

Recursive 구조를 이용한 MPEG-2 AAC 복호화기의 필터뱅크 구현

정희원 박 세 기*, 강 명 수*, 오 신 범*, 이 채 욱*

Architecture Design for MPEG-2 AAC Filter bank Decoder using Recursive Structure

Myung-Soo Kang*, Se-Ki Park*, Shin-Bum Oh*, Chae-Wook Lee** *Regular Members*

요 약

MPEG-2 AAC(Advanced Audio Coding)는 높은 압축율과 고음질의 특성을 갖는 멀티채널 오디오 압축 표준으로 알려져 있는데, 매우 낮은 비트율에서 방송 음질 수준의 오디오를 제공하기 위해서 고해상도 필터뱅크(high-resolution filter bank), 예측기법(prediction), 허프만 부호화(Huffman coding) 등을 결합하여 사용한다. MPEG-2 AAC 필터뱅크에서 부호화기와 복호화기에서 각각 수행하는 MDCT(Modified Discrete Cosine Transform)/ IMDCT(Inverse MDCT) 연산은 특히 많은 연산을 필요로 한다.

본 논문에서는 MPEG-2 AAC 실시간 복호화 시스템을 하드웨어로 구현시 IMDCT의 연산에 적합한 2단 IMDCT의 Recursive 구조를 제안한다. 그리고 메모리와 연산속도, 하드웨어의 복잡도를 기존에 제안된 알고리즘과 비교하였다.

Key Words : MPEG-2 AAC, IMDCT, recursive structure

ABSTRACT

MPEG-2 Advanced Audio Coding(AAC) is widely used in the multi-channel audio compression standards. And it combines high-resolution filter bank, prediction techniques, and Huffman coding algorithm to achieve the broadcast-quality audio level at very low data rates. The forward and inverse modified discrete transforms which are operated in the encoder and the decoder of the filter bank need many computations.

In this paper, we propose suitable recursive structure at IMDCT processing for MPEG-2 AAC real-time decoder. We confirm the memory, the computation speed and complexity of the proposed structure.

I. 서 론

MPEG-2 AAC 필터뱅크에서 낮은 비트율로 고음질을 유지하기 위하여 MDCT/IMDCT 기법을 사용한다. MDCT/IMDCT에 의한 필터뱅크는 뛰어난 스펙트럼 분해능과 시간 영역에서의 에일리어싱 제거 기법 (TDAC:Time Domain Aliasing Cancellation)

을 기반으로 하는 분석 및 합성의 기능을 수행한다. 따라서 많은 오디오 표준 및 제품에서 핵심적인 요소로 채택되고 있다. 그러나 MDCT/IMDCT의 필터뱅크는 많은 연산량과 복잡도를 가지기 때문에, 실제로 하드웨어로 구현하기 위해서는 연산량과 계산의 복잡도를 감소시켜야한다. MDCT/IMDCT의 효율적인 구현을 위해서 Chiang와 Liu가 제안한

* 대구대학교 정보통신공학과 디지털신호처리 연구실(cwlee@daegu.ac.kr)

논문번호 : 040001-0102, 접수일자 : 2004년 2월 8일

** 본 연구는 산업자원부 지역혁신인력양성사업의 연구결과로 수행되었습니다.

Regressive 알고리즘과 Nikolajevic와 Fettweis의 Recursive 알고리즘이 많이 사용되고 있다[2][3]. 두 알고리즘은 정현파의 성질을 이용하여 일반적인 길이의 MDCT/IMDCT를 연산하도록 제안한 알고리즘으로 구조가 단순함으로 VLSI로 쉽게 구현할 수 있다. 또한 MPEG 오디오 표준에서 제안하는 다양한 window switching이 포함 된 변환 길이의 MDCT/IMDCT에 대해서도 안정적으로 연산된다 [2][3]. 그러나 MPEG-2 AAC 실시간 복호화 시스템에 고음질 오디오를 실시간으로 처리할 경우 기존의 Recursive 구조로는, 많은 연산시간으로 인해 적합하지 않다.

본 논문에서는 MPEG-2 AAC 실시간 복호화 시스템을 VLSI로 구현시 적합한 Recursive 구조를 제안하고, 메모리와 연산속도, 하드웨어의 복잡도를 기존에 제안된 알고리즘과 비교하였다.

II. IMDCT의 Recursive 알고리즘

AAC 복호화기에서 사용하는 필터뱅크 IMDCT의 수식은 다음과 같이 계수 $X(k)$ 로부터 $x(n)$ 을 구하는 것이다[1].

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

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

단,

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

IMDCT 연산을 위한 Recursive 표현은 다음과 같이 몇 가지 알고리즘으로 나타낼 수 있다.

