

І. А. Клименко¹
М. В. Рудницький¹

КЛАСИФІКАЦІЯ РЕКОНФІГУРОВАНИХ ОБЧИСЛЮВАЛЬНИХ СИСТЕМ

¹Національний технічний університет України «Київський політехнічний інститут»

Запропоновано класифікацію реконфігурованих обчислювальних систем, яка ґрунтується на сучасних тенденціях збільшення продуктивності обчислювальних систем на базі ПЛІС. На підставі запропонованої класифікації визначено поняття динамічної реконфігурації, як найперспективніша технологія проектування систем на ПЛІС. Розглянуто особливості методології часткової динамічної реконфігурації та виконано порівняльний аналіз технологічних досягнень провідних виробників ПЛІС.

Ключові слова: реконфігурація, часткова динамічна реконфігурація, ПЛІС, реконфігуровані обчислювальні системи, FPGA, Reconfigurable Computer, Reconfigurable System on a Chip, Partial Dynamic Reconfiguration.

Вступ

Ідеї, що лежать в основі принципів реконфігурованих обчислювальних машин з'явилися ще в 60-х роках [1, 2]. Але значущі результати їх практичного використання стали можливими лише за створення необхідного технологічного фундаменту — на початку 2000 років технологічний процес виготовлення мікросхем дозволив значно підвищити їх швидкість та степінь інтеграції. Разом з тим, вже відомі на той час мікросхеми класу програмованих логічних інтегральних схем (ПЛІС, Field Programmable Gate Array, FPGA) отримали новий потенціал щодо продуктивності та складності розроблюваних пристроїв.

Основна особливість ПЛІС — можливість багаторазового перепрограмування. Це покладено в основу створення цифрових пристроїв і навіть потужних обчислювальних систем, архітектура яких має властивість реконфігурації в залежності від визначених умов з метою збільшення продуктивності обчислень. Такі системи належать до класу реконфігурованих обчислювальних систем (РОС, Reconfigurable Computer, RC), основна концепція їх створення — забезпечення адаптивності архітектури до класів вирішуваних задач [3, 4, 5].

Разом із надзвичайною привабливістю перепрограмування, РОС мають певну низку проблем, пошук вирішення яких обумовлює велику актуальність досліджень у цьому напрямку. До таких проблем у першу чергу належать великі накладні видатки на фізичне здійснення реконфігурації — часові затрати, непродуктивні витрати пам'яті та каналів зв'язку, енергозатрати, тощо. Як засіб для подолання багатьох з означених проблем застосовують часткову реконфігурацію (ЧР), що полягає у зміні лише частини пристрою в той час, коли інша частина продовжує працювати. Час виконання кожної задачі складається з часу реконфігурації та часу обчислення. При цьому час реконфігурації пропорційний розміру реконфігурованої області [6]. Ця технологія дозволяє створювати ефективні, з точки зору часу реконфігурації, споживання логічних ресурсів та електроенергії, проекти, що висвітлено в літературі дослідниками як нашої країни [7—10], так і інших країн світу [4, 5, 11—13]. Актуальною сферою застосування ЧР є системи, до яких висунуто високі вимоги надійності [14]. Однією з перспективних тенденцій розвитку РОС є мультизадачність, саме часткова реконфігурація дозволяє збільшити ефективність функціонування РОС в цьому режимі [15, 16].

Мультизадачний режим функціонування РОС вводить в область застосування такі поняття як динамічна реконфігурація, що базується саме на застосуванні часткової реконфігурації [16]. Така реконфігурація дозволяє відображення множини задач на той самий пристрій динамічно, визначаючи залежності відображення під час виконання, і має назву часткової динамічної реконфігурації. Це є іншою популярною галуззю сучасних досліджень, що пов'язана з динамічним плануванням та відображенням множини задач на програмовані ресурси РОС [6, 15, 16].

В роботі наводиться огляд наявних на сьогодні методологій проектування, апаратних та програмних рішень від провідних виробників ПЛІС, на базі сучасних тенденцій розробки та функціонування РОС розроблена класифікація РОС, на основі якої визначено поняття динамічної реконфігурації. На підставі цього обґрунтовані основні напрямки досліджень та розробки РОС згідно з актуальною проблемою сьогодення — досягнення високої продуктивності обчислень.

Огляд основних сімейств ПЛІС з підтримкою часткової динамічної реконфігурації

Серед мікросхем, що підтримують технологію часткової динамічної реконфігурації сьогодні, вирізняються чіпи двох компаній, що займають лідируючі позиції на ринку ПЛІС, Xilinx та Altera. Незважаючи на технічну різноманітність і широку розповсюдженість мікросхем компанії Altera, всі вони традиційно не підтримують часткову динамічну реконфігурацію [17, 18]. Ситуація змінилась лише з виходом останніх модифікацій мікросхем [19, 20], але висока ціна на пристрої і мізерний досвід реалізації ЧР поки що віддають першість засобам компанії Xilinx. Всі ПЛІС компанії Xilinx мають можливість часткової реконфігурації [21, 22]. Проте, варто зазначити, що з часом значно змінилась методологія їх реконфігурації (табл.). Щодо інструментальних засобів розробки, окремо варто зазначити, що безпосередня підтримка часткової динамічної реконфігурації з'явилась в програмному пакеті Xilinx ISE Design Suite, починаючи з версії 12 [23, 24].

