LEAST SQUARES APPROXIMATION-BASED ROM-FREE DIRECT DIGITAL FREQUENCY SYNTHESIZER

Ching-Hua Wen, Huai-Yi Hsu, Hung Yang Ko, and An-Yeu (Andy) Wu

Graduate Institute of Electronics Engineering, and
Department of Electrical Engineering,
National Taiwan University, Taipei, 106, Taiwan, R.O.C.

ABSTRACT

This paper describes a new design approach and an architecture for a *Direct Digital Frequency Synthesizer* (DDFS) based on *Least Square* (LS) approximation. It is shown that the architecture can be implemented as a low-cost, low-power, feedforward, and easily pipelineable datapath. A prototype IC has been designed and fabricated in TSMC 0.25 *um* CMOS technology. The IC produces 14-bit sine and cosine outputs with a spurious free dynamic range of 100 dBc. A 32-bit frequency word gives a tuning resolution of 0.0466 Hz at 200 MHz sampling rate.

1. INTRODUCTION

Direct Digital Frequency Synthesizers (DDFS) [1] is an important component in modern of high-performance communication systems due to their advantageous high frequency resolution, low phase noise and fast frequency switching speed. The DDFS essentially consists of the phase accumulator and phase-to-amplitude converter. Then a *Digital-to-Analog Converter* (DAC) can be used to output an analog sinusoidal waveform. A low pass filter can then be included to smooth the continuous analog sinusoid waveforms as shown in Fig. 1. Given a *Frequency Control Word* (FCW), the DDFS can output sinusoidal waveforms of frequency:

$$f_{out} = FCW * f_{CLK} / 2^{L}. \quad (1)$$

The FCW is an integer ranging from 0 to $2^{(L-1)}$, and the minimum frequency resolution is

$$f_{\min} = f_{CLK} / 2^{L}. \quad (2)$$

Traditionally, the phase-to-amplitude converter is implemented with ROM. However the ROM-based design will lead to large area requirement and power consumption for good synthesized signals and frequency resolution. Many techniques [2] have been proposed to reduce the problems of the power and hardware cost. In this paper, we propose a novel ROM-free DDFS design based on LS algorithm [3]. Based on LS algorithm, 4th order polynomials can be generated to achieve 100dBc *Spurious Free Dynamic Range* (SFDR) performance. Besides, the proposed DDFS design can be implemented with only 2 squarer circuits and 1 multiplier.

This work was supported in part by the MediaTek Inc., under NTU-MTK wireless research project.

Hence the DDFS can synthesize superior sinusoidal signal with fewer arithmetic operations. The FCW is 32-bit wide and the sine/cosine output is 14-bit wide to achieve 100dBc SFDR. These properties will make low cost, low power, high frequency resolution and high spectral purity for the DDFS operation. The DDFS has been realized with TSMC 0.25 *um* 1P5M CMOS technology. The core area is 0.58x0.58 mm^2 and the die size is 1.8x1.8 mm^2.

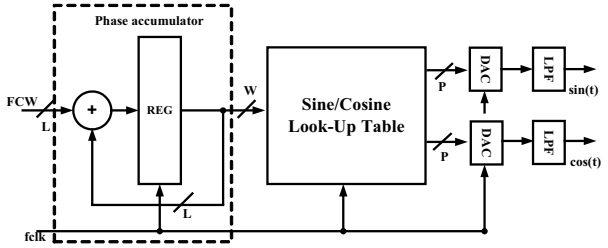

Fig.1. Block diagram of DDFS

2. LEAST-SQUARE (LS) APPROXIMATION

LS algorithm [3] can calculate the best-fit curve that has the minimal sum of the deviations squared (LS errors) from a given set of data. Suppose that the data points are (x_1, y_1), (x_2, y_2), …., (x_n, y_n), where x is the independent variable and y is the dependent variable. The fitting curve $f(x)$ has error d_i from each data point y_i. That is, $d_1=y_1-f(x_1)$, $d_2=y_2-f(x_2)$, …., $d_n=y_n-f(x_n)$. According to Eq. (3), the best fitting curve can be achieved by minimizing

$$\xi = \min\left\{ d_1^{\,2} + d_2^{\,2} + ... + d_n^{\,2} = \sum_{i=1}^{n} d_i^{\,2} \right\} = \min\{ \sum_{i=1}^{n} [y_i - f(x_i)]^2 \}. \quad (3)$$

A general polynomial is one of the most commonly used types of curves in regression. The applications of the method of least squares curve fitting using polynomials are briefly discussed as follows. We take the general case of m^{th} order polynomial as an example. LS algorithm can use m^{th} order polynomial to approximate the given set of data. The following is the complete derivation.

