**Academic Journal of Science and Technology**
ISSN: 2771-3032 | Vol. 5, No. 1, 2023

# Research and Implementation of a Numerical Control Oscillator with Improved Pipelined CORDIC Algorithm

**Yuechuan Chen, Yu Liu**

Chongqing University of Posts and Telecommunications, Sensor Components and Microsystem Laboratory, Chongqing, 400065, China

**Abstract:** As the recognized core of electronic systems, frequency synthesizers have been applied in many communication fields. NC oscillator (NCO) is the main component of the frequency synthesizer. It helps to generate high-precision and high-frequency signals, so it has been widely used. NCO implementation methods include table lookup method, polynomial expansion method, coordinate rotation digital computer (CORDIC) algorithm, etc. CORDIC algorithm is one of many commonly used methods in trigonometric function calculation and digital signal processing, and is often used as the core of DDS (direct digital synthesis) to generate signals. Compared with table lookup and polynomial expansion, CORDIC algorithm has higher efficiency in signal generation and hardware utilization. In view of the disadvantages of traditional CORDIC algorithm, which takes up large resources and has relatively slow calculation speed, in order to improve the output efficiency, this paper uses an efficient 12-stage pipelined CORDIC architecture and a very small lookup table (LUT) to implement a sine wave generator. The system is coded and simulated in Quartus and ModelSim. The results show that the proposed structure can increase the operating speed of the system from 217.77 MHz to 291.04 MHz, and improve the output efficiency of the system.

**Keywords:** CNC oscillator, Waveform generator, CORDIC algorithm, Parallel pipeline structure, Improved pipeline.

## 1. Introduction

Digital controlled oscillator (NCO) is an important part of software radio, direct digital synthesizer (DDS), fast Fourier transform (FFT), etc., and also one of the main factors that determine its performance. It has the characteristics of high frequency accuracy, short conversion time, high spectral purity and easy programming of frequency phase, Therefore, it is widely used in software radio digital up-conversion, down-conversion and various frequency and phase digital modulation and demodulation systems. With the improvement of chip integration, digital controlled oscillators are more and more widely used in signal processing, digital communication, modulation and demodulation, frequency conversion and speed regulation, guidance control, power electronics and other fields.

In order to effectively improve the accuracy of MEMS gyroscope, it is an effective and stable way to convert the analog measurement and control circuit in the measurement and control circuit of micro gyroscope into digital measurement and control circuit. In the digital measurement and control circuit, the driving circuit plays a decisive role. The main function of this circuit is to stabilize the output frequency and amplitude of the MEMS gyroscope and realize the stable operation of the gyroscope. The digital phase-locked loop is mainly used to stabilize the output frequency. As an important part of the digital phase-locked loop, the digital controlled oscillator (NCO) plays a vital role in the performance of the whole digital phase-locked loop. Therefore, the design of high-precision numerical control oscillator is the focus and difficulty of the whole circuit. The traditional implementation methods of NCO mainly include table lookup method, polynomial expansion method or approximation method [1], but these methods are difficult to give consideration to speed, accuracy and resources. At present, the numerical control oscillator is mainly realized by CORDIC algorithm. Compared with the traditional implementation method of NCO, CORDIC algorithm can generate high-precision sine and cosine waveforms without using multipliers, but only simple shift and addition operations, especially suitable for FPGA hardware implementation [2].

CORDIC (Coordinate Rotation Digital Computer) algorithm, namely coordinate rotation digital calculation method, is an effective and practical method for generating sine and cosine waveforms. It was first proposed by J.D. Volder et al. in 1959 [3], mainly used in the calculation of trigonometric functions, hyperbolas, exponents and logarithms [4]. CORDIC algorithm has strong portability, convenient implementation and controllable precision [5-7]. In the long process of development, many scholars have carried out various researches on the algorithm and proposed different improvement methods for different defects of the algorithm. In 2005, Maharatna K et al. proposed a serial structure based on CORDIC algorithm [8] and implemented it on FPGA in hardware. This method takes less resources, making the design of control unit slightly complex, timing control more cumbersome, and system processing speed low. In 2013, Huang Jun et al. proposed the pipelined CORDIC algorithm, which improved the computational efficiency of CORDIC, but greatly increased the logical resources occupied by the algorithm [9]. In 2014, Liu Zhangfa et al. proposed the binary angle recoding CORDIC algorithm [10], which can omit the next rotation direction judgment to be performed in each iteration, thus reducing the resource occupation and improving the speed of understanding. In the same year, Xu Cheng and others proposed an iterative merging CORDIC algorithm based on angle recoding [11], which further reduced the resource occupation.