Підтримка часткової динамічної реконфігурації в ПЛІС компанії Xilinx

ТИП РЕКОНФІГУРАЦІЇ	РІК	СІМЕЙСТВО ПЛІС	ПІДТРИМКА САПР
Перше покоління ПЛІС підтримує часткову реконфігурацію (знято з виробництва)	1996	XC6200	—
Друге покоління ПЛІС — реконфігурація стовпцями	1999	Virtex, VirtexII, Virtex II Pro, Virtex E	Xilinx ISE Design Suite 10.X (2008 рік) 11.X (2009 рік)
Сімейства невисокої ціни без офіційної підтримки часткової реконфігурації. Проте, групами ентузіастів було успішно перевірено її можливість на деяких пристроях	2003	Spartan 3, Spartan 3E, Spartan 3A	Xilinx ISE Design Suite 10.X (2008 рік) — інтегрований Plan Ahead
Третє покоління ПЛІС, підтримує часткову динамічну реконфігурацію фреймами, розміром 16-20 CLB, розпочинаючи з 2010 — розділами	2006	Virtex 4	Xilinx ISE Design Suite 10.1 (2008 рік) — інтегрований Plan Ahead; 11.X (2009 рік) — інтегрований Plan Ahead; 12.X (2010 рік) — автономний Plan Ahead
	2006	Virtex 5	
Сімейство четвертого покоління, підтримує часткову динамічну реконфігурацію фреймами, розміром 40 CLB, розпочинаючи з 2010 — розділами	2009—2010	Virtex 6, Spartan 6	Xilinx ISE Design Suite 11.3 (2009 рік) — інтегрований Plan Ahead, підтримка порту ICAP; 12.X (2010 рік) — вбудована підтримка часткової реконфігурації (розділами); підтримка автономної версії Plan Ahead (модульна); 13.X (2011 рік)
Сучасні сімейства підтримують часткову динамічну реконфігурацію на базі розділів	2011—2013	Virtex 7, Artix 7, Kintex 7, Zynq 7000	Xilinx ISE Design Suite 13.X (2011 рік) 14.X (2013 рік)

Надалі розглянемо основні особливості методології часткової динамічної реконфігурації компанії Xilinx, що реалізована на базі широко використовуваних розробниками стандартних засобів проектування цифрових пристроїв на ПЛІС.

Основна проблема реалізації ЧДР — технологічні можливості і обмеження мікросхем та інструментальних засобів розробки. Методологія здійснення ЧДР, в першу чергу, потребує розділення ПЛІС на статичні та динамічні області. Найрозповсюдженіший метод — розбиття прогрованої площини ПЛІС на основну статичну область (базову), де розміщуються статичні компоненти, які завантажуються в мікросхему один раз під час ініціалізації пристрою і не змінюються протягом роботи системи, та динамічну область (реконфігуровану), де розміщуються всі динамічні компоненти [11]. ПЛІС може мати одну або декілька областей часткової реконфігурації, в той же час кожна така область може складатися з декількох вузлів або модулів, що мають здатність рекон-

фігурації різного ступеня деталізації, аж до найменшої одиниці — логічного блока ПЛІС.

В зв'язку з цим інше важливе питання, що зумовлює особливості різних методологій ЧДР, це саме питання ступеня деталізації реконфігурації, яке напряму пов'язане з сімействами, типами та технологічними можливостями мікросхем ПЛІС. Відповідно до цього, розглядають локальну та модульну реконфігурацію [8]. Локальна реконфігурація — заміна одного конфігураційного логічного блоку [7]. Модульна реконфігурація — заміна частини ПЛІС, використовуючи окремо створені апаратні модулі, що додаються або замінюються під час реконфігурації. Частину ПЛІС, що змінюється, називають реконфігурованим модулем.

Для пояснення методології часткової реконфігурації ПЛІС необхідно, перш за все, звернутись до їх внутрішньої структури. Для конфігурації ПЛІС створюється конфігураційний бітовий потік, який на прикладі сімейства Xilinx Virtex II складається із заголовку й конфігураційної інформації [18]. Остання має ієрархічну структуру — на першому рівні складається з трьох блоків, далі кожен блок поділяється на стовпці, що конфігурують множину базових компонентів ПЛІС. Наприклад, стовпці типу IOI — конфігурують блоки регістрів, мультиплексорів та буфери з трьома станами лівого та правого боків пристроїв; CLB (англ. — Configurable Logic Block) — конфігурують логічні блоки та блоки вводу/виводу верхнього та нижнього боків пристрою; BRAM та BRAM_INT — конфігурують частини, визначені користувачем в BlockRAM [23]. Структура бітового потоку зображена на рис. 1. Можна розглядати повний бітовий потік, що створюється для конфігурації всієї мікросхеми, та частковий бітовий потік, що відповідає тільки тій області мікросхеми, що змінюється під час реконфігурації — часткової реконфігурації.

Тип стовбця	GCLK	IOB	IOI	CLB	CLB	...	CLB	IOI	IOB	...	BRAM	...	BRAM_INT
Кількість фреймів	4	4	22	22	22	...	22	22	4	...	64	...	22

