

# 부동 소수점 DSP를 이용한 MPEG-4 HVXC 인코더 및 디코더의 실시간 구현

## Real-time Implementation of MPEG-4 HVXC Encoder and Decoder on Floating Point DSP

강 경 옥\*, 나 훈\*\*, 홍 진 우\*, 정 대 권\*\*

(Kyeongok Kang\*, Hoon Na\*\*, Jin-Woo Hong\*, Dae Gwon Jeong\*\*)

### 요 약

본 논문에서는 인터넷 폰, 디지털 이동통신 등과 같이 낮은 비트율이 요구되는 분야에 사용될 수 있는 MPEG-4 오디오의 HVXC(Harmonic Vector eXcitation Coding) 알고리즘을 부동 소수점 DSP인 TMS320C6701에 실시간 구현한 내용을 기술한다.

실시간 동작을 위한 하드웨어 구조를 채택하였으며, 소프트웨어 최적화의 경우 연산 시간이 많이 소요되는 함수 루틴에 대한 C 언어 및 어셈블리 언어 레벨의 최적화를 수행하였다. 또한, DSP의 내부 프로그램 메모리의 프로그램 캐쉬로의 활용, DSP의 내부 데이터 메모리의 영역의 중첩 활용 및 background DMA 방식을 이용한 최적화를 수행하였다. 최적화 결과 2kbps 및 4kbps의 비트율에서 압축 및 복원을 실시간으로 수행할 수 있으며, 인코더의 경우 2kbps의 경우에는 최적화 전에 비해 약 96% 정도로 수행시간을 단축하였다. 또한, 비공식 주관품질 평가에 의하면 2kbps의 비트율에서 약 MOS 2.45를 얻었다.

핵심용어: 실시간 구현, 최적화

분류번호: 음성처리(2.2)

### ABSTRACT

In this paper, we described the real-time implementation effort of MPEG-4 audio HVXC (Harmonic Vector eXcitation Coding) algorithm for very low bitrates, which has target applications from mobile communications to Internet telephony, on current high performance floating point TMS320C6701 DSP.

We adopted a hardware structure for real-time operation. In order for software optimization, we used C- and assembly-language level optimizations for time-critical functional codes. Utilizing the internal program memory of the DSP as the program cache, the internal data memory overlap technique and DMA functionality, we could get a goal of realtime operation of HVXC codec both at 2 kbit/s and at 4 kbit/s. For an encoder at 2 kbit/s, the optimization ratio to original code is about 96 %. Finally, we got the subjective quality of MOS 2.45 at 2 kbit/s from an informal quality test.

Key words: HVXC, TMS320C6701, Realtime implementation, Optimization

### I. 서 론

MPEG-4 오디오 부호화 기술에서는 2kbps에서 64kbps까지의 넓은 범위의 비트율(bitrate)을 지원하고 있으며, 원하는 비트율에서 가장 좋은 음질을 얻기 위하여 다음과 같은 3가지 형태의 코덱(codec)을 제공하고 있다. 즉, 저 비트율(2 ~ 6kbps)의 parametric 코덱, 중간 비트율(6 ~ 24kbps)

의 Code Excited Linear Predictive(CELP) 코덱, 그리고 고 비트율(16kbps 이상)의 Time-to-Frequency(T/F) 코덱이 있다[1][2].

이 중에서 parametric 코덱은 음성/음악 성분으로 구성된 오디오 입력 신호를 2kbps에서 약 6kbps까지의 매우 낮은 비트율로 부호화하기 위해 사용되며, 2개의 블록으로 구성되어 있다. 그 중 하나는 HVXC(Harmonic and Vector eXcitation Coding) 코덱으로써 음성신호를 2 ~ 4kbps사 이로 압축 복원하는 블록이고, 또 하나는 HILN(Harmonic and Individual Line plus Noise) 코덱으로써 4kbps이상의

\* ETRI 무선방송기술연구소 방송미디어 연구부

\*\* 한국항공대학교 항공전자공학과

접수일자: 1999년 12월 8일

비트율로 음악과 같은 음성 이외의 신호를 압축 복원하는 블록이다.

한편, ITU-T에서 표준으로 제정된 음성 부호기로는 6.3/5.3kbps의 G.723, 8kbps의 G.729, 16kbps의 G.728, 32kbps의 G.721등이 있다. 그러나 MPEG-4에서 정의한 음성 부호기는 8kHz의 표본화 주파수에서는 2 ~ 24kbps, 16kHz의 표본화 주파수에서는 16 ~ 64kbps까지의 다양한 비트율을 지원할 수 있다. 즉 ITU-T의 음성 부호기는 8 kHz의 표본화 주파수에서 2kbps와 같은 낮은 비트율을 지원하지 못하고 16kHz의 표본화 주파수에서 높은 비트율을 지원하지 못하는 단점이 있다. 더욱이 MPEG-4 음성 부호기에서는 bitrate scalability, complexity scalability 등의 기능이 제공되는 장점이 있다. 특히 HVXC에 의한 2.0kbps의 비트율은 국제표준 중에서 가장 낮은 비트율이며, 음질 측면에서는 일상적인 대화를 하는데 지장이 없을 정도이다. 또한, 그림 1에 나타난 것과 같이 MPEG 그룹에 의한 주관 품질평가(Mean Opinion Score(MOS) test) 결과에 의하면, 미 연방 표준으로서 4.8kbps의 CELP 알고리즘을 사용하는 FS1016표준 보다 더 좋은 주관 품질을 보이고 있다[3].

