

## MPEG-1 Layer III 오디오 디코더의 실시간 DSP 구현

김 시 호, 권 홍 석, 배 건 성

경북대학교 전자·전기공학부

전화 : (053) 950-5527 / 팩스 : (053) 950-5505

### Real-Time DSP Implementation of MPEG-1 Layer III Audio Decoder

Si Ho Kim, Hong Seok Kwon, Keun Sung Bae

School of Electronic and Electrical Engineering Kyungpook National University

E-mail : si5@mmirl1.knu.ac.kr

#### Abstract

본 논문에서는 높은 압축률과 고음질을 제공하는 MPEG-1 Layer III 오디오 디코더를 고정소수점 DSP인 TMS320C6201을 이용하여 실시간으로 동작하도록 구현하였다. ISO/IEC에서 제공하는 부동소수점 C 프로그램을 음질의 손실 없이 고정소수점 연산으로 변환하였고 실시간 동작을 위하여 최적화 작업을 수행하였다. 연산의 정확성을 높이기 위해서 Descaling 모듈에 중점을 두어 부동소수점 연산을 고정소수점 연산으로 변환하였고 IMDCT 모듈과 Synthesis Polyphase Filter Bank 모듈에 대해 고속 알고리즘을 적용하여 연산량과 프로그램 크기를 크게 줄일 수 있었다. 구현된 디코더는 TMS320C6201 DSP가 수행할 수 있는 최대 연산량의 26%만으로 실시간 동작이 가능하였고 부동소수점 연산 결과와 고정소수점 연산 결과를 비교하여 60 dB 이상의 높은 SNR을 가짐을 확인하였다. 또한 사운드 입출력과 호스트 통신을 통하여 EVM 보드에서 실시간으로 동작함을 확인하였다.

#### I. 서론

디지털 오디오의 장점은 아날로그에 비해 음질이 훨씬 깨끗하고, 음질의 변화가 거의 없다는 것이다. 그러나 아날로그 오디오 신호를 샘플링해서 얻어지는 고음질의 디지털 오디오 신호는 데이터량이 아주 많아서 막대한

용량의 저장매체를 요구하며, 전송시 매우 넓은 주파수 대역을 차지하게 된다. 따라서 오디오 신호를 음질의 손실 없이 효과적으로 압축하는 기술이 필요하게 되었다. 이러한 요구를 충족시키는 오디오 압축기술 중에서 현재 세계적으로 가장 널리 사용되고 있는 것이 유럽의 MUSICAM 방식에 기반을 둔 MPEG이다[1]. 지금까지 상용화된 MPEG-1 오디오 압축 방식 중에서 가장 좋은 음질과 낮은 전송률을 가진 것이 Layer III이고 이는 CD 수준의 오디오 신호를 12:1 이상의 압축률로 재생가능하다[2-3].

본 논문에서는 MPEG-1 Layer III 오디오 디코더를 ISO/IEC 11172-3[4]에서 제공하는 부동소수점 연산 프로그램을 이용하여 고정소수점 DSP인 TMS320C6201 EVM 보드에서 실시간으로 동작하도록 구현하였다. 사용된 TMS320C6201 DSP는 2개의 16\*16 곱셈기와 6개의 arithmetic logic unit으로 이루어진 총 8개의 functional unit을 가지고 있어서 한 클럭 사이클 당 최대 8개의 명령을 동시에 수행할 수 있다. 따라서 200 MHz의 클럭 rate에서 최대 1600 MIPS(Million Instructions Per Second)의 성능을 발휘할 수 있다. 또한 칩 내부에는 32개의 32-bit 범용 레지스터와 64 kbyte의 데이터 메모리와 프로그램 메모리가 있다[5].

본 논문의 구성은 다음과 같다. 먼저 2장에서 MPEG-1 Layer III 오디오 디코더의 디코딩 과정에 대해서 간략히 설명하고 3장에서 부동소수점 연산을 고정소수점 연산으로 변환하는 방법과 실시간 처리를 위한 최적화 방안에 대해서 설명한다. 4장에서는 구현된 고정소

수점 MPEG-1 Layer III 오디오 디코더의 성능 및 결과를 제시하고 마지막으로 결론을 맺는다.

## II. MPEG-1 Layer III 오디오 디코더

MPEG-1 Layer III 오디오 비트스트림의 각 프레임은 압축된 1152개의 PCM 샘플을 복원하는데 필요한 정보와 오디오 데이터를 담고 있으며 그림 1과 같이 헤더, CRC, 부가 정보(side information), 주 데이터(main data), 그리고 보조 데이터(ancillary data)로 나누어진다. 헤더 부분은 프레임의 정보를 담고 있으며 부가 정보는 인코딩 과정에서 사용된 block type, 허프만 데이터를 디코딩하는데 필요한 정보 등을 가지고 있다. 주 데이터 부분은 두 개의 그래놀로 구분되며 각각은 인코딩된 오디오 데이터를 담고 있다.

