

## DVD 시스템에 있어서 DOLBY AC-3 디코더의 실시간 구현에 관한 연구

### A Study of Real-Time Implementation of Dolby AC-3 Decoder in a DVD System

이 원 우\*, 김 승 호\*, 장 성 철\*, 이 회 수\*, 허 재 훈\*  
(Won Woo Lee\*, Sung Ho Kim\*, Sung Chul Jang\*, Hee Soo Lee\*, Jae Hun Heo\*)

#### 요 약

본 연구에서는 20bit 고정 소숫점 범용 DSP 칩을 이용하여 Dolby AC-3 디코더를 실시간으로 구현하였으며, PC를 이용하여 구현한 AC-3 C-simulator와의 성능비교를 통하여 본 연구의 실시간 디코더가 성능이 뒤지지 않음을 보였다. 또한 향후 DVD 재생기에 있어서 본 연구의 AC-3 디코더가 어떻게 설계되었는지에 대해서도 논하였다. 특히 본 연구에서 제안하는 DVD AC-3 디코더는 최소한의 지연시간을 갖도록하여 기존의 디코더보다 향상된 성능을 가짐을 보였다.

#### ABSTRACT

A real-time Dolby AC-3 decoder has been implemented using a 20-bit fixed point general purpose DSP chip. It is shown that AC-3 decoder of this paper has same performance as decoder of C-simulator on PC. And also, it is applied to DVD player. Especially, in this paper, we proved that it has shorter latency time than that of previous AC-3 decoder.

#### I. 서 론

과거 몇년동안 디지털 오디오 데이터 압축은 오디오 산업에 있어서 중요한 기술중의 하나로 자리잡게 되었다. 한편, 종래의 기법을 사용함으로써 요구되는 높은 데이터 전송률이 필요없어도 고품질 오디오 재생이 가능한 새로운 포맷이 등장하였다[1-4].

AC-3 부호화 기법은 돌비연구소에서 개발한 저비트레이트, 고품질의 다채널 오디오의 재생이 가능한 오디오 압축 기법이다[5]. AC-3 비트스트림(bitstream)의 규격은 다양한 샘플링레이트(sampling rate : 48kHz, 44.1kHz, 32kHz)를 지원하며 32kbps에서 640kbps까지의 데이터 전송률을 지원한다. 또한 미국 HDTV의 사운드 표준으로 채택되었고, 최근에는 LD(Laser Disc)에도 적용되어 가정에서 멀티사운드를 즐기게 되었으며, 특히 DVD(Digital Video Disc)의 사운드 포맷으로 확정되는 등 차세대 디지털 오디오 포맷으로의 응용이 확대되고 있다. 따라서 본 논문에서는 점차 응용영역이 확대되고 있는 다채널 오디오 압축기법중의 하나인 돌비 AC-3 디코딩 알고리즘의 실시간 구현에 대해 논하고 특히 DVD

시스템에 적용한 설계기법에 대해서 다루어보고자 한다.

#### II. AC-3 디코딩 알고리즘 구현

AC-3 디코딩의 프로세스의 기본 개요는 그림 1에 보인다. 그림 1에서 보는바와 같이 디코딩 알고리즘은 여러 단계로 나누어져 있으나 이것은 좀더 크게 나누면 인코딩된 데이터를 unpack하는 데이터 unpack부와 unpack된 데이터를 역 MDCT(Modified Discrete Cosine Transform)와 윈도우 처리, 그리고 overlap-add를 통한 PCM 데이터 출력부의 2 단계로 나눌수 있다[6]. 본 연구에서는 편의상 전자를 입력 프로세싱이라고 하며 후자를 디코딩 출력 프로세싱이라 하였다.

##### 2.1 디코딩 입력 프로세싱

AC-3 디코더의 unpack부는 크게 동기화/에러 검출, BSI/side information 추출, 지수 디코딩, 비트 할당 계산, 가수 디코딩(unpack, ungroup, dequantizer, dither), decoupling rematrixing, dynamic range compression 등의 단계로 구분된다.

AC-3의 동기화/에러 검출은 bitstream 선두에서 sync

\*삼성전자 기술총괄 DVD개발센터  
DVD R&D Center, Corporate Technical operatims  
접수일자:1995년 11월 24일

word인 16진수 0xB77을 찾는 것으로 시작한다. 이 sync word가 발견되어야 그 이후를 AC-3 bitstream으로 간주할 수 있다. Sync word를 찾은 후에는 먼저 synchronization information을 추출한다. Sync.Info.인 fscod (Sampling frequency code)와 frmsizecod(frame size code)를 추출하면 프레임의 길이를 계산하여 프레임 전체의 CRC를 계산한다. 이 값은 프레임의 전송도중에 bit error등이 발생했는지를 알려준다. 한편 AC-3의 실시간 제약조건 때에 전체의 CRC를 프레임 시작에서 계산하면 그 프레임의 block 0에서 실시간 제약조건을 만족하지 못할 우려가 있기 때문에 CRC 계산을 두부분으로 나누어 프레임 앞쪽 5/8의 CRC를 block 0 시작 전에, 프레임의 나머지 뒤쪽 3/8의 CRC를 block 1이나 block 2의 시작 전에 계산하는 방법이 쓰인다. 이는 대부분의 경우 block 0의 길이가 전체 프레임의 5/8을 넘지 못한다는 점을 이용하는 것이다.

BSI/side information 추출은 CRC 계산으로 프레임에 bit error등이 없는 것으로 밝혀지면 시작된다. Bitstream information은 전체 프레임에 관한 정보로, audio coding mode, audio production information, mixing/compression/normalization 정보, language/time code등이 있다. 이들 정보가 추출된 후에는 이에 따라 여러 변수값들을 초기화한다. Side information은 audio information을 가리키며, 이들은 AC-3의 한 블록에 포함된 decoding 관련 flag등의 정보인 coupling, delta bit allocation, rematrixing/phase flag, exponent strategy, dither flag, bit allocation 정보등이 이에 포함된다.