Рис. 1. Структура бітового потоку

Одна зі сторін задачі часткової реконфігурації зводиться до виділення площини ПЛІС для розміщення реконфігурованих модулів, та розведення зв'язків між модулями системи. Для реалізації цього розглядають декілька способів. Один з них має назву реконфігурація *стовпцями*. Кожен стовпець (рис. 2), реалізує певну кількість вертикальних фреймів, висота яких дорівнює висоті пристрою, а ширина одному логічному блоку CLB. Для Virtex II і Virtex II Pro Xilinx мінімальна область, що може бути реконфігурована, відповідає повному стовпцю ПЛІС [23]. Цільова область реконфігурації складається з певної кількості стовпців і розміщується на всю висоту пристрою. Частковий бітовий потік включає тільки ті конфігураційні стовпці, що мають бути змінені. Виходячи з цього, час затрачений на часткову реконфігурацію, буде менший часу завантаження більшого за розміром повного бітового потоку. При цьому на розмір часткового бітового потоку звичайно впливає тип пристрою, а саме його розмір. Наприклад, розмір конфігураційного фрейму мікросхеми Virtex II XC2V8000 складає 286 біт, а Virtex II XC2V40 — 26 біт. Виходячи з цього, час реконфігурації, пропорційний довжині бітового потоку, та кратний висоті стовпця мікросхеми, визначає один з недоліків розглянутої технології. Серед інших недоліків: непродуктивні витрати ресурсів ПЛІС та неефективне розміщення внутрішніх зв'язків між модулями зі значною розбіжністю розмірів реконфігурованої області і завантажуваного модуля.

Під час створення мікросхем сімейств, розпочинаючи з Virtex 4, з метою подолання недоліків реконфігурації стовпцями, з'явилася нова технологія реконфігурації, що отримала назву блочної або модульної реконфігурації [11]. Організація бітового потоку також базується на фреймах, але їх структура тепер не залежить від висоти пристрою, а має стандартний розмір — 16 логічних блоків CLB в висоту і 1 CLB в ширину [18, 25]. Під час розводки ПЛІС та створення бітової послідовності відповідним програмним забезпеченням можна вибрати, який фізичний розмір на ПЛІС займатиме реконфігурований модуль. Відповідна реконфігурована область, таким чином, буде будь-якого прямокутного розміру, але кратного одному конфігураційному CLB-фрейму. В подальших сімействах простежується тенденція до зростання кількості логічних блоків у фреймі — 20 логічних блоків CLB у висоту і 1 CLB у ширину в мікросхемі Virtex 5, 40 CLB блоків у висоту — Virtex 6. Завдяки цьому також зменшується кількість фреймів для реконфігурації, а, отже, зменшується розмір бітового потоку і час здійснення реконфігурації [23].

Найгнучкіший спосіб розміщення реконфігурованих модулів на площині ПЛІС передбачає, що

кожен апаратний модуль реалізується прямокутними групами вузлів і розміри його не обмежені задалегідь зумовленими розмірами частково реконфігурованої області [8]. Під час генерування модуля, площу та співвідношення сторін можна оптимізувати відповідно до внутрішніх зв'язків модуля. Саме таку методологію пропонує так звана підтримка часткової динамічної реконфігурації в сучасних сімействах мікросхем ПЛІС (Virtex 6, Virtex 7) реконфігурація *розділами* (Partitions) [7, 26]. Вона підтримується й на рівні інструментальних засобів, що вирішує проблему забезпечення однорідності вузлів та встановлення зв'язків між ними. Таке розміщення забезпечує найбільш ефективну реалізацію з точки зору використання ресурсів, розведення зв'язків, розміру бітової послідовності, часу реконфігурації.

Мікросхеми Virtex 4, Virtex 5 підтримують як модульний підхід, так і розподіл розділами, засобами спеціального програмного забезпечення (Xilinx ISE Design Suite 12.X з 2010 року). Сучасні пристрої Virtex 6, Virtex 7 засобами ISE 13.X 2011 року та 14.X 2013 року підтримують найсучасніші технології ЧДР, що на підставі своєї затребуваності активно розвиваються виробниками ПЛІС (див. табл.).

Розпочинаючи з версії ISE 10.1 2008 року в системі автоматизації розробки інтегрований потужний пакет для проектування цифрових пристроїв на ПЛІС Plan Ahead 10.1. Версія пакету підтримує актуальні на той час ПЛІС — Virtex 4, Virtex 5. Програма надає широкі можливості для проектування частково реконфігурованих пристроїв за модульним принципом, крім того ця можливість залишається і в сучасних версіях (за автономної підтримки), в той час коли подальші версії САПР ISE модульне проектування не підтримують [7].

Особливості проектування реконфігурованої системи на базі мікросхеми Virtex 4 SX35 у складі плати ML402 наведено в роботі [27]. У вихідному стані базова конфігурація завантажена в ПЛІС через порт JTAG засобами персонального комп'ютера. Часткові бітові потоки завантажені в спеціально відведені для цього області зовнішньої флеш-пам'яті. Часткова реконфігурація ініціюється логікою управління, реалізованою в ПЛІС. Фізичне управління процесом реконфігурації здійснює керувальний контролер у складі реконфігурованої системи. Сигнал початку реконфігурації передається з боку ПЛІС до керувального контролера, разом з чим передається інша інформація керування, зокрема ідентифікатор визначення бітового потоку, який має бути отримано з флеш-пам'яті. Контролер управляє процесом формування адрес для вибірки конфігураційних даних з флеш-пам'яті, встановлює сигнали дозволу прийому бітів конфігурації та фіксації цих бітів в порту конфігурації ПЛІС. У такий спосіб часткові бітові потоки завантажуються через порт JTAG за внутрішньосистемного управління, зокрема процесора Power PC на борту платформи ML402, але для цього може бути використано і зовнішній персональний комп'ютер. Ця технологія також актуальна для всіх інших сімейств мікросхем.

Аналогічна методологія проектування та склад компонентів системи зберігається за використання подальших сімейств мікросхем ПЛІС. В роботі [28] розглянута реконфігурована система на платформі ML507 з ПЛІС Virtex 5 FX70 на борту. У складі ПЛІС виділені статична та реконфігурована області, а також область управління, що здійснює фізичне управління реконфігурацією, зокрема вибір часткової конфігурації з внутрішньої пам'яті і безпосередньо передавання її на модуль ICAP.

