

# MP3 엔코더/디코더 설계

김태훈, 임재영, 박주성

부산대학교 전자공학과

## The Design of MP3 Encoder/Decoder

Tae-hoon Kim, Jae-Young Lim, Ju-sung Park

Dept. of Electronics, Pusan national University

kthn@voiso.com

### 요 약

MP3는 현재 디지털 오디오 압축 표준으로 널리 사용되고 있으며 12:1의 높은 압축률을 가진다. MP3 encoding의 경우 현재는 대부분 PC를 이용하고 있으므로 MP3 encoding과 decoding을 동시에 할 수 있는 칩이 나오면 이러한 불편함 없이 portable이 가능해지며 라디오, CD, 카세트 테이프 등으로부터 고품질 녹음 후 재생이 가능해진다. TMS320C30과 호환되는 DSP 코어를 이용하였으며 MP3 엔코딩과 디코딩 실시간 수행을 위하여 알고리즘 개선과 FFT block, 주변 interface block을 설계하였다. 최종적으로 MP3 encoder와 decoder 칩 설계하고 이를 하드웨어 에뮬레이션을 이용하여 검증하였다.

### 제 1 장 서 론

기존의 반도체 시장은 대개 MP3 디코더를 위한 칩 개발에 주력하고 있었다. 그 원인으로는 반도체 휴대용 오디오를 목표로 했기 때문에 비교적 구현이 간단한 디코더에 관심을 가졌던 것으로 생각된다. 하지만 최근에 개인이 보유하고 있는 테이프나 CD 음악을 디지털화하는 요구가 증가하고 있다. 오디오 신호를 단순히 디지털화시키면 많은 반도체 메모리를 필요로 하기 때문에 음질의 손상 없이 데이터를 압축시킬 수 있는 오디오 엔코더가 필요하다.

디지털 오디오 엔코더 개발을 위하여 부동소수점 처리를 기본으로 한 CISC(Complex Instruction Set Computer) 타입의 DSP를 이용하여 MP3 엔코더/디코더 알고리즘 처리를 고속으로 수행할 수 있도록 하였으며, 또한 DSP 내부에 특수 기능블럭을 추가하였다.

제 1장의 서론에 이어 제 2장에서는 부동소수점 DSP에 대하여 설명하고, 제 3장에서는 MP3를 설명한다. 제 4 장에서는 칩 구현을 위한 계산량 감소 방법을, 제 5장에서는 하드웨어 에뮬레이션 및 검증에 대하여 설명하고, 마지막으로 제 6장에서 결론을 나타내었다.

### 제 2 장 부동소수점 DSP

32 비트 부동소수점 DSP인 TMS320C30 호환 코어를 이용하여 디지털 오디오 엔코더/디코더 칩을 개발하였으며, DSP는 4-level Pipeline 구조와 40bit floating point, 32 bit integer point multiply operation을 수행한다.

### 제 3 장 MP3

MP3 엔코더의 전체 블록도를 그림 1에서 나타내었고, 그림 3에서는 MP3 디코더 전체 블록도를 나타내었다.

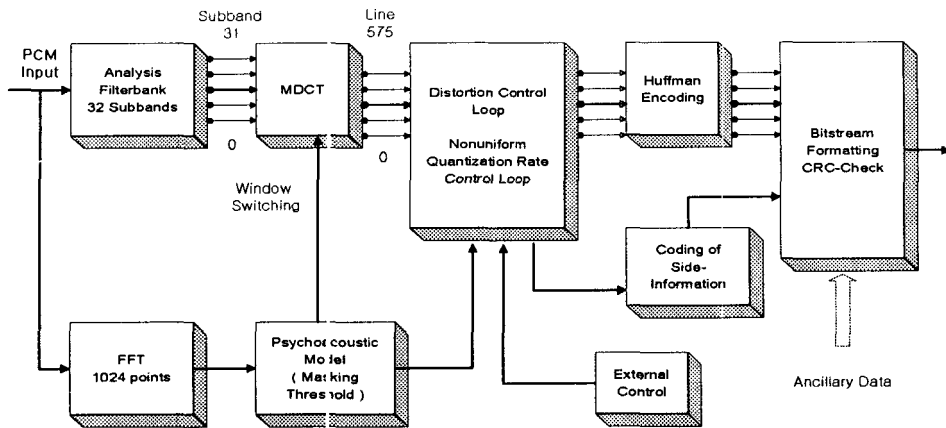


그림 1. MP3 인코더 블록도

1. 분석 필터뱅크

주파수 영역에서 균일한 대역폭을 가지는 32개의 filter band를 이용하여 입력을 32개의 서브밴드 대역의 신호로 바꾼다.

