

УДК 004.232, 004.274, 004.318

**М.О. БЕРЕЗОВСЬКИЙ, Р.Б. ДУНЕЦЬ**

*Національний університет «Львівська політехніка», Україна*

## **ОПТИМІЗАЦІЯ ФОРМАТУ МАШИННИХ КОМАНД ДИНАМІЧНО РЕКОНФІГУРОВАНОГО ПРОЦЕСОРНОГО ЯДРА**

*Представлений підхід, який може бути застосований для скорочення розрядності машинної команди динамічно реконфігурованого процесорного ядра при збереженні гнучкої структури зв'язків між функціональними блоками. Показаний один із можливих способів оптимізації формату машинної команди динамічно реконфігурованого процесорного ядра. Враховуючи низьку ймовірність одночасного використання значної кількості зв'язків, досягнуто суттєвого скорочення розрядності машинної команди відповідно до кількості зв'язків між функціональними блоками ядра процесора, що веде до підвищення швидкості зчитування машинних команд, а також зменшення розмірів виконавчих файлів.*

**Ключові слова:** машинні команди реконфігурованого процесора, динамічна реконфігурація ядра процесора, реконфігурований процесор.

### **Вступ**

Одним із шляхів забезпечення гарантоздатності та підвищення надійності комп'ютерних систем є застосування НВІС.

З одного боку, підвищення ступеня інтеграції само по собі забезпечує підвищення надійності апаратури, а, з другого боку, – достатня кількість апаратних засобів дає змогу реалізувати ефективні алгоритми та методи побудови гарантоздатних та високо надійних комп'ютерних систем.

Для створення високо продуктивних та високо надійних комп'ютерних систем застосовують одночасно два відомі шляхи – конвеєризація та паралелізація [1 – 3]. Серед методів паралелізації виразно виділяються та інтенсивно вдосконалюються методи побудови реконфігурованих комп'ютерних систем [4 – 6].

Стосовно того, в які моменти часу проводиться переналаштування, комп'ютерні системи поділяються на дві групи. До першої групи відносяться ті комп'ютерні системи, у яких пере налаштування проводиться лише перед розв'язанням завдань і до кінця їх виконання налаштування залишається незмінним.

До другої групи відносяться ті комп'ютерні системи, у яких пере налаштування може проводитися безпосередньо у процесі розв'язання завдань, причому багатократно.

Такі системи отримали назву динамічно реконфігурованих систем [7 – 12].

Значних результатів у побудові динамічно реконфігурованих комп'ютерних систем, що реалізу-

ються на основі FPGA, добилася фірма Xilinx, яка налагодила випуск FPGA, починаючи з Virtex, Spartan-2 та закінчуючи останніми моделями, що дають змогу проводити реконфігурацію логічної структури розроблених систем [13 – 15].

Стосовно структурних елементів комп'ютерних систем, які можуть розглядатися як динамічно реконфігурованими, розглядають в основному процесори чи процесорні ядра.

Процесорне ядро з можливістю гнучкої реконфігурації внутрішньої структури може забезпечити значне підвищення швидкодії виконання алгоритмів завдяки оптимізації цієї структури під конкретний алгоритм для максимально ефективного використання ресурсів ядра.

Сучасні дослідження у сфері реконфігурованих процесорів [5, 6, 17] в основному зосередженні на розробці систем, у яких поряд із процесором з класичною (нереконфігурованою) архітектурою (найчастіше RISC або VLIW-процесор), розміщуються реконфігуровані блоки без самостійного пристрою керування [5]. Конфігурація такого ядра здійснюється відповідною групою машинних команд основного (нереконфігурованого) процесора.

Іншим подібним випадком реконфігурованих процесорів є розміщення поряд із стандартним арифметико-логічним пристроєм реконфігурованих акселераторів [6] – комірок, подібних за структурою до вентилів FPGA, для використання яких передбачені відповідні команди у наборі машинних команд. Крім того, здійснюються спроби використання інших реконфігурованих елементів у ядрі процесора, зокрема регістрового файлу [17].

