

MPEG-2 AAC의 MDCT/IMDCT를 위한 최적 알고리즘 개발

°김병규, 이강현

조선대학교 전자정보통신공학부 멀티미디어 ASIC설계 실험실

Web : http://vlsi.chosun.ac.kr

E-mail : khrhee@vlsi.chosun.ac.kr

A Development on the Optimization Algorithm for MDCT/IMDCT of MPEG-2 AAC

°Byung Guy Kim, Kang Hyeon Rhee

School of Electronics, Information and Communications Eng.,

Multimedia ASIC Lab., Chosun University

Tel : (062) 230-7066 / Fax : (062) 233-1120

E-mail : khrhee@vlsi.chosun.ac.kr

Abstract

MPEG-2 AAC(Advanced Audio Coding) is the most advanced coding scheme available for high quality audio coding. This MPEG-2 AAC audio standard allows for ITU-R 'indistinguishable' quality according to at data rates of 320 kb/s for five full-bandwidth channel audio signals. The compression ratio is around a factor of 1.4 better compared to MPEG Layer 3, you get the same quality at 70% of the bitrate.

This paper suggest optimization method for MDCT/IMDCT (Modified Discrete Cosine Transform /Inverse Modified Discrete Cosine Transform) in Encoder and Decoder for AAC.

1. 서 론

AAC는 크게 huffman coding, quantization and scaling, backward adaptive prediction, MDCT, gain control and hybrid filter bank(polyphase quadrature filter (IPQF)+IMDCT) 이러한 tool들로 구성되었다.[1]

본 논문에서는 소프트웨어 및 하드웨어 적으로 AAC구현할 때 최적의 성능을 발휘하도록 MDCT/I-MDCT 연산에 필요한 cosine 계수를 약 0.5%로 줄이고, 연산 량을 0.09%로 최적 화한 후 이를 검증하고,

최적화 시 발생하는 오차를 검토하였다.

2. MDCT/IMDCT 계수의 최소화

MDCT는 Time Domain Aliasing Cancellation (TDAC)을 이용한 서브밴드 코딩으로 Aliasing을 제거한다.[2]

MDCT는 시간영역 데이터를 주파수영역으로 변환하며, Inverse Modified Discrete Cosine Transform (IMDCT)는 역으로 주파수영역 데이터를 시간영역 데이터로 환원한다. 디코딩과정에서 IMDCT는 전체 처리과정에서 CPU점유율 30%이상 많은 연산을 필요로 한다.[3]

AAC의 MDCT 및 IMDCT는 각각 식 1, 식 2와 같다.[4]

$$X(i,k) = 2 \cdot \sum_{n=0}^{N-1} x(i,n) \cos\left(\frac{2\pi}{N}(n+n_0)\left(k+\frac{1}{2}\right)\right)$$

for $0 \leq k < N/2$

$$\begin{cases} x_{in} = \text{원도된 입력} \\ n = \text{샘플 인덱스} \\ k = \text{cosine계수인덱스} \\ i = \text{블럭 인덱스} \\ N = \text{원도 길이} \\ n_0 = \left(\frac{N}{2}+1\right)\frac{1}{2} \end{cases} \quad (1)$$

$$X(i, n) = \frac{2}{N} \sum_{k=0}^{N/2-1} x(i, k) \cos\left(\frac{2\pi}{N}(n+n_0)\left(k+\frac{1}{2}\right)\right)$$

for $0 \leq n < N$

$$\begin{cases} x_{in} = \text{원도된 입력} \\ n = \text{샘플 인덱스} \\ i = \text{원도 인덱스} \\ k = \text{spectral계수인덱스} \\ N = \text{원도 길이} \\ n_0 = \left(\frac{N}{2} + 1\right)\frac{1}{2} \end{cases} \quad (2)$$

식 1과 식 2는 계수인덱스 길이와 누산 후 2 또는 2/N을 곱하는 것 말고는 차이가 없다. 따라서 식 1의 경우를 최적화 하여 식 2에도 적용가능 하다.

