DVB-T를 위한 멀티채널 HE-AAC 디코더의 최적화

우원희

성균관 대학교

bash34@paran.com

Optimization of Multichannel HE-AAC decoder for DVB-T

Woo, Won-Hee

Sungkyunkwan University

요약

최근 유럽에서 DVB-T HDTV 방송 표준이 정하지면서 오디오 포맷으로 HE-AAC가 채택되었다. HE-AAC는 압축효율은 높지만 연산량이 높아 낮은 성능의 DSP에서 수행하기에는 어려움이 있다. DVB-T에서는 5.1채널을 사용하고 있어 더욱더 많은 연산을 필요로 한다. 본 논문은 ISO/DEC 14496-3 MPEG4 HE(High Efficiency)-AAC의 Level4에 해당하는 Multichannel Decoder를 최적화하여 구현하고. 가장 많은 연산을 필요로 하는 Synthesis Filter Bank에 제안된 알고리즘을 적용하여 연산량을 줄였고 대부분의 연산부를 어셈블리로 코드 최적화를 하여 작은 성능의 DSP를 사용하여 실시간 Multichannel HE-AAC Audio Decoder의 구현이 가능하게 하였다. DVB-T 오디오 시스템에 필수로 필요한 Audio Description , Dynamic Range Control, Downmix 등을 함께 구현하여 실제 수신기에 사용이 가능하도록 하였다. DSP는 Samsung의 CalmRISC16 + MAC24 core 를 사용하였다.

1. 서론

최근 유럽에서 HD DVB-T의 표준화가 진행되면서 bandwidth를 줄이기 위해 최신 오디오 코덱인 HE-AAC와 DolbyDigitalPlus를 오디오 표준으로 사용하게 되었다. 여기서 사용하는 HE-AAC는 Level4에 해당한다. Level4는 방송용으로 Multichannel의 오디오를 지원하고 최대 48Khz 의 Sample Rate를 지원해야 한다.

기존의 스테레오만 사용하더라도 HE-AAC 디코더는 SBR(Spectral Band Replication)을 사용하여 고주파 영역을 합성하기 때문에 연산량이 비교적 높다. 이런 HE-AAC가 Multichannel을 사용하는 경우는 반드시 연산 량에 대한 최적화 작업이 반드시 필요하게 된다.

본 논문에서는 Multi Channel HE-AAC (ISO/IEC 14496-3) 표준을 Embedded DSP 환경에서 최적화하기 위한 설계 방법 제시하고 이를 기반으로 가장 연산이 많은 SBR(Spectral Band Replication)에 연산을 줄이는 새로운 방법을 적용한 후 어셈블리 최적화를 통하여 실시간 동작이 가능한 Multi Channel HE-AAC CODEC을 구현한다. Target DSP Platform은 Samsung의 CalmRISC16 + MAC24 core 로 Single MAC을 가지고 있는 24bit DSP이다. HE-AAC 뿐만 아니라 HD DVB-T를 위해 Audio Description,, Downmix, Audio Transcoding, LATM 등을 함께 개발하여 DVB-T 오디오 시스템을 완벽하게 구현하였다.

2. DVB-T 오디오 시스템

가. DownMix

Downmixing 과정은 멀티채널 오디오를 Stereo 출력으로 변환하는 과정이다.MPEG4 AAC를 택하고 있는 DVB-T에서는 다양한 Downmixing 방식을 사용하고 있다. 다양한 방식 중 Bit Stream의 Syntax 정보를 가지고 하나의 방식이 선택되어 처리 된다.

MPEG Style Downmix

MPEG의 방식은 5채널 이상의 오디오 소스에 대해서만 동작하도록 설계되어 있다. 방송의 규격중 5채널에 해당하는 3/2 규격이 아닌 다른 채널 규격에서는 사용되면 안 된다. 아래와 같은 수식을 이용하여 처리되는데 HE-AAC의 PCE(Program Config Element)에서 추출된 정보를 사용하야 한다.

DVB Style Downmix

DVB에서는 별도로 Downmixing 방식을 정의하고 있다. 여기서 정의하고 있는 Downmixing 관련 정보는 MPEG4 시스템에 전달되어야 한다. 전달하는 방식으로는 DSE(data stream element)를 사용한다. DSE(data stream element)로부터 얻은 Downmixing 정보는 PCE(Program Config Element)에서 얻은 정보와는 다르게 구체적으로 Mixing Level을 Center와 Surround로 분리하여 전송해주고 있다. On/Off를 알려주는 Flag와 Level을 알려주는 Index 값을 전송하게 된다. Index는 Mix_Level을 dB로 표현한다.

