

MPEG-2 BC/AAC 오디오 공용 합성 필터 구조에 관한 연구

강명수, 박세기, 오신범, 이채욱
대구대학교 정보통신공학부

e-mail : mskang@verygood.daegu.ac.kr, skpark@verygood.daegu.ac.kr, osb58@daegu.ac.kr,
cwlee@daegu.ac.kr

A Study on Common Synthesis Filter Architecture for MPEG-2 BC and AAC Audio

Dept. of Computer and Communication, Daegu University

Abstract

본 논문에서는 MPEG-2 BC와 AAC의 복호화 과정 중 합성 필터링 과정의 알고리즘을 분석하여 공통된 구조로 연산을 수행할 수 있는 공용 합성 필터 구조에 대하여 논하였다. 제안된 공용 합성 필터 구조는 Regressive 구조를 이용하여 MPEG-2 BC와 AAC의 복호화를 효과적으로 공용 수행하도록 하였다. 제안한 구조는 FFT를 사용할 경우에 필요한 전처리 및 후처리 과정을 고려해주지 않아도 되고 복소수 연산이 아닌 실수연산이 되어 하드웨어 구조가 단순하게 된다. 또한 MPEG-2 AAC의 다양한 원도우 변환에도 안정적으로 연산되는 구조임을 확인하였다.

I. 서론

MPEG-1, MPEG-2, MPEG-4, Dolby AC-3와 같은 오디오 압축 표준은 인간 청각계의 지각현상 및 CMFB(Cosine Modulated Filter Bank) 등을 이용한 압축 알고리즘이다. 그러나 각각의 CMFB는 표준안의 계층, 부호화 및 복호화 과정, 블록 길이에 따라 그 형태가 다양하다[1]. 따라서 복수개 표준안의 복호화 과정을 효율적으로 수행하기 위한 합성필터 구조에 대한 관심이 증가하고 있다[2-5]. CMFB는 WOA(Window and Overlapping Addition)와 MCT(Modulated Cosine Transform)의 두단계 과정으로 볼 수 있으며 MCT과정은 DCT 모듈로 구성 되어진다. MCT 과정은 반복적인 많은 연산량으로 인하여 FFT와 같은 고속 알고리즘을 널리 사용한다. 하지만 이 방법은 DCT를 변환영역으로 매핑함으로써 전처리 및 후처리 연산과

정이 추가적으로 요구되고 복소수 연산이 된다.

따라서 본 논문에서는 전처리 및 후처리 연산과정을 없앤 MPEG-2 BC/AAC의 복호화기에서 공용으로 사용될 수 있는 공용 합성 필터 구조를 제안하고자 한다. 공통된 연산 구조는 Regressive 구조를 적용하였고, 그 결과로 서브밴드 필터뱅크의 매트릭성 과정 및 IMDCT 과정을 효율적으로 수행할 수 있으며 VLSI에 적합한 단순한 하드웨어 구조를 가지게 된다.

II. MPEG-2 BC/AAC 합성 필터링 과정

2.1 MPEG-2 BC

MPEG-2 BC에서의 합성 필터링 과정의 매트릭성 과정은 다음과 같다[6].

$$V(i) = \sum_{k=0}^{31} S(k) \cos \left[\frac{\pi}{64} (2k+1)(i+16) \right] \quad (1)$$

for $i = 0, 1, \dots, 63$

여기서 $S(k)$ 는 서브밴드 샘플이다.