지수 디코딩은 AC-3 한 프레임분(256샘플)을 TDAC (Time Domain Aliasing Cancellation)변환 후에 그 transform coefficient값을 floating point값으로 바꾸어 추출한 지수를 압축한 것을 원래의 지수 값으로 복원하는 것을 말한다. AC-3 encoder는 spectral envelope를 1/2/4개의 변환계수마다 하나씩 값을 추출한 후에 이 값의 차분값을 범위 -2~2의 값으로 변환하고 이 값 3개를 묶어 5bit로 전송한다. Audio information과 지수의 추출은 xdcall이란 함수에서 수행하는데 이 작업은 단순한 bit unpack의 반복이다. 지수의 경우는 먼저 지수 절대값을 추출하고 그 뒤에 따라오는 차분지수 3개가 그룹된 값을 추출하여 계속 더하면서 1/2/4개씩 저장하는 방법으로 원래의 spectral envelope를 복원한다.

비트 할당 계산은 바로 전 단계에서 생성된 지수값을 이용하여 TDAC의 각 주파수에 해당하는 변환 계수의 가수 (mantissa)가 어떠한 방식으로 인코딩이 되었는지 알아내는 과정이다. 이 부분은 SNR 오프셋, 비트 할당 정보, 지수, 델타 비트 할당 정보(delta bit allocation information)를 토대로 하여 bap(bit allocation pointer)를 계산하는 작업이다. 먼저 지수에서 P.S.D.(Power Spectral density)를 계산하고, P.S.D.에서 마스킹 함수(masking function)을 계산한 후 P.S.D.값과 마스킹 함수값을 비교하여 비트 할당 값을 계산해 낸다.

가수 unpacking/requantization이란 인코더에 보내진 packed 가수(mantissa) data에서 bap값을 이용하여 각 변환 계수에 해당하는 bit data를 분리해내고 이것을 원래의 가수 값으로 회복시키는 작업이다. bap값은 6~15까지는 각 값에 해당하는 길이의 가수가 전송됨을 의미하지만, bap값이 1, 2, 3, 4, 5인 경우는 table에 저장된 가수의 인덱스 자체나 인덱스 2~3개의 그룹이 전송됨을 의미한다. bap가 0인 경우는 가수의 길이가 0 bit로 아무런 정보도 보내지지 않았음을 뜻한다. bap 값에 따라 디코더는 입력된 비트스트림에서 가수나 그 인덱스 또는 인덱스 그룹을 추출하여 원래의 가수 값을 복원하며, bap가 0인 경우에는 dither flag에 따라 dither를 부가하거나 값을 0으로 한다.

다음 과정은 coupling에 의해 분리된 변환계수를 다시 원래의 채널에 결합시키는 decoupling이다. Coupling 채널

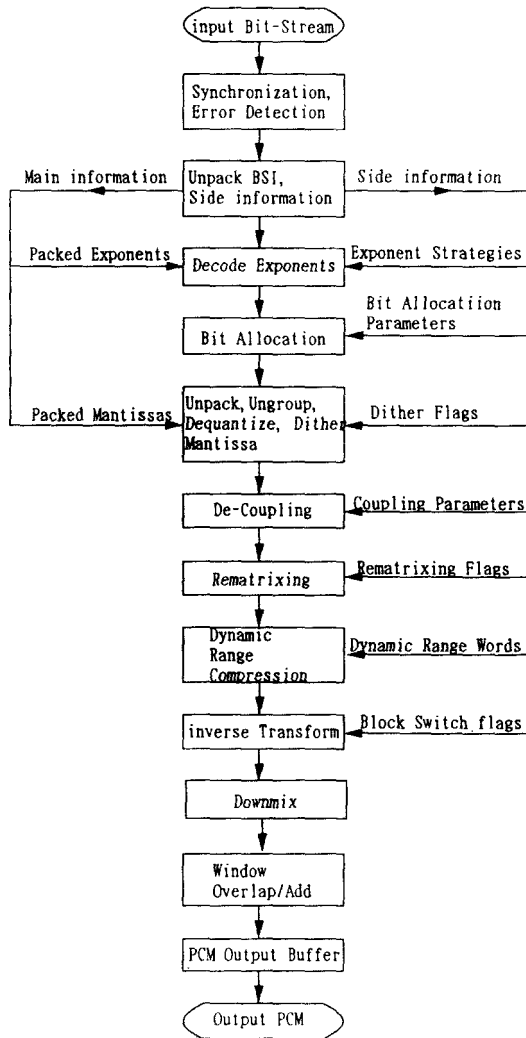


그림 1. decoding process

널에 의해 공유된 변환계수에 coupling된 각 채널별로 전송된 coupling coordinate mantissa, coupling coordinate exponent를 이용하여 scale factor를 곱하고 coupling coordinate exponent만큼 shifting하여 각 채널의 coupling된 변환계수들이 다른 값을 갖도록 한다.

Dynamic range compression에는 dynamic range compression, digital dialog normalization의 두부분이 들어있는데 각 처리는 coupling과 비슷하여 bitstream에 전송된 compression word와 dialog normalization word등을 이용하여 shift factor와 scaling mantissa를 계산하고 그 값을 지수와 가수에 적용하는 작용을 한다.

마지막으로 현재까지 결정된 가수값을 지수값만큼 right shift하여 ITDAC로 입력될 최종적인 변환계수값을 계산해 낸다.

2.2 디코더 출력 프로세싱

디코딩 출력 프로세싱은 디코딩 입력 프로세싱이 끝난 후 각 변환 계수들을 시간축상의 PCM 샘플로 변환 출력하는 부분으로 그림 1에서 보면 rematrixing 이후부터 출력 PCM 단계까지를 말한다.

