

# 음원 압축을 위한 다채널 MPEG 오디오 복호화기의 설계 및 구현

## (Design and Implementation of Multi-channel MPEG Audio Decoder to compress Sound)

金泰勳\*, 張豪根\*, 白光烈\*, 朴柱成\*

(Tae Hoon Kim, Ho Keun Jang, Kwang Ryul Baek, and Ju Sung Park)

### 요 약

본 연구에서는 사운드 합성에 응용할 목적으로 설계된 오디오 복호화기에 대한 내용을 담고 있다. 악기음을 ROM에 저장한 후 그 데이터를 이용하여 사운드 합성을 하는 PCM 방식에서 많은 악기음 데이터를 저장하면 할수록 더욱 좋은 음질의 사운드를 합성할 수 있다. 따라서 한정된 용량에 더욱 많은 악기음을 저장하기 위해서는 압축이 꼭 필요하다. 이를 위해서는 미리 압축한 악기음을 ROM에 저장한 후 그것을 필요시 복호화해 줄 복호화기만 있으면 된다. 그와 동시에 많은 악기음을 내기 위해서는 빠른 복호화가 필수적이다. 그래서 MPEG-1 오디오의 layer-1을 기본으로 하여 44.1 KHz의 샘플링 주파수로 32가지 악기음에 대한 실시간 복호화를 수행한다. 여기서는 음원 압축이라는 특수한 목적에 맞추어 압축의 효율성을 높이고 스스로 루프를 만들어서 합성의 편의를 제공하고 기존의 MPEG-1 오디오 layer-1에서 변형이 된 새로운 포맷과 추가된 기능에 관한 내용을 담고 있으며 이러한 복호화기를 설계하고 FPGA를 이용하여 검증하였다.

### Abstract

This research deals with the implementation procedures of an audio decoder for the musical application. It is very helpful in synthesizing high quality sound that the large amount of the sampled PCM data from original acoustic instruments are stored in ROM as possible as we can. Thus the compression of the sampled data is necessary. The music synthesis systems need only the decoder because the compressed data in ROM are prepared by offline. The algorithm implemented in this research is the MPEG-1 audio layer I, which employs the psychoacoustic model and have good data compression rate. New frame format and additional function are included in this paper to make efficient sound synthesis. The implement of an audio decoder is verified by using FPGA.

### I. 서 론

악기음 합성 방식중에서 PCM 합성 방식은 생성하고자하는 악기음 그 자체를 Sampling하여 디지털 부

호 형태로 기억장치에 저장해 두었다가 디지털/아날로그 변환을 통해 출력시키는 방식으로서 충분한 저장 용량이 있으면 거의 원음과 동일한 음색을 표현할 수 있다<sup>[1][2]</sup>. 따라서 제한된 저장 용량에 많은 Sampling 데이터를 저장하여 좋은 음질을 유지하기 위해서는 압축이 필수적이다. 본 논문에서 그러한 음원 압축 목적으로 필요한 복호화기에 대하여 논하려고 한다. 기존의 논문에서는 대부분 2 channel이나 혹은 5.1 channel인 경우이고 속도보다는 구현된 chip의 크기에 초점을 맞추는 경우가 많다. 이런 경우 32

\* 正會員, 釜山大學校 電子工學科

(Pusan national university electronics)

※ 본 연구는 부산대학교 기성회 연구비와 반도체설계

교육센터(IDEC)의 지원에 의해 수행되었습니다.

接受日字:1998年9月22日, 수정완료일:1999年3月12日

channel을 실시간에 복호화할 수 있을 정도의 성능을 얻을 수 없다<sup>[6] [12]</sup>. 또한 기존의 MPEG audio standard에서 조금 변형된 형태로 구현할 경우 더욱 효과적인 부분이 많다.

MPEG(Moving Picture Expert Group) 오디오 압축 방식은 사람의 청각 구조를 이용한 압축방식으로 최근 들어 널리 사용되고 있다<sup>[3-5] [11]</sup>. 사람의 청각 구조는 주파수 영역에서 필터 बैं크와 같이 시간 영역의 신호를 주파수 영역으로 바꾸어서 인지하며 이 때 각 주파수 대역에 따라서 민감도 또는 가청 한계가 다르다. 그리고 어떤 주파수 대역에 큰 에너지를 갖는 신호가 있을 때 주변 대역의 약한 신호를 듣지 못하는 마스킹 현상(Masking effect)이 생긴다. 입력 신호를 32개의 서브밴드 신호로 나눈 뒤에 각 서브밴드에서 마스킹 현상에 의해서 마스킹되어 인지할 수 없을 만큼의 양자화 잡음을 발생시키는 양자화 레벨을 결정한 후 그것을 이용하여 각 서브밴드 샘플이 필요한 만큼의 비트수를 할당하여 데이터를 압축할 수 있다. 그러면 주관적으로 좋은 음질을 유지하면서도 높은 압축률을 얻을 수 있는 장점이 있다.

부호화와 복호화 방법으로는 MPEG-1 오디오 Layer 1을 기본으로 하며 음원 압축이라는 특수성에 따라 약간의 변형으로 좀 더 간단하고 효율적으로 구현될 수 있다. 본 논문에서는 MPEG 오디오 방식으로 압축된 음원을 이용하여 32 채널음을 동시에 복호화해 낼 수 있는 복호화기 구현에 관한 내용을 다룬다. 본 논문은, II장에서는 MPEG 오디오 알고리즘에 대해 살펴보고, III장에서는 음원 압축인 경우 더욱 효율적인 압축을 위해서 부호화 과정이 어떻게 변해야 하느냐에 대해서, IV장에서는 III장에서 살펴본 부호화 과정을 수행할 음원 압축을 위한 복호화기의 설계에 대해서, V장에서는 시뮬레이션 결과 및 FPGA를 이용한 검증, 그리고 마지막으로 VI장에서 결론을 맺는 형태로 구성되어 있다.

