

論文2002-39TC-12-1

OFDM 시스템을 위한 고속 FFT 프로세서

(A High Speed FFT Processor for OFDM Systems)

趙炳珏*, 孫秉秀**, 鮮于明勳***

(Byung Gak Jo, Byung Soo Son, and Myung Hoon Sunwoo)

요약

본 논문에서는 고속 데이터 전송을 위한 OFDM(Orthogonal Frequency Division Multiplex) 시스템용 고속 FFT 프로세서를 제안한다. 제안된 구조는 단일 메모리 구조를 채택하였으며 고속 연산을 위해 Radix-4 알고리즘과 메모리 뱅크 구조를 사용하였다. 또한, 버터플라이 출력이 입력 데이터의 위치에 저장되는 In-place 메모리 구조를 사용하여 메모리의 크기를 줄였다. 설계한 프로세서는 내부 데이터와 회전인자는 각각 20 비트로 설계되었으며, 약 80dB의 SQNR 성능을 갖는다. 그리고 VHDL로 모델링한 후 삼성 0.5 μ m SOG 공정으로 합성하여 메모리를 제외한 전체 게이트 수가 98,326개를 보였으며 제안된 구조는 1,024-포인트부터는 기존의 파이프라인 구조보다 하드웨어 측면에서 이득을 가진다. 동작속도는 42MHz로 256-포인트 연산이 6 μ s에 처리 가능한 구조로 HomePlug 표준안의 8.4 μ s의 처리속도를 만족시킨다.

Abstract

This paper proposes a high-speed FFT processor for orthogonal frequency-division multiplexing (OFDM) systems. The proposed architecture uses a single-memory architecture and uses a radix-4 algorithm for high speed. The proposed memory is partitioned into four banks for high-speed computation. It uses an in-place memory strategy that stores butterfly outputs in the same memory location used by butterfly inputs. Therefore, the memory size can be reduced. The SQNR of about 80dB is achieved with 20-bit input and 20-bit twiddle factors. The architecture has been modeled by VHDL and logic synthesis has been performed using the SamsungTM 0.5 μ m SOG cell library (KG80). The implemented FFT processor consists of 98,326 gates excluding memory. It has smaller hardware than existing pipeline FFT processors for more than 1024-point FFTs. The processor can operate at 42MHz and calculate a 256-point complex FFT in 6 μ s. It satisfies the required processing speed of 8.4 μ s in the HomePlug standard.

Keywords : FFT, Radix, In-place, OFDM

* 正會員, 國防科學研究所

(Agency for Defense Development)

學生會員, * 正會員, 亞洲大學校 工科大學 電子工學部
(School of Electrical and Computer Eng., Ajou Univ.)

※ 본 논문은 국가지정연구실사업(NRL) 및 IDEC 사업의 지원을 받아 수행되었음.

接受日字:2002年7月4日, 수정완료일:2002年11月15日

I. 서론

OFDM 및 DMT(Discrete MultiTone) 변복조는 입력 데이터를 부 반송파의 수 만큼 직·병렬 변환하여 각각에 대응되는 부 반송파(subcarrier)로 변조하는 방식이다. OFDM 방식은 주파수 선택적 페이딩에 강하고, 채널을 효율적으로 이용할 수 있으며, 심벌간 간섭이 심한

채널에서 복잡한 등화기 없이 효율적인 전송을 보장한다는 점에서 고속의 데이터 전송방식으로 널리 채택되고 있다. 부 반송파 변조 방식은 고속 데이터 통신을 위한 방식으로 WLAN(Wireless Local Area Network), DAB(Digital Audio Broadcasting), DVB (Digital Video Broadcasting), VDSL(Very High-speed Digital Subscriber Line), HomePlug 등의 표준에 채택 또는 채택될 예정이며 활발히 연구가 진행되고 있다¹⁻⁴.

본 논문은 DC~25 MHz의 주파수 대역을 갖고 256개의 부 반송파를 사용하는 다중 반송파 방식의 HomePlug 표준안에 초점을 맞추었다⁴. 전력선망은 현재 구축되어 있는 기존의 망 중에서 가장 완벽하고 세밀하게 구성되어 있는 망으로써 전력선을 이용하여 통신망을 구축할 경우 신규 투자를 최소화할 수 있다는 장점이 있다.