특히 역TDAC는 전체 오디오 성능에 큰 영향을 주는 부분으로 변환식은 다음과 같이 정의된다.

$$\begin{aligned}
 x[n] &= -\sum_{k=0}^{N-1} X[k] \cos\left(\frac{2\pi}{N} \left(k + \frac{1}{2}\right) (n + n_0)\right) \\
 (\text{for } n &= 0, 1, \dots, N-1 \quad n_0 = (N/2 + 1)/2) \\
 &= -\sum_{k=0}^{N-1} X[k] \operatorname{Re}\{e^{j\frac{2\pi}{N} (k+1/2)(n+n_0)}\} \\
 x[n] &= -\sum_{k=0}^{N-1} X[k] \cos\left(\frac{2\pi}{N} \left(k + \frac{1}{2}\right) (n + n_0)\right) \\
 (\text{for } n &= 0, 1, \dots, N-1 \quad n_0 = (N/2 + 1)/2) \\
 &= -\sum_{k=0}^{N-1} X[k] \operatorname{Re}\{e^{j\frac{2\pi}{N} (k+1/2)(n+n_0)}\} \\
 &= -\sum_{k=0}^{N-1} X[k] \operatorname{Re}\{e^{j\frac{2\pi}{N} nk} \cdot e^{j\frac{2\pi}{N} n_0 k} \cdot e^{j\frac{2\pi}{N} \frac{n}{2}} \cdot e^{j\frac{2\pi}{N} \frac{n_0}{2}}\} \\
 &= -\operatorname{Re}\{e^{j\frac{\pi}{N} (n+n_0)} \sum_{k=0}^{N-1} (X(k) e^{j\frac{2\pi}{N} n_0 k}) e^{j\frac{2\pi}{N} nk}\} \\
 &= -\operatorname{Re}\{e^{j\frac{\pi}{N} (n+n_0)} \operatorname{IDFT}(X'(k))\} \quad (1) \\
 X'(k) &= X(k) e^{j\frac{2\pi}{N} n_0 k}
 \end{aligned}$$

여기에서는 사용되는 IFFT 루틴은 오버플로우 방지를 위해 radix-2 루틴을 거칠 때마다 1/2로 스케일링하였기 때문에 IFFT 루틴후에 신호성분에 대한 스케일링 보상을 해주어야 한다. 그러나 소신호 입력시 IFFT 루틴 후에 보상을 해주면 소신호는 이 루틴에서 그 레벨이 더욱 작아져 결과적으로 SNR 성능이 나빠지게 된다.

소신호 입력시의 SNR 개선을 위한 가장 일반적인 방법은 신호처리전에 일정 팩터의 up-scaling을 행하고, 소정의 신호처리 후에 다시 일정 팩터의 down-scaling을 행하는 것인데, 고정 팩터를 이용한 스케일링과 가변 팩터를 이용한 스케일링의 두가지 방법이 있다.

본 연구에서는 가변 팩터를 이용한 스케일링을 사용하였는데 그 이유는 고정 팩터를 이용하게 되면 up-scaling 할 때에 대신호가 입력되면 오버플로우가 발생할 경우가 생기므로 입력신호에 따라 스케일링 팩터를 적당히 가변시키는 방법을 이용한 것이다.

2.3 가변 팩터를 이용한 스케일링

이 방법은 앞에서 설명한 바와 같이 입력신호의 레벨에 따라 스케일 팩터(scaling factor)를 자동적으로 가변함으로써 최적의 SNR 특성을 제공할 수 있는 방법이다. (그림 2)

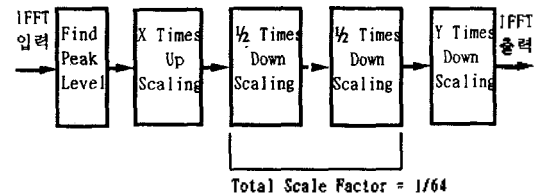


그림 2 가변 팩터를 이용한 플로우

상기에 도시한 블록도는 현재 적용되어 있는 IFFT 프로그램의 up/down scaling factor의 개념도로서 128-포인트 IFFT를 수행할 경우의 예를 보인 것이다. 블록도에서 보는 바와 같이 128-포인트 IFFT의 경우 1/64 Scaling 이 내부 연산과정에서 이루어지기 때문에 이를 보상하기 위해서 IFFT 수행전에 X배 up-scaling을, IFFT 수행후에 다시 Y배 down-scaling을 함으로써 SNR 특성의 향상을 꾀할 수 있다. 한편, 이 방법에서 가장 중요한 포인트는 scaling factor X, Y의 값을 구하는 것으로, 이것은 다음과 같은 flow로 구해진다.

- ① IFFT 입력 데이터중 가장 큰 레벨을 가지는 데이터를 찾는다.
- ② 현재의 IFFT 최대 입력 데이터가 어느 범위의 레벨 인지를 파악하고, 이로부터 up-scaling factor X를 정한다.
- ③ 과정 ②에서 정해진 scaling factor X에 의해 IFFT 입력 데이터를 up-scale 한다.
- ④ IFFT 계산을 수행한다.
- ⑤ 과정 ②에서 구해진 scaling factor X로부터 IFFT 처리후의 scaling factor Y를 다음과 같이 계산한다.  
- 128-포인트 IFFT의 경우 :  
Scaling Factor Y = 6-Scaling Factor X

- 64-포인트 IFFT의 경우 :

Scaling Factor  $Y = 5 \cdot \text{Scaling Factor } X$

⑥과정 ⑤에서 구해진 Scaling Factor Y에 의해 출력 IFFT 데이터를 down-scaling (Y가 양수인 경우)하거나, up-scaling (Y가 음수인 경우) 한다.

2.4 DSP 구현에 의한 AC-3 디코더의 특성 비교

DSP에 의한 AC-3 디코더의 실시간 구현에 있어서 중요한 지표중의 한가지는 C-simulator 에 비교하여 DSP로 구현함으로써 발생하는 성능의 차이를 비교해 보는 것이다. C-simulator가 동작되는 PC나 UNIX 환경은 연산정도에 제한이 없으나, DSP의 경우에는 연산정도의 제한이 있기 때문에 (ZR38001 DSP에서는 데이터의 정도가 20 바이트, 누산기의 정도는 48 비트로 제한됨) C-simulator에 비해 많은 양자와 잡음이 발생할 가능성이 높으며, 이것은 재생되는 신호의 레벨에 따라서도 큰 차이를 보이게 된다. 따라서, 본 절에서는 DSP 구현에 의한 AC-3 디코더의 성능을 정량적으로 분석하기 위하여 C-simulator와의 디코딩 특성을 비교, 분석하였다. 오디오 압축부호화 시스템에 있어서의 객관적 성능평가는 이 분야의 기술자들이 해결해야 할 문제점중의 하나로서, 현재 이를 명확하게 해결해 줄 수 있는 평가법은 제시되어 있지 않다. 따라서, 본 비교분석에서는 간이적인 평가법으로서 각 디코더의 SNR을 측정하여 이를 비교하고자 한다.

