

논문 2009-46SD-12-13

# MIMO-OFDM 기반 SDR 시스템을 위한 효율적인 FFT 프로세서 설계

( Design of Efficient FFT Processor for MIMO-OFDM Based  
SDR Systems )

양기정\*, 정윤호\*\*

( Gijung Yang and Yunho Jung )

## 요약

본 논문에서는 MIMO-OFDM 기반의 SDR 시스템을 위한 효율적인 FFT 구조를 제안한다. 제안한 scalable FFT/IFFT 프로세서는 64/128/512/1024/2048-point FFT 연산을 가변적으로 수행할 수 있다. 또한 mixed radix (MR) 기법과 multi-path delay commutator (MDC) 구조를 사용하여 비단순 승산을 줄임으로써 기존의 설계 구조에 비해 시스템 수율 변화 없이 하드웨어 복잡도를 크게 감소시켰다. 제안된 scalable FFT/IFFT 프로세서는 하드웨어 설계 언어 (HDL)를 이용하여 설계 되었고, 0.18um CMOS 스탠다드 셀 라이브러리를 이용하여 논리 합성되었다. 논리합성 결과 4채널 radix-2 single-path delay feed-back (R2SDF) FFT 프로세서와 비교시 59% 감소된 게이트 수와 39% 감소된 메모리로 구현 가능함을 확인하였고, 4채널 radix-2 MDC (R2MDC) FFT 프로세서와 비교시 16.4% 감소된 게이트 수와 26.8% 감소된 메모리로 구현 가능함을 확인하였다.

## Abstract

In this paper, an area-efficient FFT processor is proposed for MIMO-OFDM based SDR systems. The proposed scalable FFT processor can support the variable length of 64, 128, 512, 1024 and 2048. By reducing the required number of non-trivial multipliers with mixed-radix (MR) and multi-path delay commutator (MDC) architecture, the complexity of the proposed FFT processor is dramatically decreased without sacrificing system throughput. The proposed FFT processor was designed in hardware description language (HDL) and synthesized to gate-level circuits using 0.18um CMOS standard cell library. With the proposed architecture, the gate count for the processor is 46K and the size of memory is 64Kbits, which are reduced by 59% and 39%, respectively, compared with those of the 4-channel radix-2 single-path delay feedback (R2SDF) FFT processor. Also, compared with 4-channel radix-2 MDC (R2MDC) FFT processor, it is confirmed that the gate count and memory size are reduced by 16.4% and 26.8%, respectively.

**Keywords :** MIMO-OFDM, MRMDC, Scalable FFT, SDR

## I. 서 론

최근 무선 통신은 고속 이동 데이터 전송에 대한 요구와 함께 급변하게 발전하고 있다. 특히, orthogonal

frequency division multiplexing (OFDM) 방식은 무선 채널에서 고속 데이터 전송에 적합한 방식으로  $N$ 개의 부 반송파를 사용하여 병렬로 전송하기 때문에 다중 경로 페이딩 채널 환경에 강한 장점을 가지며, 이로 인해 IEEE 802.11n WLAN, IEEE 802.16e mobile WiMax, 4G 시스템 등에서 고속 데이터 전송을 위한 변조방식으로 폭넓게 사용되고 있다<sup>[1~3]</sup>. 최근 보다 더 높은 고속 데이터 전송에 대한 요구와 함께 multiple-input multiple-output (MIMO) 기술이 활발하게 연구되고 있

\* 학생회원, \*\* 평생회원, 한국항공대학교  
(Korea Aerospace University)

※ 이 논문은 2009년 정부(교육과학기술부)의 재원으로  
한국연구재단의 지원을 받아 수행되었고(2009-  
0071983), CAD Tool은 IDEC으로부터 지원받았음.  
접수일자: 2009년 9월 30일, 수정완료일: 2009년 12월 2일

다. MIMO 기술은 채널 대역폭을 증가시키지 않고 시스템의 용량을 증대시킬 수 있는 장점을 가지고 있어 주파수 자원 제한에 대한 해결책으로 주목 받고 있으며, 이로 인해 MIMO와 OFDM을 결합한 MIMO-OFDM 기술이 고속 데이터 전송을 위한 핵심 기술로 각광 받고 있다<sup>[4~7]</sup>. 하지만, MIMO-OFDM은 다수개의 데이터 패스를 가지기 때문에 single-input single-output OFDM (SISO-OFDM)에 비해 하드웨어 복잡도가 증가하게 되므로, 증가하는 복잡도 문제를 해결하는 것이 가장 큰 이슈다. 특히, MIMO-OFDM 시스템에 있어서 fast Fourier transformation (FFT) 프로세서는 가장 큰 복잡도를 갖는 블록 중 하나로 최적의 FFT/IFFT 프로세서를 설계하는 것이 MIMO-OFDM 방식의 고속 데이터 통신을 위한 핵심 사항이라 할 수 있다<sup>[8~9]</sup>.

또한, 최근 software defined radio (SDR) 시스템이 차세대 무선 통신 시스템의 핵심 기술로 조명 받고 있다<sup>[10~12]</sup>. SDR 시스템은 주파수나, 출력, 변조 방식 등을 소프트웨어적으로 변경하여 다른 무선 통신 방식의 서비스를 제공받을 수 있게 해 주는 기술이다. SDR 모듈 탑재로 하나의 단말기에 두 개 이상의 시스템을 동시에 지원하는 것이 가능하므로, 사용 환경에 따라 무선 통신 방식을 변경할 수 있다. 특히, 실내에서는 IEEE 802.11n WLAN 시스템으로, 실외에서는 IEEE 802.16e mobile WiMax 시스템으로 사용 가능한 시스템이 큰 응용으로 관심 받고 있다. 이러한 시스템을 위한 FFT 프로세서는 IEEE 802.11n WLAN 시스템일 때 64/128-point, IEEE 802.16e WiMax 시스템일 때 128/512/1024/2048-point의 FFT 연산을 가변적으로 수행할 수 있어야 한다<sup>[13~15]</sup>.

