УДК 004.043

**Гамаюн В.П.,** д-р техн. наук
**Андрєєв А.А.**
**Чайка М.П.**

# ПРИСКОРЕННЯ ОБЧИСЛЕННЯ АДДИТИВНО-МУЛЬТИПЛІКАТИВНИХ ОПЕРАТОРІВ ПРИ РОЗРЯДНО-ЛОГАРИФМІЧНОМУ КОДУВАННІ

**Інститут комп'ютерних технологій**
**Національного авіаційного університету**

*Розглянуто моделі виконання арифметичних базових операцій при розрядно-логарифмічному кодуванні даних, яке забезпечує обробку у великому діапазоні чисел та значне зменшення впливу округлення. Запропоновано підвищити швидкість виконання базових операцій за рахунок градієнтного розміщення даних при реалізації операцій сортування, які є складовою частиною кожної базової операції*

## *Вступ*

Арифметика чисел великого діапазону є актуальною для розв'язання задач, які чутливі до похибок та потребують безпомилкових обчислень при перетворенні багаторозрядних структур даних. Вимога до високої точності ЕОМ набуває особливого значення при вирішенні класу погано обумовлених задач, де не допускається накопичення помилок округлення. Багаторозрядна арифметика з використання нових систем числення є одним з напрацьованих напрямків розв'язання проблеми реалізації безпомилкових обчислень. Розробка та вдосконалення арифметико-алгоритмічного базису є своєчасним завданням при впровадженні нових засобів обробки на комп'ютерах.

## *Постановка проблеми*

Комп'ютерна арифметика, що пропонується, повинна відповідати характеристикам відомих розв'язань, які вже знайшли впровадження на практиці. Багаторозрядна арифметика, що пропонується як альтернативна до відомої двійкової арифметики має арифметико-алгоритмічні відмінності, які впливають на час реалізації операцій. Такими відмінностями є операції сортування цифрових кодів при виконанні-обчисленні адитивно-мультиплікативних операторів. Наближення часу реалізації багаторозрядних операцій до двійкових складає проблему у розвитку сучасних ЕОМ.

## *Методика досліджень*

Багаторозрядна арифметика реалізується з використанням нетрадиційних систем числення. Розрядно-логарифмічна система числення забезпечує розширення діапазону даних та обробку з використанням арифметики логарифмів. Така система є розширенням двійкової системи числення за умов кодування кожного ненульового розряду двійкового операнду. Такий код дорівнює значенню логарифму від ваги цього ненульового розряду:

$$N_i = \log_2 a_i p^i.$$

Нехай $A$ – двійкове число з розрядністю $n$ представлене у форматі з фіксованою комою:

$$A = \sum_i a_i p_i.$$

де $a_i = \{0,1\}$, $p = 2$ – основа системи числення.

Кожен не нульовий розряд $a_{\text{i}}$ , представляється у вигляді номеру позиції (розряду) $N_i$:

$$A \to A^N = \{N_n \ldots N_2, N_1\}$$

Наприклад: $10001101_2$=7.3.2.0$_{\text{РЛ}}$.

Крім того РЛ кодування забезпечує розширення діапазону представлення чисел, що особливо важливо при обрахунку даних, де точність є критичною характеристикою. В табл. 1 представлена порів-

няльна характеристика двійкового та РЛ кодування по діапазону даних.

*Таблиця 1.* Діапазони даних при двійковому та РЛ кодуванні

| Розрядність двійкового коду | Діапазон двійкових чисел | Розрядність РЛ коду | Діапазон чисел в РЛ коду |
|---|---|---|---|
| 4 | $2^{-3} \le A \le 2^{+3}$ | 16 | $2^{-7} \le A \le 2^{+7}$ |
| 8 | $2^{-7} \le A \le 2^{+7}$ | 256 | $2^{-127} \le A \le 2^{+127}$ |
| 16 | $2^{-15} \le A \le 2^{+15}$ | 65536 | $2^{-32767} \le A \le 2^{+32767}$ |
| 32 | $2^{-31} \le A \le 2^{+31}$ | 4294967296 | $2^{-2147483647} \le A \le 2^{+2147483647}$ |

Реалізація адитивно-мультиплікативних операторів базується на законах виконання розрядних операцій при РЛ кодуванні. Реалізація операторів в РЛ кодуванні зводиться до виконання адитивних операцій над кодами розрядів чисел, що є однією із переваг цієї системи, оскільки обробка відбувається лише над значущими розрядами. Правила виконання розрядних операцій при додаванні РЛ має наступний вигляд:

$$N_i + N_k = N_i, N_k, \text{ якщо } N_i < N_k,$$

$$N_i + N_k = N_i + 1, \text{ якщо } N_i = N_k, \qquad (1)$$

де $N_i$, $N_k$ – коди ненульових розрядів операндів.

