

論文2003-40SP-6-12

MDCT/IMDCT의 계산 복잡도를 개선하기 위한 효율적인 알고리즘

(An Efficient Algorithm for Improving Calculation Complexity of the MDCT/IMDCT)

趙良基*, 李元彪*, 金熙碩**

(Yang-Ki Cho, Won-Pyo Lee, and Hi-Seok Kim)

요약

Modified Discrete Cosine Transform (MDCT)와 역변환인 IMDCT는 서브밴드 및 변환 코딩 기법에서 시간 영역 에일리어징 제거(Time Domain Aliasing Cancellation)를 기반으로 하는 분석/합성 필터뱅크로서 채택되고 있으며, MPEG 오디오 표준의 레이어-III에서 가장 많은 연산량을 필요로 한다. 본 논문에서는 MDCT/IMDCT를 효율적으로 계산할 수 있는 새로운 알고리즘을 제안하였다. 제안한 알고리즘은 DCT를 이용하여 MDCT/IMDCT를 계산하는 알고리즘에 기반을 두고 있기 때문에, MDCT/IMDCT 계산을 위해 두 개의 DCT-II를 이용한다. 더불어, 제안한 알고리즘은 4로 나누어지는 길이의 입력을 갖는 MDCT/IMDCT의 계산에도 적용할 수 있다. 제안한 알고리즘은 계산 복잡도 면에서 기존의 알고리즘들과 비교하여 적은 계산량을 필요로 하며, 구조적인 면에서 병렬적인 구조로 나타낼 수 있기 때문에, VLSI 구현에 매우 적합하다.

Abstract

The modified discrete cosine transform (MDCT) and inverse MDCT (IMDCT) are employed in subband/transform coding schemes as the analysis/synthesis filter bank based on time domain aliasing cancellation (TDAC). And the MDCT and IMDCT are the most computational intensive operations in layer III of the MPEG audio coding standard. In this paper, we propose a new efficient algorithm for the MDCT/IMDCT computation in various audio coding systems. It is based on the MDCT/IMDCT computation algorithm using the discrete cosine transforms (DCTs), and it employs two discrete cosine transform of type II (DCT-II) to compute the MDCT/IMDCT. In addition, it takes advantage of ability in calculating the MDCT/IMDCT computation, where the length of a data block is divisible by 4. The proposed algorithm in this paper requires less calculation complexity than the existing method does. Also, it can be implemented by the parallel structure, therefore its structure is particularly suitable for VLSI realization.

Keyword : MDCT, TDAC, MPEG Audio, Lifting, Rotation

* 學生會員, 淸州大學校 電子工學科
(Dept. of Electronics Engineering, Chongju University)

** 正會員, 淸州大學校 情報通信工學部
(School of Computer & Communication Engineering,

Chongju University)

※ 본 연구는 과학기술부·한국과학재단 지정 청주대학교 정보통신연구센터 지원에 의한 것입니다.

接受日字:2003年7月4日, 수정완료일:2003年11月22日

I. 서론

MDCT/IMDCT는 서브밴드 및 변환 코딩 (subband/transform coding) 기반의 오디오 (audio) 코딩에 관련한 많은 표준 및 제품에서 핵심적인 요소로서 채택되고 있으며, 시간 영역에서의 에일리어징 제거 기법 (Time Domain Aliasing Cancellation, TDAC)^[1]을 기반으로 하는 분석 및 합성 필터뱅크 (analysis/synthesis filter bank)로서의 역할을 하고 있다^[2-6]. MPEG 오디오-III (layer-III)와 같은 오디오 코딩 시스템에서 MDCT/IMDCT 블록은 계산 복잡도 (calculation complexity)가 가장 높은 요소이기 때문에, 하드웨어로 구현하기 위해서는 계산 복잡도를 감소시키기 위한 효율적인 계산 알고리즘이 필요하다. MDCT/IMDCT의 효율적인 구현에 대한 기존의 연구 결과들은 재환형 (recursive algorithm), Fast Fourier Transform (FFT)를 이용한 방식, 그리고 Discrete Cosine Transform (DCT)를 이용한 방식으로 분류할 수가 있다^[7]. 재환형 알고리즘은 MDCT 수식을 IIR (Infinite Impulse Response) 필터와 같이 재환을 갖는 수식으로 변형하는 방식으로서^[8-11], 하드웨어 복잡도는 작으나, 계산에 필요한 사이클 (cycle)의 수가 크고, 고정 소수점 (fixed-point) 계산을 이용하여 구현할 경우, 반복적 계산을 위해 라운드-오프 에러 (round-off error)가 비교적 크다^[7]. 따라서 에러를 줄이기 위해서는 내부 연산을 위한 비트 할당 (bit assignment)을 더 크게 해야 하므로 하드웨어 비용이 증가 한다^[7].

