



**HAL**  
open science

# Direct Digital Frequency Synthesizer with CORDIC Algorithm and Taylor Series Approximation for Digital Receivers

Maher Jridi, Ayman Alfalou

► **To cite this version:**

Maher Jridi, Ayman Alfalou. Direct Digital Frequency Synthesizer with CORDIC Algorithm and Taylor Series Approximation for Digital Receivers. *European Journal of Scientific Research*, 2009, 30 (4), pp.542-553. hal-00516790

**HAL Id: hal-00516790**

**<https://hal.science/hal-00516790>**

Submitted on 11 Sep 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Direct Digital Frequency Synthesizer with CORDIC Algorithm and Taylor Series Approximation for Digital Receivers

**Maher Jridi**

*Corresponding author Laboratoire Brest-ISEN (L@BISEN)-Département optoélectronique  
Institut Supérieur de l'Electronique et du Numérique de Brest, France*

E-mail: maher.jridi@isen.fr

Tel: +33-298-038-440; Fax: +33-298-038-410

**Ayman Alfalou**

*IEEE senior member, Laboratoire Brest-ISEN (L@BISEN)-Département optoélectronique  
Institut Supérieur de l'Electronique et du Numérique de Brest, France*

## Abstract

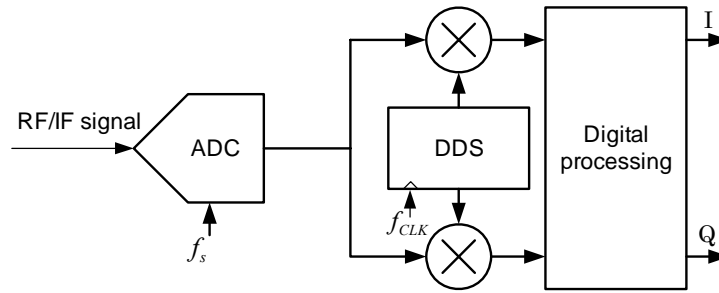
In this document we are presenting a new approach to design an optimised Direct Digital Frequency Synthesizer (DDFS) for complex demodulation used in digital receivers. For that, we suggest an adaptation of the phase to sine converter by combining the two following techniques: 1) an optimized COordinate Rotation DIgital Computer (CORDIC) algorithm 2) the principle of Taylor series approximation. To validate our proposed approach, a DDFS with 8 Hz tuning frequency resolution and 20 bits output data (for sine and cosine waves) is being implemented in Xilinx FPGA device giving a maximum operating frequency of more than 306MHz and a Spurious Free Dynamic Range (SFDR) of 112 dBc. To prove the good performances of the proposed approach, we compared it favorably with several existing DDFS architectures.

**Keywords:** Digital design, Baseband communications, DDFS survey, CORDIC algorithm, Taylor series, FPGA implementation.

## 1. Introduction

Modern digital communication systems apply Direct Digital Frequency Synthesizers (DDFS) rather than Phase Locked Loop (PLL). Indeed, the first architecture allows us to get lower energy consumption, higher spectral parameter, fine frequency tuning resolution and especially reconfiguration of the system [1]. In this paper, we will focus on the digital receiver diagram for a complex demodulation, based on a DDFS permitting the generation of signals with a high spectral purity. The goal is to obtain directly from RF (Radio Frequency) or IF (Intermediate Frequency) the baseband signal in only one chip as mentioned in figure (1). Obviously, Analog to Digital Converter (ADC) is used as interface between the RF/IF signal and multipliers inputs.

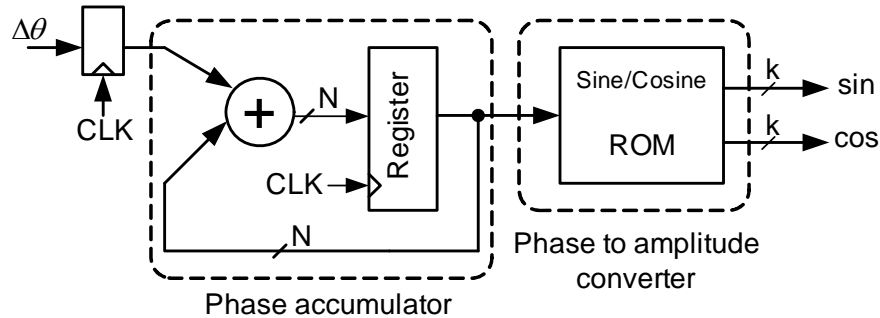
**Figure 1:** Digital quadrature demodulation diagram



Let’s remind that the most common way for digitally generating of a sine wave requires two operators:

- A digital phase accumulator to increment a constant number on each cycle of the system clock. The output of the phase accumulator is a sawtooth waveform that represents the linearly changing phase of a sinusoid.
- A phase amplitude converter to associate a sine and cosine value to each phase is generated by the phase accumulator block. This operation is achieved by using static Read Only Memory (ROM) to store the sine and cosine values for each generated phase.