MIMO-OFDM 시스템에서 FFT 프로세서의 가장 보편적인 설계 방법은 데이터 패스 수만큼 FFT 프로세서를 구현하는 것이다<sup>[16]</sup>. 이 경우 single-path delay feedback (SDF) 파이프라인 FFT 구조가 단일 패스에서 가장 복잡한 비단순 승산을 최소로 하는 특징으로 인해 보편적으로 사용되며<sup>[17~18]</sup>, 논문 [8]에서는 64/128-point의 1~4개의 채널을 지원 가능한 FFT/IFFT 프로세서를 제안하였다. 하지만, MIMO-OFDM은 다수개의 데이터 패스를 가지기 때문에 하드웨어 복잡도가 데이터 패스 수에 따라 선형적으로 증가하는 문제가 발생하게 된다. 이를 해결하기 위해 multi-path delay commutator (MDC)<sup>[19]</sup> 구조를 사용하여 다중 데이터 패스를 하나의 FFT 프로세서로 동

시에 처리함으로써 복잡도를 줄이는 방법이 제안되었다<sup>[20]</sup>. 이 논문에서는 MIMO-OFDM 시스템을 위한 FFT 구조의 하드웨어 복잡도를 분석하였으며,  $k$ 개의 입력 데이터 패스를 갖는 MIMO-OFDM 시스템의 경우,  $k$ 개의 radix-2 SDF (R2SDF) 구조보다 radix-4 MDC (RkMDC) 구조가 더 면적 효율적이며 제시되었다. 이후, 논문 [21~22]를 통해 4개의 송수신 안테나를 갖는 (4x4) MIMO-OFDM 시스템을 위한 radix-4와 radix-2를 혼합 mixed-radix MDC (MRMDC) 구조가 제안되었고, MRMDC 구조가 RkMDC 구조에 비해 비단순 승산의 수를 더 줄일 수 있음이 확인되었다. 그러나 [21~22]의 MRMDC 구조는 단순한 radix-4/2 연산의 반복으로 인해 다양한 길이의 FFT를 지원하지 못하는 문제점을 갖는다.

본 논문에서는 MIMO-OFDM 기반 SDR 시스템을 위한 최적 FFT 프로세서의 하드웨어 구조를 제안하였다. 제안된 FFT 프로세서는 최대 4채널 64/128/512/1024/2048-point 가변 길이 FFT 연산을 위해 고유의 mixed-radix 알고리즘과 MDC 구조에 기반하여 구현된다. 논문의 구성은 다음과 같다. II장에서 FFT 알고리즘 및 하드웨어 구조를 정리하고, III장에서 제안하는 FFT 프로세서의 하드웨어 구조를 설명한다. 이어서 IV장에서는 설계한 FFT 프로세서의 설계 및 구현 결과를 제시하고, 마지막으로 V장에서 본 논문의 결론을 맺는다.

### III. FFT 알고리즘 및 하드웨어 구조

고속 OFDM 기반 무선 통신 시스템을 위한 FFT 구조는 radix-2, radix-4 알고리즘을 바탕으로 하고 있다. 파이프라인 구조일 경우, radix-4는 radix-2에 비해 수율 면에서 이득을 얻을 수 있지만, 하드웨어 측면에서는 더 복잡한 구조를 가진다. 즉, radix-2 알고리즘이 radix-4 알고리즘에 비해 덜 복잡한 구조를 가짐으로서 면적 면에서의 이득을 얻을 수 있다. 일반적으로 radix가 높을수록 복소수 승산이 적어지고 하드웨어 구조는 더 복잡해진다. 이에 높은 radix 알고리즘의 복잡도를 줄이기 위해 radix-2<sup>2</sup>, radix-2<sup>3</sup>이 제안되었다<sup>[17~18]</sup>. Radix-2<sup>2</sup> 알고리즘은 radix-2 알고리즘과 같은 버터플라이 구조를 가지면서 복잡도를 낮추었고, radix-4 알고리즘과 같은 복소수 승산을 가진다<sup>[17]</sup>. 마찬가지로 radix-2<sup>3</sup> 알고리즘은 복소수 승산을 단순 승산으로 교체되어 radix-

$2^2$ 에 비해 복소수 승산을 더 줄일 수 있는 특성을 갖는다<sup>[18]</sup>.

파이프라인 구조는 SDF, MDC, single-path delay commutator (SDC) 으로, 크게 3가지 형태로 구분된다<sup>[23]</sup>. 이 중 OFDM 방식에서는 입출력 구조가 OFDM 시스템 구현에 적합하다는 점과 적은 하드웨어 자원을 사용하는 특징으로 인해 SDF 방식이 보편적으로 사용되어 왔다<sup>[18]</sup>. 하지만 MIMO-OFDM 시스템에서는 다수개의 입력 데이터 패스를 가지기 때문에 SDF 방식의 FFT 프로세서를 이용하면 패스 수에 따른 복소수 프로세서를 사용하게 됨으로써 하드웨어 복잡도 증가를 가져온다. 그에 반해 MDC 방식의 FFT 프로세서는 하나의 교환기를 추가하여 하나의 FFT 프로세서로 설계할 수 있어 하드웨어 복잡도를 크게 증가시키지 않고도 다수개의 데이터 패스를 처리 할 수 있다. 따라서 MIMO-OFDM 시스템에서는 MDC 방식이 더 효율적이다<sup>[22~23]</sup>.

