

# TMS320C80을 이용한 실시간 처리 AC-3 Encoder 하드웨어 구현에 관한 연구

여 경 현\*, 박 인 규  
홍익대학교 전기제어공학과  
121-791 서울 마포구 상수동 72-1  
spagetti@wow.hongik.ac.kr

## A study on implementing real-time AC-3 audio encoder hardware based on TMS320C80

Kyung-hyun Yeo\*, In-gyu Park  
Dept. of Electrical & Control Eng. Honk Ik Univ.  
72-1 Sangsu-dong Mapo-gu Seoul 121-791  
spagetti@wow.hongik.ac.kr

### 요약

차세대 DVD system의 audio 규격인 Dolby AC-3를 구현하는 방법으로 DSP 프로세서인 TMS320C80을 사용하여 실시간 처리 가능한 하드웨어 바탕의 firmware 소프트웨어를 개발하는 방법으로 구현하고자 한다.

본 논문에서는 먼저 TMS320C80을 바탕으로 한 하드웨어 구현에 관해 논의한다.

하드웨어의 구조는 TMS320C80과 시스템 메모리로서의 DRAM, 오디오 입력부인 ADC, 입력 데이터를 효과적으로 사용하기 위한 FIFO memory, 오디오 출력용인 DAC, 디버깅 및 통신포트로 USB, RS-232, LPT와 MPEG-2 encoding보드 등 다른 보드와 연계를 위한 local-bus를 위한 dual port ram으로 구성된다. 오디오 입력은 최대 24bit 48kHz sampling까지 받을 수 있다.

### 1. 서론

고품질의 디지털 오디오는 데이터 량이 많기 때문에 전송을 위해선 디지털 압축 알고리즘을 이용하여 데이터 량을 줄여야만 한다.

디지털 오디오의 압축방법에는 크게 AC-3와 MPEG-2 Audio가 있다. 이 두 방법의 성능은 비슷하지만 미국과 유럽 등의 선호도가 각각 다르기 때문에 전 세계 공통의 표준은 한가지가 아닐 것이다.

또한 코딩의 각 단계의 세부적인 알고리즘은 더 나은 방법이 계속해서 개발 중에 있기 때문에 앞으로 많은 발전이 있을 것이다.

본 논문에서는 압축의 모든 알고리즘이 소프트웨어적으로 처리되므로 MPEG-2 오디오, AC-3등의 어떠한 압축 방법이 적용되더라도 그에 효과적으로 대처할 수 있으며 차후 개발될 알고리즘에 대해서도 적용이 가능 할 것이다.

AC-3는 HDTV 전송을 위한 디지털 텔레비전 표준인 ATSC 표준 A/53으로 채택되어 있다.

본 논문의 AC-3는 Digital Audio Compression (AC-3) ATSC Standard에서 제시한 규격을 준수하며, 정작실시효과를 바탕으로 한 AC-3 알고리즘을 TMS320C80을 사용하여 소프트웨어적으로 처리 할 수 있도록 한다.

약 5Mbps( 6채널×48kHz×18bits = 5.184 Mbps )의 PCM 데이터를 AC-3 인코더로 압축한다면 384 kbps로 압축할 수 있다.

TMS320C80(MVP)은 120-MFLOP의 IEEE floating point unit를 갖고있는 32-bit RISC프로세서 하나(MP)와 4개의 32-bit 병렬처리 DSP(PP)를 내장하고 있으며, 480Mbyte/s의 off-chip transfer rate의 transfer controller를 내장하고 있어, AC-3 실시간 인코딩을 소프트웨어적으로 구현하기에 적합한 프로세서이다.

먼저 하드웨어의 전체 구조를 알아보고 MVP의 Packet Transfer를 이해한 후 압축 알고리즘의 구현

을 위한 방법을 알아본다. 다음엔 실험에 대한 평가를 내리고 검토와 결론을 맺어 보고자 한다.