When using an m^{th} order polynomial

$$f(x_i) = a_0 + a_1 x + ... + a_{m-1} x^{m-1} + a_m x^m. \quad (4)$$

to approximate the given set of data, (x_1, y_1), (x_2, y_2), ….. , (x_n, y_n), where $n \geq m + 1$, the best fitting curve $f(x)$ has the least square error as,

$$\xi = \min\{\sum_{i=1}^{n}[y_i - f(x_i)]^2\} = \min\{\Pi\}. \quad (5)$$

a_0, a_1, …, and a_m are unknown coefficients whereas all x_i and y_i are given. To obtain the least square error, the unknown coefficient a_0, a_1, …, and a_m can be obtained by setting first derivatives to zero.

$$\begin{cases} \frac{\partial \Pi}{\partial a_0} = 2\sum_{i=1}^{n}[y_i - f(x_i)] = 0, \\ \frac{\partial \Pi}{\partial a_1} = 2\sum_{i=1}^{n}x_i[y_i - f(x_i)] = 0, \\ \frac{\partial \Pi}{\partial a_2} = 2\sum_{i=1}^{n}x_i^2[y_i - f(x_i)] = 0, \\ \quad . \\ \quad . \\ \frac{\partial \Pi}{\partial a_m} = 2\sum_{i=1}^{n}x_i^m[y_i - f(x_i)] = 0. \end{cases} \quad (6)$$

Expanding the above equations, we have

$$\begin{cases} \sum_{i=1}^{n}y_i = a_0\sum_{i=1}^{n}1 + a_1\sum_{i=1}^{n}x_i + ...+ a_m\sum_{i=1}^{n}x_i^m, \\ \sum_{i=1}^{n}x_iy_i = a_0\sum_{i=1}^{n}x_i + a_1\sum_{i=1}^{n}x_i^2 + ...+ a_m\sum_{i=1}^{n}x_i^{m+1}, \\ \sum_{i=1}^{n}x_i^2y_i = a_0\sum_{i=1}^{n}x_i^2 + a_1\sum_{i=1}^{n}x_i^3 + ...+ a_m\sum_{i=1}^{n}x_i^{m+2}, \\ \quad . \\ \quad . \\ \sum_{i=1}^{n}x_i^my_i = a_0\sum_{i=1}^{n}x_i^m + a_1\sum_{i=1}^{n}x_i^{m+1} + ...+ a_m\sum_{i=1}^{n}x_i^{2m}. \end{cases} \quad (7)$$

Then, the unknown coefficients a_0, a_1, …, and a_m can be obtained from the $m+1$ linear equations. LS algorithm can generate m^{th} order approximation polynomials with the above derivations.

3. PHASE-TO-AMPLITUDE CONVERTER

Synthesizing sinusoid waves with high spectral purity is one of the major goals for the frequency synthesizer designs. For high spectral purity, a good phase-to-amplitude converter design is the most important issue in DDFS design. In this paper, we adopt a LS algorithm to propose a novel DDFS design. We also use the symmetric properties about sinusoidal signals to reduce the hardware requirement. Besides, the symmetric properties can improve the approximated waveforms. The comparison with Taylor-series [4] can demonstrate its advantage in hardware cost and SFDR performance.

3.1. Symmetry Property of Sine and Cosine

Forth Wave Symmetric (π/2-symmetry) and *Eighth Wave Symmetry (π/4-symmetry)* are often applied on the phase-to-amplitude converter to reduce the hardware complexity. The equation for π/2-symmetry and π/4-symmetry are shown in Eqs. (8)~(11).

