

TMS320C6701 DSP 용 MPEG-4 오디오 HVXC 부호기의 최적화 라이브러리 개발

Library Optimization of the MPEG-4 Audio HVXC Coder using TMS320C6701 DSP

나 훈*, 이지웅*, 강경옥**, 임영권**, 홍진우**, 정대권*

Hoon Na, Ji Woong Lee*, Kyeong Ok Kang**, Young Kwon Lim**, Jin Woo Hong**, Dae Gwon Jeong*

* 한국항공대학교 항공전자공학과

** ETRI 방송기술연구부

요 약

MPEG-4 오디오 부호기의 일부인 HVXC(Harmonic and Vector eXcitation Coding) 부호기는 음성
의 무성음 구간에서는 CELP 코덱, 유성음 구간에서는 MBE 코덱을 이용하여 부호화하는 구
조로서, 많은 연산량을 필요로 하여 범용 DSP 를 이용한 실시간 구현의 장애요소로 작용한다.
본 논문에서는 TMS320C6701 DSP 를 이용하여 많은 연산 시간을 요하는 함수들에 대한 C 언어
및 어셈블리 레벨의 최적화를 수행하여 HVXC 함수들의 실행시간을 단축하고 이를 라이브러
리화 하여 실시간 구현에 이용가능 하도록 하였다.

1. 서 론

MPEG-4 오디오는 2~6kbps 의 비트율을 갖는
parametric 부호기, 4~24kbps 의 CELP 부호기,
16kbps 이상의 비트율을 갖는 General
Audio(GA)부호기로 구성되며, parametric 부호기
는 음성 신호를 압축하기 위한 HVXC(Harmonic
and Vector eXcitation)와 비음성 신호를 압축하기
위한 HILN(Harmonic and Individual Lines plus
Noise)의 두가지 부호기로 이루어진다. HVXC
부호기는 음성신호를 유/무성음 구간으로 분류
한 후 유성음 구간에서는 MBE(Multi Band
Excitation) 부호기를 이용하여 부호화하며, 무
성음 구간은 CELP(Code-Excited Linear
Prediction) 부호기에서 사용되는 방법과 같은

랜덤 코드북을 이용하여 벡터 양자화한다[1].
HVXC 의 이러한 구조로 인하여 LPC 분석,
FFT/IFFT, 유/무성음 결정, 코드북 탐색, 스펙트
럼 포락선의 양자화 알고리즘 등에 많은 연산
량을 필요로 하므로 범용의 DSP 를 이용한 실
시간 구현시 최적화 과정을 필요로 한다.

본 논문에서는 TMS320C6701 DSP 를 사용하
여 HVXC 부호기를 실시간으로 구현하기 위해,
HVXC 에 대한 각 함수별 machine cycle 을 구한
후 많은 연산 시간을 요하는 함수들을 결정하
였다. 이들 함수를 대상으로 DSP 의 하드웨어
구조에 적합하도록 software pipelining, loop
unrolling, 워드/더블워드 접근 기법 등의 방법
을 사용하여 C 언어 레벨의 최적화를 수행한

이터의 처리 속도를 높인다.

표 1. HVXC 최적화 대상 함수 분석

Functional analysis	cycles per 1 execution	연산량(%)
Pitch Estimation	2,423,522	41.7
V/UV decision	35,317	0.1
VQ of spectral envelope	2,020,386	9.4
VXC	431,832	21.4
Etc.	509,137	5.0
Total	5,420,194	77.6

● Software pipelining

루프내에서 명령어의 스케줄을 재 조정해 다수의 명령어를 병렬로 처리한다.

● Loop unrolling

pipelining 을 효과적으로 수행하기 위해 루프를 확장하여 명령어를 병렬로 처리할 수 있는 구조로 조정한다.

● Using intrinsic functions

C6x 컴파일러가 제공하는 내부 함수를 사용해 C 코드를 TMS320C6701 에 최적화된 구조로 표현한다.

● Using single floating-point precision

배정도 실수형 데이터는 두개의 32bit word 를 사용하므로 메모리 용량과 access time 이 증가하게 된다. 따라서 HVXC 부호기의 정밀도가 감소되지 않는 범위에서 단정도 실수형으로 변환하여 사용한다[4][6]. 표 2 는 C 레벨 최적화 결과를 나타낸다.