이러한 알고리즘과 파이프라인 구조를 이용하여 효율적인 FFT 프로세서를 설계하기 위한 많은 방법들이 연구되어 왔으며, 그 중 비단순 승산을 줄임으로서 하드웨어 복잡도를 줄이는 단일 채널 MRMDC FFT 프로세서<sup>[21]</sup>와 다중 채널 MRMDC FFT 프로세서가 제안되었다<sup>[22]</sup>. 이 논문들에서는 MR 알고리즘과 MDC 구조를 적용하여 기존의 방식에 비해 비단순 승산의 수를 줄일 수 있음이 제시 되었으며, 표 1은 각 구조에 따른 하드웨어 복잡도를 비교하여 보여준다. 상수  $T$ 는 비단순 승산의 수를 의미하며,  $[D]$ 는  $D$ 를 넘지 않는 최대 정수를 의미한다. 정리된 결과를 통해 채널수가 4일 경우 MRMDC 방식의 FFT 프로세서가 가장 작은 하드웨어 복잡도를 갖게 됨을 알 수 있다.

### III. 제안된 FFT 프로세서의 알고리즘 및 하드웨어 구조

#### 1. 제안된 FFT 알고리즘

$N$ -point discrete Fourier transform (DFT) 의 식을 정의하면 다음과 같다.

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

IEEE 802.11n WLAN 및 IEEE 802.16e mobile

표 1. 다채널 N-point FFT 프로세서의 연산량 비교  
Table 1. Comparison fo the computational complexity for  $k$ -MIMO channel FFT processor.

	프로세서 수(P)	P당 복소곱셈수	P당 복소덧셈수
R2SDF	$k$	$\log_2 N - 1$	$2\log_2 N$
R4SDF	$k$	$\log_4 N - 1$	$8\log_4 N$
R4SDC	$k$	$\log_4 N - 1$	$3\log_4 N$
$R2^2$ SDF	$k$	$\log_4 N - 1$	$4\log_4 N$
$R2^2$ SDF	$k$	$\log_8 N - 1$	$(6+2T)\log_8 N$
R2MDC	$[k/2]$	$\log_2 N - 2$	$2\log_2 N$
R4MDC	$[k/4]$	$3(\log_4 N - 1)$	$8\log_4 N$
MRMDC	$[k/4]$	$4(\log_8 N - 1)$	$(12+3T)\log_8 N$

WiMax의 표준 규격<sup>[13~14]</sup>을 보면 MIMO-OFDM을 위한 FFT 크기는 IEEE 802.11n WLAN 시스템일 때 64/128-point이고, IEEE 802.16e mobile WiMax 시스템일 때 128/ 512/1024/2048-point이다. 따라서 제안하는 MIMO-OFDM 기반 SDR 시스템을 위해서는 하나의 FFT 프로세서로 64/128/512/1024/2048-point를 가변적으로 지원할 수 있어야 하므로 MR-2/2/4/2/4/2/4/2 분해 방법을 제안한다. 이 분해 방법을 이용함으로써 다양한 길이의 FFT 연산을 지원할 수 있을 뿐만 아니라, 발생하는 twiddle factor (TF) 비단순 승산을 최적화시킬 수 있다.

2048-point FFT 식을 표현하기 위해 수식 (1)의  $n, k$ 를 수식 (2)와 같이 정리한다. 수식 (2)을 수식 (1)에 대입하여 정리하면, 2048-point의 FFT 식을 수식 (3)과 같이 표현할 수 있으며, 수식 (3)의 변수  $x_{2048}, A_1, A_2, A_3, B_1, B_2$ 는 수식 (4)의 값을 갖는다. 수식 (3)을 통해 2048-point FFT는 2-point FFT와 1024-point FFT로, 1024-point FFT는 2-point FFT와 512-point FFT로, 512-point FFT는 4-point FFT와 128-point FFT로, 128-point FFT는 2-point FFT와 64-point FFT로 연산됨을 확인할 수 있다. 즉, 제안된 분해 방법은 처음의 2-point FFT와 복소수 승산  $W_4^{n_2 k_1}$ 을 건너뛰면 쉽게 1024-point FFT를 구현할 수 있다. 마찬가지로 두 번째의 2-point FFT와 복소수 승산  $W_{256}^{B_1 A_1}$ 을 건너뛰면 512-point FFT를, 3번째의 4-point FFT와 복소수 승산  $W_8^{n_3 k_3}$ 을 건너뛰면 128-point FFT를, 4번째의 2-point FFT와 복소수 승산  $W_{64}^{B_2 A_2}$ 을 건너뛰면 64-point FFT를 쉽게 구현할 수 있다. 이는 하나의 FFT

$$\left\{ \begin{array}{l} n = \frac{N}{2}n_1 + \frac{N}{4}n_2 + \frac{N}{16}n_3 + \frac{N}{32}n_4 + \frac{N}{128}n_5 + \frac{N}{256}n_6 + n_7 \\ n_1 \in \{0,1\}, n_2 \in \{0,1\}, n_3 \in \{0,1,2,3\}, n_4 \in \{0,1\}, n_5 \in \{0,1,2,3\}, n_6 \in \{0,1\}, n_7 \in \left\{0, \dots, \frac{N}{256} - 1\right\} \\ k = k_1 + 2k_2 + 4k_3 + 16k_4 + 32k_5 + 128k_6 + 256k_7 \\ k_1 \in \{0,1\}, k_2 \in \{0,1\}, k_3 \in \{0,1,2,3\}, k_4 \in \{0,1\}, k_5 \in \{0,1,2,3\}, k_6 \in \{0,1\}, k_7 \in \left\{0, \dots, \frac{N}{256} - 1\right\} \end{array} \right. \quad (2)$$