Konstantinos가 제안한 알고리즘에 따르면 식(1)은 코사인함수의 대칭성을 이용해서 64개의 $V(i)$ 대신에 다음과 같이 32개의 $V'(i)$ 값만으로 연산이 가능하다 [7].

$$V'(i) = \sum_{k=0}^{31} S(k) \cos \left[\frac{\pi}{64} (2k+1)(i) \right] \quad (2)$$

for $i = 0, 1, \dots, 31$

DCT 연산은

$$Y(k) = \sum_{i=0}^{N-1} y(i) \cos \left[\frac{\pi}{2N} (2i+1)k \right] \quad (3)$$

for $k = 0, 1, \dots, N-1$

로 나타내었을때 Regressive 커널은 다음과 같다[8,9].

$$P_m \sin(x) = \sum_{i=m}^{N-1} y(i) \sin[(i-m+1)x] \quad (4)$$

식 (4)를 전개하여 정리하면 P_m 은 다음과 같다.

$$P_m = y(m) + 2 \cos(x) P_{m+1} - P_{m+2} \quad (5)$$

$m = N-1, \dots, 1, 0$

그리고 식 (3)에 $\sin(x)$ 을 양변에 곱한 후 정리하면

$$Y(k) = y(0) \cos\left(\frac{\pi k}{2N}\right) + \cos\left(\frac{3\pi k}{2N}\right) P_1 - \cos\left(\frac{\pi k}{2N}\right) P_2 \quad (6)$$

이 된다.

그림 1은 식 (6)에 의한 Chau의 Regressive 구조를 나타내었다.

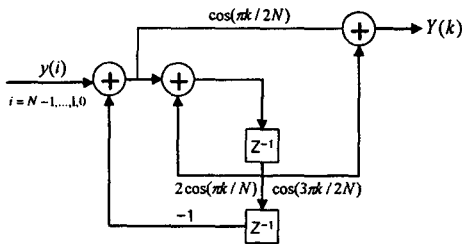


그림 1. Chau의 Regressive 알고리즘 구조

그리고 식(5), (6)의 관계를 이용한 Chau의 Regressive 구조는 다음 그림과 같다.

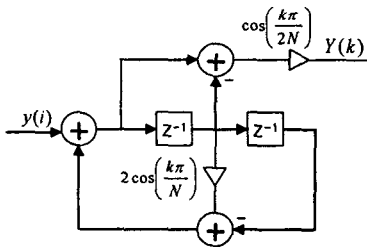


그림 2. Chau의 Regressive 알고리즘 구조

여기서 $N=32$ 이다. 식 (2)에서 $V'(i)$, $S(k)$ 는 식 (3)의 $Y(k)$, $y(i)$ 에 해당된다.

2.2 MPEG-2 AAC

MPEG-2 AAC 복호화기에서 사용하는 IMDCT는 다음과 같다[10].

$$x(n) = \sum_{k=0}^{M-1} X(k) \cos \left[\cos \frac{\pi}{M} \left(k + \frac{1}{2} \right) \left(n + \frac{M+1}{2} \right) \right] \quad (7)$$

$n = 0, 1, \dots, N-1$

여기서

$$\begin{cases} n = \text{샘플 인덱스} \\ k = \text{스펙트럼 계수 인덱스} \\ N = \text{윈도우 길이} \\ M = N/2 \end{cases}$$

이다.

IMDCT연산을 위한 식 (7)에 대한 Nikolajevic Regressive 알고리즘의 구조를 그림 3에 나타내었다 [11].

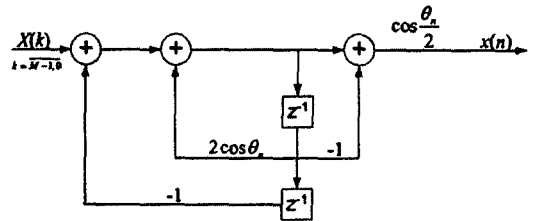


그림 3. Nikolajevic의 Regressive 알고리즘 구조

여기서 $\theta_n = \left(n + \frac{M+1}{2} \right) \frac{\pi}{M}$ 이다.

