



## Direct Digital Frequency Synthesis Based on a Two-Level Table-Lookup Scheme

SAU-GEE CHEN, JEN-CHUAN CHIH AND JUN-YI CHOU

*Department of Electronics Engineering and Institute of Electronics, National Chiao Tung University,  
1001 Ta Hsueh Road, Hsinchu, Taiwan, Republic of China*

*Received: 18 December 2003; Revised: 15 April 2005; Accepted: 24 April 2006*

**Abstract.** In this work, a new direct digital frequency synthesizer (DDFS) is proposed, which is based on a new two-level table-lookup (TLTL) scheme combined with Taylor's expansion. This method only needs a lookup-table size of total  $n \times 2^{n/4+1} + (n/4 - 2) \times 2^{n/4}$  bits, one  $(n + 1) \times 3n/4$ -bit multiplier, one  $n \times 3n/4$ -bit multiplier and two additional smaller multipliers, to generate both sine and cosine values (where  $n$  is the output precision). Compared with several notable DDFS's, the new design has a smaller lookup-table size and higher SFDR (Spurious Free Dynamic Range) for high-precision output cases, at comparable multiplier and adder complexities. The DDFS is verified by FPGA and EDA tools using Synopsys Design Analyzer and UMC 0.25  $\mu\text{m}$  cell library, assuming 16-bit output precision. The designed 16-bit DDFS has a small gate count of 2,797, and a high SFDR of 110 dBc.

**Keywords:** direct digital frequency synthesizer, DDFS algorithm, two-level table lookup scheme

### 1. Introduction

The direct digital frequency synthesizer (DDFS) [1] offers fast switching speed, high resolution, excellent phase noise, transient-free frequency changes, small size, and the reliability of digital designs, compared with analog frequency synthesizers. Since a DDFS can be flexibly programmed over a wide range of frequency band, it plays a crucial role in numerous digital and wireless communication applications [2], [3], such as wireless local area network (WLAN), 3G communication, software radio, and so on. Therefore, it is highly desirable to design a DDFS with high speed, low complexity, wide frequency range and high accuracy.

A conventional DDFS structure can be separated into three parts, including phase accumulator, sine/

cosine generator, and DAC (digital-analog-converter). The phase accumulator performs digital phase accumulation and provides phase inputs to the sine/cosine generator block for digital sine/cosine wave generation with a particular frequency. Finally, the DAC converts the digital waveforms to analog waveforms.

DDFS's can be categorized as recursive [4, 5], and non-recursive types [6–20]. The recursive DDFS's are notorious for their error-propagation problem, due to their error feedback nature, although the algorithm [4] has the advantage of only needing two multiplication and two addition operations to generate one sine and one cosine values. Also, they are hard to pipeline, because only when the current function value is obtained, can the next function value be calculated. A solution to reduce error-

propagation is presented in [5]. However, it needs four multiplication and four addition operations per output data sample. Besides, its output SFDR performance is poorer than those existing designs with the same internal data path word length. In addition, the recursive DDFS's [4, 5] are only suitable for the application of fixed frequency control word  $\theta$  and zero initial angle value. To make  $\theta$  flexible and adjustable, they need a very large table to store a wide range of  $\cos\theta$  value.

This paper will focus on the design of non-recursive DDFS's. In contrast to the recursive-type DDFS's, the non-recursive DDFS's do not have the error-propagation problem. Some key non-recursive DDFS's are discussed as follows. The direct table-lookup DDFS needs a large table size of  $2^n/8$  words to store all the sine/cosine values. Therefore, they are only feasible for short word lengths with low SFDR's. For the consideration of smaller table sizes, the bipartite table-lookup method [6] can be applied to DDFS designs. Nevertheless, the reduction is not significant enough. By simplifying the trigonometric equations, the DDFS design [7] reduces ROM size to only about 1/236 of conventional ROM-based designs, for the 12-bit output case. The optimized linear interpolation method in [9] only needs two multiplication and addition operations to generate both sine and cosine functions. Nonetheless, it needs a large lookup table of about  $2^{n/3}$  words to store the segment slope coefficients and initial segment amplitudes for the optimized segmented linear interpolation equations. In addition, the design realizes the required ROM table and multipliers altogether with somewhat complicated multiplexers and adders.

