Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 897381, 8 pages
http://dx.doi.org/10.1155/2014/897381

Research Article

Design and Implementation of Hybrid CORDIC Algorithm Based on Phase Rotation Estimation for NCO

Chaozhu Zhang, Jinan Han, and Ke Li

College of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China

Correspondence should be addressed to Jinan Han; hanjinanbuaa@163.com

Received 17 January 2014; Revised 20 June 2014; Accepted 23 June 2014; Published 7 July 2014

Academic Editor: Brajesh K. Kaushik

The numerical controlled oscillator has wide application in radar, digital receiver, and software radio system. Firstly, this paper introduces the traditional CORDIC algorithm. Then in order to improve computing speed and save resources, this paper proposes a kind of hybrid CORDIC algorithm based on phase rotation estimation applied in numerical controlled oscillator (NCO). Through estimating the direction of part phase rotation, the algorithm reduces part phase rotation and add-subtract unit, so that it decreases delay. Furthermore, the paper simulates and implements the numerical controlled oscillator by Quartus II software and Modelsim software. Finally, simulation results indicate that the improvement over traditional CORDIC algorithm is achieved in terms of ease of computation, resource utilization, and computing speed/delay while maintaining the precision. It is suitable for high speed and precision digital modulation and demodulation.

1. Introduction

Numerical controlled oscillator (NCO) is an important part of digital downconversion. It is widely used in radar wireless transceiver system and software radio system [1–3]. The main function of NCO is to produce two path sine and cosine data samples with variable frequency, discrete time, and mutually orthogonal. It has an advantage of high frequency precision and fast response.

The traditional implement method of NCO is lookup table and polynomial expansion method. Data accuracy of lookup table method depends on the size of the lookup table ROM. The size of the memory and the precision of phase accuracy are exponential relationship, which enlarges the resource consumption and reduces the processing speed of the system. In [4], it solves this problem by using store content mapping technology of odd-even symmetry to optimize the storage unit and reduce the storage resources to 12.5%. However, under the request of high precision, it still consumes a lot of resources. Polynomial expansion method is a real-time computing method which needs multiplier resources and has certain restrictions on the complexity and speed of the hardware. It is too hard for the two methods to trade off speed, accuracy, and resource. Coordinate rotation digital compute algorithm (CORDIC) is proposed to solve the problem. CORDIC algorithm uses a basic algorithm to replace the complex algorithm. CORDIC algorithm is easy to hardware implementation. It does not require hardware multiplier and all operations are only shift accumulation, which meets the hardware requirements of modular and regularization algorithm requirements.

Along with proposing high speed broadband receiver, the data accuracy and processing speed have a higher request. Under the background, traditional CORDIC algorithm has some inherent drawbacks, such as limited coverage angle and too much pipeline series which increase resource consumption and limit data processing speed. Aiming at these shortcomings, this paper puts forward an efficient pipeline architecture CORDIC algorithm for NCO design.

2. Traditional CORDIC Algorithm

Volder CORDIC algorithm was proposed in 1959, and in 1971, Walther unified the form of the algorithm. Meyer-base realized the algorithm [5, 6], using FPGA implementation for the first time. CORDIC algorithm has been applied in many fields, such as direct digital frequency synthesizer, fast

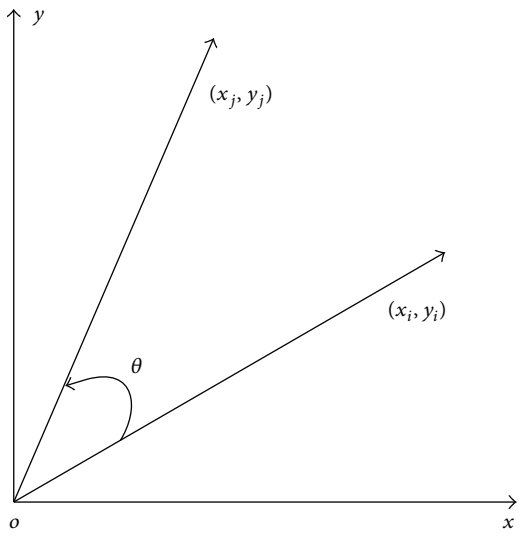


Figure 1: CORDIC vector rotation diagram.

Fourier transform, discrete cosine transform, digital modulation/demodulator, and stream processors [7–10]. According to certain phase, starting point (x_i, y_i) rotates continuously and approaches the final point gradually. Rotation vector diagram is shown in Figure 1.

