

УДК 004.274

В.М. ОПАНАСЕНКО, О.М. ЛІСОВИЙ

Інститут кібернетики ім. В.М. Глушкова НАН України, Україна

ДВА ПІДХОДИ ДО ФОРМАЛІЗАЦІЇ ПРОЦЕСУ ПРОЕКТУВАННЯ  
ПРОБЛЕМНО-ОРІЄНТОВАНИХ ПРИСТРОЇВ

В роботі запропоновано два підходи до формалізації процесу проектування проблемно-орієнтованих пристроїв: перший представлено прямонаправленим графом, другий графом з підзадачею вибору множини оптимально реалізованих функцій. Розроблено загальний алгоритм процесу проектування на ПЛІС, що відповідає формалізації процесу проектування, дозволяє обрати одну з множини розроблених реалізацій пристрою за заданими обмеженнями параметрів або за постановкою задачі оптимізації. Запропоновано структуру віртуальної лабораторії з проектування обчислювальних систем на основі ПЛІС

**Ключові слова:** FPGA, віртуальна лабораторія, ПЛІС, проблемна орієнтація, реконфігуровні пристрої.

## Вступ

Автором ідеї адаптації апаратних засобів до алгоритму задачі, тобто реалізації проблемно-орієнтованих пристроїв, був американський фізик і математик Джон Паста, а в 1960 р. Джеральд Естрін (Каліфорнійський університет Берклі) доповів на конференції Western Joint Computer Conference [1] про комп'ютерну систему з жорсткою та гнучкою частинами, яка представляла собою набір модулів для збірки спеціалізованих комп'ютерів під задану задачу. Мікроелектронна технологія 60-х років не дала можливості реалізувати реконфігуровні комп'ютери, однак теоретичні основи реконфігуровного комп'ютинга вже було сформовано.

Реконфігуровні комп'ютери на базі ПЛІС ефективно застосовуються в багатьох областях: реконфігуровні високопродуктивні багато процесорні обчислювальні системи (з можливістю реконфігурування по телеметричних каналах зв'язку, наприклад у радіоастрономії); емуляція й проектування нових бездротових систем зв'язку; наукове обчислення в реальному масштабі часу й моделювання; спеціалізовані автономні вбудовані пристрої; кодування–декодування інформації (криптографія); цифрова обробка сигналів (фільтри, обробка радіотелескопічних сигналів, когнітивні радіосистеми, обробка зображень); робототехніка й нейронні мережі; бортові (авіаційна, космічна й морська техніка) і віддалені на велику відстань (газові й нафтові вишки) обчислювальні системи; комунікаційні засоби (інтерфейси, порти, канали); контролери для керування складними об'єктами (АЕС, ТЕЦ, ГЕС та ін.).

Базовою компонентою сучасних реконфігурованих комп'ютерів є кристали FPGA (англ. field-

programmable gate array) – програмовні користувачем вентиляльні матриці. У внутрішній області яких розташовані: матриця регулярно розташованих ідентичних конфігурованих логічних блоків – Configurable logic blocks (CLB) між якими проходять канали трасувань, множина вбудованих мікропроцесорів PowerPC (PPC), блоків статичної пам'яті (Block RAM), модулів цифрової обробки сигналів Digital Signal Processing (DSP) а також швидкодіючих приймачів-передавачів Rocket IO GTP/GTX transceivers (MGT). На границях кристала розташовано матрицю блоків вводу/виводу Input/output Blocks (IOB) та засоби синхронізації Clocks & Delay-Locked Loop (DLL).

## 1. Постановка задачі

У реконфігурованих пристроях (РП) логічна структура може динамічно змінюватися, залежно від специфіки розв'язуваної задачі, шляхом передачі спеціальної інформації про кожну поточну конфігурацію. Застосовуючи технологію реконфігурування, вдається простим перепрограмуванням структури РП налаштувати його на ефективну реалізацію заданого алгоритму, зберігаючи структурну універсальність пристрою [2, 3].