The CORDIC (COordinate Rotation Digital Computation) algorithm is considered to be effective in realizing a DDFS [10, 11]. Generally, the CORDIC-based DDFS's have the disadvantages of slow sequential operations and overhead of the scale factor compensation. The CORDIC DDFS in [11] is considered a very efficient design, which combines CORDIC algorithm with a table-lookup scheme. It has the advantages of pipelined operation, fast decision of rotation directions, and is free of scale factor compensation operations normally required by CORDIC algorithms. Still, it needs a ROM table of size  $2^{n/3}$  to store some predetermined function values. Despite its efficiency, the pipelined CORDIC-based DDFS has the problem of slow switching speed, due to inherent pipelined latency. Therefore, in the

latter performance comparison, we exclude the CORDIC-based DDFS's. Recently, we proposed a DDFS design [20], which has a similar pipelined structure and speed performance to that of [11]. However, the DDFS [20] has a smaller ROM table size of  $2^{n/4}$  than  $2^{n/3}$  of [11].

The polynomial-based DDFS's [13] does not need any coefficient table but has high hardware complexities. In [14], by using a CSD (Canonical Signed Digit) hyper-folding technique, the polynomial-based DDFS achieves a low-complexity design result. Still, its SFDR performance is poor, because the hyper-folding technique sacrifices the output precision. The DDFS's [15–19] combine polynomial approximation with table-lookup operations. They involve multiplication operations, in addition to considerable table sizes. Among those designs, DDFS of [18] has a smaller table size and computational complexity. However, for high output SFDR's, it has the trouble of large table size and high computational complexity. The DDFS of [15] has a small table size, but low SFDR and long switching delay.

Here, based on our recent preliminary design [12], we would like to propose a new non-recursive polynomial-based DDFS. To reduce both table size and computational complexity of a polynomial-based DDFS, by combining Taylor's expansion, trigonometric equations and Karatsuba algorithm [21] we develop a new two-level table-lookup scheme. As a result, the new DDFS achieve a good performance with small table size and high SFDR, at a comparable computational complexity to the existing designs.

We will propose the new DDFS algorithm and its architecture realization in Section 2. Performance evaluation and comparison results will be detailed in Section 3. Section 4 discusses its verification and simulation by EDA tools, followed by the conclusion in Section 5.

## 2. New DDFS Algorithm and Architecture Based on a Two-Level Table Lookup Scheme

The new algorithm starts with a polynomial approximation. The approximation polynomial can be Taylor's expansion, or Chebyshev expansion, or a heuristic one. Here for coefficient simplicity and for low-complexity consideration, we adhere to Taylor's expansion. Normally, computation of the expansion polynomial requires considerable multiplication

count. To reduce the complexity, one often subdivides the expansion region into several subintervals. In turns, each subinterval carries out its own function approximation by using a lower-degree polynomial, with fewer multiplications than the case without subdividing the expansion region. However, the reduction of multiplication is at the cost of extra table size to store the polynomial coefficients and the function value of the expansion point for each subinterval. To reduce table sizes, we will derive a two-level table-lookup algorithm as follows.

Let  $\theta = \alpha + \beta$ , where  $\alpha$  and  $\beta$  are the most significant and the remaining less significant parts, respectively, then  $\cos \theta = \cos(\alpha + \beta)$  and  $\sin \theta = \sin(\alpha + \beta)$  can be put in matrix form as:

$$\begin{bmatrix} \cos(\alpha + \beta) \\ \sin(\alpha + \beta) \end{bmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} \cos \beta \\ \sin \beta \end{bmatrix} \quad (2-1)$$

This equation suggests that we need four multiplications to solve the results once we know  $\cos \alpha$  and  $\sin \alpha$  from table-lookup. However, it is well known [21] that the number of multiplications can be reduced to three as defined below:

$$p_1 = \cos \alpha \cos \beta \quad (2-2a)$$

$$p_2 = \sin \alpha \sin \beta \quad (2-2b)$$

$$p_3 = (\cos \alpha + \sin \alpha)(\cos \beta + \sin \beta) \quad (2-2c)$$

Then Eq. (2-1) can be reduced to:

$$\cos \theta = \cos(\alpha + \beta) = p_1 - p_2 \quad (2-3a)$$

$$\sin \theta = \sin(\alpha + \beta) = p_3 - p_1 - p_2 \quad (2-3b)$$

Here  $\cos \alpha$  and  $\sin \alpha$  are obtained from table-lookup, together with some simple calculation. For example, without loss of generality, let  $\theta$  be represented by  $n$  fractional bits and  $\pi/4 \geq \theta \geq 0$ . If  $\alpha$  is the  $n/4$  MSBs of  $\theta$ , then  $\beta$  is the remaining  $3n/4$  bits. Furthermore, let's expand  $\cos \beta$  and  $\sin \beta$  at  $\gamma$ , which is the  $n/4$  MSBs of  $\beta$ , then  $\cos \beta$  and  $\sin \beta$  can be written as:

$$\cos \beta = \sum_{i=0}^{\infty} \left[ \left( \frac{1}{i!} \right) (\beta - \gamma)^i (\cos \gamma)^{(i)} \right] \quad (2-4a)$$

$$\sin \beta = \sum_{i=0}^{\infty} \left[ \left( \frac{1}{i!} \right) (\beta - \gamma)^i (\sin \gamma)^{(i)} \right] \quad (2-4b)$$

where  $(\cos \gamma)^{(i)}$  and  $(\sin \gamma)^{(i)}$  are the  $i$ th derivatives of  $\cos \gamma$  and  $\sin \gamma$ , respectively. Eqs. (2-4a) and (2-4b) can be rewritten as:

$$\cos \beta = \cos \gamma - (\beta - \gamma) \sin \gamma + \varepsilon_{c\beta} \quad (2-5a)$$

$$\sin \beta = \sin \gamma + (\beta - \gamma) \cos \gamma + \varepsilon_{s\beta} \quad (2-5b)$$

The error terms  $\varepsilon_{c\beta}$  and  $\varepsilon_{s\beta}$  are defined below and can be shown confined to the following ranges.

$$-2^{-(n+1)} < \varepsilon_{c\beta} \equiv \sum_{i=2}^{\infty} \left[ \left( \frac{1}{i!} \right) (\beta - \gamma)^i (\cos \gamma)^{(i)} \right] \leq 0$$

$$-2^{-(5n/4+1)} < \varepsilon_{s\beta} \equiv \sum_{i=2}^{\infty} \left[ \left( \frac{1}{i!} \right) (\beta - \gamma)^i (\sin \gamma)^{(i)} \right] \leq 0$$

Similarly,  $\cos \gamma$  and  $\sin \gamma$  can be expanded as:

$$\cos \gamma = \sum_{j=0}^{\infty} \left[ (-1)^j \left( \frac{\gamma^{2j}}{(2j)!} \right) \right] = \left( 1 - \frac{\gamma^2}{2!} \right) + \varepsilon_{c\gamma} \quad (2-6a)$$

$$\sin \gamma = \sum_{j=0}^{\infty} \left[ (-1)^j \left( \frac{\gamma^{(2j+1)}}{(2j+1)!} \right) \right] = \left( \gamma - \frac{\gamma^3}{3!} \right) + \varepsilon_{s\gamma} \quad (2-6b)$$

The error terms  $\varepsilon_{c\gamma}$  and  $\varepsilon_{s\gamma}$  can be shown confined to the following ranges.

$$0 \leq \varepsilon_{c\gamma} \equiv \sum_{j=2}^{\infty} \left[ (-1)^j \left( \frac{\gamma^{2j}}{(2j)!} \right) \right] < 2^{-(n+4)}$$

$$0 \leq \varepsilon_{s\gamma} \equiv \sum_{j=2}^{\infty} \left[ (-1)^j \left( \frac{\gamma^{(2j+1)}}{(2j+1)!} \right) \right] < 2^{-(5n/4+6)}$$

By combining Eqs. (2-6a) and (2-6b), one can rewrite Eqs. (2-5a) and (2-5b) as:

$$\begin{aligned} \cos \beta = & \left( 1 - \beta\gamma + \frac{\gamma^2}{2} \right) \\ & + \left[ (\beta - \gamma) \left( \frac{\gamma^3}{6} - \varepsilon_{s\gamma} \right) + \varepsilon_{c\gamma} + \varepsilon_{c\beta} \right] \end{aligned} \quad (2-7a)$$

$$\sin \beta = \left( \beta - \frac{\gamma^3}{6} \right) + \left[ (\beta - \gamma) \left( \epsilon_{c\gamma} - \frac{\gamma^2}{2} \right) + \epsilon_{s\gamma} + \epsilon_{s\beta} \right] \quad (2-7b)$$

where

$$\left| (\beta - \gamma) \left( \frac{\gamma^3}{6} - \epsilon_{s\gamma} \right) + \epsilon_{c\gamma} + \epsilon_{c\beta} \right| < 2^{-(n+1)}$$

and

$$\left| (\beta - \gamma) \left( \epsilon_{c\gamma} - \frac{\gamma^2}{2} \right) + \epsilon_{s\gamma} + \epsilon_{s\beta} \right| < 2^{-(n+1)} + 2^{-(5n/4+1)}$$

