

일반논문-08-13-3-03

디지털 방송용 오디오 디코더의 DSP 최적화 구현

박 남 인^{a)}, 조 충 상^{b)}, 김 홍 국^{a)†}

Optimized DSP Implementation of Audio Decoders for Digital Multimedia Broadcasting

Nam In Park^{a)}, Choong Sang Cho^{b)}, and Hong Kook Kim^{a)†}

요 약

본 논문에서는 디지털 라디오와 지상파 Digital Multimedia Broadcasting (DMB)에서 사용되는 MUSICAM이라 불리는 MPEG-1/2 Layer-II와 MPEG-4 ER-BSAC 디코더를 330 MHz 클럭수를 가지고 동작하는 고정 소수점 digital signal processor (DSP) TMS320C64x+ 상에 실시간 구현한다. 오디오 디코더의 실시간 구현하기 위해, 다음과 같은 여러 단계의 최적화를 수행한다. 첫째, 메모리 공유, 데이터 타입 재설정 및 루프의 unrolling 과정을 통해, C 코드 레벨에서 최적화를 수행한다. 다음으로, 비트스트림 분석의 재구성, 합성 필터의 변경 및 합성 필터의 윈도우 계수의 재배열을 통해 알고리즘 레벨에서 최적화를 수행한다. 또한, MPEG-1/2 Layer-II 디코더의 합성필터 모듈을 linear assembly program 레벨로 치환한다. Linear assembly program 레벨로 치환하는 이유는 MPEG-1/2 Layer-II 디코더에서 합성 필터 모듈이 가장 많은 계산량을 차지하기 때문이다. 구현된 오디오 디코더의 성능 평가를 위해, 복호화 처리시간의 비율을 측정하고, 최적화된 MPEG 디코더와 레퍼런스 MPEG 디코더로 처리된 오디오 신호 사이의 root mean square (RMS)를 계산한다. 최적화 실시간 구현 결과, MPEG-1/2 Layer-II와 MPEG-4 ER-BSAC 디코더는 TMS320C64x+가 동작하는 최대 클럭 수의 3%와 11%의 사용으로 각각 동작하며, 오디오 디코더의 품질은 MPEG standard에 정의된 -77.01 dB의 조건을 모두 만족함을 확인할 수 있었다.

Abstract

In this paper, we address issues associated with the real-time implementation of the MPEG-1/2 Layer-II (or MUSICAM) and MPEG-4 ER-BSAC decoders for Digital Multimedia Broadcasting (DMB) on TMS320C64x+ that is a fixed-point DSP processor with a clock speed of 330 MHz. To achieve the real-time requirement, they should be optimized in different steps as follows. First of all, a C-code level optimization is performed by sharing the memory, adjusting data types, and unrolling loops. Next, an algorithm level optimization is carried out such as the reconfiguration of bitstream reading, the modification of synthesis filtering, and the rearrangement of the window coefficients for synthesis filtering. In addition, the C-code of a synthesis filtering module of the MPEG-1/2 Layer-II decoder is rewritten by using the linear assembly programming technique. This is because the synthesis filtering module requires the most processing time among all processing modules of the decoder. In order to show how the real-time implementation works, we obtain the percentage of the processing time for decoding and calculate a RMS value between the decoded audio signals by the reference MPEG decoder and its DSP version implemented in this paper. As a result, it is shown that the percentages of the processing time for the MPEG-1/2 Layer-II and MPEG-4 ER-BSAC decoders occupy less than 3% and 11% of the DSP clock cycles, respectively, and the RMS values of the MPEG-1/2 Layer-II and MPEG-4 ER-BSAC decoders implemented in this paper all satisfy the criterion of -77.01 dB which is defined by the MPEG standards.

Keyword : DSP implementation, DMB, audio decoder, MUSICAM, MPEG-4 ER-BSAC

I. 서론

디지털 멀티미디어 제품의 발달과 다양한 네트워크의 등장으로 인해 Digital Multimedia Broadcasting (DMB) 플레이어, 핸드폰 등과 같은 휴대용 전자제품에서 오디오 코덱의 사용에 대한 수요가 날로 증대되어 오고 있다^[1]. 이를 위해 언제 어디서나 멀티미디어 서비스를 제공받기 위해서는 멀티미디어 기술을 휴대용기기에 적용하는 임베디드(embedded) 시스템의 설계가 필수적이다. 하지만 멀티미디어 서비스는 많은 계산량과 central processing unit (CPU) 점유율을 요구하는 복잡한 멀티미디어 압축/전송/재생 기술에 바탕을 두고 있기 때문에 적은 계산량과 낮은 CPU 점유율을 갖는 멀티미디어 기술의 구현을 위한 연구가 요구된다.

본 논문에서는 현재 지상파 DMB에서 사용되고 있는 일명 MUSICAM이라 불리는 오디오 코덱인 MPEG-1/2 Layer-II와 MPEG-4 ER-BSAC의 디코더를 고정 소수점 digital signal processor (DSP)의 하나인 TMS320C64x+ 상에 실시간으로 구현을 수행한다. 우선, C-코드 최적화, 오디오 디코더 알고리즘 최적화, 메모리 사용량 최적화 등의 측면에서 최적화 과정을 수행한다. 또한 DSP에서 제공하는 다양한 라이브러리 및 내장함수를 이용한 최적화 과정도 병행된다. 특히 MPEG-1/2 Layer-II의 경우에는, linear assembly 기법을 통하여 합성 필터를 구현함으로써 보다 최적화된 DSP 구현이 가능하다. 구현된 오디오 디코더의 실시간 동작을 검증하기 위한 방법으로 오디오 디코더가 DSP 내에서 수행되는 시간을 측정한다. 구체적으로는 샘플링 주파수가 48 kHz인 스테레오 음원에 대해 384 kbit/s로 압축된 MPEG-1/2 Layer-II 비트스트림과 128 kbit/s로

압축된 MPEG-4 ER-BSAC 비트스트림이 디코딩될 때 요구되는 사이클 수를 각각 측정하여 실시간 동작 여부를 보인다. 또한, 구현된 오디오 디코더의 오디오 품질은 MPEG-1/2와 MPEG-4에 규정된 참조 디코더(Reference Decoder)와 본 논문에서 구현된 디코더로 각각 처리된 오디오 신호 사이의 root mean square (RMS) 값을 측정하여 MPEG에서 제시된 기준치를 만족하는 지를 보인다.

