

Запропонований метод прискорення реконфігурації та апаратні засоби його реалізації, що дозволяють мінімізувати накладні видатки часу реконфігурації. Виконано моделювання запропонованого методу для обчислювальних алгоритмів поданих в ярусно-паралельній формі. Застосування методу максимально зменшує комунікаційну складову часу реконфігурації, забезпечуючи інтенсивне прискорення реконфігурації в додатках, що містять велику кількість однотипних задач

Ключові слова: реконфігуровані обчислювальні системи, часткова динамічна реконфігурація, накладні видатки реконфігурації, прискорена реконфігурація

Предложен метод ускорения реконфигурации и аппаратные средства его реализации, которые позволяют минимизировать накладные расходы времени реконфигурации. Выполнено моделирование предложенного метода для вычислительных алгоритмов, представленных в ярусно параллельной форме. Применение метода максимально снижает коммуникационную составляющую времени реконфигурации, обеспечивая интенсивное ускорение реконфигурации в приложениях, которые содержат большое количество однотипных задач

Ключевые слова: реконфигурируемые вычислительные системы, частичная динамическая реконфигурация, накладные расходы реконфигурации, ускоренная реконфигурация

РОЗРОБКА МЕТОДУ ПРИСКОРЕННЯ РЕКОНФІГУРАЦІЇ В ДИНАМІЧНО РЕКОНФІГУРОВАНИХ ОБЧИСЛЮВАЛЬНИХ СИСТЕМАХ

Ю. О. Кулаков

Доктор технічних наук, професор*

E-mail: ya.kulakov@gmail.com

І. А. Клименко

Кандидат технічних наук, доцент*

E-mail: iklymenko@yandex.ua

М. В. Рудницький*

E-mail: miroslav.rudnitsky@gmail.com

*Кафедра обчислювальної техніки

Національний технічний університет України

«Київський політехнічний інститут»

пр. Перемоги, 37, м. Київ, Україна, 03056

1. Вступ

В галузі сучасних високопродуктивних обчислень актуальними є високопродуктивні реконфігуровані комп'ютери (*High-Performance Reconfigurable Computers, HPRC*) [1–3], які проявляють риси паралельних обчислювальних систем загального призначення. Така еволюція обумовлена технологічним рівнем сучасної програмованої елементної бази. Технологія часткової динамічної реконфігурації часу виконання [3] забезпечує динамічний режим реконфігурації і зниження затримок реконфігурації шляхом реконфігурації лише частини мікросхеми без зупинення роботи інших функціональних частин пристрою на кристалі ПЛІС. Здійснення динамічної реконфігурації супроводжується високими накладними видатками, які негативно впливають на продуктивність обчислень. Найбільш критичними є витрати часу та продуктивності на реалізацію процесу реконфігурації. Апаратні обмеження елементної бази, зокрема логічних ресурсів, ресурсів вбудованої пам'яті й інтерфейсів конфігурації ПЛІС також впливають на збільшення накладних видатків реконфігурації.

Проблема зменшення накладних витрат реконфігурації, в контексті досягнення високої продуктивності в реконфігурованих обчислювальних системах, не може бути ефективно вирішена відомими методами і

засобами. Це обумовлює високу актуальність і доцільність виконаних в статті досліджень.

2. Аналіз літературних даних та постановка проблеми

Рішення проблеми зменшення накладних витрат реконфігурації зазвичай спрямовані на розробки алгоритмів планування і розміщення задач, що водночас реалізують різні методи зменшення витрат реконфігурації. Найбільш розповсюдженими методами є попередня вибірка конфігурацій [3], повторне використання обчислювальних ресурсів [2, 4, 5], поєднання конфігурацій з метою зменшення накладних витрат і руху даних [1]. Переважна кількість розробок базуються на програмному або експериментальному моделюванні реконфігурації, часто абстраговано від фізичної сторони реалізації. Всі рішення покладаються на рівень програмної або програмно-апаратної надбудови операційної системи та мають високу складність реалізації алгоритмів управління реконфігурацією, що додатково збільшує накладні видатки реконфігурації, негативно впливаючи на продуктивність обчислювальної системи. Ряд робіт, які пропонують архітектурні та технічні вдосконалення фізичного рівня реконфігурованих систем [6, 7], навпаки, абстрагуються від засобів управління реконфігура-