1. Chiang의 Regressive 알고리즘

IMDCT 연산을 위한 Chiang의 Regressive 알고리즘은 아래의 수식에 의해 증명 된다.

IMDCT 연산식인 식 (1)을 다음과 같이 변형시킨다.

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

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

$V_k(\theta_n) \theta_n$ 과 을 다음과 같이 각각 정의하면,

$$\theta_n = \left(n + \frac{M+1}{2}\right) \frac{\pi}{M} \quad (3)$$

$$V_k(\theta_n) = \frac{\sin\left(\left(k + \frac{1}{2}\right)\theta_n\right)}{\sin\left(\frac{\theta_n}{2}\right)} \quad (4)$$

식 (2)를 θ_n 과 $V_k(\theta_n)$ 을 사용하여 다음과 같이 쓸 수 있으며,

$$x(n) = (-1)^n \sin\left(\frac{\theta_n}{2}\right) \sum_{k=0}^{M-1} X(M-1-k) V_k(\theta_n) \quad (5)$$

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

$$V_{k+1}$$

과 V_{k-1} 의 합은 다음과 같다.

$$\begin{aligned} V_{k+1}(\theta_n) + V_{k-1}(\theta_n) &= \\ \frac{\sin\left(\left(k+1 + \frac{1}{2}\right)\theta_n\right)}{\sin\left(\frac{\theta_n}{2}\right)} + \frac{\sin\left(\left(k-1 + \frac{1}{2}\right)\theta_n\right)}{\sin\left(\frac{\theta_n}{2}\right)} &= \\ = 2 \cos(\theta_n) \cdot \frac{\sin\left(\left(k + \frac{1}{2}\right)\theta_n\right)}{\sin\left(\frac{\theta_n}{2}\right)} &= \\ = 2 \cos(\theta_n) \cdot V_k(\theta_n) \end{aligned} \quad (6)$$

따라서 $V_k(\theta_n)$ 는 식 (6)에 의해 다음과 같은 관계식이 성립한다.

$$V_{k+1}(\theta_n) = 2 \cos(\theta_n) V_k(\theta_n) - V_{k-1}(\theta_n) \quad (7)$$

식(4)로부터 V의 초기값을 각각 구할 수 있다.

$$\begin{aligned} V_0(\theta_n) &= 1 \\ V_{-1}(\theta_n) &= -1 \end{aligned} \quad (8)$$

$Q_m(n)$ 을 식 (9)과 같이 정의하면,

$$Q_m(n) = \sum_{k=0}^m X(m-k) V_k(\theta_n) \quad (9)$$

식 (7)에 의해 식 (10)이 성립한다.

$$\begin{aligned} Q_m(n) &= X(m) + \sum_{k=0}^{m-1} X(m-1-k) V_{k+1}(\theta_n) \\ &= X(m) + 2 \cos(\theta_n) Q_{m-1}(n) + X(m-1) - Q_{m-2}(n) \end{aligned} \quad (10)$$

식 (10)에서 보는 것과 같이 $Q_m(n)$ 은 $Q_{m-1}(n)$, $X(m)$, $X(m-1)$ 의해 구할 수가 있다.

IMDCT 연산을 위한 Chiang과 Liu가 제안한 Regressive 구조는 식 (5), 식 (10)에 의해 그림 1 과 같이 구성할 수 있다[2].

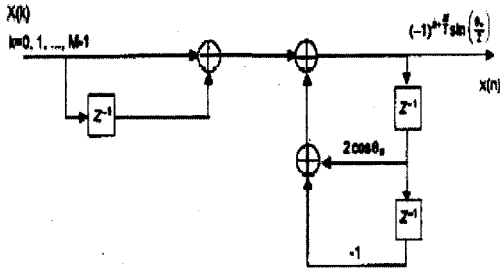


그림 1. Chiang와 Liu의 Regressive 구조

Chiang과 Liu가 제안한 구조의 Regressive 부분에 대해 1개의 Multiplier, 3개의 Adder 그리고 3개의 Delay로 구성되어진다. 연산순서는 입력 $X(k)$ 가 $X(0)$ 부터 $X(M-1)$ 까지 순차적으로 입력되며, $X(k)$ 의 k의 값이 하나씩 증가 될 때마다 식 (10)에 의한 순환연산을 한 후, $X(M-1)$ 이 입력 되면 $(-1)^n \sin(\frac{\theta}{2})^n$ 을 곱하여 $x(n)$ 이 출력 된다.