## II. MPEG 오디오

### 1. MPEG 오디오 압축 원리<sup>[3] [4]</sup>

인간의 청각 구조의 가장 큰 특징 중의 하나가 마스킹 효과(masking effect)이다. 마스킹이란 주파수 혹은 시간의 차이를 둔 소리들 사이에서 하나의 소리가 다른 소리를 들리지 않게 하는 현상이다. 이 현상을 이용하면 높은 압축율을 얻을 수 있다.

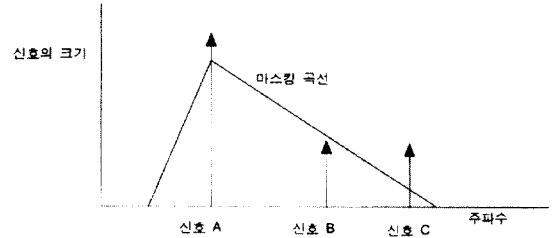


그림 1. 마스킹 곡선  
Fig. 1. Masking curve.

그림 1은 신호 A, B, C 중에서 제일 큰 신호 A에 의해 주파수 영역에서 마스킹되는 양을 표시한 삼각형 모양의 마스킹 곡선을 그려 놓았다. 이 마스킹 곡선의 아래에 있는 신호 B는 신호 A에 의해 모두 마스킹 되어 우리 귀에 들리지 않지만 신호 C의 일부는 들을 수 있다. 따라서 이 같은 경우에는 신호 B는 없어도 되며 신호 C는 약간의 변형을 하여도 우리가 인지할 수 없다. 이러한 성질을 이용하여 각 서브밴드 샘플을 지정된 비트로 표시할 경우 생기는 양자화 잡음이 마스킹 될 수 있다.

### 2. 부호화 과정

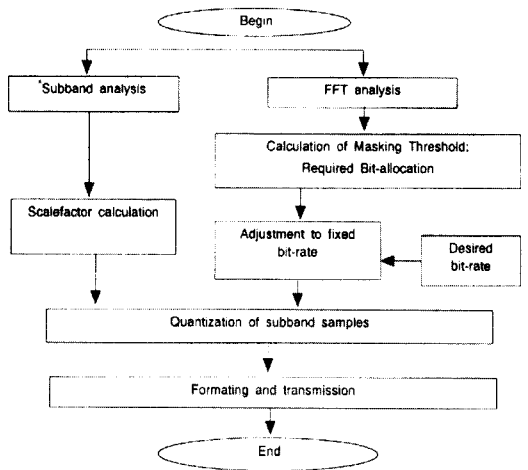


그림 2. 부호화 과정  
Fig. 2. Encoding procedure.

그림 2는 MPEG audio layer I, II의 부호화 과정을 나타내고 있다. 새로운 오디오 샘플이 입력으로 들어오면 주파수 영역에서 균일한 크기의 대역으로 분할된 32개의 서브밴드에서의 신호 크기를 구하기

위해서 서브밴드 분석 필터를 통과시켜 32개의 서브밴드 샘플을 구한다. 이와 같은 과정을 반복하여 각 서브밴드의 샘플이 12개씩 모이면 그 중 최대값을 이용하여 scalefactor를 구하며 그 값으로 나누면 서브밴드 샘플값은 -1에서 1사이로 정규화된다. 또한 오디오 샘플은 FFT를 이용하여 주파수 영역으로 변환된 후 심리음향모델을 이용하여 32개의 서브밴드에서의 마스킹 임계치를 구하게 된다. 여기서 구해진 마스킹 임계치로 각 서브밴드에서 필요한 비트수를 알 수 있다. 마스킹값이 큰 서브밴드 샘플은 샘플 표현을 위한 비트를 적게 할당하여 양자화 잡음이 많이 발생하여도 그 양자화 잡음이 전부 마스킹될 수 있지만 마스킹값이 작은 서브밴드 샘플은 많은 비트를 할당하여 양자화 잡음이 작아지도록 해야 한다.

심리음향모델에서 얻을 수 있는 것은 32개 서브밴드의 MNR(Masking to Noise Ratio)이다. 이것은 마스킹되는 양과 양자화 잡음의 비를 나타낸다. 즉 양자화 잡음이 얼마만큼 마스킹 되었는가를 나타낸다. 그러므로 MNR이 가장 작은 서브밴드부터 비트 할당을 시작한다. 전송률이 정해져 있는 경우에는 한 프레임에서 사용가능한 총 비트 수가 정해져 있으므로 할당 가능한 비트를 모두 사용할 때까지 비트를 할당한다. 비트 할당 정보를 이용하여 32개의 각 서브밴드 샘플을 마스킹값을 이용하여 구한 비트수로써 표시하는 양자화 과정을 거친 후 일정한 포맷에 따라 한 프레임을 구성하게 된다. 이렇게 구해진 MPEG-1 layer I의 한 프레임 포맷은 그림 3과 같다<sup>[5]</sup>. Header에는 syncword, layer, ID, 그리고 bit rate 등의 정보가 있고, bit allocation에서는 32개의 각 서브밴드에 할당된 bit 수를 나타낸다. Scalefactor index는 각 서브밴드에서의 scalefactor를 알 수 있는 index 값을 저장한다. 마지막으로 subband sample이 들어 있다.

Header (32 bit)	Bit allocation (128 bit)	Scalefactor index (최대 192 bit)	Subband samples
--------------------	-----------------------------	-----------------------------------	-----------------

그림 3. MPEG 오디오 계층 I의 포맷  
Fig. 3. MPEG audio layer I format.

3. 복호화 과정

복호화 과정은 그림 4와 같은 과정을 따라 수행된다<sup>[7]</sup>.

