

Системы телекоммуникации, связи и защиты информации

УДК 004.27

І.А. Клименко, канд. техн. наук, **В.В. Ткаченко**, канд. техн. наук, **О.М. Сторожук**
Національний технічний університет України «Київський політехнічний інститут»,
Проспект Перемоги, 37, м. Київ, 03056, Україна.

Засоби адаптивного відображення задач на реконфігуровану обчислювальну структуру в паралельних обчислювальних системах, що керуються потоком даних

Запропоновано засоби організації адаптивного відображення завдань на обчислювальну структуру, яка реконфігурується в процесі обчислень. Для підвищення ефективності реконфігурованих обчислювальних систем запропоновано нова концепція відображення завдань на апаратному рівні системи на базі моделі управління потоками даних. Запропоновано та реалізовано ефективний підхід до скорочення часу реконфігурованих обчислень за рахунок завчасної реконфігурації. Розроблена та досліджена програмна модель запропонованих засобів адаптивного відображення алгоритму на реконфігуровану обчислювальну структуру на ПЛІС. Бібл. 7, рис. 4.

Ключові слова: реконфігуровані обчислювальні системи; реконфігуровані архітектури; накладні витрати реконфігурації; моделювання; прискорення реконфігурації; ПЛІС.

Вступ

В останні роки поява динамічно перепрограмованих ПЛІС [1 - 3] створило передумови і нові можливості для підвищення ефективності паралельних обчислювальних систем за рахунок можливості створення гнучких обчислювальних структур, які мають властивість адаптуватися до вимог вирішуваних задач. Такі рішення актуальні на сьогодні в рамках парадигми реконфігурованих обчислень і претендують на досягнення лінійного приросту користувальницької ефективності в широких класах вирішуваних задач, що з одного боку характерно задачеорієнтованим обчислювальним системам, а з іншого боку забезпечує економічну привабливість і підтримку ефективних технологій паралельного програмування, що характерно універсальним обчислювальним структурам.

Однак динамічно реконфігуровані системи мають певні обмеження, пов'язані функціональними особливостями процесу реконфігурації і з

технологічними обмеженнями ПЛІС. Ці обмеження не дозволяють ефективно використовувати існуючі технології паралельного програмування, а також традиційні методи і засоби підвищення ефективності паралельних обчислень. Слід зауважити, що традиційні технології паралельних обчислень, на відміну від реконфігурованих, орієнтовані на фіксоване обчислювальне середовище, без будь-яких просторових і функціональних обмежень. Вищесказане обґрунтовує актуальність досліджень, їх наукову та практичність доцільність.

У статті запропоновано засоби організації адаптивного відображення задач на обчислювальну структуру, яка реконфігурується в процесі обчислень. Запропоновано спосіб реалізації моделі управління потоками даних в реконфігурованих обчислювальних системах в рамках запропонованої концепції відображення завдань.

Огляд відомих рішень

Велика кількість сучасних досліджень в галузі реконфігурованих обчислень, детальний аналіз яких представлений в попередній роботі авторів [4], спрямовані на розробки алгоритмів планування й відображення задач на обчислювальну структуру та водночас підтримують різні механізми зменшення накладних витрат реконфігурації. Найбільш характерні серед них: повторне використання ресурсів [1], кешування конфігураційних даних [2], випереджуюча реконфігурація [5], апаратні засоби прискорення процесів вводу/виводу [3], оптимізація структури на апаратному рівні [6]. При цьому прискорення реконфігурації забезпечується за рахунок зменшення часу реконфігурації, частіше за все ціною надлишкового використання апаратних ресурсів ПЛІС. Це спричинює великий відсоток відмов виконання задач і додаткові витрати часу й продуктивності на подолання апаратних обмежень ПЛІС.

