

Design and Implementation of Improved NCO based on FPGA

Yanshuang Chen ^a, Jun Yang ^{b, *}

School of Information Science and Engineering Yunnan University, Yun'nanKunming650091,
China

^a yschenat@163.com, ^{b, *} junyang@ynu.edu.cn

Abstract. A numerically controlled oscillator (NCO) is used to generate quadrature controllable sine and cosine waves and is an important part of software radio. The traditional NCO module is implemented based on the lookup table structure, which requires a large amount of hardware storage resources inside the FPGA. Therefore, the CORDIC algorithm is used to implement the NCO module, and the output accuracy is improved by improving the CORDIC algorithm. At the same time, the FPGA technology is characterized by strong reconfigurability, good scalability, low hardware resources. The module is designed with Verilog HDL language. Finally, the NCO model based on FPGA design has the characteristics of low hardware resource consumption and high output precision. The model was simulated by Modelsim and downloaded to the target chip verification of Altera DE2's EP2C35F672C6. The digitally controlled oscillator met the design requirements.

Keywords: Digital controlled oscillator (NCO); CORDIC algorithm; Pipeline structure; FPGA; precision.

1. Introduction

The numerically controlled oscillator (NCO) is an important component of the signal processing system [1], and is mainly used to generate orthogonally controllable sine and cosine waves. With the continuous development of modern communication systems, digital control oscillators have been widely used in digital communication, signal processing and other fields as an important part of software radio [2].

The traditional NCO implementation method is based on read-only memory table lookup (ROM LUT), which requires complex multiplication operations and therefore consumes a large amount of storage resources. In this paper, the CORDIC algorithm is improved, and the algorithm pipeline structure is adopted. Each level of iteration uses independent arithmetic units and increases the calculation data bits, which improves the output precision of the algorithm while reducing the resource occupation. FPGA technology itself has the characteristics of strong reconfigurability, good scalability, and low hardware resources [3], Therefore, this paper uses FPGA technology to build NCO model, uses Verilog HDL language to design and implement the module, and finally based on FPGA design. NCO has the characteristics of less hardware resources and high output accuracy.

2. The structure and Principle of Digitally Controlled Oscillator

2.1 NCO Structure and Principle based on Table Lookup Method

The structure of the digitally controlled oscillator is shown in Figure 1. The traditional NCO is implemented by the look-up table method (LUT), that is, the sine and cosine values of the corresponding phase are calculated according to the phase of the sine and cosine waves, and the phase angle is used as the sine and cosine of the phase. The address is used to construct the phase amplitude conversion circuit, and finally the sample value of the sine and cosine signal is obtained by looking up the table [4]. The look-up table method requires a large amount of storage resources.

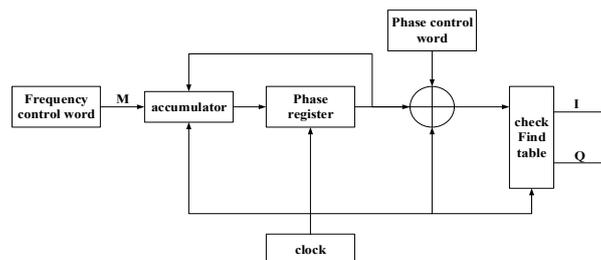


Figure 1. NCO structure diagram based on table lookup method

2.2 Principle of CORDIC Algorithm

Coordinate Rotation Digital Computer (CORDIC) was first proposed by D.J. Volder in 1959. The core idea of the CORDIC algorithm is to approximate the desired angle by continuously rotating a series of fixed, small angles associated with the calculation base [5].

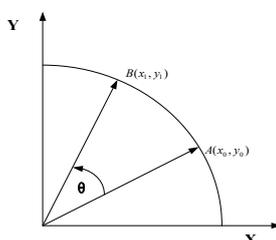


Figure 2. Schematic diagram of CORDIC vector rotation

As shown in Fig. 2, the vector $A(x_0, y_0)$ in the xy coordinate system is rotated counterclockwise by θ angle around the origin to obtain the vector $B(x_1, y_1)$, and the θ angle between the vector A and the vector B is divided into N θ_i , can get a single rotation expression. Let d_i be the direction of rotation, $d_i = 1$ when rotating clockwise, $d_i = -1$ when counterclockwise, get $\cos\theta_i$, let $\theta_i = \arctan 2^{-i}$, select the transformation method according to the coordinates:

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \cos \theta_i \begin{bmatrix} 1 & -d_i 2^{-i} \\ d_i 2^{-i} & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad (1)$$

Since the sum of N θ_i is equal to θ after N rotations, $\theta = \sum_{i=0}^{N-1} d_i \theta_i$. The above formula can be obtained:

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \left(\prod_{i=0}^{N-1} \cos \theta_i \begin{bmatrix} 1 & -d_i \tan \theta_i \\ d_i \tan \theta_i & 1 \end{bmatrix} \right) \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = K_N \left(\prod_{i=0}^{N-1} \begin{bmatrix} 1 & -d_i \tan \theta_i \\ d_i \tan \theta_i & 1 \end{bmatrix} \right) \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \quad (2)$$

Where K_N is the modulus expansion factor, $K_N = \prod_{i=0}^{N-1} \cos \theta_i = \prod_{i=0}^{N-1} 1/\sqrt{1+2^{-2i}}$. When $N \rightarrow \infty$, K_N converges to a constant, that is $K_N = 0.607252$. It can be seen that when the number of iterations N is determined, the value of K_N is a constant. You can simplify (1) by removing $\cos\theta_i$ during the rotation:

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} 1 & -d_i 2^{-i} \\ d_i 2^{-i} & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad (3)$$

Finally, the residual angle value formula is introduced, $z_{i+1} = z_i - d_i \theta_i$ z_0 is θ , and z_{i+1} is the angle residual value. As i increases, the z_{i+1} angle residual value approaches zero. When the initial

value $(x_0, y_0, z_0) = (K_N, 0, \theta)$, (x_n, y_n) converges to $(\sin\theta, \cos\theta)$. The CORDIC algorithm performs the calculation of the sine and cosine in the above manner.

3. Cordic Algorithm Improvement

3.1 Algorithm Pipeline Structure

The traditional CORDIC algorithm structure uses the same set of hardware to iterate repeatedly, and needs to continuously feed back the output data to the input end, which undoubtedly affects the running speed of the whole system, and the throughput is small [6]. This paper uses a pipeline structure to achieve, each level of CORDIC iterative operation uses a separate arithmetic unit to improve data throughput.

In the CORDIC algorithm pipeline structure, there is a certain angular error δ between the rotational accumulation angle and the actual angle[7], $\theta = \sum_{i=0}^N d_i \theta_i + \delta$. It can be seen from the above equation that as the number of iterations increases, the accumulated angle is infinitely close to θ , the angular error δ becomes infinitely small, and the accuracy of the CORDIC algorithm is thus maximized. Combined with the actual situation and the design requirements of the NCO, this paper adopts an 8-stage pipeline structure, and each stage of the iteration uses an independent arithmetic unit to achieve high-speed and high-precision output. The pipeline hardware structure is shown in Figure 3.

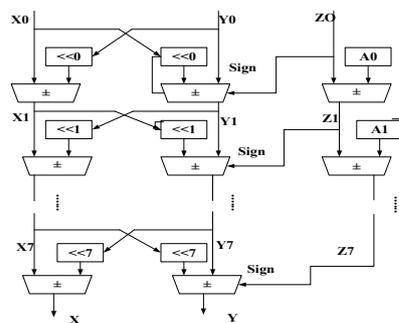


Figure 3. CoRDIC pipeline hardware structure

3.2 Coverage of the Angle in the Circle

According to the principle of CORDIC algorithm, the rotation angle $\theta = \sum_{i=0}^{N-1} d_i \arctan 2^{-i}$, the rotation range is $[-99.88^\circ, 99.88^\circ]$, and cannot cover the entire circumference period. In this paper, 24-bit data bits are used to represent the rotation angle. When the rotation angle is not in the first quadrant, it is mapped to the first quadrant by cutting off the highest two digits and doing the complement 0 processing in front, and then correspondingly processing the angle values. The iterative operation, the obtained value is then based on the symmetry output corresponding to the quadrant restored amplitude value. The angle mapping relationship is shown in Table 1.

