка».

Впровадження результатів роботи. Теоретичні та практичні результати роботи впроваджено при виконанні держбюджетної науково-дослідної роботи “Створення програмних засобів високопродуктивних комп'ютерних систем на основі універсальних комп'ютерів”, виконаної в 2009-2010 роках за договором з Міністерством освіти і науки України, державний реєстраційний номер 0109U007349, шифр роботи ДЗ/465-2009, при виконанні держбюджетної науково-дослідної роботи «Розроблення теорії побудови багатопортової пам'яті комп'ютера на принципах паралельного доступу до даних», виконаної в 2012-2013 роках в рамках пріоритетного тематичного напрямку Національного університету «Львівська політехніка» «Нові інтелектуальні, комп'ютерні, радіоелектронні, інфокомунікаційні вимірювальні технології, системи, пристрої та бортові системи космічних апаратів», державний реєстраційний номер 0112U001213, шифр роботи «ППК», у роботах, що проводились на науково-виробничому підприємстві “Інтрон” (м. Львів), а також у навчальному процесі кафедри безпеки інформаційних технологій Національного університету “Львівська політехніка” в рамках дисциплін “Архітектура комп'ютерних систем” та “Комп'ютерні методи високорівневого проектування пристроїв захисту”. Дані про впровадження підтверджені відповідними актами.

Особистий внесок здобувача полягає в теоретичному обґрунтуванні одержаних результатів, їх експериментальній перевірці та дослідженнях, а також у створенні програмних засобів для практичного використання одержаних результатів. Основний зміст роботи, всі теоретичні та практичні розробки, висновки та рекомендації, виконані автором особисто. З наукових праць, опублікованих у співавторстві, в дисертаційній роботі використовуються результати, які одержані

автором самостійно. В друкованих працях, опублікованих в співавторстві, здобувачу належать: [1] – частини 1, 2, 3, 5, 6; [2] – розроблення формалізованої моделі та структури алгоритму; [3] – формалізація опису компонентів архітектури спеціалізованого процесора; [4] – розроблення генератора програмних моделей комп'ютерних систем на кристалі; [5] – розроблення програмної моделі процесора ШПФ, його реалізація в реконфігуровному прискорювачі фірми PLDA на базі ПЛІС *Virtex4* фірми *Xilinx*, вибір платформ для виконання експериментальних досліджень, проведення цих досліджень та оцінювання отриманих результатів; [7] – формування вимог до інтерфейсу між комп'ютером і реконфігуровним прискорювачем, розроблення моделі взаємодії між універсальним комп'ютером і реконфігуровним прискорювачем та базової архітектури програмно-апаратної системи для її реалізації через інтерфейс PCI-Express; [8] – формулювання теоретичної бази та обґрунтування доцільності створення самоконфігуровних апаратних прискорювачів, розроблення концепції побудови самоконфігуровних апаратних прискорювачів та основ організації їх функціонування; [9] – розроблення технології генерування програмних моделей спеціалізованих процесорів на основі бібліотеки їх компонент та реалізація програми генерування файлів верхнього рівня мовою VHDL; [10] – визначення вихідної інформації для синтезу процесора ШПФ, розроблення його програмної моделі мовою VHDL, її синтез в ПЛІС та оцінювання отриманих результатів; [13] – вибір характеристик пам'яті з впорядкованим доступом з паралельним надходженням даних та індексів, розроблення її програмної моделі мовою VHDL, її синтез в ПЛІС та оцінювання отриманих результатів; [14] – розроблення технічних вимог до системи розподілу обчислювального навантаження і методики оцінювання ефективності типу фрагменту програмного коду, а також принципів функціонування системи розподілу обчислювального навантаження; [15] – розроблення методики оцінювання отриманого в результаті розподілу тестової програми прискорення; [20] – розроблення базових структур пристроїв пам'яті з впорядкованим доступом та спеціалізованих процесорів опрацювання зображень; [21] – визначення особливостей застосування автономних інтелектуальних агентів та багатоагентних систем в автоматичній і обчислювальній техніці; [22] – розроблення засобів генерування програмних моделей процесорів шифрування за алгоритмами *DES* і *Triple DES* та основ застосування цих засобів при проектуванні комп'ютерних систем на кристалі; [24] – проведення аналізу можливостей конфігурування моделей спеціалізованих процесорів механізмами мов опису апаратних засобів; [25] – розроблення технології організації бібліотек стандартизованих та замовних програмних моделей спеціалізованих процесорів для високопродуктивних реконфігуровних прискорювачів; [26] – формування та оцінювання підходів до проектування програмних моделей обчислювальних пристроїв з мови високого рівня у високопродуктивних реконфігуровних прискорювачах; [28] – формування та класифікація вимог до системи генерування спеціалізованих процесорів в контексті її використання в самоконфігуровному апаратному прискорювачі; [29] – розроблення методу самоконфігурування реконфігуровного апаратного прискорювача; [31] – вибір характеристик пам'яті з впорядкованим доступом з попереднім налаштуванням, розроблення її програмної моделі мовою VHDL, її синтез в ПЛІС та оцінювання отриманих результатів.

Апробація результатів дисертації. Викладені в дисертаційній роботі результати було апробовано і схвалено на 13 наукових конференціях різного рівня, а саме:

- Літній Школі Центрально-Європейської програми обміну в області університетської освіти (Central European Exchange Programme for University Studies (CEEPUS) Summer School) «*Intelligent Control Systems*», Брно, Чехія, 29 серпня – 11 вересня 2005 р.;
- V Міжнародній конференції студентів і молодих науковців „*Telecommunication in XXI Century*“, Вулка Мілановска, Польща, 24-26 листопада 2005 р.;
- III Міжнародній науково-технічній конференції «*Сучасні комп'ютерні системи та мережі: розробка та використання*» (ASCN-2007), м. Львів, 20-22 вересня 2007 р.;
- IV Всеукраїнській науково-практичній конференції «*Комп'ютерні технології: наука і освіта*», м. Луцьк, 9-11 жовтня 2009 р.;
- V Міжнародній науково-технічній конференції «*Dependable Systems, Services and Technologies*» (DESSERT'10), м. Кіровоград, 11-15 травня 2010 р.;
- II Міжнародній науково-практичній конференції «*Програмне забезпечення в освіті і науці*», м. Київ, 12-13 травня 2010 р.;
- V Міжнародній науково-технічній конференції «*Сучасні комп'ютерні системи та мережі: розробка та використання*» (ASCN-2011), м. Львів, 29 вересня – 01 жовтня 2011 р.;
- V Міжнародній науковій конференції «*Комп'ютерні науки та інженерія 2011*» (CSE-2011), м. Львів, 24-26 листопада 2011 р.;
- Міжнародному науково – практичному семінарі «*Програмовані логічні інтегральні схеми та мікропроцесорна техніка в освіті і виробництві*», м. Луцьк, 18-19 травня 2012 р.;
- I-й Міжнародній науково-технічній конференції “*Захист інформації і безпека інформаційних систем*”, м. Львів, 31 травня – 1 червня 2012 р.;
- XI міжнародній науково-технічній конференції “*Вимірювальна та обчислювальна техніка в технологічних процесах*” (ВОТПП_11_2012), м. Хмельницький, 5-8 червня 2012 р.;
- Проблемно-науковій міжгалузевій конференції “*Інформаційні проблеми комп'ютерних систем, юриспруденції, енергетики, економіки, моделювання та управління*” (ПНМК-2012), Бучач – Скоморохи, 7-9 червня 2012 р.;
- V Міжнародній науково-технічній конференції «*Комп'ютерні системи та мережні технології*» (CSNT-2012), м. Київ, 13-15 червня 2012 р.

Публікації. За результатами виконаних досліджень опубліковано 33 роботи, з них 1 монографія, 19 статей у фахових наукових журналах та збірниках, перелік яких затверджений ВАК України, 13 статей у матеріалах конференцій.

Структура та обсяг роботи. Дисертаційна робота складається з вступу, семи розділів, висновку, списку використаних джерел і додатку. Загальний обсяг роботи складає 329 сторінок. Робота містить 84 рисунки, 10 таблиць, додатки на 6 сторінках. Список використаних джерел містить 256 найменувань.

ОСНОВНИЙ ЗМІСТ РОБОТИ

У вступі наведено загальну характеристику роботи, обґрунтовано її актуальність, сформульовано її мету та основні задачі досліджень, визначено методи вирішення поставлених задач, сформульовано наукову новизну роботи та практичну цінність одержаних результатів. Наведено дані про впровадження результатів роботи, її апробацію та публікації.

У першому розділі «Аналіз принципів побудови, технологій проектування та напрямків розвитку високопродуктивних комп'ютерних систем» досліджено методи підвищення продуктивності комп'ютерних систем. Для цього проаналізовано архітектуру і оцінено технічні характеристики сучасних суперкомп'ютерів, включаючи персональні, та показано, що значна їх частина побудована з використанням спеціалізованих апаратних прискорювачів, причому саме завдяки цьому ними досягнуто високих показників продуктивності. Визначено проблемні питання побудови суперкомп'ютерів, що дозволило окреслити перспективні напрями діяльності в галузі високопродуктивних комп'ютерних систем.

Розглянуто питання прискорення універсальних комп'ютерних систем використанням паралельних апаратно-орієнтованих спеціалізованих процесорів (СП). На основі дослідження архітектури СП та аналізу відповідності різних типів пам'яті вимогам до пам'яті сучасного СП, показано переваги застосування в СП пам'яті з впорядкованим доступом. Проведено аналіз архітектури та технічних характеристик спеціалізованих апаратних прискорювачів (САП), запропоновано їхні узагальнені структури, та встановлено, що САП належать до багатопроцесорних комп'ютерних систем класів ОКМД та частково МКМД, в структурах яких наявна спільна або/і локальна пам'ять. Визначено загальні тенденції розвитку САП та встановлено, що подібні тенденції сьогодні характерні й для універсальних програмовних процесорів. Показано проблеми, що обмежують застосування САП.

Розглянуто передумови створення та приклади перших реалізацій, а також наведено структури РККС на основі програмовних логічних інтегральних схем. Проведено порівняльний аналіз основних технічних характеристик універсальних мікропроцесорів і програмовних логічних інтегральних схем та показано потенційну перевагу ПЛІС для побудови високопродуктивних комп'ютерних систем, що дозволило виявити тенденцію їх широкого впровадження в суперкомп'ютерах та перспективних персональних суперкомп'ютерах.

Подано огляд сучасних технологій і засобів проектування СП для РККС. Показано, що актуальною тенденцією розвитку цих технологій є підвищення рівня проектування від міжрегістрових передач до системного з метою спрощення та скорочення часу проектування, результатом чого є створення засобів автоматичного генерування програмних моделей процесорів з їх конфігурованих моделей та з опису алгоритму їх роботи мовою високого рівня.

На основі проведеного аналізу принципів побудови, технологій проектування та напрямків розвитку високопродуктивних комп'ютерних систем виокремлено такі способи підвищення їхньої продуктивності: 1) підвищення тактової частоти роботи; 2) паралелізм; 3) конвеєризація; 4) спеціалізація. Встановлено, що застосування ПЛІС в комп'ютерних системах дає можливість використати всі чотири зазначені способи підвищення їхньої продуктивності. Визначено проблемні питання

використання високопродуктивних комп'ютерних систем з реконфігуровною логікою, а саме: 1) необхідність визначення алгоритмів, що підлягають реалізації в реконфігуровному середовищі, та розроблення ПМСП, які реалізуються в ньому; 2) необхідність виконання розподілу обчислювальних завдань між універсальним комп'ютером і реконфігуровним середовищем в процесі написання програми; 3) неможливість завчасного розроблення для реалізації в реконфігуровному середовищі ПМСП, які були б ефективні на всіх класах задач. Сформульовано вирішувану в дисертації науково-технічну проблему та перелік завдань, які потрібно виконати для її вирішення.

У другому розділі “Розроблення основ організації функціонування та дослідження шляхів удосконалення архітектури реконфігуровних комп'ютерних систем” розвинуто теорію побудови РККС та здійснено пошук шляхів підвищення їх ефективності.

Запропоновано визначення основних понять РККС, що спрощує розуміння принципів їх функціонування. Проведено класифікацію базових структур РККС за ступенем зв'язаності універсального процесора з реконфігуровним середовищем та виділено відповідні їм типи архітектури РККС: 1) з слабозв'язаною реконфігуровною логікою; 2) з тіснозв'язаною реконфігуровною логікою, підключеною до шини пам'яті; 3) з тіснозв'язаною реконфігуровною логікою, підключеною до шини процесора; 4) з інтегрованою в АЛП універсального процесора реконфігуровною логікою; та 5) з інтегрованими в реконфігуровну логіку універсальним і спеціалізованим процесорами.

В РККС першого типу універсальний процесор взаємодіє із реконфігуровним середовищем через інтерфейс введення/виведення, тобто як із зовнішнім пристроєм. Швидкість обміну інформацією тут є достатньо низькою, тому така архітектура придатна для обчислювальних завдань, виконання яких не вимагає проведення швидкого та об'ємного обміну інформацією між універсальним і спеціалізованим процесорами. В РККС другого і третього типів реконфігуровне середовище під'єднане до універсального процесора через системну шину, що забезпечує високу швидкість взаємодії, причому в першому з них СП під'єднується до шини універсального процесора, а в другому – до шини пам'яті. РККС четвертого і п'ятого типів забезпечують найвищу швидкість взаємодії універсального процесора з спеціалізованим, причому в першому з них СП використовується як один з операційних пристроїв універсального процесора шляхом розширення його системи команд, а в другому універсальний процесор є вбудованим в реконфігуровне середовище комп'ютерної системи з реконфігуровною логікою і виконує функції керування СП, будучи реалізованим «жорстко» під час виготовлення кристалу ПЛІС, або як програмна модель. Останній підхід, по суті, є реалізацією «комп'ютерної системи на програмному кристалі» (англ. *SoPC – System on a Programmable Chip*). Принциповим його недоліком є обмеження ресурсів РККС ємністю кристалу. Для нарощення ресурсів та підвищення продуктивності, а також надання РККС властивості масштабованості, до універсального процесора підключають зовнішні пристрої реконфігуровної логіки, що, власне, і є реалізацією тіснозв'язаної та слабозв'язаної типів архітектури. Це доводить доцільність застосування реконфігуровних прискорювачів для

побудови РККС та виконання подальших досліджень на тіснозв'язаній та слабозв'язаній типах архітектури.