Більшість з розглянутих та досліджених засобів відображення задач зазвичай є модифікацією традиційних технологій паралельного програмування, що стосуються планування та розподілу задач на обчислювальне середовище [1 – 4]. Традиційні технології розроблені для фіксованих архітектур або комутованих обчислювальних середовищ та функціонують на рівні програмного забезпечення надбудови операційної системи. В динамічно реконфігурованих обчислювальних системах їх застосування призводить до високої обчислювальної складності алгоритмів управління, додаткових витрат продуктивності та реконфігурованих обчислювальних ресурсів, що обумовлено необхідністю врахування фізичних параметрів реконфігурованої обчислювальної структури фізичного рівня абстракції на рівні операційної системи, що негативно впливає на їх ефективність. Вищесказане обґрунтовує невирішені задачі в напрямку рішення проблеми ефективного відображення задач в динамічно реконфігурованих обчислювальних системах, а також необхідність і доцільність виконаних в роботі досліджень.

Постановка задачі

Реалізація моделі управління потоками даних, на відміну від традиційних послідовно-паралельних або паралельно-конвексних обчислень [7], дозволяє підвищити ефективність розпаралелювання на рівні макрокоманд за рахунок реалізації прихованого паралелізму.

На відміну від відомої моделі потокової обчислювальної системи [7] в якості компонентів бібліотеки використовуватися синтезовані на мові програмування апаратури апаратні реалізації функціональних ядер. Таким чином заздалегідь створена бібліотека конфігурацій функціональних блоків зберігається в пам'яті конфігураційних даних. Процес управління відображенням задач в реконфігурованій обчислювальній системі, таким чином, складається з процесів планування послідовності виконуваних задач, в результаті чого визначається чергова задача на виконання; реконфігурація обчислювальної структури для виконання даної задачі, що складається з пошуку і розміщення конфігураційних даних на поверхні ПЛІС (перепрограмування ПЛІС); виконання задачі. При цьому специфіка організації реконфігурованих обчислень потребує врахування фізичних параметрів на всіх рівнях реалізації процесу відображення, включаючи процес планування. Окрім того значною є проблема непродуктивних витрат часу та продуктивності під час здійснення

реконфігурації, пов'язана з необхідністю передавання великих об'ємів конфігураційних даних з віддалених бібліотек і складністю та специфічністю процесу управління реконфігурацією. На підставі чого слід зробити висновок, що відомі методи реалізації управління паралельними обчисленнями та відображенням задач, зокрема і за застосування моделі обчислень під управлінням потоку даних, неефективні в реконфігурованих обчислювальних системах і виникає необхідність їх модернізації та вдосконалення.

Запропонована та детально описана авторами в попередніх роботах [4] концепція ефективного управління обчисленнями в реконфігурованих обчислювальних системах базується на локалізації засобів управління на програмно-апаратному рівні обчислювального модуля. При цьому обґрунтовано, що апаратна реалізація засобів відображення задач на структурному рівні реконфігурованих обчислювальних систем є ефективною парадигмою, за рахунок простоти врахування фізичних параметрів обчислювального середовища на ПЛІС та апаратного прискорення. В рамках даної статті запропоновані апаратні засоби відображення задач для реконфігурованих обчислювальних систем, що базуються на модифікації відомого методу автоматичного динамічного розпаралелювання [7] для моделі обчислень під управлінням потоку даними. Модифікація методу стосується втілення в модель управління потоками даних механізмів зменшення непродуктивних витрат часу реконфігурації.

Математична модель формування заявок

Обчислювальний процес представляється за допомогою макрографу потоку даних MDG (Macro Dataflow Graphs) [5, 6] $G = (W, D)$, де W – множина вершин, D – множина дуг. В вершинах графа розміщуються макрозадачі (М-задачі) $A_i (i = \overline{1, m})$ (завдання, функція, певний обсяг роботи), а кожній дузі D_{ij} – потік даних, необхідних для виконання М-задачі. Вихідні дані для кожної задачі підготовляються на підставі графа і являють собою набір дескрипторів даних, а також бібліотеку реалізації функцій. Детально механізм формування заявок описаний в роботі [7].

Завдання описується актором (множиною елементів). Для i -ої вершини графа актор має вигляд:

$$W_i = \{n_i, N_i, I_i, Q_i, p_i, M_i, R_i\},$$

де n_i – ознака актора; N_i – ім'я актора; I_i – ідентифікатор завдання (команди, програми, процесу); Q_i – сумарна кількість потоків даних для завдання (дуг графа, що входять в вершину); p_i – рівень пріоритету завдання; M_i – множина імен акторів, для яких i -е завдання готує дані; R_i – множина імен потоків даних, які формуються при виконанні i -го завдання.

