The Double Rotation CORDIC Algorithm: New Results for VLSI Implementation of Fast Sine/Cosine Generation

Tze-Yun Sung* Chich-Sin Chen** Ming-Chou Shih*
*Department of Electrical Engineering
**Institute of Engineering Science
Chung Hua University, Hsinchu, Taiwan 300-12
{sung, sin, shih}@btcx.com.tw

Abstract- *The COordinate Rotation DIgital Computer (CORDIC) algorithm is an arithmetic algorithm to evaluate various elementary functions through a series of iterative operations. In this paper, a high-speed sine/cosine generator is based on double rotation of the original CORDIC algorithm by predicting all the rotation directions from the initial input angle. The proposed architecture has a simple prediction scheme through an efficient determination strategy of rotation direction. The critical delay path is reduced by utilizing the carry-save adder (CSA). Thus, the computation complexity of the proposed architecture is evaluated; the proposed architecture improves the latency of 37.5% in 16-bit operand, 40.6% in 24-bit operand and 42.5% in 32-bit operand, respectively. While in the large number-bit operand, the speed should be improved by 48%.*

Keywords: double rotation algorithm, CORDIC, σ -prediction algorithm, carry-save adder, sine/ cosine generator.

1. Introduction

In direct digital frequency synthesizer (DDFS) system [1] and orthogonal frequency division multiplexer (OFDM) system [2], [3], the key component is the sine/cosine function generator to computes $\sin\theta$ and $\cos\theta$ to a precision of N fraction bits. In this paper, a high-speed sine/cosine generator based on the CORDIC algorithm is proposed. CORDIC (COordinate Rotation DIgital Computer) is an algorithm for performing a sequence of iteration computations using coordinate rotation [4], [5]. It can generate some powerful elementary function realized only by a simple set of adders and shifters. The basic CORDIC iteration equations are

$$x_{i+1} = x_i - m\sigma_i 2^{-s(m,i)} y_i \tag{1}$$

$$y_{i+1} = y_i + \sigma_i 2^{-s(m,i)} x_i \tag{2}$$

$$z_{i+1} = z_i - \sigma_i \alpha_{m,i} \tag{3}$$

where m identifies circular (m=1), linear (m=0), and hyperbolic (m=-1) coordinate systems, i=0, 1,2,….,n-1,

$$s(m,i) = \begin{cases} 0,1,2,3,4,5,.... & m=1 \\ 1,2,3,4,5,6,.... & m=0 \\ 1,2,3,4,4,5,.... & m=-1 \end{cases}$$

in a hyperbolic coordinate system, the iterations are repeated at $3i+1$.

$$\alpha_{m,i} = m^{-1/2} \tan^{-1}[\sqrt{m}\, 2^{-s(m,i)}] \tag{4}$$

the rotation σ_i for rotation mode $(z_n \to 0)$ is $\sigma_i = sign(z_i)$, while for vectoring mode $(y_n \to 0)$, it is $\sigma_i = -sign(x_i)\cdot sign(y_i)$.

For the i-th iteration, a scale factor becomes $k_{m,i} = \sqrt{1 + m\sigma_i^2 2^{-2s(m,i)}}$. After n-iterations, the product of all the scale factors is

$$K_m = \prod_{i=0}^{n} k_{m,i} = \prod_{i=0}^{n} \sqrt{1+m\sigma_i^2 2^{-2s(m,i)}} = \prod_{i=0}^{n} \sqrt{1+m2^{-2s(m,i)}} \tag{5}$$

where the rotation directions are defined to $\sigma_i = \{-1,+1\}$.

2. Double Rotation CORDIC Algorithm

The basic concept of the accelerated CORDIC algorithm is to reduce the iterations. The double rotation CORDIC algorithm is developed to reduce the iterations or computation time [6]. The double rotation CORDIC iteration equations should be derived and the computation complexity should be also evaluated.

The CORDIC iteration equations in circular coordinate system are also written in the form of matrix multiplications.

$$\begin{bmatrix} x_{2i+1} \\ y_{2i+1} \end{bmatrix} = \begin{bmatrix} 1 & -\sigma_{2i} 2^{-2i} \\ \sigma_{2i} 2^{-2i} & 1 \end{bmatrix} \begin{bmatrix} x_{2i} \\ y_{2i} \end{bmatrix} \tag{6}$$

According to eqs.(6), we obtain

$$\begin{bmatrix} x_{2i+2} \\ y_{2i+2} \end{bmatrix} = \begin{bmatrix} 1 & -\sigma_{2i+1} 2^{-(2i+1)} \\ \sigma_{2i+1} 2^{-(2i+1)} & 1 \end{bmatrix} \begin{bmatrix} x_{2i+1} \\ y_{2i+!} \end{bmatrix} \tag{7}$$

the eq. (7) is an iteration equation of the double rotation CORDIC algorithm.

