

論文 2002-39SD-11-8

## 고속 FFT 연산을 위한 새로운 DSP 명령어 및 하드웨어 구조 설계

## (Design of New DSP Instructions and Their Hardware Architecture for High-Speed FFT)

李在晟\*, 鮮于明勳\*\*

(Jae Sung Lee and Myung Hoon Sunwoo)

## 요 약

본 논문은 고속의 FFT 연산을 위한 DSP(Digital Signal Processor) 명령어와 그 하드웨어 구조를 제안한다. 제안된 명령어는 MAC 연산에 의존하는 기존의 DSP 칩과는 다른 새로운 연산 과정을 수행한다. 본 논문은 새로운 명령어의 원활한 수행을 위한 데이터 연산 유닛(Data Processing Unit : DPU)의 하드웨어 구조를 제안한다. 제안된 명령어 및 하드웨어 구조는 기존의 DSP 칩과 비교하여 FFT 연산 속도가 2 배 향상되었다. 제안된 구조는 Verilog HDL을 사용하여 설계되었으며 0.35  $\mu\text{m}$  표준 셀 라이브러리를 사용하여 수행되었다. 분석 결과 최대 동작 주파수는 약 144.5 MHz이다.

## Abstract

This paper presents new DSP (Digital Signal Processor) instructions and their hardware architecture for high-speed FFT. The instructions perform new operation flows, which are different from the MAC (Multiply and Accumulate) operation on which existing DSP chips heavily depend. The proposed DPU (Data Processing Unit) supporting the instructions shows two times faster than existing DSP chips for FFT. The architecture has been modeled by the Verilog HDL and logic synthesis has been performed using the 0.35  $\mu\text{m}$  standard cell library. The maximum operating clock frequency is about 144.5 MHz.

**Key Words** : FFT, DSP, 명령어, 하드웨어 구조, arithmetic unit.

## 1. 서 론

현재의 통신기술 동향은 반도체기술의 급격한 발달

\* 正會員, 韓國電子通信研究院  
(ETRI)

\*\* 正會員, 亞州大學校 電子工學部

(School of Electrical and Computer Eng., Ajou Univ.)

※ 본 연구는 과학기술부에서 시행하는 국가지정연구  
실사업(National Research Laboratory)과 IDEC(IC  
Design Education Center)의 부분적인 지원을 받아  
수행되었습니다.

接受日字: 2001年12月7日, 수정완료일: 2002年10月28日

과 더불어 유선으로는 VDSL (Very-high-data-rate Digital Subscriber Line) 모뎀, 케이블모뎀, 전력선모뎀 등의 개발이 활발하게 진행되고 있고, 무선으로는 WCDMA(Wideband Code Division Multiple Access)와 CDMA2000을 바탕으로 하여 IMT-2000(International Mobile Telecommunications -2000)이라는 3세대 이동통신서비스가 등장하고 있는 등 통신 분야의 급속한 발전이 거듭되고 있다. 통신 시스템의 변복조 방식 및 알고리즘이 한 번 결정되고 검증되면 비용, 면적, 전력 소모 등을 최소화하기 위해 시스템을 ASIC(Application-Specific IC) 칩으로 구현한다. 그러나, 다양한 통신 표준에 대한 유연성(flexibility) 부재, 하드웨어 비용 및 개발비용 상승, 개발 기간 장기화 등이 문제화되면서

통신 시스템에서 ASIC 사용에 한계를 나타내고 있다.

DSP(Digital Signal Processor)는 설계자들이 프로그램을 작성하거나 수정하는 것에 의해 쉽고 빠르게 새로운 알고리즘을 구현 가능하도록 하는 플랫폼 역할을 한다. 또한, 공정 기술이 발달함에 따라 DSP로도 ASIC의 장점을 유지하는 것이 가능하게 되었다. DSP를 이용한 구현은 신속한 시장 대응이 가능하며 빠른 변경과 업그레이드, 시스템 변경 후의 기존 시스템과의 호환성(backward compatibility) 등의 장점을 갖는다. 오늘날, 공정 기술이 발달하고 신속한 업그레이드 요구가 증가함에 따라 DSP 기반의 통신용 시스템 개발로 그 구현 방식이 전환되고 있다.<sup>[1]</sup>

그러나, DSP 역시 초고속 광대역 통신 알고리즘이 요구하는 방대한 연산량에 비해 아직까지 그 성능이 부족한 실정이다. 특히, 좁은 주파수 대역을 활용하여 고속 데이터 전송을 위하여 사용되는 OFDM(Orthogonal Frequency Division Multiplexing), DMT(Discrete Multi-Tone)<sup>[2]</sup> 등의 통신 방식은 연산점이 큰 복소수 FFT(Fast Fourier Transform) 연산의 고속 수행을 필요로 하는데, 지금까지 DSP 기술로는 어려운 실정이다.<sup>[3-5]</sup> 따라서, 본 논문에서는 차세대 고속 유무선 통신 기술의 핵심 알고리즘인 고속 FFT 연산을 프로그램 래머블 DSP 칩에서도 수행 가능하도록 하는 명령어 및 하드웨어 구조를 제안한다.

제안하는 명령어 및 하드웨어 구조는 기존 DSP 칩의 MAC 연산 구조로는 수행할 수 없었던 연산 흐름을 지원할 수 있는 구조이다.<sup>[6,7]</sup> 기존 DSP의 연산 유닛 종류와 그 구조<sup>[8-17]</sup>를 최대한 보존하고, 최소의 데이터 패스 회로만을 추가하여 구현 가능하므로 기존 연산 모듈들의 재사용도 용이하게 할 수 있고, 칩 구현 시 오버 헤드도 최소화하여 하드웨어 부담을 줄일 수 있도록 하였다. 또한, 제안한 FFT 명령어의 하드웨어 구조는 Verilog HDL로 모델링한 후, HYUNDAI™ 0.35  $\mu\text{m}$  표준 셀 라이브러리를 사용하여 SYNOPSIS™ 캐드 툴로 논리합성을 수행하였으며, CADENCE™ Verilog-XL을 사용하여 타이밍 분석을 수행하였다. 논리합성 결과 레지스터간 데이터 패스 회로의 최대 지연 시간은 6.92 nsec 이며, 따라서 144.5 Mhz까지 동작할 수 있다. 제안한 구조를 수행 사이클 측면에서 기존의 DSP 칩<sup>[8-12]</sup>들과 비교할 경우 2배 빠른 수행 속도를 얻을 수 있었다.