본 논문의 구성은 다음과 같다. 서론에 이어, II장에서는 MPEG-1/2 Layer-II와 MPEG-4 ER-BSAC의 오디오 디코더 알고리즘과 DSP TMS320C64x+에 대해 각각 설명하고, III장에서는 오디오 디코더의 DSP 상의 실시간 구현을 위한 최적화 과정에 대해 자세히 설명하기로 한다. IV장에서는 오디오 디코더의 최적화 전과 후의 계산량을 비교하여 실시간 동작 여부를 보이며 오디오 품질에 관해 MPEG의 기준치와의 비교를 수행한다. 마지막으로 V장에서는 논문의 결론을 맺도록 한다.

II. DMB용 오디오 디코더

1. MPEG-1/2 Layer-II 디코더

MPEG-1/2 Layer-II 디코더의 알고리즘은 그림 1에서와 같이 크게 세 개의 블록으로 구성된다. 패킷 해제(frame unpacking) 블록은 입력받은 MPEG-1/2 Layer-II 오디오 비트스트림으로부터 32 비트로 구성된 헤더를 분석하여 샘플링 주파수, 비트율, 모드정보, 레이어(layer) 정보 등을 얻는다. 이 정보를 바탕으로 비트스트림을 분석하기 위한 데이

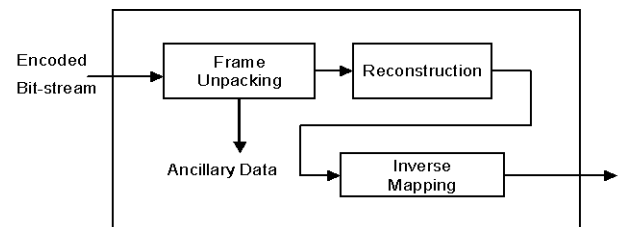


그림 1. MPEG-1/2 Layer-II 디코더 구조
Fig. 1. Structure of the MPEG-1/2 Layer-II decoder

a) 광주과학기술원 정보통신공학과
Department of Information and Communications
b) 한국전자부품연구원 SoC 연구센터
Gwangju Institute of Science and Technology 2SoC Research Center,
Korea Electronics Technology Institute
교신저자 : 김홍국(hongkook@gist.ac.kr)
※ 이 논문은 2008년도 광주과학기술원의 재원인 기본연구사업 및 2008년도 정부(교육과학기술부)의 재원으로 한국과학재단의 지원을 받아 수행된 연구(R01-2008-000-10243-0)이고 또한 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음(ITA-2008-C1090-0804-0007).

블을 선택하여 비트스트림으로 부터 양자화된 서브밴드 값, 스케일팩터 인덱스 값 그리고 스케일팩터의 양자화정보 등을 또한 분리해 낸다. 재생(reconstruction) 블록에서는 양자화된 값에 대해서 역양자화 과정을 수행한다. 즉, 역양자화 과정은 양자화된 샘플 값을 양자화 계수와 곱하고 더하는 과정을 수행한 후, 복원된 스케일팩터를 곱하여 샘플 값을 복원한다. 역맵핑(inverse mapping) 블록에서는 32차 필터뱅크로 구성된 합성 필터를 사용하여 PCM 오디오 신호를 복원한다^{[2][3]}.

2 MPEG-4 ER-BSAC 디코더

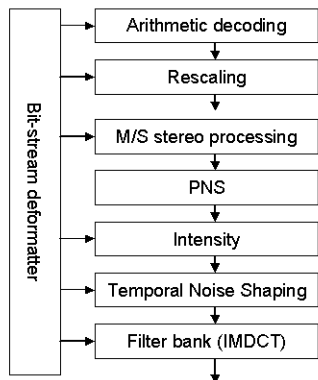


그림 2. MPEG-4 ER-BSAC 디코더 구조
Fig. 2. Structure of the MPEG-4 ER-BSAC decoder

현재 지상파 DMB 규격에 의하면 샘플링 주파수가 24, 44.1, 48 kHz 중의 하나인 오디오를 입력으로 하여 최대 비트율 128 kbit/s로 전송가능한 지상파 DMB 오디오 규격을 처리할 수 있는 MPEG-4 ER-BSAC을 사용하여 오디오 서비스를 제공하고 있다^{[4][5]}. MPEG-4 ER-BSAC은 그림 2에 보인 바와 같이 양자화를 위한 산술 부호화, 다채널 오디오 처리를 위한 M/S(Mid/Sum)-stereo, intensity stereo, perceptual noise shaping (PNS), temporal noise shaping (TNS) 블록으로 구성되어 있으며, 마지막으로 시간 축과 주파수 축 간의 변환을 위해서 modified discrete cosine transform (MDCT) 블록을 사용한다.

3. DSP TMS320C64x+ 보드의 특징

본 논문에서 사용한 고정 소수점 DSP TMS320C64x+는 330 MHz 클럭수를 가지며, 저장 공간으로는 내부 메모리, 외부 메모리, 플래시메모리가 장착되어 있다. 또한 그림 3과 같이 마이크, 헤드폰 등 외부 오디오 신호의 입출력을 제공하기 위한 TI의 AIC23 코덱 칩이 연결되어 있다. AIC32 코덱 칩의 입출력 오디오 신호는 multi-channel buffered serial port (McBSP)를 통하여 DSP 프로세서와 통신한다. 프로세서와 주변 장치의 효율적인 데이터 전송을 위해서 프로세서가 데이터 이동에 직접 관여하지 않도록 하여 프로세서의 작업량을 줄일 수 있는 enhanced direct