**Figure 2:** Simplified block diagram of the direct digital frequency synthesizer



A simplified block diagram of the DDFS is detailed in figure (2). In this system, the output frequency is function of the clock frequency  $f_{CLK}$ , the length (in bits) of the phase accumulator “ $N$ ” and the phase increment value “ $\Delta\theta$ ”. The output frequency is defined by (1).

$$f_{out} = \frac{f_{CLK}\Delta\theta}{2^N} Hz \tag{1}$$

The spectral purity of the conventional DDFS, presented in figure (2), is determined by the phase to amplitude converter resolution. Unfortunately, a large resolution means a large ROM size and consequently higher power consumption, slower access time and increasing cost. [1].

To reduce this large ROM size, two interesting approaches [2],[3] are investigated in this manuscript. The first applies the Taylor series approximation for sine function. In this approach, the compression ROM size ratio is not high enough. The second is based on the CORDIC (COordinate Rotation DIGital Computer) algorithm. Unfortunately, for CORDIC approach, the improvement is paid by the latency and the introduction of additional arithmetic circuitry [4].

The basic idea behind this paper is to combine the two approaches mentioned above (CORDIC algorithm and the Taylor series expansion) into a unified hybrid hardware model to improve DDFS performances. The aim is to model and to design a mixed architecture in order to take advantage first,

of the CORDIC algorithm implementation in terms of power and area and second, of the Taylor series approximation in terms of fast speed access times.

The paper is organized as follows:

In Section 2, techniques of phase to sine amplitude conversion recently reported in literature are reviewed. In Section 3 a novel DDFS architecture is proposed. Some implementation issues in the design of the prototype are discussed in Section 4.

## 2. DDFS Survey

The challenge in DDFS architectures is to reduce significantly the ROM size without decreasing the SFDR (Spurious Free Dynamic Range). For  $N$  Word ROM, the ROM size is equal to  $k * 2^N$  bits where  $k$  represents the length in the ROM. In literature, several techniques are used to reduce the ROM size or to design a ROM-less DDFS architecture especially for wireless applications [4].

### 2.1. Sine Symmetry

One simple technique can be used that consists in storing only  $0$  to  $\frac{\pi}{2}$  of the sine phase information and exploiting the sine symmetry to generate the sine ROM samples for the full range of  $2\pi$ . In fact, the two Most Significant Bits (MSBs) are used to encode the quadrant: the MSB determines the sign of the result and the second MSB determines the gradient of the amplitude. Using this technique a ROM compression ratio of 4: 1 is obtained.

### 2.2. Sunderland Technique

With the Sunderland technique, the phase of the sine function is divided into three terms. This technique applies the following identity:

$$\sin(A + B + C) = \sin(A + B)\cos(c) + \cos(A)\cos(B)\sin(C) - \sin(A)\sin(B)\sin(C) \quad (2)$$

In [5], this method has been used for  $N = 12$  where  $A$ ,  $B$  and  $C$  have the same size and represent respectively the MSBs, the middle bits and the LSBs.

The equation (2) can be written as (3):

$$\sin(A + B + C) \approx \sin(A + B) + \cos(A)\sin(C) \quad (3)$$

The ROM size, using the Sunderland technique, is reduced by a factor of 51: 1 is compared to conventional DDFS. Improvements of this method are studied in [6] and summarized in table 1 with the respected compression ratio:

**Table 1:** Compression ratio improvement for Sunderland architecture

Method	Reference	Compression ratio
Sunderland	Sunderland [5]	51
	Essenwanger [7]	59
	Nicholas III [8]	128
	Kent [9]	165

### 2.3. Taylor Series Approximation

The phase address  $\theta$  is divided into the upper address  $\varphi_0$  and the lower address  $\theta - \varphi_0$ . The second order Taylor series approximation computed around  $\varphi_0$  is expressed in (4)

$$\sin\left(\frac{\pi}{2}\theta\right) = \sin\left(\frac{\pi}{2}\varphi_0\right) + k_1(\theta - \varphi_0)\cos\left(\frac{\pi}{2}\varphi_0\right) - \frac{k_2(\theta - \varphi_0)^2 \sin\left(\frac{\pi}{2}\varphi_0\right)}{2} + O(3) \quad (4)$$

where  $k_1, k_2$  are two constants to adjust Taylor series terms [2] and  $O(3)$  is the Taylor approximation of the sine function with higher orders (greater than two). This architecture needs two ROMs to store  $\sin\left(\frac{\pi}{2}\varphi_0\right)$  and  $\cos\left(\frac{\pi}{2}\varphi_0\right)$ . Improvements of basic Taylor series method are presented in table 1. In fact, Compression ratio of 157: 1 is compared to conventional DDFS technique and is obtained in [11] by using a parabolic interpolation function to further compress the ROM size.

