УДК 621.373.54

С.Р. Бікташева, Л.В. Мороз, М.Ю. Стахів
Національний університет "Львівська політехніка",
кафедра захисту інформації

CORDIC-МЕТОД ОБЧИСЛЕННЯ КВАДРАТНОГО КОРЕНЯ

Проведено дослідження CORDIC-методу обчислення квадратного кореня.
Ключові слова – CORDIC, квадратний корінь.

CORDIC-method for square root calculating has been researched in this paper.
Keywords – CORDIC, square root.

Вступ

Операція добування кореня квадратного є дуже поширеною в цифровій обробці сигналів. Одним з можливих методів виконання цієї операції є ітераційний метод CORDIC – COordinate Rotation DIgital Computer, теоретичні основи якого описані у [1–4, 7, 8]. Однак цей метод майже не досліджений для практичного використання саме цієї операції, за винятком роботи [5], де наводяться лише деякі результати, що вказують на його високу ефективність.

Цілі роботи

Мета роботи – повноцінно дослідити CORDIC-метод обчислення квадратного кореня з метою вироблення рекомендацій для практичного використання.

Огляд досліджень

Існує кілька підходів до обчислення функції кореня квадратного за допомогою методу CORDIC. Вони детально описані в [3, 4, 7, 8]. Ми скористаємось результатами робіт [2, 5] і опишемо, на нашу думку, найпростіший варіант реалізації цього методу.

Основні результати досліджень

Рівняння (1)–(4) описують традиційний гіперболічний ітераційний CORDIC-метод для обчислення квадратного кореня $\sqrt{W}$, що працює у векторному режимі (hyperbolic vectoring mode):

$$\begin{aligned} x_{i+1} &= x_i - \sigma_i y_i 2^{-S(i)}, \\ y_{i+1} &= y_i - \sigma_i x_i 2^{-S(i)}, \end{aligned} \tag{1}$$

де

$$s_i = \begin{cases} -1 \ if \ y_i < 0 \\ +1 \ if \ y_i \geq 0 \end{cases} \qquad S(i) = 1,2,3,4,4,5,\ldots,12,13,13,14,\ldots,m, \qquad i = \overline{1,n}; \tag{2}$$

m – визначає число правильних двійкових розрядів дробової частини результату (або похибку обчислення); n – кількість ітерацій.

Зауважимо, що

$$x_0 = W + 0.25 \ , y_0 = W - 0.25 \ , W \in [0.03, \ 2.33]. \tag{3}$$

Як випливає з (2), деякі ітерації за номерами, наприклад 4, 13, повторюються з метою забезпечення умов збіжності [2].

Для компенсації деформації модуля вектора використовують післяітераційне ділення величини x_{n+1} на коефіцієнт деформації вектора K_n:

$$\sqrt{W} \approx x_{n+1} / K_n = x_{n+1} P_n \ , \tag{4}$$

де

$$K_n = \prod_{i=1}^{n} \sqrt{1 - 2^{-2S(i)}} \ ,$$

або в розгорнутому вигляді

$$\mathrm{K_n} = \sqrt{1-2^{-2}}\sqrt{1-2^{-4}}\sqrt{1-2^{-6}}\sqrt{1-2^{-8}}\sqrt{1-2^{-8}}\sqrt{1-2^{-10}}\sqrt{1-2^{-12}}\ldots$$

$$\ldots\sqrt{1-2^{-24}}\sqrt{1-2^{-26}}\sqrt{1-2^{-26}}\ldots\sqrt{1-2^{-2\mathrm{m}}}$$

$$P_n = 1/K_n .$$

Таблиця 1

Залежність максимальної абсолютної похибки обчислення кореня квадратного від числа ітерацій

Кількість ітерацій	Δ_n
6	$9.939\cdot10^{-4}$
7	$2.413\cdot10^{-4}$
8	$4.594\cdot10^{-5}$
9	$1.158\cdot10^{-5}$
10	$2.760\cdot10^{-6}$
11	$7.385\cdot10^{-7}$
12	$2.099\cdot10^{-7}$
13	$4.766\cdot10^{-8}$
14	$2.477\cdot10^{-8}$
15	$1.133\cdot10^{-8}$
16	$2.832\cdot10^{-9}$
17	$7.082\cdot10^{-10}$
18	$1.778\cdot10^{-10}$
19	$4.415\cdot10^{-11}$
20	$1.107\cdot10^{-11}$

Як бачимо, метод надзвичайно простий, містить на одну ітерацію лише дві операції зсуву на i розрядів вправо та дві операції віднімання. Після завершення ітерацій виконується операція множення на сталий (для цього n) коригуючий коефіцієнт P_n, яка і закінчує процес обчислень.

Ми дослідили залежність максимальної абсолютної похибки обчислення кореня квадратного від числа ітерацій:

$$\Delta_n = \max\left|x_{n+1}P_n - \sqrt{W}\right|. \qquad (5)$$

Основні результати цих досліджень зведені в табл. 1.