У реконфігурованих пристроях (рис. 1) задана базова (нульова) архітектура, що представлена п'ятіркою:

$$A_0 = \langle F, C, M, E, I \rangle, \quad (1)$$

де  $F = \{F_{ij}\}$ ,  $(i = 1 \div n; j = 1 \div m)$  – функціональне обробне поле, тобто матриця кристалів FPGA/CPLD;  $C$  – контролер шини host-комп'ютера, який визначає розрядність, частоту й тип шини (PCI, PCI-E,

VME й так далі); M – поле SRAM– або DRAM–пам'яті для кешування даних відповідної розрядності та ємності; E – роз'єми для підключення зовнішніх пристроїв (плат розширення) з метою створення складних замкнених систем; I – пам'ять для зберігання файлів конфігурації (PROM– або FLASH–пам'ять).

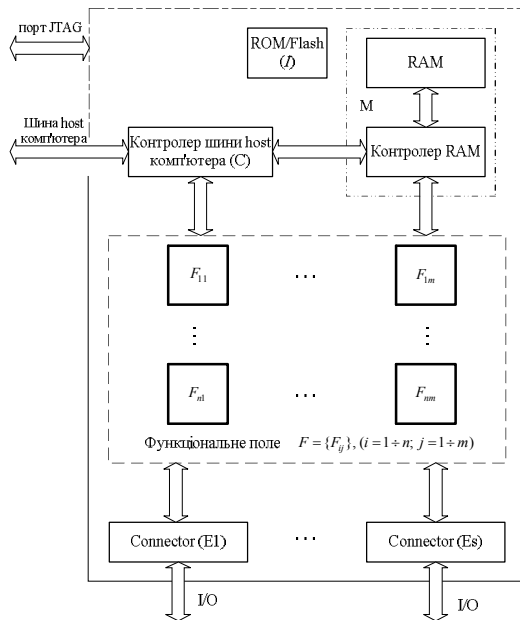


Рис. 1. Узагальнена структурна схема РП

Для забезпечення працездатності РП, як співпроцесора, що встановлюється на системну плату базового комп'ютера, використовується контролер шини, до якої виконується підключення. Контролер може бути реалізований як в окремому кристалі, що виконує тільки функції контролера шини, так й в кристалі, призначеному для розміщення структури співпроцесора. У більшості випадків РП повинен містити пам'ять (SRAM або DRAM), що використовується як Cache для зберігання вхідних/вихідних даних і/або проміжних результатів.

Наведена структура РП дозволяє, при підключенні комп'ютера до мережі Internet, робити модифікацію конфігураційних файлів через цю мережу. РП розробляються під промислові стандарти, такі як PCI, PCI Express, CompactPCI, PMC (PCI Mezzanine Cards), VME та інші. Специфікація PMC дозволяє додавати дочірні модулі (плати розширення) до базових плат через відповідну локальну шину.

На процес проектування обчислювальних пристроїв впливають безпосередньо характеристики вибраного кристала (наявні логічні ресурси, пам'ять, наявні hard core, й інше) та набір інструментальних засобів для проектування, які можуть включати бібліотеки готових технічних рішень – soft core й інші бібліотеки. Архітектури сучасних кристалів FPGA сімейства Virtex 6 оптимізовано для використання

hard core й soft core, наприклад кристал XC6VLX365T сімейства Virtex 6 має наступні вбудовані блоки: 576 вдосконалених перемножувачів DSP48E1Slices, контролер шини PCI-Express й інші.

В відомих методах формалізованого проектування обчислювальних пристроїв [4] сам процес відображається послідовністю етапів, на кожному з яких проект представлено сукупністю математичних моделей, яка описує їх різні частини. Розрізняють три головні види моделей – функціональні, динамічні, структурні. Функціональні моделі встановлюють функції, які виконує система що проектується, динамічні встановлюють процеси функціонування системи або процеси обчислень, структурні – відображають систему в вигляді композиції взаємозв'язаних компонентів.

Відповідно до [2] модель пристрою що проектується можна представити, як  $S = \langle M, A, B, P \rangle$ , де M – множина математичних методів, A – множина алгоритмів реалізації метода,  $B = \{b\}$  – алфавіт конструктивів, з яких будується структура, P – процедура опису проекту. Процес проектування полягає в розв'язанні задачі синтезу структури на базі конструктивів  $\{b\}$  алфавіту B для виконання алгоритму A за методом M. Результатом процедури P є опис проекту вхідною мовою САПР. Відмітимо, що критерієм ефективності методу (алгоритму) є загальні характеристики розробленого пристрою: апаратні затрати, швидкодія, похибка обчислень, складність структури, надійність або спеціальні критерії.