Thus, the double rotation CORDIC iteration equation in circular coordinate system is modified as shown below

$$x_{2i+2} = (1-\sigma_{2i}\sigma_{2i+1} 2^{-(4i+1)}) x_{2i} - (\sigma_{2i} 2^{-2i} + \sigma_{2i+1} 2^{-(2i+1)}) y_{2i} \tag{8}$$

$$y_{2i+2} = (\sigma_{2i} 2^{-2i} + \sigma_{2i+1} 2^{-(2i+1)}) x_{2i} + (1 - \sigma_{2i}\sigma_{2i+1} 2^{-(4i+1)}) y_{2i} \quad (9)$$

$$z_{2i+2} = z_{2i} - \sigma_{2i} \tan^{-1} 2^{-2i} - \sigma_{2i+1} \tan^{-1} 2^{-(2i+1)} \quad (10)$$

The computation complexity of parallel processing is increased to two carry-save additions ((3,2)CSAs) and one shift for each iteration [7]. In n-bit operand system, while $i \geq \frac{n}{4} - 1$, eqs.(8) and (9) becomes

$$x_{2i+2} = x_{2i} - (\sigma_{2i} 2^{-2i} + \sigma_{2i+1} 2^{-(2i+1)}) y_{2i} \quad (11)$$

$$y_{2i+2} = (\sigma_{2i} 2^{-2i} + \sigma_{2i+1} 2^{-(2i+1)}) x_{2i} + y_{2i} \quad (12)$$

Thus, the computation complexity of parallel processing is one (3,2)CSA and one shift for each iteration.

3. A Novel σ-Prediction Algorithm

The basic intention to realize the double rotation CORDIC algorithm is to generate more σ values in each step. Now, the proposed architecture requires two σ values in each step. The σ-value prediction algorithm is described as below:

In this algorithm, the σ_{2i} and σ_{2i+1} are generated in following steps, the σ_{2i} is determined by sign of $z(2i)$. The series of new constants can be defined as

$$z_1(2i+2) = z(2i) - \sigma_{2i}(\tan^{-1} 2^{-2i} + \tan^{-1} 2^{-(2i+1)}) \quad (13)$$

$$z_2(2i+2) = z(2i) - \sigma_{2i}(\tan^{-1} 2^{-2i} - \tan^{-1} 2^{-(2i+1)}) \quad (14)$$

$$z_3(2i+2) = z(2i) - \sigma_{2i} \tan^{-1} 2^{-2i} \quad (15)$$

Three equations for determining σ_{i+1} and $z(i+2)$ are defined as

$$z_1(2i+2) = z(2i) - \sigma_{2i}\Delta_1(2i) \quad (16)$$

$$z_2(2i+2) = z(2i) - \sigma_{2i}\Delta_2(2i) \quad (17)$$

$$z_3(2i+2) = z(2i) - \sigma_{2i}\Delta_3(2i) \quad (18)$$

The determination strategy of $\{\sigma_{2i}, \sigma_{2i+1}\}$ and $z(2i+2)$ is illustrated in Figs. 1, 2 and 3.

The flowchart for the σ_{2i+1}-prediction and $z(2i+2)$ determination algorithm is illustrated in Fig. 1, detailed flowcharts for specific cases are illustrated in Fig. 2 and 3, respectively. Now, the σ-prediction and $z(2i+2)$ determination algorithm is analyzed and developed, this algorithm is simple and easy to implement on hardware. Thus, the algorithm is very suited to VLSI implementation. The determination circuit of σ_{2i+1} and $z(2i+2)$ is shown in Fig. 4.

4. The Accelerated CORDIC Architecture for Sine/Cosine Generator

The proposed architecture has n-bit word length, so it makes n-iteration to compute the circular coordinate system. In this architecture, the (4,2) carry-save adder (CSA) and carry-propagation adder (CPA) consists of two three-input, two-output (3,2) carry-save adders/subtractors and one carry-look-ahead adder [8]. Fig. 5 shows the proposed accelerated architecture with the rotation mode in a circular coordinate system.