Based on the traditional serial structure, parallel structure, pipelined structure and other technologies of CORDIC, this paper proposes an improved parallel pipelined CORDIC implementation method. In order to improve the output efficiency and reduce the resource consumption, this paper proposes an improved parallel pipelined CORDIC architecture to implement a numerical control oscillator. This structure can increase the maximum clock speed of the system from 217.77 MHz to 291.04 MHz, and improve the output efficiency of the system.

## 2. CORDIC Algorithm Principle

The operation method of CORDIC algorithm is based on triangle operation, which relies on vector rotation to continuously map and coordinate between polar coordinates and rectangular coordinates. Unlike LUT, CORDIC algorithm does not completely rely on the predetermined phase, frequency and amplitude values to calculate the coordinates on the sine wave. It is more flexible than LUT. CORDIC is superior to the traditional LUT method because of its ability to generate orthogonal components and in-phase components at the same time. It can calculate the required trigonometric function values, such as sine, cosine and hyperbolic function d (ex-sinh, cosh and tanh), and can achieve any required accuracy.

CORDIC algorithm is divided into rotation mode and vector mode. Due to the consistency of hardware architecture, the basic principle of the final implementation of rotation mode and vector mode is the same. This paper mainly analyzes and studies the rotation mode. The geometric principle of rotation mode is shown in Figure 1. There is a point P1 in the coordinate system. After rotating P1 by 2 degrees, the point P2 is obtained. The angle 1 is $\alpha$, called initial angle; Angle 2 is the rotation angle $\beta$ express.

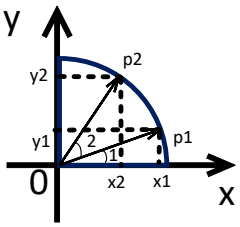


**Figure 1.** Geometric schematic diagram of CORDIC algorithm order

$$OP1 = OP2 = \sqrt{x_1^{\,2} + y_1^{\,2}} = \sqrt{x_2^{\,2} + y_2^{\,2}} = R \quad (1)$$

Get the expression of P1 point:

$$\begin{cases} x_1 = R\cos\alpha \\ y_1 = R\sin\alpha \end{cases} \quad (2)$$

Then the expression of P2 point can be obtained:

$$\begin{cases} x_2 = R\cos(\beta+\alpha) = x_1\cos\beta - y_1\sin\beta \\ y_2 = R\sin(\beta+\alpha) = y_1\cos\beta + x_1\sin\beta \end{cases} \quad (3)$$

Therefore, after P1 and rotation angle are known, point P2 can be calculated. The coordinates of P2 can be further written as:

$$\begin{cases} x_2 = \cos\beta\,(x_1 - y_1\tan\beta) \\ y_2 = \cos\beta\,(y_1 + x_1\tan\beta) \end{cases} \quad (4)$$

In order to facilitate calculation, the vector modulus value is not considered in rotation, and will be $\cos\beta$ removed. At this time, the degree of rotation coordinate can be easily obtained. This operation is called pseudorotation. At this time, 1-4 becomes a pseudorotation equation:

$$\begin{cases} x_2 = x_1 - y_1\tan\beta \\ y_2 = y_1 + x_1\tan\beta \end{cases} \quad (5)$$

