

심리음향모델기반 디지털 오디오 워터마킹 알고리즘의 효율적인 연산을 위한 Radix-4 256/1024 Point FFT 설계

옥승호, 문병인

경북대학교 전자전기컴퓨터공학부

전화: (053)940-8680, E-mail: wintiger@ee.knu.ac.kr, bihmoon@knu.ac.kr

Radix-4 256/1024 Point FFT Design for Efficient Operation of Audio Watermarking Algorithm based on Psychoacoustic Model

SeungHo Ock, ByungIn Moon

Kyungpook National University, School of Electrical Engineering & Computer Science

요 약

본 논문에서는 심리음향모델기반 오디오 워터마킹 시스템을 위한 Radix-4 256/1024 포인트 FFT를 설계하였다. FFT 연산의 효율성 및 데이터 처리율 (throughput)을 높이기 위해 단일지연교환기 (Single Delay Commutator; SDC)를 사용한 파이프라인 구조를 채택하였으며, 2단계 수렴 블록 부동점 (Two Step Convergent Block Floating Point; TS_CBFP) 스케일링 방법을 적용하여 신호 대 양자화 잡음비 (Signal to Quantization Noise Error; SQNR)를 약 3dB가량 향상시켰다. 그리고 주파수축음 알고리즘 (Decimation in Frequency; DIF)을 사용함으로써 출력데이터의 순서가 변경되는 문제를 10bit카운터를 사용한 어드레스발생기를 구현함으로써 해결하였다. RTL 수준으로 구현된 FFT는 0.25-um CMOS 셀 라이브러리를 사용하여 합성하였으며, 시뮬레이션 툴을 사용하여 정상 동작함을 확인하였다.

Abstract

In this paper, we described and designed the Radix-4 256/1204 point FFT for audio watermarking algorithm based on psychoacoustic model. We adopted a pipeline FFT architecture with the single delay commutator (SDC) for efficient operation and data throughput. In order to improve the signal to quantization noise ratio (SQNR), we adopted the two step convergent block floating point (TS_CBFP) algorithm. And we solved the problem caused by decimation in frequency (DIF) algorithm with 10bit counter. The FFT core designed in RTL level is synthesized with the 0.25-um CMOS cell library and verified with a HDL simulation tools.

Keywords : FFT, Radix-4, Psychoacoustic, Audio Watermarking

I. 서 론

고품질 오디오데이터를 제한된 대역으로 전송하거나 휴대용 저장장치에 기록하기 위해서는 효율적인 데이터

※ 본 연구보고서는 정보통신부 출연금으로 ETRI, SoC 산업진흥센터에서 수행한 IT SoC 핵심설계인력양성사업의 연구결과입니다.

※ 본 연구보고서의 내용을 발표할 때에는 반드시 ETRI, SoC산업진흥센터 IT SoC 핵심설계인력양성사업의 결과임을 밝혀야 합니다.

압축방법이 필요하다. 이와 관련된 연구는 MPEG (Moving Picture Experts Group)를 중심으로 활발히 진행되어 왔으며, 주로 가청주파수 대역인 20Hz ~ 20,000Hz 범위 내에서 특정 레벨 이하의 소리는 들을 수 없는 최소가청한계 및 큰 소리가 작은 소리를 마스킹(Masking)하는 마스킹 효과(Masking Effect)등의 인간의 청각특성이 사용된다. 마스킹이란 ‘어떤 음에 대한 최소가청한계가 다른 음의 존재에 의해 상승하는 현상’으로 정의되며, 이러한 현상은 소리가 인간의 청각신경계에 존재하는 수많은 필터뱅크에 의해 분석되는 과정에서 발생한다^[1]. 심리음향모델은 인간의 청각특성을 사

용하여 주파수 영역으로 변환된 오디오데이터의 마스크 곡선을 구하는 과정이며, 오디오 부호화 과정 및 오디오 워터마킹 알고리즘 등에 사용된다.

