# Phase-Adjustable Pipelining ROM-Less Direct Digital Frequency Synthesizer With a 41.66-MHz Output Frequency

Chua-Chin Wang, *Senior Member, IEEE*, Jian-Ming Huang, Yih-Long Tseng, Wun-Ji Lin, and Ron Hu

***Abstract*—A high-speed phase-adjustable read-only-memory less direct digital frequency synthesizer employing trigonometric quadruple angle formula is presented. A ten-stage pipelining architecture is employed based upon decomposition of phase operands. Spectral purity is better than −130 dBc for the worst case spurious-free dynamic range. The resolution is up to 12 bits. Most importantly, the output sinusoidal frequency is higher than 40 MHz, which is far higher than the 32-MHz requirement of Korean personal communications system, global system for mobile communications, 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.**

***Index Terms*—Direct digital frequency synthesizer (DDFS), frequency synthesizer, read-only-memory (ROM)-less, squarer decomposition, wireless network.**

## I. Introduction

EVER SINCE the low-cost radio frequency (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 because of their advantages over phase-locked loop (PLL)-based solutions, e.g., fast settling time, subhertz frequency resolution, continuous-phase frequency switching, and low phase noise [1], [2]. The bottleneck of the DDFS method is the generation of a pure sinusoidal output. Many prior works were proposed to resolved this problem, including read-only-memory (ROM)-based look-up tables [1], [3]–[6], complex pipelined structure with a low FSM, and a ROM [7], or scaling and error correction tables [8]. All of the ROM-based solutions suffer from ROM's intrinsic drawbacks, namely low speed, large area, and high power consumption. Sodagar and Lahiji proposed a ROM-less DDFS by using second-order parabolic approximation [8]. However, to reduce the conversion error, a scaling table and an error correction table (or generator) are needed in their design. Thus, not only does it deteriorate the speed performance, but it also affects the resolution of the output word length. Although the DDFS proposed in [9] resolved most of the mentioned problems, it could only generate a very low frequency output, i.e., a 5-kHz sine wave, which is not adequate for most of wireless applications. In this brief, we propose a novel pipelining ROM-less design for DDFSs that utilizes a trigonometric quadruple angle formula to attain a smaller error range. The pipelining methodology and tunable phase selections are adopted to enhance the processing speed as well as the throughput. The output is a 12-bit resolution and the worst case spurious-free dynamic range is −130 dBc, whereas the output frequency is higher than 40 MHz.

Manuscript received March 31, 2005; revised October 3, 2005. This work was supported in part by the National Science Council under Grant NSC94-2213-E-110-024 and Grant NSC94-2213-E-110-022 and the National Health Research Institutes under Grant NHRI-EX95-9319EI, and by "Aim for Top University Plan" of NSYSU. This paper was recommended by Associate Editor S. L. Netto.

C.-C. Wang is with the Department of Electrical Engineering, National Sun Yat-Sen University, Kaohsiung 804, Taiwan, R.O.C. (e-mail: ccwang@ee.nsysu.edu.tw).

R. Hu is with Asuka Microelectronics Inc., Hsinchu, Taiwan, R.O.C.

Digital Object Identifier 10.1109/TCSII.2006.882236

## II. 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 [8] will be eliminated. In addition, the upper bound of the error range can be analytically solved.

### A. Trigonometric First-Order Quadruple Angle Approximation

The quadruple angle formula can be rearranged as follows:

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

Since $\cos 2\pi$ to $\cos(\pi/2)$ can be derived from $\cos 0$ to $\cos(\pi/2)$, the range of $4x$ is limited in $[0, \pi/2]$ [1], and hence, the range of $x$ is $[0, \pi/8]$. Thus, $\sin x \approx x$. Equation (1) now becomes

$$\cos 4x \approx 1 - 8x^2(1 - x^2), \qquad 0 \le x \le \frac{\pi}{8}. \quad (2)$$

