

Скорочення Необхідного Ресурсу ПЛІС Для Обчислювальних Систем на Кристалі

Валерій Жабін
каф. обчислювальної техніки
НТУУ «КПІ ім. Ігоря Сікорського»
Київ, Україна
vizhabin@comsys.kpi.ua

Валентина Жабіна
каф. програмного забезпечення комп'ютерних систем
НТУУ «КПІ ім. Ігоря Сікорського»
Київ, Україна
val.zhabina2@gmail.com

Reduction of Necessary FPGA Source for Computing System on Chip

Valerii Zhabin
dept. of Computer Engineering
NTUU "Igor Sikorsky Kyiv Polytechnic Institute"
Kyiv, Ukraine
vizhabin@comsys.kpi.ua

Valentina Zhabina
dept. of Computer Systems Software
NTUU "Igor Sikorsky Kyiv Polytechnic Institute"
Kyiv, Ukraine
val.zhabina2@gmail.com

Анотація—Показана можливість зменшення необхідного апаратного ресурсу ПЛІС за рахунок порозрядної обробки даних в системах з безпосередніми зв'язками між обчислювальними модулями. Наведено приклад реалізації такої системи для обчислення степеневих функцій з порозрядною обробкою операндів зі старших розрядів в симетричних надлишкових системах числення. Запропонована реалізація обчислень дозволяє при порівняно високій швидкодії скоротити необхідний ресурс ПЛІС завдяки малому числу зв'язків між модулями

Abstract—In the article there is shown a possibility to reduce necessary FPGA hardware sources by means of digit-by-digit data processing in data-flow systems with direct couplings between processing units. One of the approaches to solving the problem of reduction of number of connections between processing units is to use quasi-parallel processors, which enable to realize concurrently the processes of digit-by-digit operands input, as well as the process of digit-by-digit result output starting from higher positions. The digit-by-digit data interchange starting from higher positions enables not only to reduce the number of connections between processing units, but also to provide realization of data-flow operations in concurrency mode. In a simple case at each step of calculations one operand position is put into processing unit starting from the higher one, as well as one result position is formed. In this calculation mode, the next operation execution will not be started after previous operation is performed, but right after first result position is received. Samples of algorithms of calculation of power functions in on-line mode with the use of symmetrical redundant numerical system with digits $\{-1, 0, 1\}$ for operands representation are

considered. To raise numbers to powers there is used the binary tree method. Elements of internal sources (execution and communication units) as well as FPGA output pins are saved. As a result of the above-mentioned, a possibility to place the system on one microchip is increased. It enables to increase system reliability and reduce energy usage and dimensions.

Ключові слова—неавтономний режим; степеневі функції; порозрядна обробка; системи на кристалі; ПЛІС

Keywords—on-line mode; power functions; digit-by-digit processing; SoC; FPGA

I. ВСТУП

Тривалість обробки інформації у паралельних системах залежить не тільки від часу виконання операцій, але і від затрат часу на обмін інформацією між гілками алгоритмів, тобто між обчислювальними модулями (ОМ) паралельної системи. Зменшити затрати часу на обмін даними дозволяє використання потокових систем з безпосередніми зв'язками (ПСБЗ) між ОМ [1, 2].

У ПСБЗ виходи одних ОМ підключаються до входів інших ОМ (можливо через елементи комутації) відповідно до графа потоку даних (ГПД). ОМ працюють у неавтономному (on-line) режимі. У процесі обчислень дані пересилаються безпосередньо від одних ОМ до інших, перетворюючись на кожному кроці відповідно до операцій, що задані вершинами ГПД. У такому випадку відсутні складні процедури пересилання даних між ОМ через загальну пам'ять, тобто створюються передумови до

зменшення непродуктивних витрат часу в процесі обміну інформацією.

Досягнення в області інтегральної технології дозволяють створювати паралельні системи за технологією SoC (System on Chip), які можна реалізувати на ПЛІС з використанням інтерактивних засобів проектування, що суттєво прискорюють термін розробки систем. Однак з боку елементної бази накладається ряд обмежень, пов'язаних з числом виводів (pins) мікросхем, наявністю вбудованих функціональних вузлів і пристроїв. Наприклад, ресурсів однієї мікросхеми може не вистачити для реалізації всієї системи, тобто може виникнути необхідність застосування кількох мікросхем, пов'язаних між собою зовнішніми лініями зв'язку. Це визначається як розмірністю задач, так і складністю обчислювальних засобів, що залежить від алгоритмів виконання операцій.