Therefore, from Eqs. (2-7a) and (2-7b), if  $n$ -bit output precision (for the fractional part) is desired,  $\cos \beta$  and  $\sin \beta$  can be approximated as:

$$\cos \beta \approx 1 - \beta\gamma + \frac{\gamma^2}{2} = 1 - \left( \beta - \frac{\gamma}{2} \right) \gamma \quad (2-8a)$$

$$\sin \beta \approx \beta - \frac{\gamma^3}{6} \quad (2-8b)$$

Equation (2-8a) needs one small  $3n/4$ -bit by  $n/4$ -bit multiplication for  $(\beta - \gamma/2)\gamma$ . In Eq. (2-8b),  $\gamma^3/6$  values can be stored in a  $(n/4 - 2) \times 2^{n/4}$ -bit lookup table, addressed by  $n/4$  bits. Note that,  $\gamma^3/6$  is very close to zero, which makes up a very small ROM table, because  $(3n/4 + 2)$  MSBs of  $\gamma^3/6$  are all zero.

As a result, in realizing  $p_1$ ,  $p_2$ , and  $p_3$  with  $n$ -bit fractional precision, Eqs. (2-2a), (2-2b), and (2-2c) can be reduced to Eqs. (2-9a), (2-9b) and (2-9c), respectively, by combining Eqs. (2-8a) and (2-8b).

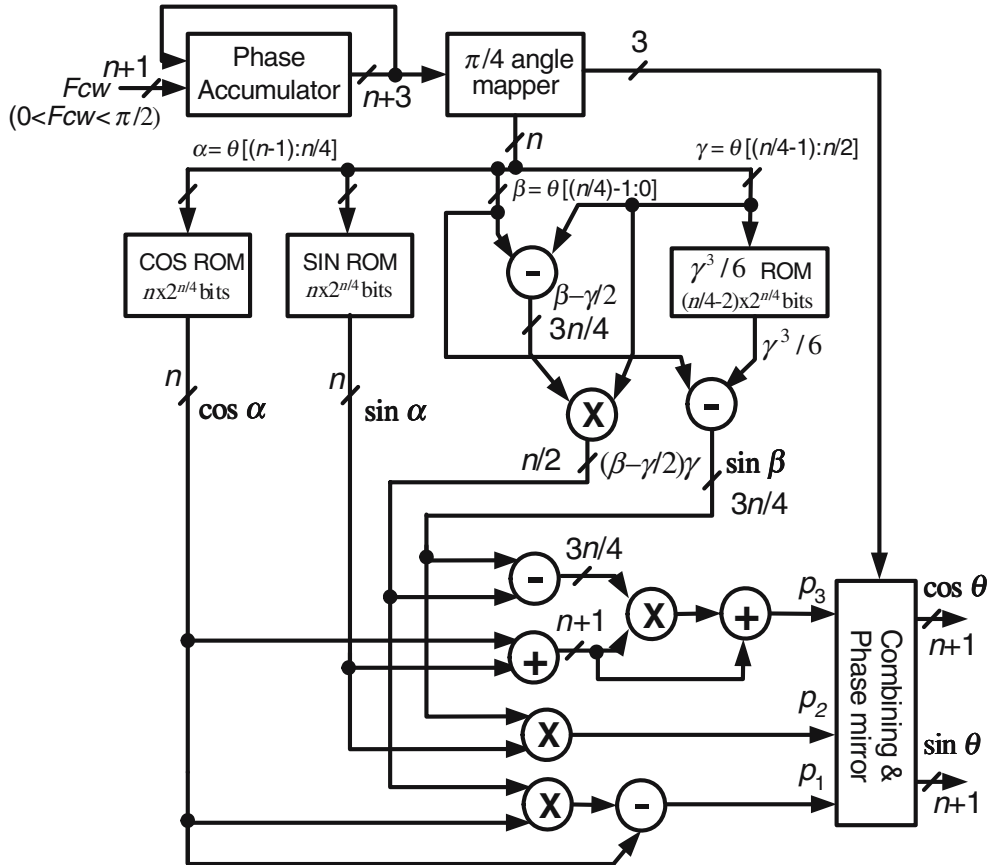


Figure 1. Architecture of the new DDFS.