Notably, the maximum amount of error occurs at $\cos(\pi/2)$. In order to minimize the amount of error, the upper bound must be chosen to be smaller than $(\pi/8) \approx 0.3927$. This bound should also be easily converted into a digital representation that makes the physical implementation feasible. The Simulink of MATLAB is employed to find such a proper bound that meets the requirement of at least a 12-bit output resolution. The simulation results suggest a nice selection at 3135/8192 with an error $\le 1/2^{12}$. Hence, we redefine our first-order approximation method, which is called $\mathrm{TA1}(x)$ (first-order trigonometric approximation), as follows:

$$\mathrm{TA1}(x) = 1 - 8x^2(1 - x^2), \qquad 0 \le x \le \frac{3135}{8192}. \quad (3)$$

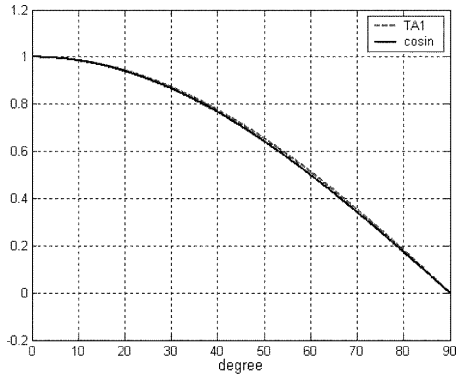


Fig. 1. Comparison of cosine and TA1$(x)$.

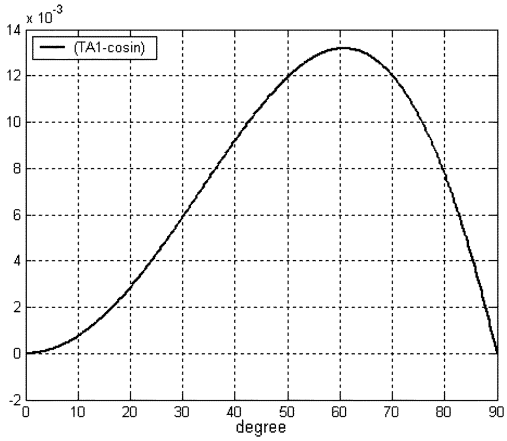


Fig. 2. Plot of the first-order error function err1$(x)$.

Fig. 1 illustrates the actual cosine function and TA1$(x)$, whereas the difference of these two functions, which is TA1$(x) - \cos 4x$, is given in Fig. 2. The maximum error attained graphically is $13 \cdot 10^{-3}$, which is smaller than $15.625 \cdot 10^{-3} = 1/2^6$. It indicates that the first-order approximation has at least a 6-bit resolution.

As shown in Fig. 2, TA1$(x) - \cos 4x$ is like a second-order parabolic curve. A second-order polynomial err1$(x) = K \cdot (0.5)^B \cdot \text{TA1}(x)(1 - \text{TA1}(x))$ is used to fit the error function TA1$(x) - \cos 4x$. The steps are summarized as follows.

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

The optimization procedure is carried out by the Simulink of MATLAB. The final optimized error function is now defined as follows ("$K$" is decided):

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

where $K = (0.84375)_{10} = (0.11011)_2$, $B = 4$, and $0 \le x \le (3135/8192)$.

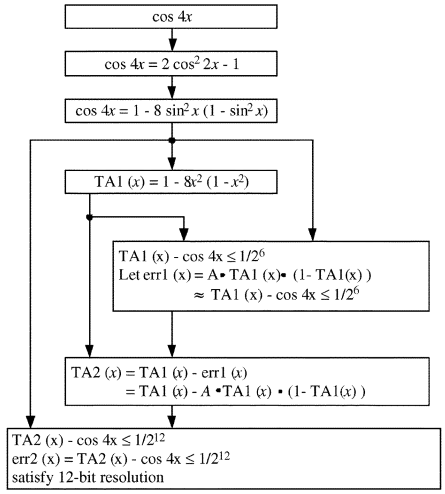


Fig. 3. Calculation flow of TA2$(x)$.

