

УДК 004.315

Жабін В. І., Токар А. Г.

**АНАЛІЗ ЕФЕКТИВНОСТІ РЕАЛІЗАЦІЇ
ДОДАВАННЯ В НЕАВТОНОМНОМУ РЕЖИМІ****ANALYSIS OF EFFICIENCY OF ADDITION
OPERATION REALIZATION IN ON-LINE MODE**

Розглядається алгоритм та апаратна реалізація операції додавання різної кількості операндів в неавтономному режимі. Такий режим дозволяє з використанням надлишкової системи числення поєднувати процеси порозрядного введення, обробки і порозрядного виведення даних, починаючи зі старших розрядів. Показано, що застосування багатооперандної функції додавання дозволяє зменшити затримку формування розрядів результату та апаратні витрати.

Ключові слова: неавтономні обчислення, надлишкові системичислення, суміщення залежних операцій.

Рис.: 3. Табл.: 1. Бібл.: 6.

The article deals with an algorithm and hardware implementation of two-place and multi-place functions of addition in on-line mode, which allows to combine the processes of bit-by-bit input, processing and bit-by-bit output of information, starting with the high-order bit with the use of redundant number system. It is shown that the implementation of a multi-place function enables to reduce delay in the result digit obtaining, as well as scope of equipment.

Key words: on-line computations, redundant number system, dependent operations overlap.

Fig.: 3. Tabl.: 1. Bibl.: 6.

Актуальність теми дослідження. Рішення складних завдань обробки інформації в реальному часі обумовлює застосування паралельних обчислювальних систем. Час рішення задач в таких системах залежить не тільки від часу виконання паралельних гілок алгоритмів, але і від витрат часу на обмін інформацією між обчислювальними модулями (ОМ). Зменшити витрати часу на обмін даними дозволяє використання потокових систем з безпосередніми зв'язками (ПСБЗ) між ОМ [1, 2]. Підвищення ефективності зазначених систем є актуальним завданням, що вимагає додаткових досліджень.

Постановка проблеми При побудові систем на базі ПЛІС важливою проблемою є економне використання ресурсу мікросхем (внутрішніх функціональних блоків і зовнішніх висновків) з метою зменшення числа мікросхем для реалізації обчислювальної системи. У роботі розглядається можливість вирішення цього завдання за рахунок зменшення кількості з'єднань між компонентами системи.

Аналіз останніх досліджень і публікацій. Виконання операцій методами паралельної арифметики вимагає передачі даних між ОМ багаторозрядними кодами, що не дозволяє економити елементи комутації ПЛІС. Для скорочення числа зв'язків розроблені методи квазіпаралельний арифметики [1, 2]. Обчислення в цьому випадку в кожному ОМ виконуються при порозрядному введенні операндів і порозрядному формуванні результату зі старших розрядів. Для виключення переносів в старші розряди використовуються надлишкові

системи числення. Завдяки цьому при виконанні ланцюжків залежних за даними операцій виникає можливість часткового суміщення операцій на рівні обробки розрядів операндів, що показано в ряді робіт [3-6].

Виділення недосліджених частин загальної проблеми. При виконанні ланцюжків залежних операцій кожен ВМ працює в неавтономному (on-line) режимі, приймаючи на кожному кроці чергові розряди операндів від попередніх ОМ і передаючи розряди результату наступного в ланцюжку ОМ. Зі збільшенням довжини ланцюжка операцій збільшується час обробки даних. Для скорочення кількості операцій в ланцюжку доцільно використовувати ОМ, які виконують багатомісні операції. Розробка і дослідження алгоритмів багатомісних операцій є важливим завданням підвищення ефективності виконання залежних операцій.

Постановка завдання. У даній роботі на прикладі операції додавання досліджується ефективність реалізації багатомісних операцій при порозрядної обробці операндів. Розглядається апаратна реалізація операції при різному числі операндів. Операція складання кількох чисел може бути корисна при цифровій обробці сигналів, реалізації штучних нейронів, обробці таблиць даних.

