

MIMO 시스템을 위한 다채널 FFT 프로세서의 설계 및 구현

Design and Implementation of Multi-channel FFT Processor for MIMO Systems

정용철 · 조재찬 · 정운호*

한국항공대학교 항공전자정보공학부

Yongchul Jung · Jaechan Cho · Yunho Jung*

School of Electronics and Information Engineering, Korea Aerospace University, Gyeonggi-do, 10540, Korea

[요 약]

본 논문에서는 MIMO(multiple input multiple output) 시스템을 위한 저복잡도 FFT(fast Fourier transform) 프로세서의 설계 및 구현 결과를 제시하였다. 무선랜을 이용한 다양한 멀티미디어 서비스 등을 이용하기 위해 높은 채널 용량과 Gbps급 전송이 가능한 시스템에 대한 요구와 함께 IEEE 802.11ac 규격이 채택되었다. MIMO-OFDM (orthogonal frequency duplex multiplexing) 기술을 사용하는 IEEE 802.11ac 규격의 무선랜 시스템은 최대 8개의 안테나 구성 및 20-160 MHz 대역폭을 지원해야 한다. 따라서, 제안된 FFT 프로세서는 8채널 64, 128, 256, 512 point 가변길이를 지원한다. 또한, 비단순 승산기의 수를 감소시키기 위해서 MRMDC(mixed-radix multipath delay commutator) 구조를 적용하였고, 이로 인해 제안된 FFT 프로세서는 기존 FFT 프로세서에 비해 현저히 낮은 복잡도로 구현 가능하다. 구현 결과, 제안된 FFT processor는 기존 방식인 radix-2 SDF 구조 대비 gate count가 50% 감소 가능하였고, 8 채널 MR-2/2/2/4/2/4/2 MDC 구조와 8채널 MR-2/2/2/8/8 MDC 구조 대비 logic gate 수를 각각 18%와 17% 감소 가능성이 확인되었다.

[Abstract]

In this paper, a low complexity fast Fourier transform(FFT) processor is proposed for multiple input multiple output(MIMO) systems. The IEEE 802.11ac standard has been adopted along with the demand for a system capable of high channel capacity and Gbps transmission in order to utilize various multimedia services using a wireless LAN. The proposed scalable FFT processor can support the variable length of 64, 128, 256, and 512 for 8x8 antenna configuration as specified in IEEE 802.11ac standard with MIMO-OFDM scheme. By reducing the required number of non-trivial multipliers with mixed-radix(MR) and multipath delay commutator(MDC) architecture, the complexity of the proposed FFT processor was dramatically decreased. Implementation results show that the proposed FFT processor can reduced the logic gate count by 50%, compared with the radix-2 SDF FFT processor. Also, compared with the 8-channel MR-2/2/2/4/2/4/2 MDC processor and 8-channel MR-2/2/2/8/8 MDC processor, it is shown that the gate count is reduced by 18% and 17% respectively.

Key word : Fourier transform, Implementation, Multiple input multiple output systems, Mixed radix, Multipath delay commutator.

<https://doi.org/10.12673/jant.2017.21.6.659>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 23 October 2017; Revised 27 October 2017

Accepted (Publication) 27 November 2017 (30 December 2017)