In Figure 1, it is easy to get

$$\begin{bmatrix} x_j \\ y_j \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix}. \tag{1}$$

From the start to the end position, spinning process can be done by several steps and each step only rotates a certain phase:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} \cos\theta_n & -\sin\theta_n \\ \sin\theta_n & \cos\theta_n \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix}. \tag{2}$$

After extracting $\cos\theta_n$, formula (2) can be expressed as follows:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \cos\theta_n \begin{bmatrix} 1 & -\tan\theta_n \\ \tan\theta_n & 1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix}. \tag{3}$$

In order to simplify the hardware implementation, every operation sets each rotation phase to $\theta_n = \arctan(2^{-n})$. The total rotation phase is $\theta = \sum S_n\theta_n$. So $\tan\theta_n = S_n 2^{-n}$. Formula (3) can be expressed as follows:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \cos\theta_n \begin{bmatrix} 1 & -S_n 2^{-n} \\ S_n 2^{-n} & 1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix}. \tag{4}$$

From formula (4), in addition to the $\cos\theta_n$ coefficient, the operation is simple shift and addition.

In the final result, $\cos\theta_n$ can be eliminated by multiplying a known constant. For example, P, the number of iterations is 16 and $|\theta| \le \pi/4$. K can be expressed as follows:

$$\begin{aligned} K &= \prod_{n=1}^{16} \cos\theta_n = \prod_{n=1}^{16} \cos\left(\arctan\left(2^{-n}\right)\right) \\ &= \prod_{n=1}^{16} \left(1 - 2^{-2i}\right)^{-1/2}. \end{aligned} \tag{5}$$

In the phase rotation process, approximative rotational iterative formula is

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & -S_n 2^{-n} \\ S_n 2^{-n} & 1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix}. \tag{6}$$

Parameter z is used to judge when the iteration is over: $z_{n+1} = z_n - \theta_n$, $z_0 = \theta$. When $z_n < \theta$, $S_n = -1$. When $z_n \ge \theta$, $S_n = +1$. If the initial value is $(x_i, y_i) = (x_0, y_0) = (K, 0)$, (x_n, y_n) of Pth iteration will converge to $(\cos\theta, \sin\theta)$. The phase convergence satisfies the CORDIC convergence theorem [6]. The constant scaling factor K is fixed and can be precomputed as long as the precision N is determined. After analysis of traditional CORDIC algorithm calculation accuracy, the iteration number and phase precision are expressed as follows:

$$P \ge -\log_2\left[\tan\left(\Delta\theta_{\min}\right)\right], \tag{7}$$

where $\Delta\theta_{\min} = 2\pi/2^N$ and the input phase data width is N.

3. Hybrid CORDIC Algorithm Based on Phase Rotation Estimation

Common operation structures are iteration, pipeline, and differential CORDIC algorithm. Iterative structure occupies less hardware resources, but the processing data efficiency is low. Although the pipeline structure occupies more hardware resources, it can improve the throughput. Based on the two realization structures, implementation schemes have parallel pipelines, hybrid rotation CORDIC, angle encoding method, and so forth [11, 12]. The work in [13] puts forward the way of prediction rotation direction. The algorithm, applied in error analysis and elimination, has the advantages of fast speed. But it does not optimize hardware structure. Using the structure of the parallel hybrid CORDIC algorithm, the prediction scheme of [14] is more regular and simpler compared to previous approaches, which can reduce the number of iterations by more than 50 percent. However, the judgment of rotation direction is not optimized, which increases latency time and resources, so that it affects the throughput. The work in [15] puts forward a modified hybrid CORDIC algorithm and improved the precision of output data, but the method is more complex. Trading off the disadvantages of the above methods and advantages of pipeline structure and iterative structure, this paper simplifies the CORDIC algorithm further. By using the arctangent function property, it reduces the rotating judgment and add-subtract unit operation.

In this paper, attention is focused mainly on techniques that reduce the number of iterations, while keeping the low latency. The hybrid CORDIC algorithm based on phase rotation estimation is presented in this section, which can be addressed by digit-on-line pipelined CORDIC circuits and repetitive multiple accumulations architecture.

3.1. Rotation Phase Estimation. Assuming that the input phase length of CORDIC algorithm is N and pipeline series is P, rotation phase θ can be represented as follows:

$$\theta = \sum_{n=1}^{P} S_n \theta_n = \sum_{n=1}^{P} S_n \arctan\left(2^{-n}\right), \tag{8}$$

where $S_n = \pm 1$. It is noted that the initial value of n is 1, and the reason is that we restrict the rotation angle within the range $|\theta| \le \pi/4$ in the application example of NCO.

With the increase of rotational coefficient n, $\arctan(2^{-n})$ gets close to 2^{-n}. When $n \ge 1$, $2^{-n} > \arctan(2^{-n})$. Error is $\varepsilon_n = 2^{-n} - \arctan(2^{-n})$. Arctangent function is developed through the tailor equation:

$$\begin{aligned} \varepsilon_n &= 2^{-n} - \left[2^{-n} - \frac{1}{3}\left(2^{-n}\right)^3 + \frac{1}{5}\left(2^{-n}\right)^5 - \cdots\right] \\ &= \frac{1}{3}\left(2^{-n}\right)^3 - \frac{1}{5}\left(2^{-n}\right)^5 + \cdots, \end{aligned} \tag{9}$$