Модуль ICAP (Internal Configuration Access Port) — це апаратний засіб ПЛІС, який дає можливість динамічно завантажувати в пам'ять потік бітів [29] без застосування порту JTAG. При цьому для реконфігурації не потрібно використовувати зовнішній пристрій і вся необхідна конфігураційна інформація зберігається на тій же платі, що зменшує час передачі даних [7]. Недоліком є низька оперативність проведення модифікації алгоритму роботи реконфігурованих модулів, оскільки для цього необхідно підключати ПК зі спеціалізованим програмним забезпеченням та завантажувати бітову послідовність через порт JTAG [7]. Пристрої Xilinx, такі як Virtex II, Virtex 4, Virtex 5, підтримують технологію ICAP [29]. Для реалізації ЧДР в сучасних сімействах мікросхем порт ICAP — невід'ємна технологія [26].

Для зберігання часткових послідовностей використовується внутрішня пам'ять ПЛІС ціною обмеження на розмір і кількість послідовностей, котрі можуть бути збережені. Але така реалізація забезпечує більшу швидкість вибірки даних, і простіший інтерфейс, ніж у зовнішньої пам'яті. Часткові конфігурації можна зберігати у зовнішній енергонезалежній пам'яті. Якщо враховувати співвідношення внутрішньої площі кристала до ціни, то кращим виходом буде зберігати неактивні бітові послідовності у зовнішній енергонезалежній пам'яті [28].

У роботі [13] розглянута реалізація системи, що динамічно реконфігурується, для обробки ве-

ликих масивів даних. Під час дослідження, авторами були зроблені важливі висновки з погляду оцінки площі реконфігурованої області та завантажуваних модулів. Мікросхема повинна мати необхідну кількість вбудованих функціональних блоків у виділеній області реконфігурації для реалізації необхідних модулів. За відсутності таких пристроїв вони можуть бути реалізовані засобами логічних ресурсів мікросхеми, використовуючи більшу кількість логічних блоків CLB блоків та LUT, але це збільшить розмір області, що частково реконфігурується. У випадку, коли площини модулів, що завантажуються в одну і ту саму область, повністю різні, часткова реконфігурація неефективна через значні втрати CLB, що приводить до значного підвищення вартості реалізації. До того ж, якщо області, що частково реконфігурується, дуже великі, час на реконфігурацію можна порівняти з часом повної конфігурації мікросхеми, що нівелює саму ідею використання часткової динамічної реконфігурації.

У роботі [28] виконаний ряд експериментів з метою оцінки часу виконання часткової реконфігурації. Проведено реконфігурування ПЛІС сімейства Virtex 5 з максимальною швидкістю 3,2 Гбіт/с. Експерименти проводилися з різними розмірами реконфігурованих модулів. При цьому встановлено, що мінімальний час реконфігурації для ПЛІС сімейства Virtex 5 дорівнює 0,016 мс для найменшого модуля, котрий займає 0,36 % ПЛІС і складається з 160 логічних елементів. Час реконфігурації великого модуля, який займає 10 % логічних елементів ПЛІС, становить 0,42 мс.

Класифікація реконфігурованих обчислювальних систем

В роботі [9] пропонується класифікація РОС, що всебічно відображає ситуацію на момент її створення — 2006 рік, але вимагає розширення та вдосконалення згідно з сучасними технічними можливостями мікросхем ПЛІС і методами проектування РОС. Таким чином, метою запропонованої нами класифікації є сучасна систематизація РОС, обґрунтування актуальних напрямків їх розробки в контексті проблеми інтенсивного збільшення продуктивності суперкомп'ютерних, кластерних та спеціалізованих архітектур [3—5], зокрема визначення поняття динамічної реконфігурації, як одного з перспективних засобів. Відповідно до цілей статті, автори детально розглядають фізичний аспект здійснення реконфігурації. Класифікація РОС наведена на рис. 1.

За часом виконання фактором класифікації є стан системи під час реконфігурації, при цьому розрізняють *реконфігурацію часу зупинки* (РЧЗ, Shutdown Reconfiguration, SD) та *реконфігурацію часу виконання* (РЧВ, Run Time Reconfiguration, RT) (рис. 1). Відповідно до назви, реконфігурація виконується або за зупинки системи або під час її роботи. Особливостями першого способу є зупинка системи та необхідність наявності зовнішніх засобів здійснення реконфігурації. Цей спосіб є значно дешевшим та простішим за реалізацію. Другий спосіб реалізується за збереження працездатності системи під час реконфігурації та реалізації алгоритмів реконфігурації як частини системи. Під системою в даному контексті розуміють реконфігуровану частину та пов'язане з нею апаратне оточення.

В інших джерелах [8, 12] поняття *Run Time* реконфігурації визначається як *динамічна реконфігурація* (Dynamic Reconfiguration, DR). Але для визначення поняття динамічної реконфігурації першочерговим є аспект її управління. Відповідно до класифікації [9] управляючий вплив на реконфігурацію може відбуватися як з боку зовнішніх, так і з боку внутрішніх алгоритмів управління. Важливим параметром є джерело запиту на реконфігурацію, що ініціює саме процедуру реконфігурації. Запит так само може бути зовнішнім, спричиненим, наприклад, користувачем системи, або внутрішнім, ініційованим алгоритмом управління. Щодо цього, існують поняття *пасивної* реконфігурації, коли реконфігурація системи відбувається лише згідно із зовнішніми запитами з використанням зовнішнього алгоритму управління, та *активної* реконфігурації, за якої наявна певна підсистема управління, що може ініціювати реконфігурацію незалежно від зовнішніх факторів [9]. Остання визначає основний фактор динамічності процесу реконфігурування — незалежність від зовнішніх запитів, у той час як статична реконфігурація є пасивною. Відповідно до цього визначимо поняття динамічної та статичної реконфігурації.