Потік даних, який відповідає дузі графа, що з'єднує j -у вершину з i -ю вершиною, характеризується дескриптором, що є множиною елементів:

$$D_{ji} = \{n_{ji}, N_j, Q_j, A_{ji}\},$$

де n_{ji} – ім'я (номер) потоку даних (дуги, що входить до i -у вершину); A_{ji} – елемент адресації даних. Всі вершини графа мають унікальні імена $N_i = (i = \overline{1, m})$. Елемент адресації визначає місце розташування даних у пам'яті системи, доступної для кожного процесора. Ім'я потоку одночасно є ознакою, що відрізняє дескриптор даних від актора.

З елементів дескрипторів відповідно до певної процедури F формуються заявки на виконання i -го завдання:

$$F(W_i, D_i) \rightarrow Z_i = \{I_i, p_i, A_i, M_i\},$$

де $A_i = \{A_j \mid j \in J\}$ – множина елементів адресації даних для i -го завдання; $D_i = \{D_{ji} \mid j \in J\}$ – множина дескрипторів даних для завдання W_i ; J – множина імен вершин графа, пов'язаних вихідними дугами з i -ю вершиною; M_i – маска, що визначає, які дескриптори вихідних даних необхідно формувати при повторному виконанні завдання.

З метою зменшення непродуктивних витрат часу в рамках вдосконалення методу реалізованого механізму завчасної реконфігурації. Таким чином, в модифікованій реалізації умова активізації заявки представлена наступним виразом:

$$w_i = r_i \& v_i \& \delta_{ji}, \quad (1)$$

де v_i – умова отримання дескриптора i -го завдання; δ_{ji} – умова отримання елемента адресації r_i – ознака завчасно виконаної реконфігурації на ПЛІС для виконання обчислення чергової задачі.

Структурна модель обчислювального модуля

Вдосконалена структура пристрою автоматичного розподілу задач (ПАРЗ) для обчислень під управлінням потоку даних. Структура вдосконаленої реалізації ПАРЗ зображена на рис. 1. У склад пристрою автоматичного розподілу задач на відміну від відомого введений контролер реконфігурації та буферна пам'ять реконфігурації.

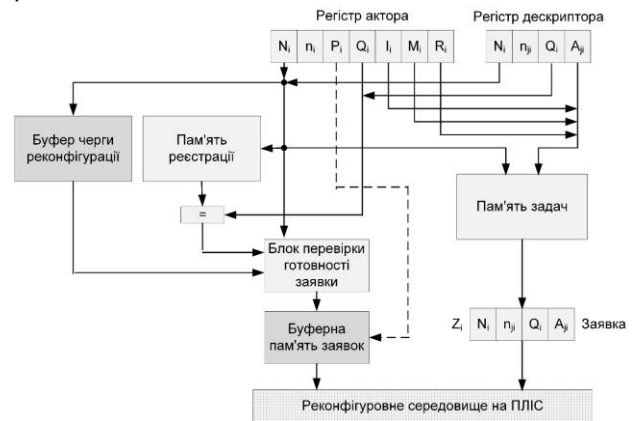


Рис. 1. Структурна модель вдосконаленого пристрою автоматичного розподілу задач

Запропонований підхід дозволяє найбільш ефективно реалізувати послідовність реконфігурації на структурному рівні обчислювального модуля. При цьому наявність контролера реконфігурації дозволяє вдосконалити апаратні засоби автоматичного розпаралелювання завдань, за рахунок розпаралелювання як на рівні макрокоманд так і на рівні процесів управління реконфігурованими обчисленнями. Запропоновані засоби функціонують на локальному рівні обчислювального модуля, і забезпечують управління паралельними алгоритмами і реконфігурованими апаратними ресурсами без участі операційної системи.