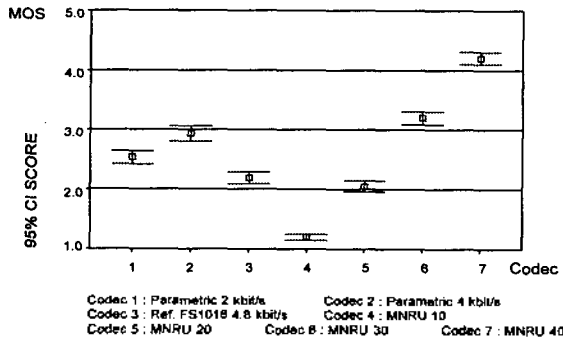


그림 1. HVXC 코덱의 주관 품질 평가 결과  
Fig. 1. Subjective quality test result for HVXC codec. This test was conducted in Nokia, FhG and NTT. In Nokia and FhG, the test languages were English, German and Swedish. In NTT, the language was Japanese. The above was the Nokia test result averaged for all European items.(source : ISO/IEC N2424, October 1998).

본 논문에서는 인터넷 폰이나 디지털 이동통신에서와 같이 낮은 비트율이 요구되는 응용분야에서 사용될 수 있는 HVXC 부호화 및 복호화 알고리즘을 부동 소수점 DSP인 TMS320C6701 DSP에 실시간 구현한 내용을 기술한다. 이를 위하여 제2장에서는 HVXC 알고리즘의 개요를, 제3장과 제4장에서는 TMS320C6701 DSP를 이용한 인코더 및 디코더의 실시간 구현내용과 인코더 및 디코더의 연동시험을 각각 기술하고, 끝으로 제5장에서 결론을 맺는다.

## II. HVXC 알고리즘의 개요

### 2.1. HVXC 인코더

HVXC 알고리즘은 신호가 유성음일 때는 LPC(Linear Predictive Coding) 잉여신호(residuals)의 스펙트럼 포락선(spectral envelope)을 하모닉 벡터 양자화하여 부호화하고, 무성음일 때는 vector excitation coding(VXC) 기법을 사용하기 때문에 HVXC 부호화 알고리즘이라고 한다[1]. 그림 2에 HVXC 인코더 알고리즘의 블록도를 나타낸다.

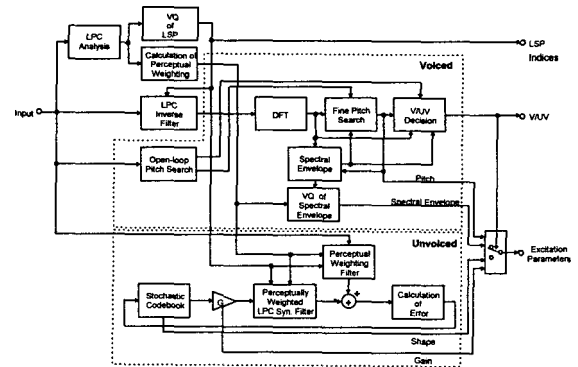


그림 2. HVXC 인코더 블록도

Fig. 2. Blockdiagram of the HVXC encoder. For voiced signal harmonic VQ of spectral envelope of LPC residuals is used. On the other hand, for unvoiced signal vector excitation coding (VXC) is used.

그림 2에서 볼 수 있듯이 여기신호에 대한 파라미터로 유성음에 대해서는 피치와, 고조파 진폭의 형태(shape)에 대한 VQ 코드북 인덱스 및 이득(gain)에 대한 스칼라 코드북 인덱스를 전송한다. 반면에 무성음에 대해서는 형태에 대한 stochastic 코드북 인덱스 및 이득에 대한 스칼라 코드북 인덱스를 전송한다[1][2].

HVXC 인코더의 비트율은 기본 계층(base layer)만을 사용하면 2kbps이고 확장 계층(extension layer)을 사용할 경우에는 4kbps까지 가능하다. 또한, 가변 비트율 모드에서는 평균적으로 1.2 ~ 1.7 kbps의 비트율을 가지고 있다. 표 1에 고정 비트율에서의 부호화 파라미터의 비트 할당표를 나타낸다.

알고리즘 지연은 정상 지연(normal delay) 모드일 때 56ms(인코더 46ms, 디코더 10ms)이고, 저 지연(short delay) 모드일 때 33.5ms(인코더 26ms, 디코더 7.5ms)이다. 인코더의 알고리즘 지연은 26ms와 46ms가 가능하며 46ms가 선택되면 피치검출 시 한 프레임의 look-ahead를 고려한다. 26ms의 경우에는 현재 프레임만 가지고 피치 검출을 수행하게 된다.

그림 3에 HVXC 코덱의 프레임 구조를 나타낸다. 8kHz의 표본화 주파수에서 프레임의 길이는 160 샘플

(20ms)이며, 256 샘플 길이의 해석 창을 사용한 LPC 분석을 수행한다. 유성음에 대해서는 256 샘플 길이에 대하여 스펙트럼 포락선을 추정하고 무성음에 대해서는 80 샘플(10ms) 단위의 부 프레임에 대한 stochastic 벡터 양자화(VXC)를 수행한다.

표 1. 부호화 파라미터에 대한 비트 할당  
Table 1. Bit allocation of the encoded parameters. '1' corresponds to 2 kbps and '2' to 4 kbps.