부 반송파를 이용한 변조는 DFT(Discrete Fourier Transform)를 이용하여 구현하며, 실제 하드웨어 설계에는 DFT나 IDFT(Inverse Discrete Fourier Transform)를 사용하지 않고 연산량을 줄이기 위해서 FFT 알고리즘을 이용한다. FFT 프로세서는 OFDM 시스템에 있어 가장 큰 복잡도를 가지며 고속 연산이 요구되어 구현이 어려운 부분으로 hardwired ASIC 설계가 일반적이다².

기존의 FFT의 구조는 크게 단일 메모리 구조와 이중 메모리 구조 그리고 파이프라인 구조로 나눌 수 있다^{5,6}. 단일 메모리 구조는 적은 하드웨어를 가진다는 장점이 있고 이중 메모리구조는 복잡한 메모리 어드레싱 없이 연속처리가 용이하다는 장점을 가지고 있다. 파이프라인 구조는 고성능으로 연산을 처리할 수 있다는 장점이 있으나 하드웨어가 큰 단점이 있다^{5,6}.

본 논문에서 목표로 한 FFT 프로세서는 단일 메모리 구조로써 8.4μs내에 256-포인트 연산을 수행 가능하여 HomePlug의 표준을 만족시키는 OFDM 시스템을 위한 구조이다. 이러한 고성능을 요구하는 분야에는 여러 종류의 파이프라인 구조가 주로 사용되고 있으나 파이프라인 구조는 스테이지 수만큼의 연산부를 요구하여 포인트 수가 증가할 경우 많은 면적을 소모한다⁷. 이러한 단점을 극복하기 위하여 본 논문에서는 메모리 구조를 채택하여 적은 하드웨어 크기를 유지하며 Radix-4 FFT 알고리즘과 메모리 뱅크 구조를 사용한 적절한 어드레싱 방법으로 Homeplug 표준안의 성능을 만족시키는 높은 성능을 얻을 수 있고 256-포인트 이상의 FFT 연산 시 포인트가 증가할수록 파이프라인 구조 보다 더욱 큰

하드웨어 절감을 가지는 구조를 제안한다⁸⁻¹¹.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 FFT 구조에 대해 설명하고 3장에서는 제안하는 메모리 구조의 FFT 프로세서에 대해 기술한다. 4장에서는 제안한 FFT 프로세서의 구현 및 성능 평가를 기술하며 마지막으로, 5장에서 결론을 맺는다.

2. 기존의 FFT 구조

기존의 FFT 구조들을 <그림 1>에 나타내었다⁷. <그림 1>에서 프로세싱 부분은 버터플라이 연산부에 해당한다. <그림 1(a)>의 단일 메모리 구조는 하나의 버터플라이 연산부와 메모리를 사용하여 메모리로부터 데이터를 읽어 버터플라이 연산을 수행한 뒤 다시 같은 메모리로 결과를 저장하는 in-place 방식으로 하드웨어 비용이 적게 드는 장점이 있다¹². <그림 1(b)>는 이중 메모리 구조로 2 개의 메모리를 사용하여 한 메모리로부터 데이터를 읽어 연산을 수행한 뒤 다른 메모리에 결과를 저장하는 방식으로 FFT 연산 스테이지가 바뀔 때마다 두 메모리의 역할이 바뀐다. 이 구조는 데이터의 입력과 출력을 동시에 수행 가능하여 단일 메모리 구조보다 높은 처리 속도를 얻을 수 있다. 위의 두 메모리 구조는 적은 하드웨어를 가지는 반면 모든 FFT 심볼의 연산이 끝나야 다음 심볼의 연산이 가능하므로 성능이 떨어지며 높은 동작 주파수를 요구한다. <그림 1(c)>는 파이프라인 구조로 매 스테이지마다 버터플라이 연산부

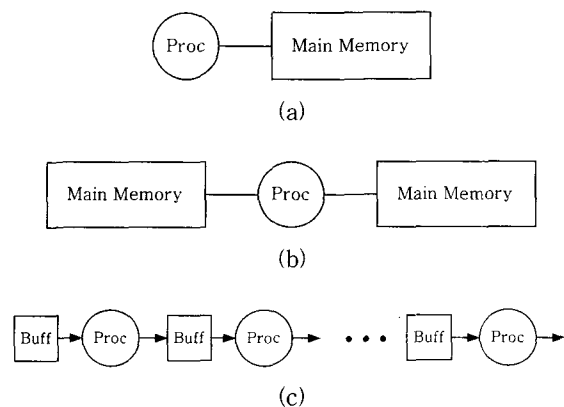


그림 1. 기존의 FFT 구조. (a) 단일메모리 구조. (b) 이중메모리 구조. (c) 파이프라인 구조.