## 2. Проектування проблемно-орієнтованих пристроїв

Розглянемо два підходи до формалізації процесу проектування проблемно-орієнтованих пристроїв: перший представлено прямонаправленим графом, другий графом з підзадачею вибору множини оптимально реалізованих функцій.

**Перший тип формалізації** – граф представлений на рис. 2. Нехай задана постановка задачі D з предметної області, яка може бути розв'язана декількома методами  $D = \bigcup M_i, (\forall i = 1 \div n)$ .

Для кожного з методів існує деяка множина алгоритмів реалізації  $M_i = \bigcup A_{ij}, (\forall j = 1 \div m_i)$ . Кожний з алгоритмів реалізується на базі множини  $B = \{F_\lambda\}, (\forall \lambda = 1 \div s)$  заданих функцій (додавання, множення й інші). Кожна функція може бути реалізована декількома варіантами  $F_\lambda = \bigcup R_\omega, (\forall \omega = 1 \div z_\lambda)$ , тому загальна кількість ймовірних варіантів реалізацій множин функцій –

$KR'_{ij} = z_{ij1} \times z_{ij2} \times z_{ij3} \times \dots \times z_{ijS}$ , однак варіанти реалізацій множин функцій  $KR''$  (кількість варіантів реалізацій множин функцій, які не задовольняють постановці задачі оптимізації) не розглядаються, оскільки для будь-якої реалізації структури пристрою, характеристики пристрою не будуть задовольняють постановці задачі. Кількість варіантів, що задовольняють постановці задачі, буде визначатися величиною  $KR_{ij} = KR'_{ij} - KR''_{ij}$ .

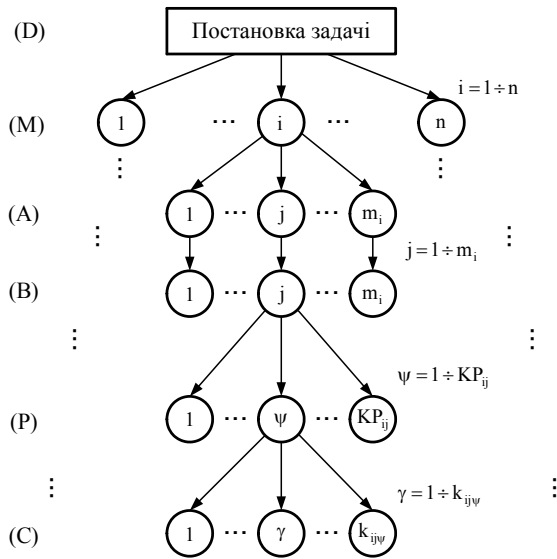


Рис. 2. Формалізація процесу проектування – перший тип

Рішення задачі синтезу реалізації структури пристрою полягає в виборі з набору існуючих  $\{C_{ij\psi\gamma}\}$ ,  $(\forall \gamma = 1 \div k_{ij\psi})$  однієї оптимальної структури, що задовольняє постановці задачі оптимізації. Розглянемо постановку задачі оптимізації для першого типу формалізації, яка може бути сформульована як задача мінімізації цільової функції з комплексними параметрами:

$$\alpha \sum_i \sum_j \sum_\psi \sum_\gamma q_{ij\psi\gamma} x_{ij\psi\gamma} + \beta \sum_i \sum_j \sum_\psi \sum_\gamma t_{ij\psi\gamma} x_{ij\psi\gamma} \Rightarrow \min; \quad (2)$$

$$(\forall i = 1 \div n); (\forall j = 1 \div m_i); (\forall \psi = 1 \div KR_{ij\psi}); (\forall \gamma = 1 \div k_{ij\psi}), \quad (3)$$