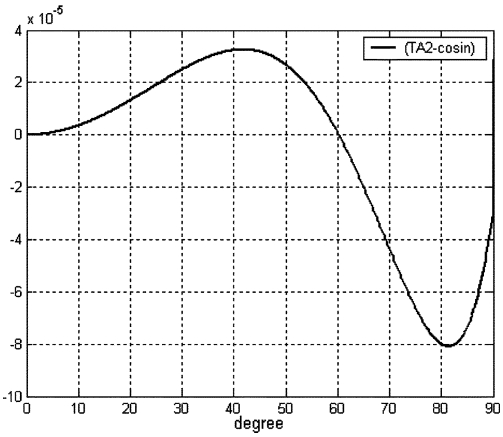


Fig. 4. Plot of the second-order error function err2$(x)$.

### B. Second-Order Approximation

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

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

$$\text{TA2}(x) = \text{TA1}(x) - A \cdot \text{TA1}(x) \cdot (1 - \text{TA1}(x)) \tag{6}$$

where $A = K \cdot (0.5)^B = 0.84375 \cdot (0.5)^4$

$$\text{err2}(x) = \text{TA2}(x) - \cos x, \qquad 0 \le x \le \frac{3135}{8192}. \tag{7}$$

The calculation flow of TA2$(x)$ is summarized in Fig. 3. We attain the maximal value of err2$(x)$ from Fig. 4 to be $0.8 \times 10^{-4} < 1.22 \times 10^{-4} = 1/2^{13}$, which concludes that the output resolution of our method is guaranteed to possess a 12-bit resolution. A trigonometric quadruple angle approximation with error correction for sinusoidal output is attained.

### C. System Design by Ten-Stage Pipelining

Fig. 5 is a typical implementation of DDFSs in prior works. "Phase accumulator" controlled by "frequency control" accu-

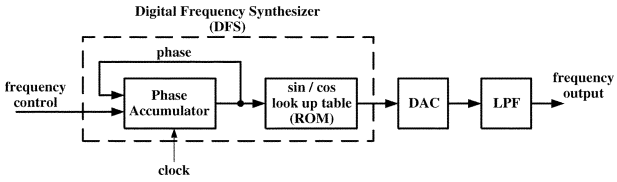


Fig. 5. Architecture of prior ROM-based DDFSs.

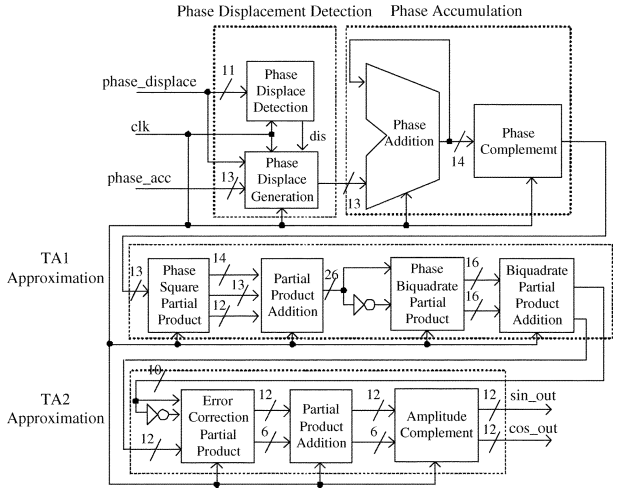


Fig. 6. Block diagram of the proposed ROM-less DDFS.

