

МЕТОДОЛОГІЧНІ ОСНОВИ РЕАЛІЗАЦІЇ КОМП'ЮТЕРНИХ ПРИСТРОЇВ ІЗ ЗАМІННИМИ МОДУЛЯМИ В ЧАСТКОВО РЕКОНФІГУРОВНИХ ПЛІС

© Мельник В. А., 2015

Висвітлено підхід часткового реконфігурування ПЛІС та описано механізм його реалізації. На основі узагальнення і доповнення відомих методичних та проектних рекомендацій сформульовано методологічні основи проектування комп'ютерних пристроїв для частково реконфігурованих ПЛІС. Виокремлено особливості проектування пристроїв із невизначеними замінами модулів.

Ключові слова: ПЛІС, часткове реконфігурування, реконфігуровні комп'ютерні системи.

METHODOLOGICAL BASICS OF IMPLEMENTATION OF COMPUTER DEVICES WITH RECONFIGURABLE MODULES IN PARTIALLY RECONFIGURABLE FPGAS

© Melnyk V., 2015

The article gives an overview of FPGA partial reconfiguration approach and shows the mechanism of its realization. Based on the summarizing and complementing of the existing methodological and design recommendations, a methodological basics of computer devices design for the partially reconfigurable FPGAs is formulated. The design features of the computer devices with undetermined reconfigurable modules are highlighted.

Key words: FPGA, partial reconfiguration, reconfigurable computer systems.

Вступ

Реконфігуровні комп'ютерні системи (РККС) набули сьогодні значного поширення, що пов'язано з можливістю досягнення ними високих показників завдяки апаратному виконанню обчислювальних завдань у кристалах реконфігуровної логіки, які входять до їх складу. РККС створили конкуренцію іншим типам високопродуктивних комп'ютерних систем завдяки високим технічним характеристикам сучасних кристалів програмовних логічних інтегральних схем (ПЛІС) – апаратній основі реконфігурованого середовища РККС та досягненням у галузі технологій проектування спеціалізованих процесорів, що підлягають синтезу в реконфігурованому середовищі РККС.

Під'єднання синтезованих у реконфігурованому середовищі спеціалізованих процесорів, у структурі яких враховано особливості виконуваних алгоритмів, до комп'ютерної системи на основі універсальних процесорів дає змогу на 2–3 порядки підвищити її продуктивність. Можливість реконфігурування та повторного синтезу в реконфігурованому середовищі спеціалізованих процесорів з новою структурою та функціями дозволяє змінювати функціональну орієнтацію створеної у такий спосіб РККС зі збереженням її високої продуктивності на нових класах задач.

Потреба створення методологічних основ реалізації комп'ютерних пристроїв із замінами модулів, запропонованих у цій статті, пов'язана з тим, що останніми роками все більшого поширення набувають ПЛІС, які дозволяють виконувати реконфігурування в деякій своїй частині, не змінюючи конфігурації решти обладнання. Такі ПЛІС отримали назву частково реконфігурованих (англ. *Partially Reconfigurable*). Їх застосування відкриває низку нових можливостей в обчислювальній техніці, зокрема дає змогу організувати апаратну багатозадачність у межах ПЛІС та практично втілити концепцію «віртуальних апаратних засобів».

Аналіз досліджень та публікацій

Реалізацію пристроїв реконфігуровної логіки, а також будову та організацію функціонування реконфігуровних комп'ютерних систем висвітлено в роботах [1] і [2]. В роботі [3] описано основні підходи до побудови частково реконфігуровних систем та визначено проблеми, пов'язані з їх розробленням. Методологію проектування програмних моделей обчислювальних пристроїв на рівні міжрегістрових передач викладено у статті [4]. Ця методологія є базовою і використовується для проектування та реалізації обчислювальних пристроїв у традиційних ПЛІС. Проектний потік з розроблення комп'ютерних пристроїв для частково реконфігуровних ПЛІС запропоновано в роботі [5], однак він не передбачає можливості створення пристроїв зі змінним складом компонентів. В роботі [6] описано проектний потік з розроблення комп'ютерних пристроїв для частково реконфігуровних ПЛІС компанії *Xilinx* з використанням програмного середовища *Vivado Design Suite*.

Постановка проблеми

Поряд зі значним обсягом літератури, яка описує різні аспекти будови, проектування та застосування традиційних ПЛІС і побудованих на їх основі РККС, методологічні основи розроблення комп'ютерних пристроїв у частково реконфігуровних ПЛІС надалі залишаються недостатньо висвітленими. Це значною мірою стримує розвиток напряму реконфігуровних обчислень загалом. У статті запропоновано методологію проектування комп'ютерних пристроїв для частково реконфігуровних ПЛІС та виокремлено особливості проектування пристроїв із невизначеними замінами модулями.

1. Підхід часткового реконфігурування ПЛІС

Підхід зміни конфігурації ПЛІС в окремій її частині після початкового конфігурування, за якої конфігурація інших її частин залишається незмінною, називають частковим реконфігуруванням (ЧР, англ. *Partial Reconfiguration*). Відповідно, ПЛІС, які надають можливість часткового реконфігурування, називають частково реконфігуровними. Традиційні ПЛІС дають змогу змінювати конфігурацію тільки повністю, тобто є повністю реконфігуровними, тоді як ЧР ПЛІС можна реконфігурувати або повністю, або у вибраній частині, що підвищує її ефективність в деяких застосуваннях.

