DESIGN OF SINE/COSINE SYNTHESIZER USING VECTOR ROTATION APPROACH WITH ANGLE RECODING

Valentina Vasileva Rankovska

Department of Electronics, Technical University - Gabrovo, 4, Hadji Dimitar str., 5300 Gabrovo, Bulgaria, phone: +359 66 821931/ 413, e-mail: Rankovska@tugab.bg

In the present paper the design of sine/cosine direct digital frequency synthesizer (DDFS) is studied based on the vector rotation approach with angle recording. The vector rotation approach is analyzed in relation to its advantages over the COordinate Rotation DIgital Computer (CORDIC) algorithm. Design methods using field programmable gate arrays (FPGA) is developed, achieving reduced hardware and latency and using the advantages of the novel components.

Keywords: Direct Digital Synthesis (DDS), sine/cosine generation, vector rotation, CORDIC algorithm, FPGA

1. INTRODUCTION

The Direct Digital Synthesis (DDS) is a novel approach, which is continuously developed and is more and more widely applied because of its undoubted advantages, such as high resolution, wide frequency range, agile frequency change with uninterruptible output signal phase, etc.

The operation principle of the Direct Digital Frequency Synthesizer (DDFS) [5] consists of accumulating continuously an input frequency control word into the so called Phase Accumulator (PA). The output of the PA is used to receive the amplitude values of the generated output signal, implementing various methods - [4] – table, analytical and table-analytical.

An important drawback of the table methods is the fact, that a large ROM table is necessary to generate a spurious-free signal, leading to high data processing latency and high power consumption. By the analytical methods the sine/cosine function is calculated using an iterative process. The implementation includes a lot of hardware stages.

Presently most implementations of DDFS are realized as specialized integrated circuits. The upper methods can be developed and applied more widely using novel components as FPGA and CPLD (Complex Programmable Logic Devices).

The aim of the present work is to develop methods for sine/cosine direct digital frequency synthesizer design based on the vector rotation approach with angle recoding using FPGA.

The main problems discussed in the report are:

1. Analysis of the vector rotation approach with angle recoding;

2. Developing methods for sine/cosine synthesizer design using the upper approach, which allows:

- decreasing the hardware;

- decreasing the datapath latency;
- using the recourses and flexibility of FPGA.

2. SINE/COSINE SIGNAL SYNTHESIZING

2.1. Principle of Sine/Cosine Function Calculation Using Vector Rotation

Let us examine the calculation of $\sin\theta$ and $\cos\theta$ in radians, using the rotation of the vector to angle θ counter-clockwise from an initial point coinciding with the axe *X* (Fig. 1). It can be considered as calculating of the coordinates (x_θ, y_θ) of the apex of the vector:

$$\begin{bmatrix} x_\theta \\ y_\theta \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \cos\theta \begin{bmatrix} 1 & -\tan\theta \\ \tan\theta & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \quad (1)$$

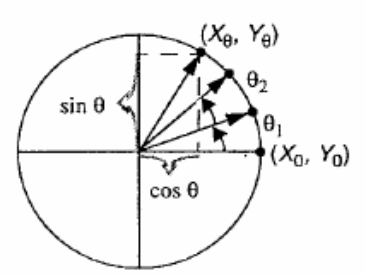

Fig. 1. Vector rotation to angle θ in Cartesian coordinates

Let us consider the rotation of the vector to an angle θ as a sum of step rotations, which values (but not their directions) are known in advance:

$$\theta = \sigma_0\theta_0 + \sigma_1\theta_1 + \ldots + \sigma_N\theta_N, \quad (2)$$

where θ_k are known fixed angles and $\sigma_k \in \{-1,0,1\}$ defines the direction of the vector rotation.

Then the equation (1) will become:

$$\begin{bmatrix} x_\theta \\ y_\theta \end{bmatrix} = K \begin{bmatrix} 1 & -\tan\sigma_N\theta_N \\ \tan\sigma_N\theta_N & 1 \end{bmatrix} \ldots \begin{bmatrix} 1 & -\tan\sigma_0\theta_0 \\ \tan\sigma_0\theta_0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \quad (3)$$

where *K* is a scale factor and is a constant as the number of the step rotations tends to infinity:

$$K = \cos\sigma_0\theta_0 \ldots \cos\sigma_N\theta_N \quad (4)$$

These dependencies are being used in various analytical methods for DDFS implementation, for instance the CORDIC algorithm [6] and its modifications.

2.2. Algorithm CORDIC

The CORDIC algorithm is studied in details in many publications. Here some differences in calculating the sine/cosine function will be marked in comparison with the suggested vector rotation approach with angel recording:

1. The scale factor *K* in Equation (4) is introduced into the initial coordinates of the vector – $(x_0,y_0)=(K,0)$.

2. The vector rotation angle is chosen such that $tg\theta_k = 2^{-k}$, where the multiplication by $tg\theta_k$ in Equations (3) is implemented as shift-and-add operations.