따라서 연산시간은 1번의 순환 연산 과정을 1Cycle이라 하면 $x(n)$ ($n=0, 1, \dots, N-1$)을 구하기 위해서는 $N^2/2$ Cycle이 필요하다. 이 구조에서 필요한 계수들은 $2 \cos \theta_n$

$$(-1)^n \sin \frac{\theta}{2}^n \text{이다.}$$

2. Nikolajevic의 Recursive 알고리즘

IMDCT 연산을 위한 Nikolajevic의 알고리즘은 다음과 같이 나타낼 수 있다[3].

식 (3)의 θ_n 을 이용하여, $V_m \sin \theta_n$ 을 다음과 같이 표현한다.

$$V_m \sin \theta_n = \sum_{k=m}^{M-1} X(k) \sin(k-m+1)\theta_n \quad (11)$$

$$\begin{aligned} &= X(k) \sin \theta_n + \sum_{k=m+1}^{M-1} X(k) \sin(k-m+1)\theta_n \\ &= X(m) \sin \theta_n + 2 \cos \theta_n V_{m+1} \sin \theta_n - V_{m+2} \sin \theta_n \end{aligned}$$

식 (11)에 의해 다음 식을 유도할 수 있다.

$$V_m = X(m) + 2 \cos \theta_n V_{m+1} - V_{m+2} \quad m = M-1, \dots, 1, 0 \quad (12)$$

한편 식 (1)의 양변에 $\sin \theta_n$ 을 곱하면 식 (13)과 같이 된다.

$$\begin{aligned} x(n) \sin \theta_n &= \sum_{k=0}^{M-1} X(k) \cos((k+\frac{1}{2})\theta_n) \sin \theta_n \\ &= X(0) \cos \frac{\theta}{2} \sin \theta_n + \cos \frac{3\theta}{2} V_1 \sin \theta_n - \cos \frac{\theta}{2} V_2 \sin \theta_n \end{aligned} \quad (13)$$

식 (13)으로부터 다음 식을 유도할 수 있다.

$$\begin{aligned} x(n) &= X(0) \cos \frac{\theta}{2} + \cos \frac{3\theta}{2} V_1 - \cos \frac{\theta}{2} V_2 \\ &= (X(0) - V_2) \cos \frac{\theta}{2} + \cos \frac{3\theta}{2} V_1 \end{aligned} \quad (14)$$

m=0일 경우 식 (12)로부터 식 (15)가 성립한다.

$$X(0) - V_2 = V_0 - 2 \cos \theta_n V_1 \quad (15)$$

식 (14)와 식 (15)에 의해 $x(n)$ 은 식 (16)과 같이 표현된다.

$$x(n) = \cos \frac{\theta}{2} (V_0 - V_1) \quad (16)$$

식 (12),(16)에 의한 Nikolajevic의 IMDCT의 Recursive 구조는 그림 2와 같다[3].

Nikolajevic와 Fettweis의 구조의 Recursive 부분은 1개의 Multiplier, 3개의 Adder 그리고 2개의 Delay로 구성되어진다. $X(k)$ 가 역순서로 입력되는 부분과, 식 (12)에 의한 Recursive 부분, 식(16)에 의한 출력부로 구성된다. 그림 2의 연산 과정을 알아보면 $X(k)$ 가 입력되어 식 (12)에 의해 순환 연산을 한 후, 식 (16)에 의해 $x(n)$ 이 출력된다.

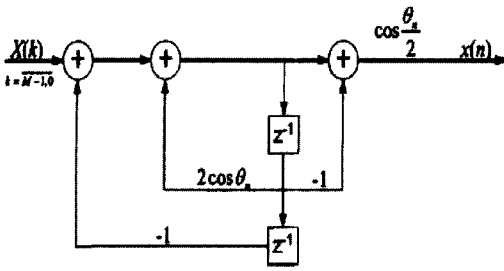


그림 2. Nikolajevic와 Fettweis의 Recursive 구조

$x(n)$ ($n=0, 1, \dots, N-1$)을 구하기 위한 연산시간은 Chiang와 Liu의 Regressive 알고리즘과 같이 $N^2/2\text{Cycle}$ 이 필요하다. 이 구조에서 필요한 계수들은 $2\cos\theta$, $\cos\frac{\theta}{2}$ 이다.