Витрата необхідних ресурсів ПЛІС багато в чому визначається способом передачі інформації між компонентами системи. При передачі даних паралельним кодом витрачається великий обсяг комунікаційних засобів мікросхеми, а також виникають проблеми, пов'язані з можливою нестачею виводів мікросхеми для введення та виведення даних. При нестачі виводів можлива ситуація, коли частина функціональних ресурсів мікросхеми може виявитися незадіяною.

З огляду на важливість цієї проблеми, компанія Virtual Machine Works розробила технологію під назвою VirtualWire (віртуальні з'єднання), представлену як технологія виробництва великих систем, для реалізації яких доводиться використовувати кілька ПЛІС [3]. Ідея, закладена в основі технології, полягає в тому, що частина ресурсів мікросхем використовується для реалізації спеціальних схем, які дозволяють підключати виводи мікросхем поперемінно до різних джерел сигналів усередині ПЛІС. Дана технологія, хоча і допомагає вирішити проблему нестачі виводів, не дозволяє в повній мірі використовувати незадіяні ресурси мікросхем для вирішення безпосередньо основної задачі. Крім того, мультиплексування створює додаткові часові затримки просування потоків даних, що суперечить самій ідеї потокової моделі обчислень.

Таким чином, існує важлива проблема зменшення апаратних ресурсів ПЛІС при побудові на їх основі обчислювальних систем класу ПСБЗ. В роботі досліджується можливість вирішення даної проблеми шляхом скорочення кількості зв'язків між компонентами системи за рахунок порозрядної передачі даних.

II. ПОСТАНОВКА ЗАДАЧІ

Одним з підходів до вирішення проблеми зменшення кількості зв'язків між компонентами системи є використання квазіпаралельних ОМ, що обмінюються між собою даними послідовним кодом. Хоча на входах і виходах таких ОМ числа представлені послідовним кодом, вони по внутрішній організації ближчі до паралельних пристроїв. У зв'язку з цим вони отримали назву квазіпаралельні [1, 4]. За швидкістю виконання

арифметичних операцій (множення, ділення тощо), такі пристрої ближче до паралельних пристроїв.

Порозрядний обмін даними дозволяє зменшити необхідну кількість зовнішніх виводів і необхідний ресурс комутаційної системи ПЛІС. Підвищується ймовірність реалізації системи на одній або на меншій кількості мікросхем. У свою чергу, це дає можливість підвищення надійності систем, зменшення енергоспоживання та габаритів.

Нижче на прикладі обчислення степеневих функцій розглядаються структури на базі квазіпаралельних ОМ, що працюють в неавтономному режимі з використанням надлишкової системи числення для подання операндів та результату. Обґрунтовано вибір надлишкової системи числення. Запропоновано структуру ОМ і алгоритми виконання операцій. Показана можливість виконання залежних операцій в режимі часткового суміщення на рівні обробки розрядів операндів. Порівнюються системи на базі ОМ з послідовним та паралельним обміном інформацією. Одержано формули для оцінки часу виконання обчислень в різних системах.

III. ОРГАНІЗАЦІЯ АПАРАТНИХ ЗАСОБІВ

Узагальнену функціональну модель систем класу ПСБЗ можна подати у вигляді

$$SYS = \langle DI, DO, M, F, C \rangle,$$

де DI – множина вхідних даних; DO – множина результатів; M – множина ОМ; F – система функцій перетворення даних; C – характеристика засобів комутації.

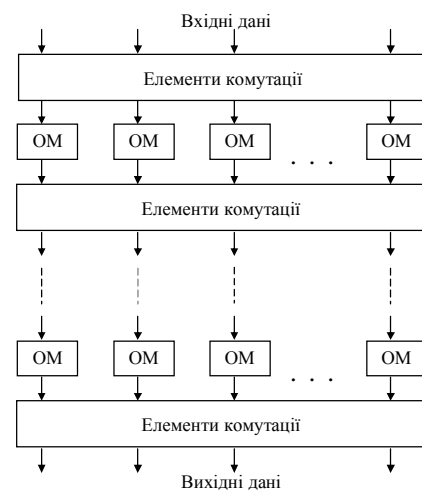


Рис. 1. Структура системи класу ПСБЗ, що реконфігурується

Система функцій F подається в вигляді алгоритмів виконання операцій в ОМ. Характеристика C визначає можливості взаємодії ОМ у просторі.