$$\sum_{n=0}^{2047} x(n) \cdot W_{2048}^{nk} = \sum_{n_7=0}^7 \left( \sum_{n_6=0}^1 \left( \sum_{n_5=0}^3 \left( \sum_{n_4=0}^1 \left( \sum_{n_3=0}^3 \left( \sum_{n_2=0}^1 \left( \sum_{n_1=0}^1 x_{2048} \cdot W_2^{n_1 k_1} \right) W_4^{n_2 k_2} \right) W_{256}^{n_3 k_3} \right) W_8^{n_4 k_4} \right) W_{64}^{n_5 k_5} \right) W_4^{n_6 k_6} \right) W_8^{n_7 k_7} \quad (3)$$

$$\left\{ \begin{array}{l} x_{2048} = x[1024n_1 + 512n_2 + 128n_3 + 64n_4 + 16n_5 + 8n_6 + n_7] \\ A_1 = k_1 + 2k_2, A_2 = k_3 + 4k_4, A_3 = k_1 + 2k_2 + 4k_3 + 16k_4 + 32k_5 + 128k_6 \\ B_1 = 16n_3 + 8n_4 + 2n_5 + n_6, B_2 = 2n_5 + n_6 \end{array} \right. \quad (4)$$

프로세서를 이용하여 다양한 크기의 FFT 연산을 지원할 수 있음을 의미하며, TF  $W_{256}^{B_1 A_1}$ ,  $W_{64}^{B_2 A_2}$ 은 각 FFT 크기에 따라 변수의 값이 달라진다. 2048-point FFT일 때 TF  $W_{256}^{B_1 A_1}$ 의  $B_1$ ,  $A_1$ 은 각각  $16n_3 + 8n_4 + 2n_5 + n_6$ ,  $k_1 + 2k_2$ 이며, 1024-point FFT일 때  $k_1$ 이 0이 되어  $A_1$ 은  $2k_2$ 이 된다. 마찬가지로 512-point FFT일 때 TF  $W_{64}^{B_2 A_2}$ 의  $B_2$ ,  $A_2$ 은 각각  $2n_5 + n_6$ ,  $k_3 + 4k_4$ 이며, 128-point FFT일 때  $k_3$ 이 0이 되어  $A_2$ 은  $4k_4$ 이 된다. 다양한 FFT 연산에 맞는 TF의 인덱스가 결정됨으로써 발생하는 TF 비단순 승산을 최적화시킨다.

또한, 수식 (3)에 나와 있는 TF  $W_{2048}^{n_7 A_3}$ 은 FFT 연산에 따라 수식 (5)처럼 표현할 있다.

$$\left\{ \begin{array}{l} W_{2048}^{n_7 A_3} = W_{2048}^{n_7(k_1 + 2k_2 + 4k_3 + 16k_4 + 32k_5 + 128k_6)} \\ W_{1024}^{n_7 A_3} = W_{1024}^{n_7(k_2 + 2k_3 + 8k_4 + 16k_5 + 64k_6)} \\ \quad = W_{2048}^{2n_7(k_2 + 2k_3 + 8k_4 + 16k_5 + 64k_6)} \\ W_{512}^{n_7 A_3} = W_{512}^{n_7(k_3 + 4k_4 + 8k_5 + 32k_6)} \\ \quad = W_{2048}^{4n_7(k_3 + 4k_4 + 8k_5 + 32k_6)} \\ W_{128}^{n_7 A_3} = W_{128}^{n_7(k_4 + 2k_5 + 8k_6)} = W_{2048}^{16n_7(k_4 + 2k_5 + 8k_6)} \\ W_{64}^{n_7 A_3} = W_{64}^{n_7(k_5 + 4k_6)} \quad = W_{2048}^{32n} \end{array} \right. \quad (5)$$

수식 (5)에서 알 수 있듯이  $A_3$ 은 2048-point FFT일

때  $k_1 + 2k_2 + 4k_3 + 16k_4 + 32k_5 + 128k_6$ , 1024-point FFT일 때  $k_2 + 2k_3 + 8k_4 + 16k_5 + 64k_6$ , 512-point FFT일 때  $k_3 + 4k_4 + 8k_5 + 32k_6$ , 128-point FFT일 때  $k_4 + 2k_5 + 8k_6$ , 64-point FFT 일 때  $k_5 + 4k_6$ 이다. 2048-point FFT일 때  $A_3$ 의  $k_1$  값이 0이면  $W_{2048}^{n_7 A_3}$ 이 1024-point FFT의  $W_{1024}^{n_7 A_3}$ 와 같아지며, 이는 1024-point FFT TF 값이 2048-point FFT TF의 처음 1024개와 같다는 의미가 된다. 마찬가지로  $W_{2048}^{n_7 A_3}$ 은  $k_1, k_2$ 의 값이 0이면 512-point FFT TF  $W_{512}^{n_7 A_3}$ 와 같아지고,  $k_1, k_2, k_3$ 의 값이 0이면 128-point FFT TF  $W_{128}^{n_7 A_3}$ 와 같아지고,  $k_1, k_2, k_3, k_4$ 의 값이 0이면 64-point FFT  $W_{64}^{n_7 A_3}$ 와 같아진다. 즉, 하나의 2048-point FFT twiddle factor ROM (TFR) 으로 모든 64/128/512/1024/2048-point TFR을 구현할 수 있으며, 이로 인해 하드웨어 복잡도를 줄일 수 있다.

제안된 MR-2/2/4/2/4/2/4/2 분해방법은 2개의 radix-2을 사용하여 2048/1024-point FFT를 지원해 주며, 3개의 비단순 승산을 사용함으로써 TF 비단순 승산을 최적화시켰다. 또한, radix-4와 radix-2를 혼용하여 512/128/64-point FFT를 지원해 주며, 8개의 비단순 승산을 사용함으로써 TF 비단순 승산을 최적화 시켰다. 즉, 제안된 MR 알고리즘은 12개의 비단순 승산으로 구현 가

능하며, 기존의 방법인 R2SDF에 비해 20개, R2MDC에 비해 4개의 비단순 승산수를 감소시킬 수 있다.

