

오디오 압축 방식을 적용한 사운드 합성 시스템의 설계

Design of Sound Synthesis System using Audio Compression Method

장 호 근*, 김 태 훈*, 광 종 태**, 박 주 성*
(Ho Keun Jang*, Tae Hoon Kim*, Jong Tae Kwak**, Ju Sung Park*)

※이 연구는 94년도 한국과학재단 연구비 지원(과제번호:94-0100-12-01-3)과 반도체설계교육센터(IDEC)의 지원에 의해 수행되었습니다.

요 약

현재 상용화된 사운드 합성 기기에서 널리 쓰이고 있는 PCM 방식에서의 문제점은 고음질의 음을 얻기 위해서 많은 메모리 용량을 필요로 하는 것이다. 이 논문에서는 이 문제를 해결하기 위해 MPEG 오디오 압축 방식을 적용하여 샘플된 음을 압축하고, 실시간으로 이를 복호화 해서 음을 합성해내는 사운드 합성 시스템을 설계하였다. 사운드 합성 시스템은 마이크로프로세서, 음원 DSP, MPEG 오디오 복호화기로 구성되며, 44.1Khz의 샘플링 주파수로 32개의 음을 동시에 합성할 수 있도록 설계되었다. 설계 과정에서 각각의 기능 요소를 C 언어로 기술하여 사운드 합성 시스템에 대한 소프트웨어 모델을 작성하였다. 이것을 통해 미리 전체 시스템의 동작을 시뮬레이션하고, 압축 방식을 적용함으로써 발생할 수 있는 여러 가지 문제점에 대한 해결 방안을 제시하였다. 시뮬레이터로 시스템의 동작을 검증한 후, DSP와 MPEG 복호화기를 포함하는 사운드 합성 시스템을 VHDL로 설계하여 시뮬레이션을 통해 하드웨어가 정상적으로 동작함을 확인하였다. MPEG 오디오 압축 방식을 이용함으로써 메모리 용량 측면에서는 약 8:1의 감소 효과를 얻을 수 있었다.

ABSTRACT

PCM or wavetable synthesis is the most widely used sound synthesis method in commercial sound synthesis systems. The problem of this method is requiring much memory capacity for high quality sound. In this paper to settle the problem MPEG audio compression method is applied to the sound synthesis system in which sound samples are compressed by MPEG encoding process and then stored in memory. The system decodes sound samples in real-time and synthesizes corresponding sound. It consists of a microprocessor, a sound DSP, and an MPEG audio decoder. It can synthesize 32 sounds simultaneously at 44.1Khz sampling rate. During design process, a software simulator was written in which each component of the system was described in C language. Through the behavioral simulation of the system, several problems were introduced and settled out. After that the VHDL description of the hardware of the system was written and simulated to confirm complete sound synthesis operation. By applying MPEG audio compression method the memory capacity was reduced by about 1/8.

I. 서 론

현재 상용화된 사운드 합성 기기에서 가장 널리 쓰이고 있는 것은 FM 방식과 PCM 방식의 합성 알고리즘이다. 이들 알고리즘은 비교적 간단한 계산과 적은 하드웨어만으로 실제 악기음에 가까운 음을 합성할 수 있음 뿐 아니라 여러 가지 다양한 사운드를 합성해 낼 수 있는 장점을 가지고 있다. FM 합성 방식은 여러 개의 정현파를 복잡한 형태로 연결하여 구성함으로써 동적 스펙트럼이 풍부

하고 여러 가지 다양한 음의 사운드를 창출할 수 있는 반면, 실제 악기음을 묘사하기 위한 적합한 파라미터를 추출하는 과정이 어렵고, 타악기와 피아노와 같은 복잡한 스펙트럼을 가지는 악기음에 대해서는 완벽한 표현이 어려운 점이 있다. 웨이브 테이블 합성 방식이라고도 부르는 PCM 합성 방식은 실제 사운드를 채취하여 메모리에 저장시켜 놓고 이것을 바탕으로 원하는 사운드를 합성해 내는 방식으로, 합성된 음이 실제의 악기음에 매우 가까운 음질을 낼 수 있는 반면 이를 위해서는 많은 메모리 용량을 필요로 한다는 단점이 있다. 그럼에도 불구하고 현재는 PCM 방식의 사운드 합성이 우위를 점하고 있는 실정인데, 그것은 멀티미디어 시대의 요구에 따라 보다 나

*부산대학교 전자공학과

**현대전자

접수일자: 1998년 1월 12일

은 음질을 추구하려는 욕구에 부응하려는 것이라 볼 수 있다.

이 논문에서는 PCM 방식의 합성 방식에서 가장 문제가 되는 메모리 크기와 음질의 문제를 오디오 압축 방식을 적용하여 해결하고자 한다. 오디오 압축은 주로 음성이나 연주된 음악을 압축, 전송하는데 적용이 되어 왔지만 사운드 합성 시스템에서 샘플링 과정의 압축에도 적용할 수 있다. 이 논문에서는 사람의 청각 구조를 고려하여 주관적인 고음질을 얻도록 설계된 MPEG 오디오 압축 기법[1][2]을 사운드 샘플에 적용시켜 압축 저장하고, 이것을 실시간으로 복호화하여 사운드를 합성해 낼 수 있는 사운드 합성 시스템을 설계하였다. II장에서는 설계된 사운드 합성 시스템의 구조와 구성 요소에 대해 살펴보고, III장에서는 시스템 시뮬레이터를 통한 기능 레벨에서의 설계에 대해서, IV장에서는 VHDL을 통한 하드웨어 설계와 검증에 대해서 살펴보고, 마지막으로 V장에서 결론을 맺도록 한다.

II. 사운드 합성 시스템의 구조와 구성 요소의 설계

설계된 사운드 합성 시스템은 그림 1과 같은 구조로 되어 있다. 이 시스템은 MIDI(Musical Instrument Digital Interface)에 기반을 둔 사운드 합성 시스템으로서 마이크로프로세서, 음원 DSP, MPEG 복호화기 등의 기능 요소로 구성된다. 각 기능 요소와 역할과 설계에 대해서 살펴보면 다음과 같다.

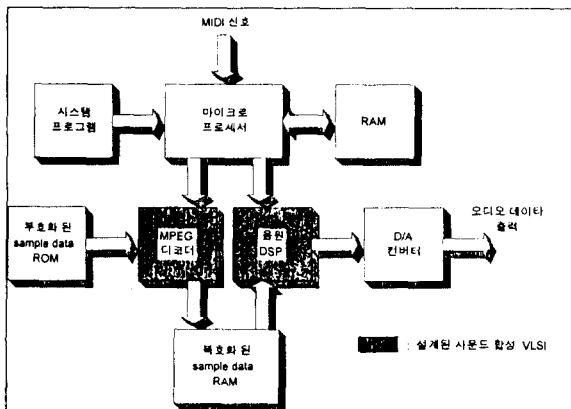


그림 1. MPEG 오디오 압축 방식을 적용한 사운드 합성 시스템
Fig. 1 Sound synthesis system with an MPEG decoder.