Thus, the computation complexity is two CPA computations, a CLA computation and a shift for each iteration at first $\frac{n}{4}$ iterations, and the computation complexity is a CPA computation, a CLA computation and a shift for each iteration at last $\frac{n}{4}$ iterations.

The Sine/Cosine generator is implemented by the accelerated CORDIC architecture with the rotation mode in the circular coordinate system. The input of x_0 is $\frac{1}{K_1}$ and input of y_0 is 0, the input of z_0 is an angle for sine and cosine function. In this architecture, the shift sequence $\{s(1,i)\}$ is pre-defined, so that the K_1 is a constant.

5. Performance Analyses of the Accelerated CORDIC Architecture

Since the computation complexity of accelerated CORDIC architecture is two additions and one shift for each iteration at first $\frac{n}{4}$-iteration. At last $\frac{n}{4}$-iteration, the computation complexity of the architecture is an addition and shift for each iteration. The total computation complexity of the accelerated CORDIC architecture is

$$(2 \times T_{CSA} + T_{CLA} + T_{shift}) \times \frac{n}{4} + (T_{CSA} + T_{CLA} + T_{shift}) \times \frac{n}{4}$$

$$= (\frac{n}{2} + 8) \cdot T_G \times \frac{n}{4} + (\frac{n}{2} + 6) \cdot T_G \times \frac{n}{4} \quad (24)$$

$$= (\frac{n^2 + 14n}{4}) \cdot T_G \quad (25)$$

where T_{CSA} (operation time of CSA)=T_{FA} (delay of full-adder)=$2T_G$ (delay of single gate), T_{shift} (operation time of hardwired shift)= T_G and T_{CLA} (operation time of CLA)= $(\frac{n}{2} + 3) \cdot T_G$ [7]. $(\frac{n}{2} + 8) \cdot T_G$ and $(\frac{n}{2} + 6) \cdot T_G$ are computation time for first $\frac{n}{4}$-iteration and last $\frac{n}{4}$-iteration, respectively.

The total computation complexity of conventional CORDIC is n CLA computations and n shifts. The computation complexity is represented as $(\frac{n^2+8n}{2})\cdot T_G$.

According to Fig. 3, the σ_{2i+1} and $z(2i+2)$ determination circuit consists of three subtractors, and one multiplexer. The determination time of σ_{2i+1} and $z(2i+2)$ is

$$T_{CLA}+T_{MUX}=(\frac{n}{2}+3)\cdot T_G+2\cdot T_G=(\frac{n}{2}+5)\cdot T_G \quad (26)$$

where $T_{MUX}\le 2\cdot T_G$.

Thus, the determination time of σ_{2i+1} and $z(2i+2)$ is less than computation time of $[\,x_{i+2}\ \ y_{i+2}\,]$, which is $(\frac{n}{2}+8)\cdot T_G$ or $(\frac{n}{2}+6)\cdot T_G$, and the process of the σ_{2i+1} and z(2*i*+2) determination can not reduce the throughput.

The percentage of latency improvement versus number of bit in each operand is shown in Fig. 6.

6. Numerical Analyses of the Double Rotation CORDIC Algorithm

The numerical analysis of the double rotation CORDIC algorithm is discussed in this section. Several error analysis researches of the CORDIC algorithm have been done [9], [10], [11]. The difference between the double rotation CORDIC algorithm and the conventional CORDIC algorithm is the term dropped in eqs. (8), (9), and (10). Now, the maximum error of the double rotation CORDIC algorithm related to the conventional CORDIC algorithm is analyzed and derived as ***Theorem 1*** [6], [9], [11]:

Theorem 1: The upper bound of the double rotation CORDIC algorithm is

$$\left\|\frac{\Delta \boldsymbol{u}}{\boldsymbol{v}}\right\|\le 2^{\frac{1}{2}}\cdot\frac{n}{2}\cdot 2^{-n-1}=2^{\frac{1}{2}}\cdot n\cdot 2^{-n-2}$$

where $\boldsymbol{v}=\begin{bmatrix}x_0\\ y_0\end{bmatrix}, \boldsymbol{u}=\begin{bmatrix}x_n\\ y_n\end{bmatrix}$, $\Delta\boldsymbol{u}$ is the error of $\boldsymbol{u}$ and *n* is the number of bits.