입력된 비트스트림은 그림 2와 같은 디코딩 과정을 거쳐 최종 PCM 샘플값을 출력하게 된다. Descaling 모듈은 인코더의 MDCT 과정에서 만들어진 frequency line을 복원하는 과정으로서 Huffman Decoding 모듈에서 계산된 frequency line과 Scalefactor Decoding 모듈에서 복원된 scalefactor를 이용하여 수행된다. 재배열(Reordering) 모듈에서는 short window에 대해 인코딩 과정의 MDCT 모듈에서 허프만 인코딩의 효율을 높이기 위해 바꾸어진 frequency line의 배열을 원래대로 바꾸는 과정이 수행된다.

Alias Reduction 모듈에서는 각 서브밴드마다 8개의 butterfly를 계산함으로써 수행되고 그 결과는 IMDCT 모듈에서 식 (1)을 이용하여 polyphase filter 서브밴드의 각 샘플로 변환된다[6].

$$x_i = \sum_{k=0}^{\frac{n}{2}-1} X_k \cos\left(\frac{\pi}{2n} \left(2i+1 + \frac{n}{2}\right)(2k+1)\right) \quad (1)$$

, for  $i=0$  to  $n-1$

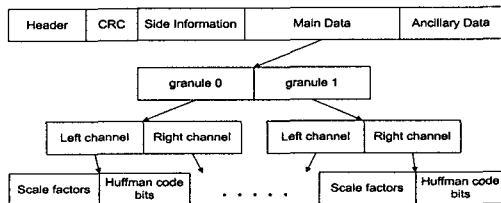


그림 1. MPEG-1 Layer III의 프레임 구성  
Fig 1. Frame structure of MPEG-1 Layer III

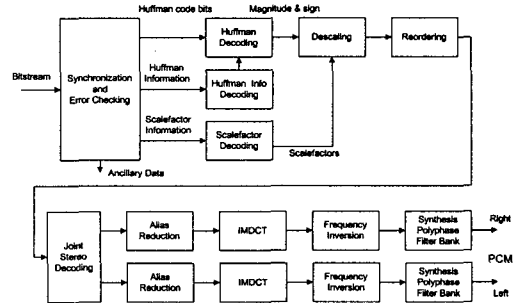


그림 2. MPEG-1 Layer III 오디오의 디코딩 과정  
Fig 2. Decoding process of MPEG-1 Layer III audio

이렇게 구해진 각 샘플값을 인코딩에서 사용된 창형에 따라 windowing한 다음 이전 그래놀의 값과 overlap-add를 함으로써 32개의 벡터를 얻을 수 있고 32개 서브밴드의 각 샘플을 Synthesis Polyphase Filter Bank를 통과 시킴으로써 최종적으로 32개의 연속적인 오디오 샘플을 얻을 수 있다.

## III. 실시간 처리를 위한 최적화 과정 및 고정소수점 변환

이 장에서는 부동소수점으로 이루어진 C 프로그램을 고정소수점으로 변환하는 과정에 적용시킨 방법을 설명하고, MPEG-1 Layer III 디코더를 TMS320C6201 EVM에서 실시간으로 동작시키기 위한 프로그램 최적화 방안에 대하여 서술한다.

Descaling 과정에서 계산되는 허프만 디코딩된 값의 크기는 그 범위가 최대 8191이고, 4/3승을 하면 결과적으로 최대 165140.3719가 된다. 이는 32 비트 레지스터를 이용하더라도 최대 Q-13 format밖에 사용할 수 없다. 이런 동적 범위의 크기를 줄이기 위하여 그림 3과 같은 형태로 표를 만들어 사용하였다. 상위 4 비트는 각 표 값의 2의 지수값에 해당되며, 나머지 27 비트는 밑수값으로서 Q-format을 이용하여 고정소수점으로 표현한 것이다.



