

ОПТИМІЗАЦІЯ РЕКОНФІГУРАЦІЇ В ДИНАМІЧНО РЕКОНФІГУРОВАНИХ ОБЧИСЛЮВАЛЬНИХ СИСТЕМАХ

Розроблений метод оптимізації процесу реконфігурації, за критеріями забезпечення необхідного додатку часу обчислення та врахування просторових обмежень реконфігурованої структури ПЛІС під час розміщення динамічного потоку задач, що дозволяє підвищити ефективність реконфігурації.

The method of optimization reconfiguration with the criteria providing timing requirements of applications and providing space requirements of reconfigurable field of FPGA when mapping of the dynamic flow of tasks is proposed. The proposed method increases the efficiently reconfiguration in the dynamic reconfigurable computer.

Вступ

В області реконфігурованих високопродуктивних обчислювальних систем [1 - 4] зменшення часу реконфігурації є актуальною проблемою сьогодення. Час виконання реконфігурації є критичним параметром зважаючи на його високу непродуктивну складову, пов'язану з необхідністю передавання великих обсягів конфігураційних даних, технологічними обмеженнями параметрів інтерфейсів обміну даними, складним процесом управління реконфігурованими ресурсами. Витрати часу на реконфігурацію, можуть бути навіть порівняні у часі з апаратним прискоренням, зводячи нанівець весь позитивний ефект від здійснення реконфігурації архітектури. Відомі способи та технології інтенсивного прискорення реконфігурації пропонують максимальне видалення непродуктивних витрат часу незважаючи на просторові обмеження реконфігурованих ресурсів ПЛІС. В той час як механізми подолання цих обмежень, такі як тимчасове вивантаження конфігурацій функціональних блоків, дефрагментація реконфігурованої області, зменшення кількості відмов під час розміщення задач, тощо, потребують додаткових ресурсів часу та продуктивності на їх реалізацію.

За умови виставлених додатком часових обмежень виконання обчислень стає задачею оптимізації процесу прискорення реконфігурації, за рахунок оптимізації обсягу зменшення часових витрат, що дозволить мінімізувати непродуктивне використання обчислювального простору реконфігурованої області настільки, щоб забезпечити виставлені додатком часові вимоги. Ця задача не вирішується відомими методами та засобами прискорення реконфігурації, що обумовлює актуальність та доцільність виконаних в статті досліджень.

Огляд існуючих рішень

Відомо багато методів та засобів зменшення непродуктивних витрат часу реконфігурації, такі як повторне використання обчислювальних ресурсів [3, 5, 6], попередня вибірка конфігурацій [7], кластеризація конфігурацій [2], реконфігурація за принципом «як можна раніш» [7], різноманітні апаратні засоби прискорення процедури обміну даними та збільшення швидкодії інтерфейсів [8]. Всі вони забезпечують максимально можливе прискорення обчислень, за рахунок максимально зменшення накладних витрат часу реконфігурації, частіше за все ціною надлишкового використання як апаратних ресурсів, так і надлишкових витрат продуктивності і енергоспоживання, що призводить до додаткових накладних витрат реконфігурації. Одне з найбільш ефективних із відомих рішень прискорення реконфігурації, що спрямовано на використання необхідного об'єму реконфігурованого обчислювального ресурсу, запропоноване для кластероподібної обчислювальної системи із багатоядерними процесорами в вузлах і розділеної між ними однорідної реконфігурованої структури [4, 9]. Але задіяна реконфігурована структура має передумовлену архітектуру і складається з набору однотипних дрібнозернистих і крупнозернистих модулів, поєднаних між собою загальною комунікаційною мережею. Основна ідея розподілу задач складається з визначення ефективної апаратної структури для реалізації задачі, що дозволить забезпечити необхідний час виконання вирішуваного додатку це дозволяє позбавитись від збитковості використання апаратури забезпечуючи загальні її обмеження. Негативні риси даного підходу складаються в наявності передумовленої архітектури, що вимагає попереднього при-

ведення будь якої вхідної задачі до визначної форми, та реалізації спеціалізованого програмного проширення із залучанням вузькопрофільних спеціалістів для їх підтримки та експлуатації. Проблематика зменшення накладних видатків не розглядається, фактично задача вирішується абстраговано від фізичного процесу реконфігурації і зводиться до класичної концепції апаратного прискорення.

На підставі огляду відомих рішень проблеми прискорення реконфігурації в сучасних реконфігурованих обчислювальних системах визначено необхідність вдосконалення відомих, або розробки нових, методів та засобів прискорення реконфігурації, які забезпечують необхідний додаток часу обчислення та мінімізуючи при цьому витрати обчислювального ресурсу ПЛІС.