2.1 마이크로프로세서와 시스템 프로그램

여기서 마이크로프로세서는 미디 컨트롤러라고 볼 수 있다. 마이크로프로세서는 입력되는 미디 신호를 해독하고 이에 따라 음원 DSP와 MPEG 복호화기에 적절한 파라미터와 제어 신호를 전달함으로써 이들이 조화롭게 동작될 수 있도록 제어하는 역할을 수행한다. 설계된 시스템

에서는 인텔의 8비트 마이크로컨트롤러인 8032를 사용하고 있다. 8032를 통하여 이루어지는 모든 시스템의 동작은 시스템 프로그램에 의해 기술된다. 시스템 프로그램은 크게 초기화 루틴, 미디 신호 처리 루틴, 그리고 실시간 처리 루틴으로 나눌 수 있다[3]. 초기화 루틴은 시스템의 전원이 켜졌을 때 행하는 일들로 8032의 초기화, 외부 메모리 초기화, 음원 DSP와 MPEG 복호화기의 초기화 등의 일을 수행한다. 초기화 루틴이 끝나면 미디 신호가 입력되기를 기다리다가, 미디 신호가 들어오면 이를 해독하고 관련된 명령을 수행하는 루틴으로 가게 된다. 여기에는 악기음의 연주 시작과 연주의 끝, 음의 높이와 세기 정보를 처리하는 루틴, 미디 프로그램 변경과 관련된 루틴, 미디 컨트롤 신호를 처리하는 루틴, 그리고 전자악기의 피치 밴드 휠을 변경할 때 발생하는 미디 신호를 처리하는 루틴 등으로 구성된다. 실시간 처리 루틴은 현재 합성되고 있는 음에 대해 음정, 음색, 음량의 변화를 주기 위한 것으로 일정 시간 간격마다 인터럽트를 발생시켜 합성음의 주파수와 진폭에 관련된 파라미터 값을 변경시키게 된다.

2.2 음원 DSP

음원 DSP는 FM 방식과 PCM 방식을 합성할 수 있는 사운드 합성 전용의 DSP로 설계되었다[3]. 이 DSP는 44.1Khz의 샘플링 주파수로 16비트로 된 32개의 음을 동시에 합성해 낼 수 있다. 서로 다른 구조와 합성 방식으로 된 8개의 합성 알고리즘을 구현하는 코드를 내장하고 있으며, 4개의 출력 채널을 통하여 음이 출력되는 채널을 지정할 수 있어서 입체감 있는 사운드를 발생시킨다. 또한 향상된 마이크로아키텍처를 채택하여 여러 개의 명령어가 한 클럭 사이클 내에서 동시에 수행될 수 있게 함으로써 낮은 클럭 주파수에서 높은 동시발음수를 얻을 수 있다.

2.3 MPEG 복호화기

MPEG 오디오 압축 기법은 기존의 서브밴드 코딩 기법에 사람의 심리 음향 모델을 적용한 것으로, 낮은 비트율에서 고음질의 음을 얻을 수 있다. 설계된 MPEG 복호화기는 ISO MPEG/오디오 계층 1을 근간으로 하였으며, 음원 DSP의 32개 음 합성에 맞추어 32개의 복호화 채널을 실시간으로 처리할 수 있도록 하였다. 44.1Khz의 샘플링 주파수로 32개의 음을 합성하기 위해서는 0.709μs(1/(44100×32))동안에 심리음향으로 압축된 32종의 음의 현 샘플씩을 복호화해야 한다. 이러한 문제를 해결하기 위하여 복호화 과정을 채널에 따라 순차적으로 처리하지 않고, 3단계의 파이프라인 구조로 설계하여 빠른 처리 속도를 얻을 수 있도록 하였다[4]. MPEG 알고리즘을 사운드 샘플의 압축에 적용하는 경우 알고리즘의 기본 개념은 그대로 적용시키고, 부호화시 헤더에 들어가는 여러 가지 정보들은 필요에 의해 변경되거나 추가될 수 있다. 이것은 3.1절에서 살펴보도록 한다.

이 외에도 설계된 사운드 합성 시스템에는 PCM 샘플 데이터를 저장하기 위한 것으로 한 개의 롬과 한 개의 램이 있다. PCM 롬은 MPEG으로 부호화된 샘플 데이터를 저장해 놓은 것이고, PCM 램은 MPEG 복호화기가 복호화된 샘플을 저장하고 DSP가 여기에서 필요한 샘플 데이터를 읽어가도록 하기 위해 필요한 메모리이다

이상과 같은 기능 요소로 이루어진 사운드 합성 시스템의 동작은 다음과 같이 이루어진다. 마이크로프로세서에 미디 신호가 입력되면 마이크로프로세서는 이것을 해독해서 해당 사운드를 합성하기 위한 파라미터 값들을 결정한다. 그리고 먼저 MPEG 복호화가 쪽으로 PCM 롬에 저장된 사운드 샘플 데이터의 어드레스와 복호화 채널 번호, 복호화 시작 신호 등을 전달하게 되면, MPEG 복호화기는 메모리에서 샘플을 읽어와서 복호화를 수행하고 이것을 PCM 램의 지정된 채널 번지에 써넣게 된다. MPEG 복호화기에서 복호화가 진행되고 있는 동안 마이크로프로세서는 음원 DSP의 합성 알고리즘 수행에 필요한 파라미터 값들을 계산해서 음원 DSP에 전달하고, MPEG 복호화기가 PCM 램에 데이터를 써 놓은 후 DSP로 합성 시작 신호를 보내게 된다. 음원 DSP의 합성이 이루어지는 동안 MPEG 복호화기는 계속 마이크로프로세서로부터 그 다음 채널의 정보를 받아서 복호화 동작을 수행하게 된다.

합성 시스템의 전체 동작에 대한 개념은 간단하게 보이지만, 실제 이것을 하드웨어로 구현하는데는 여러 가지 문제점이 발생하게 된다. 예를 들어 악기음 루프의 처리 문제라든가 PCM 램의 채널당 바퍼 크기, 그리고 음원 DSP와 MPEG 복호화기의 메모리 공유 문제 등이 해결되어야 한다. 이를 위해 실제 사운드 합성 시스템의 동작에 대한 시뮬레이터를 작성하여 각 기능 요소의 하드웨어 동작을 소프트웨어적으로 묘사하고, 이것을 시스템 레벨에서 동작시켜봄으로써, 최적화된 시스템 환경과 하드웨어 구조를 결정할 수 있었다. 이것을 다음 장에서 설명하도록 한다.