Статична реконфігурація характеризується виконанням під дією зовнішніх алгоритмів управління та ініціалізацією запитом ззовні. Така реконфігурація за способом управління належить до пасивного типу (рис. 2), є повністю суб'єктивним процесом, відбувається на вимогу користувача системи та зазвичай застосовується для рішення певної користувацької задачі засобами даної РОС.

Джерела [12, 17] описують *статичну реконфігурацію* (Static Reconfiguration, SR) або *реконфігурацію часу компіляції* (Compile Time Reconfiguration, CTR), за якої конфігурація завантажувалась

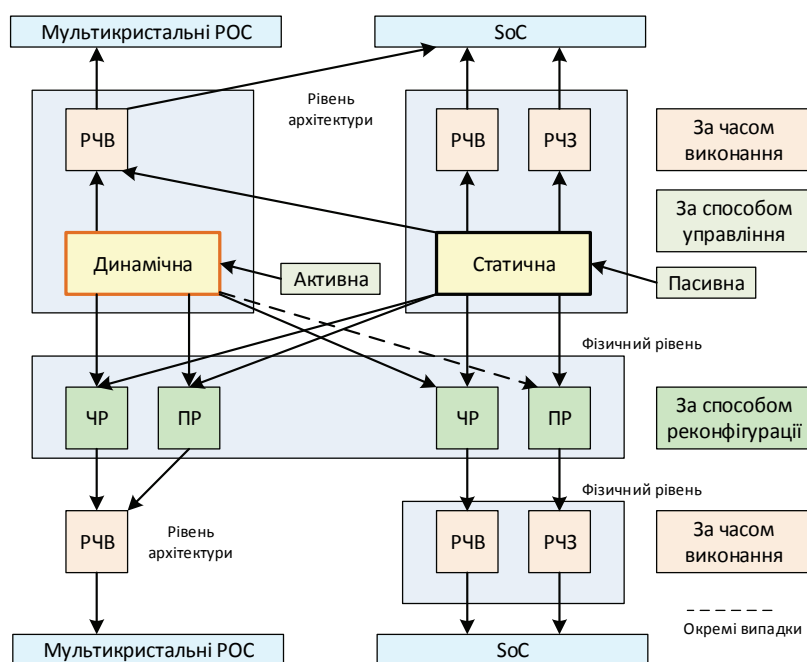


Рис. 2. Класифікація реконфігурованих обчислювальних систем

сучасні суперкомп'ютери на ПЛІС. В якості обґрунтування цього, варто врахувати деталізацію рівня на якому розглядається РОС, та зазначити, що параметри часу зупинки та часу виконання ми розглядаємо на рівні архітектури системи, який для SoC відповідає фізичному рівню, згідно їх визначення. Тому зупинка РОС для SoC відповідає зупинці реконфігурованого фізичного модуля і пов'язаного з ним апаратного оточення. На відміну від мультикристальних РОС, де реконфігурація часу зупинки не розглядається взагалі. Відповідно до місця розташування управляючих алгоритмів та джерел запиту на реконфігурацію, статична реконфігурація в цьому класі систем відбувається у режимі часу виконання. Наприклад, реконфігуровані суперкомп'ютери працюють в мультипрограмному режимі і надають користувачам обчислювальні площини для реалізації їх проєктів — реконфігурація системи відбувається за вимогою користувача під управлінням зовнішніх засобів — статична реконфігурація, але цілком система залишається працездатною під час реконфігурації окремих модулів [3].

Динамічна реконфігурація характеризується динамічним або автоматичним режимом здійснення, незалежно від будь-яких зовнішніх впливів. Ініціалізація, так само як і сам процес реконфігурації, відбувається під час роботи системи під дією алгоритмів управління, що є частиною системи. Лише внутрішні фактори, що породжуються станом системи або логікою управління визначають процес динамічної реконфігурації (рис. 1).

Надалі за способом виконання реконфігурації розрізняють *повну* та *часткову реконфігурацію*. Повна реконфігурація (ПР, Full Reconfiguration, FR) — це реконфігурація одного чи кількох фізичних модулів системи у повному об'ємі. Часткова реконфігурація (ЧР, Partial Reconfiguration, PR) — це процес реконфігурації частини фізичного модуля системи. Якщо перший спосіб відбувається за повної зупинки роботи фізичного модуля на час виконання реконфігурації, то у другому випадку модуль зможе виконувати певні функції, часткова реконфігурація відбувається в режимі часу виконання (рис. 1) [9, 26].

Повна реконфігурація за застосування в SoC відповідає статичній реконфігурації. Як окремий випадок повної динамічної реконфігурації в роботах [8, 31] розглядають *реконфігурацію реального часу* (Run Time Reconfiguration, RTR) — з розширенням та вдосконаленням технології ПЛІС стає можливо зменшити час, необхідний на виконання реконфігурації, що дозволяє виконувати повну реконфігурацію між різними стадіями чи режимами роботи ПЛІС. В мультикристальних РОС як повна, так і часткова реконфігурація виконується як в статичному, так і в динамічному режимах.

