논문 2009-46SD-2-3

MIMO-OFDM 시스템을 위한 고속 저면적 128/64-point Radix-2^4 FFT 프로세서 설계

(A High-Speed Low-Complexity 128/64-point Radix-2^4 FFT Processor for MIMO-OFDM Systems)

리우 항*, 이 한 호**

(Liu Hang and Hanho Lee)

요 약

본 논문은 높은 데이터 처리율을 요하는 MIMO-OFDM 시스템을 위하여 고속의 낮은 하드웨어 복잡도를 가진 128/64-point radix-2^4 FFT/IFFT 프,로세서 설계에 대해 제안한다. 높은 Radix 다중경로 지연 피드백 (MDF) FFT구조는 고속의 데이터 처리율과 낮은 하드웨어 복잡도를 제공한다. 제안하는 프로세서는 128-point와 64 point FFT/IFFT의 동작을 지원할 뿐만 아니라 4-병렬 데이터 경로를 사용함으로써 높은 데이터 처리율을 지원한다. 또한, 제안하는 프로세서는 기존의 128/64-point FFT/IFFT 프로세서에 비해 낮은 하드웨어 복잡도를 지닌다. 제안된 FFT/IFFT 프로세서는 IEEE 802.11n 표준의 요구사항을 만족시키며 140MHz 클락 속도에서 560 MSample/s의 높은 데이터 처리율을 가진다.

Abstract

This paper presents a novel high-speed, low-complexity flexible 128/64-point radix-2^4 FFT/IFFT processor for the applications in high-throughput MIMO-OFDM systems. The high radix multi-path delay feed-back (MDF) FFT architecture provides a higher throughput rate and low hardware complexity by using a four-parallel data-path scheme. The proposed processor not only supports the operation of FFT/IFFT in 128-point and 64-point but can also provide a high data processing rate by using a four-parallel data-path scheme. Furthermore, the proposed design has a less hardware complexity compared with traditional 128/64-point FFT/IFFT processors. Our proposed processor has a high throughput rate of up to 560 Msample/s at 140 MHz while requiring much smaller hardware expenditure satisfying IEEE 802.11n standard requirements.

Keywords : FFT, MDF, radix-2^4, MIMO-OFDM, communications

I. Introduction

The growing demand of data, multimedia and communication has requested the need for generating many disparate devices into a high speed and efficiency ·bandwidth network capacity, with seamlessly supporting and integrating each sector's unique requirements. The multiple-input multiple-output (MIMO) schemes have been widely studied and received great attention by both academy and industry in wireless communication applications. The application of multiple antennas at both transmitter and receiver provides enhanced performance over wireless communications. This technique can significantly increase the data rates of wireless system without increasing system power or

* 학생회원, ** 평생회원, 인하대학교 정보통신공학부
(Dept. of Information and Communication Engineering, Inha University)
※ 본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT 연구센터 지원사업의 연구결과로 수행되었음 (IITA-2008-C1090-0019)
접수일자: 2008년7월10일, 수정완료일: 2009년2월5일

bandwidth. Besides, the cost of increasing data rates is just the additional multiple antennas, multi-dimensional signal processing block and the extra system equipment space. Now, the MIMO scheme has been adopted by the IEEE 802.11n Wireless LAN and IEEE 802.16e WiMAX standards.

Orthogonal Frequency Division Multiplexing) is a modulation method known for its capability to mitigate multipath[1]. It can provide a flat with the sub-carrier, and achieve high efficiency and ability to deal with frequency selective fading and narrowband interference. Also the flexibility using MIMO with OFDM can support the nonflat fading channels. The transceiver structure of MIMO-OFDM is depicted in Fig. 1. It consists of Channel Encoding and Interleaving, Mapping, IFFT, MUX, RF/IF Unit, antennas, DEMUX, FFT, Demapping, Deinterleaving, Channel Decoding and Synchronization. Based on the demand of data rate, the modulation scheme can be binary phase shift keying (BPSK), quaternary phase shift keying (QPSK), or quadrature amplitude modulation (QAM). The encoding rates in this specification are 1/2, 2/3, 3/4, or 7/8. The number of spatial sequence is supported by 1, 2, 3, or 4. The guard interval period is 400 ns or 800 ns. The bandwidth of the transmitted signal is 20 or 40 MHz [2]. The trend of MIMO design in FPGA implementation demands a low device expenditure and low system clock solution. Thus, we need to concentrate on the optimization of MIMO-OFDM transceiver system algorithm and structure.