## 2. 제안하는 FFT 구조

그림 1은 MIMO-OFDM 기반 SDR 시스템을 위한 제안된 64/128/512/1024/2048-point scalable FFT 프로세서의 하드웨어 구조를 도시한다. 제안된 FFT 프로세서 구조는 data mapping module (DMM), radix-2 butterfly module 1 (R2BM1), R2BM2, R2BM3, R4BM1, R4BM2, data reordering module (DRM)으로 구성된다. 처음에는 1~4개의 입력 데이터가 DMM을 통해 FFT 크기에 맞는 정확한 길이로 재구성되어 다음 단에 입력된다. 만약 2048-point FFT일 경우, 재구성되어진 1~4개의 데이터가 MUX에 의해 R2BM1에 입력되며, 2048-point FFT 연산이 이루어진다. 1024-point FFT이면, DMM의 출력 데이터는 R2BM1을 지나쳐 첫 번째의 R2BM2의 입력으로, 512-point FFT이면 R4BM1의 입력으로, 128-point FFT이면 두 번째의 R2BM2의 입력으로, 64-point FFT이면 R4BM2으로 입력된다. 각 연산 블록들은 butterfly (BF), commutator, 단순 및 비단순 승산기 등으로 구성되며, 비단순 승산기는 TFR이 추가된다. DRM은 연산되어진 데이터들을 재정렬 (re-ordering) 해 준다.

제안된 구조는 다양한 길이의 FFT 프로세서를 지원해 줄 수 있으며 제안한 분해방법을 통해 복소수 승산

을 줄임으로써 하드웨어 비용을 줄일 수 있고, 구현에 있어서도 간단해진다. 또한, 다수 개의 하드웨어 자원을 사용하는 대신 다수개의 입력 데이터를 이용한 연산을 통해 하드웨어 효율성을 높일 수 있다.

### (1) Data Mapping Module (DMM)

제안된 DMM은 그림 2와 같이 지연소자, MUX, switch으로 구성되며, 다수개의 입력 데이터를 4개의 병렬로

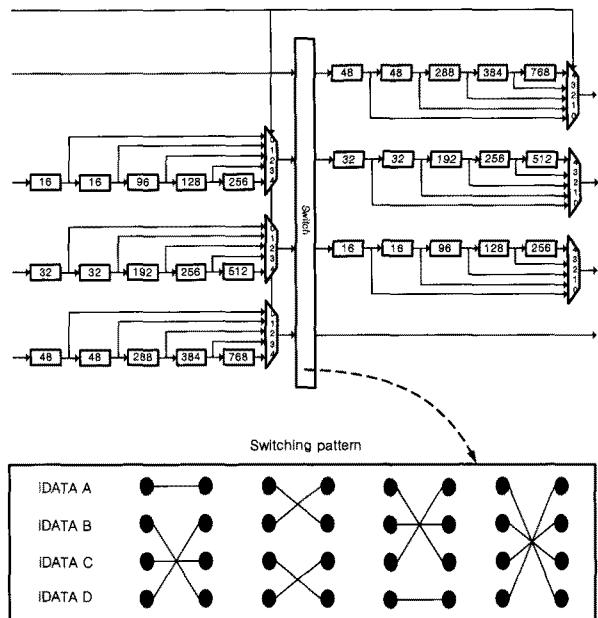


그림 2. 제안된 DMM 구조

Fig. 2. Block diagram of the proposed DMM.

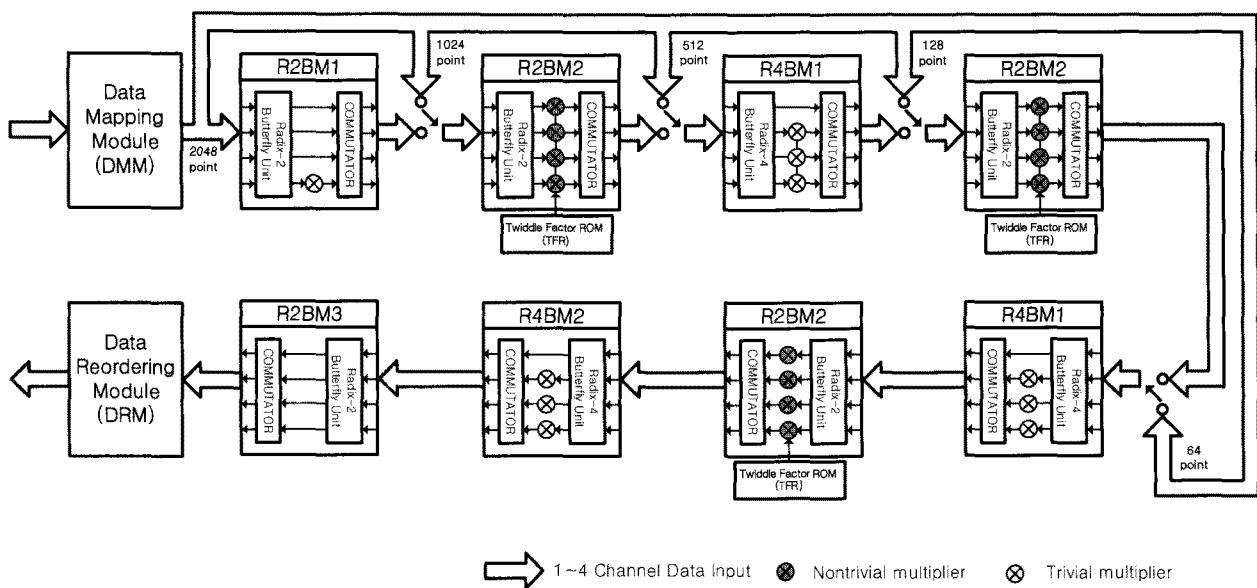


그림 1. MIMO-OFDM 기반 SDR 시스템을 위한 제안된 FFT 프로세서 블록 다이어그램.