where $\varepsilon_n < (1/3)2^{-3n}$. The minimum phase value is 2^{-N}. In the process of phase rotation, when the error estimate is $(1/3)2^{-3n} \le 2^{-N}$, error generated by estimated value 2^{-n} can be ignored. The range of n is

$$n \ge \frac{N - \log_2 3}{3}. \tag{10}$$

When $n \ge m = \lceil (N - \log_2 3)/3 \rceil$, $\arctan(2^{-n}) \approx 2^{-n}$. Through (7), pipeline series of CORDIC algorithm is $P \ge N - 2$. The less the pipeline series are, the faster the speed is. When $P = N - 2$, we define the hybrid radix set:

$$\theta = \sum_{n=1}^{N-2} S_n \theta_n = \sum_{n=1}^{m} S_n \arctan\left(2^{-n}\right) + \sum_{n=m+1}^{N-2} S_n 2^{-n}. \tag{11}$$

After iterating $m+1$ times, the sum of residual rotation phase is $\sum \theta_n$, as shown in formula (12):

$$\begin{aligned} \sum \theta_n &= \sum_{n=m+1}^{N-2} S_n 2^{-n} \\ &= S_{m+1}2^{-m-1} + S_{m+2}2^{-m-2} \cdots + S_{N-2}2^{-N+2} < 2^{-m}. \end{aligned} \tag{12}$$

The actual residual phase is z_{m+1}. According to the traditional CORDIC algorithm theory, $z_{m+1} \approx \sum \theta_n$. So $z_{m+1} < 2^{-m}$. When the $(m+1)$th rotation begins, the new rotation phase $\widetilde{\theta}_{m+1}$ is ϕ_{m+1}, where the absolute value of z_{m+1} is ϕ_{m+1}. Thus $\tan \widetilde{\theta}_{m+1} = \widetilde{S}_{m+1}\widetilde{\theta}_{m+1}$. When $z_{m+1} < 0$, $\widetilde{S}_{m+1} = -1$. When $z_{m+1} \ge 0$, $\widetilde{S}_{m+1} = +1$. Taking it into formula (3),

$$\begin{bmatrix} x_{m+2} \\ y_{m+2} \end{bmatrix} = \cos\phi_{m+1} \begin{bmatrix} 1 & -\widetilde{S}_{m+1}\widetilde{\theta}_{m+1} \\ \widetilde{S}_{m+1}\widetilde{\theta}_{m+1} & 1 \end{bmatrix} \begin{bmatrix} x_{m+1} \\ y_{m+1} \end{bmatrix}. \tag{13}$$

After the rotation, the residual phase is 0. It shows that x_{m+2} and y_{m+2} are the output of cosine data and sine data.

3.2. Rotation Function Optimization and Error Analysis. In order to obtain cosine data from the new pipeline process, we put forward unidirectional rotation method to reduce the comparator and choose addition or subtractor. $\widetilde{\theta}_{m+1}$ should be expressed firstly. N-bits input phase needs to iterate m times. The results can be expressed as W. The residual phase at this time is z_{m+1}. $\widetilde{\theta}_{m+1}$ is expressed as follows:

$$T = \sum_{i=1}^{N-2} A_i 2^i = A_{N-2}2^{N-2} + A_{N-3}2^{N-3} + \cdots + A_1 2^1, \tag{14}$$

where $A_i = 1$ or 0. When $W[N-2] = 0$, $A_i = W[i]$. When $W[N-2] = 1$, A_i is that $W[i]$ flips every bit and adds 1:

$$\widetilde{\theta}_{m+1} = \frac{2\pi T}{2^N} = \frac{\widetilde{T}}{2^N}. \tag{15}$$

From formula (15), $\widetilde{T}$ is unknown. In the hardware implementation, $\widetilde{T}$ needs to be expressed as follows:

$$\begin{aligned} \widetilde{T} &= \sum_{i=1}^{N} B_i 2^i = 2\pi T \\ &= \sum_{i=1}^{N} A_i 2^i \left(2^2 + 2 + 2^{-2} + 2^{-5} + 2^{-9}\right), \end{aligned} \tag{16}$$

where $B_i = 1$ or 0. Taking all figures of $\widetilde{T}$ into (15),

$$\begin{aligned} \widetilde{\theta}_{m+1} &= \sum_{i=1}^{N-m-1} B_i 2^{i-N} \\ &= B_{N-m-1}2^{-m-1} + \cdots + B_2 2^{2-N} + B_1 2^{1-N}. \end{aligned} \tag{17}$$

Uniting formulas (13) and (17),

$$\begin{bmatrix} x_{m+2} \\ y_{m+2} \end{bmatrix} = \begin{bmatrix} x_{m+1} - \widetilde{S}_{m+1} \cdot \sum_{i=1}^{N-m-1} B_i 2^{i-N} \cdot y_{m+1} \\ \widetilde{S}_{m+1} \cdot \sum_{i=1}^{N-m-1} B_i 2^{i-N} \cdot x_{m+1} + y_{m+1} \end{bmatrix}, \tag{18}$$

where $\widetilde{S}_{m+1} = \pm 1$ and $B_i = 1$ or 0.