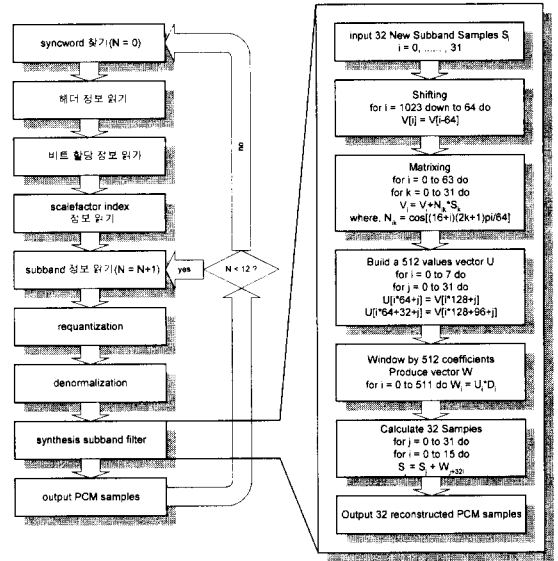


그림 4. 복호화 과정  
Fig. 4. Decoding procedure.

헤더, 비트 할당 정보, scalefactor index 정보를 읽은 다음 각 서브밴드의 비트 할당 정보를 이용하여 서브밴드 샘플을 읽어낸다. 서브밴드 샘플은 재양자화를 거친 후 scalefactor index를 이용하여 찾은 scalefactor와 곱해진 후 서브밴드 합성 필터를 통과하여 32개의 새로운 PCM 샘플이 만들어진다. 서브밴드 샘플들을 합성하는 과정은 그림 4와 같다. 새로운 32개의 서브밴드 샘플을 입력으로 받으면 matrixing 과정을 거친 후 window 계수와 곱해진다. 그러면 복호화된 32개의 샘플을 얻을 수 있다. 그리고 재양자화 공식은 다음과 같다.

$$S' = \frac{2^{nb}}{2^{nb}-1} * (S + 2^{-nb+1})$$

여기서, S'는 재양자화된 값, nb는 각 서브밴드 샘플에 할당된 비트수, S는 비트 할당 정보만큼 비트를 할당하여 양자화된 샘플값으로 복호화 과정에서 얻어지는 값이다.

III. 음원 압축을 위한 복호화 과정의 변형

1. 음원 압축의 효과

기존의 사운드 합성 시스템에서는 음원을 압축하지 않고 그대로 ROM에 저장하는 방식을 이용하였다<sup>[10]</sup>. 1Mbyte 크기의 음원 ROM에 100여 가지

의 악기음별로 몇 개의 대표음들이 저장되어 있다. 이러한 각 악기에 대한 몇 개의 대표음으로부터 interpolation과 decimation 방법을 사용하여 다른 높이의 음들을 만들어 낸다<sup>[11][8]</sup>. 대표음의 수가 많아질수록 조금만 변형하여도 나머지 음들을 만들어 낼 수 있기 때문에 원음에 더욱 가까워지며 좋은 음질을 유지할 수 있다. 이러한 장점을 얻기 위해서 음원을 MPEG 오디오 압축 방식을 이용하여 압축하는데 CD 음질이 보장되며 약 6~7배정도 더 많은 데이터를 같은 용량의 ROM에 저장할 수 있게 된다. 따라서 각 악기음별로 더 풍부하고 많은 대표음을 저장함으로써 고음질의 악기음 합성이 가능하다.

하드웨어적인 측면에서 MPEG 오디오 복호화기의 추가에 따른 부담은 있지만 사운드 합성 시스템에서는 음질이 더욱 중요한 요소로 작용하는 경우가 많기 때문에 많은 음원의 확보에 따른 음질의 개선이 더욱 중요하게 여겨질 수 있다.

2. 사운드 합성 시스템의 구조

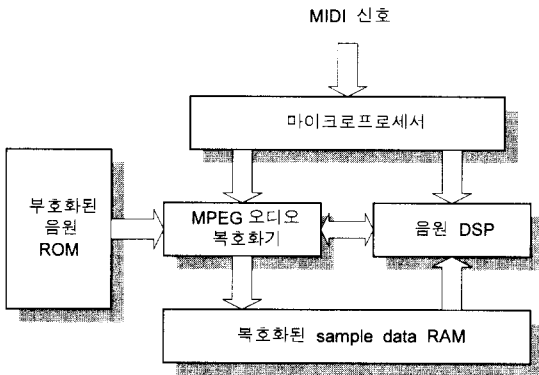


그림 5. MPEG 오디오 복호화기를 이용한 사운드 합성 시스템

Fig. 5. Sound synthesis system using MPEG audio decoder.

기존의 사운드 합성 시스템의 경우에는 음원 ROM의 데이터를 음원 DSP가 바로 읽어오는 구조로 되어 있으나 설계된 복호화기가 사용될 사운드 합성 시스템의 구조는 그림 5처럼 음원 ROM과 음원 DSP 사이에 MPEG 오디오 복호화기가 추가된다<sup>[11]</sup>.

마이크로프로세서는 외부로부터 MIDI 신호를 받고 이것을 해독하여 필요한 악기음이 복호화될 수 있도록 음원 ROM 어드레스와 복호화할 channel을 MPEG 오디오 복호화기로 보낸다. 이와 동시에 사운

드 합성에 필요한 파라미터를 음원 DSP로 보낸다. 그러면 MPEG 오디오 복호화기는 마이크로 프로세서로부터 받은 어드레스를 이용하여 음원 ROM으로부터 압축된 데이터를 읽어와서 복호화를 시작한다. 복호화된 데이터는 외부의 sample data RAM으로 보내어지게 되는데, 이 경우 마이크로 프로세서로부터 부여된 channel 정보를 이용하여 지정된 영역에 복호화된 sample data를 써 놓는다. 외부 sample data RAM에는 32 악기의 원음이 지정된 위치에 저장되어 있다. 그러면 이 데이터를 이용하여 음원 DSP에서 악기음을 합성한다. 그리고 이 합성된 sample은 D/A로 보내어지게 된다.