До безпосередніх переваг ЧР належить істотне скорочення тривалості реконфігурування і зменшення розміру коду конфігурації та ресурсів пам'яті, необхідної для його зберігання. Це пояснюється тим, що розмір коду конфігурації прямо пропорційний до кількості ресурсів, яку ним конфігурують [7]. Крім зазначених переваг, ЧР також відкриває низку нових можливостей у використанні пристроїв реконфігуровної логіки в комп'ютерній техніці, зокрема дозволяє організувати апаратну багатозадачність у межах ПЛІС (англ. *hardware multitasking* [8], [9]) та практично втілити концепцію «віртуальних апаратних засобів» (англ. *Virtual Hardware*) [10], [11].

ЧР ПЛІС застосовують для реалізації пристроїв, компоненти яких працюють не одночасно, а в режимі часового мультиплексування. Для цього в них передбачають спільні для деякої кількості компонентів пристрою ділянки і позмінно, у визначеній послідовності, синтезують компоненти у цих ділянках. Це дає змогу зменшити споживану потужність і затрати обладнання та підвищити ефективність його використання. Для порівняння, в традиційних ПЛІС повинні бути синтезовані всі компоненти пристрою, оскільки його структура не може бути змінена під час роботи. При цьому деякі з компонентів будуть простоювати.

Для підтримки ЧР виробники ПЛІС доповнюють свої середовища проектування комп'ютерних пристроїв відповідними засобами. Наприклад, компанія *Xilinx* пропонує засіб *PlanAhead* [12], який входить до складу середовищ проектування *Vivado Design Suit* та *ISE Design Suit*. Зазначимо, що історично перші ЧР ПЛІС розробила компанія *Xilinx* (починаючи від сім'ї *Virtex* першого покоління), тому більшість наукових досліджень та інженерних робіт у цьому напрямі виконано саме з використанням їхніх ПЛІС. Компанія *Altera* впровадила ЧР до своїх ПЛІС порівняно недавно –

в серії *Stratix V* [15]. Технологія та проектний потік з його реалізації тут загалом подібні до тих, що використовує компанія *Xilinx*, і підтримується середовищем проектування *Quartus II*. З-поміж інших програмних засобів проектування комп'ютерних пристроїв у ЧР ПЛІС також треба відзначити *OpenPR* [13] від компанії *Sourceforge* і *GoAhead* [14], розроблений на факультеті інформатики Університету м. Осло. Ці засоби призначені для роботи з ЧР ПЛІС компанії *Xilinx* і є у вільному доступі.

ЧР ПЛІС здійснюють завантаженням файлів часткових конфігурацій після їх початкового конфігурування, а отже – під час роботи. Ці файли задають конфігурацію тільки відповідних частин ПЛІС, які називають реконфігуровними регіонами (анг. *Reconfigurable Partition, Reconfigurable Region, Dynamic Region*), у кожному з яких у різний час працюють різні компоненти пристрою. Реконфігуровні регіони мають наперед визначені координати і межі в кристалі ПЛІС, кількість, склад і структуру обладнання. Компоненти, які завантажують до реконфігуровних регіонів, у англійській літературі називають реконфігуровними модулями (англ. *Reconfigurable Module*). На наш погляд, означення модуля терміном «реконфігуровний», з огляду на його повсюдне використання, вносить плутанину в розуміння суті й не є таким, що однозначно і належно його характеризує, тому далі в тексті називатимемо ці модулі замінними. Крім одного чи більшої кількості реконфігуровних регіонів, ЧР ПЛІС також містить статичний регіон, в якому реалізують статичні модулі й конфігурація якого під час здійснення ЧР залишається незмінною.

ЧР є двох типів – статичне і динамічне. ЧР називають статичним, якщо перед його виконанням робота усієї ПЛІС призупиняється і відновлюється після його закінчення. ЧР називають динамічним, якщо воно здійснюється під час функціонування решти обладнання ПЛІС – статичного регіону і незадіяних у процесі реконфігуровних регіонів [16]. Зрозуміло, що динамічне ЧР є ефективнішим, ніж статичне, тому далі у статті говоритимемо саме про цей його тип. Принцип динамічного ЧР ПЛІС ілюструє рис. 1.