CORDIC algorithm of efficient pipeline uses $\{\arctan 2^{-1}, \ldots, \arctan 2^{-m-1}, \widetilde{\theta}_{m+1}\}$ instead of the traditional rotation phase $\{\arctan 2^{-1}, \ldots, \arctan 2^{-m-1}, \ldots, \arctan 2^{-P}\}$. The last set of rotation phase can be expressed as binary. Rotation direction is obtained directly from the last set. One more shift and add operation reduces $N - m - 3$ rotation times. Under the premise of ensuring phase and data accuracy, it reduces the resource consumption and improves the operation speed. Finally, with fewer lines series, constant coefficient is as follows:

$$\widehat{K} = \prod_{n=1}^{m+1} \cos\left(\arctan\left(2^{-n}\right)\right). \tag{19}$$

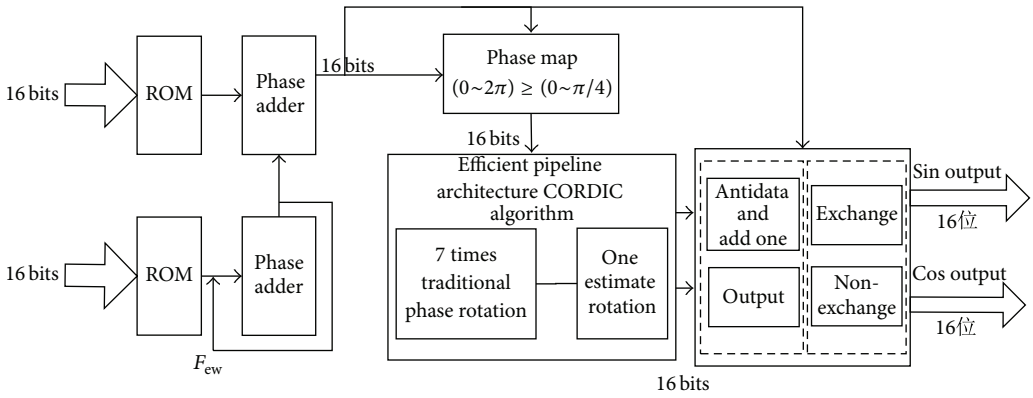


Figure 2: High speed and precision NCO structure.

At this time set the initial value (x_0, y_0) to $(\widehat{K}, 0)$. According to the above process, (x_{m+2}, y_{m+2}) converges to $(\cos\theta, \sin\theta)$.

The cosine error of this algorithm can be divided into three parts:

(1) the quantization errors are caused due to the limited word length,

(2) limited phase word length leads to approximation error,

(3) the phase estimation gives rise to the rotation estimation error.

Quantization error is in an inverse ratio to word length and output word length is set by pipeline series. The more the pipeline series are, the lower the quantization error is. But the increase of pipeline series will lead to resources consumption. So according to the data figure, it is necessary to trade off pipeline series and quantization error. Considering the hardware consumption, computing speed, and precision, [7] proposes the optimization method of data bits and pipeline series. According to [16], the quantization error consists of two parts, the quantization error produced before and this time. It can be expressed as follows:

$$\left|\mathbf{E}_n\right| \le \left|\left|\mathbf{e}_n\right| + \sum_{i=0}^{n-1}\left(\prod_{j=i}^{n-1} S_j\right)\left|\mathbf{e}_i\right|\right|, \tag{20}$$

where $\mathbf{E}_n$ is the sum of quantization error and $\mathbf{e}_n$ is the nth phase rotation quantization error with $S_j = \pm 1$.

When the output data is N and $\mathbf{e}_i = [e_{xi} \;\; e_{yi}]^T$, $|e_{xi}| \le \varepsilon$, $|e_{yi}| \le \varepsilon$, $\varepsilon = 2^{-N-1}$, $|\mathbf{e}_i|$ can be expressed as follows:

$$\left|\mathbf{e}_i\right| = \sqrt{e_{xi}^2 + e_{yi}^2} \le \sqrt{2} \times 2^{-N-1}. \tag{21}$$

According to formula (7), when phase length is N, phase resolution is $\varphi = 2\pi/2^N$. Approximation error produced by limited phase word length can be expressed as follows:

$$|A| = \frac{\left|V - V'\right|}{\left|V'\right|} \le 2\sin\left(\frac{\Delta\theta}{2}\right) \le \Delta\theta \le \varphi, \tag{22}$$

where V is the actual value and V' is the error value. $\Delta\theta$ is the difference between real phase and approximate phase. In the final rotating phase estimate, rotating phase $\arctan(2^{-n})$ is instead of 2^{-n}. Arc value of 2π can be replaced only by binary values similarly. In formula (16), the generated error can be expressed as follows:

$$\begin{aligned} |B| &= \sum_{i=m+2}^{N-3}\left(2^{-i} - \arctan\left(2^{-i}\right)\right) + 2\pi - 6.2832 \\ &\approx \sum_{i=m+2}^{N-3}\left(2^{-i} - \arctan\left(2^{-i}\right)\right) + 2 \times 10^{-5}. \end{aligned} \tag{23}$$