FFT 기반 알고리즘은 MDCT/IMDCT 계산을 복소수 또는 실수형 FFT 계산을 이용하여 처리하는 방식^[12, 13]이다. 이 방법은 MDCT/IMDCT 수식으로부터 FFT를 이용한 수식을 유도하고, FFT의 길이를 줄이기 위해서 데이터 시퀀스 (data sequence)를 치환 (permutation)하거나 버터플라이 (butterfly) 및 회전 (rotation)시킨다. FFT 기반 MDCT/IMDCT 알고리즘은 많은 오디오 코딩 시스템에 적용되고 있지만, 데이터 시퀀스의 길이가 $N = 2^n$ 일 때만 적용할 수 있다는 단점이 있다^[14, 16]. 따라서 MPEG-1이나 MPEG-2 오디오의 레이어-III에서 사용되는 길이 $N = 4 \cdot 3^m$ 인 MDCT/IMDCT의 경우, FFT 기반의 알고리즘을 적용할 수 없다.

최근 홀수 스택형 시스템에서 $N = 4 \cdot 3^m$ 인 길이를 가진 MDCT를 계산하기 위해 DCT를 이용하는 연구들

이 진행 되었는데^[14, 18], 이 알고리즘들은 MDCT 수식의 짝수 반대칭 (even anti-symmetry) 특성을 이용하며, MDCT 수식을 DCT와 Discrete Sine Transform (DST)식으로 변환하여 처리한다. DCT를 이용하는 MDCT 역시 데이터 시퀀스를 치환하거나 버터플라이 및 회전시킴으로써, DCT/DST의 길이를 감소시킨다. 뿐만 아니라, DST와 DCT는 서로 변환이 가능하기 때문에^[19], DCT 또는 DST만으로도 구현할 수 있다.

FFT 기반 알고리즘이나, DCT 기반 알고리즘을 이용할 경우, 재환 방식을 이용한 결과보다 라운드-오프 에러가 적고, 계산에 필요한 사이클의 수를 줄일 수 있다^[7].

본 논문에서는 DCT를 이용하여 MDCT를 계산하는 방식에 기반을 둔 새로운 효율적인 알고리즘을 제안하였다.

II. MDCT/IMDCT 계산 복잡도의 개선 방법

입력 시퀀스에 대한 MDCT에 대한 수식을 수식 (1)에 나타내었고, IMDCT에 대한 수식은(2)에 나타내었다^[1].

$$X_k = \sum_{m=0}^{N-1} x_m \cos \frac{\pi(2k+1)}{N} \left(m + \frac{1}{2} + \frac{N}{4} \right),$$

$$k = 0, 1, \dots, \frac{N}{2} - 1. \quad (1)$$

$$\hat{x}_m = \frac{2}{N} \sum_{k=0}^{N/2-1} X_k \cos \frac{\pi(2k+1)}{N} \left(m + \frac{1}{2} + \frac{N}{4} \right),$$

$$m = 0, 1, \dots, N-1. \quad (2)$$

MDCT의 경우, 입력 시퀀스 x_m 은 이미 윈도우 (window) 처리된 상태라고 가정한다. 수식 (1)과 식 (2)에서 m 은 시간축 인덱스 (index)를 나타내고, k 는 주파수 인덱스를, N 은 데이터 블록의 길이를 의미한다. 또한 변환 커널 (transform kernel)인 코사인 항 $1/2 + N/4$ 는 페이저 성분 (the phase factor)이다^[14]. 식 (1)과 식 (2)에서 알 수 있듯이, MDCT와 IMDCT의 코사인 계수 (coefficient)의 수가 각각 $N/2 \times N$ 과 $N \times N/2$ 이기 때문에, 수식의 변형 없이 그대로 구현할 경우, 곱셈과 덧셈의 수는 각각 $N^2/2$ 과 $N(N-1)/2$ 이기 때문에, 계산 복잡도가 매우 크고, 계산에 필요한 사이클의 수가 크다.