식 1에서 cosine 계수를 식 3과 같이 지수 함수로 나타내어 최적화를 수행한다.

$$\cos\left(\frac{2\pi}{N}(n+n_0)\left(k+\frac{1}{2}\right)\right) = e^{j\frac{2\pi}{N}(n+n_0)(k+\frac{1}{2})} = C_N^{(n+n_0)(k+\frac{1}{2})} \quad (3)$$

여기서 $e^{j\frac{2\pi}{N}} = C_N$

따라서 식 1은 다음과 같다.

$$\begin{aligned} X(i, k) &= 2 \cdot \sum_{n=0}^{N/2-1} x(i, n) C_N^{(n+n_0)(k+\frac{1}{2})} \\ &= C_N^{n_0(k+\frac{1}{2})} \sum_{n=0}^{N/2-1} x(i, n) C_N^{n(k+\frac{1}{2})} \end{aligned} \quad (4)$$

여기서 짝수 부와 홀수 부로 나누어 쓰면

$$\begin{aligned} X(i, k) &= C_N^{n_0(k+\frac{1}{2})} \left[\sum_{\text{even}} x(i, n) C_N^{n(k+\frac{1}{2})} + \sum_{\text{odd}} x(i, n) C_N^{n(k+\frac{1}{2})} \right] \\ &= C_N^{n_0(k+\frac{1}{2})} \left[\sum_{r=0}^{(N/2)-1} x(i, 2r) C_N^{2r(k+\frac{1}{2})} + \sum_{r=0}^{(N/2)-1} x(i, 2r+1) C_N^{(2r+1)(k+\frac{1}{2})} \right] \\ &= C_N^{n_0(k+\frac{1}{2})} \left[\sum_{r=0}^{(N/2)-1} x(i, 2r) C_N^{2r(k+\frac{1}{2})} + C_N^{(k+\frac{1}{2})} \sum_{r=0}^{(N/2)-1} x(i, 2r+1) C_N^{2r(k+\frac{1}{2})} \right] \end{aligned} \quad (5)$$

(단 r 은 정수 인덱스)

식 5에서 N 점 MDCT를 $N/2$ 점 MDCT로 구할 수 있다. 식 5를 다시 쓰면 식 6과 같다.

$$\begin{aligned} X(i, k) &= C_N^{n_0(k+\frac{1}{2})} \left[\sum_{r=0}^{(N/2)-1} x(i, 2r) C_{N/2}^{r(k+\frac{1}{2})} + C_N^{(k+\frac{1}{2})} \sum_{r=0}^{(N/2)-1} x(i, 2r+1) C_{N/2}^{r(k+\frac{1}{2})} \right] \end{aligned} \quad (6)$$

다시 홀수 부와 짝수 부로 따로 식 7과 같이 나타내면

$$\begin{aligned} E(i, k) &= \sum_{r=0}^{(N/2)-1} x(i, 2r) C_{N/2}^{r(k+\frac{1}{2})} \\ O(i, k) &= \sum_{r=0}^{(N/2)-1} x(i, 2r+1) C_{N/2}^{r(k+\frac{1}{2})} \end{aligned} \quad (7)$$

즉

$$X(i, k) = C_N^{n_0(k+\frac{1}{2})} \left[E(i, k) + C_N^{(k+\frac{1}{2})} O(i, k) \right] \quad (8)$$

식 8을 이용하여 흐름선도를 그리면 그림 1과 같다.

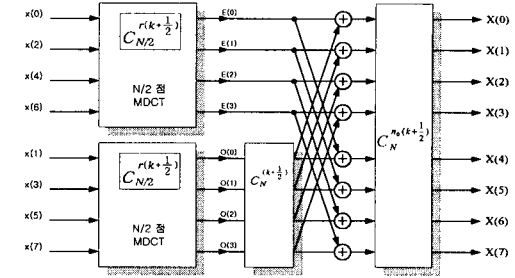


그림 1. N/2 점 MDCT의 흐름선도

이와 같은 방법을 다시 적용하여 $E(i, k)$ 와 $O(i, k)$ 을 각각 짝수 부와 홀수 부로 나누어 식 9와 같이 나타낼 수 있다.

$$\begin{aligned} E(i, r) &= \sum_{m=0}^{(N/2)-1} e(i, m) C_{N/2}^{r(k+\frac{1}{2})} \\ &= \sum_{m=0}^{(N/4)-1} e(i, 2m) C_{N/2}^{2m(k+\frac{1}{2})} + \sum_{m=0}^{(N/4)-1} e(i, 2m+1) C_{N/2}^{(2m+1)(k+\frac{1}{2})} \\ &= \sum_{m=0}^{(N/4)-1} e(i, 2m) C_{N/4}^{m(k+\frac{1}{2})} + C_{N/2}^{(k+\frac{1}{2})} \sum_{m=0}^{(N/4)-1} e(i, 2m+1) C_{N/4}^{m(k+\frac{1}{2})} \\ &= G(i, k) + C_{N/2}^{k+\frac{1}{2}} H(i, k) \end{aligned} \quad (9)$$