На рис. 1 показано приклад організації ПСБЗ для реалізації ярусно-паралельних форм алгоритмів. ОМ налагоджуються на виконання певних операцій, а

елементи комутації забезпечують зв'язок між ОМ. Формалізовані методики переходу від ГПД до структури ПСБЗ відомі [2, 5, 6].

В квазіпаралельних ОМ відбувається суміщення процесів порозрядного введення операндів, їх обробки і виведення результату. Це дає потенційну можливість виконувати залежні операції (результат однієї операції є операндом для іншої операції) у режимі часткового суміщення в часі.

На кожному кроці обчислень в ОМ вводиться по одному розряду операндів, починаючи зі старших, і формується один розряд результату. При цьому розряд проміжного результату, отриманий на i -му кроці в одному ОМ при виконанні j -ї операції, може бути використаний на $(i+1)$ -му кроці в іншому ОМ при виконанні $(j+1)$ -ї операції. При такому режимі обчислень виконання наступної операції буде починатися не після завершення виконання попередньої операції, а відразу ж після отримання першого розряду результату цієї операції. Для виконання послідовності операцій необхідна сумісна робота кількох ОМ, що обмінюються інформацією у процесі роботи.

ПЛІС дозволяють конфігурувати логічні блоки для роботи у якості ОМ, а також за допомогою спеціальних засобів можна сформувавши необхідний набір команд [3]. Цей факт є дуже важливим для побудови ОМ зі спеціальними алгоритмами виконання операцій, в тому числі, з використанням надлишкової системи числення.

ПСБЗ містить однотипні ОМ. Для обчислення степені чисел використовується метод бінарного дерева [7].

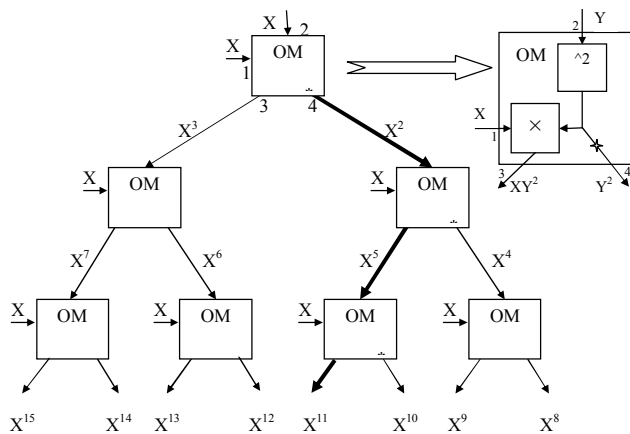


Рис. 2. Дерево однотипних модулів для обчислення степеневих функцій

На рис. 2 показано схему з'єднання ОМ, що дозволяє отримувати всі степені числа від 2-ї до 15-ї. У загальному випадку дерево, що містить m ярусів, дозволяє обчислювати функції $f(X) = X^j$, де $j = \overline{2, M}$. Максимальний показник степені визначається формулою $M = 2^{m+1} - 1$.

На рис. 2 окремо показана внутрішня організація ОМ. Модуль включає пристрій множення і пристрій для обчислення квадрата числа (позначені відповідно через « \times » і « $\wedge 2$ »). Для вирівнювання затримок формування розрядів результатів на обох виходах на відміченому зірочкою виході ОМ включені елементи затримки.

Якщо в один момент часу необхідно одержати значення будь якої із зазначених функцій, що формуються на старшому ярусі, то кількість ОМ може бути зменшена до m практично без зниження швидкодії. При обчисленні функції X^Q шлях проходження інформації крізь дерево ОМ визначається m молодшими розрядами двійкового коду показника степеня $Q = q_1 q_2 q_3 q_4$. Розряд q_i керує виходами ОМ на $(i-1)$ -му ярусі. При рівності розряду q_i одиниці в ОМ $(i-1)$ -го ярусу формується функція XY^2 (див. рис. 2), а при рівності нулю – функція Y^2 . В якості прикладу на рис. 2 потовщеною лінією показаний шлях проходження інформації через дерево ОМ при обчисленні 11-го степеня числа ($Q = 1011$).

