한국정보전자통신기술학회논문지(jkiiect)'18-06, Vol.11 No.3
논문 18-11-03-270 http://dx.doi.org/10.17661/jkiiect.2018.11.3.270

An area-efficient 256-point FFT design for WiMAX systems

Jian Yu*, Kyung-Ju Cho**

Abstract This paper presents a low area 256-point pipelined FFT architecture, especially for IEEE 802.16a WiMAX systems. Radix-24 algorithm and single-path delay feedback (SDF) architecture are adopted in the design to reduce the complexity of twiddle factor multiplication. A new cascade canonical signed digit (CSD) complex multipliers are proposed for twiddle factor multiplication, which has lower area and less power consumption than conventional complex multipliers composed of 4 multipliers and 2 adders. Also, the proposed cascade CSD multipliers can remove look-up table for storing coefficient of twiddle factors. In hardware implementation with Cyclone 10LP FPGA, it is shown that the proposed FFT design method achieves about 62% reduction in gate count and 64% memory reduction compared with the previous schemes.

Key Words : FFT, pipelined, SDF, CSD, complex multiplier, twiddle factor

1. Introduction

FFT is a very important technique in modern communications, especially for applications in OFDM systems, such as IEEE 802.11a/g/n, WiMAX, wireless personal area networks (WPANs), long-term evolution (LTE) systems [1]. FFT is one of the modules with high computational complexity in the physical layer of OFDM systems. Thus, many FFT design algorithms have been developed to reduce the computational complexities, including radix-2, radix-4, radix-22, radix-23, radix-24, and so on [2]-[5].

The radix-2 algorithm has ever been popular for FFT because it has simple butterfly unit. However, it requires more complex multiplications following with the rising point of FFT. Though the radix-4 algorithm reduce the number of complex multiplications, it needs relative complex butterfly unit. The radix-2k algorithms have been presented in [3]-[4], which can reduce the number of complex multiplication with same butterfly unit of radix-2 algorithm.

In general, FFT architectures can be divided into two different types: memory- based and pipelined architectures. Memory -based architecture consists of main processing element (butterfly unit), memory units and control logics. This kind of architecture owns low hardware cost and power consumption at the cost of low throughput and serious latency. On the other hands, the pipelined FFT architectures can satisfy real-time applications due to high throughput, but cost more hardware resources [6]-[7].

For computing FFT, complex multipliers and look-up talbe are required to multiply the twiddle factors with input signal and to store the required twiddle factors, respectively. These reads to large area and power consuming.

In this paper, to reduce hardware cost, we propose the novel CSD constant complex multipliers instead of conventional complex

This paper was supported by Wonkwang University in 2017.
*Department of Electronic Engineering Wonkwang University
**Corresponding Author: Department of Electronic Engineering Wonkwang University (kjcho@wku.ac.kr)
Received May 11, 2018 Revised May 28, 2018 Accepted June 05, 2018

multiplier which is composed of 4 multipliers and 2 adders for twiddle factor multiplication. By doing this, any look-up table is not required for storing twiddle factors.

2. Design Consideration of FFT

The N-point discrete Fourier transform $X(k)$ of input sequence $x(n)$ is defined as

$$X(k)=\sum_{n=0}^{N-1} x(n)\, W_N^{nk}, \quad 0 \le k \le N-1, \qquad (1)$$

where the twiddle factor $W_N^{nk}=e^{-j2\pi nk/N}$.

Direct implementation of (1) needs large hardware and computation time. Thus, radix-2 FFT algorithm by Cooley-Tukey [2] have been developed to improve the speed of computation and to reduce the requirement of hardware. Later, radix-2k algorithm appears to show the advantage compared to the radix-2 algorithm because it simultaneously achieves a reduced number of complex multiplication and simple butterfly unit like radix-2 algorithm. Thus, we consider radix-2k algorithm in our design.

