Design of DDS based on Hybird-CORDIC Architecture

Xiujie Qu, He Chen*, Yubin Zhang
School of Information and Electronic Engineering, Beijing Institute of Technology
Beijing 100081, China

Yingtao Ding
Department of Electrical and Computer Engineering, University of Florida
Gainesville, FL 32611, USA

Received: 23-02-2011
Accepted: 18-04-2011

Abstract

Communication technology and its signal processing are basic supportive factors of intelligent transportation system. Because of rapid progress in modern communication engineering, direct digital synthesizer (DDS) is widely used in communications system. However, traditional DDS technology is encountering challenges due to its ROM's capacity limit. The paper presents a new direct digital frequency synthesizing method based on Hybird-CORDIC (coordinated rotation digital computing) algorithm, which can calculate sin/cos value directly, improve performance, reduce size, and reduce design complexity. Accordingly, the result of field-programmable gate array (FPGA) realized with Verilog HDL language is in accord with expected requirement approvingly.

Keywords: direct digital synthesizer (DDS), CORDIC (coordinated rotation digital computing), Hybird-CORDIC DDS, rotation, phase accumulator, shift register.

1. Introduction

The rapid expansion of city in China results in traffic jam heavily. Infrastructure cannot meet the demand of city transportation, thus intelligent transportation system (ITS) is becoming a promising approach to the challenge of city expansion[1]. ITS integrates communication, information, computing, sensor, automation and control, and signal processing technology effectively to offer a real time, accurate, intelligent transportation managing system. The Direct digital synthesizer (DDS), a new technique of frequency synthesis distinctive of fast frequency transfer, high resolution, and continuous phrase, has been widely used in communication systems and electronic equipments. Frequency synthesizer is both actuating signal source of transmitter and local oscillator of receiver in communication, radar and navigation systems. It is regarded as the heart of electronic system and can be applied in ITS to improve traffic considerably.

Traditionally in DDS, ROM Look-up Table (ROM LUT) structure is applied in phrase-width transformation. The limit of ROM capacity results generally in higher cost and power consumption, and lower stability when improving DDS's performance by means of increasing ROM capacity[2,4]. General Cordic algorithm

* Corresponding author: chenhe@bit.edu.cn

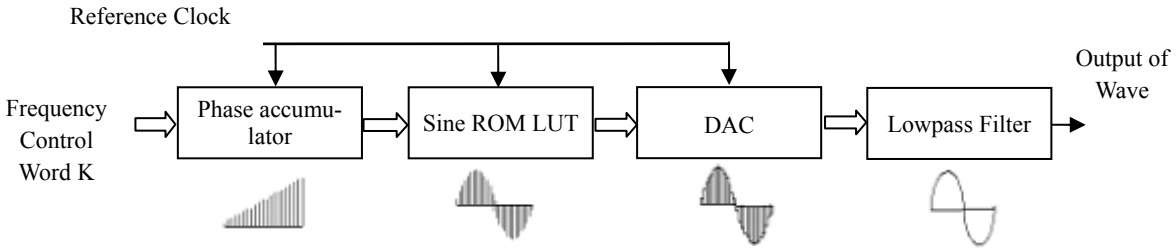

Fig. 1 Structural configuration of Basic DDS

makes circuit's structure complicated due to multiple shift operation. A Hybird-CORDIC (coordinated rotation digital computing) algorithm, refrained from ROM LUT, is outlined to simplify circuit's structure considerably compared with general Cordic algorithm.

2. Configuration of DDS

The direct digital synthesizer (DDS) is shown in Fig.1, which is composed of phase accumulator, sine ROM LUT, DAC, and lowpass filter. Controlled by reference clock K, frequency phase register accumulates frequency control word and results in addressing wave memorizer, thus a width code output is transferred to a step wave in DAC. At last, a demanded wave is achieved by lowpass filter.