The idea of pseudo rotation is to split the angle, and the specific method is to rotate the angle $\beta$ divided into several angles of equal size $\beta_i$ i.e $\beta = \sum_{i=0}^{\infty}\beta_i$。 In order to facilitate the implementation of CORDIC algorithm on hardware, set the rotation angle of each time as $\beta_i$。 regulations $\beta_i$ satisfy $tan\beta_i = 2^{-i}$, there are:

$$\beta_i = tan^{-1}\,2^{-i} \quad (6)$$

$\sum tan\beta_i$ range and rotation angle of $\beta$ scope of[ $-99.7^{\circ}$,$99.7^{\circ}$ ]Consistent. Since the direction of each rotation is related to the size of the remaining angle after the last rotation, it is necessary to set a direction for each rotation $d_i$, if the sum of rotation angles is less than $\beta$, then $d_i$ is 1, indicating that the next rotation is clockwise. If the sum of rotation angles is greater than $\beta$, then $d_i$ a value of - 1 indicates that the next rotation is counterclockwise. After rotating a certain angle from the initial position, there will be a residual angle. Set the residual angle of rotation as $z_i$， $z_i = \beta - d_i\beta_i$, bringing (1-6) into: $z_i = \beta - d_i\,tan^{-1}\,2^{-i}$。 $z_i$ initially $\beta$ i.e $z_i(0) = \beta$。 along with $i$ the value of is increasing, hello it will approach 0 and the rotation will end.

$d_i$ After defining the rotation direction, formula (1-6) can be changed to:

$$\beta_i = tan^{-1}(d_i 2^{-i}) \quad (7)$$

When using pseudo rotation, each rotation will correspond to a residual angle, and a rotation will be generated at the end of rotation $\prod cos\beta_i$ accumulation of. Assuming the number of iterations $n = a-1$, the rotation coordinate of P2 should be $(x_a * \prod cos\beta_i, y_a * \prod cos\beta_i)$ therefore, the final coordinate angle of P2 should be:

$$\begin{cases} x_a * \prod cos\beta_i = (x_1\cos\beta - y_1\sin\beta) \\ y_a * \prod cos\beta_i = (y_1\cos\beta - x_1\sin\beta) \end{cases} \quad (8)$$

Namely

$$\begin{cases} x_a = \frac{1}{\prod cos\beta_i}(x_1\cos\beta - y_1\sin\beta) \\ y_a = \frac{1}{\prod cos\beta_i}(y_1\cos\beta + x_1\sin\beta) \end{cases} \quad (9)$$

From the above formula, when $i$ the value is large enough, $\Pi cos\beta_i$ approaches to a constant, which can be set manually at this time $x_1$、 hello the value of<p mtid='539 '/> $x_1 = \Pi\, cos\beta_i, y_1 = 0$ can be calculated $\cos\beta, \sin\beta$ to realize the operation of sine and cosine.

The idea of CORDIC algorithm is to use a series of iterative algorithms to yaw at a fixed parameter angle to approximate the required rotation angle. The implementation of the algorithm is to iteratively approximate the required value. At the same time, it can be seen that due to the limitation of hardware implementation, infinite iteration is not allowed, otherwise the required resources and processing time will

increase. Therefore, in practical application, it is necessary to select the number of iterations according to the requirements of the system to achieve the accuracy required by the system.

## 3. CORDIC Algorithm Structure Analysis

From the pseudorotation equation mentioned above, the following expression can be obtained:

$$\begin{cases} x_{i+1} = x_i - d_i y_i 2^{-i} \\ y_{i+1} = y_i + d_i x_i 2^{-i} \\ z_{i+1} = z_i - d_i \tan^{-1} 2^{-i} \end{cases} \quad (10)$$

Where $x_i$ , $y_i$ ,, $z_i$ respectively represent the abscissa before rotation, the ordinate before rotation, and the remaining angle before rotation.

In combination with the selection direction mentioned above $d_i$ , there are:

$$d_i = \begin{cases} +1 & z_i \geq 0 \\ -1 & z_i < 0 \end{cases} \quad (11)$$

Equations (10) and (11) are referred to as the overall iterative process of CORDIC algorithm circular system rotation mode.

