УДК 681.3.001
© *Л. Мороз*[1]*, 2010*

АДАПТИВНИЙ CORDIC-МЕТОД ОБЧИСЛЕННЯ ДЕЯКИХ ФУНКЦІЙ

Розглянута реалізація операцій множення, ділення, ln(W) та макрооперацій -arctan(Y/X), arctanh(Y/X), помножувально-подільної, з використанням адаптивного-асинхронного режиму роботи CORDIC-method. Використання асинхронного режиму забезпечує збіжність ітераційного процесу і зменшує кількість ітерацій приблизно в 2 рази.

Considered realization of operations of multiplication, division, ln(W) and macrooperations of -arctan(Y/X), arctanh(Y/X), multiplication -division with the use of the adaptive asynchronous mode of operations of CORDIC-method. The use of the asynchronous mode is provided by convergence of iteration process and promotes a fast-acting approximately in 2 times.

1. ВСТУП

Ітераційний метод CORDIC - COordinate Rotation DIgital Computer, є одним з найпопулярніших методів обчислення багатьох елементарних функцій. Незважаючи на те, що методу уже понад 50 років, він продовжує інтенсивно розвиватись [5,6,8] .

2. МЕТА РОБОТИ

Метою даної роботи є подальше узагальнення CORDIC-методу для обчислення елементарних функцій та розширення його можливостей .

3. ОГЛЯД ДОСЛІДЖЕНЬ

Теоретичні основи методу CORDIC описані в роботах [1-4,7]. Метод базується на чисельному розв’язуванні породжуючих систем диференціальних рівнянь методом Ейлера, що використовує формулу прямокутників [7]. Найповніше узагальнення методу проведено у роботах [1,7,11]. Таке узагальнення важливе при побудові реконфігурованих обчислювачів, що працюють за методом CORDIC, тому що дозволяє побудувати універсальну структуру, яка реалізує будь-яку функцію з числа відтворюваних. Однак слід сказати, що такий узагальнений опис є неповним, оскільки не розділяє режими обертання (Rotation) та векторний (Vectoring).

---

[1] Національний університет “Львівська політехніка”

Крім того, цей опис розповсюджується лише на синхронний режим роботи CORDIC (або на знакозмінні ітерації) [4], що пов'язано з необхідністю збереження сталості значення коефіцієнта деформації вектора [4,7]. Однак існує ряд функцій, для яких непотрібно дотримуватись цих вимог, тому що коефіцієнт деформації не входить до складу виразів, за якими обчислюються ці функції або його вплив усувається за рахунок виконання операції ділення, що присутня в цих операціях (макрооопераціях). Зокрема, це стосується обчислення функцій $\tan(X)$, $\arctan(Y/X)$, $\tanh(X)$, $\arctan h(Y/X)$ та інших. В таких випадках можна використати асинхронний режим роботи CORDIC (або знакосталі ітерації), який теж забезпечує збіжність ітераційного процесу і підвищує швидкодію приблизно у 2 рази.

4. ОСНОВНІ РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ

Пропонується для уніфікації методу CORDIC використати наступний варіант узагальнення:

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \\ z_{i+1} \end{bmatrix} = \begin{bmatrix} 1 & -(R-V)M\sigma_i\delta_{M,i} & 0 & 0 \\ (R-V)\sigma_i\delta_{M,i} & 1 & 0 & 0 \\ 0 & 0 & 1 & (V-R)\sigma_i\alpha_{M,i} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} \quad (1)$$

M - параметр, що вказує на систему координат, в якій працює CORDIC (M=1 -колова, M=0 -лінійна, M=−1 - гіперболічна);

σ_i - оператор вибору напрямку обертання (вибирається з умови $\sigma_i = sign(z_i)$ або $\sigma_i = sign(y_i)$ і приймає значення $\sigma_i = \{-1,+1\}$) ;

$\delta_{M,i} = 2^{-S(M,i)}$ - оператор зсуву вправо (число розрядів зсуву $S(M,i)$ визначається режимами роботи – наприклад, для $M=1$ $S(1,i) = 0,1,2,3,4,5,...,12,13,13,14,...,m$, а для $M=-1$ $S(-1,i)$=1,2,3,4,4,5,...,12,13,13,14,...,m) ;

$\alpha_{M,i}$ - прирости змінної z_i, що зберігаються в постійній пам'яті (як правило, це константи типу $\arctan(2^{-i})$, $\arctan h(2^{-i})$ та 2^{-i}).

