A ROM-less Direct Digital Frequency Synthesizer Based on the 16-Segment Parabolic Polynomial Interpolation[§]

Jian-Ming Huang, Chia-Chuan Lee,[‡] and Chua-Chin Wang[†], *Senior Member, IEEE,*
Department of Electrical Engineering
National Sun Yat-Sen University
Kaohsiung, Taiwan 80424
Email: ccwang@ee.nsysu.edu.tw

***Abstract*— This paper presents a novel architecture for direct digital frequency synthesizer (DDFS) based on a modified parabolic polynomial interpolation method. A 16-segment parabolic polynomial interpolation is adopted to replace conventional ROM-based phase-to-amplitude conversion method. Besides, the proposed parabolic polynomial interpolation is realized in a multiplier-less fashion such that the speed can be significantly improved. The proposed DDFS is implemented in a standard 0.13 μm cell-based technology. The maximum clock rate is 227 MHz, and the core area is 0.25 mm^2. The simulation result shows that the spurious free dynamic range (SFDR) is 117 dBc.**

I. Introduction

Frequency synthesizer is an essential part for communication systems. Conventionally, phase-locked loops (PLLs) are usually adopted to synthesize the sinusoid. However, PLL-based frequency synthesizers suffer from an inherent inability to simultaneously provide both fast frequency switching and high spectral purity [1]. This imperfection makes PLL unsuitable for modern wireless communication systems, which usually require fast frequency switching. The direct digital frequency synthesizer (DDFS) has been considered as a better alternative other than PLL-based frequency synthesizer, because it can realize the fast frequency switching while keeping excellent spectral purity.

Fig. 1 shows the conventional DDFS architecture. The digital phase information are converted into samples of sine amplitude by a ROM look-up table, and then samples are converted into an analog signal by Digital-to-Analog Converter (DAC). However, this architecture demands a very large ROM as the storage of sinusoid amplitude and consequently suffers from the inherent drawback of large power dissipation, large chip area, and slow speed. Even though the ROM size can be significantly reduced by truncating the output of the phase accumulator, the added spurious noise will degrade the spectral purity. On the other hand, many researches have been reported on the designs of ROM-less DDFS, such as [2], [3], [5]-[8]. The ROM-less DDFSs employ different algorithm in stead of ROM to realize the phase-to-sine mapper.

Many prior ROM-less DDFS works developed sophisticated polynomials to carry out the phase-to-sine mapper. However, any method that is based on high-order polynomials will be hard to meet the speed requirement of modern wireless communication applications due to the massive complexity. Considering the probability of hardware realization, any polynomial whose order is large than three may be inefficient to be implemented. On the other hand, the recently reported DDFSs based on 2nd-order polynomials are still difficult to achieve high performance without sacrificing the speed. This paper investigates and proposes an improved DDFS architecture based on a modified parabolic polynomial. The proposed DDFS can simultaneously achieve an excellent spectral purity and a satisfactory speed.

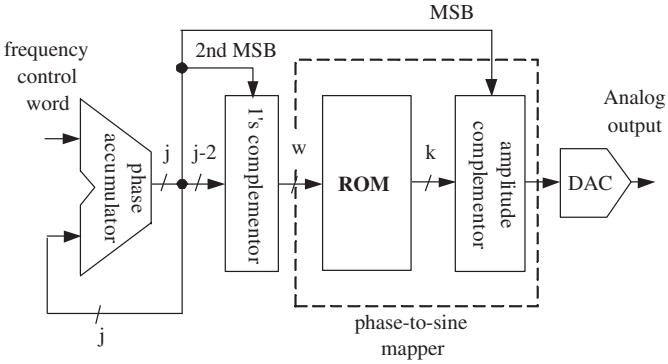

Fig. 1. Block diagram of the conventional DDFS

II. DDFS Architecture

A. The Modified Parabolic Polynomial Interpolation

To implement the phase-to-sine mapping function by a 2nd-order polynomial, the parabolic polynomial could be the most convenient choice. we can divide the first quadrant of the cosine signal ($\cos\theta$, $0 \leq \theta < \pi/2$) into $M = 2^m$ segments, and each segment can be approximated by the parabolic

[§]This research was partially supported by National Science Council under grant NSC95-2622-E-110-012-CC3.

[†] the contact author

[‡]C.-C. Lee is with the Institute of Communications Engineering, National Sun Yat-Sen University.

polynomial, which is

$$y(x) = a_i x^2 + c_i, \quad i = 1 \sim M. \tag{1}$$

The parameter a_i and c_i can be derived from the lest square method. Although the SFDR of this DDFS will be increased as the increas of M, the large M will lead to divergence when performing least square method.