**Table 2:** Compression ratio improvement for Taylor architecture

Method	Reference	Compression ratio
Taylor	Bellaouar [2]	21
	Scitech [10]	64
	Essenwanger [7]	67
	Eltawil [11]	157

## 2.4. CORDIC algorithm

### 2.4.1. Algorithm review

CORDIC is a very interesting technique for phase to sine amplitude conversion. This algorithm proposed in [12] utilizes dynamic transformation rather than ROM static addressing.

The CORDIC method can be employed in two different modes: the “rotation” mode and the “vectoring” mode. In the rotation mode, the algorithm basic idea consists in decomposing rotation operation into successive basic rotations. Each basic rotation can be realized by shifting and adding shift and add arithmetic operations. The rotation mode of the CORDIC algorithm could be used to compute sine and cosine of an angle  $\theta$ . Outputs after “n” iterations are computed according to the following algorithm:

```

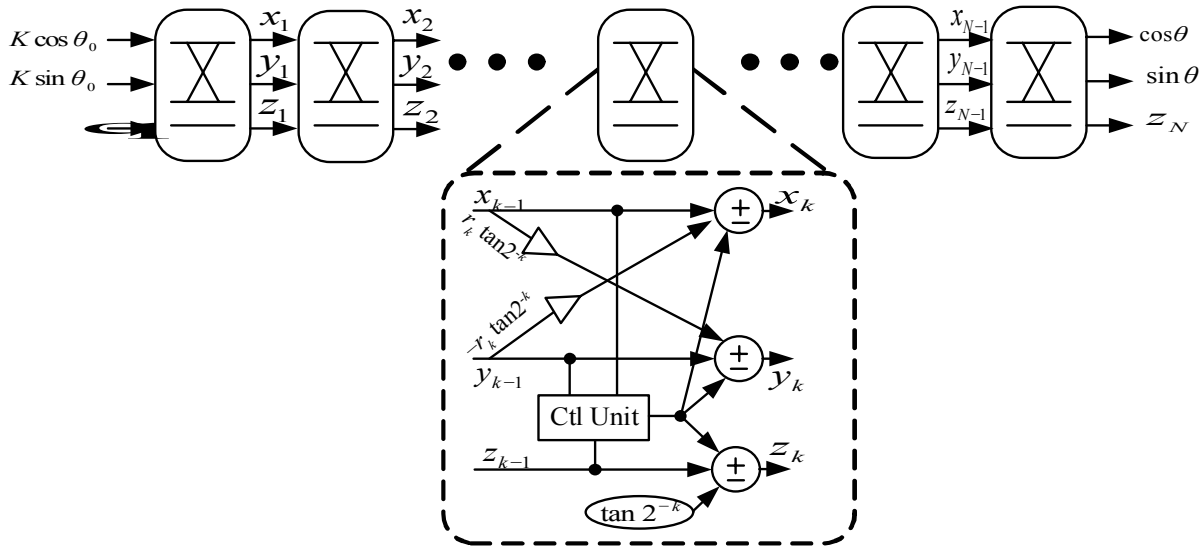
Function [x_n y_n z_n] = cordic(x,y,z,mode,N)
%% initialization, x, y, z, mode and N
For i=0:n
If (z >= 0)
d = 1;
Else d= -1;
End;
x_n = x - y * 2-i * d;
y_n = y - x * 2-i * d;
z_n = z - d * a tan(2-i);
End;
x_n = x_n * 0.6073
y_n = y_n * 0.6073
End Function;
    
```

The computation of  $\sin(\theta)$  and  $\cos(\theta)$  is based on the rotation of an initial vector of unit length, that is aligned with the abscissa ( $x_0 = 1, y_0 = 0$ ). Moreover, the accumulated angle is initialized with the desired rotation angle. For each iteration, a comparison is done between the initial angle and the resulting angle. Then, the comparison sign (represented by the variable d) is used to determine the sign of the next rotation.

**2.4.2. CORDIC Structure**

A graphical representation of the algorithmic flow is shown in figure (3). For  $N$  bits output resolution, CORDIC structure consists of a cascade of  $N$  butterflies. Each one implements a positive or negative sub-rotation by a fixed angle. The algorithm restricts angle rotation to  $\theta_k = 2^{-k}$  rad.