3. Modified Regressive 알고리즘

Modified Regressive 알고리즘은 Chiang의 Regressive 알고리즘에 기반으로 하여 MPEG-2 AAC의 실시간 복호화기의 IMDCT 연산에 응용하기 위하여 제안된 구조이다[4]. 식 (1)을 Chiang의 Regressive 알고리즘에서와 같이 식 (2)와 같이 변형시킨다. 식 (2)를 계수 $X(k)$ 를 홀수부와 짝수부로 분리하여 정리하면 다음과 같다.

$$x(n) = (-1)^n \left\{ \sum_{k=0}^{M/2-1} X(M-2k-2) \sin\left(\frac{\pi}{M}(2k+\frac{3}{2})(n+\frac{M+1}{2})\right) + \sum_{k=0}^{M/2-1} X(M-2k-1) \sin\left(\frac{\pi}{M}(2k+\frac{1}{2})(n+\frac{M+1}{2})\right) \right\} \quad (17)$$

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

다음으로 식 (18), (19)와 같이 $N/2$ 길이의 $x(n)$ 과 $x(n+\frac{N}{2})$ 두 부분으로 나눈다.

$$x(n) = (-1)^n \left\{ \sum_{k=0}^{M/2-1} X(M-2k-2) \sin\left(\frac{\pi}{M}(2k+\frac{3}{2})(n+\frac{M+1}{2})\right) + \sum_{k=0}^{M/2-1} X(M-2k-1) \sin\left(\frac{\pi}{M}(2k+\frac{1}{2})(n+\frac{M+1}{2})\right) \right\} \quad (18)$$

$n=0, 1, \dots, N/2-1$

$$x(n+\frac{N}{2}) = \sum_{k=0}^{M/2-1} -X(2k) \sin\left(\frac{\pi}{M}(2k+\frac{1}{2})(n+\frac{M+1}{2})\right) + \sum_{k=0}^{M/2-1} X(2k+1) \sin\left(\frac{\pi}{M}(2k+\frac{3}{2})(n+\frac{M+1}{2})\right) \quad (19)$$

$n=0, 1, \dots, N/2-1$

$x(n)$ 은 $x'(n) + x''(n)$ 과 같이 표현할 수 있으며. $x'(n), x''(n)$ 은 각각 식 (20), (21)이다.

$$x'(n) = (-1)^n \sum_{k=0}^{M/2-1} X(M-2k-2) \times \sin\left(\frac{\pi}{M}(2k+\frac{3}{2})(n+\frac{M+1}{2})\right) \quad (20)$$

$$x''(n) = (-1)^n \sum_{k=0}^{M/2-1} X(M-2k-1) \times \sin\left(\frac{\pi}{M}(2k+\frac{1}{2})(n+\frac{M+1}{2})\right) \quad (21)$$

$x'(n)$ 과 $x''(n)$ 각각의 Regressive 구조 식 (22), 식 (23)과 같이 유도할 수 있다.

$$Q_m(n) = X(2m) \left(4\cos^2\frac{\theta}{2} - 1 \right) + X(2m+1) + 2\cos(2\theta)Q_{m-1}(n) - Q_{m-2}(n) \quad (22)$$

$$R_m(n) = X(2m-1) \left(4\cos^2\frac{\theta}{2} - 1 \right) + X(2m+1) + 2\cos(2\theta)R_{m-1}(n) - R_{m-2}(n) \quad (23)$$

$x(n+\frac{N}{2})$ 도 마찬가지로 $x'(n+\frac{N}{2}) + x''(n+\frac{N}{2})$ 으로 표현할 수 있다.

$$x'(n+\frac{N}{2}) = \sum_{k=0}^{M/2-1} -X(2k) \sin\left(\frac{\pi}{M}(2k+\frac{1}{2})(n+\frac{M+1}{2})\right) \quad (24)$$

$$x''(n+\frac{N}{2}) = \sum_{k=0}^{M/2-1} X(2k+1) \sin\left(\frac{\pi}{M}(2k+\frac{3}{2})(n+\frac{M+1}{2})\right) \quad (25)$$

$x'(n+\frac{N}{2})$ 은 식 (26), $x''(n+\frac{N}{2})$ 은 식 (27)과 같이 Regressive 구조로 된다.

$$U_m(n) = X(M-2m) \left(4\cos^2\frac{\theta}{2} - 1 \right) + X(M-2m-2) + 2\cos(2\theta)U_{m-1}(n) - U_{m-2}(n) \quad (26)$$

$$V_m(n) = X(M-2m-1) \left(4\cos^2\frac{\theta}{2} - 1 \right) + X(M-2m+1) + 2\cos(2\theta)V_{m-1}(n) - V_{m-2}(n) \quad (27)$$

위 수식에 의해 유도된 Regressive 수식을 통해 구현한 구조가 그림 3이다.