Fig. 1. Block diagram of the proposed FFT processor for MIMO-OFDM Based SDR Systems.

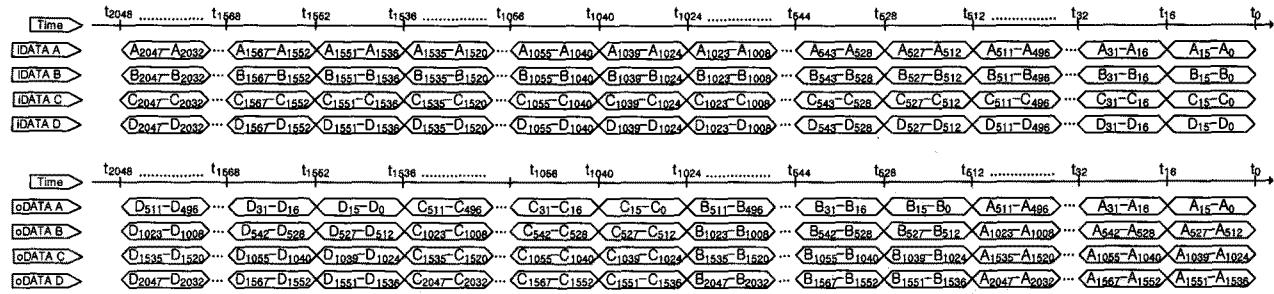


그림 3. DMM의 데이터 흐름과 연산 (2048-point)

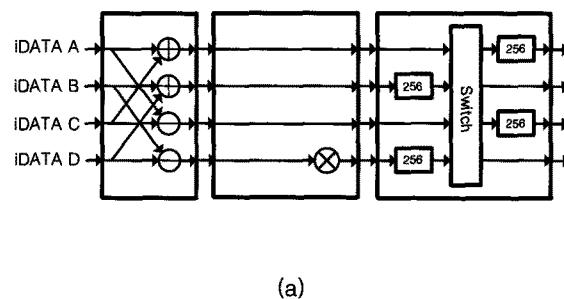
Fig. 3. Data scheduling and the operation of DMM. (2048-point).

재구성하여 다음 단으로 밀어준다. 이때 각 64/128/512/1024/2048-point FFT 지연 길이는 16, 32, 128, 256, 512 으로 MUX에 의해 결정되며, 이로 인해 하나의 DMM 으로 다양한 길이의 FFT 프로세서를 지원할 수 있다. Switch는 1~4개의 입력 데이터를 MDC 구조에 맞게 재구성해주며, switching pattern은 그림 2의 아랫부분과 같다. switch에 의해 재구성된 2048-point FFT의 입력 및 출력 데이터를 그림 3에 도시하였으며, 각 1~4개의 데이터 패스에 따라 512 cycle, 1024 cycle, 1536 cycle, 2048 cycle 동안 R2BM1으로 입력된다.

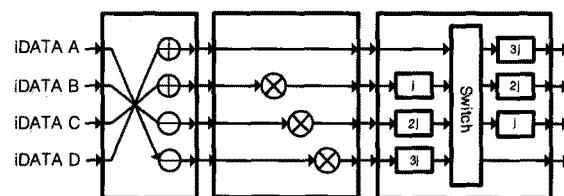
또한, 여기에 쓰이는 MUX 신호는 R2BM2의 TFR의 값 결정에 쓰이는 신호와 일치한다. 이는 FFT 크기에 따른 R2BM2의 TFR의 주소 생성을 쉽게 구현 가능하도록 한다.

## (2) Radix-2 Butterfly Module (R2BM) 및 Radix-4 Butterfly Module (R4BM)

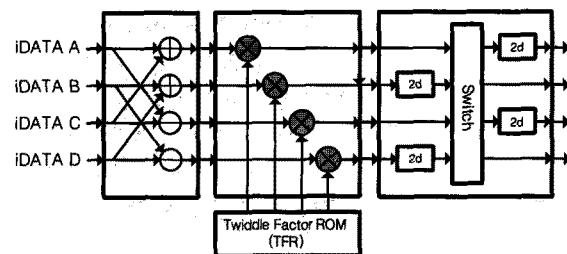
그림 4는 제안된 FFT 프로세서의  $i$ 번째 R2BM (R2BM $_i$ ) 과  $i$ 번째 R4BM (R4BM $_i$ )의 세부 구조를 도시한다. R2BM $_i$ 과 R4BM $_i$ 은  $i$ 의 값에 따라 R2BM1, R2BM2, R2BM3, R4BM1, R4BM2이 된다. R2BM1의 세부 구조는 그림 4-(a)와 같으며, 2개의 radix-2 BF, 1개의 비단순 승산기, commutator으로 구현되며, commutator의 delay factor (DF) 크기는 256이다. R2BM2의 세부 구조는 그림 4-(b)와 같이 2개의 radix-2 BF, 4개의 비단순 승산기, TFR, commutator으로 구현되며, DF 크기는  $2d$  ( $d=1, 8, 64$ )이다. 첫 번째의 R2BM2는  $d$ 의 값이 64, 두 번째의 R2BM2는  $d$ 의 값이 8, 세 번째의 R2BM2 은  $d$ 의 값이 1이 되며,  $d$ 에 따라 DF 크기는 128, 16, 2



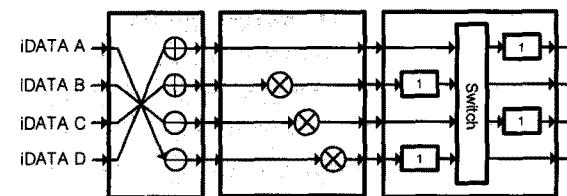
(a)



(c)



(b)



(d)

그림 4. 제안된 (a) R2BM1, (b) R2BM2, (c) R4BM1, (d) R4BM2 구조

Fig. 4. Block diagram of the proposed (a) R2BM1, (b) R2BM2, (c) R4BM1 and (d) R4BM2.