To improve the performance of the DDFS based on the parabolic polynomial, [2] proposed a quasi-linear interpolation method (QLIP). It was based on the observation that a cosine signal is close to a parabola at $\theta = 0$, but it is closer to a straight line at $\theta = \pi/2$. The QLIP method can be expressed as follows:

$$y(x) = \begin{cases} a_i x^2 + c_i, & 1 \le i \le 3M/4 \\ a_i x + c_i, & 3M/4 + 1 < i \le M. \end{cases} \tag{2}$$

According to [2], the QLIP method has 6 dBc improvement over the parabolic polynomial method.

However, both the QLIP and the parabolic polynomial method have the same difficulty in fitting the curvature of the sinusoid. The curvature of a given curve $y = f(x)$ can be expressed as follows:

$$k(x) = \frac{y''}{(1 + y'^2)^{3/2}}, \tag{3}$$

where $k(x)$ is the curvature at x. While the parabolic polynomial is adopted, its 1st-order derivative ($y' = 2a_i x$) is proportional to the 2nd-derivative ($y' = 2a_i$). Thus, we can use only one parameter a_i to determine both the slope and the curvature of the curve, which will lead to difficulties in the fitting process. In order to improve the adjustability of the parabolic polynomial, we introduce a 1st-order term into Eq. (1), and Eq. (1) can be modified as follows:

$$y(x) = a_i(x - d_i)^2 + c_i, \quad i = 1 \sim M \tag{4}$$

where d_i is the displacement factor. When Eq. (3) is used to carry out the phase-to-sine mapping function, there are two parameters that can be used to adjust the characteristics of the curve in each segment. To evaluate the performance of the modified parabolic polynomial, we design different DDFSs based on these three methods and compare their performance. The coefficients of these DDFSs are derived by the least square method of MATLAB, where the output of these DDFSs are not quantized. Fig. 2 depicts the error between the ideal cosine function and the parabolic polynomial method for $M = 4$, where the maximum error is 7×10^{-3}. Fig. 3 and Fig. 4 shows the computed error of the QLIP method and the proposed method, where the maximum error are 3.77×10^{-3} and 4.8×10^{-4}, respectively.

Table I shows the SFDR comparison between the QLIP method and the proposed method under different number of segments. The SFDR of the proposed method for $M = 16$ has almost 27 dBc improvement over that reported in [2]. As to the computation complexity, Eq. (4) need only one more addition/subtraction than Eq. (2). However, this overhead is acceptable compared to the significant improvement of the

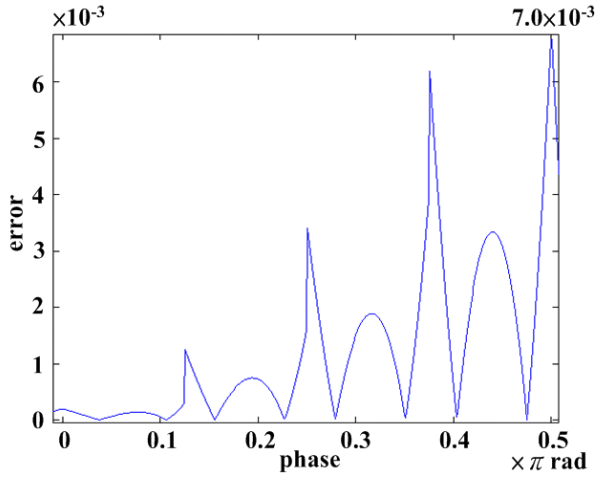

Fig. 2. The error between the ideal cosine function and the parabolic polynomial method ($M = 4$)

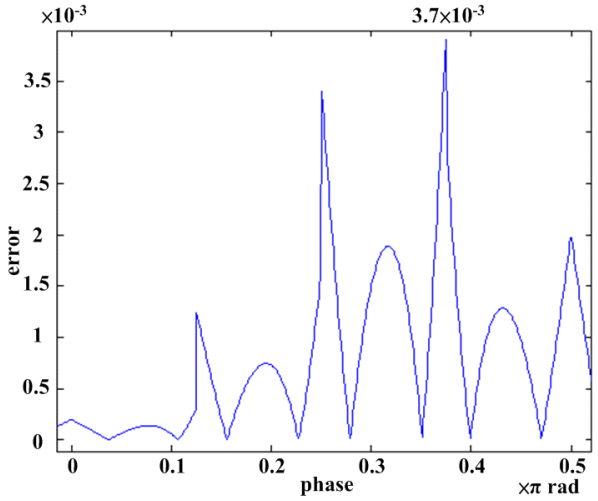

Fig. 3. The error between the ideal cosine function and the QLIP method ($M = 4$)