표 2. C 레벨 최적화 결과

Functional analysis	Unoptimized	Optimized	optimization ratio
Pitch Estimation	2,423,522	1,416,257	41.6%

V/UV decision	35,317	11,922	66.3%
VQ of spectral envelope	2,020,386	1,187,279	41.2%
VXC	431,832	184,580	57.3%
Etc.	509,137	346,514	31.9%
Total	5,420,194	3,146,552	41.9%

$$optimization\ ratio = \frac{Unoptimized - Optimized}{Unoptimized} \times 100$$

4.2 어셈블리 레벨 최적화

C 레벨 최적화 단계에서 최적화 되지 않은 코드 부분들을 linear assembly 를 사용하여 함수들을 재 프로그램해야 하며, 특히 함수의 연산량의 대부분을 차지하는 loop kernel 의 병렬로 처리되는 명령어 수를 증가시키기 위한 최적화를 수행한다.

● Word & double word access

16bit 및 32bit 데이터를 word 와 double word 로 메모리 정렬한 후 32bit load word LDW 명령어와 레지스터를 쌍으로 사용하는 load double word LDDW 명령어를 사용하여 16bit 및 32bit 데이터를 병렬처리를 한다.

● Software pipelining

C 레벨 최적화에서 수행 한 pipelining 을 개선하기 위해 각 명령어의 dependency 를 재 조정한다.

● Modulo scheduling of multicyle loops

multicycle 루프에서 같은 자원의 공유로 인해 발생하는 병렬 처리의 제한을 해결한다.

● Redundant load elimination

load 명령어는 많은 cycle 을 차지하므로, 이 명령어의 효율적인 사용은 최적화에 많은 영향을 준다.

● Determining the minimum iteration interval

루프의 최소 반복 횟수를 지정하여 루프 unrolling 을 효과적으로 수행하게 한다[4][5].

표 3은 어셈블리 레벨 최적화 결과를 나타낸다.

5. 실험결과 및 분석

최적화된 HVXC 부호기를 TMS320C6701 DSP를 사용하여 167 MHz에서 시뮬레이션 한 결과를 표 4에 나타내었다. 최적화된 라이브러리를 사용한 HVXC의 프레임당 부호화 속도가 55.3msec로, 최적화 되지 않은 경우에 비해 84.8% 단축 되었음을 알 수 있다.

표 3. 어셈블리 레벨 최적화 결과

Functional analysis	Unoptimized	Assembly Optimized	optimization ratio
Pitch Estimation	2,423,522	1,348,728	44.3%
VUV decision	35,317	11,922	66.3%
VQ of spectral envelope	2,020,386	1,169,706	42.1%
VXC	431,832	167,450	61.2%
Etc.	509,137	323,866	36.4%
Total	5,420,194	3,021,672	44.3%

표 4. TMS320C6701 DSP에서 HVXC 실행 결과

	cycles /frame	msec /frame	optimization ratio
Unoptimized	58,283,744	364.3	84.8%
Assembly level optimized	8,840,568	55.3	

6. 결 론

MPEG-4 음성신호 부호화기인 HVXC는 유성음은 MBE 부호기, 무성음은 CELP 부호기를 사용하는 구조로 인해 알고리즘의 복잡도가 매우 크다. 본 논문에서는 HVXC를 실시간으로 구현하기 위해 전체 부호기의 연산량의 대부분을 차지하는 함수들을 최적화 대상으로 선정 한 후, C언어 레벨 및 어셈블리 레벨의 최적화를 수행 하였고 이를 TMS320C6701 DSP를 사용해

구현하였다. 그 결과 최적화되지 않은 경우에 비해 부호화 속도가 84.8% 단축되었으며 이를 라이브러리화하여 실시간 구현에 이용 가능하도록 하였다[7].

참 고 문 헌

- [1] ISO/IEC JTC1/SC29/WG11 MPEG International Standard CD 14496-3 Information Technology – Coding of Audiovisual Objects, Part 3: Audio, subpart 2 : Parametric Coding, 1997.
- [2] Stefan Geyersberger, Wolfgang Fiesel, Harald Gernhardt, Doris Huhn, Martin Dietz, Dae-young Jang, Kyeongok Kang, Jinsuk Kwak, Sung Han Kim, Jin-Woo Hong, “MPEG-2 AAC Multichannel Realtime Implementation on Floating Point DSP”, AES 106th Convention, Munchen, March 1999.
- [3] Texas Instruments: TMS320C62x/C67x CPU and Instruction Set (SPRU189C).
- [4] Texas Instruments: TMS320C62x/C67x Programmer’s Guide (SPRU198B).
- [5] Texas Instruments: TMS320C6x Assembly Language Tools (SPRU186C).
- [6] Texas Instruments: TMS320C6x Optimizing C Compiler (SPRU187C).
- [7] 강경욱, 홍진우, 김진웅, 나훈, 정대권, “TMS320C6701 DSP를 이용한 MPEG-4 오디오 HVXC 코덱의 실시간 구현”, ’99 한국방송공학회 정기학술대회, 1999.11.13 발행 예정.