цією. Апаратні засоби прискорення реконфігурації зазвичай орієнтовані на збільшення пропускної здатності інтерфейсів реконфігурації. При цьому, під час розміщення пам'яті великого об'єму на ПЛІС виникає інша проблема непродуктивного використання ресурсів пам'яті ПЛІС [8].

Таким чином, постає задача розробки нових методів та засобів прискорення реконфігурації, які відповідають вимогам узгодженості логічного та фізичного рівнів реконфігурації, забезпечення механізмів зменшення накладних витрат реконфігурації, зберігання апаратних ресурсів ПЛІС.

3. Ціль та задачі дослідження

Метою дослідження є підвищення продуктивності динамічно реконфігурованих обчислювальних систем за рахунок зменшення накладних витрат реконфігурації з врахуванням апаратних обмежень ПЛІС.

Для досягнення поставленої мети в роботі вирішується задача розробки та дослідження методів та засобів мінімізації непродуктивних витрат часу, що дозволять підвищити швидкодію процесу реконфігурації.

4. Матеріали та методи дослідження впливу накладних витрат реконфігурації на продуктивність реконфігурованих обчислювальних систем

4.1. Програмні засоби та устаткування, що використовувались під час досліджень

Функціональні блоки, на базі яких виконано моделювання та дослідження процесу реконфігурації, синтезовані на мові опису апаратури *Verilog* та реалізовані на ПЛІС *Cyclone II EP2C35F672C6* компанії *Altera*. Для верифікації і дослідження часових характеристик функціональних блоків використана плата *Altera Development Kit DE2*. Програмна модель методу прискорення реконфігурації та програмний емулятор архітектури реконфігурованої обчислювальної системи розроблені на мові програмування *Java*.

4.2. Формалізація постановки задачі зменшення накладних витрат реконфігурації

В якості основного критерію оцінки ефективності процесу реконфігурації зазвичай приймають показник прискорення [9]:

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

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 задачі на поверхні реконфігурованої області обчислювального модуля. Збільшення показника прискорення досягається за рахунок зменшення складової T_{Rconf} виразу (1). Час T_{Rconf} визначає обсяг накладних витрат реконфігурації, обумовлених непродуктивними витратами часу, і визначається наступними параметрами: тривалістю циклу управління реконфігурацією T_{Contr} – обумовлює витрати продуктивності; часом передавання кон-

фігураційних даних T_{Comm} – обумовлює комунікаційні витрати; часом прошивання мікросхеми T_{Mapp} – залежить від технологічних особливостей реконфігурованої елементної бази.

Тоді проблема прискорення реконфігурації зводиться до зменшення комунікаційних витрат під час реконфігурації і зменшення складності алгоритмів управління реконфігурацією, що формально виражається наступним чином:

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

4.3. Метод прискорення реконфігурації

Запропонований авторами метод прискорення реконфігурації (МПП) базується на механізмі повторного використання апаратних ресурсів функціональних блоків і забезпечує зменшення накладних витрат часу ($T_{Rconf} \rightarrow \min$) за рахунок зменшення комунікаційних витрат ($T_{comm} \rightarrow \min$).

Час виконання послідовності B взаємозв'язаних обчислювальних задач $I_j, j=1, K$, де K – кількість типів апаратних задач, виражається як

$$T_B = \sum_{j=1}^K (P_j T_j + P_j \Delta T_j), \quad (3)$$

де T_j – час виконання задачі I_j , ΔT_j – час її завантаження та конфігурування, P_j – кількість екземплярів однотипних задач. За рахунок повторного використання апаратних ресурсів функціональних блоків, забезпечується наступний час виконання послідовності обчислень:

$$T_B^{MPP} = \sum_{j=1}^K (P_j T_j + \Delta T_j). \quad (4)$$

Складова $P_j T_j$ в виразі (4) відповідає продуктивному сумарному часу виконання всіх екземплярів апаратної задачі I_j на поверхні реконфігурованої області. Складова ΔT_j відповідає часу її одnorазової конфігурації.