*Corresponding Author; Yunho Jung

Tel: +82-2-300-0133

E-mail: yjung@kau.kr

I. 서론

MIMO 기술은 채널 대역폭을 증가시키지 않고 시스템의 용량을 증대시킬 수 있는 장점을 가지고 있으며, OFDM 방식은 N 개의 부반송파를 사용하여 병렬로 전송하기 때문에 다중 경로 페이딩 채널 환경에 강한 장점을 가진다. 이로 인해, MIMO와 OFDM을 결합한 MIMO-OFDM 기술이 고속 데이터 전송을 위한 핵심 기술로 각광 받고 있다 [1]-[4]. 무선랜 시스템인 IEEE 802.11n 시스템은 2.4 GHz의 무선 주파수 대역에서 MIMO-OFDM 기법을 적용하여 최대 600 Mbps의 전송 속도를 지원하도록 규정되어 있다 [5]. 그러나, 무선랜을 이용한 다양한 멀티미디어 서비스 등을 이용하기 위해 높은 채널 용량과 Gbps급 전송이 가능한 시스템에 대한 요구와 함께 IEEE 802.11ac 규격이 채택되었다 [6]. IEEE 802.11ac 무선랜 시스템은 최대 8개의 송수신 안테나를 지원하여 최고 6.9 Gbps의 전송속도를 지원하도록 규정하고 있으며, 20 MHz, 40 MHz, 80 MHz, 160 MHz의 가변대역폭을 지원해야 한다. 또한, 최대 8개의 데이터 경로를 갖기 때문에, 병렬 동작하는 8개의 OFDM 기저대역 프로세서가 필요하고 이는 single input single output (SISO)-OFDM에 비해 하드웨어적으로 8배의 복잡도 증가를 가져오게 된다. 무선랜 시스템에서 FFT 프로세서는 MIMO detector와 FEC(foward error correction) decoder와 함께 큰 복잡도를 갖는 구조 중 하나로, FFT 프로세서의 복잡도를 감소시키는 것이 MIMO-OFDM 무선랜 시스템의 하드웨어 복잡도를 감소시키는데 큰 기여를 한다고 할 수 있다. MIMO-OFDM 시스템에서 FFT 프로세서의 보편적인 설계 방법은 데이터경로 만큼 FFT 프로세서를 구현하는 것이며, 단일 경로에서 가장 복잡한 비단순 승산을 최소화하는 특징으로 인해 SDF(single-path delay feedback) 방식의 FFT 프로세서가 사용된다 [7]-[8]. 하지만, 병렬 구조는 하드웨어 복잡도가 데이터 경로 수에 따라 선형적으로 증가하는 문제가 생긴다. 이를 해결하기 위해, MDC (multi-path delay commutator) 방식의 FFT 프로세서를 사용하여 다중 데이터 경로를 하나의 FFT 프로세서로 처리함으로써 복잡도를 감소시키는 방법이 제안 되었다 [9]. Sansaloni에 따르면, 8 채널 FFT의 경우 radix-8 MDC 방식의 FFT 프로세서가 가장 면적이 효율적인 것으로 밝혀졌다 [10]. 그러나 제안된 MR(mixed radix)-4/2/8/8 알고리즘을 적용하면 radix-8 MDC 방식의 FFT 프로세서보다 면적을 줄일 수 있고, 제안된 알고리즘은 IEEE 802.11ac 규격에 맞게 64, 128, 256, 512 point 가변 길이 FFT 연산을 지원한다.

본 논문에서는 8x8 MIMO-OFDM 시스템에 저복잡도 FFT 프로세서를 제안하고, IEEE 802.11ac 무선랜 시스템에서 규정하는 20 MHz, 40 MHz, 80 MHz, 160 MHz의 대역폭을 지원 가능한 64, 128, 256, 512 point 가변 길이 8채널 FFT 프로세서의 최적 하드웨어 구조를 제안한다. 본 논문의 구성은 다음과 같다. 2장에서 FFT 알고리즘 및 하드웨어 구조에 대해 설명하고, 3장에서는 제안하는 FFT 프로세서의 하드웨어 구조를 설명한

다. 4장에서는 설계한 FFT 프로세서의 설계 및 구현 결과를 제시하고, 마지막으로 5장에서는 본 논문의 결론을 맺는다.

II. FFT 알고리즘 및 하드웨어 구조

FFT 알고리즘에는 radix-2, radix-4, radix-8 FFT 알고리즘, radix-2², radix-2³ FFT 알고리즘과 인덱스 분해법을 이용해 복소승산기 측면에서 이득을 얻을 수 있는 mixed radix FFT 알고리즘 등이 있다. 일반적으로 radix가 높을수록 복소수 승산 수는 적어지고 버터플라이 연산자의 복잡도는 증가한다. 따라서, 높은 radix 알고리즘의 버터플라이 연산자 복잡도를 감소시키기 위해 radix-2², radix-2³ FFT 알고리즘이 제안되었다. radix-2³ FFT 알고리즘은 radix-2 FFT 알고리즘과 같은 버터플라이 구조를 가지면서, 복소수 승산이 단순 승산으로 교체되어 radix-8과 같은 복소수 승산 수를 가진다.

FFT 프로세서의 하드웨어 구현을 위한 방법으로는 크게 단일 버터플라이 연산자를 이용하는 방법, 파이프라인 구조를 이용하는 방법, 병렬 구조를 이용하는 방법 등이 있다. 수율의 측면에서 병렬 구조가 가장 큰 이점을 갖지만, 하드웨어 복잡도가 너무 커지는 단점이 있다. 단일 버터플라이 연산자 구조는 가장 작은 하드웨어 복잡도를 갖으나 수율이 낮은 단점이 있다. IEEE 802.11ac 무선랜 시스템을 위한 FFT 프로세서의 경우, 고속 구현을 위해 수율이 커야하는 특징과 휴대용 시스템인 경우가 많으므로 복잡도가 작아야 한다는 제약이 있다. 따라서 수율과 복잡도의 교환 관계(trade-off)를 가장 잘 만족시키는 파이프라인 방식이 IEEE 802.11ac 무선랜 시스템을 위한 FFT 프로세서의 구현 방법으로 적합하다 할 수 있다. FFT 프로세서의 구현을 위한 파이프라인 방식은 크게 SDF, MDC 방식의 FFT 프로세서로 구분된다. SDF 방식의 FFT 프로세서는 단일경로를 통해 데이터를 전달하며, 변경된 버터플라이 연산자를 통해 데이터의 정렬을 수행한다. 경로당 하나의 복소 승산기를 요구하므로, 복잡도는 작아지나 단일 경로라는 특징으로 인해 수율이 낮고, 버터플라이 연산자의 구조가 복잡하므로 컨트롤이 어려운 단점을 갖는다. MDC 방식의 FFT 프로세서는 다중경로를 통해 데이터를 전달하며, 지연교환기로 SFG(signal flow graph)에 해당하는 데이터의 정렬을 수행한다. 이러한 특징으로 인해 큰 수율을 가지며 컨트롤이 쉬운 장점이 있지만, 경로당 하나의 복소 승산기를 요구하므로 복잡도가 증가하는 단점이 있다 [11].