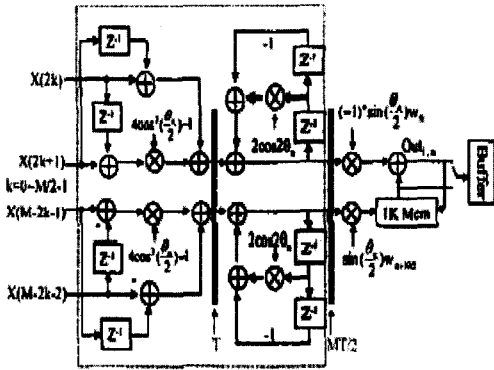


그림 3. Modified Regressive 구조

그림에서 보듯이 Regressive 부분은 4개의 Multiplier, 10개의 Adder 그리고 8개의 Delay로 구성되어진다. 연산순서는 입력 $X(k)$ 를 4부분으로 나누어 2부분은 순방향 입력, 나머지는 역입력으로 나누고 다시 홀수번째와 짝수번째로 나누어 입력한 후 각각의 Regressive 구조를 통해 연산된다. 따라서 하드웨어적 구조는 Chiang의 Regressive 구조나 Nikolajevic의 Recursive 구조에 비해 복잡하지만, 연산시간과 메모리량을 줄일 수 있다. 연산에 필요

한 계수들은 $2 \cos(2\theta_n)$, $4 \cos^2\left(\frac{\theta_n}{2}\right) - 1$, $\sin\left(\frac{\theta_n}{2}\right)$ 이다.

III. 제안하는 IMDCT의 Recursive 구조

IMDCT 연산을 위한 Chiang의 Regressive 구조나 Nikolajevic의 Recursive 구조는 연산 방식이 간단하여 하드웨어로 구현 시 간단하다. 하지만 많은 연산시간의 문제점으로 MPEG-2 AAC의 실시간 복호화기에 적합하지 않다. 그리고 Modified Regressive 알고리즘은 연산시간과 메모리의 문제점을 개선하였지만, 하드웨어로 구현시 많은 복잡도로 인해 어려움이 있다.

본 논문에서는 기존의 제안된 알고리즘의 문제점을 해결한 2단 IMDCT Recursive 구조를 제안하고자한다. IMDCT의 출력 $x(n)$ ($n=0, 1, \dots, N-1$)은 다음 같은 관계식이 성립한다.

$$x\left(\frac{N}{2} - 1 - n\right) = -x(n) \quad (28)$$

$$n = 0, 1, \dots, \frac{N}{4} - 1$$

$$x(N-1-n) = x\left(\frac{N}{2} + n\right) \quad (29)$$

$$n = 0, 1, \dots, \frac{N}{4} - 1$$

식 (28), (29)는 다음에 의해 증명 할 수 있다.

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

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

$$= - \sum_{k=0}^{M-1} X(k) \cos\left(\frac{\pi}{M}\left(k + \frac{1}{2}\right)\left(n + \frac{M+1}{2}\right)\right) \quad (30)$$

$$= -x(n)$$

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

$$= \sum_{k=0}^{M-1} X(k) \cos\left(\frac{\pi}{M}\left(k + \frac{1}{2}\right)\left(4M - \frac{3M}{2} - n - \frac{1}{2}\right)\right)$$

$$= \sum_{k=0}^{M-1} X(k) \cos\left(\frac{\pi}{M}\left(k + \frac{1}{2}\right)\left(n + \frac{3M+1}{2}\right)\right) \quad (31)$$

$$= x\left(\frac{N}{2} + n\right)$$

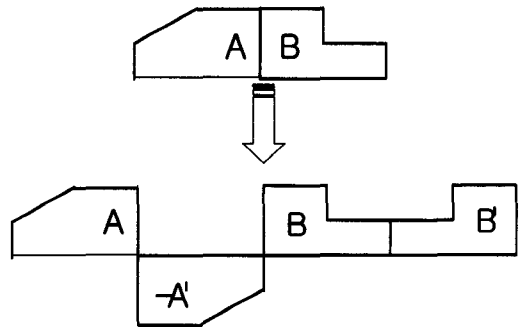


그림 4. IMDCT의 출력

그림 4에서 보는 것처럼 $N/4$ 길이의 $x(n)$ 과 $x\left(\frac{N}{2} + n\right)$ 을 각각 구하면 식 (28),(29)에 의해 $x(n)$ ($n=0, 1, \dots, N-1$)을 구할 수 있다. recursive 구조를 이용하여 $x(n)$ ($n=0, 1..N/4-1$)을 구하기 위해서는 $\cos \theta_n$ 과 $\cos \frac{\theta_n}{2}$ 의 계수가 필요하며, $x\left(\frac{N}{2} + n\right)$ ($n=0, 1, \dots, N/4-1$)을 구하기 위해서는 $\cos \theta_{M+n}$ 와 $\cos \frac{\theta_{M+n}}{2}$ 의 계수가 필요하다.

