

AC-3 복호화 알고리즘의 고정소수점 오차 최적화

°이근섭*, 유선국**, 고우석*, 윤대회*, 차일환*

* 연세대학교 전자공학과, ** 연세대학교 의용공학과

Fixed-point Error Optimization of AC-3 Decoding Algorithm

°Keun-Sup Lee*, Sun-Kook Yoo**, Woo-Suk Ko*, Dae-Hee Youn*, Il-Hwan Cha*

* Dept. of Electronic Eng., Yonsei Univ., ** Dept. of Medical Eng., Yonsei Univ.

요 약

최근 미국 내 표준안으로서 많이 사용되고 있는 AC-3 오디오 알고리즘은 그 복잡성으로 인하여 실시간 구현을 위해서는 프로세서로 구현하는 것이 적합하다. AC-3 복호화 알고리즘은 많은 부분이 실수연산으로 이루어져 있으므로 소수점을 고려한 연산이 필요한데, 프로세서로 구현할 때는 적은 비용과 빠른 속도로 실수연산을 수행하기 위해서 부동소수점보다는 고정소수점 연산이 유리하다. 그러나 고정소수점 연산시 발생하는 유한 단어길이 효과로 인하여 양자화 오차가 발생하므로 복호화된 오디오 신호의 음질저하를 최소화하기 위해서는 최적화가 필요하다. 본 논문에서는 AC-3 복호화 알고리즘의 부분별 양자화 오차를 분석하고 그 결과가 가장 많은 오차를 발생시키는 역 TDAC 변환의 오차를 최적화하였다. Fast TDAC 변환이 FFT로 이루어져 있으므로 고정소수점 연산시 오차가 적은 FFT 구조를 제안하였다. 제안된 구조를 사용하여 AC-3 고정소수점 복호화기를 C언어를 사용하여 구현하였으며, AC-3 부동소수점 복호화기와 최종 PCM을 비교하여 그 성능을 평가하였다.

I. 서 론

오늘날 많이 사용되고 있는 디지털 오디오의 장점은 아날로그에 비해 음질이 훨씬 깨끗하고, 음질의 변화가 거의 없다는 것이다. 또한 디지털 오디오는 영화나 컴퓨터와 같은 다른 매체와의 호환성이 뛰어나므로 계속해서 그 응용범위가 확장되고 있다. 그러나 고품질의 디지털 오디오 신호는 정보량이 크게 증가하여 압축하지 않으면 전송과 저장을 위한 비용이 크다는 단점이 있다.

디지털 오디오 신호를 효과적으로 압축하는 방식은

80년대에 접어들면서부터 미국의 AT&T, Dolby Lab.과 유럽의 IRT, Philips, CCETT와 일본의 Sony 등의 여러 연구소에서 개발되어 왔다. 이들 고품질 오디오 부호화 알고리즘은 오디오 신호를 부호화 후 복호화했을 때 CD 수준의 고품질을 제공해야 한다. 이들은 공통적으로 신호의 지각적 중복성을 제거하기 위해 사람의 청각적 특성을 이용한 압축방식과 신호의 통계적 중복성을 제거하기 위한 기존의 데이터 압축방식을 결합한 형태를 갖는다.

AC-3 복호화 알고리즘은 하이브리드 비트 할당 방식이라는 알고리즘의 특성상 복호화기의 계산량이 다소 많기 때문에 이를 실시간으로 구현하기 위해서는 그 복잡성으로 인해서 프로세서로 구현하는 것이 적합하며, 처리속도와 칩 면적 등의 비용을 최소화하기 위해서는 부동 소수점 연산보다 고정 소수점 연산을 사용한 구현이 유리하다. 그러나 고정 소수점 연산을 사용하면 유한단어길이 효과 (Finite Wordlength Effect) 로 인한 양자화 오차가 발생하므로 복호화된 오디오 신호에 왜곡이 생길 수 있는 위험이 있다.

본 논문에서는 고정 소수점 연산으로 발생하는 오디오 신호의 왜곡을 최소화 하기 위해 유한단어길이 효과로 인한 양자화 오차를 최적화하였다. 이를 검증하기 위하여 C언어를 사용하여 AC-3 오디오 부동/고정 소수점 복호화기를 구현하였으며, 두 복호화기의 복호화된 오디오 신호를 비교하여 최적화된 고정 소수점 복호화기의 성능을 평가하였다.