Parameter	Voiced	Common	Unvoiced
LSF 1	18 bits/20ms		
LSF 2	8bits/20ms		
V/UV	2 bits/20ms		
Pitch	7bits/20ms		
hamonic 1 shape	4+4bits/20ms		
hamonic 1 gain	5bits/20ms		
hamonic 2 split	32bits/20ms		
VXC 1 shape	6bits/10ms		
VXC 1 gain	4bits/10ms		
VXC 2 shape	5bits/5ms		
VXC 2 gain	5bits/5ms		
Total(1)2kbps	40bits/20ms		40bits/20ms
Total(1&2)4kbps	80bits/20ms		80bits/20ms

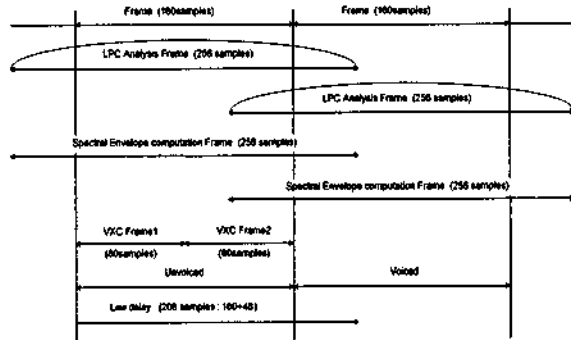


그림 3. HVXC의 프레임 구조  
Fig. 3. Frame structure of HVXC. The normal and low delays for encoder are 46ms and 26ms, respectively, and those for decoder are 10ms and 7.5ms, respectively.

2.2. HVXC 디코더

HVXC 디코더는 2kbps에서 4kbps로 부호화된 음성 신호의 복호화를 수행할 수 있으며, 또한 1.2 ~ 1.7kbps 정도의 평균 비트율에서 가변 비트율로 부호화된 음성 신호의 복호화를 수행할 수 있다. 그림 4는 HVXC 디코더의 전체 구조를 나타낸다.

HVXC 디코더의 기본적인 동작은 우선 수신된 파라미터를 디코딩한 후, 유성음인 경우에는 정현과 합성(harmonic 합성)에 의해 여기신호를 생성하고 무성음인 경우에는 코드북을 참조하여 여기신호를 생성한 후, LPC 합성을 수행하여 음성신호를 합성한다. 마지막으로 합성신호의 음질 향상을 위하여 후처리 필터를 적용한다.

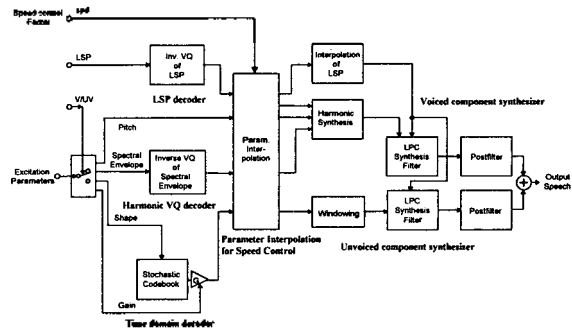


그림 4. HVXC 디코더의 블록도  
Fig. 4. Blockdiagram of the HVXC decoder.

III. HVXC 인코더 및 디코더의 실시간 구현

3.1. TMS320C6701 DSP의 기능 분석

TMS320C6x는 진보된 VLIW(very long instruction word) 아키텍처를 가지고 있으며 단정도 연산(single-precision operation)에 대해 최대 1GFOLPS(one billion floating-point operations per second)의 성능을 나타낸다[4].

TMS320C6701은 167MHz에서 동작하며 매 사이클마다 최대 8개의 32비트 명령어(instruction)를 수행한다. CPU는 32개의 32비트 범용 레지스터와 8개의 함수 유니트(functional unit)로 구성된다. 8개의 함수 유니트는 2개의 곱셈기와 6개의 산술연산기로 이루어진다.

TMS320C6701의 블록 다이어그램은 그림 5와 같으며, 메모리 맵은 다음과 같다.

- 내부 프로그램 메모리(16K words)
- 내부 데이터 메모리(16K words)
- 내부 주변기기(peripherals)
- EMIF(external memory interface)를 통하여 액세스할 수 있는 외부 메모리(최대 52MBytes)

DSP 프로세서 코어에서 I/O 동작을 수행할 경우에는 매 사이클마다 한 개 또는 두 개 이상의 instruction을 수행하는 것은 불가능한 경우가 대부분이다. 또한, 빈번한 외부 메모리 액세스를 필요로 하기 때문에 결국 프로세서의 성능저하를 가져온다. 이는 1300MIPS DSP(167MHz \* 8 instructions)의 많은 부분을 I/O 관리하는데 소모하게 되어 전체 성능을 크게 저하시키게 되며, 결국 DSP의 최대 성능은 프로그램 코드와 데이터가 내부 메모리 영역에 있을 때 얻을 수 있음을 의미한다.

표 2에 내부 메모리와 외부 메모리 상호간의 wait state를 나타낸다[5]. 표에서 알 수 있듯이 CPU에서 외부 메모리로의 데이터 액세스에는 메모리 형태에 따라 얼마간의 wait state를 보여주는 것을 알 수 있다. 반면에 프로그램 메모리 제어기나 DMA 메모리 제어기는 외부 메모리를 버스트 모드로 액세스하여 사이클 당 하나의 데이터 단위로 데이터 액세스를 할 수 있다. 메모리 형태에 따라 CPU의 데이터 액세스에서의 wait-state의 수는 중요하기 때문에 CPU를 통한 데이터 전송보다는 유용한

DMA 채널 중의 하나를 이용하여 데이터 전송을 하는 것이 더욱 효과적이다. 왜냐하면 DMA는 코어 프로세싱과 병렬로 사용할 수 있기 때문이다.