본 논문은 다음과 같이 구성된다. 2장에서는 FFT 연

산을 위한 기존 DSP 칩들의 구현 방법 및 하드웨어 구조를 조사하고, 3장에서는 새로운 FFT용 명령어 및 하드웨어 구조 제안 및 기존 DSP들<sup>[8-15]</sup>과의 성능 비교를 수행한다. 마지막으로 4장에서 결론을 맺는다.

## II. 기존 DSP 상에서 FFT 알고리즘 구현

이 장에서는 DSP에서 사용되는 FFT 연산 알고리즘 및 연산 방식을 알아보고, FFT 연산을 위한 상용 DSP 칩<sup>[8-17]</sup>들의 하드웨어 구조를 분석한다.

### 1. DSP 칩의 복소수 FFT 연산 방식

일반적으로 FFT 연산 방식은 DFT(Discrete Fourier Transform) 수식을 여러 개의 순열의 합( $\Sigma$ )들로 분할할 때 그 분할된 항의 개수 즉, 기수(radix)에 기반하여 나뉘어진다. 이때 분할된 개수를  $i$ 라고 하면,  $i$  값이 커짐에 따라 수행 속도는 빨라질 수 있으나 훨씬 더 많은 산술 연산 유닛을 필요로 하며, 연산점수(the number of point)가  $i!$ ( $i$ 은 양수)이 아니면 FFT 연산의 마지막 단계 다른 기수에 기반한 FFT 연산을 추가하여야 한다. 따라서, 전용 FFT 프로세서가 아닌 DSP에서는 곱셈기 수가 2개 정도로 제한되어 있기 때문에 실제 FFT 구현 성능은 연산점이 증가할수록 radix-2가 오히려 우수한 성능을 보이고 있다. 실제로도 상용 DSP의 구조는 radix-2 FFT 연산에 적합한 구조를 지니고 있다.

식 (1)과 <그림 1>은 각각 radix-2 FFT 알고리즘의 연산 식과 radix-2 버터플라이 구조를 나타낸다.<sup>[18]</sup> 식 (1)은 DFT 수식을 2개의 순열합의 항으로 분할하여 표현함으로써 radix-2 FFT 수식을 유도한 것이다. <그림 1>에서처럼 radix-2 FFT는 버터플라이 연산 형태를 취하며, 이러한 구조의 연속적 반복으로 모든  $2^N$  포인트 FFT 연산이 가능하다. <그림 1>의  $m-1$ 은 이전 스테이지를 나타내고  $m$ 은 현재 스테이지를 나타낸다. 즉, 이전 스테이지의 두 값들이 가중치  $k$   $j2^m / N$  과 곱한 후 가산과 감산을 수행함으로써 현재 스테이지의 값을 나타내게 된다. 그림 상의 모든 연산은 복소수 연산이며, 그러한 복소수 연산을 일반 DSP 칩에서 어떤 순서로 수행하고 있는지 <그림 2>에 나타내고 있다.<sup>[19]</sup>

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

$$X[k] = \sum_{n=0}^{N/2-1} x[2n]W_{N/2}^{nk} + W_N^k \sum_{n=0}^{N/2-1} x[2n+1]W_{N/2}^{nk}$$

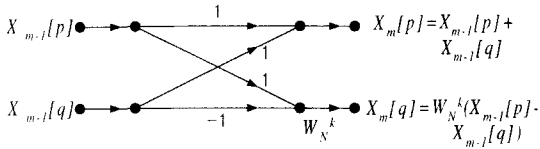


그림 1. Radix-2 FFT 버터플라이 연산 구조(DIF)  
Fig. 1. The radix-2 FFT butterfly(Decimation-In-Frequency).

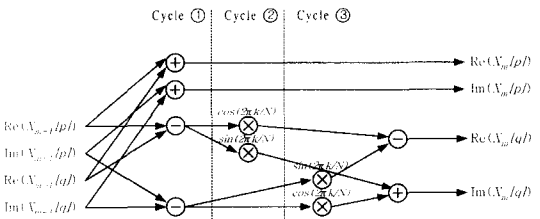


그림 2. 일반 DSP 칩에서의 radix-2 버터플라이 연산 순서도  
Fig. 2. The flowgraph of a radix-2 butterfly on general DSP chips.

1990년대 후반 이후의 DSP 칩들은 일반적으로 이중 MAC(Multiply and Accumulate) 연산 지원을 위한 하드웨어를 보유하고 있으며, ALU(Arithmetic Logic Unit)도 4 개 이상 보유하고 있는 DSP 칩들도 있다.<sup>[8~15]</sup> 그리고, MAC 구조는 곱셈기와 덧셈기 사이를 파이프라이닝하여 사용하고 있다. 따라서, <그림 2>와 같은 연산 흐름이 가능한 것이다. 그림에서 ①, ②, ③ 기호들은 1 번째, 2 번째, 3 번째 연산 사이클(각각 1 클럭 사이클 소모)을 나타낸다. 연산 유닛의 파이프라인 길이가 서로 다른 DSP 칩들인 경우라도 동작 주파수만 다를 뿐 연산 결과가 출력되는 측면에서 보면 한 개의 버터플라이 연산하는 데 걸리는 사이클 수는 3 이상으로 항상 같다.

