A Direct Digital Frequency Synthesizer Based on the Quasi-Linear Interpolation Method

Ashkan Ashrafi, *Senior Member, IEEE*, Reza Adhami, *Senior Member, IEEE*, and Aleksandar Milenković, *Member, IEEE*

***Abstract*—This paper introduces a novel direct digital frequency synthesizer (DDFS) with an architecture based on the quasi-linear interpolation method (QLIP). The QLIP method is a hybrid polynomial interpolation in which the first quarter of a cosine function is approximated by two sets of linear and parabolic polynomials. The section of the cosine function that is closer to its peak is interpolated by parabolic polynomials, due to its resemblance to a parabola. The rest of the function, which is closer to where it approaches zero, is interpolated by linear polynomials. The paper describes the proposed interpolation method and its VLSI implementation. The performance of the proposed implementations is compared to several state-of-the-art DDFS designs.**

***Index Terms*—Direct digital frequency synthesizer (DDFS), polynomial interpolation, spurious free dynamic range.**

I. Introduction

DIRECT digital frequency synthesizer is a digital system that generates frequency controlled sinusoidal signal for communication systems. High spectral purity and fast frequency hopping capability are the main characteristics of a DDFS system that make it suitable for modern and sensitive devices such as Spread Spectrum and RADAR. The block diagram of a DDFS is shown in Fig. 1. In a DDFS, the amplitude of a sinusoidal signal is digitally generated by a phase-to-sine mapper (PSM) whose input is fed by an accumulator. The input of the accumulator and its wordlength determine the output frequency and its resolution, respectively. The frequency of the output sinusoidal signal is

$$f_{\text{out}} = \frac{F_r}{2^L} f_{clk} \tag{1}$$

where, F_r, L and f_{clk} are the input of the accumulator, the accumulator wordlength and the clock frequency, respectively. The PSM was originally conceived as a simple read-only memory (ROM) that would contain the amplitude information of the waveform but, it has gradually evolved into less complex architectures that consume a less amount of power and occupy a smaller area on the chip. The lower PSM complexity was achieved at the expense of introducing nonidealities in the PSM. These nonidealities distort the output sinusoid causing spectral impurities in the form of spurious harmonics. In the past three decades, many researchers have endeavored to achieve an efficient PSM architecture with least complexity and highest spectral output signal purity. The major challenge in reducing the complexity of the PSM is maintaining the spectral purity of the output signal above a certain level. The spectrum of the signal generated by the PSM, which is called the signature sequence, will be an important factor in determining the spurious free dynamic range (SFDR) of the output signal. The SFDR of a signal is determined by the ratio of the fundamental harmonic amplitude to the maximum spur amplitude [1].

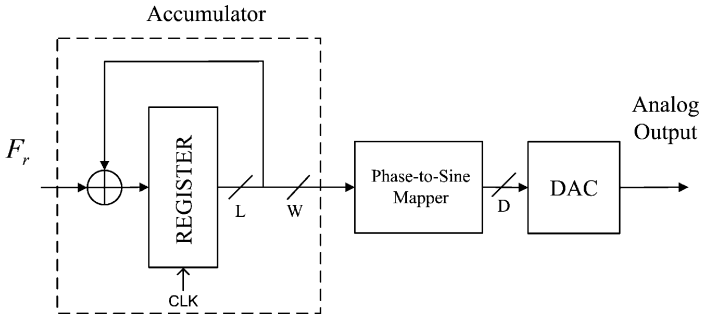

Fig. 1. Block diagram of a DDFS system.

The most important technique to reduce the complexity of the PSM utilizes the quadrature wave symmetry of the sinusoidal function. By employing this method, the complexity of the PSM can be reduced by a factor of four with no effect on the SFDR of the output [1]. Most of the methods embody the quadrature wave symmetry, which means they generate the output sinusoidal from the first quadrant. There are also some exceptions where half wave symmetry is used [2]. The other method that is also used in all DDFS systems, is called phase truncation, which is basically truncating the accumulator's wordlength from L to W where $L > W$. This truncation causes a significant reduction in the SFDR of the output signal [4]. The SFDR of a DDFS with truncated phase is calculated in [3] and [5] (assuming the signature sequence is an ideal sinusoid) which is given by

$$\text{SFDR} = 20\log_{10}\left(\frac{\sin(\frac{\pi(2^W-1)}{2^L})}{\sin(\frac{\pi}{2^L})}\right)\text{dBc}. \tag{2}$$

Quantization of the output sinusoidal signal due to the arbitrary PSM nonidealities causes another set of spurs in the output spectrum. It is shown in [3] that the spur magnitudes and locations caused by the phase truncation do not depend on the spurs

Manuscript received October 24, 2008; revised May 01, 2009. First published November 03, 2009; current version published April 09, 2010. This paper was recommended by Associate Editor B. Bakkaloglu.

A. Ashrafi is with the Department of Electrical and Computer Engineering, San Diego State University, San Diego, CA 92182 USA (e-mail: ashrafi@mail.sdsu.edu)

R. Adhami and A. Milenković are with the Department of Electrical and Computer Engineering, University of Alabama in Huntsville, Huntsville, AL 35899 USA (e-mail: adhami@ece.uah.edu; milenka@ece.uah.edu).

Digital Object Identifier 10.1109/TCSI.2009.2027645

produced by the arbitrary PSM nonidealities or simply the spectrum of the signature signal, thus they are in fact disjoint. Therefore, by choosing an appropriate value for W and knowing the spectrum of the signature sequence, one can exactly calculate the magnitude and location of the spurs at the overall DDFS output spectrum. If the only source of nonidealities in the PSM is the quantization error then the SNR of the output sinusoid would be evaluated by

$$\mathrm{SNR} = 6D + 1.7\,\mathrm{dBc} \tag{3}$$

where D is the output wordlength.

In addition to the aforementioned methods that are employed in all DDFS systems, other methods are needed for further reduction in the PSM complexity. In general, the PSM complexity reduction techniques can be classified into three major groups, polynomial interpolation [6]–[20], angle rotation, [24]–[29] and ROM compression methods [1], [30], [31].

In the polynomial interpolation method, the first quadrant of the sinusoidal signal is generally divided into several segments and each segment is approximated by a polynomial of a desired order. Normally, the number of segments s is an integer power of two i.e., $s = 2^u$ to simplify the architecture of the PSM, which is built by a binary digital circuitry.

The type of the polynomial interpolation method is chosen by the parameter to be optimized. If the optimization of the SNR of the output signal is required, the coefficients of the polynomials can be determined by Legendre polynomials [12] or Moore-Penrose pseudo-inverse method [21]. For minimizing the maximum interpolation error, Chebyshev polynomials should be used [18], [19]. If the optimization of the SFDR is required, the coefficients of the polynomials can be determined by nonlinear optimization methods such as the Nelder-Mead Nonlinear simplex method [6], genetic algorithms [10] or axial search [12]. To find the polynomial interpolation method that optimizes the SFDR, we should know the theoretical upperbound of the SFDR for the method [21]. The SFDR upperbound can help us to reduce the number of iterations in the optimization algorithm in order to achieve a faster convergence rate.

In this paper, the quasi-linear interpolation polynomial technique (QLIP), which is introduced by the authors in [22] and [23] is throughly studied. In Section II, the QLIP method is introduced as a combination of piecewise parabolic and linear polynomials interpolations employed to approximate the first quadrant of the cosine function. Moreover, the location on the first quadrant where the interpolation should be changed from even parabolic (ax^2+b) to linear is determined to maximize the SFDR of the output signal. In Section III, a realization of the DDFS utilizing the QLIP method is proposed. In Section IV, the conclusions are drawn. This section also discusses how to preserve the largest possible SFDR (by using optimization) in order to compensate for small alteration in the hardware architecture. In Section V, a VLSI implementation of the proposed method is discussed and the results are compared with the state-of-the-art designs reported in the literature. The implementation results show that the proposed design is less complex than other methods and it is suitable for inserting pipeline levels in order to increase the speed of the chip. In Section VI, the conclusion is drawn.

