

A PHASE-ADJUSTABLE ROM-LESS DIRECT DIGITAL FREQUENCY SYNTHESIZER WITH 41.66 MHZ OUTPUT FREQUENCY §

Chua-Chin Wang †, Yih-Long, Tseng, Wun-Ji Lin, and Ron Hu ‡

Department of Electrical Engineering
National Sun Yat-Sen University
Kaohsiung, Taiwan 80424
email : ccwang@ee.nsysu.edu.tw

ABSTRACT

A high-speed phase-adjustable ROM-less direct digital frequency synthesizer (DDFS) employing trigonometric quadruple angle formula is presented. The spectral purity is better than -130 dBc worst case spur. The resolution is up to 13 bits. Most important of all is that the output sinusoidal frequency is more than 40 MHz which is far more than the 32 MHz requirement of Korean PCS, GSM, and Bluetooth. Neither any scaling table nor error correction tables are required. The maximum error is mathematically analyzed. The word length of each multiplier is carefully selected in the digital implementation such that the error range is limited and the resolution is preserved.

1. INTRODUCTION

Ever since the low-cost RF CMOS technology becomes the challenger of its conventional discrete counterpart, the spectral quality of the frequency synthesizers in a single chip solution has been demanded to possess better purity. Direct digital frequency synthesizers (DDFSs) are very much preferred in some modern communication systems owing to their advantages over PLL-based solutions, e.g., fast settling time, sub-Hertz frequency resolution, continuous-phase frequency switching and low phase noise [4]. The bottleneck of the DDFS method is the generation of a pure sinusoidal output. Many prior works were proposed to resolve this problem, including ROM-based lookup tables [1], [2], [3], [4], [5], complex pipelined structure with a low FSM and a ROM [6], or scaling and error correction tables [7]. All of the ROM-based solutions suffer from ROM's intrinsic drawbacks which are slow speed, large area, and high power consumption. Sodagar *et al.* proposed a ROM-less DDFS by using 2nd-order parabolic approximation [7]. However, in order to reduce the conversion error, a scaling table and an error

correction table (or generator) are needed. It, thus, not only deteriorates the speed performance, but also affect the resolution of the output word length. Although the DDFS proposed by [8] resolved most of the mentioned problem, it could only generate a very low frequency output, i.e., 5 KHz sine wave which is not adequate for any wireless applications. In this paper, we propose a novel ROM-less design for DDFSs, which utilizes trigonometric 4θ formula to attain the smaller error range. A pipelining methodology and tunable phase selections are adopted to enhance the processing speed as well as the throughput. The output is 13-bit resolution and the worst case spur is -130 dBc, while the output frequency is 40 MHz.

2. HIGH-SPEED ROM-LESS DDFS

A basic idea to carry out the ROM-less DDFS is to utilize the trigonometric quadruple angle formula such that the irregularity of the scaling and error correction difficulties in [7] will be eliminated. In addition, the upper bound of the error range can be analytically solved.

2.1. Trigonometric 1st-order 4θ approximation

The quadruple angle formula can be re-arranged as the following equality.

$$\cos 4\theta = 2 \cos^2 2\theta - 1 = 1 - 8 \sin^2 \theta (1 - \sin^2 \theta) \quad (1)$$

Since the range of 4θ is limited in $[0, \frac{\pi}{2}]$ [4], the range of θ is $[0, \frac{\pi}{8}]$. Thus, $\sin \theta \approx \theta$. Eqn.(1) becomes

$$\cos 4\theta \approx 1 - 8\theta^2(1 - \theta^2), \quad 0 \leq \theta \leq \frac{\pi}{8} \quad (2)$$

Notably, the maximum amount of error occurs at 90° . In order to minimize the amount of error, the upper bound must be chosen to be smaller than $\frac{\pi}{8} \approx 0.3927$. This bound should also be easily converted into a digital representation which will make the physical implementation feasible. The simulink of MATLAB is employed to find such a proper bound which will meets the requirement of at least 12-bit output resolution.

§This research was partially supported by National Science Council under grant NSC 91-2218-E-110-001 and 91-2622-E-110-004.

†Prof. Wang is the contact author.

‡Dr. Hu is General Manager of Asuka Microelectronics Inc., Hsin-Chu, Taiwan.

The simulation results suggest a nice selection at $\frac{3135}{8192}$ with an error $\leq 2.4 \cdot 10^{-4}$. Hence, we re-define our 1st-order approximation method, called TA1(x) (1st-order trigonometric approximation), as follows.

$$\text{TA1}(x) = 1 - 8x^2(1 - x^2), \quad 0 \leq x \leq \frac{3135}{8192} \quad (3)$$

Fig. 1 illustrates the actual cosine function and TA1(x), while the difference of these two functions, which is TA1(x) - cosθ, is given in Fig. 2. The maximum error attained graphically is $13 \cdot 10^{-3}$ which is smaller than $15.625 \cdot 10^{-3} = \frac{1}{2^6}$. It indicates that the 1st-order approximation has at least 6-bit resolution.

