Area-Efficient Scheduling Scheme Based FFT Processor for Various OFDM Systems

Jeong Keun Jang
Dongbu Hitek
Bucheon, Korea
jeongkeun.jang@dbhitek.com

Ho Keun Kim, Myung Hoon Sunwoo
Department of Electrical and Computer Engineering
Ajou University
Suwon, Korea
hokeun92@ajou.ac.kr, sunwoo@ajou.ac.kr

Oscar Gustafsson
Department of Electrical Engineering
Linköping University
Linköping, Sweden
oscar.gustafsson@liu.se

Abstract— **This paper presents an area-efficient fast Fourier transform (FFT) processor for orthogonal frequency-division multiplexing systems based on multi-path delay commutator architecture. This paper proposes a data scheduling scheme to reduce the number of complex constant multipliers. The proposed mixed-radix multi-path delay commutator FFT processor can support 128-, 256-, and 512-point FFT sizes. The proposed processor was synthesized using the Samsung 65-nm CMOS standard cell library. The proposed processor with eight parallel data paths can achieve a high throughput rate of up to 2.64 GSample/s at 330 MHz.**

Keywords-fast Fourier transform (FFT); high throughput; low hardware complexity; mixed-radix multi-path delay commutator (MRMDC); orthogonal frequency-division multiplexing (OFDM) systems

I. INTRODUCTION

Fast Fourier transform (FFT) is a well-known mathematical algorithm for performing Fourier transform operations. The FFT plays an important role in different fields such as communication systems, biomedical applications, sensor, and radar signal processing. Moreover, an FFT processor is a high computational complexity module in the physical layer of orthogonal frequency-division multiplexing (OFDM) applications such as IEEE 802.11n/ac/ad [1], IEEE 802.15.3.c [2], and IEEE 802.16e [3]. Hence, various FFT processors have been proposed [2] to satisfy real-time processing requirements and reduce hardware complexity [3]-[11].

Most of the FFT architectures can be divided into two categories: 1) memory-based architectures and 2) pipelined architectures. Memory-based architectures were proposed to achieve smaller area [3]; whereas, pipelined FFT architectures [4]-[11] can achieve high throughput rates and low latency, which are suitable for real-time applications. Pipelined FFT architectures can be classified into single-path feedback (SDF) architectures [5], multi-path delay feedback (MDF) architectures [9]-[11], and multi-path delay commutator (MDC) architectures [6]-[8], according to the dataflow scheme.

In current real-time applications, many parallel pipelined FFT architectures have been proposed [6]-[11] to provide very high throughput rates. The number of delay elements in MDF architectures [9]-[11] is less than that in SDF architectures [5]. Recently, parallel MDC architectures have been proposed in [6]-[8] for achieving high throughput rates and hardware efficiency based on radix-2^n algorithms as an improvement on radix-2 and radix-4 algorithms. In [8], radix-8 pipelined MDC architectures improved the area efficiency by using data shuffling structures. However, the radix-8 algorithm cannot handle 128- and 256-point FFTs. Conversely, the proposed FFT processor can provide both 128- and 256-point FFTs. Moreover, the proposed processor was designed based on the radix-4 and radix-2 algorithms, which can significantly reduce the area.

In this paper, we propose an eight-parallel mixed-radix MDC architecture for low hardware complexity. An area-efficient scheduling scheme is proposed to reduce the size of read-only memories (ROMs) for storing twiddle factors.

This paper is organized as follows. Section II describes FFT algorithms for the proposed architecture. Section III provides the proposed mixed-radix MDC FFT architecture in detail. Section IV presents the design and implementation results of the proposed FFT processor. Finally, the conclusion is presented in Section V.