Застосування швидкодіючої локальної пам'яті в структурі обчислювального модуля, дозволяє отримати додаткове збільшення абсолютного показника прискорення під час застосування методу прискорення реконфігурації за рахунок зменшення затримки ΔT_j в виразі (4):

$$\Delta T_{MP}^+ = \sum_{j=1}^K (P_j \Delta T_j - \Delta T_j^+), \text{ якщо } \Delta T_j^+ < \Delta T_j. \quad (5)$$

Формальні визначення (4), (5) методу прискорення реконфігурації справедливі для будь-якого послідовного алгоритму обчислень. Математичний запис методу прискорення реконфігурації для паралельного алгоритму обчислень $\|A$ виглядає наступним чином:

$$T_{MPP}^{\|A} = \sum_{j=1}^K \Delta T_j + \frac{\sum_{j=1}^K (P_j T_j)}{F}, \quad (6)$$

де обчислювальна частина виразів (4) та (5) розгалужується ідеально на F функціональних блоків, а сумарний час реконфігурації є послідовною частиною, розгалу-

ження якої неможливе, зважаючи на послідовні засоби реконфігурації і інтерфейс конфігурації. За застосування ярусно-паралельної форми (ЯПФ) подання графів алгоритмів [10], коли вихідна М-програма подана макрографом $G_M=(V, E)$, де V – множина вершин, що відповідають макрозадачам, а E – множина ребер, що визначають відношення між макрозадачами, час виконання обчислюваної програми визначається, як сумарний часу виконання найтриваліших задач кожного ярусу:

$$T_G = \sum_{k=1}^w \max \{T_h | h = \overline{1, N_k}\}, \quad (7)$$

де $k = \overline{1, w}$ – номер ярусу, w – кількість ярусів обчислювального алгоритму, $h = \overline{1, N_k}$ – номер вузла на ярусі k , N_k – кількість вузлів на ярусі k . Відповідно до цього, наступна упорядкована множина М-задач:

$$B = \{T_{max_k} | k = \overline{1, w}\}, \quad (8)$$

де T_{max_k} – час виконання найтривалішої задачі k -го ярусу, складає найтривалішу взаємозв'язану послідовність задач, за якою визначається час виконання програми. За застосування методу прискорення реконфігурації час виконання послідовності найтриваліших задач B визначається виразом (4).

Потенційна можливість зменшення часу виконання паралельного алгоритму (6) лежить в зменшенні послідовної компоненти часу, що може бути реалізовано шляхом розпаралелювання процесів обчислення та реконфігурації. Це дозволяє реалізувати принцип реконфігурації «як можна раніше», відповідно якому всі процедури реконфігурації переміщуються на попередній ярус. Таке розпаралелювання процесів реконфігурації та обчислення призводить практично до повного знехтування часом реконфігурації і наближає до нуля непродуктивні витрати часу. Виключеннями є процедури реконфігурації першого ярусу графу алгоритму ЯПФ [10].

Відповідно до вищесказаного та виразів (6)–(8) наведемо математичну модель методу прискорення реконфігурації для макрографу, поданого в ЯПФ [10]:

$$T_G^{МПВ} = \sum_{j=1}^K T_{IO_j} + \sum_{k=1}^w \max \left(\sum_{h=1}^{N_k} \Delta T_h, \{T_h | h = \overline{1, N_k}\} \right), \quad (9)$$

де послідовна компонента T_{IO} визначає послідовність процесів, які остаточно не можуть бути розділені у часі, наприклад, процеси обміну даними під час реконфігурації, що задіюють загальне комунікаційне середовище.

