

# AC-3 Decoding Algorithm Software 구현에 관한 연구

이 건 욱 • 박 인 규

홍익대학교 대학원 전기제어공학과  
121-791 서울 마포구 상수동 72-1  
gunwook@wow.hongik.ac.kr

## A Study on Implementing of AC-3 Decoding Algorithm Software

Gun-wook Lee • In-gyu Park

Dept. of Electrical & Control Eng.,  
Graduate School, Hong Ik University  
72-1 Sangsu-dong Mapo-gu Seoul 121-791  
gunwook@wow.hongik.ac.kr

### 요약

본 논문은 Digital Audio Compression(AC-3) Standard 인 A-52를 기반으로 하였으며 Borland C++3.1 Compiler 를 사용하여 AC-3 Decoding Algorithm 구현하였다. Input Stream은 DVD VOB File에서 AC-3 Stream만을 분리하여 사용하며 최종 출력은 16 Bit PCM File이다. AC-3의 Frame구조는 Synchronization Information, Bit Stream Information, Audio Block, Auxiliary Data, Error Check로 구성된다. Audio Block은 모두 6개의 Block으로 나뉘어져 있다. BSI와 Side Information을 참조하여 Exponent를 추출하여 Exponent Strategy에 따라 Exponent를 복원한다. 복원된 Exponent 정보를 이용하여 Bit Allocation을 수행하여 각각의 Mantissa에 할당된 Bit수를 계산하고 Stream으로부터 Mantissa를 추출한다. Coupling Parameter를 참조하여 Coupling Channel을 Original Channel로 복원시킨다. Stereo Mode에 대해서는 Rematrixing을 수행한다. Dynamic Range는 Mantissa와 Exponent의 Magnitude를 바꾸는 것으로 선택적으로 사용할 수 있다. Mantissa와 Exponent를 결합하여 Floating Point Coefficient로 만든 후 Inverse Transform을 수행하면 PCM Data를 얻을 수 있다. PC에서 듣기 위해서는 Multi Channel을 Stereo나 Mono로 Downmix를 수행한다. 이렇게 만들어진 PCM Data는 PCM Data를 재생하는 프로그램으로 재생할 수 있다.

### 1. 서론

AC-3(Audio Compression Third)는 Dolby에서 개발한 차세대 Digital Audio Compression 기술로서 5개의 Channel인 R(Right), L(Left), C(Center), SR(Surround Right), SL(Surround Left)과 Low Frequency Channel인 Woofer를 지원한다. AC-3는 인간의 청각심리효과를 이용하여 압축 효율은 높고 Audio CD에 가까운 음질을 보장한다. 인간의 귀는 고주파 성분에 둔감하고 큰 소리와 함께 나는 작은 소리는 거의 들을 수 없다. 이러한 특성을 이용하여 들을 수 없는 부분은 Bit를 할당하지 않는다. AC-3는 다양한 Bit Rate를 가지고 있으므로 사용목적에 따라 Bit Rate를 조절할 수도 있다. AC-3는 Multi Channel지원함으로써 현장감 있는 음악과 함께 Multi-Lingual Mode로 자막 없이 모국어로 Dubbing된 영화도 볼 수도 있고 Karaoke Mode로 가정에서도 노래방을 즐길 수 있는 등 다양한 서비스를 지원한다. 현재 AC-3는 미국, 일본, 한국 등에서 DVD Standard로 사용되며, 또한 AC-3는 HDTV, DSS(Digital Satellite System), DCS(Digital Cable System), DTS(Digital Theater System)등에서 Standard로 채택되어 있다. 그리고 미지않아 모든 공중과 방송에서도 AC-3를 채택할 것으로 전망한다.

### 2. AC-3 Audio Frame Structure

AC-3 Stream은 Frame 단위로 구성되며 서로 다른 Frame 사이에 주고받는 정보는 없이 독립된 Frame 구조를 가진다. 하나의 Frame은 Header와 CRC를 빼고나면 6개의

Block구조로 이루어지는데 제일 먼저 나오는 Audio Block0는 나머지 Audio Block들 중에서 가장 많은 Data를 보유하고 있다. Audio Block0가 Data가 많은 이유는 모든 Channel들의 Exponent들과 Bit Allocation Information, Coupling Information, Mantissa를 모두 가지고 있다. 나머지 Block들은 이 정보들을 가질 수도 있지만 대부분은 Audio Block0의 정보를 다시 사용한다. AC-3는 한 Frame에 1536 Sample들을 가진다. 그리고 각 Audio Block들은 256 Sample들을 가진다. 다음은 AC-3의 Frame 구조를 나타내는 그림이다.