III. 시스템 시뮬레이터를 통한 기능 레벨 설계

시스템 레벨의 시뮬레이터는 기능적인 측면과 기능 요소간 인터페이스 신호에 대한 타이밍 측면에서 살펴볼 수 있다. 위에서 설명한 체계의 주요 기능 요소간의 구체적인 타이밍까지 고려된 시뮬레이터를 작성할 경우 마이크로프로세서에 대한 시뮬레이터가 필요하고, 미디 신호 입력의 단계에서부터 실시간 처리까지를 모두 고려해야 하므로 시뮬레이터가 복잡해지고 시뮬레이션 시간이 많이 걸리게 된다. 실제로 사운드 합성 기능을 수행하는 것은 음원 DSP와 MPEG 복호화기이므로, 시스템 시뮬레이터는 이 두 개의 기능 요소만을 C로 모델링 하여 기능적인 측면과 타이밍 측면을 살펴보았다. 마이크로프로세서는 시스템 프로그램의 수행 결과인 여러 가지 파라메

터와 제어신호를 음원 DSP와 MPEG 복호화기에 인가해 주는 일을 수행한다. 따라서 마이크로프로세서의 동작은 음원 DSP와 MPEG 복호화기에 전달되는 파라미터들과 제어 신호를 시스템 실행 타이밍에 맞추어 적절한 시기에 시뮬레이터의 입력으로 가해 주는 것으로 충분하다.

그림 2에는 설계된 사운드 합성 시스템의 시뮬레이터에 대한 흐름도를 나타내었다. 음원 DSP는 구체적인 하드웨어 동작을 표현하여 클럭 사이클 레벨에서 동작하는 것으로 모델링 하였고, 이것이 시뮬레이터의 기본 동작 사이클이 된다. MPEG 복호화기는 알고리즘 측면에서 모델링 되어 한 프레임의 부호화된 데이터를 읽어와서 이것을 복호화 하여 384개의 복호화된 샘플을 내놓는 것으로 되어 있다. 설계된 음원 DSP는 하나의 합성음을 만들어 내기 위해 24 사이클이 소요되므로, 44.1KHz의 샘플링 주파수로 32개의 음을 동시에 합성하기 위해서는 $44100 \times 32 \times 24 = 33.8688\text{Mhz}$ 로 동작하여야 한다. 한편, MPEG 복호화기는 각 복호화 채널에 대해 1초에 44100개의 샘플을 복호화 해 낼 수 있어야 한다. 설계된 MPEG 복호화기에서는 20 사이클마다 한 개씩 복호화된 샘플을 PCM 램에 쓰게 되어 있으므로[4], 음원 DSP가 무리 없이 샘플을 읽어가기 위한 동작 주파수는 $44100 \times 32 \times 20 = 28.224\text{Mhz}$ 이상이 되어야 한다. 실제로 MPEG 복호화기의 동작 주파수는 악기음을 샘플링 할 때 대표음의 설정과 한 대표음으로부터 음정을 변화시킬 수 있는 정도에 직결적인 연관이 있으며 이것은 3.3절에 설명되어 있다. 그림 2에 나타난 시뮬레이터에서는 음원 DSP와 MPEG 복호화기의 동작 주파수를 같게 한 경우로, MPEG 복호화기는 각 채널 복호화 시작시 384개의 샘플을 미리 복호화 해 놓고, 시뮬레이터의 20개 클럭 사이클마다 한 샘플씩 PCM 램에 써넣는 것으로 모델링 하였다. 위에서 언급한 내용들이 고려된 시뮬레이터의 전체 동작은 다음과

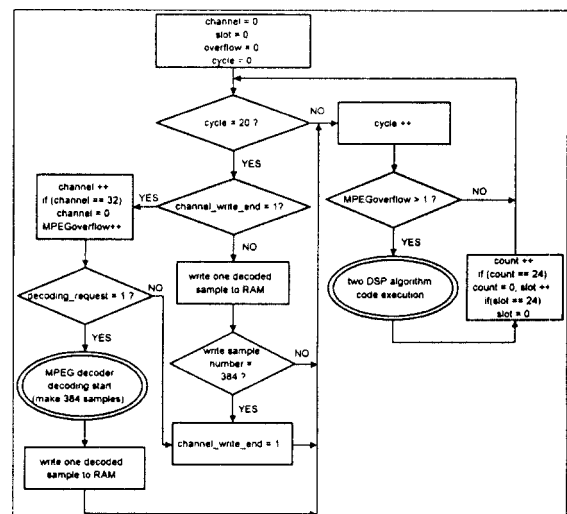


그림 2. 사운드 합성 시스템 시뮬레이터의 흐름도

Fig. 2 Flowchart of the simulator of sound synthesis system.

같다. 초기에 MPEG 복호화기 루틴에서 0번 채널부터 복호화를 시작하여 31번 채널까지 복호화를 수행해서 모든 채널에 대해 첫 번째 프레임의 복호화된 데이터가 PCM 램에 쓰여진 상태(MPEGOverflow>1)가 되면, DSP 루틴으로 들어가서 0번 채널부터 합성 알고리즘이 매 사이클 수행된다. 그 다음부터는 DSP의 합성 알고리즘이 수행되는 20 사이클마다 MPEG 복호화기 루틴으로 들어가서 현재 복호화 채널의 복호화된 데이터를 PCM 램에 써넣게 된다. MPEG 복호화기에서 한 채널에 대한 384개 샘플이 램에 쓰여지고 나면, channel_write_end 신호가 1이 된다. 다음에 MPEG 복호화기 루틴으로 들어오면 채널 번호가 증가하고, 증가된 채널의 복호화가 진행되어 384개의 샘플이 나오게 되고 20 사이클마다 한 샘플씩 PCM 램에 쓰여지게 된다.

MPEG 오디오 압축 기법을 도입함으로써 발생하는 몇 가지 문제점들을 시스템 레벨 시뮬레이터를 통하여 어떻게 해결했는지 살펴보자.

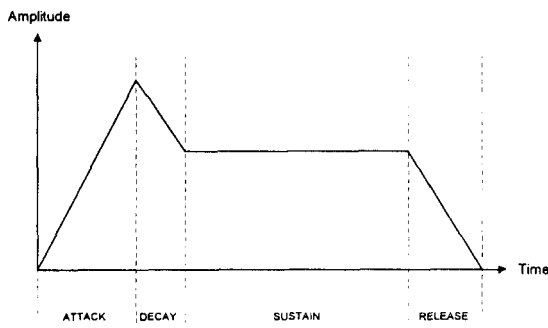


그림 3. 악기음의 일반적인 포락선 모양
Fig. 3 Envelope shape of acoustic sound.