З метою виявлення вузьких місць в організації функціонування РККС, які є причиною недостатньої їх ефективності на певних класах задач, і здійснення пошуку шляхів їх вдосконалення, сформульовано спосіб опрацювання інформації в РККС та подано перелік і описано порядок застосування необхідних для його реалізації комп'ютерних засобів. Відповідно до цього способу, процес опрацювання інформації в РККС подається як послідовне виконання чотирьох етапів. На першому етапі користувач створює програму P_{in} на мові програмування високого рівня, розподіляє цю програму на підпрограму P_{UPP} універсального комп'ютера та підпрограму P_{RE} реконфігуровного середовища, виконує компіляцію підпрограми P_{UPP} універсального комп'ютера, формує її виконавчий файл obj та зберігає його в пам'яті універсального комп'ютера. На другому етапі користувач для виконання підпрограми P_{RE} реконфігуровного середовища розробляє (або використовує готове рішення) на мові опису апаратних засобів програмну модель спеціалізованого процесора $SPPM$, виконує її логічний синтез та завантажує до реконфігуровного середовища отримані в результаті логічного синтезу файли конфігурації $\mathbf{conf} = \{conf_q, q = \overline{1 \dots K_{FPGA}}\}$, K_{FPGA} – кількість кристалів ПЛІС, які формують реконфігуровне середовище, і, таким чином, створює в цьому середовищі СП. На третьому етапі за командою користувача операційна система завантажує виконавчий файл obj підпрограми універсального комп'ютера до його основної пам'яті за допомогою стандартного завантажувача. На четвертому етапі РККС виконує програму P_{in} , причому в універсальному комп'ютері та в реконфігуровному середовищі виконуються відповідні підпрограми – P_{UPP} та P_{RE} . В разі повторного виконанні цієї ж програми користувач послідовно виконує третій і четвертий етапи, а в разі наявності виконавчого файлу та файлів конфігурації для виконання програми – завантажує файли конфігурації до реконфігуровного середовища, і далі послідовно виконує третій і четвертий етапи.

На основі цього способу розроблено методику та отримано вирази для розрахунку тривалості опрацювання інформації в РККС. Загальну тривалість опрацювання інформації в РККС можна подати виразом:

$$T_{RCCS}^{DP} = {}_U t_{distr}^P + t_{compile}^{UPP} + t_{store}^{obj} + {}_U t_{develop}^{SPPM} + {}_U t_{synth}^{SPPM} + {}_U t_{load}^{conf} + t_{load}^{obj} + t_{exe}^P, \quad (1)$$

де: ${}_U t_{distr}^P$ – тривалість виконання розподілу програми на підпрограму P_{UPP} універсального комп'ютера та підпрограму P_{RE} реконфігуровного середовища; $t_{compile}^{UPP}$ – тривалість виконання компіляції підпрограми універсального комп'ютера; t_{store}^{obj} – тривалість зберігання виконавчого файлу підпрограми універсального комп'ютера в пам'яті універсального комп'ютера; ${}_U t_{develop}^{SPPM}$ – тривалість розроблення ПМСП; ${}_U t_{synth}^{SPPM}$ – тривалість логічного синтезу ПМСП; ${}_U t_{load}^{conf}$ – тривалість завантаження до реконфігуровного середовища отриманих в результаті логічного

синтезу файлів конфігурації; t_{load}^{obj} – тривалість завантаження до основної пам'яті виконавчого файлу підпрограми універсального комп'ютера; t_{exe}^P – тривалість виконання програми P_{in} в РККС. Індексом U^* позначено дії, які безпосередньо виконуються користувачем.

При повторному виконанні цієї ж програми користувач послідовно виконує третій і четвертий етапи, тому тривалість $T_{RCCS}^{DP'}$ опрацювання інформації в РККС в цьому випадку можна подати виразом:

$$T_{RCCS}^{DP'} = t_{load}^{obj} + t_{exe}^P. \quad (2)$$

За наявності виконавчого файлу та файлів конфігурації для виконання програми користувач завантажує файли конфігурації до реконфігурованого середовища, і далі послідовно виконує третій і четвертий етапи, тому тривалість $T_{RCCS}^{DP''}$ опрацювання інформації в РККС в цьому випадку можна подати виразом:

$$T_{RCCS}^{DP''} = {}_U t_{load}^{conf} + t_{load}^{obj} + t_{exe}^P. \quad (3)$$

Тривалість t_{exe}^P виконання програми відповідно до способу опрацювання інформації в РККС можна подати виразом:

$$t_{exe}^P = T_{UPP}(P_{UPP}) + T_{SP}(P_{RE}) - T_{UPP,SP}^{PAR} + ttr_{data}, \quad (4)$$

де: $T_{UPP}(P_{UPP})$ – тривалість виконання підпрограми P_{UPP} універсальним процесором; $T_{SP}(P_{RE})$ – тривалість виконання СП заданих підпрограмою P_{RE} обчислень; ttr_{data} – тривалість пересилання даних між спеціалізованим та універсальним процесорами чи пам'яттю під час виконання програми; $T_{UPP,SP}^{PAR}$ – час, на протязі якого універсальний та спеціалізований процесори виконують обчислення паралельно.

З виразу (4) випливає, що суттєвий вплив на значення тривалості t_{exe}^P виконання програми в РККС має організація зв'язку між комп'ютером і реконфігурованим прискорювачем. Тому, з метою визначення показників тривалості опрацювання інформації в РККС за запропонованою методикою, проведено оцінювання характеристик засобів взаємодії компонентів РККС слабозв'язаної та тіснозв'язаної типів архітектури. При цьому сформульовано основи організації функціонування РККС з слабозв'язаною реконфігуровною логікою та проведено оцінювання технічних характеристик існуючих реконфігурованих прискорювачів цих РККС з інтерфейсами *PCI*, *PCI-X* та *PCI-Express*. Також сформульовано основи організації функціонування РККС з тіснозв'язаною реконфігуровною логікою та проведено оцінювання технічних характеристик існуючих реконфігурованих прискорювачів цих РККС, а саме: тісноінтегрованих в мережну структуру прискорювачів, тісноінтегрованих прискорювачів з прямим під'єднанням до мережного інтерфейсного кристалу, прискорювачів з прямим з'єднанням з пристроями пам'яті та гібридних прискорювачів.

З метою визначення областей доцільного застосування реконфігуровних прискорювачів слабозв'язаної та тіснозв'язаної типів архітектури з позиції організації зв'язку між ними і комп'ютером проведено низку експериментальних досліджень характеристик РККС залежно від складності задачі на декількох обчислювальних платформах з використанням прискорювача *XPRESSFX100-11* від компанії *PLDA*. Тестування виконувалось на алгоритмах перетворення $RGB \rightarrow YUV$, цифрової фільтрації, ДКП та ШПФ на 1024 точки. В результаті встановлено, що при виконанні тестових алгоритмів з обчислювальною складністю $O((N/2) \cdot \log N)$ на базі слабозв'язаної архітектури досягається прискорення на один порядок, а на базі тіснозв'язаної – на два порядки. У випадку виконання обчислювально складніших алгоритмів показники прискорення вищі. Разом з тим дослідження показали, що алгоритми, які характеризуються невисокою обчислювальною складністю та інтенсивним потоком даних, типу перетворення $RGB \rightarrow YUV$, на базі слабозв'язаної архітектури виконувати недоцільно, оскільки коефіцієнт прискорення буде меншим одиниці. Тому в роботі було розроблено методику визначення доцільності застосування прискорювача з позиції організації зв'язку між ним і комп'ютером, за якою перед вибором алгоритмів, які мають бути реалізовані в прискорювачі, можна оцінити ефективність такого рішення.

На основі аналізу способу опрацювання інформації в РККС виокремлено основні проблеми, що перешкоджають підвищенню їх ефективності, а саме:

1. для виконання кожної нової програми в РККС потрібно послідовно виконати всі чотири етапи, в перших двох з яких всі дії здійснюються користувачем, що вимагає значних часових затрат;
2. в разі наявності виконавчого файлу та файлів конфігурації користувач для виконання програми повинен завантажити файли конфігурації до реконфігуровного середовища перед третім і четвертим етапами, що також вимагає значних часових затрат;
3. клас завдань, на яких РККС є ефективною, є вузьким, і залежить від функціональних характеристик реалізованих в реконфігуровному середовищі СП;
4. опрацювання інформації характеризується високою складністю, оскільки користувач, крім моделювання і програмування, повинен виконувати системний аналіз, розробляти архітектуру СП, здійснювати їх логічний синтез та реалізацію в ПЛІС.

Ключовим підходом до вирішення вказаних вище проблем є автоматизація процесів, що відбуваються в РККС під час опрацювання інформації. З урахуванням того, що сьогодні існують програмні засоби, які дозволяють автоматично створювати ПМСП з опису алгоритму їх роботи мовою програмування високого рівня, в роботі запропоновано пов'язати операції генерування ПМСП, її логічного синтезу та конфігурування реконфігуровного середовища в одну автоматично виконувану послідовність, таким чином завантажуючи коди конфігурації до реконфігуровного середовища автоматично без участі користувача. Необхідно зазначити, що використання засобів генерування висуває умову наявності опису алгоритму, який повинен виконуватись СП, мовою високого рівня. В роботі запропоновано отримати цей опис способом автоматичного розподілу програми на

підпрограму універсального комп'ютера та підпрограму реконфігуровного середовища, що, крім скорочення часових затрат, дасть можливість:

Для усунення обмеження на клас завдань, на яких РККС є ефективною, в роботі пропонується вдосконалити спосіб опрацювання інформації в РККС таким чином, щоб завантаження до реконфігуровного середовища отриманих в результаті логічного синтезу файлів конфігурації здійснювалось не користувачем, а операційною системою, причому не на другому етапі, а на третьому, паралельно з завантаженням виконавчого файлу підпрограми універсального комп'ютера до його основної пам'яті за командою користувача. Для цього файли конфігурації потрібно зберігати в пам'яті універсального комп'ютера разом з виконавчим файлом, і виконувати їх завантаження до реконфігуровного середовища після подання користувачем команди ініціалізації виконання програми. При цьому, оскільки файли конфігурації формуються автоматично і паралельно з виконавчим файлом підпрограми універсального комп'ютера і зберігаються в його пам'яті, всю послідовність дій від моменту початку розподілу програми до моменту отримання виконавчого файлу і файлів конфігурації варто трактувати як єдиний етап компіляції програми.

Запропоновані вдосконалення способу опрацювання інформації в РККС відображено на діаграмі, наведеній на рис. 1.

В результаті вдосконалення способу опрацювання інформації в РККС, та за умови автоматизації перших двох його етапів, вирішуються всі зазначені вище проблеми, оскільки: 1) всі дії, починаючи від розподілу програми до отримання виконавчого файлу та файлів конфігурації, виконуються автоматично на етапі компіляції без участі користувача; 2) в разі наявності виконавчого файлу та файлів конфігурації завантаження цих файлів відповідно до основної пам'яті універсального комп'ютера та до реконфігуровного середовища виконується за командою користувача операційною системою; цим самим вирішується питання розширення класу завдань, на яких РККС є ефективною.

У третьому розділі “Розроблення теоретичних основ побудови та організації функціонування самоконфігуровних комп'ютерних систем” розроблено концепцію побудови нового типу високопродуктивних комп'ютерних засобів – самоконфігуровних комп'ютерних систем.

СККС – це комп'ютерна система з реконфігуровною логікою, в якій компіляція програми включає автоматично виконувани дії з формування конфігурації, і яка набуває цієї конфігурації автоматично під час завантаження програми до виконання.

Термін «самоконфігуровна» до комп'ютерної системи вжито вперше, і говорить він про те, що всі дії з опрацювання інформації в ній, починаючи від розподілу програми між універсальним комп'ютером і реконфігуровним середовищем до створення файлу конфігурації, а також конфігурування, виконуються комп'ютерною системою самостійно за командою користувача.

Запропоновано метод самоконфігурування комп'ютерної системи з реконфігуровною логікою та, на його основі, спосіб опрацювання інформації в СККС. Відповідно до розробленого способу, користувач створює програму P_{in} на мові програмування високого рівня та подає її до СККС. Процес опрацювання інформації в СККС можна подати як послідовне виконання трьох етапів: компіляції

програми, її завантаження та виконання. На етапі компіляції СККС автоматично виконує наступні дії: розподіляє цю програму на підпрограму P_{UPP} універсального комп'ютера та підпрограму P_{RE} реконфігуровного середовища, виконує компіляцію підпрограми P_{UPP} універсального комп'ютера, формує її виконавчий файл obj , створює на мові опису апаратних засобів програмну модель СП $SPPM$ для виконання підпрограми P_{RE} реконфігуровного середовища, виконує її логічний синтез та зберігає в пам'яті універсального комп'ютера отримані виконавчий файл obj та файли конфігурації реконфігуровного середовища $\mathbf{conf} = \{conf_q, q = \overline{1 \dots K_{FPGA}}\}$, де K_{FPGA} – кількість кристалів ПЛІС, які формують реконфігуровне середовище.

