

## DB-X 의 DSPA를 이용한 MP3 플레이어 설계 및 구현

김성훈, 손용기, 유진호, 이진우, 한동원  
한국전자통신연구원 컴소연 인터넷정보기전연구부

### A Design and Implementation of MP3 player using DSPA module of DB-X

Sung Hun Kim, Yong Ki Son, Jin Ho Yoo, Jeun Woo Lee, Dong Won Han  
Internet Appliance Technology Dept. , CSTL, ETRI

**Abstract** - 이 논문에서는 디지털 시그널 프로세싱에 적합하지 않은 RISC타입의 DB-X 마이크로컨트롤러를 이용하여 MP3 플레이어를 구현한 방법을 제안한다.

DB-X 에는 DSPA 모듈이 있어 FIR, FFT 등 계산이 많이 요구되는 부분을 DSP 수준으로 빨리 연산해 줄 수가 있으며 SSI 모듈이 있어서 44.1Khz 의 샘플링 주기를 가지는 D/A 코덱과 연결할 수가 있다. 또한, SSI 와 음성 데이터가 저장되어 있는 메모리 사이를 CPU 부담없이 데이터를 전송하기 위한 DMA 모듈이 있어 MP3 데이터를 프레임 단위로 디코딩할 때 더블 버퍼링 기법을 쉽게 사용할 수가 있다.

## 1. 서 론

지능형 휴대 정보 단말의 성능이 급속하게 발전함에 따라 기존의 데이터 저장 기능뿐만 아니라 멀티미디어 기능의 요구가 증가하고 있다. 이러한 멀티미디어 요구의 가장 기본적인 형태는 휴대 단말에서의 오디오 구현이다.

그런데, 지능형 정보 단말에서 주로 사용되고 있는 RISC 타입의 마이크로컨트롤러(microcontroller)는 디지털 시그널 프로세싱을 수행하도록 최적화 되어 있지 않으므로 MP3(MPEG-1 Layer3) 디코딩 등의 많은 계산을 요구하는 작업을 수행하기는 적합하지 않다. 특히, PDA 와 같은 내장형 시스템에서 마이크로컨트롤러를 이용하여 MP3 플레이어를 구현하기 위해서는 MP3 디코더 칩을 마이크로컨트롤러의 GPIO(General Purpose I/O) 모듈 등을 이용하여 연결하거나 소프트웨어적으로 디코딩 기법을 최적화해야 했지만 비용, 인터페이스, 전력 등의 문제로 많은 어려움이 있었다.

Motorola 가 새로 개발한 RISC 타입의 DB-X 마이크로컨트롤러에는 DSPA(DSP Accelerator) 모듈이 있어 FIR, FFT 등의 계산을 DSP 수준의 속도로 할 수 있으며 SSI(Serial Synchronous Interface), DMA 등을 적절히 이용하면 MP3 디코더 칩 등의 외부 칩을 사용하지 않고 DB-X 마이크로컨트롤러 만으로도 MP3 플레이어를 구현할 수 있다.

본 논문은 현재 Motorola와 ETRI가 함께 수행하고 있는, M341 마이크로프로세서(microprocessor)와 USB, BTA, LCDC, DSPA 등의 주변장치들이 원칩화 되어 있는 아직 미출시된 DB-X 마이크로컨트롤러를 이용한 PDA 개발중에서 DB-X 의 DSPA 모듈을 이용하여 MP3 플레이어를 설계하고 구현한 방법을 제시한다. 우선 MP3 플레이어의 전체 구현 절차에 대해 설명하고 MP3 플레이어 구현을 위한 H/W, S/W 구현 방법에 대해 기술한 후 결론을 맺는다.

## 2. 본 론

### 2.1 MP3 디코더의 전체 구현 절차

MP3에서는 MDCT(Modified Discrete Cosine Tr

ansform), 심리음향 모델, 비선형 양자화, subband coding 그리고 Huffman Coding과 같은 기법들이 복합적으로 적용되어 오디오를 압축하게 된다[1]. 그림 2.1 에 MP3 디코더의 기능 블록에 대해 나타내었다.