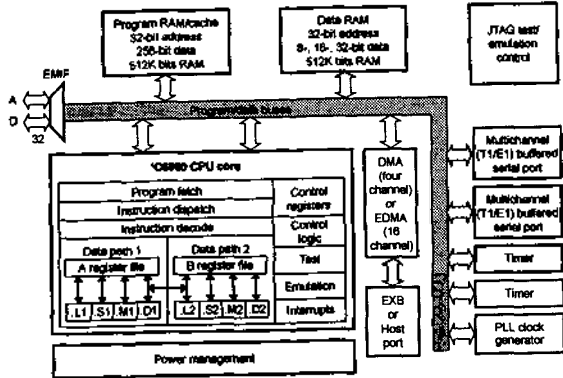


그림 5. TMS320C6701 DSP의 블록 다이어그램  
Fig. 5. Blockdiagram of the TMS320C6701 DSP.

프로그램 코드에 대해서는 TMS320C6701은 프로그램 캐쉬 아키텍처를 사용하여 메모리 로딩과 관련된 오버헤드 없이 내부 프로그램 캐쉬 영역으로 사용할 수 있다.

표 2. 메모리 위치에 따른 DSP 코어 wait state  
Table 2. DSP core wait states on memory locations [5]. For the memory-test purpose the core uses a typical memory copy function for data exchange.

source memory	destination memory	Cycles for 2048 values	average wait states
internal	internal	2165	0
SBSRAM	internal	28754	14
SDRAM	internal	37337	18
internal	SBSRAM	14416	7
SBSRAM	SBSRAM	41078	20
SDRAM	SBSRAM	49852	24
internal	SDRAM	17767	8
SBSRAM	SDRAM	45746	22
SDRAM	SDRAM	63061	30

한편, 현재 사용중인 TMS320C6701 DSP 칩의 경우 더블워드 로드(LDDW) 명령어가 내부 메모리 영역에서만 사용할 수 있는 제약이 있기 때문에, 외부 메모리 영역에서는 double 형 변수를 사용하는(배정도의) 모든 수학 함수를 float 형 함수(단정도 함수)로 변환 사용하여야 하며 double 형 변수 또한 사용할 수 없다. 뿐만 아니라 ANSI-C 변수 형인 long의 경우에도 TMS320C6701 DSP에서는 40비트의 정수를 표현하지만 동일한 이유로 인하여 32비트 정수형 변수 int로 변환하여야 하는 문제점을 가지고 있다[6].

3.2. 하드웨어 플랫폼 개발

HVXC 알고리즘의 실시간 동작을 위한 최적화 단계에서는 TMS320C6701 EVM(evaluation module) 보드를 타겟 하드웨어로 사용하였으며 Go DSP의 CODE COMPOSER emulation SW를 디버깅 도구로 사용하였다[7].

실시간 동작이 가능한 목표 사이클을 얻은 후에는 MPEG-2 Advanced Audio Coding(AAC) 및 MPEG-4 오디오 응용을 위해 개발한 시스템인 Multimedia Audio and Speech Integrated Codec(MASIC) 시스템을 하드웨어 플랫폼으로 사용하였다. 이 시스템은 그림 6에 나타난 것과 같이 Peripheral Connection Interface(PCI)를 통하여 호스트 컴퓨터에 의한 제어가 가능하며, 160MHz의 클럭 속도를 갖는 TMS320C6701 DSP를 사용하여 MPEG-2 AAC 인코더의 경우에 최대 5.1채널의 오디오 신호에 대해 DSP 2개를, 디코더의 경우에 최대 8채널의 신호에 대해 DSP 1개를 사용하여 실시간 처리를 수행할 수 있다[5][8].

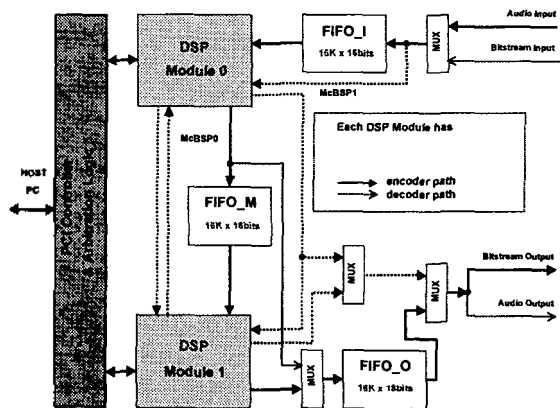


그림 6. 하드웨어 플랫폼의 구조  
Fig. 6. Structure of the hardware platform. This system is called Multimedia Audio and Speech Integrated Codec (MASIC) system and consists of 2 TMS320C6701 DSPs operating at 160 MHz.

MASIC 시스템은 원래 MPEG-2 AAC 및 MPEG-4 오디오 응용을 위하여 개발하였기 때문에 인코더로 사용될 경우에는 DSP 모듈 '0'를 통하여 MPEG-4 오디오의 다른 객체(예를 들어 AAC Scalable object)의 부호화를 수행하고 DSP 모듈 '1'에서 HVXC 인코더 알고리즘이 동작하도록 구현하였다. 따라서, DSP 모듈 '0'에서 음성신호의 입력처리를 수행하고, DSP 모듈 '1'에서 HVXC 부호화와 비트스트림 출력을 수행하였다. 반면에 디코더의 경우에는 DSP 모듈 '0'만을 사용하여 비트스트림 입력, HVXC 복호화, 음성신호의 출력을 수행하였다.