III. 제안하는 공용 합성 필터 구조

앞에서 살펴본 MPEG-2 BC/AAC의 Regressive 알고리즘은 입력값 하나에 대하여 입력 개수만큼 순환을 계속하여 곱셈과 덧셈을 한 후 값이 출력되게 된다. 그림 2, 3에서 보는 것처럼 MPEG-2 BC/AAC의 Regressive 알고리즘의 구조는 지연소자 2개, 덧셈소자 3개, 곱셈소자 2개로 동일함을 알 수 있다. 또한, 동일한 데이터 경로를 가진다. 그러나 곱해지는 코사인 계수값은 MPEG-2 BC와 MPEG-2 AAC에 따라 다르다. 그리고 Regressive 공용 합성 필터 뱅크 구조를 설계할 때 MPEG-2 AAC 같은 경우 많은 연산량을 가지기 때문에 공용 합성 필터 구조를 실제로 하드웨어로 구현하기 위해서는 연산량 및 계산의 복잡도가 문제가 된다.

한편 MCT연산은 Regressive 구조의 MPEG-2 BC/AAC 공용 합성 필터링 구조인 그림 2, 3처럼 입력은 역순으로 들어가고 출력은 순차적으로 나오게 되는 구조를 가지게 되며 다음으로 WOA과정이 필요하다. 즉 MPEG-2 BC/AAC에 따라 적절한 윈도우를 취한 후에 MPEG-2 BC는 중첩 가산 방법을 이용하여 32개씩 36번 반복하여 1152개의 PCM sample을 출력하고 MPEG-2 AAC도 중첩 가산 방법을 이용하여 long window의 경우는 1024개, short window의 경우는 128개씩 8번 반복하여 1024개의 PCM sample을 출력한다. 그림 3에서 $x(n)(n=0, 1, \dots, N-1)$ 값은 코

사인 계수의 대칭성으로 인하여 절대치가 동일한 값을 중복 계산하는 구조를 가지고 있다. 따라서 불필요한 연산이 소요됨으로 많은 연산시간의 문제점이 있다.

본 논문에서는 이러한 점을 고려하여 다음과 같이

$$x(M-1-n) = -x(n), n = 0, 1, \dots, \frac{N}{4}-1 \quad (8)$$

$$x(N-1-n) = x(M+n), n = 0, 1, \dots, \frac{N}{4}-1 \quad (9)$$

$x(n)$ 의 관계를 적용하면 $\frac{N}{4}$ 길이의 $x(n)$ 및 $x(M+n)$ 을 각각 구하면 식 (8), (9)에 의해 $x(n)$ 을 구할 수 있으므로 IMDCT의 Regressive 구조를 그림 4와 같이 2단으로 구성하면 병렬로 연산되는 구조가 되어 연산시간을 줄일 수 있다. 2단으로 구성하였을 경우 지연소자 4개, 덧셈소자 6개, 곱셈소자 4개가 필요하다.

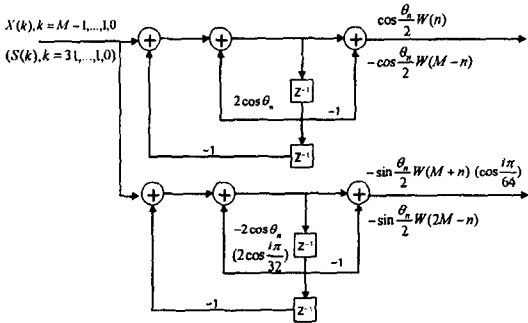


그림 4. 제안하는 MPEG-2 BC/AAC 합성필터 구조

그림 4에서 윗부분이 $x(n)(n = 0, 1, \dots, \frac{N}{4}-1)$ 의 연산부이고, 아랫부분이 $x(M+n)(n = 0, 1, \dots, \frac{N}{4}-1)$ 에 대한 연산부이다. 그리고 $W(\cdot)$ 은 윈도우를 취하기 위한 계수이다. 코사인 계수와 윈도우 계수를 미리 곱한 값을 메모리에 저장해두고 사용하도록 한다. 아랫부분의 $x(M+n)$ 에 필요한 코사인 계수값인 $2\cos\theta_{M+n}$ 와 $\cos\frac{\theta_{M+n}}{2}$ 는 다음과 같이 나타낼 수 있다.