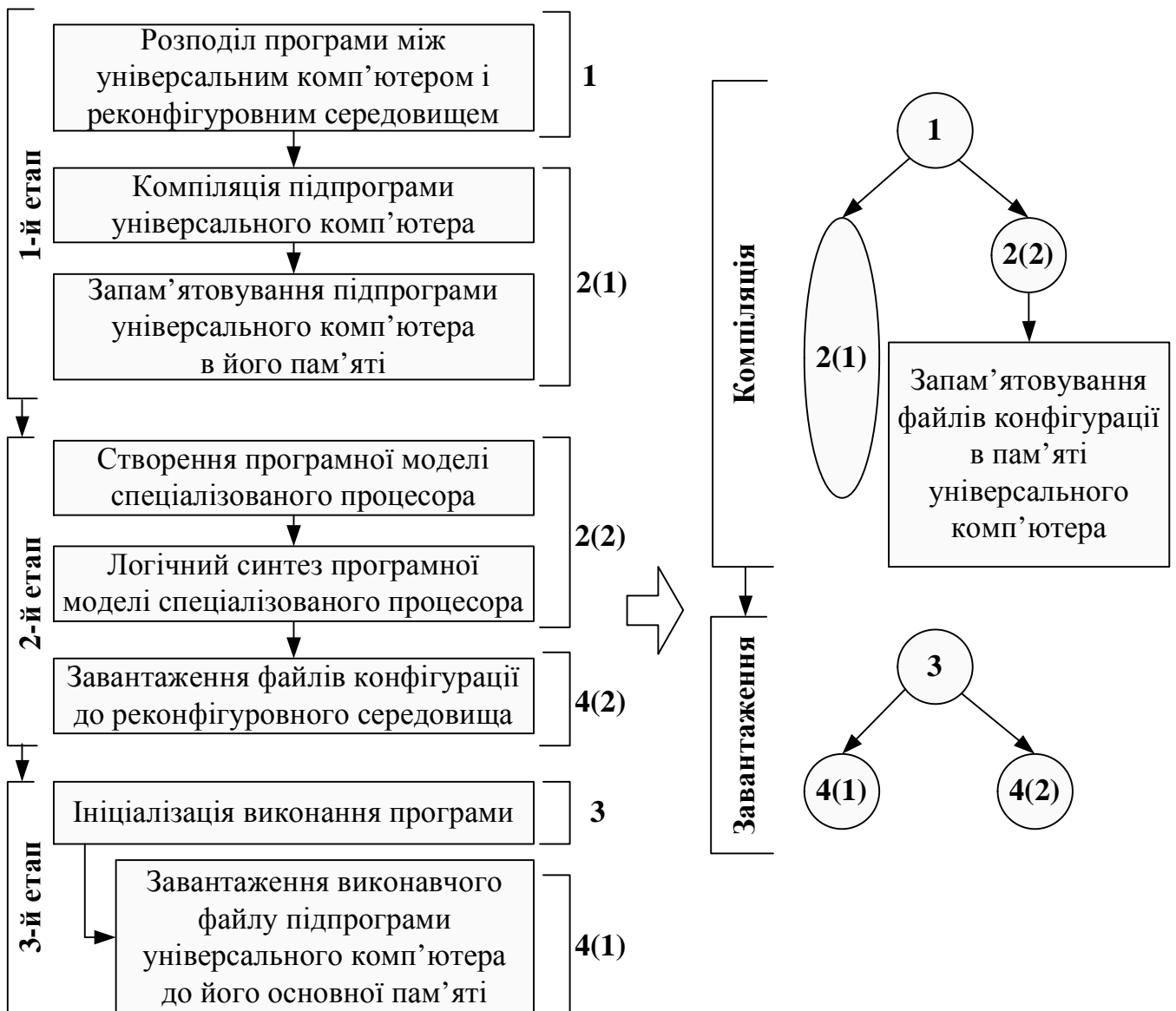


Рис. 1. Діаграма вдосконалення способу опрацювання інформації в РККС

Для виконання вказаних дій до складу СККС введено такі програмні засоби:

1. Для розподілу обчислювального навантаження між універсальним комп'ютером та реконфігуровним середовищем – систему розподілу

обчислювального навантаження, яка автоматично знаходить в програмі P_{in} такі фрагменти, виконання яких в реконфігурованому середовищі прискорює роботу комп'ютерної системи, і розподіляє програму P_{in} на підпрограму P_{UPP} універсального комп'ютера, заміняючи в ній виокремлені фрагменти на команди роботи з реконфігурованим середовищем, та сформовану із цих фрагментів підпрограму P_{RE} реконфігурованого середовища.

2. Для створення ПМСП – систему генерування, яка автоматично генерує програмну модель $SPPM$ спеціалізованого процесора з підпрограми P_{RE} .
3. Для компіляції підпрограми універсального комп'ютера і формування її виконавчого файлу, та для логічного синтезу ПМСП – засоби, що використовуються для виконання цих дій в РККС.

На етапі завантаження, після отримання від користувача команди ініціалізації виконання програми, завантажується виконавчий файл obj підпрограми універсального комп'ютера до його основної пам'яті, і паралельно завантажуються до реконфігурованого середовища файли конфігурації $\mathbf{conf} = \{conf_q, q = \overline{1..K_{FPGA}}\}$ і, таким чином, створює в цьому середовищі СП. Далі відбувається етап виконання програми, який є таким самим, як і в РККС. Для виконання вказаних дій в СККС використано ті ж засоби, що використовуються для їх виконання в РККС.

Розроблено структуру СККС (рис. 2), яка реалізує запропонований спосіб опрацювання інформації. З погляду користувача, СККС функціонує аналогічно до традиційного універсального комп'ютера, оскільки він, згідно з запропонованим способом опрацювання інформації, а) розробляє програму на мові високого рівня та подає її до СККС для виконання компіляції; б) ініціює виконання програми після її компіляції; в) завантажує дані до опрацювання та отримує результати.

На основі поданого вище способу опрацювання інформації в СККС запропоновано методику розрахунку тривалості опрацювання у ній інформації. Шляхом аналізу дій, виконуваних на кожному з етапів відповідно до способу, визначено їх тривалість. Відповідно, тривалість першого етапу опрацювання інформації в СККС можна визначити з виразу:

$$T_{SCCS}^{COMPILE} = t_{distr}^P + \max\left(t_{compile}^{UPP} + t_{store}^{obj}, \left(t_{generate}^{SPPM} + t_{synth}^{SPPM} + t_{store}^{conf}\right)\right), \quad (5)$$

де: t_{store}^{conf} – тривалість зберігання файлів конфігурації реконфігурованого середовища у пам'яті універсального комп'ютера; інші складові – це значення тривалості аналогічних дій, присутніх у виразі (1), виконувани автоматично.

Тривалість другого етапу опрацювання інформації в СККС можна визначити з виразу:

$$T_{SCCS}^{LOAD} = \max\left(t_{load}^{conf}, t_{load}^{obj}\right), \quad (6)$$

де t_{load}^{conf} – тривалість завантаження до реконфігурованого середовища отриманих в результаті логічного синтезу файлів конфігурації; t_{load}^{obj} – тривалість завантаження до основної пам'яті виконавчого файлу підпрограми універсального комп'ютера.

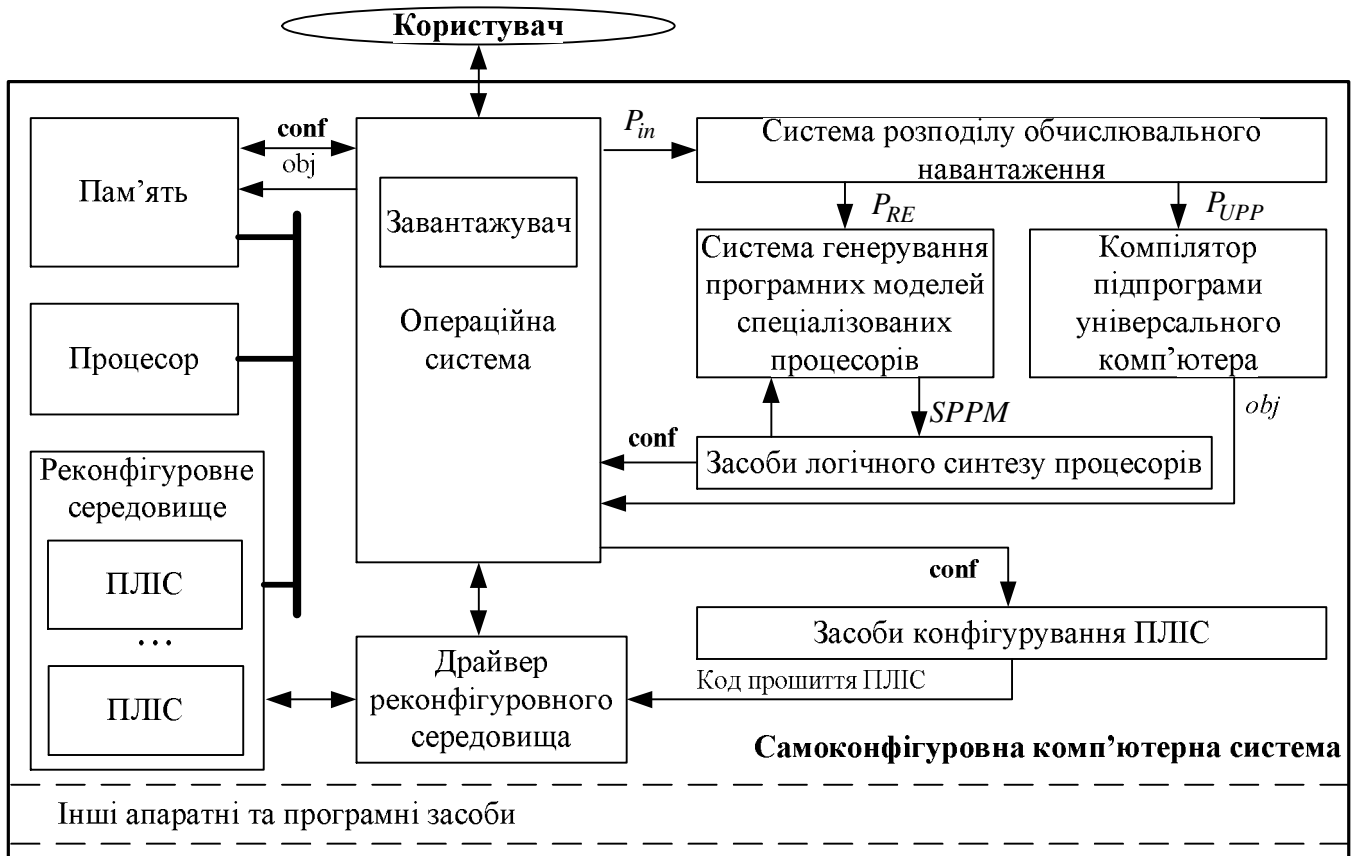


Рис. 2. Структура самоконфігуровної комп'ютерної системи

Тривалість опрацювання інформації відповідно до заданої програми в СККС можна подати виразом:

$$T_{SCCS}^{DP} = t_{distr}^P + \max\left(\left(t_{compile}^{UPP} + t_{store}^{obj}\right), \left(t_{generate}^{SPPM} + t_{synth}^{SPPM} + t_{store}^{conf}\right)\right) + \max\left(t_{load}^{conf}, t_{load}^{obj}\right) + t_{exe}^P. \quad (7)$$

В разі повторного виконання цієї ж програми або наперед сформованих виконавчому файлі підпрограми універсального комп'ютера та файлів конфігурації реконфігуровного середовища, тривалість опрацювання інформації в СККС буде однаковою і може бути подана виразом:

$$T_{SCCS}^{DP'} = \max\left(t_{load}^{conf}, t_{load}^{obj}\right) + T_{SCCS}^{EXE}(P_{in}). \quad (8)$$

Зауважимо, що можливе збільшення тривалості опрацювання інформації в СККС порівняно з РККС при повторному виконанні цієї ж програми за умови, що $t_{load}^{conf} > t_{load}^{obj}$. Тому в роботі було приділено увагу детальному дослідженню тривалості цих двох процесів. Для цього визначено основні процеси на етапах завантаження програми до виконання в СККС і її виконання і проведено аналітичні та низку експериментальних досліджень їх тривалості. При цьому проаналізовано характеристики сучасних комп'ютерних засобів, запропоновано вирази для оцінки та зроблено розрахунки тривалості завантаження виконавчого файлу підпрограми універсального комп'ютера до його основної пам'яті, а кодів конфігурації до реконфігуровного середовища. Доведено надлишковість застосування в СККС

під'єднаної до ПЛІС мікросхеми пам'яті для зберігання кодів конфігурації та показано шляхи подальшого зменшення тривалості завантаження файлів конфігурації до ПЛІС.

Встановлено, що тривалість завантаження виконавчих файлів об'ємом від кількох сотень КБ до кількох десятків МБ (переважна більшість виконавчих файлів програм мають такий об'єм) знаходиться в діапазоні від кількох одиниць до кількох сотень мілісекунд, причому значну частину цього часу займає виконання системних завдань, пов'язаних з їх завантаженням. Для сучасних кристалів ПЛІС провідних виробників значення тривалості завантаження кодів конфігурації знаходиться в діапазоні від кількох одиниць до кількох сотень мілісекунд. За результатами досліджень зроблено висновок, що значення тривалості завантаження виконавчих файлів та значення тривалості завантаження файлів конфігурації мають один порядок, тобто $t_{load}^{conf} \approx t_{load}^{obj}$, і різниця цих значень принципово не впливає на продуктивність СККС. Це підтверджує ефективність покладених в основу методу самоконфігурування та способу опрацювання інформації в СККС принципів, оскільки здійснення конфігурування ПЛІС СККС не уповільнює етапу завантаження програми порівняно з тривалістю цього етапу в РККС.