враховуючи обмеження:

$$\sum_i \sum_j \sum_\psi \sum_\gamma q_{ij\psi\gamma} x_{ij\psi\gamma} \leq Q_0; \quad (4)$$

$$\sum_i \sum_j \sum_\psi \sum_\gamma t_{ij\psi\gamma} x_{ij\psi\gamma} \leq T_0; \quad (4)$$

$$(\alpha + \beta) = 1, \sum_\gamma x_{ij\psi\gamma} = 1, \quad (5)$$

де  $\alpha, \beta$  – вагові коефіцієнти, які можуть бути визначені методом експертних оцінок ( $\alpha, \beta \in [0,1]$ );  $q_{ij\psi\gamma}, t_{ij\psi\gamma}$  – (відносні, безрозмірні) апаратні та часові оцінки  $\gamma$ -го варіанту структури пристрою побудованого на функціях з  $\psi$ -ої множини реалізацій функцій для  $j$ -го алгоритму  $i$ -го метода поставленої задачі;  $x_{ij\psi\gamma}$  – бульова змінна, яка визначає відповідний варіант реалізації структури пристрою ( $x_{ij\psi\gamma} \in \{0,1\}$ );  $Q_0, T_0$  – (відносні, безрозмірні) апаратні та часові обмеження в постановці задачі D.

Тоді вирішення задачі може бути отримано методами цілочислового математичного програмування. Методи розв'язання таких задач досить добре розроблені й дозволяють за припустимий час одержувати прийнятне рішення [5].

Для такої постановки задачі оптимізації  $KR''$  дорівнюватиме кількості тих варіантів реалізацій множин функцій, що не задовольняють системі:

$$\begin{cases} \sum_\lambda Q_\lambda < Q_0; \\ \max \{T_\lambda\} < T_0, \end{cases} \quad (6)$$

де  $\sum_\lambda Q_\lambda$  – сума апаратних характеристик реалізацій функцій  $\psi$ -ої множини,  $\max \{T_\lambda\}$  – найбільший часовий параметр реалізації з усіх функцій  $\psi$ -ої множини.

Відмінність *другого типу формалізації* (рис. 3) полягає в тому, що з множини заданих функцій  $V = \{F_\lambda\}$ ,  $(\forall \lambda = 1 \div s)$  формується множина  $V' = \{F'_\lambda\}$ ,  $(\forall \lambda = 1 \div s)$  – множина оптимально реалізованих функцій, тобто множина з перебору  $(1 \div KR_{ij})$ , яка для конкретної реалізації структури пристрою буде відповідати постановці задачі оптимізації.

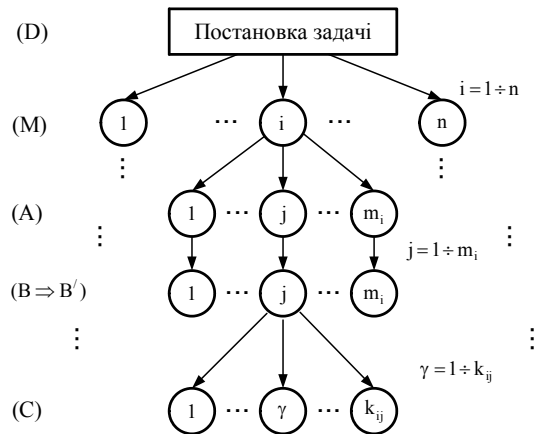


Рис. 3. Формалізація процесу проектування – другий тип

Відображення  $V \Rightarrow V'$  – рішення задачі вибору множини оптимально реалізованих функцій (рис. 4), яке в свою чергу залежить від постановки задачі оптимізації, алгоритму та реалізації структури пристрою.

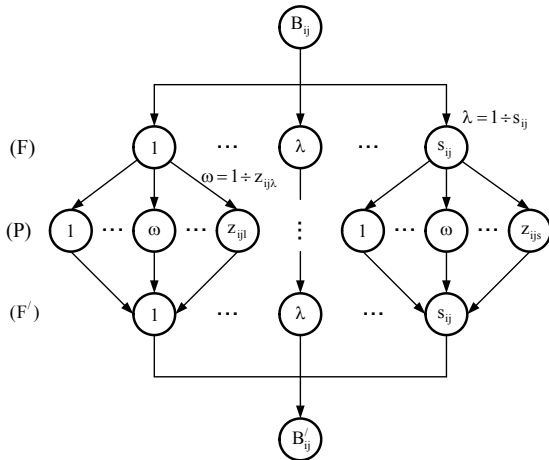


Рис. 4. Задача вибору множини оптимально реалізованих функцій

Рішення задачі синтезу реалізації структури пристрою буде полягати в виборі з набору існуючих  $\{C_{ij\gamma}\}$ ,  $(\forall \gamma = 1 \div k_{ij})$  оптимальної структури, з відміткою що структура  $C_{ij\gamma}$  буде будуватися на множині оптимально реалізованих функцій  $F'$ , а задача вибору множини оптимально реалізованих функцій розглядається окремо.