계수 $\cos \theta_{M+n}$ 과 $\cos \frac{\theta_{M+n}}{2}$ 은 식 (32)와 식 (33)에 의해 구한다.

$$\begin{aligned} \cos \theta_{M+n} &= \cos \left(M+n + \frac{M+1}{2} \right) \frac{\pi}{M} \\ &= \cos \pi \cos \left(n + \frac{M+1}{2} \right) \frac{\pi}{M} \quad (32) \\ &= -\cos \theta_n \end{aligned}$$

$$\begin{aligned} \cos \frac{\theta_{M+n}}{2} &= \cos \left(M+n + \frac{M+1}{2} \right) \frac{\pi}{2M} \\ &= -\sin \frac{\pi}{2} \sin \left(n + \frac{M+1}{2} \right) \frac{\pi}{2M} \quad (33) \\ &= -\sin \frac{\theta_n}{2} \end{aligned}$$

식 (32)에서 $\cos \theta_n$ 의 부호를 바꾸는 것으로 $\cos \theta_{M+n}$ 의 계수가 구해진다. 따라서 IMDCT의 Recursive 구조로 구현 시 메모리량을 줄일 수 있다. 연산알고리즘을 단계별로 정리하면 다음과 같은 과정을 거친다.

[Step1] 길이 N 의 $x(n)$ 는 의해 길이 $N/4$ 의 $x(n)$ 과 $x(\frac{N}{2} + n)$ 에 대한 연산으로 구할 수 있다.

[Step2] $x(n)$ 과 $x(\frac{N}{2} + n)$ 을 구하기 위하여 IMDCT의 Recursive 구조를 사용할 경우, 필요한 계수는 식 (32),(33)에 의해 $N/4$ 길이의 $\cos \theta_n$, $\cos \frac{\theta_n}{2}$, $-\sin \frac{\theta_n}{2}$ 이 된다.

[Step3] IMDCT의 Recursive 구조를 $N/4$ 길이의 $x(n)$ 과 $x(\frac{N}{2} + n)$ 이 병렬로 연산되는 구조를 그림 5와 같이 구성 할 수 있다.

그림 5에서 윗부분이 $x(n)$ 의 연산부이고, 아랫부분이 $x(\frac{N}{2} + n)$ 에 대한 연산부이다. 또한 연산 블록에 따라 A, B, C의 세 부분으로 나눌 수 있는데, A는 식(12)에 의해 순환연산 되는 부분으로 식 (32)에 의해 $N/4$ 길이의 $2\cos \theta_n$ 의 계수가 필요하다.

B는 A에서 연산 된 하나의 출력에 두개의 적당한 Window 계수와 출력단 계수를 곱하는 부분으,

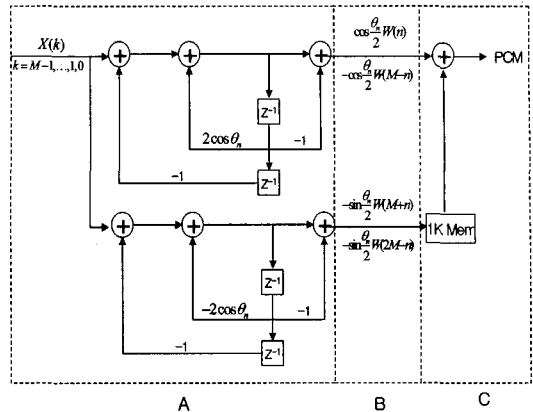


그림 5. 2단 IMDCT Recursive 구조

로 출력단의 계수와 Window 계수는 미리 곱하여 메모리에 저장하여 둔다.

C는 Overlap-Add 과정으로 그림 5의 윗부분이 ($n=0, 1, \dots, N-1$)의 앞부분으로 나타나고 아랫부분은 $x(n)$ ($n=0, 1, \dots, N-1$)의 뒷부분에 해당된다. 따라서 첫 번째 B의 $x(n)$ ($n=0, 1, \dots, N-1$)의 뒷부분에 해당하는 아랫부분의 출력을 메모리에 저장한 후 두 번째의 B의 $x(n)$ ($n=0, 1, \dots, N-1$)의 앞부분에 해당하는 윗부분의 출력을 더하면, Overlap-Add 연산이 된다.

그림 5에서 보는 것과 같이 제안한 Recursive 구조의 Recursive 부분은 2개의 Multiplier, 6개의 Adder 그리고 4개의 Delay로 구성되어진다. 연산시간은 Chiang의 Regressive 구조나 Nikolajevic의 Recursive 구조의 1/4 시간이 필요하다. 똑같은 연산시간을 가지는 Modified Regressive 알고리즘과 비교하여 2개의 Multiplier, 4개의 Adder 그리고 4개의 Delay를 줄여 하드웨어의 복잡도를 감소되었다. 또한 병렬 연산 시 시스템의 입력이 동일하며 순환계수 $2\cos \theta_n$ 에 대한 동일성으로 인해 하드웨어 구현이 간단하다.