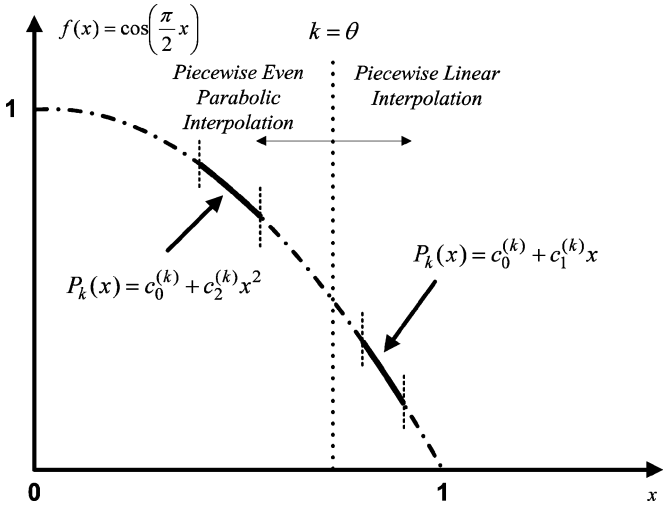

Fig. 2. Quasi-linear interpolation method (QLIP).

II. The Quasi-Linear Interpolation Method (QLIP)

Consider the first quadrant of the cosine function defined as

$$f(x) = \cos\left(\frac{\pi}{2}x\right), \quad 0 \le x \le 1 \tag{4}$$

is divided into $s = 2^u$ segments and each segment is approximated by a polynomial $P_k(x)$ such that

$$P_k(x) = \begin{cases} c_0^{(k)} + c_2^{(k)}x^2 & 1 \le k \le \theta \\ c_0^{(k)} + c_1^{(k)}x & \theta+1 \le k \le s \end{cases} \qquad 0 \le x \le 1 \tag{5}$$

where k is the segment number and $c_0^{(k)}$, $c_1^{(k)}$ and $c_2^{(k)}$ are the polynomial's coefficients and θ is the segment number where the type of polynomial is changed from even parabolic to linear. The reason behind this polynomial arrangement is the fact that the cosine function (4) behaves more like a parabola where x is closer to 0 and it behaves more like a line where x is closer to 1. Fig. 2 shows the interpolation configuration.

The first step in finding the values of the coefficients $c_0^{(k)}$, $c_1^{(k)}$ and $c_2^{(k)}$ is to find the SFDR upperbound for different values of θ and then determine the appropriate value of θ that is corresponding to the maximum SFDR upperbound. To find the SFDR upperbound, we can use the universal method introduced in [21]. The detailed analysis of finding the SFDR upperbound is explained in the Appendix . Fig. 3 exhibits the the SDFR upperbounds of the QLIP method versus θ for a different number of segments. It is obvious from Fig. 3 that the maximum SFDR upperbound occurs when

$$\theta = \frac{3s}{4}. \tag{6}$$

The maximum SFDR upperbound values for a different number of segments are given in Table I. It is worth noting that the algorithm calculating the SFDR upperbounds will also provide the

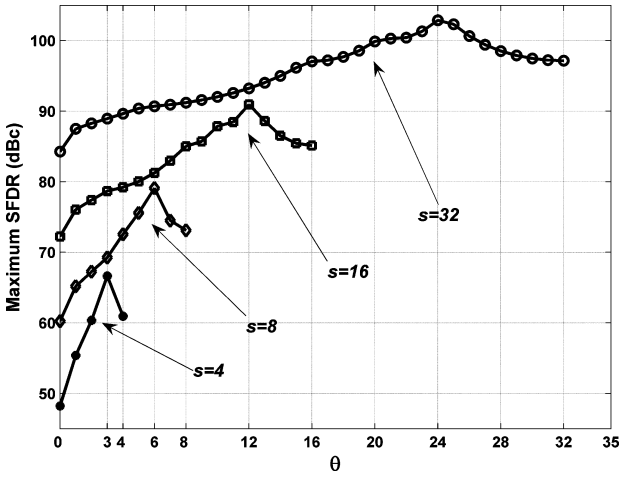

Fig. 3. SFDR upperbounds of the QLIP method versus θ for a different number of segments.

TABLE I
THE MAXIMUM SFDR UPPERBOUND VALUES

Number of Segments	$s=4$	$s=8$	$s=16.$	$s=32$
Maximum SFDR (dBc)	66.64	79.08	90.94	102.87

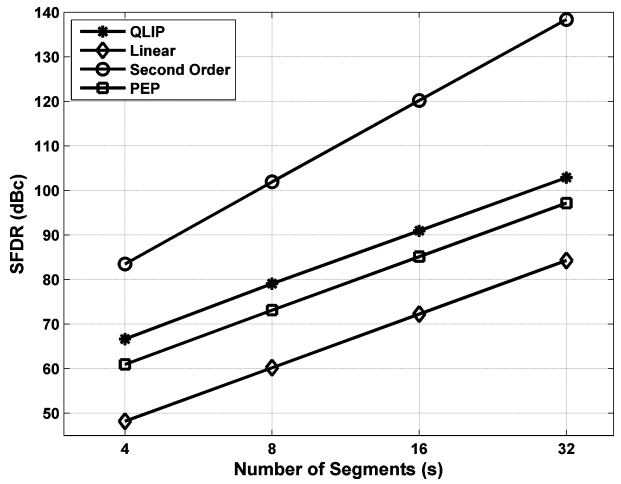

Fig. 4. SFDR comparison of different polynomial interpolation methods.

corresponding coefficients $c_0^{(k)}$, $c_1^{(k)}$ and $c_2^{(k)}$ [21]. These coefficients can be used as an initial guess for any further optimization algorithm.

Fig. 4 shows the SFDR upperbounds comparison between ideal DDFS architectures based on different polynomial interpolation methods: QLIP, Piecewise Even Parabolic (PEP), Linear and Second Order (pure parabolic). According to Fig. 4, the QLIP method offers better SFDR than linear and PEP methods. Obviously, the second order polynomial provides us with better SFDR values by increasing the complexity of the system.

III. REALIZATION OF THE QLIP-BASED DDFS

Now, by knowing the maximum SFDR upperbounds for the QLIP method, we are ready to realize the DDFS. This can be done by quantizing the polynomials (5) and then optimizing the quantized coefficients to obtain the maximum possible SFDR. To begin the process, first we should quantize the phase x. The phase quantization can be done by this relationship

$$x = \frac{n}{2^{W-2}} \tag{7}$$

where W is the phase wordlength and n is the new variable that takes on integer numbers in the range $0 \le n \le 2^{W-2}-1$. The quarter wave symmetry is used in this design, thus the maximum value of n is $2^{W-2}-1$. By substituting (7) into (5), scaling the amplitude up to 2^{D-1} and choosing $W = D+1$, we can obtain

$$P_k[n] = \begin{cases} 2^{D-1}c_0^{(k)} + c_2^{(k)}\frac{n^2}{2^{W-2}} & 1 \le k \le \frac{3s}{4} \\ 2^{D-1}c_0^{(k)} + c_1^{(k)}n & \frac{3s}{4}+1 \le k \le s \end{cases} \tag{8}$$

where D is the output wordlength of the DDFS.