## 1. Динамічно реконфігуроване процесорне ядро

Розглянута у попередніх дослідженнях структура реконфігурованого процесорного ядра [18], у порівнянні з іншими реконфігурованими ядрами включає в свою структуру й пристрій керування, завдяки якому такий процесор може самостійно виконувати програми, без підключення його до іншого процесора з класичною архітектурою.

Структурна схема такого процесора склада-

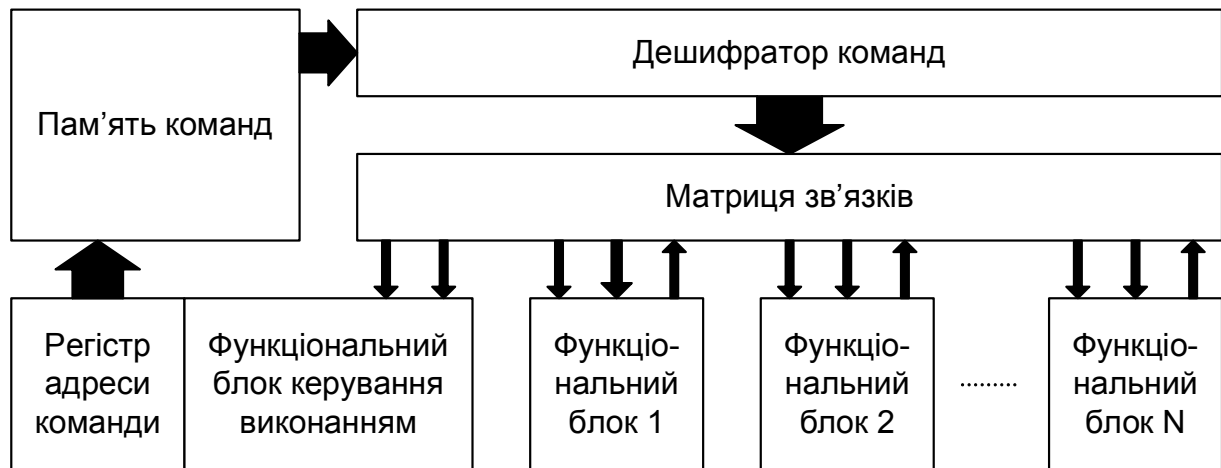


Рис. 1. Структурна схема динамічно реконфігурованого процесорного ядра

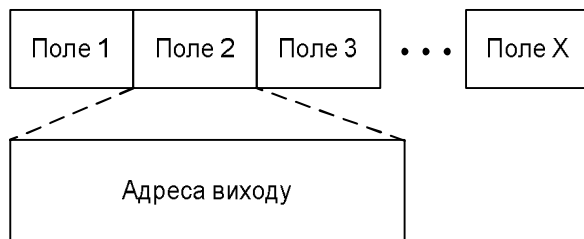


Рис. 2. Структура машинної команди до оптимізації

Загальна кількість розрядів регістру матриці зв'язків у такому випадку обчислюється за формулою  $p = X \log_2 Y$ , де  $X$  - сумарна кількість входів всіх функціональних блоків,  $Y$  - сумарна кількість виходів всіх функціональних блоків [18]. При цьому, нецілий результат логарифму необхідно заокруглювати до більшого цілого числа.

Основний недолік такої структури є велика довжина машинного слова. Нехай, наприклад, процесор містить 16 функціональних блоків, кожен з яких має лише по 2 входи і виходи. В цьому випадку машинна команда буде мати 160 розрядів. При більшій кількості функціональних пристроїв, а особливо при більшій кількості входів та виходів функціональних блоків кількість розрядів може збільшитися на порядок. Тому актуальною є задача зменшення розрядності слова машинної команди.