Постановка задачі

В якості критерію оцінки ефективності функціонування реконфігурованої обчислювальної системи приймають наступний показник прискорення:

$$\rho = \frac{T_{SW}}{T_{Rconf} + T_{HW}} \quad (1)$$

де T_{SW} – час обчислення задачі на процесорному ядрі, T_{HW} – час обчислення задачі апаратними засобами, T_{Rconf} – час реконфігурації, витрачений на розміщення конфігурації відповідної апаратної реалізації задачі на поверхні реконфігурованої обчислювальної структури. Тоді проблема прискорення реконфігурації може бути вирішена за рахунок зменшення часу реконфігурації, що формально виражається наступним чином:

$$\min(T_{Rconf}) = \min(T_{Contr}) + \min(T_{Comm}) + T_{Mapp} \quad (2)$$

Час T_{Rconf} в виразах (1) і (2) характеризує обсяг непродуктивних видатків часу реконфігурації і складається з наступних складових, відповідно виразу (2): тривалості циклу управління процесом реконфігурації $T_{control}$ – обумовлює витрати продуктивності; часу передавання конфігураційних даних T_{comm} – обумовлює комунікаційні витрати; часу прошивання мікросхеми T_{mapp} – залежить від технологічних особливостей реконфігурованої елементної бази. На підставі визначення (2) в роботі [10] запропонований метод інтенсивного прискорення реконфігурації (МІПР), який забезпечує мінімізацію непродуктивних видатків часу реконфігурації ($T_{Rconf} \rightarrow \min$), за рахунок максимального вида-

лення непродуктивних комунікаційних витрат ($T_{comm} \rightarrow \min$) шляхом повторного використання апаратних ресурсів функціональних блоків. Ефективність запропонованого методу МІПР залежить від просторових параметрів реконфігурованої області. За невідповідності розміру реконфігурованої області і кількості розміщуваних апаратних задач процес реконфігурації потребує додаткового часу, який негативно впливає на загальний час обчислень і часто призводить до зворотного ефекту від прискорення реконфігурації.

В реальних реконфігурованих обчислювальних системах стає задачею оптимізації реконфігурації, яка може бути вирішена за рахунок оптимізації обсягу видалення непродуктивних витрат часу, що дозволить мінімізувати використання обчислювальних ресурсів ПЛІС, забезпечуючи затребувані вирішуваним додатком обмеження часу виконання.

Проблема оптимізації реконфігурації

Геометричне обґрунтування оптимізації реконфігурації за критеріями забезпечення обмежень часу обчислення і просторових обмежень реконфігурованої області ПЛІС зображено на рис. 1.

Непродуктивні видатки часу на здійснення реконфігурації визначаються складовою T_{Rconf} виразу (1). Коефіцієнт реконфігурації K_{Rconf} характеризує обсяг виконаної реконфігурації з ціллю збільшення степені адаптації обчислювальної структури до вимог вирішуваної задачі. Збільшення коефіцієнта реконфігурації призводить до прискорення обчислень за рахунок зменшення складової часу T_{HW} виразу (1) ($T_{HW} \rightarrow \min$). Але прискорення за рахунок збільшення степені адаптації обчислювальної структури часто призводить до зворотного ефекту з причини високих часових витрат на здійснення реконфігурації ($T_{Rconf} \rightarrow \max$). В цьому контексті на діаграмі визначена область оптимізації реконфігурації (I) з точки зору забезпечення збільшення показника прискорення відповідно виразу (1). Застосування механізмів інтенсивного прискорення реконфігурації збільшує область оптимізації реконфігурації (II), дозволяючи реалізувати більшу степінь адаптації архітектури та прискорити обчислення.

Криві T_{Count} і $T_{Count}^{МІПР}$ зображують зміну загального часу обчислення відповідно за стандартної послідовності реконфігурації та засто-

сування механізмів інтенсивного прискорення реконфігурації в межах області оптимізації ре-
конфігурації.