The next step is to quantize $c_1^{(k)}$ and $c_2^{(k)}$. To avoid the requirement of a digital multiplier, these coefficients can be approximated by a summation of integer powers of two, which can be realized by logical left and right shifts [10]. The following formula represents the approximation

$$c_{1,2}^{(k)} = \sum_{j=0}^{r} h_{jk}2^{g_{jk}} \tag{9}$$

where $h_{jk} \in \{+1,-1\}$, $g_{jk} \in \{\ldots,-2,-1,0,+1,+2,\ldots\}$ and r is a fixed number. This is the well-known Canonic Signed Digit (CSD) representation [11]. The value of r can be indefinitely large, but to obtain a less complex architecture, it should be chosen as small as possible. By substituting (9) into (8), the final discrete polynomial will become

$$P_k[n] = \begin{cases} q_k + \sum_{j=0}^{r} h_{jk}\langle \frac{n^2}{2^{W-2}}, g_{jk}\rangle & 1 \le k \le \frac{3s}{4} \\ q_k + \sum_{j=0}^{r} h_{jk}\langle n, g_{jk}\rangle & \frac{3s}{4}+1 \le k \le s \end{cases} \tag{10}$$

where $\langle a,b\rangle$ means that the binary number a has been shifted to the right ($b \le 0$) or to the left ($b \ge 0$) by b bits and the result is truncated to an integer number. The parameter q_k can be obtained by

$$q_k = \left\lfloor 2^{D-1}c_0^{(k)} \right\rfloor \tag{11}$$

where $\lfloor . \rfloor$ is the floor function.

According to (9), the variable h_{jk} can take on $+1$ or -1. When it is $+1$, the corresponding digital realization would be an adder in the summation process of (10) but when it is -1, a digital subtractor should be employed to perform the task. The necessity of a subtractor makes the system more complex because a digital realization of a subtractor is actually a combination of an adder and a 2's complementor. To avoid subtractors in the final digital realization of the DDFS, and thus reducing the complexity of the design, we can replace the subtractor by a combination of a 1's complementor and an adder. This approach will change the architecture of the DDFS and it will be different from the one given by (10). Nevertheless, we can still use the architecture given by (10) because the final values of the other coefficients, i.e., g_{jk} and q_k are determined by the optimization algorithm. Thus, the modification of the DDFS structure would not significantly impact the SFDR.

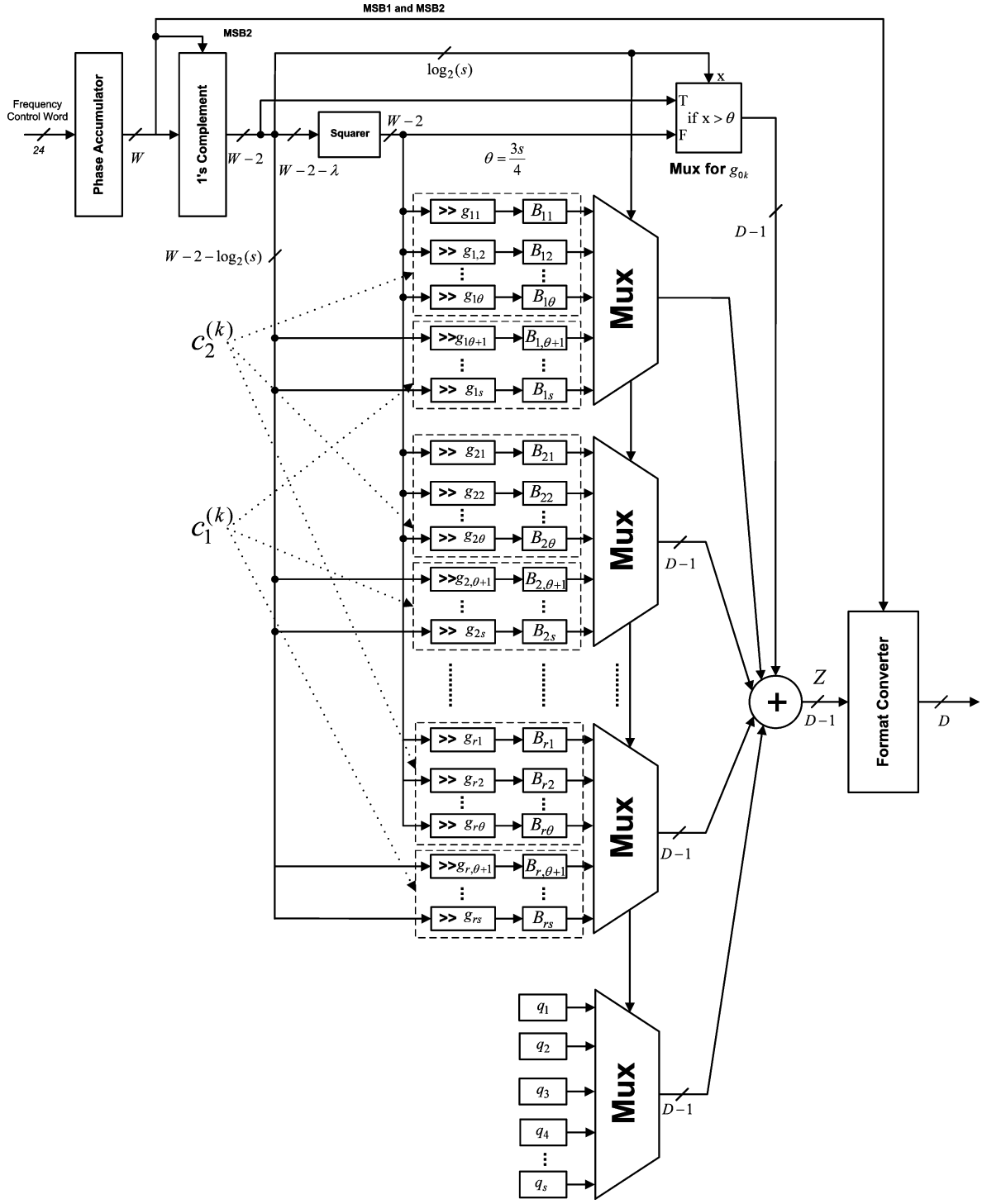

Fig. 5. Block diagram of the DDFS architecture based on the QLIP method.

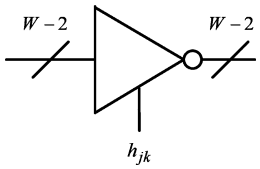

Fig. 6. Block diagram of B_{jk} blocks.

The architecture introduced in [10], which is designed to implement the piece-wise linear interpolation method, can be modified to accommodate the implementation of the proposed method. The modification would be simply adding a squarer to realize the second order polynomial section of the QLIP method and multiplexers that select between the linear and even parabolic interpolations based on the value of $\theta = 3/4$. Other architectures could have also been employed for this design such as the one with fixed-width multiplier introduced in [12]. However, the architecture of [10] is used because it is more suitable for pipelining. It will be shown later that by adding a few pipeline stages we can achieve clock frequencies up to 1 GHz. Fig. 5 illustrates the block diagram of the proposed

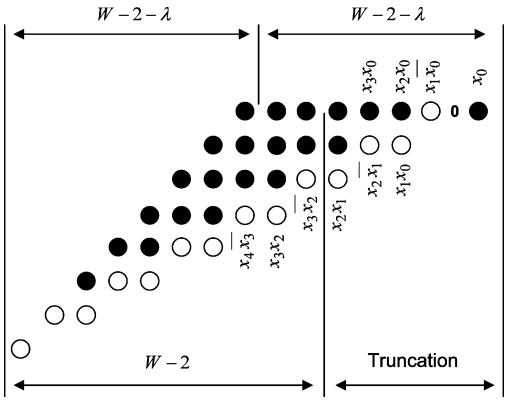

Fig. 7. The partial product table of the truncated squarer for $W = 12$ and $\lambda = 2$.