The 256-point FFT computation with radix-2k algorithm consists of eight stages. The radix-2k algorithm retains the structure of the radix-2 algorithm and has the same butterfly structure regardless of k. However, the twiddle factor multiplication is different for k. Table 1 shows the sequence of 256-point FFT twiddle factor at each stage for radix-2k algorithm. From Table 1, the radix-24 algorithm is optimal candidate for 256-point FFT since it has the least number of complex multiplication and the lowest complexity of twiddle factors, where -j means trivial multiplication.

Table 1. Base number of twiddle factors for 256-point at each stage

Algorithms	Stages							#CM
	1	2	3	4	5	6	7	
Radix-2	W256	W128	W64	W32	W16	W8	-j	642
Radix-22	-j	W256	-j	W64	-j	W16	-j	492
Radix-23	-j	W8	W256	-j	W8	W32	-j	504
Radix-24	-j	W16	-j	W256	-j	W16	-j	480

#CM: total number of complex multiplication

Generally, two design styles exist in the pipelined FFT architectures: feedforward and feedback. Feedforward architectures can be divided into single-path delay commutator (SDC) and multi-path delay commutator (MDC). Feedback architectures can be divided into SDF and multi-path delay feedback (MDF) [7]. The control logic of feedforward styles is more complex than that of feedback styles, and MDF architecture requires more hardware cost than SDF architecture.

The demand for wireless devices is increasing rapidly that requires less hardware and low power FFT architecture [8]. Thus, in this paper SDF FFT architecture was adopted since it requires less memory elements and its control unit is easy to design.

Fig. 1 shows an architecture of the radix-24 256-point SDF FFT. Two types of butterfly (BF1 and BF2) and several delay buffers of various sizes are used for data shuffling to obtain appropriate data at the butterfly input. The symbol ⊗ represents the complex multiplier. Control signals are used to switch the butterfly types and it also provides a appropriate control for twiddle factor multiplication.

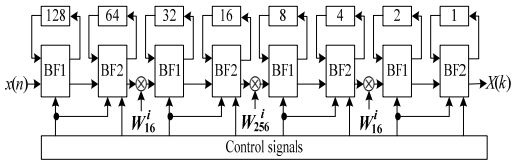

Fig. 1. 256-point radix-2^4 SDF FFT architecture.

3. Proposed FFT Design

In this section, using CSD multiplier, we present the method to reduce the required hardware of complex multiplication with twiddle factors.

3.1 Proposed W_{16}^{i} CSD complex multiplier

Twiddle factors W_{16}^{i} at stage 2 only need seven factors: W_{16}^{0}, W_{16}^{1}, W_{16}^{2}, W_{16}^{3}, W_{16}^{4}, W_{16}^{6}, and W_{16}^{9}. By using $W_{16}^{4}=-j$ and symmetry property of complex sinusoidal function, the factors can be expressed by three values of Re$\{W_{16}^{1}\}$, Re$\{W_{16}^{2}\}$, and Re$\{W_{16}^{3}\}$ as shown in Table 2, where Re{t} represents the real part of t. The three values Re$\{W_{16}^{1}\}$, Re$\{W_{16}^{2}\}$, and Re$\{W_{16}^{3}\}$ are equal to 0.9239, 0.7071 and 0.3827, respectively.

In hardware design, CSD representation and common sub-expression (CSE) sharing methods are exploited to reduce the occupied hardware resources. Table 3 shows the CSD representations of Re$\{W_{16}^{1}\}$, Re$\{W_{16}^{2}\}$, and Re$\{W_{16}^{3}\}$. The CSE block '101' (or -10-1) is enclosed by the blue ellipses. Therefore, the CSD constant complex multiplier is only composed of adders, shifters and multiplexers with low cost of hardware compared to conventional complex multiplier.