연산 순서를 설명하면 다음과 같다. 먼저 ①의 사이클에서 4 개의 ALU들을 이용한 복소수 감가산 연산을 수행하며(1 클럭 사이클 소모), ②의 사이클에 두 개의 곱셈을 처리하고(1 클럭 사이클 소모), 그 결과와 ③의 MAC 연산을 취하기 위한 준비 사이클로 1 사이클이

소모되고 최종적으로 ③의 연산을 수행함(①과 동시에 동작하므로 사이클 소모가 없음)으로써 모두 3 사이클이 걸린다. 즉, 새로운 출력데이터가 나오는데 걸리는 잠복(Latency) 사이클이 3인 것이다. ①, ②, ③ 사이클의 연산이 연속적으로 이루어져 출력단에서는 1 사이클 당 하나의 버터플라이 연산 결과가 나오는 것이 아니다. 왜냐하면 곱셈기는 2 개로 한정되어 ②, ③을 동시 수행할 수 없고, MAC 기반의 연산은 누산기가 곱셈기와 ALU 밑에 있어 ③ 사이클의 연산을 준비하는 데 1 사이클 이상 소모된다. 동시에 사용할 수 없기 때문이다. 따라서, 항상 3 클럭 사이클마다 한번씩만 버터플라이 데이터들이 입력되는 것이다. <그림 2>의 ③에 나타난 바와 같이 상용 DSP 칩들은 곱셈기와 감가산기를 연속으로 배치하여 한 사이클에 이중 MAC 연산을 수행할 수 있도록 하고 있다.

2. FFT 연산을 위한 상용 DSP 칩의 구조

본 절에서는 90년대 후반 이후 발표된 최신 상용 DSP 칩들의 하드웨어 구조를 분석한다. 대상 DSP 칩들은 모두 통신용이며, FFT 알고리즘 연산 처리를 고려한 하드웨어 구조를 가지고 있다.

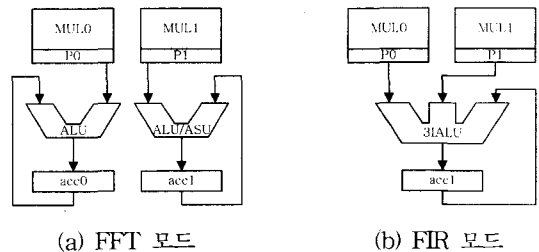


그림 3. PalmDSPCoreTM의 데이터 연산 유닛  
Fig. 3. The main computation unit of PalmDSPCoreTM.

<그림 3>은 DSP Group의 차세대 통신용 DSP 코어인 PalmDSPCoreTM의 데이터 연산부를 나타낸다[14]. PalmDSPCoreTM은 두 개의 곱셈기 외에 ALU, ASU (Adder/ Subtractor Unit), 3IALU(three input ALU) 등을 포함하여 7개의 산술 연산 유닛을 보유하고 있는 VLIW 아키텍처를 가지고 있다. <그림 3(a)>는 복소수 FFT 연산 모드 시 이중 MAC 기반의 복소수 연산을 위한 구조이며, <그림 3(b)>는 FIR(Finite Impulse Response) 필터 알고리즘 연산 모드 시 두 개의 곱셈단까지 한번에 MAC 연산을 취할 수 있도록 하는 구조이다. 두 구조는 적용 알고리즘에 따라 모드가 변경되

어 구성된다. 그림에서 P0, P1은 파이프라인 레지스터이며, acc0, acc1는 누산기를 나타낸다.

Philips 사의 차세대 통신용 DSP인 R.E.A.L (Reconfigurable Embedded DSP Architecture Low power Low cost) 코어와 Lucent Technologies 사의 DSP16000 코어 모두 <그림 3(a)>와 같은 형태의 이중 MAC 구조를 가지며 1 사이클 안에 이중 MAC을 처리할 수 있고 이러한 고속 연산 처리를 위하여 데이터의 흐름을 원활하게 하는 이중 하버드 아키텍처(dual Harvard architecture)를 가지고 있다. 이중 하버드 아키텍처는 메모리를 여러 개로 분할하여 각 메모리당 AGU(Address Generation Unit)들을 보유하여 다중 버스를 통해 데이터를 송수신하는 구조이다. 두 코어는 각각 125 Mhz, 100 Mhz까지 동작할 수 있다.<sup>[8,9]</sup>

Infineon Technologies의 CLIW™(Configurable Long Instruction Word) 구조를 갖는 CARMEL DSP 코어는 고도의 병렬 수행 능력을 보유함으로써 동시에 4개의 어드레스와 4 개의 산술연산 및 2 개의 데이터 이동을 수행할 수 있다. 그러나, CARMEL DSP도 이중 MAC에 기반한 2 사이클 복소수 연산 방식을 수행하기 위한 구조를 가지고 있다.<sup>[13]</sup>

Texas Instruments 사의 TMS320C6x는 곱셈기 2 개 (.M 유닛)와 ALU 6 개 (.L, .S, .D 유닛)를 병렬로 배열하여 동시 연산 가능하도록 하였으며, 동시에 레지스터 파일에 액세스 가능한 구조로 되어 있다.<sup>[10]</sup> TMS320C6x 역시 2 개의 곱셈기와 ALU들을 사용하여 이중 MAC의 구조를 지원하며, 1 사이클 수행이 가능하다.

Motorola Semiconductors와 Lucent Technologies 두 회사가 1999년 6월 공동 설립한 스타코어 기술센터(StarCore Technology Center)에서 3000 MIPS 또는 초당 12억 MAC의 처리성능을 갖는 세계 최초의 SC140 계열 DSP 코어를 개발하였다.

SC140의 DALU는 4개의 ALU와 8개의 쉬프터/리미터(shifters/limiter)로 이루어지며, 각 ALU는 MAC과 BFU(Bit-Field Unit)라는 연산 유닛을 가짐으로 총 8 개의 연산 유닛을 가지며, 따라서 총 16개의 연산 유닛을 보유한다.<sup>[13]</sup> 이처럼 최근 반도체 공정기술의 발달과 더불어 1 사이클 복소수 연산을 실현하기 위하여 4개의 곱셈기가 하나의 코어에 집적될 수 있게 된 것이다. 4 개의 ALU에 각각 곱셈기와 누산기가 있어 <그림 2>의 radix-2 버터플라이 연산 순서도상의 ②, ③의 부

분을 한 사이클에 처리 할 수 있다. SC140 코어는 무선통신 단말기나 개인휴대단말기(PDA) 등 차세대 절전형 메모리 절약형 애플리케이션 설계에 폭 넓게 활용할 목적으로 개발되었으며, 이미 Motorola의 MSC8101과 같은 고성능 DSP 칩에 핵심 코어로 사용되고 있다. 그러나, 하드웨어 크기만 확대되었을 뿐 다량 연산 유닛간의 데이터 패스 구조는 기존의 DSP에서와 크게 다를 바 없다.