따라서, 본 논문에서는 계산 복잡도를 효율적으로 줄일 수 있는 새로운 계산 방식을 제안하였다.

1. MDCT의 유도

식 (1)에 나타난 MDCT의 계수 X_k 는 식 (3)과 같이 짝수 반대칭 특성을 갖는다.

$$X_{N-k-1} = -X_k, \quad k = 0, 1, \dots, N/2 - 1. \quad (3)$$

입력 데이터 시퀀스 x_m 을 식 (4)와 같이 치환하면, 식 (1)을 식 (5)와 같이 나타낼 수 있다.

$$y_m = \begin{cases} -x_{m-(N/4)}, & m = 0, 1, \dots, (N/4) - 1 \\ x_{m-(N/4)}, & m = (N/4), (N/4) + 1, \dots, N - 1 \end{cases} \quad (4)$$

$$X_k = \sum_{m=0}^{N-1} y_m \cos \frac{\pi}{2N} (2k+1)(2m+1) \quad (5)$$

$$= \sum_{m=0}^{N/2-1} (y_m - y_{N-1-m}) \cos \frac{\pi}{2N} (2k+1)(2m+1) \\ = \sum_{m=0}^{N/2-1} w_m \cos \frac{\pi}{2N} (2k+1)(2m+1) \quad (6)$$

식 (5)의 코사인 항은 반대칭을 갖고 있기 때문에, 식 (6)과 같이 계수를 반으로 줄일 수 있다. 식 (6)에서의 변환 커널은 w_m 의 $N/2$ -포인트 DCT-IV 식으로 볼 수 있고, 논문 [16]에서의 MDCT 전개 방식 중 DCT-IV 형태의 중간식과 같다.

MDCT 계수 X_k 를 짝수부와 홀수부로 나누면, 식 (6)은 다음과 같이 나타낼 수가 있다.

$$X_{2k} = \sum_{m=0}^{N/2-1} w_m \cos \frac{\pi}{2N} (4k+1)(2m+1), \quad (7)$$

$$X_{2k+1} = \sum_{m=0}^{N/2-1} w_m \cos \frac{\pi}{2N} (4k+3)(2m+1), \quad (8)$$

$$k = 0, 1, \dots, N/4 - 1.$$

1) 짝수부의 유도

삼각 공식을 이용하여 식 (7)을 확장하면, 식 (9)와 같이 나타낼 수 있다.

$$X_{2k} = \sum_{m=0}^{N/2-1} w_m \cos \frac{\pi}{2N} (2m+1) \cos \frac{k\pi}{N/2} (2m+1) \\ - \sum_{m=0}^{N/2-1} w_m \sin \frac{\pi}{2N} (2m+1) \sin \frac{k\pi}{N/2} (2m+1). \quad (9)$$

식 (9)의 두 항은 모두 대칭 특성을 갖고 있기 때문에, 식 (10)과 같이 유도할 수 있다.

$$X_{2k} = \sum_{m=0}^{N/4-1} u_m \cos \frac{k\pi}{N/2} (2m+1) \\ + \sum_{m=0}^{N/2-1} v_m \sin \frac{k\pi}{N/2} (2m+1), \quad (10) \\ = A_k + B_k$$

$$u_m = w_m \cos \frac{\pi}{2N} (2m+1) \\ + w_{N/2-1-m} \sin \frac{\pi}{2N} (2m+1), \quad (11) \\ v_m = -w_m \sin \frac{\pi}{2N} (2m+1) \\ + w_{N/2-1-m} \cos \frac{\pi}{2N} (2m+1)$$

$$A_k = \sum_{m=0}^{N/4-1} u_m \cos \frac{k\pi}{N/2} (2m+1) \quad (12)$$

$$B_k = \sum_{m=0}^{N/2-1} v_m \sin \frac{k\pi}{N/2} (2m+1).$$

식 (10)에서 u_m 과 v_m 및 A_k 와 B_k 를 각각 식 (11)과 식 (12)에 나타내었다. 식 (10)에서 A_k 와 B_k 의 식은 각각 $N/4$ -포인트 DCT-II와 DST-II로 볼 수 있다. DST-II를 DCT-II로 변환할 수 있기 때문에^[19], 식 (12)에서 B_k 의 DST-II 식을 식 (13)과 같이 DCT-II의 식 B'_k 로 표현할 수가 있다. 식 (13)을 식 (10)에 적용하면 식 (14)와 같이 다시 나타낼 수 있다.

