УДК 004.2 (045)

І.А. Жуков, д-р техн. наук
В.О. Гуменюк, канд. техн. наук

СКОРОЧЕННЯ АПАРАТНИХ ВИТРАТ У МАТРИЧНИХ СУМАТОРАХ У КОДІ «*M* з *N*» НА ОСНОВІ СПЕЦІАЛЬНИХ АЛФАВІТІВ

Кафедра обчислювальної техніки, НАУ, e-mail: iit@nau.edu.ua

Запропоновано методику складання спеціальних алфавітів за умов використання коду «M з N» для подання цифрової інформації в комп'ютері. Застосування такої методики сприяє значному скороченню апаратних витрат у процесі реалізації матричних операційних блоків обчислювальних пристроїв.

Вступ

Підвищення вимог до обчислювальних засобів, пов'язаних із проектуванням літаків і ракет, фундаментальними науковими дослідженнями, прогнозуванням погоди та природних катаклізмів, у найближчі чотири роки передбачає значне збільшення їх продуктивності ($\geq 10^{25}$ Flops), внаслідок чого виникає проблема реалізації масових паралельних обчислювальних процесів. Ці вимоги перевершили можливості одного процесора, що працює за принципом фон-Неймана [1].

Постановка проблеми

Практичне розв'язання проблеми підвищення продуктивності обчислювальних систем зазвичай пов'язують, у першу чергу, зі збільшенням тактової частоти роботи елементів і кількості цих елементів, що дозволяє вводити паралелізм обробки й програмовість структур. Однак удосконалювання обчислювальних систем завжди супроводжувалося розривом між швидкодією логічних елементів та елементів пам'яті. Цей розрив під час зростання ступеня інтеграції й швидкодії надвеликих інтегральних схем має тенденцію до збільшення. На кожному рівні розвитку елементної бази через обставини, обумовлені необхідністю подолання даного розриву, обмеження на розмір і кількість виводів у корпусах мікросхем, наявні засоби автоматизації програмування одні архітектури одержували переваги над іншими, наприклад, за показником продуктивність/вартість.

Однак, як показує порівняльний аналіз характеристик архітектур найбільш продуктивних паралельних обчислювальних систем, поряд з перевагами кожна з них має істотні недоліки. Результатом цього є існуюча проблема реалізації масових паралельних обчислювальних процесів.

Мета даної статті – пошук додаткових можливостей істотного підвищення загальної продуктивності паралельних обчислювальних систем незалежно від їхньої архітектури. Розглянемо особливості подання й зберігання інформації в ЕОМ і мікропроцесорних системах.

Традиційний спосіб подання двійкової інформації в ЕОМ, при якому обидві цифри двійкового розряду представлені одним тригером (взаємозалежне подання) призвів до втрати природної здатності до контролю позиційних числень, що, у свою чергу, обумовило використання в ЕОМ різних надлишкових кодових побудов, які дозволяють виявляти або виправляти помилки. Але за такого принципу побудови апаратного контролю не може бути забезпечене безвідмовне функціонування органів контролю [2].

Застосування нероздільних кодів «*M* з *N*»

У літературі [2–5] подання цифрової інформації в обчислювальних системах як альтернативу традиційному двійковому поданню пропонується використовувати код «*M* з *N*», де *N* – кількість позицій (кожна з яких представлена окремим тригером) у розряді числа; *M* – кількість «одиниць» у цих позиціях (інші $N - M$ позицій містять «нулі»).

Таким чином, співвідношення «нулів» і «одиниць» у межах даного коду фіксоване. Такі коди відносяться до нероздільних, в яких розряди кодового слова неможливо розділити на інформаційні й контролюючі. Зміни відповідності однорозрядних кодових комбінацій і цифр обраної основи *p* системи числення приводять до утворення інших алфавітів.

