AN EFFICIENT FFT TWIDDLE FACTOR GENERATOR

Jen-Chuan Chi and Sau-Gee Chen

National Chiao Tung University
Department of Electronics Engineering and Institute of Electronics
1001 Ta Hsueh Rd, Hsinchu, Taiwan, ROC
web: http://tomato.ee.nctu.edu.tw/

ABSTRACT

Twiddle factor (TF) generator is a key component in IFFT/FFT computation for current OFDM-based wireless and wireline communication systems, such as IEEE 802.11a/g, 802.16, VDSL, DVB, DAB systems. In this paper, we propose an efficient TF generator design. The proposed design combines an efficient conventional recursive sine/cosine function generation algorithm with a small compensation lookup table. The compensation table solves the error propagation problem of a recursive TF generator. It also costs less area than the existing modified recursive sine/cosine function generator. The proposed design requires only two constant multipliers, two adders, and two correction tables, which is smaller than those polynomial-based complex sine function generators. Simulation results show that the proposed design is favourable over the existing TF generators for general FFT lengths in practical applications, except for small lengths such as 64 up to 512. For those small FFT lengths, ROM-based is a better choice.

1. INTRODUCTION

In modern communication systems, the OFDM (Orthogonal Frequency Division Multiplexing) algorithm is effective in combating the problems of frequency-selective fading, inter-symbol interference (ISI), and inter-frequency interference (IFI). It is also efficient for wideband data transmission. Therefore, OFDM techniques plays an important role in modern wireless and wireline communication systems, such as 820.11a/g, 802.16, DVB, DAB, VDSL, and so on. In these systems, DFT/IDFT (realized by FFT/IFFT S/W and H/W) is the core component in achieving OFDM transmission.

Those systems require FFT/IFFT lengths ranging from 64 to 8192. There are numerous FFT/IFFT designs in the literature. Most of them are devoted to efficient realization of the core FFT architecture and butterfly design. The required twiddle factors in FFT are generally assumed stored in memories in advance and retrieved for butterfly multiplication whenever necessary. This normally ends up with a very large lookup table in comparison with the core FFT processing elements and main data memory, particularly for large FFT lengths such as 8192. Therefore, an efficient TF generator with small area and high speed performance is indispensable, considering especially for portable and high data rate design. In the past, TF generation techniques were somewhat ignored due to the fact that OFDM systems were not as pervasive as they are now. It was mostly applied to off-line, non-real-time applications. However, there exist many popular generation techniques for trigonometric functions, which can be applied to TF generation. Those computing techniques are mostly used for the designs of direct digital frequency synthesizer (DDFS). They include CORDIC algorithms [1], [2], [3], polynomial-based approach [4], [5], [6], [7], [8], [9], [10], the most popular ROM-based table-lookup scheme [10], and the recursive function generators [12], [13].

The conventional CORDIC-based TF generations are by-products from the CORDIC-based butterfly rotation operations. As a result, virtually there is no need to generate the TF. It is the biggest advantage of CORDIC-based FFT. In addition, it is realized by simple shift-and-add operations. However, they have the disadvantage of slow sequential computation [1] and the need of extra scale factor compensation operations. In [2], [3], pipelined CORDIC architectures are proposed for boosting speed performance, at the cost of extra area, pipe latency and control complexity. In [4], the pipelined CORDIC architecture achieves a better speed performance, meanwhile costs smaller area than the previous two, by using a fast rotation prediction scheme and radix-4 CORDIC operations. However, pipelined structures are still too hardware consuming for practical FFT applications. The CORDIC-based designs will be compared with our design latter.

Twiddle factors can also be efficiently generated by the polynomial-based DDFS [5], [6], [7], [8], [9], [10], [11]. They are based on the concept of piecewise polynomial approximation to a function value, generally need memory space for the storage of polynomial coefficients and/or some function values, and some multiplication and addition operations to generate an output function point. In addition, normally complexities of polynomial-based algorithms will grow with the output precision significantly. In [5], the authors partition the parameter ranges into regions and use a second-order polynomial approximation, combine with Horner's rule to compute the sine and cosine values. This approach needs four multipliers, four adders and two lookup tables. As such, it has high complexity. In addition, if a higher output precision is desired, then the parameter range has to be partitioned into more regions, and accordingly a higher computation complexity will be required. The designs in [6], [7], [8], [9], [10], [11] are also based on similar design concepts, and they have the same problem.