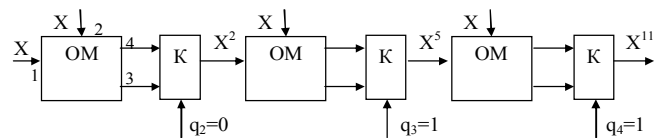


Рис. 3. Лінійне з'єднання модулів для обчислення степеневих функцій

Таким чином, для обчислення функції X^Q , де Q – ціле ($2^m \leq Q \leq 2^{m+1} - 1$), достатньо мати m ОМ, з'єднаних між собою, як вказано на рис. 3, де K – комутатор.

IV. АЛГОРИТМИ ВИКОНАННЯ ОПЕРАЦІЙ

Неавтономний режим виконання операцій зі старших розрядів можливий як у змішених, так і симетричних надлишкових системах числення [4, 8]. Наприклад, з основою $k=2$ це відповідно системи з цифрами $g = \{0, 1, 2\}$ та $g = \{-1, 0, 1\}$. В змішених системах для визначення знаку числа необхідно мати спеціальний знаковий розряд, який при виконанні операцій треба обробляти окремо, що ускладнює обробку чисел зі знаками. В симетричних системах знак числа визначається знаком старшого основного розряду числа, тобто знаковий розряд не потрібен. Всі розряди обробляються однаково, що дає можливість прискорити виконання операції. Якщо операнди можуть мати різні знаки, то доцільно виконувати операції в симетричних системах числення.

В ОМ виконуються дві операції: множення ($Z = 2^{-p} XY$) та піднесення до квадрату ($Z = 2^{-p} X^2$). В неавтономному режимі обчислень кожний операційний пристрій вносить затримку формування старшого розряду на p кроків.

Будемо вважати, що операнди є нормалізованими дробовими числами (мантисами) і представлені у формі:

$$X = \sum_{i=1}^n x_i 2^{-i}, Y = \sum_{i=1}^n y_i 2^{-i}, \quad (1)$$

де $x_i, y_i \in \{-1, 0, 1\}$ – цифри операндів, n – розрядність операндів.

В даній надлишковій системі одне і те ж саме число може мати кілька подань, наприклад, $0,1011 = 0,1\bar{1}\bar{1} = 0,11\bar{0}\bar{1}$ (символ $\bar{1}$ є більш зручної формою запису цифри -1). Це дозволяє в ОМ формувати результат порозрядно без переносів у старші розряди.

Операнди, що містять тільки i розрядів праворуч від коми, позначимо відповідно через X_i та Y_i . Наприклад, $X_3 = x_1 x_2 x_3 0 \dots 0$, $Y_2 = y_1 y_2 0 \dots 0$.

Цифри z_i результату Z формуються зі старших розрядів в такій самій системі числення порозрядно. Розряди операндів і результату передаються між ОМ за допомогою двох провідників. Цифри з множини $\{-1, 0, 1\}$ можуть кодуватись відповідно двома сигналами $\{10, 00, 01\}$.

А. Алгоритм множення.

Вимагатимемо, щоб похибка результату була знакозмінною і за абсолютною величиною не перевищувала половини ваги n -го розряду після коми. Ця вимога буде виконуватися, якщо на i -му кроці цифру z_i результату вибирати таким чином, щоб мало місце співвідношення

$$Z_i - 2^{-i-1} \leq 2^{-p} X_i Y_i < Z_i + 2^{-i-1}, \quad (2)$$

де p – число кроків латентної затримки початку формування цифр результату множення в ОМ.

Використовуючи методику, описану в [4], і формули (1), (2), можна отримати алгоритм обчислення Z у неавтономному режимі для обраної форми подання чисел.

В даному випадку $p = 2$. Для отримання n розрядів результату після коми необхідно виконати $n + 2$ кроки обчислення.

Алгоритм обробки операндів має наступний вигляд.