MIMO-OFDM 시스템에서는 다수개의 입력 데이터 경로를 가지기 때문에 SDF 방식의 FFT 프로세서를 이용하면 경로 수에 따른 복소수 승산 수가 증가되어 하드웨어 복잡도가 증가하는 반면, MDC 방식의 FFT 프로세서는 하나의 FFT 프로세서로 설계 할 수 있어 하드웨어 복잡도를 크게 증가시키지 않고도 다수개의 데이터 경로를 처리할 수 있다. 따라서 MIMO-OFDM 시스템에서는 MDC 방식의 FFT 프로세서가 효율적이다.

$$\begin{cases} n = \frac{N}{4}n_1 + \frac{N}{8}n_2 + \frac{N}{64}n_3 + \frac{N}{512}n_4 \\ n_1 \in \{0,1,2,3\}, n_2 \in \{0,1\}, n_3 \in \{0,1,2,3,4,5,6,7\}, n_4 \in \{0,1,2,3,4,5,6,7\} \\ k = k_1 + 4k_2 + 8k_3 + 64k_4 \\ k_1 \in \{0,1,2,3\}, k_2 \in \{0,1\}, k_3 \in \{0,1,2,3,4,5,6,7\}, k_4 \in \{0,1,2,3,4,5,6,7\} \end{cases} \quad (2)$$

$$\begin{aligned} \sum_{n=0}^{511} x(n) W_{512}^{nk} &= \sum_{n_4=0}^7 \sum_{n_3=0}^7 \sum_{n_2=0}^1 \sum_{n_1=0}^3 \left[x(128n_1 + 64n_2 + 8n_3 + n_4) W_{512}^{(128n_1 + 64n_2 + 8n_3 + n_4)(k_1 + 4k_2 + 8k_3 + 64k_4)} \right] \\ &= \sum_{n_4=0}^7 \left(\sum_{n_3=0}^7 \left(\sum_{n_2=0}^1 \left(x(128n_1 + 64n_2 + 8n_3 + n_4) W_4^{n_1 k_1} \right) W_{512}^{4k_1} W_2^{n_2 k_2} \right) W_{16}^{n_3 k_3} W_8^{n_4 k_4} \right) W_{512}^{n_4 B} W_8^{n_4 k_4} \end{aligned} \quad (3)$$

III. 제안된 FFT 프로세서의 알고리즘 및 하드웨어 구조

3-1 제안된 FFT 프로세서의 알고리즘

N -point discrete fourier transform (DFT)는 수식 (1)과 같이 정의된다.

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{-kn} \quad (1)$$

IEEE 802.11ac 무선랜 규격에서 MIMO-OFDM을 위한 FFT 프로세서의 크기는 64/128/256/512-point이다. 따라서 제안된 MIMO-OFDM 기반의 무선랜 시스템을 위해서 하나의 FFT 프로세서로 64/128/256/512-point를 가변적으로 지원할 수 있어야 하므로 MR-4/2/8 분해 방법을 제안한다. 이 분해 방법을 이용하면 다양한 길이의 FFT 연산을 지원할 수 있고, 발생하는 twiddle factor (TF) 비단순 승산수를 최소화 할 수 있다.

512-point FFT 식을 표현하기 위해 수식 (1)의 n, k 를 인덱스 분해법을 이용해 수식 (2)와 같이 정리한다. 수식 (2)을 DFT의 정의식인 수식 (1)에 대입하여 정리하면, 512-point FFT 식을 수식 (3)와 같이 표현할 수 있으며, 수식 (3)의 변수 A, B 는 각각 $A = 64n_1 + 8n_3 + n_4$ 와 $B = 4k_2 + 8k_3$ 의 값을 갖는다. 수식 (3)을 통해 512-point FFT는 4-point FFT와 128-point FFT로, 256-point FFT는 2-point FFT와 128-point FFT로, 128-point FFT는 2-point FFT와 64-point FFT로 연산됨을 확인할 수 있다. 즉, 제안된 분해 방법은 처음의 4-point FFT와 복소수 승산 $W_{512}^{4k_1}$ 를 연산한 후에 2-point FFT와 $W_{16}^{n_3 k_3}$ 건너뛰면 쉽게 256-point FFT를 구현할 수 있다. 마찬가지로 첫 번째의 4-point FFT와 복소수 승산 $W_{512}^{4k_1}$ 을 건너뛰고 2-point FFT와 $W_{16}^{n_3 k_3}$ 을 연산하면 128-point FFT를, 첫 번째의 4-point FFT와 $W_{512}^{4k_1}$ 연산, 두 번째 2-point FFT와 $W_{16}^{n_3 k_3}$ 연산을 건너뛰면 64-point FFT를 쉽게 구