Викладення основного матеріалу. У роботі [2] запропонований метод складання двох чисел поданих в надлишкової системі числення з природним порядком ваг, основою $k=2$ і цифрами з множини $\{-1,0,1\}$. При виконанні ланцюжка залежних за даними операцій ОМ працюють синхронно наступним чином [2]. На кожному кроці ОМ приймає по одному розряду операндів з попереднього в ланцюжку ОМ і формується один розряд результату для наступного ОМ. Операція виконується зі старших розрядів. Отриманий розряд результату не вимагає корекції, оскільки в надлишкової системі числення переноси в старші розряди результату не поширюються.

Показано [2], що розряди результату формуються з затримкою на $p=2$ циклів, тобто ОМ виконує двомісну операцію $Z = 2^{-p} \cdot (X + Y)$, причому, операнди є правильними r -розрядними дробовими числами:

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

де $x_i, y_i \in \{-1,0,1\}$.

Для отримання n розрядів функції $Z^* = (X + Y)$ необхідно сформувати $m=r+p$ розрядів функції $Z = 2^{-p} \cdot (X + Y)$. Після виконання m кроків буде отримана функція з похибкою, яка за модулем не перевищує k^{-m-1} .

Алгоритм виконання i -го циклу ($i = 1, \dots, m$) зводиться до виконання наступних обчислень [2].

По формулі

$$H_i = 2R_{i-1} + 2^{-2}x_i + 2^{-2}y_i \quad (1)$$

знаходиться проміжна перемінна H_i (початковим є значення $R_0=0$).

Цифра результату формується за правилом

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

Для виконання наступного циклу обчислюється

$$R_i = H_i - z_i. \quad (3)$$

Розглянемо модифікацію алгоритму для додавання в неавтономному режимі K r -розрядних чисел X_j ($j=1, \dots, K$), тобто обчислення функції

$$Z = 2^{-p} \sum_{j=1}^K X_j, \text{ де } X_j = \sum_{i=1}^r x_{ji} 2^{-i}.$$

Якщо в кожному i -му циклі буде виконуватися співвідношення

$$Z_i - 2^{-i-1} \leq 2^{-p} \sum_{j=1}^K X_j < Z_i + 2^{-i-1}, \quad (4)$$

то після виконання $m = n + p$ циклів буде одержано результат з похибкою, що за модулем не перевищує 2^{-m-1} .

Ввівши заміну

$$R_i = (2^{-p} \sum_{j=1}^K X_j - Z_i) 2^i, \quad (5)$$

вираз (4) можна подати у формі

$$-2^{-1} \leq R_i < 2^{-1}. \quad (6)$$

Нерівність (6) виконується у вихідному стані при $R_0 = 0$. Будемо вважати, що на $(i-1)$ -му кроці нерівність (6) також виконується, та визначимо при якому мінімальному значенні p це співвідношення буде виконуватись на будь-якому наступному кроці.

Виділимо в (5) значення R_{i-1} і зробимо заміну

$$H_i = 2R_{i-1} + \sum_{j=1}^K x_{ji}. \quad (7)$$

Тоді вираз (6) приймає вигляд

$$z_i - 2^{-1} \leq H_i < z_i + 2^{-1}. \quad (8)$$

Аналізуючи вирази (6), (7), (8) та враховуючи допустимі значення цифр операндів, отримуємо значення затримки

$$p \geq 1 + \log K, \quad (9)$$

при якій виконується умова (6) в будь-якому циклі, тобто процес обчислень буде збіжним. Звичайно, треба вибрати мінімальне ціле значення затримки.

Підставляючи в рівність (8) допустимі значення $z_i \in \{-1, 0, 1\}$, отримаємо правило формування цифри результату, яке співпадає з правилом (2) для двомісної функції додавання.

Отже, при додаванні K чисел в кожному циклі потрібно по формулі (7) знайти значення H_i , визначити цифру функції за допомогою правила (2), та знайти значення R_i для наступної ітерації за формулою (3).

Розглянемо апаратну реалізацію додавання в надлишкових системах числення.

Для побудови ВМ використовується звичайна двійкова апаратура. Цифри $-1, 0, 1$ кодуються двома двійковими розрядами, наприклад, відповідно 10, 00, 01 та пересилаються між ВМ по двом провідникам.