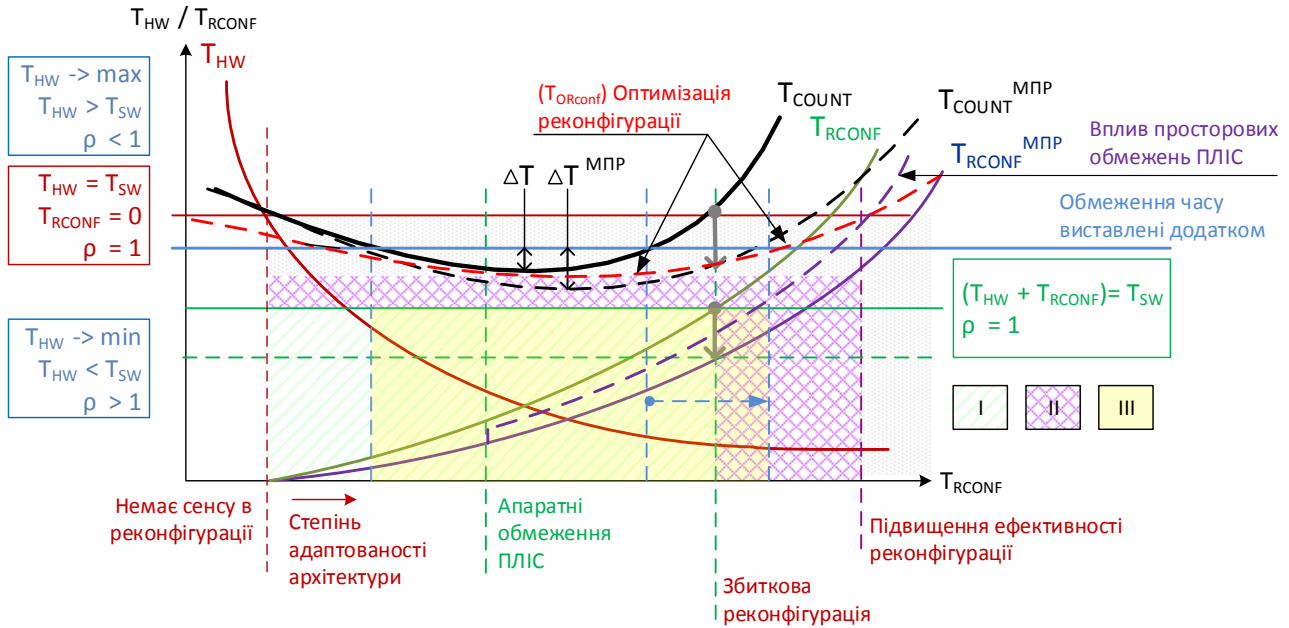


Рис. 1. Визначення критеріїв оптимізації реконфігурації

Прискорення обчислень природньо обмежується недостатчею апаратних ресурсів ПЛІС, як для реалізації найбільш адаптованої до структури задачі конфігурації на поверхні кристалу так і для досягнення максимального прискорення реконфігурації. Подолання обмежень апаратних ресурсів ПЛІС потребує додаткових витрат часу, що ще більше збільшує накладні видатки реконфігурації ($T_{Rconf} \rightarrow \max$). Апаратні обмеження ПЛІС зменшують ефективність механізмів інтенсивного прискорення реконфігурації та, як зображено на діаграмі (рис. 1), звужують область оптимізації реконфігурації (II).

Область оптимізації реконфігурації за критерієм виставлених додатком часових обмежень (III) також розширюється за рахунок механізмів прискорення реконфігурації, що, з іншого боку, обмежується недостатчею апаратних ресурсів ПЛІС. При цьому за наближення параметрів реконфігурації до точки перетину графіків T_{Rconf} та T_{Rconf}^{MPP} є деякий проміжок часу (ΔT^{MPP}), який дозволяє вкластися в обмеження часу виставлені додатком без використання, або часткового використання, механізмів прискорення реконфігурації. На підставі цього висновку в роботі запропоновано метод оптимізації процесу реконфігурації, що забезпечує видалення оптимального обсягу непродуктивних витрат часу, зменшуючи час реконфігурації настільки, щоб забезпечити при цьому мініма-

льне використання просторових ресурсів ПЛІС і задовольнити вимоги додатку до максимального часу виконання обчислень (крива T_{ORconf} на рис. 1).

Модифікований спосіб розподілу часу

Для реалізації методу оптимізації реконфігурації запропонований модифікований спосіб розподілу часу, який застосовується для розподілу заданого проміжку часу виконання додатку між послідовністю виконуваних макрозадач [11], що є складовими даного додатку. На відміну від відомого способу, заданий проміжок часу виконання розподіляється між послідовністю задач обернено пропорційно часу їх очікуваного виконання. Це призводить до виділення задачам, коефіцієнт прискорення яких більше ніж середній, додаткового часу виконання за рахунок задач, прискорення яких нижче ніж середнє. Таким чином, запропонований спосіб дозволяє визначити критичні до часу виконання задачі, що потребують максимального прискорення реконфігурації. Задачі з високим коефіцієнтом прискорення ефективно вирішуються на наявному устаткуванні засобами стандартної послідовності процесу реконфігурації.