본 측정에서 이용된 SNR 값들은 다음과 같은 계산식에 의하여 구해졌다. 먼저, 원래의 테스트신호값을  $s(n)$ , 이 테스트신호  $s(n)$ 을 인코딩하여 이를 디코딩 한 신호를 각각  $s1'(n)$ ,  $s2'(n)$  이라 하자.

( $s1'(n)$  : C-Simulator 에 의해 디코딩된 신호,  $s2'(n)$  : DSP 에 의해 디코딩된 신호)

원래의 신호와 부호화과정을 거친 출력신호의 차이에 러신호는 다음과 같이 정의된다.

$$e1(n) = s(n) - s1'(n)$$

$$e2(n) = s(n) - s2'(n)$$

샘플의 길이가 N인 데이터의 SNR을 구하기 위해서는 먼저 원신호의 에너지를 아래식과 같이 구하고,

$$Es^2 = \frac{1}{N} \cdot \sum_{n=1}^N s^2(n) - \left( \frac{1}{N} \cdot \sum_{n=1}^N s(n) \right)^2 \quad (2)$$

에러신호의 에너지를 다음식과 같이 구한다.

$$Ee1^2 = \frac{1}{N} \cdot \sum_{n=1}^N e1^2(n) - \left( \frac{1}{N} \cdot \sum_{n=1}^N e1(n) \right)^2 \quad (3)$$

$$Ee2^2 = \frac{1}{N} \cdot \sum_{n=1}^N e2^2(n) - \left( \frac{1}{N} \cdot \sum_{n=1}^N e2(n) \right)^2 \quad (4)$$

상기식으로 부터 SNR은 다음과 같이 각각 정의되며, 여기서 SNR1 은 C-simulator 에 의한 것이며, SNR2는 DSP 에 의한 것이다.

$$SNR1(dB) = 10 \log_{10} \left( \frac{Es^2}{Ee1^2} \right) \quad (5)$$

$$SNR2(dB) = 10 \log_{10} \left( \frac{Es^2}{Ee2^2} \right) \quad (6)$$

4 채널에 각각 다양한 주파수의 정현파 (10K, 5K, 1K, 500, 200, 80 Hz)를 배정한 멀티채널 테스트 벡터를 5.1 채널 부호화모드에서 384kbps의 비트레이트로 인코딩하여 이것을 테스트 벡터로 사용하였는데, 각 신호의 재생레벨의 변화에 따른 디코더 특성을 비교, 검토하기 위하여 0 dB 에서 -80 dB 사이의 다양한 레벨의 정현파로 된 테스트 벡터를 사용하였다. 그림 3은 1kHz에 대해 입력레벨에 대한 SNR을 비교한 데이터들이다. 또한 그림 4, 그림 5는 1KHZ 정현파에 대한 파형을 비교한 것이다.

그림 3.에서 보듯이 C-Simulator가 DSP로 구현한 것 보다는 -30dB이하의 신호에 대해선 약 5dB정도의 좋은 SNR을 가지고 있다. 그러나 -30dB이상의 신호에 대해서는 거의 같은 SNR 특성을 보이고 있다. 그림 4,그림 5를 보면 -70dB의 아주 작은 소신호에 대해서 C-Simulator에 비해서 왜곡 없는 거의 같은 신호가 디코딩되어 출력되고 있음을 보인다.

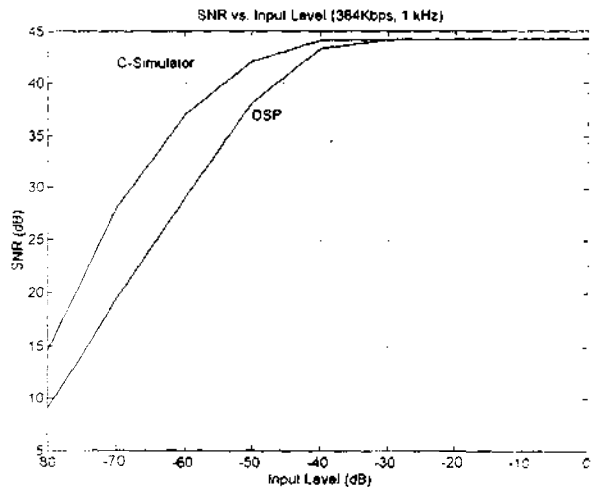


그림 3.

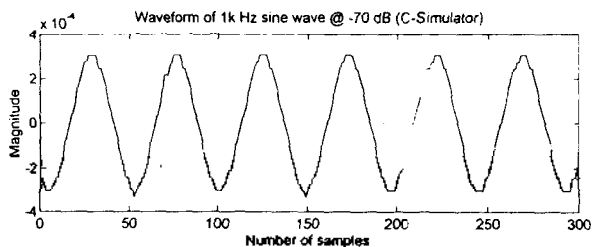


그림 4.

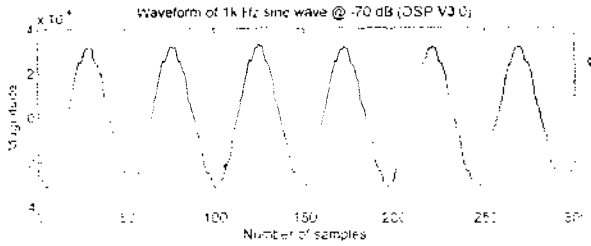


그림 5.

Ⅲ. DVD 재생기의 AC-3 디코더 시스템

DVD 시스템은 기존의 비디오/오디오를 대체할 만한 첨단 광디스크 AV 재생기로서 특별히 오디오에 있어서는 영화의 다중채널을 지원하는 AC-3를 채택하였다. 본 연구에서는 이 DVD 시스템에 있어서 AC-3 디코더의 구현방법과 그 동작원리에 대해 기술하였다.

3.1 디지털 신호처리 프로세서

본 연구에서 AC-3 디코더의 실시간 구현을 위하여 사용된 디지털 신호처리 프로세서로는 Zoran사의 범용 DSP인 ZR38001이 사용되었다. ZR38001은 시리얼 입, 출력 포트가 각각 3개씩 있기 때문에 AC-3의 5.1 채널 출력을 대응할 수가 있다. 이 ZR38001 DSP의 주요기능을 살펴보면 다음과 같다[7].

