

MPEG 계층 III 오디오 복호기 실시간 구현에 관한 연구

김수현, 김진호, 이창원, 김현중, 차형태
승실대학교 전자공학과
전화 : (02) 826-9063 / 팩스 : (02) 821-7653

Implementation of the Real-time MPEG Layer III Audio Decoder

Soo-hyun Kim, Jin-ho Kim, Chang-won Lee, Hun-joong Kim, Hyung-tai Cha
Dept. of Electronic Engineering, Soongsil University
E-mail : shkim@mmslab.soongsil.ac.kr

Abstract

In this paper, we propose a real-time implementation of the MPEG-1 layer III and MPEG-2 layer III LSF audio decoding system based on OAK DSP Core. In order to solve the problem of resolution, the system has been used floating-point operation and double precision in dequantization module. The size of ROM is reduced by using the Run-length algorithm of reordered index. The subband synthesis filter module is optimized to have low computational complexity in terms of the size of ROM or RAM. To construct a efficient system, we used both the DSP Core and Parser-Huffman decoder which is implemented with VHDL.

I. 서론

현대 기술의 급속한 발전은 대용량의 디지털 정보의 처리와 전송을 가능하게 하였고 이에 따라 급속하게 이전의 아날로그 기술을 대체해 나가고 있다. MPEG(Moving Picture Export Group) 오디오, 비디오 규격은 디지털 방송에 적합하도록 만들어 졌으며 DBS, HDTV, DAB, Multimedia등에 적용이 되고 있다. 최대 1.5 Mbps이하의 비디오 전송률과 2 채널, 최대 384 Kbps이하의 오디오 전송률을 가진 MPEG-1

규격[1]은 1992년에 MPEG 회의에서 압축과 신장 그리고 전송을 위한 표준 방식이 결정되었다. 5.1 채널과 최대 256 Kbps이하의 오디오 전송률을 가진 MPEG-2 규격[2]은 낮은 Sampling 주파수를 사용하며 1994년에 MPEG 회의에서 결정되었다. 한국에서도 DBS 시험 방송이 1996년에 시작되었고 수신기 개발을 위해 Chip의 제작을 위한 자체 기술 개발에 노력하고 있으며 이 규격을 지원하는 응용이 필수적이라 하겠다[3][4].

OAK DSP Core는 16-bit 고정 소수점 연산, 16-bit 데이터와 프로그램 버스, 16×16 비트의 곱셈 연산과 36 비트의 barrel shifter를 지니며 통신과 전자 응용 분야에 좋은 성능을 지니도록 설계되어 있으며, 40 MIPS의 계산 능력을 갖고 있다.

II. MPEG 오디오 복호기의 개요

MPEG 오디오 부호기에서 만들어진 비트열은 그림 1에서와 같이 크게 4가지 부분으로 나눌 수 있으며 다음과 같다. 첫째, Header부분은 동기신호, 비트열의 압축률, 압축 방법, Sampling 주파수, Stereo 또는 Mono 방송의 여부 등에 대한 기본적인 정보를 가지고 있으며 Error에 가장 민감한 부분이다. 둘째, CRC(Cyclic Redundancy Code)는 비트열의 주요 부분에 Error의 발생 여부를 검사하는 부분이다. 셋째로 부가정보(Side Info.)부분은 block type, block 이득, Huffman 부호화 등에 관련된 부가 정보를 가지고 있다. 넷째로

Audio Data 부분은 실제 부호화된 오디오 신호에 대한 정보를 가지고 있으며 크기 정보(scalefactor)와 허프만(Huffman) 부호화된 Sample들로 이루어져 있다. 이 정보들을 바탕으로 허프만 복호화, 역양자화, 재배열(Reorder)과 Anti-aliasing, IMDCT (Inverse Modified Discrete Cosine Tramsform), 합성 필터를 거쳐 PCM신호로 복호화된다. 마지막으로 Ancillary 부분은 기타 정보를 전송하고자 할 때 사용되는 부분이다.