3. σ_k is defined by calculating the sign of the difference between two successive angles on each iteration, leading to considerably increased hardware.

3. SINE/COSINE SYNTHESIS USING VECTOR ROTATION APPROACH WITH ANGLE RECODING

At the studied method for direct digital synthesis based on the vector rotation with angle recoding [2], the output value of the phase accumulator is used to define the current angle in degrees for which the sine and cosine functions are calculated. To simplify the calculations the current angle is transformed to another one, being in the first half of the firs quadrant - in the interval $[0, \pi/4]$.

Let us consider the angle θ, measured in radians (<1 rad), expressed as:

$$\theta = \sum_{k=1}^{N} b_k \theta_k \ , \tag{5}$$

where:

$\theta_k = 2^{-k}$ are the weight coefficients of the individual bits;

$b_k \in \{0, 1\}$ – the bits corresponding to $(N + 1)$ - bit fractional presentation of the angle θ. As the angle θ is constrained to be a positive number, $b_0 = 0$ does not appear in the above sum.

Let us divide each step rotation to two equal half-rotations in such a way that at:

$b_k=1$ two successive counter-clockwise half-rotations are made with magnitude 2^{-k-1} rad;

$b_k=0$ two successive half-rotations with magnitude 2^{-k-1} rad are made but with opposite directions. As a result the vector stays at its previous position.

As a consequence of the above consideration all the first half-rotations from each couple of half-rotations which have one and the same directions can be united in a initial vector rotation to angle θ_0:

$$\theta_0 = 1/4 + 1/8 + ... + 1/2^{N+1} = 1/2 - 1/2^{N+1} rad \ , \tag{6}$$

where $\theta_0 = f(N)$ and it is a constant at a definite hardware. So:

$$\theta = \theta_0 + \theta' , \tag{7}$$

that is the rotation of the angle θ is fragmented into an initial rotation to angle θ_0, followed by N step half-rotations.

Let us examine the sum θ' consisting of the rest step half-rotations. If all the bits of the binary number N are equal to 1, then:

$$\theta' = b_1 . \frac{1}{4} + b_2 . \frac{1}{8} + ... + b_N \frac{1}{N+1} = \sum_{k=2}^{N+1} b_{k-1} . 2^{-k} \tag{8}$$

Bu the Equation (8) is not true, when some b_k are equal to zero. The error is corrected with introducing the parameter:

$$r_i = 2b_{i-1} - 1, \qquad r_i \in \{-1,1\} \tag{9}$$

that is when $b_{i-1} = 0, r_i = -1$ and when $b_{i-1} = 1, r_i = 1$. Or:

$$\theta = \sum_{k=1}^{N} b_k 2^{-k} = \theta_0 + \sum_{k=2}^{N+1} r_k 2^{-k} \tag{10}$$

The transformation of b_k to r_k is made simply by changing all the zero bits b_k in the binary representation with value (-1).

An advantage of the method over the CORDIC algorithm is the simple defining the directions of the step-rotations.

Considering the scale factor *K* and the initial rotation angle θ_0 is realized as the step half-rotations begin from an initial point with coordinates:

$$x_0 = K\cos\theta_0$$
$$y_0 = K\sin\theta_0 \quad (11)$$

Converting the angle in radians is implemented by multiplying its value by $\pi/4$.

After considering the scale factor *K* and the angle θ_0 in the initial rotation (Equations 11) the sequence of the rest step half-rotations (Equation 3) is expressed in the following way:

$$x_{k+1} = x_k - (r_k tg\theta_k)y_k$$
$$y_{k+1} = y_k + (r_k tg\theta_k)x_k \quad (12)$$

The most complex operation in the Equations (12) is the multiplication by $tg\theta_k$. As $\theta_k = 2^{-k}$, the following approximation may be introduced for sufficiently small θ_k (considering the inequality $\theta_k - tg\theta_k \le \varepsilon$, where ε is the permissible error):

$$tg\theta_k \approx \theta_k = 2^{-k} \quad (13)$$

Thus the multiplication by $tg\theta_k$ may be implemented as a shift-and-add operation of the angle value by *k* bits.

Therefore the Equations (12) may be implemented by consecutively connected stages, accomplishing shift-and-add operations till reaching the final values (x_θ, y_θ).

But the approximation in Equation (13) introduces error which may be avoided if the first *i=f(k)* stages, implementing the first terms of the sum in Equation (10) with considerable weight coefficients are substituted with a ROM table, consisting of the amplitude values of the functions $\sin\theta, \cos\theta$ [2].

To reduce the hardware and to accelerate the calculations it is also possible to merge the end *m* stages according to the equations:

$$x_{k+m} = x_k - y_k \sum_{i=k}^{k+m-1} r_i tg2^{-i}$$
$$y_{k+m} = y_k + x_k \sum_{i=k}^{k+m-1} r_i tg2^{-i} \quad (14)$$

which are valid for every $k \ge (p-1)/2$ (*p* is the datapath word length).