4. The FPGA Design and Implementation of NCO

4.1. High Speed and Precision NCO Structure. This paper adopts efficient pipelining structure CORDIC algorithm for high speed and high precision NCO. Its structure is shown in Figure 2. We take 16-bit phase control words as an example. Firstly, input is a 16-bit phase control word and 16-bit frequency control word. Secondly, through the phase accumulator and phase adder, the output is 16-bit phase value. Phase map generates the $\{0, \pi/4\}$ phase. Thirdly, the shift-add efficient pipelining structure processes phase data. Finally according to the previous mapping relation, 16-bit sine and cosine data can be generated.

The range θ of rotation angle value is $\{-44.855^\circ, 44.855^\circ\}$ and approximates to $\{-\pi/4, \pi/4\}$. It does not meet the $\{0, 2\pi\}$ scope of phase. Before 16-bit phase values are sent into the algorithm, cosine function property can judge the highest, second highest, and third highest bit. According to certain mapping relation, the highest 3 bits of 16-bit phase value and phase can be reduced to $3'b000$ and $\{0, \pi/4\}$, respectively. The highest bit controls sine data symbol. If the bit is 1, the algorithm flips the sine data and adds 1. On the other hand, the algorithm does not process input data. The highest bit and second highest bit control cosine data symbol. If they are different, the algorithm flips sine data and adds 1. Otherwise, it remains to be the input data. The second highest bit and third highest bit control the location of cosine data and sine

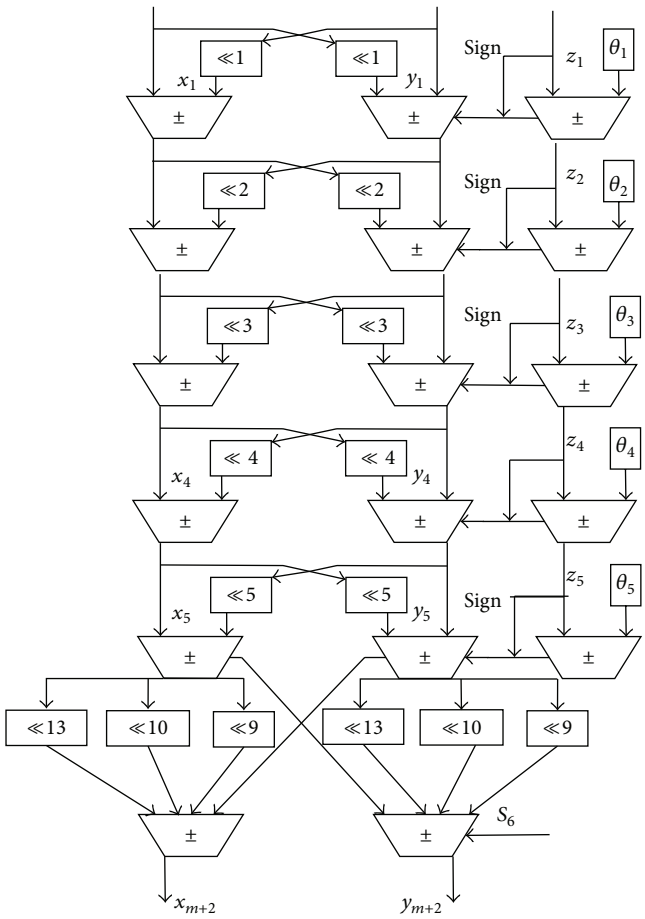


Figure 3: Phase rotation estimation based on pipeline structure.

data. If they are different, the algorithm exchanges cosine data and sine data. Or else it remains to be the input data.

4.2. Internal Architecture Design and the Major Implementation Steps. According to formula (10), our algorithm needs $m = \lceil (16 - \log_2 3)/3 \rceil = 5$ times for traditional phase rotation and one time for rotation phase estimation. If $\widetilde{\theta}_6 = 2^{-9} + 2^{-10} + 2^{-13}$, the pipeline structure is shown in Figure 3. Each level only needs three adder-subtractors, two or six phase shift registers, and a phase coefficient memory and reduces more than a half of the rotation phase judgment and shift operation. For reducing the critical path in the pipelined implementation of traditional CORDIC, the differential CORDIC (D-CORDIC) algorithm based on digit-on-line pipelined CORDIC circuits [17] can be used to achieve higher throughput and lower pipeline latency. D-CORDIC algorithm is equivalent to the usual CORDIC in terms of accuracy as well as convergence. The system architecture uses parallel and pipeline differential CORDIC architecture to reduce latency and improve throughout. Digit-on-line pipelined CORDIC circuits take place of continuous phase accumulation in Figure 3.

From what has been discussed above, the major steps of our algorithm are as follows.