Розглянемо постановку задачі оптимізації для другого типу формалізації:

$$\alpha \sum_i \sum_j \sum_\gamma q_{ij\gamma} x_{ij\gamma} + \beta \sum_i \sum_j \sum_\gamma t_{ij\gamma} x_{ij\gamma} \Rightarrow \min \quad (7)$$

$$(\forall i = 1 \div n) \quad (\forall j = 1 \div m_i) \quad (\forall \gamma = 1 \div k_{ij}) \quad (8)$$

враховуючи обмеження:

$$\sum_i \sum_j \sum_\gamma q_{ij\gamma} x_{ij\gamma} \leq Q_0; \quad (9)$$

$$\sum_i \sum_j \sum_\gamma t_{ij\gamma} x_{ij\gamma} \leq T_0, \quad \sum_\gamma x_{ij\gamma} = 1.$$

Задача вибору множини оптимально реалізованих функцій полягатиме в переборі з  $KR_{ij}$  комбінацій множин функцій та виборі множини оптимально реалізованих функцій, таким чином, щоб пристрій відповідав задачі оптимізації. Значення  $KR_{ij}$  обчислюватиметься за виразом (1).

### 3. Загальний алгоритм процесу проектування

На рис. 5 наведено блок-схему алгоритму процесу проектування, де: ВР – процедура верифікації,

умови в вершинах 5 та 7 (01) – успішне виконання верифікації, (00), (10) – неуспішне виконання верифікації за двома признаками верифікації.

Алгоритм процесу проектування включає наступні кроки:

1: Постановка задачі включає розробку технічного завдання та формулювання задачі оптимізації параметрів пристрою. Для оцінки коректності, повноти та можливості відповідності постановки задачі до задачі оптимізації (пункти 1.1, 1.2, 1.3, 2.1, 2.2, 2.3).

