TECHNICAL TRANSACTIONS 4/2018
ELECTRICAL ENGINEERING
DOI: 10.4467/2353737XCT.18.059.8371
SUBMISSION OF THE FINAL VERSION: 28/3/2018

Leonid Moroz
Department of Security of Information Technology, Lviv Polytechnic National University, Ukraine

Volodymyr Samotyy (vsamotyy@pk.edu.pl)
Department of Automatic Control and Information Technology, Cracow University of Technology

The CORDIC method of calculating the exponential function

Metoda CORDIC obliczania funkcji eksponencjalnej

Abstract

This article presents a modern method of calculating the exponential function exp(x) based on the CORDIC iterative algorithm. The proposed solution is implemented in the form of a single iterative equation, which results in the simplification of the electronic version of this algorithm, thus reducing the cost of the device. It is important to point out that the accuracy of the calculation of the analysed function is not lost.

Keywords: exponential function, hybrid CORDIC method, LUT lookup table, iterative equations, activation functions

Streszczenie

W artykule zaproponowano nowoczesną metodę obliczania funkcji eksponencjalnej exp(x) na bazie algorytmu iteracyjnego CORDIC. Przedstawione rozwiązanie jest realizowane w postaci jednego równania iteracyjnego, co prowadzi do uproszczenia wersji elektronicznej tego algorytmu i w wyniku zmniejsza koszt urządzenia. Należy podkreślić, że przy tym nie tracimy na dokładności obliczania badanej funkcji.

Słowa kluczowe: funkcja eksponencjalna, hybrydowa metoda CORDIC, tablica odnośników LUT, równania iteracyjne, funkcje aktywacji

1. Introduction

CORDIC (coordinate rotation in a digital compute) algorithms were developed a long time ago; however, to this day they remain a subject of interest to many researchers. The method has been developed by Jack E. Volder [9]. The method can be easily used for fast Fourier transform calculations, Householder transformation calculations, digital signal filtering, image recognition, etc. The idea of this algorithm involves the application of the iterative process of rotating vectors on a plane at any angle – only shifting and adding operations are used. A number of patents has been published in which CORDIC algorithms are implemented in the form of electronic devices; however, only one of them features an electronic implementation of the exponential function [10]. This approach involves a large amount of computation and requires the electronic implementation of two iterative equations, which is a time-consuming process. There is a well-known paper on simplified exponential function calculation in which only two iterations are used [7, pp. 1–4]. This simplifies the electronic implementation of the CORDIC algorithm, although in the proposed approach, the number of calculations remains unchanged.

2. Calculating the exponential function using the hybrid CORDIC method

The device designed for calculating functions is

$$x = \exp(\pm\varphi)\,, \tag{1}$$

Measurements of the input signal φ are converted into binary code (m– number of bits)

$$\varphi = \sum_{i=1}^{m} a_i 2^{-i}, \;\; a_i = \{0,1\}, \;\; \varphi \in [0,1) \tag{2}$$

The following well-known exponential function calculators [1-6, 8] use CORDIC iterative methods (Fig. 1):

$$x_{i+1} = x_i + \sigma_i y_i 2^{-i}, \tag{3}$$

$$y_{i+1} = y_i + \sigma_i x_i 2^{-i}, \tag{4}$$

$$z_{i+1} = z_i - \sigma_i \alpha_i, \tag{5}$$

$$\alpha_i = \operatorname{arctan}h\left(2^{-i}\right), \;\; i = 1,\, 2,\, 3,\, 4,\, 4,\, 5 \ldots 12, 13, 13, 14, \ldots, m\,, \tag{6}$$

$$\sigma_i = \begin{cases} -1\, if\;\, z_i < 0 \\ +1\, if\;\, z_i \geq 0 \end{cases}, \tag{7}$$