II. AC 3 오디오 알고리즘

AC-3 오디오 복호화 알고리즘은 크게 나누어서 비트정보 추출과정, 지수부 복호화 과정, 다채널 처리 과정, 역변환 과정 등으로 이루어진다[1].

비트정보 추출과정은 입력받은 비트열에 대해 여러 가지 정보를 추출해내는 과정이다. 이렇게 얻은 비트 정보를 사용하여 지수부를 복호화한다.

복호화된 지수부를 이용하여 비트할당 정보를 계산한 후에, 가수부를 비트열로부터 추출하여 MDCT (Modified Discrete Cosine Transform) 계수를 복원한다. 가수값을 양자화하기 위한 비트 할당은 지수값을 사용하여 구성된 간략화된 스펙트럼에 대해 심리 음향 모델을 적용하여 결정한다. 이러한 방법을 사용하면, 복호화 단계에서도 전송 받은 지수값들만으로 다시 비트 할당 정보를 만들어 낼 수 있기 때문에, 비트 스트림에 비트 할당에 대한 정보를 수록하지 않아도 되는 장점이 있다.

마지막으로 MDCT 계수를 이용하여 합성필터링 과정을 거치면 복호화된 오디오 신호를 얻을 수 있다. AC-3 복호화기의 합성필터링은 IMDCT (Inverse MDCT) 로 구성된다. MDCT는 시간 영역 에일리어징을 제거하는 TDAC (Time Domain aliasing Cancellation) 변환에 기초한 방식으로 FFT (Fast Fourier Transform)을 사용하여 계산량을 크게 줄일 수 있는 고속 알고리즘을 적용시킬 수 있다[2]. 또한 매우 향상된 주파수 선택성을 갖는다.

III. 고정 소수점 알고리즘 분석

3.1 고정 소수점 연산

AC-3 오디오 복호화 알고리즘은 실수 연산으로 이루어져 있으므로 실수의 소수점을 고려한 연산을 수행해야 한다. 실수의 소수점을 나타내는 방법은 크게 부동 소수점과 고정 소수점으로 나뉜다.

고정 소수점은 모든 실수를 제한된 길이의 정수로 표현하므로 고정 소수점 연산을 사용할 경우 지수부와 가수부를 나누어서 연산하는 부동 소수점 연산에 비해 산술 연산이 비교적 간단하여 전력 소모가 적고 속도도 빠르다. 따라서 하드웨어로 구현시 비용이 낮아진다는 큰 장점을 가진다. 그러나 고정 소수점 연산을 사용하면 표현할 수 있는 수의 크기가 제한되기 때문에 연산에 사용되는 데이터에는 양자화 오차 (Quantization Error) 가 존재하게 되는 단점이 있다. 이러한 현상을 유한단어길이 효과 (Finite Wordlength Effect) 라고 부르는데, 이는 결국 복호화된 오디오 신호가 왜곡되는 효과를 발생시키므로 이를 줄이기 위한 최적화 과정이 필요하다.

3.2 고정 소수점 연산 오차

고정 소수점 연산에서는 앞에서 언급한 바와 같이

유한단어길이 효과가 실수 값에 영향을 미치므로 양자화 오차가 발생한다. 이렇게 발생하는 오차는 실수 값을 표현할 때에도 물론 생기지만 산술 연산을 수행한 후 더 커지는 경향이 있다.

고정 소수점으로 표현된 두 개의 실수 값이 서로 곱해질 때는 식 (1)과 같은 과정을 거치게 된다. 식 (1)에서 보는 바와 같이 40-bit의 중간 결과 값이 20-bit의 최종 결과로 반올림될 때 오차가 발생하는 것을 발견할 수 있다.

$$\begin{aligned} [X_1]_Q \times [X_2]_Q &= (X_1 + e_1) \times (X_2 + e_2) \\ &= X_1X_2 + e_1X_2 + e_2X_1 + e_1e_2 \quad (40 \text{ bits}) \\ &\Rightarrow (\text{반올림}) \Rightarrow X_1X_2 + e' + e_3 \quad (20 \text{ bits}) \\ &\dots\dots\dots (1) \end{aligned}$$

where $[]_Q =$ 양자화, $e_1, e_2 =$ 양자화오차,
 $e' = e_1X_2 + e_2X_1 + e_1e_2$, $e_3 =$ 반올림오차

위의 식 (1)에서와 같이 고정 소수점으로 곱셈 연산을 할 때마다 $e_1X_2 + e_2X_1 + e_1e_2 + e_3$ 만큼의 오차가 발생한다. 그러나 더 큰 문제는 곱셈의 결과가 다시 곱셈 연산의 입력으로 들어갈 때 벌어진다. 두번째 곱셈에서는 앞서 얻어진 오차를 가지고 더 큰 오차를 만들어 내기 때문이다. AC-3 알고리즘은 수많은 곱셈 연산을 포함하기 때문에 고정 소수점 연산을 사용해서 복호화를 할 경우, 많은 오차를 내포할 수밖에 없다. 따라서 이러한 오차를 줄이기 위한 방안이 필요하다.