ється з набору функціональних блоків та матриці зв'язків між функціональними блоками (рис.1). Для можливості підключення кожного входу функціонального блоку до довільного виходу будь-якого функціонального блоку структура слова машинної команди повинна бути такою, яка наведена на рис. 2.

Кількість полів відповідає кількості входів функціональних блоків, а кількість розрядів кожного поля залежить від кількості виходів функціональних блоків.

## 2. Формат із змінною розрядністю машинних команди

З метою зменшення розрядності пропонується більш гнучкий формат машинної команди (рис. 3).

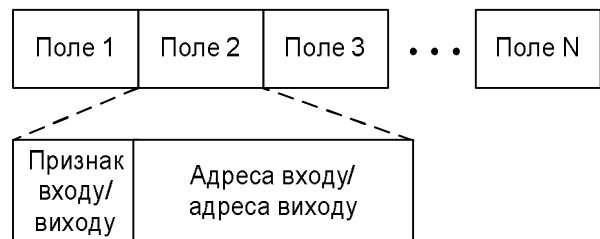


Рис. 3. Структура оптимізованої машинної команди

Цей формат передбачає залежність розміру команди від кількості встановлюваних зв'язків між функціональними блоками. У переважній більшості випадків, для зберігання номеру входу та номеру виходу функціональних блоків використовується однакова кількість біт. Тому для запису команди може бути застосований наступний принцип - машинна команда може складатися з довільної кількості полів ( $N$ ), розрядністю  $1 + \log_2 Y$ . У кожному полі старший біт вказує на те, чи задається номер входу, що відповідає стану «0», чи задається номер виходу,

що відповідає стану «1». Наступні біти відведені власне для коду номера входу чи виходу відповідно. Нумерація входів функціональних блоків загальна (наскрізна) у межах ядра. Те ж саме стосується і нумерації виходів.

Враховуючи те, що до одного входу функціонального блоку може бути підключений лише один вихід функціонального блоку, але в той час до одного виходу може бути підключена довільна кількість входів, то запис таких підключень можна проводити парами – "номер входу - номер виходу". Для позначення кінця машинної команди може слугувати поле, яке містить зарезервоване значення, що не використовується як номер входу чи виходу. Крім того, останнім у машинній команді є поле, що задає вихід, що підключається до останнього за номером входу ( $X - 1$ ).

Наприклад, машинна команда, що підключає вхід з номером 2 до виходу з номером 3, а вхід з номером 4 до виходу з номером 0, буде містити послідовно поля «вхід 2», «вихід 3», «вхід 4», «вихід 0» та «кінець команди».

З метою економії розміру машинної команди, і відповідно, підвищення швидкодії підсистеми завантаження і дешифрування машинних команд, пропонуються способи скорочення кількості полів. Якщо після поля з номером входу послідовно записані декілька полів з номерами виходів, то вказані виходи підключаються по порядку до наступних входів. Наприклад, при послідовному підключенні входів з номерами 2, 3 та 4 відповідно до виходів 3, 8 і 0, машинна команда буде складатися з такої послідовності полів: «вхід 2», «вихід 3», «вихід 8», «вихід 0» та «кінець команди». Якщо у першому полі команди вказаний номер виходу, то він підключається до нульового входу.

Наприклад, для підключення входу 0 до виходу 3, достатньо команди з послідовністю полів «вихід 3» та «кінець команди».

Максимальна кількість розрядів машинної команди з такою організацією становить  $X(1 + \log_2 Y)$ .

На рис. 4 поданий варіант функціональної схеми дешифратора для даного формату машинних команд.