$$B'_k = B_{N/4-k} \\ = \sum_{m=0}^{N/4-1} (-1)^m v_m \cos \frac{k\pi}{N/2} (2m+1), \quad (13)$$

$$k = 1, 2, \dots, N/4.$$

$$\therefore X_{2k} = A_k + B'_{N/4-k}, \quad (14)$$

$$k = 0, 1, \dots, N/4 - 1.$$

식 (13)에서 알 수 있듯이, DST의 입력 데이터 중 홀수 번째 인덱스를 갖는 데이터에 '-1'을 곱하고, 결과 계수의 순서를 반전시키면, DCT의 결과를 얻을 수 있다.

2) 홀수부의 유도

MDCT 계수의 홀수부를 나타낸 식 (8)을 MDCT의 짝수부 계수에 대한 유도식과 마찬가지로 두 개의 $N/4$ 포인트 DCT-II의 식으로 변환하기 위해서, 식 (15)와 같이 다시 표현한다.

$$X_{2k+1} = \sum_{m=0}^{N/2-1} w_m \cos \frac{\pi}{2N} (4k-1)(2m+1), \quad (15)$$

$$k = 1, 2, \dots, N/4.$$

식 (15)에서 인덱스 k 는 0이 아닌 1부터 시작한다. 식 (15)를 짝수부의 계수와 유사한 방법으로 유도하면 식 (16)을 얻을 수 있다.

$$X_{2k-1} = A_k - B'_{N/4-k}, \quad (16)$$

$$k = 1, 2, \dots, N/4.$$

식 (16)에서 인덱스 k 를 $k+1$ 로 치환하면, 식 (17)과 같이 완전한 MDCT의 홀수부 계수에 대한 식으로 표현할 수 있다.

$$\therefore X_{2k+1} = A_{k+1} - B'_{N/4-k-1}, \quad (17)$$

$$k = 0, 1, \dots, N/4 - 1.$$

MDCT의 결과인 식 (14)와 식 (17)은 w_m 의 $N/4$ -포인트 DCT-II 두 개에 의한 MDCT 식을 표현한 것으로서, <그림 1>에 식 (14)와 식 (17)의 흐름도(flowgraph)를 표현하였다.

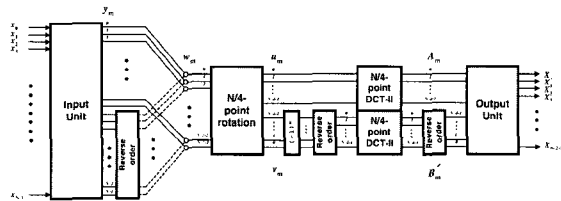


그림 1. 제안한 MDCT 계산 알고리즘의 흐름도
Fig. 1. Flowgraph of proposed algorithm for MDCT calculation.

<그림 1>에서 원(O) 기호는 덧셈을 의미하고, 점선은 부호 반전을 의미한다. 역순 (Reverse order)블록은 입력 데이터 시퀀스의 순서를 반대로 바꾸는 과정이며, 반전 $(-1)^m$ 블록은 홀수 번째 데이터의 부호를 바꾸는 과정이다. 식 (4)에 해당하는 입력부(input unit)는 입력 데이터 시퀀스를 치환하는 과정으로서, <그림 2(a)>에 내부 흐름도를 나타내었다. 출력부(output unit)는 두 개의 $N/4$ -포인트 DCT-II의 결과를 조합하여 MDCT 계수를 출력하는 과정으로서, 흐름도를 <그림 2(b)>에 나타내었다. <그림 2(b)>에 표현된 출력부는 $N/2 - 2$ 개의 덧셈이 필요하다.