The architecture of DDFS of sine/cosine signal using the vector rotation approach with angle recoding is shown in Fig. 2.

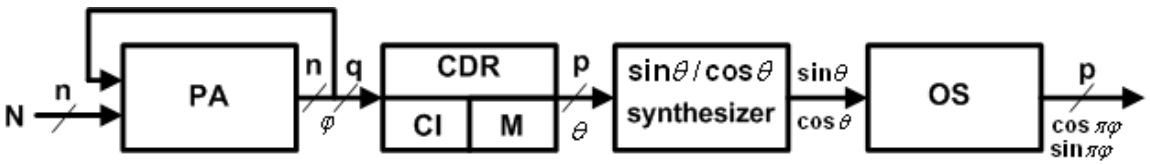

Fig. 2. Architecture of DDFS of sine/cosine signal using the vector rotation approach with angle recoding

N *n*-bits number representing the step for the changing the vector rotation angle. Its magnitude defines the output frequency.
PA Phase accumulator
CDR Converter "degrees-radians"
CI Converter of the angle θ into its image in the interval $[0, \pi/4]$
M Multiplier by $\pi/4$
OS Output stage
q Length of the truncated frequency control word
p Datapath length of the sine/cosine synthesizer

4. METHODS FOR SIN/COS DDS SYNTHESIZER DESIGN BASED ON THE VECTOR ROTATION APPROACH WITH ANGLE RECORDING

1. Definition of basic features and parameters of the output signal of the sine/cosine synthesizer – maximum output frequency, resolution, spurious-free dynamic range (SFDR), accuracy, etc.

2. Determining the datapath word length, considering the resolution of the output signal and the number of the datapath stages [2].

3. Calculation the length of the frequency control word, considering the allowed truncation error [1] and the determined in step 2 datapath word length.

4. Designing the radian converter. Two variants of the multiplier by $\pi/4$ are possible implemented in FPGA, which:

- has not a hardware multipliers – using shift-and-add operations [2]:

$$\pi/4 \approx 2^{-1} + 2^{-2} + 2^{-5} + 2^{-8} + 2^{-12} \quad (15)$$

- using hardware multipliers.

5. Determining the structure and the contents of the ROM table, replacing the first datapath stages, considering the condition:

$$\theta_k - tg\theta_k \le \varepsilon \quad (16)$$

6. Designing the rest stages of the datapath after the ROM table using the Equations (12) and (13).

7. Determining the number *m* of the merged end stages, working in parallel (Equations 14). Implementing the architecture of these stages using shift registers and adders.

8. Transforming $\sin\theta/\cos\theta$ into $\sin\pi\varphi/\cos\pi\varphi$ by the output stage [2].

9. Choosing FPGA, considering the maximum output frequency, defined in step 1, and the chosen in step 4 design variant according to its resources.

10. Simulation of the sine/cosine synthesizer design implemented in FPGA to verify the solution of the defined design problems.

5. CONCLUSION

By implementing the vector rotation approach with angle recoding using FPGA devices to generate sine/cosine signal the following is achieved in comparison with the CORDIC algorithm:

1. Reducing the hardware: because of the way in which the direction of the current rotation is determined; merging the output stages; if only sine or cosine signal is necessary.

2. Merging the *m* end stages reduces the latency for the data processing in the datapath.

3. The method allows implementing precise sin/cos generators using FPGA, which structure makes possible parallel calculations.

4. The implementation of the method using FPGA allows architectural variants of sin/cos generators, using various recourses of the hardware – PLL (Phase Locked Loop) blocks, hardware multipliers, etc.

6. REFERENCES

[1] A Technical Tutorial on Digital Signal Synthesis, Analog Devices, Inc., 1999.

[2] Madisetti, Avanindra, Alan Y. Kwentus. *A 100-MHz, 16-b, Direct Digital Frequency Synthesizer with a 100-dBc Spurious-Free Dynamic Range*. IEEE Journal of Solid-State Circuits, Vol. 34, No. 8, August 1999, pp.1034-1043

[3] Rankovska, V. *Families, Features and Recourses of FPGA and Design Technology of Devices and Systems.* Proceedings of the Int. Conference Unitech'06, Gabrovo, 2007, pp. I-202 – I-207, (in Bulgarian).

[4] Rankovska, V., H. Karailiev. *Implementation Methods of Frequency Synthesizers Based on Direct Digital Synthesis*. Proceedings of the Int. Conference Unitech'07, Gabrovo, 2007, pp. I-134 – I-139 (in Bulgarian).

[5] Tierney, Joseph, Charles M. Rader, Bernard Gold. *A digital Frequency Synthesizer*. IEEE Transactions on Audio and Electroacoustics, Vol. au-19, No. 1, March 1971, pp. 48-57

[6] Volder, J. *The CORDIC Trigonometric Computing Technique*. IRE Transactions on Electronic Computers, vol. EC-8, 1959, pp.330-334.