3. syncword 생략

부호화된 음원 ROM의 각 악기음 어드레스를 마이크로프로세서가 이미 알고 있고 음원 ROM에서 데이터를 읽어올 경우 error의 가능성은 거의 없으므로 별도로 syncword를 넣어서 각 프레임 별로 syncword를 찾아서 맞출 필요가 없다. 따라서 syncword와 그외의 압축시 알고있는 정보를 담고 있는 헤더는 필요가 없다. 그러나 한 악기음의 끝이 어디까지인지는 syncword와 비슷하게 각 frame의 처음에 decoding 비트를 넣어서 복호화를 계속할지 여부를 결정한다.

4. channel

사운드 합성시 동시에 많은 악기음을 생성하면 더욱 좋은 사운드를 합성할 수 있다. 그러기 위해서는 기존의 2 channel에서 여러개의 channel로 변환하여 한번에 많은 악기음을 동시에 복호화할 수 있어야 한다. 이를 위해서 설계된 복호화기는 32 channel을 두어 동시에 32가지 악기음을 동시에 복호화할 수 있도록 하였다. 본 연구에서는 0.8μm CMOS technology를 목표로 했기 때문에 32 채널을 설계했지만 0.6μm나 0.35μm technology를 이용하면 더 많은 채널을 복호화할 수 있을 것이다.

5. loop 기능 추가

한 가지 악기음을 처음부터 끝까지 다 저장한다는 것은 많은 데이터 저장을 요한다. 따라서 보통의 악기음은 시작 부분인 attack과 decay, 거의 변화없이 일정한 주기로 같은 파형이 반복되는 sustain 한 주기를 저장한 후 attack과 decay를 소리낸 후 sustain의 한 주기를 반복시킨다<sup>[8][10]</sup>. 따라서 적

어도 sustain 한 주기는 복호화한 후에 저장되어야 완전한 loop를 할 수 있다. Sustain의 한 주기가 짧은 경우에는 문제가 없지만 긴 경우 이 한 주기를 저장할 많은 용량의 RAM이 필요하다. channel 수가 많은 경우에는 더욱 문제가 된다. 따라서 sustain의 loop에 해당되는 정보를 미리 만들어서 이것을 헤더에 넣어주면 이 정보를 이용하여 loop의 시작시의 변수값과 상태를 그대로 저장한 후 loop가 끝나면 loop 시작할 때의 상태를 그대로 불러와서 다시 loop를 만든다. 따라서 loop가 필요한 경우 복호화시에 자동적으로 loop를 만들어 준다<sup>[9]</sup>. 그러면 음원 DSP에서는 이것을 loop에 상관없이 그대로 읽어서 합성만 하면 되므로 더욱 간단하게 loop 문제를 해결할 수 있다.

6. subband 개수 가변

악기의 종류에 따라서 신호가 주파수 모든 대역에 걸쳐서 나타나는 경우도 있고 저주파에서만 나타나는 경우도 있다. 따라서 기존의 경우 무조건 32개의 고정 subband를 이용한다는 것은 비효율적인 면이 있다. 따라서 신호의 성질에 따라서 subband 수를 다르게 하면 보다 효율적인 압축이 가능하다. 여기 구현된 복호화기는 프레임 헤더에 정보를 두어 32 subband를 이용하는 경우, 16 subband를 이용하는 경우, subband 데이터가 없는 경우로 나누어 진다. 이것을 frame마다 subband 관련 정보로 넣어둔다. 그러면 압축하는 신호가 고주파에 신호가 없는 경우에는 16 subband만 이용하고 silence인 경우에는 모든 subband 값이 0이 되도록 정보를 넣어 둔다. 그러면 비트 할당 정보에 드는 비트를 줄일 수 있고 더욱 압축율을 높일 수 있다.

7. MSB 반전 과정 생략

MSB 반전 과정은 전송시에 한 frame에서 syncword (1111 1111 1111)와 같은 값이 frame의 중간에 계속해서 나오지 않도록하기 위하여 행하는 과정이라고 볼 수 있으므로 부호화된 음원 ROM에서 데이터를 읽어 오는 경우 저장된 어드레스만 알고 있으면 syncword가 필요 없다. 따라서 복호화시에 MSB 반전 과정은 생략될 수 있다.

8. 가변 비트율 적용

압축율을 결정하는 비트 할당은 심리음향모델에서 구해지는 마스킹값을 이용하여 양자화 잡음이 모두

마스킹될 때까지만 할당하면 된다. 일반적으로 신호에 따라서 마스킹의 양이 변하게 되고 고정 비트율로 압축하면 마스킹이 많이 되는 구간에서는 양자화 잡음을 모두 마스킹하고도 비트가 남는 경우가 생길 수 있고 그 반대의 경우도 생길 수 있다. 따라서 마스킹의 양에 따라 비트율을 바꾸는 가변 비트율을 이용하여 압축율을 높일 수 있다.

9. 포맷 변형

위의 사항들을 모두 고려하여 그림 6과 같은 포맷을 만들었다. Header part는 각 악기음 데이터의 맨 처음 한번만 있고 그 이후는 Frame part가 반복된다. 각 파라미터의 의미는 다음과 같다.

loop start frame (8 bit)	loop end frame (6 bit)	loop end sample (9 bit)	decoding (1 bit)	subband number (2 bit)	bit allocation	scalefact or index	subband samples
Header part			Frame part				

그림 6. 포맷 변형  
Fig. 6. Format Modification.

- 1) Loop start frame : loop가 시작되는 frame 수를 나타낸다. 복호화시에 frame 카운터가 1부터 증가하다가 loop start frame과 같은 값이 되면 이 때부터 loop가 시작됨을 나타낸다.
- 2) Loop end frame : loop 구간내의 frame 수를 나타낸다. loop가 시작된 후 frame 카운터를 1부터 다시 증가시켜서 loop end frame과 같은 값이 되면 loop의 끝이 된다. 여기서 다시 loop start frame시의 상태로 돌아가서 loop를 다시 시작한다.
- 3) Loop end sample : 384개의 샘플이 한 개의 frame을 구성하고 loop의 샘플 갯수가 384의 배수가 되지 않는 경우가 대부분이다. 따라서 loop end sample은 loop의 마지막 frame에서 몇번째 샘플까지 loop인가를 나타낸다. loop end sample 이후 샘플은 loop 구간에 들어가지 않는다.
- 4) Decoding : 각 악기음에 대한 frame 수는 별도로 저장되어 있지 않다. 따라서 악기음의 끝을 알 필요가 있는데 decoding 비트를 이용하여 끝을 알 수 있다. decoding값이 1이면 계속 다음 frame이 존재하고 0이면 그 악기에 대한 bitstream이 끝났음을 나타낸다.