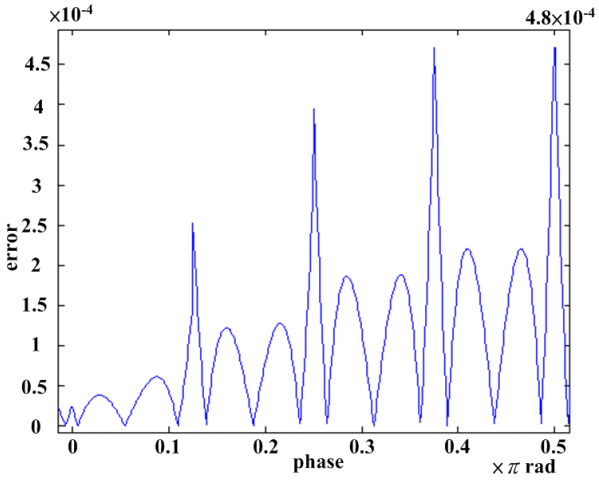

Fig. 4. The error between the ideal cosine function and the proposed method ($M = 4$)

spectral purity. Since the SFDR of 100 dBc can meet the requirement of most modern wireless communication application, we adopt the 16-segment modified parabolic polynomial method to realize the proposed DDFS.

TABLE I

THE SFDR OF THE QLIP METHOD AND THE PROPOSED METHOD UNDER DIFFERENT M

	QLIP method [2]	proposed method
$M = 4$	65 dBc	81 dBc
$M = 8$	78 dBc	99 dBc
$M = 16$	90 dBc	117.4 dBc

B. Hardware Implementation

To consider the hardware implementation of the DDFS based by the proposed modified parabolic polynomial, we must first choose a suitable length of the phase word. The reason is that the phase word-length determines the performance and the complexity of the phase-to-sine mapper. According to [4], the worst case SFDR for a DDFS with a perfect phase-to-sine mapper (unquantized output) can be expressed as follows:

$$\text{SFDR} < 20 \cdot \log 2^W, \tag{5}$$

where W is the phase word-length. Since the simulation result shows the SFDR of the proposed methos for $M = 16$ is 117 dBc, the phase word-length is chosen to be 20 bits.

Given the phase word-length of 20 bits, such a massive phase information will increase the hardware complexity and slow down the operation speed. In order to enhance the speed, the multiplication in Eq. (4) can be modified to reduce the complexity. The coefficient a_i is manipulated as a 15-digit binary sequence, and therefore the multiplication can be replaced by the shift-and-add operation (Ex. $0.75 \cdot x = 2^{-1} \cdot x + 2^{-2} \cdot x$). Fig. 5 shows the block diagram of the proposed DDFS. The frequency control word (FCW) is set as 32-bit to obtain a fine frequency tuning range. The output of the squarer is shifted and selected by Mux1 to Mux15. The symbol a_{i-j} in the Fig. 5 represents the j-th digit of the sequence for i-th segment. The summation of the total 16 multiplexer outputs is realized by a 4-level adder tree. To reduce the latency caused by the adder tree, the summation is implemented by a 4-stage pipeline.

III. SIMULATION AND IMPLEMENTATION

The proposed DDFS prototype is carried out by TSMC (Taiwan Semiconductor Manufacturing Company) 0.13 μm CMOS technology to verify the performance. All of the process corners : [0^{o}C, +100^{o}C], and (SS, TT, FF) models, are simulated. Fig. 6 shows the layout of the proposed DDFS prototype. Fig. 7 shows the simulation result of the SFDR, which is 117 dBc. The specifications of the proposed prototype is summarized in Table II. By adopting the pipeline and multiplier-less design, the maximum clock rage is up to 227 MHz. Table III presents the comparison between the proposed DDFS and the recent ROM-less DDFS. The comparison shows that the proposed DDFS has the largest SFDR while maintaining the satisfactory speed and energy efficiency.