Найкращі показники ефективності часткова реконфігурація надає в динамічному режимі, настільки, що поняття *часткової динамічної реконфігурації* (ЧДР, Dynamic Partial Reconfiguration, DPR) відповідає сучасній технології в галузі розробки РОС. Багато дослідників взагалі ототожнюють ці два поняття та вважають, що одне цілком визначає інше [15, 16]. Дослідження та застосування

в ПЛІС лише раз під час ввімкнення, і весь цикл роботи додатку не змінювалась. Тобто, фактично, реконфігурація не використовувалась. Це доволі однобокий погляд на це поняття, скажемо так, що одноразове завантаження конфігурації системи є окремим випадком статичної реконфігурації.

Натомість деякі автори [30] ототожнюють статичну реконфігурацію з поняттям реконфігурації часу зупинки. Це частково справедливо лише для РОС класу система-на-кристалі (System on a Chip, SoC). Але, вдосконалюючи класифікацію [9] згідно з сучасними реалізаціями РОС та галузями застосування, необхідно врахувати клас мультикристальних РОС, до яких належать і

ЧДР є актуальним напрямком, що відображено у багатьох роботах останніх років [23, 32, 33, 34].

Варто детально зупинитися на таких нащадках частково реконфігурованих систем — системи, що *самореконфігурується* (Self Reconfiguration, SR) [26]. Вони можуть змінювати власне апаратне забезпечення в динамічному режимі під дією внутрішніх алгоритмів управління, що надає їм надвисокої ефективності, гнучкості та автономності [18, 30]. Такі системи часто використовуються в співпроцесорах, збудованих на технології реконфігурації. Іншими прикладами їх використання є комп'ютерні мережі [35], обробка зображень [33], шифрувальні алгоритми [30].

Висновки

З огляду на проведений аналіз літературних джерел за тематикою досліджень часткової динамічної реконфігурації та на підставі розробленої класифікації РОС можна зробити такі висновки:

1. В реконфігурованих обчислювальних системах найбільш ефективно використання динамічної реконфігурації. Це дозволить перебудовувати архітектуру системи автономно, в динамічному режимі. Це забезпечує, з одного боку, підвищення продуктивності реконфігурованих обчислювальних систем взагалі, а з іншого боку, приведення реконфігурованих обчислювальних систем до класу обчислювальних систем загального використання з досягненням високої реальної продуктивності обчислень для вирішення широкого класу задач.

2. Серед методів динамічної реконфігурації найбільш доцільно використання часткової динамічної реконфігурації. Це дозволяє збільшити розміри завантаженого пристрою та зменшити накладні витрати на виконання реконфігурації, що критично негативно впливають на продуктивність реконфігурованих систем. До накладних видатків належать: час реконфігурації, непродуктивні дані обміну, область пам'яті для збереження конфігурації пристрою, енерговитрати.

3. Актуальність та доцільність часткової динамічної реконфігурації обґрунтувала втілення цих технологій у виробництво сучасних ПЛІС. Сьогодні всі ПЛІС провідних компаній виробників підтримують технологію часткової динамічної реконфігурації, як на рівні апаратних засобів, так і за інструментальної підтримки.

4. Для реалізації найбільш продуктивних проектів доцільно використовувати сучасні пристрої, які підтримують ефективні технології проектування, з підтримкою часткової динамічної реконфігурації, та забезпечують необхідну інструментальну підтримку для спрощення проектування. Окрім того, політика компаній виробників щодо оновлення своїх продуктів, дотримується принципів ієрархічної спадковості [7], що призводить, в першу чергу, до вилучення застарілих рішень в нових апаратних та програмних продуктах. Серед актуальних на сьогодні мікросхем доцільно використання пристроїв сімейств Virtex 5 — Virtex 7 компанії Xilinx, сучасних пристроїв компанії Altera, розпочинаючи з 2010 року виробництва, зокрема Stratix V, Cyclone V. Для широкого кола дослідників серед мікросхем невисокої ціни та найбільш розповсюджених можливо використання пристроїв сімейства Xilinx Virtex 4, яке підтримує як модульну, так і динамічну часткову реконфігурацію [26], засобами актуального на сьогодні програмного забезпечення Xilinx ISE Design Suite 12.X та з підтримкою інтерфейсу ICAP [29].

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Estrin G. Organization of computer system: the fixed plus variable structure computer / G. Estrin. // Western Joint IRE-AIEE-ACM Computer Conference (New York, May 3—5, 1960). — New York, 1960. — P. 33—40.
2. Каляев А. В. Теория цифровых интегрирующих машин и структур / А. В. Каляев. — М. : Советское радио, 1970. — 472 с.
3. Baxter R. Maxwell — a 64 FPGA Supercomputer / R. Baxter, S. Booth, M. Bull, A. Trew [and all]. // Adaptive Hardware and Systems, 2007 (AHS 2007). (Edinburgh, August 5—8, 2007). — IEEE Computer Society, 2007. — P. 287—294.
4. Каляев И. А. Высокопроизводительные реконфигурируемые вычислительные системы на основе плив Virtex-6 и Virtex-7 / И. А. Каляев, А. И. Дордопуло, И. И. Левин, Е. А. Семерников // Параллельные вычислительные технологии (ПаВТ'2012) (Новосибирск, 26—30 марта 2012 г.). — Челябинск : Издательский центр ЮУрГУ, 2012. — С. 449—458.
5. Каляев И. А. Реконфигурируемые мультимедийные вычислительные структуры / И. А. Каляев, И. И. Левин, Е. А. Семерников, В. И. Шмойлов. — М. : ЮНЦ РАН, 2008. — 420 с.
6. Guan N. Schedulability analysis of preemptive and nonpreemptive EDF on partial runtime-reconfigurable FPGAs / N. Guan, Q. Deng, Z. Gu, W. Xu [and all]. // ACM Transactions on Design Automation of Electronic Systems. — New York, ACM, 2008 — Volume 13, No. 4, Article 56.
7. Дунець Р. Б. Проблеми побудови частково реконфігурованих систем на ПЛІС / Р. Б. Дунець, Д. Я. Тиханський // Радіоелектронні і комп'ютерні системи, 2010. — № 7 (48). — С. 200—204.
8. Дунець Р. Б. Дослідження часткової реконфігурації ПЛІС / Р. Б. Дунець, Д. Я. Тиханський // Радіоелектронні і комп'ютерні системи, 2009. — № 6 (40). — С. 240—244.

