

논문 2017-54-7-4

저면적 Mixed-radix MDC FFT 프로세서를 위한 효율적인 스케줄링 기법

(Efficient Scheduling Schemes for Low-Area Mixed-radix MDC FFT Processor)

장 정 근*, 선우 명 훈**

(Jeong Keun Jang and Myung Hoon Sunwoo[©])

요 약

본 논문에서는 고속 데이터 전송을 위해 orthogonal frequency division multiplexing (OFDM) 시스템에 적용 가능한 고속 fast Fourier transform (FFT) 프로세서를 제안하였다. 제안하는 FFT 프로세서는 높은 처리율을 만족하기 위해 mixed-radix 알고리즘과 8개의 병렬 경로를 가지는 multipath delay commutator (MDC) 파이프라인 구조를 채택하였다. 하드웨어 복잡도를 줄이기 위해서 새로운 스케줄링 기법들을 적용하여 twiddle factor 연산을 위한 read-only memories (ROM)의 크기를 줄이는 구조와 복소 상수 곱셈기의 수를 줄이는 구조를 제안한다. 제안하는 구조는 지연 소자와 연산 사이클의 증가 없이 하드웨어 복잡도를 줄일 수 있다. 또한, IEEE 802.11 ac/ad와 같은 고속 OFDM 시스템을 위해 64/128/256/512-포인트 FFT 연산이 가능하다. 제안하는 FFT 프로세서는 Verilog-HDL로 모델링하여 Samsung 65nm 공정 라이브러리로 합성하여 0.36mm²의 면적과 330MHz의 동작 주파수에서 2.64 GSample/s를 보이고 있다.

Abstract

This paper presents a high-throughput area-efficient mixed-radix fast Fourier transform (FFT) processor using the efficient scheduling schemes. The proposed FFT processor can support 64, 128, 256, and 512-point FFTs for orthogonal frequency division multiplexing (OFDM) systems, and can achieve a high throughput using mixed-radix algorithm and eight-parallel multipath delay commutator (MDC) architecture. This paper proposes new scheduling schemes to reduce the size of read-only memories (ROMs) and complex constant multipliers without increasing delay elements and computation cycles; thus, reducing the hardware complexity further. The proposed mixed-radix MDC FFT processor is designed and implemented using the Samsung 65nm complementary metal-oxide semiconductor (CMOS) technology. The experimental result shows that the area of the proposed FFT processor is 0.36 mm². Furthermore, the proposed processor can achieve high throughput rates of up to 2.64 GSample/s at 330 MHz.

Keywords : FFT, Mixed-radix algorithm, MDC processor, IEEE 802.11.ac/ad, OFDM system

* 정회원, 티맥스, ** 정회원, 아주대학교 전자공학과
(Department of Electrical and Computer Engineering,
Ajou University)

© Corresponding Author (E-mail : sunwoo@ajou.ac.kr)

※ 본 연구는 한국연구재단 특별협력사업(한-스웨덴 연구교류 사업)으로 지원된 연구 결과입니다. (2016 K2A9A2A12003787),

※ 본 연구는 미래창조과학부 및 정보통신기술진흥센터의 대학ICT연구센터 육성지원사업 (IITP-2012016-0-00309) 결과로 수행되었음.

Received ; December 15, 2016 Revised ; May 24, 2017

Accepted ; June 16, 2017

I. 서 론

FFT (Fast Fourier Transform) 알고리즘은 1965년 Cooley와 Tukey에 의해 DFT (Discrete Fourier Transform) 연산에서 수행할 계산의 양을 실질적으로 줄이는 알고리즘으로 개발되었다. 이 알고리즘으로 인하여 계산량은 DFT 연산량 $O(N^2)$ 에서 $O(N \log N)$ 으로 줄어든다. 이러한 FFT 알고리즘은 통신 시스템, 진동 및 소음 계측 시스템 및 의료 시스템 등 다양한 분야에서 사용된다.