$N/4$ -포인트 회전 (rotation) 블록은 식 (11)에 해당하는 부분으로서, N 개의 곱셈과 N 개의 덧셈이 필요하

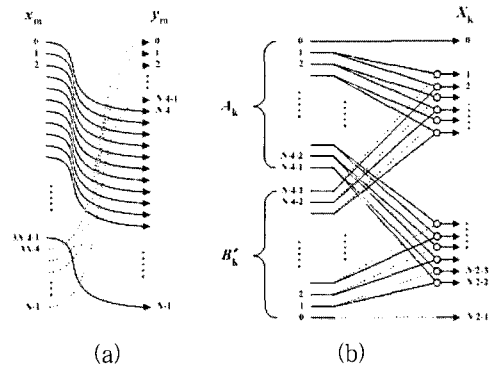


그림 2. 입력부와 출력부의 흐름도 : (a) 입력부, (b) 출력부

Fig. 2. Flowgraph of the input unit and the output unit : (a) input unit, (b) output unit.

기 때문에, 제안한 구조에서 가장 복잡도가 높은 부분이다. 본 논문에서는 $N/4$ -포인트 회전 블록의 곱셈 복잡도를 줄이기 위해서, 리프팅 (lifting) 방법을 사용하였다^[16]. 식 (11)을 행렬 곱 형태로 나타내고, 리프팅 방법을 이용하여 전개하면 식 (18)과 같이 나타낼 수 있다^[16].

$$\begin{aligned} v_m &= 1 \quad \frac{c-1}{s} \quad 1 \quad 0 \quad 1 \quad \frac{c-1}{s} \quad w_m \\ u_m &= 0 \quad 1 \quad s \quad 1 \quad 0 \quad 1 \quad w_{N/2-1-m} \end{aligned} \quad (18)$$

$$= \begin{aligned} &1 \quad a_m \quad 1 \quad 0 \quad 1 \quad a_m \quad w_m \\ &0 \quad 1 \quad b_m \quad 1 \quad 0 \quad 1 \quad w_{N/2-1-m} \end{aligned}$$

$$a_m = \frac{\cos \frac{\pi}{2N} (2m+1) - 1}{\sin \frac{\pi}{2N} (2m+1)} \quad (19)$$

$$b_m = \sin \frac{\pi}{2N} (2m+1).$$

식 (18)에서 a_m 과 b_m 은 리프팅 계수^[16]이며, 식 (19)와 같다. 식 (18)에서 알 수 있듯이, 리프팅 방법을 이용하면, $N/4$ -포인트 회전 블록을 $3N/4$ 개의 곱셈과 $3N/4$ 개의 덧셈으로 구현할 수 있다.

2. IMDCT의 유도

본 논문에서 제안한 MDCT의 구조를 IMDCT의 계산에 적용하기 위해서, 식 (20)의 치환을 식 (2)에 적용해서 정리하면 식 (21)과 같이 나타낼 수 있다^[16].

$$\hat{x}_m = \begin{cases} y_{m+(N/4)}, & m = 0, 1, \dots, (3N/4) - 1 \\ -y_{m-(3N/4)}, & m = (3N/4), (3N/4) + 1, \dots, N - 1 \end{cases} \quad (20)$$

$$y_m = \sum_{k=0}^{(N/2)-1} X_k \cos \frac{\pi}{2N} (2k+1)(2m+1), \quad (21)$$

$$m = 0, 1, \dots, (N/2) - 1$$

식 (21)에 나타낸 y_m 은 짝수 반대칭 특성을 갖고 있기 때문에, 식 (22)가 성립된다.

$$y_m = -y_{N-1-m}, \quad m = 0, 1, \dots, (N/2) - 1 \quad (22)$$

식 (22)를 고려하면, 식 (21)은 식 (6)과 완전히 동일한 식으로 볼 수 있기 때문에, 제안한 MDCT 계산 알고리즘에 의한 구조를 이용하여 추가적인 연산 없이 IMDCT의 계산을 직접 처리할 수 있다.

III. 알고리즘 비교

본 논문에서는 제안한 알고리즘의 성능을 평가하기 위하여 기존의 DCT 기반 MDCT 계산 알고리즘들과 비교하였다. 각 알고리즘을 이용한 MDCT의 계산에서 DCT와 DST의 계산은 동일한 알고리즘을 이용한다고 가정한다. 즉, 각 알고리즘에서 사용하는 DCT와 DST

는 계산 복잡도 및 특성은 모두 같다.