Операція додавання розпочинається з перетворення кодів, що мають найменшу вагу. За класичним алгоритмом [1] операція додавання складається з наступних етапів:

1) запис операндів у РЛ коді;

2) перевірка операндів на рівність нулю. Якщо один із них дорівнює нулю, то в РЛ структуру результату записується значення другого операнда і додавання закінчується (перехід до етапу 6). Якщо обидва операнди дорівнюють нулю, то РЛ код результату дорівнює нулеві (перехід до етапу 6);

3) об'єднання РЛ структур операндів в РЛ структуру результату;

4) сортування РЛ структури результату;

5) зведення подібних розрядів РЛ структури результату, використовуючи правила (1);

6) закінчення операції.

Об'єднання масивів розрядів двох доданків, сортування та зведення займає більшу частину часу обрахунку кінцевого результату. ***Операцію сортування можливо виключити*** взагалі, враховуючи властивості вхідних операндів, а саме їх впорядкованість.

Наведемо прискорений алгоритм додавання розпочинається так само як і за класичним алгоритмом з молодших розрядів і складається з наступних етапів:

1) запис операндів У РЛ формі;

2) перевірка операндів на рівність нулю. Якщо один із них дорівнює нулю, то в РЛ структуру результату записується значення другого операнда і додавання закінчується (перехід до етапу 4). Якщо обидва операнди дорівнюють нулю, то РЛ код результату залишається без змін (перехід до етапу 4);

3) обробка чергових розрядів РЛ структур доданків:

3.1) вибірка наступних розрядів кожного з доданків – $N_a$, $N_b$ та формування переносу $P$.

3.2) порівняння вибраних розрядів. Якщо $N_a > N_b$, то $P=N_b$ та при вибірці (етап 3.1) чергових розрядів вибірка буде відбуватися лише із другого операнда. Якщо $N_a < N_b$, то $P=N_a$ та при вибірці (етап 3.1) чергових розрядів вибірка буде відбуватися лише із першого операнда. Якщо $N_a=N_b$, то $P=N_b+1$ та при вибірці (етап 3.1) чергових розрядів вибірка буде відбуватися з обох операндів.

3.3) порівняння переносу $P$ з останнім елементом результату $N_c$. Якщо $P>N_c$, то $P$ заноситься в результат. Якщо $P=N_c$, то $N_c=N_c+1$.

3.4) перевірка на завершеність обробки обох операнів. Якщо обробка виконана для всіх розрядів обох доданків перехід до етапу 4, інакше повтор починаючи з етапу 3.1.

4) закінчення операції.

Для оцінки прискорення було проведене моделювання додавання випадкових чисел з різною кількістю розрядів. Результати моделювання представлено в табл. 2.

*Таблиця 2.* Результати моделювання додавання

| Кількість розрядів | Час обрахунку за класичним алгоритмом (сек.) | Час обрахунку за прискореним алгоритмом (сек.) | Ступінь прискорення (%) |
|---|---|---|---|
| 2048 | 0,02 | 0,01 | 50,00 |
| 4096 | 0,03 | 0,01 | 66,67 |
| 8192 | 0,03 | 0,02 | 33,33 |
| 16384 | 0,08 | 0,05 | 37,50 |
| 32768 | 0,17 | 0,11 | 35,29 |
| 65536 | 0,34 | 0,20 | 41,18 |
| 131072 | 0,72 | 0,39 | 45,83 |
| 262144 | 1,48 | 0,78 | 47,30 |
| 524288 | 3,03 | 1,56 | 48,51 |

Останній стовпчик таблиці 2 показує прискорення обчислення суми двох операндів в РЛ кодуванні в процентному відношенні. Незалежно від розрядності операндів середній виграш у часі складає 45%, що показує доцільність використання прискореного алгоритму.

***Множення*** двох операндів, представлених в РЛ коді, реалізується як поелементна сума кожного розряду першого множника з кожним розрядом другого. Правило виконання порозрядного множення має наступний вигляд :

$$N_i * N_k = N_i + N_k,$$

де $N_i$, $N_k$ – коди ненульових розрядів операндів.