Since the error function, err1(x) = TA1(x) - cosθ, is not a good function to be implemented digitally. We propose to use a polynomial function to fit the error function. The steps are summarized as follows.

- 1). Keep dividing TA1(x)(1 - TA1(x)) by 2 until the maximum of TA1(x)(1 - TA1(x)) is close to the maximum of err1(x).
- 2). A scaling factor, K, is chosen to further reduce the error between TA1(x)(1 - TA1(x)) and cos x. The K must be digitally representable. Besides, the final error must be less than $\frac{1}{2^{12}} = 2.4 \cdot 10^{-4}$ to ensure the resolution.

The optimization procedure is carried out by simulink of MATLAB. The final optimized error function becomes as follows.

$$\begin{aligned} \text{err1}(x) &= K \cdot (0.5)^4 \text{TA1}(x) \cdot (1 - \text{TA1}(x)), \quad (4) \\ &\approx \text{TA1}(x) - \cos x, \end{aligned}$$

where $K = (0.84375)_{10} = (0.11011)_2$, $0 \leq x \leq \frac{3135}{8192}$, and $0 \leq \theta \leq \frac{\pi}{2}$.

2.2. 2nd-order approximation

A simple thought to further reduce the amount of error between the cosine function and the approximation equation is to utilize a 2nd-order difference method, which is given as follows.

$$\text{TA2}(x) = \text{TA1}(x) - \text{err1}(x), \quad 0 \leq x \leq \frac{3135}{8192}, \quad (5)$$

We attain the maximal amount of error from the figure is $0.8 \times 10^{-4} < 1.22 \times 10^{-4} = \frac{1}{2^{13}}$, we conclude that the output resolution of our proposed method is guaranteed to be 13 bits, which is more accurate any prior work. In other words, a trigonometric 4θ approximation with error correction for sinusoidal output is attained.

2.3. Analytic solutions

It is also an interesting thing to find out where the maximal error is. We represent the difference between TA2 and cosine as another error function.

$$\begin{aligned} \text{err2} &= \text{TA2}(x) - \cos \theta, \\ &\text{where } 0 \leq x \leq \frac{3135}{8192}, \quad 0 \leq \theta \leq \frac{\pi}{2} \quad (6) \end{aligned}$$

$$\begin{aligned} \text{TA2}(x) &= \text{TA1}(x) - A \cdot \text{TA1}(x)(1 - \text{TA1}(x)), \\ &\text{where } A = 0.84375 \cdot (0.5)^4 \quad (7) \end{aligned}$$

$$\text{TA1}(x) = 1 - 8x^2(1 - x^2) \quad (8)$$

By substituting Eqns.(7) and (8) into Eqn.(6), we obtain the entire err2. Then, we take the first order derivative of err2 and solve the solution given that $\text{err2}' = 0$ to attain the following equations.

$$\begin{aligned} \text{err2}' &= \text{TA2}'(x) - (\cos \theta)' = 0, \\ 0 &= (32x^3 - 16x)(16Ax^4 - 16Ax^2 + A + 1) - (\cos \theta)' \\ 0 &= (32x^3 - 16x)(16Ax^4 - 16Ax^2 + A + 1) + \\ &\quad \frac{8192\pi}{6270} \sin \theta, \quad \text{where } \theta = \frac{8192\pi}{6270}x \quad (9) \end{aligned}$$

By graphically solving the two terms in Eqn.(9) as shown in Fig. 3, there are two intersections between the two curves which denotes where the maximum errors are located.

2.4. System design by pipelining

Fig. 4 is a typical implementation of DDFSs in prior works. The slow and large ROM not only occupies a significant portion of the chip (or board) area, it also degrades the speed. If a direct implementation of Eqn. (7), (8), and (9) is adopted, a large number of multipliers are required which will result in low speed and large area. Hence, We propose our pipelining digital implementation basing upon the proposed 4θ approximation method in Fig. 5. Besides the pipelining design, the phase computation is also simplified.

Phase Accumulation : A bottle-neck in the early stage of the DDFS is the generation of phase square in Eqn. (2) in which the resolution of the phase is 13 bits. It will be a disaster to use either a squarer or a multiplier at this stage. We decompose the 13 bits into 6 high bits (denoted as H) and 7 low bits (denoted as L).

$$(HL)^2 = H^2 \cdot 2^{14} + L^2 + 2 \cdot H \cdot L \cdot 2^7 \quad (10)$$

Notably, both of the $H^2 \cdot 2^{14}$ and L^2 are 14-bit terms, which can be calculated by combinational logic without using any long adders. Hence, the result in Eqn. (10) can be generated by a single adder.