3.1 루프 구간의 복호화 방법

대부분의 악기음의 파형은 그림 3에 나타난 바와 같이 attack, decay, sustain, 그리고 release의 네 개의 영역으로 구분될 수 있다. 여기서 attack과 decay 부분은 악기음 자체의 특성이 가장 두드러지게 나타나는 것으로 파형 자체도 불규칙 하게 나타나지만, sustain과 release 부분은 주기성이 강하게 나타난다. 이것을 이용하여 PCM 합성 방식에서는 주기성이 나타나는 부분에서 루프 구간을 설정해서 메모리에 저장하고, sustain과 release는 이 부분을 반복하여 파형을 만들어 내게 된다[5]. 압축 방식이 포함되지 않은 PCM 방식에서는 파형의 루프 구간에 대한 이드레스를 미리 저장하고 있다가 파형의 끝부분에 도달하면 루프 시작 어드레스로 가서 샘플을 읽어 오게 된다[3]. MPEG 오디오 압축 기법을 이용하는 경우에는 부호화시 루프 시작 구간이 항상 프레임의 시작에 오도록 조정해 주어야 한다. 만약, 샘플된 악기음 파형을 처음부터 384

(MPEG의 한 프레임에 해당하는 샘플 개수)개씩 그룹화시켜 나간다고 하면, 대부분의 경우 미리 설정된 루프 시작이 프레임 시작과 일치하지 않게 된다. 이 때 루프 시작 샘플 번호를 일정한 양만큼 지나쳐서 새로운 프레임이 시작되었다고 하면, 루프 구간의 끝 프레임에서 그 지나친 양만큼의 샘플들이 다시 반복된다고 보고 마지막 프레임을 설정할 수 있다. 즉, 루프 구간의 파형이 일정한 양만큼 순환 쉬프트 된다고 볼 수 있다. 악기음 샘플에서 루프 구간이 새로운 프레임으로 시작되도록 맞추고 나면, 루프의 시작과 끝이 몇 번째 프레임이고 마지막 프레임의 몇 번째 샘플에서 루프 파형이 끝나는지를 알 수 있다. 이러한 정보들은 부호화된 악기음 샘플의 헤더에 저장시켜 사용한다.

MPEG 오디오 복호화기의 서브밴드 합성 필터에서는 현재 프레임에 대한 PCM 샘플 값들을 계산하기 위해서는 이전 프레임들에서 계산되어 온 중간값들이 필요하다. 따라서 루프 끝 부분에서 다시 루프 시작 부분으로 복호화를 수행하기 위해서는 루프가 시작되는 이전 프레임에 대한 중간값들을 가져와야만 루프 시작 부분을 올바르게 복호화할 수 있다. 이를 위해 MPEG 복호화기는 복호화를 수행하다가 루프 시작 프레임에 도달하게 되면 이전 프레임들에서 계산되어 온 중간값들을 메모리에 저장시켜 놓아야 한다. 이상을 종합하여 루프 구간에 대한 복호화 방법은 다음과 같이 설명될 수 있다. MPEG 복호화기는 부호화된 악기음 샘플의 헤더 정보를 읽어 들여서 각 채널당 할당된 메모리에 저장한다. 프레임 별로 복호화를 수행할 때마다 카운터를 증가시켜 현재의 프레임 번호와 저장된 루프 시작 프레임 번호를 비교하고, 이것이 같아지게 될 때 현재의 중간 계산값들을 메모리에 저장하게 된다. 또한 PCM 룬의 현재의 샘플 어드레스를 메모리에 저장시켜 놓는다. 계속적으로 복호화를 수행하다가 루프 구간 마지막 프레임의 마지막 샘플에 도달하면 다시 루프 시작 구간의 복호화를 위해 미리 저장해 두었던 PCM 룬의 어드레스와 중간값들을 메모리로부터 가져오게 된다. 이렇게 하여 현재 복호화 환경을 처음 루프 구간을 복호화 할 때와 동일한 환경이 되게 한다. 이상의 과정을 그림 4의 흐름도에 나타내었고, 그림 5에는 부호화된 비트 스트림 포맷을 나타내었다.

3.2 타악기 음의 부호화 기법

타악기 음은 루프 구간이 존재하지 않으므로, 전체 파형에 대한 샘플을 다 읽어 오면 진폭값을 강제로 0으로 만들어서 소리가 나지 않게 할 수 있다. 압축 방식이 포함되지 않은 합성 알고리즘에서는 음원 DSP에서 파형의 시작 어드레스와 마지막 어드레스를 파라미터로 전달받아, PCM 룬으로부터 샘플을 읽어 올 때마다 읽어 가는 어드레스와 마지막 어드레스를 비교하여 이것이 같아질 때 진폭값을 0으로 만들게 된다. 압축 방식을 적용할 경

우 음원 DSP는 PCM 램으로부터 단지 복호화된 샘플을 읽어가기만 하면 되므로 PCM 샘플 어드레스에 대한 파라미터는 필요가 없게 된다. 그림 5에 보인 부호화 비트 스트림의 포맷에서 1비트의 decode_bit는 이 문제를 해결하기 위하여 사용한다. 이 비트의 의미는 "0" 일 때 해당 프레임의 복호화가 이루어지는 것이고, "1" 일 때는 복호화가 수행되지 않는 것이다. 보통 부호화된 프레임의 헤더에서 이 바트는 "0"이 된다. 타악기 음의 부호화 시에는 전체 과정의 모든 샘플들이 부호화 된 다음에 새 프레임의 0을 삽입하고, 그 다음으로 decode_bit를 "1"로 하여 여분의 한 프레임을 삽입하여 저장한다. 타악기 음의 경우 전체 음의 복호화가 끝난 후에는 계속되는 새 프레임의 0의 값에 의해 PCM 램의 해당 채널에 모두 0의 값이 쓰여지게 된다. 이렇게 하면 음원 DSP에서 잔폭값을 0으로 만들지 않고도 더 이상 소리가 나지 않게 할 수 있다. 마지막 프레임의 decode_bit는 이 복호화 채널에 대해서는 더 이상 복호화를 수행하지 않는다는 것을 의

미한다. 즉, PCM 램의 타악기 합성 채널에 모두 0이 채워진 후에는 복호화를 정지함으로써 새로운 악기음이 할당되기까지는 0의 값이 계속 읽혀지게 된다.

3.3 대표음의 샘플링 간격과 채널당 필요한 PCM 램의 크기 설정

PCM 합성 방식에서는 샘플된 대표음으로부터 주변의 음을 합성해 내게 된다. 이 때 디지털 신호 처리 기법인 데시메이션과 인터플레이션이 이용된다[5]. 샘플된 음보다 높은 음을 합성하는 경우는 데시메이션으로 메모리에 저장된 샘플을 몇 개의 간격을 두고 읽어오게 된다. 예를 들어 데시메이션 비율이 2라고 하면 메모리에 저장된 샘플을 1개씩 건너 뛰어 읽어오는 것이 되며, 합성되는 음은 샘플된 음보다 한 옥타브 높은 음이 된다. 데시메이션의 상한은 대표음으로부터 몇 배 높은 주파수의 음까지를 합성해 낼 것인지를 결정한다. 샘플된 음보다 낮은 음을 합성하는 경우는 인터플레이션 기법을 이용하며, 메모리에 저장된 샘플을 읽어 오는 속도는 출력 샘플링 주파수(여기서는 44.1Khz)보다 낮다. 압축 방식이 포함되지 않은 PCM 합성 알고리즘에서는 데시메이션 비율의 상한이 대표음을 샘플링 하는 간격에 의해 결정된다고 볼 수 있다. 압축 방식이 포함된 PCM 합성 방식에서는 데시메이션 비율의 상한은 MPEG 복호화기의 속도에 의해 결정되며, 이것에 따라 대표음이 샘플링 되어야 한다. 데시메이션 비율이 D라고 하고, 복호화 채널이 C개이고, 한 샘플이 복호화 되어 PCM 램에 쓰여지는데 K 사이클이 소요된다면, 한 채널을 복호화 하고 다시 그 채널을 복호화 시작할 때까지 DSP가 읽어 간 샘플 데이터 개수는 다음과 같이 표시된다.