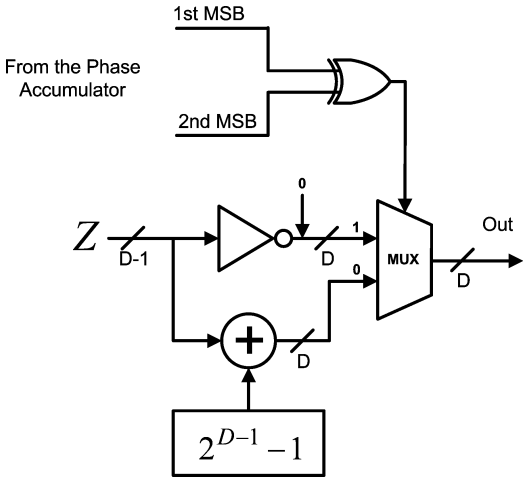

Fig. 8. Internal structure of the format converter.

architecture. The internal architecture of the B_{jk} blocks in Fig. 5 is illustrated in Fig. 6. Basically, each B_{jk} block is a conditional complementor. When the corresponding h_{jk} is -1, the block negates the input word and when the corresponding h_{jk} is $+1$, the block acts as a simple buffer.

To implement the squarer utilized in the design, the pre-truncated squarer introduced in [33] is employed. The input wordlength of the pre-truncated squarer is $W - 2 - \lambda$, and its output wordlength is $W - 2$ where $\lambda < W/2 - 1$. In this situation, the partial product matrix has $2W - 4 - 2\lambda$ columns. To implement this kind of pre-truncated squarer, we can keep the $W - 2$ most significant columns of the squarer's partial product array and discard the rest of the array (the least significant $W - 2 - 2\lambda$ columns). Now, by using carry-save adders, the array can be shrunk to two final numbers and using a fast adder, such as a carry-look-ahead adder, we can obtain the result of the operation. Fig. 7 illustrates the partial product array of the pre-truncated squarer [33]. It is worth noting that the truncation error will be almost compensated in evaluating the final parameters of the proposed DDFS.

The discrete polynomial (10) (with the aforementioned modifications) would only generate the first quadrant of the cosine signal. To generate the complete cosine signal, we should reproduce the other quadrants using the first one. In this case, let's assume Z is the output of the digital system that generates the first quadrant. The following equation creates the entire cosine signal by using Z

$$\text{Out} = y.(\overline{Z})_{D-1} + \overline{y}(Z + 2^{D-1} - 1) \tag{12}$$

where $y = l_1 \oplus l_2$ ($\oplus$ is the XOR operator), l_1 and l_2 are respectively the 1*st* and 2*nd* most significant bits of the phase and $(\overline{Z})_{D-1}$ is the bit extension of the logical NOT of Z, i.e., $\{0, \overline{Z}\}$, where $\{.\}$ represents concatenation of two or more words. Fig. 8 shows the structure of this section of the design, which is shown by the "format converter" block in Fig. 5.

IV. Optimization of the Quantized Coefficients for Maximum SFDR

In the final architecture, the parameters q_k, h_{jk}, g_{jk}, λ, r, s, W and D are to be determined for the desired SFDR and output wordlength. The parameters W and D are determined based on the chosen SFDR and output SNR. The number of segments s should be determined based on the desired SFDR and the SFDR upperbound derived for each value of s (Fig. 4). The parameter r is to be determined based on the desired level of complexity of the target DDFS architecture and the maximum achievable SFDR. The greater the value of r, the more complex the final design would become, but the higher SFDR can be achieved. At some point, increasing r would only add to the complexity of the system and will not have any further impact on SFDR.

The parameters h_{jk} and g_{jk} can be determined by representing the ideal coefficients, $c_1^{(k)}$ and $c_2^{(k)}$ in the CSD format. The value of r directly depends on the number of segments s. For $s = 4$, 8, 16, the value of r is 3, 3 and 4, respectively. In choosing the parameters g_{jk}, we always consider that $g_{0k} = 0$ in order to reduce the hardware complexity. The parameter q_k can be determined by using the Nelder-Mead nonlinear simplex algorithm (available in MATLAB) to maximize the SFDR of the system. The above procedure can be repeated for different values of g_{jk}'s (except for g_{0k}'s which are always zero) over a limited range of ± 2 around their values obtained from CSD representation of the ideal coefficients $c_1^{(k)}$ and $c_2^{(k)}$. This exhaustive search will lead to the optimized coefficients corresponding to the maximum possible SFDR. The CPU time of this numerically intensive optimization method could be from several minutes for the architecture with $s = 4$ to a few hours for the architecture with $s = 16$. Since the SFDR theoretical upperbound is known, we can terminate the optimization algorithm when an SFDR close enough to the SFDR theoretical upperbound is reached (for example, 1 dBc to 2 dBc below the upperbound). This can significantly reduce the CPU time. Moreover, randomly choosing the values of g_{jk} within their predefined ranges can help us to reach the optimized SFDR faster.

The parameter λ is determined for the given architecture to maximize the SFDR. Its value should be changed between 0 and $W/2-1$. For each value of λ, the parameter q_k should be determined to find the maximum SFDR. The value of λ that makes the highest SFDR will be chosen as the final value. One might expect that the maximum SFDR can be obtained at $\lambda = 0$, because it is the case where the squarer is not pre-truncated and,

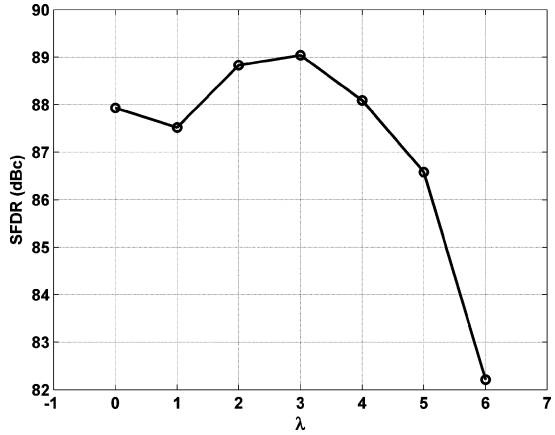

Fig. 9. Variations of the maximum SFDR ($s = 16$, $W = 15$ and $r = 3$) with respect to the parameter λ .

TABLE II
PARAMETERS OF A QLIP-BASED DDFS FOR $s = 4, W = 11, D = 10, r = 2$ AND $\lambda = 3$. THE PARAMETER h_{jk} IS ALWAYS -1

	g_{0k}	g_{1k}	g_{2k}	q_k
$k = 1$	0	-2	-10	504
$k = 2$	0	-3	-6	498
$k = 3$	0	-5	-8	483
$k = 4$	0	-1	-6	778

TABLE III
THE PARAMETERS OF A QLIP-BASED DDFS FOR $s = 16, W = 15, D = 14, r = 3$ AND $\lambda = 3$

	g_{0k}	h_{0k}	g_{1k}	h_{1k}	g_{2k}	h_{2k}	g_{3k}	h_{3k}	q_k
$k = 1$	0	-1	-2	-1	-6	+1	-9	+1	8183
$k = 2$	0	-1	-2	-1	-6	+1	-8	+1	8184
$k = 3$	0	-1	-2	-1	-5	+1	-6	-1	8184
$k = 4$	0	-1	-2	-1	-5	+1	-7	+1	8179
$k = 5$	0	-1	-3	-1	-4	-1	-7	-1	8171
$k = 6$	0	-1	-3	-1	-5	-1	-6	-1	8151
$k = 7$	0	-1	-3	-1	-5	-1	-	-	8134
$k = 8$	0	-1	-3	-1	-	-	-	-	8087
$k = 9$	0	-1	-4	-1	-5	-1	-10	-1	8024
$k = 10$	0	-1	-4	-1	-	-1	-	-1	7941
$k = 11$	0	-1	-6	-1	-7	-1	-9	-1	7821
$k = 12$	0	-1	-6	-1	-8	-1	-	-1	7678
$k = 13$	0	-1	-1	+1	-7	+1	-7	+1	12208
$k = 14$	0	-1	-1	-1	-6	-1	-7	-1	12519
$k = 15$	0	-1	-1	-1	-4	-1	-7	+1	12744
$k = 16$	0	-1	-1	-1	-4	-1	-8	-	12834