첫 번째 방안으로 위의 식 (1)에서 오차가 적은 반올림 방법을 선택하는 것이다. e' 의 경우에는 고정 소수점 연산에서 피할 수 없는 오차이므로 줄이기 힘들다. 그러나 e_3 는 반올림 방법을 개선함으로써 어느 정도 오차를 줄일 수 있다.

본 논문에서 선택한 반올림 방법은 Round-to-nearest-odd 방법이다. [4] Round-to-nearest-odd 방법을 대략적으로 설명하면 다음과 같다. 일반적인 반올림 방법과 같이 0.5보다 크면 올림을 하고 0.5보다 작으면 내림을 하는 것은 같지만, 0.5이면 LSB (Least Significant bit) 을 보고 결정을 한다. 즉, LSB가 0이면 올림을 하고 1이면 내림을 하는 것이다. 이 Round-to-nearest-odd 방법의 장점은 오차의 평균이 '0' 이라는 것이다. [4] 따라서 반올림에 대한 오차를 최대한 줄일 수 있다.

그리고 두번째 방안은 AC-3 오디오 복호화 알고리즘 자체를 최적화해서 곱셈의 수를 줄이는 것이다. 전

계 곱셈의 수를 줄임으로써 위에서 설명한 양자화 오차를 줄일 수 있다.

IV. 고정 소수점 오차의 최적화

4.1 고정 소수점 오차의 분포

본 논문에서는 AC-3 알고리즘의 블록별 고정 소수점 연산 오차를 얻기 위하여 AC-3 부동/고정 소수점 복호화기를 C언어를 사용하여 구현하고 이를 사용하여 모의 실험을 수행하였다. 고정 소수점 복호화기는 20-bit의 정수형을 사용하였으며, 시험 비트열들을 입력으로 하여 부동 소수점 복호화기와 비교하여 각 블록별로 고정 소수점의 최대 오차를 계산하였다. 얻은 결과는 그림 1과 같다.

그림 1에서 알 수 있듯이 IMDCT 루틴에서 발생하는 고정 소수점 오차의 최대값은 전체 오차의 약 89.1% (2.6561×10^{-5}) 를 차지하는 것으로 나타났다. 이것은 IMDCT 루틴에서 대부분의 오차가 발생한다는 것을 뜻한다. 따라서 고정 소수점 복호화기에서의 오차 최적화는 IMDCT 루틴에 집중되었을 때 가장 큰 효과를 얻을 수 있다.

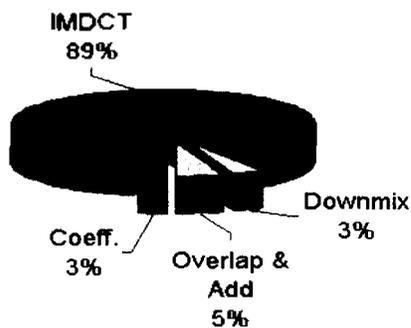


그림 1. 알고리즘 블록별 고정 소수점 오차

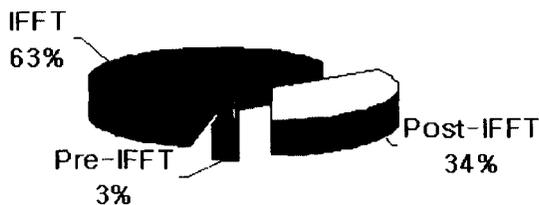


그림 2. IMDCT의 루틴별 최대 오차

4.2 Inverse MDCT 최적화

본 논문에서 사용된 IMDCT의 고속 알고리즘은 Pre-IFFT, IFFT, Post-IFFT의 3가지 루틴으로 나누어진 다. [2] 따라서 가장 많은 오차를 발생시키는 IMDCT를 이러한 3가지 부분으로 나누어 다시 오차를 분석하였다. 결과로 얻은 IMDCT의 루틴별 오차는 그림 2와 같다.