우선 MDCT의 길이 N 이 2^n , 12, 36라고 가정할 때, 각 알고리즘의 계산 복잡도 비교를 하였고, <표 1>과 <표 2>에 각각 곱셈과 덧셈의 복잡도를 비교한 결과를 나타내었다. 결과에서 알 수 있듯이, 곱셈의 복잡도는 적용한 모든 알고리즘에 대해서 같고, 덧셈의 복잡도의 경우, 제안한 알고리즘의 복잡도가 Lee의 알고리즘과 같으며, 나머지의 알고리즘들보다 우수함을 알 수 있다. 두 번째 비교 평가는 각 알고리즘에 대한 대기 사이클 (또는 지연 사이클, latency)과 데이터 처리율(throughput)이다. 비교 평가를 위해, 모든 알고리즘을 최소 레벨의 파이프라인으로 구현하였다고 가정한다. 즉, 하나의 병렬 연산을 하나의 파이프라인 레벨로 처리한다.

제안한 알고리즘을 파이프라인화된 하드웨어로 구현할 때의 흐름도를 <그림 3>에 나타내었다. <그림 3>에서 $N/4$ -포인트 DCT-II의 파이프라인 레벨이 n_c 이고 데이터 처리율이 1 사이클이라고 가정할 때, 최종 전체의 파이프라인 레벨은 $8 + n_c$ 이다. 따라서 $N/4$ -포인트 DCT-II의 대기 사이클은 n_c 이고, 전체의 대기

표 1. 기존의 알고리즘과의 곱셈 복잡도 비교
Table 1. Complexity comparison between the known algorithm and the proposed algorithm(Multiplication).

N=2^n일 때	
Algorithm	Num. of mul. (MDCT : IMDCT)
Britanak's[14,15]	3N/4 + n _m : 3N/4 + n _m
Lee's[16]	3N/4 + n _m : 3N/4 + n _m
Krishnan's[18]	3N/4 + n _m : 3N/4 + n _m
proposed	3N/4 + n _m : 3N/4 + n _m
N=12일 때	
Algorithm	Num. of mul. (MDCT : IMDCT)
Britanak's[14,15]	9 + n _m : 9 + n _m
Lee's[16]	9 + n _m : 9 + n _m
Krishnan's[18]	9 + n _m : 9 + n _m
proposed	9 + n _m : 9 + n _m
N=36일 때	
Algorithm	Num. of mul. (MDCT : IMDCT)
Britanak's[14,15]	27 + : 27 + n _m
Lee's[16]	27 + : 27 + n _m
Krishnan's[18]	27 + : 27 + n _m
proposed	27 + : 27 + n _m

n_m : the number of multiplications for the DCT and DST

표 2. 기존의 알고리즘과의 덧셈 복잡도 비교
Table 2. Complexity comparison between the known algorithm and the proposed algorithm(Addition).

N=2^n일 때	
Algorithm	Num. of add. (MDCT : IMDCT)
Britanak's[14,15]	(11N/4) - 2 + n _a : (9N/4) - 2 + n _a
Lee's[16]	(7N/4) - 2 + n _a : (5N/4) - 2 + n _a
Krishnan's[18]	(11N/4) - 2 + n _a : (9N/4) - 2 + n _a
proposed	(7N/4) - 2 + n _a : (5N/4) - 2 + n _a
N=12일 때	
Algorithm	Num. of mul. (MDCT : IMDCT)
Britanak's[14,15]	31 + n _a : 25 + n _a
Lee's[16]	19 + n _a : 13 + n _a
Krishnan's[18]	31 + n _a : 25 + n _a
proposed	19 + n _a : 13 + n _a
N=36일 때	
Algorithm	Num. of mul. (MDCT : IMDCT)
Britanak's[14,15]	97 + n _a : 81 + n _a
Lee's[16]	61 + n _a : 43 + n _a
Krishnan's[18]	97 + n _a : 81 + n _a
proposed	61 + n _a : 43 + n _a

n_a : the number of additions for the DCT and DST

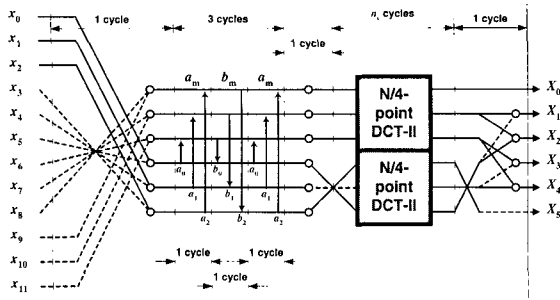


그림 3. 제안한 알고리즘에 대한 파이프라인화된 구조
Fig. 3. Pipelined structure by the proposed algorithm.