1. X_0, Y_0, R_0 надати значення 0.
2. Для $i = \overline{1, n+2}$ виконувати пункти 3-7.
3. $H_i = 2R_{i-1} + 2^{-2} X_{i-1} y_i + 2^{-2} Y_{i-1} x_i + 2^{-2-i} x_i y_i$.
4. $X_i = X_{i-1} + x_i 2^{-i}$.
5. $Y_i = Y_{i-1} + y_i 2^{-i}$.

$$6. \quad z_i = \begin{cases} -1, & \text{якщо } H_i < -2^{-1}; \\ 0, & \text{якщо } -2^{-1} \leq H_i < 2^{-1}; \\ 1, & \text{якщо } 2^{-1} \leq H_i. \end{cases}$$

$$7. \quad R_i = H_i - z_i.$$

В алгоритмі H_i і R_i – допоміжні змінні.

В. Алгоритм обчислення квадрату числа.

Аналогічно операції множення висунемо вимогу, щоб похибка результату була знакозмінною і за абсолютною величиною не перевищувала половини ваги n -го розряду після коми. Ця вимога буде виконуватися, якщо на i -му кроці цифру z_i результату вибирати таким чином, щоб мало місце співвідношення

$$Z_i - 2^{-i-1} \leq 2^{-p} X_i^2 < Z_i + 2^{-i-1}. \quad (3)$$

Використовуючи методику, описану в [4], і формули (1), (3), можна отримати алгоритм обчислення Z у неавтономному режимі.

1. X_0, R_0 надати значення 0.
2. Для $i = \overline{1, n+2}$ виконувати пункти 3-6.
3. $H_i = 2R_{i-1} + 2^{-1} X_{i-1} x_i + 2^{-2-i} x_i^2$
4. $X_i = X_{i-1} + x_i 2^{-i}$.
5. $z_i = \begin{cases} -1, & \text{якщо } H_i < -2^{-1}; \\ 0, & \text{якщо } -2^{-1} \leq H_i < 2^{-1}; \\ 1, & \text{якщо } 2^{-1} \leq H_i. \end{cases}$
6. $R_i = H_i - z_i$.

В. ПОРІВНЯЛЬНА ОЦІНКА СИСТЕМ

Послідовність виконання операцій у неавтономному режимі ілюструється діаграмою на рис. 4.

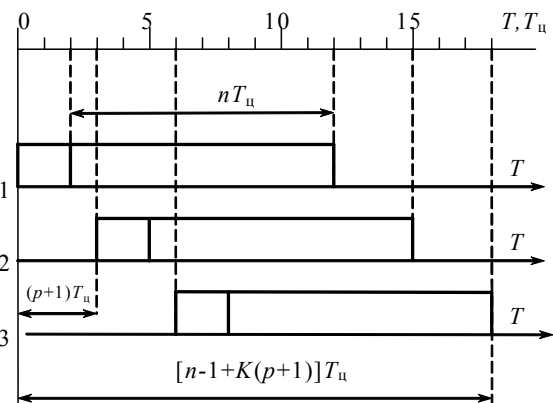


Рис. 4. Часова діаграма виконання залежних операцій в 3-х ОМ

Операції виконуються в неавтономному режимі за допомогою ланцюга ОМ, що формують цифри проміжних результатів із затримкою на p циклів. Наступний у ланцюгу ОМ починає свою операцію через $(p+1)$ циклів відносно попереднього ОМ. Це означає, що час обчислень, необхідний для отримання остаточного результату, складає

$$T = \left[n - 1 + \sum_{j=1}^K (p_j + 1) \right] t_0, \quad (4)$$

де n – розрядність результату; K – число операцій у ланцюгу; p_j – затримка формування розрядів результату j -ї операції; T_0 – тривалість циклу формування одного розряду результату в ОМ.

У випадку, що розглядається, для кожної операції затримка складає $p = 2$. Наприклад, для обчислення 15-го степеня числа необхідний ланцюг з 3-х ОМ, в кожному з яких зв'язані між собою 2 операційних пристроїв, тобто в неавтономному режимі працюють 6 пристроїв. Наприклад, для 64-розрядних операндів знадобиться виконати $64 \cdot 1 + 18 = 81$ цикл обчислень. Тривалість одного циклу визначається в основному обчисленням змінної H_i у приведених вище алгоритмах. Вказаний час визначається додаванням 3-х чисел для операції обчислення квадрату числа і 4-х чисел для операції множення за допомогою дерева комбінаційних суматорів і складає одиниці наносекунд при реалізації на ПЛІС.