IV. 실험 및 분석

1. 시뮬레이션 1

Matlab을 사용하여 제안하는 Recursive 구조와 식(1)에 의한 합성필터를 각각 구현하였다. 그리고

입의 오디오데이터를 MDCT 변환한 계수로부터 식(1)에 의하여 계산되어진 결과와 제안하는 Recursive 구조에 의한 결과를 비교하였을 때, 그림 6과 같은 동일한 결과를 얻을 수 있었다.

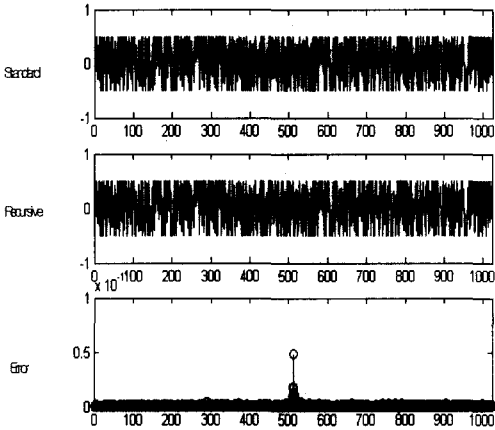


그림 6. Standard와 제안한 Recursive구조의 결과 비교

2. 시뮬레이션 2

VHDL을 사용하여 제안하는 Recursive 구조를 그림 7과 같이 구현하였다.

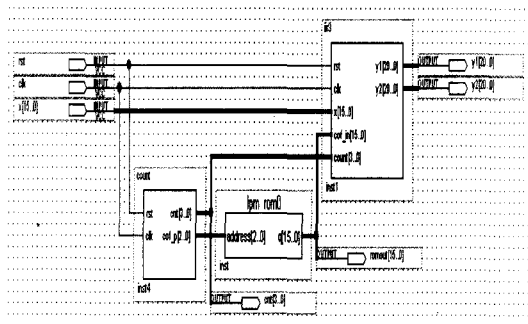


그림 7. 제안하는 Recursive 구조의 VHDL 구현

N=32일 때의 모의실험을 통해, matlab을 통해 계산되어진 결과와 그림 8에 의한 Time simulation을 통한 결과를 비교 및 검증하였다.

3. 결과 및 비교

제안하는 2단 IMDCT Recursive 구조와 참고문헌 [2][3][4]에 대하여 연산시간, 하드웨어의 복잡도,

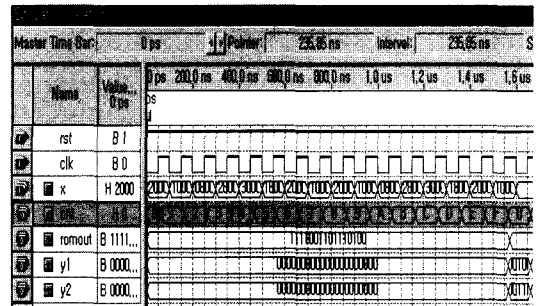


그림 8. Timing Simulation

필요한 메모리량을 비교하였다. 표 1에 2단 Recursive 구조에서 필요한 계수들의 메모리량을 정리하였다. 표 2에서 Modified Regressive 구조와 제안하는 2단 IMDCT의 Recursive 구조가 Chiang의 구조와 Nikolajevic의 구조보다 하드웨어의 복잡도는 증가하였지만, 연산시간을 비교했을 때 75%의 연산시간이 줄었음을 알 수 있다. 표 3의 필요 메모리의 비교에서도 제안하는 2단 IMDCT의 Recursive 구조가 가장 적다

따라서 연산시간과 하드웨어 복잡도, 메모리를 모두 고려하였을 경우, 제안하는 2단 IMDCT의 Recursive 구조가 가장 적합하다. 또한 Modified Regressive 구조와 입력 $X(k)$ 를 비교했을 때 Modified Regressive 구조가 4개의 입력인데 반해, 제안하는 알고리즘의 구조는 1개의 입력이라는 장점이 있다.