Як видно з праць [2; 3], при використанні коду «*M* з *N*» можливе практичне розв'язання проблеми автоматичного контролю в обчислювальних системах. Максимальна кількість комбінацій в одному розряді (вибір основи числення) обчислюється за формулою

$$C_N^M = N! / [M! * (N - M)!],$$

де $N! = N * * (N - 1) * (N - 2) * \ldots * 1...$

Основу числення *p* вибирають з умови

$$C_N^M \geq p.$$

Подання даних у паралельних обчислювальних системах у коді «*M* з *N*» сприяє підвищенню продуктивності кожного процесора й загальної продуктивності системи.

При цьому практично розв'язується проблема автоматичного контролю процесорів [2; 3].

Вибір основи числення з множини $p > 2$ за умови $M \approx N/2$ дозволяє скоротити апаратурну надмірність S_p.

Особливістю структури суматора у коді «*M* з *N*» є те, що кожний його розряд містить матрицю 2*M*-вхідних логічних елементів І.

Авторами статті запропоновано спосіб перетворення прямокутної матриці додавання, що містить 2*M*-вхідних логічних елементів І, у неповну трикутну матрицю додавання з такими самими логічними елементами. У короткому викладі суть способу полягає в такому.

Пари кодових комбінацій операндів X_{n1} і Y_{n2} подані набором сигналів:

$$X_{n1} + Y_{n2} = \bigcap_{(m)} x_{n1,m}\, y_{n2,m} \bigcap_{(k)} J_{n1n2,k}, \qquad (1)$$

де $x_{n1,m}$ й $y_{n2,m}$ – сигнали, які представляють «одиниці» в *m*-х позиціях цих комбінацій, причому $0 \le m \le N-1$; $0 \le n1 \le p-1$; $0 \le n2 \le p-1$; $J_{n1n2,k}$ – сигнали, які відповідають номерам *k*-х позицій кодових комбінацій, в яких утримуються протилежні (0 й І) сигнали, тобто «одиниці» у *k*-й позиції однієї комбінації відповідає «нуль» у тій самій позиції іншого кодового слова.

Виходячи з умов складання наборів сигналів (1) необхідно, щоб

$$\bigcap_{(m)} x_{p-1,m}\, y_{1,m} \bigcap_{(k)} J_{(p-1)1,k} = \bigcap_{(m)} x_{1,m}\, y_{(p-1),m} \bigcap_{(k)} J_{1(p-1),k};$$

$$\bigcap_{(m)} x_{p-2,m}\, y_{2,m} \bigcap_{(k)} J_{(p-2)2,k} = \bigcap_{(m)} x_{2,m}\, y_{(p-2),m} \bigcap_{(k)} J_{2(p-2),k} \text{ і т.п.}$$

Тому максимальна кількість $K_н$ компіляцій сигналів, що являє собою суму часткових кількостей компіляцій значень $\sum_{(m)} x_{n1,m} y_{n2,m}$ і $\sum_{(k)} J_{n1n2,k}$, обчислюється як

$$K_н = [\sum_{i=1}^{M} C_N^i C_{N-i}^{2(M-i)}] + C_N^{2M}. \qquad (2)$$

Кількість елементів І, що входять до складу трикутної матриці додавання, визначається як

$$K_{M\Delta} = p(p+1)/2. \qquad (3)$$

З порівняльного аналізу виразів (2) і (3) випливає, що для $N \ge 4$ справедлива нерівність $K_{M\Delta} > K_H$. Отже, існує необхідність подальшого дослідження особливостей синтезу таких матриць при $N \ge 4$.

Методика складання спеціальних алфавітів у коді «*M* з *N*»

Сутність такої методики полягає в наступному.

1. Уводиться довільний алфавіт у даному коді як вихідний.