Table 1: Comparison of resource use.

Algorithm	Delay		
	$N = 16$	$N = 24$	$N = 32$
Traditional pipeline structure	$64T_{FA}$	$110T_{FA}$	$160T_{FA}$
Hybrid CORDIC algorithm	$43T_{FA}$	$72T_{FA}$	$96T_{FA}$
Ours	$26T_{FA}$	$37T_{FA}$	$72T_{FA}$

Step 1. Phase rotation is limited in the range of $\{-\pi/4, \pi/4\}$.

Step 2. Traditional or differential CORDIC algorithm implements partial phase rotation.

Step 3. Using a relatively simple prediction scheme, we divide original CORDIC rotations into the lower part and the higher part.

Step 4. Differential CORDIC or traditional architecture is proposed to compute rotation direction. The lower part is computed by continuous accumulation or online architecture [18] based on differential CORDIC and the higher part is predicted by rotation phase estimation.

Step 5. According to phase mapping relationship, the required high precision and high speed cosine data is produced.

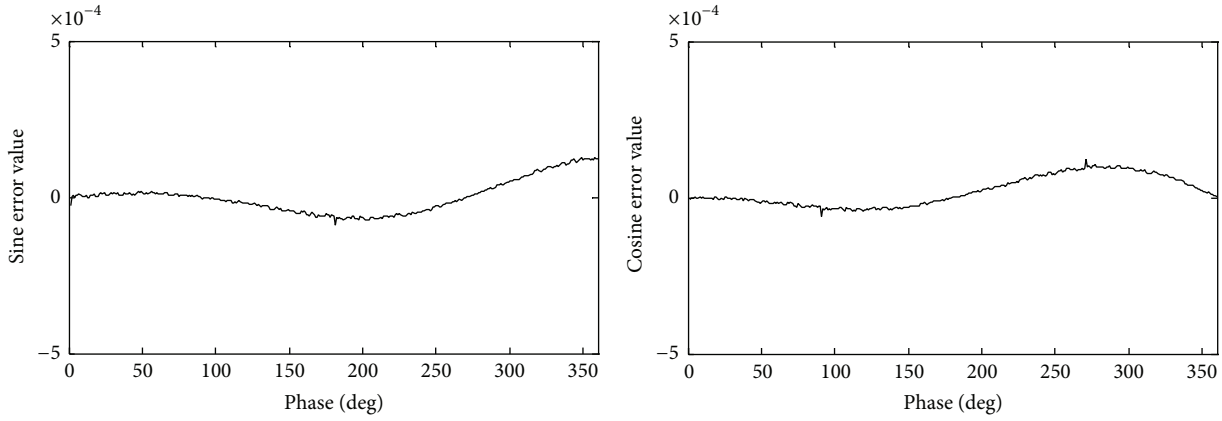
4.3. Simulation Results. Table 1 compares the delay of some CORDIC rotation methods. Our proposed algorithm could obtain good performance in delay and resource.

To compare our pipeline CORDIC algorithm with other previously proposed methods fairly, we assume CSA is universal adder in all algorithms and fast carry-propagate adders (CPA) are used in the last stage to take carry-save forms back to the input initial phase value.

In [13], the first m iterations use the traditional continuous comparison method, the same as the traditional CORDIC. The delay increases logarithmically with the maximum number of shifts. If the delay of carry-propagate adder (CPA) is $\lceil \log_2 N \rceil \cdot T_{FA}$, the latency of $(N - m)$ iterations increases linearly with the word length and the delay is $(4N/3) \cdot T_{FA}$.

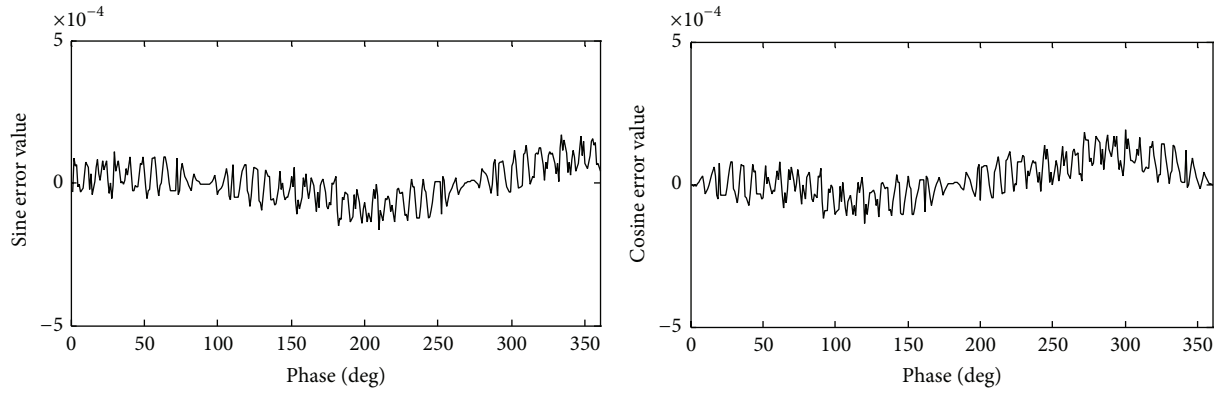
Based on the calculation method above, the traditional CORDIC based on pipeline architecture has the delay of $\lceil \log_2 N \rceil \cdot T_{FA} \cdot N$.