TABLE II

SPECIFICATIONS OF THE PROPOSED DDFS

Technology	0.13 μm CMOS process
Power supply	1.2 V
Frequency control word (FCW)	32 bits
Phase word	20 bits
Frequency tuning range	0.053 Hz
SFDR	117 dBc
Max. clock rate	227 MHz
Power dissapation	79 mW @ 227 MHz
Area	2.015 mm^2 (whole chip)
	0.33 mm^2 (core area)

TABLE III

COMPARISON

	[5]	[6]	[7]	[8]	ours
Process (μ)	0.5	0.25	0.18	0.28	0.13
SFDR (dBc)	91.51	80	84	110	117
Phase word length (bit)	16	n/a	n/a	n/a	20
Energy (mW/MHz)	0.56	0.127	0.6	0.4	0.35
Clock (MHz)	106	600	150	250	227
year	2005	2005	2003	2003	2007

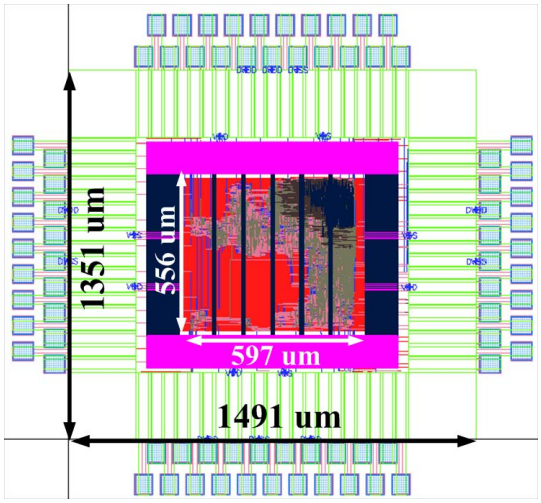

Fig. 6. The layout view of the proposed DDFS

IV. CONCLUSION

A modified parabolic polynomial to realize the phase-to-sine mapping function of a DDFS is proposed in this paper. The proposed method can achieve a significant improvement in SFDR compared to the QLIP method [2]. A 16-segment parabolic polynomial is employed to implement a ROM-less DDFS. The logic operation of the proposed DDFS is manipulated to achieve a multiplier-less design. The pipeline design is adopted to further reduce the latency in signal processing. The proposed DDFS achieves a SFDR of 117dBc, and the maximum clock rate is 227 MHz.

ACKNOWLEDGEMENT

The authors would like to express their deepest gratefulness to CIC (Chip Implementation Center) of NAPL (National