Table 1. Complexity and speed comparisons of DDFS designs.

DDFS	ROM size (in bit)	No. of multipliers	Delay time (16 bits)	Critical path
Proposed	$n \times 2^{n/4+1} + (\frac{n}{4} - 2) \times 2^{n/4}$	1 $(n+1) \times 3n/4$ 1 $n \times n/2$ 1 $n \times 3n/4$ 1 $n/4 \times 3n/4$	6.7 ns	$2t_{mul}^a + 2t_{add}^a$
[1]	$n \times 2^{n/2+1}$	0	2.4 ns	$2t_{add}$
[2]	$3n \times 2^{n/2}$	2 $n/2 \times n/2$	3.8 ns	$t_{mul} + t_{add}$
[7] <sup>b</sup>	$\approx 5n \times 2^{n/2-1}$	2 $n/2 \times n/2$	4.5 ns	$t_{mul} + 2t_{add}$
[13]	0	6 $(n+1) \times (n+1)$	10.1 ns	$3t_{mul} + 2t_{add}$
[18]	$(n+1) \times 2^{n/3+1}$	4 $(n+1) \times (n+1)$	7.4 ns	$2t_{mul} + 2t_{add}$

<sup>a</sup> $t_{add}$  represents one adder delay, and  $t_{mul}$  represents one multiplier delay.

<sup>b</sup>The ROM size is slightly smaller than  $5n \times 2^{n/2-1}$ .

$$p_1 \approx \cos \alpha - \cos \alpha \times (\beta - \gamma/2)\gamma \quad (2-9a)$$

$$p_2 \approx \sin \alpha \times (\beta - \gamma^3/6) \quad (2-9b)$$

$$p_3 \approx (\cos \alpha + \sin \alpha) + (\cos \alpha + \sin \alpha) \times [\sin \beta - (\beta - \gamma/2)\gamma] \quad (2-9c)$$

(c)  $(\cos \alpha + \sin \alpha) \times [\sin \beta - (\beta - \gamma/2)\gamma]$ : Based on Eq. (2-8b),  $[\sin \beta - (\beta - \gamma/2)\gamma]$  is bounded by:

$$2^{-n/4} > \sin \beta \geq \left[ \sin \beta - \left( \beta - \frac{\gamma}{2} \right) \gamma \right] \approx \beta(1 - \gamma) + \frac{\gamma^2}{2} \left( 1 - \frac{\gamma}{3} \right) \geq 0 \quad (2-10)$$

In implementation, the multiplier complexities for these three multiplications are as explained below.

- (a)  $\cos \alpha \times (\beta - \gamma/2)\gamma$ : Since  $2^{-n/4} > \beta \geq \gamma \geq 0$ , then  $2^{-n/2} > (\beta - \gamma/2)\gamma \geq 0$  and  $n/2$  bits are enough to represent  $(\beta - \gamma/2)\gamma$ , if  $n$ -bit fractional results are desired. Therefore, the multiplier size is  $n \times n/2$ .
- (b)  $\sin \alpha \times (\beta - \gamma^3/6)$ : Since  $2^{-n/4} > \beta \geq \gamma \geq 0$ , then  $2^{-n/4} > (\beta - \gamma^3/6) \geq 0$  and  $3n/4$  bits are enough to represent  $(\beta - \gamma^3/6)$ . Hence, the multiplier size is  $n \times 3n/4$ .

Therefore,  $3n/4$  bits are enough to represent  $[\sin \beta - (\beta - \gamma/2)\gamma]$ . On the other hand, there needs  $n+1$  bits to represent  $(\cos \alpha + \sin \alpha)$ , because both operands are  $n$ -bit long. As a result, the multiplier size is  $(n+1) \times 3n/4$ .

For better understanding of the new two-level table-lookup DDFS algorithm, Fig. 1 shows its data path realization. In the figure,  $F_{cw}$  represents the input frequency control word. Note that the final outputs are  $(n+1)$  bits in length, including one sign bit and  $n$  fractional bits.

Table 2. Area and speed performance comparison of DDFS designs vs. four different minimum target SFDR's.