Proof:

$$\boldsymbol{u}=\begin{bmatrix}x_n\\ y_n\end{bmatrix}\ ,\boldsymbol{v}=\begin{bmatrix}x_0\\ y_0\end{bmatrix}, \boldsymbol{P}_i=\begin{bmatrix}1 & -\sigma_i 2^{-i}\\ \sigma_i 2^{-i} & 1\end{bmatrix}$$

where *i*= 0, 1, 2,…., *n*,

$$\boldsymbol{u}=\frac{1}{K_1}\prod_{i=0}^{n}\boldsymbol{P}_i\boldsymbol{v}$$

$$\boldsymbol{u}=\frac{1}{K_1}\prod_{i=0}^{\frac{n}{2}-1}\boldsymbol{P}_{2i}\boldsymbol{P}_{2i+1}\boldsymbol{v}$$

$$\boldsymbol{u}+\Delta\boldsymbol{u}=\frac{1}{\boldsymbol{K}_1}\prod_{i=0}^{\frac{n}{2}-1}(\boldsymbol{P}_{2i}\boldsymbol{P}_{2i+1}+\Delta\boldsymbol{P}_{2i,2i+1})(\boldsymbol{v}+\Delta\boldsymbol{v})$$

$$=\frac{1}{K_1}\prod_{i=0}^{\frac{n}{2}-1}\boldsymbol{P}_{2i}\boldsymbol{P}_{2i+1}\boldsymbol{v}+\frac{1}{K_1}\prod_{i=0}^{\frac{n}{2}-1}\boldsymbol{P}_{2i}\boldsymbol{P}_{2i+1}\Delta\boldsymbol{v}+\frac{1}{K_1}\prod_{i=0}^{\frac{n}{2}-1}\Delta\boldsymbol{P}_{2i,2i+1}\Delta\boldsymbol{v}+\frac{1}{K_1}\prod_{i=0}^{\frac{n}{2}-1}\Delta\boldsymbol{P}_{2i,2i+1}\boldsymbol{v}$$

$$=\boldsymbol{u}+\frac{1}{K_1}\prod_{i=0}^{\frac{n}{2}-1}\boldsymbol{P}_{2i}\boldsymbol{P}_{2i+1}\Delta\boldsymbol{v}+\frac{1}{K_1}\sum_{i=0}^{\frac{n}{2}-1}\boldsymbol{P}_0\boldsymbol{P}_1\cdots\Delta\boldsymbol{P}_{2i,2i+1}\cdots\boldsymbol{P}_{n-2}\boldsymbol{P}_{n-1}\boldsymbol{v} \quad (27)$$

where $\Delta\boldsymbol{v}$, $\Delta\boldsymbol{u}$ and $\Delta\boldsymbol{P}_{2i,2i+1}$ are errors of $\boldsymbol{v}$, $\boldsymbol{u}$, and $\boldsymbol{P}_{2i}\boldsymbol{P}_{2i+1}$, respectively.

Here, the error analysis of $\Delta\boldsymbol{v}$ is beyond this paper [6], [11], so we assume that $\Delta\boldsymbol{v}=0$, and $\Delta\boldsymbol{P}_{2i,2i+1}$ is the error introduced by double rotation. According to eq. (27), we obtain

$$\Delta\boldsymbol{u}=\frac{1}{K_1}\sum_{i=0}^{\frac{n}{2}-1}\boldsymbol{P}_0\boldsymbol{P}_1\cdots\Delta\boldsymbol{P}_{2i,2i+1}\cdots\boldsymbol{P}_{n-2}\boldsymbol{P}_{n-1}\boldsymbol{v} \quad (28)$$

$$\|\Delta\boldsymbol{u}\|=\frac{1}{K_1}\left\|\sum_{i=0}^{\frac{n}{2}-1}\boldsymbol{P}_0\boldsymbol{P}_1\cdots\Delta\boldsymbol{P}_{2i,2i+1}\cdots\boldsymbol{P}_{n-2}\boldsymbol{P}_{n-1}\boldsymbol{v}\right\|$$

$$\le\frac{1}{K_1}\left\|\sum_{i=0}^{\frac{n}{2}-1}\boldsymbol{P}_0\boldsymbol{P}_1\cdots\Delta\boldsymbol{P}_{2i,2i+1}\cdots\boldsymbol{P}_{n-2}\boldsymbol{P}_{n-1}\right\|\cdot\|\boldsymbol{v}\| \quad (29)$$

According to eq. (31), for all *i*, we obtain

$$\|\boldsymbol{P}_{2i}\boldsymbol{P}_{2i+1}\|=\frac{1}{k_{2i}k_{2i+1}}\|\cos(\alpha_{2i}+\alpha_{2i+1})+\sin(\alpha_{2i}+\alpha_{2i+1})\| \quad (30)$$

where $k_{2i}=\sqrt{1+2^{-4i}}$ and $k_{2i+1}=\sqrt{1+2^{-(4i+2)}}$.