Вдосконалений пристрій управління автоматичним розподілом задач розроблений та досліджений на базі ПЛІС Cyclone II EP2C35F672C6 компанії Altera. Використані стандартні засоби проектування систем-на кристалі SOPC Builder. Для реалізації управління процесорів та контролеру реконфігурації застосовані вбудовані процесорні ядра Nios II/s. Пристрій управління автоматичним розподілом задач реалізований як апаратна надбудова процесорного ядра.

Загальне управління потоковими обчисленнями та формування заявок Z_i виконує управляючий процесор. Апаратні обчислювачі, реалізовані на поверхні ПЛІС з точки зору управління розглядаються, як пасивні функціональні елементи. Дані для i -го завдання формуються в різні моменти часу різними апаратними обчислювачами на ПЛІС. Контроль за надходженням даних для виконання чергового завдання виконує пристрій автоматичного розподілу задач.

Завчасна реконфігурація здійснюється наступним чином. Під час формування заявок готових для виконання для вхідного актору, формується черга попередньої реконфігурації із множини M_i , що зберігається в кожному акторі. А під час виконання заявки перевіряється ознака r_i згідно виразу (1) в блоці перевірки готовності заявки (рис. 1).

Розробка самоподібного алгоритму управління реконфігурацією

З метою забезпечення механізму завчасної реконфігурації, при реалізації запропонованого методу відображення в потокових алгоритмах, виникає необхідність визначення критичного шляху графа алгоритму. При статичному аналізі графів можливе використання алгоритмів, заснованих на направленому аналізі часу виконання всіх нащадків вершин графа алгоритму. Однак в динамічному режимі використання механізмів спрямованого перебору може призводити до невизначеним тимчасових витратах при непостійних умовах відображення. Це пояснюється тим, що при скороченні часу виконання задач на критичному шляху, за рахунок попередньої реконфігурації, може призводити до того, що час виконання послідовностей задач на інших шляхах графа ставатиме більше ніж час виконання завдань на критичному шляху.

В роботі запропонований спосіб скорочення критичного часу виконання потокового алгоритму, який заснований на особливості деякої самоподібності потокових алгоритмів, ця особливість дозволила оптимізувати процедуру формування критичного шляху в динамічному режимі за рахунок організації самоподібних функцій організації процесу відображення в потокових алгоритмах. Запропоновано спосіб скорочення критичного часу являє собою модифікацію методу розгалужень і меж, заснований на визначенні часу нащадків поточної задачі з найбільшим часом виконання.

Алгоритмічна модель реалізації самоподібної функції відображення представлена на рис. 2. Застосовуються наступні позначення: T_i – очікуваний час виконання i -ї задачі; $i = \overline{1, q}$, де q – кількість вершин-нащадків батьківської вершини T_g , i_max – індекс задачі, що має максимальний час виконання серед набору вершин-нащадків $Q_g = \{T_i | i = \overline{1, q}\}$; T_D – час виконання обчислювального алгоритму, обрахований за шуканим ефективним маршрутом. Граф обчислювального алгоритму подається в вигляді матриці зв'язності M_D розмірністю $(v \times v)$, де v – кількість вершин графу G_D . Рядки $M_D[g] | g = \overline{1, v}$ відповідають виконуваним задачам. В стовпцях $M_D[s] | s = \overline{1, v}$ кожного рядку $M_D[g]$ розміщується одиниця, якщо задача T_g готує дані для задачі T_s . Під час аналізу кожної вершини графу формується множина всіх батьківських задач які готують дані для задачі T_s , які є елементами стовпця $M_D[s]$, в якому індекси g вказують на шукані батьківські задачі T_g , якщо $M_D[g, s] = 1$. На підставі множини нащадків кожної вершини графу обчислювального алгоритму формується черга завчасної реконфігурації.