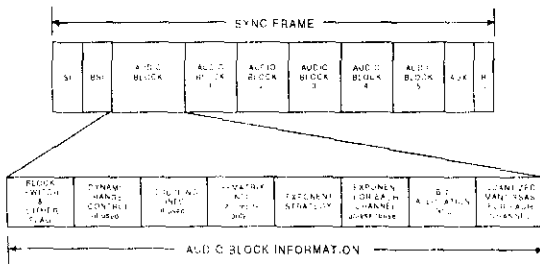


그림 1. AC-3 Frame 구조 및 Audio Block 구조

### 3. AC-3 Decoding 순서

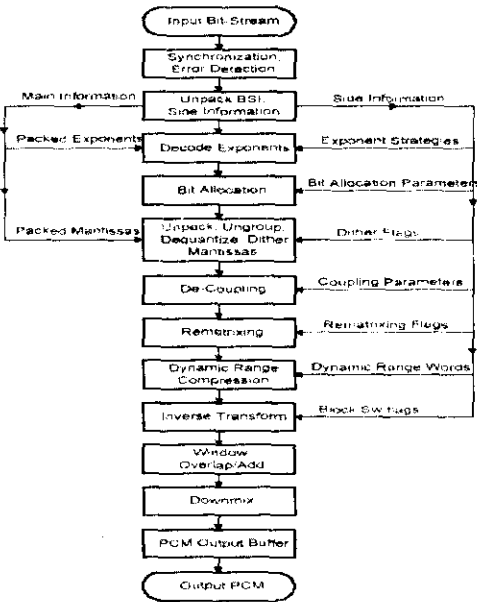


그림 2. AC-3 Decoding Flow Chart

### Input Bit Stream

현재 AC-3 Stream 만으로 구성된 File을 구하기란 쉽지가 않다. 따라서 Input Stream으로는 DVD System File (\*.vob)에서 AC-3 Stream만을 따로 분리해서 Input Stream으로 사용하였다. 이 Test Stream은 44kHz Sampling Rate와 5.1 Channel을 지원하는 AC-3 Stream이다.

### Synchronization and Error Detection

Frame의 SI(Synchronization Information) Block은 *syncword*, *crcl*, *iscod*, *frmsizccod*로 구성되며 모두 40 Bits로 구성 되어있다. *syncword*는 Frame의 제일 처음 16Bits(0B77)이다. *crcl*은 5/8 Frame Size에 대한 CRC Code이다. 따라서 *crcl*을 Check하기전에 *iscod*와 *frmsizccod*로부터 Frame Size가 결정되어야 한다. 길쭉된 Frame Size만큼의 Buffer를 할당하고 File로부터 읽어서 Buffer에 저장한다. 그리고 Frame 전체에 대한 Error Check를 구간별로 수행한다. AC-3는 CRC Code가 2개인 것에 주의해야 한다. Frame의 마지막 16bits *crcl2*는 3/8 Frame에 대한 Error Code이다. Frame Size는 첫 번째 Frame에서만 결정하면 된다. 하나의 File 내에서는 고정된 Frame Size를 가진다. 두 번째 Frame부터는 *syncword* Check와 *crcl1*, *crcl2* Test만 수행한다.

### Unpack BSI(Bit Stream Information)

BSI까지가 AC-3의 Header부분으로서 가변적 구조로 되어있어 AC-3 Test Stream의 BSI는 다음과 같다.

```

bsid = 01000
bsmod = 000
acmod = 111
cmixlev = 00
surmixlev = 00
lfeon = 1
dialnorm = 11111
compre = 1
compr = 00000000
langcode = 1
langcod = 00001001
audprodic = 1
mixlevel = 11001
roomtyp = 10
copyrightb = 1
origins = 1
timecodele = 0
timecoe2e = 0
adbsic = 0
    
```

그림 3. BSI Stream Extract

*acmod*와 *lfcorn*은 Audio Block의 Channel의 개수를 결정하는 정보이다. AC-3 Channel은 1-5개를 사용할 수 있으며 Woofer를 0.1 Channel로 따로 처리한다. 위에서 *acmod*=111은 3/2 Mode에서 5개의 Channel(L,R,C,SL,SR)을 사용한다는 것을 나타낸다. *lfcorn*=1은 Woofer Channel에 대한 정보가 있음을 나타낸다. 따라서 이 Stream은 5.1 Channel을 모두 사용한다는 것을 나타낸다.