thus it does not introduce any error in the squaring block. Nevertheless, the maximum SFDR never happens at $\lambda = 0$. The reason for this phenomenon is perhaps the excessive truncations in the squarer that creates significant nonlinearities. However, a clear analysis that can explain this effect has not been devised. For example, the maximum SFDR occurs at $\lambda = 3$ for the case of $s = 16$, $W = 15$ and $r = 3$. This fact is illustrated in Fig. 9. Tables II and III show the parameters obtained for $s = 4$ and $s = 16$ from the optimization procedure. They respectively achieve the SFDRs of 63.2 dBc and 89.04 dBc, which are as close as possible to the corresponding SFDR upperbounds given in Table I. The spectrum of the signature sequences of the DDFS architectures, whose coefficients are given in Tables II and III, are respectively shown in Figs. 10 and 11. The spectrums are compared with the spectrum of the ideal cases in which the SFDR upperbounds are obtained by the

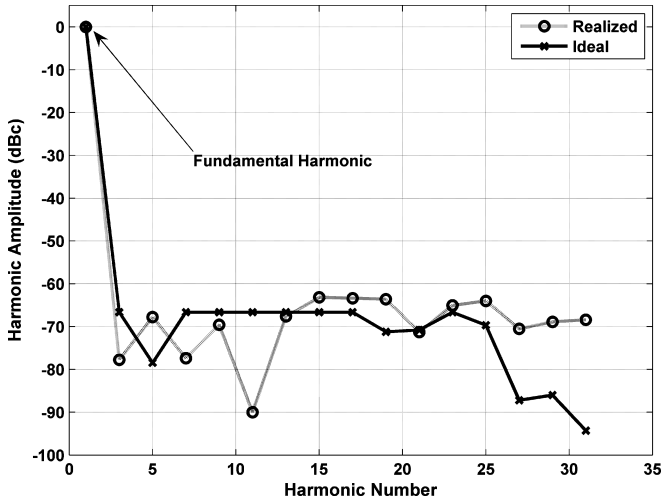

Fig. 10. Spectrum of the signature sequence of the QLIP DDFS ($s = 16$, $W = 15$, $D = 10$ and $r = 3$) whose parameters are given in Table II.

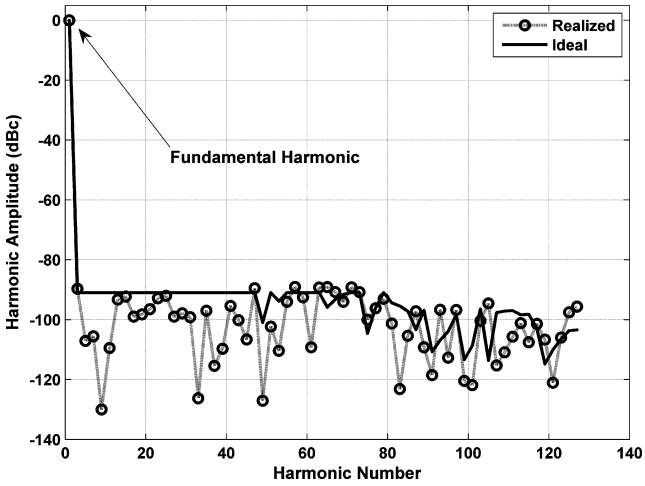

Fig. 11. Spectrum of the signature sequence of the QLIP DDFS ($s = 16$, $W = 15$, $D = 14$ and $r = 3$) whose parameters are given in Table III.

method given in the Appendix . The deviation of the spectrum of the signature sequences from the ideal ones is caused by the quantization error introduced in the digital implementations of the systems. These deviations are minimized by optimizing the coefficients explained above.

V. VLSI IMPLEMENTATION AND SIMULATION RESULTS

The proposed designs are implemented using TSMC-0.13 μm, 1.2 V supply standard cell library. The design is modeled by Verilog RTL and then synthesized by Cadence PKS tool. The place and route procedure as well as power analysis are also performed by Cadence Encounter SOCE tool. The performance of the proposed method is compared with some of the state-of-the-art designs in Table IV. Due to different implementation methods, SFDR choices, standard cell libraries, fabrication processes, accumulator wordlengths and so on, a fair comparison between the performances of different DDFS designs is not easy. To overcome this problem, the normalized area is introduced, which is defined as the total silicon area of the chip divided

TABLE IV
COMPARISON BETWEEN THE PROPOSED METHOD AND SOME RECENT DESIGNS

Design	Method	s	SFDR (dBc)	Max. Freq. (MHz)	Power (μW/MHz)	Area (μm^2)	Normalized Area $\times 10^5$	Process (μm)	MOS #
This paper $L=24, D=10, W=11$	QLIP (Single Phase)	4	63.2	313	4.9	3756	2.22	0.13	2950
This paper $L=24, D=10, W=11$	QLIP (pipelined) (Single Phase)	4	63.2	1000	8.2	9784	5.79	0.13	7685
This paper ($\theta=0$) $L=24, D=10, W=11$	Piecewise Linear (Single Phase)	8	60.2	275	5.6	4528	2.68	0.13	3557
Langlois [7] $L=16, D=12, W=11$	Piecewise Linear (Single Phase)	32	60	80	30	11000	3.39	0.18	NA
De Caro [12] $L=24, D=9, W=11$	Piecewise Linear (Quadrature)	32	62.3	224	39.3	16800	2.69	0.25	3545
This paper $L=24, D=14, W=15$	QLIP (Single Phase)	16	89.04	178	13.8	8791	5.20	0.13	6905
This paper $L=24, D=13, W=14$	QLIP (Single Phase)	16	82.7	203	10.7	6499	3.85	0.13	5105
This paper ($\theta=0$) $L=24, D=13, W=14$	Piecewise Linear (Single Phase)	32	83.2	172	12.2	9172	5.42	0.13	7205
De Caro [12] $L=24, D=12, W=15$	Piecewise Linear (Quadrature)	64	83.6	156	57.3	25700	4.11	0.25	5530

by the feature size squared ($0.13^2 = 0.0169$ in our case). Other parameter that can be used for a fair comparison is the approximate number of MOS transistors, which is obtained by the method introduced in [10] (The area of the chip divided by the area of a four input NAND gate times the number of MOS transistors in a NAND gate, which is four).

For the DDFS design with the SFDR in 60 dBc range, the normalized area of the proposed design is 52% smaller than that of the design in [7] and it is 20% smaller than that of the design in [12]. Moreover, the percentage of the silicon area occupied by the interpolator is 20% of the total area of the chip, which is 10% less than that of the design introduced in [12] for the same SFDR range. These comparisons can also be confirmed by using the number of MOS transistors given in the last column of Table IV. In addition, the proposed method is used when $\theta = 0$ where the QLIP interpolation becomes the linear interpolation. The normalized area of the corresponding chip is very close to that of the design given in [12]. This shows the effectiveness of the QLIP method in reducing the silicon area for the SFDR range of 60 dB.

For the DDFS designs with the SFDR in 80 dBc range, the normalized area of the proposed design (with SFDR of 89.04 dBc) is 20% larger than that of the design in [12] but, it should be noted that its SFDR is 6 dB greater and the output wordlength is two bits larger. To have a fair comparison, we have reduced the phase wordlength W by one bit ($W = 14$) and designed the DDFS whose SFDR is 82.7 dB. The normalized area of this new design is 25% less than that of the original design and it is 6.7% less than that of the design given in [12]. Obviously a reduced accumulator size contributes to a smaller silicon area but, this reduction occurs while the SFDR is not changed dramatically. This shows that the QLIP design is very effective in obtaining the optimum architecture for the maximum possible SFDR. The linear interpolation method using the same architecture (when $\theta = 0$) is also designed. The results show that the normalized chip area of the design with $\theta = 0$ is larger than those of the QLIP method and the method given in [12]. This also exhibits the inherent optimized nature of the QLIP method.