FFT 프로세서는 OFDM (Orthogonal Frequency Division Multiplexing) 전송 방식을 채택한 IEEE 802.11 WiFi^[1], IEEE 802.15.3 WPAN (Wireless Personal Area Network)^[2], IEEE 802.16 WMAN (Wireless Metropolitan Area Network)^[3] 등의 표준에서 가장 큰 복잡도를 가지며 고속 연산이 요구되어 하드웨어로 구현함에 있어 가장 중요한 모듈 중 하나이다. 특히, 주파수나 사용 환경, 전송속도 등 많은 요구조건에 맞추어서 시스템을 구현해야 하므로 시스템이 가지고 있는 특성을 잘 파악하고 하드웨어 자원을 잘 활용하여 성능을 최적화해야 한다. 이러한 OFDM 전송방식에서 요구하는 높은 처리율을 만족하며 실시간 신호 처리를 위해 다양한 FFT 프로세서^[4~8] 들이 활발히 연구되고 있다.

FFT 구조는 크게 메모리 기반의 구조와 파이프라인 구조로 나뉜다. 메모리 기반의 구조^[4] 는 적은 하드웨어 크기를 만족하기 위해 제안되었다. 그러나 이 구조들은 많은 연산 사이클을 요구하기 때문에 높은 처리 속도를 얻는데 어려움을 가지고 있어 처리 속도를 높이기 위해서는 높은 동작 주파수를 사용하여야 한다. 때문에 고속 동작을 요구하는 분야에서는 파이프라인 구조가 주로 사용된다. 파이프라인 구조^[5~8] 는 데이터 교환 방식에 따라 SDF (Single-path Delay Feedback), MDF (Multi-path Delay Feedback), SDC (Single-path Delay Commutator), MDC (Multi-path Delay Commutator) 등으로 분류할 수 있다. SDF 구조는 다른 파이프라인 구조에 비해 낮은 하드웨어 복잡도를 보이지만, 단일 경로를 통해 데이터를 전달하여 처리율이 낮다. 이에 반해 MDC 구조는 다른 구조에 비해 높은 하드웨어 복잡도를 보이지만 다중 경로를 통해 데이터를 전달하기 때문에 높은 처리율을 가진다. 각 구조에 따라 전체 구조의 하드웨어 복잡도와 데이터 처리율이 결정되기 때문에 시스템에서 요구되는 데이터 처리율과 하드웨어 복잡도를 고려하여 적합한 구조와 병렬 구조의 수를 결정하여야 한다.

본 논문에서는 8개의 경로를 가지지는 MDC 파이프라인 구조를 채택하여 높은 데이터 처리율을 만족시켰다. 또한, 제안하는 데이터 스케줄링 기법을 통해 지연소자와 연산 사이클이 증가 없이 ROM (Read-only Memories) 의 크기 및 복소 상수 곱셈기의 수를 감소시키는 구조를 제안한다. 제안하는 프로세서는 64/128/256/512-포인트 FFT 연산을 지원하기 때문에 IEEE 802.11ac/ad 표준을 사용하는 시스템에 적합하다.

본 논문의 구성은 다음과 같다. II장에서는 기존의

FFT 구조에 대하여 설명하고, III장에서는 제안하는 FFT 프로세서의 알고리즘, 구조 및 제안하는 스케줄링 기법에 대해 기술한다. IV장에서는 제안하는 FFT 프로세서의 구현 및 성능 평가를 기술하여, 마지막으로 V장에서 결론을 맺는다.

II. 기존 FFT 프로세서

기존 FFT 구조는 하드웨어 면적이 작으면서 상대적으로 데이터 처리율이 낮은 메모리 기반의 구조와 하드웨어 면적이 크지만 처리율이 높은 파이프라인 구조로 크게 두 가지의 구조로 나뉜다. FFT 연산 구조 특성상 포인트 수가 증가하면 하드웨어 크기와 소요시간 또한 증가한다. 따라서 제한되어 있는 자원을 고려하여 적용하고자하는 시스템에 요구되는 포인트 길이, 데이터 처리율, 하드웨어 복잡도 등에 적합한 구조를 채택하여야 한다.