В режимі обертання (Rotation) R=1, V=0 та у векторному(Vectoring) –навпаки R=0, V=1.

Асинхронний режим $\sigma_i = \{0,+1\}$ дозволяє дуже просто реалізувати адаптивний режим роботи. В даному випадку адаптивність полягає у виборі необхідної кількості ітерацій, а також повторень деяких ітерацій (особливо для гіперболічної системи координат) з метою забезпечення діапазону перетворень та наперед заданої похибки обчислень $Rref = 2^{(-m)}$, де m - число дробових розрядів в режимі обчислень з фіксованою комою.

Розглянемо декілька операцій (множення, ділення, ln(W)) та макрооперацій (arctan(Y/X), arctanh(Y/X), помножувально-подільна), які можна виконати з використанням адаптивного асинхронного режиму роботи. Псевдокоди подані нижче.

```
ft= arctan(Y/X)
m:=16;Rref:=2^(-m); i:=0;
X:=1;Y:=8; Z:=0; x:=X;  y:=Y; z:=Z;
if Y>X then x:=Y;y:=X;fi; dx:= x; dt:= 1;
ft:= arctan(Y/X);
while dt>Rref do
while y<dx do
dx:=dx/2;
i:=i+1;
od;
x:=x+y*2^(-i);
y:=y-dx;
dx:=x*2^(-i);
dt:= arctan(2^(-i));
z:=z+dt;
if y=0 then dt:=0; fi;
od;
if Y>X then z:=evalf(Pi/2-z); fi;
delta:=z-ft;

ft= arctanh(Y/X)
m:=16; Rref:=2^(-m); i:=0;
X:=1;Y:=0.999999; Z:=0;  x:=X;y:=Y;z:=Z;
dx:= x; dt:= 1;
ft:= arctanh(Y/X);
while dt>Rref do
while y<dx do
dx:=dx/2;
i:=i+1;
od;
x:= x-y*2^(-i);
y:= y-dx;
```

```
dx:=x*2^(-i);
dt:= arctanh(2^(-i));
z:=z+dt;
if y=0 then dt:=0; fi;
od;
delta:=z-ft;
```

```
ft= ln(W)
m:=16; i:=0;
W=10;Rref:=2^(-m);x:=W+1;y:=W-1;z:=0;
dt:= 1; dx:= x;
ft:= ln(W);
while dt>Rref do
while y<dx
do dx:=dx/2;
i:=i+1;
od;
x:= x-y*2^(-i);
y:= y-dx;
dx:=x*2^(-i);
dt:= arctanh(2^(-i));
z:=z+dt;
if y=0 then dt:=0;fi;
od;
delta:= z*2-ft;
```

```
ft= Z*X/Y
m:=16; i:=0; Rref:=2^(-m);
X:=101;Y:=4093;Z:=35;
ft:= Z*X/Y; x:=X;y:=Y;z:=Z-Y;
dz:= Y*2^(-i); dx:=X*2^(-i);sigma:=sign(z);
if z=0 then dz:=0;dx:=0;fi;
while dx> Rref do
while abs(z)<dz do
dz:=dz/2;
i:=i+1;     od;
dx:= X*2^(-i);
x:= x+sigma*dx;
z:=z- sigma *dz;
if z=0 then dx:=0;fi;
   od;
delta:=x-ft;
```

```
ft=Y/X
m:=16; i:=0; Rref:=2^(-m);
i:=-1;
X:=1001;Y:=11; Z:=0; x:=X;y:=Y; z:=Z;
dx:=2^(-i)*X;dt:=1;
ft:=Y/X;
while dt >Rref do
while y<dx do
dx:=dx/2;
i:=i+1;
od;
dt:= 2^(-i);
y:= y-dx;
z:= z+dt;
if y=0 then dt:=0;fi;
od;
delta:=z-ft;

ft:= Z*X
m:=16; i:=-1; Rref:=2^(-m);
X:=0.55;Y:=0;Z:=0.33;x:=X;y:=Y;z:=Z;
ft:= Z*X;
dx:=2^(-i)*x;dt:=1;
 while dt>Rref do
while z<2^(-i) do
dx:=dx/2;
i:=i+1;
od;
dt:= 2^(-i);
y:= y+dx;
z:= z-dt;
if z=0 then dt:=0;fi;
od;
delta:=y-ft;
```