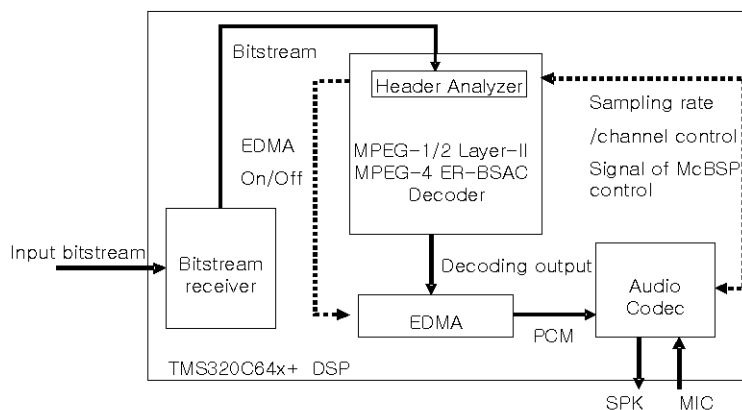


그림 3. DSP TMS320C64x+를 이용한 실시간 오디오 디코딩 시스템 구조
Fig. 3. Structure of real-time audio decoding using DSP TMS320C64x+

memory access (EDMA)를 지원한다. 뿐만 아니라, basic input/output system (BIOS)이라 불리는 실시간 운영체제를 제공한다. BIOS는 독립적인 작업들을 각각의 task를 구성함으로써 한 프로세서가 여러 작업들을 독립적으로 수행할 수 있도록 해 준다¹⁾.

III. MPEG Decoder 최적화 과정

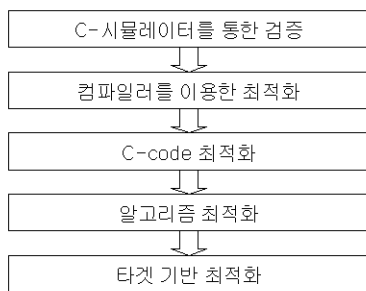


그림 4. 실시간 구현을 위한 MPEG 디코더의 최적화 과정
 Fig. 4. Optimization steps for the real-time implementation of MPEG decoders

DSP 보드에 구현된 디지털 방송용 오디오 디코더를 최적화하기 위하여 그림 4와 같이 5단계에 걸쳐 최적화 과정¹⁾을 수행한다. 먼저, MPEG에서 규정된 MPEG-1/2 Layer-II와 MPEG-4 ER-BSAC 레퍼런스 디코더를 통해 DSP 보드에 구현될 오디오 디코더의 소스코드에 대한 성능검증 작업을 수행한다. 성능검증을 완료한 후, 컴파일러에서 제공하는 최적화 기능을 사용하여 오디오 디코더의 성능을 개선한다. 여기서 컴파일러의 최적화 레벨을 파일 레벨로, 스피드/사이즈 옵션을 'speed most critical'로 그리고 프로그램 레벨 옵션을 'program mode compilation'으로 각각 정하여 TI사의 Code Composer Studio (CCS) version 3.2⁶⁾로 컴파일한다. 그 다음 단계로는, C-code 측면에서 최적화 과정을 수행한다. C-code 측면에서 최적화 과정이 완료되면, 알고리

즘을 분석한 결과를 바탕으로 알고리즘 개선을 통한 최적화 과정을 수행한다. 특히, MPEG-1/2 Layer-II 내의 합성필터는 linear assembly 기법을 적용하여 부가적으로 최적화를 수행한다. 알고리즘 최적화 과정을 완료한 후, 타겟 DSP 보드인 TI사의 TMS320C64x+의 DSP 특성을 최대한 활용하도록 하는 최적화 과정을 거친다.

1. C-code 최적화

1.1 MPEG-1/2 Layer-II 디코더의 C-code 최적화

MPEG-1/2 Layer-II 디코더에서는 윈도우 계수를 위한 배열, 합성필터 계수를 위한 배열 등 많은 배열이 사용된다. DSP 상에서 배열은 메모리의 일정 주소에서 시작하여 설정된 개수만큼 연속적으로 나열된다. 데이터를 연속적으로 접근할 경우에는 DSP는 파이프라인(pipeline) 기법을 통해 데이터에 접근하는 시간이 줄어드는 반면, 그렇지 않는 경우에는 데이터를 읽기 위한 처리시간이 증가하게 된다. 따라서 본 논문에서는 특히 for 문 등의 반복문을 수정하여 배열의 인덱스 순으로 접근하게 함으로써 DSP 처리시간을 줄일 수 있다. 이와 더불어, 배열 데이터를 순차적으로 접근하는 경우는 배열의 인덱스로 접근하는 대신 배열의 시작 주소를 기점으로 주소를 post increment 형태로 증가시키는 방법을 통해 처리시간을 줄일 수 있다.

수신된 비트스트림을 실질적으로 복호화에 사용되는 데이터 버퍼에 옮기는 작업은 일정 개수 이하의 비트스트림이 데이터 버퍼에 남아 있을 경우에만 데이터 버퍼를 채우는 작업을 수행한다. 비트스트림 버퍼로부터 데이터 버퍼를 채우기 위해서는 데이터 버퍼에 남아있는 비트스트림을 데이터 버퍼의 앞단으로 이동시키고 데이터 버퍼에 남아있는 공간을 확인하여 비트스트림 버퍼로부터 비트스트림을 데이터 버퍼에 채운다. 이런 동작은 현재 데이터 버퍼에 채워야 하는 비트스트림의 크기를 정확히 모르기 때문에 발생한다. 따라서 데이터 버퍼를 채우면서 남아있는 공간을 체크하는 조건들을 제거하기 위해서, 데이터 버퍼에 남아있는 비트수와 채워야하는 비트수 그리고 비트스트림 버퍼에서 공급할 수 있는 비트수를 미리 계산하여 데이터 이동시 확인해야 하는 조건들을 제거할 수 있다. 또한, MPEG-

1) 본 논문에서의 최적화에 따른 계산량은 DSP에서의 메모리 access 시간과 알고리즘의 수행을 위한 연산량 등을 합한 처리 속도를 모두 포함한다.

1/2 Layer-II 디코더에 입력된 비트스트림을 원하는 데이터 크기만큼 가져오는 작업은 반복적으로 이루어진다. 이러한 동작을 최적화하기 위해서 8 비트 단위로 들어오는 비트스트림을 32 비트로 확장하여 비트 인덱스와 데이터 배열의 인덱스를 조절하는 작업의 횟수를 줄일 수 있다.