The FFT/IFFT processor is one of the kernel modules having high computational complexity in the physical layer of the MIMO-OFDM system, which is used to process multi-carrier modulation. Since the MIMO-OFDM system needs more independent channel operators and processors, the system complexity and hardware cost dramatically increase. For example, a MIMO-OFDM system with N transceiver inputs and outputs require N basebands to be operated, and therefore N FFT/IFFT processors are necessary, also the system complexity increase N times. Thus, a scheme of high-speed, low-complexity FFT/IFFT processor can apparently reduce the complexity of MIMO-OFDM systems[3~4]. And, as the data transmission rate of MIMO-OFDM systems increases, generating OFDM symbols with high data rate requires high-efficiency FFT algorithm and architecture. This paper proposes a high-speed, low-complexity 128/64-point radix-2^4 FFT/IFFT processor with a four-parallel data-path and a multipath delay feed-back (MDF) structure to deal with the issues of the high-throughput and hardware complexity for MIMO-OFDM applications. It can provide a higher throughput rate and low hardware complexity.

This paper is organized as follows. Section II describes the design issues of the FFT/IFFT processor. Section III describes the proposed 128/64-point radix-2^4 MDF (R2^4MDF) FFT/IFFT

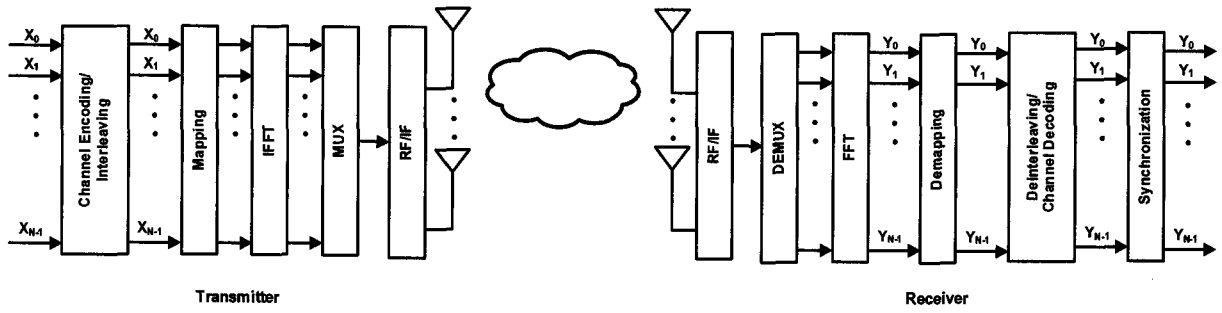

그림 1. 일반적인 MIMO-OFDM 시스템의 송수신부
Fig. 1. General transceiver/receiver of MIMO-OFDM system.

architecture. In Section Ⅳ, the hardware cost and throughput rate of the proposed FFT/IFFT architecture is compared with that of the traditional 128/64-point FFT/IFFT architectures. Conclusions are described in Section Ⅴ.

Ⅱ. Design Issue of the FFT/IFFT Processor

1. Processor architecture design issue

In order to implement the physical layer of the MIMO-OFDM system more efficiently and low clock frequency, the four data-path approach has been adopted to reduce the data sampling rate from the analog-digital converter (ADC), such that, after the serial-to-parallel (S/P) converter, the data sampling rate of each path can generally be reduced to 132 M samples/s[2, 5]. However, the hardware cost is also increased significantly, because more memory and complex multipliers are needed to allow multiple data to be operated simultaneously. Therefore, radix-2^4 SDF FFT/IFFT architecture has been proposed to offers high throughput, low hardware complexity and low power consumption by reduce the number of complex multiplications[5]. In pipelined FFT hardware scheme, the multipath delay commutator (MDC) scheme can achieve higher throughput rate by using multiple data paths, while the single-path delay feed-back (SDF) scheme needs less memory and hardware complexity with the delay feed-back scheme. Therefore, MDF FFT/IFFT architecture has been proposed to provide high throughput, low hardware complexity, and low power consumption[2]. Proposed radix-2^4 MDF architecture can provide higher throughput rate with minimal hardware cost by combining the features of MDC and SDF.