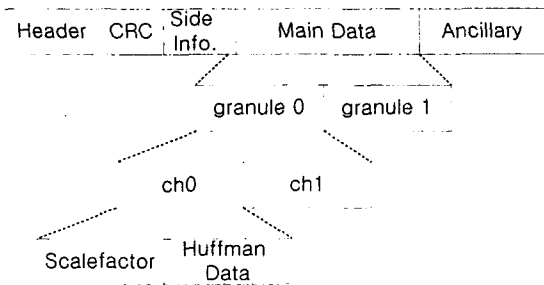


그림 1 비트열의 구조
Fig. 1 Bitstream architecture

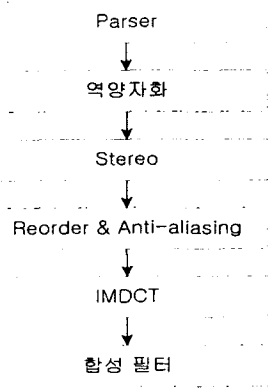


그림 2 MPEG 계층 III 복호화기
Fig. 2 MPEG layer III decoder

이 비트열이 복호기에 입력되고 동기신호가 검출되면 그림 2와 같이 Parser부에서 동기 검출된 비트열로부터 Header 정보, 부가 정보와 Main Data등을 각 부분별로 정보들을 분류하여 저장한다. MPEG 오디오 비트열에는 각 Frame의 시작을 표시하는 동기 신호가 들어있으며 12개의 '1'로 구성되어 있다. 동기 신호는 Sampling 주파수가 고정되었을 때 각 Frame간의 간격이 일정하다는 사실을 이용하여 검출될 수 있다.

Parser부에서 Header 정보는 비트열의 특성에 관한 자세한 기록, 즉, Sampling 주파수, Bitrate, Stereo mode등의 정보를 가지고 있다. Header 정보에 들어

있는 ID 비트에 의해 MPEG-1('1') 비트열과 MPEG-2('0') 비트열이 구분된다. 부가 정보에는 MPEG-1은 두 개의 granule로, MPEG-2 LSF(Low Sampling Frequency)는 한 개의 granule로 구성되며 각 granule은 두 개의 채널 정보를 가질 수 있다. 역양자화부에서 한 개의 granule에 대해 채널 단위로 허프만 부호화된 Sample들의 복호화와 크기 정보와 부가 정보를 이용하여 원래의 크기로 Sample들을 역양자화한다. 한 개의 granule에 대해 이 Sample들은 stereo에서 해당되는 경우에만 Joint-stereo 처리를 하게 된다. 한 개의 granule에 대해 채널단위로 재배열 및 Anti-aliasing부와 IMDCT부 처리를 통하여 합성 필터부에서 합성되고 PCM신호로 재생되어 출력하게 된다.

III. OAK DSP Core를 이용한 MPEG 오디오 구현

MPEG 계층 III는 허프만 부호를 사용하기 때문에 Main Data의 크기가 일정하지 않다. 이 문제 해결을 위해 이전 Frame의 비트열에 현재 Frame의 비트열의 Main Data를 일부 넣을 수 있도록 구성되어 있다. 따라서 한 Frame의 비트열의 복호화 후 다음 Frame의 비트열을 분석하여 이전 Frame에서 필요한 정보를 얻게 되므로 3 Frame 정도의 Main Data들을 보관해야 한다. 본 논문에서는 이 부분을 별도의 모듈로 설계했으며 3 Frame에 대한 포인터들을 이용하여 Main Data의 다른 저장 없이 바로 원래의 비트열에서 가져오도록 구현하였다.

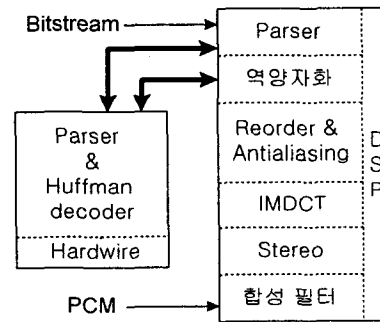


그림 3 시스템 구성
Fig. 3 System architecture

그림 3과 같이 일부 Parser부와 허프만 복호화를 위한 모듈을 VHDL을 이용하여 설계하였다. OAK DSP Core는 Parser부에 대한 특별한 명령을 제공하지 않으므로 DSP에서는 포인터 및 가져올 비트를 전송하여

Parser/허프만 모듈이 해당 위치의 비트열에서 요구된 비트만큼의 데이터를 읽어 DSP에 전송하도록 설계하였다.

역양자화부는 다음과 같이 효율적으로 구현하였다. 허프만 복호화 값과 global 이득과 scalefactor를 이용하여 역양자화를 하게 된다.

$$\begin{aligned}
 xr_i &= \text{sign}(is_i) \times |is_i|^{\frac{4}{3}} \times 2^{\frac{1}{4}(gg-210)} \\
 xr_i & * = 2^{-\frac{1}{4}(8 \times sg)} \times 2^{-(sm \times scf)}, \quad \text{short block} \quad (1) \\
 xr_i & * = 2^{-(sm \times (scf + pft))}, \quad \text{long block} \\
 sm &: \text{scalefactor multiplier} \\
 pft &: \text{preflag} \times \text{pretab}
 \end{aligned}$$

허프만 복호화 값(is), global 이득(gg), subblock 이득(sg)과 scalefactor(scf)가 식 (1)과 같은 연산을 하게 되므로 높은 resolution일 필요하게 된다. 고정 소수점 OAK DSP Core에서는 지수연산을 지원하지 않으므로 미리 계산된 결과 값을 테이블화하여 색인(index)을 이용하는 방식(table look-up)을 사용한다. resolution 문제를 해결하기 위해서 역양자화부를 부동소수점 방식으로 연산하게 하였으며 double precision을 통하여 24 비트의 resolution을 가지도록 구현하였다.