타겟 보드가 고정 소수점 방식의 코드이기 때문에, DSP 보드 상에서 2 사이클이 소요되는 지수승에 대한 곱셈을, 1 사이클인 시프트 연산으로 대체한다. 그리고 16 비트 곱셈 연산과 32 비트 곱셈연산에 따라 많은 계산량 차이를 보인다는 점에 착안하여 32 비트를 두 개의 16 비트로 전환하여 상위 16 비트 곱과 하위 16 비트 곱으로 따로 수행하는 동작을 통해 32 비트 곱셈을 수행하도록 한다. 이와 같은 최적화 과정을 수행하기 위해, MPEG-1/2 Layer-II 디코더 프로그램에서 역 양자화된 샘플 값에 Q15 포맷을 적용하여 16 비트로 전환하고, 스케일팩터도 Q15 포맷을 적용하여 16 비트로 전환하여 사용한다.

1.2 MPEG-4 ER-BSAC 디코더의 C-code 최적화

MPEG-4 ER-BSAC의 각 함수는 필요한 데이터를 함수의 파라미터로 입력받는 구조를 가지고 있다. 이러한 구조는 스택(stack) 사이즈가 충분한 경우에는 특별한 문제가 되지 않지만, DSP와 같이 스택 사용이 제한된 경우에는 각 함수에 필요한 데이터를 전달하기 위해 불필요한 오버헤드가 발생한다. 이러한 문제를 해결하기 위해, 함수 호출시 스택을 사용한 파라미터 전송 방식이 아닌 직접 메모리 주소를 참조할 수 있도록 각 함수의 파라미터를 전역 변수로 수정함으로써 수행시간을 줄일 수 있다. 그리고 캐쉬(cache)를 효율적으로 사용하기 위해서, 각 변수의 데이터 선언을 각 변수의 적정 범위가 되도록 수정하였다. 특히, MPEG-4 ER-BSAC 디코더는 여러 가지 블록들로 구성되어 있으며, 각 블록은 서로 다른 기능을 수행한다. 그러므로 각 블록의 기능을 수행하기 위해 선언된 메모리는 대부분 다른 블록을 처리할 때는 사용되지 않는다. 이러한 구조는 메모리 공간이 충분하거나 캐쉬가 충분한 경우는 필요한 데이터를 캐쉬에 상주하도록 하여 사용할 수 있지만, 제한된 조건에서는 캐쉬의 내용을 계속 갱신해야 하는 문제가 발생되어 수행시간 증가를 유발할 수 있다. 이러한 문제점

을 해결하기 위해서 이전 블록에서 사용된 변수 즉 메모리를 현재 수행중인 블록에서 사용하는 방식을 사용하였다. 결과적으로 한번 캐쉬에 저장된 메모리 공간을 계속 유지하여 사용할 수 있기 때문에 캐쉬의 내용을 갱신하기 위해 요구되는 수행시간을 감소시켜 성능을 개선할 수 있다.

또한, MPEG-4 ER-BSAC 디코더에서는 각 계층에 따른 많은 배열을 사용한다. 배열은 메모리의 일정 주소에서 시작하여 연속적으로 선언된 변수형이 설정된 개수만큼 나열된다. 프로그램에서 데이터를 연속적으로 접근하지 않은 경우에는 for 문 등의 반복문을 수정하여 배열의 인덱스 순으로 데이터를 접근함으로써 수행시간을 줄일 수 있다. 또한, 배열 데이터를 순차적으로 접근하는 경우는 배열의 인덱스로 접근하는 대신, 배열의 시작 주소를 기점으로 주소를 post increment 형태로 증가시키는 방법으로 최적화를 수행한다. 또한, 상술한 바와 같이 MPEG-4 ER-BSAC 디코더에서도 8 비트 단위로 들어오는 비트스트림을 32 비트로 전환하고, 원하는 데이터 비트스트림을 가져가는 작업 또한 8 비트 범위에서 32 비트로 확장하여 비트 인덱스와 데이터 배열의 인덱스를 조절하는 작업의 횟수를 줄일 수 있다. 그리고 산술 양자화 부분은 비트스트림으로 부터 읽을 추가 정보 및 주파수 측 데이터를 산술 양자화하기 위하여 레이어 개수만큼 for 루프를 반복적으로 돌면서 산술 복호화 함수를 지속적으로 호출하게 된다. 이와 같이 루프에서 함수를 호출하면 파이프라인이 형성되지 않는다. 그러므로 최대한 파이프라인이 형성되도록 하기 위해서는 주기적으로 호출되는 함수가 함수 호출에 의해 수행되는 것이 아니라, 루프 내에서 직접 수행되는 형태로 전환하도록 개선해야 한다.

2. 알고리즘 최적화

2.1 MPEG-1/2 Layer-II 디코더 알고리즘 최적화

MPEG-1/2 Layer-II 디코더 과정에서 가장 많은 계산량이 요구되는 모듈인 합성필터는 다상 변조 필터 बैं크 형태를 갖고 있으며, 매 프레임 당 72번씩 반복해서 수행된다. 그러므로 합성필터 모듈의 계산량을 최대한 줄임으로서 복호화 과정의 수행 시간을 상당히 단축시킬 수 있다. 합성필

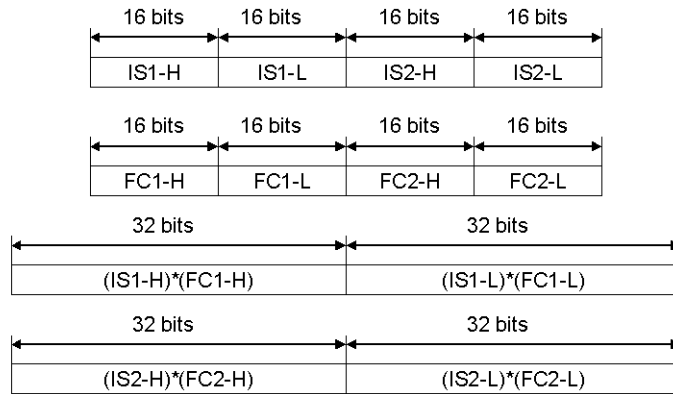


그림 6. Linear assembly programming을 이용한 합성 필터에서 처리 구조
 Fig. 6. Structure of processing a synthesis filter by using the linear assembly programming