Table 1. CORDIC algorithm angle mapping table

| The highest two angles | Quadrant | Mapped angle value θ | Sinusoidal output value | Cosine output value |
|------------------------|----------|-----------------------------|-------------------------|---------------------|
| 00 | I | θ | $\cos\theta$ | $\sin\theta$ |
| 01 | II | $\theta - 90^\circ$ | $-\sin\theta$ | $\cos\theta$ |
| 10 | III | $\theta - 180^\circ$ | $-\cos\theta$ | $-\sin\theta$ |
| 11 | IV | $\theta - 270^\circ$ | $\sin\theta$ | $-\cos\theta$ |

3.3 Increase the Calculated Data Bit

the CORDIC algorithm uses finite precision algebra to do the calculation, so another error occurs. The error caused by the limited data bit width is called the rounding error ϵ . According to the formula $\epsilon = 2^{-b-1}$, the operation data bit width b affects the rounding error ϵ , and the rounding error

is reduced to half of the original by the bit width of the operand. In the internal iteration of CORDIC designed in this paper, by adding 6 data bits, the data bit width is extended to 14 bits, which effectively improves the operation precision. The data bit increase diagram is shown in Figure 4.

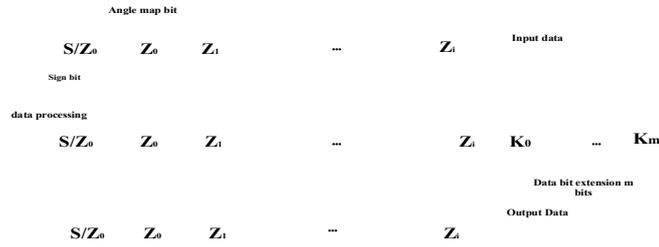


Figure 4. Shows the calculation of the increase in data bits

4. Design and Implementation of Improved Nco based on Fpga

The improved NCO top-level structure diagram based on FPGA is shown in Figure 5. The structure consists of three parts: pre-processing, CORDIC iteration and truncated output processing.

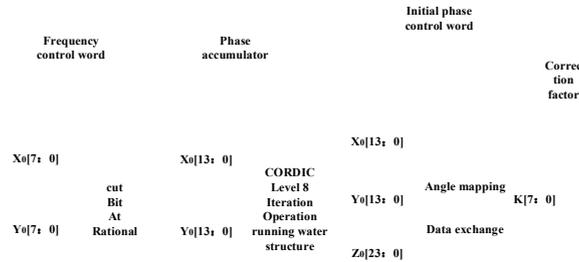


Figure 5. Is based on the improved CORDIC algorithm NCO top-level structure

In the pre-processing, the initial value of the phase accumulator is 0. Under the control of the clock pulse, the frequency control word and the phase accumulator are accumulated, and the obtained phase value is added to the phase control word to obtain the current phase value. Let the frequency control word be K , then the frequency of the output signal is $f_{out} = K \times f_c / 2^m$, where f_c is the input frequency and m is the phase accumulator bit width, then the minimum resolution of the frequency is the output frequency $f = f_c / 2^n$. Since the number of pipeline stages has been determined, simply push the number of pipeline stages into the formula to find K_N , and then determine the correction factor.

The CORDIC iteration part is an 8-stage pipeline structure. Each level of the iterative structure contains two shift registers and three adders and subtractors. Each level structure is equivalent to one iteration of the CORDIC algorithm. The hardware structure is shown in Figure 3.

The truncated output processing part is a rounding truncation of the data, and outputs two sinusoidal and cosine waveform signals.

The design of the NCO module based on FPGA is shown in Figure 6, where clk is the clock pulse signal, the active level is high, rst_n is the reset signal, $phase_in$ is the input of phase data, and sin_out and cos_out are the sine and cosine signals of the output respectively.

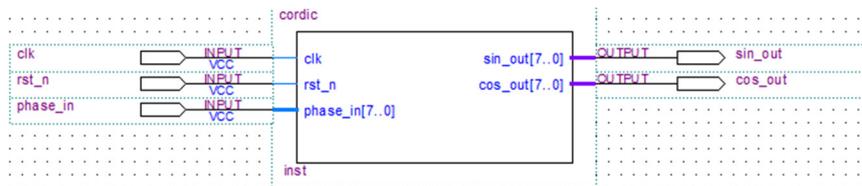


Figure 6. Nco module diagram

5. System Simulation and Results Analysis

This design simulates the NCO model in Quartus II 13.0 test software, simulates the model with Modelsim, and downloads it to the development board with Altera DE2 EP2C35F672C6 as the target chip, which verifies the correctness of the design method. And feasibility, the simulation results are shown in Figure 7.