**Figure 3:** Block diagram of CORDIC architecture.



As  $\theta_k$  is a power of two ( $\theta_k = 2^{-k}$ ), the implementation of  $\tan \theta_k$  can be simplified by employing the approximation  $\tan \theta_k \approx \theta_k$  for sufficiently small  $\theta_k$  (i.e., for sufficiently large  $k$ ), [13]. Hence, all multiplication operations are reduced to simple shift operations.

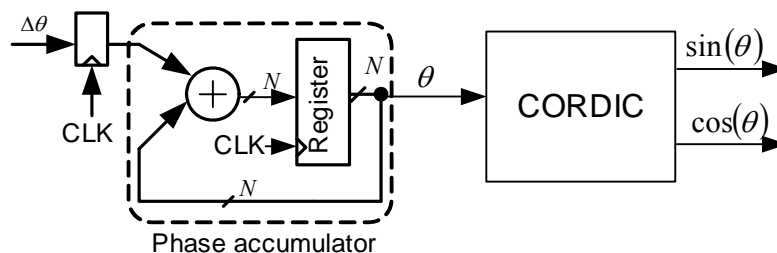
**3. Improved DDFS Based on the CORDIC Algorithm**

Below, some techniques for ROM size compression generating a sine wave are briefly mentioned. It is obvious that CORDIC method is the most suitable for high SFDR and for VLSI implementation in terms of area and power consumption [1], [13]. Nevertheless, the number of butterflies and the length of internal and external signals used in iterative CORDIC architecture decrease the speed system. In the present section, we propose a ROM-less technique reducing the latency of the whole system.

**3.1. Conventional DDFS Based on the CORDIC ALGORITHM**

In figure (4), CORDIC algorithm replaces the ROM block of figure (2) to generate sine and cosine functions. This conventional architecture is used in [3] with an iterative CORDIC implementation.

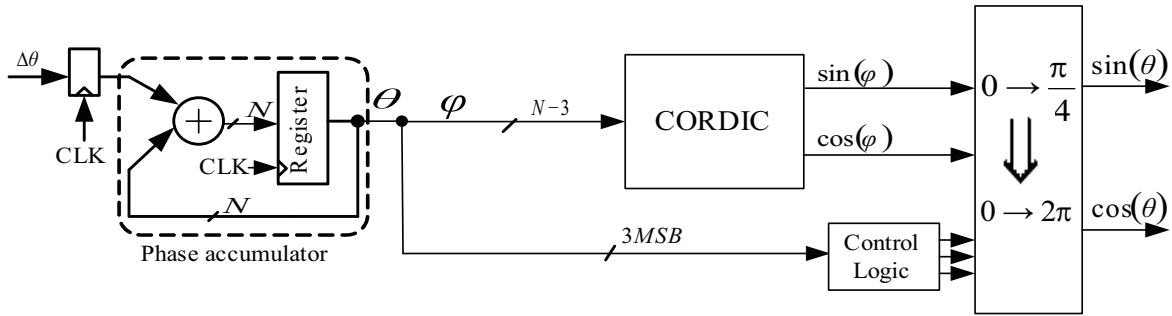
**Figure 4:** Simplified block diagram of the DDFS based on the CORDIC algorithm



### 3.2. Angle Decomposition

Conventionally, sine and cosine symmetries are performed by using  $MSB_1$  and  $MSB_2$  of  $\theta$  angle. These two bits represent the quadrant occupied by  $\pi\theta$  and increase the compression ratio by 4: 1. In order to ameliorate this compression ratio, the  $MSB_3$  is used to determine whether the angle is in the upper or lower part of the quadrant. These “3” bits are used in a control unit to convert results from  $0 \rightarrow \frac{\pi}{4}$  to  $0 \rightarrow 2\pi$  as shown in figure (5). This compression technique is possible because the sine wave from  $\frac{\pi}{2} \rightarrow \frac{\pi}{4}$  is equal to the cosine from  $0 \rightarrow \frac{\pi}{4}$  and the cosine wave from  $\frac{\pi}{2} \rightarrow \frac{\pi}{4}$  is equal to the sine wave from  $0 \rightarrow \frac{\pi}{4}$ . Hence, the compression ratio is equal to 8:1 in case of ROM utilization. In the case of CORDIC, this angle decomposition allows the use of a smaller size vector ( $N - 3$  rather than  $N$ ) in all butterflies to obtain the same resolution.

Figure 5: Angle decomposition for CORDIC algorithm



The use of small signal length decreases the required area of the phase to amplitude converter and improves the system speed. To further improve the speed parameter, a new method is proposed in next section.