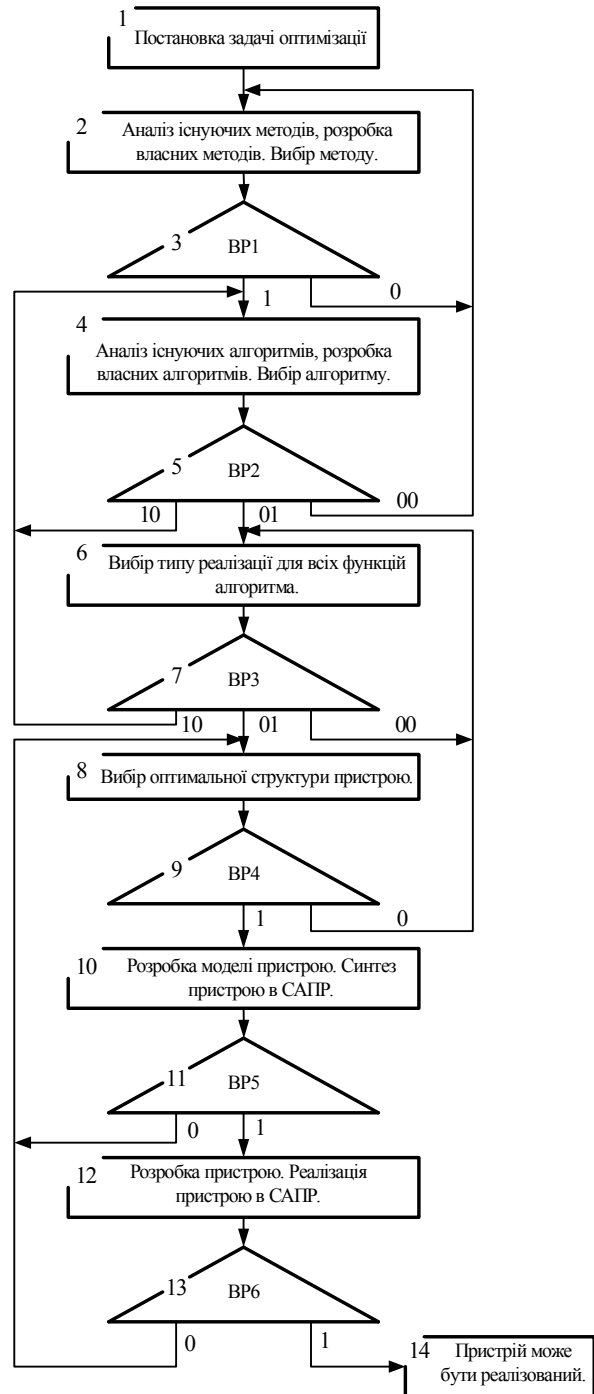


Рис. 5. Блок-схема алгоритму процесу проектування

2: Виконується аналіз існуючих методів розв'язання задачі, що розглядається, в разі відсутності таких, розроблюються власні методи  $M_i \subset M$ , ( $\forall i = 1 \div n$ ). Обирається один з методів  $M_i$ .

3: Перевіряється відповідність набутих внаслідок використання обраного методу похибок до обмежень заданих в обмеженнях задачі оптимізації, використовується верифікація методом технічного огляду документації. В разі невідповідності виконується перехід на вершину 2 ( $i = i + 1$ ). В випадку коли похибки всіх існуючих та власних методів не задовольняють задачі оптимізації, пристрій не може бути реалізовано, в цьому випадку необхідно спрощувати задачу оптимізації.

4: Виконується аналіз існуючих алгоритмів та розробка власних, обирається один з алгоритмів  $A_{ij} \subset A_i$ , ( $\forall j = 1 \div m_i$ ).

5: Перевіряється відповідність набутих внаслідок використання обраного алгоритму похибок до обмежень заданих в задачі оптимізації. В разі невідповідності задачі оптимізації й не всі алгоритми розглянуті, тоді виконується перехід (10) на вершину 4 ( $j = j + 1$ ). В разі невідповідності та розглянуті всі алгоритми виконується перехід (01) на вершину 2 для вибору іншого методу ( $i = i + 1$ ).

6: Виконується аналіз функцій, що використовуються в обраному алгоритмі, оскільки джерел одержання функцій може бути досить велика кількість для кожної з функцій матимемо одну або більше реалізацій.

7: В разі відсутності хоча б однієї функції, що використовується в алгоритмі, відбувається перехід на вершину 4 (10) для вибору іншого алгоритму ( $j = j + 1$ ). Якщо обрані реалізації функцій не можуть бути використана для даного пристрою (00), внаслідок різних причин, обирається інша реалізація даної функції ( $\psi = \psi + 1$ ).

8: Обирається або розробляється структура пристрою.

9: В разі неможливості побудувати структуру пристрою на обраних реалізаціях функцій, обираються нові реалізації ( $\psi = \psi + 1$ ).

10: Виконується реалізація поведінкової моделі пристрою та синтез пристрою засобами САПР.

11: Перевіряється функціональність пристрою методом поведінкового моделювання.

12: Виконуються етапи: компонування та трасування пристрою (MAP, Place and Route, Trace).

13: Перевіряється функціональність пристрою методом моделювання з реальними затримками. Отримані характеристики перевіряються на відповідність обмежень постановки задачі. В разі невідповідності виконується перехід в вершину 8

( $\gamma = \gamma + 1$ ). Якщо ж підтверджена функціональність та характеристики пристрою відповідають постановці задачі оптимізації, то проект пристрою вважається реалізованим.

#### 4. Віртуальна лабораторія з проектування обчислювальних систем на ПЛІС

Віртуальна лабораторія «Проектування обчислювальних систем на основі ПЛІС» – це програмно-апаратний комплекс для колективної розробки і функціональної верифікації проектів пристроїв комп'ютерної техніки на елементній базі ПЛІС з загальною бібліотекою конструкторських компонентів IP-Core [6, 7].