From the iterative process expression, the basic processing unit of CORDIC algorithm for hardware implementation in rotation mode can be obtained. As shown in Figure 2.

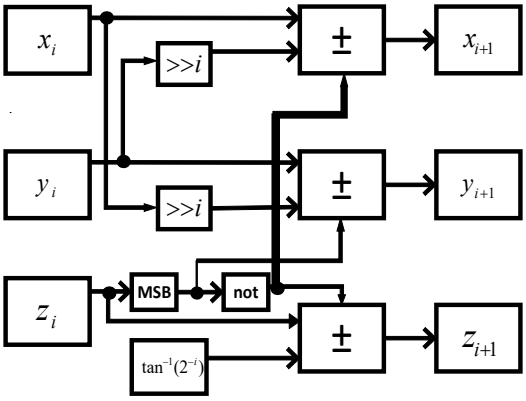


**Figure 2.** CORDIC algorithm circle system rotation mode processing unit

It can be seen from the above figure that the basic processing unit of CORDIC algorithm is composed of three adders, a lookup table LUT (storage angle) and two shift operators. It can be seen from this structure that the advantages of CORDIC algorithm are perfectly reflected. Since the basic structure of the processing unit used in each iteration is the same, only in terms of shift amount and storage angle, two hardware implementation architectures of CORDIC algorithm can be obtained, namely, serial architecture and parallel architecture. The general block diagram of the serial architecture is shown in Figure 3.

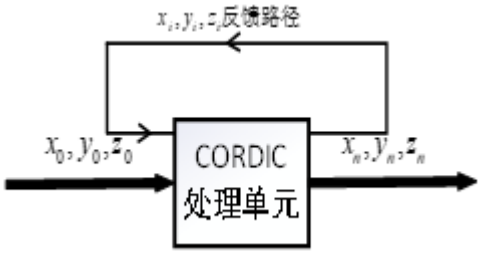


**Figure 3.** CORDIC algorithm serial structure

The block diagram of parallel structure is shown in Figure 4.

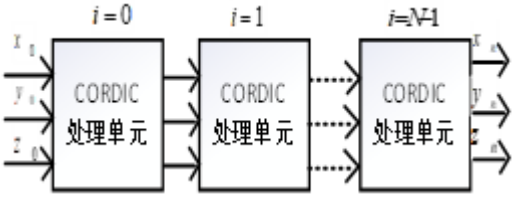


**Figure 4.** CORDIC algorithm parallel structure

The detailed block diagram of the serial structure is shown in Figure 5.

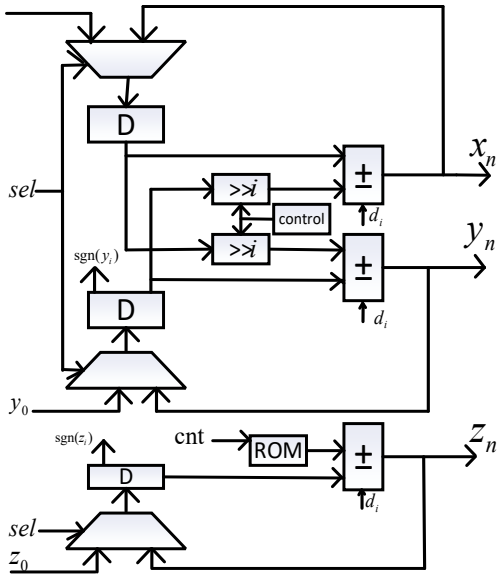


**Figure 5.** Detailed diagram of CORDIC algorithm serial structure

In Figure 5, sgn (yi) and sgn (zi) respectively represent the sign bits of yi and zi, that is, the highest bit. In operation, CORDIC algorithm will choose different operation modes (rotation mode and vector mode), and choose one of them to assign to di. Secondly, the adder determines the working mode (addition operation, subtraction operation) according to the selected value of di. ">>i" means that the input data is shifted to the right by i bit, which is controlled by the control module. The data stored in ROM is the rotation angle at each iteration $tand_i = 2^{-i}$, can be expressed in angle or radian. Take 16 iterations as an example, that is, i=0, 1, 2,..., 15, design a counter cnt count value of module 16 from 0 to 15, and generate the sel signal in the figure by decoding the count value, that is, when cnt>0, sel is 1, otherwise it is 0. At the same time, the count value can also dynamically adjust the shift amount, as shown in Figure 6. In addition, the count value can also be used as the read address of the ROM. This

is the architecture used by the traditional CORDIC algorithm.