**Side Information**

Side Information에는 Exponent Strategies, Bit Allocation Parameters, Dither Flags, Coupling Parameters, Rematrixing Flags, Dynamic Range Words, Block Switch Flags들이 있다. 이 Information들을 이용하여 Stream으로부터 Exponent, Mantissa, Coupling Coordinate들을 추출한다. Side Information은 Audio Block마다 값이 갱신되거나 그 전의 Audio Block에서 쓰인 값을 다시 사용한다. 그러나 Audio Block0에서는 모든 값을 갱신한다. 각각의 Information들은 Block, Channel, Band 또는 Sub-Band 에 대한 Information이다. Information들은 이전의 Parameter에 의해서 종속적으로 선택되어지며 한 Frame또는 하나의 Audio Block 내에서만 유효하다.

**Exponent Decode**

AC-3의 Exponent Strategy에는 D15, D25, D45 세 가지 방법이 쓰인다. D15는 모든 Exponent를 전송하고 D25는 2개중 하나의 Exponent만 전송한다. D45 는 4개중 하나의 Exponent만 전송한다. 그리고 AC-3 Stream에서는 Exponent전송은 Differential Value로 Group Code화하여 Stream에 Packing한다. Exponent의 Range는 0 ~ 24이며 각 Channel의 첫 번째 Exponent만 Absolute Value이며 나머지는 그 전 값들에 대한 Differential Value 들이다

diff exp	mapped value( Mi )
+2	4
+1	3
0	2
-1	1
-2	0

**Encode Exponent**

Coded 7 bit Grouped Value  
 $gexp = (25 \times M1) + (5 \times M2) + M3$

**Decode Exponent**

$M1 = \text{truncate}(gexp/25)$   
 $M2 = \text{truncate}((gexp\%25)/5)$   
 $M3 = \text{truncate}(gexp\%25)\%5$   
 $dexp = Mi - 2$   
 $exp[n] = exp[n-1] + dexp[n]$

**Bit Allocation**

Bit Allocation은 Bit Allocation Parameter 와 Stream 으로부터 추출한 Exponent로 Mantissa에 할당된 Bit 수를 계산하는 것이다. Bit Allocation 계산과정은 모두 7단계를 거쳐야한다.

1) Initialization

Full Bandwidth Channel, Coupling Channel, LFE Channel에 대한 Range 조기화 작업을 한다.

2) Exponent Mapping into PSD

Exponent를 13 Bit Log Power Spectral Density로 변환한다.

3) PSD Integration

각 Channel을 50개의 Band로 통합하고 각 Band에 대해 Mantissa개수를 할당한다.

4) Compute Excitation Function

Integrated PSD spectrum에 대해서 Masking Curve Model을 적용하는 것이다.

5) Compute Masking Curve

Exciting Function으로부터 Masking Noise Level Threshold Curve를 계산한다. 인간의 귀로 들을 수 없는 영역은 Bit를 할당하지 않는다.

6) Apply Delta Bit Allocation

Masking Curve를 Level을 올리거나 내린다.

7) Compute Bit Allocation

각각의 Mantissa에 4 Bit BAP(Bit Allocation Pointer)를 할당한다. BAP의 Range는 0-15이다.

**Dequantize Mantissas**

Mantissa의 BAP 0~5 까지는 Symmetric Quantization을 적용하며 6~15는 Asymmetric Quantization을 적용한다. 그리고 1,2,4 Bit가 할당된 Mantissas는 다음과 같이 Group Code로 Packing되어 있으므로 다음과 같이 Group 을 해체해야 한다. 그리고 mantissa\_code는 해당하는 Table을 참조하여 각각의 Mantissa로 복원하여야 한다.

**Decoder Equations**

$bap = 1:$   
 $\text{mantissa\_code}[a] = \text{truncate}(\text{group1} / 9)$   
 $\text{mantissa\_code}[b] = \text{truncate}((\text{group1} \% 9) / 3)$   
 $\text{mantissa\_code}[c] = (\text{group1} \% 9) \% 3$   
 $bap = 2:$   
 $\text{mantissa\_code}[a] = \text{truncate}(\text{group2} / 25)$   
 $\text{mantissa\_code}[b] = \text{truncate}((\text{group2} \% 25) / 5)$   
 $\text{mantissa\_code}[c] = (\text{group2} \% 25) \% 5$   
 $bap = 4:$   
 $\text{mantissa\_code}[a] = \text{truncate}(\text{group4} / 11)$   
 $\text{mantissa\_code}[b] = \text{group4} \% 11$