그림 3. Descaling 과정에서의 4/3승 표 구성  
Fig 3. Table structure of power of 4/3 for descaling processing

TMS320C6201 DSP에서는 Descaling 과정에서 필요한 2의 지수승 연산을 제공하지 않으므로 식 (2)와 같이 수식을 정리함으로써 지수값의 소수부분을 단지 1/4승, 2/4승, 3/4승만으로 표현 가능하게 한다.

$$x_{r_i} = \text{sign}(is_i) \cdot \text{abs}(is_i)^{\frac{4}{3}} \cdot \frac{2^{\frac{1}{4}(global\_gain[gr][ch]-210)}}{2^{(multiplier \cdot (scalefac\_l[gr][ch][sfb]+preflag[gr][ch] \cdot pretab[sfb]))}} = (\text{sign}(is_i) \cdot \text{abs}(is_i)^{\frac{4}{3}} \cdot 2^{\frac{b}{4}}) \ll a \quad (2)$$

그러나 허프만 디코딩이 이루어진 값은 0~8191의 값을 가지게 되므로 이 값을 위에서 언급한 방법으로 표현하더라도 최소한 Word32(4 bytes) 크기의 데이터가 8192개 필요하게 된다. 이는 표의 크기가 크므로 식 (3) 처럼 허프만 디코딩된 결과의 4/3승 부분을 8로 나누어서 이 값을 저장함으로써 표의 크기를 줄일 수 있고, 그 오차를 선형 보간하여 보상한다[7].

$$\text{abs}(is_i)^{\frac{4}{3}} = \text{abs}\left(\frac{is_i}{8} \times 8\right)^{\frac{4}{3}} = \text{abs}(is_i' \times 8)^{\frac{4}{3}} = \text{abs}(is_i')^{\frac{4}{3}} \times 16 = \text{abs}(is_i')^{\frac{4}{3}} \ll 4 \quad (3)$$

식 (1)을 이용하여 수행되는 IMDCT 과정은 계산량이 많아서 연산 속도를 증가시키기 위한 알고리즘이 반드시 필요하다. 일반적으로 DCT 연산에 대한 고속화 알고리즘으로는 factorization 방법[8]과 Fast Discrete Cosine Transform(FDCT) 방법[9] 등이 있다. Short window의 경우에는 IMDCT의 수가 12이며 long window의 경우에는 36인데, 이것은 2의 지수승을 요구하는 알고리즘인 FDCT 방법만으로는 효과적으로 연산 속도를 증가시킬 수 없다. 따라서 short window인 경우에는 3-point에 대해서 factorization을 수행하고 그 결과를 가지고 FDCT에 적용하였으며, long window인 경우에는 9-point에 대해서 factorization을 수행하고 그 결과를 FDCT에 적용하였다.

Synthesis Polyphase Filter Bank는 전체 MPEG-1 오디오 디코딩 과정에서 가장 연산량이 많이 소요되는 모듈로서 IDCT 연산과 유사한 식 (4), (5)에 의해서 구해진다. 이 계산 과정도 앞에서 언급한 IMDCT 과정에서 수행한 것과 마찬가지로 FDCT 방법을 적용하여 수행속도를 증가시켰다. 그리고 cosine 함수의 대칭성을 이용하여 64개의 V[i] 대신에 32개의 V[i] 값만으로 연산이 가능하게 하였다.

$$N_{ik} = \cos\left[(16+i)(2k+1)\frac{\pi}{64}\right] \quad (4)$$

, for  $0 \leq i \leq 63$ ,  $0 \leq k \leq 31$

$$V[i] = \sum_{k=0}^{31} N_{ik} S_k \quad (5)$$

#### IV. MPEG-1 Layer III 오디오 디코더의 실시간 DSP 구현 결과

이 장에서는 3장에서 설명한 방법으로 구현한 프로그램을 이용하여 고정소수점 연산 결과를 부동소수점 연산 결과와 비교하여 고정소수점 연산 결과의 정확성을 평가하고 최적화 알고리즘을 적용한 프로그램의 크기와 수행 사이클 수를 측정하여 최적화 정도를 확인하였다. 실험 비트스트림으로는 표 1에 주어진 것처럼 Fraunhofer IIS의 MPEG-1 Layer III 실험 비트스트림을 사용하였다. 표 2는 3가지 실험 비트스트림에 대한 결과를 비교한 것이다. 이때 SNR은 식 (6)으로 정의되며,  $PCM_{fixed}$  는 고정소수점 연산으로,  $PCM_{float}$  는 부동소수점 연산으로 계산된 최종 PCM 샘플값을 나타낸다.

$$SNR[dB] = 20 \times \log_{10} \frac{|PCM_{fixed}|}{|PCM_{float} - PCM_{fixed}|} \quad (6)$$

표 2에서 보듯이 SNR은 전체적으로 상당히 높게 나타나고 실제로 청취하더라도 부동 소수점 연산 결과와 차이점을 느낄 수 없으며 파형에서도 그 차이를 볼 수 없다.