The maximum clock frequencies and the power consumptions of the designed chips are also given in Table IV. The maximum clock frequencies of the proposed designs are higher than those of the previous designs. This improvement is partly due to using a better technology and partly due the QLIP architecture. However, the contribution of each of the aforementioned reasons cannot be accurately determined due to different parameters of the employed technologies (feature size, standard cell libraries, etc.) and different synthesis tools employed in the designs. The same argument is valid for the lower power consumptions but, due to the less number of MOS transistors in the QLIP designs, the contribution of the new architecture to the lower power consumption is more significant than the contribution of the small MOS size.

By inserting three pipeline registers in the QLIP design with SFDR of 63.2 dBc, we increased the speed of the DDFS to 1 GHz [23]. The chip area of the proposed pipelined DDFS is 2.6 times more than that of the original design while its power consupmtion is only 1.6 times more than that of the original design. This shows that the proposed design has a perfect architecture for pipelining, which makes it the best choice for high speed DDFS architectures.

Based on the simulation results shown in Table IV, the QLIP method can reduce the complexity of the DDFS architecture due to two effects. The first effect is the reduction of the number of segments required for a target SFDR. With a lower number of segments, the multiplier (the MUX sections in Fig. 5) becomes less complex at the cost of an additional squarer. By using a pre-truncated squarer, its impact on complexity can be mitigated. The other effect is that the structure of the design is kept similar to the linear interpolation method. The combination of these two effects make an efficient architecture. In fact, the name "quasi linear" has been adopted due to this combination.

VI. Conclusion

We have proposed the Quasi-Linear Interpolation method to design DDFS architecture. In this method, we used a combination of a linear and even parabolic polynomial interpolation techniques to approximate the first quarter of a cosine signal. The SFDR upperbounds of the proposed method are evaluated for different numbers of segments per quarter cosine signal and their corresponding polynomial coefficients are evaluated to be used in the final optimization algorithm that generates the final quantized coefficients. The results show the effectiveness of the employed optimization algorithm to obtain the closest possible SFDRs to the SFDR upperbounds.

It has also shown that the VLSI architectures of the proposed designs demonstrate better performance compared to some of the state-of-the-art designs reported in the literature in terms of power consumption, chip area and speed. The architecture of the proposed design is proved to be suitable for pipelining. A DDFS design with the SFDR of 63.2 dBc and 1 GHz clock frequency is designed to show the effectiveness of pipelining in the proposed architecture.

Appendix

To obtain the SFDR upperbound of the QLIP method, we should find the Fourier series of the approximate cosine function whose first quadrant is defined by (5)

$$P(x)=\sum_{n=1}^{\infty} a_n \cos\left(\frac{n\pi}{2}x\right) \tag{13}$$

where $P(x)$ and a_n are the interpolating function and the coefficients of the Fourier series, respectively. The coefficients a_n can be determined by

$$a_n = 2\int_0^1 P(x)\cos\left(\frac{n\pi x}{2}\right)dx \tag{14}$$

where n counts the odd harmonics. Considering the segmentation of the first quadrant, (14) can be transformed to

$$a_n = 2\sum_{k=1}^{s}\int_{x_k}^{x_{k+1}} P_k(x)\cos\left(\frac{n\pi x}{2}\right)dx \tag{15}$$

where s is the number of segments, $P_k(x)$ is the polynomial that approximates the k^{th} segment $x_k \le x \le x_{k+1}$ and $x_k = (k-1)/s$ is the starting point of the k^{th} segment. Based on (5), the interpolating polynomials have two different degrees, thus (15) can be written as

$$\begin{aligned} a_n = {} & 2\sum_{k=1}^{\theta}\int_{x_k}^{x_{k+1}} (c_0^{(k)}+c_2^{(k)}x^2)\cos\left(\frac{n\pi x}{2}\right)dx \\ & +2\sum_{k=\theta+1}^{s}\int_{x_k}^{x_{k+1}} (c_0^{(k)}+c_1^{(k)}x)\cos\left(\frac{n\pi x}{2}\right)dx \end{aligned} \tag{16}$$

where $\theta = 3s/4$. We can simplify (16) to

$$\begin{aligned} a_n = {} & 2\sum_{k=1}^{\theta}(c_0^{(k)}\alpha_{n0}^{(k)}+c_2^{(k)}\alpha_{n2}^{(k)}) \\ & +2\sum_{k=\theta+1}^{s}(c_0^{(k)}\alpha_{n0}^{(k)}+c_1^{(k)}\alpha_{n1}^{(k)}) \end{aligned} \tag{17}$$

where

$$\alpha_{ni}^{(k)} = \int_{x_k}^{x_{k+1}} x^i \sin\left(\frac{n\pi}{2}x\right)dx, \quad 0\le i\le 2. \tag{18}$$

One can easily find the harmonic coefficients a_n by transforming (17) to the following matrix form

$$a_n = 2\sum_{k=1}^{s}\mathbf{B}_k\mathbf{c}_k \tag{19}$$

where $\mathbf{c}_k \in \mathbb{R}^{2\times 1}$, $\mathbf{B}_k \in \mathbb{R}^{N\times 2}$

$$\begin{aligned} \mathbf{c}_k &= [\,c_0^{(k)} \quad c_2^{(k)}\,]^{\mathrm{T}}, \quad 1\le k\le\theta \\ \mathbf{c}_k &= [\,c_0^{(k)} \quad c_1^{(k)}\,]^{\mathrm{T}}, \quad \theta+1\le k\le s \\ \mathbf{B}_k &= [\,\alpha_{n0}^{(k)} \quad \alpha_{n2}^{(k)}\,], \quad 1\le k\le\theta \\ \mathbf{B}_k &= [\,\alpha_{n0}^{(k)} \quad \alpha_{n1}^{(k)}\,], \quad \theta+1\le k\le s \end{aligned} \tag{20}$$

$$n = 1,3,\ldots 2N-1 \tag{21}$$

and N is the number of odd harmonics chosen for the optimization. By using (20) and (21), one can rewrite (19) as a matrix multiplication

$$\mathbf{a} = \mathbf{B}\mathbf{c} \tag{22}$$

where $\mathbf{B}\in\mathbb{R}^{N\times 2s}$, $\mathbf{c}\in\mathbb{R}^{2s\times 1}$, $\mathbf{a}\in\mathbb{R}^{N\times 1}$,

$$\mathbf{B} = 2[\mathbf{B}_1|\mathbf{B}_2|\cdots|\mathbf{B}_s] \tag{23}$$

$$\mathbf{c} = [\mathbf{c}_1^{\mathrm{T}}|\mathbf{c}_2^{\mathrm{T}}|\mathbf{c}_3^{\mathrm{T}}|\cdots|\mathbf{c}_{s-1}^{\mathrm{T}}|\mathbf{c}_s^{\mathrm{T}}]^{\mathrm{T}} \tag{24}$$

and $\mathbf{a}$ is a vector containing the harmonic values of the output signal.

Now we can treat (22) as an equation in which the coefficients of the polynomials i.e., the vector $\mathbf{c}$ is to be found when the spectrum of the output sinusoid is ideal. In the ideal spectrum of a sinusoid, the first harmonic has the unit magnitude and the other harmonics have zero magnitudes. This situation can be represented by

$$\mathbf{a}' = [\,1 \quad 0 \quad \cdots \quad 0\,]^{\mathrm{T}}, \quad \mathbf{a}'\in\mathbb{R}^{N\times 1}. \tag{25}$$

By solving the matrix equation

$$\mathbf{B}\mathbf{c} = \mathbf{a}' \tag{26}$$

one can find the coefficients of the polynomials.