It makes matrix of $\Delta\boldsymbol{P}_{2i}$ be changed as

$$\Delta\boldsymbol{P}_{2i}=\begin{bmatrix}\sigma_{2i}\sigma_{2i+1}2^{-n-1} & 0\\ 0 & \sigma_{2i}\sigma_{2i+1}2^{-n-1}\end{bmatrix} \quad (31)$$

$$\|\Delta\boldsymbol{P}_{2i}\|_\infty=|\sigma_{2i}|\cdot|\sigma_{2i+1}|\cdot 2^{-n-1}=2^{-n-1} \quad (32)$$

Hence,

$$\|\Delta\boldsymbol{u}\|\le 2^{-n-1}\cdot\left|\sum_{i=0}^{\frac{n}{2}-1}\cos(\sum_{j=0}^{n-1}\alpha_j-\alpha_{2i}-\alpha_{2i+1})+\sin(\sum_{j=0}^{n-1}\alpha_j-\alpha_{2i}-\alpha_{2i+1})\right|\cdot\|\boldsymbol{v}\|$$

$$\left\|\frac{\Delta\boldsymbol{u}}{\boldsymbol{v}}\right\|\le\sqrt{2}\cdot\frac{n}{2}\cdot 2^{-n-1}=n\cdot\sqrt{2}\cdot 2^{-n-2} \quad (33)$$

The error upper bound would be indicated, and the extra bits and iterations to gain same accuracy of the conventional CORDIC algorithm could be estimated.

When $n=32$, we have $\left\|\frac{\Delta\boldsymbol{u}}{\boldsymbol{v}}\right\|\le 2.63378\times10^{-9}$ and the errors are smaller than the upper bound, which is also illustrated in Fig. 7.

7. Conclusion

This paper presents double rotation architecture and a novel σ-prediction algorithm of CORDIC iteration applying them to the sine/cosine generator. The proposed architecture does not require extra ROM or complicated determination hardware. The speed is improved by using carry-save adder (CSA) with reduce the delay time of the critical path.

The double rotation CORDIC architecture with a novel σ-prediction algorithm improves the latency of the conventional CORDIC algorithm at least 37.5%, the efficiency of the CORDIC computation is increased by bits and iterations, and it makes the latency of the conventional CORDIC algorithm improve at most 48%.

8. References

[1] A. Madisetti, Y. K. Wentus, and A. N. Wilson, Jr., “A 100-MHz, 16-bit, Direct Digital Frequency Synthesizer with a 100-dBc Spurious-Free Dynamic Range,” *IEEE J. Solid-State Circuits*, Vol. 34, Aug. 1999, pp. 1034-1043.

[2] S. Y. Park, N. I. Cho, S. U. Lee, K. Kim, “Design of 2K/4K/8K-Point FFT Processor Based on CORDIC Algorithm in OFDM Receiver,” *2001 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing* (*2001-PACRIM*), Vol. 2, Aug. 2001, pp. 457-460.

[3] K. Mowak, R. Nouta, M. Bekooij, and E. Deprettere, “An Efficient Implementation of a 256-Point FFT Processor with CORDIC for OFDM Systems,” *Proc. ProRISC*, 1998.

[4] J. E. Volder, “The CORDIC Trigonometric Computing Technique,” *IRE Transactions on Electronic Computers*, Vol. EC-8, 1959, pp. 330-334.

[5] J. S. Walther, “A Unified Algorithm for Elementary Functions,” *Spring Joint Computer Conference Proceedings*, Vol.38, 1971, pp.379-385.

[6] S. Wang, E. E. Swartzlander Jr., “Merged CORDIC Algorithm,” *Proc. Int’l Symp. Circuits and Systems*, 1995, pp.1988-1991.