Subband number : 부호화시에 필요한 subband 수를 나타낸다. 값이 2이면 32개의 subband를 다 사용한다. 즉 bit allocation 정보가 32개 존재하고 그 이후 scalefactor도 32개 존재하고 subband sample도 32개 존재한다. 하지만 고주파가 없을 시에는 그 값이 1이 되어 16개의 subband만 사용한다. 그리고 silence 구간에서는 주파수영역에서 모든 서브밴드에서 신호가 없으므로 subband number값을 0으로 두어 bit allocation 정보와 scalefactor와 subband sample 값이 하나도 존재하지 않는다. 2개의 비트로써 4가지의 subband number가 가능하다. 하지만 구현된 하드웨어 구조상 0, 16, 32의 외의 변환은 쉽지 않기 때문에 이 3가지 경우에 대해서만 subband number가 가능하다.

#### IV. 음원 압축을 위한 복호화기 설계

##### 1. 파이프라인 구조 설계

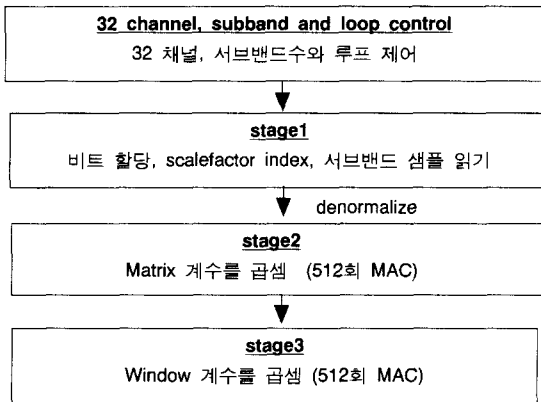


그림 7. 3 단계 파이프라인 구조  
Fig. 7. Structure of 3-staged pipeline.

복호화기가 32 채널을 실시간으로 처리하기 위해서는 빠른 처리 속도가 요구된다. 그러므로 복호화 과정을 순차적으로 처리하는 구조보다 복호화기를 파이프라인 구조로 설계하면 보다 빠른 처리 속도를 얻을 수 있다. 복호화에 필요한 과정을 분석하여 각 단계별로 작업량을 비슷하게 분산될 수 있게 그림 7과 같이 3단계 파이프라인을 구성하였다.

##### 2. 전체 시스템 구성 및 동작

32 채널을 실시간으로 복호화할 수 있는 MPEG 오디오 복호화기의 전체 시스템 구성은 그림 8에 나

타냈다.

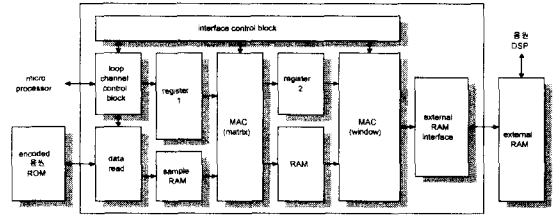


그림 8. 전체 블록도  
Fig. 8. Top block diagram.

마이크로 프로세서로부터 합성하고자하는 악기음이 있는 어드레스와 복호화를 원하는 채널 정보를 채널 컨트롤 블록에 써 주면 그 정보에 따라 음원 ROM로부터 압축된 데이터를 읽어와서 각 채널을 순차적으로 복호화를 수행하게 되고 각 stage사이의 인터페이스에 관련된 register1과 register2는 각 stage에서의 변수를 저장한다. Data read 블록은 loop 정보, 비트 할당 정보, scalefactor index와 서브밴드 샘플을 읽은 후 matrix 계수를 계산하는 MAC(matrix) 블록으로 넘겨주는 역할을 한다. 또한 Loop 정보를 현재의 frame 카운터와 비교하여 loop 시작점을 찾아내어 loop의 시작을 다른 블록에 알려주고, loop 시작의 상태를 저장한다. 각 채널의 복호화를 수행하다가 loop가 끝나는 점에서도 역시 loop가 끝났음을 다른 블록에 알려주는 역할을 한다. Data read 블록에서 읽혀진 서브밴드 샘플은 다음 파이프라인 스테이지로 넘겨주는 과정에서 scalefactor와 곱해지게 되어 있다. MAC(matrix)에서는 넘겨받은 서브밴드 샘플을 matrix 계수와 MAC(Multiply and Accumulate)을 수행하여 32개의 buffer 값을 만든다. MAC(window)에서는 512개의 buffer 값을 window 계수와 512회의 MAC을 수행하여 32개의 복호화된 16 비트 PCM 샘플을 만들어 낸다. PCM 데이터의 loop가 시작하면 512개의 buffer 값은 외부 RAM에 저장된다. 그리고 이 값들은 loop의 한 주기가 끝나서 다시 loop의 처음으로 돌아갈 때 쓰이게 된다. 복호화한 후 외부 RAM에 저장하게 되며 음원 DSP가 외부 RAM의 데이터를 읽어서 악기음을 합성하게 된다. 복호화기의 출력은 32 채널 PCM 데이터가 각각 출력되도록 되어 있다. MPEG 오디오 알고리즘 특성상 layer-1인 경우는 각 채널의 384개 샘플을 동시에 복호화해야 하므로 32 채널의 데이터

가 1차적으로 복호화 되는 데까지는 latency가 있다.