터에 사용되는 필터계수는 식 (1)로 계산된다.

$$N_{ik} = \cos\left[(16+i)(2k+1)\frac{\pi}{64}\right] \quad (1)$$

for $0 \leq i \leq 64, 0 \leq k \leq 31$,

여기서 N은 합성필터의 계수로써 행렬 형태를 취하고 있으며, i와 k는 행렬의 인덱스이다. 또한 식 (1)로 부터 계산된 필터계수로 부터 합성필터는 식 (2)과 같이 표현된다.

$$V[i] = \sum_{k=0}^{31} N_{ik} S_k \quad \text{for } 0 \leq i \leq 64, \quad (2)$$

여기서 S_k 는 k번째 입력 샘플 값이고 $V[i]$ 는 i번째 필터링된 결과 값이다. 필터계수가 대칭성을 가지는 코사인 함수를 사용하고 있기 때문에, 이 특성을 이용하여 32개의 $V[i]$ 값만을 계산하여 64개의 $V[i]$ 로 확장할 수 있다^[6].

2.2 MPEG-4 ER-BSAC 디코더 알고리즘 최적화

MPEG-4 ER-BSAC 디코더는 각 샘플에 대해 시작 지점과 폭에 대한 값을 갱신하게 된다. 갱신된 폭을 이용하여 다음 샘플 복호화를 위해 필요한 데이터 사이즈를 결정하게 된다. 그림 5(a)는 갱신된 폭을 이용하여 읽을 데이터 사이즈를 결정하는 방법에 대한 예를 보여 준다. 그림 5(a)에서와 같이 갱신된 폭이 2의 배수 단위로 나누어진 구간

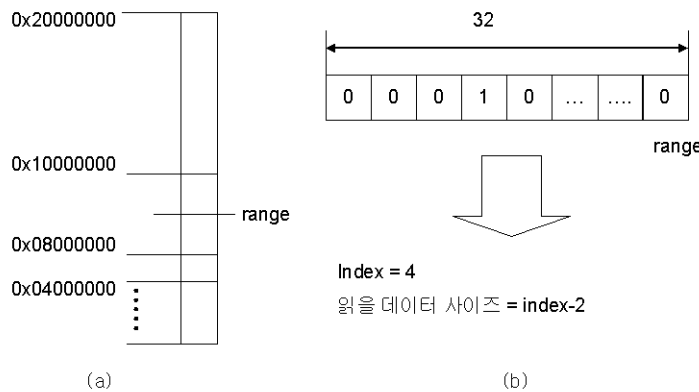


그림 5. 읽을 데이터 크기 결정 방법의 예; (a) 기존 방법 (b) 제안된 방법
 Fig. 5. An example of determining data size to read; (a) the conventional method and (b) the proposed method

어디에 포함되는지 확인한 후, 양 인덱스가 2이므로 다음 복호화를 위해 2 비트를 읽는다. 이러한 작업은 매 샘플마다 수행되므로 BSAC 복호화 알고리즘에서 상당히 복잡하게 된다. 또한 어느 위치에 속하는 지를 찾기 위해서는 매번 각 구간을 나타내는 값과 폭과의 일대일 비교가 불가피하다. 이러한 구조를 개선하기 위해 그림 5(a)는 그림 5(b)와 같은 구조로 변환된다.

3. 타겟 기반 최적화

3.1 MPEG-1/2 Layer-II 디코더의 타겟 기반 최적화

MPEG-1/2 Layer-II 디코더에서 가장 많은 계산량을 요구하는 합성필터를 개선하기 위해 합성필터에서 반복적으로 수행되는 곱의 합(sum of product) 부분에 linear assembly 기법을 적용한다. 어셈블리어 명령어 중에서 데이터를 읽어 들이는 load 명령어와 루프 명령어에서 다음 사이클 시작 위치로 이동시키는 branch 명령어는 각각 5 사이클과 6 사이클의 처리 시간이 요구된다^{[7][8]}. 따라서 성능 개선을 위해서는 load 명령어와 branch 명령어의 사용을 최대한 줄여야 한다. 본 논문에서는 load 명령어와 branch 명령어 사용 횟수를 최대한 줄이기 위해 합성필터링 과정을 그림 6과 같은 구조로 구성한다. 입력 샘플 값 (IS)과 필터 계수 (FC)가 16 비트 사이즈를 갖고 있으므로 64 비트 단위로 값을 읽어 들일 수 있는 LDDW 명령어를 사용함으로써 이를 구현할 수 있다. 즉, LDDW 명령어를 사용하면 64 비트 단위로 입력 샘플과 필터 계수들을 한 번에 메모리에서 읽어 들일 수 있게 된다. 필터 계산은 32 비트 단위 블록으로 곱셈 연산을 하는 PROD2 명령어를 사용하여 입력 샘플과 필터계수의 32 비트 단위 블록의 상위와 하위에 대해서 필터 계산을 취하도록 표현된다. 최종적인 결과에서는 상위와 하위의 필터 계산 값을 더하여 저장하는 구조로 구성된다. 위와 같은 구조를 사용하여 필터 처리 과정의 루프의 반복 횟수가 C-언어 기반의 합성 필터에서의 루프의 반복 횟수에 비해 그 횟수를 반으로 감소시킬 수 있다. 그러므로 루프의 반복 횟수의 감소로 데이터를 읽어 들이는 load 명령어와 다음 사이클 시작 위치로 이동시키는 branch 명령어의 사용 횟수가 감소하여 필터 처리에 요구되는 사이클 수를

감소시킬 수 있다. 또한 필터 계산에 DOTP2 명령어를 선택하여 적용함으로써 4번의 SUM 명령어와 4번의 MPY 명령어를 호출해야 처리가 가능하던 것을 DOTP2 명령어를 2번 호출하는 것으로 처리할 수 있게 된다. 게다가 DOTP2 명령어에 입력되는 값이 서로 독립적이므로 2번 호출되는 DOTP2 명령어가 파이프라인으로 구성되어 필터의 처리 성능을 향상시킬 수 있다^[9].