허프만 복호화 값은 13비트의 색인을 가질 수 있으나 13비트 모두 표로 만들 수가 없기 때문에 16비트로 표현할 수 있는 범위의 2436개만을 사용하였다. 허프만 부호화에서 많은 비트를 할당하는 경우 허프만 부호화의 효율성이 떨어지는 경우이며 빈번히 발생하는 경우가 아니다. 따라서 본 논문에서는 다른 색인에 대하여 2436으로 제한을 가하여 사용하도록 구현하였다. 허프만 복호기는 입력된 비트열을 사전에 정해진 단위 길이로 만들고 ROM의 테이블을 이용하여 비교하여 값을 DSP에 되돌려 준다. 이때 소요되는 클럭은 1 ~ 6 클럭이 소요된다.

재배열부에서는 다음과 같은 방법을 통하여 ROM의 소요를 줄일 수 있었다. 재배열은 3가지의 색인 연산을 수행하며 각 576 Sample들에 대한 색인을 살펴보면 일정한 규칙이 없이 변화나 색인들 사이에 분산이 낮고, 일부는 규칙적이다. 이 특성을 이용하여 색인들의 차등을 Run-length를 적용하여 117개의 표로 ROM을 절약하여 사용할 수 있다.

합성 필터부는 다음과 같은 알고리즘의 최적화를 사용하여 속도 및 ROM과 RAM의 효율적인 사용을 하였다. 합성 필터는 다음의 식 (2)에 의해 처리된다.

$$Y[i] = \cos[(16+i)(2k+1)\frac{\pi}{64}] \times S_k \quad (2)$$

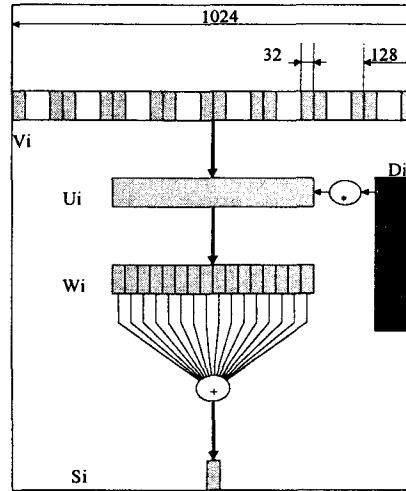


그림 4 Subband 합성 필터
Fig. 4 Subband synthesis filter

식(2)의 방정식을 이용하여 32개의 입력 S_k 로부터 64개의 $Y[i]$ 를 구하여 사용해야 하므로 Stereo mode를 기준으로 한 Frame당 2304개의 Sample을 처리해야 하므로 오디오 복호기 실시간 구현에서 합성필터 블록이 가장 많은 연산량을 차지하였다. 이러한 연산량을 줄이기 위해 cosine 함수의 대칭성을 이용하여 32개의 $Y[i]$ 만으로 64개의 $Y[i]$ 를 구하도록 하였다. 즉,

$$\begin{aligned}
 Y[i] &= -Y[32-i] \quad i=0,1,2,\dots,15 \\
 Y[i+32] &= Y[64-i] \quad i=1,2,3,\dots,15 \\
 Y[16] &= 0 \\
 Y[48] &= -\sum S_k \quad k=0,1,2,\dots,31
 \end{aligned} \quad (3)$$

이렇게 구하여진 $Y[i]$ 는 64개씩 Shifting되는 V_i 로 대입된다.

$$\begin{aligned}
 V[i] &= V[i-64] \quad i=1023,\dots,65,64 \\
 V[i] &= Y[i] \quad i=63,\dots,2,1,0
 \end{aligned} \quad (4)$$

V_i 는 그림 4에서와 같이 1024개의 Word로 이루어져 있으나 Cosine 함수의 대칭성을 이용하여 512개의 Word만 저장하고 이로부터 1024개의 계수들을 구할 수 있다. U_i 는 V_i 를 8개의 블록으로 만들어 1블록의 128개에서 앞 뒤 32개씩을 취하여 만들어진다. 이 U_i 에 Window 계수 D_i 를 곱하여 512개의 W_i 를 만들고, W_i 를 32개의 계수를 가진 16개의 블록으로 만들어 각 블록을 더하여 32개의 S_i 를 만든다. 또 32개의 PCM값인 S_i 를 구할 때 U_i 와 W_i 와 같은 RAM을 거치지 않고 직접 계산함으로써 불필요한 RAM 부분을 줄일 수