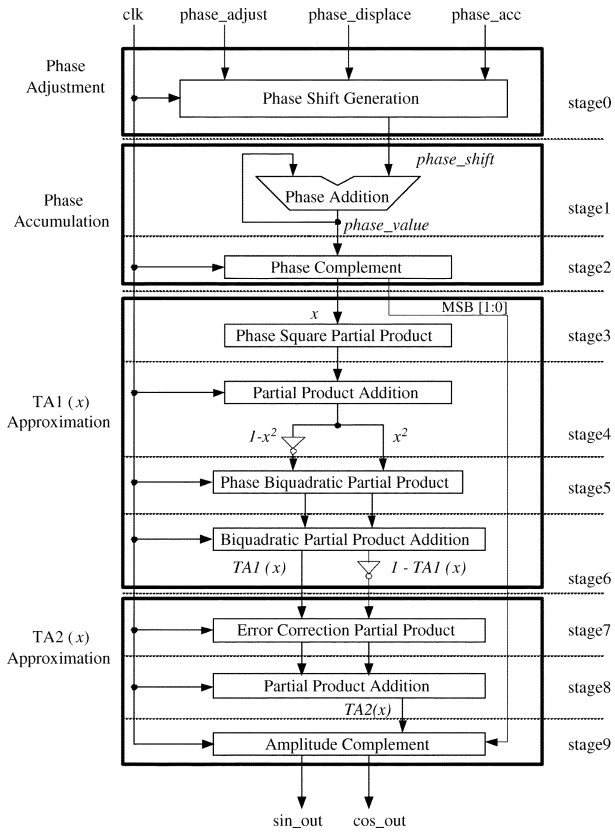


Fig. 7. Signal flow of the proposed ROM-less DDFS.

mulates "phase" at each clock cycle. "$\sin/\cos$ look-up table" outputs digital "$\sin/\cos$" values, given the value of "phase." Digital "$\sin/\cos$" values output by the "$\sin/\cos$ look-up table" pass through a digital-to-analog converter (DAC) and a low-pass filter (LPF) such that the "frequency output" is generated. The slow and large ROM not only occupies a significant portion of the chip (or board) area but also degrades the operation frequency. If a direct implementation of (3) and (6) is adopted, a large number of long multipliers are required, which will also degrade the operation frequency and require a large chip area. Hence, we propose our pipelining digital implementation based upon the proposed quadruple angle approximation method in Fig. 7.

The proposed pipelining design is used to replace the ROM-based digital frequency synthesizer (DFS) in Fig. 5 and speed up the operation frequency of the DDFS. The phase computation is also simplified. The detailed block diagram of the proposed DDFS is shown in Fig. 6. Meanwhile, the detailed signal flow of the proposed DDFS is depicted in Fig. 7. Before the detailed description of the pipeline architecture, several topics should be introduced.

*Squarer Decomposition:* A bottleneck in the early stage of the DDFS is the generation of phase square in (3) 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. The parallel squarer scheme [10] is adopted to reduce the complexity of computation. The 13 bits are decomposed into six high bits (denoted as $H$) and seven low bits (denoted as $L$). Thus, the squarer can be formulated as follows:

$$(HL)^2 = H^2 \cdot 2^{14} + L^2 + 2 \cdot H \cdot L \cdot 2^7. \tag{8}$$

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 (8) can be generated by a single adder. The squarer decomposition is divided into two pipeline stages, and the operation frequency can be kept.

*Phase Adjustment:* Equation (3) tells that the maximum count of the phase input is 3135 ($\approx$12 bits). Meanwhile, the resolution of the design is proved to be 13 bits. Hence, the maximum count of the phase_acc in Fig. 7 is 6270. The phase accumulation from 0 to 6270 corresponds to the cosine output from 0 to $\pi/2$. The 11 bits input at phase_displace is fed by a digital PLL (DPLLS; not shown) such that the phase comparison and adjustment can be carried out. The frequency of output sinusoidal wave is

$$f_{\text{out}} = \frac{\text{phase\_acc}}{6270 \times 4} \cdot f_{\text{clock}} \tag{9}$$

where $f_{\text{out}}$ is the frequency of the output sinusoidal waves, and $f_{\text{clock}}$ is the system clock. The theoretically maximum $f_{\text{out}}$ is $1/4 \cdot f_{\text{clock}}$.

In some applications such as the DPLLs, the DDFS needs to carry out the phase adjustment. "stage0" realizes the function of the phase adjustment and generates the next phase shift (phase_shift). The adjustable phase angle is

$$\text{adjustable phase angle} = \frac{\text{phase\_displace}}{6270} \cdot \frac{\pi}{2}. \tag{10}$$

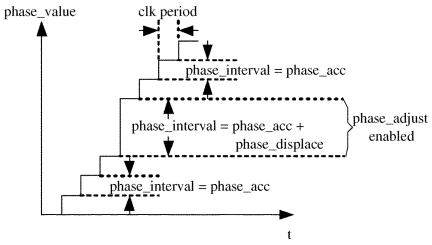


Fig. 8. Diagram of phase adjustment.