Fig. 1. Conventional FFT architectures. (a) Single-memory architecture. (b) Dual-memory architecture. (c) Pipeline architecture.

를 사용하는 구조로 높은 성능을 요구하는 분야에 사용된다^[6]. 그러나 이 구조는 매 스테이지마다 버터플라이 연산 구조가 사용되므로 하드웨어 크기가 큰 단점이 있다.

3. 제안하는 FFT 구조

제안하는 FFT 프로세서는 높은 성능을 만족하도록 설계하였다. 메모리는 뱅크로 나누어 사용하고, 요구되는 성능을 만족하기 위해 Radix-4 알고리즘을 사용하여 제안한 구조는 파이프라인 구조에 가까운 성능을 얻을 수 있다.

제안한 FFT 프로세서는 기본적으로 in-place 알고리즘을 적용한 단일 메모리 구조이며 <그림 2>에 대략적인 구조를 나타내었다. 1개의 메인 메모리를 가지고 있으며 각 메모리는 4개의 뱅크로 이루어져 있다. 1개의 버터플라이로 반복적인 연산을 수행하며 고속 수행 및 256(44)-point FFT 연산에 적합한 Radix-4 알고리즘을

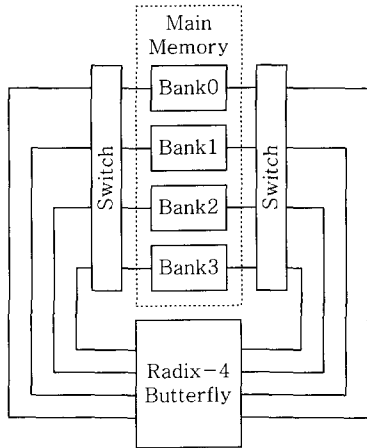


그림 2. FFT 프로세서의 구조.
Fig. 2. The proposed FFT architecture.

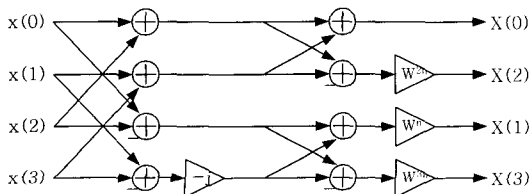


그림 3. Radix-4 DIF 버터플라이.
Fig. 3. The radix-4 DIF butterfly.

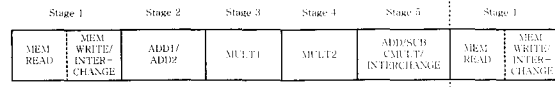


그림 4. 파이프라인 다이어그램
Fig. 4. The pipeline diagram.

사용하였다^[3]. <그림 3>에 설계한 Radix-4 DIF (Decimation-In-Frequency) 버터플라이 연산부를 나타내었다.

고속의 프로세서에서 동작 주파수가 증가하게 되면, 이에 따르는 메모리의 액세스 속도도 증가해야 하지만 메모리의 액세스 속도는 한정되어 있으므로 데이터를 읽거나 쓸 때 충분한 시간을 가져야 한다. 본 구조는 Radix-4 알고리즘을 사용함으로써 요구되는 버터플라이 연산 회수를 줄여 동작 속도를 낮추었고, 또한 메모리를 4개의 뱅크로 나누어 사용하여 동시에 4개의 데이터를 액세스 함으로써 동작 속도를 낮추었다. 256-포인트 심볼에 대한 버터플라이 연산 회수가 $\log_2 256$ (스테이지) $\times 64$ (스테이지 당 버터플라이 연산 회수) = 256 회이고 8.4 μs 에 연산이 완료되어야 하므로 버터플라이는 30MHz 이상의 동작 주파수를 가져야 한다. 또한, 메모리는 in-place 연산 수행을 위해 버터플라이 동작 주파수의 두 배인 약 60MHz 이상의 동작 주파수를 가져야 한다.