### V. 설계 검증

복호화기 설계를 위하여 C 언어를 이용한 MPEG 오디오 복호화기의 정수 시뮬레이터, 구조 및 논리 설계, FPGA를 이용한 검증 등의 과정을 거쳤다. MPEG 오디오 복호화기 시뮬레이터는 MPEG 오디오 디코더 구현 표준에 따라 코딩된 16 비트 정수형 시뮬레이터로 설계의 검증 기준으로 사용했다.

#### 1. 로직 시뮬레이션

본 연구에 사용된 툴은 COMPASS이며 사용된 라이브러리는 0.8 $\mu$ m CMOS SOG 셀이다, 로직시뮬레이션을 통하여 디코딩된 악기음과 정수형 시뮬레이터의 결과와 정확하게 일치하는 것을 확인하였다. 현악기, 관악기, 타악기에 대한 두 경우의 결과를 그림 9-14에 나타냈다. 그림 9 - 10은 Acoustic Guitar G4, 그림 11 - 12는 Baritone Saxophone A#3, 그림 13 - 14는 타악기인 Cowbell에 대한 시뮬레이션 결과이다.

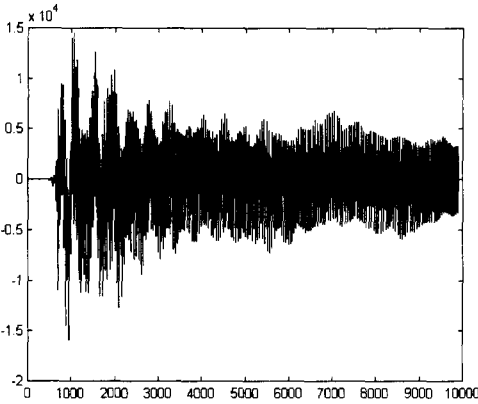


그림 9. 복호화기의 로직 시뮬레이션 결과 (Acoustic Guitar G4)  
Fig. 9. The logic simulation result of decoder (Acoustic Guitar G4).

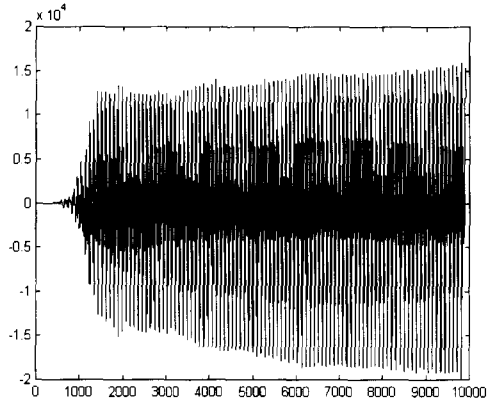


그림 11. 복호화기의 로직 시뮬레이션 결과 (Baritone saxophone A#3)  
Fig. 11. The logic simulation result of decoder (Baritone saxophone A#3).

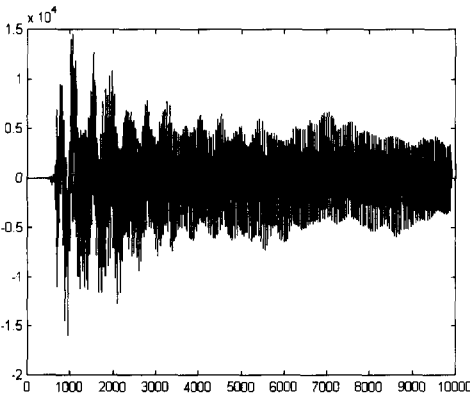


그림 10. C 프로그램을 이용한 복호화 결과 (Acoustic Guitar G4)  
Fig. 10. The decoding result using C program (Acoustic Guitar G4).

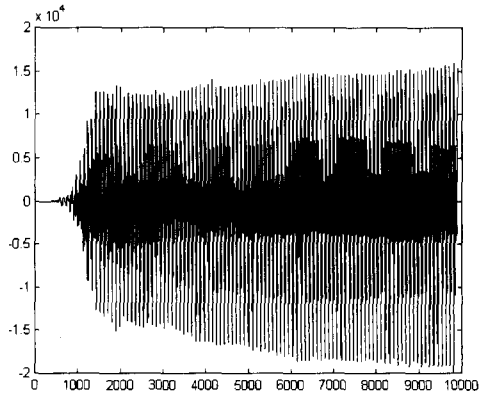


그림 12. C 프로그램을 이용한 복호화 결과 (Baritone Saxophone A#3)  
Fig. 12. The decoding result using C program (Baritone Saxophone A#3)

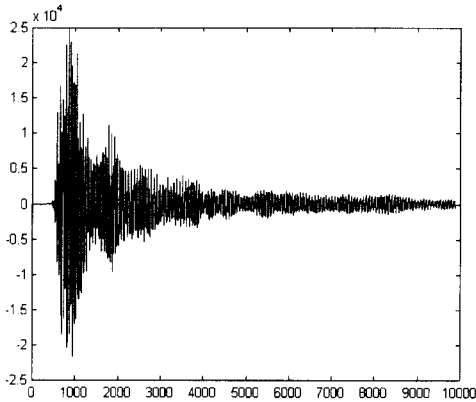


그림 13. 복호화기의 로직 시뮬레이션 결과 (Cowbell)  
Fig. 13. The logic simulation result of decoder (Cowbell).

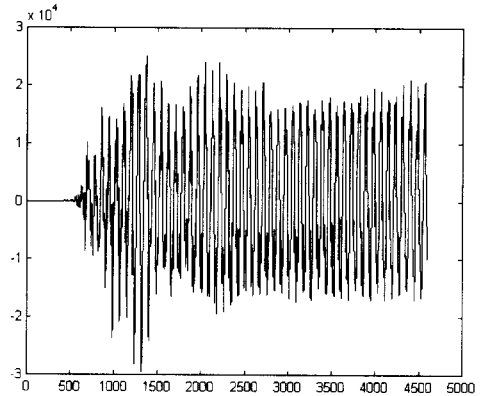


그림 15. Grand piano C4의 파형  
Fig. 15. Waveform of Grand piano C4.