9. Баркалов А. А. Классификация реконфигурируемых вычислительных систем / А. А. Баркалов, Р. В. Мальчева, А. А. Гриценко // III научно-практическая конференция «Донбасс 2020: Наука и Техника — Производству» (Донецк, 3—4 февраля 2006). — Донецк : ДонНТУ Министерства образования и науки, 2006. — С. 473—478.
10. Палагин А. В. Реконфигурируемые вычислительные системы: Основы и приложения / А. В. Палагин, В. Н. Опанасенко. — К. : Прогресс, 2006. — 280 с.
11. Lie W. Dynamic partial reconfiguration in FPGAs / W. Lie, Wu Feng-yan. // Third International Symposium on Intelligent Information Technology Application (IITA 09) (NanChang, China, November 21—22, 2009). — IEEE Computer Society Washington, 2009. — Volume 2. — P. 445—448.
12. Azarian A. Reconfigurable computing architecture survey and introduction / A. Azarian, M. Ahmadi // Computer Science and Information Technology, 2009 (ICCSIT 2009). (2nd IEEE International Conference, August 8 — 11, 2009). — IEEE Computer Society, 2009. — P. 269—274.
13. Hussain H. M. Novel Dynamic Partial Reconfiguration Implementation of K-Means Clustering on FPGAs: Comparative Results with GPPs and GPUs / H. M. Hussain, K. Benkrid, A. Ebrahim, A. T. Erdogan [and all]. // International Journal of Reconfigurable Computing. — Hindawi Publishing Corporation, 2012. — Volume 2012, Article ID 135926, 15 pages.
14. Saldana M. Using Partial Reconfiguration and Message Passing to Enable FPGA-Based Generic Computing Platforms / M. Saldana, A. Patel, H. J. Liu, P. Chow // International Journal of Reconfigurable Computing. — Hindawi Publishing Corporation, 2012. — Volume 2012, January 2012. Article No. 3.
15. Bassiri M. Mitigating reconfiguration overhead in on-line task scheduling for reconfigurable computing systems / M. M. Bassiri, S. H. Shahhoseini // Computer Engineering and Technology (ICCTET). (2nd International Conference, Jodhpur, India, 16—18 April 2010). — IEEE Computer Society, 2010. — Volume 4. — P. 4—402.
16. Wang X. Migration between Software and Hardware Task on Preemptive Multitasking CPU/FPGA Hybrid Architecture / X. Wang, D. Feng, C. Tian-zhou, Hu Tong-sen // High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems (HPCC-ICCESS). (Liverpool, 25—27 June 2012). — IEEE Computer Society, 2012. — P. 1329—1336.
17. Fons Lluís F. Embedded electronic systems driven by run-time reconfigurable hardware: дис. д-ра техн. наук : 21.03.2012 / F. Fons Lluís; Universitat Rovira I Virgili. — Tarragona, 2012. — 242 p.
18. Upegui Posada A. Dynamically reconfigurable bio-inspired hardware: Doctor of Science in Electronics Thesis : 13.10.2006 / A. E. Upegui Posada; École Polytechnique Federale De Lausanne. — 2006. — 187 p.
19. Stratix V Partial Reconfiguration [Електронний ресурс]. — Altera wiki, 2014. — Режим доступу: http://www.alterawiki.com/wiki/Stratix_V_Partial_Reconfiguration.
20. About Partial Reconfiguration [Електронний ресурс]. — Altera Quartus II Help, 2014. — Режим доступу: http://quartushelp.altera.com/13.0/master.htm#mergedProjects/comp/comp/comp_about_part_reconfig.htm?GSA_pos=1&WT.os_s_r=1&WT.oss=partial reconfiguration
21. Gonzalez I. Dynamically Reconfigurable Coprocessors in FPGA-based Embedded Systems: Doctor of Science in Electronics Thesis : 21.03.2006 / I. Gonzalez; Universidad Autónoma de Madrid. — Madrid, 2006. — 56 p.
22. Schleupen K. Dynamic Partial FPGA Reconfiguration in a Prototype Microprocessor System / K. Schleupen, S. Lelaich, R. Mannion, G. Zhi [and all]. // Field Programmable Logic and Applications, 2007. (FPL 2007). (Amsterdam, 27—29 August 2007). — IEEE Computer Society, 2007. — P. 533—536.
23. Paiz Gatica C. V. Dynamically Reconfigurable Hardware for Embedded Control Systems: Doctor of Science in Electronics Thesis : 21.12.2012 / C. V. Paiz Gatica; Universität Bielefeld. — Bielefeld, 2012. — 220 p.
24. What's New in Xilinx ISE Design Suite 12 [Електронний ресурс]. — Xilinx, 2014. — Режим доступу: http://www.xilinx.com/support/documentation/sw_manuals/xilinx12_3/whatsnew.htm.
25. Sverre H. Framework for self reconfigurable system on a Xilinx FPGA: Master of Science in Electronics Thesis : 11.06.09 / H. Sverre; Norwegian University of Science and Technology. — Trondheim, 2009. — 59 p.
26. Partial Reconfiguration in the ISE Design Suite [Електронний ресурс]. — Xilinx, 2014. — Режим доступу: <http://www.xilinx.com/tools/partial-reconfiguration.htm>.
27. Birla M. Partial run-time reconfiguration of FPGA for computer vision applications / M. Birla, K. N. Vikram. // Parallel and Distributed Processing, 2008 (IPDPS 2008). (Miami, April 14—18, 2008). — IEEE Computer Society, 2008. — P. 1—6.
28. Тиханський Д. Я. Метод підвищення ефективності частково реконфігурованої системи / Д. Я. Тиханський, Р. Б. Дунець, Р. В. Грица // Комп'ютерні системи та мережі: збірник наукових праць / Міністерство освіти і науки, молоді та спорту України, Національний університет «Львівська політехніка»; відповідальний редактор А. О. Мельник. — Львів : Видавництво Львівської політехніки, 2011. — С. 192—198.
29. Ming L. Run-time Partial Reconfiguration speed investigation and architectural design space exploration / L. Ming, W. Kuehn, L. Zhonghai, A. Jantsch. // Field Programmable Logic and Applications, 2009. (FPL 2009). (Prague, August 31—September 2, 2009). — IEEE Computer Society, 2009. — P. 498—502.
30. Abidine El Z. Self-Partial and Dynamic Reconfiguration Implementation for AES using FPGA / Z. El Abidine, A. Ismaili, A. Moussa // IJCSI International Journal of Computer Science Issues — 2009. — Volume 2. — P. 33—40.
31. Danne K. Periodic real-time scheduling for FPGA computers / K. Danne, M. Platzner // Intelligent Solutions in Embedded Systems. — IEEE Computer Society, 2005. — P. 117—127.
32. Berthold O. Self-reconfiguring System-on-Chip using Linux on a Virtex-5 FPGA: Master of Science Thesis : 26.04.2012 / O. Berthold; Universität zu Berlin. — Berlin, 2012. — 101 p.
33. Wildermann S. Systematic Design of Self-Adaptive Embedded Systems with Applications in Image Processing: дис. д-ра техн. наук : 16.04.2012 / S. Wildermann; Universität Erlangen. — Nurnberg, 2012. — 215 p.