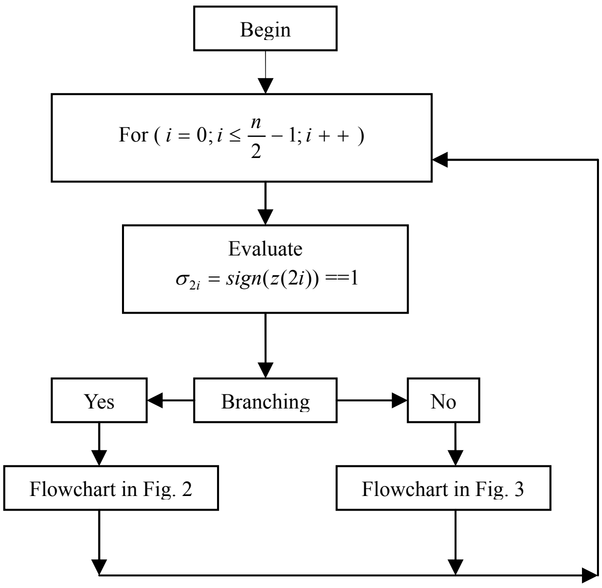
[7] I. Koren, “Computer Arithmetic Algorithms, Second Edition,” *A. K. Peters*, Natick, MA, 2002, Chapter 5.

[8] J. K. Kwak, J. H. Choi, E. E. Swartzlander, Jr., “High-Speed CORDIC Based on an Overlapped Architecture and a Novel σ-Prediction Method,” *Journal of VLSI Signal Processing*, Vol. 25, 2000, pp.167-177.

[9] Y. H. Hu, “The Quantization Effects of the CORDIC Algorithm”, *IEEE Transactions on Signal Processing*, Vol. 40, No. 4, 1992, pp.834-844.

[10] K. Kota, J. R. Callaro, “Numerical Accuracy and Hardware Trade-offs for CORDIC Arithmetic for Special-Purpose Processors”, *IEEE Transactions on Computers*, Vol. 42, No. 7, 1993, pp.769-779.

[11] T. Y. Sung, Y. H. Sung, “The Quantization Effects of CORDIC Arithmetic for Digital Signal Processing Applications”, *The 21st Workshop on Combinatorial Mathematics and Computation Theory*, Taichung Healthcare and Management University, Taichung, Taiwan, May 21~22, 2004, pp. 16-25.



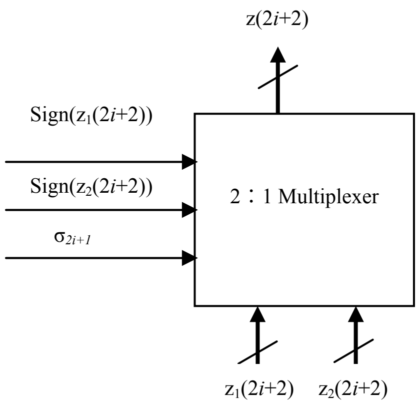


Fig. 4. (a) Determination circuit of $z(2i+2)$

Fig. 1. Flowchart for the σ_{2i+1}-prediction and $z(2i+2)$ determination algorithm. Detailed flowcharts for specific cases when sign(*z(2i)*) evaluation returns +1 , -1, and when the algorithm is in a branching are illustrated in Figs. 2 and 3, respectively.