	80 dBc		100 dBc		120 dBc		160 dBc		200 dBc	
	Total gates	Max. delay	Total gates	Max. delay	Total gates	Max. delay	Total gates	Max. delay	Total gates	Max. delay
Proposed	1,570	3.5 ns	2,153	4.6 ns	<b>4,150</b>	5.0 ns	<b>8,314</b>	5.4 ns	<b>23,989</b>	6.0 ns
[1]	1,460	1.8 ns	4,321	2.2 ns	13,510	2.4 ns	96,778	2.7 ns	872,864	3.2 ns
[2]	1,430	2.1 ns	2,082	2.8 ns	10,371	2.9 ns	26,297	3.5 ns	77,042	4.0 ns
[7]	<b>1,235</b>	2.1 ns	<b>1,856</b>	3.2 ns	4,416	3.7 ns	16,158	3.9 ns	70,171	4.1 ns
[13]	1,853	6.8 ns	2,595	8.2 ns	8,320	8.4 ns	14,250	8.9 ns	31,911	9.5 ns
[18]	1,864	5.6 ns	3,561	6.5 ns	7,798	6.9 ns	21,660	7.4 ns	71,996	7.9 ns

Table 3. Synthesized gate count results of the new DDFS.

Block name	Combinational circuit	Non-combinational circuit	Sum
Phase accumulator	530	210	740
sin/cos generator	1,321	0	1,321
Control logic	442	168	610
Rom table	126	0	126
Total	2,419	378	2,797

### 3. Performance Evaluations and Comparisons

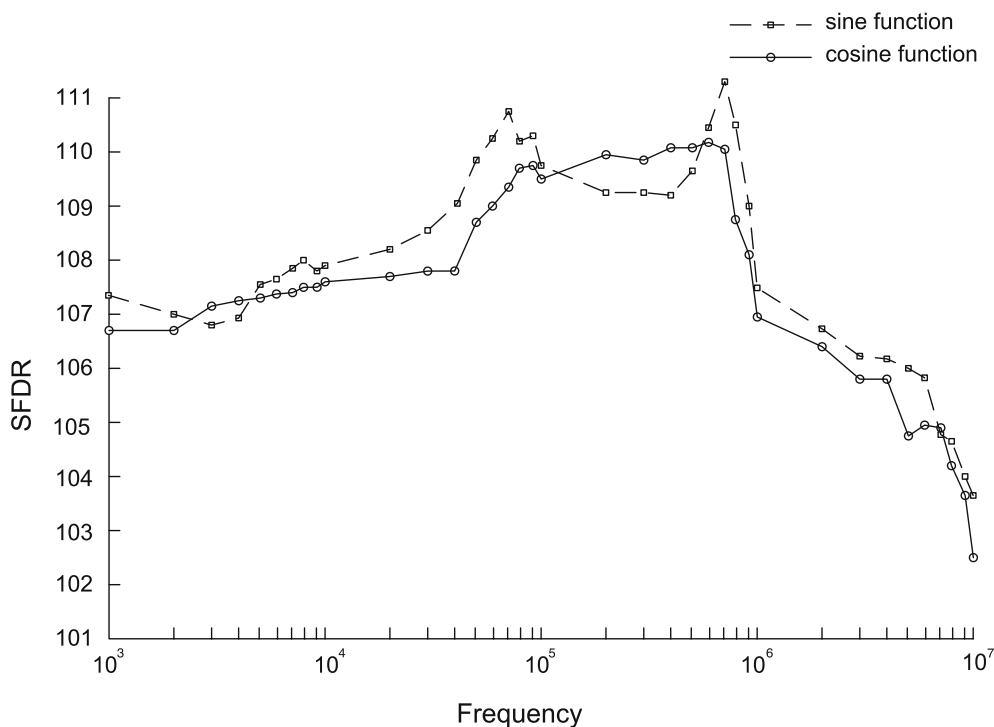
We synthesized those designs by using Synopsys Design Analyzer. Although CORDIC-based DDFS designs are efficient, they are not compared here due to their slow switching speed. The hardware and speed performance comparisons of the proposed DDFS design and some key table lookup-based and polynomial-based DDFS's are listed in Table 1. The delay figures are obtained assuming 16-bit outputs.