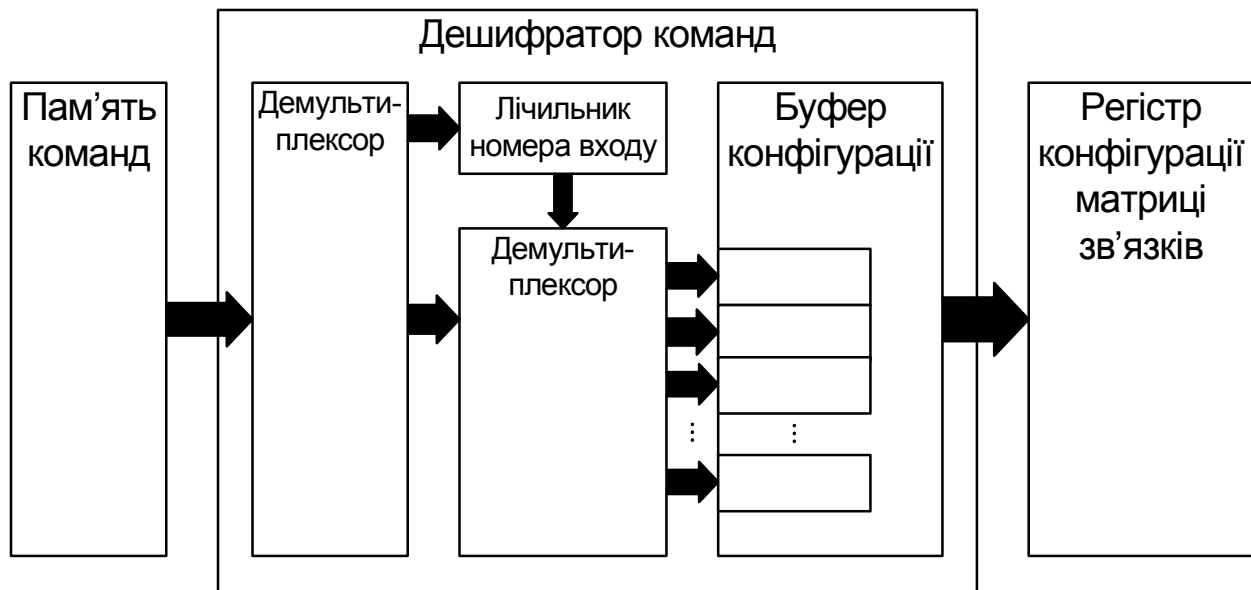


Рис. 4. Дешифратор машинних команд

### 3. Принципи формування набору машинних інструкцій

При формуванні набору машинних інструкцій однією із ключових задач у подібних системах є проблема нумерації входів та виходів функціональних блоків.

Послідовна нумерація входів та виходів з довільною їх послідовністю допускається, проте такий підхід може спричинити до значного погіршення швидкодії процесорного ядра внаслідок збільшення розрядності середньостатистичної машинної команди.

Враховуючи те, що під час використання послідовних за нумерацією входів, номер входу не вказується явно, а інкрементується автоматично, що дає економію в одне поле машинної команди, тому слід намагатися нумерувати входи підряд тоді, коли вони часто використовуються разом в одній машинній команді. Зокрема, у командах, що містять умовний перехід, досить часто при їх реалізації виникає потреба в одночасному здійсненні разом з ними виконання порівняння чисел та логічних функцій.

При визначенні того, якому входу слід призначити нульовий номер, також слід керуватися подібними міркуваннями: оскільки перед входом з

нульовим номером не потрібно ставити поле машинної команди з номером, слід використовувати початкові номери для тих функціональних блоків, що використовуються найчастіше.

В залежності від конкретної спеціалізації системи можуть застосовуватися різні типи функціональних блоків.

Так, в універсальних системах, таких як персональний комп'ютер, це може бути функціональний блок операції цілочисельного віднімання, оскільки ця операція використовується не тільки для віднімання, але і для порівняння чисел, а остання використовується часто у зв'язку з виконанням умовних переходів.

У спеціалізованих системах це можуть бути інші функціональні блоки: блоки логічних функцій, блок інтерфейсу пам'яті, чи будь-який вузькоспеціалізований блок.

## Висновки

У даній роботі показаний один із можливих способів оптимізації формату машинної команди динамічно реконфігурованого процесорного ядра.