메모리 기반의 구조는 단일 메모리, 이중 메모리 구조가 대표적이다. 단일 메모리 구조는 메모리로부터 데이터를 읽어 버터플라이 연산 후 다시 메모리에 연산 결과를 저장하는 방식이며, 이중 메모리 구조는 하나의 메모리에서 데이터를 읽어 버터플라이 연산 후 나머지 하나의 메모리에 연산 결과를 저장하는 방식을 취한다. 두 가지의 메모리 구조는 파이프라인 구조에 비해 낮은 하드웨어 복잡도를 갖지만 데이터 처리율을 낮다.

파이프라인 구조는 크게 SDF, MDF, SDC, MDC 구조로 나뉜다. 이 중 SDF 구조는 버터플라이 연산 후 일부의 결과를 시프터 레지스터에 저장하고 나머지 결과를 다음 스테이지에 전달하여 레지스터를 효율적으로 사용하는 구조이다. 이를 통해 하드웨어 복잡도를 감소시킬 수 있으나 단일 경로를 통해 데이터가 전달됨으로 처리율이 낮고 버터플라이 연산부의 제어가 비교적 복잡하다. MDC 구조는 SDF, SDC 구조와 달리 여러 경로를 통해 데이터를 전달하므로 데이터 처리율이 높고 제어가 간단하지만 교환기 (Commutator)를 사용하여 데이터 교환이 이루어지기 때문에 레지스터가 추가로 사용되어 하드웨어 복잡도를 증가 시킨다.

최근 수 GSample/s의 데이터 처리율을 만족시키기 위해 파이프라인 구조인 SDF 구조를 병렬 처리하여 높은 데이터 처리율을 만족시키는 MDF 구조가 제안되었다^[5~6]. 유사한 방식으로 데이터 처리율이 높은 파이프라인 구조인 MDC 구조에 병렬 처리 기법을 적용하여 더 높은 데이터 처리율을 가지는 구조^[7~8] 역시 제안되

의미하며, 이로 인해 복소 곱셈기는 FFT 프로세서의 하드웨어 복잡도를 증가시키는 가장 큰 요인이 된다. 이에 제안하는 구조는 스테이지 2와 3에서 각각 데이터 스케줄링을 통해 데이터를 재구성하여 상위 4개의 경로에 복소 곱셈을 위한 ROM의 크기를 감소시킬 수 있었으며, 추가적으로 복소 상수 곱셈기의 수를 줄여 하드웨어 복잡도를 감소시킬 수 있었다. 그림 2는 FFT 수에 따른 스테이지 2에서 필요한 twiddle factor의 순서를 나타내며, stage 2에서 짝수 순서는 위로 올려서 W_{256} 복소 곱셈기로 사용하고, 홀수 순서는 아래로 내려서 W_{512} 복소 곱셈기로 연산을 한다. 따라서 W_{512} 를 필요한 stage 2에서 스케줄링 방법을 적용하여 W_{256} 으로 낮추어서 하드웨어 면적이 줄었다.

그림 3는 데이터 스케줄링을 위한 스테이지 2와 3의 제안하는 교환기 구조를 나타낸 것으로 지연 소자와 멀티플렉서, 동작 모드 신호로 구성된다. 교환기는 각 동작모드 신호에 따라 4-to-1 멀티플렉서에서 4개의 입력 중 하나의 신호를 선택하여 출력하게 되고 이에 따라 데이터들이 재정렬된다.