- 33MIPS (33MHz 외부 클럭)
- 4K × 20 on-chip dual 액세스(access) 데이터 RAM
- 12K × 32 on-chip 프로그램/데이터 ROM
- 1K × 32 on-chip 프로그램 RAM
- Dual address generation per cycle
- Single cycle, 3-operand ALU/multiplier instructions
- 4-cycle radix-2 FFT butterfly
- 20-bit 데이터 paths
- 32-bit instruction opcodes
- 1M × 32 word 프로그램/데이터 address 영역
- 3개의 직렬 입력 포트
- 3개의 직렬 출력 포트

3.2 DVD 오디오 팩(pack)과 AC-3 sync frame

1) DVD AC-3 오디오 팩(pack)

DVD의 AC-3 오디오 pack의 구조는 그림 6.와 같다. (95. 7월 현재 tentative format임.)

DVD의 AC-3의 오디오 pack의 기본단위는 2048 byte이다. 이중에서도 오디오 데이터는 2016byte를 차지하고 있고, 48kHz 샘플링에 64-448kbps로 압축되어 있는 AC-3 bitstream이 기록되어 있다.

호스트는 오디오 데이터의 헤더(header)에 있는 정보를 읽어 기록되어 있는 오디오 데이터의 종류를 분석한

다음 2016 byte의 AC-3 오디오 데이터만을 DSP로 전송하게 된다.

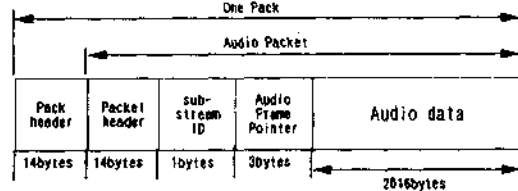


그림 6. DVD Audio Pack 구조

2) AC-3 Sync Frame

AC-3 sync frame은 압축된 6개의 오디오 블럭 (오디오 블럭)을 포함하는데, 이 각각의 오디오 블럭은 256 PCM 샘플에 해당된다. 따라서 한 AC-3 sync frame은 1536 PCM 샘플에 해당하는 데이터인 것이다.

각 sync frame은 블럭 동기 정보(SI)로 시작해서 그 뒤에는 bitstream 정보 블럭(BSI)가 있다. 그 다음에 6개의 오디오 블럭(AB)이 오게 되고 마지막으로 auxiliary 데이터 블럭과 CRC word로 구성되어 있다. (그림 7.)

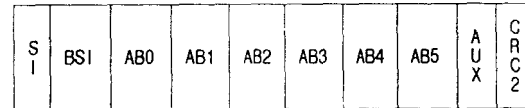


그림 7. AC-3 Sync Frame

이 AC-3 sync frame 크기는 압축율에 따라 변하는데 fs가 48kHz이고 384kbps의 압축인 경우 768 word의 크기를 갖는다.

DVD의 AC-3 디코더를 구현하는데 있어서 중요한 것은 DVD의 AC-3 오디오 데이터의 크기(2016bytes = 1008 words)와 AC-3 sync frame(768 words)의 크기가 다르다는 점이다. 이에 대해서는 다음 절에서 자세히 논하기로 한다.

3.3 AC-3 디코더의 구성 및 동작

DVD 시스템에 있어서 본 연구에서 제안하는 AC-3 디코더는 크게 다음과 같은 블럭으로 이루어져 있다. 1)호스트와의 AC-3 데이터 인터페이스를 위한 메모리 버퍼부와 그 버퍼를 제어하는 제어부, 2)입력된 AC-3 데이터를 디코딩하는 DSP부와, 3)디코딩된 데이터를 출력시키는 출력부로 구성되어 있다. 이 출력부는 디지털 신호를 아날로그로 변화시켜주는 디지털 아날로그 변환기 (DAC)와 디지털 데이터 bitstream을 출력시키는 디지털 트랜스미터(transmitter)부로 나누어 진다. DVD 재생기의 AC-3 디코더 블럭을 그림 8에 나타내었다.

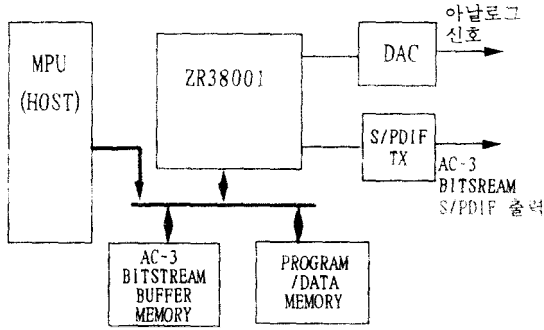


그림 8. DVD 재생기의 AC-3 디코더

1) 메모리 버퍼 및 제어부

위 3.2절에서 설명하였듯이 DVD 오디오 packet과 AC-3 sync frame의 크기는 다르다. 이는 DSP가 외부의 AC-3 데이터 저장용 버퍼 메모리를 효율적으로 활용해야함을 말한다.

본 연구에서는 AC-3 데이터 저장용 버퍼로서 2K word의 메모리를 이용하였다. 실제로 호스트가 한번의 DMA 요구가 있을 때마다 전송하는 데이터 양은 1008 words이지만 AC-3 sync frame의 크기(fs = 48kHz, 384kbps인 경우 768words)와 다르기 때문에 더블 버퍼의 구조로 하여야 한다. 호스트(MPU)가 AC-3 데이터 DMA 전송을 위한 어드레스는 10bit로 0 - 3EF까지 액세스(access)하는데 이 호스트 어드레스는 어드레스 mux에 의해 한번은 0번지 부터 쓰게 되고 한번은 3F0부터 1008 words씩 쓰게 된다. (그림 9.) 여기서 MPU가 AC-3 데이터를 전송하는데 16bit DMA 병렬전송을 하는 이유는 전송속도를 최대한 빠르게 하고 이로 인해 DVD 시스템의 호스트가 비디오 신호처리 및 서보제어등의 다른 작업들을 할 수 있도록 한 것이다.