단 $G(i, k) = \sum_{m=0}^{(N/4)-1} e(i, 2m) C_{N/4}^{m(k+\frac{1}{2})}$

$H(i, k) = \sum_{m=0}^{(N/4)-1} e(i, 2m+1) C_{N/4}^{m(k+\frac{1}{2})}$

$e(r) = x(2r) \quad \therefore \quad e(2m) = x(4m)$
 $e(2m+1) = x(4m+2)$

또한 $N/2$ 가 홀수일 때 다음과 같이 $N/4$ 점 MDCT으로 나타낼 수 있다.

$$\begin{aligned} O(i, k) &= \sum_{r=0}^{(N/2)-1} x(i, 2r+1) C_{N/2}^{r(k+\frac{1}{2})} \\ &= \sum_{m=0}^{(N/4)-1} o(i, 2m) C_{N/2}^{2m(k+\frac{1}{2})} + \sum_{m=0}^{(N/4)-1} o(i, 2m+1) C_{N/2}^{(2m+1)(k+\frac{1}{2})} \\ &= \sum_{m=0}^{(N/4)-1} o(i, 2m) C_{N/4}^{m(k+\frac{1}{2})} + C_{N/2}^{k+\frac{1}{2}} \sum_{m=0}^{(N/4)-1} o(i, 2m+1) C_{N/4}^{m(k+\frac{1}{2})} \end{aligned} \quad (10)$$

식 10을 이용하여 흐름선도를 그리면 그림 2와 같다.

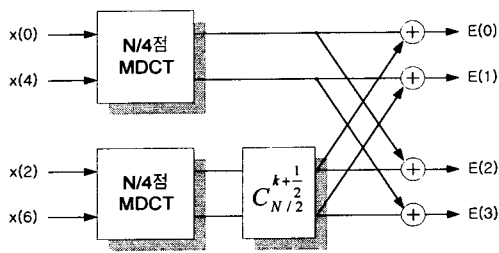


그림 2. N/4점 MDCT

따라서 N/4점 MDCT는 그림 3과 같이 나타난다.

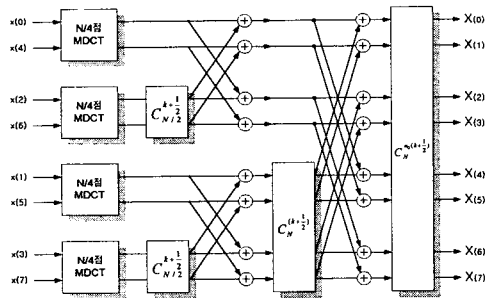


그림 3. N/4점 MDCT를 이용한 8점 MDCT

이러한 과정에서 사용되는 최소연산요소 즉 2점 MDCT는 2개의 입력에 짝수 부에만 계수를 곱하여 더하는 과정으로 그림 4와 같은 흐름선도를 갖는다.

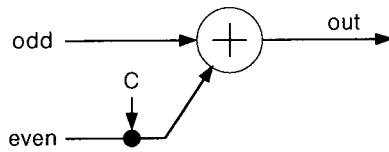


그림 4. MDCT 최소연산요소

3. 연산 단계의 일반화

위 절과 같이 MDCT/IMDCT의 연산과정에서 중복되는 부분을 묶음으로서 연산에 필요한 계수의 개수를 대폭 줄일 수 있다. 일반화 과정은 위의 과정에서 규칙을 찾는 것이다.

MDCT의 단계를 나누는 것은 $N=2^p$ 인 p 단계까지 적용할 수 있다. 즉 연산 단계 p 는 $\log_2 N$ 으로 구할 수 있다. 블록크기 N 의 MDCT의 계수 값은 $p = \log_2 N$ 인 p 까지 단계를 가지므로

256점이면 $\rightarrow 128 \rightarrow 64 \rightarrow 32 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2$ 이므로 $N/2^p$ 점 MDCT로 할 수 있다.