The relationship between frequency of output, reference clock frequency, and frequency control word is formulated as:

$$f_{out} = Kf_{clk} / 2^N \quad (1)$$

Where f_{out} stands for frequency of DDS output, K is frequency control word and is also called phase increment, f_{clk} stands for reference clock frequency, N is the digits of phase accumulator. On the premise of confirmed reference clock frequency and the digits of phase accumulator, change of K may result in different-frequency output. When K=1, we obtain the least frequency of DDS output, defined as the frequency accuracy of DDS system as follow:

$$\Delta f = f_{clk} / 2^N \quad (2)$$

3. CORDIC Algorithm

The Coordinate Rotation Digital Computer algorithm

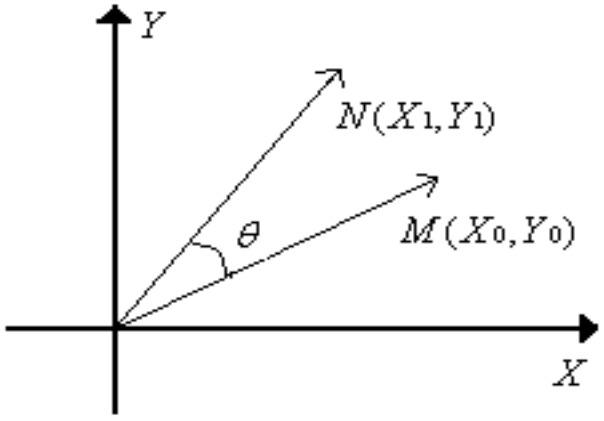

Fig. 2 the triangle spinning of CORDIC

(CORDIC) was developed by J. Volder[3] in the late 1950s, which can be used to compute trigonometric functions. The algorithm uses vector rotation to compute the sine, cosine, tangent, arcsine, arccosine, and arctangent functions. The CORDIC algorithm is an iterative method to calculate the coordinate of a vector rotation or to carry out radius and the phase of a vector. The CORDIC algorithm can be operated in either a vector "rotation" mode or an angle accumulation mode ("vectoring").[5] This paper uses the "rotation" mode, as show in Fig. 2.

In Fig.2, $M(X_0, Y_0)$ is a point in the plane right-angled coordinate system of XOY, vector $\overline{OM}$ will overlap with the vector $\overline{ON}$ after rotating the angle θ. The relationship between vector $\overline{ON}$ and vector $\overline{OM}$ can be expressed as[5,6]:

$$\begin{cases} X_1 = X_0 \cos\theta - Y_0 \sin\theta \\ Y_1 = X_0 \cos\theta + Y_0 \sin\theta \end{cases} \tag{3}$$

Where x and y are original coordinates before rotation, X_1 and Y_1 are coordinates after rotation, θ is the rotation angle. This equation can be simplified by assuming that the tangent is a power of 2, namely $\theta = \arctan 2^{-i}$, the equation (3) can be transformed into:

$$\begin{cases} X_1 = (X_0 - \sigma \cdot Y_0 \cdot 2^{-i}) \cos\theta \\ Y_1 = (X_0 + \sigma \cdot Y_0 \cdot 2^{-i}) \cos\theta \end{cases} \tag{4}$$

Where, σ refers to the rotation direction, $\sigma = 1, -1$ responding to counterclockwise rotation and clockwise respectively. The term $\cos\theta$ can be calculated in advance, which can be ignored during the iteration. Then any angle of rotation can be obtained by performing successive smaller rotations. This assumption helps us to write equation (4) in the form of an iterative operation:

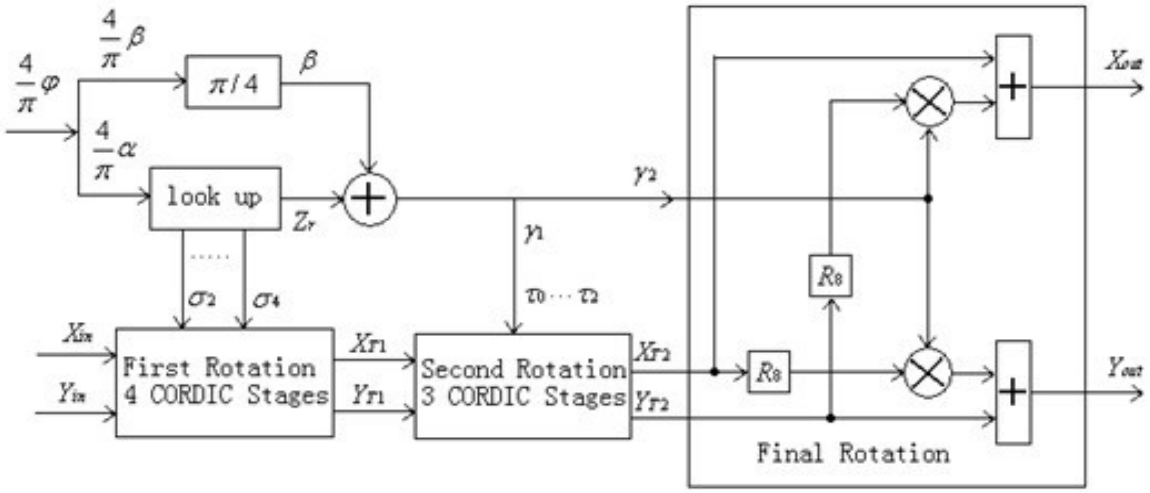

Fig. 3 the architecture of Hybird-CORDIC rotator

$$\begin{cases} X_1 = X_0 - \sigma \cdot Y_0 \cdot 2^{-i} \\ Y_1 = X_0 + \sigma \cdot Y_0 \cdot 2^{-i} \\ Z_1 = Z_0 - \sigma \cdot \arctan 2^{-i} \end{cases} \quad (5)$$

Where, the third equation is the angle accumulator. The value of sin/cos can be attained after n rotations.

4. Hybird-CORDIC Algorithm

The rotation of the Hybird-CORDIC algorithm is divided into three steps to fulfill. The rotation of the first two steps use regular CORDIC iterative algorithm to achieve, the difference of them is that the first step attain the rotating direction using a lookup table, the second step attain the rotating direction by calculating directly. The last rotation is composed of multiplier unit and adder unit, which can reduce the line series of whole design and reduce the complexity of the whole circuit.[7] The circuit architecture diagram is shown in Fig.3.

In design, we use the phase accumulators with 32 bits and calculate the phase-amplitude after intercepted 16 bits. Because the value of the positive/cosine symmetry in cycle is symmetrical, we can directly take $\varphi \in [0, \pi/4]$ to calculate, the value of the rest interval can be directly obtained according to transforming the top three values, which can be formulated in binary form as follow:

$$\frac{4}{\pi}\varphi = f_1 \cdot 2^{-1} + \cdots + f_{13} \cdot 2^{-13} \quad (6)$$

First Rotation

In the first step, the angle of φ is divided into two parts, $\varphi = \alpha + \beta$.

$$\alpha = (f_1 \cdot 2^{-1} + \cdots + f_{-m+1} \cdot 2^{-m+1} + 2^{-m}) \cdot \frac{\pi}{4} \quad (7)$$

$$\beta = (-\overline{f_m} \cdot 2^{-m} + \cdots + f_{13} \cdot 2^{-13}) \cdot \frac{\pi}{4} \quad (8)$$

Where, $\overline{f_m}$ is the complement of f_m. In order to ensure error within permitted region, the value of m must be satisfied with the relationship below:

$$2^{-m} - \arctan 2^{-m} < 2^{-B} \quad (9)$$

From above, we can obtain m=4. In the first rotation, it mainly uses the CORDIC Algorithm in equation (5) to calculate the phase α. The algorithm starts with $X_1 = -1$, $Y_1 = 0$, $Z_1 = a$ and $\sigma_1 = +1$. Through four CORDIC sub-rotations, the first rotation includes a phase error Z_r by comparison of equation (5) and equation (7). The phase error can be carried out again by an adder passed to the second rotation, shown as Fig.3.