3.2 MPEG-4 ER-BSAC 디코더의 타겟 기반 최적화

MPEG-4 ER-BSAC 디코더에서 사용되는 inverse MDCT (IMDCT)는 fast Fourier transform (FFT)과 같은 DSP 라이브러리와 내장 함수를 사용하여 최적화된다^[7]. 즉, DSP에서 제공하는 최적화된 FFT 라이브러리 함수로 C-함수의 FFT를 대체함으로써 보다 최적의 성능을 얻을 수 있다. 이를 위해, 컴파일러에 포함되어 있는 DSP 라이브러리를 추가하고, 트위들 벡터(twiddle vector)를 입력으로 추가함으로써 속도를 향상시킨다. 또한, 32 비트 연산을 위한 사칙연산 함수들을 다음과 같이 개선할 수 있다. 일반적으로 32 비트 기준의 사칙 연산 함수는 32 비트를 상위 16 비트와 하위 16 비트를 나누어 연산하며, 연산 결과에 대한 오버플로우와 언더플로우를 체크하는 구조를 가지고 있다. 그러나 고정 소수점 MPEG-4 ER-BSAC 디코더는 연산 결과의 최대값이 1 이하인 부동 소수점 ER-BSAC 디코더로 부터 변환된 것이므로 사칙 연산의 결과에서 오버플로우와 언더플로우를 확인하지 않으면서 이를 DSP의 assembly 함수를 사용가능하도록 하는 내장 함수로 대체함으로써 계산량을 줄일 수 있다. 마지막으로 IMDCT 이후에 적용된 윈도우 및 overlap-and-add를 수행하는 부분에서 윈도우와 데이터 배열의 곱에 DSP의 vector 라이브러리를 사용함으로써 branch가 많이 걸리는 배열 곱의 속도를 줄일 수 있다.

IV. 성능 평가

최적화된 MPEG-1/2 Layer-II 디코더와 MPEG-4 ER-BSAC 디코더의 계산량을 측정하기 위해, 샘플링 주파수 48 kHz, 스테레오, 비트율 128 kbit/s인 BBC 방송용 Di-

gital Audio Broadcasting (DAB) 비트스트림과 샘플링 주파수 48 kHz, 스테레오, 128 kbit/s 비트율, 레이어 수가 48인 비트스트림을 conformance test 파일로 사용하였다^[4]. 이 conformance test 파일을 가지고 CCS version 3.2의 프로파일링 기능을 사용하여, 1초 동안의 오디오 샘플을 복호화과정에 소요되는 사이클 수 (MCPS: Mega Cycles Per Second)를 측정하였다.

표 1은 본 논문에서 사용한 DSP TMS320C64x+의 메모리 구성 및 실제 사용한 메모리 구성을 보여 준다. 본 논문에서의 DSP 처리 속도는 실제 연산을 위한 계산량과 메모리로부터 데이터 등을 읽어오는 데 필요한 시간을 모두 포함한다. 따라서 실제 사용된 메모리 구성이 다를 경우 디코더에서의 연산을 위한 계산량은 동일한 반면 처리속도 면에서 서로 다른 결과를 보일 수 있다. 예를 들어 stack을 4 KB만을 사용할 경우에는 프로그램 내의 subroutine 시 필요한 파라미터의 수에 따라 다른 처리속도를 보일 수 있으며, 적은 량의 cache를 사용할 경우에도 메모리의 불필요한 load에 의해 처리속도가 증가할 수 있다. 표 1에서 보인 바와 같이, MPEG-1/2 Layer-II 디코더는 program memory (PM)와 data memory (DM)를 각각 38.25 KB와 37.48 KB를 사용해서 구현되었다. 한편, MPEG-4 ER-BSAC 디코더의 경우, BSAC 알고리즘은 순차수행이 필요하게 되어 이 부분에 program 사이즈가 증가하게 되었으며 따라서 MPEG-1/2 Layer-II 디코더보다 많은 PM이 요구되어 58.50 KB가 필요하게 되었다. 그리고 알고리즘 수행에 필요한 table들의 증가로 인해 DM도 MPEG-1/2 Layer-II 디

코더보다 많은 65.86 KB를 사용하게 되었다. 실제 사용한 메모리의 스택의 크기는 비디오 처리를 위해 4 KB를 사용하였다.

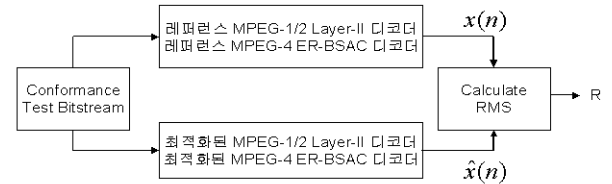


그림 7. 실시간으로 구현된 디코더의 성능을 평가하기 위한 RMS 측정 과정
Fig. 7. Procedure of obtaining RMS values for the performance evaluation of the real-time implementation

최적화된 MPEG-1/2 Layer-II 디코더와 MPEG-4 ER-BSAC 디코더의 오디오 품질을 측정하기 위해 그림 7과 같이 환경을 구축하고 root mean square (RMS) 값을 계산하였다. 그림 7에서 보는 바와 같이 conformance test 비트스트림 파일로부터 레퍼런스 MPEG-1/2 Layer-II 혹은 MPEG-4 ER-BSAC 디코더에 의해 복호화된 PCM 파일을 생성하였다. 이렇게 복호화된 파일과 conformance test에서 제공하는 레퍼런스 파일을 식 (3)을 이용하여 RMS 값을 구하였다. 마지막으로 RMS 값은 MPEG 표준에서 제시한 -77.01 dB보다 작은 지를 확인하였다. 즉 RMS를 구하는 식은

$$RMS = 20 \log_{10} \sqrt{\frac{1}{N} \sum_{n=0}^{N-1} (x(n) - \hat{x}(n))^2} \quad (3)$$

표 1. DSP TMS320C64x+의 메모리 구성 및 실시간 구현에 사용된 메모리 구성
Table 1. Memory configuration of DSP TMS320C64x+ and that used for the real-time implementation of the decoders