3.3. HVXC 인코더 소프트웨어 최적화

실시간 HVXC 인코딩 프로그램의 동작흐름은 그림 7과 같다. HVXC 인코딩은 MPEG 그룹에서 제공하고 있는

verification model(VM) 소프트웨어를 사용하여 최적화를 수행하였으며, 이 VM 소프트웨어의 경우 디코더의 경우에 알고리즘의 복잡도 및 메모리 요구사항은 표 3과 같다 [1]. 비교를 위하여 MPEG-4 오디오의 AAC LC(Low complexity), AAC Scalable 및 CELP의 경우와 비교하여 제시하였으며, HVXC 인코더의 메모리 요구사항은 다음과 같다.

- 프로그램 메모리 : 262Kbytes
- 데이터 메모리(RAM/ROM) : 289KBytes/139KBytes

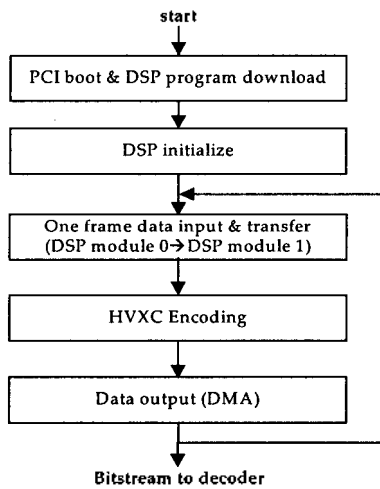


그림 7. 실시간 HVXC 인코더의 동작 흐름도  
Fig. 7. Operation flow of the realtime HVXC encoder.

표 3. MPEG-4 오디오 코덱의 알고리즘 복잡도  
Table 3. Algorithm complexity of MPEG-4 audio objects [1].  
These data are corresponding to single channel decoder.

Object type	Sampling frequency(KHz)	PCU(MOPS)*	RCU(kWords)**
HVXC	8	2	1
NB CELP	8	1	1
WB CELP	16	2	1
AAC LC	48	3	3
AAC Scalable	48	5	4

\* Processor Complexity Units, \*\* RAM Complexity Units

일반적으로 음성 코덱의 실시간 구현을 위해서는 저가의 고정 소수점(fixed-point) DSP를 사용하지만, 본 논문의 경우에는 음질 향상 측면 뿐만 아니라 HVXC 알고리즘이 MPEG-4 오디오의 하나의 객체에 해당하며 MASIC 시스템을 이용한 HVXC와 MPEG-4 오디오의 다른 객체(예를 들면 AAC)의 처리를 응용목표로 하였기 때문에 부동 소수점 DSP를 사용하였다. HVXC 인코더 알고리즘을 한 개의 TMS320C6701 160MHz DSP를 사용하여 실시간 처리를 하기 위해서는 20ms의 프레임 크기에 대해 최대 3.2 million cycle 내에서 처리가 가능하여야 한다.

그러나, VM 소프트웨어를 DSP 시스템에 포팅한 후 2kbps의 경우의 한 프레임에 대한 수행시간은 58.28 million cycle이었다.

최적화를 위하여 우선, TMS320C6701 DSP 내부 프로그램 메모리 영역을 프로그램 캐쉬로 사용하여 수행시간을 69.9% 만큼 단축하였다. 그리고, 함수별 수행시간을 분석하여 표 4와 같이 연산량이 많은 34개의 함수 모듈을 선정하여 이 모듈에 대한 C 언어 레벨 및 어셈블리 언어 레벨의 최적화를 수행하여 최적화 라이브러리를 개발하였다[9~12]. 선정된 함수 모듈은 실행 시 machine cycle(cycle per execution)로는 전체 부호기 연산량의 77.6%를 차지한다[9].

표 4. 최적화 대상 함수 분석 및 최적화 결과  
Table 4. Analysis of functions for optimization and optimization result.

Function	Unoptimized		Optimized (cycles*)	Optimized ratio(%)
	Cycles*	Ratio(%)**		
Pitch Estimation	2,423,522	41.7	1,348,728	44.3
V/UV decision	35,317	0.1	11,922	66.3
Spectral envelope VQ	2,020,386	9.4	1,169,706	42.1
VXC	431,832	21.4	167,450	61.2
Etc.	509,137	5.0	323,866	36.4
Total	5,420,194	77.6	3,021,672	44.3

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

\* Cycles every execution

\*\* Percentage to the total cycles of the encoder per execution (function call).

최적화 라이브러리의 개발 결과 34개의 최적화 대상 함수에 대하여 표 4와 같이 machine cycle로 평균 44.3% 정도로 수행시간을 단축하였으며, 개발된 최적화 라이브러리 모듈을 채용한 인코더 프로그램의 한 프레임당 처리시간은 표 5와 같이 8.84 million cycle이었다. 또한, 컴파일러가 제공하는 최적화 방법을 사용하면 VM 코드에 비해 87.1% 정도의 수행시간이 단축된 7.53 million cycle의 한 프레임당 처리시간을 얻었다.