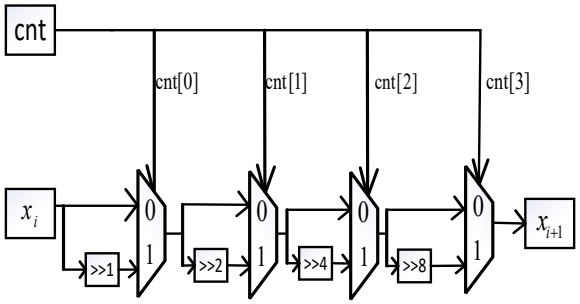


**Figure 6.** Dynamic shift circuit

The detailed block diagram of the parallel structure is shown in Figure 7.

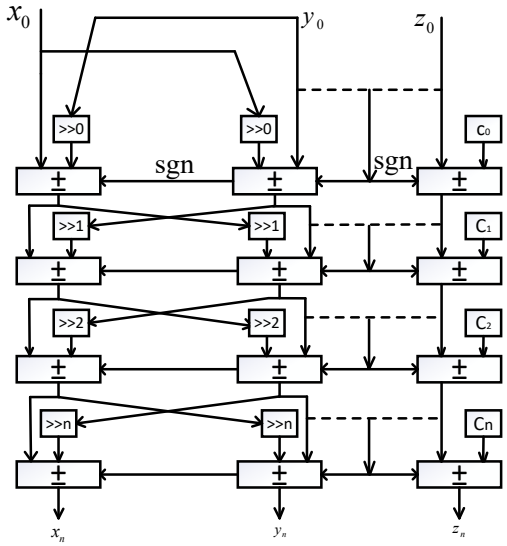


**Figure 7.** Detailed parallel structure of CORDIC algorithm

It can be seen from the above figures that in the implementation architecture of CORDIC, compared with the parallel architecture, the serial CORDIC processing unit uses synchronous time-sharing multiplexing, which leads to the minimum resources occupied by the serial structure. This makes the design of CORDIC control unit slightly more complex, the timing control is more cumbersome, and the system processing speed is lower. The parallel structure is an extension and improvement of the serial structure. This structure introduces an independent CORDIC processing unit for each iteration of the algorithm, so it does not need the addition of other control circuits, and the whole process only needs shift, addition and subtraction operations. This is the biggest difference from the CORDIC processing unit in the serial structure. In order to improve the processing speed of the system, this paper adds a pipeline register to the parallel structure, which effectively shortens the critical path and makes the length of the critical path change from N CORDIC processing units of the parallel structure to 1 CORDIC processing unit.

The block diagram of parallel pipeline structure is shown in Figure 8.

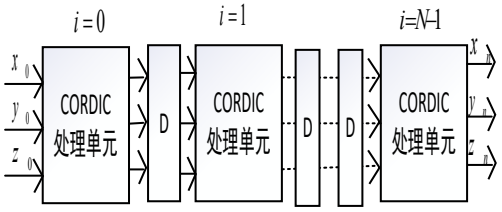


**Figure 8.** Parallel pipeline structure of CORDIC algorithm

# 4. Improved CORDIC algorithm

The implementation method and process of the traditional serial CORDIC algorithm is relatively simple, and because the angle of each rotation is fixed, the multi-level parallel pipeline structure is required to achieve high precision calculation. However, the traditional CORDIC parallel multistage pipeline structure uses a higher number of stages, resulting in a large overall circuit delay and increased hardware costs.

Based on the above conditions, this paper proposes an improved CORDIC parallel structure based on the traditional serial CORDIC and the traditional parallel pipelined CORDIC, which is composed of a 12-stage pipeline and a small-capacity sine and cosine lookup table.