Звідси бачимо, що метод має приблизно лінійну збіжність – трохи менше двох правильних бітів результату на одну ітерацію і добре узгоджується з результатами, отриманими в [5].

Як випливає з аналізу цієї таблиці та характеру поведінки абсолютної похибки в усьому діапазоні зміни аргументу W, можна дещо покращати результати обчислень, якщо відкоригувати значення P_n для післяітераційного множення величини x_{n+1} на P_n. Результати зведені у табл. 2.

Як бачимо, відбулось зменшення значення максимальної абсолютної похибки приблизно у два рази. Слід зауважити, що за апаратної реалізації описаного методу необхідний помножувач для множення на сталий коефіцієнт P_n, одним із варіантів реалізації якого може бути високоточний помножувач для таких цілей, описаний у [6].

Таблиця 2

Залежність максимальної абсолютної похибки обчислення кореня квадратного від числа ітерацій за відкоригованого значення P_n

Кількість ітерацій	Δ_n	P_n
6	$3.690 \cdot 10^{-4}$	1.2070
7	$9.717 \cdot 10^{-5}$	1.20737
8	$2.360 \cdot 10^{-5}$	1.2074660
9	$6.293 \cdot 10^{-6}$	1.2074890
10	$1.485 \cdot 10^{-6}$	1.20749515
11	$4.212 \cdot 10^{-7}$	1.207496583
12	$1.309 \cdot 10^{-7}$	1.207496940
13	$4.766 \cdot 10^{-8}$	1.207497031
14	$1.945 \cdot 10^{-8}$	1.207497051
15	$5.952 \cdot 10^{-9}$	1.2074970605
16	$1.428 \cdot 10^{-9}$	1.2074970659
17	$3.553 \cdot 10^{-10}$	1.207497067296
18	$9.397 \cdot 10^{-11}$	1.207497067650
19	$2.244 \cdot 10^{-11}$	1.207497067734
20	$5.842 \cdot 10^{-12}$	1.207497067756

Якщо ж апаратна реалізація не передбачає наявності в складі пристрою окремого помножувача (ПЛІС, мікроконтролери без виконання операції множення), то цей метод може бути модифікований. Для цього значення коригуючого коефіцієнта P_n підбирається в такий спосіб, щоб можна було замінити операцію післяітераційного множення на кілька простих операцій зсуву, додавання/віднімання.

Наприклад, для $n=14$ або $n=25$ $P_n = 1 + 2^{-2} - 2^{-8} = 319/256.$

$n=14$ $W \in [0.015,\ 4.14]$ $D_n = 1.206 \cdot 10^{-5}$

i=	1	2	3	4	5	6	7	8	9	10	11	12	13	14
S(i)=	1	2	2	3	4	5	5	5	6	6	7	7	7	8

$n=25$ $W \in [0.015,\ 4.215]$ $D_n = 4.215 \cdot 10^{-8}$

i=	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S(i)=	1	2	2	3	4	5	5	5	6	6	7	7	8	8	8

i=	16	17	18	19	20	21	22	23	24	25
S(i)=	9	9	9	9	10	10	10	11	11	12

Очевидно, що в цьому випадку збіжність методу падає і становить приблизно один правильний біт результату на одну ітерацію.

Висновки

CORDIC-метод обчислення квадратного кореня має доволі високі швидкісні та точнісні характеристики і може успішно використовуватись у цифровій обробці сигналів поряд з іншими методами.

1. Volder J.E. The CORDIC trigonometric computing technique. IRE Trans. Electron. Comput.,EC-8, 3 (Sept. 1959), 330–334. 2. Walther J.S. A unified algorithm for elementary functions. In Proceedings of AFIPS 1971 Spring Joint Computer Conference, vol. 38, AFIPS Press, Arlington, Va., 1971, pp. 379-385. 3. Оранский А.М. Аппаратные методы в цифровой вычислительной технике. – Минск: Издательство БГУ, 1977. – 208 с. 4. Байков В.Д., Смолов В.Б. Специализированные процессоры: итерационные алгоритмы и структуры. – М.: Радио и связь, 1985. – 288 с. 5. Crawford J.A. Computing Square Roots U11891. 1 June 2005. http://www.am1.us/. 6. Gustafsson O. and Qureshi F. Addition Aware Quantization for Low Complexity and High Precision Constant Multiplication, 2010, IEEE Signal Processing. – Vol.17. – № 2. – P.173–176. 7. Аристов В.В. Функциональные макрооперации. Основы итерационных алгоритмов. – К.: Наук. думка, 1992. – 280 с. 8. Евдокимов В.Ф., Стасюк А.И. Параллельные вычислительные структуры на основе разрядных методов вычислений. – К.: Наук. думка, 1987. – 312 с.