### 3.3. Proposed Hybrid CORDIC-Taylor phase converter

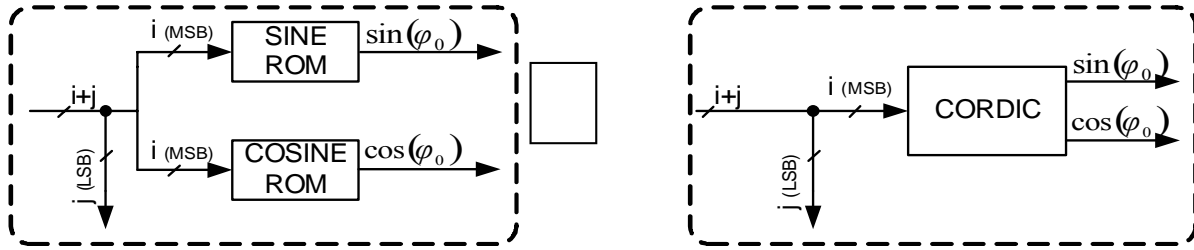
#### 3.3.1. Motivation

In literature, several improvements of the CORDIC iterative architecture are proposed. For fast VLSI implementation, pipelined CORDIC architecture gives a low latency by introducing pipelined register between successive stages [14] and [15]. Hence, each CORDIC algorithm iteration is performed in separating hardware butterflies that are pipelined. Unfortunately, the increasing of speed is paid by the increasing of the required area. Thus, for an optimized DDFS VLSI description, there is a trade-off between power and area consumption, speed, accuracy and frequency resolution.

#### 3.3.2. Principle of the proposed method

An optimized sine and cosine generation based on the CORDIC algorithm needs a low size input signals to reduce the required area and the consumed power. The basic idea of our proposed method consists first in using the Taylor series approximation and second in replacing the two required ROMs by only one CORDIC algorithm. Therefore, the input signal length, for CORDIC block, is equal to “ $i$ ” bits rather than  $i + j = N - 3$  bits. Consequently, for “ $k$ ” bits output resolution ( $k$  butterflies) shift, addition, and subtraction operations are employed with  $i$  bits vector length and not with  $i + j$  bits. A simplified diagram block of our proposed approach is presented in figure (6).

**Figure 6:** Simplified block diagram of the proposed method

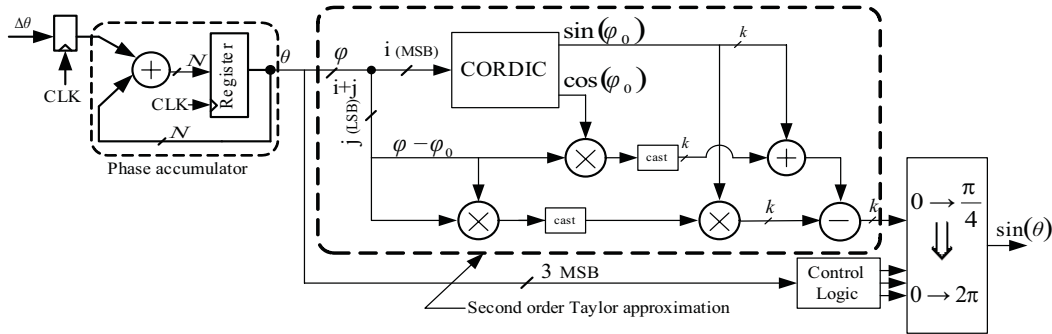


The various improvements used to achieve our proposed approach are summarized by using:

- a large phase accumulator to perform the frequency resolution ( $N \geq 16$ );
- an angle decomposition with  $N = i + j + 3$  ;
- a pipelined CORDIC architecture to optimize system speed;
- a CORDIC algorithm with an input signal length equal to “ $i$  bits” rather than “ $N$  bits” to reduce the required area;
- the second order Taylor series approximation to improve the system accuracy.

According to equation (4), sine wave generation needs sine and cosine values around a vector”  $\varphi_0$ ”. Figure (7) describes the block diagram of our proposed method. In fact, the accumulated phase  $\varphi = \frac{\theta}{8}$  is divided into two parts:  $\varphi_0$  and  $\varphi - \varphi_0$ . For the first part ( $\varphi_0$ ) the pipelined CORDIC algorithm is used to compute sine and cosine functions of  $\varphi_0$ . However, the second part of the angle ( $\varphi - \varphi_0$ ) is used to represent the Taylor series approximation. Hence, as quoted in equation (4), three multipliers and two adders are required.

**Figure 7:** Proposed method architecture



In figure (7), two cast operators are used. These blocks are very important for optimizing the implementation of the proposed method and for reducing the multiplier requirements. The method proposed is extended to cosine generation.