## 4.1. Angle binary coding

In the research of traditional researchers and scholars on CORDIC, it is known from various research data that in the traditional implementation method of the algorithm, the resource consumption for calculating the remaining angle basically accounts for 1/3 of the total resource consumption. In the actual operation process of the algorithm, the direction of each rotation and the size of the iteration are inseparable from the last rotation, which has an inevitable impact on the overall operation speed of the system. According to the calculation relationship of trigonometric function, in order to avoid such influence $[0,\pi/4]$ the rotation angle of is expressed by N-bit binary number.

$$\theta = \sum_{n=1}^{N} a_n 2^{-n} \tag{12}$$

Among them, $a_n = \{0, 1\}$ it is the bit $\theta$ value of the input rotation angle expressed in binary. So far, the angle and direction of each rotation can be determined according to the size of this value. because $a_n$ it represents a binary bit value. It does not consume additional resources in the hardware implementation process, thus eliminating the internal consumption of resources generated when calculating the remaining angle, and improving the overall running speed of the system.

## 4.2. Establishment of small lookup table

In this paper, 13-bit binary number is used $[0,\pi/4]$ the input angle in the, and $[\pi/4, 2\pi]$ the sine and cosine function values of the internal input angle can be obtained through the transformation of the trigonometric function formula, so only input is required $[0,\pi/4]$ the angle value in is enough. Some lookup tables are shown in Table 1 below.

**Table 1.** Lookup Table

| address | Binary cosine value | Binary sine value |
|---|---|---|
| 0000 | 010000000000000 | 000000000000000 |
| 0001 | 001111111110000 | 000001000000000 |
| 0010 | 001111111000000 | 000001111111101 |
| ..... | ..... | ..... |
| one thousand one hundred and one | 00101100000010 | 001011100111011 |

## 5. Simulation of Improved CORDIC Parallel Architecture

According to the traditional CORDIC parallel structure, the traditional CORDIC pipelined parallel structure and the improved CORDIC pipelined structure proposed in this paper, the Verilog digital code is written on Quartus, and the simulation is performed on ModelSim. The simulation results are as follows.