본 프로세서의 버터플라이는 한 사이클에 하나의 Radix-4 DIF 버터플라이 연산을 수행한다. 이 연산의 데이터는 복소수 고정 소수점 형식이며 실수부와 허수부는 각각 20bit으로 구성되어있다. 이 프로세서의 데이터 경로는 <그림 4>에서 나타난 것과 같이 5단의 파이프라인 단계를 가지며 첫 번째 단은 두 개의 부수적인 단으로 나뉘지며 앞부분은 버터플라이의 입력이 메모리로부터 입력되고 뒷부분은 4개의 버터플라이 입력이 상호 교환되고 이전에 계산된 버터플라이 출력이 메모리에 쓰여진다. 두 번째 단은 <그림 3>에 보여진 2단의 덧셈 연산이 수행되고 세 번째와 네 번째 단은 실수부와 허수부의 곱셈이 수행된다. 다섯 번째 단은 복소곱셈기의 연산이 완료되고 버터플라이의 결과가 상호 교환되어 메모리에 저장된다.

1. 제안하는 in-place 메모리 어드레싱

본 절에서는 Radix-4 알고리즘을 하나의 메모리로 처리하기 위한 in-place 메모리 어드레싱에 대해 기술한다^[9]. Radix-4 버터플라이 연산을 위해 4개의 데이터를

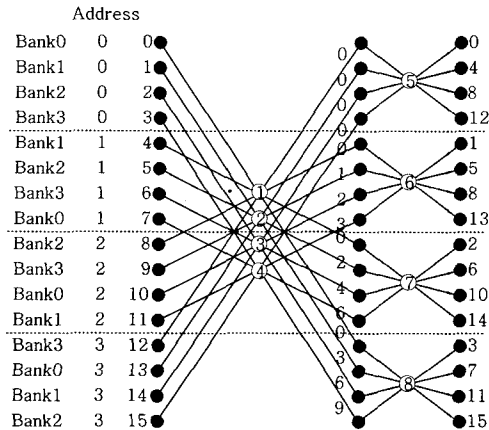


그림 5. 16-포인트 FFT의 in-place 어드레싱.
Fig. 5. The data flow graph of a in-place 16-point FFT.

표 1. 256-포인트 FFT의 주소할당

Table 1. The address assignment for a 256-point FFT.

Data Bank Address	Data Bank Address	Data Bank Address
0 0 0	16 1 4	:
1 1 0	17 2 4	:
2 2 0	18 3 4	:
3 3 0	19 0 4	:
4 1 1	20 2 5	244 3 61
5 2 1	21 3 5	245 0 61
6 3 1	22 0 5	246 1 61
7 0 1	23 1 5	247 2 61
8 2 2	24 3 6	248 0 62
9 3 2	25 0 6	249 1 62
10 0 2	26 1 6	250 2 62
11 1 2	27 2 6	251 3 62
12 3 3	:	252 1 63
13 0 3	:	253 2 63
14 1 3	:	254 3 63
15 2 3	:	255 0 63

동시에 액세스하여야 하고 액세스한 위치에 버터플라이 결과 4개를 동시에 저장하여야 한다. 이를 위해 메인 메모리를 4개의 뱅크로 나누어 사용하며 동시에 한 뱅크에서 여러 개의 데이터를 액세스하지 않기 위해 적절한 어드레싱을 수행한다.

제안하는 in-place 메모리 어드레싱의 예를 <그림 5>에 도시하였다. 만약 데이터가 <그림 5>에서 보여진 4개의 뱅크에 위치해 있다면 4개의 입력은 다른 뱅크로부터 읽혀질 수 있고 4개의 출력은 버터플라이 연산 ①에서부터 ⑧까지 전 단계 걸쳐 각각 다른 뱅크에 쓰여질 수 있다. 즉, <그림 5>에 나타난 규칙으로 입력 데이터를 각 뱅크에 나누어 저장하면 데이터 흐름도 상의