워터마킹이란 지적재산권 보호를 위한 일종의 암호화 방법이며, 오디오 워터마킹의 경우 원본 오디오 데이터에 저작권 정보를 삽입함으로써 워터마킹이 이루어진다. 하지만 저작권 정보를 삽입하는 과정은 원본 오디오 데이터를 조작하는 과정으로 이루어지며, 따라서 원본 오디오 데이터의 변형 및 손상을 줄이면서 저작권 정보를 강인하게 삽입하기 위한 방안으로 심리음향모델이 사용된다. 하지만 심리음향 모델을 워터마킹 과정에 사용하기 위해서는 연산량의 부담이 크기 때문에 이를 효율적으로 처리할 수 있는 방안이 필요하다^[2].

본 논문에서는 심리음향모델 연산의 대부분을 차지하는 256, 1024 포인트 FFT를 설계하였다. 설계된 FFT는 면적과 처리율의 교환관계(trade-off)를 적절히 만족시키는 Radix-4 파이프라인 구조를 채택하였으며, SQNR을 높이기 위해 2단계 수렴 블록 부동점 (Two Step Convergent Block Floating Point; TW_CBF) 스케일링 방법을 사용하였다^[3].

본 서론에 이어 II장에서는 FFT의 알고리즘 및 설계에 사용된 파이프라인 구조에 대해 언급하고, III장에서는 전체적인 설계 개요와 함께 세부 블록의 기능 및 동작을 설명한다. 이후 IV장에서는 설계된 FFT를 시뮬레이션 및 검증한 결과를 평가하며, 마지막 V장에서 본 논문의 결론을 맺는다.

II. FFT 알고리즘 및 아키텍처

1. FFT 알고리즘

1965년 Cooley와 Turkey에 의해 고안된 고속 푸리에 변환(Fast Fourier Transform; FFT)은 아래의 식(1)로 정의되는 이산 푸리에 변환(Discrete Fourier Transform; DFT)의 연산량을 획기적으로 줄임으로서 DFT연산을 고속으로 수행하는 알고리즘이다.

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

단, $W_N^{kn} = e^{-j\frac{2\pi}{N}nk}$

FFT의 알고리즘은 시간축음방법(Decimation in Time; DIT) 및 주파수축음방법(Decimation in Frequency; DIF)으로 구현될 수 있으며, 구현 방법에

따라 입·출력 데이터의 순서 및 회전인자의 곱셈 방법 등이 변화된다^[4].

2. FFT 구조

FFT를 구현하는 방법으로는 구조적인 측면에서 단일 나비연산기(butterfly unit) 구조를 이용하는 방법, 완전 셔플(perfect shuffle) 및 시스토릭 배열(systolic array)과 같은 병렬 구조를 이용하는 방법 그리고 파이프라인 구조를 이용하는 방법 등이 있다^[6]. 파이프라인 구조는 실시간 처리가 요구되거나 순차적인 입력에 대해 연속적인 출력이 요구되는 시스템 등에 효율적인 구조로서 구현 방법이 간단하고 높은 출력(throughput)특성을 가지고 있어 널리 사용되고 있는 구조이다^{[5][6]}.

일반적으로 파이프라인 구조는 나비연산에 입력되는 데이터를 정렬하기 위한 지연변환기(delay commutator) 및 나비연산기 그리고 복소수 승산기(complex multiplier)로 구성되며, 지연변환기의 구조 및 사용되는 radix에 따라 R2SDF(Radix-2 Single-path Delay Feedback), R4SDC(Radix-4 Single-path Delay Commutator) 등으로 구분된다^[5]. 이 중 DF구조는 나비연산기의 효율이 낮은 단점이 있는 반면, 적은 메모리로 구현이 가능한 장점이 있고, DC구조는 DF구조보다 많은 메모리가 필요하지만 나비연산기의 사용 효율이 높은 장점을 가지고 있다.