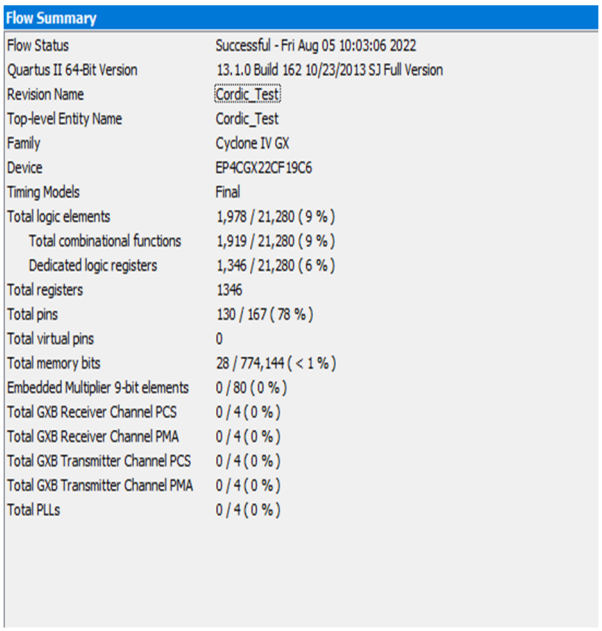


**Figure 9.** CORDIC logic resource consumption of traditional parallel architecture

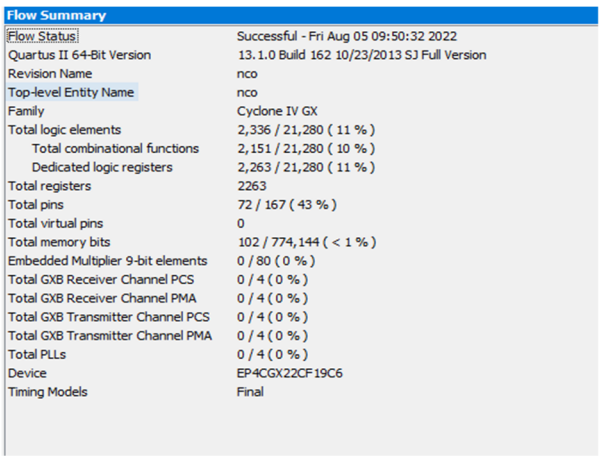


**Figure 10.** CORDIC logic resource consumption of traditional parallel pipeline structure

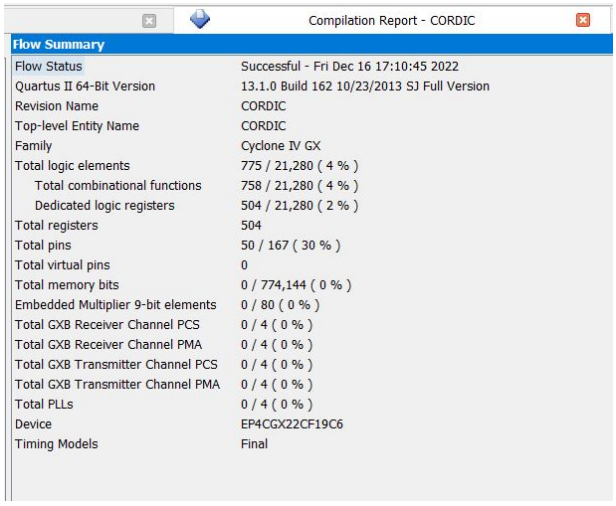


**Figure 11.** Improved parallel pipeline CORDIC logic resource consumption

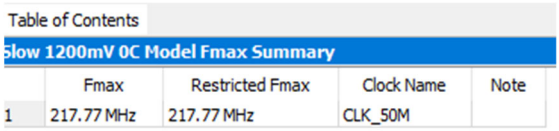


**Figure 12.** Maximum clock speed of traditional parallel architecture

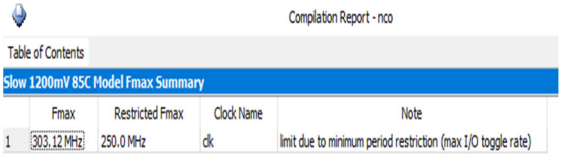


**Figure 13.** Maximum clock speed of traditional parallel pipeline structure

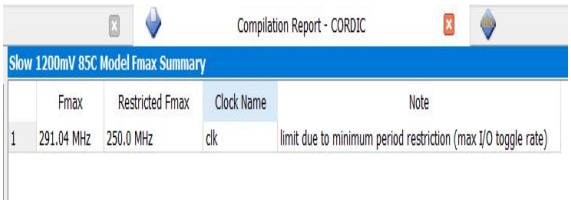


**Figure 14.** Maximum clock speed of improved parallel pipelining structure

Figure 9, Figure 10 and Figure 11 show that the traditional parallel pipeline architecture uses more registers than the traditional parallel architecture (pipeline architecture 2263, non-pipeline architecture 1346), while the improved parallel pipeline CORDIC architecture proposed in this paper uses 504 registers, which significantly reduces the built-in resource consumption compared with the previous two traditional CORDIC architectures. Figure 12, Figure 13 and Figure 14 measure the clock speed of the system. The results show that the running speed of the traditional parallel structure CORDIC algorithm is 217.77 Mhz, and the running speed of the traditional parallel pipeline structure is 303.12 Mhz, while the running speed of the improved parallel pipeline structure algorithm proposed in this paper is 291.04 Mhz. Overall, the traditional pipeline structure can greatly improve the system running speed, but the use of more registers leads to increased resource consumption, which affects the overall performance. The improved pipeline structure proposed in this paper uses the least registers,

reduces the internal resource consumption, and the overall speed is only less than 3.9% lower than the traditional parallel pipeline, and also ensures the high speed of the system.

The following figure shows the simulation results using ModelSim.