나. Audio Description

Audio Description 은 시각 장애자를 위해 방송의 부연 설명을 소리로 전달하여 이해를 도와주는 것이 목적이다. 일반적으로 우리나라의 경우 방송국에서 부연 설명을 방송국에서 믹싱하여 전송하지만 유럽 DVB-T에서는 수신기에서 믹싱을 한다. 따라서 Audio Description은 수신기의 성능이 높아야 한다. 메인 오디오와 서브오디오(부연설명)을 동시에 디코딩해야한다. 즉 듀얼 디코딩이 수행되면서 동시에 오디오 믹싱이 동작하여 하나의 소리로 만들어야한다. 일련의 작업들이 모두 수신기에서 수행되기 때문에 높은 수신기의 성능을 요구하게 된다. 이번 연구에서는 메인 오디오는 DSP를 사용하였고 서브오디오는 HOST CPU를 사용하여 구현하였다. 방송 시청 시에는 HOST CPU의 자원은 많이 여유가 있다.

다. DTS Transcoding

셋톱박스나 TV에는 디지털 오디오 출력으로 SPDIF 광 출력을 가지고 있다. DVB-T에서 오디오 코덱을 HE-AAC로 사용하면 광출력을 이용하여 HE-AAC를 출력하더라도 대부분의 AV 리시버에서 지원을 하지 않기 때문에 의미가 없어진다. 이런 문제를 인식하고 Coding Technology 사와 DTS사가 공동으로 HE-AAC를 DTS 포맷으로 변경하는 기능을 수행하는 DTS Transcoder를 개발하였다.

칩 제조사 입장에서는 그 정도의 오디오 성능을 가지기 위해서는 칩의 크기가 커져야 하는 문제가 있다. 최근에는 Dolby 사가 Conding Technology사를 인수함으로써 DTS보다는 DolbyDigital로 변환 혹은 인코딩하는 방식도 도입되고 있다. 본 연구에서는 DTS사 아닌 DD(Dolby Digital)를 사용하였고 DSP의 성능이 부족하여 Host CPU를 사용하였다.

라. LATM/LOAS

MPEG4 Audio(ISO/IEC 14496-3)에서는 MPEG4 System (ISO/IEC 14496-1)과 별도로 Audio Transport System을 가지고 있다. 이 System은 다양한 오디오를 전송하기 위해 고안되었고 LATM(Low-overhead MPEG-4 Audio Transport Multiplex) 과 LOAS(Low Overhead Audio Stream)로 구성된다.

DVB-T 방송 시스템에서는 사실상 MPEG2 시스템을 기본으로 사용하고 있고 여기에 내부적으로 MPEG4를 사용하고 있다. 즉 MPEG 2 PES의 Payload 부분을 추출하면 LATM/LOAS 포맷으로 되어 있어 다시 Syntax를 분석하고 ES 정보를 추출하는 과정을 거치게 된다.

2. HE-AAC 디코더의 분석

가. 소프트웨어 프로파일링

최적화 작업을 진행하기 전에 전체의 HE-AAC의 알고리즘 블록 중 가장 많은 연산을 가진 블록을 찾아내었다. PC 환경에서 Microsoft의 Visual Studio 6.0을 사용하여 각각의 알고리즘 함수들의 시간점유를 구하였다.

실험의 결과에 의하면 Synthesis Filter Bank가 전체의 24.6%를 사용하였고 이를 포함하고 있는 SBR Decoder는 68.3%를 사용하고 있다. SBR Decoder를 최적화하게 되면 전체의 성능 개선효과가 가장 크다.

3. 최적화 방법

최적화 작업은 알고리즘의 최적화와 수작업으로 진행되는 어셈블리 최적화로 진행 되었다. 알고리즘 최적화는 소프트웨어 분석을 통해 얻은 프로파일링 정보를 이용하여 알고리즘 최적화 대상이 되는 블록을 알고리즘을 변형하여 연산수를 줄이는 작업이다. 어셈블리 최적화는 DSP의 명령어를 가장 효율적으로 사용하여 원하는 기능을 구현하는 작업이다.