4. 4. Апаратні засоби реалізації методу прискорення реконфігурації

Реалізацію методу прискорення реконфігурації забезпечують розроблені апаратні засоби управління реконфігурованими ресурсами на базі бага-

торівневої пам'яті. Апаратні засоби забезпечують зменшення обчислювальної складності алгоритмів управління реконфігурованими ресурсами (складова $\min(T_{Contr})$ виразу (2)) та вирішують проблему обмеженості ресурсів вбудованої пам'яті ПЛІС. До складу реконфігурованого модуля додано спеціальний контролер реконфігурації, який забезпечує розпаралелювання логічної і фізичної послідовності реконфігурації і розвантажує локальний процесор. В роботі [8] авторами запропонований спосіб зберігання конфігураційних даних апаратних задач, за якого часто затребувані конфігурації апаратних задач зберігаються в заздалегідь зконфігурованому вигляді на поверхні реконфігурованої області ПЛІС. Для копіювання конфігурацій на поверхні реконфігурованої області задіяна внутрішня пам'ять ПЛІС. Такий спосіб зберігання дозволяє зменшити час повторного завантаження конфігураційних даних і задовольняє обмеженням просторових ресурсів мікросхеми ПЛІС. Облік та підтримку розкладу зберігання конфігурацій на поверхні реконфігурованої області забезпечує багаторівнева швидкодіюча пам'ять конфігурацій, що реалізована засобами декількох прошарків кеш-пам'яті між центральною бібліотекою конфігурацій і реконфігурованою структурою мікросхеми ПЛІС [8].

4. 5. Емуляція апаратних засобів прискорення реконфігурації

Розроблений емулятор реконфігурованої обчислювальної системи (РОС), структурна схема якої зображена на рис. 1.

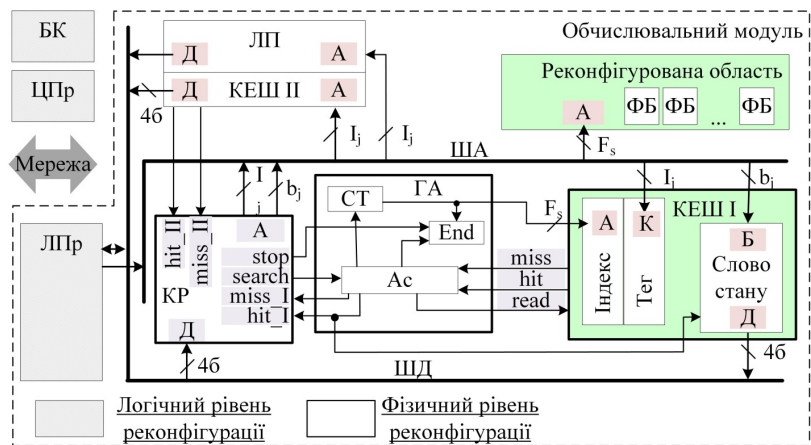


Рис. 1. Структурна схема реконфігурованої обчислювальної системи: ЦПр – центральний процесор, БК – бібліотека конфігурації, ЛППр – локальний процесор, ЛП – локальна пам'ять, КР – контролер реконфігурації, ГА – генератор адрес, ША – шина адреси, ШД – шина даних, ФБ – функціональний блок, А – адреса, Д – дані, К – компаратор, СТ – лічильник адрес, Ас – аналізатор сигналів

Емулятор РОС включає наступні блоки, що реалізовані як програмні модулі, які емулюють функції відповідних апаратних блоків: *HardwareSystem* – головний модуль, що емулює функціонал контролера реконфігурації й та включає підлегли модулі: *FPGA* – реконфігурована область; *memory* – локальна пам'ять, що реалізує функціонал КЕШ пам'яті першого рівня і механізм підтримки бонусів (*bonuses*); генератора

адрес. Модуль *HardwareSystem* виконує функції пошуку конфігурації задачі – *findConfiguration* та завантаження задачі в ПЛІС – *load*. Функція *findConfiguration* повертає значення, які встановлюються на підставі сигналів *hit* або *miss* (вдалого, або невдалого пошуку) від КЕШ пам'яті першого (КЕШ I) або другого (КЕШ II) рівнів відповідно до табл. 1.