При використанні для виконання послідовності залежних за даними операцій паралельних ОМ, що працюють в канонічній двійковій системі числення, час виконання ланцюга операцій буде визначатися формулою

$$T = T_0 + \sum_{i=1}^K t_i, \quad (5)$$

де t_i – час виконання i -ї операції; T_0 – додаткові витрати часу. До останніх витрат можна віднести, наприклад, витрати на вибір і розшифрування команд при програмній реалізації або на процедури обміну даними між ОМ (можливо, через загальну пам'ять), прийом операндів до пристрою, тощо.

Для кількісної порівняльної оцінки часу обчислень за формулами (4) та (5) необхідний більш детальний розгляд конкретної реалізації ОМ паралельного типу. Враховуючи, що при одночасному паралельному вводі операндів до ПЛІС витрачається багато зовнішніх виводів, навряд чи такий режим вводу є ефективним. Більш реальним є режим попереднього вводу 64-розрядних операндів у певному порядку з наступною їх обробкою у

паралельному пристрої, що додатково затримує обчислення.

Крім того, під час передачі інформації між ОМ паралельним кодом використовується багато комунікаційного ресурсу ПЛІС, тобто апаратний ресурс ПЛІС витрачається неекономно.

VI. ВИСНОВКИ

У роботі досліджена можливість підвищення ефективності систем з безпосередніми зв'язками між ОМ за рахунок реалізації неавтономних обчислень при порозрядній передачі даних між модулями системи.

У порівнянні з системами, в яких передача даних між ОМ проводиться паралельним кодом, для реалізації систем з порозрядною обробкою даних на базі квазіпаралельних ОМ витрачається менше ресурсів ПЛІС. Економляться як елементи внутрішнього ресурсу (функціональні і комунікаційні модулі, вбудовані пристрої), так і виводи ПЛІС.

Завдяки цьому збільшується ймовірність розміщення системи на одній мікросхемі. Реалізація системи на одній мікросхемі забезпечує підвищення надійності систем, зменшення енергоспоживання та габаритів.

Обмін даними між ОМ всередині мікросхеми виконується швидше, ніж між компонентами системи, що реалізовані на різних мікросхемах. Це дає потенційну можливість підвищити частоту тактування, що, у свою чергу, прискорює обробку інформації.

Таким чином, отримані результати підтверджують ефективність застосування неавтономних методів порозрядної обробки інформації в обчислювальних системах на кристалі.

ЛІТЕРАТУРА REFERENCES

- [1] В.И.Жабин, «Выполнение последовательностей зависимых операций в режиме совмещения», *Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка*, №46, с. 226-233, 2007.
- [2] В.И.Жабин, В.И.Корнейчук, В.П.Тарасенко, «Построение быстродействующих специализированных вычислителей для реализации многоместных выражений», *Автоматика и вычислительная техника*, №6, с. 18-22, 1981.
- [3] К. Максфилд, *Проектирование на ПЛИС. Архитектура, средства и методы*, М: Издательский дом «Додэка-XXI», 2007, 408 с.
- [4] V.I.Zhabin, V.I.Korneichuk, V.P.Tarasenko, "Computation of rational functions for many arguments", *Automation and Remote Control*, vol. 38, no 12, pp. 1864–1871, 1978. [В.И.Жабин, В.И.Корнейчук, В.П.Тарасенко, «Некоторые машинные методы вычисления рациональных функций многих аргументов» *Автоматика и телемеханика*, № 12, с. 145–154, 1977].
- [5] А.В.Палагин, В.Н.Опанасенко, *Реконфигурируемые вычислительные системы: Основы и приложения*, К: Просвіта, 2006, 280 с.
- [6] И.А.Каляев, И.И.Левин., Е.А.Семерников, «Архитектура семейства реконфигурируемых вычислительных систем на основе ПЛИС», *Искусственный интеллект*, № 3, с. 663-674, 2008.
- [7] Д.Кнут, *Искусство программирования для ЭВМ. Т. 2. Получисленные алгоритмы*, М: «Мир», 1977, 724 с.
- [8] И.А.Дичка, В.В.Жабина, «Совмещение зависимых операций на уровне обработки разрядов операндов», *Искусственный интеллект*, №3, с. 649-654, 2008.