IFFT Radix-2 DIT			
Mean	STD	Max	Min
-7.08927e-07	6.48311e-06	1.64736e-05	-1.48182e-05
IFFT Radix-4 DIT			
Mean	STD	Max	Min
-5.89717e-07	6.02072e-06	1.42488e-05	-1.48182e-05
IFFT Radix-2 DIF			
Mean	STD	Max	Min
-7.08927e-07	7.3527e-06	2.21956e-05	-1.73288e-05

표 1. FFT 구조에 따른 오차의 통계값 (128-point IFFT)

그림 2에 의하면 Inverse FFT 루틴에서의 오차가 전체 IMDCT 오차의 약 63%로 절반 이상을 차지하는 것을 확인할 수 있다. 따라서 IMDCT에 대한 최적화는 IFFT 루틴의 최적화를 통해서 얻어질 수 있다. 그런데 여기서 특이한 것은 Pre-IFFT와 Post-IFFT가 연산 형태는 두 개의 곱셈 결과를 더하는 것으로 똑같은 연산이지만 오차가 크게 차이가 나는 것이다. 이것은 Post-IFFT 루틴이 IFFT 루틴에서 발생한 큰 오차를 곱셈의 입력으로 받기 때문에 같은 연산을 하더라도 결과적으로 더 큰 오차 값이 얻어진다. 따라서 IFFT 루틴의 오차가 줄어들면 Post-IFFT 루틴의 오차도 함께 줄어들게 된다는 것을 알 수 있다.

IFFT 루틴을 최적화시키기 위해 다음과 같은 3가지 FFT 구조를 선택하고 128개의 복소수 데이터 20개를 사용하여 오차의 통계 값을 얻었다. 이 실험에서 사용한 FFT의 구조는 Radix-2 DIT (Decimation In Time) FFT, Radix-4 DIT FFT, 그리고 Radix-2 DIF (Decimation In Frequency) FFT이다. 여기에서 128개의 데이터에 대한 Radix-4 결과는 그림 3과 같은 구조를 사용하여 얻었다.

표 1은 실험을 통해 얻은 오차의 통계값이다. Mean은 오차의 평균값이며, STD는 오차의 표준 편차, Max와 Min은 각각 오차의 최대값과 최소값을 나타낸 것이다. 위의 실험에 의하면 DIT가 DIF에 비해 좋은 성능을 가지며, 또한 Radix-2에 비해 Radix-4가 나은 성능을 가진다는 결론을 얻을 수 있다. 따라서 가장 훌륭한 성능을 보이는 구조는 Radix-4 DIT라고 결론지을 수 있다.

Radix-4 구조가 Radix-2에 비해 좋은 성능을 보여주는 이유는 FFT Stage 수의 차이에서 나타난다. 4-point FFT를 예로 들면, Radix-2는 결과값을 얻기 위하여 2개의 Stage를 거쳐야 하지만 Radix-4는 1개의 Stage로 해결되므로 줄어드는 Stage 수만큼 IFFT에 필요한 양자화의 회수는 줄어든다. 따라서 Stage 수가 줄어들면 양

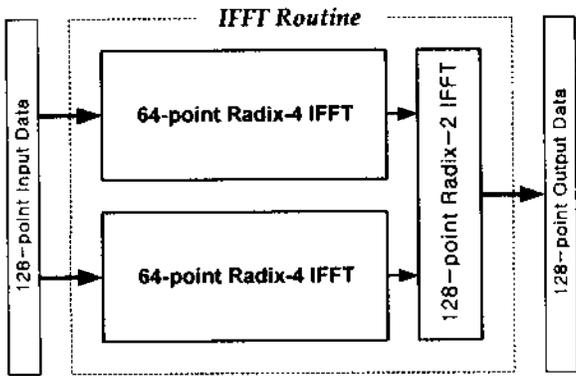


그림 3. 128-point radix-4 IFFT 구조

자화 오차도 줄어들게 된다. 그러나 Radix-4는 주소 발생 (Addressing) 이 복잡하고, 데이터의 개수가 4의 승수개여야만 하는 단점이 있다. IMDCT에서 short block은 데이터의 개수가 64이므로 Radix-4가 가능하지만 long block의 경우 128개이므로 Radix-4를 그대로 쓸 수 없다. 따라서 그림 3과 같은 구조로 Radix-4를 사용하여 Stage수를 줄일 수 있다. 이러한 방법으로 Stage수는 128-point의 경우 7개에서 5개로 줄어들게 된다. 따라서 결국 줄여준 Stage수만큼 양자화의 수는 줄어들기 때문에 그만큼 고정 소수점 오차는 줄어든다.