표 1. 실험에 사용된 비트스트림

	bit rate	채널	sample rate
비트스트림 A	32kbps~320kbps	모노	48kHz
비트스트림 B	64kbps	모노	48kHz
비트스트림 C	128kbps	스테레오	44.1kHz

표 2. 테스트 비트스트림에 대한 오차 및 SNR 비교

	최대 오차 비트	SNR[dB]
비트스트림 A	2	66.5737
비트스트림 B	2	65.4235
비트스트림 C	L:3 R:4	L:61.4032 R:61.5354

구현된 디코더의 성능을 확인하기 위해서 메모리 크기와 수행 속도 부문에 대하여 분석하였다. 표 3은 사용된 메모리의 크기를 나타낸다.

표 3. 메모리 크기

	메모리 크기 [kbyte(kWord)]
프로그램 ROM	27.07 (6.77)
데이터 ROM	12.53 (3.13)
데이터 RAM (stack 포함)	39.77 (9.94)

수행 속도 면에서는 표 4에서 보는 것처럼 디코더의 프레임당 수행 사이클수를 측정함으로써 최적화의 정도를 확인하였다. 최대 sampling rate인 48 kHz에서 한 프레임의 시간은 24 msec이고 디코딩 연산에 걸리는 시간은 최대 6.29 msec이므로 한 프레임 수행 연산량은 TMS320C6201 DSP가 수행할 수 있는 최대 연산량의 26.21%에 해당된다.

표 4. 프레임당 수행 시간

	수행 시간 [clock(msec)]
Average	1,255,087 (6.28)
Maximum	1,257,393 (6.29)
Minimum	1,235,316 (6.18)

### V. 결론

본 논문에서는 TMS320C6201 EVM 보드에서 실시간으로 동작하는 고정소수점 MPEG-1 Layer III 오디오 디코더를 구현하였다. 부동소수점 C 프로그램을 TMS320C6201 DSP에서 동작하도록 고정소수점 연산 프로그램으로 변환하고 실시간 동작을 위하여 최적화 작업을 수행하였다. 연산량과 메모리 크기를 줄이기 위하여 IMDCT 모듈과 Synthesis Polyphase Filter Bank 모듈 등에 중점적으로 최적화 알고리즘을 적용하였고 그 결과 최종 프로그램의 ROM의 크기는 6.77 kWord, 데이터 ROM의 크기는 3.13 kWord, 데이터 RAM의 크기는 9.94 kWord이었다. 구현된 디코더의 수행결과 TMS320C6201 DSP가 수행할 수 있는 최대 연산량의 26.21%만으로 충분히 실시간 동작함을 확인하였다. 또한 부동소수점 연산과 고정소수점 연산의 출력 PCM값을 비교하여 60 dB

이상의 높은 SNR를 가짐을 확인함으로써 고정소수점 연산의 정확성을 검증하였고 EVM 보드에서 실시간으로 동작함을 확인하였다.

### 참 고 문 헌

- [1] T. Painter, A. Spanial, "A Review of Algorithms for Perceptual Coding of Digital Audio Signals," IEEE 13th International Conference on Digital Signal Processing Proceedings, DSP 97., Vol. 1, pp. 179-208, 1997
- [2] P. Noll, "MPEG Digital Audio Coding," IEEE Signal Processing Magazine, Sep., 1997
- [3] D. Pan, "A Tutorial on MPEG/Audio Compression," IEEE Trans. on Multimedia, Vol. 2, No. 2, pp. 60-74, 1995
- [4] ISO/IEC 11172-3 "Information technology-Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s - Part 3 : Audio," First edition, 1993
- [5] Texas Instruemtns, TMS320C6201 CPU and Instruction Set Reference Guide
- [6] J. Princen, A. Bradley, "Analysis/Synthesis Filter-bank Design Based on Time Domain Aliasing Cancellation," IEEE Trans. on Acoust. Speech, and Signal Processing, Vol. ASSP-34, pp. 1153-1161, 1986
- [7] 김진원, 정남훈, 김준석, 이근섭, 이충용, "MPEG-1 계층 III 오디오 복호화기의 VLSI 설계," 제 12회 신호처리합동학술대회, pp. 847-850, 1999
- [8] <http://www.iro.umontreal.ca/~boyerf/IMDCT/>
- [9] Byeong Gi Lee, "A New Algorithm to Compute the Discrete Cosine Transform," IEEE Trans. on Acoust, Speech and Signal Processing, Vol. ASSP-32, No. 6, pp. 1243-1245, 1984