II. FFT ALGORITHMS

The discrete Fourier transform (DFT) of length N is defined as

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

where $x(n)$ and $X(k)$ denote the input and output of the DFT, respectively, and W_N^{nk} denotes the Nth primitive root of unity, with its exponent evaluated as modulo N [12].

$$W_N^{nk} = e^{-j(2\pi nk/N)} = \cos(2\pi nk/N) - j\sin(2\pi nk/N). \tag{2}$$

Furthermore, (1) can be reformulated as (3) using the 2-dimensional index map in (4). Moreover, (3) consists of two DFT computation 64-point DFTs, which are expressed as $G(n_2, k_1)$ and N/64-point DFT.

$$X(k_1+64k_2)=\sum_{n_2=0}^{\frac{N}{64}-1}\sum_{n_1=0}^{63}x(\frac{N}{64}n_1+n_2)W_N^{(\frac{N}{64}n_1+n_2)(k_1+64k_2)}$$
$$=\sum_{n_2=0}^{\frac{N}{64}-1}\left\{\underbrace{\left[\sum_{n_1=0}^{63}x(\frac{N}{64}n_1+n_2)W_{64}^{n_1k_1}\right]}_{G(n_2,k_1)}W_N^{n_2k_1}\right\}W_{N/64}^{n_2k_2} \tag{3}$$

where

$$\begin{cases} n_1=0,1,\cdots,63; \quad n_2=0,1,\cdots,(N/64-1) \\ k_1=0,1,\cdots,63; \quad k_2=0,1,\cdots,(N/64-1) \\ N=128,256,512. \end{cases} \tag{4}$$

Thus, when N is 128, 256, and 512, the N/64-point DFT is 4-, 4-, 2-, and 2-point DFTs, respectively. As these 2- and 4-point DFTs can be folded using radix-2, they can be calculated using radix-2, radix-2^2, and radix-2^3, respectively, as expressed in (5).

$$X(k_1+64k_2)=\sum_{n_2=0}^{7}\left\{G(n_2,k_1)W_N^{n_2k_1}\right\}W_8^{n_2k_2}$$
$$=\underbrace{\sum_{\alpha'_3=0}^{1}\underbrace{\sum_{\alpha'_2=0}^{1}\underbrace{\sum_{\alpha'_1=0}^{1}\left\{G(n_2,k_1)\overbrace{W_N^{n_2k_1}}^{Stage4\ TF}\right\}\overbrace{W_2^{\alpha'_1\beta'_1}}^{Stage5\ BU}}_{N=128}\underbrace{W_4^{\alpha'_2\beta'_1}}_{Stage5\ TF}\overbrace{W_2^{\alpha'_2\beta'_2}}^{Stage6\ BU}}_{N=256}\underbrace{W_8^{(\alpha'_3\beta'_1+2\alpha'_3\beta'_2)}}_{Stage6\ TF}\overbrace{W_2^{\alpha'_3\beta'_3}}^{Stage7\ BU}}_{N=512} \tag{5}$$

$$n_1=16\alpha_1+4\alpha_2+2\alpha_3+\alpha_4, \quad k_1=\beta_1+4\beta_2+16\beta_3+32\beta_4$$
$$G(n_2,k_1)=\sum_{n_1=0}^{63}x\left(\frac{N}{64}n_1+n_2\right)W_{64}^{n_1k_1}$$
$$=\sum_{\alpha_4=0}^{1}\sum_{\alpha_3=0}^{1}\sum_{\alpha_2=0}^{3}\sum_{\alpha_1=0}^{3}x\left(\frac{N}{64}n_1+n_2\right)W_{64}^{(16\alpha_1+4\alpha_2+2\alpha_3+\alpha_4)(\beta_1+4\beta_2+16\beta_3+32\beta_4)}$$
$$=\sum_{\alpha_4=0}^{1}\sum_{\alpha_3=0}^{1}\sum_{\alpha_2=0}^{3}\sum_{\alpha_1=0}^{3}x\left(\frac{N}{64}n_1+n_2\right)$$
$$\times\underbrace{W_4^{\alpha_1\beta_1}}_{Stage1\ BU}\overbrace{W_{16}^{\alpha_2\beta_1}}^{Stage1\ TF}\underbrace{W_4^{\alpha_2\beta_2}}_{Stage2\ BU}\overbrace{W_{64}^{(2\alpha_2+\alpha_3)(\beta_1+4\beta_2)}}^{Stage2\ TF}\underbrace{W_2^{\alpha_3\beta_3}}_{Stage3\ BU}\overbrace{W_4^{\alpha_4\beta_3}}^{Stage3\ TF}\underbrace{W_2^{\alpha_4\beta_4}}_{Stage4\ BU}. \tag{6}$$