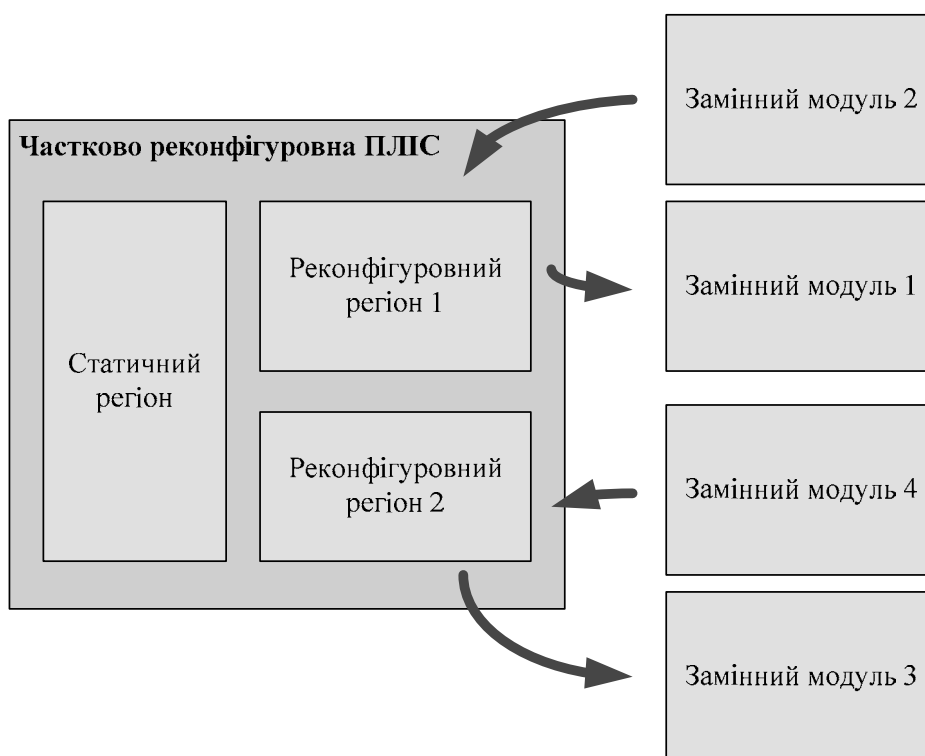


Рис. 1. Принцип динамічного часткового реконфігурування ПЛІС

У складі статичного регіону може функціонувати, наприклад, контролер часткового реконфігурування, пам'ять та інтерфейсна логіка, як це показано на рис. 2.

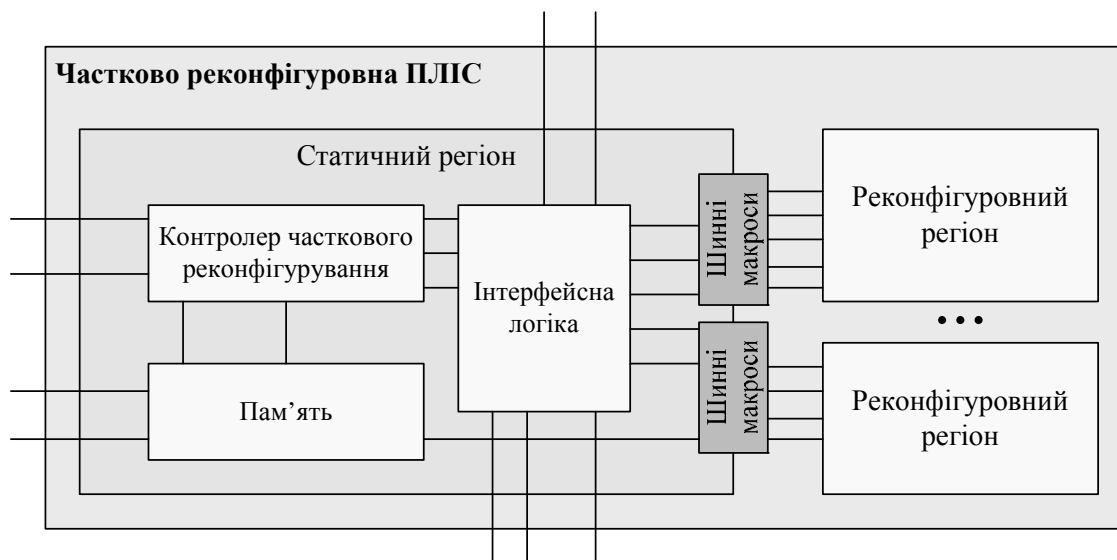


Рис. 2. Статичний та реконфігуровні регіони в складі ЧР ПЛІС

У перших поколіннях ЧР ПЛІС компанії *Xilinx* (*Virtex*, *Virtex-E*, *Virtex-II*, *Virtex-II Pro*, *Virtex-4*) взаємодія ПЛІС із зовнішнім обладнанням здійснюється лише через її статичний регіон, причому всі зв'язки замінних модулів реалізовано через блок інтерфейсної логіки. Взаємодія між модулем у реконфігуровному регіоні й рештою модулів (за винятком сигналів синхронізації, які подаються спеціальними каналами) здійснюється через так звані шинні макроси (англ. *Bus Macros*). Шинний макрос є наче «мостом» між реконфігуровним регіоном і зовнішнім відносно нього обладнанням, який створений під час початкового конфігурування ПЛІС і не змінює місця розташування під час ЧР. В новіших поколіннях ПЛІС компанії *Xilinx* шинні макроси не використовують.

Контролер ЧР (англ. *Partial Reconfiguration Controller*) автоматизує виконання ЧР. Користувач може розробити такий контролер самостійно або скористатися наявними на ринку розробками. Контролер може бути й зовнішнім щодо ПЛІС пристроєм.

2. Особливості реалізації ЧР у ПЛІС компанії *Xilinx*

У ПЛІС компанії *Xilinx* використовують два режими ЧР:

– **Модульний** (англ. *Module-based partial reconfiguration*), який передбачає повну заміну попередньо синтезованого в ПЛІС замінного модуля на новий. Цей підхід потребує сумісності інтерфейсів замінних модулів, які функціонують в одному реконфігуровному регіоні, оскільки інтерфейсне обладнання реконфігуровного регіону створюється під час початкового конфігурування і не підлягає зміні під час ЧР.

– **Різницевий** (англ. *Difference-based partial reconfiguration*), який передбачає внесення невеликих змін у схему попередньо синтезованого замінного модуля. В цьому випадку код часткової конфігурації містить інформацію про відмінності у структурах попередньо синтезованого і нового замінних модулів. Його можна сформулювати, наприклад, «злиттям» двійкових кодів попередньої та нової конфігурації за операцією *XOR* (виключне АБО) [17]. Цей підхід дає можливість суттєво зменшити об'єм коду конфігурації. Його використовують, наприклад, для заміни вмісту табличних операційних пристроїв, вмісту пам'яті тощо.

Механізм реалізації часткового реконфігурування ПЛІС з часом вдосконалюється. Наприклад, в ЧР ПЛІС *Virtex*, *Virtex-II*, *Virtex-II Pro* і *Virtex-E* компанії *Xilinx* конфігурацію можна змінювати тільки повними стовпцями реконфігуровної матриці, як це показано на рис. 3, а їх кількість повинна були кратною 4 (4, 8, 12, ...). У ПЛІС *Virtex-4* цю вимогу усунуто, натомість можна змінювати конфігурацію довільної прямокутної ділянки матриці, з деякими обмеженнями на її висоту.

