

# OakDSPCore®를 이용한 MPEG-I Layer 3 decoder 의 DSP 실시간 구현

하호진, 강상원  
한양대학교 제어계측공학과

## DSP real-time implementation of the MPEG-I Layer 3 decoder using OakDSPCore®

Jin-Ho Ha, Sang-Won Kang  
Dept. of Control & Instrumentation Eng., Hanyang Univ.  
e-mail) swkang@selab.hanyang.ac.kr

### 요약

본 논문에서는 국제 표준화기구(ISO) 산하의 동영상 전문가 그룹(MPEG)의 오디오 압축방법들중 하나인 MPEG-I layer 3의 복호화기를 고정 소수점으로 변환한 후, OakDSPCore®를 기반으로 전 과정을 어셈블리어로 실시간 구현하였다.

실시간 구현에 사용된 OakDSPCore®는 DSP Group사에서 개발된 저전력 소비형 16-비트 고정 소수점 DSPCore로서 40MIPS의 성능을 가지고 있으며, 음성/오디오, 통신, 디지털 셀룰라폰 같은 소비자의 맞게 ASIC 화할 수 있는 장점을 가지고 있다.

구현된 MP3 복호화기는 약 33 MIPS의 복잡도를 나타내며, 사용된 메모리양은 프로그램 ROM 3.1K words, 데이터 ROM(table) 10.82K words 및 RAM 6.1K words이다. 구현된 MP3 복호화기는 OMNI-MEDIASOUND에서 제공하는 4개의 test 벡터들을 bit-exact하게 통과하였다.

### 1. 서론

디지털 오디오 방송, 위성 TV 및 멀티미디어

의 다양한 응용 분야에서, CD 수준의 오디오 신호를 낮은 비트율에서 전송 및 저장하기 위해서, 오디오 신호의 압축은 필수적이다. 상용화된 디지털 오디오 압축 방식 중 현재까지 가장 좋은 음질과 가장 낮은 전송률을 가지고 있는 것으로 알려진 MPEG layer 3 (MP3) 방식은 CD 수준의 오디오 신호를 12:1 이상의 압축으로 복원음을 재생할 수 있는 고품질 오디오 부호화 기술이다[1]. 이러한 오디오 신호 압축의 핵심은 인간의 청각계 지각 특성을 이용하여 잡음을 은폐시키는 심리음향 모델과 오디오 신호의 중복성을 이용하는 방법이라고 할 수 있다.

MPEG-1,2 계층 1,2 오디오 부호화기의 경우, 32개의 일정한 대역으로 분할된 서브밴드 필터를 사용함으로써 오디오 신호를 서브밴드 대역으로 나누게 되고, 각 서브밴드 샘플에 심리음향모델을 이용하여 비트를 할당한다[2]. 이러한 임계대역과 불일치하는 서브밴드에 심리음향모델의 비트를 할당함으로써 비트의 손실을 가져오며, 낮은 비트율로 갈수록 많은 음의 왜곡을 가져오게 된다. 이에 비해서 MP3는 서브밴드 샘플에 MDCT(Modified Discrete Cosine Transform)를 수행함으로써 주파수 대역의 해상도를 증가시키는 하이브리드 변환 부호화 방식

을 사용한다. 이와 동시에 FFT를 사용하는 심리음향모델은 마스킹 임계값을 사용하여 인간 청각계의 특성에 의한 지각적인 중복성을 제거한다. 마스킹 임계값을 근거로 주관적인 잡음이 들리지 않도록 MDCT 스펙트럼에 대해서 비트할당, 양자화, 허프만 부호화 과정의 반복 루프를 통해서 양자화가 이루어지게 된다. 결과적으로 잡음이 최소화된 양자화 비트는 허프만 부호화 과정을 거쳐서 부호화되어 부가 정보와 함께 비트열로 만들어 지게 된다.

본 논문에서 사용한 CSD17C00 칩은 DSP 그룹사에서 개발된 저전력 소비형 16-비트 고정소수점 DSPCore인 OakDSPCore®를 기반으로 Miscellaneous Logic, Serial Port, Compander, Host 접속, Timer의 5가지 Peripheral과 범용 I/O Ports로 설계되었고 40MIPS의 성능을 가지고 있다.

본 논문의 구성은 2장에서 MP3 복호화기의 기본구조와 특성에 대하여 살펴보고 3장에서는 실시간 구현에 관해 논하며 마지막으로 4장에서 결론을 맺도록 하겠다.