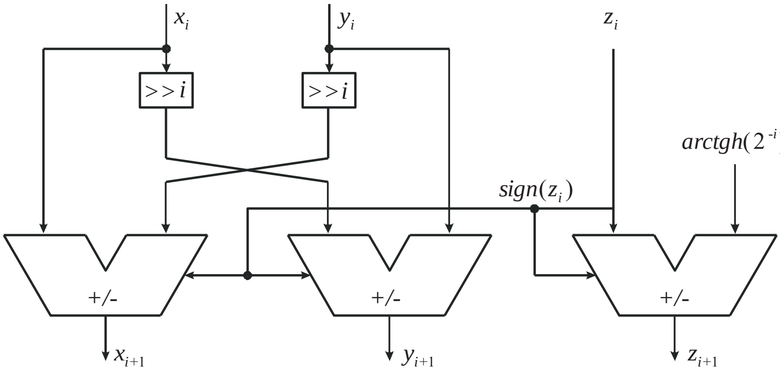

Fig. 1. Electronic implementation of the CORDIC method

$$x_1 = P^{'},\ y_1 = 0\ ,\ z_1 = \varphi,\ x_{m+1} \approx \cosh(\varphi),\ y_{m+1} \approx \sinh(\varphi),\ \varphi \in [0,\ 1.118] \qquad (8)$$

$$P^{'}_{m1} = \prod_{i=1}^{m} \cosh(\alpha_i) = \prod_{i=1}^{m} \frac{1}{\sqrt{1-2^{-2i}}};\ P^{'}_{m2} = \frac{1}{\sqrt{1-2^{-8}}} \frac{1}{\sqrt{1-2^{-26}}} \frac{1}{\sqrt{1-2^{-80}}} \qquad (9)$$

$$P^{'} = P^{'}_{m1} P^{'}_{m2} \qquad (10)$$

After the completion of the iterative calculation of the function $y_{m+1} \approx \sinh(\varphi)$ and $x_{m+1} \approx \cosh(\varphi)$, we can calculate the exponent functions:

$$x_{m+1} + y_{m+1} \approx \exp(\varphi),\ x_{m+1} - y_{m+1} \approx \exp(-\varphi) \qquad (11)$$

The disadvantage of such devices is the redundancy of electronic components for the implementation of three iterative equations for variables x_{i+1}, y_{i+1} and z_{i+1}, and the resulting long calculation time (for a calculation with accuracy of m bits, we have to implement m+2 iterative cycles). The simplified device (Fig. 2) for the exponent calculation is described in [7]. The device implements the algorithm

$$w_{i+1} = w_i + \sigma_i w_i 2^{-i},\ z_{i+1} = z_i - \sigma_i \alpha_i,\ w_1 = P^{'},\ z_1 = \varphi,\ w_{m+1} \approx \exp(\varphi) \qquad (12)$$

and

$$w_{i+1} = w_i - \sigma_i w_i 2^{-i},\ z_{i+1} = z_i - \sigma_i \alpha_i,\ w_1 = P^{'}, \qquad (13)$$

$$z_1 = \varphi,\ w_{m+1} \approx \exp(-\varphi),\ \varphi \in [0,\ 1.118] \qquad (14)$$

Here, only two iterative equations are implemented for w_{i+1} and z_{i+1}, which simplifies the structure of the device. Unfortunately, this does not lead to a reduction in the number of calculations.

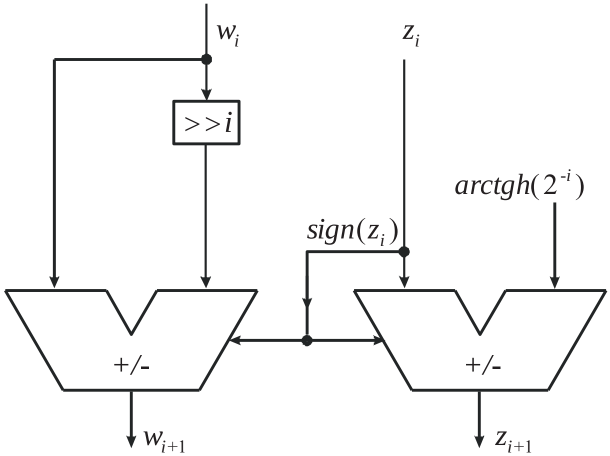

Fig. 2. Electronic implementation of the simplified CORDIC method

The end goal of our invention is to simplify the electronic implementation of the device and to reduce the number of calculation operations. First, let us consider the calculation of the function

$$x = \exp(+\varphi). \tag{15}$$

We propose dividing the input argument j into three distinct parts j_1, j_2, j_3 (Fig. 3)