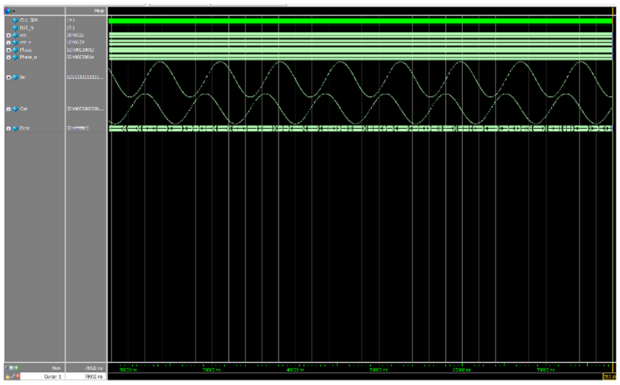

**Figure 15.** ModelSim simulation results

# 6. Conclusion

In view of the problems of the slow operation speed of the traditional serial CORDIC algorithm and the large resource consumption of the traditional parallel CORDIC algorithm, this paper proposes an improved CORDIC parallel structure based on the traditional serial CORDIC and the traditional parallel pipelined CORDIC, which is composed of a 12-stage pipeline and a small-capacity sine and cosine lookup table. Finally, the parallel architecture, parallel pipeline architecture and improved CORDIC architecture are successfully implemented on Quartus, and the resource consumption and running speed of the three are compared. The comprehensive conclusion is that the structure proposed in this paper not only reduces the resource consumption, but also improves the running speed. Finally, the required sine wave is obtained on ModelSim.

# References


[1] Wang Zixuan, Wang Xin, Cai Zhikuang, etc A low-voltage digital controlled oscillator based on multistage capacitor attenuation technology [J] Research and Progress in Solid State Electronics, 2019, 39 (5): 7.

[2] Tang Wenming, Liu Guixiong FPGA fixed-point technology of exponential function CORDIC algorithm [J] Journal of South China University of Technology: Natural Science Edition, 2016, 44 (7): 6.

[3] Jack E. Volder. The CORDIC Trigonometric Computing Technique[J]. IRE Transactions on Electronic Computers, 1959, EC-8(3): 330-334.

[4] Yong-sen Wang, Yu-qun Xue, Heng You, Shu-shan Qiao. Area-energy efficient CORDICs using new elementary-angle-set and base-2 exponent expansions scheme: LETTER[J]. IEICE Electronics Express,2021,18(2).

[5] Supriya Aggarwal, Pramod Kumar Meher, Kavita Khare. Concept, Design, and Implementation of Reconfigurable CORDIC[J]. IEEE Trans. VLSI Syst, 2016, 24(4).

[6] Mahdavi Hossein, Timarchi S. Improving Architectures of Binary Signed-Digit CORDIC With Generic/Specific Initial Angles[J]. IEEE Transactions on Circuits and Systems I: Regular Papers,2020,67(7).

[7] Mahdavi H, Timarchi S. Area–Time–Power Efficient FFT Architectures Based on Binary-Signed-Digit CORDIC[J]. Circuits and Systems I: Regular Papers, IEEE Transactions on, 2019, 66(10): 3874-3881.

[8] Maharatna K , Banerjee S , Grass E , et al. Modified virtually scaling-free adaptive CORDIC rotator algorithm and architecture[J]. IEEE Transactions on Circuits & Systems for Video Technology, 2005, 15(11): 1463-1474.

[9] Huang Jun CORDIC pipeline design of sine and cosine wave generator based on FPGA [J] Mechanical Engineering and Automation, 2013, (05): 17-19.

[10] Qi Yanjie, Liu Zhangfa FPGA implementation of high precision and high speed direct digital frequency synthesizer based on Parallel-CORDIC [J] Journal of Electronics, 2014, 42 (7): 1392-1397.

[11] Xu Cheng, Qin Yunchuan, Li Kenli, et al The two-step rotation CORDIC algorithm without scaling factor [J] Journal of Electronics, 2014, 42 (7): 1441-1445.