스테이지 2에서 twiddle factor W_{512} 연산을 요구한다. 따라서 twiddle factor W_{512} 연산을 위해 각 경로마다 복소 곱셈기가 필요로 하여 twiddle factor W_{512} 의 계수를 저장하는 8개의 ROM을 요구된다. 이에 반해 제안하는 구조는 스테이지 2에서 교환기를 통해 데이터 스케줄링을 하여 상위 4개 경로의 ROM 크기를 줄일 수 있다. 스테이지 2에서의 교환기 패턴은 그림 2(a)와 같으며, 동작모드 신호는 식 (1)을 따라 변한다.

$$\text{동작모드1} = T \times 32/N, \quad T = 0, 1, \dots, N/8 - 1 \quad (1)$$

제안하는 데이터 스케줄링을 통해 재정렬된 상위 4개의 경로의 데이터는 twiddle factor W_{512} 연산을 위한 계수가 W_{512}^{2nk} 로 짝수인 계수를 가진다. 이는 수학적 계산에 의해 $W_{512}^{2nk} = W_{512/2}^{2nk/2} = W_{256}^{nk}$ 가 되어 상위 4개의 경로에서 twiddle factor W_{256} 으로 대체 할 수 있다. 즉, 상위 4개의 경로에서 twiddle factor W_{256} 연산을 하게 되고, 하위 4개의 경로에서 twiddle factor W_{512} 연산을 한다. 따라서 8개의 복소 곱셈기에서 twiddle factor W_{512} 연산을 위해 ROM 하나당 128개의 계수를 저장하여 총 1024개의 계수를 저장해야 되지만 제안하는 구조처럼 4개의 복소 곱셈기를 twiddle factor W_{256} 연산을 하게 대체한다면 4개의 ROM에는 64개의 계수씩 256 (64×4)개의 계

수, 다른 4개의 ROM에는 128개의 계수씩 512 (128×4)개의 계수를 저장해야 함으로 총 768개의 계수를 저장해야 한다. 결론적으로 ROM에 저장해야 하는 twiddle factor 계수를 25% 감소시켜 제안하는 구조의 하드웨어

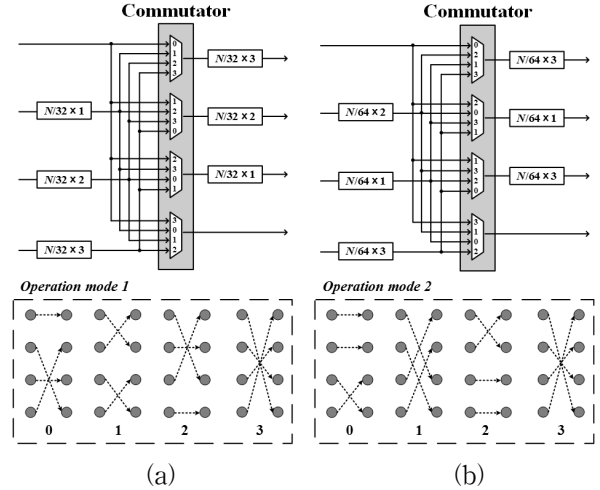


그림 3. 스테이지 2와 3의 제안하는 교환기의 구조 (a) 스테이지 2에서의 교환기 패턴 (b) 스테이지 3에서의 교환기 패턴

Fig. 3. Proposed commutator architecture of Stage 2 and 3. (a) Commutator pattern of Stage 2. (b) Commutator pattern of Stage 3.

복잡도를 감소시킬 수 있다.

스테이지 2에서 연산된 데이터를 스테이지 3에서 교환기를 통해 다시 스케줄링 시켜 스테이지 3, 4, 그리고 5에 필요한 복소 상수 곱셈기의 수를 줄일 수 있다. 제안하는 교환기 패턴은 그림 2(b)와 같으며 동작모드 신호는 다음 식 (2)을 따라 변한다.

$$\text{동작모드2} = T \times 128/N, \quad T = 0, 1, \dots, N/8 - 1 \quad (2)$$

스테이지 3는 64-포인트 FFT 연산함에 있어 사용되지 않는 포인트이기 때문에 N 은 64를 제외한 $128/256/512$ 의 값을 갖는다. 복소 상수 곱셈기의 감소를 설명하기 위해 그림 3은 스테이지 2 이후의 기존 구조^[9]와 제안하는 구조를 나타낸다. 기존 구조와 제안하는 구조 모두 스테이지 3에서 $-j$ 연산이 필요하고, 스테이지 4에서 twiddle factor W_8 연산이 요구되며, 스테이지 5에서 twiddle factor W_{16} 연산이 필요하다.