표 5. 최적화 라이브러리 모듈 및 컴파일러에 의한 최적화  
Table 5. Optimization results from the optimized modules and the Compiler optimization.

Status	Cycles*	Optimized ratio(%)**	Notes
Original VM	58.28		Unoptimized
	17.53	69.9	Cache enable
Substitution of optimized modules	8.84	84.8	Optimized library
Compiler optimization	7.53	87.1	Level 3(-o3)

\* Cycles per frame (million cycles),

\*\* Ratio to the original VM code

그러나, 이 상태에서도 실시간 수행을 위한 사이클 수보다 2.5배 가까이 초과하고 있기 때문에 LPC 분석, 파치 추정 및 스펙트럼 포락선 추정 과정에서의 공통적인 요소의 제거 등을 포함한 최적화 라이브러리 모듈 및 전체 인코더 프로그램에 대한 추가적인 C 언어 레벨의 최적화, TI의 TMS320C6701용 FFT 어셈블러 함수를 사용하여 프레임당 4.98 million cycles의 수행시간을 얻었다. 이 단계에서의 주요 기능별 프레임당 수행시간을 표 6에 나타낸다.

표 6. 2kbps 경우의 인코더 최적화 결과  
Table 6. Optimization result of the encoder at 2 kbps.

Function	Cycles*	Mid-tem optimization	Final optimization	Optimized ratio(%)
LPC analysis & VQ		1,068,600	417,980	60.7
Pitch Estimation		1,051,149	446,868	57.5
Fine pitch search & Spectral envelope estimation		693,775	293,165	57.7
V/UV decision		499,919	100,500	79.9
VQ of spectral envelope		886,351	480,653	45.8
VXC		777,231	382,393	50.8
Total		4,977,025	2,121,559	57.5

\* Cycles per frame

한편, 표 2에 나타난 것과 같이 DSP core와 SDRAM과 같은 외부 메모리 사이에는 얼마간의 wait state가 존재하기 때문에, 계산량이 많은 부분의 연산, 테이블 참조가 필요한 LSP계수의 양자화를 위한 VQ 테이블, 고조파 진폭의 벡터 양자화를 위한 형태 및 이득 코드북, 무성음 성분의 양자화(VXC)를 위한 형태 및 이득 코드북을 background DMA를 이용하여 DSP의 내부 데이터 메모리 영역으로 옮겨 연산을 수행하면 수행시간을 크게 단축할 수 있다. 뿐만 아니라, 스택(stack)을 DSP의 내부 메모리 영역에 설정하면 동일한 효과를 얻을 수 있어 효율적이다. 이와 같은 최적화 방법을 사용하여 2kbps의 경우에 표 6에 제시한 바와 같이 프레임당 2.12 million cycle의 수행시간에 이르러 최적화 전의 VM 코드에 비해 약 96% 정도로 수행시간을 단축함으로써 HVXC 인코더의 실시간 동작이 가능하게 되었다. 음성신호의 입력 및 비트스트림의 출력에도 background DMA(direct memory access) 방식을 이용하였다.

2kbps의 경우에 실시간 동작을 위해서는 무성음 성분의 부호화 연산에 대한 최적화가 더욱 필요하다. 왜냐하면, 무성음에 대한 전체 부호화는 유성음 성분과의 공통적인 연산 외에 10ms 단위의 2개의 부 프레임에 대한 VXC 연산이 수행되기 때문이다.

4kbps의 비트율에서는 LSP 계수의 양자화, 고조파 성분 진폭의 벡터 양자화, 무성음 성분의 벡터 양자화 모듈에서의 추가적인 연산이 필요하기 때문에 2kbps의 경우와

동일한 상위 레벨의 최적화, 내부 데이터 메모리 영역에서의 연산 등의 최적화를 통하여 표 7과 같은 최적화 결과를 얻을 수 있었다. 무성음의 경우에는 VXC 부호화 모듈의 연산이 5ms 단위의 부 프레임에 대해 4번에 걸쳐 수행되기 때문에 VXC 모듈에 대한 최적화가 필요하며, 유성음의 경우에는 고조파 성분 진폭의 벡터 양자화 모듈에 대한 최적화가 필수적이다.

표 7. 4kbps 경우의 인코더 최적화 결과  
Table 7. Optimization result of the encoder at 4 kbps.

Function	Cycles/frame	Ratio(%)
LPC analysis & VQ	441,246	17.8
Pitch Estimation	448,796	18.1
Fine pitch search & Spectral envelope estimation	293,114	11.8
V/UV decision	101,172	4.1
VQ of spectral envelope	494,216	19.9
VXC	480,901	19.3
Total	2,485,527	100

3.4. HVXC 디코더 소프트웨어 최적화

실시간 HVXC 디코더 프로그램의 동작 흐름은 그림 8과 같으며, HVXC 디코더의 메모리 요구사항은 다음과 같다.

- 프로그램 메모리 : 173Kbytes
- 데이터 메모리(RAM/ROM) : 160Kbytes/85Kbytes

VM 소프트웨어를 DSP 시스템에 포팅한 후 프레임당 수행시간은 17 million cycle이었으며, DSP 내부 프로그램 메모리 영역을 프로그램 캐쉬로 사용하면 약 4.64 million cycle이었다. 인코더의 경우와 같이 함수별 수행시간을 분석하여 연산량이 많은 함수에 대한 C 언어 레벨 및 어셈블리 레벨의 최적화를 수행하여 일부 함수에 대한 최적화 라이브러리 모듈을 개발하였으며, 컴파일러에 의한 최적화를 수행하였다.