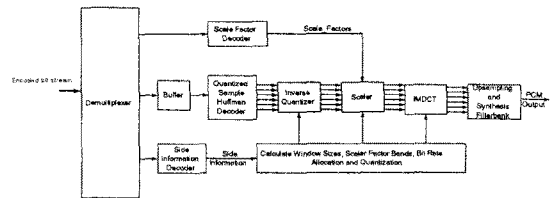


그림 2.1 MP3 decoder의 기능블록

그림 2.1의 기능 블록에 따라 디코딩 과정을 개략적으로 설명하자면 다음과 같다. header, CRC, side information, scale factor, Huffman code data이 반복적으로 구성되어진 MP3 비트스트림이 Demultiplexer를 통해 디코딩 과정에서 실제로 필요한 정보인 side information, scale factor, Huffman code data로 분리되어 각 블록별로 사용된다. 이렇게 나누어진 정보 중 Huffman code data는 디코딩되어 32개 주파수 대역별로 18개의 양자화된 주파수 계수로 바뀌게 된다. 이 주파수 계수는 심리음향 모델이 적용된 정보인 side information으로 역양자화가 수행되고, side information과 scale factor에 의해 scaling된다. scale된 576개의 주파수 계수는 IMDCT를 통해 시간영역의 sample로 바뀌게 된다. 이때 IMDCT는 시간영역에서의 aliasing을 제거하기 위해 제안된 DCT 기법이다[2]. 32개의 주파수 영역으로 나누어진 576개의 시간영역 계수는 synthesis filter bank를 통하여 합성된다.

앞의 MP3 디코딩 과정에서 비교적 계산량이 많이 요구되는 synthesis filter 부분을 DB-X 의 DSPA 모듈을 이용하여 구현하게 되며, 44.1Khz 의 샘플링 주기를 가지는 D/A 코덱과 DB-X를 연결하기 위해서는 SSI 모듈을 사용한다. 또한, SSI 와 음성 데이터가 저장되어 있는 메모리 사이를 CPU 부담없이 데이터를 전송하기 위해 DMA 모듈을 사용한다. DMA 을 이용하여 그림 2.2 와 같이 MP3 데이터를 프레임 단위로 디코딩할 때 더블 버퍼링 기법을 쉽게 사용할 수가 있다. 즉, 고정된 크기의 출력 버퍼를 이중으로 사용하여 하나의 버퍼가 D/A 버퍼로 사용될 때는 DMA를 사용하게 하여 CPU의 부담을 줄이면서 또 다른 버퍼에는 MP3 프레임을 디코딩 후 나온 PCM 데이터를 저장하게 한다. D/A 출력이 완료되면 양 버퍼를 전환하여 사용함으로써 실시간 출력이 가능하도록 하였다.

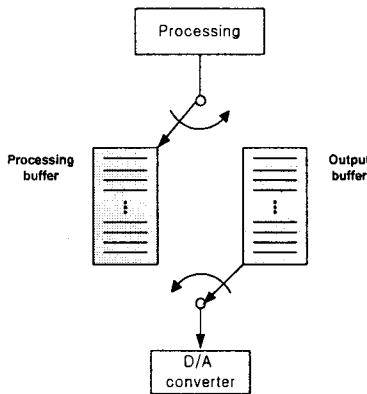


그림 2.2 출력 버퍼의 데이터 흐름도

## 2.2 DB-X를 이용한 MP3 디코더 구현

일반적으로 휴대정보단말을 위한 RISC 프로세서는 다양한 디지털 신호 처리 연산을 수행하는 데 있어 부적절한 연산구조를 가지고 있다. 그러나 Motorola의 DB-X (DragonBolt-X)는 특수한 목적의 연산 단위인 DSPA 를 가지고 있어 이러한 프로세서에 기반한 연산 구조를 최적으로 이용함으로써 디코딩 성능의 효과적으로 개선할 수 있다.

### 2.2.1 DB-X의 구조 및 특성

DB-X는 Motorola에서 개발한 RISC 타입의 마이크로컨트롤러로서 그림 2.3과 같은 블록으로 구성되어 있다. 최대 196Mhz 의 동작 속도와 USBC, BTA, LCDC, DSPA 등의 주변장치들이 원칩화되어 있으며 M341 마이크로프로세서를 코어로 MMU 등이 구현되어 있다. 또한, SSI, EIM(External Interface Module), SPI(Serial Peripheral Interface) 등의 풍부한 외부 인터페이스를 제공한다.