Table 2. Representation of W_{16}^{i}

W_{16}^{0}	1	W_{16}^{4}	−j
W_{16}^{1}	Re$\{W_{16}^{1}\}$−jRe$\{W_{16}^{3}\}$	W_{16}^{6}	−Re$\{W_{16}^{2}\}$−jRe$\{W_{16}^{2}\}$
W_{16}^{2}	Re$\{W_{16}^{2}\}$−jRe$\{W_{16}^{2}\}$	W_{16}^{9}	−Re$\{W_{16}^{1}\}$−jRe$\{W_{16}^{3}\}$
W_{16}^{3}	Re$\{W_{16}^{3}\}$−jRe$\{W_{16}^{1}\}$		

Table 3. CSD representation for W_{16}^{i} with 12 bits

Re$\{W_{16}^{1}\}$	1	0	0	0	−1	0	−1	0	0	1	0	0
Re$\{W_{16}^{2}\}$	1	0	−1	0	−1	0	1	0	1	0	0	0
Re$\{W_{16}^{3}\}$	0	1	0	1	0	0	0	1	0	0	0	0

The three CSD multipliers can be obtained by simply using 6 shifters and 7 additions as

$$\begin{aligned} CSE &= d + d \gg 2 \\ d \times Re\{W_{16}^{4}\} &= d - CSE \gg 4 + d \gg 9 \\ d \times Re\{W_{16}^{2}\} &= d - CSE \gg 2 + d \gg 6 \\ d \times Re\{W_{16}^{3}\} &= d \gg 1 - d \gg 3 + d \gg 7 \end{aligned} \quad (2)$$

where, d and ≫t represent the multiplicand for twiddle factors and the right-shift operation by t.

In order to select the two sets of twiddle factor multiplications in Table 2, 4-to-1 multiplexers are needed. Fig. 2 shows the detailed structure of the proposed constant complex multiplier W_{16}^{i}. Two signals sel1 and sel2 are needed to select the proper results.

3.2 Proposed W_{256}^{i} CSD complex multiplier

As shown in Fig. 1, the butterfly output signals at stage 4 are multiplied by appropriate twiddle factors $W_{256}^{i}(i=0\sim255)$. The twiddle factors are $W_{256}^{i}=x+jy$, where i are divided into 8 regions. Only N/8 sets of constant values, i.e., $W_{256}^{p}=x_p+jy_p$, where p is from 0 to N/8 as region A in Table 4, are needed since the twiddle factors in other regions can be obtained through mapping from region A [7]. Table 4 shows the corresponding mapping.

Table 4. Twiddle factors with corresponding mapping in 8 regions

i	Real	Imaginary	Region
$0 \le i \le N/8$	x_p	y_p	A
$N/8 < i < N/4$	$-y_p$	x_p	B
$N/4 < i \le 3N/8$	y_p	$-x_p$	C
$3N/8 < i < N/2$	$-x_p$	y_p	D
$N/2 < i \le 5N/8$	$-x_p$	$-y_p$	E
$5N/8 < i < 3N/4$	y_p	x_p	F
$3N/4 < i \le 7N/8$	$-y_p$	x_p	G
$7N/8 < i < N$	x_p	$-y_p$	H

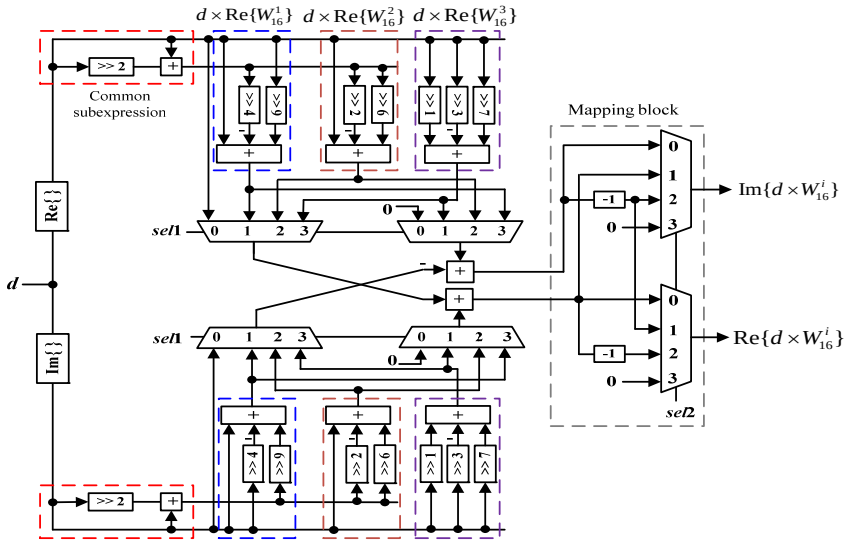

Fig. 2. Detailed structure of the CSD constant complex multiplier for W_{16}^i.