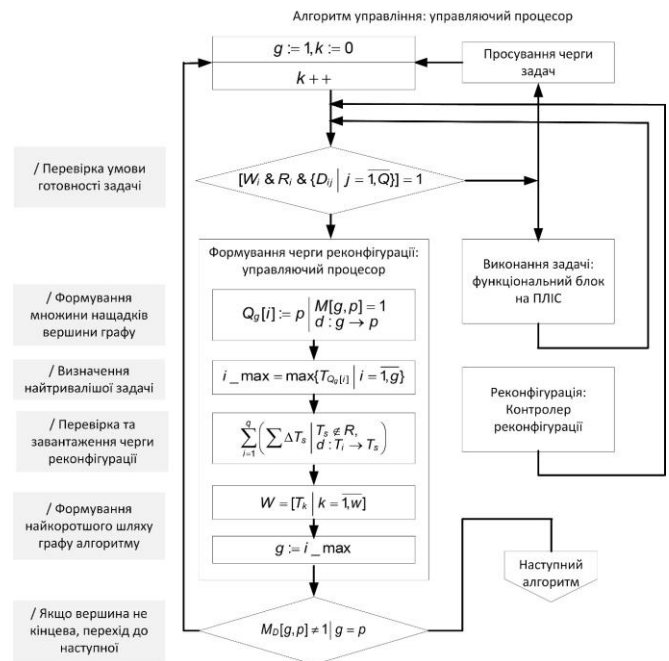


Рис. 2. Алгоритмічна модель самоподібної функції відображення

Моделювання

Для дослідження процесу завчасної реконфігурації розроблена програмна модель методу прискорення реконфігурації та програмний емулятор архітектури реконфігурованої обчислювальної системи. Програмні засоби розроблені на кросплатформенній мові Java. Програмна модель має модульну структуру та включає в себе наступні блоки, що реалізовані як програмні модулі, які емулюють функції відповідних апаратних блоків: DataFlow – головний модуль, що емулює функціонал всієї системи розпаралелювання та об'єднує підлеглі модулі: AppFormingService – модуль що відтворює роботу пристрою автоматичного розпаралелювання задач та включає в себе функціонал контролера реконфігурації; FPGA – реконфігурована область; Core – модуль, що виконує роль програмно-апаратного ядра для обчислення задач та може масштабуватися в налаштуваннях системи; Actor, Descriptor, Task, Application, Data – модулі, що виконують роль відповідних структур в потоковій системі та забезпечують збереження даних в пам'яті.

Для оцінки ефективності запропонованих засобів використовуються коефіцієнт прискорення $K = T_G / T_R$, де T_G – це час виконання потокового алгоритму без будь-якого прискорення реконфігурації, а T_R – час виконання обчислювального алгоритму із застосуванням завчасної реконфігурації на базі запропонованих засобів. Отримані графіки залежності часу виконання обчислювальних алгоритмів з різною кількістю однотипних задач (рис. 3, рис. 4).

Дослідження проводились для серії обчислювальних алгоритмів, поданих поточковими графами. Досліджувались графи алгоритмів з різною кількістю однотипних задач та різною ступеню зв'язності. На підставі проведених експериментів отримано усереднену залежність часу виконання реконфігурованих обчислень для поточкових алгоритмів з прискоренням за рахунок повторного використання реконфігурованих ресурсів (T_G) [5], та – за рахунок застосування завчасної реконфігурації на базі запропонованих засобів (T_R), представлені на рис. 3. Показник прискорення обчислення для алгоритмів з різною кількістю однотипних задач розрахований як $K = T_G / T_R$.

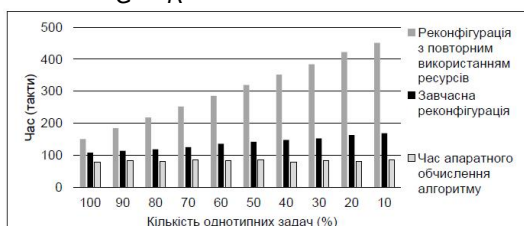


Рис. 3. Діаграма залежності часу реконфігурації від кількості однотипних задач в обчислювальному алгоритмі

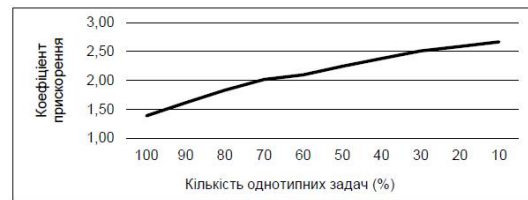


Рис. 4. Діаграма залежності коефіцієнту прискорення від кількості повторів однотипних задач