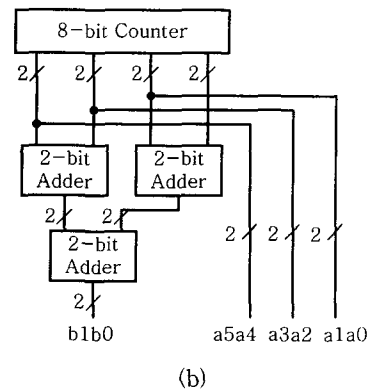
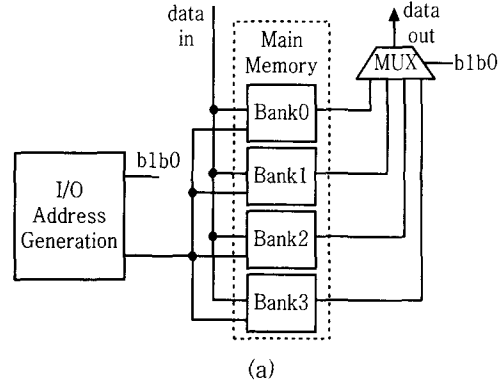


그림 6. In-place 어드레싱을 위한 구조 (a) 데이터 입출력을 위한 구조 (b) 입출력 주소 생성부
Fig. 6. The architecture for in-place addressing. (a) The architecture for data input and output. (b) The I/O address generation.

모든 버터플라이의 4개의 입력을 서로 다른 뱅크로부터 액세스가 가능하다. 본 논문에서는 <그림 5>의 방식을 256-포인트 FFT에 확장 적용하였다.

<표 1>은 256-포인트 FFT 연산에 대한 주소 지정방식과 각 데이터에 할당되어진 뱅크와 주소를 나타내었다. <그림 6(a)>는 직렬 데이터 입력을 저장하고 <표 1>에 보여진 순서로 연산되어진 데이터를 출력하기 위한 구조를 보여준다. 데이터 입력포트는 모든 메모리의 뱅크에 연결되어있다. 데이터 입력을 위한 올바른 메모리 뱅크의 선택은 쓰기 신호의 조정에 의해 수행되어진다. 데이터 출력의 경우에 멀티플렉서와 <그림 6(b)>의 입출력 주소 생성부에 의해서 생성되어진 뱅크선택 신호 b1b0가 사용된다. 주소 생성부는 8비트 카운터와 모듈로-4 연산을 수행하기 위한 2비트 덧셈기로 구성되어 있다. 각 메모리 뱅크의 주소 신호 a5a4a3a2a1a0은 8비트 카운터 출력의 상위 6비트로 이루어진다.

2. OFDM 시스템을 위한 구조

OFDM 및 DMT 시스템은 IFFT를 수행한 뒤 보호구간을 삽입한다. IFFT가 수행된 하나의 OFDM 심볼을 메모리에 저장한 뒤 심볼의 뒷부분 중 일정 구간을 OFDM 심볼의 앞부분에 보호구간으로 삽입하며 이를 cyclic prefix라 한다. OFDM 시스템은 cyclic prefix를 이용하여 수신 단에서 프레임 동기를 구한다. Cyclic prefix를 삽입하기 위한 구조는 <그림 7>과 같다^[10,14].

<그림 7>에서 IFFT가 수행된 심볼을 메모리에 저장한 뒤 출력할 때 cyclic prefix 구간을 먼저 출력하고 뒤 이어 OFDM 심볼을 출력한다. 본 논문의 FFT 프로세서는 cyclic prefix 삽입 구조의 메모리를 사용하여 실시간으로 연속적으로 입력되는 데이터를 처리한다. <그림 7>에 이를 위한 구조를 나타내었다. <그림 8>에 보여지는 바와 같이 cyclic prefix 삽입을 위해 사용되는 두 개의 메모리를 FFT 프로세서의 연산을 위한 메모리에 포함시킨다. 하나의 메모리가 버터플라이 연산을 위해 사용되는 동안 다른 하나의 메모리는 실시간으로 입력되는 데이터를 저장하는 데 사용하고 나머지 하나의 메모리는 출력에 사용된다. 하나의 OFDM 심볼에 대한 연산이 끝나면 버터플라이 연산에 사용되었던 메모리는 출력을 위한 메모리로 전환되고 입력을 저장한 메모리는 버터플라이 연산을 위한 메모리로 전환된다. 또한,

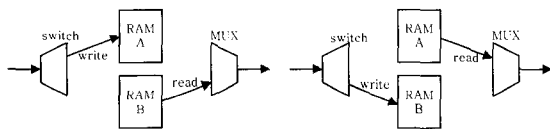


그림 7. cyclic prefix 삽입을 위한 구조.
Fig. 7. The architecture for the cyclic prefix insertion.