표 3. 기존의 알고리즘과의 대기 사이클 수 비교

Table 3. Latency comparison between the known algorithms and the proposed algorithm.

Algorithm	Latency(cycles)
Britanak's[14,15]	$7 + n_c$
Lee's[16]	$4 + N/2 + n_c$
Krishnan's[18]	$7 + n_c$
proposed	$6 + n_c$

n_c : the latency of the DCT and DST

사이클은 $8 + n_c$ 이며, 데이터 처리율은 1 사이클이다. 각 알고리즘을 구현한 회로에 대해 대기 사이클과 처리율의 비교 결과를 각각 <표 3>과 <표 4>에 나타내었다.

<표 3>에서, Lee의 알고리즘에 의한 구조는 캐환 구조를 갖고 있기 때문에, 대기 사이클이 비교적 큰 반면, 제안한 알고리즘에 의한 구조가 가장 작다. Britanak과 Krishnan 알고리즘에 의한 구조들은 입력 버퍼플라이가 두 개이므로, 제안한 구조보다 대기 사이클이 1만큼 크다.

표 4. 기존의 알고리즘과의 데이터 처리율 비교

Table 4. Throughput comparison between the known algorithms and the proposed algorithm.

Algorithm	Throughput(cycles)
Britanak's[14,15]	1
Lee's[16]	$N/4$
Krishnan's[18]	1
proposed	1

<표 4>에 나타난 데이터 처리율의 비교에서는, 캐환 구조를 갖는 Lee의 구조가 가장 데이터 처리 속도가 늦다는 것을 알 수 있다.

V. 결론

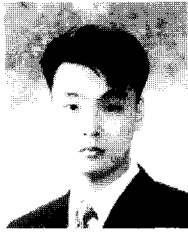
본 논문에서는 DCT를 기반으로 하여 MDCT와 IMDCT를 효율적으로 계산하는 새로운 알고리즘을 제안하였다. 기존의 알고리즘들과 비교하여, 제안한 알고리즘은 적은 계산 복잡도를 갖고 있으며, 제안한 알고리즘에 의한 구조는 적은 대기 사이클과 적은 데이터 처리율을 갖는다. 또한 제안한 알고리즘은 길이가 2^n 인 MDCT/IMDCT 뿐만 아니라, 길이가 $N = 4 \cdot 3^m$ 인 MDCT/IMDCT에도 적용할 수 있으며, 구조적인 측면에서 병렬적 하드웨어 구현에 매우 적합함을 알 수 있다.

참고 문헌

- [1] J. P. Princen, A. W. Johnson, and A. B. Bradley, "Subband/Transform Coding Using Filter Banks Designs Based on Time Domain Aliasing Cancellation," in Proc. IEEE Int. Conf. Acoustic, Speech and Signal Processing, Dallas, TX, Apr. 1987, pp. 2161-2164.
- [2] ISO/IEC JTC1/SC29/WG11 MPEG, IS 11172-3, Information Technology - Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mb/s - Part 3 : Audio, MPEG-1, 1992.
- [3] T. H. Tsai, "A System Level Integration Design for MPEG Layer III Audio Decoder," in Proc. Int. Conf. Signal Processing (WCCC-ICSP), Beijing, China., Aug. 2000, pp. 1333-1336.
- [4] M. S. Vinton and L. E. Atlas, "A Scalable and Progressive Audio Codec," in Proc. IEEE Int. Conf. Acoustic, Speech, and Signal Processing, Salt Lake City, UT, May. 2001, pp. 3277-3280.