2. Для кожної кодової комбінації визначається сума S_H номерів позицій, що містять «одиниці»:

$$S_H = \sum_{(n)} m_n,$$

де m_n – номер позиції, що містить «одиницю» в *n*-му кодовому слові:

$$0 \le n \le C_N^M - 1.$$

3. У тому разі, коли $M \ne N/2$ й C_N^M – парне число, визначається загальна сума *S* номерів позицій, що містять «одиниці» у всіх кодових комбінаціях, що входять у базовий алфавіт

$$S = \sum_{n=0}^{C_N^M - 1} S_n.$$

4. Кодові комбінації переміщаються так, щоб для всіх пар кодових комбінацій, симетрично розташованих в алфавіті, виконувалася умова

$$S_{n1} + S_{n2} = 2S / C_N^M,$$

де S_{n1} й S_{n2} – суми номерів позицій, що містять «одиниці» першого і другого кодових слів, які утворюють пари відповідно.

Причому при $M = N/2$ алфавіт складається за принципом самодоповнення до «одиниць» у всіх позиціях.

Наприклад, для коду «2 з 4», p = 6 такі пари кодових комбінацій можуть мати вигляд:

0 – 0011 й **5** – 1100;
1 – 0101 й **4** – 1010;
2 – 0110 й **3** – 1001.

5. При $M \ne N/2$ визначається підмножина наборів, що складаються тільки із сигналів

$$\bigcap_{(m)} x_{n1,m}\, y_{n2,m} \text{ й } \bigcap_{(k)} J_{n1n2,k},$$

причому

$$\sum_{(k)} J_{n1n2,k} = N - 2.$$

6. Кожному набору підмножини ставиться у відповідність група композицій операндів, кодові комбінації яких відповідно до алфавіту можуть утворювати такий набір сигналів.

7. Послідовними взаємними переміщеннями кодових комбінацій, яким, як правило, відповідають рівні суми S_n, визначають такий алфавіт, при якому загальна кількість композицій операндів, правильно об'єднаних у групи, є максимальною.

8. Кодові комбінації, до подальших переміщень яких ставляться суперечливі вимоги, з алфавіту вилучаються.

9. Кодові комбінації, що залишилися, зрушуються вбік нуля на місця, що звільнилися в послідовності, якщо виключені кодові комбінації не відповідали найбільшим цифрам.

Розглянемо приклад складання спеціального алфавіту для коду «2 з 5», що має 10 кодових комбінацій. Отже, початково $p = 10$.

Уведемо довільний алфавіт (табл. 1).

Таблиця 1

Довільний алфавіт

n	Позиції					S_n
	4	3	2	1	0	
0	0	0	0	1	1	1
1	1	0	0	0	1	4
2	0	1	1	0	0	5
3	0	0	1	0	1	2
4	0	1	0	1	0	4
5	0	0	1	1	0	3
6	1	1	0	0	0	7
7	1	0	1	0	0	6
8	1	0	0	1	0	5
9	0	1	0	0	1	3

Визначимо S_n й S

$S_{n1} + S_{n2} = 2 * 40/10 = 8;\ S = 40.$

Відповідно до п. 4 уведемо новий алфавіт (табл. 2).

Таблиця 2

Проміжний варіант алфавіту

n	Позиції					S_n
	4	3	2	1	0	
0	0	0	0	1	1	1
1	0	0	1	0	1	2
2	0	1	0	0	1	3
3	1	0	0	0	1	4
4	0	0	1	1	0	3
5	0	1	1	0	0	5
6	0	1	0	1	0	4
7	1	0	0	1	0	5
8	1	0	1	0	0	6
9	1	1	0	0	0	7

Оскільки $M \neq N/2$, визначимо підмножину наборів, що містять тільки символи:

$\bigcap_{(k)} J_{n1n2,k}$: $J_3J_2J_1J_0$; $J_4J_3J_2J_1$;

$J_4J_3J_2J_0$; $J_4J_3J_1J_0$; $J_4J_2J_1J_0$.