## II. 하드웨어 구조

전체 보드의 구조는 다음 그림과 같이 TMS320C6x과, 시스템 메모리로 DRAM, EPROM을 가지고 있으며, ADC, FIFO, DAC, USB port, RS-232, LPT, dual port ram으로 구성된다.

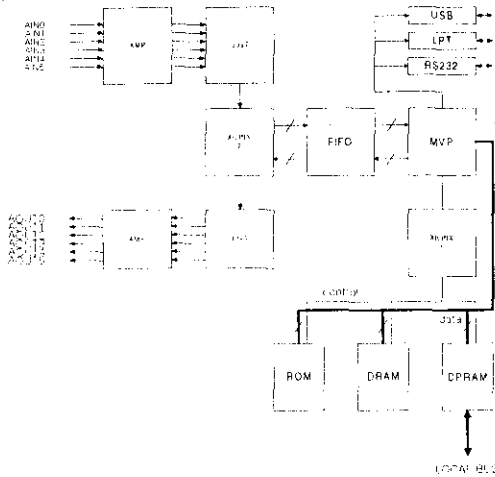


그림 1. 시스템 블록도

Input PCM Word의 길이는 최대 24Bit 까지 converting이 가능하며,  $512 \times 24 \times 6 = 73728 \text{bit} (9\text{kB})$  씩 처리한다. 입력부분의 data loss를 보완하고, 병렬처리를 최적화 시키기 위해 버퍼로서  $512 \times 4 \text{byte}$ 의 FIFO를 사용하였다.

프로그램 메모리인 EPROM은 버스 폭이 32bit이며, 용량은 512kByte이다. 설계 Stand alone으로 동작하게 되면, AC-3 인코딩 프로그램을 탑재하게 되며, 입력PCM word, 출력 스트림을 임시 저장하는 데에는 DRAM을 사용하였다. DRAM은 버스폭 32bit로, 용량은 32MB를 저장할 수 있다.

아날로그 오디오 입력을 디지털 오디오로 변환할 수 있도록 ADC(CS5397)를 사용하였는데 이는 sampling rate를 입력하는 clock에 따라 가변적으로 조절할 수 있다. 입력 clock의 주파수에 따라 32kHz, 44.1kHz, 48kHz를 선택할 수 있도록 하였다. 입력 중에 데이터의 유실을 막기 위해 FIFO (74ACT3638)를 사용하여 신뢰도를 높일 수 있게 하였다.

소프트웨어로 코딩하기 때문에 AC-3 오디오 디코더도 구현가능 하다. AC-3 디코더를 위한 아날로그 오디오 출력용으로 DAC(CS4390)를 사용한다.

입력부분은 1 채널부터 5.1채널까지 모두 6채널의

오디오 신호를 입력받아 사용할 수 있으며 그중 0.1 채널은 저주파 신호(subwoofer)를 전달하기 위한 lfe( low frequency enhancement ) 채널이다. Input signal은 3Hz single pole filter를 이용하여 High-pass filtering된 signal을 사용하며, LFE Channel 은 120Hz에서 low-pass filtering 되어진다.

데이터 출력용으로는 MPEG-2 encoding보드 등 여타보드와의 연계를 위한 local-bus를 갖고 있으며 이를 통한 데이터 공유를 위해 dual port ram으로 구성된다.

또한 디버깅 및 통신 포트로 USB port, RS-232, LPT를 사용하였고, 이는 각각 초당 약 10MB, 0.1MB, 0.5MB의 속도를 낸다.

## III. FPGA 설계

MVP와 여러 chip들을 인터페이스 하는 데에 디버깅의 편의성을 위하여 두 개의 FPGA를 사용하여 설계하였다. part 1은 Reset, MVP Bus control, Address decoding, Ready Signal 생성, part2와의 Interface를 담당하며, part 2는 ADC 설정 및 초기화, ADC의 serial interface, FIFO reset, FIFO Write, MVP의 FIFO access를 위한 interface 부분을 담당한다.