앞서 살펴본 바와 같이 최근 DSP 칩들은 FFT 연산 시 복소수 연산을 빠르게 하기 위하여 이중 MAC을 지원하는 구조를 모두 갖추고 있다. 따라서, 거의 대부분의 DSP 칩들이 <그림 2>의 연산 흐름에 따라 FFT 연산을 수행하고 있으며 하드웨어 구조도 SC140 코어의 경우처럼 그 연산 흐름에 맞추어 확대되고 있다. 즉, FFT 알고리즘 연산의 흐름 자체를 분석하여 DSP 칩의 구조에 적용하는 사례는 아직까지 발표되고 있지 않은 실정이다.

### Ⅲ. 새로운 FFT용 명령어 및 하드웨어 구조 제안

본 장에서는 기존 방식과 다른 복소수 연산 구조를 가지는 FFT 연산 방식에 기반하여 새로운 명령어를 제안하며 그 명령어 처리를 위한 하드웨어 구조를 제안한다.<sup>[6,7]</sup> 제안한 구조의 검증을 위하여 Verilog HDL 모델링 및 논리 합성과 타이밍 시뮬레이션도 수행하였다. 제 1절에서 FFT 용 명령어 및 하드웨어 구조를 설계하고, 제 2절에서 시뮬레이션 결과를 이용하여 기존 DSP들과의 성능 비교를 수행한다.

#### 1. 제안하는 FFT 전용 명령어 및 하드웨어 구조 설계

제 2 장의 상용 DSP 칩들의 분석에서도 나타난 바와 같이 최근 발표되고 있는 고성능 DSP 칩들은 일반적으로 두 개의 곱셈기와 여러 개의 ALU를 보유하고 있어, 한 번에 2개의 MAC 연산을 수행할 수 있다. radix-2 버터플라이 연산 구조는 두 개의 복소수를 한 번은 더하고, 한번은 차에 복소수 계수를 곱하는 구조이다. 복소수 곱셈을 수행하는 부분을 2 개의 곱셈기로 한번에 처리 할 수 없었기 때문에 두 번에 나누어 처리되었다.

제안하는 새로운 FFT 연산 순서에서는 복소수 곱셈 부분에서 <그림 4>와 같은 연산 순서를 따라 복소수

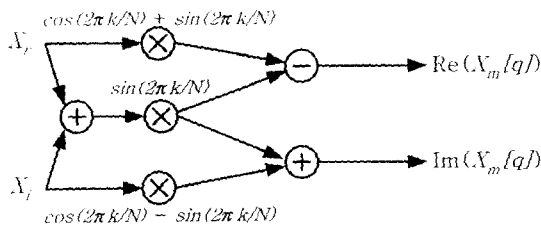


그림 4. 향상된 복소수 곱셈 구조

Fig. 4. The data flow of the enhanced complex multiplication.

곱셈 연산 부분을 처리하면 전체 FFT 연산을 2 배 빠르게 연산을 수행할 수 있다.

<그림 4>는 데이터의 실수부와 허수부가  $Re(X_m[q])=X_r[\cos(2\pi k/N)+\sin(2\pi k/N)] - (X_r+X_i)\sin(2\pi k/N)$ ,  $Im(X_m[q])=X_i[\cos(2\pi k/N)-\sin(2\pi k/N)] + (X_r+X_i)\sin(2\pi k/N)$ 로 표현될 수 있다는 점에서 착안한 것이다. 이러한 연산 구조는 기존 연산 구조에서 4번의 곱셈이 필요하던 것을 3개의 곱셈으로 줄여준다.<sup>[3)]</sup> <그림 5>는 이러한 복소수 곱셈 연산 구조를 사용하여 radix 2 FFT를 수행하는 순서를 나타내고 있

다. 즉, radix-2 FFT 버터플라이 연산 구조 두 개를 3 사이클 안에 수행하는 구조이며, 필요한 연산 유닛수는 기존 DSP가 보유하고 있는 2 개의 곱셈기와 5개의 ALU로 충분히 구현 가능하다. <그림 5>의 기호 ①, ②, ③은 한 사이클씩 처리 순서를 나타낸다. 이때 ①, ②의 과정에서 기존의 MAC 연산이 아니라 덧셈을 먼저 하는 AMPY(Add and Multiply) 연산이 필요하게 된다. AMPY 명령어는 MAC 명령어를 구현하는 방식의 역으로 단지 간단한 멀티플렉서(Multiplexer) 만을 사용하여 덧셈 후 곱셈을 수행하도록 데이터 패스 스위칭을 하여 구현될 수 있다. 두 개의 AMPY 연산이 동시에 수행되어야 하므로 dual AMPY 명령어가 사용된다. ①, ② 과정의 dual AMPY에 의한 두 개의 뺄셈 결과값을 더하기 위해 consecutive ADD 명령어가 사용된다. ③의 과정에서 한 번의 곱셈 후에 덧셈과 뺄셈을 수행하는 MDAC(Multiply and Double Accumulate) 명령어가 수행된다. 두 개의 MDAC 명령어가 동시에 수행되어야 하므로 ③의 과정에서 dual MDAC 명령어가 사용된다. FFT의 계수 값은 삼각함수 값이며, 내장