To reduce the hardware cost for multiplication of W_{256}^i, the cascade CSD constant complex multiplier structure is proposed as follows.

1. By using the 1/8 symmetry property, the exponent i $(i=0\sim255)$ of W_{256}^i can be reduced as p $(0\le p\le 32)$.

2. In order to further reduce the number of twiddle factors, decompose p into p_1 and p_2 as $p=4p_1+p_2$ $(p_1=0\sim8, p_2=0\sim3)$.

3. Tabulate CSD representations for $W_{256}^{p_1}$ and $W_{256}^{p_2}$ and find the optimized CSE.

The proposed cascade CSD complex multiplier needs two stage complex multiplication operation to achieve a complete multiplication of twiddle factor.

$$\begin{aligned} d\times W_{256}^{p} &= d\times W_{256}^{4p_1+p_2} = d\times W_{256}^{4p_1}\times W_{256}^{p_2} \\ &= d\times\left(Re\left\{W_{256}^{4p_1}\right\}+jIm\left\{W_{256}^{4p_1}\right\}\right) \\ &\quad\times\left(Re\left\{W_{256}^{p_2}\right\}+jIm\left\{W_{256}^{p_2}\right\}\right) \end{aligned} \tag{3}$$

For example, $d\times W_{256}^{13}$ can be decomposed into $d\times W_{256}^{4\times3}\times W_{256}^{1}$ $(p_1=3,\ p_2=1)$.

Note that 32 twiddle factors are further reduced to 12 different values. Table 5 lists the CSD representation of the 12 values. Multiplier-less realization of twiddle factor multiplication W_{256}^i can be accomplished by optimizing the CSE eliminations of the 12 values. From Table 5, one of the term '101' (or -10-1) as a CSE block is enclosed by blue ellipses, and '10-1' (or -101) and '1000-1' (or -10001) are also considered as the CSE block enclosed by red and purple ellipses, respectively.

Table 5. CSD representation of 12 values for composing twiddle factors with 12-bit

i_1	$Re\{W_{256}^{4i_1}\}$											
1	1	0	0	0	0	0	0	0	−1	0	−1	0
2	1	0	0	0	0	0	−1	0	−1	0	0	0
3	1	0	0	0	−1	0	1	0	1	0	0	−1
4	1	0	0	0	−1	0	−1	0	0	1	0	0
5	1	0	0	−1	0	0	0	1	0	0	−1	0
6	1	0	−1	0	1	0	1	0	1	0	−1	0
7	1	0	−1	0	0	1	0	−1	0	0	0	−1
8	1	0	−1	0	−1	0	1	0	1	0	0	0
i_2	$Re\{W_{256}^{2i_2}\}$											
0	1	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	−1
2	1	0	0	0	0	0	0	0	0	−1	0	1
3	1	0	0	0	0	0	0	0	−1	0	1	0
i_1	$-Im\{W_{256}^{4i_1}\}$											
1	0	0	0	1	0	−1	0	0	1	0	0	0
2	0	0	1	0	−1	0	0	1	0	0	0	−1
3	0	0	1	0	0	1	0	1	0	0	1	0
4	0	1	0	−1	0	0	0	1	0	0	0	−1
5	0	1	0	0	0	−1	0	0	0	1	0	1
6	0	1	0	0	1	0	0	−1	0	0	0	1
7	0	1	0	1	0	0	0	1	0	1	0	−1
8	1	0	−1	0	−1	0	1	0	1	0	0	0
i_2	$-Im\{W_{256}^{2i_2}\}$											
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	1	0	−1	0	0	1	0
2	0	0	0	0	1	0	−1	0	0	1	0	0
3	0	0	0	0	−1	0	1	0	−1	0	−1	0

The detailed architecture of W_{256}^{i} CSD constant multiplier is shown in Fig. 3. The CSE block in Fig. 3 consists of adders, and shifters and the rectangle boxes represent right shifters, and these shifters are realized using simple hardware connection. Three 4-to-1 and six 8-to-1 multiplexers are used to obtain the appropriate results.