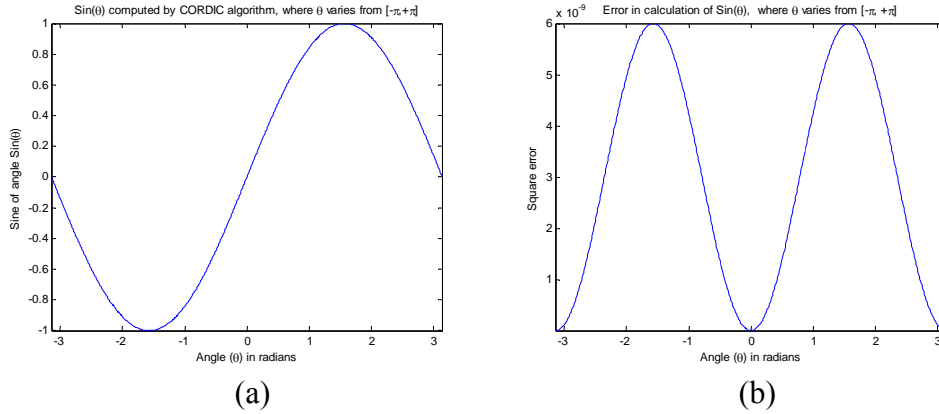
**3.3.3. Simulation results**

This section presents simulation results of our proposed DDFS architecture. Simulations are done with an angle length  $\varphi$  of 20 bits ( $i= 10$  bits and  $j = 10$  bits). The CORDIC output resolution is fixed to 20 bits to have acceptable spectral parameters. Output simulation results obtained using our approach are shown in figure (8). As it can be seen, the error curve has a very small magnitude compared to the generated signal. The resultant amplitude spectrum using 2048 point FFT shows, in figure (9), a SFDR value equal to  $112.6dBc$ . Obviously, the SFDR depends on the CORDIC precision. In fact, for an output precision of 16bits and 24 bits, SFDR is about  $82dBc$  and  $148dBc$  respectively. It is also

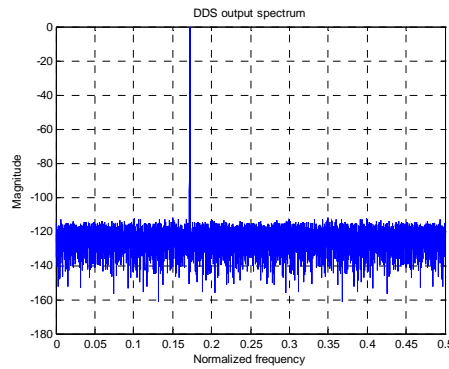


important to note that the error curve takes a sinusoidal waveform and techniques to error calibration are reported in literature but this aspect is not treated in the present paper.

**Figure 8:** DDFS output (a), Error evolution (b)



**Figure 9:** DDFS output spectrum



#### 4. Implementation

The proposed hardware model has been validated using the VHDL language. This standard language gives the choice of implementing target devices (FPGA family, CPLD, ASIC) at the end of the implementation flow. It means that the models reported here are synthesized and may be implemented on arbitrary technologies.

In the present section, synthesis results are presented for a VIRTEX 5 XC5VFX200T device. Since all Xilinx FPGA, in particular VIRTEX 5 families, give a clock signal at 50 MHz, the accumulator length is set to have a small tuning frequency (around 8 Hz) and a high SFDR. These three parameters are related by (5).

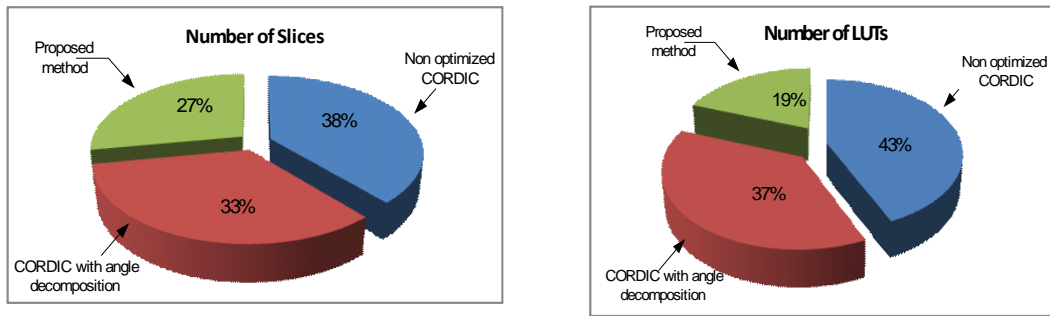
$$N = \log_2 \frac{\text{Clock\_frequency}}{\text{Tuning\_frequency}} = \log_2 \frac{50 \cdot 10^6}{8} \approx 23 \text{bits} \quad (5)$$

Therefore, 3 bits are used for angle decomposition, 10 bits represent the angle (10 MSB) and the 10 last bits are used to perform the Taylor series expansion.