메모리 구성	<ul style="list-style-type: none"> • Stack : 16 KB • Cache <ul style="list-style-type: none"> - program : 32 KB - Data : 32 KB - L2 : 32 KB 	<ul style="list-style-type: none"> • Program <ul style="list-style-type: none"> - External memory • Data <ul style="list-style-type: none"> - External memory
실시간 구현에 사용된 메모리 구성	<ul style="list-style-type: none"> • Stack : 4 KB • Cache <ul style="list-style-type: none"> - program : 32 KB - Data : 32 KB - L2 : 32 KB 	<ul style="list-style-type: none"> • MPEG-1/2 Layer-II decoder <ul style="list-style-type: none"> - Program memory: 38.25 KB - Data memory: 37.48 KB • MPEG-4 ER-BSAC decoder <ul style="list-style-type: none"> - Program memory: 58.50 KB - Data memory: 65.86 KB

여기서 N 은 오디오 신호에서의 샘플 수이며, $x(n)$ 과 $\hat{x}(n)$ 은 그림 7에 보여준 바와 같이 conformance test 비트 스트림을 레퍼런스 MPEG-1/2 Layer-II 혹은 MPEG-4 ER-BSAC 디코더에 의해 복호화된 오디오 신호와 구현된 MPEG-1/2 Layer-II와 MPEG-4 ER-BSAC 디코더에 의해 복호화된 오디오 신호를 각각 나타낸다.

1. MPEG-1/2 Layer-II 디코더 성능 평가

표 2는 MPEG-1/2 Layer-II 디코더의 각 단계별 최적화 전·후의 계산량을 측정하는 것이다. MPEG-1/2 Layer-II 디코더는 C-코드 최적화, 알고리즘 최적화 및 타겟 기반 최적화를 수행함으로써 계산량이 각각 65.62%, 81.99%, 69.77%로 감소하였다. 표 3은 최적화 후 레퍼런스 MPEG-1/2 Layer-II 디코더 대비 계산량 감소량과 CPU 점유율을 나타낸 것이다. 표 3에서 보는 바와 같이 레퍼런스 디코더는 486 MCPS이므로 타겟 보드인 TMS320C64x+의 최대 동작 사이클 수인 330 MHz를 초과하는 것을 알 수 있었다. 이에 반해, 실시간 구현을 위해 최적화를 수행한 결과, MPEG-1/2 Layer-II 디코더의 계산량은 9.1 MCPS로 DSP 최대 동작 사이클의 2.76%만 사용하게 되므로 실시간으로 동작하는 것을 확인할 수 있었다.

표 2. MPEG-1/2 Layer-II 디코더의 최적화에 따른 계산량 비교
Table 2. Complexity comparison according to each optimization step applied to the MPEG-1/2 Layer-II decoder

Optimization Step	최적화 전	최적화 후
C-코드 최적화	486.0	167.1
알고리즘 최적화	167.1	30.1
타겟 기반 최적화	30.1	9.1

표 3. MPEG-1/2 Layer-II 디코더의 최적화 전·후 성능 비교
Table 3. Comparison of decoding time of the MPEG-1/2 Layer-II decoder before and after performing the optimization process

Reference Decoder	구현된 Decoder		계산량 감소율
	계산량 (MCPS)	CPU 점유율	
486.0	9.1	2.76%	98.13%

표 4는 최적화 전·후 MPEG-1/2 Layer-II 디코더 각 블록

의 계산량을 나타낸 것이다. 최적화 전의 합성필터는 463.4 MCPS의 계산량이 소요되었으나, 최적화 이후 5.4 MCPS로 약 98.83%의 계산량이 상대적으로 감소되었음을 알 수 있었다. 또한 구현된 MPEG-1 Layer-II 디코더와 MPEG-2 Layer-II 디코더의 RMS 평균을 측정하는 결과, 표 5와 같이 각각 -86.35 dB와 -85.01 dB로 -77.01 dB보다 작은 RMS 값을 갖는 것을 표 5에서 확인할 수 있었다. 이러한 결과로부터 DSP에 구현된 MPEG-1/2 Layer-II 디코더가 정상적으로 비트스트림을 실시간으로 복원하고, 실시간 구현시 성능열화가 없음을 확인할 수 있었다.

표 4. 최적화 전·후 MPEG-1/2 Layer-II 디코더의 각 블록의 계산량 비교
Table 4. Comparison of the decoding time for each processing block of the MPEG-1/2 Layer-II before and after performing the optimization process

Processing Block	최적화 전 (MCPS)	최적화 후 (MCPS)
Decoding Initialization	2.2	0.2
Frame Unpacking	6.2	2.2
De-quantization	14.2	1.3
Synthesis Filtering	463.4	5.4
Total	486.0	9.1

표 5. MPEG-1/2 Layer-II 디코더의 RMS 비교
Table 5. RMS Comparison of the MPEG-1/2 Layer-II decoder

Audio Decoder	평균 RMS (dB)
MPEG Reference	-77.01
최적화된 MPEG-1 Layer-II	-86.35
최적화된 MPEG-2 Layer-II	-85.01

2. MPEG-4 ER-BSAC 디코더 성능 평가

표 6은 MPEG-4 ER-BSAC 디코더의 각 단계별 최적화 전·후의 계산량을 측정하는 것이다. MPEG-4 ER-BSAC 디코

표 6. MPEG-4 ER-BSAC 디코더에서 최적화에 따른 계산량 비교
Table 6. Complexity comparison by each optimization step applied to the MPEG-4 ER-BSAC decoder

Optimization Step	최적화 전	최적화 후
C-코드 최적화	458.0	371.0
알고리즘 최적화	371.0	360.0
타겟 기반 최적화	360.0	34.2

표 7. MPEG-4 ER-BSAC 디코더의 최적화 전·후 성능 비교
Table 7. Comparison of decoding time of the MPEG-4 ER-BSAC decoder before and after performing the optimization process

Reference Decoder	최적화된 Decoder		계산량 감소율
계산량 (MCPS)	계산량 (MCPS)	CPU 점유율	
458.0	34.2	10.36%	92.53%

더의 계산량은 C-코드 최적화, 알고리즘 최적화 및 타겟 기반 최적화를 수행함으로써 각각 18.99%, 2.96%, 90.50%로 감소하였다. 표 7은 최적화 후 레퍼런스 MPEG-4 ER-BSAC 디코더 대비 계산량 감소량과 CPU 점유율을 나타낸 것이다. 표 7에서 보는 바와 같이 레퍼런스 디코더는 458 MCPS이므로 타겟 보드인 TMS320C64x+의 최대 동작 사이클 수인 330 MHz를 초과하였다. 그러나 실시간 구현을 위해 최적화를 수행한 결과, MPEG-4 ER-BSAC 디코더의 계산량은 34.2 MCPS이므로, DSP 최대 동작 사이클인 330MHz의 10.36%만 사용하게 되어, 실시간으로 동작하는 것을 확인할 수 있었다.