## 2. MP3 복호화기

MP3 복호화기는 3부분으로 크게 나눌 수 있다. 프레임 분해, 역 양자화, 역 매핑블록이다. 개략적인 블록도는 다음과 같다.

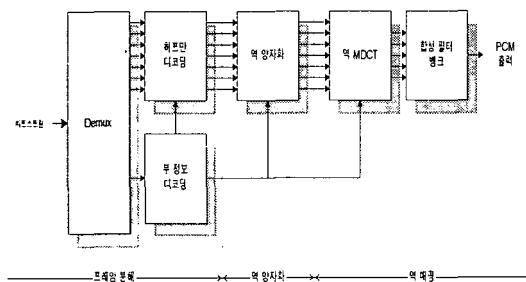


그림 1. MPEG-1 계층 3의 기본 구조

### 2.1 프레임 분해

입력 비트 스트림에서 헤더정보와 이후 블록의 디코딩을 위한 부 정보 그리고 양자화된 스펙트럼 값으로 변환하기 위한 scalefactor 값을 추

출하는 과정이 수행된다. 헤더정보에는 인코딩된 비트 스트림의 layer, 샘플링 주파수 그리고 스테레오 모드등의 정보들이 들어있다. 스테레오 모드에서는 스테레오 신호의 중복적인 성질을 이용해서 코딩이 이루어진다. 이러한 스테레오 모드는 인간의 청각적인 특성을 이용해서 효과적인 코딩을 수행한다. 부 정보에서는 scalefactor 디코딩, 허프만 디코딩 그리고 역 양자화를 하기위한 정보들이 들어있다. 그림 2는 부 정보에 대한 비트 스트림의 구조를 나타내었다. 부 정보는 그래놀 2개 채널 2개, 즉 4개가 존재하게 된다. Main\_data\_begin에는 9비트가 할당되며, 이 값은 메인 데이터의 첫번째 비트의 위치를 결정하는데 이용되며, 오디오 비트 스트림의 상크 워드의 첫번째 바이트로부터 바이트 단위로 음수의 오프셋으로서의 위치를 정한다. 하지만 헤더와 부 정보에 속해있는 바이트 수는 계산에 들어가지 않는다. 따라서 하나의 AAU(Audio Access Unit)를 코딩하는데 비트를 유동적으로 사용함으로써, 효과적인 비트할당을 하게 된다. Scalefactor 디코딩은 역 양자화를 위한 파라미터로서, 각각의 그래놀 채널마다 값을 가지게 된다. Scalefactor\_compress에 의해서 slen1과 slen2에 할당된 비트에 따라 scalefactor 값을 읽어 들인다. 그리고 slen1과 slen2는 블록의 형태에 따라 적용 범위가 달라지게 된다. 다음 테이블은 블록의 형태에 따른 slen1과 slen2의 적용범위를 나타내고 있다. scfsi(Scalefactor select info.)는 한 프레임 내의 2개의 그래놀에 대한 scalefactor를 모두 전송하는지에 대한 정보를 제공한다. 0이면 각각의 그래놀에 대해서 scalefactor가 전송되고, 1이면 그래놀 0에서 전송되었던 scalefactor가 그래놀 1에서도 사용된다. 허프만 디코딩은 각 그래놀의 스펙트럼 값의 범위에 따라 다른 허프만 코드 테이블을 가지고 디코딩을 하게 된다.