있다. Vi부터 Si까지는 한 경로를 사용하여 계산함으로써 설계의 효율성을 기할 수 있고, 불필요한 출력부분의 시간 지연을 최소화 할 수 있다.

$$\begin{aligned}
 S_0 &= \sum_{k=0}^7 [V_{64k} D_{64k} - V_{64k+32} D_{64k+32}] \\
 S_i &= \sum_{k=0}^7 [V_{64k+i} D_{64k+i} + V_{64k+i+47} D_{64k+i+32}] \\
 &\quad i = 1, 2, \dots, 15 \\
 S_{16} &= \sum_{k=0}^7 [V_{64k+63} D_{64k+48}] \\
 S_i &= \sum_{k=0}^7 [V_{64k+79-i} D_{64k+32+i} - V_{64k+32-i} D_{64k+i}] \\
 &\quad i = 17, 18, \dots, 31
 \end{aligned} \tag{5}$$

이때 Vi와 Si의 관계는 주어진 식 (5)로부터 직접 구할 수 있다.

IV. 실험 결과

본 논문에서는 MPEG 계층 III 복호기에 대한 검증으로서 어셈블러 프로그램의 성능을 OAK DSP core의 시뮬레이션을 통해 측정된 결과 약 28 MIPS의 연산량을 필요로 하였다. 이 알고리즘이 갖는 프로그램 ROM의 크기는 약 2.6 K Word이며, 약 12 K Word의 데이터 ROM과 약 6 K Word의 데이터 RAM으로 구성되어 있다. Parser/허프만 복호화 모듈은 VHDL로 설계하여 시뮬레이터에서 정상 동작함을 확인하였다. 역양자화부에서 Double precision을 사용함으로써 ROM을 두 배로 사용되는 문제점을 가지고 있으나 이는 음질과의 상호보상 관계를 가지며 24 비트의 정확도를 확보할 수 있었다.

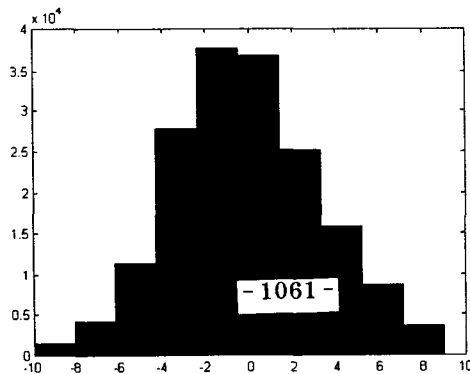


그림 4 모의 실험 결과
Fig. 4 Experimental results

C언어로 구현된 결과와 본 시스템의 시뮬레이션 결과를 비교하면 그림 4와 같이 최대 5 비트의 에러를 가지며 rms 값으로 약 2 비트의 에러율을 보인다. 실험 입력은 Fraunhofer IIS의 Layer III test bitstream package v2.2 중 he_48khz.bit를 사용하였다[5].

V. 결론

본 논문에서는 OAK DSP Core를 이용하여 MPEG-1 계층 III와 MPEG-2 계층 III LSF 오디오 복호기를 구현하였다. cosine 함수의 대칭성과 하나의 경로 사용으로 합성 필터부의 알고리즘을 효율적으로 적용함으로써 RAM과 ROM의 크기 그리고 각 부분의 연산에 필요한 계산량을 최적화하였다. 효율적인 시스템의 구성을 위해 Parser부의 일부 기능과 허프만 복호기를 VHDL로 구현하였고 나머지 모듈은 DSP로 구현되었다. 본 논문에서 구현된 시스템은 약 28 MIPS의 속도와 5 비트이하의 에러를 유지할 수 있었다. 속도와 에러율은 OAK DSP Core가 가지고 있는 한계에 의한 것으로 16 비트를 사용하는 연산과 저장 방식에 기인하며 이를 보정하기 위하여 특정 부분에 Double precision의 연산을 수행하여 복호화 속도 저하 및 ROM의 크기가 커지게 되었다. 앞으로는 알고리즘의 연구를 통하여 더욱 빠른 속도를 제공하는 알고리즘에 대한 연구가 필요하다고 생각된다.

참고문헌

- [1] ISO/IEC 11172-3, "Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to 1.5 Mbit/s Audio Part," International Standard, Nov., 1992.
- [2] ISO/IEC 13818-3, "Generic Coding of Moving Pictures and Associated Audio-Audio Part," International Standard, Nov., 1994.
- [3] Mark, K. and Karlheinz, B., Applications of Digital Signal Processing to Audio and Acoustics. Kluwer Academic Pub., 1998.
- [4] 차형태, "MPEG 오디오 합성 필터의 구현을 위한 연구," 제 13회 음성통신 및 신호처리 워크샵 논문집, pp. 79-84, 8월 16일, 1996.
- [5] ftp://ftp.tnt.uni-hannover.de