Виконано теоретичні дослідження ефекту від застосування запропонованого модифікованого способу розподілу часу. Досліджений додаток є гіпотетичною послідовністю задач, за часові параметри яких прийняті часові характеристики

синтезованих на ПЛІС Cyclone II Altera арифметичних прискорювачів. На діаграмах (рис. 2) видно, що обернено пропорційна корекція T_{QoS_Mod} усереднених проміжків часу T_{QoS_Av} надає задачам, що мають високий коефіцієнт

прискорення додаткового часового ресурсу. При цьому задачі, коефіцієнт яких наближається до середнього, також отримують додаткового часу для виконання.

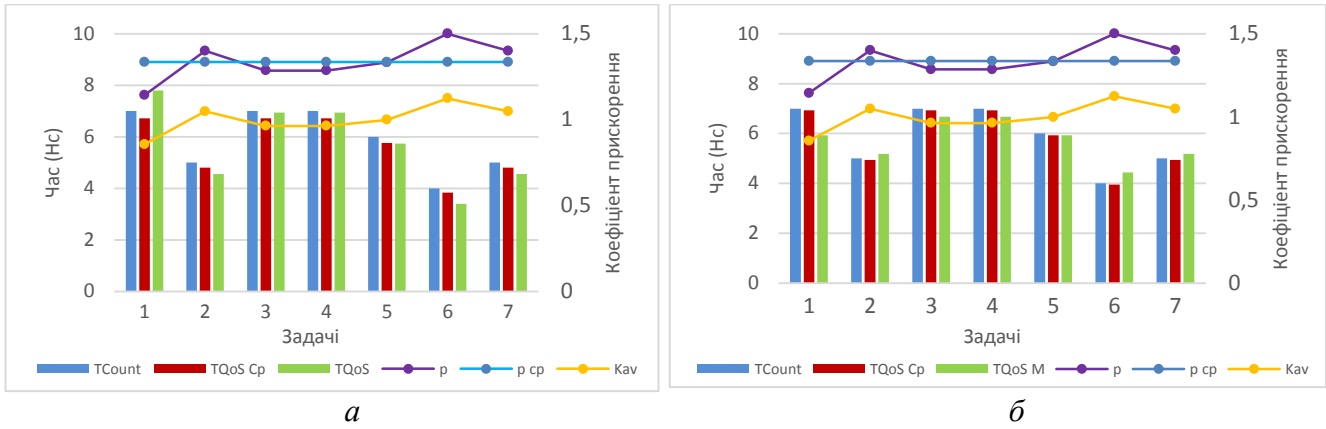


Рис. 2. Модифікований спосіб розподілу часу: а – пропорційна корекція (відомий спосіб), б – обернено пропорційна корекція усереднених проміжків часу обмеження

Математичний запис запропонованого модифікованого способу розподілу часу представлений виразом (3), де T_{QoS} – виставлені додатком обмеження часу виконання, T_{QoS_Mi} – обмеження часу, розраховані для кожної i -ї задачі, $T_{Count\ i}$ – очікуваного час виконання i -ї задачі, ρ_i – прискорення продуктивності, розраховане

за формулою (1), усереднене прискорення продуктивності ρ_{Av} , $i = \overline{1, n}$, де n – кількість задач в послідовності B , що надходить на виконання. На відміну від відомого способу в вираз (3) введена корекція похибки усереднення, що дозволило підвищити точність розрахунків.

$$T_{QoS_Mod\ i} = T_i \times \left[\frac{T_{QoS}}{\sum_{i=1}^n T_{C\ Count}} \times \frac{\sum_{i=1}^n T_{SW\ i}}{\sum_{i=1}^n T_{Count} \times \rho_{Av}} \right] \times \frac{\rho_i}{\rho_{Av}} \tag{3}$$

Виконання часових обмежень в загальному випадку визначається наступним виразом:

$$T_{Count\ i} \leq T_{QoS\ i}, \tag{4}$$

Таким чином, модифікований спосіб розподілу часу складається в наступному:

1. Заданий час обмеження виконання розподіляється на частини обернено пропорційні часу виконання кожної задачі згідно виразу (3) у наступній послідовності:

– виконується корекція часу обмеження до середнього значення,

– виконується розподіл середнього часу обмеження пропорційно середньому значенню прискорення продуктивності – отримання усереднених проміжків часу обмеження виконання кожної макрозадачі,

– корегуються усереднені проміжки часу згідно зворотного напрямку коефіцієнта усереднення кожної задачі.

2. Отримані проміжки часу аналізуються згідно загальної ознаки виконання обмежень (4).

На підставі (3) та (4) визначені наступні умови застосування способу розподілу часу та сформовані наступні межі його застосування, за яких раціонально використання запропонованого методу:

$$\begin{cases} \rho_{Cp} > 1 \\ \rho_{max} > \rho_{QoS} > 1 \\ \rho_{max} \rightarrow \rho_{Cp}, \rho_{min} \rightarrow \rho_{Cp} \end{cases}, \tag{5}$$

де $\rho_{QoS\ k}$ – прискорення, що вимагає накладені часові обмеження (шукане прискорення), ρ_{max}, ρ_{min} – граничні обмеження шуканого прискорення, в межах яких раціонально оптимізація реконфігурації.