Unlike the above methods, our proposed method reduces the number of iterations and simplifies the Z datapath. The first iterations still adopt the traditional CORDIC algorithm where a delay of $\lceil \log_2 N \rceil \cdot T_{FA}$ is assumed for an N-bit CPA. The accumulations of final iteration use repetitive multiple accumulations architecture [19], which has much higher throughput and less delay compared with serial accumulator and pipelined adder based on carry-save addition as well. The last iteration increases linearly and the delay is $(4K/3) \cdot T_{FA}$, where K is the full-adder number for the accumulations based on adder-tree architecture.



(a) Sine value error (b) Cosine value error

Figure 4: Sine and cosine error statistics of traditional pipeline structure.



(a) Sine value error (b) Cosine value error

Figure 5: Sine and cosine error statistics of ours.

Table 2: Algorithm resource use comparison.

Algorithm	Resource		
	Logic unit	Register	Storage size
Traditional pipeline structure	1177	754	63
Hybrid CORDIC algorithm	1034	576	26
Ours	819	393	26

According to the structure shown in Figure 2, traditional pipeline structure and efficient pipeline structure based on rotation phase estimation are implemented by verilog language, respectively. Hardware platform is a Cyclone II series EP2C8Q208C8 chip and software platform is in Quartus II of Altera company. Modelsim 10.0 simulation software tests the experience result. Firstly, the input frequency control word, phase control word, and clock frequency are set to $16'h1999$, $16'd0$, and 100 MHz. Output frequency is 10 MHz. Compared to the use of resources, the result can be expressed in Table 2.

Through the comparison in Table 2, our proposed algorithm reduced resource obviously.

This algorithm precision is the same as traditional CORDIC algorithm, $\Delta\widehat{\theta}_{\min} = 2\pi/2^N$. The input frequency control word, phase control word, and clock frequency are set to $16'h00B6$ and $16'd0$. The output frequency is 0.3125 MHz. Compared with the theoretical value and experiment value, the error statistic is shown in Figures 4 and 5. The simulation runtime of our proposed algorithm costs less than the traditional CORDIC algorithm in Figure 6.

Compared with Figures 4 and 5, our proposed algorithm has the larger error volatility, while the two kinds of the algorithm error will be controlled in $(-5 \times 10^{-4}, 5 \times 10^{-4})$.

Though our algorithm structure reduces logic unit, it guarantees the cosine data accuracy. Figure 7 shows the NCO simulation waveform of efficient pipeline structure.

It is necessary to obtain efficient bits of phase, optimum iteration number, and data width. We do the above experiment 200 times. The random angle value is restricted from 0 to 45°. When the iteration number is 5~8 and the series of data width are 15, 16, 18, and 21, we can obtain the effective bits. The relationship of effective bit number with iteration times and data width is shown in Table 3. The data unit is degree.

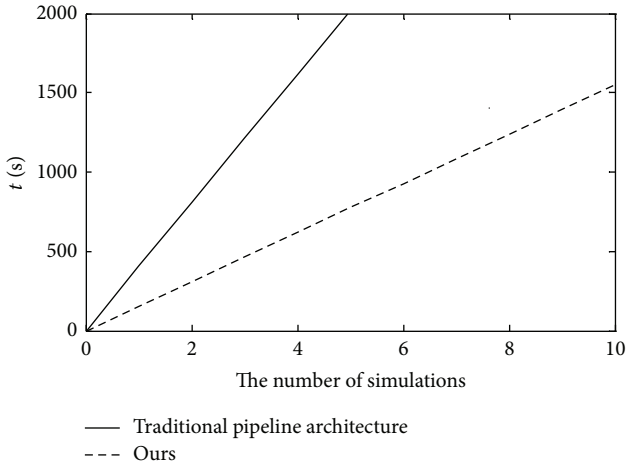


Figure 6: The runtime of algorithms' comparison.

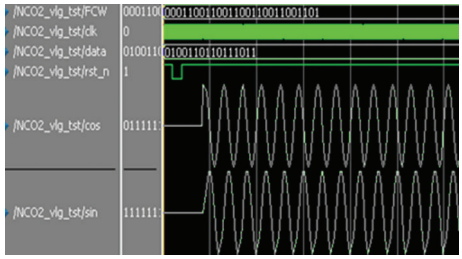


Figure 7: NCO simulation waveform.

Table 3: Relationship of effective bit number with iteration number and data width.