Second Rotation

In order to complete the operation, the second and third stages of the Hybrid CORDIC architecture rotate the vector [X_{T1}, Y_{T1}] (the output of the first stage) by an angle:

$$\gamma = Z_r + \beta \quad (10)$$

The angle is computed by using the multiplier and the adder shown in Fig. 3. The multiplier is needed to calculate from its scaled representation, see equation (5). Since, as we have observed before, the absolute values of Zr and β are both lower than 2^{-4}, the absolute value of γ is lower than 2^{-3}. By representing with 11 bits, we have

$$\gamma = 2^{-3}(-g_0 + g_1 2^{-1} + \cdots + g_{10} \cdot 2^{-10}) \quad (11)$$

Similar to equation (7) and (8), the angle γ is then split as the sum of two sub-angles $\gamma = \gamma_1 + \gamma_2$, where

$$\gamma_1 = 2^{-3}(-g_0 + g_1 \cdot 2^{-1} + g_2 \cdot 2^{-2} + 2^{-3}) \quad (12)$$

$$\gamma_2 = 2^{-3}(-\overline{g_3} \cdot 2^{-3} + g_4 \cdot 2^{-4} \cdots + g_{10} \cdot 2^{-10}) \quad (13)$$

The second rotation block is aimed to perform the rotation by the angle, whereas the rotation by the angle is assigned to the final rotation block. In the second rotation we employ a CORDIC algorithm without the Zi computation. The rotation directions τ_i are obtained directly by the bits of γ_1 as follow:

$$\tau_0 = 2\overline{g_0} - 1, \tau_i = 2g_i - 1 \quad i = 1,2 \quad (14)$$

The corresponding CORDIC equations are similar to the first rotation, we can use the equation (5) to calculate. In fact, when *i* is not large enough, there exists the relationship: $\arctan 2^{-i} \neq 2^{-i}$. As a consequence, the second rotation block introduces a phase error:

$$\varphi_{err} = \gamma'_1 - \gamma_1 < 0.77 \cdot 2^{-13} \quad (15)$$

The phase error of the second rotation introduces an error on each component of the DDFSM output. From equation (15), it is much lower than the weight of the output LSB (2^{-11}).

Final (Third) Rotation

The final rotation block in Fig. 3 implements the rotation by γ2. This final rotation could also be worked out by using the CORDIC algorithm. However, as observed in[8, 9], when the rotation angle is small a complex multiplier is able to reduce the latency and improve the performances.