표 1. 2단 Recursive 구조의 계수 메모리량

계수	window size	Short_win (256)	Long_win (2048)	Total
$2 \cos \theta_n$		64	512	576
$\cos \frac{\theta}{2} W(n)$		128*2	1024*2	2304
$-\sin \frac{\theta}{2} W(n+N/2)$				
Total		320	2560	2880

표 2. 하드웨어의 복잡도 비교

구분	IMDCT			
	Mpy	Add	Z ⁻¹	clock cycles
Regressive Ref[2]	1	3	3	1024*2048
Recursive Ref[3]	1	3	2	1024*2048
Modified Ref[4]	4	10	8	512*1024
2단 Recursive	2	6	4	1024*512

표 3. 계수 메모리량 비교

구분	IMDCT		
	N=256	N=2048	Total
Standard	16384	1048576	1064960
Reference [2][3]	768	6720	7488
Modified Ref[4]	352	3392	3744
2단 Recursive	320	2560	2880

VI. 결 론

IMDCT의 Recursive 구조는 하드웨어로 구현 하면 단순하고, 병렬 VLSI 구조에 적합하다. 또한 이러한 구조는 MPEG Audio Standard의 다양한 Window Switching을 포함한 가변길이의 IMDCT에도 적용할 수 있다. 하지만 윈도우 길이 N이 커질 수록 연산시간이 많이 걸린다는 단점이 있다.

본 논문에서는 IMDCT의 성질을 이용한 2단 IMDCT의 Recursive 구조를 제안하여 기존의 구조보다 연산시간을 75% 단축시켰다. 그리고 연산시간과 하드웨어 복잡도, 필요한 메모리량을 기존에 제안된 알고리즘과 비교하였으며, 그 결과 제안하는 2단 IMDCT의 Recursive 구조가 AAC의 필터뱅크에 가장 적합하다는 것을 알 수 있었다. 제안된 알고리즘은 다양한 Window Switching을 포함한 가변길이의 IMDCT에도 적용 가능하므로 2가지 이상의 알고리즘을 복호화 할 수 있는 공용복호화기의 공용 합성필터에 적용가능하다.

앞으로의 연구과제는 고정소수점에서의 유효비트에서의 오차에 대한 검증을 통해, 본 논문에서 제안

된 알고리즘을 적용한 MPEG-2 AAC 복호화기를 구현하는 것이다.

참 고 문 헌

- [1] ISO/IEC JTC1/SC29/WG11 No. 1650 "IS 138-7(MPEG-2 Advanced Audio Coding, AAC)", April 1997.
- [2] Hwang-Cheng Chiang and Jie-Cherng Liu, "Regressive implementations for the forward and inverse MDCT in MPEG audio coding," Signal Processing Letters, IEEE, Volume 3 Issue 4, April 1996, pp.116 -118.
- [3] Nikolajevic, V. and Fettweis, G., "New recursive algorithms for the forward and inverse MDCT", Signal Processing Systems, 2001 IEEE Workshop on, Sept. 2001, pp. 51 -57.
- [4] Tsung-Han Tsai and Jiun-Nan Liu, "Architecture design for MPEG-2 AAC filterbank decoder using modified regressive method", IEEE International Conference, Volume 3, May 2002, pp. III-3216-3219, vol.3.
- [5] Sakamoto, T.; Taruki, M.; Hase, T., "A fast MPEG-audio layer III algorithm for a 32-bit MCU" Consumer Electronics, IEEE Transactions, Volume 45, Issue 3, Aug 1999, pp. 986 -993.
- [6] Mu-Huo Cheng; Yu-Hsin Hsu, "Fast IMDCT and MDCT algorithms - a matrix approach" Signal Processing, IEEE Transactions [see also Acoustics, Speech and Signal Processing, IEEE Transactions, Volume 51 Issue 1, Jan 2003, pp. 22.

박 세 기(Se-Ki Park)

정회원

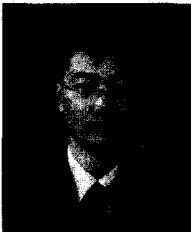


2002년 2월 : 대구대학교 정보
통신공학과 졸업
2004년 2월 : 대구대학교 정보
통신공학과 석사

<관심분야> 디지털신호처리, 적응신호처리

강 명 수(Myung-Soo Kang)

정회원

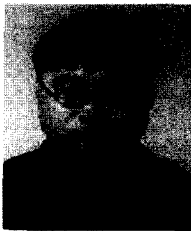


2002년 2월 : 대구대학교 정보
통신공학과 졸업
2004년 2월 : 대구대학교 정보
통신공학과 석사

<관심분야> 디지털신호처리, 적응신호처리

오 신 범(Shin-bum Oh)

정회원



1997년 2월 : 대구대학교 정보
통신공학과 졸업
1999년 8월 : 대구대학교 정보
통신공학과 석사
2004년 2월 : 대구대학교
정보통신공학과 박사

<관심분야> 디지털신호처리, 적응신호처리

이 채 욱(Chae-wook Lee)

정회원



1980년 2월 : 한국항공대학교
통신공학과 졸업
1987년 3월 : 동경공업대학교
전기전자공학과 석사
1990년 3월 : 동경공업대학교
전기전자공학과 박사
1990년 3월~현재 : 대구대학교
정보통신공학과 교수

<관심분야> 신호처리, 광통신시스템