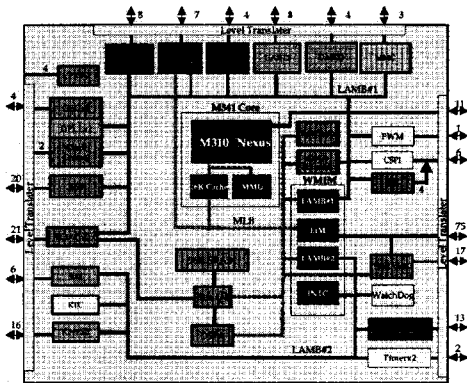


그림 2.3 DB-X

(1) DSPA : 그림 2.4 와 같은 블록 다이어그램을 가지는 DSPA 는 FFT 나 FIR 과 같은 계산량이 많이 요구되는 기능을 구현하는데 사용되어 질 수 있다.

16X16, 32X16 의 곱셈을 수행할 수 있으며, 40비트의 덧셈기를 가지고 있다. 또한, 2개의 FIFO 가 있으며 그 중 1개의 FIFO 는 데이터를 메모리에서 가져오기 위한 FIFO 이며 다른 하나의 FIFO 는 연산기의 출력을 위한 FIFO 이다.

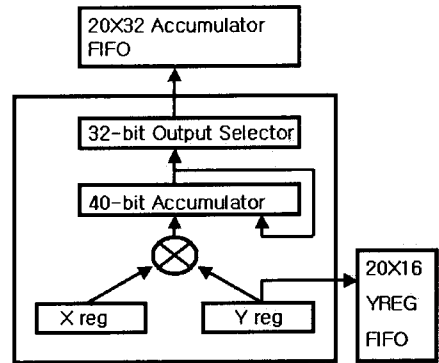


그림 2.4 DB-X 의 DSPA 모듈

(2) SSI : SSI는 코덱 등의 외부 주변 장치나 DSP, MCU 등과 통신하기 위한 full-duplex 시리얼 포트이며 주로 주기적으로 데이터를 전송하기 위해서 사용되어 진다. 독립적인 클락 발생과 프레임 싱크 신호를 가지는 독립적인 송, 수신 파트로 구성되어 지기 때문에 원하는 클락신호를 내부에서 만들수도 있고 외부에서 제공해 줄 수도 있다.

(3) DMA : DB-X에는 모두 11개의 DMA 채널이 있으며 그 중에서 1 개는 메모리와 메모리 사이에서의 DMA 채널을 위한 것이고 나머지 10개는 메모리와 SSI 등의 주변 장치를 위한 DMA 채널을 위한 것이다.

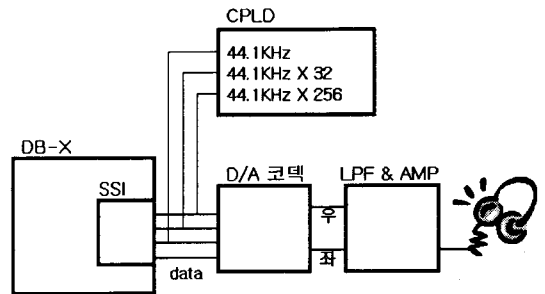


그림 2.5 MP3를 위한 H/W 구조도

### 2.2.2 MP3를 위한 H/W 구조

그림 2.5는 MP3 디코딩을 위한 H/W 블록 다이어그램이다. DB-X 내부에는 MP3 디코딩을 위한 44.1Khz 의 샘플링 주기를 가지는 D/A 코덱이 없기 때문에 외부 D/A 코덱을 DB-X 의 SSI 모듈을 이용하여 연결해야 한다. 또한, 44.1Khz 의 샘플링 주기 신호와 좌,우 스테레오를 구현하기 위한 프레임 싱크 신호, 코덱에 들어가는 메인 클락 신호 등은 DB-X 자체적으로 만들기

가 어려우므로 CPLD를 이용하여 그림 2.6처럼 외부에서 구현한다.