그림 3(a)에 보이듯이 기존 구조는 스테이지 3에서 $-j$ 연산을 위해 경로 2, 4, 6, 그리고 8에 복소 상수 곱셈기가 필요하다. 하지만 그림 3(b)의 제안하는 구조는 데이터 스케줄링을 통해 데이터를 재정렬 하여 경로 4

와 8에만 복소 상수 곱셈기가 필요하게 된다. 또한, 기존 구조의 경우 스테이지 4에서 twiddle factor W8 연산을 위해 8개의 경로 모두에 복소 상수 곱셈기가 요구된다. 반면 제안하는 구조는 지속적으로 데이터를 스케줄링 하여 twiddle factor W8을 위한 복소 상수 곱셈기를

덧셈 연산을 이용하여 구현하는 방법으로 공통의 패턴을 정의하여 서로 공유한다. CSS 기법은 CSD 구현함에 있어 중복되는 연산을 미리 연산 처리하여 연산량과 하드웨어 복잡도를 줄일 수 있다.

IV. 실험

제안한 FFT 프로세서는 Verilog-HDL 기반의 하드웨어로 구현하였고, Synopsys Design Compiler를 통해 Samsung 65nm 공정 라이브러리로 논리합성을 하였다. 다른 구조에 비해 높은 데이터 처리율을 갖는 MDC 구조에 8개의 경로를 가지도록 병렬 처리 기법을 적용하여 고속 동작이 가능한 구조를 설계하였으며 64/128/256/512-포인트 FFT 연산이 모두 가능하다. 내부 워드 길이는 실수부와 허수부를 각각 12비트로 같은 입력에 대한 제안한 FFT 프로세서의 시뮬레이션 결과를 MATLAB 결과와 비교하면서 동작 검증을 하였다. SQNR(Signal to Quantization Noise Ratio)은 64/128-포인트 연산의 경우 36dB, 256/512-포인트 연산의 경우 35dB 성능을 보였다.

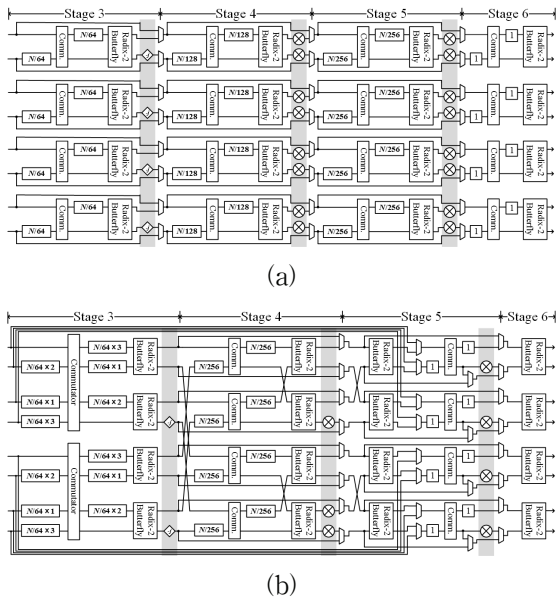


그림 4. FFT 프로세서의 스테이지 3, 4, 5, 그리고 6 구조 (a) 기존 FFT 구조 (b) 제안하는 FFT 구조
Fig. 4. Structures of Stages 3, 4, 5, and 6 of FFT processor. (a) Existing FFT architecture. (b) Proposed FFT architecture.