가 된다. 첫 번째 R2BM2에서 연산되는 TFR은  $W_{256}^{B_1 A_1}$ , 두 번째 R2BM2에서 연산되는 TFR은  $W_{64}^{B_2 A_2}$ , 세 번째

R2BM2에서 연산되는 TFR은  $W_{2048}^{n_7 A_3}$ 으로, R2BM2의 TFR은 각 64/128/512/1024/2048-point FFT에 따라 가변적으로 생성된다. R2BM3의 세부 구조는 그림 4-(a)에서 단순 승산기를 제외한 구조로, 2개의 radix-2 BF, commutator으로 구현되며, DF 크기는 1이다. R4BM1의 세부 구조는 그림 4-(c)와 같이 radix-4 BF, 3개의 단순 승산기, commutator으로 구현되며, DF 크기는  $j, 2j, 3j$  ( $j=4, 32$ ) 이다. 첫 번째 R4BM1의  $j$  값은 32로 DF 크기는 32, 64, 96이 되며, 두 번째 R4BM1의  $j$  값은 4로 DF 크기는 4, 8, 12이 된다. R4BM2의 세부 구조는 그림 4-(d)와 같이 radix-4 BF, 3개의 단순 승산기, commutator으로 구현되며, DF 크기는 1이다.

#### IV. 제안된 FFT 프로세서의 설계 및 구현 결과

제안된 FFT 프로세서는 Verilog HDL을 이용하여 설계되었고, ModelSim을 이용한 기능검증과 Matlab을 이용한 부동점 FFT 연산을 수행하였다. 표 2는 내부 데이터 길이에 따른 SQNR을 정리하여 보여준다. 정리된 결과를 통해 내부 데이터 패스는 SQNR 42.5dB인 12bit를 선택하였다. 또한, 0.18um의 CMOS 스탠다드 셀 라이브러리를 이용하여 논리합성 되었으며, 합성된 결과 제안된 FFT 프로세서는 약 46K 게이트 수와 약 90K 메모리로 구현되는 것을 확인하였으며, 40MHz로 동작하는 제안된 FFT 프로세서의 시뮬레이션 결과를 그림 5에 도시하였다. 제안된 FFT 프로세서가 하드웨어 복잡도 면에서 효율적이라는 것을 검증하기 위해, 4

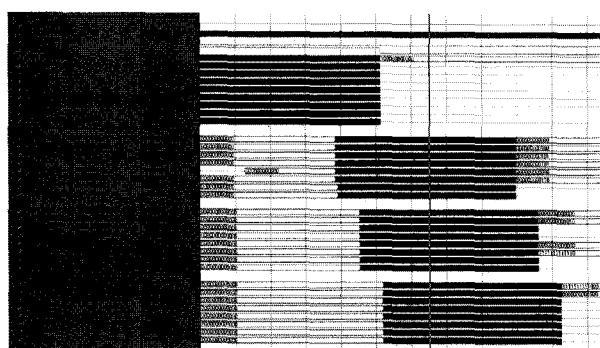


그림 5. 제안된 FFT 프로세서의 시뮬레이션 결과  
Fig. 5. Simulation result of the proposed FFT processor.

표 2. 제안된 FFT 프로세서의 부동점 시뮬레이션  
Table 2. Fixed point simulation for proposed FFT processo.

Data path(bits)	11	12	13	14
SQNR[dB]	38	42.5	53	62

표 3. Scalable FFT/IFFT 논리 합성 결과 비교  
Table 3. Comparison for the logic synthesis results of the scalable FFT/IFFT processors (2048-point)

	총 게이트 수	메모리(bits)
4*R2SDF	111K (100%)	147K (100%)
2*R2MDC	55K (50%)	123K (84%)
Proposed	46K (41%)	90K (61%)

채널 2048-point R2SDF 및 2048-point R2MDC 방식의 하드웨어를 설계 및 구현하였고, 표 3은 비교 결과를 정리하여 보여준다. 논리합성 결과, R2SDF 방식의 FFT 프로세서는 약 111K 게이트 수와 약 147K 메모리로 구현되었으며, R2MDC 방식의 FFT 프로세서는 약 55K 게이트 수와 약 123K 메모리로 구현되는 것을 확인하였다. 이 결과를 통해 MIMO-OFDM 시스템은 MDC 방식이 SDF 방식보다 효율적임을 알 수 있으며, R2MDC 방식보다 제안된 방식이 더 효율적임을 알 수 있다.

#### V. 결 론

본 논문에서는 MIMO-OFDM 기반 SDR 시스템을 위한 scalable FFT/IFFT 프로세서 구조를 제안하고 1-4채널 64/128/512/1024/2048-point FFT 프로세서를 하드웨어로 구현 및 검증하였다. MDC 방식을 이용하여 다수개의 입력 데이터를 하나의 FFT 프로세서로 처리했으며, 제안된 MR 분해 방법을 통해 복잡도 측면에서 가장 큰 비중을 차지하는 비단순 승산의 수를 줄임으로써 복잡도를 크게 낮추었다.

제안된 구조를 갖는 FFT 프로세서의 복잡도 비교를 위해 R2SDF 구조, R2MDC 구조, 제안된 MRMDC 구조의 FFT 프로세서를 구현하였다. 그 결과 제안된 프로세서는 R2SDF FFT 프로세서에 비해 약 59% 감소된 게이트 수와 약 39% 감소된 메모리로 구현 가능함을 확인하였고, R2MDC FFT 프로세서에 비해 약 16.4% 감소된 게이트 수와 약 26.8% 감소된 메모리로 구현 가능함을 확인할 수 있었다. FFT 프로세서는 MIMO-OFDM 기반 SDR 시스템에서 가장 큰 블록 중 하나이므로 제안된 scalable FFT/IFFT 프로세서는 SDR 시스템의 하

드웨어 복잡도를 줄이는데 큰 공헌을 할 수 있을 것으로 예상된다.