Modules	Multiplications	Additions
Analysis FilterBank	65536	64512
Synthesis FilterBank	262144	258048
Total	327680	322560

표 1 .Required Operation before Optimization

가. Filter Bank 알고리즘 최적화

Synthesis Filter Bank는 DCT 연산이 아니지만 이것의 형태를 DCT-II Form이 되도록 수식을 변형하고 이것을 Split Radix DCT 방식으로 구현을 하면 기존의 방식보다 적은 연산 량을 가지게 된다.

Step 1:

Synthesis Filter Bank의 알고리즘을 간략하게 표현하면 아래와 같다.

$$Output[n] = \sum_{k=0}^{k=63} x(k)\cos(\pi(k+0.5)(2n-64)/128) \quad (1)$$

Where n varies from 0 to 127

(1)의 수식의(2n-64)에서 2를 제거하면

$$Output[n] = \sum_{k=0}^{k=63} x(k)\cos(\pi(k+0.5)(n-32)/64) \quad (2)$$

Where n varies from 0 to 127

(2)의 수식은 DCT form과는 두 가지 측면에서 다르다.

인덱스 n 의 범위가 0부터 64가 되어야 하지만 0부터 127의 범위로 다르고 Cos 항에 (n-32)의 delay가 존재한다.

Step 2:

위의 수식 (2)에서 n에 (64-n)을 대입한다.

$$Output[64-n] = \sum_{k=0}^{k=63} x(k)\cos(\pi(k+0.5)(-n+32)/64) \quad (3)$$

Cos의 성질을 이용하면

$$Output[64-n] = \sum_{k=0}^{k=63} x(k)\cos(\pi(k+0.5)(n-32)/64) \quad (4)$$

즉 (4)와 (2)는 같게 된다. 따라서

$$Output[n] = Output[64-n] \quad \text{for } 0 \le n \le 64$$

이와 같이 128개의 샘플 중 64~127에 해당하는 결과는 앞의 64개의 결과와 같기 때문에 다시 연산을 할 필요가 없다.

Step 3:

cos에 delay를 처리하기 위해 index 32부터 63까지를 하는 수식과 64부터 95까지 연산하는 수식에 (n-32)를 m으로 치환하면 아래와 같다.

$$Output[m+32] = \sum_{k=0}^{k=63} x(k)\cos(\pi(k+0.5)(m)/64) \quad (5)$$

Where m varies from 0 to 31

$$Output[m+32] = \sum_{k=0}^{k=63} x(k)\cos(\pi(k+0.5)(m)/64) \quad (6)$$

Where m varies from 32 to 63

(5)와 (6)의 식의 병합하고 m을 n으로 다시 복원하면,

$$Output[n+32] = \sum_{k=0}^{k=63} x(k)\cos(\pi(k+0.5)(n)/64) \quad (7)$$

Where n varies from 0 to 63

(7)의 수식은 64 point DCT II의 형식을 가진다. 여기에 Split Radix 알고리즘을 이용하여 연산을 하게 되면 많은 연산 량을 줄일 수 있게 된다. Analysis Filter Bank도 같은 방법을 적용하면 DCT-III 형식으

로 연산을 줄일 수 있다. 아래의 표는 Frame을 처리하는데 필요한 연산 량을 보여 주고 있다.

Modules	Multiplications	Additions
Analysis FilterBank	2304	5440
Synthesis FilterBank	14336	23616
Total	16640	29056

표 2 Required Operation after Optimization

4. 성능평가

본 논문에서는 세 가지로 분리하여 성능을 평가하겠다. 처음으로 알고리즘 최적화에 대한 평가와 레퍼런스 디코더의 결과물과 비교하는 음질 평가를 하고 마지막으로 난이도가 높은 Stream을 이용하여 소모 MCPS(Million Cycle Per Second)를 평가한다.

가. 알고리즘 최적화 평가

Filter bank의 연산 량을 줄이기 위해 수식은 정형화된 DCT로 표현하고 이것을 Split Radix DCT를 이용하여 구현하였다. Visual Studio 6.0에서 Floating Point Code를 이용하여 최적화 전후의 변화를 측정하였다. Test Vector,"TC022.mp4"에서 전체 디코딩 시간이 52% 감소되는 효과를 확인하였다.