	Data width (iteration number)			
	14 (5)	16 (6)	18 (7)	21 (8)
Estimated value	22.137	22.137	22.137	22.137
Simulated value	23.672	22.458	22.281	22.132

The algorithm error will be controlled in $(-5 \times 10^{-4}, 5 \times 10^{-4})$, when the iteration number is greater than 6. The experimental results show that the effective bit number is 13. Through calculating the minimum number of microrotation, the effective bit number is generally seven greater than iteration number. The calculation of total quantization errors could be calculated through this method.

5. Conclusion

In this paper, the hybrid CORDIC algorithm based on phase rotation estimation is proposed to design NCO. In the case of assuring the high precision output, the efficient CORDIC algorithm reduces more than a half of the rotation phase judgment and shift operation. Resource consumption, operation speed, and system delay have much better performance than traditional CORDIC algorithm. In terms of electronic countermeasures, it has a certain practicality. The algorithm has been successfully used in high speed broadband ADS-B receiver and shows good performance.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant no. 61172159) and the Fundamental Research Funds for the Central Universities (HEUCFT1101).

References

[1] L. Guo, S. Tian, Z. Wang, and J. Luo, "Study of NCO realization in parallel digital down anversion," *Chinese Journal of Scientific Instrument*, vol. 33, no. 5, pp. 998–1004, 2012.

[2] Q. Zhang, Y. Luo, S. Chen, and J. Yan, "Design and implementation of NCO based on phase rotation," *Systems Engineering and Electronics*, vol. 32, no. 5, pp. 908–911, 2010.

[3] X.-N. Yang, Y.-C. Lou, and J.-L. Xu, *Software Radio Technology and Application*, Beijing Institute of Technology Press, Beijing, China, 2010.

[4] W.-B. Qin, L.-Y. Luo, and T.-Y. Li, "Study on the efficient technology applied to high precision and high resolution storage in high speed NCO," *Journal of Sichuan University (Engineering Science Edition)*, vol. 39, no. 1, pp. 156–159, 2007.

[5] J. Volder, "The CORDIC trigonometric computing technique," *IRE Transactions on Electronic Computers*, vol. 8, no. 3, pp. 330–334, 1959.

[6] J. Walther, "A unified algorithm for elementary functions," in *Proceedings of the Spring Joint Computer Conference*, vol. 38, pp. 379–385, 1971.

[7] S.-Q. Wan, W.-F. Chen, S.-R. Huang, H. Ji, and Z. Yu, "Implementation of a high-speed direct digital frequency synthesizer based on improved CORDIC algorithm," *Chinese Journal of Scientific Instrument*, vol. 31, no. 11, pp. 2586–2591, 2010.

[8] S. Y. Park and Y. J. Yu, "Fixed-point analysis and parameter selections of MSR-CORDIC with applications to FFT designs," *IEEE Transactions on Signal Processing*, vol. 60, no. 12, pp. 6245–6256, 2012.

[9] S. Aggarwal, P. K. Meher, and K. Khare, "Scale-free hyperbolic CORDIC processor and its application to waveform generation," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 2, pp. 314–326, 2013.

[10] H. Huang and L. Xiao, "CORDIC based fast radix-2 DCT algorithm," *IEEE Signal Processing Letters*, vol. 20, no. 5, pp. 483–486, 2013.

[11] T. Juang, "Low latency angle recoding methods for the higher bit-width parallel CORDIC rotator implementations," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 55, no. 11, pp. 1139–1143, 2008.

[12] P. K. Meher and S. Y. Park, "CORDIC designs for fixed angle of rotation," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 21, no. 2, pp. 217–228, 2013.

[13] S. Wang, V. Piuri, and E. E. Swartzlander Jr., "Hybrid CORDIC algorithms," *IEEE Transactions on Computers*, vol. 46, no. 11, pp. 1202–1207, 1997.

[14] S.-F. Hsiao, Y.-H. Hu, and T.-B. Juang, "A memory-efficient and high-speed sine/cosine generator based on parallel CORDIC rotations," *IEEE Signal Processing Letters*, vol. 11, no. 2, pp. 152–155, 2004.

[15] X. Zhang, R. Xin, Q. Wang, and H. Li, "Design of direct digital frequency synthesizer based on improved hybrid CORDIC algorithm," *Acta Electronica Sinica*, vol. 36, no. 6, pp. 1144–1148, 2008.

[16] Y. H. Hu, "The quantization effects of the CORDIC algorithm," *IEEE Transactions on Signal Processing*, vol. 40, no. 4, pp. 834–844, 1992.

[17] H. Dawid and H. Meyr, "The differential CORDIC algorithm: constant scale factor redundant implementation without correcting iterations," *IEEE Transactions on Computers*, vol. 45, no. 3, pp. 307–318, 1996.

[18] M. D. Ercegovac and T. Lang, "Redundant and on-line CORDIC: application to matrix triangularization and SVD," *IEEE Transactions on Computers*, vol. 39, no. 6, pp. 725–740, 1990.

[19] P. K. Meher, "New approach to scalable parallel and pipelined realization of repetitive multiple accumulations," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 55, no. 9, pp. 902–906, 2008.