## 참 고 문 헌

- [1] N. Weste and D. J. Skllem, "VLSI for OFDM," *IEEE Commun. Mag.*, vol. 36, no. 10, pp. 127-131, Oct. 1998.
- [2] R. van Nee and R. Prasad, "OFDM for wireless multimedia communications," Boston: Artech House, 2000.
- [3] M. D. Batariere, J. F. Kepler, T. P. Krauss, S. Mukthavaram, J. W. Porter, and F. W. Vook, "An experimental OFDM system for broadband mobile communications," *Proc. IEEE Veh. Technol. Conf.*, vol. 4, pp. 1947-1951, 2001.
- [4] 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 Commun. Mag.*, vol. 40, no. 9, pp. 143-149, Sept. 2002.
- [5] A. van Zelst, Tim C. W. Schenk, "Implementation of a MIMO OFDM-Based wireless LAN system," *IEEE Trans. on Signal Processing*, vol. 52, no. 2, pp. 483-494, Feb. 2004.
- [6] 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*, vol. 92, no. 2, pp. 271-297, Feb. 2004.
- [7] H. Bolcskei and E. Zurich, "MIMO-OFDM wireless systems: basics, perspectives, and challenges," *IEEE Trans. Wireless Commun.*, vol. 13, no. 4, pp. 31-37, Aug. 2006.
- [8] Y. W. Lin, and C. Y. Lee, "Design of an FFT/IFFT processor for MIMO OFDM systems," *IEEE Trans. on Circuits and Systems*, vol. 54, no. 4, pp. 807-815, Apr. 2007.
- [9] 리우 항, 이한우, "MIMO-OFDM 시스템을 위한 고속 저면적 128/64-point radix-24 FFT 프로세서 설계," *전자공학회논문지 제 46권 SD편 제 2호*, pp. 15-23, 2009년, 2월.
- [10] J. Mitola, "The Software Radio Architecture," *IEEE Commun. Mag.*, vol. 33, no. 5, pp. 26-38, May 1995.
- [11] C. K. Rauwerda, P. M. Heyters, and C. J. M. Smit, "Towards software defined radios using coarse-grained reconfigurable hardware," *IEEE Trans. on VLSI Systems*, vol. 16, no. 1, pp. 3-13, Jan. 2008.
- [12] T. Shono, Y. Shirato, H. Shiba, K. Uehara, K. Araki, and M. Umehira, "IEEE 802.11 wireless LAN implemented on software defined radio with hybrid programmable architecture," *IEEE Trans. on Wireless Commun.*, vol. 4, no. 5, pp. 2299-2308, Sep. 2005.
- [13] IEEE Std. 802.11n, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications," 2005.
- [14] IEEE Std. 802.16e, "Local and Metropolitan Area Networks-Part 16: Air Interface for Fixed Broad-band Wireless Access System", Oct. 2004.
- [15] 최원철, 전형구, 이현, 오현서, "재구성 가능한 가변 포인트 IFFT/FFT 프로세서 설계에 관한 연구," *전자공학회논문지 제 41권 TC편 제 12호*, pp. 61-68, 2004년, 12월.
- [16] Petrus, P. et al. "An integrated draft 802.11n compliant MIMO baseband and MAC processor," *Proc. of ISSCC'07*, pp. 266-268, Feb. 2007.
- [17] S. He and M. Torkelson, "A New Approach to Pipeline FFT Processor," *Parallel Processing Symposium*, pp. 766-770, Apr. 1996.
- [18] S. He and M. Torkelson, "Designing pipeline FFT processor for OFDM (de)modulation," *Proc. IEEE URSI Int. Signals, Systems, and Electron.*, vol. 2, PP. 257-262, Oct. 1998.
- [19] L.R. Rabiner and B. Gold, "Theory and Application of Digital Signal Processing," Prentice-Hall, 1975.
- [20] 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.
- [21] Y. Jung, H. Yoon, and J. Kim, "New efficient FFT algorithm and pipeline implementation results for OFDM/DMT applications," *IEEE Trans. Consum. Electron.*, vol. 49, no. 1, pp. 14-20, Feb. 2003.
- [22] S. Lee, Y. Jung, and J. Kim, "Low complexity pipeline FFT processor for MIMO-OFDM systems," *IEICE Electronics Express*, vol. 4, no. 23, pp. 750-754, Dec. 2007.
- [23] G. Bi and E. V. Jones, "A pipelined FFT processor for word-sequential data," *IEEE Trans. on Acoust., Speech, and Signal Processing*, vol. 37, no. 12, pp. 1982-1985, Dec. 1989.

저 자 소 개



양 기 정(학생회원)  
2009년 한국항공대학교 항공전자  
및 정보통신공학부 학사  
졸업.  
2009년~현재 한국항공대학교  
항공전자 및 전자공학과  
석사 과정

<주관심분야 : 무선 통신 시스템용 모뎀 프로세서의 알고리즘 및 SoC 설계>



정 윤 호(평생회원)  
1998년 연세대 전자공학과 학사  
졸업.  
2000년 연세대 전기컴퓨터공학과  
석사 졸업.  
2005년 연세대 전기전자공학과  
박사 졸업.

2005년~2007년 삼성전자 통신연구소 책임연구원  
2007년~2008년 연세대학교 전자공학과 연구교수  
2008년~현재 한국항공대학교 항공전자 및 정보  
통신공학부 조교수

<주관심분야 : 무선 통신 시스템용 모뎀 및 영상  
처리 프로세서의 알고리즘 및 SoC 설계>