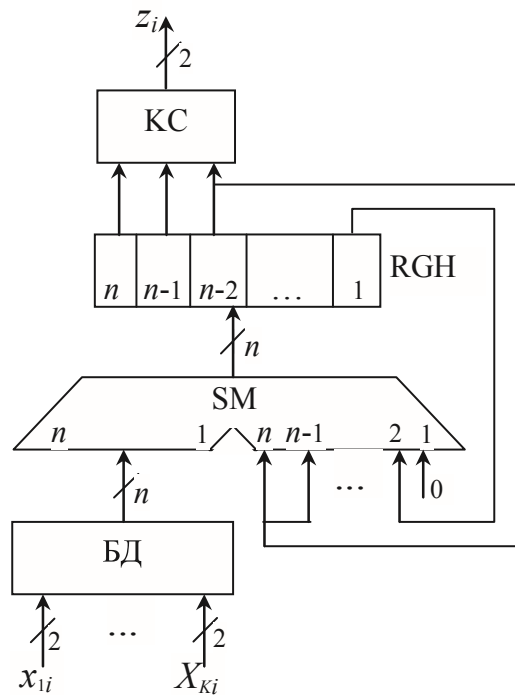


Рис. 1. Структура обчислювального модуля

Схема пристрою (рис. 1) містить регістр RGH для зберігання H_i . Розрядність регістра визначається формулою $n=2+p=3+\log_2 K$, де $\log_2 K$ відповідає максимальній розрядності прирощення для H_i . Регістр має 2 знакових розряди. Значення H_i формується в доповняльному коді відповідно з формулою (7). Блок додавання (БД) формує суму i -х розрядів всіх операндів, що подаються в ВМ в i -му циклі. Кінцеве значення H_i формується на суматорі (SM). На рис. 2 для випадку $K=2$ показано перетворення цифр операндів -1 та 1 в n -розрядний доповняльний код, що подається на вхід суматора.

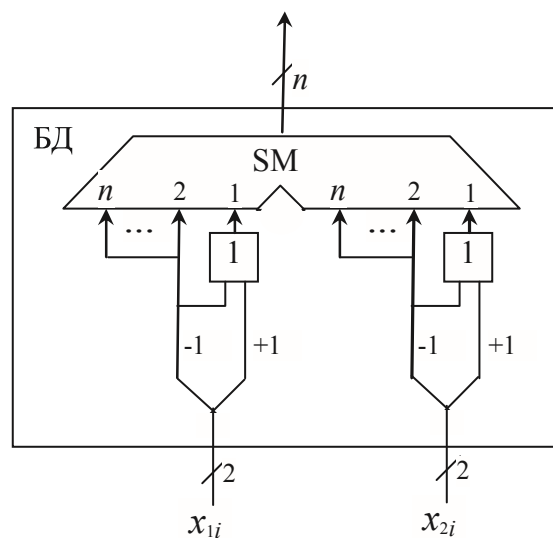


Рис. 2. Блок додавання

Значення розряду результату Z_i визначається комбінаційною схемою (КС), яка аналізує три старші розряди коду H_i відповідно з таблицею істинності (табл. 1).

Таблиця 1.

Таблиця істинності функцій

H_i	Функція $Z_i=-1$	Функція $Z_i=1$
10,1... 11,0...	1	0
00,0... 11,1...	0	0
00,1... 01,0...	0	1
01,1... 10,0...	-	-

Віднімання Z_i відповідно з формулою (3) замінюється просторовим функціональним зсувом на регістрі RGH. Як показано на рис. 1, вихід $(n-2)$ -го розряду переписується водночас в n -й та $(n-1)$ -й розряди регістру.

Розглянемо два варіанти реалізації додавання декількох чисел в неавтономному режимі.

В першому варіанті сума K чисел визначається за допомогою бінарного дерева ВМ, призначених для підсумування двох чисел. Дерево має $N = \log_2 K$ ярусів однакових ВМ, що працюють в неавтономному режимі. тобто ланцюжок ВМ, що відповідає максимальній затримці формування розрядів результату, також складає $N = \log_2 K$ модулів. Загальна кількість ВМ у бінарному дереві, як відомо, складає суму перших $N = \log_2 K$ членів геометричної прогресії із знаменником 2 та дорівнює $M = 2^{\log_2 K} - 1 = K - 1$.