4. Results and Comparison

To evaluate the proposed CSD constant complex multipliers, the proposed and conventional complex multipliers with 12-bit wordlength were designed using Verilog HDL, and synthesized using Cyclone 10LP and QUARTUS PRIME design tool. Table 6 shows results of the design. Note that the proposed complex multipliers for W_{16}^{i} and W_{256}^{i} can reduce about 76% and 34% in gate counts, compared with the conventional complex multiplier using modified Booth multiplier, respectively.

In addition, the proposed FFT and previous FFT for 256-point FFT for IEEE 802.16a WiMAX systems were designed using Verilog HDL and synthesized using Cyclone 10LP.

Table 7 shows the performance comparison between the proposed scheme and the other schemes. Note that the proposed design achieves 62% gate count reduction and 64% memory reduction compared to conventional design.

The throughput rate of the implementation is suitable for WiMAX applications, whose the maximum sample rate is 32MHz.

Table 6. Hardware comparison for W_{16}^{i} and W_{256}^{i}

Methods	Logics
Conventional complex mul.	2,090 (1)
W_{16}^{i} using CSD mul.	501 (0.24)
W_{256}^{i} using cascade CSD mul.	1,369 (0.66)

Table 7. Hardware comparison of 256-point FFT

Methods	Logic elements	Registers	Memory bits
Radix-2^4	9,450 (1)	509 (1)	19,044 (1)
Radix-2^4 [1]	8,763 (0.93)	509 (1)	13,640 (0.72)
Radix-2^4 [8]	4,981 (0.53)	489 (0.96)	12,900 (0.68)
Proposed	3,578 (0.38)	489 (0.96)	6,756 (0.36)

5. Conclusion

In this paper, we proposed a hardware efficient FFT design method for WiMAX systems using