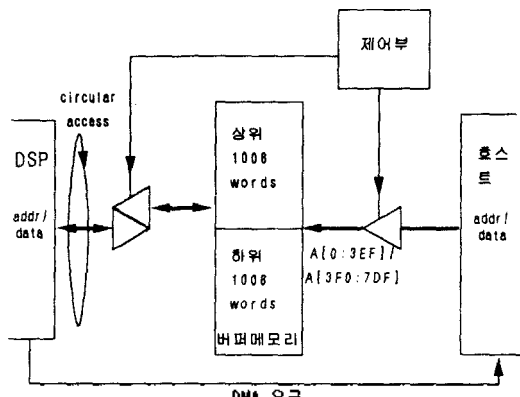


그림 9. 메모리 Access

반면 DSP는 이 외부 버퍼메모리를 회전 링(circular ring) 버퍼의 구조로 액세스(access)하도록 한다. 이렇게 DSP가 외부 버퍼 메모리를 액세스(access)하다가 한

packet안에 있는 AC-3 sync frame을 다 디코딩하고 나면 다시 호스트에게 데이터를 요구하는 DMA 요구신호를 보낸다.

이렇게 함으로써 ZR38001 DSP는 호스트가 전송한 데이터들이 저장되어 있는 버퍼 메모리가 마치 연속된 AC-3 sync frame이 있는 것처럼 데이터를 읽어갈 수 있는 것이다.

제어부에서는 어드레스 및 데이터 버스가 충돌이 나지 않도록 tri-state 양방향 버퍼의 방향 및 인에이블(enable)신호를 제어한다.

2) AC-3 디코딩부

AC-3 디코딩부에서는 호스트로부터 전송되어 외부 버퍼 메모리에 저장된 AC-3 데이터로부터 한 AC-3 frame sync 데이터를 내부 버퍼로 가져와 이를 디코딩하고 매 샘플마다 PCM 데이터를 출력시키는 동작이 이루어진다.

출력 버퍼는 256 PCM 샘플의 크기를 갖는 더블 버퍼의 구조로 한다. 256 샘플은 AC-3 sync frame내의 한 블록을 디코딩하면 생기는 PCM 샘플의 수이다. 즉, AC-3 디코더는 입력은 frame단위로 받지만 출력은 매 블록단위로 256 샘플씩 나가게 된다. 더블 버퍼의 구조로 하는 것은 디코딩하고 있는 동안에도 매 샘플링 주기마다 출력이 되어야 하기 때문이다.

fs가 48kHz이고 384kbps 압축(1 frame sync = 768 words)일때의 AC-3 디코더의 flow chart를 그림 10.에 나타내었다.

여기서 외부버퍼에서 내부 버퍼로 하나의 AC-3 frame sync를 로드(load)하는데 걸리는 시간을 계산해보면, 한 word를 로드하는데에는 2 cycle이 걸리므로,

한 Frame 로드(load) 시간  
 $= 1/33M \cdot (\text{cycles}) \cdot 768 = 46.5\mu s$

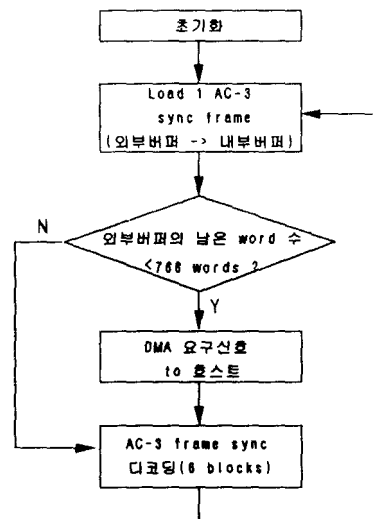


그림 10. AC-3 디코딩 flow chart

이다. 실제로 한 블럭을 디코딩하는데에는 256샘플에 해당하는 시간인 5,33ms가 보장되므로 이 시간은 디코딩할 때 오버헤드(overhead)에 영향을 주지 않는다.

3) 출력부

출력부는 두 부분으로 나뉜다. 디코딩된 PCM 출력을 아날로그 변환시켜 출력해 주는 디지털-아날로그 변환기와 bitstream 데이터를 직접 광출력이나 동축출력으로 보내내주는 디지털 트랜스미터(transmitter)가 그것이다.

디지털 아날로그 변환기로는 BURR BROWN사의 PCM1710을 이용하였다. 이 디바이스는 8배 오버샘플링에 16/20bit을 대응하는 고정도 디지털 아날로그 변환기이다.

디지털 출력으로는 PCM 데이터와 AC-3 bit stream 중에서 선택적으로 출력할 수 있게 하였다. 특히 AC-3 bitstream 출력은 외부의 또 다른 전용 AC-3 디코더를 사용할 때를 위해 출력하게 한 것이다. 여기서 외부 디코더와의 전송방식은 돌비(Dolby)사가 제안하는 S/PDIF format에 맞추었고 이를 위해 bit packing 작업이 추가되어야 한다.

DVD는 AV 기기로서 오디오와 비디오간의 동기가 중요하다. 따라서 외부 전용 디코더를 광/동축 케이블로 연결하여 사용하는 경우에는 내부 디코더와 외부 디코더의 동기가 중요하게 된다. 이는 다음 장에서 설명하기로 한다.

IV. AC-3 디코더 Latency

모든 AC-3 디코더는 예측 가능한 어느정도의 지연시간 후에 PCM 데이터를 출력하게 된다. 이 때문에 디코더의 latency를 정의하는 것이 필요하게 된다.

디코더의 latency를 알기 위해서는 재생중에 lip-sync를 보장하는데 필요한 오디오와 비디오 간의 오프셋을 계산하는 것이 필요하게 된다.

여기서 lipsync란 오디오와 비디오를 동시에 재생하는데 있어서 오디오/비디오간의 동기를 맞추어주는 것을 말한다. 특히 사람의 말소리와 입모양은 반드시 일치시켜야 한다.

2.1 시스템 definition

디코더의 latency를 정의하는데 있어서 시스템의 AC-3 오디오 디코더만 고려하기로 한다. 또한 여기서는 AC-3 데이터가 어떻게 이 디코더에 전송되는지는 고려하지 않았다.