$$\begin{aligned} 2\cos\theta_{M+n} &= 2\cos\left[\left(M+n+\frac{M+1}{2}\right)\frac{\pi}{M}\right] \\ &= 2\cos(\pi)\cos\left[\left(n+\frac{M+1}{2}\right)\frac{\pi}{M}\right] \quad (10) \\ &= -2\cos\theta_n \end{aligned}$$

$$\begin{aligned} \cos\frac{\theta_{M+n}}{2} &= \cos\left[\left(M+n+\frac{M+1}{2}\right)\frac{\pi}{2M}\right] \\ &= -\sin\left(\frac{\pi}{2}\right)\sin\left[\left(n+\frac{M+1}{2}\right)\frac{\pi}{2M}\right] \quad (11) \\ &= -\sin\left(\frac{\theta_n}{2}\right) \end{aligned}$$

식 (6)에서 $x(M+n)$ 에 필요한 코사인 계수값은 같은 클럭에서 $x(n)$ 의 연산시 필요한 코사인 계수값의 부호만 반전시켜주면 된다. 그 결과로 메모리량을 줄일 수 있다. 그리고 Overlap-Add 과정을 위하여 아랫부분의 계산 결과값은 1K 워드 메모리에 저장하도록 하고 윗부분의 계산값과 1K 워드 메모리에 저장된 값을 더해 Overlap-Add 과정을 끝낸후 PCM sample 값을 출력한다.

그리고 MPEG-2 BC의 Regressive 과정을 그림 4에 적용할 경우 아랫부분에서 $V'(i)(i = 0, 1, \dots, 31)$ 를 연산하도록 한다. 연산한 결과인 $V'(i)$ 데이터를 적절하게 배치하면 원하는 결과인 $V(i)(i = 0, 1, \dots, 63)$ 를 구할 수 있다. 이때 곱셈에 필요한 코사인 계수값인 $2\cos\left(\frac{i\pi}{32}\right)$ 및 $\cos\left(\frac{i\pi}{64}\right)$ 은 메모리에서 읽어오도록 한다.

Window Overlap-Add 과정은 MPEG-2 AAC와는 다르다. 하지만 MPEG-2 AAC의 WOA 과정에 필요한 1K 워드 메모리내에서 연산이 가능하고 $\cos\left(\frac{i\pi}{64}\right)$ 와 합성 윈도우 계수값을 미리 곱한 결과를 사용하여 연산을 하면 곱셈 횟수를 줄일 수 있다. 그리고 Overlap-Add 연산을 거친 후 32개의 오디오 샘플값을 출력한다.

따라서, 그림 4의 구조는 MPEG-2 BC/AAC의 합성 필터링 과정에서 공통적으로 Regressive 구조를 가지게 되어 FFT를 사용한 방법처럼 전처리 및 후처리 과정을 고려하지 않는 실수연산을 함으로써 효율적인 공용 합성 필터 구조의 구현이 가능해진다. 다만, MPEG-2 BC와 AAC의 WOA 과정은 서로 상이하므로 별개의 연산 구조를 가지게 된다.

IV. 실험결과

공용 합성 필터 구조의 검증을 위하여 Matlab을 사용하여 컴퓨터 시뮬레이션 하였다. MPEG-2 AAC의 합성 필터 부분을 부동소수점 연산을 이용하여 표준으로 계산한 결과와 2단 Regressive 구조로 계산한 결과를 그림 5에 나타내었다.

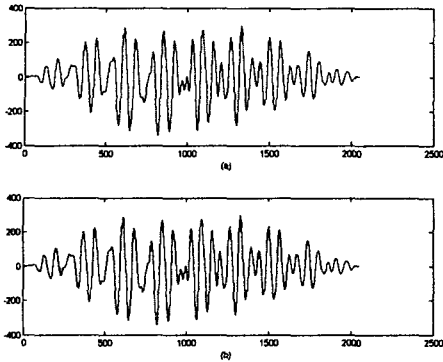


그림 5. Matlab을 이용한 시뮬레이션 결과

그림 5는 16bit, 44.1kHz로 샘플링된 타악기 Cowbell 에 대한 시뮬레이션이며 1채널에 대한 1024 샘플의 복원된 2048개의 값을 나타내었다. 표준으로 구한 (a) 그림과 2단 Regressive 구조로 연산된 (b) 그림에서 성능의 차이가 없음을 알 수 있다. MPEG-2 BC 또한 동일한 값을 확인하였다.