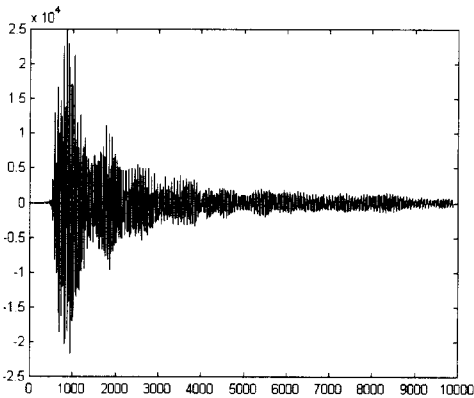


그림 14. C 프로그램을 이용한 복호화 결과 (Cowbell)  
Fig. 14. The decoding result using C program (Cowbell).

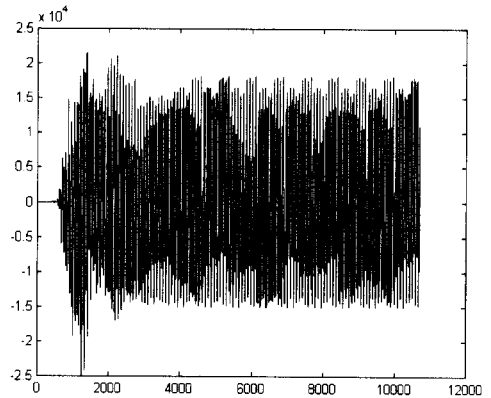


그림 16. 루핑 로직 시뮬레이션 결과  
Fig. 16. The logic simulation result of looping.

복호화기는 압축된 음을 복호화하는 기능외에 일정한 파형을 반복해주는 looping 기능이 있어야 한다. 그림 15는 Grand piano C4 파형인데 looping하기 전이다. 이 파형을 이용하여 sustain 영역에서 761 샘플을 주기로 looping 시키면서 복호화한 결과를 그림 16에 나타냈다. 여기서 사용된 파라미터는 loop\_start\_frame=10, loop\_end\_frame=2, loop\_end\_sample =377이다. 이것은 3840 samples(10 frame \* 384 samples/frame)을 복호화 다음 loop가 시작하게 되며 loop 주기는 768 samples(2 frame=loop\_end\_frame)에서 7 samples(384 samples/frame

- loop\_end\_sample)을 빼면 된다.

설계된 복호화기가 CD급 음질(44.1KHz, 16 비트)을 보장해주기 위해서는 32MHz 클럭을 사용하여 20 사이클마다 하나의 샘플 데이터를 만들어 내야 한다. 논리 시뮬레이션을 통하여 이것을 확인하기 위하여 worst case(4.75V, 70℃) 조건에서 동작시켜본 결과 정상적으로 샘플을 만들어 내는 것을 확인하였다.

## 2. FPGA를 이용한 검증

그림 17에서는 설계된 MPEG 오디오 복호화기는 ALTERA사의 FLEX10K100 FPGA 1개를 이용하여 로직을 검증한 블록도를 나타내고 있다. MPEG 오디오 복호화기는 COMPASS tool을 이용하여 설



계 되어있기 때문에 이것을 ALTERA사의 FPGA를 이용하기 위해서는 변환 과정이 필요하다. COMPASS schematic으로 설계된 회로를 VHDL로 변환 후 SYNOPSIS를 이용하여 ALTERA의 FLEX10K library로 합성하였다. 그리고 이것을 edif 포맷으로 저장한 다음 maxplus tool과 FLEX10K100 FPGA를 이용하여 설계를 검증하였다. 음원 ROM대신 PC에서 부호화된 음원 데이터를 DRAM에 다운로드시키면 FPGA는 DRAM의 다운로드된 데이터를 이용하여 복호화한 후 복호화된 샘플을 D/A 컨버터로 보내면 스피커로 청취할 수 있다. 32 channel인 경우 약 32MHz의 빠른 동작주파수를 요구하므로 FPGA로 검증하기 위해서 2 channel로 줄였다. 외부 clock은 8MHz를 사용하였고 이것으로 DRAM을 제어하기 위해서 4분주하여 FPGA 내부의 clock으로 사용한다. FPGA를 이용하여 복호화기의 로직 에뮬레이션한 결과 파형은 그림 18에서 볼 수 있으며 이것을 C 프로그램을 이용하여 복호화한 결과 파형은 그림 19에 나타내었다.

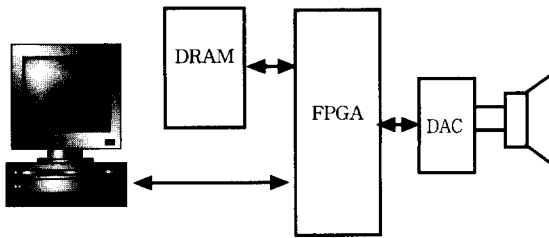


그림 17. FPGA를 이용한 검증  
Fig. 17. Verification Using FPGA.

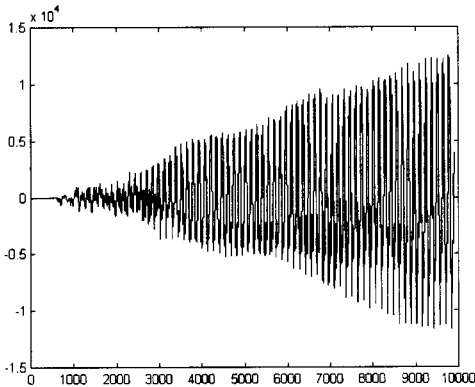


그림 18. 로직 에뮬레이션 결과(Cello E3)  
Fig. 18. The logic emulation result(Cello E3).