Випадки, коли усереднене значення прискорення обчислювального процесу (яке є вихідним параметром для обчислення) менше за одиницю ($\rho_{Cp} < 1$), тобто коли фактично відбувається уповільнення загального обчислюва-

льного процесу, не розглядаються. Такі алгоритми можуть бути виконані засобами процесорних ядер. Це має сенс розглядати випадки, коли жодна задача не вкладається в часові обмеження, якщо $\rho_{max} < \rho_{QoS} > \rho_{Cp}$, або всі задачі вкладаються в обмеження, коли $\rho_{min} > \rho_{QoS} > 1$. Значення ρ_{max}, ρ_{min} розраховуються на підставі показника середньоквадратичного відхилення δ , який є мірою розкидання величин від їх середніх значень. Задачі, у яких прискорення продуктивності наближаються до значення $\pm 3\delta$, значним чином впливають на рівномірність розподілу часу обмеження, що з практичного боку реалізації призведе до надлишкових дії щодо мінімізації часу виконання задачі, і, як слідство, до збиткового використання апаратних та програмно-апаратних ресурсів системи. За цима межами застосування оптимізації реконфігурації недоцільно.

Метод оптимізації реконфігурації

Вихідний обчислюваний додаток є паралельною програмою зі змішаним типом паралелізму, для опису якого застосовується модель програмування M-задач [11]. Відповідно до цього програма (M-програма) задається макрографом потоку даних MDG (Macro Dataflow Graphs) в вершинах якого розміщуються макрозадачі (M-задачі), а ребра вказують на залежності між вершинами. Тоді, представимо вихідну програму макрографом $G_M = (V_M, E)$, де V_M – множина вершин, що відповідають макрозадачам, а E – множина ребер, що визначають відношення між макрозадачами. Зважаючи на архітектурні особливості реконфігурованої обчислювальної системи, розповсюдженим методом відображення задач на архітектуру є відображення рівнями. При цьому граф програми подається у ярусно-паралельній формі (ЯПФ) і кожний рівень підграфу послідовно відображується на архітектуру обчислювальної системи [11]. Час виконання обчислюваної програми визначається, як сума часу виконання послідо-

вності найтриваліших задач кожного рівня, і описується виразом:

$$T_{G_M_япф} = \sum_{k=1}^w \max \{T_{v_k} | v_k = \overline{1, H_k}\},$$

де $k = \overline{1, w}$ – номер ярусу, w – кількість ярусів обчислювального алгоритму, $v_k = \overline{1, H_k}$ – номер вузла на ярусі k , H_k – кількість вузлів на ярусі k . Відповідно до цього наступна упорядкована множина макрозадач:

$$B_{G_M} = \{T_{\max_k} | k = \overline{1, w}\}, \tag{6}$$

де T_{\max_k} – час виконання найтривалішої задачі k -го ярусу, складає найтривалішу взаємозв'язану послідовність задач, яка є вихідною для визначення загального часу виконання програми.

Під обмеженнями часу виконання розуміють обмеження часу встановлені на підставі певних зовнішніх факторів, наприклад, вихідних часових вимог виконуваного додатку.

Метод оптимізації процесу реконфігурації для динамічно реконфігурованих обчислювальних систем складається в наступному:

Етап 1

Визначення доцільної межі забезпечення часовим умовам:

$$\begin{cases} \rho_{Cp} > 1 \\ \rho_{max} > \rho_{QoS} > 1 \end{cases}$$

Етап 2

Видалення із послідовності задач B (6), множини задач D ($T_i \in D$, якщо $T_{Reconf\ i} = \max$) час яких неможливо зменшити, зокрема задачі першого рівня ЯПФ [11]:

$$T_{v_1} \in D, \text{ якщо } T_{v_1} = \max \{T_{v_1} | v_1 = \overline{1, H_1}\}.$$

тоді

$$T_{QoS} = T_{QoS} - \sum_{i=1}^n (T_i | \forall T_i \in D).$$

Етап 3

Забезпечення виконання умови

$$\rho_{max} \rightarrow \rho_{QoS}.$$

Етап 4

Розрахунок часу обмеження для задач, що залишилися в послідовності, на підставі (3):

$$T_{QoS\ i} = T_i \times \left[\frac{T_{QoS}}{\sum_{i=1}^n T_{Count\ i}} \times \frac{\sum_{i=1}^n T_{SW\ i}^\uparrow}{\sum_{i=1}^n T_{Count} \times \rho_{Cp}} \right] \times \frac{\rho_i}{\rho_{Cp}} \Big|_{\forall (T_i \notin D)}.$$