$$\varphi = \varphi_1 + \varphi_2 + \varphi_3 \tag{16}$$

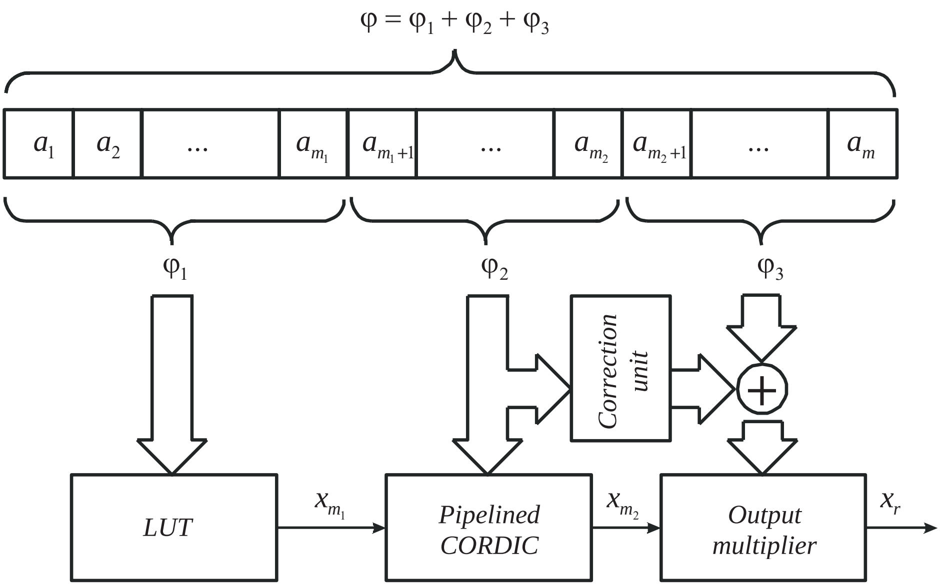

Fig. 3. Hybrid CORDIC

The first part j_1 occupies m_1 older bits of argument j. These are fed into the LUT lookup table (Fig. 3).

$$\varphi_1 = \sum_{i=1}^{m_1} a_i 2^{-i} \tag{17}$$

The second part (j_2) of the input argument j is computed using the CORDIC method, which is implemented in the form of a single iterative equation (Fig. 4). The second part (j_2) of the input argument j occupies the subsequent $m_2–m_1$ bits

$$\varphi_2 = \sum_{i=m_1+1}^{m_2} a_i 2^{-i} \tag{18}$$

At the end, the third calculation block provides for the multiplication of the correction angle j_3, which occupies $m–m_2$ bits

$$\varphi_3 = \sum_{i=m_2+1}^{m} a_i 2^{-i} \tag{19}$$

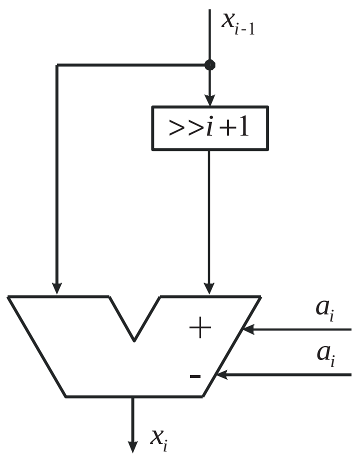

Fig. 4. Proposed pipelined CORDIC

Here, m_1 – the number of older bits of argument j, fed into the LUT table, which performs the function

$$x_{m_1} = P \cdot \exp(\varphi_1 + D_c) \tag{20}$$

where D_c and P – constants, which we calculate according to the formulas:

$$D_c = \sum_{i=m_1+2}^{m_2+1} \arctan h(2^{-i}) \tag{21}$$

$$P = \left(\prod_{i=m_1+1}^{m_2} \sqrt{1-2^{-2i-2}} \right)^{-1} \tag{22}$$

The values of x_{m_2} which we read from the LUT outputs, contain m bits. The minimum value of m_1 is obtained from the condition:

$$m_{1\min} = \left\lceil \frac{m - 10 - 2 \cdot \log_2 3}{6} \right\rceil . \tag{23}$$

This quantity of older bits $m_{1\min}$ gives us the ability to maintain the accuracy of calculations in m bits. The upper limit of m_1 is limited by the value of m_2 and depends solely on the LUT memory space.