It is shown in [21] that the number of odd harmonics involved in the calculations is always greater than Ms where M is the number of coefficients in the interpolating polynomials i.e., $N > 2s$. Therefore, the matrix (26) is an inconsistent overdetermined equation and it has no solution. However, we can find an

approximate solution to (26) that minimizes the infinity norm of the error vector i.e.,

$$\|\eta(\mathbf{c})\|_\infty = \|\mathbf{a}' - \mathbf{Bc}\|_\infty . \tag{27}$$

This approximate solution is called the Chebyshev minimax solution to (26). It is shown in [21] that this approximate solution gives us the coefficients of the polynomial interpolation corresponding to the SFDR upperbound.

To find the Chebyshev minimax solution to (26), we can define the positive real parameter ρ as the upperbound of the error vector $|\mathbf{a}' - \mathbf{Bc}|$, such that

$$|\mathbf{a}' - \mathbf{Bc}| \leq \rho. \tag{28}$$

The Chebyshev minimax solution of (26) is the vector $\tilde{\mathbf{c}}$ that minimizes the parameter ρ. This is equivalent to the following linear programming problem

$$\begin{cases} \min\limits_{\mathbf{x}} \mathbf{px} \\ \mathbf{Dx} \geqslant \hat{\mathbf{a}} \end{cases}, \quad \mathbf{p} = [\mathbf{0}_{2s} \quad 1], \quad \mathbf{p} \in \mathbb{R}^{1\times(2s+1)} \tag{29}$$

where

$$\begin{gathered} \mathbf{D} \in \mathbb{R}^{2N\times(2s+1)}, \quad \mathbf{x} \in \mathbb{R}^{(2s+1)\times 1}, \quad \hat{\mathbf{a}} \in \mathbb{R}^{2N\times 1} \\ \mathbf{D} = \begin{bmatrix} \mathbf{B} & \mathbf{1}_{N\times 1} \\ -\mathbf{B} & \mathbf{1}_{N\times 1} \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} \mathbf{c} \\ \rho \end{bmatrix}, \quad \hat{\mathbf{a}} = \begin{bmatrix} \mathbf{a}' \\ -\mathbf{a} \end{bmatrix}. \end{gathered} \tag{30}$$

The linear programming problem (29) can be solved numerically by computer packages such as MATLAB. The solution of (29) will be a vector $\tilde{x} = [\tilde{\mathbf{c}}^{\mathrm{T}} | \tilde{\rho}^{\mathrm{T}}]^{\mathrm{T}}$ where $\tilde{\mathbf{c}}$ is the Chebyshev minimax solution to (26) and $\tilde{\rho}$ is its corresponding minimax value (the minimum value of the infinity norm (27)). Then the SFDR upperbound can be found by

$$\mathrm{SFDR}_{\mathrm{upperbound}} = \frac{1-\tilde{\rho}}{\tilde{\rho}}. \tag{31}$$

To find a suitable value for N we can use Theorem 1 of [21]. We can start finding the SFDR upperbound by choosing $N = 2s+1$, solve (29) and find whether the error vector $|\mathbf{a}' - \mathbf{Bc}|$ has $2s+1$ equal minimax error values or not. If $2s+1$ number of minimax values do not occur, N must be increased by one and the entire process must be repeated until the error vector has $2s+1$ equal minimax error values. The obtained solution would be the Chebyshev minimax solution, which determines the SFDR upperbound. It is worth noting that since the matrix $\mathbf{B}$ is not a Haar matrix (the definition of a Haar matrix is given in [21]), the Chebyshev minimax solution is not unique and the solution obtained by the linear programming method would be the solution that generates the sinusoid with the worst SNR. However, the final sinusoid will be obtained by the quantized coefficients that are determined by optimization and it will have a better SNR than the ideal case discussed above (see Figs. 10 and 11).

REFERENCES

[1] J. Vankka, "Methods of mapping from phase sine amplitude in direct digital synthesis," *IEEE Trans. Ultrason., Ferroelectr. Freq. Contr.*, vol. 44, no. 2, pp. 526–534, Mar. 1997.

[2] A. McEwan and S. Collins, "Direct digital frequency synthesis by analogue interpolation," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 53, no. 11, pp. 1294–1298, Nov. 2006.

[3] A. Torosyan and A. N. Willson Jr., "Exact analysis of DDFS spur and SNR due to phase truncation and arbitrary phase-to-amplitude errors," in *Proc. IEEE Int. Freq. Contr. Symp. Expo.*, Aug. 2005, pp. 50–58.

[4] H. T. Nicholas and H. Samueli, "An analysis of the output spectrum of direct digital frequency synthesizers in the presence of phase-accumulator truncation," in *Proc. 41st Ann. Freq. Contr. Symp.*, 1987, pp. 495–502.

[5] F. Curticpean and J. Niittylahti, "Exact analysis of spurious signals in direct digital frequency synthesizers due to phase truncation," *Electron. Lett.*, vol. 39, no. 6, pp. 499–501, Mar. 2003.

[6] D. De Caro, E. Napoli, and A. G. M. Strollo, "Direct digital frequency synthesizers with polynomial hyperfolding technique," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 51, no. 7, pp. 337–344, Jul. 2004.

[7] J. M. P. Langlois and D. Al-Khalili, "Low power direct digital frequency synthesizers in 0.18 mm CMOS," in *Proc. IEEE Custom Integr. Circuits Conf.*, Sep. 2003, pp. 283–286.

[8] A. M. Sodagar and G. R. Lahiji, "A pipelined ROM-less architecture for sine-output direct digital frequency synthesizers using the second-order parabolic approximation," *IEEE Trans. Circuit Syst. II, Analog Digital Signal Process.*, vol. 48, no. 9, pp. 850–857, Sep. 2001.

[9] A. M. Sodagar and G. R. Lahiji, "A pipelined ROM-less architecture for sine-output direct digital frequency synthesizers using the second-order parabolic approximation," *IEEE Trans. Circuit Syst. II, Analog Digital Signal Process.*, vol. 48, no. 9, pp. 850–857, Sep. 2001.

[10] J. M. P. Langlois and D. Al-Khalili, "Novel approach to the design of direct digital frequency synthesizers based on linear interpolation," *IEEE Trans. Circuit Syst. II, Analog Digital Signal Process.*, vol. 50, no. 9, pp. 567–578, Sep. 2003.

[11] G. W. Reitweisner, "Binary arithmetic," *Adv. Comput.*, vol. 1, pp. 232–313, 1960.

[12] D. De Caro and A. G. M. Strollo, "High-performance direct digital frequency synthesizers using piecewise-polynomial approximation," *IEEE Trans. Circuit Syst.*, vol. 52, pp. 324–336, Feb. 2005.

[13] D. De Caro and A. G. M. Strollo, "High-performance direct digital frequency synthesizers in 0.25 mm CMOS using dual-slope approximation," *IEEE J. Solid-State Circuits*, vol. 40, no. 11, pp. 2220–2227, Nov. 2005.

[14] C. C. Wang, Y. L. Tseng, H. C. She, C. C. Li, and R. Hu, "A 13-Bit resolution ROM-less direct digital frequency synthesizer based on a trigonometric quadruple angle formula," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 12, no. 9, pp. 895–900, Sep. 2004.

[15] C. C. Wang, J. M. Huang, Y. L. Tseng, W. J. Lin, and R. Hu, "Phase-adjustable pipelining ROM-less direct digital frequency synthesizer with a 41.66-MHz output frequency," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 53, no. 10, pp. 1143–1147, Oct. 2006.