어떤 오디오 프로그램에 있어서 AC-3 bitstream은 일정한 전송율을 가지고 AC-3 디코더에 전송되고 또한 다른 일정한 전송율로 디코딩된 PCM 데이터가 출력된다. 입력 데이터 전송율은 S/PDIF format의 continuous나 burst 전송율이 있다. 물론 출력 전송율은 샘플링 전송율이어야 한다.

AC-3 bitstream을 디코딩하는데에는 다음의 2가지가 정이 있다.

1. 디코더는 AC-3 sync frame을 받기 전에는 디코딩을 시작하지 않는다.

2. 디코더는 한번에 한 오디오 블럭을 처리한다.

2/3 frame의 제약은 CRC1(그림 7 참조)이 frame의 처음 5/8을 포함한다는 것에 기인한다. 즉, 적어도 2/3은 읽어야 CRC를 검사하여 데이터가 타당한지 안전지를 확인하게 된다. 디코더는 또한 한번에 하나의 오디오 블럭을 디코딩하고 출력할 256 PCM 샘플을 만들어 낸다. 이 경우라면 latency는 더이상 디코딩 과정에 좌우되지 않고 256 PCM 샘플을 출력하는데 걸리는 시간에 의해 결정된다. 256 샘플을 출력하는 동안에 디코더는 반드시 다음의 한 블럭을 디코딩하여 다음의 256 샘플이 나와 있어야 한다.

2.2 Latency의 정의

AC-3 디코더 latency는 AC-3 sync frame의 첫번째 word의 도달 시점과 그 frame에 의해 나타나는 첫번째 PCM 샘플 출력의 지연시간으로 정의된다.

$$\text{Latency} = (\text{frame의 } 2/3\text{을 읽는데 걸리는 시간}) + (\text{256 PCM 샘플을 출력하는데 걸리는 시간}) \quad (7)$$

위 식은 다음과 같이 표현될 수도 있다.

$$\text{Latency} = (\text{frame의 } 2/3\text{을 읽는데 걸리는 시간}) + (\text{한 오디오 블럭을 디코딩하는데 걸리는 시간}) \quad (8)$$

단, 위의 변수들은 샘플링 전송율, 입력 데이터 전송율, 그리고 전송 모드에 의해 변할 수 있다. 예나하면 위 파라메타에 따라 AC-3 sync frame의 크기(word수)가 변하기 때문이다.

2.3 내부와 외부 디코더의 동기

이절에서는 내부 디코더와 외부 디코더가 어떻게 오디오 출력의 동기를 맞추는지에 대해 설명하기로 한다.

본 연구의 구현 시스템에서 처럼, AC-3 bit stream을 입력으로 받아 디코딩하여 2 채널 PCM 데이터를 출력

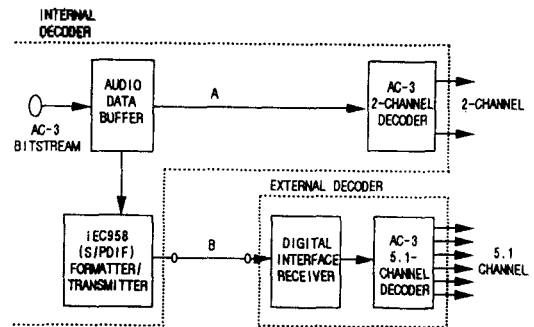


그림 11. 일반적인 시스템의 동기 방식

하고 동시에 외부의 디코더로 AC-3 S/PDIF 출력을 내보내는 시스템에 있어서 디코더 latency는 AC-3 sync frame의 2/3 도달시간과 하나의 블록을 디코딩하는데 걸리는 시간의 합이라 하였다. 하나의 오디오 블록을 디코딩하는데 걸리는 시간을 알고 따라서 디코더의 동기는 이제 2/3 샘플 도달 시간의 함수라 할 수 있다.

내부와 외부 디코더의 동기는 양 디코더에서 AC-3 sync frame의 2/3 점 도달 시점을 일치시키면 된다.

이 동기를 맞추기 위한 일반적인 구현 시스템을 AC-3 384kbps, 48kHz 샘플링의 예를 들어 보도록 하자.(그림 11, 12)

여기서 A점에서는 AC-3 bitstream이 continuous mode로 384kbps의 전송율로 내부 디코더로 전송되어지고 B점에서는 burst mode로서 1.5Mbps로 전송되어진다. 이러한 시스템에 있어서 동기를 맞추기 위해서는 AC-3 sync frame의 1.5배의 용량을 갖고 입력과 출력 포인터를 따로 제어할 수 있는 오디오 데이터 버퍼를 두어야 한다는 것이다.

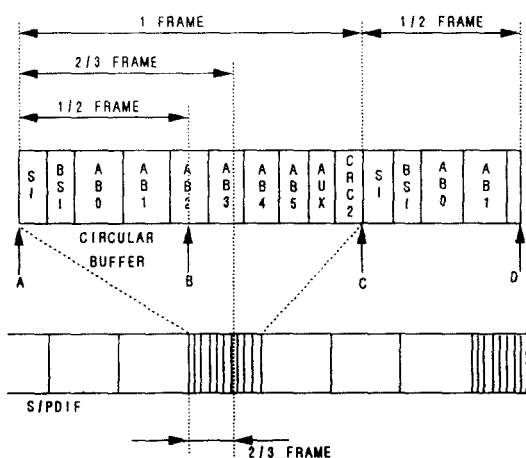


그림 12. S/PDIF 전송을 위한 AC-3 버퍼링

- 동기방식

1. A에서 시작하여 들어오는 오디오 샘플을 저장한다.
2. 입력 포인터가 C에 오면 하나의 AC-3 sync frame이 저장된 것이다. 그리고 오디오 버퍼는 내부 디코더에 들어오는 샘플과 같은 전송율로 A로부터 샘플의 출력을 시작한다.
3. 첫번째 출력 포인터가 B에 도달하면 외부 디코더에 S/PDIF 전송 전송율로 데이터의 출력을 시작한다.
4. 이 버퍼는 회전(circular) 버퍼로 한다. 이제 그림 8과 같이 AC-3 sync frame의 2/3 점이 양 디코더에 동시에 도달하여 동기되어 디코딩하게 된다.