Визначено умови скорочення тривалості етапу виконання програми в СККС:

$$\left\{ \begin{array}{l} \frac{T_{UPP}(P_{UPP}) + T_{SP}(P_{RE})}{T_{UPP}(P_{in})} \rightarrow \min; \\ \min(T_{UPP}(P_{UPP}), T_{SP}(P_{RE})) - T_{UPP, SP}^{PAR} \rightarrow \min; \\ ttr_{data} \rightarrow \min. \end{array} \right. \quad (9)$$

де $T_{UPP}(P_{in})$ – тривалість виконання програми P_{in} універсальним процесором.

Виконання цих умов вирішується в СККС під час розподілу обчислювального навантаження між універсальним комп'ютером та реконфігуровним середовищем, та під час створення ПМСП, що було враховано при розробленні концептуальних основ побудови системи розподілу обчислювального навантаження та системи генерування.

Запропоновано модифікацію розробленого способу опрацювання інформації та структури самоконфігуровної комп'ютерної системи для випадку її багато процесорної реалізації та розроблено методику розрахунку характеристик тривалості процесів опрацювання в ній інформації. При цьому прийнято, що СККС містить K_{UPP} універсальних процесорів та K_{FPGA} кристалів ПЛІС, а програма P_{in} після виконання розпаралелення розпаралелюючим компілятором представлена кількістю k паралельних підпрограм P_{in_i} , $i = \overline{1..k}$, причому $k \leq K_{UPP}$.

Розроблено концептуальні основи побудови складових частин СККС – системи розподілу обчислювального навантаження та системи генерування програмних моделей спеціалізованих процесорів. При цьому сформовано вимоги, яким повинна відповідати система розподілу обчислювального навантаження, щоб забезпечити

виконання поданої виразом (9) системи умов. В результаті досліджень встановлено, що виконання цих умов залежить від ефективності вибору типу фрагменту програмного коду, над яким виконуються оптимізаційні процедури під час розподілу обчислювального навантаження. Якщо позначити множину типів фрагментів програмного коду як \mathbf{FC} , причому $\mathbf{FC} = \{FC_i, FC_j; i = \overline{1..n}; j = \overline{1..n}\}$, де FC_i, FC_j – i -тий та j -тий типи фрагменту програмного коду, серед яких виконується вибір, n – кількість типів фрагментів програмного коду, то вибір типу фрагменту програмного коду FC_i при розподілі обчислювального навантаження є ефективним, якщо для $FC_i \in \{FC\}, FC_j \in \{FC\}$ виконуються співвідношення:

$$\begin{cases} \frac{\sum_n T_{DT}(FC_i)_k}{\sum_n T_{FCE}(FC_i)_k} < \frac{\sum_m T_{DT}(FC_j)_l}{\sum_m T_{FCE}(FC_j)_l}, \\ Q_{RE}(FC_i) = Q_{RE}(FC_j) \end{cases} \quad (10)$$

де: $T_{DT}(FC_i)_k$ – час обміну даними з k -м, $k = \overline{1..n}$, фрагментом програмного коду при i -му типі фрагменту програмного коду; $T_{FCE}(FC_i)_k$ – час виконання k -го фрагменту програмного коду при i -му типі фрагменту програмного коду; $T_{DT}(FC_j)_l$ – час обміну даними з l -м, $l = \overline{1..m}$, фрагментом програмного коду при j -му типі фрагменту програмного коду; $T_{FCE}(FC_j)_l$ – час виконання l -го фрагменту програмного коду при j -му типі фрагменту програмного коду; n – кількість фрагментів програмного коду в об'ємі обчислень, що припадають на підпрограму реконфігурованого середовища, при i -му типі фрагменту програмного коду; m – кількість фрагментів програмного коду в об'ємі обчислень, що припадають на підпрограму реконфігурованого середовища при j -му типі фрагменту програмного коду; Q_{RE} – частка виконуваних реконфігурованим середовищем об'єму обчислень в програмі P_{in} , яка визначається з виразу:

$$Q_{RE} = \frac{V_{RE}}{V_{UPP} + V_{RE}} \cdot 100\%, \quad (11)$$

де V_{RE} – об'єм обчислень, що припадають на реконфігуроване середовище; V_{UPP} – об'єм обчислень, що припадають на універсальний процесор.

Сформовано три групи вимог до системи генерування ПМСП, а саме: 1) в частині її функціональної повноти; 2) в частині технічних характеристик генерованих ПМСП; 3) в частині архітектури генерованих ПМСП. Проведено аналіз та визначено умови забезпечення виконання цих вимог.

Сформовано вимоги до інших складових частин СККС – засобів логічного синтезу СП та конфігурування ПЛІС, реконфігурованого середовища та його

драйвера. Розроблені концептуальні основи побудови та вимоги покладено в основу проектування та реалізації складових частин СККС.

У четвертому розділі “Розроблення архітектури та основ проектування паралельних спеціалізованих процесорів для виконання в реконфігурованому середовищі алгоритмів з інваріантною до даних структурою” виходячи з потреби забезпечення високої продуктивності СП як одного з центральних елементів СККС та за результатами проведеного в Розділі 1 аналізу основних архітектурних підходів, використовуваних для побудови СП, обґрунтовано необхідність розроблення для СККС нової архітектури процесорів з паралельним опрацюванням даних, яка орієнтована на виконання визначених системою розподілу обчислювального навантаження частин алгоритмів з інваріантною до даних структурою і може бути налаштована під виділену кількість ресурсів реконфігурованого середовища СККС.

З метою максимального наближення структури СП до структури виконуваного ним алгоритму проведено аналіз алгоритму та запропоновано формалізований опис його параметрів. З метою виявлення зв'язків між параметрами алгоритму побудовано модель його структури.

Розроблено спосіб опрацювання інформації в СП для виконання алгоритмів з інваріантною до даних структурою. Враховуючи потребу забезпечення орієнтації архітектури процесора на опрацювання масивів даних, в ньому використано пам'ять з впорядкованим доступом (ПВД), доступ до даних в якій здійснюється у порядку, що задається привласненими їм індексами, які вказують їхнє місце у вихідному масиві. Згідно з цим способом, комп'ютерна програма подається як набір компонент – даних, команд, та індексів. Компоненти комп'ютерної програми записуються до комірок блоків ПВД як матриці, сформовані з дотриманням порядку виконання алгоритму. Обчислення здійснюються шляхом зчитування з блоків ПВД рядків даних та команд і їх подання до операційних пристроїв, при цьому за один такт роботи процесора здійснюється паралельне зчитування рядка даних, впорядкованих відповідно до їхніх індексів, і рядка команд, їх подання до операційних пристроїв, паралельне виконання операційними пристроями операцій над даними відповідно до вказівок цих команд, та запис результатів до ПВД. Отримані під час виконання обчислень проміжні дані записуються до ПВД в зарезервовані для них місця у вихідній матриці, причому місце запису кожного проміжного даного вказується його індексом. Рядок даних зчитують з пам'яті за наявності всіх потрібних даних у ньому, а отримані результати записують до пам'яті, привласнюючи їм відповідні індекси. Кількість тактів виконання в процесорі алгоритму за наведеним способом рівна кількості кроків цього алгоритму. Розроблений спосіб покладено в основу організації функціонування СП для виконання алгоритмів з інваріантною до даних структурою.

Проведено розроблення та дослідження принципів структурної організації та функціонування СП для виконання алгоритмів з інваріантною до даних структурою згідно з розробленим способом. Розроблено три типи структур СП з двома блоками ПВД даних: ітераційну, з шинною організацією з розподіленими входами і виходами ПВД даних, та з шинною організацією з об'єднаними входами і виходами ПВД даних.

На рис. 3 показано ітераційну структуру СП для виконання алгоритмів з інваріантною до даних структурою з двома блоками ПВД даних. Використання двох блоків ПВД даних зумовлено тим, що необхідно зберігати рядки отриманих з виходу АЛП даних та виконувати їх впорядкування в той час, коли ПВД, з якої зчитуються дані на вхід АЛП, вміщує ще не зчитані рядки даних, а це не дозволяє використовувати цей блок ПВД для записування даних.

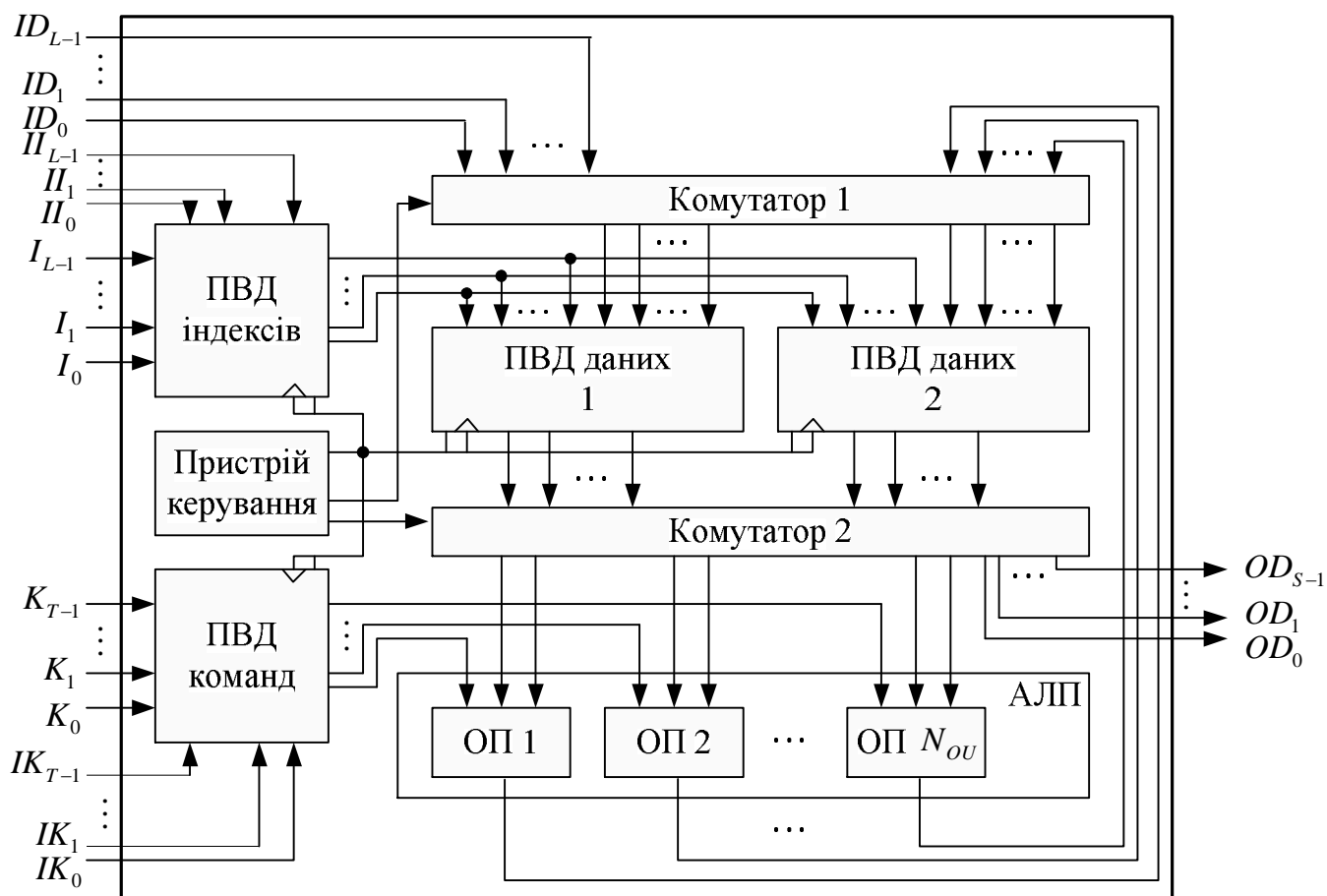


Рис. 3. Ітераційна структура паралельного СП для виконання алгоритмів з інваріантною до даних структурою з двома блоками ПВД даних

Розроблені типи структур характеризуються гранично високою ефективністю при виконанні таких алгоритмів, у яких на кожному кроці використовуються всі наявні в ПВД дані. Головним недоліком цих структур є потреба здійснення пересилань всього масиву вихідних даних одного ярусу графа алгоритму між блоками ПВД, що для алгоритмів з невеликою часткою використовуваних на наступному ярусі даних зумовлює надлишкові пересилання та відповідні часові втрати. Для усунення цього недоліку розроблено структуру СП для виконання алгоритмів з інваріантною до даних структурою з адаптованою до алгоритму кількістю блоків ПВД даних (рис. 4). Наведена структура забезпечує найвищий рівень відповідності СП виконуваному алгоритму, який може бути досягнутий застосуванням розробленого способу опрацювання інформації, а відповідно й найвищу швидкість виконання алгоритму порівняно з іншими запропонованими варіантами, однак характеризується вищими затратами обладнання.