현할 수 있다. 이는 하나의 FFT 프로세서를 이용하여 다양한 크기의 FFT 연산을 지원할 수 있음을 의미한다. TF $W_{512}^{4k_1}$ 은 FFT 크기에 따라 변수의 값이 달라진다. 512-point FFT일 때 TF $W_{512}^{4k_1}$ 의 A 는 $64n_1 + 8n_3 + n_4$ 이며, 128-point FFT일 때 n_1 이 0이 되어 A 는 $8n_3 + n_4$ 가 된다. 또한, 수식 (2)에 나와 있는 TF $W_{512}^{n_4 B}$ 는 FFT 연산에 따라 수식 (4)처럼 표현 할 수 있다.

$$\begin{cases} W_{512}^{n_4 B} = W_{512}^{n_4(4k_2 + 8k_3)} \\ W_{256}^{n_4 B} = W_{256}^{n_4(2k_2 + 4k_3)} = W_{512}^{n_4(4k_2 + 8k_3)} \\ W_{128}^{n_4 B} = W_{128}^{n_4(k_2 + 2k_3)} = W_{512}^{n_4(4k_2 + 8k_3)} \\ W_{64}^{n_4 B} = W_{64}^{n_4(k_3)} = W_{512}^{8n_4 k_3} \end{cases} \quad (4)$$

수식 (4)에서 알 수 있듯이, B 는 512-point FFT일 때 $4k_2 + 8k_3$, 256-point FFT일 때 $2k_2 + 4k_3$, 128-point FFT 일 때 $k_2 + 2k_3$, 64-point FFT일 때 k_3 이다. 512-point FFT일 때 B 의 k_2 값이 0이면 $W_{512}^{n_4 B}$ 이 256-point FFT의 $W_{256}^{n_4 B}$ 와 같아지며, 이는 256-point FFT TF 값이 512-point FFT TF의 처음 256개와 같다는 의미가 된다. 이와 같이, $W_{512}^{n_4 B}$ 의 처음 128개가 128 point FFT TF인 $W_{128}^{n_4 B}$ 와 같고, $W_{512}^{n_4 B}$ 의 처음 64개가 64-point FFT TF인 $W_{64}^{n_4 B}$ 와 같다. 따라서, 다양한 FFT 연산에 맞는 TF의 인덱스가 결정됨으로써 발생하는 TF 비단순 승산을 최적화시켰다. 이렇게 최적화된 TF는 TF ROM (TFR) 하나에 저장되어 사용될 수 있으므로 64/128/256/512-point에 적합한 TFR을 구현할 수 있으며, 이로 인해서 하드웨어 복잡도를 감소시킬 수 있다. 표 1은 각 구조별 FFT 프로세서의 연산기 수를 비교하여 보여 준다. 제안된 MR-4/2/8 알고리즘은 16개의 비단순 승산으로 구현이 가능하며, 기존의 방법인 radix-2 SDF 방식에 비해 16개, MR-2/2/2/4/2/4/2 MDC 방식에 비해 6개, MR-2/2/2/8/8 MDC 방식에 비해 2개의 비단순 승산 수를 감소시킬 수 있다.

표 1. 각 구조별 FFT 프로세서의 연산기 수 비교

Table 1. Operator of various FFT architecture.

Architecture	Complex Adder	Complex Multiplier		Memory (N)
		Tri	Non-Tri	
R-2 SDF	144	32	32	4088
R-2/2/2/4/2/4/2 MDC	72	14	22	4088
R-2/2/2/8/8 MDC	72	34	18	4088
R-4/2/8/8 MDC	72	35	16	4088

*Tr : Trivial Multiplier *Non-Tri : Non-Trivial Multiplier

즉, 제안된 분해 방법을 이용함으로써 다양한 길이의 FFT 연산을 지원할 수 있을 뿐만 아니라 발생하는 TF 비단순 승산을 최적화시킬 수 있어, 복잡도 측면에서 가장 효율적인 알고리즘으로 판단된다.