File name	TC022.mp4
Total decoding time (ms)	56709.887
SBR Decoder	50202.437
Synthesis FB.	34996.583
Analysis FB.	7775.04
Envelop Calc.	6336.602
HF generation	781.239

표 3 decoding time before optimization

File name	TC022.mp4
Total decoding time (ms)	27167.761 (-52%)
SBR Decoder	21463.943
Synthesis FB.	11193.514
Analysis FB.	3748.177
Envelop Calc.	5494.406
HF generation	723.368

표 4 decoding time after optimization

나. 음질 평가

음질 평가는 24Bit Fixed Point C code 과정에서 수행되어야 한다. PSNR의 값을 이용하여 레퍼런스 코드 대비 어느 정도 일치하는가를 판단하게 된다. PSNR의 수치가 비교적 낮으면 구현상에 문제가 있는 것으로 판단하고 Code를 다시 수정하여 PSNR을 높이는 과정을 수행한다.

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) = 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right)$$

여기서 MAXi는 24bit Data로 표현되는 가장 큰 수를 의미한다. MSE는 매 Sample의 차이를 Mean Square Error로 표현한 값이다.

PSNR은 전체적인 추이를 보는 것이고 순간적으로 발생하는 Error는 검출이 안 될 가능성이 높다. 이런 이유로 매 샘플의 최대 차이값(Max Difference)도 성능평가의 요소로 사용하게 된다.

본 논문에서는 ISO의 AAC Conformance Test 파일과 Nero사의 HE-AAC Encoder을 사용하여 만든 몇몇의 Multichannel HE-AAC 파일을 사용한다. 이 모든 Test Vector를 디코딩하는데 문제가 없어야 한다.Test Vector의 수가 많기 때문에 자동화된 평가 방법이 필요하다. PSNR을 구하는 Tool을 구현하고 이를 스크립트 언어를 사용하여 Batch Test가 가능하도록 환경을 구축하였다. 모든 Vector에 대해 레퍼런스 디코더의 출력 파일과 최적화된 Decoder의 출력 파일의 PSNR을 측정하였다.

Test Vector	-ISO conformance streams -Encoded Streams by Nero Tools
Average PSNR	122.37 dB

표 5 .Average PSNR

다. MCPS 평가 .

MCPS의 평가는 L3 어셈블리 코드 구현 단계에서 진행되어야 한다. 구현된 Code의 성능을 최적화하기 위해 각각의 알고리즘 별 MCPS도 필요하다. 여기서 말하는 MCPS는 Cache Hit Ratio 나 Bus Latency 간은 요소들은 고려하지 않은 MCPS이다. HE-AAC가 구현된 후 Simulator를 이용하여 측정이 가능하다. 사용된 Simulator는 CalmShine16 Ver1.62를 사용하였고 Worst Stream으로 사용된 파일은 ISO의 "al_sbr_cm_48_5.1.mp4"이다.

파일의 각각의 Frame중 Peak로 확인된 MCPS는 아래와 같다.

4. 결론

본 논문에서는 Multi Channel HE-AAC Decoder를 구현하기 위해 개발 방법론을 제안하고 그 방법론을 따라 CalmRISC16/MAC24 DSP에 구현을 완료하였다. 최적화 방법으로는 알고리즘 적으로 Synthesis Filter Bank의 연산 량을 줄이고 어셈블리로 구현하여 107 MCPS를 소모하는 방송용 Multichannel HE-AAC를 구현하였다. 이와 함께 DVB-T를 위한 오디오 시스템들도 함께 구현하였고 결과물은 유럽향 DTV 제품에 적용되었다.

참고문헌

[1] Gerlind Plonka and Manfred Tasche, "Spli radix algorithms for discrete trigonometric transforms", 2002

[2] ISO/IEC 14496-3, "Coding of audio-visual objects Part3: Audio",2005

[3] ETSI TS 101 154 V1.7.1, "Digital Video Broadcsting, Implementation guidelines for the use of MPEG2 system" , 2005

[4] Hyung-Jung Kim, Deock-Gu Jee, "Optimization of HE-AAC for Korean S-DMB Using TMS320C55x DSP core", 2006

[5] Samsung, CalmRISC16/MAC24 Instruction Set Reference Guide Book.,