Таблиця 1

Значення функції *findConfiguration*

Від КЕШ I		→	Від КЕШ II		Стан пошукової процедури	
h	I		h	II	h_I h_II	Результат пошуку, (ім'я змінної)
1	<i>hit_I</i>	0	–	–	10	<i>TSK_FPGA</i>
0	<i>miss_I</i>	1	0	<i>miss_II</i>	01	<i>TSK_MEM</i>
–	–	–	1	<i>hit_II</i>	00	<i>TSK_LIB</i>

Реконфігурована область представлена масивом об'єктів, що відповідають завантаженим в даний момент функціональним блокам апаратних задач. Система обліку та управління реконфігурованими ресурсами забезпечує підтримку нарахування бонусів та стану лічильників та таймерів для функціональних блоків апаратних задач, що включені в реконфігуровану область. Завантажений функціональний блок представляється набором властивостей, що зберігається в пам'яті КЕШ I. Локальна пам'ять представлена множиною об'єктів, що відповідають апаратним задачам, витісненим із реконфігурованої області. Апаратна задача в локальній пам'яті представляється набором властивостей, що зберігається в пам'яті КЕШ II. Модуль *lib* центральної бібліотеки конфігурацій представлений масивами даних (*data*) і назвою файлу з даними (*LIBRARY_FILE*). Модуль містить набір функцій, що забезпечують доступ до даних. Апаратні задачі, синтезовані заздалегідь, і у вихідному стані зберігається в централізованій бібліотеці конфігурацій у вигляді об'єктів конфігураційних файлів разом зі своїми властивостями.

4. 6. Моделювання методу прискорення реконфігурації

Модуль (*modeller*) моделює функціонал реконфігурованої обчислювальної системи. Функція моделювання *modell* повертає результат у вигляді діаграми Ганта в текстовому форматі. Для оцінки часових залежностей під час моделювання методу прискорення реконфігурації використані наступні константи: час звернення до пам'яті – *MEMORY_ACCESS_TIME*, час завантаження на мікросхему – *LOAD_LAST_TIME*, час завантаження даних – *LOAD_DATUM_TIME*, максимальний час для випадкової складової доступу в мережі – *NETWORK_MAX_RANDOM_TIME*.

Згідно методу прискорення реконфігурації (9) конфігурація та виконання апаратної задачі відбувається за різними послідовностями дій в залежності від місця знаходження конфігураційних даних:

I – на поверхні реконфігурованої області,
 II – в локальній пам'яті,
 III – в центральній бібліотеці конфігурацій. Відповідно до цього виконується розрахунок часу обчислення T_j :

$$T_j^I = T_{count j}, T_{reconf j} = 0,$$

$$T_j^{II} = T_{comm j} + T_{count j}, T_{reconf j} = T_{comm j},$$

$$T_{sum j}^{III} = T_{comm_net j} + T_{count j}, T_{reconf j} = T_{comm_net j},$$

де $T_{count j}$ – час виконання апаратної задачі I_j , включаючи процеси вводу вихідних даних та виводу результатів (*LOAD_DATUM_TIME*), завантаження (*LOAD_LAST_TIME*), виконання (*workTime* або *correctTime*); $T_{comm j}$ – час передавання конфігураційних даних із локальної пам'яті, залежить від часу звернення до локальної пам'яті (*MEMORY_ACCESS_TIME*) та розміру конфігураційного файлу (*bytestreamWords*); T_{comm_net} – час передавання конфігураційних даних із бібліотеки конфігурацій на рівень обчислювального модуля мережевими засобами зв'язку, залежить від випадкової мережевої складової комунікаційного часу (*NETWORK_MAX_RANDOM_TIME*), та розміру конфігураційного файлу. В усіх послідовностях враховується час пошуку конфігурації апаратної задачі (*TIME_findConfiguration*), розрахований експериментально.