For synthesis results, two strategies are employed. First, we make the choice to optimize the design to obtain the best speed. Then, we force the ISE synthesis tool to disable the using of DSP blocks in order to translate the whole netlist into LUTs and Slices. So, for all designed blocks only required multipliers are provided by VIRTEX 5 FPGA.

Synthesis results are summarized in figure (10). The number of slices and Look Up Tables (LUTs) are calculated for three designs: the non-optimized CORDIC method, the CORDIC method with angle decomposition and the proposed method. These results point out the area optimization of the proposed method. In fact, the proposed method uses 57 slice elements and 148 LUTs against 69 slices and 289 LUTs for the CORDIC method with angle decomposition and 79 slices and 333 LUTs for the non-optimized CORDIC method.

**Figure 10:** FPGA Resource Utilization



In terms of speed, our proposed Hybrid Taylor-CORDIC method presents a maximum operating frequency of about 306.091 MHz against 254.959 MHz for the Conventional CORDIC algorithm.

Finally, to enhance the DDFS survey quoted in section 2, comparisons with some previous works are presented in table 3. In this table, it is shown that two types of targets are used for implementation (ASIC, FPGA). Objectively, the synthesis tools for ASIC are better optimized, so we restrict comparison to implementations made on FPGA.

Nevertheless, ASIC implementations are listed to show that we can achieve a high SFDR of 100 dBc [13], a low power of 0.1 mW/MHz [17] and a high maximum frequency of 600 MHz [18].

For FPGA implemented phase to sine converter, the proposed architecture gives high spectral performance (SFDR = 112 dBc) and high operating frequency. We also note that for all FPGA implementation, the power consumed is not mentioned because synthesis tools give an approximating value.

**Table 3:** Comparison of performance

Reference	Target	SFDR (dBc)	Technique	Input bits	Maximum frequency (MHz)	Power (mW/MHz)	Comments
Madiseti [13] JSSC 2004	ASIC	100	Angle rotation	36	100	14	Similar to CORDIC
De Caro [17] in Trans. CAS 2005		83.6	Piecewise linear	24	536	0.102	High area consumption
De Caro [18] JSSC 2005		80	Dual slope Piecewise linear	24	600	0.127	
Song [19] Trans VLSI 2004		100	Interpolation + angle rotation	32	150	2.333	
Sodagar [20] Trans. CAS 2001		64	Pipelined ROM	32	175	--	
Jyothi [21] Trans. IEFC 2006	FPGA	89	Slope piecewise	32	25	--	Needs ROM
WANG [22] ICSP 2006		105	Taylor Series corrected	32	61.44	--	
Moran [23] Trans. CAS 2006		110	Modulus change	28	160	--	
<b>This work</b>	<b>FPGA</b>	<b>112</b>	<b>CORDIC - Taylor</b>	<b>23</b>	<b>306.091</b>	<b>--</b>	

## 5. Summary and Concluding Remarks

It has been proven that mixed CORDIC and Taylor approximation is a good alternative for frequency synthesizers. This proposed method takes advantage of second order Taylor series expansion which uses a low ROM size and of CORDIC algorithm for high performance VLSI. In fact, the proposed method uses a large phase accumulator to perform a high frequency resolution. In addition, angle decomposition and pipelined CORDIC architecture are used to decrease the whole system latency. Comparison of performances shows that the proposed method has the best SFDR and a high maximum frequency. These results may improve (especially in terms of maximum operating frequency) for ASIC synthesis. Our future works will relate to ASIC synthesis to estimate the circuit power consumption and Canonical Signed Digit (CSD) encoding to minimize the hardware complexity.