V. 결론

본 논문에서는 MPEG-2 BC와 MPEG-2 AAC 오디오 복호화기에 공용으로 사용될 수 있는 공용 합성필터의 구조를 제안하였다.

MPEG-2 BC와 AAC의 합성 필터링 과정은 Regressive 구조로 구현될 수 있다. 제안한 구조는 코사인 계수의 대칭성을 이용하여 MPEG-2 AAC Regressive 구조를 변형한 2단 Regressive 구조를 사용하였고 여기에 MPEG-2 BC의 서브밴드 필터링도 연산되도록 하여 MPEG-2 BC와 MPEG-2 AAC 합성 필터링 과정 모두 효과적으로 처리할 수 있도록 하였다.

앞으로의 과제는 부동소수점에 대한 결과를 바탕으로 고정소수점에서의 오차에 대한 값을 최소화하는 것과 제안된 알고리즘을 실제 복호화기에 적용하여 설계하는 연구가 요구된다.

참고문헌

[1] Ch.M. Liu, W.Ch. LEE, A unified fast algorithm for cosine modulated filter banks in current audio coding standards, J. Audio Eng. Soc. 47(12)(December 1999) 1061-1075
 [2] Winnie Lau and Alex Chwu, "A Common

Transform Engine for MPEG & AC3 Audio Decoder", IEEE Trans. on Consumer Electronics, Vol. 43, No. 3, pp. 559-566, Aug. 1997.

- [3] Y. Jhung and S. Park, "Architecture of Dual Mode Audio Filter for AC-3 and MPEG", IEEE Trans. on Consumer Electronics, Vol. 43, No. 3, pp. 575-585, Aug. 1997.
 [4] W. S. Ko, S. K. Yoo, S.W Park, J. S. Kim and D. H. Youn, "A vlsi implementation of dual AC-3 and MPEG-2 audio Decoder", IEEE Trans. on Consumer Electronics, Vol. 44, No. 3, pp. 872-877, Aug. 1998.
 [5] 고우석, 유선국, 박성욱, 정남훈, 김준석, 이근섭, 윤대희, "AC-3와 MPEG-2 오디오 디코더 공용 합성 필터의 설계", 신호처리 합동 학술대회, pp. 299-302, Sep. 1997.
 [6] ISO/IEC JTC1/SC29/WG11 No.703 "Generic Coding of Moving Pictures and Associated Audio - CD 13813-3 (Part3. MPEG-Audio)", Mar. 1994.
 [7] K. Konstantinides, "Fast Subband Filtering in MPEG Audio Coding", IEEE Signal Processing, Letters vol.1, no.2, pp. 26-28, 1994.
 [8] L.-P. Chau and W.-C. Siu, "Recursive algorithm for the discrete cosine transform with general lengths", ELECTRONICS LETTERS, VOL. 30, NO. 3, Feb 1994.
 [9] Lap-Pui CHAU, Wan-Chi SIU, "Recursive Algorithm for the realization of the Discrete Cosine Transform", ISCAS 2000 - IEEE International Symposium on Circuits and Systems, May 28-31, 2000.
 [10] ISO/IEC JTC1/SC29/WG11 No.1650 "IS 13818-7 (MPEG-2 Advanced Audio Coding, AAC)", Apr., 1997..
 [11] Vladimir Nikolajevic, "Computation of Forward and Inverse MDCT Using Clenshaw's Recurrence Formula", IEEE Trans, SIGNAL PROCESSING, VOL. 51, NO.5, May 2003.