2. Radix-2^4 Algorithm

Discrete Fourier Transform of length N is defined as follows:

$$X(k)=\sum_{n=0}^{N-1}x(n)W_n^k \quad , \quad k=0,1,\ldots,N-1 \tag{1}$$

where W_n, the so called "twiddle factor", denotes the N-th primitive root of unity, with its exponent evaluated modulo$N$. k is the frequency index and n is the time index .In order to derive the radix-2^4 algorithm, consider the following first 4steps of decomposition. Applying a 5-dimensional linear index map.

$$\begin{aligned} n&=\left\langle \frac{N}{2}n_1+\frac{N}{4}n_2+\frac{N}{8}n_3+\frac{N}{16}n_4+n_5 \right\rangle_N \\ k&=<k_1+2k_2+4k_3+8k_4+16k_5>_N \end{aligned} \tag{2}$$

The Common Factor Algorithm (CFA) takes the form of

$$\begin{aligned} &X(k_1+2k_2+4k_3+8k_4+16k_5) \\ &=\sum_{n_5=0}^{\frac{N}{16}-1}\sum_{n_4=0}^{1}\sum_{n_3=0}^{1}\sum_{n_2=0}^{1}\sum_{n_1=0}^{1}x(\tfrac{N}{2}n_1+\tfrac{N}{4}n_2+\tfrac{N}{8}n_3+\tfrac{N}{16}n_4+n_5)W_N^{nk} \\ &=\sum_{n_5=0}^{\frac{N}{16}-1}[G_{\frac{N}{16}}(n_5,k_1,k_2,k_3,k_4)W_N^{n_5(k_1+2k_2+4k_3+8k_4)}]W_{\frac{N}{16}}^{n_5k_5} \end{aligned} \tag{3}$$

The twiddle factors can be expressed in the form of

$$\begin{aligned} W_N^{nk}&=(-1)^{n_1k_1}(-j)^{n_2(k_1+2k_2)}W_{16}^{(2n_3+n_4)(k_1+2k_2+4k_3)} \\ &\bullet(-1)^{n_4k_4}W_N^{n_5(k_1+2k_2+4k_3+8k_4)}W_{\frac{N}{16}}^{n_5k_5} \end{aligned} \tag{4}$$

The fourth butterfly unit has the expression of

$$\begin{aligned} &G_{\frac{N}{16}}(n_5,k_1,k_2,k_3,k_4)=H(n_5)+(-1)^{k_4}W_{16}^{(k_1+2k_2+4k_3)}H(n_5+\tfrac{N}{16}) \\ &+W_{16}^{2(k_1+2k_2+4k_3)}H(n_5+\tfrac{N}{8})+(-1)^{k_4}W_{16}^{3(k_1+2k_2+4k_3)}H(n_5+\tfrac{3N}{16}) \end{aligned} \tag{5}$$

where $H(n)$ denotes the second butterfly unit.

$$H(n)=H(n,k_1,k_2)=B(n,k_1)+(-j)^{(k_1+2k_2)}B(n+\tfrac{N}{4},k_1)$$

where $B(n,k_1)$ denotes the first butterfly unit as follows.

$$B(n,k_1)=x(n)+(-1)^{k_1}x(n+\tfrac{N}{2})$$

The twiddle factor in equation (5) has four complex numbers. The algorithm can take a complex constant multiplier instead of a programmable complex multiplier. Hence, the complex multiplication of twiddle factors, W_{16}^n can be implemented in the

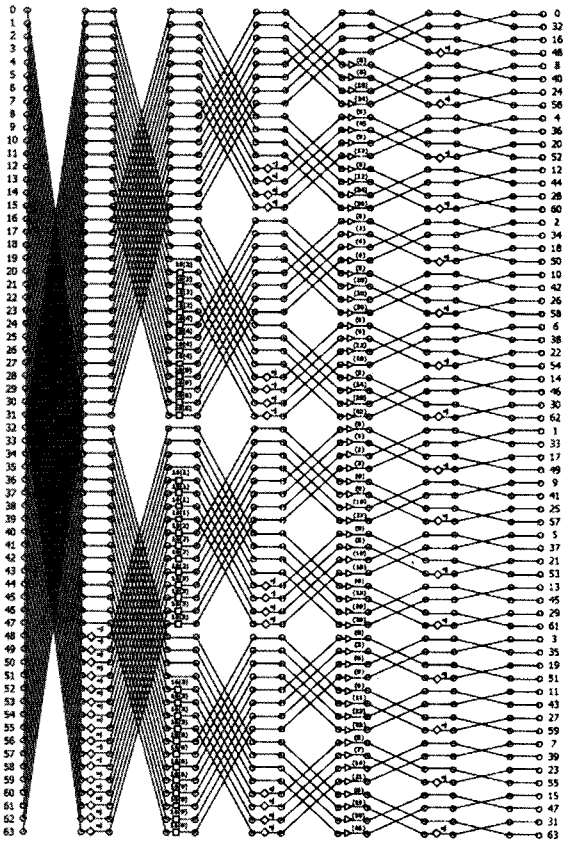

그림 2. 64-point radix-2^4 FFT 신호 흐름도
Fig. 2. The 64-point radix-2^4 FFT signal flow graph.