Етап 5. Визначення концепції зменшення часу виконання задач, що не вкладаються в часові

обмеження, шляхом застосування методів інтенсивного прискорення:

$$\rho_i \mid \forall (T_{Count\ i} < T_{QoS\ i}) = \frac{T_{SW\ i}}{T_{Count\ i}} = \frac{T_{SW\ i}}{T_{Rconf\ i} + T_{HW\ i}} \Rightarrow \begin{cases} T_{Rconf\ i} \rightarrow \min \\ T_{Count\ i} \rightarrow T_{HW\ i} \end{cases}$$

Для зменшення часу реконфігурації використовується запропоновані авторами робіт [10, 12] метод та засоби інтенсивного прискорення реконфігурації.

Моделювання методу оптимізації реконфігурації

Розроблений емулятор реконфігурованої обчислювальної системи (РОС) та програмна модель реалізації прискорення реконфігурації. Достовірність функціонування програмної моделі забезпечена шляхом застосування наближених до реальних часових характеристик функціональних елементів РОС, що отримані на базі розроблених моделей складових обчислювального модуля та функціональних блоків апаратних задач. Блоки апаратних задач синтезовані на мові Verilog та реалізовані на ПЛІС Cyclone II Altera.

Дослідження проводились для серії додатків, поданих графами алгоритмів в ЯПФ. Досліджувались графи алгоритмів з різною кількістю однотипних задач. На підставі проведених експериментів отримані залежності часу реконфігурації від місця розташування конфігурацій

них даних й від кількості виконуваних задач одного типу (рис. 3), та показник прискорення реконфігурації для алгоритмів з різною кількістю наборів однотипних задач (рис. 4). Виконані порівняння процесів інтенсивного оптимального прискорення реконфігурації.

Реконфігуровані обчислення виконані засобами стандартної послідовності реконфігурації потребують значного часу, як зображено на рис. 3, у зв'язку зі значною перевагою часу реконфігурації над часом виконання апаратної задачі, і не залежать від типів виконуваних задач. Метод прискорення реконфігурації за рахунок видалення всіх непродуктивних комунікаційних витрат часу забезпечує інтенсивне прискорення реконфігурації за умови відповідності структури задачі до структури реконфігурованого середовища, тобто коли ширина графу ЯПФ алгоритму відповідає розміру ПЛІС [11]. Інакше відбуваються витіснення апаратних задач в локальну пам'ять і завантаження затребуваних алгоритмом конфігурацій. Це потребує додаткового часу, але у порівнянні зі стандартною послідовністю реконфігурації МПР забезпечує загальне прискорення виконання обчислювального алгоритму.