Механізми повторного використання реконфігурованих ресурсів дозволяють прискорити швидкість реконфігурації в середньому на 63%, що детально досліджено в попередній роботі авторів [5]. При цьому ефективність алгоритмів повторного використання ресурсів залежить від кількості однотипних задач в обчислювальному алгоритмі (рис. 3). За результатами експериментів видно, що завчасна реконфігурація дозволяє видалити практично весь непродуктивний час реконфігурації для будь-яких обчислювальних алгоритмів, незалежно від кількості однотипних задач. Під час дослідження показники прискорення реконфігурованих обчислень за застосування запропонованих засобів з реалізацією завчасної реконфігурації отримано середній коефіцієнт прискорення, що дорівнює 2 (рис. 4).

Висновки

На підставі досліджень можна зробити наступні висновки щодо ефективності та практичної значимості запропонованих засобів відображення задач в обчислювальну структуру в реконфігурованих обчислювальних системах:

- Механізми завчасної реконфігурації підвищують ефективність процесу відображення задач на реконфігуровану структуру у порівнянні з механізмами повторного використання ресурсів, за рахунок видалення практично всіх непродуктивних витрат часу реконфігурації, незалежно від кількості повторів однотипних задач в обчислювальному алгоритмі, що забезпечує інтенсивне прискорення реконфігурації.

- Вдосконалений метод автоматичного розподілу задач на базі моделі обчислень, керованих потоком даних за рахунок реалізації механізму попередньої реконфігурації дозволяє скоротити непродуктивні витрати часу в процесі автоматичного розподілу потоку задач і підвищити ефективність відображення обчислювальних алгоритмів на динамічно реконфігуровану обчислювальну структуру.

Список використаних джерел

1. Bassiri M.M. Mitigating Reconfiguration Overhead In On-Line Task Scheduling For Reconfigurable Computing Systems [Text] / M.M. Bassiri, S.H. Shahriar // Proceeding of 2nd International Conference on Computer Engineering and Technology (IC CET), (Chengdu, 16 – 18 April 2010). – 2010. – Vol. 4. – P. V4 – 397. – V4 – 402.
2. Al-Wattar A. Efficient On-line Hardware/Software Task Scheduling for Dynamic Run-time Reconfigurable Systems [Text] / A. Al-Wattar, S. Areibi, F. Saffih // Proceeding in 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW). – IEEE, 2012. – P 401 – 406.
3. Liu S. Achieving Energy Efficiency through Runtime Partial Reconfiguration on Reconfigurable Systems [Text] / S. Liu, R.N. Pittman, A. Forin, J.-L. Gaudiot // Transactions on Embedded Computing Systems (TECS). – USA, NY, New York, ACM, 2013. – Vol. 12. – P.72:1 – 72:21.
4. Клименко І.А. Аналіз ефективності управління ресурсами та додатками в реконфігурованих обчислювальних системах [Текст] // І.А. Клименко / Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: Зб. наук. пр. – К.: Век+, – 2015. – № 62. – С. 11 – 21.
5. Кулаков Ю.О. Розробка методу прискорення реконфігурації в динамічно реконфігурованих обчислювальних системах [Текст] // Ю. О. Кулаков, І. А. Клименко, М. В. Рудницький 2015 // Східно-Європейський журнал передових технологій. – Харків.: УДА залізничного транспорту. – 2015. – №4/4 (76). – С. 25 – 29.
6. Кулаков Ю.О. Організація багаторівневої пам'яті в реконфігурованих обчислювальних системах [Текст] // Ю.О.Кулаков, І.А.Клименко / Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: Зб. Наук. Пр. – К.: Век+, 2014. – №61. – С. 18 – 26.
7. Жабин В.И. Реализация вычислений под управлением потока дескрипторов данных в мультипроцессорных системах [Текст] // В.И. Жабин / Электронное моделирование. – К.: ИПМЕ, 2003. – Т.25, №1. – С. 35 – 47.