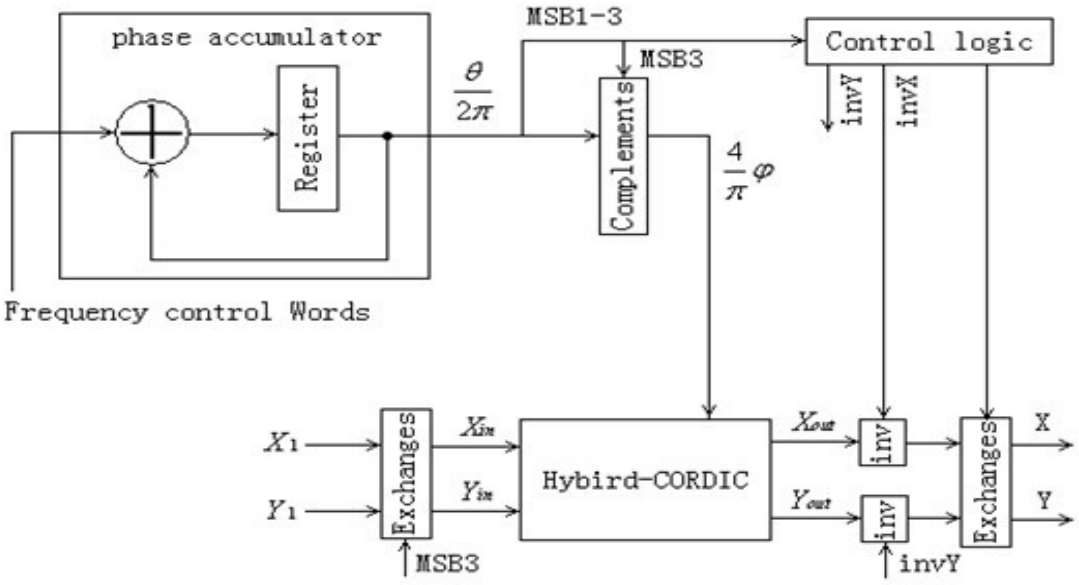

Fig. 4 the DDS structure based on Hybird-CORDIC algorithm

Table 1 comparison with recently proposed designs

Method	Accumulators （bits）	SFDR （DBc）	Output （bits)	Number of Slices	Number of Slice Flip Flops	Power Consumption （mW）
CORDIC	32	100	13	819 /5472	652 /10944	340
Hybird -CORDIC	32	100	13	280/5472	394 /10944	280.56

In this case, the absolute value of phase is lower than 2^{-6}. Therefore, we can approximate sine and cosine functions in the CORDIC algorithm as:

$$\sin \gamma_2 \approx \gamma_2, \cos \gamma_2 \approx 1 \tag{16}$$

In this way, the final rotation is realized without the need of lookup tables to store sine and cosine values.

An analytical derivation of the joint effect of all algorithmic and quantization errors are not easy. We performed bit-level simulations, by simulation of two methods. Table 1 shows the performances of the CORDIC algorithm and Hybird -CORDIC algorithm respectively.

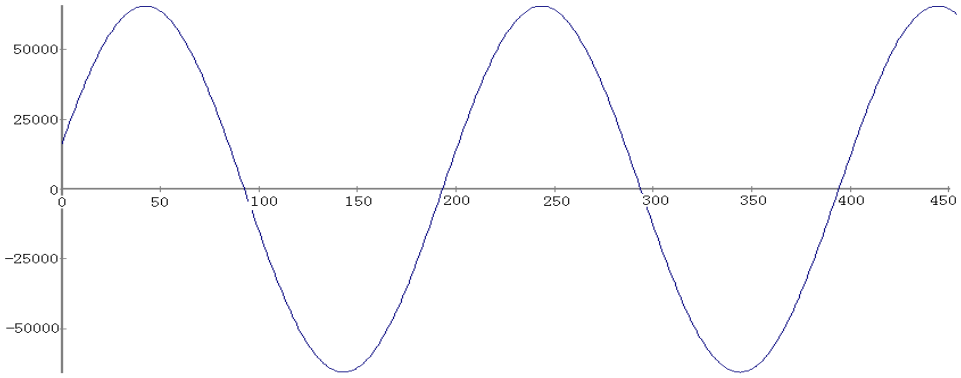

Fig. 5 result of new FPGA structure

As shown in Fig. 3, we have introduced two R8 modules in the final rotation stage for the sake of only intercepting 8 bits from X_{T2} and Y_{T2} into calculating, to reduce the complexity of multipliers input. The two modules introduce an additional error ε_2 at the output. We have

$$\left|\varepsilon_2\right| = \left|r_2\right| \cdot 2^{-7} \le 2^{-13} \tag{17}$$

5. DDS based on FPGA

In our design, in order to improve the frequency resolution ratio, we adopt the phase accumulator with 32 bit, intercepting16 bit into calculating and using the 13 binary to represent the consequence trigonometric function value. We use the Virtex4 to fulfill the design, which is produced by the Xlinx company, shown as Fig.4.

6. Experiment Result

A Field-programmable Gate Array (FPGA) is an integrated circuit designed to be configured by the customer or designer after manufacturing, hence field programmable. The FPGA configuration is generally specified using a hardware description language (HDL), similar to that used for an application-specific integrated circuit (ASIC) (circuit diagrams were previously used to specify the configuration, as they were for ASICs, but this is increasingly rare). FPGAs can be used to implement any logical function that an ASIC could perform. Hence, we use FPGA to testify our new DDS design, and download the DDS design into FPGA after layout wiring, and then Chipscope will show the wave in fig.5.