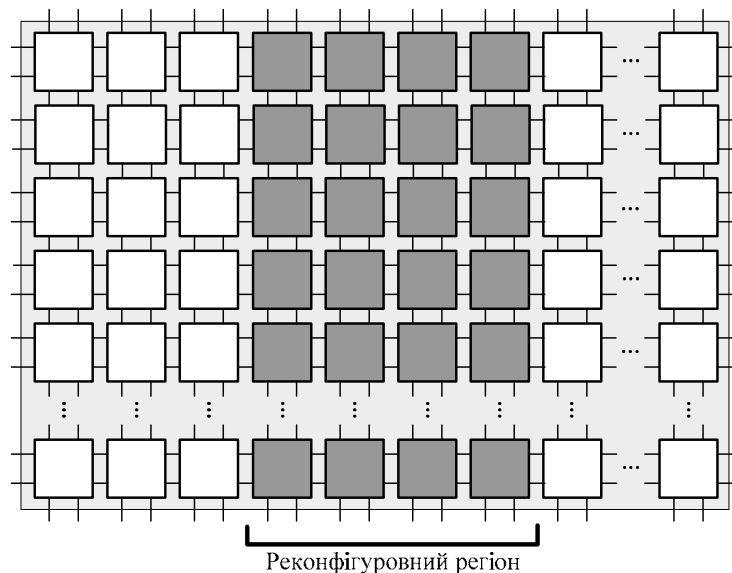


Рис. 3. Реконфігуровний регіон у ПЛІС *Virtex-II* і *Virtex-II Pro* компанії *Xilinx*

Починаючи від ПЛІС *Virtex-4* компанії *Xilinx* і до сучасних (сьогодні це пристрої сьомого покоління: *Artix-7*, *Kintex-7*, *Virtex-7*, і *Zynq-7000 SoC*) мінімальні ділянки, конфігурація яких може бути змінена, називають реконфігуровними рамками (англ. *Reconfigurable Frames*). Ширина реконфігуровної рамки становить один стовпець. Кількість елементів у стовпці реконфігуровної рамки залежить від їх типу. Наприклад, у пристроях сьомого покоління компанії *Xilinx*, це 50 CLB, 10 блоків DSP48, 10 блоків RAM [6]. У пристроях архітектури *UltraScale* [18], це 60 CLB, 24 блоки DSP48, 12 блоків RAM. Усі елементи в реконфігуровній рамці синхронізовані одним тактовим сигналом.

Компанія *Xilinx* накладає деякі обмеження на створення реконфігуровних регіонів, зокрема:

- Рекомендується створювати регіони лише прямокутної форми. Застосування інших форм, наприклад, Т- або L-подібної, ускладнює розміщення і трасування компонентів у них, результатом чого є гірші технічні характеристики реалізованих у таких регіонах модулів.
- Рекомендується залишати деякі проміжки між межами реконфігуровних регіонів, оскільки їхні інтерфейси повинні взаємодіяти із модулями статичного регіону.
- Не дозволяється реалізовувати вкладені реконфігуровні регіони або з перекриттям.
- Одна реконфігуровна рамка повинна належати тільки одному реконфігуровному регіону.

Потрібно звернути увагу й на те, що не всі типи елементів ПЛІС піддаються ЧР. Перелік цих елементів у різних ПЛІС різний, наприклад, у згаданих вище пристроях сьомого покоління ЧР піддаються елементи CLB, BRAM, DSP. Натомість не можна конфігурувати елементи, задіяні в генеруванні й поширенні синхросигналу в кристалі: BUFG, BUFR, PLL тощо, елементи інтерфейсу введення–виведення і внутрішніх інтерфейсів: ISERDES, OSERDES, BSCAN, ICAP, STARTUP тощо. В пристроях архітектури *UltraScale* перелік конфігуровних елементів ширший і не містить лише внутрішні інтерфейси й елементи, що їх реалізують. Під час проектування потрібно враховувати, що нереконфігуровні елементи можуть міститися виключно в статичному регіоні ПЛІС.

Файл часткової конфігурації складається з деякої цілої кількості конфігураційних рамок (англ. *Configuration Frames*, не плутати з реконфігуровними рамками). Конфігураційна рамка є мінімальною одиницею інформації цього файлу і задає конфігурацію однієї реконфігуровної рамки.

Тривалість часткового реконфігурування залежить від розміру файла конфігурації і ширини інтерфейсу, який використовується для її завантаження. Характеристики пропускної здатності інтерфейсів, що використовуються для завантаження конфігурації у пристроях сьомого покоління компанії *Xilinx*, наведені в табл. 1 [6]. Зазначимо, що дані конфігураційних рамок зберігаються у високошвидкісних елементах статичної пам'яті (SRAM).