Canonic Signed Digit (CSD) constant multiplier, which contains the fewest number of non-zero bits. As such, the area and power consumption can be reduced[6]. Fig. 2 shows the signal flow graph (SFG) of the 64-point radix-2^4 FFT algorithm.

III. Proposed FFT/IFFT Architecture

The radix-2^4 algorithm can take complex constant multiplier instead of programmable complex multiplier. The Canonic Signed Digit (CSD) constant multiplier contains the fewest number of non-zero bits, so it can reduce the area and power consumption[6]. Fig. 3 shows the architecture of proposed four-parallel data-path 128/64-point R2^{4}MDF FFT/IFFT processor. The proposed design can support the operation of FFT/IFFT in 128-point or 64-point. It consists of memory units, butterfly units (BF_64, BF_128, BF1, BF2), complex Booth multipliers, CSD complex constant multipliers, multiplexers and adders. Also the four-parallel design requests four inputs and outputs, which operate the real and imaginary data separately.

Butterfly unit is the kernel of a FFT processor. For the multiply factors j in different butterfly steps, we design two kinds of similar butterfly units. The BF1 stores all of N/2-th input data in RAM. Meanwhile, the RAM keeps the swapping buffer space for the next input data. When (N/2+n)-th input data are fed to BF1, the input data x[n] stored in RAM are read and are added by new input data. And then the subtracted output data (x[n] - x[N/2+n]) are stored in the location of previous input data x[n]. When the new set of input data are fed to BF1 after

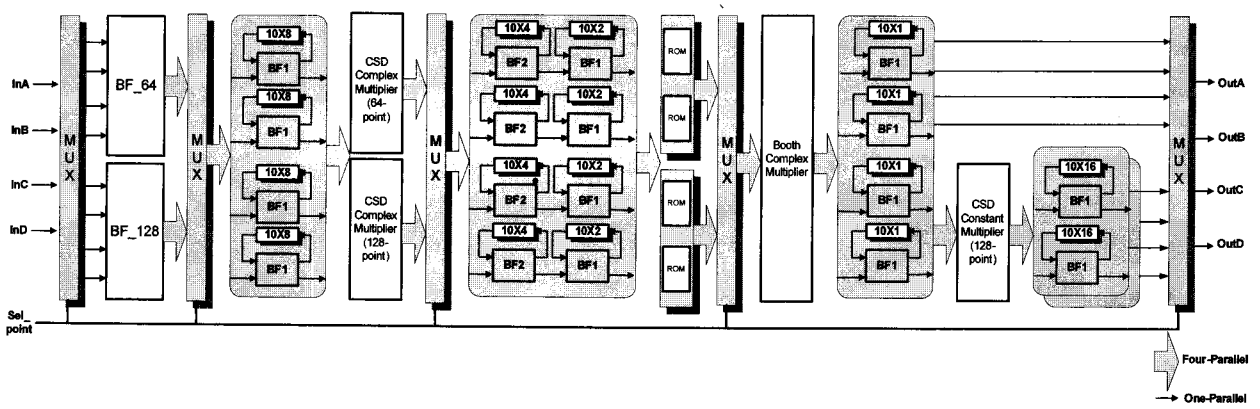

그림 3. 제안하는 four-parallel data-path 128/64-point Radix-2^4 MDF FFT/IFFT 프로세서
Fig. 3. Proposed four-parallel data-path 128/64-point Radix-2^4 MDF FFT/IFFT processor.