В другому варіанті всі K чисел обробляються в одному ВМ, що реалізує K -місну функцію додавання.

Можна порівняти приблизно апаратні витрати на реалізацію двох варіантів додавання. В першому випадку M блоків БД в складі ВМ містять по одному суматору (рис.2), а в другому варіанті один БД містить дерево із M суматорів, тобто на БД витрачається приблизно однаковий склад апаратури.

Вся інша апаратура, що є в одному ВМ за другим варіантом, в системах за першим варіантом повторюється в кожному ВМ, тобто M разів. Слід зауважити, що розрядність суматорів і регістрів в різних ВМ може несуттєво відрізнятися, але це не впливає на висновок, що перший варіант програє другому за складністю.

Порівняємо час виконання обчислень в системах за двома варіантами у передбаченні, що різні ВМ працюють в складі системи в синхронному режимі, тобто цикл обчислень в різних ВМ однаковий. Такий режим є звичайним для ПСБЗ, що реалізовані на ПЛІС.

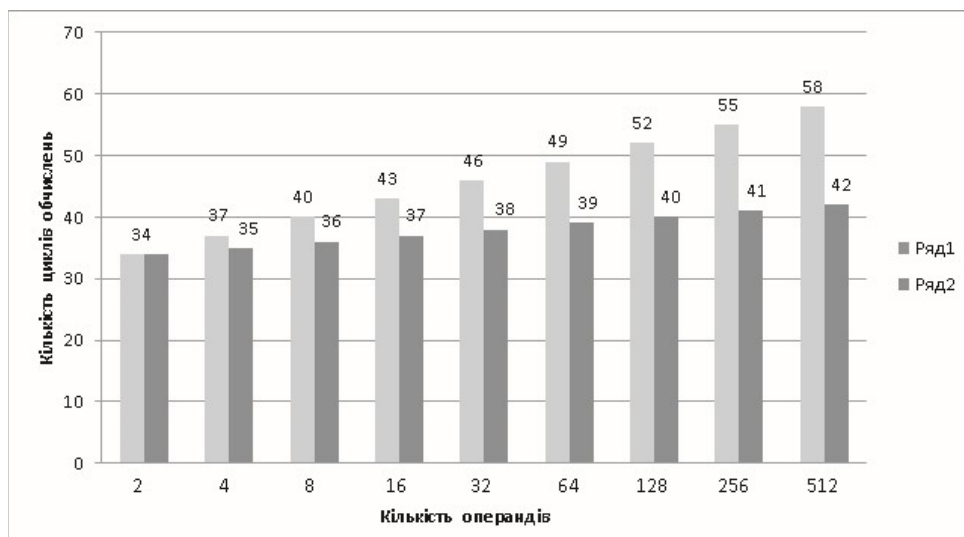


Рис. 3. Залежність кількості циклів обчислень від кількості операндів: ряд 1 – для використання двомісних функцій; ряд 2 – для K -місних функцій

Час виконання послідовності з M операцій в неавтономному режимі складає [4]

$$T = [m - 1 + M(p + 1)]t_0, \quad (10)$$

де t_0 – тривалість циклу.

При $m = 32$ для двомісної та K -місної функції додавання формула (10) приймає відповідно вигляд:

$$T = (31 + 3\log_2 K)t_0, \quad T = (32 + 1 + \log_2 K)t_0$$

З гістограми (рис. 3) видно, що для будь-якого $K > 2$ використання багатомісної функції додавання забезпечує виграв в швидкодії відносно двомісних функцій. Виграш в швидкодії зростає із збільшенням місності функції додавання.

Висновки. Запропоновано алгоритм та реалізація багатомісної функції додавання чисел в неавтономному режимі з використанням надлишкової симетричної системи числення з цифрами $\{-1, 0, 1\}$ та основою $k = 2$. Операція виконується порозрядно зі старших розрядів із суміщенням процесів введення, обробки та виведення розрядів чисел. Це дозволяє частково поєднувати залежні операції в різних модулях, тобто виконувати ланцюжки залежних операцій в режимі суміщення на розрядному рівні.

