ARM 프로세서를 이용한 MP3 인코딩용 고속 MDCT 구현

조 경 연(趙 庚 衍), 최 종 찬(崔 鍾 讚), 이 철 동(李 哲 東)
전자부품연구원 시스템IC연구센터
전화 : (0333) 610-4054 / 팩스 : (0333) 610-4048

Implementation of MDCT for MP3 using ARM Processor

Kyoung-Youn Cho, J Chan Choi, Chul-Dong Lee
Korea Electronics Technology Institute
System IC Research Center
E-mail : choky@nuri.keti.re.kr

Abstract

MDCT(Modified Discrete Cosine Transform) is one of the most compute-intensive operations in the MPEG audio coding standard. In this paper a fast algorithm to perform MDCT operation is presented. The algorithm presented in the MPEG audio coding standard requires $(N/2) \times N$ multiplications and $(N/2) \times (N-1)$ additions to generate the result, but the algorithm presented in this paper requires $(N/2) \times (N/2)$ multiplications and $(N/2) \times (N/2)$ additions to perform the same task. In this algorithm N should be multiple of 4. The algorithm was implemented using ARM processor and the processing time comparison between the original algorithm and the fast algorithm is presented.

Ⅰ. 서 론

MPEG 오디오 신호 코딩은 Motion Pictures Expert Group(MPEG)에 의해서 ISO/IEC 11172-3 International standard[1]로 제정되었다. MPEG 오디오 압축 방식은 스테레오 채널에 대해서 CD와 유사한 음질을 유지하면서 128 Kbps의 비트률로 오디오 신호를 압축한다. 본 논문에서는 MPEG-1 Layer 3(MP3)에서 사용되는 MDCT 연산을 고속으로 수행할 수 있는 알고리즘을 제안하고 그 구현 결과를 제시한다. MPEG 오디오 인코딩 과정에서 많은 연산량을 요구하는 부분은 Subband Filter 연산과 MDCT 연산이다. 실시간 MP3 인코더를 구현하기 위해서는 Subband Filter 연산과 MDCT 연산을 고속으로 수행해야 하는데, Subband Filter 연산에 대한 고속 알고리즘은 [2]에 제시되었다. 본 논문에서는 MDCT 연산을 위한 고속 알고리즘을 제시하고, 기존의 알고리즘과 고속 알고리즘을 32 비트 RISC 프로세서인 ARM 프로세서에서 수행하여 검증한다. Ⅱ장에서는 고속 MDCT 연산 알고리즘을 유도하고, Ⅲ장에서는 MPEG-1 Layer 3에 적용 예를 제시한다. Ⅳ장에서는 ARM 프로세서를 이용하여 알고리즘을 검증하고, 실험 결과를 제시한다.

Ⅱ. MDCT 고속 알고리즘 유도

MP3 인코딩에서 MDCT 연산은 식 (1)로 정의된다.

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

$$\text{for } i=0 \text{ to } \frac{N}{2}-1 \qquad (1)$$

식 (1)에서 N 은 short block 일 경우 12이고, long block, start block, stop block 일 경우 36이다. $x(k)$는 Subband Filter 연산의 결과에 적당한 window 연산을 취함으로써 구해지는 값이다.

Lemma 1 : $x'(k)$를 식 (2)와 같이 정의한다.

$$x'(k) = -x\left(k+\frac{N}{4}\right) \quad \text{for } 0 \le k \le \frac{3N}{4}-1$$

$$x\left(k-\frac{3N}{4}\right) \quad \text{for } \frac{3N}{4} \le k \le N-1 \qquad (2)$$

이때 식 (1)은 식 (3)으로 변환될 수 있다.

$$X(i) = (-1)^i \sum_{k=0}^{N-1} x'(k)\sin\left(\frac{\pi}{N}\left(k+\frac{1}{2}\right)(2i+1)\right) \qquad (3)$$

Proof : 식 (1)에서 $k' = k - N/4$ 로 대체하면 식 (4)로 변환할 수 있다.

$$X(i) = \sum_{k'=-N/4}^{\frac{3}{4}N-1} x\left(k'+\frac{N}{4}\right)\cos\left[\frac{\pi}{N}\left(k'+\frac{1}{2}+\frac{N}{2}\right)(2i+1)\right] \qquad (4)$$

식 (4)에서 summation 구간을 분리하면 식 (5)가 구해진다.

$$X(i) = \sum_{k'=-N/4}^{-1} x\left(k'+\frac{N}{4}\right)\cos\left[\frac{\pi}{N}\left(k'+\frac{1}{2}+\frac{N}{2}\right)(2i+1)\right]$$
$$+ \sum_{k'=0}^{\frac{3N}{4}-1} x\left(k'+\frac{N}{4}\right)\cos\left[\frac{\pi}{N}\left(k'+\frac{1}{2}+\frac{N}{2}\right)(2i+1)\right]$$
$$= S_1(i) + S_2(i) \qquad (5)$$

식 (5)에서 첫 번째 합인 $S_1(i)$에 대해서 $k = k' + N$ 로 대체하면 식 (6)이 구해진다.