5. Результати досліджень показників часу реконфігурації

Дослідження проводились для серії додатків, поданих графами алгоритмів в ЯПФ. Досліджувались графи алгоритмів з різною кількістю однотипних задач та різною ступеню зв'язності. На підставі проведених експериментів отримані залежності часу реконфігурації від місця розташування конфігураційних даних й від кількості типів виконуваних задач (рис. 2), та показник прискорення реконфігурації для алгоритмів з різною кількістю наборів однотипних задач (рис. 3). Показник інтенсивності прискорення реконфігурації розрахований як $K_n = (K_n / K_{n-1}) \times 100\%$, де K – коефіцієнт прискорення, що є відношенням часу реконфігурації під час застосування методу прискорення реконфігурації і часу виконання стандартної послідовності реконфігурації, n – номер виміру.

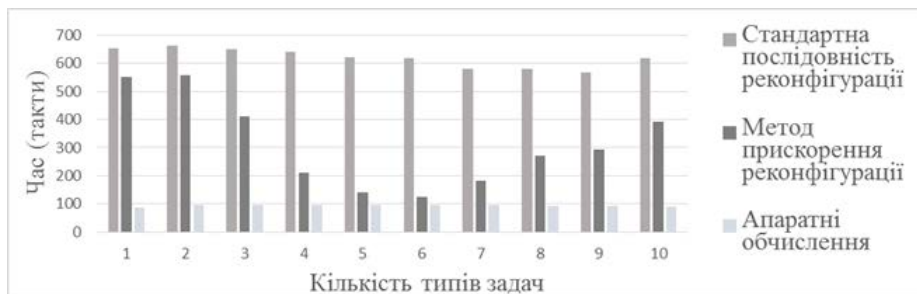


Рис. 2. Дослідження часу реконфігурації

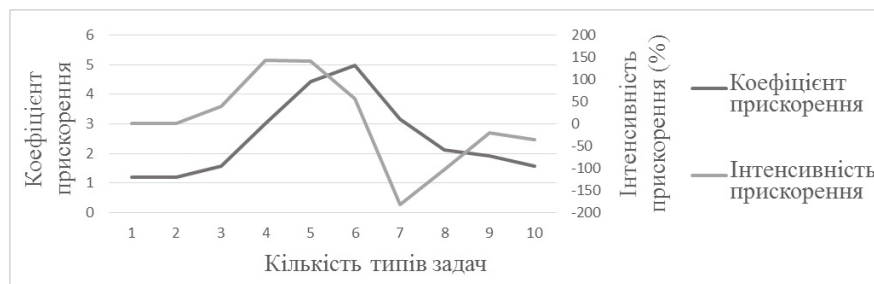


Рис. 3. Дослідження показників прискорення

Під час дослідження показників прискорення реконфігурації за застосування запропонованого методу отримано середній коефіцієнт прискорення реконфігурації, що дорівнює 2,5. Зменшення інтенсивності прискорення реконфігурації в середньому на 85 % відслідковується за збільшення кількості типів задач. За сумірної з розміром реконфігурованої області ширини графа ЯПФ та високої кількості однотипних задач досягається збільшення інтенсивності прискорення реконфігурації в середньому на 63 %. Середній показник збільшення інтенсивності прискорення реконфігурації під час застосування методу прискорення реконфігурації у порівнянні зі стандартною послідовністю реконфігурації досягає 4,3 %.