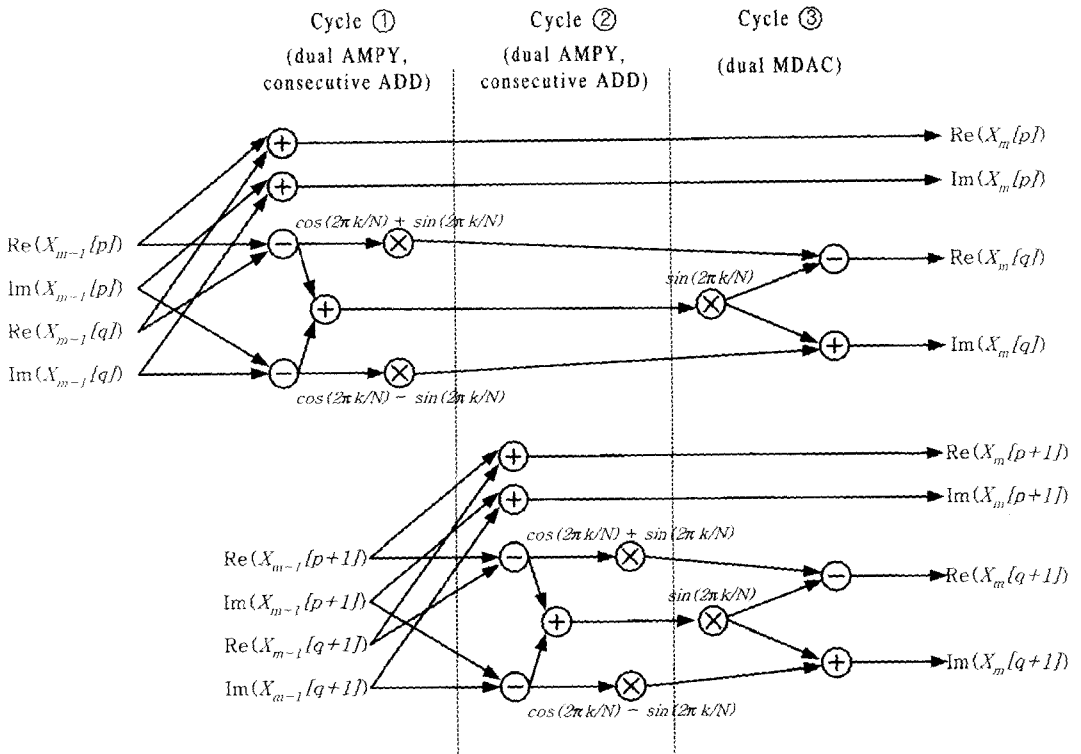


그림 5. 향상된 FFT 연산 순서도

Fig. 5. The flowgraph of the enhanced FFT.

메모리나 롬 테이블 형식으로 저장되어 있으며, 대개 1 ~ 1 값을 가지므로 cos 값과 sin 값의 합은  $\sqrt{2} - \sqrt{2}$  사이의 값이 되어 1 bit의 데이터 폭만 추가하여 가지면 되므로, 데이터 포맷 설정 시 문제가 되지 않는다.

<그림 5>의 곱셈 연산 구조는 반드시 승수가 확정적이어야 미리 감가산 연산을 취하여 준비될 수 있으며, 만약 승수가 불확실한 랜덤 값을 가지면 미리 감가산 연산을 취할 수 없으므로 연산 사이클이 한 사이클 더 필요하고 ALU도 2 개나 더 필요해 오히려 기존 복소수 연산 방식보다 못한 성능을 보인다. 이러한 이유로 <그림 4>의 복소수 연산 구조는 프로그래머블 칩에서 잘 사용하지 않았던 것이다. 그러나, FFT의 계수는 결정적이기 때문에 <그림 4>의 복소수 연산 구조를 도입할 수 있는 것이다. 실제로 FFT의 계수는 위에 설명한 바와 같이 내장 메모리나 롬 테이블 형태로 미리 저장되어 사용된다. 반도체 공정 기술의 발달에 따른 집적도의 증가에 따라 곱셈기를 2개 이상 활용하여 병렬 처리하는 구조로 변경하더라도 본 구조는 기존 방법보다 2 배 속도를 향상시킬 수 있다. 즉 한 개의 버터플라이 연산 당 평균 1.5 사이클이 걸리는 셈이다.

그 밖에 ADMAC(Add and Dual MAC)이라는 명령어를 두어 <그림 6>과 같은 순서로 복소수 연산을 처리한다면 효율적이다. <그림 6>의 ①의 과정에서 한 번의 뺄셈 후에 두 번의 곱셈을 수행하는 ADMPY(Add and Double Multiply) 명령어가 요구된다. ②의 과정에서 뺄셈을 먼저 수행한 뒤 dual MAC 연산을 수행하는 ADMAC 명령어가 사용된다. <그림 6>의 연산 방법은 한 개의 버터플라이 연산에 2 사이클이 소요된다.

제안한 명령어들은 VLIW 구조<sup>[10]</sup>에서와 같이 '||' 구문을 사용해 다른 명령어들과 병렬로 수행 가능하여야 한다.

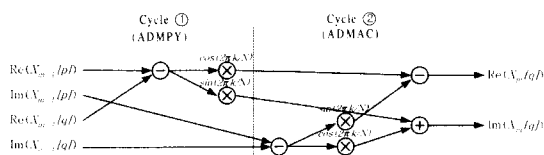


그림 6. AMAC 명령어를 사용한 FFT 연산 순서도  
Fig. 6. The flowgraph of FFT implementation using AMAC instruction.

<그림 7>은 dual AMPY, consecutive ADD, dual MDAC, ADMPY, ADMAC 명령어를 이용해 고속 FFT 연산을 처리하기 위하여 제안한 하드웨어 구조이다. PR은 파이프라인 레지스터를 의미하며 <그림 7> 하단에 명령어들이 어떤 데이터 흐름으로 수행되는지를 나타내었다.

<그림 7>에서 나타나 듯이 복소수 연산을 빠르게 처리할 수 있도록 곱셈기 사이에 덧셈기 구조를 넣었으며, 기존 DSP 연산 유닛은 그대로 사용 가능하다. 즉, 일반 DSP 명령어 및 연산을 처리하기 위한 연산 유닛의 배치 구조를 유지하면서 고속 FFT 연산을 수행할 수 있는 구조로써 사용되는 감가산기 및 곱셈기는 기존 연산 유닛으로도 구현 가능하다.