where

$$\begin{cases} \alpha_1=0,1,2,3;\ \alpha_2=0,1,2,3;\ \alpha_3=0,1;\ \alpha_4=0,1 \\ \beta_1=0,1,2,3;\ \beta_2=0,1,2,3;\ \beta_3=0,1;\ \beta_4=0,1. \end{cases} \tag{7}$$

Therefore, this paper proposes decomposition for calculating the 128-, 256-, and 512-point DFTs using (5) and (6). In these decompositions, the required twiddle factors for each stage are summarized in Table I; the mixed method in Table I indicates that the twiddle factors should be calculated according to the FFT size.

TABLE I. 128-, 256-, AND 512-POINT FFT TWIDDLE FACTOR COMPUTATION.

Stage / FFT size	1	2	3	4	5	6
128-point	W_{16}	W_{64}	$-j$	W_{128}		
256-point	W_{16}	W_{64}	$-j$	W_{256}	$-j$	
512-point	W_{16}	W_{64}	$-j$	W_{512}	$-j$	W_8
Mixed method	W_{16}	W_{64}	$-j$	W_{512}	$-j$	W_8

III. PROPOSED FFT ARCHITECTURE

Using the radix-4^2 and radix-2^2 FFT algorithms in Module-1 and the radix-2^n FFT algorithm in Module-2, we proposed the mixed-radix MDC FFT architecture illustrated in Fig. 1. To perform 128-, 256-, and 512-point FFT operations, the proposed FFT processor consists of seven stages. Stages 1, 2, 3, and 4 are used in common, but stages 5, 6, and 7 are selectively reconfigured according to two selection bits as presented as shown in Table II. The proposed FFT architecture employs MDC architectures including butterfly units (BU), complex multipliers, complex

TABLE II. MULTIPLEXER SELECTION BITS

FFT size	S_1	S_0
128	0	0
256	0	1
512	1	1

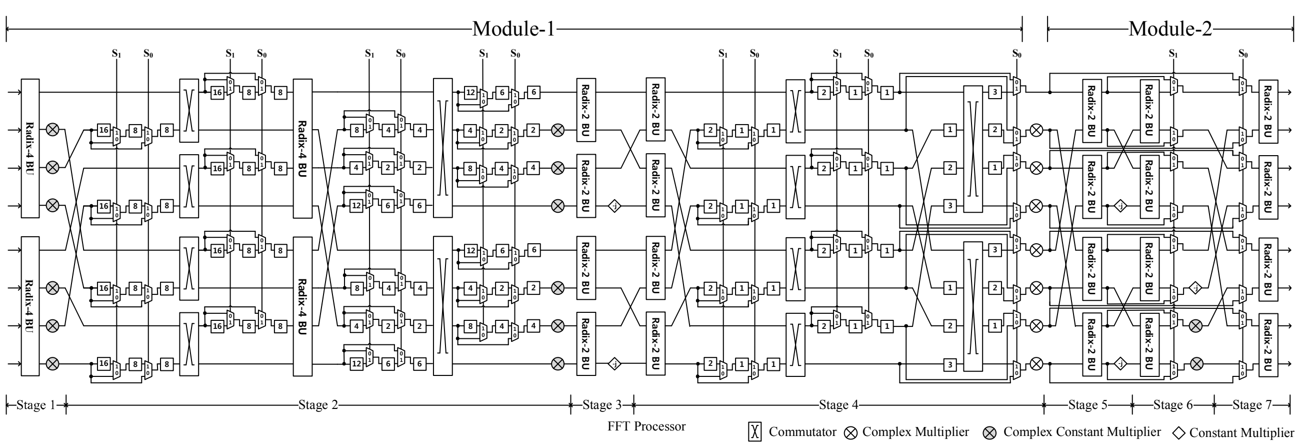


Figure 1. Proposed mixed-radix MDC FFT architecture

constant multipliers, delay elements, and commutators.