2. Convergent Block Floating Point

고정점(fixed-point) FFT의 경우 각 stage에 입력되는 데이터의 비트수를 n, 복소 승산기에 입력되는 회전인자의 데이터 비트수를 r 이라고 할 때, Radix-4 나비연산기의 출력 데이터 비트수는 최대 (n+2)비트로 확장되며, 복소 승산기의 출력 데이터 비트수는 ((n+2)+r-1)비트로 확장된다. 따라서 각 stage의 출력 데이터 비트수를 적절한 방법으로 조절해야 하며, 이를 위해 출력 데이터의 비트수를 고정된 비트수로 자르는 경우 하드웨어의 구현이 용이한 장점이 있지만 FFT의 SQNR이 나빠지는 문제가 발생한다.

FFT의 SQNR을 향상시키기 위한 방법으로 CBF가 사용되며, 이는 각 stage의 연산결과 중 가장 큰 값을 기준으로 데이터를 스케일링함으로써 연산단계 중간 결과 값의 정확도를 높이는 방법이다. 하지만 각 stage의 연산결과 중 가장 큰 값을 찾기 위해서는 출력 데이터의 크기에 비례하는 내부 버퍼가 필요하며, 따라서 하드웨어의 면적을 증가시키는 요소가 된다.

III. FFT 설계

1. 설계 개요

본 논문에서는 TW_CBFP 스케일링 방법을 적용함으로써 고정점 구조의 FFT에 비해 SQNR을 향상시킬 수 있는 구조로 설계되었으며, 심리음향모델 연산에 사용되는 256, 1024 포인트 FFT를 선택 가능하도록 하였다. 그리고 DIF알고리즘을 사용함으로써 출력데이터의 순서가 변경되는 문제를 10bit 카운터를 사용한 어드레스발생기를 구현함으로써 해결하였다.

설계된 FFT는 Radix-4 나비연산기(R4BF) 및 지연교환기(SDC), 복소 승산기(CMUL), CBFP, 회전인자(TW_ROM), 주소발생기(ADDR_GEN)등으로 구성되며, FFT의 전체 구조는 그림 1과 같다. Radix-4 256 포인트 FFT의 경우 연산에 필요한 stage의 수는 $\log_4 256=4$ 로서 전체 4단계의 stage가 필요하며, 1024 포인트 FFT의 경우 연산에 필요한 stage수는 $\log_4 1024=5$ 로서 전체 5단계의 stage가 필요하다. 그림 2의 Mux는 FFT의 각 동작 포인트에 따른 stage를 선택하는 모듈로써, 외부 컨트롤 신호인 len을 통해 제어된다. 즉, 외부 컨트롤 신호 len이 0일 경우, FFT로 입력된 데이터는 stage2부터 stage5까지 4단계의 stage를 거치면서 256 포인트 FFT 연산이 이루어지며, len이 1일 경우 FFT로 입력된 데이터는 stage1부터 stage5까지 5단계를 거치면서 1024 포인트 FFT 연산이 이루어진다.

2. Stage 연산 단계

FFT의 각 stage는 나비연산을 위해 데이터의 순서를 재배치시키는 지연교환기와, 나비연산기, 그리고 복소 승산기로 구성된다. 각 stage의 나비연산기는 그림 2와 같은 구조로 연산이 수행되며, 6개의 가산기와 2개의

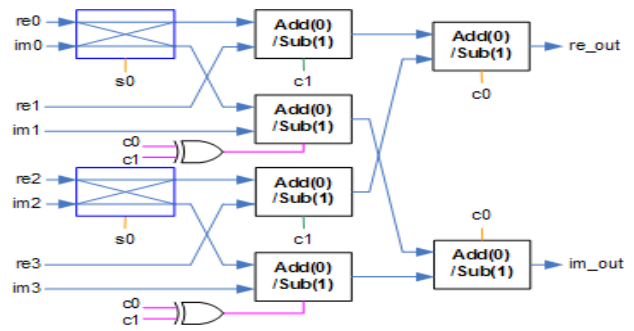


그림 2. 나비연산기 구조
Fig 2. Butterfly unit architecture

표 1. 나비연산기 제어 신호

Table 1. Butterfly unit control signal

Step	c1	c0	s0
1	0	0	
2	0	1	
3	1	0	
4	1	1	

선택기로 구현하였다.