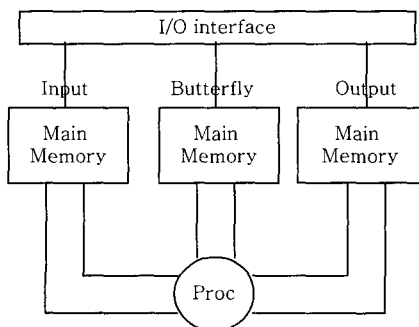


그림 8. 실시간 처리를 위한 구조.
Fig. 8. The architecture for non-stopping processing.

출력에 사용되었던 메모리는 다음 심볼의 입력을 저장하는 메모리로 전환된다. 이와 같은 구조를 사용하면 메모리의 증가없이 단일 메모리 구조를 갖는 FFT 프로세서로 실시간 OFDM 시스템의 구현이 가능하다.

4. 구현 및 성능 평가

제안한 구조를 VHDL 언어를 이용하여 하드웨어로 구현하였다. 내부 워드 길이를 20 비트로 하였으며 메인 메모리도 실수부와 허수부가 각각 20 비트의 워드 길이를 갖는 구조를 사용하여 약 80dB의 SQNR 성능을 갖는다. 버터플라이 연산부는 4 단의 파이프라인을 갖도록 설계하였다. VHDL로 설계한 FFT 프로세서를 삼성 0.5 μm KG80 SOG 공정 라이브러리를 이용하여 논리합성을 수행하였다. 논리합성 수행결과를 <그림 9>에 나타내었다.

논리합성 수행결과 메인 메모리를 제외한 전체 FFT 프로세서가 약 98,000개의 게이트 수를 보였으며 버터플라이 연산부가 약 86,000개, 계수 저장을 위한 롬이 약 1,800개, 제어부 및 메모리 어드레싱부가 10,150개의 게이트 수를 보였다. 버터플라이 연산부의 최대 지연경로는 23.46ns로 $1/23.46 = 42\text{MHz}$ 의 동작이 가능한 것으로 나타났다. 256-포인트 FFT 연산을 수행할 경우 $\log_2 256 \times 64 \times 23.46\text{ns} = 6\mu\text{s}$ 에 처리가 가능하다. 이는 전력선 모뎀의 표준인 HomePlug가 8.4μs 이내에 256-포인트 연산을 수행해야 하는 사양을 만족한다^[4].

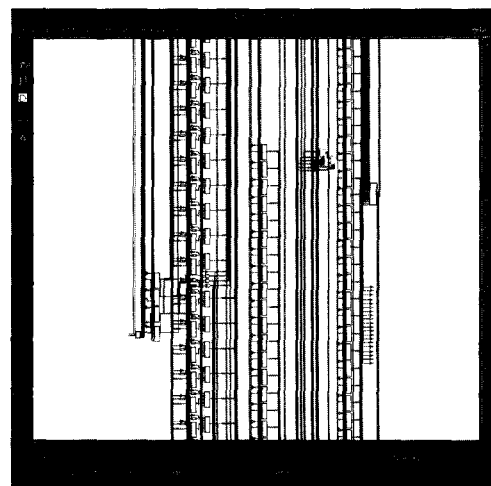


그림 9. 논리합성 수행결과.
Fig. 9. The result of logic synthesis.

또한, 4,096-포인트는 $\log 44,096 \times 1,024 \times 23.46\text{ns} = 144\mu\text{s}$ 에 처리가 가능하며 이는 DMT 방식의 VDSL 모듈이 250 μs 이내에 처리하여야 하는 것을 만족한다^[15].

제안된 구조는 radix-4 알고리즘을 기본으로 한 in-place 연산과 메모리 뱅크 구조를 사용함으로 기존의 이와 같은 알고리즘을 사용하지 않는 구조에 비해 4배의 연산사이클을 줄일 수가 있었으며 전체적인 동작 주파수를 낮추었다.

<그림 10>은 설계한 코어의 타이밍 시뮬레이션 결과를 나타낸다. 256개의 데이터가 입력된 후 연산을 마치고 FFT_OUT으로 올바른 결과 데이터가 출력되는 것을 볼 수 있다.