Відповідно до п. 6 маємо:

а) $J_3J_2J_1J_0 \rightarrow \underline{0\text{ і }5,\ 1\text{ і }6,\ 2\text{ і }4}$;

б) $J_4J_3J_2J_1 \rightarrow \underline{4\text{ і }9,\ 5\text{ і }7,\ 6\text{ і }8}$;

в) $J_4J_3J_2J_0 \rightarrow 1\text{ і }9,\ 2\text{ і }8,\ \underline{3\text{ і }5}$;

г) $J_4J_3J_1J_0 \rightarrow 0\text{ і }9,\ 2\text{ і }7,\ 3\text{ і }6$;

д) $J_4J_2J_1J_0 \rightarrow 0\text{ і }8,\ 1\text{ і }7,\ \underline{3\text{ і }4}$.

Тут підкреслено пари, що підлягають переміщенням.

Найкраще сполучення пар операндів, як показує аналіз відповідностей **а** і **б**, відбувається при переміщенні кодових комбінацій, що представляють цифри 5 і 6. Отже:

а) $J_3J_2J_1J_0 \rightarrow 0\text{ і }6,\ 1\text{ і }5,\ 2\text{ і }4$;

б) $J_4J_3J_2J_1 \rightarrow 4\text{ і }9,\ 6\text{ і }7,\ 5\text{ і }8$;

в) $J_4J_3J_2J_0 \rightarrow 1\text{ і }9,\ 2\text{ і }8,\ \underline{3\text{ і }6}$;

г) $J_4J_3J_1J_0 \rightarrow 0\text{ і }9,\ 2\text{ і }7,\ \underline{3\text{ і }5}$;

д) $J_4J_2J_1J_0 \rightarrow 0\text{ і }8,\ 1\text{ і }7,\ \underline{3\text{ і }4}$.

Перестановка кодової комбінації, що представляє цифру 3, не приводить до поліпшення результату сполучення, тому вилучимо це кодове слово з подальшого розгляду, а кодові слова, які наведені у табл. 2 нижче, ніж це слово, переміщуємо вгору. Тоді дістанемо:

а) $J_3J_2J_1J_0 \rightarrow 0\text{ і }5,\ 1\text{ і }4,\ 2\text{ і }3$;

б) $J_4J_3J_2J_1 \rightarrow 3\text{ і }8,\ 4\text{ і }7,\ 5\text{ і }6$;

в) $J_4J_3J_2J_0 \rightarrow 1\text{ і }8,\ 2\text{ і }7$;

г) $J_4J_3J_1J_0 \rightarrow 0\text{ і }8,\ 2\text{ і }6$;

д) $J_4J_2J_1J_0 \rightarrow 0\text{ і }7,\ 1\text{ і }6$.

У табл. 3 наведено остаточний скорочений варіант алфавіту при $p = 9$.

Таблиця 3

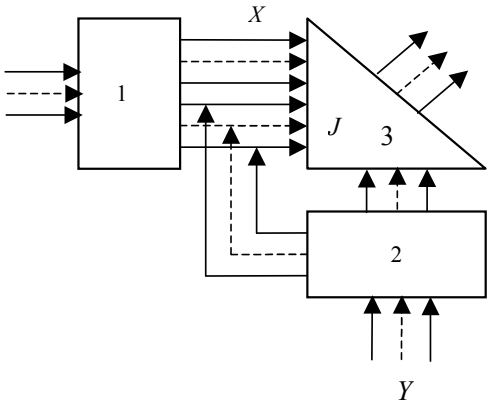
Остаточний варіант алфавіту

n	Позиції					S_n
	4	3	2	1	0	
0	0	0	0	1	1	1
1	0	0	1	0	1	2
2	0	1	0	0	1	3
3	0	0	1	1	0	3
4	0	1	0	1	0	4
5	0	1	1	0	0	5
6	1	0	0	1	0	5
7	1	0	1	0	0	6
8	1	1	0	0	0	7

На основі цього алфавіту можна побудувати одноступінчасту неповну трикутну матрицю додавання, що містить 38 чотиривхідних елементів І, замість 81 таких самих елементів, які містить аналогічна прямокутна матриця додавання. За необхідності отриманий алфавіт можна скоротити (наприклад, для одержання основи $p = 8$) вилученням будь-якої кодової комбінації й наступного зрушення інших комбінацій убік нуля на вільні місця.