Phase Adjustability : Eqn. (3) tells that the maximum count of the phase input is 3135 (≈ 12 bits). Meanwhile, the resolution of the design is proved to be 13 bits. Hence, the maximum count of the phase_acc

in Fig. 5 is 6270. The 10-bit input at phase_displace is fed by a digital PLL (not shown) such that the phase comparison and adjustment can be carried out.

$$f_{out} = \frac{\text{phase_acc}}{6270 \times 4} \cdot f_{clock}, \theta = \frac{\text{phase_displace}}{6270} \cdot \frac{\pi}{2}, \quad (11)$$

where f_{out} is the frequency of the output sinusoidal waves, and f_{clock} is the system clock. As soon as there is a phase-adjusted command, the “dis” at the Fig. 5 is pulled high to add “phase_displace” and “phase_acc”, which is shown in Fig. 6.

3. SIMULATION AND IMPLEMENTATION

Modelsim of Mentor and MATLAB of Mathworks are the S/W tools to proceed the system-level simulations. The design in Fig. 5 is coded by RTL Verilog which is then simulated by Modelsim. The RTL code is then synthesized by SYNOPSIS and post-layoutedly verified by TimeMill and PowerMill. Fig. 6 shows the transition from a 5.55 MHz sine output to a 41.66 MHz O/P given a worst condition (SS Model, 0°C, $f_{clock} = 166$ MHz). Meanwhile, the decimal output data in a 12-bit format are collected. The FFT command of MATLAB is executed to attain the spectrum as shown in Fig. 7, which illustrates the spurious performance of the proposed method is as high as -130 dBc. It is far better than any prior works. Table 1 summarizes the performance of our work and prior methods.

	resolution	spurious
[1]	10 bits	-55 dBc
[4]	12 bits	-55 dBc
[5]	12 bits	-98.75 dBc
[6]	12 bits	-70 dBc
[7]	12 bits	-62.8 dBc
ours	13 bits	-130 dBc

Table 1: Performance comparison

We, then, follow the standard cell-based design flow to implement the our work. Fig. 8 is the layout of the proposed design. Table 2 summarizes the characteristics of the chip.

f_{clock}	166 MHz
max. f_{out}	41 MHz
SFDR	-78 dB
THD	-41 dB
avg. power	16.774 mW
resolution	13 bits
core area	1118.8×1118.8 μm^2
gate count	13259

Table 2: Characteristics of the proposed DDFS

4. CONCLUSION

In this paper, we have presented a novel implementation utilizing pipelining to carry out a ROM-less DDFS which is based on the quadruple angle equality equation. Not only are the spurious tones reduced, the 2nd-order error correction has been simulated to justify the capability of subsiding the noise power of the harmonics.

5. REFERENCES

- [1] G. Van Andrews, *et al.*, “Recent progress in wideband monolithic direct digital synthesizers,” *IEEE MTT-S Inter. Microwave Symp. Digest*, vol. 3, pp. 1347-1350, 1996.
- [2] M. J. Flanagan, and G. A. Zimmerman, “Spur-reduced digital sinusoid synthesis,” *IEEE Trans. on Comm.*, vol. 43, no. 7, pp. 2254-2262, July 1995.
- [3] V. F. Kroupa, V. Cizek, J. Stursa, and H. Svan-dova, “Spurious signals in direct digital frequency synthesizers due to the phase truncation,” *IEEE Trans. on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 47, no. 5, pp. 1166-1172, Sep. 2000.
- [4] G. W. Kent, and N.-H. Sheng, “A high purity, high speed direct digital synthesizer,” *1995 49th IEEE Inter. Frequency Control Symp.*, pp. 207-211, 1995.
- [5] R. Larson, and S.-L. Lu, “Interpolation-based digital quadrature frequency,” *13th Annual IEEE Inter. ASIC/SOC Conf.*, pp. 48-52, 2000.
- [6] K. I. Palomaki, and J. Niittylahti, “Direct digital frequency synthesizer architecture based on Chebyshev approximation,” *Thirty-Fourth Asilomar Conference on Signals, Systems and Computers*, vol. 2, pp. 1639-1643, 2000.
- [7] A. M. Sodagar, and G. R. Lahihi, “A novel architecture for ROM-less sine-output direct digital frequency synthesizers by using the 2nd-order parabolic approximation,” *2000 IEEE/EIA Inter. Frequency Control Symp. and Exhibition*, pp. 284-289, 2000.
- [8] C.-C. Wang, H.-C. She, and R. Hu, “A ROM-less direct digital frequency synthesizer by using trigonometric quadruple angle formula,” *9th IEEE International Conference on Electronics, Circuits and Systems (ICECS 2002)*, pp. 65-68, Sep. 2002.