Through Matlab simulation on our design for stray case, the results show that the stray inhibit is more than 100dB, as shown in Fig.6.

In the same FPGA device and design software cases, comparing our design with DDS based on the general CORDIC algorithm, the results show that our structure greatly reduces the complexity and power consumption of the circuit

7. Conclusion

This paper discusses the DDS structure based on Hybird-CORDIC algorithm in detail. This new structure divides the phase of $\pi/4$ into three parts, and uses different mathematics methods responding to each part. The experiment shows that our design provides a great improvement in complexity, performance and area.

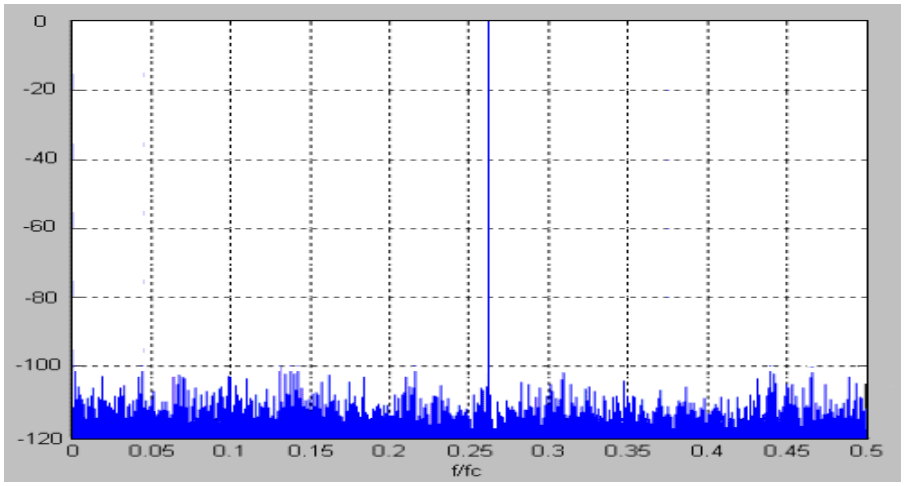

Fig. 6 simulating result

Consequently, the result of field-programmable gate array (FPGA) realized with Verilog HDL language is in accordance with expected requirement approvingly.

References

1. W H Wang, Fuguo Hou, Huachun Tan, H Bubb, A Framework for Function Allocations in Intelligent Driver Interface Design for Comfort and Safety, *International Journal of Computational Intelligence Systems,* 5(2010), 531-541.
2. Yuan Ruan, Jianhua Xie, Feng Sun, Design of pipelining DDS based on CORDIC algorithm, *Electronic Design*, 24(2) (2008) 286-287.

3. Pengwei Lu, Shaochun Mei, Implementation of trigonometric function based CORDIC, *Theory and method*, 27(1) (2008) 10-11.
4. Hui Li, Ming LU, CORDIC- Based DirectDigital Frequency Synthesizer, *Experiment Science& Technology*, 2 (2006) 115-118.
5. U. Meyer-Bäse, A. Meyer-Bäse and W. Hilberg, Coordinate Rotation DIgital Computer (CORDIC) synthesis for FPGA, *Field-Programmable Logic Architectures, Synthesis and Applications*. 849 (1994) 397-408.
6. Sun Zhen, Lu Binbin, Study of DDFS based on hybrid CORDIC algorithm, *Electronic Measurement Technology*, 32(7) (2009) 58-61.
7. Davide De Caro, Nicola Petra, A 380 direct digital synthesizer / mixer with hybrid CORDIC architecture in 0. 25μm CMOS, *IEEE Journal Of Solid-State Circuits*, 42(1) (2007) 151-160.
8. H. M. Ahmed, "Signal processing algorithms and architectures," *Ph.D. dissertation, Dept. Electr. Eng., Stanford Univ.* (Stanford, CA, Dec. 1981).
9. H. M. Ahmed, "Efficient elementary function generation with multipliers," *in Proc. 19th Symp. Computer Arithmetic* (Sep. 1989), pp. 52-59.