$$S_1(i) = \sum_{k=\frac{3N}{4}}^{N-1} x\left(k-\frac{3N}{4}\right)\cos\left[\frac{\pi}{N}\left(k+\frac{1}{2}-\frac{N}{2}\right)(2i+1)\right] \qquad (6)$$

식 (6)을 식 (5)에 적용하면 식 (7)이 구해진다.

$$X(i) = \sum_{k=\frac{3N}{4}}^{N-1} x\left(k-\frac{3N}{4}\right)\cos\left[\frac{\pi}{N}\left(k+\frac{1}{2}-\frac{N}{2}\right)(2i+1)\right]$$
$$+ \sum_{k'=0}^{\frac{3N}{4}-1} x\left(k'+\frac{N}{4}\right)\cos\left[\frac{\pi}{N}\left(k'+\frac{1}{2}+\frac{N}{2}\right)(2i+1)\right] \qquad (7)$$

식 (7)에서 cosine 항은 식 (8)과 같이 변환된다.

$$\cos\left[\frac{\pi}{N}\left(k+\frac{1}{2}\pm\frac{N}{2}\right)(2i+1)\right] =$$
$$\cos\left[\frac{\pi}{N}\left(k+\frac{1}{2}\right)(2i+1)\pm\frac{\pi}{2}(2i+1)\right] =$$
$$\mp(-1)^i\sin\left(\frac{\pi}{N}\left(k+\frac{1}{2}\right)(2i+1)\right) \text{ (복호동순)} \qquad (8)$$

식 (8)을 식 (7)에 적용하면 식 (9)가 구해진다.

$$X(i) = (-1)^i \sum_{k=\frac{3N}{4}}^{N-1} x\left(k-\frac{3N}{4}\right)\sin\left[\frac{\pi}{N}\left(k+\frac{1}{2}\right)(2i+1)\right]$$
$$-(-1)^i \sum_{k=0}^{\frac{3N}{4}-1} x\left(k+\frac{N}{4}\right)\sin\left[\frac{\pi}{N}\left(k+\frac{1}{2}\right)(2i+1)\right] \qquad (9)$$

식 (9)에서 식 (1)의 cosine 항이 sine 항으로 변경된 것을 주의해야 한다. 식 (9)에 식 (2)를 적용하면, 식 (3)이 구해진다.

Lemma 2 : $y(k)$를 식 (10)과 같이 정의한다.

$$y(k) = x'(k) + x'(N-1-k) \qquad (10)$$

이때 식 (3)은 식 (11)로 변환될 수 있다.

$$X(i) = (-1)^i \sum_{k=0}^{\frac{N}{2}-1} y(k)\sin\left[\frac{\pi}{N}\left(k+\frac{1}{2}\right)(2i+1)\right] \qquad (11)$$

Proof : 식 (3)에서 summation 구간을 분리하면 식 (12)가 구해진다.

$$X(i) = (-1)^i \sum_{k=0}^{\frac{N}{2}-1} x'(k)\sin[\frac{\pi}{N}(k+\frac{1}{2})(2i+1)]$$
$$+(-1)^i \sum_{k=\frac{N}{2}}^{N-1} x'(k)\sin[\frac{\pi}{N}(k+\frac{1}{2})(2i+1)] \quad (12)$$

식 (12)의 두 번째 summation에 대해서 $k = (N-1)-k$ 를 적용하면, 식 (13)과 같이 변환된다.

$$(-1)^i \sum_{k=0}^{\frac{N}{2}-1} x'(N-1-k) \times$$
$$\sin[\pi(2i+1) - \frac{\pi}{N}(k+\frac{1}{2})(2i+1)] \quad (13)$$

식 (13)의 sine 항을 변환하면 식 (14)와 같이 변환된다.

$$(-1)^i \sum_{k=0}^{\frac{N}{2}-1} x'(N-1-k)\sin[\frac{\pi}{N}(k+\frac{1}{2})(2i+1)] \quad (14)$$

식 (14)를 식 (12)에 적용하면 식 (15)가 구해진다.

$$X(i) = (-1)^i \sum_{k=0}^{\frac{N}{2}-1} x'(k)\sin[\frac{\pi}{N}(k+\frac{1}{2})(2i+1)]$$
$$(-1)^i \sum_{k=0}^{\frac{N}{2}-1} x'(N-1-k)\sin[\frac{\pi}{N}(k+\frac{1}{2})(2i+1)] \quad (15)$$

식 (15)에 식(10)을 적용하면 식 (11)이 구해진다. Lemma 1과 Lemma 2 의 정의를 이용하면 $x(k)$로부터 $y(k)$를 구할 수 있는데, 식 (16)에 제시하였다.

$$\begin{aligned} y(k) &= -x(k+\frac{N}{4}) + x(\frac{N}{4}-1-k) \\ &\qquad \text{for } 0 \le k \le \frac{N}{4}-1 \\ &\quad -x(k+\frac{N}{4}) - x(\frac{5N}{4}-1-k) \\ &\qquad \text{for } \frac{N}{4} \le k \le \frac{N}{2}-1 \quad (16) \end{aligned}$$