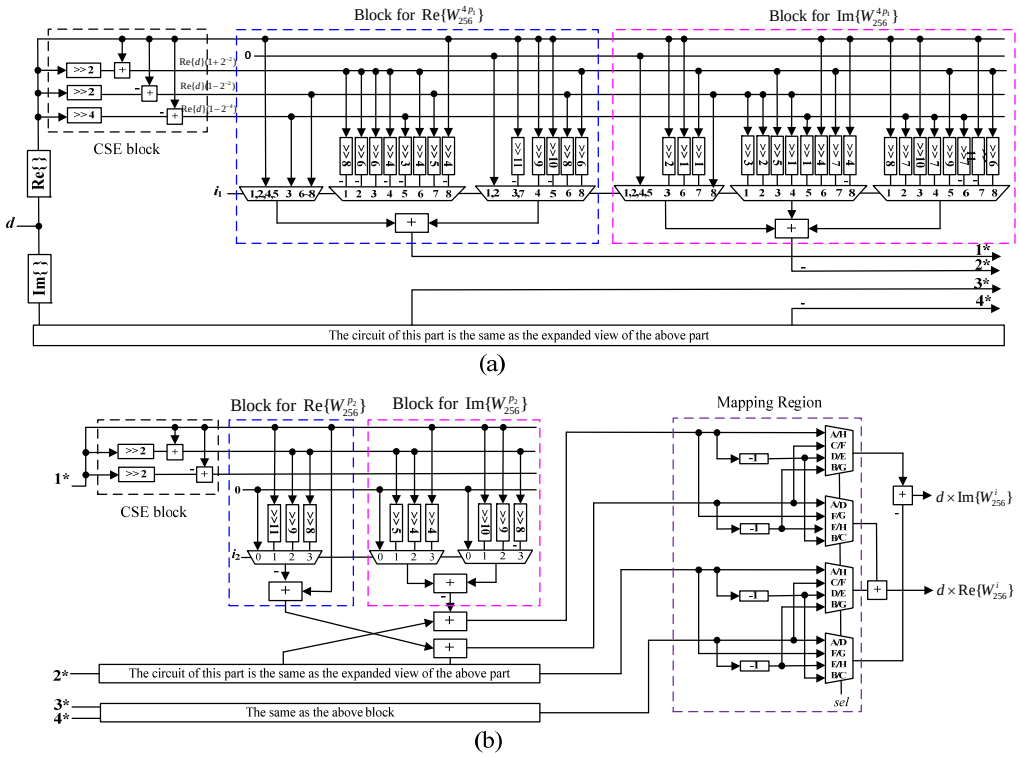

Fig. 3. Detailed structure of the cascade CSD constant complex multiplier for W_{256}^{i}.

radix-24 algorithm and SDF architecture. To reduce the hardware cost, we proposed the CSD constant complex multipliers which replace conventional complex multiplier and remove look-up table for storing twiddle factors. By simulation, it was shown that the proposed FFT design method achieves about 62% reduction in gate count and 64% reduction in memory size compared with the previous schemes.

REFERENCES

[1] J. H. Kim and I. C. Park, "Long-point FFT processing based on twiddle factor table reduction", IEICE Trans. Fundam. Electron. Commun. Comput. Sci., E90-A. no. 11, pp. 2526-2532, 2007.

[2] J. W. Cooley, J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series", Math. Comput., 19(90), pp.297-301, 1965.

[3] S. He and M. Torkelson, "Designing pipeline FFT processor for OFDM (de) modulation", Proc. URSI Int. Symp. Signals. Syst., Electron., 1998. pp. 257-262.

[4] J. Y. Oh and M. S. Lim, "New radix-2 to the 4th power pipeline FFT processor", IEICE trans. Electron., vol. E88-C, no. 8, pp.1740-1746, 2005.

[5] C. Wang, Y. Yan, and X. Fu, "A high-throughput low-complexity radix-22-22 -23 FFT/IFFT Processor with parallel and normal input/output order for IEEE 802.11ad systems", IEEE Trans. Very Large Scale Integr. (VLSI) Syst, vol. 23, no. 11, pp.2728-2732, 2015.

[6] S. J. Huang, S. G. Chen, "A high-throughput radix-16 FFT processor with parallel and normal input/output ordering for IEEE 802.15. 3c systems", IEEE Tran. Circuits Sys. I: Reg. Papers, vol. 59, no. 8, pp.1752-1765, 2012.

[7] G. K. Ganjikunta and S. K. Sahoo, "An area-efficient and low-power 64-point pipeline Fast Fourier Transform for OFDM applications", Integration, the VLSI Journal, vol. 57, pp.125-131, 2017.

[8] C. P. Fan, M. S., Lee and G. A. Su, "A low multiplier and multiplication costs 256 -point FFT implementation with simplified radix-24 SDF architecture", Proc IEEE APCCAS 2006. pp. 1935-1938.

Author Biography

Jian Yu [Member]

- Jun. 2001: Hebei Normal Univ., Electronic Engr., BA
- Mar. 2008: Tianjin Polytechnic Univ., Electronic Engr., MS
- Mar. 2016 ~ current : Wonkwang Univ., Dept. of Electronic Engr., PhD course

〈Research Interests〉 Low-power system, VLSI Design

Kyung-Ju Cho [Member]

- Feb. 2002 : Chonbuk National Univ., Info. & Comm. Engr., MS
- Aug. 2006 : Chonbuk National Univ., Info. & Comm. Engr., PhD
- Mar. 2012 ~ current : Wonkwang Univ., Dept. of Electronic Engr., Professor

〈Research Interests〉 Low-power system, VLSI Design, SOC