<그림 7>에서 가는 실선은 <그림 5>의 ①, ② 연산 시 데이터 흐름이고, 굵은 점선은 ③의 연산시 데이터 흐름이다. 스위칭 네트워크로부터 덧셈기2, 덧셈기3 그리고 덧셈기2, 덧셈기3으로부터 곱셈기0, 곱셈기1로의 가는 실선은 dual AMPY 명령어의 데이터 흐름을 나타낸다. 덧셈기2, 덧셈기3으로부터 덧셈기4로의 가는 실선은 consecutive-ADD 연산 수행을 위한 데이터 흐름이다. 다른 사이클 지연 없이 곱셈기2, 3으로부터 덧셈기0, 1, 2, 3으로의 굵은 점선의 데이터 흐름으로 ③의

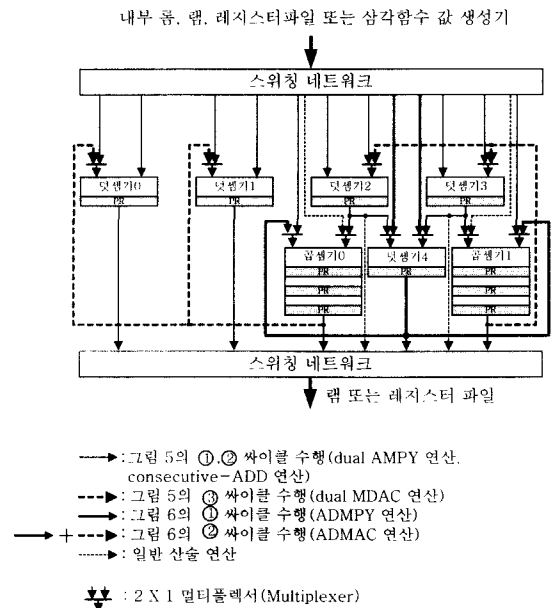
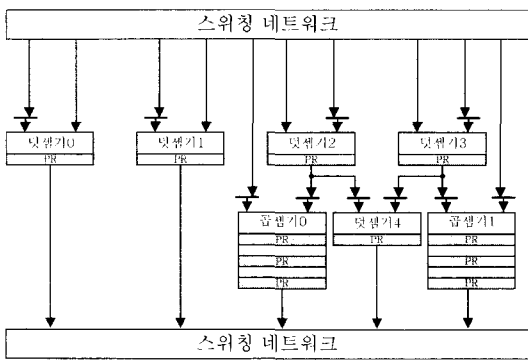


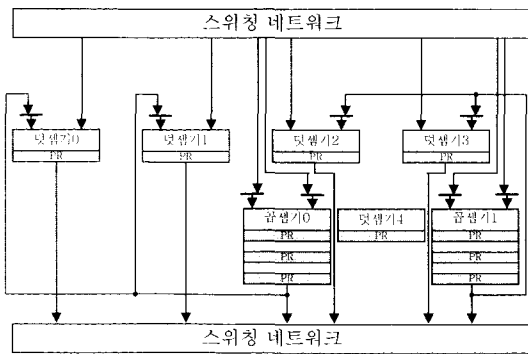
그림 7. 제안한 FFT 명령어용 하드웨어 구조  
Fig. 7. The proposed architecture for the FFT instruction.

dual MDAC 연산을 수행한다. 그림에서 나타난 바와 같이 기존 상용 DSP가 보유하고 있는 연산 유닛의 수와 같으면서 두 배 이상의 연산 속도를 증가시킬 수 있다. 제한한 하드웨어 구조에서는 곱셈기를 3단 파이프라이닝 하였으나 앞서 설명한 바와 같이 파이프라인 깊이가 더 깊거나 얕더라도 감가산기와 스테이지 보조를 맞추어 파이프라이닝을 한다면 파이프라인 깊이에 상관없이 출력 데이터의 잠복 사이클은 같은 결과(1.5 사이클)를 나타낸다.

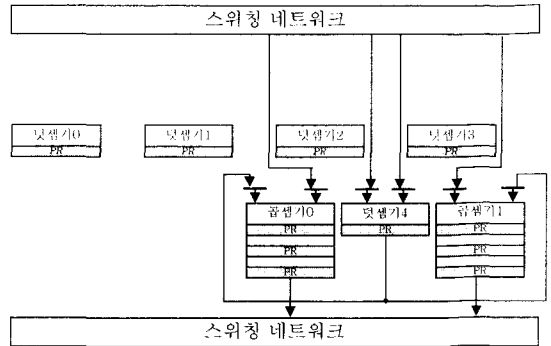
본 구조는 <그림 6>에 제안한 ADMAC 명령어를 처리할 수 있는 구조도 포함하고 있다. 본 도에서 <그림 6>에 ①의 ADMPY 연산은 스위칭 네트워크로부터 덧셈기4 그리고 덧셈기4로부터 곱셈기0, 1로의 짧은 실선의 데이터 패스로 처리하고, ②의 ADMAC 연산은 짧은 실선의 데이터 흐름 뒤에 곱셈기0으로부터 덧셈기0, 1 그리고 곱셈기1로부터 덧셈기2, 3으로의 짧은 점선으로 표시된 데이터 패스를 따라 처리 가능하며 따라서 출력데이터의 잠복사이클은 2 사이클이 되어 전용 radix-2의 잠복 사이클과 같게 된다. 또한, 얇은 점선의 흐름으로 일반 산술 연산의 수행이 가능하다.



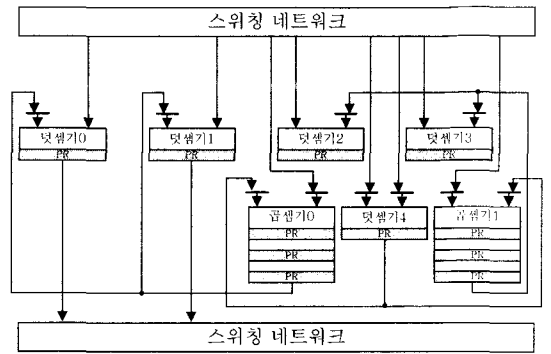
(a)



(b)



(c)



(d)

그림 8. 각 명령어의 데이터 패스 (a) dual AMPY || consecutive ADD 명령어 (b) dual MDAC 명령어 (c) ADMPY 명령어 (d) ADMAC 명령어

Fig. 8. The data path for each instruction (a) The dual AMPY || consecutive ADD instruction (b) The dual MDAC instruction (c) The ADMPY instruction (d) The ADMAC instruction.

<그림 8>은 각 명령어에 대한 데이터 패스를 보여준다. <그림 8(a)>는 dual AMPY || consecutive ADD 명령어를 위한 데이터 패스이고 <그림 8(b)>는 dual MDAC 명령어의 데이터 패스이다. <그림 8(c), (d)>는 각각 ADMPY와 ADMAC 명령어의 데이터 패스를 나타낸다.