나비연산을 통해 출력된 데이터는 회전인자와 함께 복소승산기로 입력된다. 이때 회전인자의 저장에 사용되는 ROM의 크기를 줄이기 위해 표2 와 같이 내부 컨트롤 신호 및 삼각함수의 성질을 이용하여 회전인자를 제어함으로써 ROM의 크기를 약 30%줄일 수 있었다.

표 2. 회전인자 제어

Fig 2. Twiddle factor control

θ	real factor, $\cos(\theta)$	imag factor, $-\sin(\theta)$
0~90	$\cos(\theta)$	$\sin(\theta)$
90~180	$\sin(\theta)$	$-\cos(\theta)$
180~270	$-\cos(\theta)$	$-\sin(\theta)$
270~360	$-\sin(\theta)$	$\cos(\theta)$

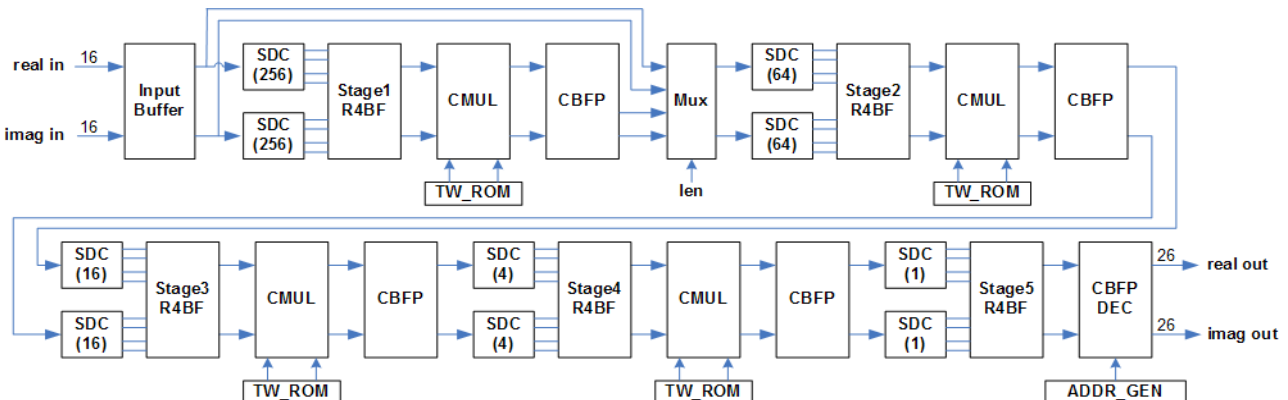


그림 1. FFT의 전체 구조
Fig 1. FFT block diagram

3. Convergent Block Floating Point

복소 승산기에서 출력된 데이터는 CBFP로 입력되어 스케일링과정을 거친다. 이 과정은 CBFP로 입력된 데이터의 연속된 부호비트를 제거함으로써 이루어지며, 제거된 부호비트의 수는 스케일링된 데이터와 함께 다음 stage의 SDC에 보존된다. 예를 들어 “11110110101”이라는 데이터가 CBFP로 입력될 경우 스케일링된 데이터는 “101101010000”이 되며, 제거된 부호비트의 수가 SDC에 함께 저장된다. 한편 각 연산블록의 전체 스케일링 인자가 구해지면, SDC에 저장해 두었던 데이터를 보정하는 과정을 거치며, FFT 연산이 종료되는 시점에서 CBFP 디코더를 통해 전체 데이터의 스케일링 보정과정이 수행된다^[3].