3개로 줄일 수 있다. 마지막으로 기존 구조는 스테이지 5에서 8개의 복소 상수 곱셈기가 twiddle factor W16 연산을 위해 필요로 하지만 제안하는 구조는 교환기와 복소 상수 곱셈기의 위치를 바꿈으로서 4개의 복소 상수 곱셈기만을 가지게 된다. 즉, 기존 구조의 경우 스테이지 3, 4, 그리고 5에 요구되는 복소 상수 곱셈기는 총 20개인 반면 제안하는 구조는 총 9개의 복소 상수 곱셈기를 요구한다.

따라서 제안하는 구조는 데이터 스케줄링을 통해 지연 소자와 연산 사이클의 증가 없이 기존 구조^[9]보다 55%의 복소 상수 곱셈기 수를 감소시켜 하드웨어 복잡도를 줄일 수 있다. 또한, 복소 상수 곱셈기의 사용률을 50%에서 94.4%로 증가시킬 수 있었다.

하드웨어 복잡도를 더욱 감소시키기 위해서 모든 복소 상수 곱셈기는 CSD(Canonical Signed Digit)와 CSS(Common Sub-expression Sharing) 기법을 사용하여 구현하였다. CSD 기법은 곱셈 연산을 시프트와

기존의 FFT 구조와의 공정한 비교를 위해 병렬 수가 8개이며, 512-포인트 FFT 연산이 가능한 프로세서로 선정하였다. 표 1은 제안하는 구조와 기존의 FFT 구조^[5~7]과 성능 비교를 나타낸다. 제안하는 구조는 330MHz의 주파수에서 2.64 GSample/s의 처리율을 가진다. 이 처리율은 기존 논문^[5~6]과 비교하였을 때 가장 높다. 일반화 면적을 비교하였을 때도 기존 논문^[5~7]보다 작은 하드웨어 면적을 가지며, 특히^[7] 논문의 약 41%의 하드웨어 크기를 가진다는 것을 알 수 있다.

각각의 논문과 자세히 비교하면 다음과 같다. 논문^[5]의 경우 동일한 SQNR을 갖지만 데이터 처리율이 높으며, 일반화 면적이 36% 더 작다. 논문^[6]과 비교하면 제안하는 논문이 데이터 처리율은 1.5배 높으며, 일반화 면적에서도 34% 더 작음을 보인다. 같은 MDC 구조를 가진 논문^[7]은 동일한 데이터 처리율을 보이지만 제안하는 구조가 일반화 면적에서 59% 감소를 보인다. 특히, 기존의 구조^[5~7]의 경우 512-포인트 FFT 연산만을 지원하지만 제안하는 구조는 64/12/256/512-포인트 FFT 연산이 모두 가능하다. 일반화 면적은 면적/(공정/65nm)²/병렬 수/log2(포인트 수)로 나타낸다.

V. 결 론

본 논문에서는 초고속 데이터 전송을 위한 OFDM 시스템에 적용 가능한 고속 FFT 프로세서를 제안하였다. 제안하는 FFT 프로세서는 64/128/256/512-포인트 모두 연산이 가능하며, 높은 데이터 처리율을 만족하기 위하여 파이프라인 구조와 병렬 처리 기법을 사용하였다. 특히, 기존 구조 중 데이터 처리율이 높은 MDC 구조에 radix-8와 radix-4 알고리즘을 적용하여 높은 처리율을 만족 시키면서 연산 사이클을 줄였으며, radix-2 알고리즘을 기반으로 스테이지를 구성해 다수의 포인트 길이를 지원하도록 하였다. 또한, 하드웨어 복잡도의 감소를 위해 새로운 스케줄링 기법을 제안하였다. 이는 복소 곱셈을 위한 ROM의 크기를 줄일 수 있으며, 제안하는 새로운 구조를 적용하여 지연 소자와 연산 사이클의 증가 없이 복소 상수 곱셈기 및 연산 복잡도를 감소시킬 수 있다.