A. Proposed data scheduling scheme in stage 2

The proposed FFT processor requires the twiddle factor $W_{64}^{(2\alpha_2+\alpha_4)(\beta_1+4\beta_2)}$ in stage 2. By using the proposed commutator, we modified the conventional structure by changing the connection. Therefore, the proposed commutator blocks between stage 2 and stage 3 reduce the number of multipliers by rearranging the output data samples of radix-4 BU.

By using the new data scheduling scheme, the proposed architecture can remove complex multipliers in paths 1 and 5 as shown in Fig. 2. Therefore, the new data scheduling scheme can reduce the number of complex constant multipliers from eight to six.

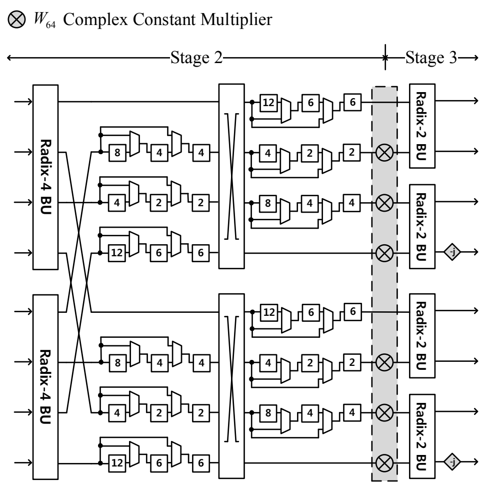


Figure 2. Stages 2 and 3 of the proposed FFT architecture.

B. Proposed data scheduling scheme in stage 4

The twiddle factor is $W_{512}^{n_2k_1}(=W_{512}^{n_2(\beta_1+4\beta_2+16\beta_3+32\beta_4)})$ in stage 4. Changing the location of the data samples in stage 4 affects the twiddle factor multiplications. As shown in Fig. 3, using the proposed scheduling scheme, three of the eight W_{512} could be replaced with two W_{256} and one W_{128}, and one of the eight W_{512} is not required.

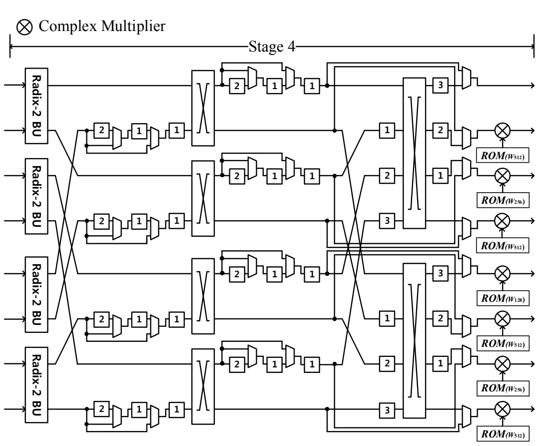
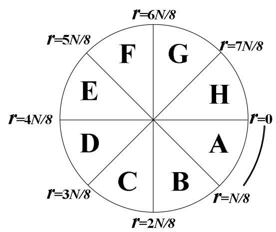


Figure 3. Decomposition of three different FFT lengths.

The twiddle factor multiplication is one of the major contributors to the area of the FFT processor, which requires both memories and complex multipliers [15]. The existing processor [10] requires ROMs with 1024 stored words. However, the proposed FFT processor requires ROMs with 672 stored words, by using the data mapping scheme as shown in Fig. 4. Therefore, the size of the twiddle factor LUTs in stage 4 can be reduced to 34.4% compared with the existing structure [10].