Оптимізована машинна команда може мати розмір в межах від  $1 + \log_2 Y$  до  $X(1 + \log_2 Y)$ , в залежності від кількості зв'язків, що встановлюються між функціональними блоками.

Враховуючи низьку ймовірність одночасного використання значної кількості зв'язків, досягнуто суттєвого скорочення розрядності машинної команди відповідно до кількості зв'язків між функціональними блоками ядра процесора, що веде до підвищення швидкості зчитування машинних команд, а також зменшення розмірів виконавчих файлів.

## Література

1. Мельник, А. Питання побудови високопродуктивних процесорів паралельної архітектури [Текст] / А. Мельник // Матеріали 5-ої Міжнародної науково-технічної конференції ACSN-2011. - Львів, НВФ «Українські технології», 2011. - С. 253 – 254.
2. Мельник, В. Тенденції розвитку персональних суперкомп'ютерів [Текст] / В. Мельник // Матеріали 5-ої Міжнародної науково-технічної конференції ACSN-2011. - Львів, НВФ «Українські технології», 2011. - С. 10 – 13.
3. Рак, Ю.П. Проектування технологічних ліній оперативної поліграфії: системний підхід [Текст] / Ю.П. Рак, Р.Б. Дунець. - Дрогобич: НВЦ "Каменярь" ДДПУ, 2002. - 112 с.
4. Палагин, А.В. Реконфигурируемые вычислительные системы: Основы и приложения [Текст] / А.В. Палагин, В.Н. Опанасенко. - К.: Просвита, 2006. - 280 с.
5. Low Power Coarse-Grained Reconfigurable Instruction Set Processor [Text] / F. Barat, M. Jayapala, T.V. Aa, G. Deconinck, et al. // Proc of 13th International Conference on Field Programmable Logic and Applications FPL. - 2003. - P. 230 – 239.
6. Iqbal, M.A. Computational Unit Design for High Speed Reconfigurable Processors [Text] / M.A. Iqbal, S.A. Khan, U.S. Awan // International Journal of Intelligent Information Technology Application. - 2009. - V. 2, No. 5. - P. 229 – 236.
7. Dynamic partial reconfiguration on FPGA [Електронний ресурс]. - Режим доступу: <http://www.vlsi-world.com/content/view/48/47>. - 17.03.2011.
8. Mermoud, Grégory. A Module-Based Dynamic Partial Reconfiguration Tutorial [Text] / Grégory Mermoud. - Ecole Polytechnique Fédérale de Lausanne. - 2004.
9. Gokhale, M.B. Reconfigurable Computing: Accelerating Computation with Field-Programmable Gate Arrays [Text] / M.B. Gokhale, P.S. Graham // Springer. - 2005.
10. Sheikh Secure Partial Reconfiguration of FPGAs, George Mason University, Fairfax, VA, Summer 2005 [Electronic resource]. - Access mode: [http://ece.gmu.edu/courses/Crypto\\_resources/web\\_resources/theses/GMU\\_theses/Zeineddini/Zeineddini\\_Summer\\_2005.pdf](http://ece.gmu.edu/courses/Crypto_resources/web_resources/theses/GMU_theses/Zeineddini/Zeineddini_Summer_2005.pdf). - 17.03.2011.
11. Thorvinger, Jens. Dynamic Partial Reconfiguration of an FPGA for Computational Hardware Support [Text] / Jens Thorvinger. - Lund Institute of Technology. - 2004.
12. Custodio, Evan. Self-Healing Partial Reconfiguration of an FPGA [Text] / Evan Custodio, Brian Marsland // Worcester polytechnic institute, A Major Qualifying Project Report. - 2007.
13. Kao, Cindy. Benefits of Partial Reconfiguration [Electronic resource] / Cindy Kao. - Access mode: [http://www.xilinx.com/publications/archives/xcell/Xcell\\_55.pdf](http://www.xilinx.com/publications/archives/xcell/Xcell_55.pdf). - 17.03.2011.
14. Дунець, Р.Б. Проблеми побудови частково реконфігурованих систем на ПЛІС [Текст] / Р.Б. Дунець, Д.Я. Тиханський // Радіoeлектронні і комп'ютерні системи. - 2010. - Т.48, № 7. - С. 200 – 204.
15. Дунець, Р.Б. Дослідження часткової реконфігурації ПЛІС [Текст] / Р.Б. Дунець, Д.Я. Тиханський // Радіoeлектронні і комп'ютерні системи. - 2009. - № 6 (40). - С. 240 – 244.
16. Березовський, М.О. Динамічно реконфігуроване процесорне ядро [Текст] / М.О. Березовський, Р.Б. Дунець // Матеріали 5-ої Міжнародної науково-технічної конференції ACSN-2011. - Львів, НВФ «Українські технології», 2011. - С. 204 – 205.
17. Wong, S. Dynamically reconfigurable register file for a softcore VLIW processor [Electronic resource] / S. Wong, F. Anjam, F. Nadeem // In Proc. DATE, 2010. - 2010. - P. 969 – 972. - Access mode: [http://www.date-conference.com/proceedings/PAPERS/2010/DATE10/PDFFILES/IP3\\_10.PDF](http://www.date-conference.com/proceedings/PAPERS/2010/DATE10/PDFFILES/IP3_10.PDF). - 17.03.2011.