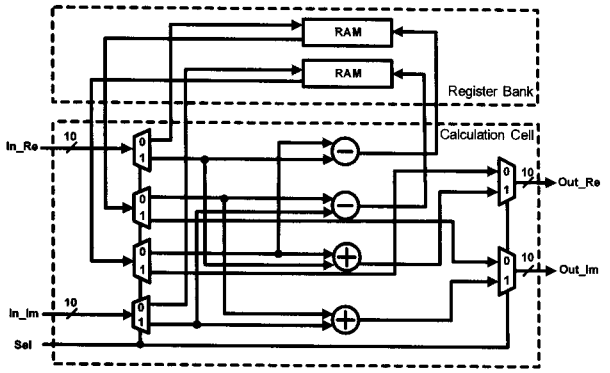

그림 4. Radix-2^4 butterfly 유닛 (BF1)
Fig. 4. Radix-2^4 butterfly units (BF1).

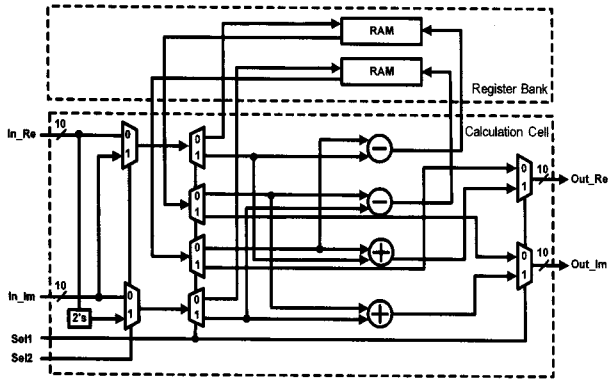

그림 5. Radix-2^4 butterfly 유닛 (BF2)
Fig. 5. Radix-2^4 butterfly units (BF2).

the N-th input operation, the new input data are stored in the RAM and (x[n]-x[N/2+n]) RAM. The BF2 architecture is almost same with the BF1 architecture except the operation of 3N/4-th input data multiplying of $-j$. The output results of butterfly units are complex addition and complex subtraction of two input data x[n] and x[N/2+n], where N is the value of point, as shown in Fig. 4 and 5. Based on these two kinds of butterfly unit architecture, the constant data processing ability is realized by the RAMs. Thus, this scheme can operate the input data and stored data efficiently. Besides, the two different kinds of BF design can attain the required speed with minimal silicon area.

The BF_64 block, which is MDC hardware design, consists of adders, subtracters, multiplexers and RAMs, which can store complex data. The architecture of BF_64 block is shown in Fig. 6. We design the individual processor module for each 64

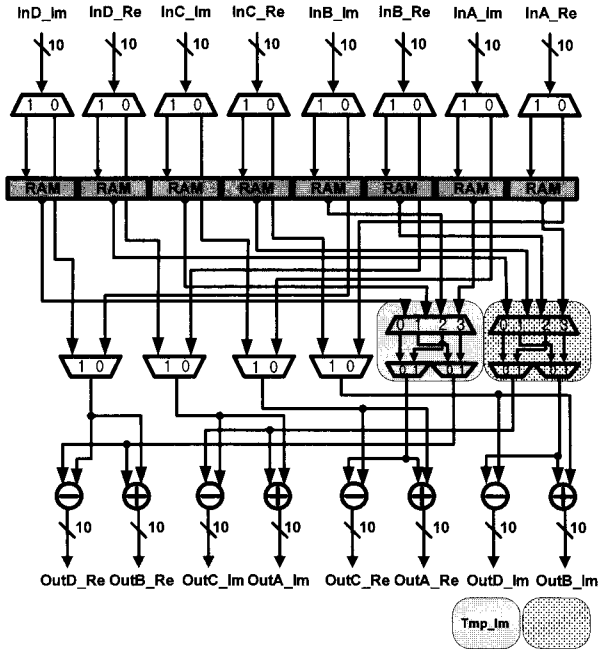

그림 6. BF 64의 블록도
Fig. 6. Block diagram of BF_64.

and 128-point with four-parallel data path through the different complex data storage. It operates as following steps: at the beginning 16 cycles, the first 32 data are stored in the register file. When the next N/2-th input data are stored in the register file, the previously stored input data x[2n-1] are read from the register file and generates the outputs. Then these outputs are added and subtracted by x[2n-1] and the new input data x[N/2+2n-1] simultaneously. And the input data x[N/2+2n] written into the module is stored in the location where the previous x[2n-1] data was. After x[N] data is stored in the register file, it write out data x[2n] and x[N/2+2n], those are added and subtracted in the module at the next cycle simultaneously. Contrast to most design using the 64 point butterfly units, our scheme can have a high operation speed by the multi-parallel scheme, and efficiently utilize the RAMs to accomplish the input data processing continuously.