<표 2>는 실시간 응용에 가장 많이 사용되는 파이프라인 FFT 구조들의 하드웨어 복잡도를 나타내었다^[11]. N은 FFT 포인트 수를 의미하며 곱셈기 수와 덧셈기 수는 복소 곱셈기와 복소 덧셈기를 의미한다. 실제 FFT 프로세서의 하드웨어 복잡도는 곱셈기와 메모리가 좌우한다^[16]. 본 논문에서 제안한 메모리 구조는 복소 곱셈기의 수가 3개로 고정되어 있으며 덧셈기 또한 8개로 고정되어 있다. 메모리 크기는 N으로 포인트 수와 같다.

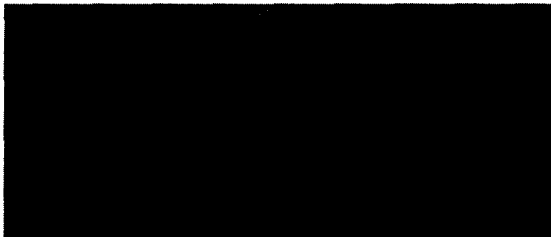


그림 10. 타이밍 시뮬레이션 결과
Fig. 10. Timing simulation result.

표 2. 파이프라인 구조의 하드웨어 복잡도
Table 2. The hardware requirements of the pipeline architectures and the proposed architecture.

	곱셈기수	덧셈기수	메모리크기
R2MDC	$2(\log_4 N - 1)$	$4\log_4 N$	$3N/2 - 2$
R2SDF	$2(\log_4 N - 1)$	$4\log_4 N$	$N - 1$
R4SDF	$\log_4 N - 1$	$8\log_4 N$	$N - 1$
R4MDC	$3(\log_4 N - 1)$	$8\log_4 N$	$5N/2 - 4$
R4SDC	$\log_4 N - 1$	$3\log_4 N$	$2N - 2$
R ² SDF	$\log_4 N - 1$	$4\log_4 N$	$N - 1$
제안하는 FFT	3	8	N

표의 곱셈기 수를 비교하여 볼 때 R4SDF(Radix-4 Single-path Delay Feedback) 구조와 R4SDC(Radix-4 Single-path Delay Commutator), R22SDF(Radix-22 Single-path Delay Feedback) 구조가 256-포인트에서는 제안한 구조와 같은 곱셈기 수를 가지나 1,024-포인트 후부터는 제안한 구조가 하드웨어 측면에서 이득을 가진다.

V. 결론

본 논문에서는 고속 데이터 전송을 위한 OFDM 시스템에 적용 가능한 고속 FFT 프로세서를 제안하였다. 단일 메모리 구조를 채택하였으며 고속 연산을 위해 Radix-4 알고리즘과 메모리 뱅크 구조를 사용하여 연산 사이클을 현저하게 줄였다. 또한, in-place 연산을 위해 새로운 메모리 어드레싱 방법을 제안하였다. 제안된 구조는 1,024-포인트부터는 기존의 파이프라인 구조보다 하드웨어 측면에서 이득을 가진다. 설계한 프로세서는 VHDL로 모델링하여 삼성 0.5 μm SOG 공정으로 합성하였으며 메모리를 제외한 전체 게이트 수가 98,326개를 보였으며 내부비트를 20 비트를 사용하여 80dB의 SQNR 성능을 갖는다. 동작속도는 42MHz로 256-포인트 연산에 6 μs 에 처리 가능한 구조이다. 제안한 고속 FFT 프로세서는 향후 VDSL, 전력선 모뎀, DVB, DAB, WLAN 등과 같은 OFDM, DMT 변복조 방식의 통신 시스템 개발에 활용될 수 있다.