$$M = \frac{384 \times K \times (C - 1) \times 44100 \times D}{f_{mpeg}} \quad (1)$$

여기서 384는 MPEG 복호화기가 한 채널에 대해 한 프레임을 복호화 할 때 나오는 샘플 개수이다. 따라서 DSP가 샘플을 읽어갈 때 샘플 데이터가 모자라지 않기 위해서는 M이 384보다 작거나 같으면 된다. 그러므로 식 (1)로부터

$$D \leq \frac{f_{mpeg}}{K \times (C - 1) \times 44100} \quad (2)$$

이고,

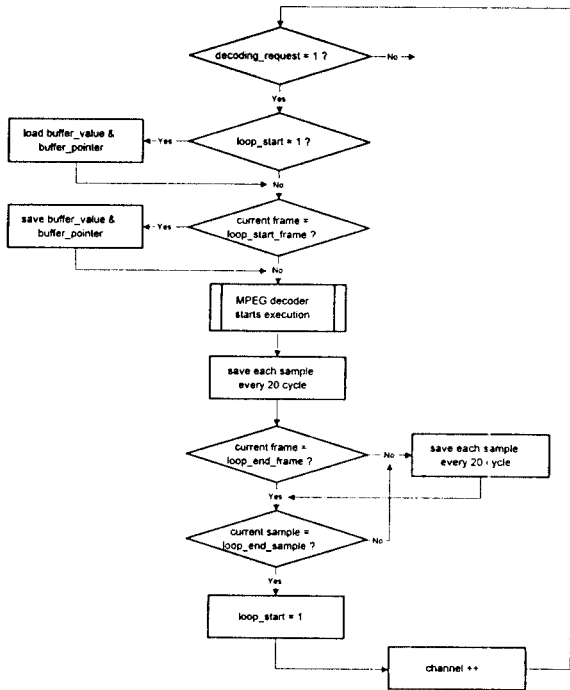


그림 4. 루프 구간의 처리 방법에 대한 흐름도
Fig. 4 Flowchart of loop processing.

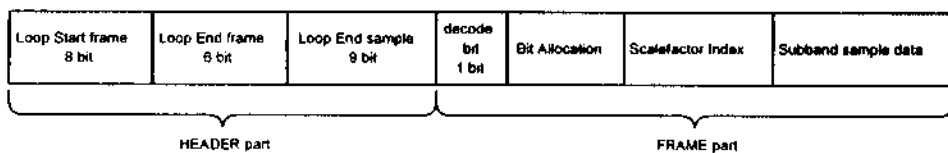


그림 5. 최종 부호화된 비트 스트림 포맷
Fig. 5 Encoded bit stream format.

$$D_{max} = \frac{f_{mpeg}}{K \times (C-1) \times 44100} \quad (3)$$

가 된다. 예를 들어, 이 논문에서 설계된 사운드 합성 시스템에서 MPEG 복호화기의 동작 주파수를 DSP와 같은 33MHz, 샘플 하나를 복호화 해서 PCM 램에 쓰기 위해 필요한 사이클이 20, 채널 개수가 32라고 할 때 D_{max} 는 1.21이 나오게 된다. 이것은 샘플된 대표음으로부터 1.21배의 주파수에 해당하는 음정까지 낼 수 있다는 것이므로 이것을 고려하여 대표음을 샘플링 하여야 한다.

MPEG 복호화기는 복호화 된 샘플 데이터를 PCM 램의 해당 채널 영역에 써넣게 되고, 음원 DSP는 해당 채널의 사운드 합성시에 이 영역으로부터 복호화 된 샘플을 읽어 와서 합성을 하게 된다. 따라서 PCM 램은 일종의 FIFO(First In First Out) 메모리라고 볼 수 있다. MPEG 복호화기가 샘플을 써넣는 속도와 음원 DSP가 이를 읽어 가는 속도가 다르므로 올바른 동작을 위해서는 채널당 필요한 메모리 크기를 적절하게 설정해 주어야 한다. 채널당 메모리 크기는 MPEG 복호화기가 쓰는 속도보다 DSP가 읽어 가는 속도가 느린 경우로부터 유도되어야 한다. 즉, MPEG 복호화기가 DSP가 읽어 가는 속도보다 빨리 데이터를 쓰는 경우에는 DSP가 채 읽어가지 않은 데이터 영역에 새로운 복호화 데이터를 덮어쓰는 현상이 발생할 수 있다. MPEG 복호화기가 한 채널을 복호화하고 다시 그 채널을 복호화 하기 전까지 DSP가 읽어 가는 샘플 개수는 식 (1)과 유사하게 다음과 같이 표시된다.

$$N = \frac{384 \times K \times C \times 44100 \times D}{f_{mpeg}} \quad (4)$$

MPEG 복호화기가 다시 그 채널을 복호화 하려고 할 때 N보다 많은 샘플이 남아 있다면 그 채널을 복호화 할 필요가 없고, N보다 작은 개수의 샘플이 남아 있다면 복호화를 수행해야 한다. 한 번 복호화 할 때 나오는 샘플 개수는 384개이므로 채널당 필요한 최대 램 크기는

$$G = 384 \times N \quad (5)$$

이 된다. $f_{mpeg} = 33\text{MHz}$, $K = 20$, $C = 32$ 에 대해 G는 780이므로 780워드 이상의 램을 한 채널당 메모리 크기로 할당해야 한다. MPEG 복호화기가 채널에 대해 복호화 수행 여부를 결정하는 것은 DSP가 읽어 가는 빈지와 MPEG 복호화기가 그 채널에 대해 가장 최근에 쓴 빈지를 비교함으로써 이루어진다. 즉, 이 빈지수 차이가 N보다 작은 경우에는 복호화를 수행하고, N보다 큰 경우에는 복호화를 수행하지 않게 된다.

3.4 외부 램의 영역 구분

그림 6은 MPEG 오디오 복호화기에 연결된 외부 램을 나타낸 것이다. 세 개로 나누어져 있는 메모리 영역은 실제로 하나의 메모리 칩으로 구현되는 것이다.