Згідно правилам можливо реалізувати множення, як обчислення елементів матриці часткових добутків (2).

| | *A*0 | *A*1 | … | *An* |
|---|---|---|---|---|
| *B*0 | *A*0+*B*0 | *A*1+*B*0 | … | *An*+*B*0 |
| *B*1 | *A*0+*B*1 | *A*1+*B*1 | … | *An*+*B*1 |
| … | ... | ... | … | ... |
| *Bm* | *A*0+*Bm* | *A*1+*Bm* | … | *An*+*Bm* |

(2)

де $A_0 \ldots A_n$ – розряди першого множника;
$B_0 \ldots B_n$ – розряди другого множника.

Загальний алгоритм множення для розрядно-логарифмічних даних реалізується, як виконання наступних етапів:

1) запис операндів;

2) перевірка операндів на рівність нулю. Якщо один із них дорівнює нулю, то в РЛ структуру результату залишається без змін (перехід до етапу 6);

3) обчислення елементів матриці часткових добутків та занесення їх в РЛ структуру результату;

4) сортування РЛ структури результату;

5) зведення подібних розрядів РЛ структури результату, використовуючи правила (1) ;

6) кінець.

Розрядно-логарифмічна структура результату складається з елементів матриці часткових добутків (2). Розмірність такого масиву значна та таким чином головні часові витрати складає обробка такого масиву. Впорядкованість масиву значною мірою впливає на такі часові витрати, тому порядок обчислення проміжних добутків розглянемо детальніше розрядів.

При класичному виконанні множення кожний із рядків матриці (2) є відсор-

тованим, але в цілому результат сформований в такому порядку є майже не впорядкованим.

Наприклад: $A$=0.2.3.7.13.19.
$B$=1.2.6.11.13.

| | 0 | 2 | 3 | 7 | 13 | 19 |
|---|---|---|---|---|---|---|
| 1 | 1 | 3 | 4 | 8 | 14 | 20 |
| 2 | 2 | 4 | 5 | 9 | 15 | 21 |
| 6 | 6 | 8 | 9 | 13 | 19 | 25 |
| 11 | 11 | 13 | 14 | 18 | 24 | 30 |
| 13 | 13 | 15 | 16 | 20 | 26 | 32 |

(3)

Якщо записувати масив часткових добутків по строках, то маємо наступний результат

$C=A*B$=1.3.4.8.14.20.2.4.5.9.15.21.6.8.9.13.19.25.11.13.14.18.24.30.13.15.16.20.26.32.

Отримання результату обчислень $C$ потребує окрім сортування ще й операцію «зведення подібних». Для прискорення виконання множення пропонується наступне. Напрямок прискорення операції множення полягає у зміні порядку обрахунку часткових добутків. Мінімальний та максимальний елементи проміжного результату знаходяться на кінцях головної діагоналі, тобто елементи матриці поступово зростають в напрямку від верхнього лівого кута до нижнього правого. Таке розміщення назвемо ***градієнтністю.*** Використовуючи його, змінимо порядок обрахунку елементів матриці (2) з рядків на діагоналі, паралельні побічній (рис. 1).