지금까지 설명한 데이터 패스의 흐름은 <그림 7>의 하단에 나타난 멀티플렉서를 사용하여 제어하는 것이며, 2×1 멀티플렉서를 사용하므로 기존 DSP의 연산 유닛들에 추가되는 데이터 패스 회로의 크기는 매우 작고 간단하다. FFT 계수인 삼각함수는 어떤 형태로 생성 또는 저장되는가에 따라 롬 테이블, 내부 램, 삼각 계수 생성기의 형태를 취할 수 있으며, 처리되는 피

승수 값들은 내부 램, 레지스터 파일 등의 형태로 저장된다. 상용 DSP에는 원활한 데이터 흐름을 위해서 AGU가 존재하며, 본 구조는 이러한 AGU가 지원되는 환경이라는 가정 하에 설계된 것이다. 결론적으로 본 구조는 기존 DSP 칩 구조보다 2 배의 FFT 연산 성능을 높일 수 있으며, 기존 DSP의 연산 유닛 종류와 그 구조를 최대한 보존하고, 최소의 데이터 패스 회로만을 추가하여 구현할 수 있으므로, 기존 연산 모듈들의 재사용도 용이하게 할 수 있고, 칩 구현시 오버 헤드도 최소화하여 하드웨어 부담을 줄일 수 있다.

2. 제안한 구조의 성능 평가

제안하는 FFT용 명령어 및 그 하드웨어 구조의 성능을 검증하기 위하여 제 1 절의 DPU 구조를 Verilog HDL로 모델링하여 HYUNDAI™ 0.35 μm 표준 셀 라이브러리와 SYNOPSIS™ 툴을 사용하여 합성하였다.

표 1. DSP 상에서의 complex FFT 구현 성능 비교

Table 1. The performance comparison of complex FFT on DSP architectures.

DSP 구조 \ 포인트 수(N)	256	1024
Lucent Technologies의 DSP1620	16065	.
Motorola Semiconductors의 DSP56002	9600	49680
Texas Instruments의 TMS320C54x	8542	42098
Motorola Semiconductors의 DSP56303	9096	.
Philips Semiconductors의 R.E.A.L Core(125Mhz)	8355	41777
Analog Devices의 ADSP-21062 SHARC	4045	19245
Texas Instruments의 TMS320C62x	5140	20780
Infineon Technologies의 CARMEL DSP Core(120 Mhz)	2448	11628
DSP Group의 PalmDSP Core(130 Mhz)	2047	10240
Motorola, Lucent의 STAR CORE™(SC140)(300 Mhz)	2040	10239
제안하는 DSP용 FFT 명령어 하드웨어 구조	1536	7680

CADENCE™ Verilog-XL을 사용하여 타이밍 분석을 수행한 결과 최대 지연 시간은 6.92 ns가 되었다. 합성 가능한 HDL 코드 작성시 감가산기는 1 단 파이프라인 CLA(Carry Lookahead Adder)를 사용하였고, 곱셈기로는 wallace tree 기반의 3 단 파이프라인 곱셈기를 사용하였다. Wallace tree 직전에 파이프라인 레지스터를 사용하였고, wallace tree 안에서 파이프라인 레지스터를 삽입하였으며, 마지막으로 CPA(Carry Propagate Adder) 직전에 한 번 더 레지스터를 삽입하였다.

<표 1>은 기존 DSP 구조상에서의 FFT 실행 시간과 제안하는 FFT 연산 구조의 실행 사이클을 비교하여 나타내고 있다.<sup>[7-12]</sup> N=256인 경우  $256/2 \times \log_2 256$  개의 버터플라이 연산을 수행해야하며, 1 개의 버터플라이 연산에서 1.5 사이클이 걸리므로  $256/2 \times \log_2 256 \times 1.5 = 1536$  클럭 사이클이 필요하며, N=1024인 경우  $1024/2 \times \log_2 1024 \times 1.5 = 7680$  클럭 사이클이 필요하게 된다. 합성결과에서 6.92 ns의 최대 지연 시간을 나타내었으므로 256 포인트 연산의 경우  $1536 \times 6.92$  ns로 10.63 us가 요구되며 1024 포인트의 경우  $7680 \times 6.92$  ns로 53.15 us가 요구된다. Texas Instruments 사의 TMS320C62x의 경우는 동작 주파수가 200 Mhz 이상 되므로, 연산 수행 시간이 빠르고, Infineon Technologies 사의 CARMEL DSP 코어 (120 Mhz)와 DSP Group 사의 PalmDSPCore™(130 Mhz) 경우는 코어가 사용되는 공정에 따라 클럭 주파수가 다르나 일반 공정(0.25 μm 이하)에서 대개 100 Mhz에서 150 Mhz 정도에서 동작 가능하며 두 개의 MAC 연산기를 가지는 것은 일반 DSP와 같은 사양이나 산술 연산기(ALU)가 무려 7 개 정도 되므로 <그림 2>에서 ① 번의 연산을 이전 스테이지의 ③ 번 연산과 동시에 수행 할 수 있어 1개의 버터플라이 연산 수행 사이클을 2 사이클로 줄일 수 있다. 따라서, 표에 나타난 바와 같이 빠른 FFT 연산 수행이 가능하다. STARCORE™ (SC140)는 300 Mhz까지 동작 가능하며, 4개의 MAC 연산기가 있어 복소수 연산을 1 사이클 안에 수행 할 수 있다. 즉, 1 개의 버터플라이 연산을 2 사이클 안에 수행할 수 있기 때문에 고속 FFT 연산이 가능한 것이다.

제안한 FFT 전용 연산 명령어 및 하드웨어 구조는 최신 DSP 칩에서 포함되어 있는 유연성 있는 AGU (Address Generation Unit)가 지원되었다고 가정된 것이며, 실제로 최신 DSP 칩들은 AGU 안에 어드레스 연산 전용 ALU를 2 개씩 보유하고 있는 실정이다. 비



고 결과표에 나타난 바와 같이 제안하고 있는 하드웨어 구조는 최신 상용 DSP들과 비교하여 우수한 성능을 가질 수 있음이 나타났다.