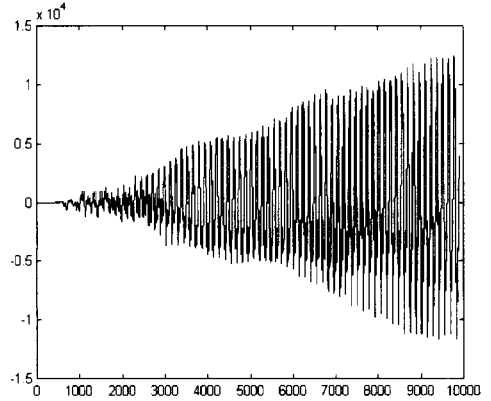


그림 19. C 프로그램을 이용한 복호화 결과 (Cello E3)  
Fig. 19. The decoding result using C program (Cello E3).

### VI. 결 론

같은 용량에 더 많은 악기음을 압축하여 저장함으로써 보다 좋은 음질의 악기음을 합성하기 위한 음원 압축을 위한 MPEG-1 layer-1 오디오 복호화기를 0.8  $\mu$ m SOG 라이브러리와 설계 tool인 COMPASS를 이용하여 설계하였다. 그리고 COMPASS로 설계된 로직을 ALTERA사의 Flex10K100 FPGA를 이용하여 검증하였다. 설계된 MPEG 오디오 복호화기는 3 단계 파이프라인 구조로 되어 있으며 32 MHz로 동작하여 CD 음질로 32 악기음을 동시에 복호화할 수 있음을 확인하였다. 악기음의 압축율은 음질을 기준으로 양자화 잡음이 모두 마스킹될 때까지 비트 할당이 이루어졌기 때문에 악기에 따라 다르다. 4:1에서 11:1까지 다양한 압축률을 보였으며 평균 7:1 정도이다. 그리고 부호화할 데이터의 중간에 silence가 포함될 경우는 더욱 높은 압축율을 얻을 수 있으며 합성의 편의를 위해서 루핑 기능을 갖추고 있다. 2 채널 악기음을 복호화할 수 있는 칩을 FPGA로 구현하여 여러 가지 악기음을 복호화와 루핑 과정을 거쳐서 하여 주관적인 평가를 하여 만족할 만한 결과를 얻었다. 그리고 MPEG-1 layer-1의 정수형 시뮬레이터와 동일한 결과를 얻었다. 이러한 압축을 통하여 더욱 풍부한 악기음과 다양한 효과음을 저장함으로써 보다 좋은 음질의 사운드합성이 가능하리라고 본다. 여기에 설계된 회로는 음원뿐만 아니라 음성이나 음악을 압축하고 그것을 실시간으로 복호화하여 응용하는 많은 분야에 적용될 수 있을 것이다.

참 고 문 헌

- [ 1 ] 장호근, 김태훈,곽중태, 박주성, “오디오 압축 방식을 적용한 사운드합성 시스템의 설계”, 한국음향학회, Vol.17, No.3, pp. 27-36, 1998
- [ 2 ] 박주성 외, “고음질 악기음 합성을 위한 핵심 기술 연구”, 한국과학재단 보고서, 1997
- [ 3 ] Y. F. Dehery, et al. “a MUSICAM source codec for digital audio broadcasting and storage.”, Proc. ICASSP pp.3605-3608, 1991.
- [ 4 ] Davis Pan, “A Tutorial on MPEG/Audio Compress”, IEEE Multimedia, pp.60-74,1995.
- [ 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, 1994 October.
- [ 6 ] Tsung-Han Tsai, Thou-Ho Chen, and Liang Gee Chen, “An MPEG Audio Decoder Chip”, IEEE trans. on Consumer Electronics, Vol.41, No.1, pp.89-96, Feb. 1995.
- [ 7 ] 김태훈, “MPEG 오디오 복호화에 관한 연구”, 부산대학교 전자공학과 석사학위논문, 1997
- [ 8 ] Curtis Roads, “The Computer Music Tutorial,” MIT press, pp. 120-123, 1996.
- [ 9 ] Thomas D.rossing, “The Science of Sound”, Addison Wesley pub., 1990.
- [ 10 ] 장호근, “악기음 합성을 위한 음원 DSP의 설계 및 FPGA 구현”, 부산대학교 전자공학과 석사학위논문, 1995
- [ 11 ] Yves-francois dehery, “Musicam source coding”, AES 10th international conference, pp. 71-79.
- [ 12 ] Sung-Chul Han, Sun-Kook Yoo, Nam-Han Jeong, Sung-Wook Park, Joon-Suk Kim, Young-lae Han, Dae-Hee Yoon, “An Asic Implementation of The MPEG-2 Audio Decoder”, IEEE trans. on Consumer Electronics, Vol.42, No.3, pp.540-545, Jan, 1996

저 자 소 개



金 泰 勳(正會員)

1972년 4월 25일생. 1995년 2월 : 부산대학교 전자공학과 (공학사). 1997년 2월 : 부산대학교 전자공학과 (공학석사). 1997년 2월 - 현재 : 부산대학교 전자공학과 대학원 박사과정, 컴퓨터 및 정보통신연구소 연구원.

주관심분야는 사운드 압축 및 복원, DSP 설계

白 光 烈(正會員) 第 35卷 S編 第 8號 參照



張 豪 根(正會員)

1968년 1월 15일생. 1993년 2월 : 부산대학교 전자공학과 (공학사). 1995년 2월 : 부산대학교 전자공학과 (공학석사). 1995년 2월 - 현재 : 부산대학교 전자공학과 대학원 박사과정. 주관심분야는 DSP 설계, 사운드

드 합성



朴 柱 成(正會員)

1953년 12월 19일생. 1976년 2월 : 부산대학교 전자공학과 (공학사). 1978년 2월 : 한국과학기술원 전기 및 전자공학과 (공학석사). 1979년 3월~1985년 7월 : 한국전자기술연구소 1989년 8월~1991년 3월 : 한국

전자통신연구소 책임연구원. 1991년 3월~현재 : 부산대학교 전자공학과 부교수, 컴퓨터 및 정보통신연구소 겸임연구원. 주관심분야는 DSP 설계 및 응용, 사운드 합성, 반도체 소자 모델링