Based on the radix-2^4 FFT algorithm, after the BF_32 block, it will be multiplied with trivial constant multipliers.

The radix-2^4 FFT algorithm with four-parallel data-path architecture has fewer multipliers than

other schemes of lower radix FFT algorithm. Thus, this algorithm can reduce the degree of multiplicative complexity efficiently[5]. The twiddle factors, W(8), W(16), W(24), and W(48) correspond to the cos(π/8) and sin(π/8) by using the formula of Trigonometric Functions. Table 1 shows the twiddle factors, which illustrates the 10-bits coefficients in the decimal representation, the 2's complement representation, and the CSD representation. To efficiently compensate the quantization error, the truncated bits are divided into two groups (major group and minor group) depending upon their effects on the quantization error. The error compensation bias is first expressed in terms of the truncated bits in the major group. The total CSD complex multiplier block consists of CSD constant multipliers, 2's complement logics, and multiplexers as shown in Fig. 7. When the real and imaginary values of twiddle factors are same, the two CSD constant multipliers are used and their two outputs are added to generate the output of the CSD complex multiplier. Otherwise, the CSD constant multipliers are used for the multiplication of input and twiddle factors. If inputs don't need to multiply with twiddle factor, the output is generated from the input directly.

For the four-parallel approach to implement the radix-2^4 FFT algorithm, we design the four-parallel data path complex Booth multipliers module. This kind of Booth multiplier needs a ROM to store the twiddle factor. The only 1/8 period of cosine and sine twiddle factors are needed, which are stored in ROM [5, 7]. To reduce the truncation error in the fixed-width multiplier, the Dadda reduction network with error-compensation circuit[8~9] was used, as shown in Fig. 8. It consists of Booth encoder, partial product generator (PPG), carry save adder (CSA), carry lookahead adder (CLA), subtractor and ROM. Each ROM stores the various twiddle factors. The Partial Product Generator, which generates appropriate partial products to be added with a Wallace tree. The proposed Dadda reduction network with error-compensation design can reduce full adder about 50% obviously than conventional Wallace tree.

표 1. Twiddle Factor의 CSD 이진 표현
Table 1. CSD Binary representation of Twiddle Factor.

Coefficients	Decimal	2's Comp.	CSD
$\sin(\frac{\pi}{8})$	0.3827	0011000011	$010\bar{1}00\bar{0}1\bar{0}$
$\cos(\frac{\pi}{8})$	0.9239	0111011001	$1000\bar{1}0\bar{1}0\bar{0}$

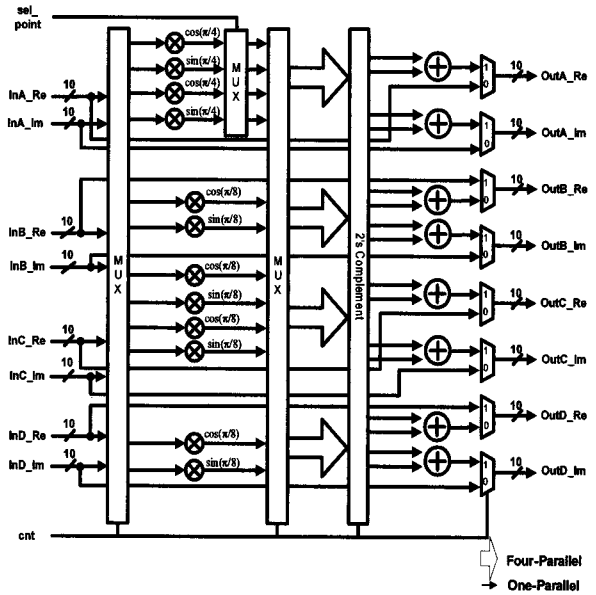

그림 7. 복소 곱셈기
Fig. 7. CSD complex constant multiplier.