[16] A. Bellaouar, M. S. O'brecht, A. M. Fahim, and M. I. Elmasry, "Low-power direct digital frequency synthesis for wireless communications," *IEEE J. Solid-State Circuits*, vol. 35, no. 3, pp. 385–390, 2000.

[17] A. Ashrafi and R. Adhami, "Comments on A 13-Bit resolution ROM-less direct digital frequency synthesizer based on a trigonometric quadruple angle formula," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 9, pp. 1096–1098, Sep. 2005.

[18] A. Ashrafi, Z. Pan, R. Adhami, and B. E. Wells, "A novel ROM-less direct digital frequency synthesizer based on Chebyshev polynomial interpolation," in *Proc. 36th Southeastern Symp. Syst. Theory, SSST'04*, Mar. 2004, pp. 393–397.

[19] A. Ashrafi, R. Adhami, L. Joiner, and P. Kaveh, "Arbitrary waveform DDFS utilizing Chebyshev polynomials interpolation," *IEEE Trans. Circuit Syst. I, Reg. Papers*, vol. 51, pp. 1468–1475, Aug. 2004.

[20] L. S. J. Chimakurthy, M. Ghosh, F. F. Dai, and R. C. Jaeger, "A novel DDS using nonlinear ROM addressing with improved compression ratio and quantization noise," *IEEE Trans. Ultrason., Ferroelectr., Freq. Contr.*, vol. 53, no. 2, pp. 274–283, Feb. 2006.

[21] A. Ashrafi and R. Adhami, "Theoretical upperbound of the spurious free dynamic range in direct digital frequency synthesizers realized by polynomial interpolation methods," *IEEE Trans. Circuit Syst. I, Reg. Papers*, vol. 54, no. 10, pp. 2252–2261, Oct. 2007.

[22] A. Ashrafi and R. Adhami, "A direct digital frequency synthesizer utilizing quasi-linear interpolation method," in *Proc. IEEE 37th Southeastern Symp. System Theory*, Mar. 2005, pp. 114–118.

[23] A. Ashrafi, A. Milenkovic, and R. Adhami, "A 1 GHz direct digital frequency synthesizer based on the quasi-linear interpolation method," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2007, pp. 2766–2769.

[24] J. M. P. Langlois and D. Al-Khalili, "Phase to sinusoid amplitude conversion techniques for direct digital frequency synthesis," *Proc. IEE Circuit Devices Syst.*, vol. 151, no. 6, pp. 519–528, Dec. 2004.

[25] G. C. Gielis, R. van de Plassche, and J. van Valburg, "A 540-MHz 10-b polar-to-cartesian coverter," *IEEE J. Solid-State Circuits*, vol. 26, no. 11, pp. 1645–1650, Nov. 1991.
[26] A. Madisetti, A. Y. Kwentus, and A. N. Willson, "A 100-MHz, 16-b, direct digital frequency synthesizer with a 100-dBc spurious-free dynamic range," *IEEE J. Solid-State Circuits*, vol. 34, no. 8, pp. 1034–1043, Aug. 1999.
[27] Y. Song and B. Kim, "Quadrature direct digital frequency synthesizers using interpolation-based angle rotation," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 12, no. 7, pp. 701–710, Jul. 2004.
[28] F. Cardells-Tomoro and J. Valls-Coquillat, "Optimization of direct digital frequency synthesizers based on CORDIC," *Electron. Lett.*, vol. 37, no. 21, pp. 1278–1280, Oct. 2001.
[29] F. Curticapean, K. I. palomaki, and J. Niittylahti, "Quadrature direct digital frequency synthesizer using angle rotation algorithm," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS'03)*, May 2003, vol. 2, pp. 81–84.
[30] A. G. M. Strollo, D. De Caro, and N. Petra, "A 630 MHz, 76 mW direct digital frequency synthesizer using enhanced ROM compression technique," *IEEE J. Solid-State Circuits*, vol. 42, pp. 350–360, Feb. 2007.
[31] D. De Caro, N. Petra, and A. G. M. Strollo, "Reducing lookup-table size in direct digital frequency synthesizers using optimized multipartite table method," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 7, pp. 2116–2127, Aug. 2008.
[32] R. K. Kolagotla, W. R. Griesbach, and H. R. Srinivas, "VLSI implementation of 350 MHz 0.35 mm 8 bit merger squarer," *Electron. Lett.*, vol. 34, no. 1, pp. 47–48, Jan. 1998.
[33] A. Ashrafi and S. Thota, "A novel pre-truncated fixed-width digital squarer," in *Proc. IEEE Midwest Symp. Circuits Syst. MWSCAS-2008*, Aug. 2008, pp. 959–961.
[34] A. G. M. Strollo, N. Petra, and D. De Caro, "Dual-tree error compensation for high performance fixed-width multipliers," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 52, no. 8, Aug. 2005.

Ashkan Ashrafi (M'98–S'02–M'06–SM'08) was born in Tehran, Iran, in 1969. He received the B.Sc. and M.Sc. degrees in electronics engineering from K. N. Toosi University of Technology, Tehran, Iran, and the M.S.E. and Ph.D. degrees in electrical engineering from the University of Alabama, Huntsville, in 1991, 1995, 2003, and 2006, respectively (all with honors).

He served as a faculty member and Associate Chair of Electrical Engineering at Azad University, Tehran Central Branch, Tehran, Iran from 1997 to 2001 where he was also the IEEE student branch counselor. He also served as a Visiting Assistant Professor at the Electrical and Computer Engineering Department of the University of Alabama in Huntsville from 2006 to 2007. He is currently an Assistant Professor at the Electrical and Computer Engineering Department of San Diego State University, San Diego, CA where he is the director of the Signal Processing Research Group. His research interests are circuits and systems, digital signal processing and direct digital frequency synthesizers.

Dr. Ashrafi is the recipient of the 2005 Iliana Martin Chittur Outstanding Graduate Student Award and the 2005 National Engineer's Week Outstanding Graduate Student Award from the College of Engineering, University of Alabama in Huntsville. He has published more than 25 technical papers in well known journals and conferences. He has served as a reviewer for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I: REGULAR PAPERS, IEEE TRANSACTIONS ON AUTOMATIC CONTROL, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, IEEE TRANSACTIONS ON ULTRASONIC FERROELECTRIC AND FREQUENCY CONTROL, IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE and IEEE SIGNAL PROCESSING LETTERS. He is also a member of Phi Kappa Phi and Eta Kappa Nu.

Reza Adhami (M'83–SM'01) received the B.S.E., M.S.E., and Ph.D. degrees in electrical engineering, in 1980, 1981, and 1985, respectively, from the University of Alabama , Huntsville.

Currently, he is Professor and Chair of Electrical and Computer Engineering at the University of Alabama in Huntsville. His research interests include VLSI, applications of biometrics in information assurance, communication systems, and signal processing. As a faculty member, he has successfully supervised numerous Masters theses and Ph.D. dissertations. He has published 2 books and more than 100 articles in different journals and conferences.

Dr. Adhami has been recognized by different organizations for his contributions to high quality research, education, and leadership. He is a recipient of several national and regional awards including the Maurice Simpson Award for excellence in technical publication, IEEE Region III Outstanding Educator Award, and IEEE USA Professional Achievement Award for leadership in building collaborative relationships between industry and academia. He has multiple years of industrial experience during which he has designed and implemented numerous hardware/software systems.

Aleksandar Milenković (M'97) received the Dipl. Ing., M.Sc., and Ph.D. degrees in computer engineering and science from the University of Belgrade, Serbia, in 1994, 1997, and 1999.

He is an Associate Professor of Electrical and Computer Engineering at the University of Alabama in Huntsville. He currently directs the LaCASA Laboratory (http://www.ece.uah.edu/ lacasa/). His research interests include advanced architectures for the next generation computing devices, low-power VLSI, reconfigurable computing, embedded systems, and wireless sensor networks.