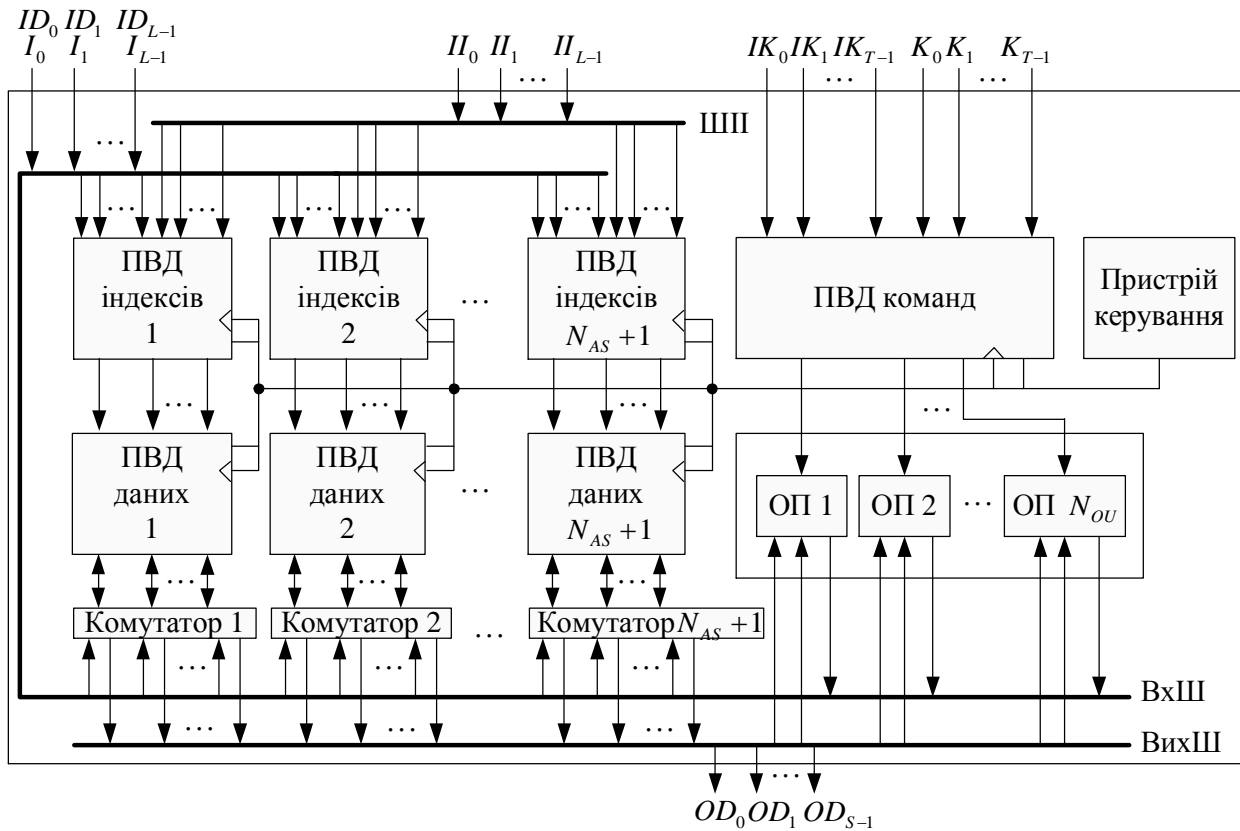


Рис. 4. Структура паралельного СП для виконання алгоритмів з інваріантною до даних структурою з адаптованою до алгоритму кількістю блоків ПВД даних

Розроблено принципи формування параметрів компонентів програми СП для виконання алгоритмів з інваріантною до даних структурою, а саме: початкових даних, проміжних даних, кінцевих даних, команд, виконуваних команд, індексів початкових даних, індексів проміжних та кінцевих даних, індексів команд, індексів проміжних даних, індексів індексів початкових даних, та індексів індексів кінцевих даних. Кожен із зазначених компонентів подано як двовимірну матрицю, кількість рядків якої визначається кількістю тактів запису відповідного компонента до ПВД або його зчитування звідти, а кількість стовбців – відповідно, кількістю вхідних або вихідних каналів цієї ПВД. Наприклад, матриця початкових даних має такий вигляд:

$$\begin{vmatrix} ID_{0,0} & \dots & ID_{0,l} & \dots & ID_{0,L-1} \\ \dots & \dots & \dots & \dots & \dots \\ ID_{k,0} & \dots & ID_{k,l} & \dots & ID_{k,L-1} \\ \dots & \dots & \dots & \dots & \dots \\ ID_{K-1,0} & \dots & ID_{K-1,l} & \dots & ID_{K-1,L-1} \end{vmatrix} \cdot \quad (12)$$

Тут $ID_{k,l}$ – початкове дане, яке знаходиться в k -му рядку та l -му стовбці. При цьому $l=0,1,\dots,L-1$, де L – кількість стовбців матриці початкових даних і, відповідно, входів, якими початкові дані надходять до СП, $k=0,1,\dots,K-1$, де K – кількість рядків матриці початкових даних, яка рівна кількості тактів запису цих даних в ПВД даних. Розмір матриці початкових даних $Q_{ID} = K \times L$ визначає кількість цих даних. Матриця виконуваних команд має такий вигляд:

$$\begin{pmatrix} PK_{0,0} & \dots & PK_{0,f} & \dots & PK_{0,F-1} \\ \dots & \dots & \dots & \dots & \dots \\ PK_{e,0} & \dots & PK_{e,f} & \dots & PK_{e,F-1} \\ \dots & \dots & \dots & \dots & \dots \\ PK_{E-1,0} & \dots & PK_{E-1,f} & \dots & PK_{E-1,F-1} \end{pmatrix}. \quad (13)$$

Тут $PK_{e,f}$ – виконувана команда, яка знаходиться в e -му рядку та f -му стовбці, де e – номер рядка матриці виконуваних команд, $e=0,1,\dots,E-1$, де E – кількість рядків матриці виконуваних команд і, відповідно, кількість тактів зчитування команд з пам'яті команд; f – номер стовпця, $f=0,1,\dots,F-1$, де F – кількість стовбців, рівна кількості каналів надходження команд до АЛП і, відповідно, кількості операційних пристроїв в АЛП. Розмір матриці виконуваних команд $Q_{PK} = E \times F = Q_K$ визначає кількість команд, які надходять до АЛП.

На основі розроблених принципів формування параметрів компонентів програми проведено розрахунок та дослідження технічних характеристик СП для виконання алгоритмів з інваріантною до даних структурою, а саме: об'єму ПВД даних, об'єму ПВД команд, об'єму ПВД індексів, кількості операційних пристроїв в АЛП СП та його продуктивності.

Розроблено методики розрахунку та отримано вирази для оцінювання тривалості виконання програми в СП для виконання алгоритмів з інваріантною до даних структурою з адаптованою до алгоритму кількістю блоків ПВД даних і з двома блоками ПВД даних.

Розглянуто питання проектування СП для виконання алгоритмів з інваріантною до даних структурою. Сформовано послідовність проектування і подано перелік методів, розроблення та реалізація яких є основою побудови одного з базових компонентів СККС – системи генерування програмних моделей цих спеціалізованих процесорів.

У п'ятому розділі “Розроблення та дослідження методів побудови в самоконфігурованих комп'ютерних системах засобів генерування програмних моделей спеціалізованих процесорів” розвинуто теорію високорівневого проектування ПМСП.

Проведено класифікацію методів проектування ПМСП для реконфігурованих прискорювачів і на їх основі запропоновано чотири методи побудови в СККС засобів генерування ПМСП, а саме: перший – метод побудови засобів генерування з використанням бібліотек ПМСП, другий – метод побудови засобів генерування з

використанням бібліотек компонент ПМСП, третій – метод побудови засобів генерування з використанням конфігурованих ПМСП, та четвертий – метод побудови засобів генерування ПМСП з описів алгоритмів мовою високого рівня.

Проведено оцінювання доцільності застосування в СККС запропонованих методів побудови засобів генерування ПМСП. Розроблено структури та досліджено відповідність побудованих за цими методами засобів вимогам до системи генерування в частині її функціональної повноти, в частині технічних характеристик генерованих нею ПМСП, та в частині архітектури генерованих ПМСП. Встановлено, що єдиним типом засобів генерування, які характеризуються функціональною повнотою, є засоби генерування ПМСП з описів алгоритмів мовою високого рівня. Також встановлено, що вимогам в частині архітектури та в частині технічних характеристик СП відповідають генератори на основі бібліотек ПМСП та компонентів ПМСП, а також на основі конфігурованих ПМСП. При цьому програмні моделі в бібліотеках доцільно представляти на рівні файлів конфігурації ПЛІС реконфігурованого середовища СККС, що дозволить скоротити та спростити процес компіляції програми в СККС, уникнувши потреби виконання логічного синтезу процесорів. Використання в СККС третього методу побудови засобів генерування програмних моделей спеціалізованих процесорів є недоцільним з огляду на те, що вони поступаються можливостями генераторам на основі бібліотек ПМСП.

Сформульовано вирази для оцінювання характеристик тривалості виконання генерування ПМСП різними типами засобів та встановлено, що мінімальним часом генерування характеризуються генератори на основі бібліотек ПМСП, дещо більшим – генератори на основі бібліотек компонентів ПМСП і на основі конфігурованих ПМСП, і найбільшим – засоби генерування ПМСП з описів алгоритмів мовою високого рівня, тобто:

$$t_{gen}(HLDS) > t_{gen}(G_{LPCPM}) \approx t_{gen}(G_{CPM}) > t_{gen}(G_{LPPM}). \quad (14)$$

В результаті проведених досліджень прийнято, що четвертий метод, зважаючи на його функціональну повноту, може бути застосований як основний для побудови системи генерування, а інші – як допоміжні у випадку використання СККС для виконання наперед визначеного класу алгоритмів.

Сформульовано теоретичні основи організації функціонування та визначено основні етапи роботи засобів генерування ПМСП з описів алгоритмів мовою високого рівня, а саме: введення алгоритму, перетворення в проміжний код, визначення та аналіз структури алгоритму та оптимізаційний синтез схеми СП. Запропоновано таку класифікацію засобів проектування ПМСП від алгоритму до рівня міжрегістрових передач: 1) переведенням опису алгоритму мовою високого рівня в модель апаратно-програмної системи, 2) переведенням опису алгоритму мовою високого рівня в логічні вентиля ПЛІС, та 3) конфігуруванням їх базової конфігурованої програмної моделі.

За результатами проведених досліджень та оцінювання доцільності застосування в СККС запропонованих методів побудови засобів генерування ПМСП розроблено структуру (рис. 5) і алгоритмічні основи роботи системи генерування, до складу якої включено декілька типів засобів, а саме – генератори на основі бібліотек ПМСП та

компонентів ПМСП і засоби генерування ПМСП з описів алгоритмів мовою високого рівня.

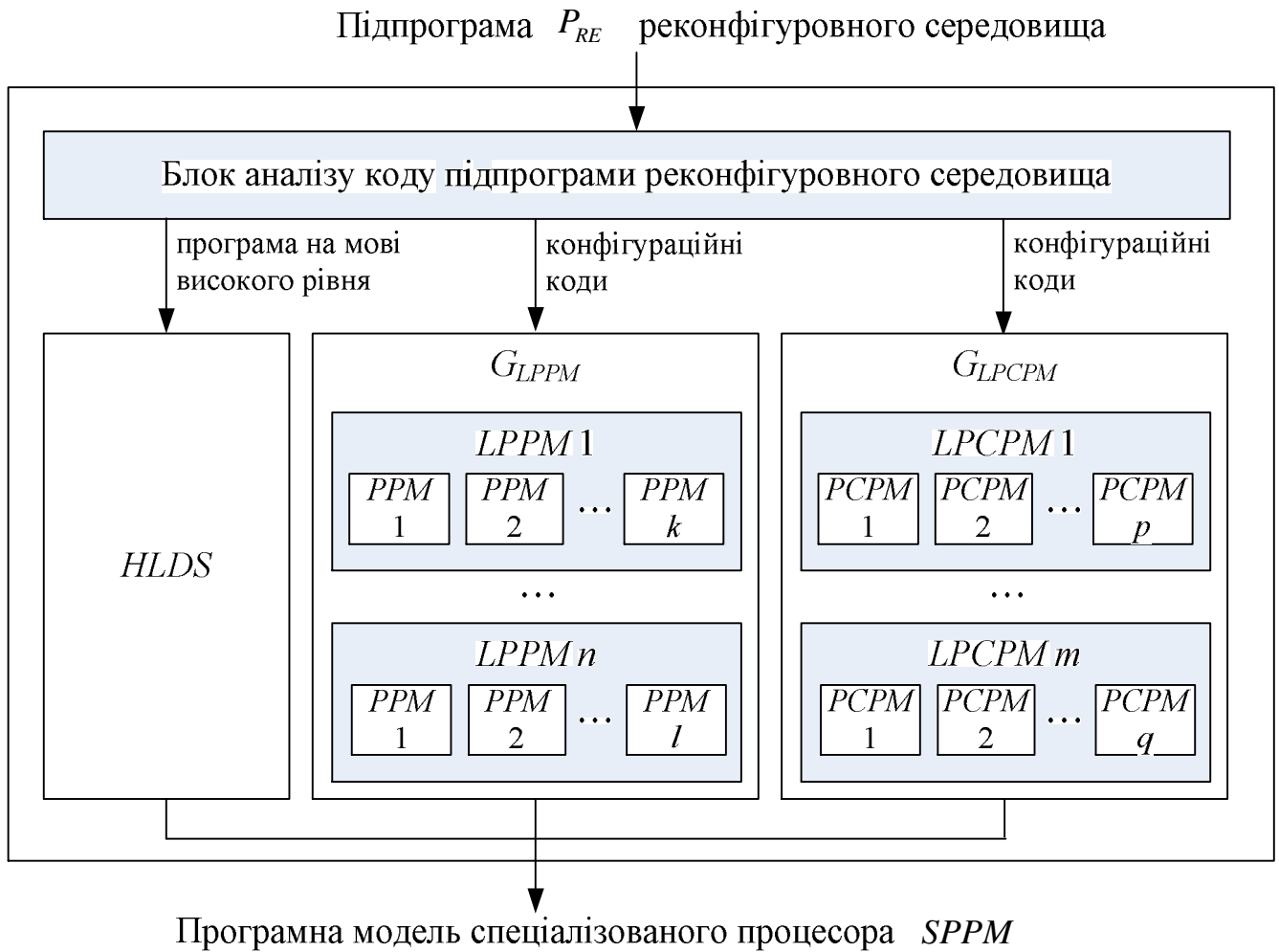


Рис. 5. Структура системи генерування ПМСП в СККС

В наведеній структурі генератор G_{LPPM} на основі бібліотек ПМСП містить множину бібліотек $LPPM_i$, $i = \overline{1..n}$, де n – кількість бібліотек, а кожна i -та бібліотека складається з множини ПМСП PPM_j , $j = \overline{1..z}$, де z – кількість ПМСП в бібліотеці, причому $z = k$ при $i = 1$; $z = l$ при $i = n$, тобто:

$$\begin{aligned} G_{LPPM} &= (\{LPPM_i, i = \overline{1..n}\}, SEM); \\ LPPM_i &= \{PPM_j, j = \overline{1..z}\} \end{aligned} \quad (15)$$

де SEM – засоби пошуку і вибірки відповідної вхідним конфігураційним кодам ПМСП з бібліотеки. Кожна з програмних моделей PPM_j представлена у вигляді файлу конфігурації ПЛІС реконфігуровного середовища СККС.