3-2 제안된 FFT 프로세서의 하드웨어 구조

그림 1은 MIMO-OFDM 기반 IEEE 802.11ac 무선랜 시스템을 위한 제안된 64/128/256/512-point FFT 프로세서의 하드웨어 구조를 도시하였다. 제안된 FFT 프로세서의 하드웨어 구조는 data mapping module1 (DMM1), DMM2, DMM3, data reordering module (DRM), radix-4 butterfly module (R4BM), radix-2 butterfly module (R2BM), radix-8 butterfly module1 (R8BM1), R8BM2으로 구성된다. 동작 원리는 처음 1-8개의 입력 data가 DMM1을 통해 FFT 크기에 맞는 정확한 길이로 재구성 되어 다음 단계 입력된다. 만약 512-point FFT 일 경우, 재구성 되어진 8개의 data가 mux에 의해 R4BM에 입력되어 512-point FFT 연산이 이루어진다. 256-point FFT일 경우는 DMM1을 통해 256-point FFT에 맞게 재구성된 입력 데이터가 R4BM에 입력되고, 그 출력 데이터는 R8BM1으로 입력된다. 128-point FFT일 경우, DMM1을 통해 128-point에 맞게 재구성된 입력 데이터가 R4BM을 지나쳐 R2BM으로 입력된다. 64-point FFT일 경우, DMM1을 통해 64-point에 맞게 재구성된 입력 데이터가 R4BM과 R2BM을 지나 R8BM1으로 입력된다.

제안된 구조는 가변 길이의 FFT 프로세서를 지원하며 제안한 분해 방법을 통해 복소수 승산 수를 줄임으로써 하드웨어 비용을 줄일 수 있고, 구현도 간단해진다. 제안된 가변 길이 FFT 프로세서는 각 point 연산에 수행되는 연산과정의 수식을 통해 공통으로 사용되는 연산 블록을 공유하게 함으로써 하드웨어 복잡도를 줄일 수 있는 하드웨어 구조를 제안하였다. 또한 다수개의 하드웨어 자원을 사용하는 대신 다수개의 입력 데이터를 이용한 연산을 통해 하드웨어 효율성을 높였다.

1) Data Mapping Module1 (DMM1) 및 Data Reordering module (DRM)

DMM1은 다수개의 입력 데이터를 지연 소자를 이용해 일정한 데이터 정렬 패턴으로 8개의 병렬로 재구성하여 다음 단계로 밀어준다. 기존 FFT 프로세서 하드웨어 구조는 FFT point 수와 데이터 경로 수가 증가할수록 지연 소자로 인한 하드웨어 면적이 증가하는 문제가 있다. 기존 구조에서는 R4BM에 맞게 데이터를 재구성 해주는 DMM1 부분의 지연소자의 면적이 가장 크다. 따라서 본 논문에서는 그림 2와 같이 pre-commutator, post-commutator와 8개의 dual port RAM을 기존 DMM1의 지연소자와 commutator를 대신해 하드웨어 면적을 감소시키는 구조를 제안했다 [12]. Pre-commutator는 8개의 입력 데이터를 dual-port RAM에 맞게 재구성 해주는 역할을 한다. 이때, 64/128/256/512-point의 가변 길이를 지원해야 하므로 mux를 이용해 각각 64/128/256/512-point FFT에 맞게 데이터 정렬 패턴을 결정한다. dual port RAM의 출력은 post-commutator 구조에 맞게 재구성 해주며, post-commutator는 pre-commutator와 마찬가지로 mux를 이용해 MDC 구조에 맞게 재구성 해준다. 처음의 입력 구조에 맞게 재구성해 출력해주는 DRM도 DMM과 유사한 방법으로 구현된다.

2) Radix-4 Butterfly Module (R4BM)

R4BM의 세부 구조는 그림 3와 같으며, 2개의 radix-4 버터플라이 연산자, 3개의 단순 승산기, round & clip 그리고 교환기

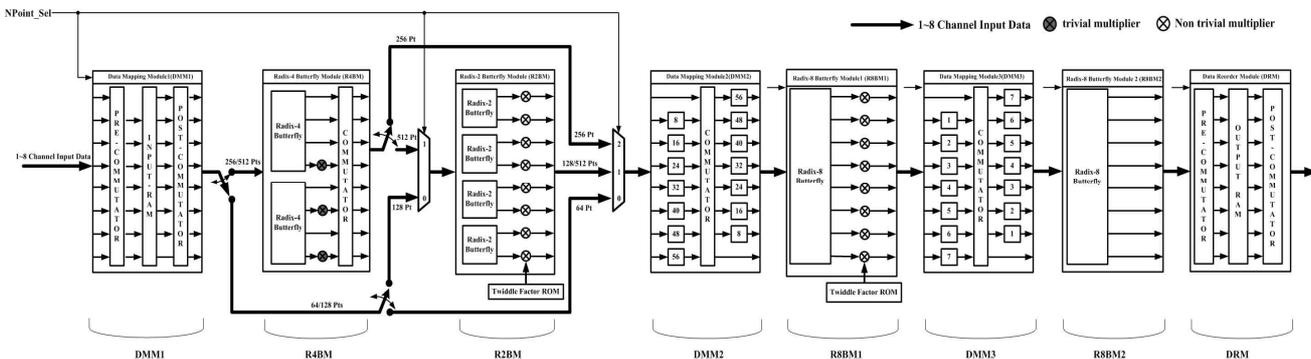


그림 1. 제안된 FFT 프로세서의 하드웨어 구조도
Fig. 1. Hardware block diagram of the proposed FFT processor.