$$\sin(\theta) = \begin{cases} \sin(\theta), & \theta \in [0, \frac{\pi}{2}] \\ \sin(\pi - \theta), & \theta \in [\frac{\pi}{2}, \pi] \\ -\sin(\theta - \pi), & \theta \in [\pi, \frac{3\pi}{2}] \\ -\sin(2\pi - \theta), & \theta \in [\frac{3\pi}{2}, 2\pi] \end{cases} \quad (8)$$

$$\cos(\theta) = \begin{cases} \cos(\theta), & \theta \in [0, \frac{\pi}{2}] \\ -\cos(\pi - \theta), & \theta \in [\frac{\pi}{2}, \pi] \\ -\cos(\theta - \pi), & \theta \in [\pi, \frac{3\pi}{2}] \\ \cos(2\pi - \theta), & \theta \in [\frac{3\pi}{2}, 2\pi] \end{cases} \quad (9)$$

$$\sin(\theta) = \begin{cases} \sin(\theta), & \theta \in [0, \frac{\pi}{4}] \\ \cos(\frac{\pi}{2} - \theta), & \theta \in [\frac{\pi}{4}, \frac{\pi}{2}] \end{cases} \quad (10)$$

$$\cos(\theta) = \begin{cases} \cos(\theta), & \theta \in [0, \frac{\pi}{4}] \\ \sin(\frac{\pi}{2} - \theta), & \theta \in [\frac{\pi}{4}, \frac{\pi}{2}] \end{cases} \quad (11)$$

As shown in Fig. 2, for the synthesized Sine waveform, the smaller phase range can result in better approximation result in LS algorithm. This phase limitation [0, π/4] will improve the approximation result over the limitation [0, 2π]. The approximated sine and cosine waveforms in [0, π/4] can be much better than the approximated waveforms in one full period.

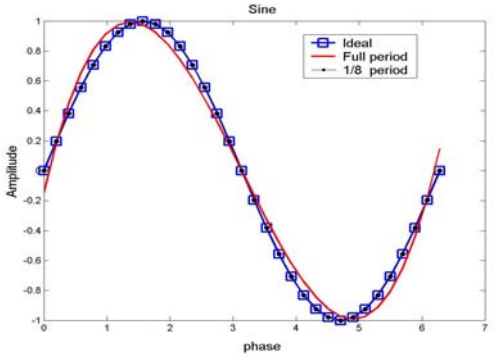

Fig. 2. Synthesized Sine Waveform in [0, 2π], [0, π/4] and Ideal Sine Waveform

Therefore, the required of amplitude values to be represented can be reduced to one eighth of the original when the sine and cosine symmetry is utilized. In this small interval, [0, π/4], the complexity of the phase-to-amplitude converter can be reduced substantially.

3.2. SFDR Performance

In order to express the advantage of the proposed LS approximation we use Fig. 3 to illustrate LS algorithm can achieve the same SFDR by using lower order number than Taylor series. For our expected SFDR=100dBc, LS approximation only requires 4th order polynomial while the Taylor series requires a 6th order polynomial to achieve it. We can use fewer multipliers to implement DDFS with higher spectral purity using the LS algorithm.

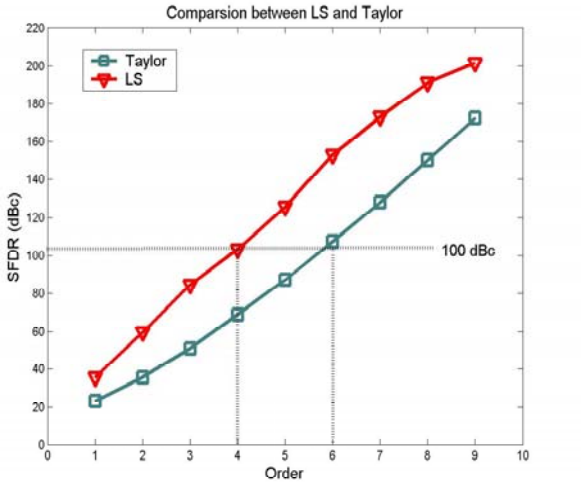

Fig. 3. SFDR Comparison between LS and Taylor approximation.

4. SYSTEM ARCHITECTURE

4.1 Finite Wordlength Effect

The optimization of DDFS performance involves trading off the finite wordlength and sinusoid computation method against the sine and cosine wave spectral purity and maximum clock rate. Fig. 4 shows a basic block diagram of a DDFS that identifies the three basic sources of noise inherent to all DDFS designs. These noise sources are phase quantization, amplitude quantization and sine/cosine function compression distortion. For this figure, pipelining isn't considered. In this paper, the wordlength of the accumulator, *L*, is designed as 32-bit for tuning frequency resolution < 1 Hz and the phase-to-amplitude converter is based on a 4th order LS approximation. We will use the SFDR criterion to decide the value of *W* and *P*. From Matlab and FFT simulation, we set the truncated accumulated phase wordlength to *W*=17 bits and amplitude wordlength to *P*=14 bits. These hardware parameters can achieve SFDR=100dBc spectral purity performance.