**Характеристики пропускної здатності інтерфейсів, що використовуються
для завантаження конфігурації у пристроях сьомого покоління компанії Xilinx**

Режим конфігурування	Максимальна частота, МГц	Розрядність інтерфейсу, бітів	Максимальна пропускна здатність
ICAP	100	32	3,2 Гб/с
SelectMAP	100	32	3,2 Гб/с
Serial	100	1	100 Мб/с
JTAG	66	1	66 Мб/с

Точний розмір конфігурації вказано в заголовку конфігураційного файла. Приклад заголовка файла з конфігурацією розміром 1211072 біти наведено на рис. 4.

```
Xilinx ASCII Bitstream
Created by Bitstream 2015.1
Design name: led_shift_count;UserID=0xFFFFFFFF
Architecture:kintex7
Part:      7k325tffg900
Date:      Mon Mar 16 16:42:05 2015
Bits:      1211072
111111111111111111111111111111111111
```

Рис. 4. Приклад заголовка конфігураційного файла

**3. Методологія проектування комп'ютерних пристроїв із замінними модулями
для частково реконфігурованих ПЛІС**

Методологію проектування комп'ютерних пристроїв із замінними модулями для ЧР ПЛІС ілюструє рис. 6. Вона узагальнює і доповнює подані в роботах [5] і [6] підходи і дещо відрізняється від традиційної методології проектування та реалізації комп'ютерних пристроїв у ПЛІС [4].



Рис. 5. Методологія проектування комп'ютерних пристроїв із замінними модулями для частково реконфігурованих ПЛІС

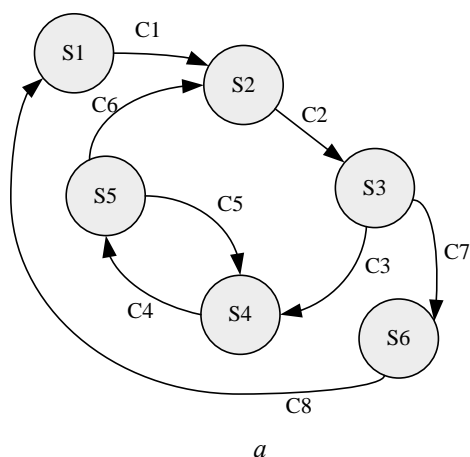
Аби не ускладнювати наведену на рис. 6 діаграму, етапи, пов'язані із тестуванням і відлагодженням – створення системи тестування пристрою і його модулів, функціональна симуляція, часова симуляція і системне тестування, які, вочевидь, передбачені в методології, на ній не показано. Нижче розглянемо основні етапи методології проектування комп'ютерних пристроїв із замінними модулями для частково реконфігуровних ПЛІС, акцентуючи на відмінностях від традиційної методології проектування.

3.1. Проектування архітектури пристрою

Як і в традиційній методології проектування, на цьому етапі здійснюють функціональну декомпозицію і розробляють архітектуру пристрою. Характерною особливістю проектування для ЧР ПЛІС є те, що пристрій містить динамічну частину, яка в різні моменти його роботи складається з різних функціональних модулів. Тому під час проектування архітектури пристрою додатково необхідно виконати такі роботи:

- 1) визначити його статичні модулі;
- 2) визначити його замінні модулі й описати порядок їх взаємозаміни під час роботи пристрою;
- 3) згрупувати замінні модулі за можливістю їх розміщення в одному реконфігуровному регіоні ПЛІС у режимі часового мультиплексування;
- 4) описати структуру пристрою як множину його конфігурацій (тут під конфігурацією розуміємо кортеж компонентів – статичних і замінних модулів);
- 5) визначити кількість реконфігуровних регіонів і склад замінних модулів у кожному з них.

Порядок заміни модулів пристрою у частково реконфігуровній ПЛІС можна задати кількома способами, наприклад, таблично чи діаграмою станів, де кожному стану відповідає деяка його конфігурація (кортеж компонентів). Зміна стану відбувається під час виконання відповідної умови переходу. Приклад специфікації порядку заміни модулів у деякому пристрої з шістьма можливими конфігураціями наведено на рис. 6.



Поточний стан	Конфігурація	Наступний стан	Умова переходу
S1	Конфігурація 1	S2	C1
S2	Конфігурація 3	S3	C2
S3	Конфігурація 2	S4	C3
		S6	C7
S4	Конфігурація 4	S5	C4
S5	Конфігурація 6	S2	C6
		S4	C5
S6	Конфігурація 5	S1	C8

Рис. 6. Приклад специфікації порядку заміни модулів у деякому пристрої з шістьма можливими конфігураціями:
а – діаграмою станів; б – таблично

Множину конфігурацій пристрою також можна описати таблично, визначивши для кожної з них вміст статичного регіону і усіх реконфігуровних регіонів, як показано у табл. 2 (де N_{CONF} – кількість конфігурацій, N_{RP} – кількість реконфігуровних регіонів, N_{SM} – кількість статичних модулів).

Опис множини конфігурацій пристрою

Конфігурація 1	Статичний регіон	Статичний модуль 1 ... Статичний модуль N_{SM}
	Реконфігуровний регіон 1	Замінний модуль RM_i
	... Реконфігуровний регіон N_{RP}	... Замінний модуль RM_j
...
Конфігурація N_{CONF}	Статичний регіон	Статичний модуль 1 ... Статичний модуль N_{SM}
	Реконфігуровний регіон 1	Замінний модуль RM_p
	... Реконфігуровний регіон N_{RP}	... Замінний модуль RM_q

Подані на рис. 6 і в табл. 2 описи є, по суті, алгоритмічним забезпеченням контролера часткового реконфігурування.

3.2. Розроблення програмної моделі пристрою мовою опису апаратних засобів

а) Розроблення опису верхнього рівня.

На найвищому рівні ієрархії пристрій повинен бути описаний тільки на структурному рівні опису апаратури. Використовувати тут інші рівні опису апаратури, наприклад, поведінковий, міжрегістрових передач, не дозволяється. Опис верхнього рівня повинен містити декларації інтерфейсу пристрою, специфікації статичних модулів, специфікації замінних модулів, карти портів і декларації сигналів. Також у разі використання в цьому (і тільки в цьому) описі задають всі шинні макроси, що з'єднують замінні модулі пристрою зі статичними. Під час розроблення опису необхідно враховувати, що заміна одних компонентів пристрою іншими під час виконання ЧР повинна відповідати інтерфейсним специфікаціям компонентів і картам портів у описі верхнього рівня, який є спільним для всіх конфігурацій.