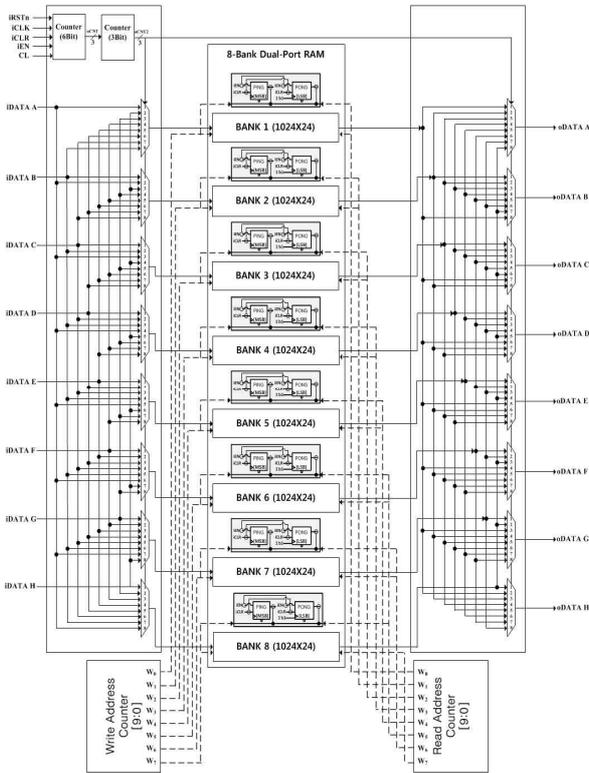


그림 2. 제안된 DMM1의 하드웨어 구조도
Fig. 2. Hardware block diagram of the proposed DMM1.

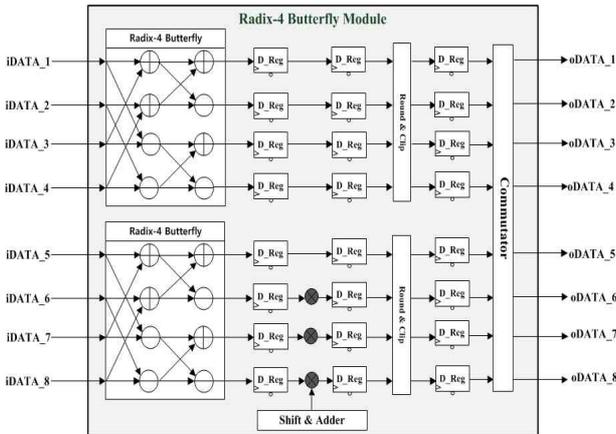


그림 3. 제안된 radix-4 버터플라이 모듈의 하드웨어 구조도
Fig. 3. Hardware block diagram of the proposed radix-4 butterfly module.

로 구성 되어있다. 512-point FFT 연산의 경우 DMM의 출력이 R4BM으로 입력된다. 이 8개의 데이터 경로는 2개의 radix-4 버터플라이 연산이 수행된 다음 트위들 팩터를 승산하기 위한 승산기를 통과하며, 이때의 승산기는 단순 승산기이다. 단순 승산기는 하드웨어적으로 복잡도가 큰 복소 승산기로 구현하지 않고, shift & adder로 구현한다. 단순 승산기를 통과한 데이터들은 round & clip을 거치고 교환기에 의해 다음 단계에 필요한 입력 데이터로 재구성된다.

3) Radix-2 Butterfly Module

R2BM의 세부 구조는 그림 4와 같으며, 4개의 radix-2 버터플라이 연산자, 8개의 비단순 승산기, round & clip으로 구성되어 있다. 512-point FFT 연산일 경우 R4BM의 8개의 출력이 입력으로 들어온다. 8개의 데이터 경로는 4개의 radix-2 버터플라이 연산이 이루어진 다음 8개의 비단순 승산기를 통과하고 round & clip을 거쳐 출력된다.

4) Radix-8 Butterfly Module

R8BM1의 세부 구조는 그림 5와 같으며, 1개의 radix-8 버터플라이 연산자, 8개의 비단순 승산기, round & clip으로 구성되어 있다. 입력으로 들어오는 8개의 데이터 경로는 1개의 radix-8 버터플라이 연산자를 수행한 다음 8개의 비단순 승산기를 통과하고 round & clip을 거쳐 출력된다. R8BM2의 세부 구조는 radix-8 버터플라이 연산자를 수행한 다음, 복소 승산기 연산 없이 바로 round & clip을 거쳐 출력된다.