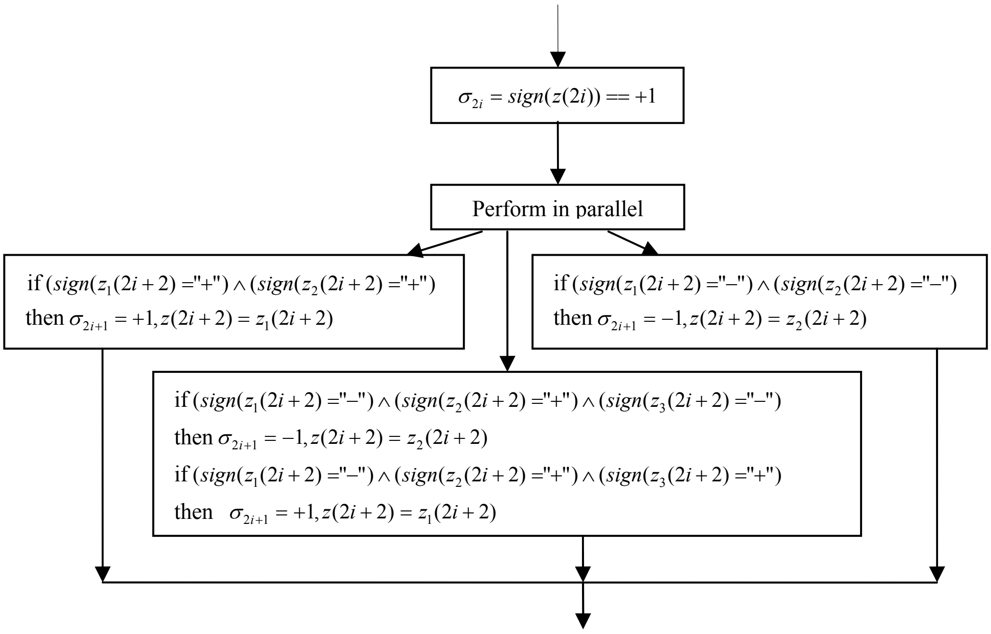


Fig. 2. Flowchart for *i*-iteration for the case when $\sigma_{2i} = sign(z(2i))$ evaluation returns +1.

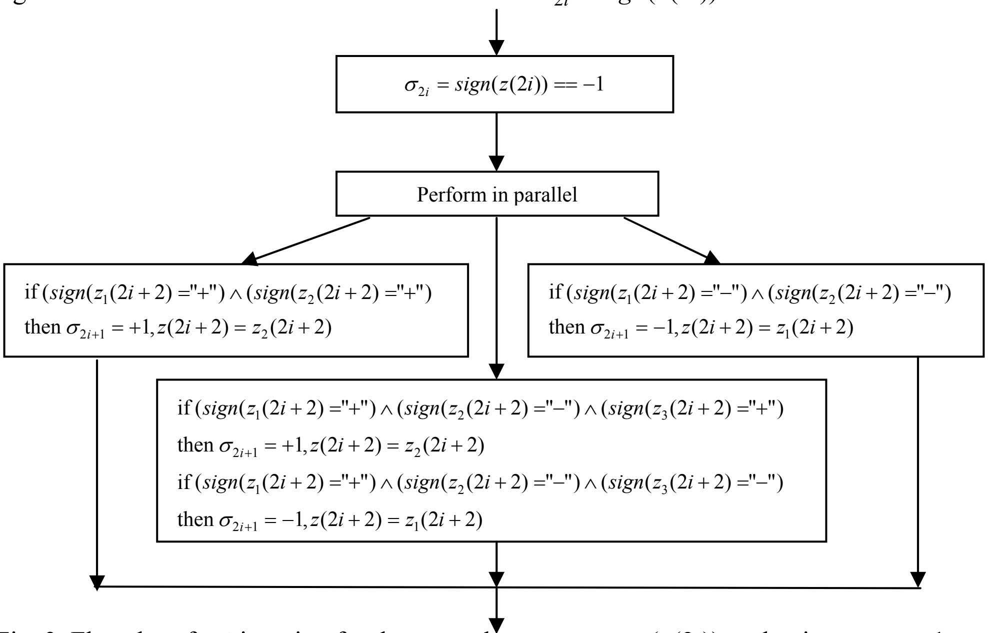


Fig. 3. Flowchart for *i*-iteration for the case when $\sigma_{2i} = sign(z(2i))$ evaluation returns -1.