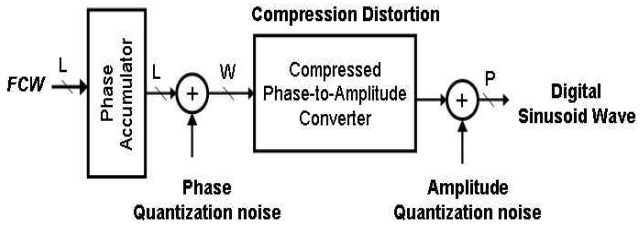

Fig.4. Finite Wordlength Effects of DDFS.

4.2 System Design

Fig. 5 is the hardware architecture of the proposed LS-based DDFS design with 100 dBc SFDR. In order to improve the system clock, we arrange 6 pipeline stages in the DDFS design. The output spectrum after fixed-point and FFT simulation is shown in Fig. 6. It shows this hardware architecture can achieve 100 dBc SFDR successfully.

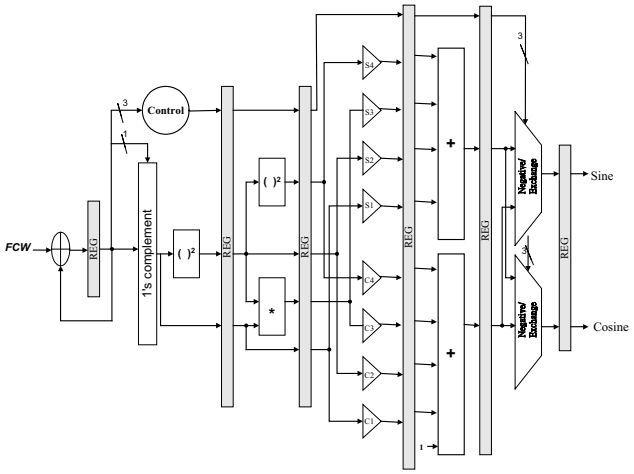

Fig. 5. Detailed and pipelined Architecture of LS-based DDFS.

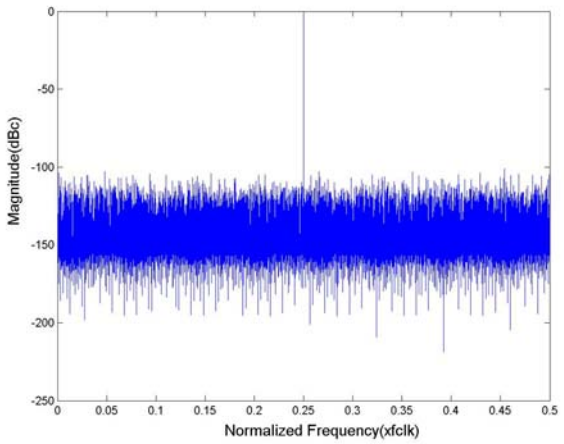

Fig. 6. Output Spectrum of LS-based DDFS with Finite Wordlength(FCW=1/4).

5. IMPLEMENTATION RESULTS AND COMPARISION

The proposed DDFS design described was implemented in TSMC 0.25 *um* 1P5M CMOS technology. The microphotograph of the LS approximation-based DDFS is shown in Fig. 7.

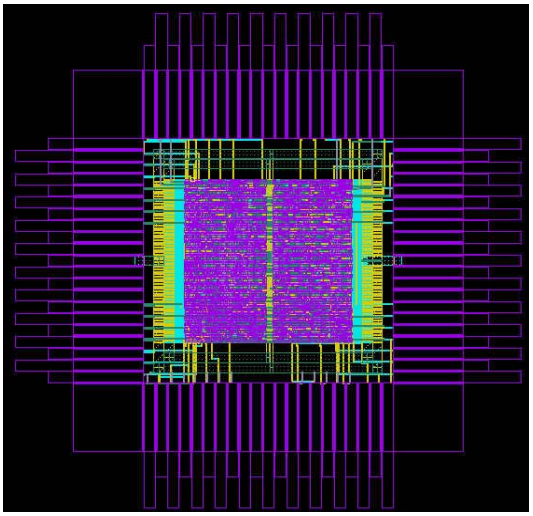

Fig. 7. Microphotograph of the proposed DDFS.