## References

- [1] J. Vandka, 2005. "Digital Synthesizers and Transmitters for Software Radio", Editor Springer.
- [2] A. Bellaouar, M. Obrecht, A. Fahim and M.I. Elmasry, 1999 "A low-power direct digital frequency synthesizer architecture for wireless communications". Proceeding IEEE Custom Integrated circuits, pp. 593-596.
- [3] S.W. Mondwurf, 2002. "Versatile COFDM Demodulation Based on the CORDIC-Algorithm", IEEE Transaction on Consumer Electronics, vol.48 no. 3, pp. 718-723.
- [4] D.D. Caro and A.G. Strollo, 2005. "High-Performance Direct Digital Frequency Synthesizers Using Piecewise-Polynomial Approximation". IEEE Transaction on Circuit and Systems. vol. 52, no. 2, pp. 324-337.
- [5] D.A. Sunderland, R.A. Strauch, S.S. Wharfield, H.T. Peterson and C.R. Colc, 1984. "CMOS/SOS frequency synthesizer LSI circuit for spread spectrum communication". IEEE Journal of Solid-State Circuits, vol.19 no. 4, pp. 497-506.
- [6] L. Cordesses, 2004. "Direct Digital Synthesis: A Tool for Periodic Wave Generation (Part 1)", IEEE Signal processing magazine, pp. 49-54.
- [7] K.A. Essenwanger, V.S. Reinhardt and A.Zarowin, 1998. "Sine output DDSs. A survey of the state of the art". Proceeding Int. Frequency control symp., pp. 370-378.
- [8] H.T. Nicholas III and H. Samueli, 1991. "A 150-MHz direct digital synthesizer in 1.25  $\mu\text{m}$  CMOS with 90-dBc spurious performances", IEEE Journal of Solid-State Circuits, vol.26 no. 12, pp. 1959-1969.
- [9] G.W. Kent and N.H. Sheng, 1995 "A high purity high speed direct digital synthesizer". Proceeding Int. Frequency control symp., pp. 207-211.
- [10] SciTech, 1992. "DDS tutorial", SciTech Electronics, Inc, San Diego, CA, Tech. Rep. V3.
- [11] A.M. Eltawil and B. Dancshrad, 2002. "Piece-wise parabolic interpolation for direct digital frequency synthesis". Proceeding IEEE Custom Integrated circuits, 31, pp. 401-404.
- [12] J.E. Volder, 1959. "The CORDIC Trigonometric Computing Technique". IRE Transaction on Electronic Computers, pp. 226-230.
- [13] A. Madiseti, A.Y. Kwentus and A.N. Wilsson, 1999. "A 100-MHz, 16-b, Direct Digital Frequency Synthesizer with a 100-dBs Spurious-Free Dynamic Range". IEEE J. Solid-State Circuits, vol.34 no. 8, pp. 1034-1043.
- [14] E. Antelo, J. Villabla and E.L. Zapata, 2008. "Low-Latency Pipelined 2D and 3D CORDIC Processors". IEEE Transaction on Computers, vol. 57, no. 3, pp. 404-417.
- [15] J. Valls, T. Sansaloni, A. P. Pascual, V. Torres and V. Almenar, 2006. "The use of CORDIC in software defined radios: A tutorial" IEEE Communications. Magazine. vol. 44, no. 9, pp. 46-50.
- [16] Y. Park and N. I. Cho, 2004. "Fixed-point error analysis of CORDIC processor based on the variance propagation formula". IEEE Transaction Circuits and system I, Reg. Papers, vol. 51, no. 3, pp. 573-584.
- [17] D. De Caro and A. G. M. Strollo, 2005. "High performance direct digital frequency synthesizers using piecewise polynomial approximation". IEEE Transaction Circuits and system I, Reg. Papers, vol. 52, no. 2, pp. 324-337.
- [18] D. De Caro and A. G. M. Strollo, 2005. "High performance direct digital frequency synthesizers in 0.25  $\mu\text{m}$  CMOS using dual-slope approximation". IEEE Journal Solid-State Circuits, vol. 40, no. 11, pp. 2220-2227.
- [19] Y. Song and B. Kim, 2004. "Quadrature direct digital frequency synthesizer using interpolation based angle rotation". IEEE Transaction on Very Large-Scale Integr. (VLSI) Systems, vol. 12, no. 7, pp. 701-710.
- [20] A.M. Sodagar and G.R. Lahiji, 2001. "A pipelined ROM-less architecture for sine-output direct digital frequency synthesizers using the second-order parabolic approximation". IEEE Transaction on Circuits and Systems II. vol. 48, no. 9, pp. 850-857.

- [21] L.S Jyothi, M. Ghosh, F.F. Dai and R.C. Jaeger, 2006. "A novel DDS using nonlinear ROM addressing with improved compression ratio and quantization noise". IEEE Transactions on Ultrasonics Ferroelectrics and Frequency Control vol. 53, no. 2, pp. 274—283.
- [22] W. Wang, Z. Yifang and Y. Yang, 2006. "Efficient wireless Digital Up Converters design using system generator". International conference on Signal Processing, ICSP, pp 443—446
- [23] D.R. Moran, J.G. Menoyo and J.L. Martin, 2006. "Digital Frequency Synthesizer Based on Two Coprime Moduli DDS". IEEE Transaction on Circuit and Systems II, vol. 53, no.12, pp. 1388—1392.