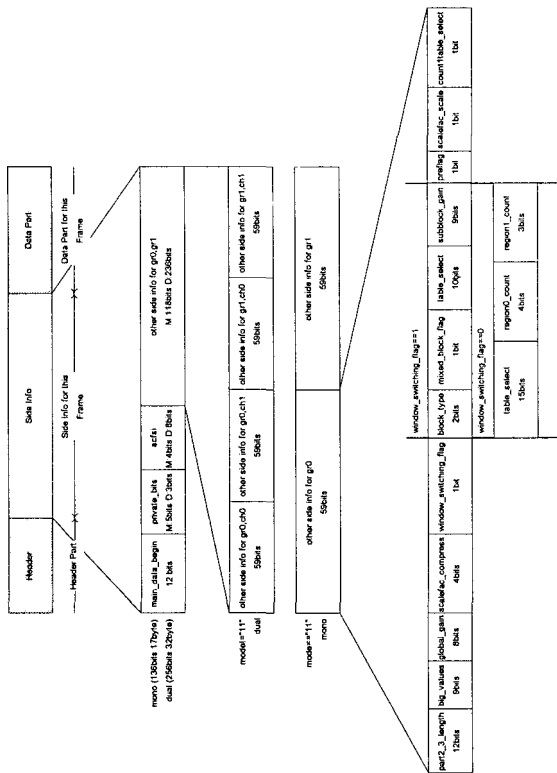


그림 2. 부 정보 블록도

표 1. 블록형태에 따른 slen1 과 slen2 의 적용범위

블록 형태	블록 형태 (mixed)	Slen1	Slen2
0,1,3	Any	0-10 long 블록	11-21 long 블록
2	0	0-5 short 블록	6-11 short 블록
	1	0-7 long 블록 3-5 short 블록	6-11 short 블록

0에서 Nyquist 주파수까지의 전체 주파수 범위는 여러 지역으로 나뉘어지며, 각 지역마다 다른 허프만 테이블을 사용하여 디코딩을 하게 된다. 이렇게 지역으로 나누는 작업은 양자화된 값의 최대값에 따라 결정된다. 이것은 MDCT에 의해서 고주파 값들은 작거나 거의 0의 값을 가지므로 코딩할 필요가 없다는 가정을 가지고 수행된다. 높은 주파수에서 시작하여, 0으로 양자화된 값을 2의 배수로 세고 이것을 rzero라고 정한다. 그 다음부터 1을 넘지 않는 절대값을 가진 값의 범위를 count1 지역으로 정해지고, 이 범위는 4의 배수로서 결정된다. 나머지 부분을 2의 배수로 세어지고 이

것은 big value 지역으로 정한다. 이러한 과정은 그림 3에 나타나 있다. Big value 지역은 허프만 디코딩의 성능을 향상시키기 위해서 short window의 경우에는 2부분 그리고 long window의 경우에 3부분으로 나누어서, 각각 다른 허프만 테이블을 사용하여 디코딩을 한다. 역 양자화를 위한 부 정보로 전체 이득값이 사용되며, short window의 경우에는 각각의 전체 이득값이외에 3개의 서브블럭 이득값이 사용된다.

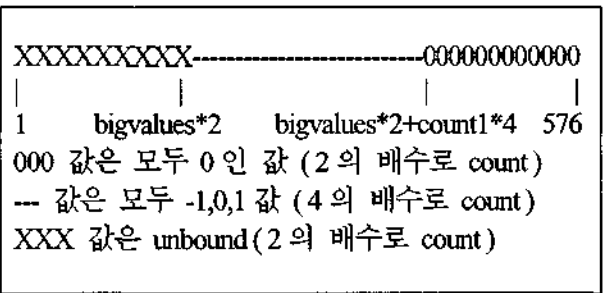


그림 3. 그레놀에 대한 partition

### 2.2 역 양자화

허프만 디코딩된 결과값을 역 매핑을 하기 위한 값으로 변환하는 과정이다. 역 양자화 과정은 다음 식 (1)과 (2)에 의해 각각 계산된다. Short 블록인 경우는 수식(1)에 의해 계산되고, short 블록을 제외한 나머지 블록은 수식(2)에 의해서 계산된다.

$$x_i = \text{sign}(is_i) * |is_i|^{4/3} * 2^{(\text{sum}0 - \text{sum}1)} \quad (1)$$

$$\text{sum}0 = (\text{global\_gain} - 210 - 8 * \text{subblock\_gain}) / 4$$

$$\text{sum}1 = \text{scalefac\_multiplier} * \text{scalefac\_s}$$

$$x_i = \text{sign}(is_i) * |is_i|^{4/3} * 2^{(\text{sum}2 - \text{sum}3)} \quad (2)$$

$$\text{sum}2 = (\text{global\_gain} - 210) / 4$$

$$\text{sum}3 = \text{calefac\_multiplier} * (\text{scalefac\_l} + \text{preflag} * \text{pretab})$$

여기서 is는 허프만 디코딩 결과이고, pretab은 고주파 성분의 값에 대한 증폭을 위한 값이며, pretab을 사용할 것인지에 대한 결정은 preflag에 의해서 결정된다.