PCM 데이터 영역은 복호화된 샘플을 저장하고 DSP가 읽어 가는 메모리 영역으로서 MPEG 복호화기와 음원 DSP가 서로 공유하는 부분이다. 이 영역의 메모리 공유 방법에 대해서는 다음 절에서 설명하도록 한다.

또 하나의 영역은 비퍼 영역이라고 표시되어 있는 부분인데 이것은 앞에서 설명한 바와 같이 MPEG 복호화시 서브밴드 합성 필터에서는 이전 프레임에서 계산된 중간값들이 사용되므로 채널 변환시 현재 채널에 대한 이러한 값들을 저장해 놓기 위한 메모리 영역으로 쓰이게 된다. 이 영역에 저장된 데이터는 복호화 수행시 매 사이클마다 액세스 되어야 하는데, 외부의 램 영역으로부터 직접 읽기/쓰기가 일어나면 이에 대한 타이밍 제한상 MPEG 복호화기의 동작 속도가 제한되고, PCM 데이터를 저장하거나 DSP가 데이터를 읽어 올 시간적 여유가 충분하지 않게 될 수도 있다. 이에 따라 MPEG 복호화기 내부에 512 워드 크기의 두 개의 캐쉬 램을 사용한다. 만약 MPEG 복호화기가 n번째 채널에서 BUF1을 액세스하고 있다면, 이와 동시에 미리 PCM 램의 n+1번째 채널에 해당하는 비퍼 영역에서 중간값을 BUF2로 읽어와서 저장해 놓는다. 그러면 n+1번째 채널에서는 BUF2에 있는 값을 액세스 하면서 복호화를 수행하고, 그 동안 n번째 채널에 해당하는 중간값들이 BUF1에서 외부 램의 n번째 채널의 비퍼 영역으로 저장되고, n+2번째 채널에 해당하는 중간값이 읽혀져 BUF1에 저장된다. 이와 같은 방식을 사용하면 현재 복호화에 필요한 중간값들을 액세스 하기 위해 외부 램을 바로 액세스 하지 않으므로 메모리 공유 문제를 해결하기도 쉽고, 내부 캐쉬 램을 사용함으로써 훨씬 속도를 빠르게 할 수 있는 이점이 있다.

외부 램에 구현된 또 하나의 메모리 영역은 3.1절에서 설명된 두프 구간의 복호화시 두프 시작 프레임의 복호화에 필요한 중간값들을 저장하는데 사용되는 영역이다.

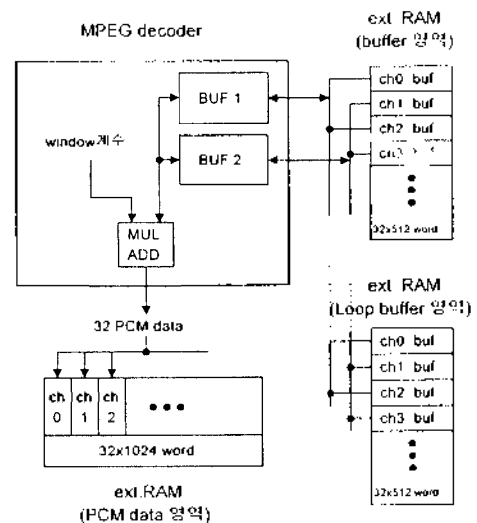


그림 6. 외부 메모리의 영역 분할
Fig. 6 External memory structure.

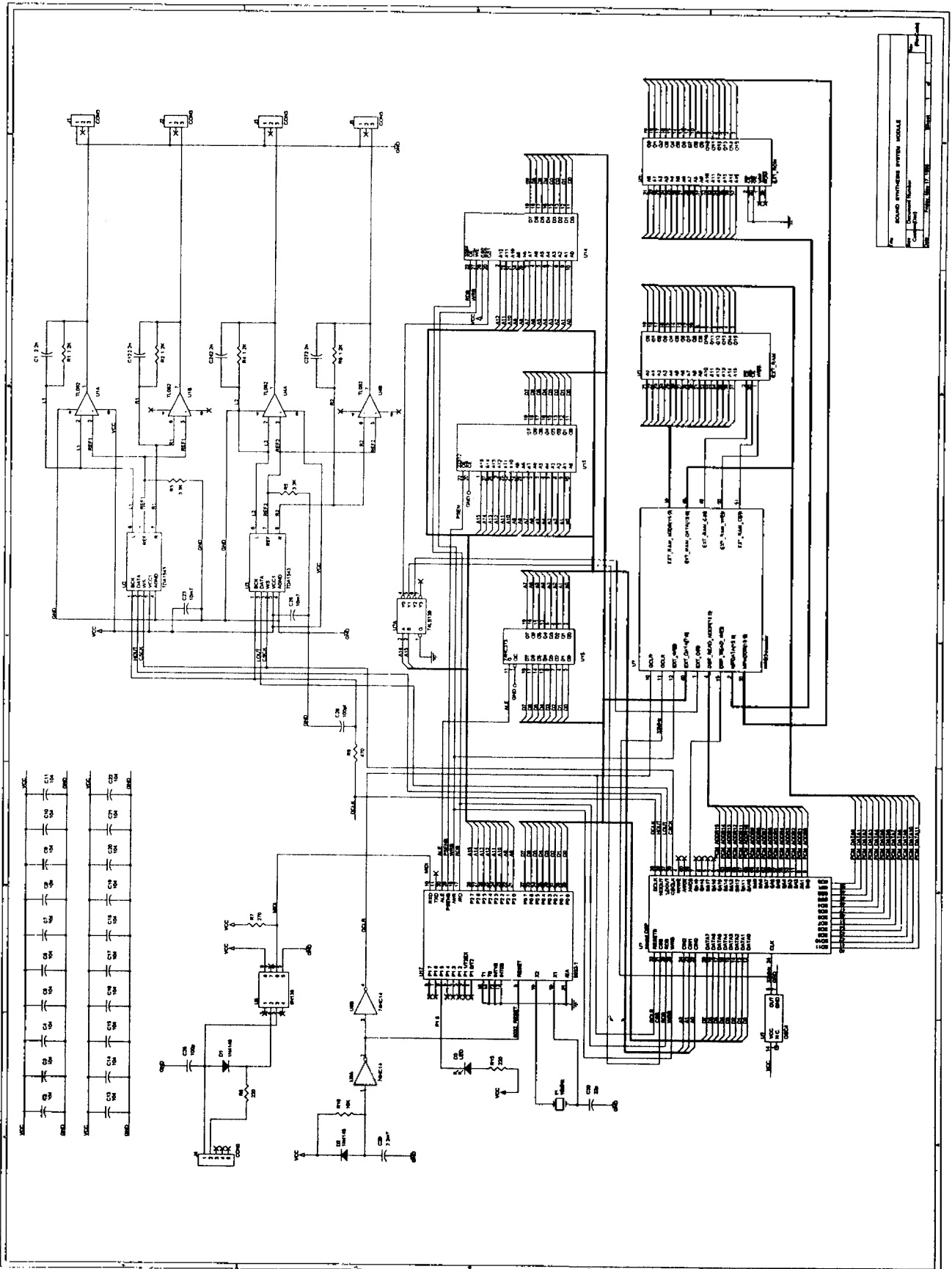


그림 7. 설계된 사운드 합성 시스템의 회로도
 Fig. 7 Schematic of sound synthesis system.