34. Paya-Vaya G. Dynamic Data-Path Self-Reconfiguration of a VLIW-SIMD Soft-Processor Architecture / G. Paya-Vaya, R. Burg, H. Blume. // Workshop on Self-Awareness in Reconfigurable Computing Systems (SRCS) — Oslo, 2012. — P. 26—29.
35. Bobda C. Enabling Self-Organization in Embedded Systems with Reconfigurable Hardware / C. Bobda, K. Cheng, F. Muhlbauer, K. Drechsler [and all] // International Journal of Reconfigurable Computing — Hindawi Publishing Corporation, 2009. — Volume 2009, Article ID 161458, 9 pages.

Рекомендована кафедрою обчислювальної техніки

Клименко Ірина Анатоліївна — канд. техн. наук, доцент, докторант, e-mail: iklymenko@yandex.ru;

Рудницький Мирослав Валентинович — магістрант кафедри обчислювальної техніки.

Національний технічний університет України «Київський політехнічний інститут», Київ

I. A. Klymenko¹
M. V. Rudnytskyi¹

Classification of reconfigurable computer systems

¹National Technical University of Ukraine «Kyiv Polytechnic Institute»

The classification of reconfigurable computer systems, which is based on the modern trends of improve performance FPGA computing systems, is suggested. On the basis of the given classification the concept of dynamic reconfiguration is defined, as the most promising technology of FPGA systems development. Partial dynamic reconfiguration methodology features are considered and comparative analysis of the technological achievements of leading FPGA manufactures is conducted.

Keywords: reconfiguration, FPGA, reconfigurable computer, reconfigurable system on chip, partial dynamic reconfiguration.

Klymenko Iryna A — Cand. Sc.(Eng.), Assistant Professor, Postdoctoral student, e-mail: iklymenko@yandex.ru;

Rudnytskyi Myroslav V. — Undergraduate Student of the Chair of Computer Engineering

И. А. Клименко¹
М. В. Рудницький¹

Классификация реконфигурируемых вычислительных систем

¹Национальный технический университет Украины «Киевский политехнический институт»

Предложено классификацию реконфигурируемых вычислительных систем, которая основывается на современных тенденциях повышения производительности вычислительных систем на ПЛИС. На базе предложенной классификации определено понятие динамической реконфигурации как наиболее перспективной технологии разработки систем на ПЛИС. Рассмотрены особенности методологии частичной динамической реконфигурации и выполнен сравнительный анализ технологических достижений передовых производителей ПЛИС.

Ключевые слова: реконфигурация, частичная динамическая реконфигурация, ПЛИС, реконфигурируемые вычислительные системы, FPGA, Reconfigurable Computer, Reconfigurable System on a Chip, Partial Dynamic Reconfiguration.

Клименко Ирина Анатольевна — канд. техн. наук, доцент, докторант, e-mail: iklymenko@yandex.ru;

Рудницький Мирослав Валентинович — магістрант кафедри вычислительной техники