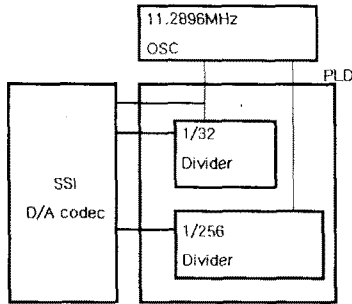


그림 2.6 MP3를 위한 CPLD 구조도

### 2.2.3 DSPA를 이용한 연산 최적화

일반적으로 DSP는 MAC과 같은 연산을 효율적으로 수행할 수 있는 하드웨어적인 구성을 가지고 있다. MP3에 있어서 연산이 가장 많은 부분인 DCT, synthesis filter bank에서 DSP 기능을 사용하면 디코딩 성능을 개선할 수 있다. 특히 synthesis filter bank는 filter들의 조합으로 구성되어 있기 때문에 그 효과가 두드러진다. MP3에서의 filter bank는 기본이 되는 저역필터를 중심으로 cosine 변조된 형태로 구성되어 있어 통과대역이 여러개 있는 것과 동일하게 볼 수 있다. 디코딩 과정에서는 이러한 각 filter bank들의 시간상의 계수들을 합성하여 오디오 신호를 만들게 된다. 계수들과의 곱 또는 덧셈으로 표현되는 이러한 필터형태는 DSP 구조에 가장 적합한 형태라고 할 수 있다. 이러한 synthesis filter bank에서의 합성과정을 그림 2.7에 나타내었다.

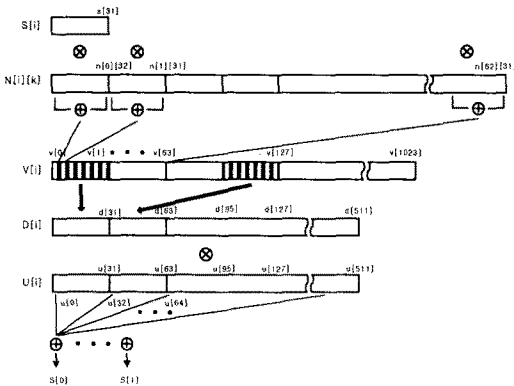


그림 2.7 synthesis filter bank의 합성 과정

32개의 새로운 subband sample,  $S[i]$ 이 들어오면 filter matrix,  $N[i][k]$ 에서 32개씩 MAC을 수행한 다음 64개의 결과를 64개 shift 시킨 버퍼,  $V[i]$ 에 저장한다. 버퍼,  $V[i]$ 에 저장된 결과들은 그림과 같이 32개씩 재배열하여 512개의 버퍼,  $D[i]$ 를 만든 다음 window,  $U[i]$ 와 곱하여  $j+32i$ 에 해당하는 결과들로부터 복원된 신호,  $S[i]$ 를 얻는다. 이러한 복잡한 합성 과정을 DSPA를 사용하면 다음과 같은 의사코드로 구현

할 수 있다.

```

SynthesisFilter()
{
    filtering()
    result_rearrange();
    windowing()
}

filtering()
{
    mac32_init();
    for(i=0;i(64;i++)
    {
        enable_DSPA();
        *(V_buffer++) = dspa_output;
    }
}

windowing()
{
    mac16_init();
    for(i=0;i(32;i++)
    {
        enable_DSPA();
        *(synthesis_result++) = dspa_out;
    }
}
    
```

DSPA 모듈을 사용하기 위하여 제어 레지스터를 설정해 주는 부분들이 mac32\_init()와 mac16\_init()이다. 이 부분에서는 피연산자들의 메모리 번지들과 연산 횟수, 반복 횟수, 다음 번지의 증가 정도 등과 같은 설정을 해주어야 한다. DSPA를 enable하면 설정한 값에 따라 연산을 고속으로 수행하고 결과를 버퍼에 저장한다.

### 3. 결 론

RISC 타입의 DB-X을 이용하여 MP3 플레이어를 성공적으로 구현할 수 있었다. 우선, 계산량이 많이 요구하는 synthesis filter를 DSPA를 이용하여 계산 속도를 빠르게 했으며 SSI 모듈을 이용하여 DB-X 와 외부 D/A 코덱 사이를 연결하였다. 또한, SSI 와 음성 데이터가 저장되어 있는 메모리 사이를 CPU 부담없이 데이터를 전송하기 위해 DMA 모듈을 사용했으며 이를 이용하여 MP3 데이터를 프레임 단위로 디코딩할 때 더블 버퍼링 기법을 쉽게 사용할 수가 있었다.

본 논문은 RISC 타입의 CPU에서 계산량이 많은 응용 프로그램을 구현할 때 사용될 수 있는 방법을 제시하고 있다.

### (참 고 문 헌)

- [1] ISO/IEC JTC1/SC29. "Information Technology - Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to About 1.5 Mbit/s - ISO/IEC 11172-3(Part 3, Audio)," 1992
- [2] J.Princen, A.Johnson, A.Bradley, "Subband/ Transform Coding Using Filter Bank Designs Based on Time Domain Aliasing Cancellation," *Proc. of the ICASSP*, pp.2161 - 2164, 1987