여기에 저장된 데이터는 루프 시작 구간의 복호화가 요구될 경우 해당 채널 영역에서 MPEG 복호화기의 BUF1 혹은 BUF2로 읽혀져 저장된다.

3.5 PCM 램 영역의 공유 방법

MPEG 오디오 복호화기에 연결된 외부 램은 앞 절에서 설명한 바와 같이 세 개의 영역으로 구분되어 사용된다. MPEG 복호화기는 이 세 가지 영역을 모두 액세스하고, PCM 데이터 영역은 MPEG 복호화기와 음원 DSP가 함께 액세스 하는 영역이다. 따라서 두 프로세서가 하나의 램을 공유하기 위해서는 우선 순위를 정해두고 정확한 타이밍에 따라 동작하도록 해야 한다. 합리적인 우선 순위는 다음과 같이 정해진다고 볼 수 있다. 가장 높은 우선 순위는 DSP가 PCM 데이터를 읽어 가는 것이고, 그 다음은 MPEG 복호화기가 복호화된 데이터를 써넣는 것, 그리고 다음이 복호화에 필요한 중간값들의 저장과 읽기가 된다. 외부 램에 대한 읽기/쓰기는 시스템 클럭의 두 사이클을 기본으로 해서 이루어지므로, 두 사이클마다 램에 대한 액세스 여부를 샘플링 해서 우선 순위에 따라 제어 신호가 발생된다.

IV. VHDL을 이용한 사운드 합성 시스템의 설계 및 시뮬레이션

III장에서는 C 언어로 표현된 시뮬레이터를 이용하여 MPEG 오디오 복호화기를 적용한 사운드 합성 시스템에서 발생하는 여러 가지 문제점에 대해 살펴보고 이것을 해결하는 방안을 제시하였다. 음원 DSP와 MPEG 복호화기는 이러한 모든 점들을 고려하여 수정, 보완되어 VHDL로 설계되었다. 그림 7은 설계된 사운드 합성 시스템의 회로도도를 나타내고 있다. 미디 신호 입력부와 함께 마이크로프로세서, 시스템 프로그램 롬, 음원 DSP, MPEG 복호화기, PCM 롬과 램이 나타나 있고, 음원 DSP의 출력을 받아 최종적으로 아날로그 신호로 변환시켜 내보내는 D/A 컨버터와 저역통과 필터 등이 나타나 있다. VHDL로 설계되어 검증된 부분은 미디 입력 부분과 마이크로프로세서, 그리고 D/A 컨버터를 제외한 부분이다. C로 된 시뮬레이터에서와 마찬가지로 마이크로프로세서의 동작은 DSP와 MPEG 복호화기의 입력 파라미터와 제어 신호들만으로 충분히 표현할 수 있으므로, 이것을 VHDL로 설계된 사운드 합성 시스템의 입력으로 하여 시뮬레이션을 수행하였다. 사용한 툴은 SYNOPSIS의 VHDL 시뮬레이터인 VSS(VHDL System Simulation)이다.

검증은 다음과 같이 이루어졌다. 먼저 C로 된 시뮬레이터 자체에 대한 검증을 위해서 음원 DSP만으로 합성한 출력 파형과 압축 방식을 적용한 시스템 시뮬레이터의 시뮬레이션 결과 파형을 비교하여 시뮬레이터의 올바른 동작을 확인하였다. 다음으로 시뮬레이터 결과 파형과 VHDL로 설계된 시스템의 시뮬레이션 결과 파형을

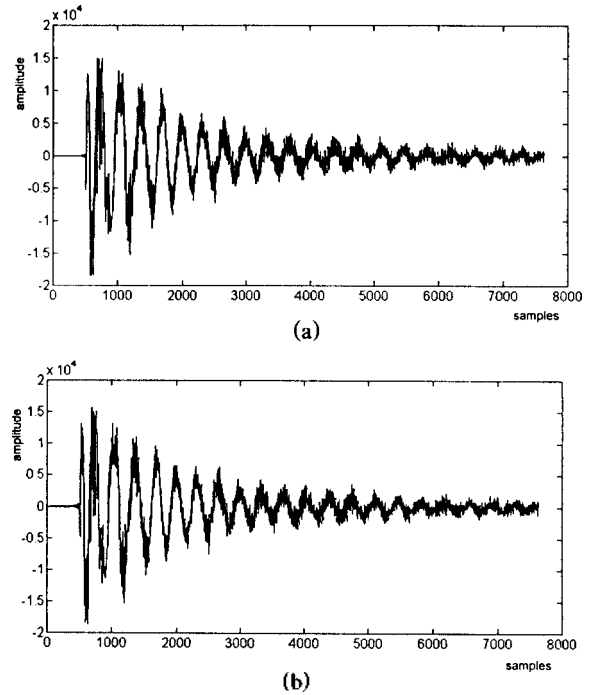


그림 8. 타악기(Snare drum)에 대한 시뮬레이션 결과
(a) 시뮬레이터 결과 (b) 하드웨어 시뮬레이션 결과
Fig. 8 Simulation waveform of snare drum :
(a) Output waveform of software simulator,
(b) Output waveform of hardware simulation.

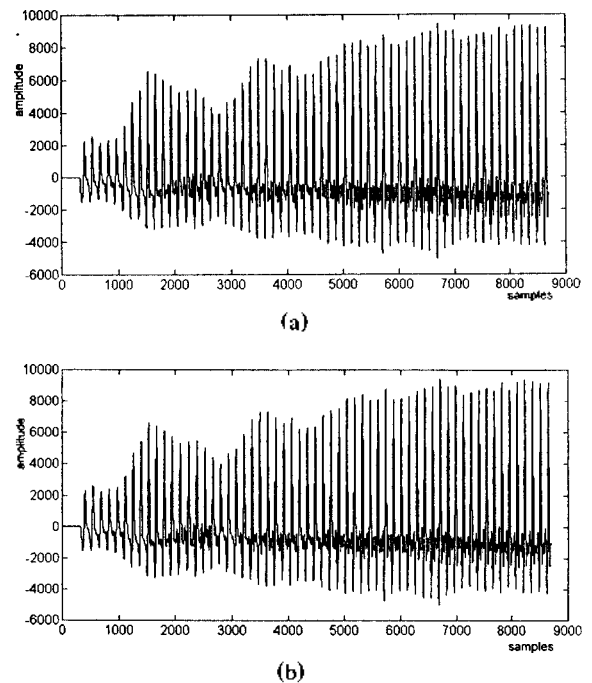


그림 9. 금관악기(Horn)에 대한 시뮬레이션 결과
(a) 시뮬레이터 결과 (b) 하드웨어 시뮬레이션 결과
Fig. 9 Simulation waveform of horn :
(a) Output waveform of software simulator,
(b) Output waveform of hardware simulation.