Із застосуванням такої методики складено спеціальний алфавіт для коду «3 з 6», що дозволяє при основі числення $p = 16$ синтезувати одноступінчасту матрицю додавання, що містить тільки 103 елементи І (замість 256 таких самих елементів, що містить відома прямокутна матриця).

Структуру пристрою, що включає подібну матрицю, подано на рисунку.



Структура *i*-го розряду суматора з неповною трикутною матрицею додавання

Особливістю регістрів *1* і *2* операндів є те, що кожний їхній тригер має два ідентичних, але гальванічно незалежних «одиничних» виходи.

Матриця *3* додавання має три групи входів: *X, Y, J*. Діагональні шини, об'єднуючи виходи елементів I, що відповідають однаковим результатам (з урахуванням переносу) додавання, з'єднані з виходами матриці *3*.

Кількість входів елемента I матриці дорівнює 2*M*.

Уведений такий алфавіт:

0→100011, **1**→100101, **2**→111000, **3**→110010, **4**→110100, **5**→101010, **6**→101100, **7**→100110, **8**→011001, **9**→010011, ***A***→010101, ***B***→001011, ***C***→001101, ***D***→000111, ***E***→011010, ***F***→011100.

Висновок

Позитивний ефект від застосування таких спеціальних алфавітів полягає в скороченні апаратурних витрат щонайменше у два рази, а також у підвищенні швидкодії пристрою внаслідок зменшення паразитних реактивностей та геометричних розмірів матриць додавання.

Література

1. *Бурцев В.С.* Новые подходы к оценке качества вычислительных средств // Параллелизм вычислительных процессов и развитие архитектуры суперЭВМ. – М.: Нефть и газ, 1997. – С. 28–40.

2. *Брюхович Е.И.* О проблеме автоматического контроля в ЭВМ и контролеспособности позиционных счислений // УСиМ. – 1977. – №4. – С. 71–75.

3. *Брюхович Е.И., Гуменюк В.А.* Исследование достоверности результатов вычислений при представлении чисел в ЭВМ «*M* из *N*» кодом // Техническая диагностика электронных систем: Сб. науч. тр. – К.: Наук. думка, 1982. – 172 с.

4. *Гуменюк В.А., Жуков И.А., Гуменюк А.В.* Применение неразделимых кодов «*M* из *N*» в высокопроизводительных параллельных вычислительных системах // Пробл. інформатизації та управління: Зб. наук. пр. – К.: НАУ, 2004. – Вип. 3(14). – С. 256–263.

5. *Жуков І.А., Гуменюк В.О.* Представлення двійкової інформації парафазним кодом у високопродуктивних паралельних обчислювальних системах // Вісн. НАУ. – 2005. – №1. – С. 13–18.

Стаття надійшла до редакції 19.09.05.

И.А. Жуков, В.А. Гуменюк

Сокращение аппаратных затрат в матричных сумматорах в коде «*M* из *N*» на основе специальных алфавитов

Предложена методика составления специальных алфавитов при условии использования кода «*M* из *N*» для представления цифровой информации в компьютере. Применение такой методики способствует значительному сокращению аппаратных затрат при реализации матричных операционных блоков вычислительных устройств.

I.A. Zhukov, V.A. Gumenyuk

Shortening hardware expenditures in matrix adders in «M out of N» code on the base of special alphabets

A method of constituting specialized alphabets while using «M out of N» code for representing digital information in computers is proposed. Using such alphabets leads to substantial shortening hardware expenditures while realizing matrix operational units of computing devices.