6. Обговорення результатів дослідження впливу накладних видатків реконфігурації на продуктивність реконфігурованих обчислювальних систем

Реконфігуровані обчислення виконані засобами стандартної послідовності реконфігурації (П) потребують значного часу, як зображено на рис. 1, у зв'язку зі значною перевагою часу реконфігурації над часом виконання апаратної задачі, і не залежать від типів виконуваних задач. За результатами досліджень видно, що запропонований метод прискорення реконфігурації за рахунок видалення практично всіх комунікаційних витрат часу забезпечує інтенсивне прискорення реконфігурації за відповідності структури задачі до структури реконфігурованого середовища, тобто коли ширина графу ЯПФ алгоритму відповідає розміру ПЛІС. В іншому випадку відбуваються витіснення апаратних задач в локальну пам'ять і завантаження затребуваних алгоритмом конфігурацій. Це потребує додаткового часу, але у порівнянні зі стандартною послідовністю реконфігурації відслідковується загальне прискорення

виконання обчислювального алгоритму за рахунок максимально можливого зменшення комунікаційних витрат. Також на час реконфігурації впливає щільність надходження задач різного типу. Зважаючи на те, що механізми повторного використання ресурсів надають значне прискорення за великої кількості однотипних задач на протязі виконання послідовності обчислень, досліджено, що метод прискорення реконфігурації надає інтенсивне прискорення зі збільшенням типів задач до кількості сумірної з шириною графа ЯПФ.

7. Висновки

На підставі досліджень можна зробити наступні висновки щодо ефективності та практичної значимості запропонованих засобів прискорення реконфігурації:

- спосіб зберігання та підтримки конфігурацій апаратних задач на поверхні реконфігурованої області ПЛІС та апаратні засоби управління розкладом підтримки конфігурацій на функціональному рівні, забезпечують зменшення часу реконфігурації і сприяють зменшенню використання ресурсів внутрішньої пам'яті ПЛІС;
- метод прискорення реконфігурації, за рахунок реалізації механізму повторного використання ресурсів функціональних блоків на підставі запропонованого способу зберігання задач, дозволяє видалити практично всі непродуктивні витрати часу, забезпечуючи інтенсивне прискорення реконфігурації;
- організація багаторівневої пам'яті дозволяє реалізувати спосіб зберігання та підтримки конфігурацій, а розроблені апаратні засоби управління розкладом розміщення та підтримки конфігурацій функціональних блоків дозволяють зменшити обчислювальну складність алгоритмів управління реконфігурацією;
- емулятор реконфігурованої обчислювальної системи та програмна модель методу прискорення реконфігурації дозволяють в режимі реального часу моделювати процес управління реконфігурованими ресурсами, та є зручними інструментами для дослідження часових характеристик реконфігурованої обчислювальної системи.

Література

1. Huang, M. Reconfiguration and Communication-Aware Task Scheduling for High-Performance Reconfigurable Computing [Text] / M. Huang, V. K. Narayana, H. Simmler, O. Serres, T. El-Ghazawi // ACM Transactions on Reconfigurable Technology and Systems (TRETs). – 2010. – Vol. 3, № 4. – P. 1–25. doi:10.1145/1862648.1862650
2. 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 (ICCEt), Chengdu, 16-18 April 2010. – 2010. – Vol. 4. – P. V4-397–V4-402. doi:10.1109/iccet.2010.5485509
3. El-Araby, E. Exploiting Partial Runtime Reconfiguration for High-Performance Reconfigurable Computing [Text] / E. El-Araby, I. Gonzalez, T. El-Ghazawi // ACM Transactions on Reconfigurable Technology and Systems (TRETs). – 2009. – Vol. 1, № 4. – P. 1–23. doi:10.1145/1462586.1462590
4. Panella, A. A Design Workflow for Dynamically Reconfigurable Multi-FPGA Systems [Text] / A. Panella, F. Redaelli, F. Cancare, D. Sciuto // Proceeding of 18th IEEE/IFIP VLSI System on Chip Conference (VLSI-SoC), Spain, Madrid, 27-29 Sept. 2010. – 2010. – P. 114–119. doi:10.1109/vlsisoc.2010.5642697