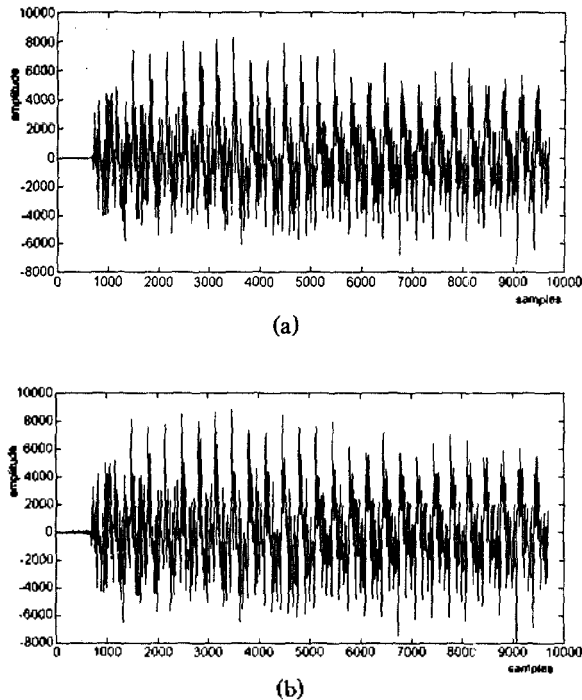


그림 10. 피아노에 대한 시뮬레이션 결과
 (a) 시뮬레이터 결과 (b) 하드웨어 시뮬레이션 결과
 Fig. 10 Simulation waveform of piano ;
 (a) Output waveform of software simulator,
 (b) Output waveform of hardware simulation.

비교함으로써 최종적으로 압축 방식을 적용한 사운드 합성 시스템의 하드웨어 동작을 검증하였다. 그림 8, 9, 10에 각각 타악기, 금관악기, 건반악기 등 세 가지 종류의 악기 음에 대한 C 시뮬레이션 결과 파형과 VHDL 시뮬레이션 결과 파형을 나타내었다. 그림에서 보듯이 세 가지 종류의 악기에 대한 결과 파형이 거의 동일함을 알 수 있다. 파형에서 약간씩 차이가 나는 것은 VHDL 시뮬레이션에서는 MPEG 복호화기가 게이트 레벨에서 합성 가능한 하드웨어 수준에서 설계되어 고정 소수점 연산을 수행하기 때문이다.

V. 결 론

본 논문은 고음질의 사운드 합성을 위하여 MPEG 오디오 압축 기법을 적용한 사운드 합성 시스템의 설계에 대해서 논하였다. MPEG 오디오 압축 기법을 적용하여 사운드 샘플을 압축시켜 본 결과 평균적으로 8:1 정도의 메모리 감소 효과를 얻을 수 있음을 발견하였다. 이것은 동일한 메모리 용량에 대해서는 8배 정도 더 많은 음을 저장할 수 있다는 것이므로 고음질의 사운드를 합성해 낼 수 있게 된다. 사운드 합성 시스템을 이루는 기능 요소들은 독립적으로 설계되어 FPGA와 같은 하드웨어를 통하여 그 동작을 검증하였으며, 이것을 합성 시스템에 적

용하기 위해 먼저 각 기능 요소의 동작을 C 언어로 모델링 하여 전체적인 시스템 레벨에서의 시뮬레이터를 작성해 미리 동작을 살펴봄으로써, 여러 가지 문제점을 발견하고 해결 방법을 제시하였다. 시뮬레이터를 통해 충분히 검증된 기능 요소들은 VHDL로 수정, 보완되어 최종적으로 시스템 레벨의 하드웨어 시뮬레이터를 완성하였다. 이것으로써 VHDL 시뮬레이션 결과를 통하여 사운드 합성 시스템이 정상적으로 동작함을 검증하였다. 실제로 MPEG 오디오 압축 기법을 이용해 사운드 합성 시스템을 설계하는 경우 시스템 프로그램과 특정 DSP의 사용에 따라 여러 가지 형태로 구현이 가능하다. 그러나 결국 압축 기법이 들어가게 되면 복호화기의 동작 속도가 DSP의 동시 합성수에 큰 영향을 미치게 되므로 복호화 알고리즘의 설계와 최적화된 하드웨어 구현에 더 많은 연구가 이루어져야 할 것이다.

참 고 문 헌

1. Karlheinz Brandenburg, "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.
2. Davis Pan, "A Tutorial on MPEG/Audio Compression," IEEE Multimedia, pp. 60-74, 1995.
3. 박주성, 장호근 외, "스테레오 사운드 카드 개발," 정보통신부 보고서, 1997.
4. 김태훈, "MPEG 오디오 복호화기에 관한 연구," 부산대학교 전자공학과 석사학위논문, 1997.
5. Curtis Roads, "The Computer Music Tutorial," pp. 117-133, MIT press, 1996.
6. Nikitas Alexandridis, "Design of microprocessor-based systems," Prentice Hall, 1993.
7. Steve Carlson, "Introduction to HDL-based design using VHDL," Synopsys, 1990.
8. "VHDL Compiler Reference Manual," Synopsys, 1996.

▲장 호 근(Ho Keun Jang) 1968년 1월 15일생
 1993년 2월: 부산대학교 전자공학과(공학사)
 1995년 2월: 부산대학교 전자공학과(공학석사)
 1995년 3월~현재: 부산대학교 전자공학과 대학원 박사과정
 ※주관심분야: DSP 설계, 사운드 합성

▲김 태 훈(Tae Hoon Kim) 1972년 4월 25일생
 1995년 2월: 부산대학교 전자공학과(공학사)
 1997년 2월: 부산대학교 전자공학과(공학석사)
 1997년 3월~현재: 부산대학교 전자공학과 대학원 박사과정
 ※주관심분야: 오디오 압축, DSP 설계

▲곽 증 태(Jong Tae Kwak) 1972년 11월 5일생
 1996년 2월: 부산대학교 전자공학과(공학사)
 1998년 2월: 부산대학교 전자공학과(공학석사)
 1998년 3월~현재: 현대전자
 ※주관심분야: 마이크로프로세서 설계, DSP 설계

▲박 주 성(Ju Sung Park) 1953년 12월 19일생
 1976년 2월: 부산대학교 전자공학과(공학사)
 1978년 2월: 한국과학기술원 전기 및 전자공학과(공학석사)
 1978년 3월~1985년 7월: 한국전자기술연구소
 1985년 8월~1989년 7월: University of Florida, Ph.D.
 1989년 8월~1991년 3월: 한국전자통신연구소 책임연구원
 1991년 3월~현재: 부산대학교 전자공학과 부교수
 ※주관심분야: DSP 설계 및 응용, 사운드 합성, 반도체 소
 자 모델링