- [5] S. Hashimoto, A. Niwa, H. Okuhata, and I. Shirakawa, "VLSI Implementation of Portable MPEG-4 Audio Decoder," in Proc. 13th Annual IEEE Int. ASIC/SOC Conf., Arlington, VA, Sep. 2000, pp. 80-84.
- [6] M. A. Watson and P. Buettner, "Design and Implementation of AAC Decoders," IEEE Trans. Consumer Electronics, Vol. 46, Iss. 3, pp. 819-824, Aug. 2000.
- [7] X. Yang, S. Shi, and A. K. Wong, "Tradeoffs in Modified Discrete Cosine Transform Implementations," in Proc. Int. Conf. ASIC, Shanghai, China, Oct. 2001, pp. 370-373.
- [8] V. Nikolajevic and G. Fettweis, "New Recursive Algorithm for the Forward and Inverse MDCT," IEEE Workshop Signal Processing Systems, Sep. 2001, Antwerp, Belgium, pp. 51-57.
- [9] H. C. Chiang and J. C. Liu, "Regressive Implementations for the Forward and Inverse MDCT in MPEG Audio Coding," IEEE Signal Processing Letters, Vol. 3, Iss. 4, pp. 116-118, Apr. 1996.
- [10] C. H. Chen, C. B. Wu, B. D. Liu, and J. F. Yang, "Recursive Architectures for the Forward and Inverse Modified Discrete Cosine Transforms," IEEE Workshop Signal Processing Systems, Lafayette, LA., Oct. 2000, pp. 50-59.
- [11] D. Y. Chan, J. F. Yang, and S. Y. Chen, "Regular Implementation Algorithm of Time Domain Aliasing Cancellation," IEEE Proc. Vision, Image, and Signal Processing, Vol. 143, Iss. 6, pp. 387-392, Dec. 1996.
- [12] M. Iwadare, A. Sugiyama, F. Hazu, A. Hirano, and T. Nishitani, "A 128 kb/s Hi-Fi Audio CODEC Based on Adaptive Transform Coding with Adaptive Block Size MDCT," IEEE Journal Selected Areas in Communication, Vol. 10, Iss. 1, pp. 138-144, Jan. 1992.
- [13] P. Duhamel, Y. Mahieux, and J. P. Petit, "A Fast Algorithm for the Implementation of Filter Banks Based on Time Domain Aliasing Cancellation," Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP), Apr. 1991, Toronto, Ont., Canada, pp. 2209-2212.
- [14] V. Britanak and K. R. Rao, "A New Fast Algorithm for the Unified Forward and Inverse MDCT/MDST Computation," Signal Processing, Vol. 82, Iss. 3, pp. 433-459, Mar. 2002.
- [15] V. Britanak and K. R. Rao, "An Efficient Implementation of the Forward and Inverse MDCT in MPEG Audio Coding," IEEE Signal Processing Letters, Vol. 8, Iss. 2, pp. 48-51, Feb. 2001.
- [16] S. W. Lee, "Improved Algorithm for Efficient Computation of the Forward and Backward MDCT in MPEG Audio Coder," IEEE Transaction Circuits and Systems II : Analog and Digital Signal Processing, Vol. 48, Iss. 10, pp. 990-994, Oct. 2001.
- [17] W. Jianxin and D. Zaiwang, "A Fast Algorithm for Modified Discrete Cosine Transform," Int. Conf. Communication Technology Proceedings (ICCT), Beijing, China, May. 1996, pp. 445-448.
- [18] T. Krishnan and S. Orantara, "Fast and Lossless Implementation of the Forward and Inverse MDCT Computation in MPEG Audio Coding," Int. Sym. Circuit and Systems (ISCAS), Arlington, Texas, 2002, pp. 181-184.
- [19] P. Lee and F. Y. Huang, "Restructured Recursive DCT and DST Algorithm," IEEE Trans. Signal Processing, Vol. 42, Iss. 7, pp. 1600-1609, Jul. 1994.

저 자 소 개



趙 良 基(學生會員)

1997년 : 청주대학교 전자공학과 학사. 1999년 : 청주대학교 전자공학과 석사. 2002년 : 청주대학교 전자공학과 박사 수료. <주관심분야 : 디지털 신호 처리용 VLSI 설계, 마이크로프로세서 설계, 시스템온

칩.>



李 元 彪(學生會員)

1997년 : 청주대학교 전자공학과 학사. 1999년 : 청주대학교 전자공학과 석사. 2002년 : 청주대학교 전자공학과 박사 수료. <주관심분야 : 마이크로프로세서 설계, 영상처리, 시스템온칩>



金 熙 碩(正會員)

1977년 : 한양대학교 전자공학과 학사. 1980년 : 한양대학교 전자공학과 석사. 1985년 : 한양대학교 전자공학과 박사. 1988~1989년 : University of Colorado 교환교수.

1997~1998년 : UC Irvine 교환 교

수. 2001~2002년 : 대한전자공학회 CAD & VLSI 설계 연구회 전문위원장. 현 청주대학교 정보통신공학부 교

수. <주관심분야 : 시스템/상위/논리레벨 합성, 시스템온 칩, DSP 설계.>