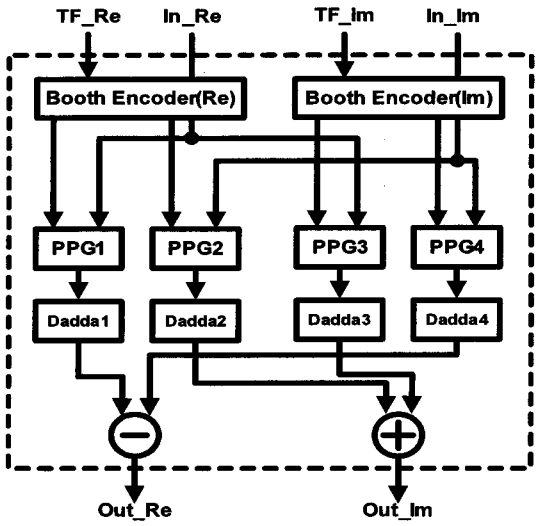

그림 8. 에러 보정 회로를 가진 Dadda 부스 복소 곱셈기
Fig. 8. Booth complex multiplier with Dadda reduction network with error-compensation circuit.

The architecture of proposed combining 128/64-point R2^{4}MDF FFT/IFFT architecture involves a selective MUX, the BF_64 or BF_128 modules, the CSD complex constant multiplier for each different point group, and the sharing butterfly units (BF1, BF2). If the operation of 64-point FFT/IFFT is performed, the calculated data will go into the BF_64 module, otherwise, it will be into the BF_128 module. The architecture can separate the input data into 64-point processing group or 128-point processing group, respectively. Each group contains four data sequences, A, B, C, and D through the different complex data storage. And then, the each data sequence will be processed by the 128/64 point sharing butterfly units (BF1, BF2). The last MUX can selective the 64 or 128 point output with four-parallel data path. This proposed design includes the individual processor groups for each 128 and 64-point with four-parallel data path. The scheme can support the operation of FFT/IFFT in 128-point or 64-point. The performance of this processor is switched by a control signal "sel_point". The logic 0 state of the "sel_point" signal sets the processor to operate in 64-point FFT/IFFT mode while its logic 1 state enables the processor to perform 128-point operation.

The proposed design was done with the aim to provide applicable throughput rates to achieve the requirements of IEEE 802.11n standard. And the 128-point or 64-point FFT/IFFT with four-parallel data path can be operated in our design. Besides, the hardware complexity of the complex multipliers can be reduced by using the radix-2^4 FFT algorithm and the CSD constant multipliers. And last, higher radix FFT algorithm can be implemented to save power dissipation regardless of the operation of 128-point or 64-point FFT.

Ⅳ. Implementation and Performance

The appropriate word length in the proposed processor is determined by a fixed-point simulation before hardware implementation. Based on the simulation results, 10 word lengths of the proposed FFT/IFFT architecture were determined in both real and imaginary parts. In addition, the SQNR of the proposed FFT/IFFT processor is about 30 dB.

After the appropriate word length was chosen, the FFT/IFFT processor was implemented using Xilinx Virtex-4 FPGA and functionally verified using Mentor Graphics ModelSim simulator. The operation of the FFT/IFFT processor is controlled by the control signal "mode", with the "0" is FFT and "1" is IFFT. And the switch of 128/64-point depends on the input signal "sel_point" at "0" or "1".

표 2. FFT/IFFT 프로세서의 성능 비교표
Table 2. Performance of the FFT/IFFT Processor.

	4-parallel R2^{4}MDF [Proposed]	8-parallel MRMDF [10]	4-parallel MRMDF [2]
Technology	Virtex-4 FPGA	Virtex-4 FPGA	0.13μm CMOS
No. of complex Multipliers	4 (50%)	8 (100%)	4 (50%)
No. of registers	64 (52%)	120 (97%)	124 (100%)
No. of Memory	32x10RAM 8 16x8ROM 8	-	64x10RAM 4 128x10ROM14
No. of Slice	5,633 (47%)	12,105 (100%)	-
Logic Gate Count	86k	-	-
Word Length	10 bit	-	12 bit
Parallel format	4 data-path	8 data-path	4 data-path
Radix	Radix-2^4	Mixed Radix (Radix 2, Radix 8)	Mixed Radix (Radix 2, Radix 8)
Max Clock rate	140 MHz	125 MHz	100 MHz
Throughput rate (R: clock rate)	4R (560 Msample/s)	8R (1 Gsample/s)	4R (400 Msample/s)
Standard	802.11n	802.15.3a	802.11n
SQNR	30 dB	-	-

Table 2 shows performance comparisons between the proposed processor and the traditional FFT/IFFT processors. From the comparison of the implementation results, the four-parallel 128/64-point R2^4MDF FFT/IFFT processor consists of 5,633 slices, and the highest throughput rate of the proposed architecture is as high as 560 Msample/s at 140 MHz.