The ROM-based table-lookup method is the simplest and most popular approach for TF generation. It takes advantage of sine function symmetry that reduces the table size to one-eighth of its original. It is good for short-length FFT/IFFT computation, such as 64-point FFT in 802.11a. However, the table size will grows intolerably large for large FFT lengths such as 8192-point FFT in the DVB and VDSL systems.

The recursive sine function generator [12] is based on the well-known recursive feedback difference equation for the computation of sine and cosine functions. The approach's key advantage is low complexity. For combined sine and cosine function generations, the approach only needs two multiplication and two addition operations per output point, which is less complicated than the mentioned CORDIC, table-lookup and polynomial designs. However, its serious drawback is the introduced error propagation problem in finite-precision computation, due to its error feedback nature. In [13], the authors proposed a solution to alleviate the error-propagation problem. However, this approach needs two extra multiplications and corresponding area to correct the propagation error.

In this paper, considering the low-complexity merit of the recursive sine function generator, we attempt to reduce its main drawback of error-propagation problem with low overhead. We will first analyze the error propagation bound of this approach and derive two error correction tables (for both sine and cosine functions), for the compensation the propagated finite-precision error. In all, the new design needs two constant multipliers and two adders to calculate these two functions. The correction table entries are at most three bit long, regardless of output word lengths. As a result, the proposed design has a smaller area than those mentioned designs in most cases, as will be detailed later. In section 2, we will introduce the new design, followed by simulations, discussion and performance comparison in section 3. In this section, we will also conduct EDA hardware simulation to verify its performance.

2. THE PROPOSED RECURSIVE TWIDDLE FACTOR GENERATOR

Let's first consider the generation of the imaginary part of a twiddle factor, that is, the sine function

$$x[n] = (\sin n\theta)u[n]$$

In z-transform, one can get the following equation [14]

$$H(z) = Z\{[\sin n\theta]u[n]\} = \sum_{n=0}^{\infty}(\sin n\theta)z^{-n} = \frac{1}{2j}\sum_{n=0}^{\infty}(e^{jn\theta} - e^{-jn\theta})z^{-n} \quad (2\text{-}1)$$
$$= \frac{1}{2j}\left(\frac{1}{1-e^{j\theta}z^{-1}} - \frac{1}{1-e^{-j\theta}z^{-1}}\right) = \frac{(\sin\theta)z^{-1}}{1-2(\cos\theta)z^{-1}+z^{-2}}, |z|>1$$

From the z-transform result, the corresponding difference equations for the generation of sine function, and similarly the cosine functions are:

$$\sin n\theta = 2\cos\theta \times \sin(n-1)\theta - \sin(n-2)\theta$$
$$\cos n\theta = 2\cos\theta \times \cos(n-1)\theta - \cos(n-2)\theta \quad (2\text{-}2)$$

Since in the equations, there are feedback terms of the previous generated output to the present output computation, there will introduce error propagation problem in finite-precision calculation. The reduction method for finite-precision error propagation in [13] needs extra multipliers, adders and computation. To achieve a low-complexity error correction, here we include two error correction tables to recover the full-precision outputs. To decide the word length of the correction tables, we first analyze the upper bound of the finite-precision error propagation as follows.

$$\begin{aligned}\overline{\sin m\theta} &= 2\overline{\cos\theta}\cdot\overline{\sin(m-1)\theta} - \overline{\sin(m-2)\theta} \\ &= 2(\cos\theta \pm 2^{-(n+1)})[\sin(m-1)\theta \pm 2^{-(n+1)}] - [\sin(m-2)\theta \pm 2^{-(n+1)}] \\ &= 2\cos\theta\cdot\sin(m-1)\theta - \sin(m-2)\theta \\ &\quad \pm 2^{-n}[\sin(m-1)\theta \pm \cos\theta \pm 2^{-1}] \pm 2^{-(2n+1)} \\ &= \sin m\theta \pm 2^{-n}[\sin(m-1)\theta \pm \cos\theta \pm 2^{-1} \pm 2^{-(n+1)}]\end{aligned} \quad (2\text{-}3)$$

Here $\overline{\cos\theta}$, $\overline{\sin(m-1)\theta}$ and $\overline{\sin(m-2)\theta}$ represent the finite-precision values of $\cos\theta$, $\sin(m-1)\theta$ and $\sin(m-2)\theta$, respectively, all with n-bit fractional precision. Then, the maximum error of the computed $\overline{\sin m\theta}$ is bounded by:

$$\max\left|2^{-n}[\sin(m-1)\theta \pm \cos\theta \pm 2^{-1} \pm 2^{-(n+1)}]\right| < 2^{-(n-2)} \quad (2\text{-}4)$$

This result shows a maximum two-bit error. Similar analysis results can be obtained with the cosine function.