In the aspect of hardware cost, as listed in Table 1, the proposed design has fewer multipliers and a

smaller table size than [18]. Correspondingly, the critical path of the proposed design is shorter than that of [18]. Regarding table size, the proposed design requires the smallest table size of all, except the design of [13]. However, the latter design needs six multipliers. Regarding multiplier complexity, although the design of [1] does not need any multiplier, it needs a larger table size than the proposed design. Therefore, they are only suitable for applications with low SFDR's and output precisions. Similarly, the design of [7] has a larger table size than that of the proposed design, though it only costs two multipliers.

Table 2 shows the total required synthesized circuit areas (excluding phase accumulator and angle mapper) and speed performances of the compared designs to achieve the same minimum SFDR's.

As shown, in most cases (especially for high SFDR cases), the proposed design has the smallest areas (i.e., those quantities in bold faces) of all, with moderate delays. In the low SFDR case of 80 dBc, the proposed design's area is only a bit larger than the designs of [1, 2, 7]. From the simulation results in Table 2, we can roughly locate a SFDR limit around 110 dBc. Beyond that limit, the proposed

Figure 2. SFDR performance of the proposed DDFS vs. synthesized frequency,  $n=16$ .

design required a lower complexity than all the compared designs. Note that in the above comparisons, we do not discuss the power consumption performance, because it is hard to do a precise and fair power consumption comparison for those designs, from their architectures and those subjective designer-dependent synthesized results.

#### 4. Hardware Verification

Function correctness of the new DDFS has been verified by FPGA with Altera Quartus II system. In addition, Table 3 lists the synthesized gate count results by using Design Analyzer, assuming 16-bit output resolution. The synthesized design has a low delay and gate count of 6.7 ns and 2,797, respectively, with a power consumption of 18 mW. To verify SFDR performance, we test the proposed design with many output frequencies ranging from 1 kHz to 10 MHz, as shown in Fig. 2, assuming a system clock is 100 MHz clock and 16-bit output resolution. As shown, the synthesized waveforms achieve at least 100 dBc and maximum 110 dBc.

#### 5. Conclusion

In the work, we proposed a two-level table-lookup direct digital frequency synthesizer (DDFS), then realized and verified it by MATLAB simulation, HDL simulation/synthesis, and FPGA implementation. It has 6.7 ns delay and a small gate count of only 2,797 to achieve a high averaged SFDR of 110 dBc, with a power consumption of 18 mW. Since its main advantages are small ROM size and high SFDR, it is particularly suitable for the applications that need high SFDR and output resolution.

#### Acknowledgment

This work is supported in part by the grants NSC 94-2220-E-009-034 and MOEA 94-EC-17-A-01-S1-048, Taiwan.

#### References

1. D. A. Sunderland, R. A. Strauch, S. S. Wharfield, H. T. Peterson and C. R. Cole, "CMOS/SOS Frequency Synthesizer LSI Circuit for Spread Spectrum Communications," *IEEE J. Solid-State Circuits*, vol. 19, 1984, pp. 497–505, Aug.
2. 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, 2000, pp. 385–390.
3. S. Morteza pour and E. K. F. Lee, "Design of Low-Power ROM-Less Direct Digital Frequency Synthesizer Using Nonlinear Digital-to-Analog Converter," *IEEE J. Solid-State Circuits*, vol. 34, 1999, pp. 1350–1359.
4. L. Lo Presti and G. Cardamone, "A Direct Digital Frequency Synthesizer Using an IIR Filter Implemented with a DSP Microprocessor," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, 1994, pp. 201–204.
5. M. M. Al-Ibrahim, "A Simple Recursive Digital Sinusoidal Oscillator with Uniform Frequency Spacing," *Proceedings of IEEE International Symposium on Circuits Systems*, vol. 2, 2001, pp. 689–692.
6. D. Das Sarma and D. W. Matula, "Faithful Bipartite ROM Reciprocal Tables," *Proceedings of 12th Symposium on Computer Arithmetic*, 1995, pp. 17–29.
7. F. Curticepean and J. Niittylahti, "A Hardware Efficient Direct Digital Frequency Synthesizer," *Proceedings of IEEE International Conference on Electronics, Circuits and Systems*, vol. 1, 2001, pp. 51–54.
8. F. Curticepean and J. Niittylahti, "Low-Power Direct Digital Frequency Synthesizer," *Proceedings of IEEE International Symposium on Circuits Systems*, 2000, pp. 822–825.
9. J. M. P. Langlois and D. Al-Khalili, "Novel Approach to the Design of Direct Digital Frequency Synthesizers Based on Linear Interpolation," *IEEE Transactions on Circuits System II: Analog and Digital Signal Processing*, vol. 50, no. 9, 2003, pp. 567–578.
10. Y. K. Chang and E. E. Swartzlander, "An Analysis of the CORDIC Algorithm for Direct Digital Frequency Synthesis," *Proceedings of Application-Specific Systems, Architectures and Processor*, 2002, pp. 111–119.
11. A. Madiseti, A. Y. Kwentus and A. N. Wilson, "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, 1999, pp. 1034–1043.
12. J. C. Chih, J. Y. Chou and S. G. Chen, "An Efficient Direct Digital Frequency Synthesizer Based on Two-level Table-Lookup," *Proceedings of Frequency Control System and PDA Exhibition*, 2001, pp. 824–827.
13. C. C. Wang, H. C. She and H. Ron, "A ROM-Less Direct Digital Frequency Synthesizer by Using Trigonometric Quadruple Angle Formula," *Proceedings of IEEE International Conference on Electronics, Circuits, and Systems*, vol. 1, 2002, pp. 65–68.
14. D. De Caro, E. Napoli and A. G. M. Strollo, "Direct Digital Frequency Synthesizers with Polynomial Hyperfolding Technique," *IEEE Trans. Circuits Syst.*, vol. 51, 2004, pp. 337–344.
15. A. M. Sodagar and G. Roientan, "A Novel Architecture for ROM-less Sine-Output Direct Digital, Frequency Synthesizers by Using the 2-Order Parabolic Approximation," *Proceedings of Frequency Control System and PDA Exhibition*, 2000, pp. 284–289.
16. K. I. Palomaki and J. Niittylahti, "Direct Digital Frequency Synthesizer Architecture Based on Chebyshev Approximation," *Proceedings of Signals, Systems and Computers*, vol. 2, 2000, pp. 1639–1643.