Аналогічно, генератор G_{LPCPM} на основі бібліотек компонентів ПМСП містить множину бібліотек, $r = \overline{1..m}$, де m – кількість бібліотек, а кожна r -та бібліотека

складається з множини програмних моделей компонент процесорів $PCPM_s$, $s = \overline{1...w}$, де w – кількість компонент в бібліотеці, причому $w = p$ при $r = 1$; $w = q$ при $r = m$, тобто:

$$G_{LPCPM} = \left(\{LPCPM_r, r = \overline{1...m}\}, SETLGM \right); \quad (16)$$

$$LPCPM_r = \{PCPM_s, s = \overline{1...w}\}$$

де *SETLGM* – засоби пошуку і вибірки відповідних вхідним конфігураційним кодам компонент ПМСП з бібліотеки та генерації файлів опису верхнього рівня СП. Кожна з програмних моделей компонент $PCPM_s$ подається мовою опису апаратних засобів.

Застосування вибраних засобів в системі генерування забезпечує її відповідність вимогам в частині функціональної повноти, в частині технічних характеристик ПМСП та в частині архітектури ПМСП.

У шостому розділі “Розроблення та дослідження засобів взаємодії компонентів самоконфігуровної комп’ютерної системи” сформовано вимоги до інтерфейсу між компонентами СККС, а саме: цей інтерфейс повинен належати до стандартних інтерфейсів, якими зазвичай обладнано універсальний комп’ютер, характеризуватися високою пропускнуою здатністю, надавати можливість керування живленням реконфігуровного прискорювача та можливість його приєднання під час роботи. Проведено дослідження найпоширеніших сучасних інтерфейсів, зокрема *USB*, *FireWire*, також відомого як стандарт *IEEE 1394*, *PCI*, *PCI-X* та *PCI Express* та встановлено, що серед них наведеним вимогам найкраще відповідає інтерфейс *PCI Express*. Крім того встановлено дуже важливу перевагу цього інтерфейсу, а саме – він використовується в РККС як слабозв’язаної архітектури, так і тіснозв’язаної – у платформі *Intel* з підключенням до процесора хост-комп’ютера через північний міст, а у платформі *AMD* – через міст *HyperTransport / PCI-Express*.

Розроблено базову архітектуру програмно-апаратної системи для організації взаємодії між комп’ютером та реконфігуровним прискорювачем.

Розроблено модель взаємодії апаратних засобів комп’ютера та реконфігуровного прискорювача СККС через інтерфейс *PCI Express*, та на її основі розроблено структуру апаратно-програмної системи для організації цієї взаємодії, яку можна побудувати шляхом розроблення та введення до складу СККС трьох компонент: інтерфейсу прикладного програмування додатків користувача в комп’ютері, драйвера реконфігуровного прискорювача, та апаратної частини інтерфейсу спеціалізованого процесора в реконфігуровному прискорювачі, як це показано на рис. 6.

Визначено склад та розглянуто особливості проектування програмних засобів для організації взаємодії між комп’ютером та реконфігуровним прискорювачем, а саме: драйвера реконфігуровного прискорювача, що відповідає за конфігурацію та обмін даними з ним на низькому рівні, бібліотеки рівня користувача, яка дозволяє програмісту абстрагуватись від запитів рівня драйвера прискорювача та працювати з ним на рівні абстрактних запитів, та системних утиліт для роботи з реконфігуровним прискорювачем. Визначено та реалізовано функції цих системних

утиліт, а саме: отримати інформацію про стан прискорювача (підключений/не підключений); створити область пам'яті в прискорювачі та записати до неї масив даних; запустити на виконання завдання на прискорювачі; виконувати моніторинг кількості опрацьованих даних; повідомити операційну систему комп'ютера про завершення виконання обчислень; вичитати масив результатів обчислень. За цільову платформу при реалізації з міркувань поширеності обрано *Microsoft Windows*.

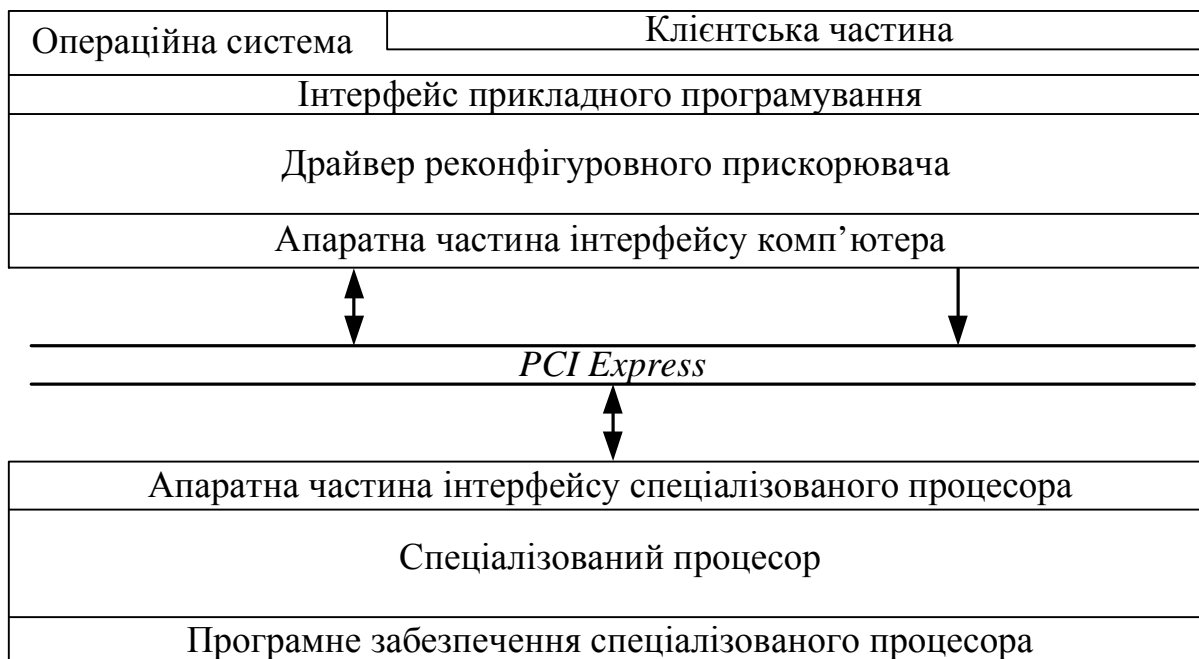


Рис. 6. Структура програмно-апаратної системи для організації взаємодії між комп'ютером та реконфігуровним прискорювачем

Розроблено драйвер реконфігуровного прискорювача СККС, який виконує наступні базові операції: початкове конфігурування прискорювача, його динамічне реконфігурування на запит клієнтської програми, запис масиву даних до прискорювача, зчитування масиву даних з прискорювача і надсилання відповідного інформаційного повідомлення клієнтській програмі, та інформування клієнтської програми про зміну режиму живлення прискорювача.

Запропоновано метод організації віддаленого використання реконфігуровних прискорювачів, який полягає у наданні реконфігуровного прискорювача як сервісу через комп'ютерну мережу. Його використання сприятиме ширшому впровадженню реконфігуровних прискорювачів як альтернативи традиційним багатопроесорним комп'ютерним системам і таким чином забезпечить економію коштів навчальним, науковим та іншим організаціям відповідного напрямку діяльності, а також зробить їх доступнішими. Розроблено принципи структурної організації системи віддаленого доступу до реконфігуровних прискорювачів та методіку її налаштування. Розроблено схему та два методи взаємодії користувачів з реконфігуровним прискорювачем в межах системи віддаленого доступу – 1) від сервера та 2) від клієнта.

У цьому розділі “Реалізація та експериментальні дослідження самоконфігурованих високопродуктивних комп’ютерних систем” викладено результати використання розроблених основ теорії побудови паралельних спеціалізованих процесорів для виконання алгоритмів з інваріантною до даних структурою, удосконалень теорії високорівневого проектування ПМСП, та практичної реалізації базових компонентів СККС.

Відповідно до розроблених в Розділі 3 концептуальних основ побудови системи розподілу обчислювального навантаження між комп’ютером і реконфігурованим середовищем здійснено реалізацію цієї системи. Систему реалізовано згідно з методом розподілу обчислювального навантаження між комп’ютером та реконфігурованим середовищем, в основу якого покладено виконання проходу розподілу коду. При виконанні проходу здійснюються наступні дії:

1. Аналіз коду вхідної програми з метою виділення в ній лінійних ділянок, виконання яких вимагає значних ресурсів процесорного часу комп’ютера;
2. Визначення спільного регіону для цих лінійних ділянок програми;
3. Переміщення визначеного регіону в окрему функцію, яка буде виконуватися в реконфігурованому середовищі.

Якщо під час виконання проходу розподілу коду система виділила регіон, доцільний для виконання в реконфігурованому середовищі, тоді вона винесе його з основного коду програми у вигляді функції в окремий вихідний файл, призначений для виконання в реконфігурованому середовищі, а на місце регіону поставить віддалений виклик цієї функції.

Під час розроблення системи розподілу обчислювального навантаження використано технологію та набір бібліотек *LLVM*. Проведено експериментальні дослідження, результати яких підтвердили коректність покладених в основу побудови системи розподілу обчислювального навантаження методів.

Відповідно до розробленого в Розділі 5 методу побудови системи генерування ПМСП розроблено структуру та здійснено реалізацію генератора на основі бібліотеки компонент ПМСП. На рис. 7 показано його структуру.

Генератор побудовано з використанням таких складових: бібліотеки *VHDL*-файлів з описами компонент спеціалізованих процесорів; програми генерації файлів верхнього рівня; головного конфігураційного файлу (файлу найвищого рівня ієрархії, кількість полів у якому визначає кількість програмних моделей спеціалізованих процесорів, що можуть бути згенеровані системою з використанням даної бібліотеки компонент); набору вторинних конфігураційних файлів (файлів нижчих рівнів ієрархії, кожен з яких описує підмножину компонент моделі процесора на певному рівні ієрархії).

Відповідно до розробленої в Розділі 4 архітектури проведено проектування паралельного спеціалізованого процесора для виконання алгоритмів з інваріантною до даних структурою на прикладі алгоритму ШПФ за методом Кулі-Тьюкі. Для цього визначено вихідну інформацію, а саме: параметри матриць вхідних даних, параметри матриць вихідних даних, сформовано матриці вхідних даних та правила впорядкування даних для них, а також сформовано матриці команд для операційного пристрою процесора шляхом проведення маркування ПГА ШПФ. З використанням мови *VHDL* розроблено програмну модель паралельного

спеціалізованого процесора для виконання алгоритму ШПФ, проведено синтез його вузлів та отримано відповідні схемотехнічні рішення. Здійснено моделювання та тестування процесора, що показало коректність його функціонування. Здійснено реалізацію в ПЛІС розробленої програмної моделі процесора та проведено оцінку досягнутих ним характеристик. Зазначений процесор, реалізований в ПЛІС XC4085XLABG560 фірми *Xilinx*, порівняно з існуючими аналогами характеризується майже в 2 рази нижчими затратами обладнання при підвищенні швидкодії в 1,5 рази.

Конфігураційні параметри

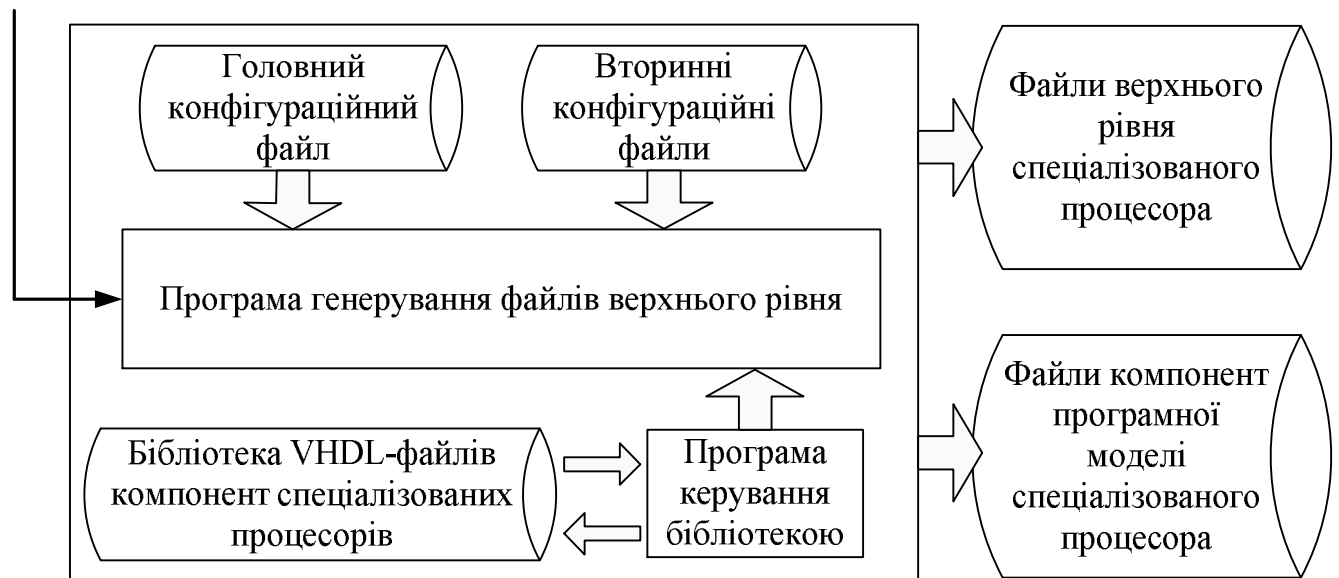


Рис. 7. Структура генератора на основі бібліотеки програмних моделей компонент спеціалізованих процесорів

Розроблено з використанням мови *VHDL* програмні моделі таких типів паралельної ПВД для паралельних спеціалізованих процесорів для виконання алгоритмів з інваріантною до даних структурою: з попереднім налаштуванням, з паралельним надходженням даних та індексів, із змінним впорядкованим доступом та з фіксованим впорядкованим доступом. Проведено синтез програмних моделей паралельної ПВД вказаних типів і отримано відповідні схемотехнічні рішення, здійснено моделювання їх роботи та показано коректність функціонування. Здійснено реалізацію в ПЛІС розроблених програмних моделей ПВД та проведено оцінку досягнутих характеристик. В результаті їх реалізації в ПЛІС *bvcsx75tff484-2* фірми *Xilinx* отримано значення тактової частоти від 802 до 1012 МГц, що означає, що ця пам'ять може працювати на максимальній частоті ПЛІС, а це, з врахуванням надання нею паралельного доступу до даних, забезпечує їй продуктивність, недосяжну для інших типів пам'яті.