Table 1. Implement result of the LS-based DDFS.

Technology	TSMC 0.25 um 1P5M CMOS
Voltage	3.3 V
Amplitude resolution	14 bits
Core Size	0.58x0.58 mm^2
Die Size	1.36x1.36 mm^2
Max Frequency	200MHz
Latency	6 cycles
Power consumption	96 mW
SFDR	100 dBc

In order to eliminate the factor of different fabrication technology, we adopt the *Normalize Index* [5]. The *Normalized Area* is the silicon area normalized to a 1 *um* technology, as shown below:

$$Normalized\,Area = \frac{Area}{(Technology/1\mu m)^2} \tag{12}$$

The *Power Efficiency* (*Pe*), which compares the number of DDFS calculation per MHz is shown in below:

$$Power\ Efficiency = \frac{Power}{Frequency} \tag{13}$$

From the performance table in Table 2, we can see that the proposed LS-DDFS scheme has pretty good performance according to the normalize index.

Table 2. Comparison with the existing DDFS designs.

DDFS	CMOS tech. (mm)	SFDR (dBc)	Nor. Area (mm2)	Latency (cycle)	Max. Freq. (MHz)	*Pe* (mw/MHz)
This Work	0.25	100	5.44	6	200	0.48
LUT [6]	0.25	90.36	5.76	13	150	2.67
Appro. [7]	0.35	80	3.6	1	88	0.44
Appro. [8]	0.5	59 (DAC)	5.6	-	100	0.08
CORDIC [9]	1.0	100	12	16	100	14

6. CONCLUSIONS

Based on LS algorithm, we define and realize the LS-based ROM-free DDFS. The LS-based DDFS can avoid the speed-down by memory device and achieve 100dBc spectral purity. In the comparison, the proposed DDFS requires less area and power consumption. For today and future communication application, the proposed LS-based DDFS can meet the needs of portability, cost, power, speed, and spectral purity for modern SOC designs.

7. REFERENCES

[1] J. Tierney, C. M. Rader, and B. Gold, "A digital Frequency Synthesizer," *IEEE Trans. on Audio Electroacoust*., vol. AU-19, no. 1, pp. 48-57, Mar. 1971.

[2] J. Vankka, "Methods of mapping from phase to sine amplitude in direct digital synthesis" *IEEE Trans. on Ferroelectrics and Frequency Control*, vol. 44, pp. 526-534, Mar. 1997.

[3] M. Flickner, J. Hafner, and E.J. Rodriguez, and J.L.C. Sanz, "Fast least-squares curve fitting using quasi-orthogonal splines," *IEEE Int. Image Processing*, vol. 1, pp. 686-690, Nov. 1994.

[4] K.I. Palomaki and J. Niittylahti, to "Methods to improve the performance of quadrature phase-to-amplitude conversion based on Taylor series approximation," in *Proc. 43th IEEE Symp. on Circuits and Systems*, vol. 1, pp. 14-17, 2000.

[5] M. Baas, "A Low-Power High-Performance, 1024-Point FFT Processor," *IEEE J. of Solid-State Circuits*, vol. 34 no. 3, pp. 380-387, Mar 1999.

[6] A. Torosyan, Dengrwei Fu, Jr. Willson A. N., "A 300 MHz quadrature direct digital synthesizer/mixer in 0.25 μm CMOS," in *IEEE Solid-State Circuits Conference*, vol. 1, pp. 132 -133, 2002.

[7] D. De Caro, E. Napoli, A. G. M. Strollo, "Direct digital frequency synthesizers using high-order polynomial approximation," *IEEE* Solid-State Circuits Conference, vol. 1 , pp. 134 -135, 2002.

[8] A. N. Mohieldin, A.A. Emira, E. Sanchez-Sinencio, "A 100-MHz 8-mW ROM-less quadrature direct digital frequency synthesizer," *IEEE J. of Solid-State Circuits*, vol. 37, pp. 1235-1243, Oct. 2002.

[9] A. Madisetti, A.Y. Kwentus, A.N. Willson and Jr., "A 100-MHz, 16-b, direct digital frequency synthesizer with a 100-dBc spurious-free dynamic range" *IEEE J. of Solid-State Circuits*, vol. 34, pp. 1034-1043, Aug. 1999.