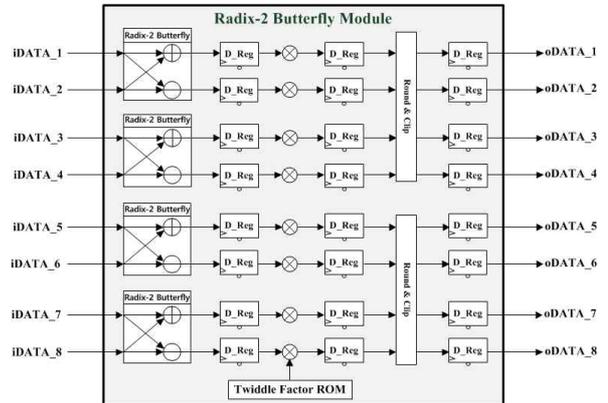


그림 4. 제안된 radix-2 버터플라이 모듈의 하드웨어 구조도
Fig. 4. Hardware block diagram of the proposed radix-2 butterfly module.

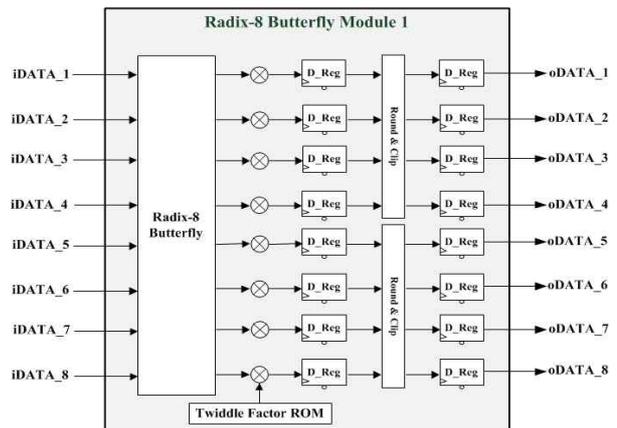


그림 5. 제안된 radix-8 버터플라이 모듈의 하드웨어 구조도
Fig. 5. Hardware block diagram of the proposed radix-8 butterfly module.

표 2. 제안된 FFT 프로세서의 부동 소수점 설계결과

Table 2. Fixed point design results for proposed FFT processor.

Data path (bits)	11	12	13	14
SQNR (dB)	42	49.5	55	62

표 3. 제안된 8채널 FFT 프로세서와 기존 FFT 프로세서와의 논리 합성 결과 비교 (DMM1, DRM 제외)

Table 3. Comparison for the logic synthesis results of the scalable FFT processors (without DMM1, DRM).

구조	Gate Count	Reduction (%)		
R2SDF	235K	100	-	-
MR-2/2/2/4/2/4/2 MDC	142K	60	100	-
MR-2/2/2/8/8 MDC	140K	59	99	100
MR-4/2/8/8 MDC	117K	50	82	83

IV. 제안된 FFT 프로세서의 설계 및 구현 결과

제안된 구조의 FFT 프로세서는 Matlab을 이용한 부동점 FFT 연산과 Verilog-HDL로 설계 후, 65nm의 CMOS 스탠다드 셀 라이브러리를 이용하여 논리합성 하였다. 표 2는 내부 데이터의 비트수에 따른 SQNR 값을 정리하여 보여준다. 정리된 결과를 통해 내부 비트 수는 SQNR이 49.5 dB인 12 bit를 선택하였다. 표 3는 논리 합성 결과를 정리한 표이며, 제안된 구조는 약 117K gate count로 구현이 되는 것을 확인 할 수 있다. 제안된 FFT 프로세서가 하드웨어 복잡도 면에서 효율적이라는 것을 검증하기 위해, 8 채널 512-point R2SDF, MR-2/2/2/4/2/4/2 MDC, MR-2/2/2/8/8 MDC 방식과 제안된 FFT 프로세서를 비교 하였다. 제안된 FFT 프로세서는 R2SDF 방식의 FFT 프로세서 구조보다 50%의 gate count를 감소시킬 수 있다. 또한 MR-2/2/2/4/2/4/2 MDC 구조와 MR-2/2/2/8/8 MDC 구조와 비교하면 각각 18% 및 17%의 gate count 감소 효과가 있다. 이 결과를 통해 MIMO-OFDM 시스템은 MDC 방식이 SDF 방식보다 효율적임을 알 수 있으며, R2SDF, MR-2/2/2/4/2/4/2 MDC, MR-2/2/2/8/8 MDC 방식보다 제안된 방식이 더 효율적임을 알 수 있다.

V. 결론

본 논문에서는 MIMO 시스템을 위한 FFT 프로세서 구조를 제안하고, 1-8 채널 64/128/256/512-point FFT 프로세서를 하드웨어로 구현 및 검증하였다. MDC 방식을 이용하여 다수개의 입력 데이터를 하나의 FFT 프로세서로 처리했으며, 제안된 mixed radix 분해 방법을 통해 복잡도 측면에서 가장 큰 비중을 차지하는 비단순 승산의 수를 줄임으로써 복잡도를 크게 낮추

었다.