Віртуальна лабораторія використовується для створення сумісних віртуальних лабораторій – "collaboratories". Collaboratories – це відрита металабораторія, яка об'єднує множину географічних областей, де співробітники взаємодіють за допомогою електронних засобів – «працюють сукупно на відстані». Collaboratories спроектовано для: підтримки взаємовідношень між науковцями у даній галузі досліджень; допомоги в співпраці шляхом залучення науковців із різних галузей науки; прискорення розвитку та розповсюдження базових знань; мінімізації часу затримки між винаходом й застосуванням. Подолання перешкоди дистанції дозволить регулярне й щільне партнерство головних академічних та дослідних науковців у плануванні й виконанні складних експериментів. Ці умови є типовою щоденною рутинною сьогоденних дослідників, які повинні користуватися обладнанням великої складності для виконання експериментів, зустрічатися з колегами для планування нових експериментів.

Головними напрямками дослідження та розробок цієї лабораторії є: мови опису типу VHDL; розробка систем реального часу на основі технології графічного програмування (Lab View та ін.); on-line експерименти в Інтернет середовищі на базі технології "reconfigurable computing"; on-line проектування і моделювання цифрових структур на сучасних кристалах ПЛІС для різних застосувань, включаючи побудову інтерфейсів, цифрову обробку сигналів, динамічні системи керування, спеціалізовані пристрої й ін.

Загальну структуру віртуальної лабораторії наведено на рис. 6.

Пакет ChipScope Pro, розроблений фірмою Xilinx, складається з трьох модулів: ChipScope Pro Core Generator, ChipScope Pro Core Inserter й ChipScope Pro Analyzer, що забезпечують перевірку кристалів фірми Xilinx. Взаємодія host-комп'ютера і кристала, що верифікується, здійснюється через порт JTAG

(IEEE Std. 1149.1). Верифікації проектів та процес моделювання на базі типових проектів виконується в середовищах Simulink Matlab й ModelSim. Simulink Matlab виконує лише моделювання математичної моделі функції, яка засобами System Generator може бути представлена вихідною мовою ISE Foundation. Система ModelSim виконує моделювання моделі пристрою (функціональне та з реальними затримками).

Платформа LabVIEW надає єдине універсальне середовище розробки для програмування різнотипних вузлів. Використання середовища розробки LabVIEW дозволяє створити код, що може викону-

ватися на таких обчислювальних платформах, як персональні комп'ютери, пристрої реального часу, пристрої і підсистеми на базі ПЛІС. У єдиній оболонці LabVIEW поєднує специфічні функції для рішення зовсім різнотипних задач, наприклад, функції розпізнавання образів і класифікації об'єктів відеоконтролю, вимір аналогових і цифрових сигналів. Традиційно кожна із цих задач вимагала застосування окремих спеціалізованих програмних продуктів. Платформа LabVIEW містить бібліотеку розширеного аналізу сигналів, а також розвинені засоби комунікації з Інтернетом для віддаленого керування і моніторингу.



Рис. 6. Загальна структура віртуальної лабораторії

Таким чином, платформа LabVIEW, будучи ефективною та простою у використанні оболонкою для проектування, керування, запуску та синхронізації розподілених систем, дозволяє об'єднати окремі контрольно-вимірювальні системи в розподілені системи більш високого рівня з повною інтеграцією обчислювальних і керуючих ресурсів.

## Висновки

Розроблена методологія проектування високопродуктивних комп'ютерних систем дозволяє реалізувати структуру проблемно-орієнтованого при-

строю з довільною розрядністю з урахуванням обмежень на логічну ємність, кількість виводів та інші параметри кристалу. реалізувати нові проблемно-орієнтовані обчислювальні системи для обчислень з підвищеною точністю.

Запропоновано два методи формалізації процесу проектування, які призначені для опису процесу розробки цифрового пристрою на ПЛІС та вибору одного варіанту структурної реалізації шляхом перебору відповідно до однієї з постановок задач оптимізації. Перший метод розглядає всі варіанти структурних реалізацій, тобто для кожної з  $C_{ij}$

структури  $KR_{ij}$  варіантів. В другому методі зменшується кількість структурних реалізацій для рішення задачі синтезу структури, за рахунок рішення задачі вибору множини оптимально реалізованих функцій. Розроблена блок-схема алгоритму процесу проектування на ПЛІС, що відповідає формалізації процесу проектування, дозволяє встановити реалізацію пристрою за заданими обмеженнями параметрів або за постановкою задачі оптимізації.