2. MDCT

각 서브밴드 신호에 대해 MDCT를 수행한다. 이는 주파수 resolution을 높여 압축률을 높이기 위함이다. 심리음향모델에서 나오는 window switching 신호에 따라 window의 크기가 결정된다.

3. 심리음향모델

일반적으로 MP3에서는 심리음향모델 II를 사용하며 심리음향모델에서 얻어지는 것은 모든 스케일팩터밴드(scalefactor band)에서의 SMR(signal-to-mask ratio)이다.

4. Distortion Control loop

scalefactor와 global\_gain이 주된 변수로써 distortion 제어 역할을 하며 iteration loop에서 증감되는 변수이다. global\_gain은 전체 샘플크기를 조정하여 전체 사용 비트를 조절하는 역할하고 scalefactor는 각 scalefactor band내의 샘플 크기를 조정하여 각 band에서의 distortion을 제어한다.

(1) Iteration loop

그림 2에서 iteration loop의 전체 흐름 도를 나타내고 있다. outer loop내에서의 반복 loop를 돌면서 신호의 왜곡(distortion)이 제어된다. iteration loop는 크게 outer loop와 inner loop로 구성된다. Outer loop는 scalefactor band 단위에서 각 band내의 신호 왜곡을 제어하며, inner loop는 global\_gain을 이용하여 전체 사용 비트수를 제어한다.

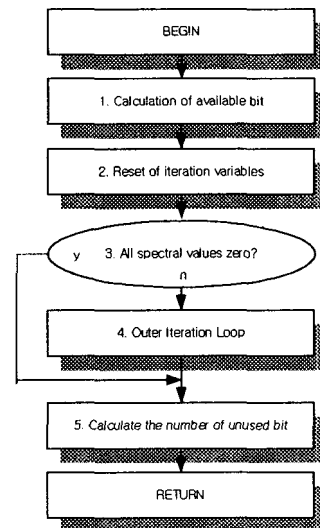


그림 2. Iteration Loop

제 4 장 칩 구현을 위한 계산량 감소 방법

제 1 절 MP3 Encoding 계산량

주요 함수의 계산량은 실시간 수행시 약 87 MIPS 이상을 요구하나 DSP의 성능을 고려할 때 약 50 MIPS로 줄일 필요성이 있다. 심리음향 모델에서는 FFT 블록을 설계하여 계산량을 줄였으며 iteration loop에서는 loop 횟수를 줄이는 precalculating 방법을 이용하여 줄일 수 있었다.

제 2 절 Precalculation 방법

Iteration loop에서 scalefactor가 1 증가하면 global\_gain은 2가 증가하는 것과 같은 효과가 있다. Iteration loop의 최종 목표는 사용 가능한 비트수 내에서 distortion을 최대로 masking하는 것이다. 이는 초기에 masking 모양과 비슷한 distortion 할당을 하는 방

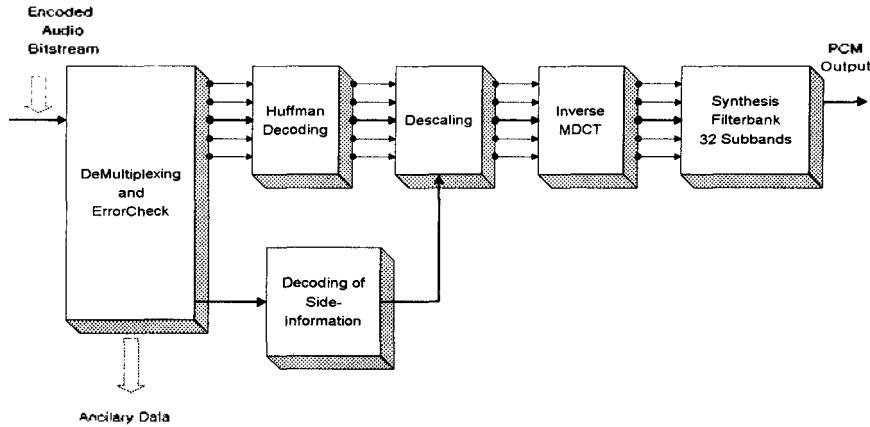


그림 3. MP3 디코더 블록도

범으로 loop 횟수를 줄일 수 있다. precalculating 방법은 아래와 같다.