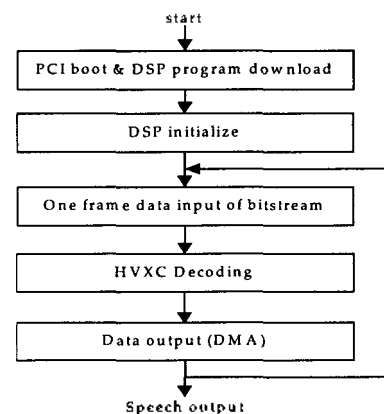


그림 8. 실시간 HVXC 디코더의 동작 흐름도  
Fig. 8. Operation flow of the real-time HVXC decoder.

인코더의 경우에서처럼 계산량이 많은 부분의 연산, 테이블 참조가 필요한 연산 등을 background DMA를 이용하여 DSP의 내부 데이터 메모리 영역으로 옮겨 연산을 수행하였고, 스택을 DSP의 내부 메모리 영역에 설정하여 수행시간을 단축하였다. 이와 같은 최적화 방법을 사용하여 4kbps의 경우에 표 8에 나타난 것과 같이 디코더의 실시간 동작이 가능하게 되었다.

표 8. 디코더 최적화 결과(4kbps, short delay)  
Table 8. Optimization result of the decoder at 4 kbps and short delay.

Function	Cycles*	Pre-optimization	Post-optimization	Optimized ratio(%)
Parameter decoding(LSP, Harmonic VQ, VXC)		385,567	298,920	22.5
Parameter interpolation		20,362	13,394	34.2
Voiced component synthesizer		3,174,592	1,661,316	47.7
Unvoiced component synthesizer		774,269	363,230	53.1
Total		4,354,790	2,246,860	48.4

\* Cycle per frame

디코더의 경우에는 비트스트림으로부터 파라미터를 추출하는 부분의 오버헤드는 무시할 수 있으며, 대부분의 수행시간이 유성을 및 무성을 성분의 합성 과정에 소요됨을 알 수 있다. 인코더의 경우도 그렇지만 본 연구에서는 TMS320C6701 DSP 자원을 이용한 실시간 동작이 목표이기 때문에 안정적인 실시간 동작을 확인한 후 추가적인 최적화는 수행하지 않았다.

#### IV. 인코더 및 디코더 연동시험

HVXC 인코더 및 디코더의 실시간 동작을 확인하기 위한 연동시험을 수행하였다. 연동시험을 위한 시스템 구성도를 그림 9에 나타낸다. 편의상 인코더와 디코더를 각각의 호스트 PC와 분리하여 나타내었으나, 실제로는 인코더는 인코더 DSP 보드가 장착된 개인용 컴퓨터이고 마찬가지로 디코더도 디코더 DSP 보드가 장착된 또 다른 개인용 컴퓨터이다. 인코더와 디코더는 그림 6의 MASIC 시스템을 사용하였으나, 인코더는 TMS320C6701 DSP 2개를 디코더는 1개를 채용하였다.

입력 박스(Input Box) 및 출력 박스(Output Box)는 멀티 채널 오디오 또는 음성신호의 입출력을 위한 I/O 장치로서, 최대 8채널 아날로그 신호 또는 4채널의 디지털 신호를 입출력할 수 있으며, ADI(Alesis Multi-channel Optical Digital Interface) 포맷의 Optical 인터페이스를 구비하고 있어 MASIC DSP 보드와 디지털 오디오 신호를 교환할 수 있다. 한편, 입출력 채널 중 아날로그 2채널 또는 디지털 1채널을 음성신호 채널로 사용할 수 있기 때문에, 아날로그 한 채널을 연동시험을 위한 음성신호의 입출력

채널로 사용하였다.

입력/출력 박스와 인코더/디코더 상호간의 음성신호의 입출력을 위하여 별도의 드라이버 소프트웨어를 작성하여 사용하였으며, PCI 인터페이스를 통하여 HVXC 부호화 및 복호화 알고리즘을 DSP에 탑재한 후 인코더 및 디코더의 연동시험을 수행하였다. 3.2절에서 설명한 바와 같이 인코더의 경우 DSP 모듈 '0'에서 음성신호의 입력처리를 DSP 모듈 '1'에서 HVXC 인코더 프로그램의 처리 및 비트스트림의 출력을 각각 수행하였으며, 디코더는 DSP 모듈 '0'만을 사용하여 비트스트림 입력, HVXC 디코더 프로그램의 처리 및 음성신호의 출력을 수행하였다. DAT(Digital Audio Tape Recorder)를 음성신호원으로 사용하여 연동시험을 수행한 결과 음성신호 1채널에 대하여 2kbps 및 4kbps에서 HVXC 인코더 및 디코더의 실시간 동작을 확인할 수 있었다.

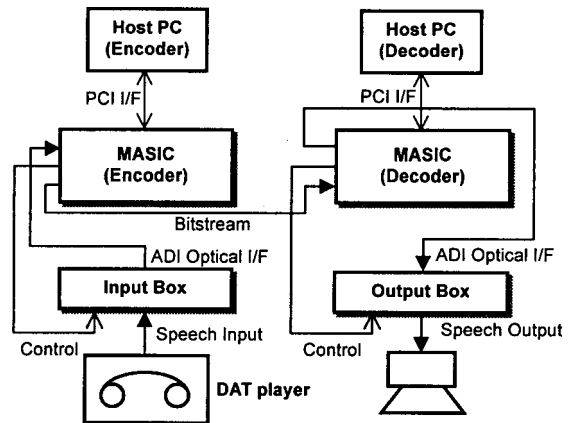


그림 9. 실시간 연동시험을 위한 시스템 구성

Fig. 9. System configuration for realtime operation test. The MASIC system in the encoder mode uses 2 TMS320C6701 DSPs. On the other hand, The MASIC system in the decoder mode uses only one DSP.