Розроблена структура віртуальної лабораторії «Проектування обчислювальних систем на основі ПЛІС» дозволяє організувати роботу колективу проектувальників пристроїв на ПЛІС з сумісним ресурсом (бібліотекою проектних рішень, що включає реалізації ряду арифметичних функцій та систем) та обмінюватися результатами роботи через мережу Internet.

### Література

1. Estrin G. Organization of computer system: the fixed plus variable structure computer / G. Estrin // Proc. Western Joint Computer Conf. – 1960. – N5. – P. 33-40.

2. Палагин А.В. Реконфигурируемые вычислительные системы / А.В. Палагин, В.Н. Опанасенко. – К.: Просвіта, 2006. – 295 с.

3. Опанасенко В.Н. Высокопроизводительные реконфигурируемые компьютеры на базе FPGA / В.Н. Опанасенко // Проблемы информатизации та управління: Зб. наукових праць НАУ. – Вып. 3 (27). – Київ, 2009. – С. 114-118.

4. Капитонова Ю.В. Математическая теория проектирования вычислительных систем / Ю.В. Капитонова, А.А. Летичевский. – М.: Наука. Гл. ред. физ.-мат. лит, 1988. – 296 с.

5. Сергиенко И.В. Задачи дискретной оптимизации. Проблемы, методы решения, исследования / И.В. Сергиенко, В.П. Шило. – К.: Наук. думка, 2003. – 261 с.

6. Палагин А.В. Проектирование реконфигурируемых систем на ПЛИС / А.В. Палагин, В.Н. Опанасенко, А.Н. Лисовый // Технология и конструирование в электронной аппаратуре. – 2007. – №3. – С. 15-19.

7. Опанасенко В.М. Реалізація проблемно-орієнтованих цифрових пристроїв на кристалах FPGA / В.М. Опанасенко, О.М. Лисовий // Радіоелектронні і комп'ютерні системи. – 2009. – № 5. – С. 176-183.

Надійшла в редакцію 5.02.2010

**Рецензент:** д-р техн. наук, ст.н.с., провідний науковий співробітник Н.І. Алішов, Інститут кібернетики ім. В.М. Глушкова, Київ.

### ДВА ПОДХОДА К ФОРМАЛИЗАЦИИ ПРОЦЕССА ПРОЕКТИРОВАНИЯ ПРОБЛЕМНО-ОРИЕНТИРОВАННЫХ УСТРОЙСТВ

**В.Н. Опанасенко, А.Н. Лисовый**

В работе предложены два подхода к формализации процесса проектирования проблемно-ориентированных устройств: первый представлен прямонаправленным графом, второй – графом с подзадачей выбора множества оптимально реализованных функций. Разработан общий алгоритм процесса проектирования на ПЛИС, который соответствует процессу формализации, позволяет выбрать одну из множества разработанных реализаций устройств в зависимости от заданных ограничений по параметрам либо в зависимости от постановки задачи оптимизации. Предложена структура виртуальной лаборатории по проектированию вычислительных систем на основе ПЛИС.

**Ключевые слова:** ПЛИС, FPGA, виртуальная лаборатория, проблемная ориентация, реконфигурируемые устройства.

### TWO METHODS OF FORMALIZATION OF TASK-ORIENTED DEVICES DESIGN PROCESS

**V.M. Opanasenko, O.M. Lisovyi**

Two methods of formalization of task-oriented devices design process are proposed. The first method produced as a straight directional graph, second produced as a graph with subtask of selection of range of optimally embodied functions. It's proposed general design process algorithm of PLD-based device. It allows selecting one of the sets of the designing implementation of devices by parameters limitation or target setting of optimization. The structure of virtual laboratory for designing of PLD-based computer systems is proposed.

**Key words:** PLD, FPGA, Virtual laboratory, problem alignment, reconfigurable devices.

**Опанасенко Володимир Миколайович** – д-р техн. наук, ст.н.с., провідний науковий співробітник Інституту кібернетики ім. В.М. Глушкова, Київ, Україна, e-mail: vlopanas@ukr.net.

**Лисовий Олександр Миколайович** – молодший науковий співробітник Інституту кібернетики ім. В.М. Глушкова, Київ, Україна, e-mail: Lan-Sasha@yandex.ru.