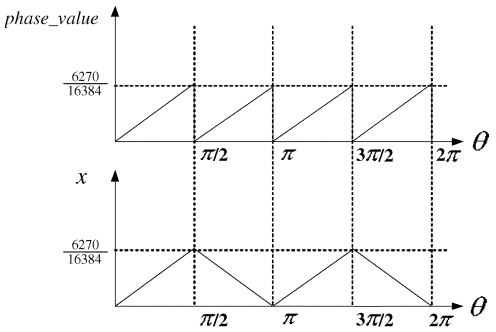


Fig. 9. Mapping from the output phase angle ($\theta$) to "phase_value" and $x$.

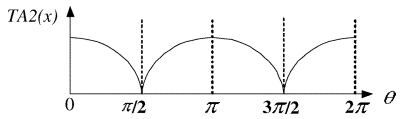


Fig. 10. Mapping from the phase angle to the output of the "partial product addition" TA2($x$).

As shown in Fig. 8, if the "phase_adjust" command is enabled, "phase_interval" equals "phase_acc" plus "adjust_displace". Hence, the function of phase adjustment is achieved.

*Phase Accumulation:* Referring to Fig. 7, "stage1" realizes the phase addition, which is accomplished by a modulo 6270 adder. The "phase_value", which is the output of the modulo 6270 adder, is fed to "stage2" to generate its complement "$x$", and "stage2" generates the 2-bit most significant bit (MSB) for the amplitude complement in "stage9". The mapping from the output phase angle of the proposed DDFS "$\theta$" to "phase_value" and "$x$" is shown in Fig. 9.

*TA1($x$) Approximation:* "stage3" to "stage6" realize TA1($x$) approximation, which was mentioned in (3), and generate $1 - \mathrm{TA1}(x)$ for TA2($x$) approximation. Both the phase square and phase biquadratic calculation are realized by the mentioned squarer decomposition.

*TA2($x$) Approximation:* "stage7" to "stage9" realize TA2($x$) approximation, which was mentioned in (6). After the processing of "phase complement" shown in Fig. 9, the output of "stage8" is $|\cos\_\mathrm{out}|$ as shown in Fig. 10. "stage9" implements the amplitude complement of "stage8". The mapping of "phase_value" and "cos_out" is shown in Fig. 11. "stage3" to "stage8" are duplicated to generate "$\sin\_\mathrm{out}$" and "$\cos\_\mathrm{out}$" simultaneously.

## III. Implementation and Physical Measurement

Modelsim of Mentor and MATLAB of Mathworks are the software tools to proceed the system-level simulations. The design in Fig. 7 is coded by register transfer level (RTL) Verilog, which is then simulated by Modelsim. The RTL code is then synthesized by SYNOPSYS and postlayoutedly verified by TimeMill and PowerMill. Fig. 12 shows the transition from a 5.55-MHz sine output to a 41.66-MHz O/P given the worst condition (SS Model, 0 °C, $f_{\mathrm{clock}} = 166$ MHz). Meanwhile, the decimal output data in a 12-bit format are collected. The fast Fourier transform (FFT) command of MATLAB is executed to attain the spectrum as shown in Fig. 13, which illustrates that the spurious performance of the proposed method is as high as $-130$ dBc. It is far better than any prior works. Table I summarizes the performance of our work and prior methods.

We then follow the standard cell-based design flow to implement our work. Fig. 14 is the die photo of the proposed design. Table II summarizes the characteristics of the chip. The proposed design has been silicon-proved by the Taiwan Semiconductor Manufacturing Company (TSMC) 0.35-$\mu$m 2P4M CMOS process. The proposed prototype is verified by the IMS100 Logic Master, which shows that the measured data exactly match the expected results. The time-domain measurement is performed using an Agilent 17702B logic analyzer. Fig. 15 shows the test condition showing that the measured maximum system clock rate is 180 MHz. Fig. 16 shows the

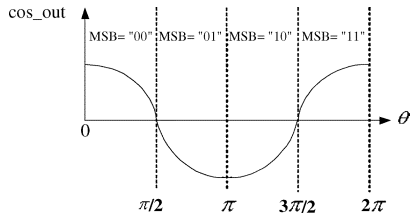


Fig. 11. Mapping of the output phase angle ($\theta$) to "$\cos_$out."