다음은 BAP에 대한 Quantizer Level, Quantization Type, 그리고 할당된 Bit수를 나타낸다.

bap	quantizer levels	quantization type	mantissa bits (qntztab[bap]) (group bits/ num in group)
0	0	none	0
1	3	symmetric	1.67 (5/3)
2	5	symmetric	2.33 (7/3)
3	7	symmetric	3
4	11	symmetric	3.5 (7/2)
5	15	symmetric	4
6	32	asymmetric	5
7	64	asymmetric	6
8	128	asymmetric	7
9	256	asymmetric	8
10	512	asymmetric	9
11	1024	asymmetric	10
12	2048	asymmetric	11
13	4096	asymmetric	12
14	16,384	asymmetric	14
15	65,536	asymmetric	16

### De-Coupling

Coupling Channel은 여러 Channel들에 대한 고주파 성분 Data를 하나의 Channel로 만들어 압축한 것이다. Coupling Channel의 Data 구조는 4Bit의 *cplcoexp*, *cplcomant*와 2 Bit의 *mstrcplco*, 그리고 Exponent와 Mantissa로 구성된다. 또한 Coupling Channel은 몇 개의 Band로 구성되는데 하나의 Band 내에서는 동일한 Coordinate를 가진다.

Coupling Channel을 Original Channel로 복원하기 위해서는 각각의 Channel에 대한 Coupling Coordinate정보를 이용하여 Original Channel을 복원한다. Coupling Coordinate는 Exponent와 Mantissa로 분리되어 Packing 되어 있으므로 다음과 같은 과정을 통해서 Coordinate로 바꾼다.

```

if( cplcoexp[ch][band] < -15){
    temp[ch][band] = cplcomant[ch][band]/16;
}
else{
    temp[ch][band] = (cplcomant[ch][band]+16)/32;
}
cplcol[ch][band] = temp[ch][band] >>
                    (cplcoexp[ch][band] +
                    3mstrcplcol ch);
    
```

Original Channel의 Mantissa는 복원된 *cplco*와 Stream으로부터 추출된 *cplmant*을 사용하여 다음과 같이 복원한다.

$$chmant = cplmant * cplco * 8$$

### Inverse Transform

모든 Channel의 Mantissa와 Exponent가 복원되면 DCT Coefficient를 복원하기 위해서 Mantissa를 Exponent Value만큼 Right Shift한다.

복원된 Coefficient는 Block Switch에 의해서 *blksw*=0 인 경우는 512-Sample IMDCT Transform을 수행하게 된다. *blksw*=1 인 경우는 256-Sample IMDCT Transform을 수행하게 된다. IMDCT를 수행하고 나면 각 Channel에 대한 16Bit PCM Data가 생성되는 것이다. 필요에 따라서는 Multi-Channel을 Downmix 과정을 통해서 Stereo나 Mono로 Channel수를 줄일 수 있다.

### 4. 결론 및 향후 계획

AC 3 Test 환경은 Intel MMX200 CPU를 장착한 PC에서 Borland C++3.1 Compiler를 사용하였다. Input Stream은 Frame Size 1792 Byte, 48kHz Sampling Rate의 5.1 Channel AC 3 Stream을 사용하였다.

따라서 한 Frame이 가지고 있는 Data는 시간상으로 32ms 정도의 Data이다. Decoding Algorithm을 Test 할 경우 한 Frame을 Decoding 하는데 걸리는 시간은 32ms를 초과한다. 아직은 실시간으로 AC-3를 Play 하는데는 문제가 있고 Program을 최적화가 필요하다. AC-3 Decoder를 구현하는데 있어서 Mantissa Ungrouping과 Bit Allocation을 이해하는데 다소 어려움이 따랐다. 그러나 AC-3 Decoding Algorithm을 Software로 구현함으로써 Software AC-3 Player의 개발과 함께 Software DVD Player를 개발할 수 있는 기반을 마련하였다. 더 나아가 AC-3 Encoder를 구현함으로써 Software DVD 제작장비나 음향, 방송장비를 개발하는데 기여할 수 있을 것으로 생각된다.

### 참고문헌

- [1] ATSC Digital Audio Compression(AC-3) Standard (A/52). Dec 1995
- [2] Ken C. Pohlmann, Principle of Digital Audio, Mc Grow Hill, Third Edition
- [3] K.R Rao and J.J Hwang, Techniques & Standards for Image-Video & Audio Coding, Prentice Hall Second Edition
- [4] N. S. Jayant and P. Noll, "Digital Coding of Waveform : Principle and Application to Speech and Video," Prentice Hall, 1984