위와 같은 방식의 latency를 계산하면,

$$\text{Latency} = 2/3 \cdot \text{Frame Period}(32\text{ms}) + \text{블럭 Period}(5.33\text{ms}) = 26.67\text{msec이다.}$$

2.4 본 연구의 동기 시스템

본 연구의 구현 시스템에 있어서는, 내부 디코더가 continuous mode의 384kbps로 AC-3 bit stream을 받는 것이 아니고, 앞장에서 설명한 바와 같이 호스트가 외부 버퍼 메모리에 저장한 데이터를 46.5us만에 내부 버퍼로 가져오기 때문에 latency는 전적으로 외부 디코더로 전송되는 S/PDIF 전송율에 의해 결정된다. 따라서 내부 디코더는 외부 디코더로 가는 전송 전송율에서 AC-3 sync frame의 2/3이 도달할 때까지 기다려야 한다. 이를 시간 축에서 보면 그림 13과 같이 나타낼 수 있다.

48kHz 샘플링에 384kbps 압축인 경우, latency를 수식으로 나타내면,

$$\text{Latency} = 2/3 \cdot 1/96000 \cdot 768 + \text{블럭 Period}(5.33\text{ms}) + 46.5\text{us} = 12.125\text{msec}$$

이는 앞 절의 일반적인 구현시스템과 비교해 볼 때 16msec의 지연시간을 감소시켰다.

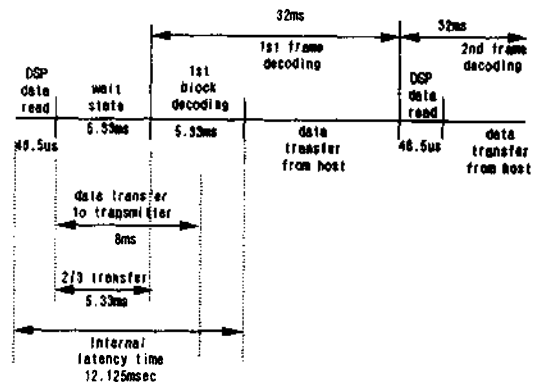


그림 13. 본 구현 시스템의 시간 흐름도

V. 고찰 및 결론

본 연구에서는 AC-3 디코딩 알고리즘을 20bit 고정소숫점 범용 DSP 칩을 이용하여 구현하였으며 또한 차세대 첨단 AV기기가 할 수 있는 DVD 재생기에 적용함으로써 그 실현 가능성을 검토하였다.

본 연구를 통하여 PC를 이용한 AC-3 디코딩 C-언어 시뮬레이션 결과와 그 성능을 비교함으로써 실시간으로 처리한 AC-3 디코더가 뒤지지 않는 성능을 가짐을 보였다.

또한 DVD 재생기에 본 연구의 AC-3 디코더를 적용하는데 있어서 데이터 통신 방식을 DMA전송을 이용함으로써 전체 DVD 시스템을 관장하는 MPU에게 부담을 주지 않고 디코딩할 수 있음을 보였다. 뿐만 아니라, DVD 재생기내의 디코더이외에 S/PDIF 디지털 출력을 통한 외부 AC-3 디코더와의 동기문제도 다룬 결과 본 연구에서는 기존의 지연시간(latency time)보다 향상된 성능을



가짐을 보였다.

향후 DVD뿐 아니라 HDTV등의 멀티미디어 제품에 있어서 이 AC-3 디코더는 중요한 디바이스가 될 것이며 따라서 본 연구와 같은 AC-3 디코더가 전용 IC로 개발하는 것 또한 시급한 과제라고 할 수 있다.

## REFERENCES

1. K. Brandenburg, H. Gerhaeuser, D. Seitzer, and T. Sporer, "Transform Coding of High-Quality Digital Audio at Low Bit Rates: Algorithms and Implementation," proc. of IEEE Intl. Conf. on Comm., Atlanta, April 1990.
2. L. D. Fielder, and G. Davidson "AC-2: A Family of Low-Complexity Transform-Based Music Coders," AES Workshop on Digital Audio, London, October 1991.
3. G. Stoll and Y. F. Deherly, "High-Quality Audio Bit Rate Reduction System Family for Different Applications," Proc. of IEEE Intl. Conf. on Comm., Atlanta, pp. 937-941, April 1990.
4. Grant A. Davidson, Marina Bosi "AC-2: High-quality digital audio coding broadcasting and storage," NAB 1992 Broadcast Engineering Conference Proceedings, 1992.
5. Crag C. Todd, Grant A. Davidson, Mark F. Davis, Louis D. Fielder, Brian D. Link, Steve Vernon, "AC-3: Flexible Perceptual Coding for Audio Transmission and Storage," AES convention 26 Feb., 1994.
6. Dolby Lab, "Digital Audio Compression(AC-3)," (draft) ATSC STANDARD, Aug. 12, 1994.
7. Zoran Corporation, "ZR38xxx Digital Signal Processors," Architectual specifications-Revision 3.4, Jun 24. 1994.

- ▲이 원 우(Won Woo Lee) 1967년 11월 15일생  
1990년 2월: 연세대학교 전자공학과 졸업  
1990년 3월~현재: 삼성전자 DVD 개발센터 전임연구원



- ▲김 승 호(Sung Ho Kim) 1964년 9월 20일생  
1988년 2월: 연세대학교 전자공학과 졸업  
1988년 3월~현재: 삼성전자 DVD 개발센터 전임연구원

- ▲장 성 철(Sung Chul Jang) 1962년 5월 1일생  
1985년 2월: 한양대학교 전자공학과 졸업  
1993년 2월: 미국 워싱턴대 대학원 전기공학과 졸업  
1985년 3월~현재: 삼성전자 DVD 개발센터 선임연구원

- ▲이 희 수(Hee Soo Lee) 1964년 3월 20일생  
1986년 2월: 한양대학교 전자공학과 졸업  
1986년 3월~현재: 삼성전자 DVD 개발센터 선임연구원



- ▲허 재 훈(Jae Hun Heo) 1968년 10월 5일생  
1991년 2월: 포항공대 전자전기공학과 졸업  
1993년 2월: 포항공대 대학원 전산학과 졸업  
1986년 3월~현재: 삼성전자 DVD 개발센터 전임연구원