참고 문헌

- [1] C. L. Wang and C. H. Chang, "A novel DHT-based FFT/IFFT processor for ADSL transceivers," in Proc. IEEE Int. Symp. Circuits Syst., 1999, pp. 51~54.
- [2] J. R. Choi, S. B. Park, D. S. Han, and S. H. Park, "A 2048 complex point FFT architecture for digital audio broadcasting system," in Proc. IEEE Int. Symp. Circuits Syst., 2000, vol. V, pp. 693~696.
- [3] VDSL Alliance, VDSL Alliance Draft Standard Proposal, April 1999.
- [4] HomePlug Powerline Alliance, HomePlug 0.5 Draft Medium Interface Specification, Nov. 2000.
- [5] B. M. Bass, "A low power, high performance,

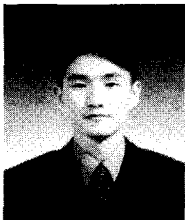
- 1024-point FFT processor, *IEEE J. Solid-State Circuits*, vol. 34, pp. 380~387, Mar. 1999.
- [6] H. F. Lo, M. D. Shieh, and C. M. Wu, "Design of an efficient FFT processor for DAB systems," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2001, pp. 654~657.
- [7] L. Jia, Y. Gao, and H. Tenhunen, "A pipelined shared-memory architecture for FFT processor," in *Proc. IEEE 42nd Midwest Symp. Circuits Syst.*, 1999, pp. 804~807.
- [8] B. S. Son, B. G. Jo, M. H. Sunwoo, and Y. S. Kim, "A high-speed FFT processor for OFDM system," to appear in *Proc. IEEE Int. Symp. Circuits Syst.*, 2002.
- [9] L. G. Johnson, "Conflict free memory addressing for dedicated FFT hardware," *IEEE Trans. Circuits Syst. II*, vol. 39, pp. 312~316, May 1992.
- [10] 김재석, 조용수, 조중휘, 이동통신용 모뎀의 VLSI 설계, 대영사, 2001
- [11] S. He and M. Torkelson, "Design and implementation of a 1024-point pipeline FFT processor," in *Proc. IEEE Custom Integrated Circuits Conf.*, 1998, pp. 131~135.
- [12] J. A. Hidalgo, J. Lopez, F. Aruguello, and E. L. Zapata, "Area-efficient architecture for fast Fourier transform," *IEEE Trans. Circuits Syst. -II*, vol. 46, pp. 187~193, Feb. 1999.
- [13] M. K. Rudberg, M. Sandberg, and K. Ekholm, "Design and implementation of an FFT processor for VDSL," in *Proc. Asia-Pacific Conf. Circuits Syst.*, 1998, pp. 611~614.
- [14] K. W. Park, S. H. Park, and Y. S. Cho, "An OFDM transmission scheme using cyclic suffix," *IEICE Trans. Commun.*, vol. E84-B, pp. 1100~1103, Apr. 2001.
- [15] ETSI Technical Specification, *Transmission and Multiplexing (TM); Access Transmission Systems on Metallic Access Cables; Very high speed Digital Subscriber Line (VDSL); Part 2: Transceiver Specification*, Feb. 2001.
- [16] N. Weste and D. J. Skellern, "VLSI for OFDM," *IEEE Commun. Mag.*, pp. 127~131, Oct. 1998.

저 자 소 개



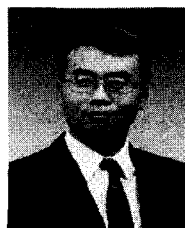
趙炳瑛(正會員)

2000년 2월 : 아주대학교 전자공학 학사. 2002년 2월 : 아주대학교 전자공학 석사. 2002년 2월~현재 : 국방과학 연구소 연구원. <주관심분야 : 통신 및 신호처리용 ASIC 설계>



孫秉秀(學生會員)

2001년 8월 : 아주대학교 전자공학 학사. 2001년 9월~현재 : 아주대학교 전자공학과, 석사과정 재학중. <주관심분야 : 통신 및 신호처리용 ASIC 설계>



鮮于明勳(正會員)

1980년 2월 : 서강대학교 전자공학 학사. 1982년 2월 : 한국과학기술원 전자공학 석사. 1982년 3월~1985년 8월 : 한국전자통신연구소(ETRI) 연구원. 1985년 9월~1990년 8월 : Univ. of Texas at Austin 전자공학 박사. 1990년 8월~1992년 8월 : Motorola, DSP Chip Division(미국). 1992년 8월~1996년 10월 : 아주대학교 전기전자공학부 조교수. 1996년 10월~2001년 9월 : 아주대학교 전자공학부 부교수. 2001년 10월~현재 : 아주대학교 전자공학부 교수. <주관심분야 : VLSI 및 Parallel Architecture, 통신 멀티미디어용 DSP 칩 및 ASIC 설계>