б) Розроблення статичних модулів.

Статичні модулі належать до тієї частини пристрою, яка залишається незмінною під час виконання часткового реконфігурування. Ці модулі реалізують відповідно до традиційної методології проектування комп'ютерних пристроїв у ПЛІС [4].

в) Розроблення замінних модулів.

Загалом проектування замінних модулів здійснюють за традиційною методологією, але до них застосовують важливе обмеження: замінні модулі, які розміщені в одному реконфігуровному регіоні, повинні мати однакові інтерфейси, разом з іменами і властивостями портів. Це зумовлено тим, що з'єднання (зокрема з використанням шинних макросів), через які здійснюється взаємодія реконфігуровного регіону зі статичним, мають фіксоване розміщення в ПЛІС, а отже, мають бути ідентичними логічно і фізично. Компанія *Xilinx* в останніх версіях своїх програмних засобів дозволяє відмінність інтерфейсів замінних модулів за умови, що декларація і карта портів компонента, що відповідає цим модулям в описі верхнього рівня, повинні містити повний набір (англ. *superset*) портів усіх цих модулів [6, с. 11].

3.3. Встановлення проектних обмежень

Після розроблення програмної моделі пристрою необхідно задати проектні обмеження (англ. *Design Constraints*). Зокрема, ними вказують для кожного компонента пристрою належність до групи статичних або групи замінних модулів (англ. *area group*) і часові обмеження

(англ. *timing constraints*), а також режим виконання часткового реконфігурування – модульний чи різницевий (англ. *reconfiguration mode*). Встановлення часових обмежень виконується так само, як і в традиційному проектуванні. Окрім названих, можуть використовуватись й інші проектні обмеження.

3.4. Логічний синтез статичних модулів і опису верхнього рівня пристрою

На цьому етапі за допомогою відповідних програмних засобів виконують перевірку коректності опису верхнього рівня і проектних обмежень та синтез статичних модулів і опису верхнього рівня за методикою «знизу–доверху». Під час логічного синтезу компілятор замість замінних модулів встановлює так звані «чорні скриньки», тобто компоненти, які містять тільки інтерфейсний опис. Результатом цього етапу є програмна модель пристрою на рівні примітивів цільової ПЛІС (нетліст) і технічні характеристики модулів, зокрема, кількість обладнання і частота роботи.

3.5. Логічний синтез замінних модулів

Логічний синтез замінних модулів виконують з урахуванням результатів перевірки коректності опису верхнього рівня і проектних обмежень. Під час його виконання потрібно вимкнути опцію встановлення буферів введення–виведення, оскільки замінні модулі не можуть мати прямих зв'язків з выводами ПЛІС. Результатом цього етапу є нетліст і технічні характеристики кожного замінного модуля.

3.6. Планування розміщення реконфігурованих регіонів

Відповідно до даних, отриманих під час проектування архітектури пристрою – опису порядку заміни модулів і опису множини конфігурацій, та на основі отриманих у результаті логічного синтезу технічних характеристик замінних модулів визначають кількість ресурсів кожного реконфігурованого регіону та здійснюють планування їх розміщення в ПЛІС (англ. *Floorplanning*). Кількість ресурсів реконфігурованого регіону вибирають відповідною до потреб найбільшого замінного модуля, який у ньому працюватиме, з урахуванням обмежень на форму регіону і висоту реконфігурованої рамки. Важливим питанням є ефективність використання ресурсів реконфігурованого регіону, яку можна визначити, скориставшись таким виразом:

$$E(RP_k) = \sum_{i=1}^{N_{RM}^{RP_k}} \frac{Q(RM_i)}{Q(RP_k)} \times \frac{T(RM_i^{RP_k})}{T(RP_k)}, \quad (1)$$

де $E(RP_k)$ – показник ефективності використання обладнання реконфігурованого регіону RP_k ($\max E(RP_k) \rightarrow 1$); $Q(RM_i)$ і $T(RM_i^{RP_k})$ – відповідно, кількість обладнання, необхідного для реалізації замінного модуля RM_i та час його роботи в реконфігурованому регіоні RP_k ; $Q(RP_k)$ і $T(RP_k)$ – відповідно, кількість обладнання реконфігурованого регіону RP_k та загальний час його роботи в ПЛІС; $N_{RM}^{RP_k}$ – кількість замінних модулів, які використовують реконфігурований регіон RP_k у режимі часового мультиплексування.

Використання модуля, що займає невелику частку обладнання реконфігурованого регіону, не призведе до суттєвого зниження ефективності, якщо відсоток часу його роботи в регіоні незначний.

Компанія *Xilinx* розробила методологію планування своїх ПЛІС [19] і пропонує для цього спеціальний інструментарій. Важливим завданням є автоматизація розміщення реконфігурованих регіонів у ПЛІС. Її мета – автоматичне групування замінних модулів за реконфігурованими регіонами і визначення оптимального місця їх розташування в ПЛІС. Роботи в цьому напрямі сьогодні тривають [20], [21], однак програмних засобів автоматичного розміщення реконфігурованих регіонів у ПЛІС до систем проектування поки що не впроваджено.

3.7. Імплементация

Етап імплементации (англ. *Place and Route* – розміщення і трасування; його також розглядають як останню фазу логічного синтезу) виконують ітераційно.