Результати моделювання та синтезу процесора ШПФ і пристроїв ПВД довели ефективність розроблених теоретичних рішень з проектування спеціалізованих процесорів для виконання алгоритмів з інваріантною до даних структурою і їх коректність.

Розроблено засоби для проведення експериментальних досліджень і оцінювання характеристик РККС залежно від складності задачі.

ОСНОВНІ РЕЗУЛЬТАТИ РОБОТИ ТА ВИСНОВКИ

У дисертаційній роботі вирішено актуальну науково-практичну проблему підвищення продуктивності та ефективності використання комп'ютерних систем з реконфігуровною логікою шляхом створення нового класу комп'ютерних засобів – самоконфігуровних високопродуктивних комп'ютерних систем, методів та засобів їх побудови та організації функціонування. При цьому отримано наступні результати:

1. Запропоновано визначення основних понять і класифікацію базових типів архітектури РККС, розроблено спосіб опрацювання в них інформації, принципи його реалізації та методику розрахунку тривалості опрацювання інформації, що дозволило виявити вузькі місця, які є причиною недостатньої ефективності РККС на певних класах задач, та знайти шляхи їх вдосконалення.
2. Проведено дослідження і порівняльний аналіз типів архітектури РККС та обґрунтовано доцільність застосування в них реконфігурованих прискорювачів, проведено дослідження засобів взаємодії компонентів РККС слабозв'язаної та тіснозв'язаної типів архітектури з позиції організації зв'язку між комп'ютером і реконфігуровним прискорювачем та показано організацію їх функціонування, оцінено технічні характеристики існуючих реконфігурованих прискорювачів та проведено експериментальні дослідження характеристик РККС залежно від складності задачі, що дало можливість визначити області доцільного застосування реконфігурованих прискорювачів.
3. На основі аналізу способу опрацювання інформації в РККС виокремлено основні проблеми, що перешкоджають підвищенню їх ефективності, та запропоновано підхід до вирішення цих проблем, який полягає в автоматизації виконання розподілу обчислювального навантаження між універсальним комп'ютером та реконфігуровним середовищем, автоматизації створення програмних моделей спеціалізованих процесорів, та вдосконаленні способу опрацювання інформації в РККС таким чином, щоб завантаження до реконфігуровного середовища отриманих в результаті логічного синтезу файлів конфігурації здійснювалось не користувачем, а операційною системою паралельно з завантаженням виконавчого файлу підпрограми універсального комп'ютера до його основної пам'яті після подання користувачем команди ініціалізації виконання програми.
4. Запропоновано концепцію, принципи побудови та організації функціонування нового типу високопродуктивних комп'ютерних засобів – самоконфігурованих комп'ютерних систем, які, у порівнянні з РККС, мають вищу продуктивність та ефективні на довільних класах алгоритмів. Розроблено концептуальні основи побудови їх складових частин – системи розподілу обчислювального навантаження та системи генерування програмних моделей спеціалізованих процесорів, сформовано вимоги до засобів логічного синтезу спеціалізованих процесорів і конфігурування ПЛІС, реконфігуровного середовища та його драйвера, що дало можливість здійснити їх проектування та реалізацію. Розроблені структура, принципи побудови та організації функціонування СККС є основою створення нових високопродуктивних комп'ютерних систем з реконфігуровною логікою, в яких буде використано всі потенційні можливості,

надані властивістю зміни конфігурації реконфігуровного середовища, що забезпечує їм одне з чільних місць серед найперспективніших засобів високопродуктивних обчислень.

5. Розроблено метод самоконфігурування комп'ютерних систем з реконфігуровною логікою та відповідний йому спосіб опрацювання інформації в таких комп'ютерних системах, в якому всі дії виконуються автоматично, включаючи розподіл програми на підпрограми універсального комп'ютера та реконфігуровного середовища, створення на мові опису апаратних засобів програмної моделі спеціалізованого процесора для виконання підпрограми реконфігуровного середовища, виконання її логічного синтезу та завантаження до реконфігуровного середовища отриманих в результаті логічного синтезу файлів конфігурації, та який, на відміну від способу опрацювання інформації в РККС, де ці дії виконуються користувачем, дозволяє значного підвищити продуктивність та розширити клас вирішуваних комп'ютерною системою задач.
6. Розроблено методику та подано вирази для розрахунку характеристик тривалості процесів опрацювання інформації в СККС, проведено дослідження основних процесів на етапах завантаження та виконання програми, та розроблено методику оцінювання тривалості виконання програми в СККС і забезпечуваного нею прискорення, що дало можливість довести її принципові переваги над РККС і підтвердило ефективність положень, покладених в основу способу опрацювання в ній інформації.
7. Запропоновано модифікацію розробленого способу опрацювання інформації та структури СККС для випадку її багатопроекторної реалізації, розроблено методику розрахунку часу опрацювання в ній інформації та методику оцінювання забезпечуваного нею прискорення, що підтвердило доцільність застосування методу самоконфігурування у високопродуктивних багатопроекторних комп'ютерних системах.
8. Обґрунтовано необхідність та розроблено новий спосіб опрацювання інформації з впорядкованими компонентами програми, який, на відміну від відомих способів опрацювання інформації, враховує в програмі просторові та часові параметри виконуваного алгоритму та забезпечує паралельне виконання алгоритмів з інваріантною до даних структурою.
9. Розроблено нову архітектуру паралельних СП для виконання в реконфігуровному середовищі СККС алгоритмів з інваріантною до даних структурою, яка базується на новому способі опрацювання інформації з впорядкованими компонентами програми. Розроблено базові структури таких СП, проведено розрахунок і дослідження їхніх технічних характеристик, розроблено методику розрахунку і отримано вирази для оцінювання тривалості виконання ними програми, розроблено основи їх проектування, що дало можливість сформулювати теоретичну базу для побудови систем автоматичного проектування програмних моделей спеціалізованих процесорів для виконання алгоритмів з інваріантною до даних структурою. Розроблена архітектура спеціалізованих процесорів для виконання алгоритмів з інваріантною до даних структурою з паралельним опрацюванням даних може бути покладена в основу

побудови високопродуктивних спеціалізованих процесорів для виконання завдань широкого кола застосувань.

10. Запропоновано чотири методи побудови в СККС засобів генерування програмних моделей спеціалізованих процесорів та проведено оцінювання доцільності їх застосування в цих системах, розроблено структури засобів генерування та досліджено їх на відповідність вимогам до системи генерування, сформульовано вирази для оцінювання часових характеристик роботи цих засобів, що дало можливість розробити структуру і основи організації функціонування системи генерування програмних моделей спеціалізованих процесорів в СККС, яка відповідає вимогам в частині функціональної повноти, в частині технічних характеристик моделей процесорів, та в частині архітектури моделей процесорів.
11. Сформовано вимоги до інтерфейсу між компонентами СККС та розроблено базову архітектуру програмно-апаратної системи для організації взаємодії між комп'ютером та реконфігуровним прискорювачем і модель такої взаємодії в СККС, що дало можливість здійснити проектування та реалізацію програмних засобів взаємодії між комп'ютером та реконфігуровним прискорювачем. Розроблені засоби організації взаємодії компонентів СККС можуть бути використані для організації взаємодії реконфігуровних прискорювачів в довільних комп'ютерних системах, у тім числі й з віддаленим доступом.
12. Проведено експериментальні дослідження СККС та здійснено реалізацію системи розподілу обчислювального навантаження між комп'ютером і реконфігуровним середовищем і системи генерування програмних моделей спеціалізованих процесорів, проведено проектування та розроблено з використанням мови *VHDL* програмну модель паралельного спеціалізованого процесора для виконання алгоритмів з інваріантною до даних структурою на прикладі алгоритму ШПФ за методом Кулі-Тьюкі та програмні моделі чотирьох типів паралельної пам'яті з впорядкованим доступом. Результати проведених експериментальних досліджень підтвердили коректність покладених в основу побудови та організації функціонування СККС методів та ефективність реалізованих засобів.

СПИСОК ПРАЦЬ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

1. Мельник В.А. Персональні суперкомп'ютери: архітектура, проектування, застосування: монографія / А.О. Мельник, В.А. Мельник. – Львів: Видавництво Львівської політехніки, 2013. – 516 с.
2. Melnyk W. Parametry algorytmu / A. Melnyk, W. Melnyk // Zeszyty Naukowe Politechniki Świętokrzyskiej. Elektryka, №43. – Wydawnictwo Politechniki Świętokrzyskiej, Kielce, 2005 r. – Str. 115-121.
3. Мельник В.А. Модель архітектури спеціалізованого процесора / А.О. Мельник, В.А. Мельник, Мохаммад Аль Хабабсах // Науковий вісник Чернівецького університету: Збірник наукових праць, вип. 479: Комп'ютерні системи та компоненти. Чернівці, 2009. – С.6-10.
4. Мельник В.А. Побудова генераторів програмних моделей комп'ютерних систем на кристалі / В.А. Мельник, З. Сарайрех // Науково-технічний журнал

Національного аерокосмічного університету ім. М.Є. Жуковського “Харківський авіаційний інститут” «Радіоелектронні і комп’ютерні системи», – №7 (48), Харків "ХАІ" 2010, – С.215-219.

5. Мельник В.А. Використання реконфігуровних прискорювачів для підвищення продуктивності персональних комп’ютерів / А.О. Мельник, В.А. Мельник, З.Т. Сарайрех // Науковий вісник Чернівецького ун-ту. Комп’ютерні системи та компоненти. – Чернівці: Чернівецький національний університет імені Юрія Федьковича, 2010. - Т.1 Вип.1. – С. 20-25.
6. Мельник В.А. Віртуальна лабораторія реконфігуровних прискорювачів з дистанційним доступом / В.А. Мельник // Науковий вісник Чернівецького ун-ту. Комп’ютерні системи та компоненти. – Чернівці: Чернівецький національний університет імені Юрія Федьковича, 2010. - Т.1 Вип.2. – С. 6-11.
7. Мельник В. Організація взаємодії реконфігуровного прискорювача з універсальним комп’ютером / В. Мельник, А. Юрчук, К. Ліпич, З. Сарайрех // Вісник Національного університету „Львівська політехніка” «Комп’ютерні науки та інформаційні технології». – № 686, Львів, 2010, – С.81 - 86.
8. Мельник В.А. Самоконфігуровні апаратні прискорювачі обчислень в комп’ютерах / В.А. Мельник, З.Т. Сарайрех // Вісник Національного університету „Львівська політехніка” «Комп’ютерні системи та мережі». – №688, Львів, 2010, – С.163 – 171.
9. Мельник В.А. Засоби генерації програмних процесорів на основі бібліотеки їх компонент / В.А. Мельник, З.Т. Сарайрех // Вісник Національного університету „Львівська політехніка” «Автоматика, вимірювання та керування». №695, Львів, 2011. –С.82 – 87.
10. Мельник В.А. Проектування багатоканального процесора ШПФ з використанням пам’яті з впорядкованим доступом / А.О. Мельник, Мохаммад аль Хабабсах, В.А. Мельник // Міжвузівський збірник "Комп’ютерно-інтегровані технології: освіта, наука, виробництво". Науковий журнал Луцького національного технічного університету. Луцьк, 2011. Випуск №5. –С. 171-177.
11. Мельник В.А. Стан та перспективи розвитку високопродуктивних обчислювальних систем / В.А. Мельник // Вісник Національного університету „Львівська політехніка” «Комп’ютерні системи та мережі». №717, Львів, 2011. –С.96 – 104.
12. Мельник В.А. Реалізація в програмованих логічних інтегральних схемах паралельної пам’яті з фіксованим впорядкованим доступом / В.А. Мельник // Науковий вісник Чернівецького університету. Комп’ютерні системи та компоненти. – Чернівці: Чернівецький національний університет імені Юрія Федьковича. 2011. Т. 2. Вип. 4. –С.12-16.
13. Мельник В.А. Реалізація в ПЛІС пам’яті з впорядкованим доступом з паралельним надходженням даних та індексів / А.О. Мельник, В.А. Мельник // Міжвузівський збірник "Комп’ютерно-інтегровані технології: освіта, наука, виробництво". Науковий журнал Луцького національного технічного університету. Луцьк, 2012. Випуск №8. – С.71-76.
14. Мельник В.А. Система розподілу обчислювального навантаження між хост-комп’ютером та самоконфігуровним прискорювачем / В.А. Мельник, В.А.