표 8은 MPEG-4 ER-BSAC 디코더의 최적화 전·후 각 블록에서 요구되는 계산량을 나타낸 것이다. 최적화 전의 산술 디코딩과 IMDCT는 각각 123.4 MCPS와 296.1 MCPS의 계산량이 소요되었으나, 최적화 이후 32.4 MCPS와 1.5 MCPS로 약 73.74% 및 99.49%의 계산량이 감소되었다. 또한 표 9과 같이, MPEG-4 ER-BSAC 디코더의 RMS 평균값은 -91.87dB로 MPEG에 규정된 RMS 값인 -77.01 dB보다 작아, 실시간 구현시 성능열화가 없음을 확인할 수 있었다.

표 8. 최적화 전·후 MPEG-4 ER-BSAC 디코더의 각 블록의 계산량 비교
Table 8. Comparison of the decoding time of each processing block of the MPEG-4 ER-BSAC decoder before and after performing the optimization process

Processing Block	최적화 전	최적화 후
Arithmetic Decoding	123.4	32.4
MS-Stereo	0.9	0.2
TNS	37.6	0.1
IMDCT	296.1	1.5
Total	458.0	34.2

표 9. MPEG-4 ER-BSAC 디코더의 RMS 비교
Table 9. RMS comparison of the MPEG-4 ER-BSAC decoder

Audio Decoder	평균 RMS (dB)
MPEG Reference	-77.01
최적화된 MPEG-4 ER-BSAC	-91.87

V. 결 론

본 논문에서는 TMS320C64x+ DSP를 이용하여 디지털 방송용 오디오 코덱인 MPEG-1/2 Layer-II 디코더와 MPEG-4 ER-BSAC 디코더를 실시간으로 구현하였다. 이를 위해, C-코드 최적화, 알고리즘 최적화 그리고 타겟 기반 최적화 등의 최적화 과정을 수행하였다. 그 결과 MPEG-1/2 Layer-II 디코더와 MPEG-4 ER-BSAC 디코더는 TMS320C64x+가 동작하는 최대 클럭수인 330 MHz의 3.02%와 10.35%로 각각 동작되어 실시간 범위에 있음을 확인하였다. 또한, MPEG-1/2 Layer-II 디코더와 MPEG-4 ER-BSAC 디코더의 오디오 품질은 각각 -85.01 dB와 -91.87 dB로 MPEG에서 정의한 기준인 -77.01 dB 보다 작아 실시간 구현시 성능 열화가 없음을 확인할 수 있었다.

참 고 문 헌

- [1] Y. F. Dehery, "A MUSICAM source CODEC for digital audio broadcasting and storage," in Proc. ICASSP, vol. 5, pp. 3605-3608, May 1991.
- [2] M. Bosi and R. E. Goldberg, Instruction to Digital Audio Coding and Standards, Kluwer Academic Publishers: Norwell, MA, 2003.
- [3] ISO/IEC 11172-3, Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s part 3: Audio, Apr. 1996.
- [4] ISO/IEC 14496-4, Information technology - Coding of audio-visual objects part 4: Conformance testing bitstream, Feb. 2004.
- [5] S. Lee, M. Choi, J. Kim, D. Kim, N. Eum, and H. Jung, "The MPEG-4 BSAC audio decoder implementation in terrestrial DMB receiver," in Proc. IEEE Consumer Electronics, pp. 257-258, Jan. 2006.
- [6] J. Huang and G. Gu, "A direct approach the design of QMF banks via frequency domain optimization," IEEE Trans. Signal Processing, vol. 46, no. 8, pp. 2131-2138, Aug. 1988.
- [7] Texas Instruments, TMS320C6000 Peripherals Reference Guide,

Feb. 2001.

[8] R. Chassaing, Digital Signal Processing and Applications with the C6713 and C6416 DSK, Wiley-Interscience, 2005.

[9] Texas Instruments, TMS320C6000 CPU and Instruction Set

Reference Guide, Oct. 2000.

[10] ISO/IEC 11172-4, Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s part 4: Compliance test, Mar. 1995.

저 자 소 개



박 남 인

- 2000년 3월 ~ 2007년 2월 : 광운대학교 전자통신공학과 (공학사)
- 2007년 3월 ~ 현재 : 광주과학기술원 정보통신공학과 석사 과정
- 주 관심분야 : 음성 및 오디오 코딩



조 충 상

- 1999년 3월 ~ 2006년 2월 : 수원대학교 전자공학과 (공학사)
- 2006년 2월 ~ 2008년 2월 : 광주과학기술원 정보통신공학과 (공학석사)
- 2008년 3월 ~ 현재 : 전자부품연구원 SoC 센터
- 주 관심분야 : 오디오 코딩, 프레임 손실은닉, 3D 오디오



김 흥 국

- 1984년 3월 ~ 1988년 2월 : 서울대학교 제어계측공학과 (공학사)
- 1988년 3월 ~ 1990년 2월 : 한국과학기술원 전기 및 전자공학과 (공학석사)
- 1990년 3월 ~ 1994년 2월 : 한국과학기술원 전기 및 전자공학과 (공학박사)
- 1990년 3월 ~ 1998년 2월 : 삼성종합기술원 전문연구원
- 1998년 2월 ~ 1998년 9월 : (주)엠씨테크놀로지 선임연구원
- 1998년 11월 ~ 2003년 6월 : AT&T Labs-Research Senior Technical Staff Member
- 2003년 8월 ~ 현재 : 광주과학기술원 정보통신공학과 부교수
- 주 관심분야 : 멀티미디어 신호처리, 실시간 User Interface, 음성 및 오디오 코딩