또한 표 1을 참조하면 DIT 구조와 DIF 구조를 서로 비교했을 때 DIF 구조가 오차가 더 큰 것으로 나타났다. 따라서 오차의 최적화에 가장 적합한 구조로써 Radix-4 DIT IFFT를 선택하였다.

V. 성능 평가

앞서 설명한 바와 같이 고정 소수점 모의 실험상의 오차를 줄이기 위하여 가장 많은 오차를 발생시키는 IMDCT 에서 최적화를 수행하였다. AC-3 오디오 알고리즘에 있어서 고정 소수점 연산의 오차는 결국 최종 복호화된 음질의 저하를 가져오므로 최적화의 성능 평가는 복호화된 오디오 신호의 음질을 기준으로 수행되어야 한다. 부동 소수점 복호화기와 최적화된 고정 소수점 복호화기의 16 bits 출력 PCM을 비교해 본 결과 그림 5-1과 같이 0.1 이내의 오차를 얻을 수 있었다.

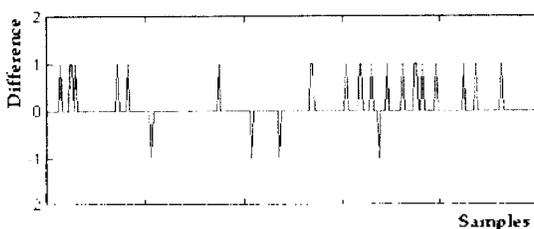


그림 4. 고정/부동 소수점 복호화기의 오차

VI. 결론

본 논문에서는 AC-3 오디오 고정 소수점 복호화 알고리즘의 오차를 최적화하였다. AC-3 오디오 복호화 알고리즘에 대한 부동/고정 소수점 모의 실험을 수행하였고, 고정 소수점 모의 실험의 내부 부분별 오차에 대해 연구, 분석하였다.

고정 소수점 연산에는 양자화 잡음이 존재하므로 고정 소수점 모의 실험에서는 이것을 최적화해 주어야 한다. 특히 양자화 잡음은 곱셈 연산을 수행하면서 더 증가하게 된다. 이를 개선하기 위하여 알고리즘의 최적화를 통해 곱셈의 수를 줄이는 방법과 Round-to-nearest-odd 방법과 같은 향상된 반올림 기법을 사용하는 방법 등을 이용하였다. 그리고 모의 실험을 통하여 각 블록별 고정 소수점 오차를 분석하였다. 그 결과 고속 IMDCT 루틴의 내부에 존재하는 IFFT 루틴에서 IMDCT 오차의 약 63%, 전체 오차의 약 55%에 달하는 오차가 발생된다는 것을 발견하였다. 그래서 모의 실험을 통해 오차가 적은 Radix-4 DIT IFFT 구조를 선택하고, 이를 128-point의 Long block에 사용 가능한 구조를 제안하였다.

본 연구에서는 고정 소수점 모의 실험의 최적화를 통해 기존 PCM인 Dolby 부동 소수점 모의 실험 프로그램의 결과와 비교해 0.1이내의 작은 오차를 갖는 16 bits의 최종 출력 PCM을 얻을 수 있었다.

참고 문헌

- [1] Advanced Television System Committee(ATSC) Standard Doc. A/52, "Digital Audio Compression Standard(AC-3)", Nov., 1994
- [2] M. Iwadare, et al. "A 128 kbit/s hi-fi audio codec based on adaptive transform coding with adaptive block size MDCT." IEEE J. Selected Areas Comm. pp. 138-144, 1992.
- [3] Craig C. Todd, et al. "AC-3: Flexible Perceptual Coding for Audio Transmission and Storage," Audio Engineering Society, 1993.
- [4] Israel Koren, "Computer Arithmetic Algorithm", Prentice Hall, 1993
- [5] 고우석, 유선국, 박성욱, 정남훈, 김준석, 이근섭, 윤내화, "AC-3와 MPEG-2 오디오 공용 복호화기의 설계", 신호처리 합동 학술대회, pp.299-302, Sep., 1997