한편, HVXC 알고리즘에 의한 부호화 및 복호화 과정을 거친 신호에 대하여 비공식적인 주관 품질평가를 실행하였다. 음성 시료는 각각 약 2초 간격의 한국어, 영어 음성을 사용하였으며, 8명의 비 전문가를 대상으로 하여 원음과 2kbps의 HVXC 코덱을 통한 음성을 비교 제시하여 코덱 음성에 대한 주관 품질평가를 실시하였다. 품질 평가 척도는 1점(매우 나쁘다)에서 5점(아주 좋다)까지 연속적인 수치를 사용하였으며, 평가 결과로부터 MOS(Mean Opinion Score)를 구하였다. 그 결과에 의하면 약 2.45 정도의 MOS를 나타내고 있어 그림 1에 표시한 MPEG 그룹의 결과와 유사함을 확인할 수 있었다.

#### V. 결 론

지금까지 HVXC 부호화 및 복호화 알고리즘을 TMS320C6701 160MHz DSP를 사용하여 실시간 동작을

구현한 내용을 기술하였다.

연산 시간이 많이 소요되는 합수 루틴에 대한 C 언어 레벨의 최적화 및 어셈블리 언어 레벨의 최적화를 수행하였고, TMS320C6701 160MHz DSP내부 프로그램 메모리를 프로그램 캐쉬로 사용하고, 계산량이 많은 부분과 테이블 참조가 필요한 연산을 DSP의 내부 데이터 메모리 영역에서 수행하여 소요시간을 단축하였으며, 음성신호 및 비트스트림의 입출력에는 background DMA 방식을 이용하였다. 이와 같은 최적화를 수행한 후 인코더 및 디코더의 연동시험을 통하여 2kbps 및 4kbps의 비트율에서 압축 및 복원을 실시간으로 수행할 수 있음을 확인하였다.

본 연구는 MPEG-2 AAC 및 MPEG-4 오디오 응용을 위해 개발한 시스템을 하드웨어 플랫폼으로 사용하였기 때문에 부동 소수점의 TMS320C6701 DSP를 사용하였으나, 저가의 고정 소수점 DSP를 이용한 HVXC 코덱 알고리즘의 최적화가 필요할 것으로 생각되며, 정확한 주관품질을 측정하기 위한 공식적인 품질평가를 실시할 예정이다.

#### 감사의 글

본 연구는 정보통신부의 "차세대 멀티미디어용 고품질 오디오 처리기술 개발" 과제의 일환으로 수행된 연구결과이며, 정보통신부 관계자와 ETRI 방송미디어연구부의 관련 연구원의 도움에 감사 드립니다.

#### 참 고 문헌

1. ISO/IEC FDIS 14496-3, "Information Technology-Coding of Audiovisual Objects, Part 3: Audio," 1998.
2. 한민수외, "MPEG-4 오디오 기술 동향," 방송공학회지, 제4권 1호, pp. 62-79, 1999년 3월.
3. ISO/IEC JTC1/SC29/WG11, "Report on the MPEG-4 speech codec verification tests," N2424, October 1998.
4. Texas Instruments: TMS320C6x Technical Brief (SPRU197).
5. Stefan Geyersberger et al, "MPEG-2 AAC Multichannel Realtime Implementation on Floating Point DSP," AES 106<sup>th</sup> Convention, Munchen, March 1999.
6. Texas Instruments: TMS320C6201/6701 Evaluation Module, Users Guide, October 1998.
7. Texas Instruments: TMS320C62x/C67x Programmers Guide (SPRU198B).
8. Dae Young Jang et al, "A Multichannel Audio Codec System for Multichannel Audio Authoring," AES 106<sup>th</sup> Convention, Munchen, March 1999.
9. 나훈외, "TMS320C6701 DSP용 MPEG-4 오디오 HVXC 부호기의 최적화 라이브러리 개발," 방송공학회 학술대회, 서울대, 1999년 11월.
10. Texas Instruments: TMS320C6x Source Debugger - User's Guide (SPRU188).
11. Texas Instruments: TMS320C6x Assembly Language Tools (SPRU186C).
12. Texas Instruments: TMS320C6x Optimizing C Compiler (SPRU187C).

#### ▲ 강 경 옥(Kyeongok Kang)

한국음향학회지 제16권 3E호 참고

#### ▲ 나 훈(Na, Hoon)



1996년 2월 : 한국항공대학교 항공전자공학과 졸업(학사)

1998년 2월 : 한국항공대학교 항공전자공학과 학원 졸업(석사)

1998년 2월 ~ 현재 : 한국항공대학교 항공전자공학과 대학원 박사과정 재학중

※ 주관심 분야: 음성/영상 신호처리, 멀티미디어 신호처리

#### ▲ 홍 진 우(Jin Woo Hong)

한국음향학회지 제16권 3E호 참고

#### ▲ 정 대 권(Dae Gwon Jeong)

한국음향학회지 제16권 3E호 참고