Mapping Region	Twiddle factor		Multiplication (Input = a + *j*b)	
	Real	Imag.	Real	Imag.
A	*p*	*-q*	*ap+bq*	*bp-aq*
B	*q*	*-p*	*aq+bp*	*bq-ap*
C	*-q*	*-p*	*-aq+bp*	*-bq-ap*
D	*-p*	*-q*	*-ap+bq*	*-bp-aq*
E	*-p*	*q*	*-ap-bq*	*-bp+aq*
F	*-q*	*p*	*-aq-bp*	*-bq+ap*
G	*q*	*p*	*aq-bp*	*bq+ap*
H	*p*	*q*	*ap-bq*	*bp+aq*

Figure 4. Eight regions of the mapping scheme.

IV. RESULTS AND COMPARISONS

Based on the fixed-point simulation results, 12-bit word length of the proposed FFT processor is synthesized using a Samsung 65-nm CMOS standard cell library. The proposed processor can operate up to 330 MHz. For comparison with different technologies, the normalized area based on [8] is expressed in the following equation:

$$\text{Normalized Area} = \frac{\text{Area}}{(\text{Tech.} / 65\text{ nm})^2} \tag{8}$$

As summarized in Table III, the proposed FFT proces-

TABLE III. PERFORMANCE COMPARISONS

	Proposed	[8]	[10]	[11]
Technology	65 nm	65 nm	90 nm	130 nm
Architecture	MDC	MDC	MDF	MDF
Size	512/256/128	512	512	512
Datapath type	8	8	8	8
Algorithm	Mixed radix-2/4	Radix-8	Modified radix-2^5	Mixed radix-$2^2/2^3/2^4$
Word length (bits)	12	12	12	14
SQNR (dB)	33	N/A	35	N/A
Frequency (MHz)	330	330	310	220
Throughput (GSample/s)	2.64	2.64	2.48	1.76
Area (mm^2)	0.21	0.88	0.78	1.69
Normalized area (mm^2)	0.21	0.88	0.41	0.42

sor operates at 330 MHz and its throughput is 2.64 GSample/s. The throughput is the same as that in [8] and faster than those in [10] and [11]. The normalized areas in [8], [10], [11], and the proposed FFT processor are 0.88 mm^2, 0.41 mm^2, 0.42 mm^2, and 0.21 mm^2, respectively. In summary, the proposed FFT processor can additionally support 128-/256-point operations compared with [8], [10], and [11]. Furthermore, the clock rate and throughput are faster than in [10] and [11] as the proposed FFT architecture is MDC. Therefore, the proposed FFT processor achieves the best area efficiency and throughput compared with the other FFT processors in [8], [10], and [11] and can be applied to an OFDM system such as IEEE 802.11n/ac/ad, because the proposed FFT processor can support various FFT points compared with [8], [10], and [11].

V. Conclusion

This paper proposed an area-efficient mixed-radix MDC FFT processor for various OFDM systems such as 802.11n/ac/ad. The proposed FFT processor can be reconfigured for 128-, 256-, and 512-point FFTs. The proposed processor adopts a scheduling scheme to reduce the number of complex multipliers and complex constant multipliers. The performance results show that the proposed FFT processor can achieve 2.64 GSample/s at 330 MHz. Moreover, the proposed FFT processor can support various FFT points compared with [8], [10], and [11]. Thus, it can be applied to various OFDM systems such as 802.11n/ac/ad.

Acknowledgment

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2018-2016-0-00309-002) supervised by the IITP(Institute for Information & communications Technology Promotion), by the National Research Foundation of Korea under the framework of international cooperation program (NRF-2016K2A9A2A12003787) and by IDEC (IC Design Education Center).

References

[1] IEEE P802.11-Task Group AD, http://www.ieee802.org/11/

[2] M. Garrido, F. Qureshi, J. Takala, and O. Gustafsson, Hardware architectures for the fast Fourier transform, 3rd ed. Handbook of Signal Processing Systems, Springer, 2018.