$\log_2(256)=8$, $8-1=7$ 따라서 전체 계수 값 256개

에서 256/128 즉 2개의 기본 계수 값(MDCT 최소연산요소) + 각 단계별 최고차 계수 값, 여기서는 8단계이므로 $\cos(256x)$, $\cos(128x)$, $\cos(64x)$, $\cos(32x)$, $\cos(16x)$, $\cos(8x)$, $\cos(4x)$, $\cos(2x)$ 가 각각 필요하다. 따라서 초항의 계수 값은 무조건 2개이고, 다음 단계의 수만큼 계수 값이 필요하게 된다. 2048점 MDCT의 경우 총 계수의 개수는 $\log_2 2048=11$ 이다. 따라서 $10+2=12$ 개의 계수 값만 있으면 2048점 MDCT를 구할 수 있다. 또한 각 단계의 입력순서는 앞 단계의 출력에 순서에 맞게 나오므로 역시 역비트순 입력을 적용해주면 된다. 또한 각 단계의 MDCT 최소연산요소 연산은 $N/2$ 만큼 수행하게 된다. 따라서 필요한 메모리는 $N/2$ 개이다.

4. 연산 량의 최소화

위 절과 같은 과정에서 연산에 필요한 계수의 량은 혁신적으로 감소했다. 다음은 식 1의 cosine계수인덱스 k 에 대해 최적화를 실시한다. 실제로 연산 량의 감소는 k 에 대해 최적화 함으로서 가능하다. k 인덱스의 변화에 따라 계수인덱스가 달라지는데, k 의 변화에 따라 각각 위의 단계별 연산 과정을 거치게되므로 만약 k 에 관계해서 중복부분이 있다면, 연산 량은 감소할 것임에 착안한다. 3 절의 일반화과정에서 각 단계별로

$$C_2^{(n-n_0)(k+\frac{1}{2})}, C_4^{(n-n_0)(k+\frac{1}{2})}, C_8^{(n-n_0)(k+\frac{1}{2})}, \dots, C_{2^p}^{(n-n_0)(k+\frac{1}{2})}$$

이므로 단계 p 에 대해 주기는 2^p 이 된다. 그런데 그림 3에서와 같이 매 단계들은 가산으로만 연결되고, 또한 최종결과에서는 실수부만 사용하므로 식 3은 cosine 함수만 남게되며, cosine함수의 주기적 이용하여 부호 성분을 제외하면 주기는 2^{p-1} 이 된다. 표 1은 이런 결과를 볼 수 있는 Matlab 처리결과이다.

	단계 0	단계 1	단계 2	단계 3	단계 4
k=0	1	0.7071	0.9239	0.9808	0.9952
k=1	-1	-0.7071	0.3827	0.8315	0.9569
k=2	1	-0.7071	-0.3827	0.5556	0.8819
k=3	-1	0.7071	-0.9239	0.1951	0.7730
k=4	1	0.7071	-0.9239	-0.1951	0.6344
k=5	-1	-0.7071	-0.3827	-0.5556	0.4714
k=6	1	-0.7071	0.3827	-0.8315	0.2903
k=7	-1	0.7071	0.9239	-0.9808	0.0980
k=8	1	0.7071	0.9239	-0.9808	-0.0980
k=9	-1	-0.7071	0.3827	-0.8315	-0.2903
k=10	1	-0.7071	-0.3827	-0.5556	-0.4714
k=11	-1	0.7071	-0.9239	-0.1951	-0.6344
k=12	1	0.7071	-0.9239	0.1951	-0.7730
k=13	-1	-0.7071	-0.3827	0.5556	-0.8819
k=14	1	-0.7071	0.3827	0.8315	-0.9569
k=15	-1	0.7071	0.9239	0.9808	-0.9952

표 1. cosine 인덱스 k에 따른 단계별 실수 출력결과

표 1에서 회색음영 부분만 계산해주면 되며, 나머지는 버퍼에 저장하였다가 사용할 수 있다. 또한 매 단계별 연산을 연결할 때 필요한 가산도 표 1에 의해 줄일 수 있다.

5. 성능평가 및 결론

위와 같은 처리과정에서 이론상 오차는 발생하지 않는다. 그러나 본 알고리즘을 테스트하는 시스템의 유효 비트 크기에 따라 오차는 다르게 나타날 수 있다. 본 논문에서는 Pentium II 266MHz windows NT40 시스템에서 Matlab 5.1과 VC 6.0으로 64비트 처리하여 검증하고, 구현되었다.