4. 주소발생기

DIF 알고리즘을 사용한 FFT의 경우 CBFP의 구현이 용이한 장점이 있는 반면, 출력데이터의 순서가 다르기 때문에 이를 위한 부가적인 회로가 필요하다. 주소발생 회로는 256, 1024 포인트 FFT를 모두 지원 가능한 구조로 설계되었으며, 그림 3과 같이 내부 10bit 카운트 및 Mux를 사용하여 구현하였다.

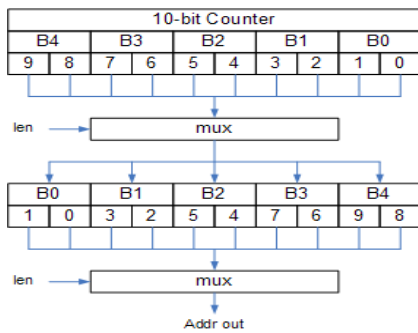


그림 3. 주소발생기 구조
Fig 3. The Address generator diagram

IV. 설계 검증 및 성능 측정

RTL 수준으로 설계된 FFT는 시뮬레이션 및 합성틀을 사용하여 검증과정을 진행하였다. TW_CBFP 스케일링 방법을 적용한 FFT의 성능을 측정하기 위해 Radix-4 SDC 구조의 256, 1024 포인트 고정점 FFT를 설계하였으며, MATLAB을 사용하여 설계된 FFT를 비교 및 평가한 결과, 표 3에서와 같이 동일 구조의 고정점 FFT에 비해 TW_CBFP 구조의 FFT에서 SQNR가 향상되었음을 확인할 수 있었다.

표 3. FFT SQNR의 비교결과

Table 3. The comparison result of FFT SQNR

	고정점 FFT		CBFP FFT	
FFT Point	256	1024	256	1024
SQNR	51.68	50.48	54.18	50.53

V. 결론

본 논문에서는 TW_CBFP 스케일링방법을 적용하여 심리음향모델 연산에 사용 가능한 256, 1024 포인트 FFT를 설계 하였다, 연산효율과 데이터 처리율을 높이기 위해 Radix-4 SDC 파이프라인 구조를 사용하였으며, 동일구조의 고정점 FFT를 사용하여 설계된 FFT의 성능을 평가한 결과, 고정점 FFT에 비해 SQNR이 향상되었음을 확인하였다. 설계된 FFT의 내부 스케일링 비트를 조절할 경우 SQNR이 더욱 향상될 수 있으며, 따라서 설계된 FFT를 심리음향모델 연산에 사용함으로써 오디오 워터마킹 알고리즘의 효율적인 연산을 위한 구조라 할 수 있다.

참고 문헌

- [1] Johnston, J.D, “Transform coding of audio signals using perceptual noise criteria,” IEEE Trans, Selected Areas in Communications, Vol6, no. 2, pp. 314-232, Feb 1988.
- [2] Hyen-O Oh, Joon-Seok Kim, Chang-Jun Song, Young-Cheol Park and Dae-Hee Youn, “Low Power MPEG/AUDIO Encoders Using Simplified Psychoacoustic Model and Fast Bit Allocation,” IEEE Trans, On Consumer Electronics, Vol 47, no. 3, pp. 613-621, Aug 2001.
- [3] 이승기, 양대성, 신경욱, “2단계 수렴 블록 부동점 스케일링 기법을 이용한 8192점 파이프라인 FFT/IFFT 프로세서”, 전자공학회논문지, 제27권 제10C호, 963-972쪽, 2002년 10월
- [4] Alan V. Oppenheim, Ronald W. Scafer, “Discrete-Time Signal Processing 2nd”, Prentice Hall, pp. 629-669, 1999
- [5] Shousheng He, Mats Torkelson “Design and Implementation of a 1024-point Pipeline FFT Processor”, IEEE Conf on Custom Intergrated Circuits Conference, pp. 131-134, May 1998.
- [6] Erling H. WORD, Alvin M. Despain, “Pipeline and Parallel-Pipeline FFT Processors for VLSI Implementations”, IEEE Tran. Computers, Vol. C-33, no 5, pp. 414-426, May 1984.