[3] S. J. Huang and S. G. Chen, “A high-throughput radix-16 FFT processor with parallel and normal input/output ordering for IEEE 802.15.3c systems,” IEEE Trans. on Circuits and Syst. I, vol. 59, no. 8, pp. 1752–1765, Aug. 2012.

[4] Fang-Li Yuan, Yi-Hsien Lin, Chih-Feng Wu, Muh-Tian Shiue and Chorng-Kuang Wang, “A 256-Point dataflow scheduling 2×2 MIMO FFT FFT/IFFT processor for ieee 802.16 WMAN,” in Proc. IEEE Asian Solid-State Circuits Conference (A-SSCC), Nov. 2008, pp.309-312., doi:10.1109/ASSCC.2008.4708789.

[5] C. T. Lin, Y. C. Yu and L. D. Van, “Cost-effective triple-mode reconfigurable pipeline FFT/IFFT/2-D DCT processor,” IEEE Trans. on Very Large Scale Integr. (VLSI) Syst., vol. 16, no.8, pp. 1058-1071, Aug. 2008.

[6] M. Garrido, J. Grajal, M. S´anchez, and O. Gustafsson, “Pipelined radix-2^k feedforward FFT architectures,” IEEE Trans. on Very Large Scale Integr. (VLSI) Syst., vol. 21, no. 1, pp. 23-32, Jan. 2013.

[7] M. Ayinala and K.K. Parhi, “Parallel Pipelined FFT Architectures with Reduced Number of Delays,” in Proc. ACM Great Lakes Symp. on VLSI (GLSVLSI), May 2012, pp. 63-66, doi: 10.1145/2206781.2206798.

[8] T. Ahmed, M. Garrido, and O. Gustafsson, “A 512-point 8-parallel pipelined feedforward FFT for WPAN,” in Proc. 2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR), Nov. 2011, pp. 981–984, doi: 10.1109/ACSSC.2011.6190157.

[9] Y. Chen, Y.-W. Lin, Y.-C. Taso, and C.-Y. Lee, “A 2.4-Gsample/s DVFS FFT processor for MIMO OFDM communication systems,” *IEEE J. of Solid-State Circuits*, vol. 43, no. 5, pp. 1260–1273, May 2008.

[10] T. Cho and H. Lee, “A High-Speed Low-Complexity Modified Radix-2^5 FFT Processor for High Rate WPAN Applications,” IEEE Trans. on Very Large Scale Integr. (VLSI) Syst., vol. 21, pp. 187-191, Jan. 2013.

[11] C. Wang, Y. Yan, and X. Fu, “A High-Throughput Low-complexity Radix-2^4-2^2-2^3 FFT/IFFT Processor with Parallel and Normal Input/ Output Order for IEEE 802.11ad Systems,” IEEE Trans. on Very Large Scale Integr. (VLSI) Syst., vol. 23, no. 11, pp. 2728-2732, Nov. 2015.

[12] A. V. Oppenheim and R. W. Schafe, Discrete-time signal processing. Englewood Cliffs: Prentice Hall. 1989.

[13] F. Qureshi and O. Gustafsson, “Low-complexity reconfigurable complex constant multiplication for FFTs,” in Proc. IEEE Int. Symp. on Circuits and Systems (ISCAS), 2009, pp. 1137-1140, doi: 10.1109/ISCAS.2009.5117961.

[14] M. Garrido, F. Qureshi, O. Gustafsson, “Low-Complexity Multiplierless Constant Rotators Based on Combined Coefficient Selection and Shift-and-Add Implementation (CCSSI),” IEEE Trans. on Circuits and Syst. I, vol. 61, no. 7, pp. 2002-2012, Jul. 2014.

[15] F. Qureshi, S.A. Alam and O. Gustafsson, “4K-Point FFT Algorithms based on optimized twiddle factor multiplication for FPGAs,” in proc. 2010 Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics (PrimeAsia), Sep. 2010, pp. 225-228, doi: 10.1109/PRIMEASIA.2010.5604921.