The bits of the φ_1 part, that is bits with numbers $m_1+1 \ldots m_2$, are processed according to the CORDIC method. The value of m_2 is chosen based on the condition

$$m_2 = \left\lceil \frac{m}{2} \right\rceil \tag{24}$$

In practice, the CORDIC equations

$$b_i = 2 \cdot a_i - 1 \tag{25}$$

and

$$x_i = x_{i-1} + b_i \cdot x_{i-1} \cdot 2^{-i-1} \tag{26}$$

are implemented as follows:

$$\text{if } a_i = 1, \text{ then } b_i = 1 \text{ and } x_i = x_{i-1} + x_{i-1} \cdot 2^{-i-1}, \tag{27}$$

$$\text{if } a_i = 0, \text{ then } b_i = -1 \text{ and } x_i = x_{i-1} - x_{i-1} \cdot 2^{-i-1}. \tag{28}$$

It follows that CORDIC contains only one iterative equation (28). The CORDIC output gives us the code x_{m_2}. In the final stage, we use the values of z corrected for the D_v component of φ_3, which is calculated according to the formula

$$z = \varphi_3 + D_v , \tag{29}$$

where

$$D_v = \left(\sum_{i=m1+1}^{m_3} a_i [2^{-i} - 2 \cdot \operatorname{arctan} h(2^{-i-1})] \right] \tag{30}$$

$$m_3 = \left\lceil \frac{m - 5 - \log_2 3}{3} \right\rceil . \tag{31}$$

$$x_r = x_{m_2} + z \cdot x_{m_2} . \tag{32}$$

A patented device can be used to calculate activation functions in neural networks – this gives the possibility for a significant reduction in the amount of computation in the network teaching process. There are three types of activation functions in which the exponential function is used. These include functions

$$\text{logistic} \quad f_{active}(x) = \frac{1}{1 - e^{-x}}\,, \tag{33}$$

$$\text{hyperbolic} \quad f_{active}(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}\,, \tag{34}$$

$$\text{exponential} \quad f_{active}(x) = e^{-x}\,, \tag{35}$$

$$\text{softmax} \quad f_{active}(x) = \frac{e^{x}}{\sum_{i} e^{x_i}}\,. \tag{36}$$

3. Conclusions

The characteristic feature of the patented device is that the argument j of function $y=\exp(\pm j)$ is divided into three parts and only the second part j_2 is calculated using the CORDIC iteration algorithm. This solution reduces the number of computations by several times and as a result, increases the speed of the device for calculating the function $y=\exp(\pm j)$.

The CORDIC algorithm is implemented in the proposed device in the form of only one iterative equation. This leads to the simplification of the electronic version of this algorithm and thus reduces the cost of the device.

The time pause in the patented device will be several times shorter than the currently known solution [10]. For example, for $m=32$ and $m_1=4$, the time pause in the patented device has 14 steps, and the current solution requires 34 steps. For $m=32$ and $m_1=8$, we have 10 and 34 steps, respectively.

References

[1] Walther J.S., *A unified algorithm for elementary functions*, [in:] Proc. Spring Joint Comput. Conf., 1971, 379–385.

[2] Walther J.S., *Elementary floating-point Cordic function processor and shifter*, US Patent 3766370, 1973.

[3] Muller J.M., *Elementary functions: algorithms and implementation*. Birkhauser Boston, 2nd edition, 2006.

[4] Hu X., Huber R., Bass S., *Expanding the Range of Convergence of the CORDIC Algorithm*, IEEE Transactions on Computers, Vol. 40, 1, 1991, 13–21.

[5] Llamocca-Obregon D.R., Agurto-Rios C.P., *A Fixed-Point Implementation of the Expanded Hyperbolic CORDIC Algorithm*, Latin American Appl. res., Vol. 37, 1, 2007, 1–10.

[6] Sudha J., Hanumantharaju M.C., Venkateswarulua V., Jayalaxmi H., *A Novel Method for Computing Exponential Function Using CORDIC Algorithm*, Procedia Engineering, 30, 2012, 519–528.

[7] Pottathuparambil R., Sass R., *Implementation of a CORDIC based double-precision exponential core on an FPGA,* Proceedings of RSSI, 2008, 1–4.
[8] Boudabous A., Ghozzi F., Kharrat M., Masmoudi N., *Implementation of hyperbolic functions using cordic algorithm,* in Proc. of the 16th Inter. Conf. on Microelectronics, 2004, 738–741.
[9] Volder J.E., *The CORDIC Trigonometric Computing Technique,* IRE Transactions on Electronic Computers, 1959, 330–334.
[10] Kantabutra V., *Apparatus For Computing Exponential And Trigonometric Functions,* United States Patent, US006055553A, Apr. 25, 2000.

If you want to quote this article, its proper bibliographic entry is as follow: Moroz L., Samotyy V., *The CORDIC method of calculating the exponential function*, Technical Transactions, Vol. 4/2018, pp. 119–126.