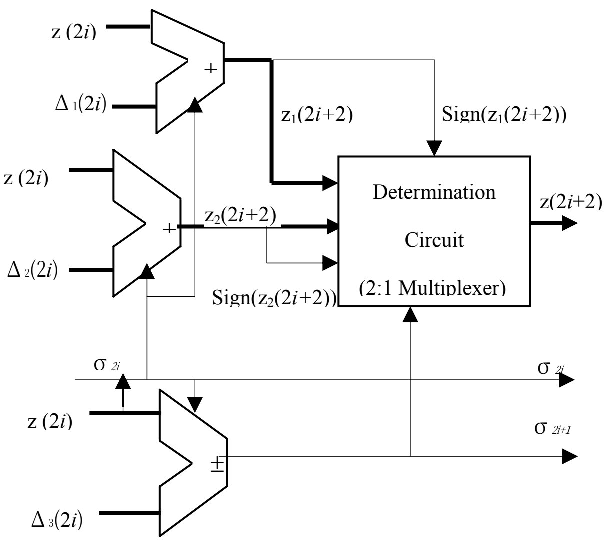


Fig.4 (b) $\sigma_{2i}, \sigma_{2i+1}$ and $z(2i+2)$ generator

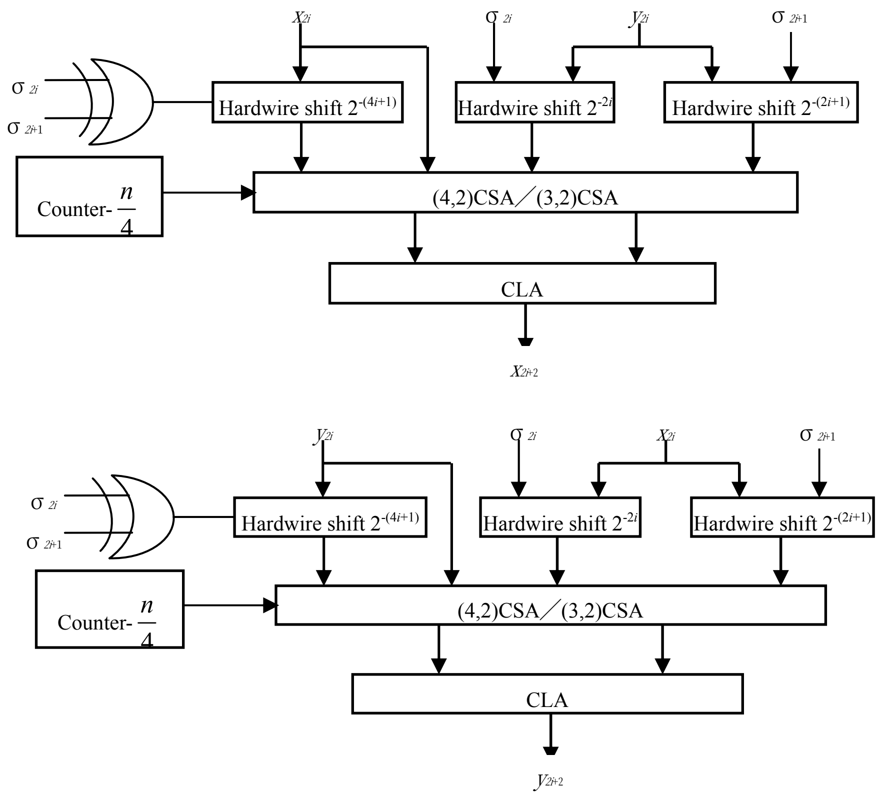


Fig. 5. The accelerated CORDIC architecture with the rotation mode in the circular coordinate system

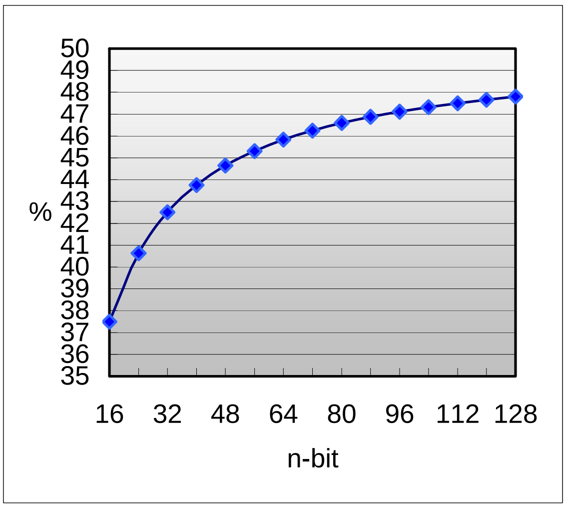


Fig. 6. The percentage of latency improvement versus number of bit in each operand

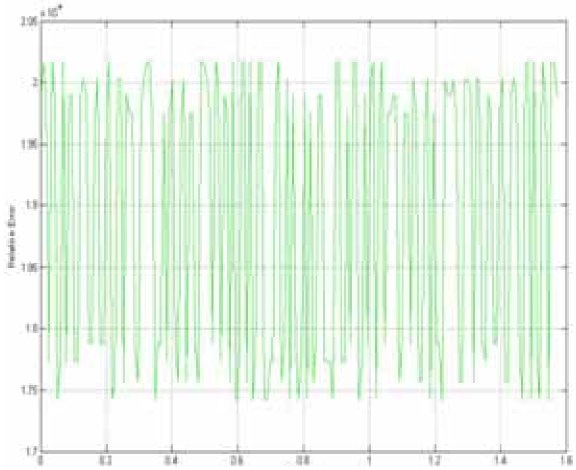


Fig. 7. The $\left\|\frac{\Delta \boldsymbol{u}}{\boldsymbol{v}}\right\|$ versus α of input angles