그림 5는 최적화하지 않은 MDCT와 최적화 후 MDCT의 Impulse 응답의 오차를 구한 Matlab 출력 파형이다.

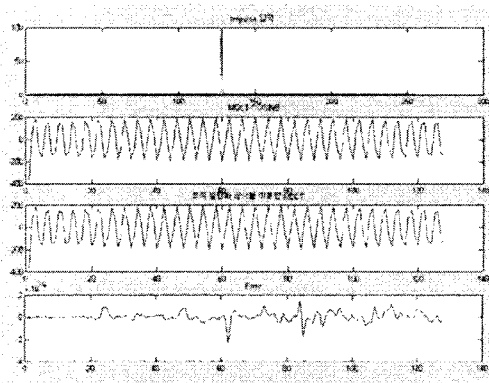


그림 5. Impulse 응답 Matlab simulation

그림 5에서 오차의 범위는 $\pm 1 \times 10^{-12}$ 으로 오디오용 D/A 변환기의(24bit) 오차보다 충분히 작음으로 무시할 수 있다. 그림 5에서 k 값이 증가함에 따라 오차가 증가하는 것은 최적화 시 MDCT의 승산의 누적가산에서 64비트 한계에 따른 버림에 의해 발생한 것으로 볼 수 있으며, 결과적으로 최적화 시 반올림 및 버림 오차가 적어 더욱 정밀하다고 할 수 있다.

연산량은 $k=0$ 일 때만 모든 단계를 계산하며, 이 결과를 버퍼에 저장하여 사용하므로, 단계 별로 다음과 같은 승산회수를 필요로 한다.

$$MA(p) = 2^{p-1} \quad (11)$$

단) $MA(p)$: 단계별 승산량
 p : 정수 연산단계 인덱스 ($0 < p < \log_2 N$)

따라서 초기 승산을 제외한 k 인덱스에 따른 총 승산량은 M_a 는 식 12와 같다.

$$M_a = \sum_{p=1}^Z MA(p) = \sum_{p=1}^Z 2^{p-1} \quad (12)$$

단) $Z = (\log_2 N) - 2$

그리고 초기 승산량($k=0$ 일 때) M_i 는 식 13과 같다.

$$M_i = \sum_{n=0}^P \log_2 \left(\frac{N}{4} \right) 2^n \quad (13)$$

단) $P = (\log_2 N) - 1$

따라서 2048점 MDCT의 경우

(초기 승산량) + (k 인덱스에 따른 승산량) = $1024 + 1024 = 2048$ 번의 승산으로 모든 처리를 할 수 있다. 가산의 경우도 중복부분을 제거할 수 있으나, 현재 Pentium급 PC는 모두 1 clock에 계산 가능하므로 여러 조건을 검증하는 코드를 작성할 경우 처리과정이 복잡하고, 성능개선이 별로 없으므로 단계별 연결에 가산을 모두 사용하면 2048점 MDCT의 경우 $12 \times 1024 = 12288$ 번 가산으로 모든 처리가 끝나게 된다.

최적화하지 않은 경우 $2048 \times 1024 = 2097152$ 번의 승산과, 가산으로 결과를 구할 수 있다. 따라서 최적화 전에 비해 승산에서 0.098%, 가산에서 0.58% 연산량만으로 결과를 구할 수 있는 효율을 보인다.

참고문헌

- [1] ISO/IEC JTC1/SC29/WG11 N1650, "IS 13818-7(MPEG-2 Advanced Audio Coding, AAC)" April 1997, p3.
- [2] J.Princen, A.Johnson, A.Bradley: "Subband/Transform Coding Using Filter Bank Designs Based on Time Domain Aliasing Cancellation", Proc. of the ICASSP 1987, pp. 2161-2164.
- [3] "Presented at the 101st Convention 1996 November 8-11 Los Angeles, California", AN AUDIO ENGINEERING SOCIETY PREPRINT. p28
- [4] ISO/IEC JTC1/SC29/WG11 N1650, "IS 13818-7(MPEG-2 Advanced Audio Coding, AAC)" April 1997, p49, p77.
- [5] Mark Kahrs, Karlheinz Brandenburg, "APPLICATIONS OF DIGITAL SIGNAL PROCESSING TO AUDIO AND ACOUSTICS", 1998 by Kluwer Academic Publishers.