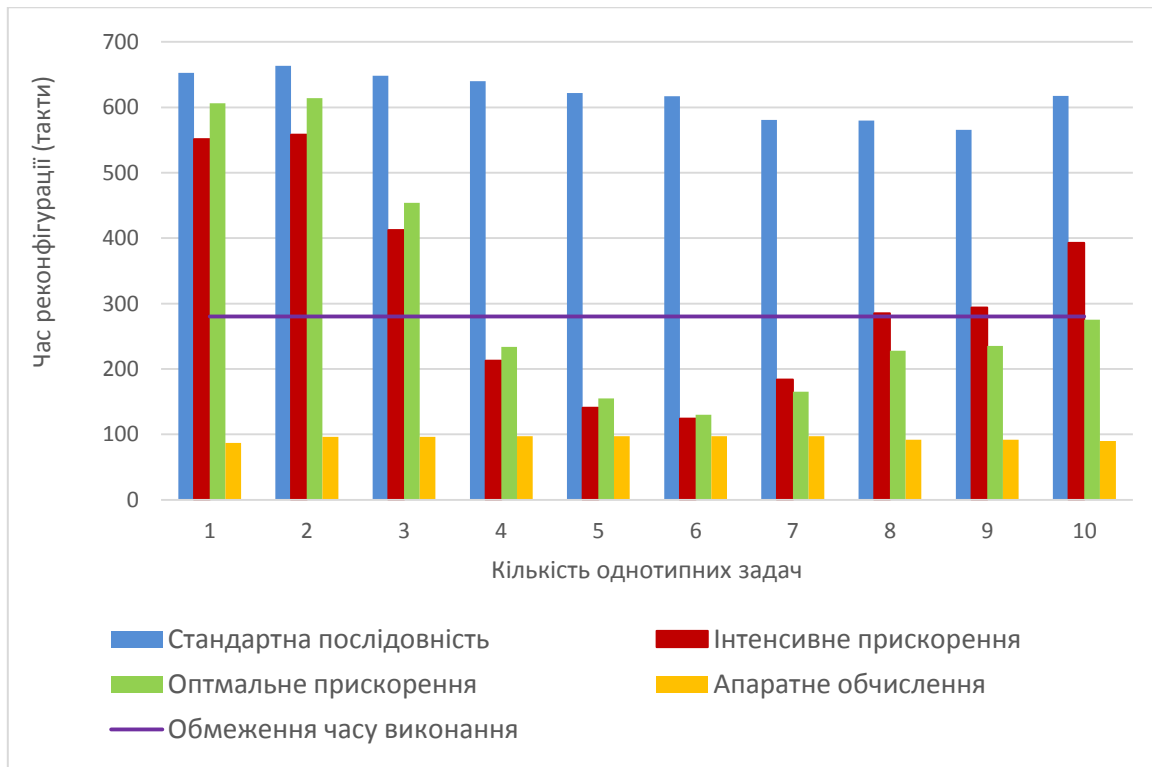


Рис. 3. Дослідження часу реконфігурації

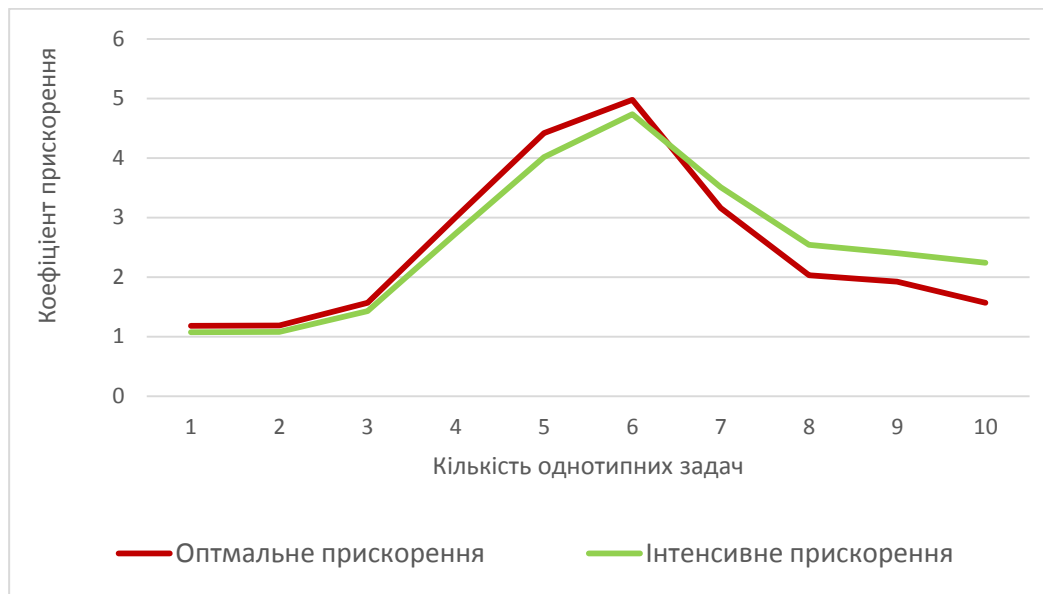


Рис. 4. Дослідження показників прискорення

За результатами експериментів (рис. 3, рис. 4) отримано, що механізм повторного використання ресурсів, на якому базується МПР, надає інтенсивне прискорення реконфігурації зі збільшенням типів задач до кількості сумірної з шириною графа ЯПФ. За сумірної з розміром реконфігурованої області ширини графа ЯПФ та високої кількості однотипних задач досягається збільшення інтенсивності прискорення реконфігурації в середньому на 63%. При цьому в процесі подолання просторових обмежень ПЛІС відбувається різке зменшення інтенсивності прискорення реконфігурації в середньому на 85%.

Метод оптимізації реконфігурації зменшує інтенсивність впливу просторових обмежень на швидкість реконфігурації приблизно на 10% у порівнянні з методом інтенсивного прискорення. З діаграми (рис. 4) видно, що більша кількість задач потрапила в межі часових обмежень. Метод оптимізації реконфігурації доцільно також застосовувати для зменшення кількості відмов виконання задач.

Висновки

Вперше запропоновано та обґрунтовано метод оптимізації процесу реконфігурації, що шляхом визначення та видалення оптимального обсягу накладних витрат часу, за критеріями забезпечення часових обмежень, виставлених вирішуваним додатком та забезпечення апаратних обмежень ресурсів ПЛІС під час розподілу динамічного потоку задач, дозволяє підвищити швидкодію реконфігурованих обчислень та зменшити кількість відмов виконання задач.