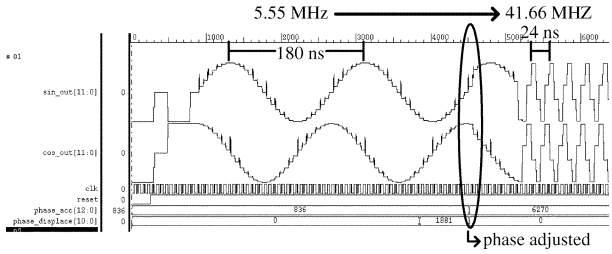


Fig. 12. Postlayout simulation results.

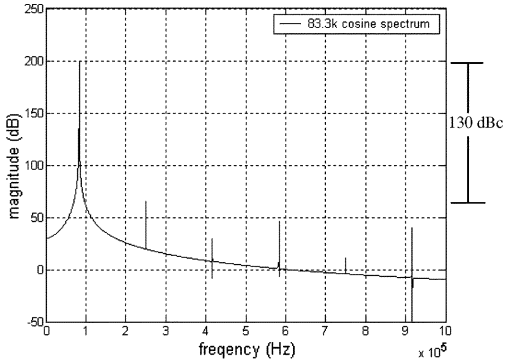


Fig. 13. Spurious performance of the proposed DDFS.

TABLE I
PERFORMANCE COMPARISON

| | resolution | spurious | max. clock rate | area |
|---|---|---|---|---|
| [3] | 10 bits | -55 dBc | 800 MHz | 15.58 $mm^2$ |
| [1] | 12 bits | -55 dBc | 500 MHz | 2.8 $mm^2$ |
| [6] | 12 bits | -98.75 dBc | N/A | N/A |
| [7] | 12 bits | -70 dBc | 160 MHz | N/A |
| [11] | 12 bits | -66.8 dBc | N/A | N/A |
| ours | 13 bits | -130 dBc | 180 MHz | 5.07 $mm^2$ |

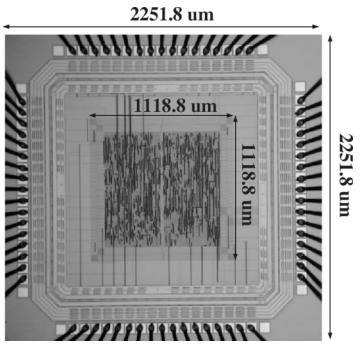


Fig. 14. Die photo of the proposed design.

TABLE II
CHARACTERISTICS OF THE PROPOSED DDFS

| | |
|---|---|
| $f_{clock}$ | 180 MHz |
| max. $f_{out}$ | 41.66 MHz |
| avg. power | 16.774 mW |
| resolution | 13 bits |
| core area | 1118.8×1118.8 $\mu m^2$ |

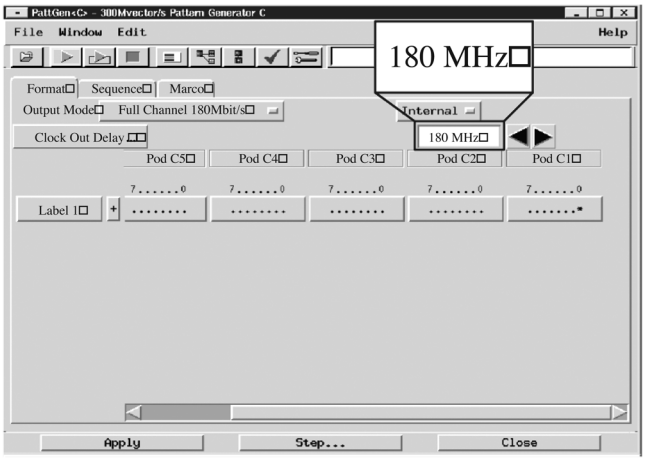


Fig. 15. Testing condition in the Agilent 16702B logic analyzer with the maximum operating frequency (180 MHz).

waveform of the synthesized cosine wave derived from the measured samples.