5. 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), 21-25 May 2012. – IEEE, 2012. – P. 401–406. doi:10.1109/ipdpsw.2012.50
6. Liu, S. Achieving Energy Efficiency through Runtime Partial Reconfiguration on Reconfigurable Systems [Text] / S. Liu, R. N. Pittman, A. Forin, J.-L. Gaudiot // ACM Transactions on Embedded Computing Systems (TECS). – 2013. – Vol. 12, № 3. – P. 1–21. doi:10.1145/2442116.2442122
7. Liu, S. Minimizing the runtime partial reconfiguration overheads in reconfigurable systems [Text] / S. Liu, R. N. Pittman, A. Forin, J.-L. Gaudiot // The Journal of Supercomputing. – 2012. – Vol. 61, № 3. – P. 894–911. doi:10.1007/s11227-011-0657-6
8. Кулаков, Ю. О. Організація багаторівневої пам'яті в реконфігурованих обчислювальних системах [Текст]: зб. наук. пр. / Ю. О. Кулаков, І. А. Клименко // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка. – К.: Век+, 2014. – № 61. – С. 18–26.
9. Ahmed, W. Adaptive Resource Management for Simultaneous Multitasking in Mixed-Grained Reconfigurable Multi-core Processors [Text] / 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. doi:10.1145/2039370.2039426
10. Кулаков, Ю. О. Метод оптимізації ярусно-паралельної форми подання задачі для реконфігурованих обчислювальних систем [Текст] / Ю. О. Кулаков, І. А. Клименко // Електроніка та зв'язок. – К: НТТУ «КПІ», 2014. – Том 19, № 4 (81). – С. 90–96.

В статті розглянуто питання моделювання динаміки процесу дроблення руди з використанням апарату ортонормованих функцій Лагерра. Виконано порівняльний аналіз якості наближення виходу моделі до характеристик досліджуваного об'єкту. В результаті визначено структуру і масштабний коефіцієнт моделі Лагерра, а також інтервал дискретизації, які забезпечують мінімальну середньоквадратичну похибку ідентифікації

Ключові слова: процес дроблення, модель Лагерра, ідентифікація, інтервал дискретизації, моделювання

В статье рассмотрен вопрос моделирования динамики процесса дробления руды с использованием аппарата ортонормированных функций Лагерра. Выполнен сравнительный анализ качества приближения выхода модели с характеристиками изучаемого объекта. В результате определена структура и масштабный коэффициент модели Лагерра, а также интервал дискретизации, которые позволяют обеспечить минимальную среднеквадратичную ошибку идентификации

Ключевые слова: процесс дробления, модель Лагерра, идентификация, интервал дискретизации, моделирование

УДК 65.011.56:681.3

DOI: 10.15587/1729-4061.2015.47318

МОДЕЛЮВАННЯ ДИНАМІКИ ПРОЦЕСУ ДРОБЛЕННЯ РУДИ З ВИКОРИСТАННЯМ МОДЕЛІ ЛАГЕРРА

О. Ю. Михайленко

Асистент

Кафедра електропостачання
та енергетичного менеджменту

ДВНЗ «Криворізький
національний університет»

вул. XXII Партз'їзду, 11,

м. Кривий Ріг, Україна, 50027

E-mail: epem.mykhailenko@gmail.com

1. Вступ

Для рішення задачі підвищення якісних показників процесу дроблення руди у конусних дробарках шляхом автоматизованого керування ним, необхідно отримати математичний опис залежностей між вхідними параметрами і вихідної величиною, що характеризує протікання цього процесу.

Використання існуючих аналітичних і експериментальних моделей процесу дроблення для рішення задач автоматизації обмежено. Так, матричні моделі [1] дозволяють дослідити загальні закономірності форму-

вання гранулометричного складу готового продукту при різних значеннях вхідних параметрів (розподілу за класами крупності живлення і ширини розвантажувальної щілини). Проте, даний клас моделей не враховує вплив неконтрольованих збурень, а також вони описують процес дроблення лише у статичних режимах. Наявність складних функціональних залежностей, здебільшого нелінійних, ускладнює налаштування моделей стосовно реального процесу.

Регресійні та стохастичні моделі [2] мають більшу точність, у порівнянні з аналітичними, проте вони також мають недоліки, які полягають в їх неінтерпре-