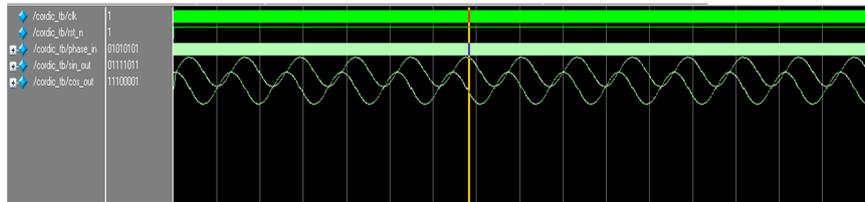


Figure 7. Is a simulation diagram of NCO results based on improved CORDIC algorithm design.

It can be seen from the above figure that the sinusoidal waveform signal generated by the improved NCO has orthogonal characteristics. By mapping the angle to achieve coverage of the entire circumferential interval, it can replace the traditional NCO design method based on the lookup table structure. After compiling and synthesizing the code through Quartus II 13.0 software, the hardware resource information such as the logic unit consumed by the design is obtained:

Table 2. Comparison of lookup table structure and CORDIC algorithm hardware resource consumption

| Method to realize | Data width | Pipeline series | Logical unit used | Resource saving rate |
|-------------------|------------|-----------------|-------------------|----------------------|
| Lookup table | 8 | ~ | 275 | ~ |
| CORDIC | 8 | 8 | 240 | 13.75% |

In Table 3, in the 8-bit data bit and 8-stage pipeline iteration, the NCO based on the lookup table structure uses 275 logic cells, while the NCO based on the CORDIC algorithm uses only 240 logic cells, saving 13.75%. Hardware resources.

According to the data analysis of Table 4, in the case of 8-bit data bits, the operation error of the traditional CORDIC algorithm can reach 10^{-4} , and the improved CORDIC algorithm can improve the accuracy to $10^{-5} - 10^{-6}$, so this design improves the computational accuracy while reducing hardware resource consumption.

Table 3. Comparison of Errors between Traditional CORDIC Algorithm and Improved CORDIC Algorithm

| achieve the way | | 0° | 10° | 25° | 45° |
|------------------------------|-----|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|
| Traditional CORDIC algorithm | sin | 0.000231 2.42×10^{-4} | 0.172485 2.34×10^{-4} | 0.413586 2.25×10^{-4} | 0.707892 1.76×10^{-4} |
| | cos | 1 0 | 0.910031 0.43×10^{-4} | 0.903261 1.06×10^{-4} | 0.708152 1.71×10^{-4} |
| Improve CORDIC algorithm | sin | 0.000030 3.07×10^{-5} | 0.161650 2.16×10^{-5} | 0.399557 2.89×10^{-5} | 0.709113 7.01×10^{-6} |
| | cos | 1 0 | 0.970801 5.92×10^{-5} | 0.809308 3.87×10^{-6} | 0.709569 2.38×10^{-5} |

6. Conclusion

In this paper, based on FPGA, the NCO module in digital down conversion is improved and designed. The CORDIC algorithm is optimized by using pipeline structure, circumferential angle

coverage and increasing data bits. The design is verified by ModelSim. The experimental results show that the design improves the computational accuracy of the system while reducing the hardware resource consumption.

References

- [1]. Wang Shenbiao, Hu Chunlin, Hu Guangyu, et al. NCO design based on improved CORDIC algorithm [J]. *Electronic Technology Application*, 2017, 43(3): 43-57.
- [2]. Song Hongyan. Research and implementation of digital down conversion based on FPGA [D]. Shandong University, 2016.
- [3]. Wang Wei. Design and implementation of multi-stage CIC filter based on FPGA, *Journal of Yunnan University*, 2018, 40(4): 676-681.
- [4]. CaiQuan, Gao Bo, Gong Min. Implementation of CORDIC algorithm based on FPGA[J]. *Electronic Devices*, 2018, 41(05): 1242-1246.
- [5]. Zhu Kui. A Digital Frequency Conversion Design Based on FPGA [D]. Nanjing University of Posts and Telecommunications, 2017.
- [6]. Peng Qi, Xi Youbao. Design and Implementation of NCO Based on FPGA and Improved CORDIC Algorithm[J]. *Microcomputer & Applications*, 2013, 32(05): 60-62.
- [7]. Zhou Yun. Research and FPGA Implementation Based on Software Radio Digital Down Conversion Technology [D]. Southwest Jiaotong University, 2016.