1. Global\_gain을 고정(이전 프레임에서 구한 값에서 8을 증가시킴)
2. Scalefactor를 증가시키면서 distortion 계산
3. 모든 scalefactor band에 대해서 distortion이 masking 되는 최소 scalefactor 각각 계산
4. 계산된 Global\_gain과 scalefactor가 iteration loop이 초기값으로 사용
5. Iteration loop 과정은 그림 4와 같이 변형된다.

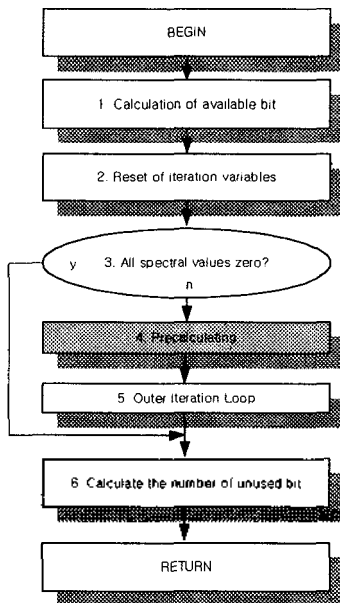


그림 4. Iteration loop의 변형

### 제 3 절 특수기능블록

설계된 FFT 블록은 MP3 encoder에서 사용되는 Hann Window 부분과 FFT 처리 부분을 포함하여 설

계하였다. FFT 블록의 입력 값 중에서 Long/Short 비트에 의해서 Long/Short 비트 값이 1인 경우에는 총 1024개의 데이터를 처리하고, Long/Short 비트 값이 0인 경우에는 256개의 데이터 블록 세 개를 동시에 처리한다. 설계된 FFT 블록은 외부의 호스트 프로세서 (Host Processor)와는 독립적으로 동작하며 처리된 연산 결과를 RAM 블록에 저장함으로써 호스트 프로세서와 데이터를 전달한다.

먼저 Hann window 부분과 FFT 처리 부분을 각각 설계한 후, 통합하여 설계하여 C로 모델링한 결과와 비교하여 그 동작을 검증하였다.

### 제 5 장 하드웨어 에뮬레이션 및 검증

MP3 엔코더/디코더는 C 레벨에서 알고리즘을 검증한 후 이를 어셈블리어로 작성하여서 C와 수행 결과를 비교하였다. 또한 작성된 어셈블리어와 FFT 블록, 외부 CODEC interface는 하드웨어 에뮬레이션을 통하여 검증할 수 있었다. 특수 블록 추가와 알고리즘 개선을 통하여 실시간 수행을 위하여 50 MIPS 정도가 요구되며, 전체 DSP 코어는 13만 gates 정도이며 RAM은 70 Kbytes, program ROM 46K words 정도가 필요하다. 하드웨어 에뮬레이션을 수행하기 위한 전체 시스템은 그림 5와 같다. 하드웨어 에뮬레이션 동작을 확인하기 위하여 하드웨어 에뮬레이터, 데이터 입출력을 위한 PC 시스템, 인터페이스 보드로 검증용 시스템을 구성하였다. MP3 엔코더 동작을 수행하기 위하여 프로그램을 ROM에 저장하고, PC 시스템에서 오디오 데이터를 입력하면, 하드웨어 에뮬레이터에서 오디오 데이터를 엔코딩하고, 이를 다시 PC로 전송하였다. PC 상에서 엔코딩된 데이터를 WINAMP라는 어플리케이션 프로그램을 통하여 동작을 검증하였다. 또한 엔코딩된 데이

터를 MP3 디코더를 통하여 복호화한 후 청취한 결과 고음질을 유지할 수 있음을 확인하였다.

## 제 6 장 결 론

본 논문에서 디지털 신호처리의 응용분야에서 많이 사용되고 있는 TMS320C30 칩과 호환성을 가지면서 상용의 제품보다 고속 동작하는 DSP 코어를 이용하고 특수 기능블럭을 첨가하고 이를 이용하여 MP3 인코더/디코더를 구현하였다. 먼저 MP3 인코더/디코더 알고리즘을 C 언어를 이용하여 알고리즘을 구현한 후 어셈블리어를 이용하여 빠른 수행이 가능하도록 최적화시켰다. 응용 알고리즘의 결과를 IKOS Emulator를 이용하여 구현하고 실행을 시켜 PC 인터페이스를 통하여 인코딩/디코딩을 수행시킨 후 스피커를 통해 청음 테스트를 수행하여 회로와 알고리즘이 정상적으로 동작함을 확인하였다.