### 2.3 역 매핑

역양자화를 거쳐 얻어진 주파수 도메인 값을 시간 도메인으로 변환하기 위해서 IMDCT(Inverse MDCT)와 합성 필터링하는 과정이다. 하지만 long window 를 사용한 경우, 인코딩 과정시 MDCT 과정에서 aliasing 을 막기 위한 aliasing butterfly 을 보상하기 위해서, 디코딩시에는 aliasing butterfly 역과정을 수행한다. 이 과정은 그림 4 과 같이 수행된다.

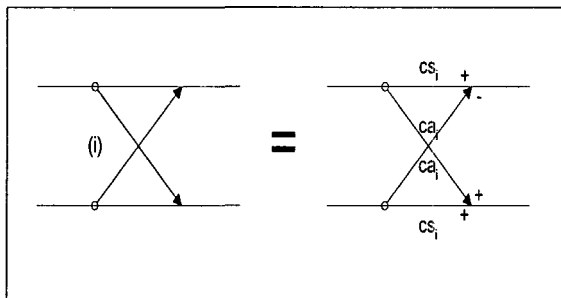


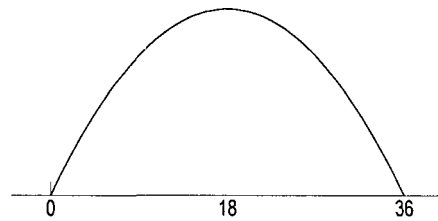
그림 4. Aliasing butterfly 과정

그리고 short window 의 경우 각 스펙트럼 값들이 scalefactor 밴드순으로 나열되어 있으며, 각 scalefactor 밴드내에는 3개의 window 에 대한 스펙트럼 값들이 순서대로 나열되어 있다. 이러한 값들은 IMDCT 를 수행하는데 적절하지 않은 순서이므로, 이 값들을 재 배치하게 된다. 이러한 과정은 short window 의 경우에만 수행된다.

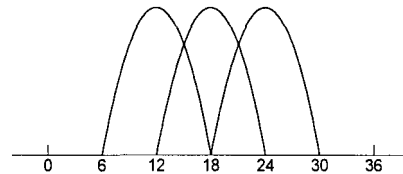
위의 과정이 끝난 후, IMDCT 는 하나의 그래늘마다 18 개의 값을 읽어와서 IMDCT 를 수행한 후, 그 이전 프레임의 값과 overlap-add 를 사용하여 디코딩을 하게 된다. IMDCT 에 사용되는 식은 다음과 같다.

$$x_i = \sum_{k=0}^{n-1} X_k \cos\left(\frac{\pi}{2n}\left(2i+1+\frac{n}{2}\right)(2k+1)\right) \text{ for } i=0, \dots, n-1 \quad (3)$$

여기서 long window 와 short window 의 경우, n 값은 각각 36 과 12 를 사용한다. 사용되는 윈도우의 형태인 long window 와 short window 의 형태를 그림 5 에 나타내었다.



(a) Long window



(b) Short window

그림 5. Window 형태

합성 필터 과정은 IMDCT 를 수행하고 난 후 각각의 그래늘 단위로 수행된다. 서브밴드 당 32 개의 값을 입력 받아 버퍼에 입력하고, 그 이전의 64 개의 값을 버린다. 합성 필터는 합성행렬을 이용해서 처리된다. 합성행렬은 다음식을 이용해서 구해진다.

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

for  $i = 0$  to 63, and  $k = 0$  to 31

합성 필터링 후, 합성 윈도우를 이용하여 32 개의 결과값을 얻는다.

### 3. 실시간 구현

MP3 의 실시간 구현에 사용된 보드는 OakDSPCore®를 기반으로 설계된 C&S Technology 사의 CSD17C00 칩이 탑재되어 있고 PC 와 RISC 기타 MCU 와도 Host 접속이 용이하며 독립 보드로도 사용 가능하다. 또한 CSD17C00 칩에는 4K words 의 데이터 RAM 과 9K words 의 데이터 ROM 이 내장되어있고 software 볼륨 조절등의 기능이 내장되어 있다.