MVP는 DRAM의 access를 기본으로 설계되었기 때문에 DRAM access모드만을 맞추어 주면 된다.

BUS의 설정은 AS[2:0], P[3:0], BS[1:0], CT[2:0]으로 각각 Address shift, Page size, Bus size, Clock timing을 어드레스맵에 따라 맞도록 외부에서 입력하여 주면 된다. Row Address time에 크게 영역이 결정되기 때문에 이것으로 적절히 핀들을 동작시켜 주면 된다. 이것을 MVP access timing에 맞도록 10ns 안에 해주는 것이 관건이다. 될 수 있으면 핀들의 변화가 적도록 어드레스맵을 정비해 주어야 한다.

리셋은 MVP를 reset과 함께 halted mode로 동작시켜야 한다. XILINX가 configure되는 동안 MVP는 동작이 불가능하기 때문이다. XILINX가 활성화되면 MVP를 running시켜주는 방식으로 되어 있다.

ROM은 속도가 느린 단점이 있지만 MVP내부의 cache를 이용하므로 전체 performance에는 지장이 없다. 그밖에 UART등과 같이 느린 chip에 대해서는 Ready 신호를 이용하여 타이밍을 맞추어 주었다.

ADC로부터 FIFO까지의 입력은 ADC로부터 출력되는 clock을 이용하여 입력되는 즉시 FIFO로 write해 주게 된다. 이 부분은 MVP와는 별개로 XILINX에서 logic으로만 처리해 주었다. 반대쪽 FIFO에서부터 MVP로의 출력부는 MVP의 cas를 ACT3638의 clock으로 사용하여 동작시켜 주었다. FIFO의 입출력 포트는 독립적으로 동작한다.

ADC는 두 채널의 오디오를 24비트 PCM형식의 데이터로 변환한다. 출력된 신호의 포맷은 I<sup>2</sup>S형식이며, SCLK, LRCK, SDATA로 구성되어 있다. LRCK는 channel 0또는 1을 나타내며 이는 sampling rate와 같은 clock이 된다. SCLK는 LRCK의 32배가 되는 clock이며, LRCK의 edge에 맞추어 시작하면 SCLK의 상승 edge에서 MSB먼저 한 비트씩 읽을 수 있게 된다. 24비트는 PCM데이터이며 마지막 8bit는 Peak level로 활용할 수 있다. XILINX내부에서 알맞게 shift시커 FIFO로 저장할 수 있도록 한다.

Audio 출력부는 일정 clock마다 FIFO에서 읽어와 DAC로 출력하여 준다. MVP은 FIFO의 almost empty flag를 참조하여 data loss가 생기지 않도록 해 주었다.

IV. packet transfer

오디오 입력 과실중 fifo를 사용한다. 사용한 fifo의 용량은 512sample인데, ADC로부터의 입력으로 512sample이 차는데 예는 48kHz sampling의 경우 6채널이므로 약 1.7ms가 걸린다. 448 sample이 차면 flag가 세팅되어 MVP에서 데이터를 읽어야 한다. 읽는 속도는 일반적인 access방법으로 약 1.5ms가 걸렸다. FIFO에서 읽는 속도가 쓰는 속도에 비해 별로 빠르지 않기 때문에 좀더 빠른 방법을 쓰지 않으면 MP의 효율이 떨어져 PP를 사용하는 데에 무리가 따른다.

이를 위해 packet transfer를 사용하여 전송 속도를 높이게 된다.

Packet Transfer(PT)는 C80의 두 메모리간의 데이터 블록 전송을 말한다. 어떤 프로세서가 PT를 요청하면, 그 프로세서는 다음 프로그램 수행을 계속할 수 있다. PT는 추가의 프로세서 사이를 없이 TC(Transfer Controller)에 의하여 수행된다.