## IV. CONCLUSION

In this brief, 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, but the second-order error correction has also been simulated to justify the capability of subsiding the noise power of the harmonics. The proposed design is fully proven on silicon to reveal its feasibility.

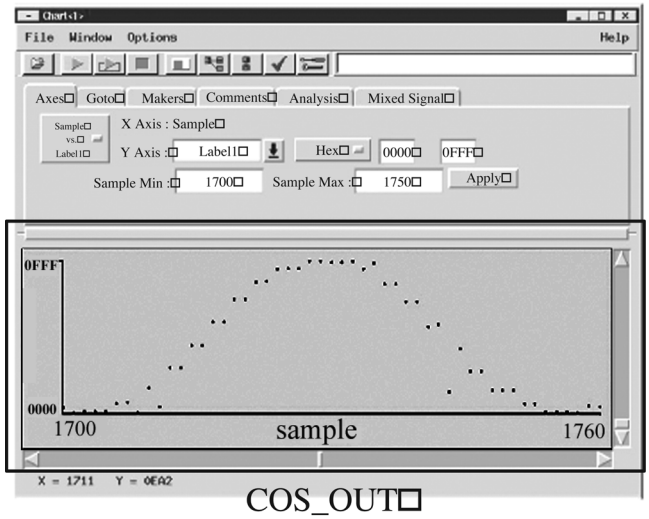


Fig. 16. Waveform of the synthesized cosine wave derived from the measured samples the Agilent 16702B logic analyzer.

## ACKNOWLEDGMENT

The authors would like to thank the Chip Implementation Center (CIC) of National Applied Research Laboratories (NAPL), Taiwan, R.O.C., for their thoughtful chip fabrication service.

## REFERENCES

[1] G. W. Kent and N.-H. Sheng, "A high purity, high speed direct digital synthesizer," in *Proc. 49th IEEE Int. Frequency Control Symp.*, 1995, pp. 207–211.

[2] V. F. Kroupa, *Direct Digital Frequency Synthesizers*. Piscataway, NJ: IEEE Press, 1998.

[3] G. Van Andrews *et al.*, "Recent progress in wideband monolithic direct digital synthesizers," in *Proc. IEEE MTT-S Inter. Microw. Symp. Dig.*, 1996, vol. 3, pp. 1347–1350.

[4] M. J. Flanagan and G. A. Zimmerman, "Spur-reduced digital sinusoid synthesis," *IEEE Trans. Commun.*, vol. 43, no. 7, pp. 2254–2262, Jul. 1995.

[5] V. F. Kroupa, V. Cizek, J. Stursa, and H. Svandova, "Spurious signals in direct digital frequency synthesizers due to the phase truncation," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 47, no. 5, pp. 1166–1172, Sep. 2000.

[6] R. Larson and S.-L. Lu, "Interpolation-based digital quadrature frequency synthesizer," in *Proc. 13th Annu. IEEE Int. ASIC/SOC Conf.*, 2000, pp. 48–52.

[7] K. I. Palomaki and J. Niittylahti, "Direct digital frequency synthesizer architecture based on Chebyshev approximation," in *Proc. 34th Asilomar Conf. Signals, Syst. and Comput.*, 2000, vol. 2, pp. 1639–1643.

[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. Circuits Syst. II, Expr. Briefs*, vol. 48, no. 9, pp. 850–857, Sep. 2001.

[9] C.-C. Wang, Y.-L. Tseng, H.-C. She, C.-C. Li, and R. Hu, "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.

[10] J. Yoo, K. F. Smith, and G. Gopalakrishnan, "A fast parallel squarer based on divide-and-conquer," *IEEE J. Solid-State Circuits*, vol. 32, no. 6, pp. 909–912, Jun. 1997.

[11] A. M. Sodagar, G. R. Lahiji, and A. Azarpeyvand, "A novel architecture for sine-output direct digital frequency synthesizers using parabolic approximation," in *Proc. 7th IEEE ICECS*, Sep. 2000, vol. 1, pp. 256–259.