Ⅲ. MPEG-1 Layer 3 에 적용

MPEG-1 Layer 3 에서는 4 종류의 block type을 사용하는데, long block, start block, stop block, short block이다. MPEG-1 Layer 3에서는 block type에 따라 식 (11)로 정의된 MDCT 연산의 N 값이 달라진다. Block type이 long, start, stop일 경우 식 (11)에서 N은 36이고, short일 경우 12이다. N 이 36일 경우 MDCT 연산은 식 (17)로 계산된다.

$$\begin{aligned} X(i) &= (-1)^i \sum_{k=0}^{17} y(k)\sin[\frac{\pi}{36}(k+\frac{1}{2})(2i+1)] \\ y(k) &= -x(k+9) + x(8-k) \quad \text{for } 0 \le k \le 8 \\ y(k) &= -x(k+9) - x(44-k) \quad \text{for } 9 \le k \le 17 \quad (17) \end{aligned}$$

N이 12일 경우 MDCT 연산은 식 (18)로 계산된다.

$$\begin{aligned} X(i) &= (-1)^i \sum_{k=0}^{5} y(k)\sin[\frac{\pi}{12}(k+\frac{1}{2})(2i+1)] \\ y(k) &= -x(k+3) + x(2-k) \quad \text{for } 0 \le k \le 2 \\ y(k) &= -x(k+3) - x(14-k) \quad \text{for } 3 \le k \le 5 \quad (18) \end{aligned}$$

Ⅳ. 구현 및 실험 결과

기존의 MDCT 알고리즘과 고속 MDCT 알고리즘을 비교하기 위해서 32비트 RISC 프로세서인 ARM 프로세서에서 각각의 알고리즘을 구현하여 실험하였다. 먼저, C 언어를 이용하여 알고리즘을 검증하였다. 검증된 C 언어 프로그램을 ARM assembly 언어로 변환하여 ARM 프로세서를 이용하여 동작을 검증하였고, 연산 시간을 표 1에 제시하였다.

표 1 실험 결과

구 분	연산 클록 수 (cycles)	연산 시간 (ms)
기존의 알고리즘	97589	2.440
본 논문의 알고리즘	31678	0.792

표 1은 36-point MDCT 연산에 대한 실험 결과이다. 연산 클록 수는 ARMulator를 이용하여 측정하였고, ARM 프로세서가 40 MHz의 클록으로 동작한다고 가정하고, 식 (19)를 이용하여 연산 시간을 계산하였다.

$$\text{연산 시간} = (\text{연산 클록 수}) \times \frac{1}{40MHz} \quad (19)$$

본 논문에서 제안된 알고리즘의 연산 시간이 기존 알고리즘의 연산 시간의 약 1/3로 감소하였다.

V. 결 론

본 논문에서는 MDCT 연산을 위한 고속 알고리즘을 ARM 프로세서를 이용하여 구현하였다. MPEG-1 Layer 3 인코딩을 위해서는 MDCT 연산이 필요한데, 기존의 알고리즘은 N-point 연산에 대해서 $(N/2) \times N$ 곱셈 연산과, $(N/2) \times (N-1)$ 덧셈 연산이 필요하다. 본 논문에서 제안된 알고리즘은 같은 연산에 대해서 $(N/2) \times (N/2)$ 곱셈 연산과 $(N/2) \times (N/2)$ 덧셈 연산이 필요하다. 덧셈 연산은 식 (11)과 식 (16)의 계산을 위해서 필요한데, 식 (11)을 계산하기 위한 덧셈 연산은 $(N/2) \times (N/2-1)$ 이고, 식 (16)을 계산하기 위한 덧셈 연산은 $(N/2)$ 이다. 본 논문에서 N 은 4의 배수가 되어야 한다. MPEG-1 Layer 3의 경우 long block에 대해서 곱셈 연산 횟수는 648에서 324로 줄었고, 덧셈 연산 횟수는 630에서 324로 줄었다. 곱셈 연산 횟수와 덧셈 연산 횟수가 약 1/2 로 줄었고, 실험 결과 ARM 프로세서의 연산 시간은 약 1/3 로 줄었다.

참고문헌

[1] ISO/IEC 11172-3,"Coding of moving pictures and associated audio for digital storage media at up to about 1.5 MBit/s-Part3: Audio,"ISO/IEC JTC 1/SC 29, 5/20/1993.

[2] Konstantinides,"Fast Subband Filtering in MPEG Audio Coding,"*IEEE Signal Processing Letters*, vol. 1. No. 6. pp 26-28. Feb. 1994.

[3] 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. No. 4, pp 116-118, Apr. 1996.

[4] K. R. Rao and P. Yip, *Discrete Cosine Transform*. New York: Academic, 1990.

[5] Gluth R, Regular, "FFT-Related Transform Kernels for DCT/DST-based polyphase filter banks," ICASSP 91, pp.2205-8 vol.3.