На першій ітерації до структури пристрою вносять всі статичні модулі й ті замінні модулі, які мають бути синтезовані під час першого, початкового конфігурування ПЛІС. Отриману в результаті виконання першої ітерації схему розміщення і трасування запам'ятовують. Після цього замінні модулі вилучають із проекту, а на їхні місця встановлюють «чорні скриньки».

На наступних ітераціях замість «чорних скриньок» підставляють інші замінні модулі, здійснюють їх імплементацию у складі пристрою і запам'ятовують схеми їх розміщення і трасування. Так по черзі імплементують всі конфігурації пристрою. Статичні модулі повторно не імплементують, а використовують отриману на першій ітерації схему їх розміщення і трасування. Варто зазначити, що наявність реконфігурованих регіонів дещо ускладнює оптимізаційні процедури під час імплементации, оскільки оптимізація схеми може здійснюватись лише в межах окремого регіону, а не усього кристала ПЛІС.

Після імплементации всіх модулів рекомендовано виконати перевірку сумісності конфігурацій, яка дає змогу виявити можливі відмінності в межах статичного регіону й у інтерфейсах, що сполучають його з реконфігурованими регіонами.

3.8. Генерування кодів конфігурації

На цьому етапі на основі результатів імплементации генерують бінарні файли з кодами конфігурації, один – що задає початкову конфігурацію ПЛІС, і по одному файлу з частковими конфігураціями для кожного замінного модуля, зокрема й тих модулів, що входять до початкової конфігурації. Якщо пристрій не містить статичних модулів, можливе генерування тільки файлів часткових конфігурацій.

4. Особливості проектування комп'ютерних пристроїв із невизначеними заміними модулями

Описана вище методологія проектування застосовується для реалізації комп'ютерних пристроїв із наперед визначеними заміними модулями. Функціональні й технічні вимоги до таких пристроїв і їхніх компонентів відомі заздалегідь, що дає можливість розробити їх архітектуру, здійснити проектування усіх модулів, визначити порядок їх заміни і налагодити роботу у всіх конфігураціях. Однак є категорія пристроїв, в яких склад замічних модулів на час реалізації невизначений або визначений не повністю. До них належать, наприклад, пристрої багатоцільового використання, в яких замінні модулі визначають функціональну спрямованість і постачаються в процесі експлуатації, пристрої, що підлягають регулярному оновленню тощо.

Порядок проектування таких пристроїв дещо відрізняється від описаного вище. В деяких публікаціях, наприклад [22], для них запропоновано окрему методологію. Ми ж встановимо особливості проектування пристроїв із невизначеними заміними модулями і виділимо відмінності від наведеного вище порядку.

Згідно з наведеним вище порядком проектування, планування розміщення реконфігурованих регіонів здійснюється на основі даних, отриманих в результаті логічного синтезу пристрою, разом з усіма його заміними модулями. Якщо ж потрібно імплементувати пристрій без замічних модулів (або їх частини) і згенерувати початковий код конфігурації ПЛІС, розробник змушений діяти навпаки – спочатку здійснити планування розміщення реконфігурованих регіонів, заповнивши їх «чорними скриньками», визначити кількість обладнання й інтерфейси кожного з них, а після цього здійснювати проектування та імплементацию замічних модулів.

Отже, порядок проектування пристроїв із невизначеними заміними модулями можна подати двома рознесеними в часі етапами: 1) створення пристрою без замічних модулів (або їх частини); 2) проектування та імплементация замічних модулів. Результатом зміни порядку проектування є введення додаткових обмежень до проектування й імплементации замічних модулів, якими є тип інтерфейсу, кількість обладнання реконфігурованого регіону і місце його розміщення у ПЛІС. Ще

одним обмеженням є кількість реконфігурованих регіонів у ПЛІС, яка визначає максимальну кількість замінних модулів, що одночасно працюють у пристрої.

Питання визначення інтерфейсів замінних модулів вирішується їх уніфікацією, що сьогодні не становить проблеми, адже розроблено і затверджено стандартами спеціальні інтерфейси, наприклад, *AMBA* [23], *Wishbone* [24], *CoreConnect* [25], які використовують у комп'ютерних системах на кристалі.

Висновки

У статті висвітлено підхід часткового реконфігурування ПЛІС, який застосовують для реалізації пристроїв, функціональні модулі яких працюють у режимі часового мультиплексування. Безпосередніми перевагами цього підходу є істотне зменшення тривалості реконфігурування і зменшення необхідного для зберігання конфігурації об'єму пам'яті. Також його застосування надає низку нових можливостей в обчислювальній техніці, зокрема дає змогу організувати апаратну багатозадачність у межах ПЛІС та практично втілити концепцію «віртуальних апаратних засобів».

На основі узагальнення і доповнення різних підходів та проектних рекомендацій запропоновано методологію проектування комп'ютерних пристроїв для частково реконфігурованих ПЛІС, яка передбачає виконання таких етапів:

- проектування пристрою на архітектурному рівні;
- створення програмної моделі пристрою мовою опису апаратних засобів, реалізація структури верхнього рівня і статичних модулів;
- реалізація замінних модулів;
- встановлення проектних обмежень;
- логічний синтез статичних модулів і верхнього рівня пристрою;
- логічний синтез замінних модулів;
- планування розміщення реконфігурованих регіонів;
- імплементація;
- генерування кодів конфігурації.

Виокремлено особливості проектування пристроїв із невизначеними замінними модулями. Цей порядок можна подати двома рознесеними в часі етапами: 1) створення (від розроблення архітектури до генерування початкового коду конфігурації) пристрою без замінних модулів (або їх частини); 2) проектування інших замінних модулів.