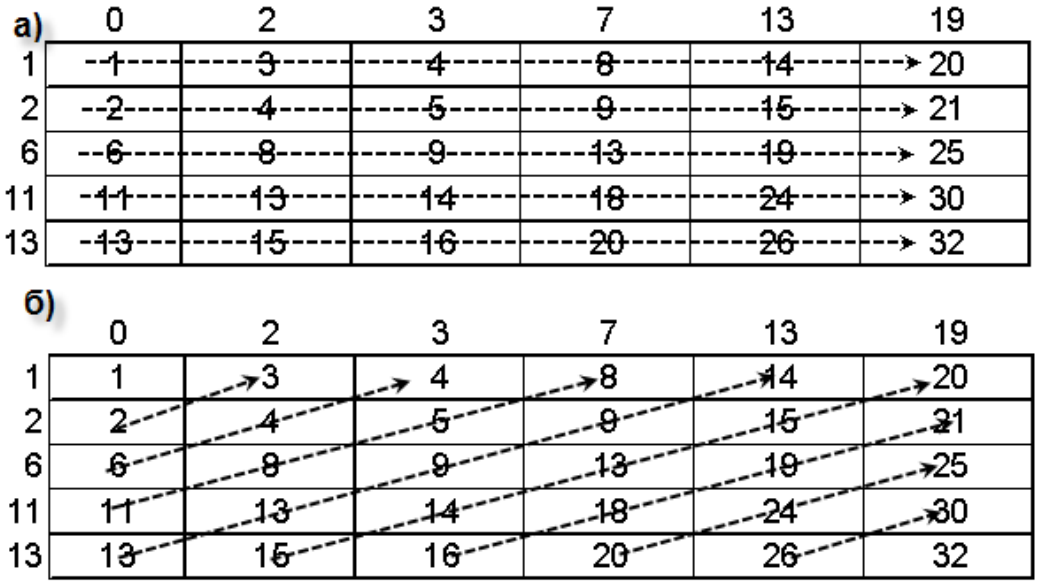


Рис. 1. Порядок обчислення елементів матриці (3).
а) за класичним методом; б) за прискореним методом

Використаємо даний метод для обрахунку проміжного масиву матриці (3):

$C$□=$A*B$=1.2.3.6.4.4.11.8.5.8.13.13.9.9.14.15.14.13.15.20.16.18.19.21.20.24.25.26.30.32

Даний масив є більш упорядкованим, що дає можливість уже на етапі обрахунку елементів використовувати поелементне «зведення елементів» та сортування, тому проміжний масив матиме вигляд:

$C$□□□□□=$A*B$=1.2.3.5.6.8.5.8.11.10.14.14.13.16.16.18.19.20.20.21.24.25.26.30.32

Дана методика дає можливість отримати проміжний масив результату

меншого розміру. При умові, що операнди мають різну розрядність, розбиття матриці можливо зробити на три матриці спеціального типу (рис. 2), що також дає прискорення при обчисленні.

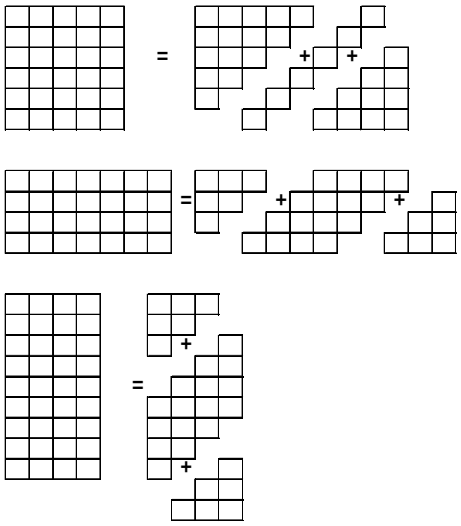

Рис. 2. Розбиття матриці часткових добутків

Для оцінки прискорення було проведено моделювання множення чисел з різною кількістю розрядів. Результати моделювання представлено в табл. 3.

*Таблиця 3.* Результати моделювання множення

| Кількість розрядів | Час обрахунку за класичним методом (сек.) | Час обрахунку за прискореним методом (сек.) | Ступінь прискорення (%) |
|---|---|---|---|
| 128 | 0,05 | 0,03 | 40,00 |
| 256 | 0,17 | 0,16 | 5,88 |
| 512 | 0,64 | 0,61 | 4,69 |
| 1024 | 2,64 | 2,48 | 6,06 |
| 2048 | 10,89 | 10,25 | 5,88 |
| 4096 | 45,23 | 41,98 | 7,19 |
| 8192 | 227,08 | 174,56 | 23,57 |

Останній стовбець таблиці 3 показує прискорення обчислення добутку двох операндів в РЛ кодуванні в процентному відношенні. Виграш у часі залежить від розрядності операндів – чим більша розрядність, тим більший виграш.

Більшого прискорення для обох операцій можливо досягнути використовуючи паралельні обчислення (на паралельних архітектурах та асоціативних процесорах).

### Висновки

Таким чином, в результаті досліджень запропоновано альтернативні алгоритми обчислення адитивно-мультиплікативних операторів при розрядно-логарифмічному кодуванні, та експериментально підтверджено ефективність їх використання в порівнянні з класичними алгоритмами. З алгоритму обчислення суми повністю виключено сортування, що прискорило обрахунки майже в два рази. Прискорення обчислення добутку досягнуто зміною порядку обчислень часткових добутків.

### Список літератури

1. *Гамаюн В.П.* Разрядно-логарифмическая арифметика, Методы и

алгоритмы. – К.: Книжное издательство НАУ, 2007. – 272 с.