DSP 프로그램의 개발은 준비 단계, 코딩 그리고 검증 단계의 3가지 부분으로 나누어서 수행하였다. 우선 준비 단계에서는 알고리즘을 분석하고 부동 소수점 C소스를 고정 소수점 DSPCore 에서 작동되도록 부동소수점 C소스로

변환하는 작업을 수행하였다. 그리고 MP3 디코더에 필요한 모듈을 파악하고 call tree 를 작성하였다. 효율적인 구현을 위해서는 알고리즘의 분석이 반드시 선행 되어야 한다. 계산량을 줄이기 위해서는 dual operation 이 필수적인데 이 부분에서 OakDSPCore®는 2개의 오퍼랜드가 각각 X-메모리와 Y-메모리상에 있어야 하는 제약이 있었다. 따라서 모든 메모리를 조사하여 OakDSPCore®의 성격과 알고리즘상의 메모리의 특성을 분석하여 전체적인 메모리의 위치를 배치하였다. 준비 단계에서는 프로그램의 형식, 각 모듈의 depth 에 따른 메모리의 위치, 지역 변수와 전역 변수의 지정 그리고 함수 호출 시 인자의 전달방법을 미리 프로그래머 상호간에 규약을 정해서 프로그램을 개발하였다. MP3 복호화기의 코딩은 프레임 분해, 역 양자화 그리고 스테레오 처리, 역 매핑의 3가지로 구분 지을 수 있다. 프로그램을 코딩할 때에는 어느 정도 관련이 있는 모듈들로 나누어서 하는 것이 효과적이므로 알고리즘 분석을 통해서 프로그램의 partition 을 나누었다. 본 논문에서는 프로그램의 전 과정이 어셈블리어[3][4]로 이루어 졌으며 프로그램된 각각의 모듈은 test 벡터들을 이용한 검증을 통해 확정되었다. 프레임 분해모듈은 그 내부의 모듈이 서로 유기적으로 연결되어있기 때문에 각 모듈을 먼저 프로그램한 후, test 벡터를 통하여 검증하였다. 나머지 모듈은 각각을 프로그램한 후, 계속 연결시키면서 test 벡터를 검증하였다. Test 벡터는 OMIN-MEDIASOUND사에서 제공하는 4가지 벡터값을 사용하였으며, 구현된 프로그램은 test 벡터와 bit-exact 하게 일치 하였다. 이러한 과정을 통해 구현된 MP3 복호화기의 평균 MIPS 를 표 3에 나타내었다. 사용된 메모리량은 프로그램 ROM 3.1K words, 데이터 ROM(table) 10.82K words, RAM 6.1K words 이다.

표 3. 구현된 MP3 복호화기의 각 모듈별 MIPS

모듈	MIPS
프레임 분해	6.9
역 양자화	4.1
스테레오처리	7.1
역 매핑	15.2
합계	33.3

#### 4. 결 론

본 논문에서는 16-비트 고정 소수점 DSPCore 인 OakDSPCore®를 기반으로 설계된 C&S Technology사의 CSD17C00 Chip 을 이용하여 전 과정을 어셈블리어로 실시간 구현하였다. 본 연구를 통해 MP3 오디오 복호화기가 실시간 동작되도록 구현되었으며, test 벡터의 검증을 통하여 C 시뮬레이션 결과와 bit-exact 함을 확인 하였다.

구현된 MP3 복호화기는 약 33.3 MIPS 의 복잡도를 나타내고 있으며 사용된 메모리량은 프로그램 ROM 3.1K words, 데이터 ROM(table) 10.82K words, RAM 6.1K words 이다. 본 논문에서 구현된 MP3 복호화기는 OakDSPCore®의 저 전력 소비, ASIC 구현의 용이함, 편리한 디버깅 환경과 더불어 휴대형 CD 재생기, 카세트 재생기를 대체할 수 있는 차세대 휴대형 디지털 오디오 기기등 다양한 응용에 부합하는 경쟁력 있는 시스템을 구축할 것이라 사료된다.

#### [ 참 고 문 헌 ]

- [1] ISO/IEC JTC1/SC29/WG11 No.71, "Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5Mbits -CD 11172-3(part3, MPEG-audio)."
- [2] D. Pan, "A tutorial on mpeg/audio compression," IEEE Trans. on Multimedia, Vol.2, No.2, pp.64-74, 1995.
- [3] SSP1820 OakDSPCore® Architecture Specification

Samsung Electronics Co. Ltd, Revision 3.0, March  
12, 1998.

[4] DSP Group Assemblers & Linker User's manual  
DSP Group Inc., 3120 Scott Blvd, Santa Clara, CA  
95054, Revision 7.2, March 1998.