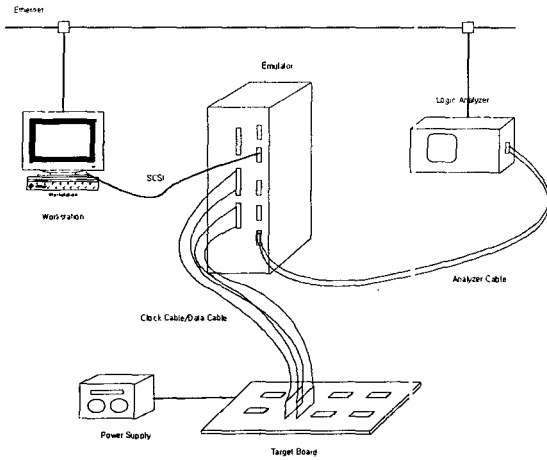


그림 5. Hardware Emulation 시스템

그림 6은 오디오 원음을 표현한 것이며 그림 7은 그림 4.16의 원음을 설계된 DSP를 이용하여 MP3로 인코딩한 후에 PC 상에서 플레이되는 데이터를 캡처한 결과를 나타낸 것이다. 두 데이터를 청음으로 비교하였을 때 고음질을 유지할 수 있었다.

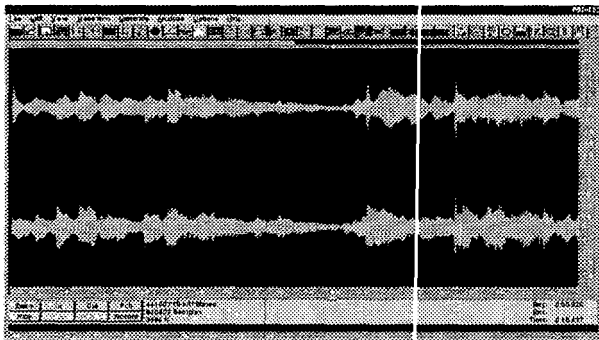


그림 6. 원음 오디오 데이터

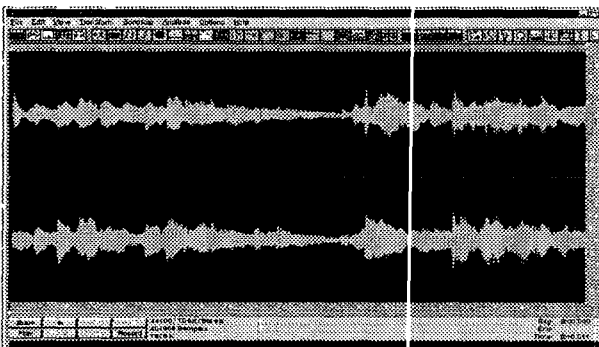


그림 7. DSP 인코딩/디코딩 결과

## 참고 문헌

1. TMS320C3x User's Guide - Texas Instrument
2. TMS320 Floating-Point DSP Assembly Language Tools User's Guide - Texas Instrument
3. VirtualLogic Emulation System Version 2.0 - IKOS system Inc
4. Davis Pan, "A Tutorial on MPEG/Audio Compress", IEEE Multimedia, pp. 60-74, Aug. 1995
5. Karlheinz Brandenburg and Gerhard Stoll, "ISO-MPEG-1 Audio: A generic Standard for Coding of High-Quality Digital Audio", J. Audio Eng. Soc. Vol. 42, No. 10, pp. 780-792, Oct. 1994
6. Yves-francois deherly, "Musicam source coding", AES 10th international conference, pp. 71-79
7. [2] Davis Pan, "A Tutorial on MPEG/Audio Compress", IEEE Multimedia, pp. 60-74, Aug. 1995
8. Karlheinz Brandenburg and Gerhard Stoll, "ISO-MPEG-1 Audio: A generic Standard for Coding of High-Quality Digital Audio", J. Audio Eng. Soc. Vol. 42, No. 10, pp. 780-792, Oct. 1994
9. 장호근, 김태훈, 광종태, 박주성, "오디오 압축 방식을 적용한 사운드합성 시스템의 설계", 한국음향학회, 제17권 제3호, pp.27-36, Mar.1998.