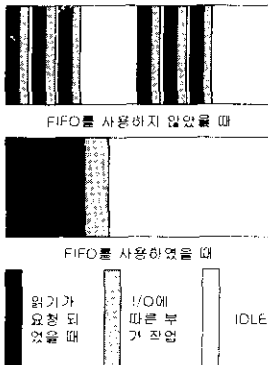


그림 2. FIFO와 Processor의 부하

약 500개의 data를 FIFO지만 연속적인 데이터로

보아도 상관없다. 이것을 PT로 읽는데 예는 약 0.5ms정도 소요되었다. 시간으로만 따졌을 때 MP의 전체 사용량의 약 90%를 데이터 전송에 사용하였지만 PT를 이용하여 28%만을 사용하게 되었다. access time이 15ns인 FIFO를 사용하였기 때문에 wait cycle을 없애면 더 빠르게 읽을 수 있지만 그렇지 않아도 충분한 속도이다.

패킷 트랜스퍼의 순서는 다음과 같다.

- 1) Packet Transfer request parameter 초기화
- 2) 64바이트의 버퍼에 packet transfer parameter를 넣는다.
- 3) MP의 parameter memory에 linked-list pointer로 packet transfer parameter의 어드레스를 넣는다.
- 4) PKTREQ 레지스터를 사용하여 TC로 packet transfer를 요청한다.

V. 압축 operation

AC-3 압축의 알고리즘은 다음의 블록도와 같다.

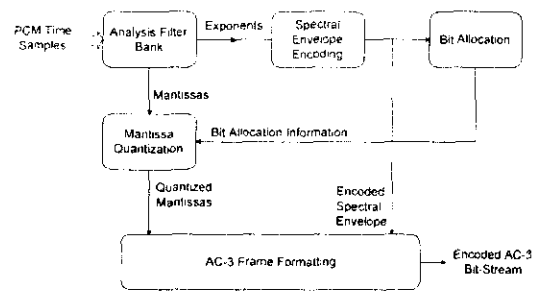


그림 3. AC-3 Encoding Summary

위의 모든 과정은 소프트웨어로 처리되며, 이로써 각각의 세부적인 알고리즘을 구현하는 데에 있어 다양한 방법이 적용 가능하게 된다. 실제 소프트웨어에서 처리하는 Processing은 그림 4의 순서도와 같다. 결국 이것은 계속되는 입력에 대하여 같은 처리를 반복하게 되는 것이므로, DSP를 사용하였을 때 일반 프로세서를 사용하는 것 보다 더 빠른 연산이 가능하다. 이것을 단계별로 PP 4개로 나누어 처리한다면 4배 이상의 속도 향상을 얻을 수 있다.

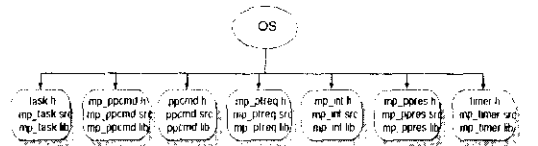


그림 4. MVP Kernel의 구조

프로그램은 ATSC의 AC-3표준안을 기반으로, TI사에서 제공한 커널을 사용하여 C로 코딩하였다.

Kernel이 직접적으로 multitasking, semaphores,

message passing, exception handling과, private contexts등을 지원한다.

각 단계마다 한번에 처리하는 데이터의 양은 Time domain에서는 한 채널당 512 샘플씩 취하여 처리하며 frequency domain에서는 한 channel당 256개의 coefficient를 가진다. 따라서 AC-3 싱크 프레임은 한 프레임당 1536 (256×6) 오디오 샘플을 가지게 된다.