1. Scott Hauck, André DeHon. *Reconfigurable Computing: The Theory and Practice of FPGA-Based Computation* / Morgan Kaufmann, 2008. – 944 p. 2. Мельник А. О. *Персональні суперкомп'ютери: архітектура, проектування, застосування: монографія* / А. О. Мельник, В. А. Мельник. – Львів: Видавництво Львівської політехніки, 2013. – 516 с. 3. 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. 4. Мельник А. *Технологія проектування ядер комп'ютерних пристроїв* / А. Мельник, В. Мельник // *Вісник Національного університету "Львівська політехніка" "Комп'ютерні системи та мережі"*. – 2002. – № 463. – С. 3–9. 5. Wang Lie, Wu Feng-yan. *Dynamic Partial Reconfiguration in FPGAs* // *Intelligent Information Technology Application*, 2009. IITA 2009. *Third International Symposium on*, vol. 2. – P. 445–448, 21–22 Nov. 2009. 6. *Vivado Design Suite User Guide. Partial Reconfiguration. UG909 (v2015.2) June 24, 2015.* [Електронний ресурс]. – Режим доступу: http://www.xilinx.com/support/documentation/sw_manuals/xilinx2015_2/ug909-vivado-partial-reconfiguration.pdf. 7. McDonald E. J. *Runtime FPGA Partial Reconfiguration. Aerospace Conference, 2008 IEEE, Los Angeles, 2008.* – P. 1–7. 8. Iturbe X., Benkrid K., Arslan T., Torrego R. and Martinez I. *Methods and mechanisms for hardware multitasking: executing and synchronizing fully relocatable hardware tasks in Xilinx FPGAs* // in *Proceedings of the 21st International Conference on Field Programmable Logic and Applications (FPL '11)*. – P. 295–300, September 2011. 9. Kalte H. and Pormann M. *Context Saving and Restoring for Multitasking in Reconfigurable Systems* // *Proc. of the International Conference on Field Programmable Logic and Applications*. – 2005. –

P. 223–228. 10. Brebner G. J. A Virtual Hardware Operating System for the Xilinx XC6200 // *Proc. of the International Workshop on Field-Programmable Logic, Smart Applications, New Paradigms and Compilers*, 1996. 11. Brebner G. The swappable logic unit: a paradigm for virtual hardware // In K. L. Pocek and J. M. Arnold, editors, *The 5th Annual IEEE Symposium on FPGAs for Custom Computing Machines (FCCM'97)*. – P. 77–86, Los Alamitos, CA, Apr. 1997. IEEE Computer Society Press. 12. Partial Reconfiguration Tutorial. PlanAhead Design Tool. UG743 (v14.1) May 8, 2012. [Електронний ресурс]. – Режим доступу: http://www.xilinx.com/support/documentation/sw_manuals/xilinx14_2/PlanAhead_Tutorial_Partial_Reconfiguration.pdf. 13. Sohaghpurwala A., Athanas P., Frangieh T. and Wood A. OpenPR: An Open-Source Partial-Reconfiguration Toolkit for Xilinx FPGAs // In *Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW)*, 2011 IEEE International Symposium on, May 2011. – P. 228–235. 14. Beckhoff C., Koch D. and Torresen J. GoAhead: A Partial Reconfiguration Framework // In *Field-Programmable Custom Computing Machines (FCCM)*, 2012 IEEE 20th Annual International Symposium, Toronto, Canada, 29.04.2012–1.05.2012. – P. 37–44. 15. Increasing Design Functionality with Partial and Dynamic Reconfiguration in 28-nm FPGAs. July 2010, Altera Corporation. [Електронний ресурс]. – Режим доступу: https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/wp/wp-01137-stxv-dynamic-partial-reconfig.pdf. 16. Virtex-4 Configuration Guide, Xilinx, Inc. [Електронний ресурс]. – Режим доступу: http://www.xilinx.com/support/documentation/user_guides/ug071.pdf. 17. Sedcole P., Blodget B., Becker T., Anderson J. and Lysaght P. Modular dynamic reconfiguration in Virtex FPGAs // *IEE Proceedings Computers and Digital Techniques*, 153(3):157–164, 2006. 18. UltraScale Architecture. [Електронний ресурс]. – Режим доступу: <http://www.xilinx.com/products/technology/ultrascale.html>. 19. Floorplanning Methodology Guide. UG633 (v14.5) April 10, 2013. [Електронний ресурс]. – Режим доступу: http://www.xilinx.com/support/documentation/sw_manuals/xilinx14_7/Floorplanning_Methodology_Guide.pdf. 20. Vipin K., Fahmy S. A. Efficient Region Allocation for Adaptive Partial Reconfiguration // *Proc. of IEEE International Conference on Field Programmable Technology (FPT)*, New Delhi, 2011. 21. Vipin K., Fahmy S. A. An Approach to a Fully Automated Partial Reconfiguration Design Flow // *Field-Programmable Custom Computing Machines (FCCM)*, 2013 IEEE 21st Annual International Symposium on. – P. 231–231. 22. Conger C., Hymel R., Rewak M., George A. D. and Lam H. FPGA design framework for dynamic partial reconfiguration // In *Proceedings of Reconfigurable Architectures Workshop (RAW)*, 2008. 23. AMBA Specification (Rev 2.0). ARM Limited, 1999. – 230 p. 24. Wade D. Peterson, Silicore Corporation. Specification for the WISHBONE System-on-Chip (SoC) Interconnection Architecture for Portable IP Cores. Revision: B.3, Released: September 7, 2002. – 140 p. 25. Milica Mitić and Mile Stojčv. A Survey of Three System-on-Chip Buses: AMBA, CoreConnect and Wishbone // *ICEST 2006*.