Модифіковано спосіб розподілу обмежень часу виконання між задачами обчислювального алгоритму, що дозволило забезпечити його використання в межах вирішення проблеми оптимізації реконфігурації за рахунок визначення задач, що не потребують залучення механізмів інтенсивного прискорення реконфігурації, а також підвищити точність розрахунків, у порівнянні з відомим способом.

Запропонований метод оптимізації реконфігурації дозволяє забезпечити широкий клас задач найбільш ефективною цільовою структурою для досягнення необхідної продуктивності обчислень, а також зменшити кількість відмов під час розподілу потоку динамічно надходжуваних задач.

Список посилань

1. El-Araby E. Exploiting Partial Runtime Reconfiguration for High-Performance Reconfigurable Computing / E. El-Araby, I. Gonzalez, T. El-Ghazawi // ACM Transactions on Reconfigurable Technology and Systems (TRETs). - USA, NY, New York, ACM, 2009. – Vol. 1, Issue 4, Article № 21.

2. Huang M. Reconfiguration and Communication-Aware Task Scheduling for High-Performance Reconfigurable Computing / M. Huang, V.K. Narayana, H. Simmler, O. Serres, T. El-Ghazawi // Transactions on Reconfigurable Technology and Systems (TRETs). – USA, NY, New York, ACM, 2010. – Vol. 3, Issue 4, Article № 20.
3. Bassiri M.M. Mitigating Reconfiguration Overhead In On-Line Task Scheduling For Reconfigurable Computing Systems / M.M. Bassiri, S.H. Shahriar // Proceeding of the 2nd International Conference on Computer Engineering and Technology (ICCET), (China, Chengdu, 16 – 18 April 2010). – IEEE, 2010. – Vol. 4. – P. V4-397 – V4 – 402.
4. Ahmed W. Adaptive Resource Management for Simultaneous Multitasking in Mixed-Grained Reconfigurable Multi-core Processors / W. Ahmed, M. Shafique, L. Bauer, J. Henkel // Proceedings of the 9th International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS) (Taiwan, Taipei, 9-14 October 2011). – IEEE, 2011. – P. 365 - 374.
5. Panella A. A design workflow for dynamically reconfigurable multi-FPGA systems / A. Panella, M.D. Santambrogio, F. Redaelli [and all] // Proceeding of the 18th VLSI System on Chip Conference (VLSI-SoC), (Spain, Madrid, 27 – 29 September, 2010). – IEEE/IFIP, 2010. – P. 414 - 419.
6. Al-Wattar A. Efficient On-line Hardware/Software Task Scheduling for Dynamic Run-time Reconfigurable Systems / A. Al-Wattar, S. Areibi, F. Saffih // Proceeding of the 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW) (China, Shanghai, 21 – 25 May 2012). – IEEE, 2012. – P 401 - 406.
7. Taher M. Virtual Configuration Management: A Technique for Partial Runtime Reconfiguration / M. Taher, T. El-Ghazawi // IEEE Transactions on Computers. – IEEE, 2009. – Vol. 58, Issue: 10. – P. 1398 - 1410.
8. Liu S. Achieving Energy Efficiency through Runtime Partial Reconfiguration on Reconfigurable Systems / S. Liu, R.N. Pittman, A. Forin, J.-L. Gaudiot // Transactions on Embedded Computing Systems (TECS). – USA, NY, New York, ACM, 2013. – Volume 12, Issue 3, Article № 72. – 21 p.
9. Ahmed W. mRTS:Run-Time System for Reconfigurable Processors with Multi-Grained Instruction-Set Extensions / W. Ahmed, M. Shafique, L. Bauer, J. Henkel // Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE) (France, Grenoble, 14 – 18 March 2011). – IEEE, 2011. – P. 1 - 6.
10. Кулаков Ю.О. Розробка методу прискорення реконфігурації в динамічно реконфігурованих обчислювальних системах / Ю. О. Кулаков, І. А. Клименко, М. В. Рудницький // Восточно-европейский журнал передовых технологий. – 2015. – Том 4, № 4(76) (2015). – С. 25 – 30.
11. Кулаков Ю.О. Метод оптимізації ярусно-паралельної форми подання задачі для реконфігурованих обчислювальних систем / Ю.О. Кулаков, І.А. Клименко // Електроніка та зв'язок. – К: НТУУ «КПІ», 2014. – Том 19, №4(81). – С. 90 - 96.
12. Кулаков Ю.О. Організація багаторівневої пам'яті в реконфігурованих обчислювальних системах // Ю.О. Кулаков, І.А. Клименко / Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: Зб. Наук. Пр. – К.: Век+, 2014. - №61. – С. 18 - 26.