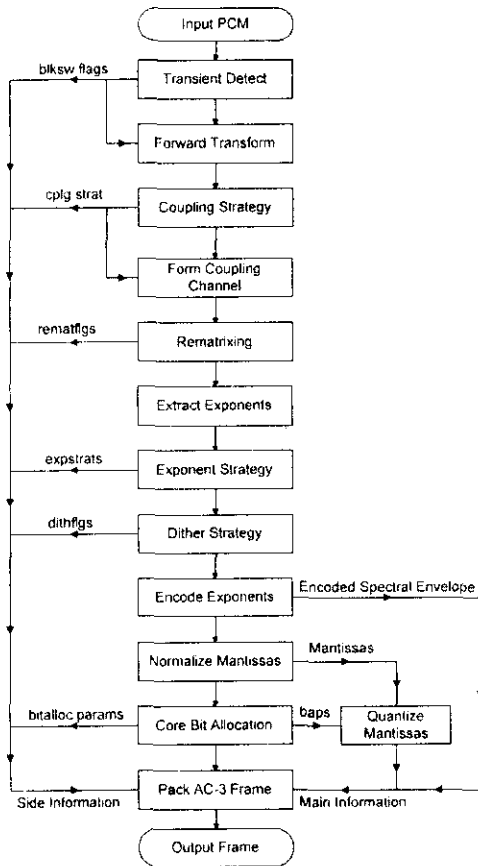


그림 5. AC-3 encoding Flow chart

VI. 테스트환경

TI사에서 제공하는 TMS320c80용 emulator인 XDS510을 사용하여 디버깅을 하였다. XDS510은 TMS320c80과 Sun SPARC workstation을 JTAG (IEEE 1949.1) 인터페이스로 연결하여 테스트 환경을 제공하는 Emulator 이다. 이를 통해 실험한 결과 입

력한 오디오 시그널로부터의 AC-3 인코딩 과정의 부분별 알고리즘을 검증하였고 AC-3 스트림의 생성을 test 하였다.

VII. 결론

1 채널부터 5.1채널까지 모두 6채널의 오디오 신호를 입력받아 사용할 수 있으며 인코딩된 비트스트림은 로컬버서나 USB port로 출력하여 MPEG-2 비디오 스트림과 Multiplex되거나 독자적으로 AC-3 스트림을 전송할 수 있다.

입력중 0.1 채널은 저주파 신호(subwoofer)를 전달하기 위한 lfe( low frequency enhancement ) 채널이다. Input signal은 3Hz single pole filter를 이용하여 High-pass filtering된 signal을 사용하며, LFE Channel은 120Hz에서 low-pass filtering 되어진다. Input PCM Word의 길이는 최대 24Bit 까지 converting이 가능하며, 512\*24\*6=73728bit (9kB)씩 처리한다. 완성된 Data Rate는 32kbps부터 640kbps 까지 가능하다.

AC-3의 최대 PCM word length는 24비트이지만 실제 구현 시 오디오 입력부에 잡음이 발생하여 이를 사용하지 못하였다. 24비트의 고품질의 오디오 소스는 보통 DAT와 같은 디지털 저장장치에 있다. 이를 아날로그로 바꾸어 다시 디지털로 변환하는 데에는 문제가 있으므로 디지털 입력을 받을 수 있도록 해야 한다.

AC-3은 결국 만들어지는 스트림의 포맷이 AC-3의 syntax에만 맞으면 되므로 단계별의 코딩 알고리즘은 무궁무진한 응용 가능성이 있다. TMS320c80의 고속 병렬처리 특성을 이용하여 더 나은 performance를 낼 수 있는 방법을 찾아야 하며, 이와 함께 MPEG-2 오디오 스트림에 대한 연구도 가능하다.

참고문헌

[1] ADVANCED TELEVISION SYSTEMS COMMITTEE, "DIGITAL AUDIO COMPRESSION (AC-3) ATSC STANDARD", Dec. 1995  
 [2] Texas Instrument, "TMS320c80 Digital Signal Processor Datasheet", Texas Instrument, Oct. 1997  
 [3] TI, "TMS320c80 Online Reference", Oct. 1998