Показано, що реалізація багатомісної функції додавання дозволяє прискорити обчислення та зменшити апаратні витрати. Обмін даними між вузлами системи послідовним кодом зменшує необхідні функціональні та комутаційні ресурси ПЛІС (функціональні комірки та комірки введення-виведення), що, в свою чергу, підвищує надійність систем.

Список використаних джерел

1. Zhabin V.I. Design of High-Speed Specialized Computers for the Realization of Many-Place Expressions / V.I.Zhabin, V.I.Korneichuk, V.P.Tarasenko // Automatic Control and Computer Sciences. – 1981. – vol. 15, no 6. – P. 15-18.
2. Zhabin V.I. Computation of Rational Functions for Many Arguments / V.I.Zhabin, V.I.Korneichuk, V.P.Tarasenko // Automation and Remote Control. – 1978. – vol. 38, no 12. – P. 1864-1871.
3. Дичка И.А. Совмещение зависимых операций на уровне обработки разрядов операндов / И.А.Дичка, В.В.Жабина // Искусственный интеллект. – 2008. – №3. – С. 649-654.
4. Жабин В.И. Выполнение последовательностей зависимых операций в режиме совмещения / В.И.Жабин // Вісник Національного технічного університету України “КПІ”. „Інформатика, управління та обчислювальна техніка”. – 2007. – №46. – С. 226-233
5. Максфилд К. Проектирование на ПЛИС. Архитектура, средства и методы / К. Максфилд. – М.: Издательский дом «Додэка-XXI», 2007. – 408 с.
6. Жабин В.И. Реализация неавтономных вычислений в избыточных системах счисления на ПЛИС / В.И. Жабин, В.В. Жабіна, А.В. Скоріченко // Вісник НТУУ "КПІ". Інформатика, управління та обчислювальна техніка. – 2016. – №64. – С. 150-155.

ДОВІДКА ПРО АВТОРІВ

Жабін Валерій Іванович – професор, кафедра обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського».

Valerii Zhabin – professor, Department of Computer Engineering, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

E-mail: viz.kpi@gmail.com

Токар Андрій Геннадійович – студент, кафедра обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського».

Andrii Tokar – student, Department of Computer Engineering, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

E-mail: andrey-naum@ukr.net

**Zhabin V.I.,
Tokar A.H.**

ANALYSIS OF EFFICIENCY OF ADDITION OPERATION REALIZATION IN ON-LINE MODE

Relevance of the research topic. The solution of complex tasks of information processing in real time determines the use of parallel computing systems. In such systems the time of solving problems depends not only on the time of execution of parallel branches of algorithms, but also on the time spent on the exchange of information between computing modules. Reducing the time spent on data exchange is an urgent task.

Problem statement. When building systems based on FPGA, an economical use of the microcircuit resource (internal functional blocks and external pins) is important in order to reduce the number of chips for the implementation of the computing system. The paper considers the possibility of solving this problem by reducing the number of connections between components of the system.

Analysis of recent researches and publications. Performing operations using parallel arithmetic requires data transfer between blocks with multi-bit codes. This does not save the switching elements of the FPGA. To reduce the number of links, quasi-parallel arithmetic methods have been developed. Calculations in this case in each block are performed with bitwise introduction of operands. To exclude transfers to the upper digits, redundant number systems are used. When performing chains dependent on data operations, it becomes possible to overlap operations at the processing level of the operand bits. This is shown in a number of works. **Singling out of unexplored parts of the general problem.** When performing the chain of dependent operations, each module works in on-line mode. With the increase in the length of the chain of operations, the processing time increases. To reduce the number of operations in a chain, it is advisable to use modules that perform multiple operations.

Task definition. In the present paper, the efficiency of realizing a multi-operand function in the case of redundant data representation is shown.

Statement of the main material. The algorithm and hardware implementation of the addition function in on-line mode using redundant number systems is considered. A comparative analysis of the speed and complexity of devices with different number of operands is performed. The result of comparison showed greater efficiency of devices with a large number of operands.

Key words: on-line computations, redundant number system, dependent operations overlap