V. Conclusion

This paper presented a four-parallel data-path pipelined combining 128/64-point radix-2^4 MDF FFT/IFFT processor. This design can operate the 128-point or 64 point FFT/IFFT processing flexibly. Besides, the four-parallel data-path scheme structure enables a high performance at a comparatively low clock frequency of 140 MHz, thus it saves mass of power dissipation without the limitation of the signal processing ability. In the proposed architecture, high-speed data processing and low hardware complexity can be achieved due to application of the CSD complex multipliers and a fixed-width Booth multiplier with a Dadda reduction network, which maintains the input and output at 10 bit width at 30dB SQNR. Also our design of twidder factors in CSD representation can utilize the CSD complex multipliers instead of a mount of programmable complex multipliers. Furthermore, the number of memories is effectively reduced by using the radix-2^4 MDF FFT algorithm. All these schemes can accomplish a high throughput performance by using low device expenditure.

Also based on this solution we compared the different architectures of 128/64-point FFT/IFFT processor. The performance results show that the data processing rate is 560 Msample/s while requiring small hardware complexity. The proposed architecture is expected to be incorporated in MIMO-OFDM systems such as IEEE 802.11n WLAN, IEEE 802.16e mobile WiMAX and 4G.

Reference

[1] J. A. C. Bingham, "Multicarrier modulation for data transmission: an idea whose time has come," IEEE Commun. Mag., vol. 28, no. 5, pp. 5-14, May 1990.

[2] Y-W. Lin, and C-Y. Lee, "Design of an FFT/IFFT processor for MIMO OFDM systems," IEEE Transactions on Circuits and Systems, vol. 54, no. 4, pp. 807-815, April 2007.

[3] G. L. Stuber, J. R. Barry, S. W. McLaughlin, Y. Li, M.A. Ingram, and T.H. Pratt, "Broadband MIMO-OFDM wireless communications", Proc. IEEE, 92, (2), pp. 271 - 294, 2004

[4] D. Borkowski, and L. Bruhl, "Optimized hardware architecture for real-time equalization in single and multi-carrier MIMO systems", 3rd Workshop on Software Radio, Karlsruhe, Germany, 2004

[5] J. Lee and H. Lee, "A High-Speed 2-Parallel Radix-2^4 FFT/IFFT Processor for MB-OFDM UWB Systems," IEICE Trans. on Fundamentals of Electronics, communications, and Computer Science, vol. E91-A, no. 4, pp. 1206-1211, April 2008

[6] S-M. Kim, J-G. Chung, and K. K. Parhi, "Low Error Fixed-width CSD Multiplier with Efficient Sign Extension," IEEE Trans. on Circuits and Systems-II, vol. 50, no. 12, December 2003.

[7] W-C. Yeh and C-W. Jen, "High-speed and low-power split-radix FFT," IEEE Trans. Acoust., Speech, Signal Process., vol. 51, no. 3, pp.864 - 874, March 2003.

[8] L. Dadda, "On Parallel Digital Multipliers," Alta Frequenza 45:574-580, 1976.

[9] K. J. Cho, K. C. Lee, J. G. Chung, and K. K. Parhi, "Design of Low-error. Fixed-width Modified Booth Multiplier," IEEE Trans. Very Large Scale. Integr. (VLSI), vol.12, no.5, pp.522 - 531, May 2004.

[10] L. Liu, J. Ren, X. Wang, F. Ye, "Design of Low-power, 1GS/s Throughput FFT Processor for MIMO-OFDM UWB Communication System," IEEE International Symposium on Circuits and Systems, vol.2, no.25, pp.2594 - 2597, May 2007.

저 자 소 개

리 우 항(학생회원)
2006년 베이징공과대학교 통신공학과 학사 졸업
2008년~현재 인하대학교 정보통신공학부 석사졸업
<주관심분야 : 통신용 VLSI설계 및 SoC설계>

이 한 호(정회원)
1993년 충북대학교 전자공학과 학사 졸업.
1996년 Univ. of Minnesota 전기컴퓨터공학 석사
2000년 Univ. of Minnesota 전기컴퓨터공학 박사
2002년 Member of Technical Staff, Lucent Technologies, USA.
2004년 Assistant Prof. Dept. of Electrical and Computer Engineering, Univ of Connecticut, USA
2004년~현재 인하대학교 정보통신공학부 부교수
<주관심분야 : 통신용 집적회로 및 SoC설계>