From this result, we can conclude an error correction word of three-bit long per output point. Specifically, in every iteration cycle, we will replace the 3 LSB's of the currently generated output with the correction value stored in the correction tables. As a result, there will be no error propagation problem, and all the generated TF's will be guaranteed full n-bit precision. In all, we need two tables (one for cosine and the other for sine values) of $3\times N/8$ bits each for total N TF values in the angle range of 0 to 2π. The factor 1/8 is due to table redundancy reduction by considering the symmetric property of trigonometric functions. This greatly reduces the required table size. Figure 1 shows the architecture of the proposed generator for the real part (the cosine function) of a TF. The same structure can be obtained for the TF imaginary part (the sine function), except that an additional table for the initial sine values $\sin\theta$ should be included. Since both sine and cosine generators are exactly the same, they can be also combined as one interleaving structure by including simple control, a few registers, and a MUX after the correction-table stage, as shown in Fig. 2. Considering the radix-2 DIF N-point FFT as an example, where $N=2^K$, operation steps of the proposed design can be summarized as:

(1) Set the FFT stage number k=0
(2) Set the FFT subgroup number i=0, within the kth stage
(3) Set the Initial values $\overline{\sin n\theta}=0$ and $\overline{\cos n\theta}=1$, n=0, and output the TF values to the FFT processing element.
(4) Load $\overline{\cos\theta}=\overline{\cos(2\pi/M)}$ and $\overline{\sin\theta}=\overline{\sin(2\pi/M)}$ from a small table of size of $K-2=(\log N)-2$ words, where $M=N/2^k$. Output the TF values to the FFT processing element. Set the twiddle factor sequence number n=2.
(5) Compute the finite-precision $\overline{\sin n\theta}$ and $\overline{\cos n\theta}$ according to equation (2-2).

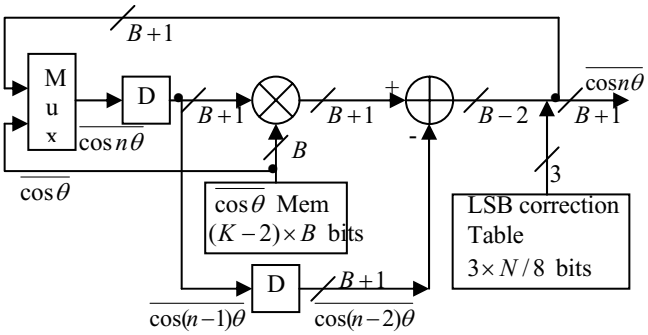

Figure 1. Architecture of the proposed generator for real part (cosine) of a twiddle factor.

Table 1. Area and speed simulation results of the proposed TF generator, wordlength B=16 bits.

FFT length	Total area	Delay	Max. operation freq
64	2316	3.01	332MHz
256	2437	3.21	311MHz
512	2798	3.33	300MHz
1024	3267	3.45	290MHz
2048	3612	3.50	286MHz
4096	4512	3.65	274MHz
8192	8983	3.71	270MHz

Note: The area unit is gate; The Delay time unit is ns

Table 2. The area comparison of TF generators, B=8 bits.

	64 points	4096 points	8192 points
Proposed	1128	3818	6924
CORDIC [2]	7721	7721	7721
CORDIC [3]	7238	7238	7238
Polynomial [6]	7255	7255	7255
Polynomial [7]	8021	8021	8021
Polynomial [8]	7217	7217	7217
ROM based	264	10121	20131
Recursive [13]	7559	7559	7559

Table 3. Performance comparison of TF generators, B=16.