설계한 FFT 프로세서는 Verilog-HDL로 코딩하였고, Synopsys Design Compiler를 통해 Samsung 65nm 공정 라이브러리로 합성하였다. 메모리를 제외한 전체 게이트 수가 약 163,000개이며, 동작속도는 330MHz로 2.64 Gs/s의 데이터 처리율을 보인다. 제안하는 고속 FFT 프로세서는 802.11ac/ad와 같은 다수의 포인트 길이를 지원하는 OFDM 변복조 방식의 통신 시스템 개발에 활용될 수 있다.

REFERENCES

[1] Anonymous, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," in Proc. IEEE Advanced Communication Technology (ICACT), Feb. 2012, pp. 804-808, Part 15.3.

[2] Anonymous "Wireless MAC and PHY Specifications for High Rate WPANs," IEEE Std 802.15.3-2003, Local and metropolitan area networks Part 15.3.

[3] IEEE 802.16 Working Group, "IEEE Standard for Local and Metropolitan Area Networks. Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems. Amendment 3: Management Plane Procedures and Services," IEEE Standard, vol. 802, 2006.

[4] S. Huang and S. Chen, "A high-throughput Radix-16 FFT processor with parallel and normal input/output ordering for IEEE 802.15. 3c

Systems," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 59, no. 8, pp. 1752-1765, Oct. 2012.

[5] T. Cho and H. Lee, "A High-speed low-complexity modified Radix-25 FFT processor for high rate WPAN applications," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 21, no. 1, pp. 187-191, Feb. 2013.

[6] C. Wang, Y. Yan, X. Fu, "A High-Throughput Low-Complexity Radix-24-2²-2³ FFT/IFFT Processor with Parallel and Normal Input/Output Order for IEEE 802.11ad Systems," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., 2015. (online published)

[7] T. Ahmed, M. Garrido, and O. Gustafsson, "A 512-point 8-parallel pipelined feedforward FFT for WPAN," in Proc. ASILOMAR, Nov. 2011, pp. 981-984.

[8] K. Yang, S. Tsai and G.C. Chuang, "MDC FFT/IFFT processor with variable length for MIMO-OFDM systems," IEEE Trans. Very Large Scale Integration (VLSI) Syst., vol. 21, no. 4, pp. 720-731, Apr. 2013.

[9] M. Garrido, J. Grajal, M. Sánchez, and O. Gustafsson, "Pipelined radix-2k feedforward FFT architectures," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 21, no. 1, pp. 23-32, Jan. 2013.

[10] F. Qureshi, S.A. Alam and O. Gustafsson, "4K-Point FFT Algorithms based on optimized twiddle factor multiplication for FPGAs," in proc. IEEE Asia Pacific postgraduate research microelectron, 2010, pp. 225-228.

저 자 소 개



장 정 근(정회원)
2008년~2015년 아주대학교 전자공
학부 학사
2015년~2017년 아주대학교 전자공
학부 석사
2017년~현재 티맥스 OS 선임연
구원

<주관심분야: SoC 설계, 신호처리, 통신알고리즘>



선우 명 훈(평생회원)
1980년 서강대학교 전자공학 학사
졸업.
1982년 한국과학기술원 전자공학 석
사 졸업.
1982년~1985년 한국전자통신연구
소(ETRI) 연구원

1985년~1990년 Univ. of Texas at Austin 전자공
학 박사.

1990년~1992년 Motorola, DSP Chip Division (미국)

1992년~1996년 아주대학교 전기전자공학부 조교수

1996년~2001년 아주대학교 전자공학부 부교수

2001년~현재 아주대학교 전자공학부 정교수

2011년~현재 IEEE Fellow Member

2011년~2016년 IEEE CASS Board of Governor

2012년~2014년 대한전자공학회 반도체 소사이어
티 회장

2014년~현재 대한전자공학회 반도체 소사이어티
자문위원

2016년~현재 아주대학교 ITRC uDSD 연구센터장

<주관심분야: ICT SoC 설계, ASIP 설계, 저전력
의료기기 디바이스 설계>