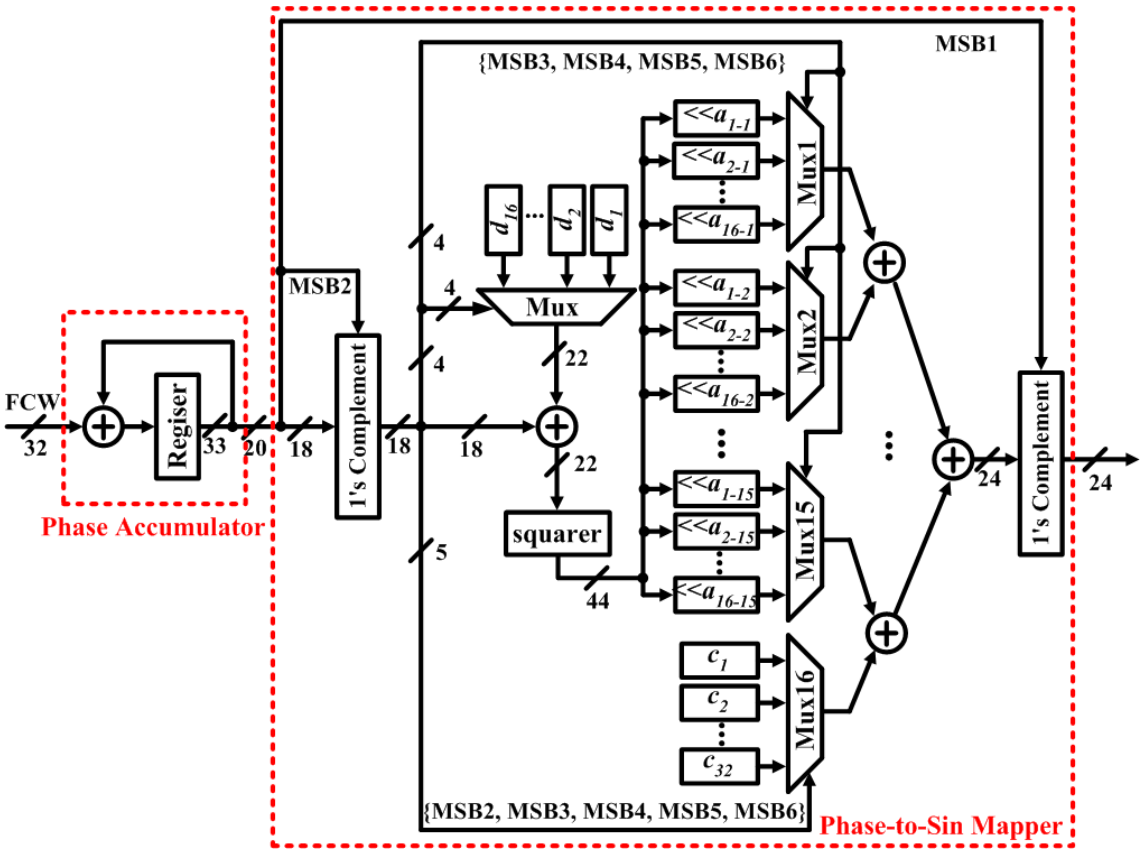

Fig. 5. The block diagram of the proposed DDFS

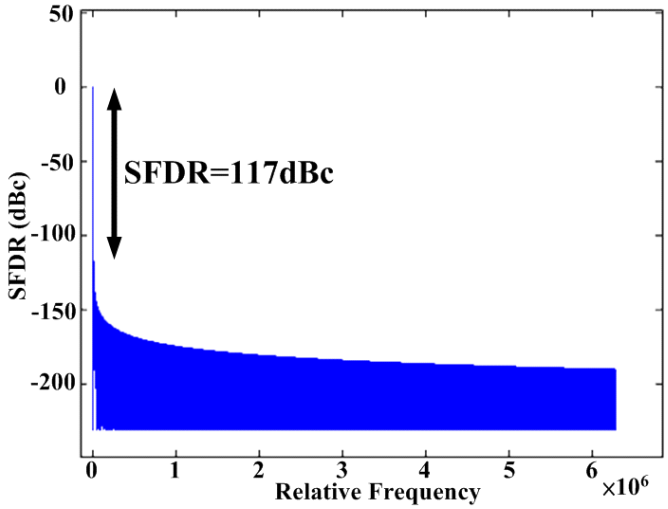

Fig. 7. The plot of the simulated SFDR

Applied Research Laboratories), Taiwan, for their thoughtful chip fabrication service. The authors also like to thank "Aim for Top University Plan" project of NSYSU and MOE, Taiwan, for partially supporting this investigation.

REFERENCES

[1] A. Yamagishi, M. Ishikawa, T. Tsukahara, and S. Date, "A 2-V, 2-GHz Low-Power Direct Digital Frequency Synthesizer Chip-set for Wireless Communication," *IEEE J. of Solid-State Circuits*, vol. 33, no. 2, pp. 210-217, Feb. 1998.

[2] A. Ashrafi, and R. Adhami,"A Direct Digital Frequency Synthesizer Utilizing Quasi-Linear Interpolation Method," in Proc. of 37th IEEE Southeastern Symposium on System Theory, pp. 114-118, Mar. 2005.

[3] A. Ashrafi, and R. Adhami, "An Optimized Direct Digital Frequency Synthesizer Based on Even Fourth Order Polynomial Interpolation," in Proc. of 38th IEEE Southeastern Symposium on System Theory, pp. 109-113, Mar. 2006.

[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," 41st Annual Frequency Control Symposium, pp. 495-502, 1987.

[5] H. Jafari, A. Ayatollahi, and S. Mirzakuchaki, "A Low Power, High SFDR, ROM-Less Direct Digital Frequency Synthesize," 2005 IEEE Conf. on Electron Devices and Solid-State Circuits, pp. 829 - 832, Dec. 2005.

[6] D. De Caro, and A. G. M. Strollo, "High-Performance Direct Digital Frequency Synthesizers in 0.25 μm CMOS Using Dual-Slope Approximation," IEEE J. of solid-state Circuits, vol. 40, no 11, pp. 2220 - 2227, Nov. 2005.

[7] J. M. P. Langlois, and D. Al-Khalili, "Low Power Direct Digital Frequency Synthesizers in 0.18 um CMOS", in Proc. of IEEE Custom Integrated Circuits Conference, pp. 283-286, Sept. 2003.

[8] Y. Song, and B. Kim, "A 250 MHz Direct Digital Frequency Synthesizer with Delta-Sigma Noise Shaping," 2003 IEEE Int. Solid-State Circuits Conf. (ISSCC 2003), vol. 1, pp. 472-509, Feb. 2003.