Поступила в редакцию 01 июня 2016 г.

УДК 004.27

І.А. Клименко, канд. техн. наук, **В.В. Ткаченко**, канд. техн. наук, **А.Н. Сторожук**

Национальный технический Университет Украины «Киевский политехнический институт»,
проспект Победы, 37, г. Киев, 03056, Украина.

Средства адаптивного отображения задач на реконфигурируемую вычислительную структуру в параллельных вычислительных системах, под управлением потоком данных

Предложены средства организации адаптивного отображения задач на вычислительную структуру, которая реконфигурируется в процессе вычислений. Для повышения эффективности реконфигурируемых вычислительных систем предложена новая концепция отображения задач на аппаратном уровне системы на базе модели управления потоком данных. Предложен и реализован эффективный подход к сокращению времени реконфигурируемых вычислений за счет упреждающей реконфигурации. Разработана и исследована программная модель предложенных средств адаптивного отображения алгоритма на реконфигурируемую вычислительную структуру на ПЛИС. Библ. 7, рис. 4.

Ключевые слова: реконфигурируемые вычислительные системы; реконфигурируемые архитектуры; накладные расходы реконфигурации; моделирование; ускорение реконфигурации; ПЛИС.

UDC 004.27

I. Klymenko, Ph.D, V. Tkachenko, Ph.D, A. StorozhukNational Technical University of Ukraine "Kyiv Polytechnic Institute",
37, Prospect Peremohy, 03056, Kyiv, Ukraine.

Adaptive tasks mapping tools for reconfigurable computing structure in parallel computing systems under data flow control

The means of adaptive tasks mapping on a computing structure that is reconfigured during the computation was proposed. The new concept of tasks mapping on hardware level that based on a data flow computations for increased effectiveness reconfigurable computer systems was proposed. The effective approach of reducing the execution time of reconfigurable computation by preemptive reconfiguration was proposed and investigated. Program model of proposed means for adaptive tasks mapping on reconfigurable FPGA computing structure. References 7, Fig.4.

Keywords: reconfigurable computing systems; reconfigurable architectures; reconfiguration overhead; modeling; rapid configuration; FPGA.

References

1. Bassiri, M., and Shahriar, H. (2010). Mitigating Reconfiguration Overhead In On-Line Task Scheduling For Reconfigurable Computing Systems. In Proc. of the 2nd International Conference on Computer Engineering and Technology (ICCET), China, Chengdu, V4-397–V4-402.
2. Al-Wattar, A., Areibi, S., and Saffih, F. (2012). Efficient On-line Hardware/Software Task Scheduling for Dynamic Run-time Reconfigurable Systems. In Proc. of the 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), China, Shanghai, 401–406.
3. Liu S., Pittman, R .N., Forin, A., and Gaudiot J.-L. (2013). Achieving Energy Efficiency through Runtime Partial Reconfiguration on Reconfigurable Systems. Transactions on Embedded Computing Systems (TECS), 12, 72:1–72:21.
4. Klymenko, I. A. (2015). The effectiveness analysis of resources management in reconfigurable computer systems. Visnyk NTUU "KPI". Informatyka, upravlinnia ta obchislyvalna technika: Zb. Nauk. Pratsc, no 62, pp. 11 – 21. (Ukr.)
5. Kulakov, Y. O., Klymenko, I. A, and Rudnytskyi, M. V. (2015). Development of the reconfiguration acceleration method in the dynamically reconfigurable computing systems. Vostochno-Evropeyskiy zurnal peredovyh tehnologiy [Eastern-European Journal of Enterprise Technologies], 4/4 (76), 25–29.
6. Kulakov, Y. O., and Klymenko, I. A. (2014). The multilevel memory in the reconfigurable computing system. Visnyk NTUU «KPI». Informatyka, upravlinnia ta obchislyvalna technika: Zb. nauk. pr., 61, 18–26.
7. Zhabin, V. I. (2003). Computation implementation under control of descriptors data flow in multiprocessor systems. Electronic modeling. IPME. Vol.25, no1, pp. 35 - 47.