- Степанов, З.Т. Сарайрех // Науковий вісник Чернівецького університету. Комп'ютерні системи та компоненти. – Чернівці: Чернівецький національний університет імені Юрія Федьковича, 2012. – Т. 3. Вип. 1. – С.6-16.
15. Мельник В.А. Експериментальні дослідження засобів розподілу обчислювального навантаження в реконфігурованих комп'ютерних системах / В.А. Мельник, В.А. Степанов, З.Т. Сарайрех. // Технічні вісті. – Львів, 2012. – С. 21-25.
 16. Мельник В.А. Основи побудови та організації функціонування реконфігурованих комп'ютерних систем / В.А. Мельник // Науковий журнал «Вісник Хмельницького Національного Університету», Серія «Технічні науки». – №6. – 2012 р. – С. 212 – 217.
 17. Мельник В.А. Структурна організація паралельної пам'яті з змінним впорядкованим доступом та її реалізація в ПЛІС / В.А. Мельник // Вісник Національного університету „Львівська політехніка” «Автоматика, вимірювання та керування». – №741, Львів, 2012, –С.11 – 16.
 18. Мельник В.А. Засоби автоматичного генерування програмних моделей спеціалізованих процесорів у самоконфігурованій комп'ютерній системі / В.А. Мельник // Міжнародний науково-технічний журнал «Інформаційні технології та комп'ютерна інженерія», - №3 (25), 2012. – С.65-70.
 19. Мельник В.А. Дослідження характеристик засобів генерування програмних моделей спеціалізованих процесорів в самоконфігурованій комп'ютерній системі / В.А. Мельник // Вісник Національного університету „Львівська політехніка” «Комп'ютерні науки та інформаційні технології». № 744, Львів, 2012. – С. 86-93.
 20. Мельник В.А. Використання пам'яті з упорядкованим доступом у процесорах опрацювання зображень / А.О. Мельник, В.А. Мельник, І.В. Мороз, Я.С. Парамуд // Вісник Національного університету „Львівська політехніка” «Комп'ютерні системи та мережі». - № 745, Львів, 2012. – С. 118-123.
 21. Melnyk V. Intelligent Agents and Multi-Agent Systems in Automation, Computing and Networking / V. Melnyk, M. Plaza // Central European Exchange Programme for University Studies (CEEPUS) Summer School 2005, Intelligent Control Systems. Brno, Czech Republic, 2005. –P. 61-65.
 22. Melnyk W. IP Cores Generators in SoC Design / A. Melnyk, W. Melnyk // Proceedingd of the 5th international Conference for Students and Young Scientists „Telecommunication in XXI Century“, Poland, Wolka Milanowska, 2005. –P. 23-28.
 23. Melnyk V. IEEE 802.15.4b-Compatible AES-CCM* Security Module Application in IEEE 802.11i Environment / V. Melnyk // Proceedings of 3-rd International Conference “Advanced Computer Systems and Networks: Design and Application”, 2007, Lviv, Ukraine. – P. 139-142.
 24. Мельник В.А. Методи конфігурування моделей спеціалізованих процесорів / В.А. Мельник, Мохаммад Аль Хабабсах // IV Всеукраїнська науково-практична конференція «Комп'ютерні технології: наука і освіта», м. Луцьк, Луцький інститут розвитку людини Університету «Україна», 2009р. –С.121-125.
 25. Мельник В.А. Організація бібліотек ядер стандартизованих та замовних комп'ютерних пристроїв для високопродуктивних реконфігурованих

- прискорювачів / А.О. Мельник, В.А. Мельник // IV Всеукраїнська науково-практична конференція «Комп'ютерні технології: наука і освіта», м. Луцьк, Луцький інститут розвитку людини Університету «Україна», 2009р. –С.113-117.
26. Мельник В.А. Технології та засоби проектування ядер стандартизованих та замовних комп'ютерних пристроїв для високопродуктивних прискорювачів / А.О. Мельник, В.А. Мельник // IV Всеукраїнська науково-практична конференція «Комп'ютерні технології: наука і освіта», м. Луцьк, Луцький інститут розвитку людини Університету «Україна», 2009р. –С.117-121.
27. Мельник В.А. Тенденції розвитку персональних суперкомп'ютерів / В.А. Мельник // Матеріали 5-ої Міжнародної науково-технічної конференції «Сучасні комп'ютерні системи та мережі: розробка та використання» (ASCN-2011). – Львів, 2011. – С.10 – 13.
28. Мельник В.А. Вимоги до системи генерування моделей процесорів самоконфігуровного апаратного прискорювача / В.А. Мельник, З.Т. Сарайрех // Матеріали 5-ої Міжнародної науково-технічної конференції «Сучасні комп'ютерні системи та мережі: розробка та використання» (ASCN-2011). – Львів, 2011. – С.255 – 258.
29. Мельник В.А. Метод самоконфігурування апаратного прискорювача / В.А. Мельник, З.Т. Сарайрех // Матеріали V Міжнародної конференції молодих вчених “Комп'ютерні науки та інженерія 2011” (CSE-2011), Львів. –С.126-127.
30. Мельник В.А. Паралельна пам'ять з фіксованим впорядкованим доступом для апаратних засобів захисту інформації / В.А. Мельник // Матеріали I-ої Міжнародної науково-технічної конференції “Захист інформації і безпека інформаційних систем”. Львів, Вид-во Львівської політехніки, 2012р. –С.106-107.
31. Мельник В.А. Реалізація в ПЛІС паралельної пам'яті з впорядкованим доступом з попереднім налаштуванням / А.О. Мельник, В.А. Мельник // Матеріали Одинадцятої міжнародної науково-технічної конференції “Вимірювальна та обчислювальна техніка в технологічних процесах” (ВОТТП_11_2012). Хмельницький національний університет, м. Хмельницький, 2012р. – С.68-69.
32. Мельник В.А. Проектування та синтез багатоканального потокового процесора ШПФ / В.А. Мельник // Поступ в науку. Збірник наукових праць міжнародної проблемно-наукової міжгалузевої конференції “Інформаційні проблеми комп'ютерних систем, юриспруденції, енергетики, економіки, моделювання та управління” (ПНМК-2012). Випуск №8. Бучач –2012р. С.125-129.
33. Мельник В.А. Реалізація в ПЛІС паралельної пам'яті з змінним впорядкованим доступом / В.А. Мельник // Збірник тез V Міжнародної науково-технічної конференції «Комп'ютерні системи та мережні технології» (CSNT-2012). – К.: Вид-во Нац. авіац. ун-ту «НАУ-друк», 2012. – С.96.

АНОТАЦІЇ

Мельник Віктор Анатолійович. Самоконфігуровні високопродуктивні комп'ютерні системи. – На правах рукопису.

Дисертація на здобуття наукового ступеня доктора технічних наук за спеціальністю 05.13.05 – комп'ютерні системи та компоненти. – Національний університет “Львівська політехніка” Міністерства освіти і науки України, Львів, 2013.

Дисертація присвячена розробленню теоретичних основ, методів та засобів побудови та організації функціонування нового класу високопродуктивних комп'ютерних засобів – самоконфігуровних комп'ютерних систем і їхніх компонентів, їх дослідженню та реалізації. Розроблено основи організації функціонування та досліджено шляхи вдосконалення архітектури РККС, визначено проблеми, що перешкоджають підвищенню їх ефективності і запропоновано методи їх вирішення. Розроблено теоретичні основи побудови та організації функціонування СККС, в яких враховано запропоновані вдосконалення РККС. Досліджено характеристики тривалості опрацювання інформації в СККС і доведено їх принципові переваги над реконфігуровними. Розроблено принципи структурної організації СККС і концептуальні основи побудови їхніх компонентів. Розроблено нову архітектуру, спосіб опрацювання інформації, базові типи структур та основи проектування паралельних спеціалізованих процесорів для виконання в реконфігуровному середовищі СККС алгоритмів з інваріантною до даних структурою. Розроблено та досліджено методи побудови в СККС засобів генерування програмних моделей спеціалізованих процесорів, структуру та принципи функціонування системи генерування. Здійснено розроблення, дослідження та реалізацію засобів взаємодії компонентів СККС. Запропоновано методологічні основи побудови, принципи структурної організації та функціонування системи віддаленого доступу до реконфігуровних прискорювачів. Проведено реалізацію та експериментальні дослідження СККС і їх базових компонентів.

Ключові слова: високопродуктивна комп'ютерна система, самоконфігуровна комп'ютерна система, реконфігуровна комп'ютерна система, реконфігуровний прискорювач, спеціалізований процесор.

Мельник Виктор Анатольевич. Самоконфигурируемые высокопроизводительные компьютерные системы. – На правах рукописи.

Диссертация на соискание ученой степени доктора технических наук по специальности 05.13.05 – компьютерные системы и компоненты. – Национальный университет "Львовская политехника", Министерства образования и науки Украины, Львов, 2013.

Диссертация посвящена разработке теоретических основ, методов и средств построения и организации функционирования нового класса высокопроизводительных компьютерных средств – самоконфигурируемых компьютерных систем и их компонентов, их исследованию и реализации.

Проведен анализ принципов построения, технологий проектирования и направлений развития высокопроизводительных компьютерных систем. Исследованы основные подходы к построению этих систем, определены

проблемные вопросы повышения их производительности и эффективности, установлена перспективность построения высокопроизводительных компьютерных систем на основе устройств реконфигурируемой логики.

Разработаны основы организации функционирования и исследованы пути совершенствования архитектуры РККС. Проведено исследование и сравнительный анализ типов архитектуры РККС и обоснована целесообразность применения в них реконфигурируемых ускорителей. Проведены исследования средств взаимодействия компонентов РККС слабосвязанной и тесносвязанной типов архитектуры с позиции организации взаимодействия между компьютером и реконфигурируемым ускорителем, оценены характеристики существующих реконфигурируемых ускорителей и проведены экспериментальные исследования характеристик РККС в зависимости от сложности задачи, что позволило определить области целесообразного применения реконфигурируемых ускорителей. Определены проблемы, препятствующие повышению эффективности РККС, и предложены методы их решения.

Разработаны теоретические основы построения и организации функционирования самоконфигурируемых компьютерных систем, в которых учтены предложенные усовершенствования РККС, в частности предложена концепция их построения, метод самоконфигурирования и соответствующий ему способ обработки информации. Исследованы характеристики продолжительности процессов обработки информации в СККС и путем проведения аналитических и экспериментальных исследований доказаны их принципиальные преимущества перед РККС. Разработаны принципы структурной организации СККС и концептуальные основы построения их базовых компонентов – системы распределения вычислительной нагрузки между компьютером и реконфигурируемой логикой, и системы генерирования программных моделей специализированных процессоров. Сформированы требования к остальным компонентам – средствам логического синтеза специализированных процессоров, реконфигурируемой среде и средствам взаимодействия с ней.

Обоснована необходимость и разработана новая архитектура, способ обработки информации и основы проектирования параллельных специализированных процессоров для выполнения в реконфигурируемой среде СККС алгоритмов с инвариантной к данным структурой. Разработаны четыре типа структур таких процессоров, исследованы их технические характеристики и разработана методика расчета продолжительности выполнения в них программы. Сформированы основы проектирования специализированных процессоров для выполнения алгоритмов с инвариантной к данным структурой, что позволило сформировать теоретическую базу для построения систем их автоматического проектирования.

Разработаны и исследованы методы построения в СККС средств генерирования программных моделей специализированных процессоров. По результатам этих исследований разработана структура и принципы функционирования системы генерирования программных моделей специализированных процессоров.

Осуществлены разработка, исследование и реализация средств взаимодействия компонентов СККС. Предложены методологические основы построения, принципы

структурной организации и функционирования системы удаленного доступа к реконфигурируемым ускорителям.

Проведена реализация и экспериментальные исследования СККС и реализованы их базовые компоненты – система распределения вычислительной нагрузки и средства генерирования программных моделей специализированных процессоров на основе библиотеки их компонентов. На примере алгоритма быстрого преобразования Фурье осуществлены проектирование и реализация параллельного специализированного процессора для выполнения алгоритмов с инвариантной к данным структурой, реализовано четыре базовых типа параллельной памяти с упорядоченным доступом для процессоров этой архитектуры, оценены характеристики реализованных устройств.

Ключевые слова: высокопроизводительная компьютерная система, самоконфигурируемая компьютерная система, реконфигурируемая компьютерная система, реконфигурируемый ускоритель, специализированный процессор.

Melnyk Viktor. Self-Configurable High-Performance Computer Systems. – In manuscript.

Thesis for D.Sc degree in the specialty 05.13.05 – computer systems and components. – Lviv Polytechnic National University of the Ministry of Education and Science of Ukraine, Lviv, 2013.

Thesis is dedicated to development of the theoretical foundations, methods and means of design and operation of a new class of high-performance computer facilities – self-configurable computer systems (SCCS) and their components, their study and implementation. Theoretical foundations of operation and ways for improvements in the reconfigurable computer systems architecture are developed and explored. The problems, hindering these systems effectiveness, are identified, and methods to solve them are proposed. Theoretical principles of design and operation of the SCCS, which include the proposed improvements, are developed. Characteristics of information processing duration in the SCCS are investigated and SCCS's fundamental advantages over the reconfigurable ones are proved. Principles of structural organization of the SCCS and conceptual bases of their components design are developed. A new architecture, information processing method, the basic structures and design foundations of the specialized processors for parallel execution in the SCCS's reconfigurable environment of the algorithms, invariant to the data structure, are developed. The methods of construction of the specialized processors VHDL-models generation means for the SCCS, structure and operating principles of the generation system are developed and studied. Design, research and implementation of the software for the SCCS's components interaction are done. Methodological base of the structural organization and functioning of the system of remote access to the reconfigurable accelerators are proposed. An implementation and experimental study of the SCCS and their basic components are performed.

Keywords: high-performance computer system, self-configurable computer system, reconfigurable computer system, reconfigurable accelerator, specialized processor.