17. L. Fanucci, R. Roncella and R. Saletti, "A Sine Wave Digital Synthesizer Based on A Quadratic Approximation," *Proceedings of Frequency Control System and PDA Exhibition*, 2001, pp. 806–810.
18. A. M. Sodagar and G. R. Lahiji, "Mapping From Phase to Sine-Amplitude in Direct Digital Frequency Synthesizers Using Parabolic Approximation," *IEEE Trans. Circuits Syst.*, vol. 47, 2000, pp. 1452–1457, Dec.
19. S. Yonachul and K. Beomsup, "A 16b Quadrature Direct Digital Frequency Synthesizer Using Interpolative Angle Rotation Algorithm," *VLSI Circuits Digest of Technical Papers*, 2002, pp. 146–147.
20. M.-S. Lin, C.-S. Yang, C.-Y. Chu and S.-G. Chen, "An Efficient Pipeline Direct Digital Frequency Synthesizer Based on a Novel Interpolation Algorithm," *Proceedings of the 2001 European Conference on Circuit Theory and Design*, 2001, pp. 273–276.
21. A. A. Karatsuba and Y. P. Ofman, "Multiplication of Multi-digit Numbers on Automata," *Sov. Phys. Dokl.*, vol. 7, 1963, pp. 595–596.



**Sau-Gee Chen** received the BS degree from National Tsing Hua University, Taiwan, in 1978, the MS degree and PhD degree in electrical engineering, from the State University of New York at Buffalo, NY, in 1984 and 1988 respectively.

He currently is a Professor and the Director of Institute of Electronics, Department of Electronics Engineering, National Chiao Tung University, Taiwan. His research interests

include digital communication, multi-media computing, digital signal processing, and VLSI circuit design.



**Jen-Chuan Chih** was born in Taipei, R.O.C. He received the BS and MS degrees all in electronics engineering from the Department of Electronics Engineering and Institute of Electronics, National Chiao Tung University, Taiwan, in

1998 and 1999 respectively. Currently, he is pursuing the PhD degree. His research interests are in fast DSP algorithms and VLSI implementation.



**Chun-Yi Chou** received the BS degree in Electrical Engineering from the National Yunlin University of Science and Technology, Taiwan, and MS degree in Degree Program of Electrical Engineering and Computer Science, National

Chiao Tung University, Taiwan, in 1996 and 2002 respectively. Currently, he is an Engineer at Novatek Microelectronics Corp., Hsinchu, Taiwan, R.O.C., working in the field of Mobile Phone TFT LCD Driver IC design. His research interests are mix-mode architecture design and VLSI implementation.