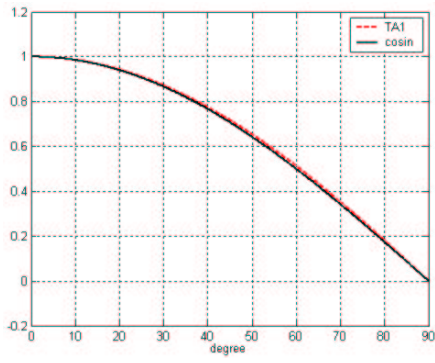


Figure 1: comparison of cosine and $TA1(x)$

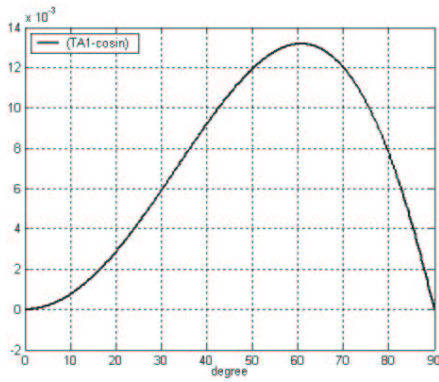


Figure 2: $err1(x)$

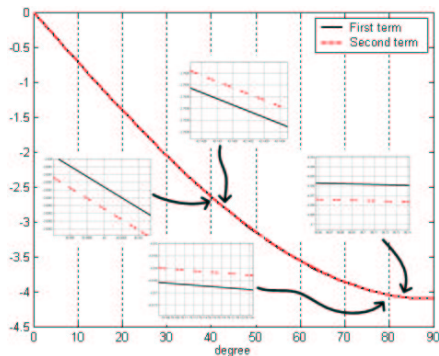


Figure 3: graphical solutions for maximum error in $err2$

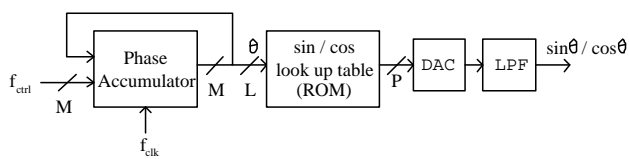


Figure 4: the architecture of prior ROM-based DDFSs

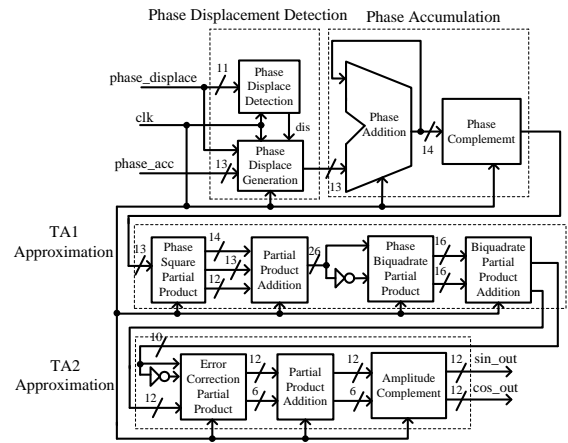


Figure 5: the proposed ROM-less DDFS

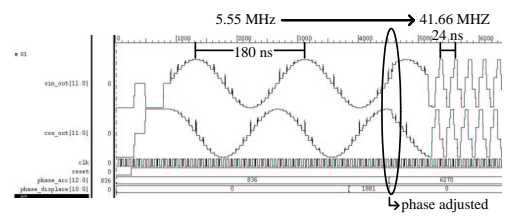


Figure 6: Post-layout simulation results

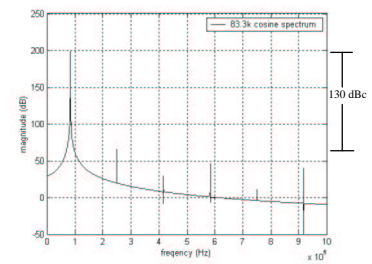


Figure 7: spurious performance of the proposed DDFS

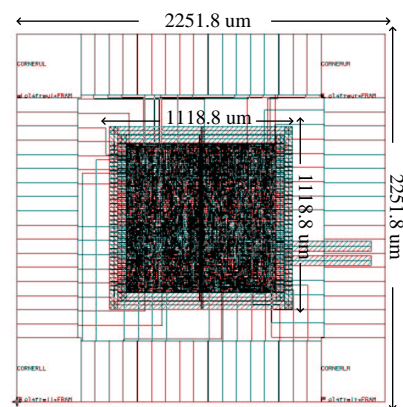


Figure 8: measured spectrum of the proposed DDFS