Тут прийняті наступні позначення: ft - аналітичний вираз, значення якого обчислюється; X,Y,Z,W- вхідні операнди; delta –абсолютна похибка.

Слід відмітити, що початкове значення i для $M=0$ може змінюватись в сторону зменшення ($i=-2,-3,-4...$) згідно з рекомендаціями [10]. При цьому розширюється діапазон перетворень та зменшується число необхідних ітерацій.

5. ВИСНОВКИ.

Адаптивний асинхронний CORDIC-метод має достатньо високі швидкість (загальне число ітерацій зменшено приблизно у два рази порівняно з традиційним синхронним CORDIC-методом) та точність (забезпечує задане значення похибки у всьому діапазоні перетворення) і може успішно використовуватись в цифровій обробці сигналів.

1.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. 2. Volder J.E. The CORDIC trigonometric computing technique, IRE Trans. Electron. Comput., Vol. EC-8, NO.3, Sept. 1959,pp. 330-334. 3.Оранский А.М. Аппаратные методы в цифровой вычислительной технике.- Минск: Издательство БГУ,1977.-208с. 4.Байков В. Д., Смолов В. Б. Специализированные процессоры: итерационные алгоритмы и структуры . –Москва: "Радио и связь", 1985.-288с. 5.Beuler M. CORDIC-Algorithmus zur Auswertung elementarer Funktionen in Hardware. FH-Report. Juni 2008. http://dok.bib.fh-giessen.de/opus/volltexte/2009/4148/ 6. Pramod K. Meher, Javier Valls,Tso-Bing Juang,K. Sridharan,Koushik Maharatna. 50 Years of CORDIC: Algorithms, Architectures,and Applications. IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS, VOL. 56, NO. 9, SEPTEMBER 2009, pp. 1893-1907. 7. Аристов В.В. Функциональные макрооперации. Основы итерационных алгоритмов. –Киев: "Наукова думка", 1992.-280с. 8. Lakshmi B. and Dhar A. S. CORDIC Architectures: A Survey. VLSI Design. Volume 2010, Article ID 794891, 19 pages, January 8, 2010. 9. Llamocca-Obregón D.R., Agurto-Ríos C.P. A Fixed-Point Implementation of the Natural Logarithm Based on a Expanded Hyperbolic CORDIC Algorithm. XIIWORKSHOP IBERCHIP, p.p. 322 -323, 2006. 10. X. Hu, R. Huber, S. Bass, "Expanding the Range of Convergence of the CORDIC Algorithm", IEEE Transactions on Computers. Vol. 40, N° 1, pp. 13-21,1991. 11.Beuler, M.; Bonath, W. Berechnung elementarer Funktionen für das Huber/Braun-Neuronenmodell in digitaler Architektur; MPCWorkshop Konstanz, Juli 2008. http://dok.bib.fh-giessen.de/opus/volltexte/2009/4149/