	64-point		4096-point		8192-point	
	Area	Delay	Area	Delay	Area	Delay
Proposed	2316	3.01	4512	3.65	8983	3.71
CORDIC [2]	10181	2.51	10181	2.51	10181	2.51
CORDIC [3]	9781	2.72	9781	2.72	9781	2.72
Polynomial [6]	9812	4.12	9812	4.12	9812	4.12
Polynomial [7]	9125	4.23	9125	4.23	9125	4.23
Polynomial [8]	9274	4.57	9274	4.57	9274	4.57
ROM based	534	2.21	18198	2.40	36742	2.83
Recursive [13]	9879	2.89	9879	2.89	9879	2.89

(6) Replace the 3 LSB's of $\overline{\sin n\theta}$ and $\overline{\cos n\theta}$ with the corresponding 3 LSB correction bits retrieved from the correction table.

(7) Output the corrected $\overline{\sin n\theta}$ and $\overline{\cos n\theta}$ values to FFT processing element for TF multiplication.

(8) If $n<M-1$, then n=n+1, go to step (4), otherwise proceed to the next step.

(9) If $i<k$, $i=i+1$, go to step (3); otherwise, $k=k+1$.

(10) If $k<K$, go to step (2); otherwise end of the TF computation.

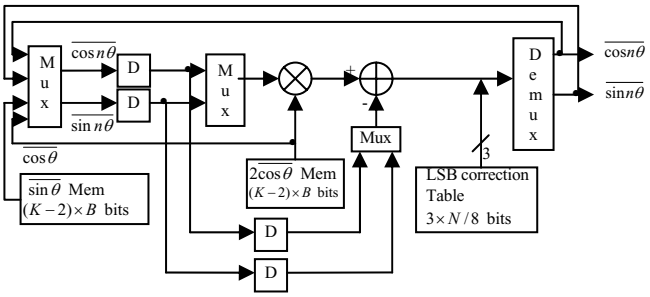

Figure 2. Combined TF generator architecture for interleaving computation of real and imaginary parts.

3. SIMULATIONS AND COMPARISON

In our simulation, we assume B-bit output precision and N-point FFT operations, where N=64, 256, 512, 1204, 2048, 4096, and 8192, as defined in the mentioned OFDM systems. The required hardware complexity of the proposed complete non-interleaving TF generator is:

Area =2[A_m + A_a + 3*N/8 ROM* bits + (*K*-2)*B* ROM bits]

Where A_m and A_a are multiplier and adder areas, respectively. Compared with the conventional ROM-based TF generators, the proposed design obviously costs much less area, especially for the cases of moderate to high output word lengths, as will be shown in the latter comparison. Compared with the 1st-order polynomial-based TF generators, the proposed design required the same numbers of multipliers and adders, but with a much smaller table. Compared with the 2nd-order polynomial-based designs, the new designs of course costs less multipliers and adders, meanwhile requires a smaller table size. In comparison, the modified recursive design [13] needs four multipliers and four adders to generate the TF, where half of them are for compensating the error propagation. As such, it is more area-consuming than the proposed design as will be verified later. Compared with the conventional sequential CORDIC based designs, the new design has a shorter latency and higher speed. Compared with the pipelined CORIDIC based designs [2], [3], [4] the new design has less complicated hardware and simpler control.

To verify the above comparison, we conduct hardware synthesis of the proposed design and some key current techniques. In our synthesis, we set word length B to 8 bits and bits for various FFT lengths commonly used in OFDM systems. The synthesized total gate counts and delays are shown in Table 1, based on TSMC 0.25 μm standard cell library, subject to the maximum delay constraint of 8ns. From Table 1, since the areas and delay times increase very slowly with the increasing FFT lengths, we can conclude that both areas and delay are dominated by multipliers. The areas taken by correction tables are relatively insignificant. The synthesized small table size results are partially contributed by efficient EDA optimization process in synthesizing the table with combinational logic cells. Moreover, since our design costs the least multipliers among the existing designs (except for the table-lookup approach), we will expect a good area performance, compared with other designs as shown in Tables 2 and 3 for 8-bit and 16-bit generators, respectively. The tables summarizes the simulated performance results of our design, the CORDIC-based designs [2], [3], polynomial-based de-

signs [5], [6], [7], the ROM-based design, and the improved recursive TF generator [13], in terms of gate counts and the maximum operation frequency, assumed 8-bit and 16-bit output precisions, with FFT lengths of 64, 4096 and 8192. For other output word lengths, similar results are obtained.

From above simulation results, we can find that the ROM-based design is only suitable for a small FFT lengths less than about 512, while the proposed design has better area performance and comparable speed performance to the existing design in most cases, varying from small lengths to large lengths.