18. Березовський, М.О. Динамічно реконфігуроване процесорне ядро [Text] / М.О. Березовський, Р.Б. Дунець // Матеріали 5-ої Міжнародної науково-

технічної конференції ACSN, 2011 - Львів, НВФ «Українські технології», 2011. – Р. 204 – 205.

Надійшла до редколегії 26.12.2011

**Рецензент:** д-р техн. наук, проф. Ю.П. Кондратенко, Чорноморський державний університет ім. Петра Могили, Миколаїв, Україна.

#### ОПТИМИЗАЦИЯ ФОРМАТА МАШИННЫХ КОМАНД ДИНАМИЧЕСКИ РЕКОНФИГУРИРОВАННОГО ПРОЦЕССОРНОГО ЯДРА

*М.О. Березовский, Р.Б. Дунец*

Представленный подход, который может быть применен для сокращения разрядности машинной команды динамически реконфигурированного процессорного ядра при сохранении гибкой структуры связей между функциональными блоками. Показан один из возможных способов оптимизации формата машинной команды динамически реконфигурируемого процессорного ядра. Учитывая низкую вероятность одновременного использования значительного количества связей, достигнуто существенного сокращения разрядности машинной команды в соответствии с количеством связей между функциональными блоками ядра процессора, который ведет к повышению скорости считывания машинных команд, а также уменьшению размеров исполняемых файлов.

**Ключевые слова:** машинные команды реконфигурированного процессора, динамическая реконфигурация ядра процессора, реконфигурированный процессор.

#### INSTRUCTION FORMAT OPTIMIZATION OF DYNAMICALLY RECONFIGURABLE PROCESSOR CORE

*M.O. Berezowsky, R.B. Dunets*

Presented approach, which can be applied to reduce instruction size in the dynamically reconfigurable processor core, saving flexible connections structure between the functional blocks. One of possible methods of optimization of format of computer instruction of dynamically reconfiguration processor kernel is retined. Taking into account low probability of the simultaneous use of far of connections, substantial reduction of bit of computer instruction is attained in accordance with the amount of connections between the function boxes of kernel of processor, which conduces to the increase speed of read-out of computer instructions, and also diminishing of sizes of executive files.

**Key words:** reconfigurable processor instructions, dynamical reconfiguration of processor core, reconfigurable processor.

**Березовський Маріян Орестович** – аспірант кафедри ЕОМ Національного університету “Львівська політехніка”, e-mail: mar12@ukr.net.

**Дунець Роман Богданович** – д-р техн. наук, проф., зав. кафедри СКС Національного університету “Львівська політехніка”, e-mail: dunets@polynet.lviv.ua.