제안된 구조를 갖는 FFT 프로세서의 복잡도 비교를 위해 R2SDF 구조, MR-2/2/2/4/2/4/2 구조, MR-2/2/2/8/8 구조, 제안 MRMDC 구조의 FFT 프로세서를 구현하였다. 그 결과 제안된 프로세서는 R2SDF 방식의 FFT 프로세서 구조보다 50%의 gate count를 감소시킬 수 있다. 또한 MR-2/2/2/4/2/4/2 와 MR-2/2/2/8/8 구조와 비교하면 각각 18% 및 17%의 감소된 gate count로 구현이 가능함을 확인 할 수 있었다.

Acknowledgments

본 논문은 산업통산자원부 및 한국산업기술평가관리원의 산업핵심기술개발사업 (10080619)의 일환으로 수행되었으며, CAD Tool은 IDEC으로부터 지원 받았음

References

- [1] H. Sampath, S. Talwar, J. Tellado, V. Erceg, and A. Paulraj, "A fourth-generation MIMO-OFDM: broadband wireless system: Design, performance, and field trial results," *IEEE Communications Magazine*, vol.40, No.9, pp.143-149, Sep.2002.
- [2] A. van Zelst, Tim C. W. Schenk, "Implementation of a MIMO OFDM-Based wireless LAN system," in *IEEE Transactions on Signal Processing*, vol. 52, no. 2, pp. 483-494, Feb. 2004.
- [3] G. L. Stuber, J. R. Barry, S. W. McLaughlin, Y. Li, M. A. Ingram, and T. H. Pratt, "Broadband MIMO-OFDM wireless communications," in *Proceedings of the IEEE*, vol.92, no.2, pp.271-297, Feb.2004.
- [4] H. Boleskei, and E. Zurich, "MIMO-OFDM wireless systems: basics, perspectives, and challenges," *IEEE Wireless Communications*, vol.13, no.4, pp.31- 37, Aug.2006.
- [5] IEEE Std. 802.11n, "Wireless LAN medium access control (MAC) and Physical Layer (PHY) specifications," 2005.
- [6] Standard for wireless LAN medium access control(MAC) and physical layer(PHY) Specifications : Enhancements for very high Throughput for Operation in Bands below 6 GHz, IEEE I. Std. P802.11ac, 2014
- [7] M. Dali, R. M. Gibson, A. Amira, A. Guessoum and N. Ramzan, "An efficient MIMO-OFDM radix-2 SDF FFT implementation on FPGA," in proceeding of Conference on Adaptive Hardware and Systems, Montreal, pp. 1-7, 2015.
- [8] S. Lee and S. C. Park, "Modified SDF Architecture for Mixed DIF/ DIT FFT," in Proceeding of IEEE international Symposium Circuit and Systems, New Orleans, pp. 1-5, 2006
- [9] L. R. Rabiner and B. Gold. "Theory and Application of Digital Signal Processing," *Prentice-Hall*, 1975.

[10] T. Sansaloni, A. Perex-Pascual, V. Torres, and J. Valls, "Efficient pipeline FFT processors for WLAN MIMO-OFDM systems," *Electronics Letters*, vol. 41, no. 19, pp. 1043-1044, Sep. 2005.

[11] Y. Jung, H. Yoon, and J. Kim, "New efficient FFT algorithm and pipeline implementation results for OFDM/DMT

applications," *IEEE Transactions on Consumer Electronics*, vol.49, no.1, pp.14- 20, Feb.2003.

[12] S. Yoshizawa, K. Nishi and Y. Miyanaga "An Area and Power Efficient Pipeline FFT Processor for 8x8 MIMO-OFDM Systems," in *IEEE International Symposium on Circuits and Systems*, pp.1248-1251, May 2008.



정 용 철 (YongChul Jung)

2015년 8월 : 한국항공대학교 전자 및 항공전자공학과 (공학사)
2017년 2월 : 한국항공대학교 항공전자정보공학부 (공학석사)
2017년 3월 ~ 현재 : 한국항공대학교 항공전자정보공학부 박사과정
※관심분야 : Radar SoC, VLSI 신호처리



조 재 찬 (Jaechan Cho)

2015년 8월 : 한국항공대학교 전자 및 항공전자공학과 (공학사)
2017년 2월 : 한국항공대학교 항공전자정보공학부 (공학석사)
2017년 3월 ~ 현재 : 한국항공대학교 항공전자정보공학부 박사과정
※관심분야 : 영상처리 SoC, VLSI 신호처리



정 윤 호 (Yunho Jung)

1998년 2월 : 연세대학교 전자공학과
2000년 2월 : 연세대학교 전기전자공학과 (공학석사)
2005년 2월 : 연세대학교 전기전자공학과 (공학박사)
2005년 ~ 2007년 : 삼성전자 책임연구원
2007년 ~ 2008년 : 연세대학교 연구교수
2008년 ~ 현재 : 한국항공대학교 교수
※관심분야 : 무선 통신 시스템, 항공통신 시스템, 영상처리 시스템, 모뎀 SoC, 영상처리 SoC