4. CONCLUSION

In this work we propose an efficient twiddle factor generator, which has the merit of low area complexity and high speed. Analysis and simulations show that it is favourable over the existing TF generators for practical FFT operations of lengths varying from 64 to 8192, which covers all possible FFT lengths in current OFDM-based wireless and wireline communication systems, including 802.11a, 802.16a, DVB, DAB, ADSL and VDSL. The proposed design is particularly suitable for the situations where FFT lengths are long and adjustable, as required by the multi-mode and/or multi-standard operations defined in the mentioned systems. Note that the proposed design includes two error correction tables, which guarantee full-precision results. In fact, the complexity of the proposed design can be reduced by using only 2-bit long error correction words, instead of 3 bits, at the cost of sacrificing minor quantization error. Another further possible hardware reduction approach is to do the error compensation only once per few recursive computations, instead of one correction per recursive computation. Of course, these two simplification techniques can be combined for a best hardware reduction results. All these possible improvement are under investigation and will be reported in the final version of this work.

REFERENCES

[1] C.S. Wu and A.Y. Wu, "Modified vector rotational CORDIC (MVR-CORDIC) algorithm and its application to FFT," *Proceedings of IEEE ISCAS,* Geneva, vol: 4, pp. 529-532, May. 2000.

[2] S.Y. Park, N.I. Cho, S.U. Lee, K. Kim, and J. Oh, "Design of 2K/4K/8K/-point FFT processor based on CORDIC algorithm in OFDM receiver," *proceedings of IEEE PACRIM, Pacific Rim, vol: 2, pp. 457-460, Aug. 2001*.

[3] S.W. Mondwurf. Morf, "Versatile COFDM demodulator design based on the CORDIC algorithm," *Tran. on IEEE Consumer Electronics*, vol: 48 pp. 718-723, Aug. 2002.

[4] C.F. Lin and S.G. Chen, "A CORDIC algorithm with fast rotation prediction and small iteration number," *Proceedings of IEEE ISCAS*, vol: 5, pp. 229-232, Jun. 1998.

[5] L. Fanucci, R. Roncella, and R. Saletti, "A sine wave digital synthesizer based on a quadratic approximation," *Proceedings of IEEE Frequency Control Symposium and PDA Exhibition*, pp. 806-810, Jun. 2001.

[6] A.M. Sodagar and G. Roientan, "A novel architecture for ROM-less sine-output direct digital frequency synthesizers by using the 2nd-order parabolic approximation," *Proceedings of IEEE Frequency Control Symposium and Exhibition*, pp. 284-289, Jun. 2002.

[7] A. Bellaouar, M. Obrecht, A. Fahim, and M.I. Elmasry, "A low-power direct digital frequency synthesizer architecture for wireless communications," *Proceedings of IEEE Custom Integrated Circuit*, pp. 593-596, May. 1999.

[8] F. Curticapean and J. Niittylahti, "Low-power direct digital frequency synthesizer," *Proceedings of IEEE Circuit and System*, vol: 2, pp. 8-11, Aug. 2000.

[9] A.M. Eltawil and B. Daneshrad, "Piece-wise parabolic interpolation for direct digital frequency synthesis," *Proceedings of IEEE Custom Integrated Circuits*, pp. 401-404, May. 2002.

[10] L. Xiu and Z. You, "A new frequency shnthesis method based on flying-adder architecture," *Trans. on IEEE Circuits and Systems*, vol: 50, pp. 130-134, Mar. 2003.

[11] A.M. Eltawil and B. Daneshrad, "Interpolation based direct digital frequency synthesis for wireless communications," *Proceedings of IEEE WCNC*, vol: 1, pp. 73-77, Mar. 2002.

[12] N.J. Fliege and J. Wintermantel, "Complex Digital Oscillator and FSK Modulators," *Tran. on IEEE Signal Processing*, vol:40, pp. 333-342, Feb. 1992.

[13] M.M. Al-Ibrahim, "A simple recursive digital sinusoidal oscillator with uniform frequency spacing," *Proceedings of IEEE Circuits and Systems*, pp. 689-692, Mar. 2001.

[14] A.V. Oppenheim, R.V. Schafer and J.R. Buck, Discrete-Time Signal Processing, 2nd Ed., Prentice Hall, 1999.