#### IV. 결 론

본 논문에서는 이중 MAC에 의존한 FFT 연산만을 추구하던 기존 DSP 구조상의 FFT 연산 성능 저해 요소를 탈피하고 새로운 복소수 연산 방법 및 이를 위하여 명령어를 제안하였다. 새로운 명령어 AMPY, consecutive ADD, MDAC, ADMPY, ADMAC을 제안하였으며, 이 명령어들 뿐만 아니라 일반 DSP 명령어도 처리 가능한 DPU 구조를 제안하였다. 제안된 구조는 기존 DSP의 연산 유닛 종류와 그 구조를 최대한 보존하고, 최소의 데이터 패스 회로만을 추가하여 기존 연산 모듈들의 재사용도 용이하게 할 수 있고, 칩 구현 시 오버 헤드도 최소화하여 하드웨어 부담을 줄일 수 있다. 제안된 명령어 및 하드웨어 구조는 Verilog HDL로 모델링을 수행하였으며, SYNOPSIS™ 툴을 사용하여 0.35  $\mu\text{m}$  공정에서 논리 합성을 수행하고 시뮬레이션을 하였으며, 최대 지연 경로는 6.92 nsec로 나타났다. 앞에서 보인 바와 같이 제안한 구조는 최신 VLIW 구조의 고성능 DSP를 포함한 기존 DSP칩들과 비교하여 우수한 성능을 보이며, 따라서 DMT 방식의 VDSL이 요구하는 FFT 연산 성능을 만족시킬 수 있다.

#### 참 고 문 헌

- [1] J. Glossner, J. Moreno, M. Moudgill, J. Derby, E. Hokenek, D. Meltzer, U. Shvadron, and M. Ware, "Trends in compilable DSP Architecture," in *Proc. Workshop on SiGNAL Processing Systems (SiPS)*, 2000, pp. 181~199.
- [2] VDSL Alliance, *VDSL Alliance Draft Standard Proposal*, April 1999.
- [3] J. G. Cousin, M. Denoual, D. Saille, and O. Sentieys, "Fast ASIP synthesis and power estimation for DSP application," in *Proc. Workshop on SiGNAL Processing Systems (SiPS)*, 2000, pp. 591~600.
- [4] B. R. Wiese and J. S. Chow, "Programmable implementations of xDSL transceiver systems," *IEEE Commun. Mag.*, pp. 114~119, May 2000.
- [5] 이 재성, 선우 명훈, "DSP를 이용한 VDSL 모델용 FFT 구현 및 하드웨어 플랫폼 설계," *한국통신학회 추계종합학술대회 논문집*, 2000년 11월, pp. 1303~1306.
- [6] J. S. Lee, Y. S. Jeon, and M. H. Sunwoo, "Design of new DSP instructions and their hardware architecture for high-speed FFT," in *Proc. IEEE Workshop on SiGNAL Processing Systems (SiPS)*, 2001, pp. 80~91.
- [7] 선우 명훈, 이 재성, "프로그램머블 프로세서에서 고속 FFT 연산을 위한 FFT 연산 방법 및 그 연산을 실행하기 위한 FFT 연산 회로," *출원 번호 제 10-2001-0043713호*, 2001년 7월
- [8] Philips Semiconductors Inc., Philips Semiconductors' R.E.A.L. DSP core for low-cost low-power telecommunication and Consumer Applications. *Technical Backgrounder From Philips Semiconductors*[Online], sep 1998.
- [9] Lucent Technologies Inc., *DSP16210 Digital Signal Processor Data Sheet*, Allentown, PA, 2000.
- [10] Texas Instruments Inc., *TMS320C62xx User's Manual*, Dallas, TX, 1997.
- [11] Berkeley Design Technology Inc. (1996) Evaluating DSP processor performance. *White Papers*[Online]. Available: <http://www.bdti.com>
- [12] Analog Devices Inc., *ADSP-2106x SHARC DSP Microcomputer Family Data Sheet*, Norwood, MA, 2000.
- [13] Motorola Semiconductors Inc., *SC140 DSP Core Reference Manual*, Denver, CO, 2000.
- [14] O. B. Sheva, W. Gideon, and B. Eran. (1999) Multiple and parallel execution units in digital signal processors. *Smartcores Articles*[Online]. Available: <http://www.dspg.com>
- [15] Infineon Technologies Inc., *CARMEL DSP Core Data Sheet*, 1999.
- [16] Soohwan Ong and M. H. Sunwoo, "A fixed-point DSP(MDSP) chip for portable

- multimedia," *IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E82-A, pp. 939~944, June 1999.
- [17] Soohwan Ong, Myung H. Sunwoo, and Manpyo Hong, "A fixed-point multimedia DSP chip for portable multimedia services," in *Proc. IEEE Workshop on Signal Processing Systems Design and Implementation*, Oct. 1998, pp. 94~102.
- [18] A. V. Oppenheim and R. W. Schaffer, *Discrete-Time Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
- [19] P. Pirsch, *Architectures for Digital Signal Processing*. New York, NY: Wiley, 1998.
- [20] A. Wenzler and E. Lüder, "New structures for complex multipliers and their noise analysis," in *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS)*, 1995, pp. 1432~1435.

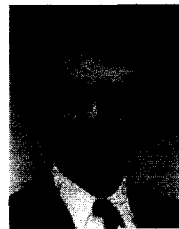
저 자 소 개



李 在 晟(正會員)

1999년 2월 아주대학교 전자공학부 학사. 2001년 2월 아주대학교 전자공학과 석사. 2001년~현재 한국 전자통신연구소(ETRI) 연구원. <주관심분야 : VLSI Architecture, 병렬 처리 프로세서 및 DSP 칩 설계,

Protocol Processing>



鮮于明勳(正會員)

1980년 서강대학교 전자공학 학사. 1982년 한국과학기술원 전기 및 전자공학 석사. 1982년~1985년 한국 전자통신연구소(ETRI) 연구원. 1985년~1990년 Univ. of Texas at Austin 전기 및 컴퓨터 공학 박사.

1990년~1992년 미국 Motorola, DSP Chip Division. 2001년~현재 IEEE Senior Member. 1992년~현재 아주대학교 전자공학부 교수. <주관심분야 : VLSI 및 SoC Architecture, 멀티미디어 통신용 DSP 칩 및 ASIC 설계>