

VHDL을 이용한 MPEG-4 HVXC 복호화기 구현

김구용, 임강희, 차형태

송실대학교 전자공학과

Implementation of MPEG-4 HVXC decoder with VHDL

Gooyong Kim, Kanghee Lim, Hyungtai Cha

Dept. of Electronics Engineering, Soongsil Univ.

E-mail : gykim@mms.ssu.ac.kr

요약

MPEG-4 Parametric Coding 중 저 비트율로 음성신호를 부호화하는 HVXC(Harmonic Vector eXcitation Coding)의 복호화 모듈인 LSP 합성필터와 무성음 합성부, 유성음 합성부를 VHDL을 이용하여 구현하였다.

MPEG-4 HVXC의 복호화 과정은 코드북을 이용하여 LSP 계수, VXC signal, 그리고 Spectral Envelope이 부호화 되어 각각 LSP 역필터, 무성음과 유성음 합성단을 통과하여 LPC계수와 유,무성음 여기신호로 변환된 후 LPC 합성필터링 과정을 거쳐 최종적으로 음성신호를 출력시킨다. LSP inverse filter에서 사용되는 cosine함수값을 위하여 Table based Approximation을 이용하여 적은 양의 Table 값을 사용하여 정확하고 고속의 cosine 연산을 수행하였다. VXC 복호화 과정에서는 신호의 중복성을 제거하는 Hidden Address in LSB 방법을 사용하여 코드북의 크기를 줄였다. 유성음 합성단에서는 IFFT 모듈을 이용하여 연산속도를 증가 시켰다. 최종적으로 위와 같이 구현된 시스템을 Simulation을 통해 Software 검증을 하였다.

1. 서론

1998년 ISO 산하 MPEG-4 오디오 표준화 위원회(Part-3)는 그의 Subpart로서 Parametric 부호화를 표준화하였다[1]. Parametric 부호화는 부호화 하려는 음원에 따라 2 가지로 나눌 수 있는데, 8kHz, 16kHz 대역폭의 음악신호(music signal)를 압축하는 HILN(Harmonic Individual Line plus Noise)과 8kHz의 음성신호(speech signal)의 부호화를 목적으로 하는 HVXC(Harmonic Vector eXcitation Coding)가 있다[2]. 이중 HVXC는 음성신호를 최소 2.0Kbps에서 최대 4.0Kbps의 bitrate으로 부호화 할 수 있는데 회선의 상태에 따라 가변적으로 부호화 할 수 있다. 또한 음성의 Pitch와 재생속도를 독립적으로 변화시킬 수 있는 틀을 제공하고 있어 MPEG-4 HVXC Speech Codec은 저 비트율을 필요로 하는 이동통신환경, 위성통신환경

이나, 트래픽이 폭주하는 인터넷 환경에서 효율적으로 음성신호를 부호화 할 수 있고, 큰 규모의 음성DB 구축에도 적용 될 수 있다[3].

본 논문에서는 HVXC 복호화기의 복호화 과정을 칩 구현에 적합하도록 알고리즘을 최적화 시켰고 하드웨어 설계언어인 VHDL을 이용하여 설계하였다.

본 논문의 구성은 2장에서 MPEG-4 HVXC 복호화기를 소개하고 3장에서는 실제 구현된 모듈의 설명과 구현 과정 및 결과를 제시하고 4장에서 결론을 맺는다.

2. MPEG-4 HVXC 복호화기의 개요

그림 1.은 MPEG-4 HVXC의 전체 복호화 과정을 나타낸다. 복호화 과정은 HVXC 부호기에 의해 부호화된 비트열의 정보를 이용하여 LSP decoder, Harmonic VQ decoder, Time domain decoder에서 LSP 계수, 유성음의 스펙트럼 포락 정보, 무성음의 시간영역 샘플들을 각각 복호화 한다. 복호화된 파라미터들을 이용하여 LSP 계수는 LPC 계수로 변환되고, 스펙트럼 포락 정보는 하모닉 합성되어 유성음의 여기신호로 쓰이며, 시간영역 샘플은 window 연산을 거쳐 무성음의 여기신호로 사용된다. 생성된 유무성음의 여기신호는 복호화된 LPC 계수를 이용하여 각각 LPC 합성필터를 거치고 유무성음이 더해지게 되어 최종적으로 음성신호의 복호화가 이루어진다.

또한 파라미터 복호화들과 여기신호 생성부분에 Parameter Interpolation을 둘 수 있어 재생음성의 속도조절이 가능하고, LPC 합성 필터 후에 Postfilter를 두어 합성음성의 음질을 향상시킬 수 있다[4].

3. HVXC 복호화기의 VHDL구현

MPEG-4 HVXC 복호화기의 하드웨어 구현을 위한 구성은 그림 2.와 같이 3개의 codebook을 저장하는

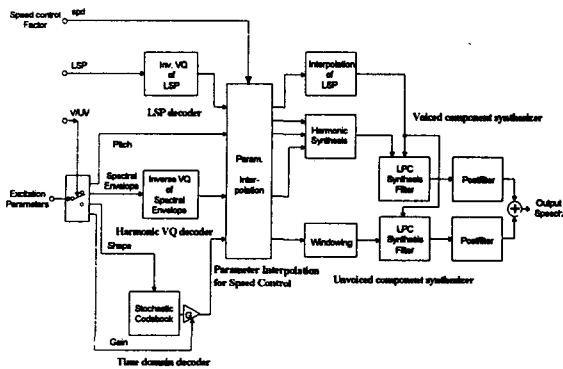


그림 1. HVXC 복호화기의 블록 다이어그램

ROM과 연산을 위한 RAM이 있으며 24bit의 덧셈, 뺄셈과 곱셈 연산을 하는 ALU와 LSP 합성필터와 무성음 여기신호 복호화기 그리고 유성음 여기신호 복호화기 모듈로 구성되어지며 고정소수점(Fixed point)을 통해 연산을 수행한다.

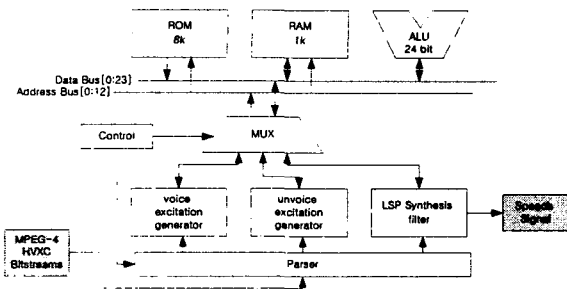


그림 2 HVXC 복호화기의 하드웨어 블록도

3-1 LSP inverse filter의 구현

LSP 합성필터는 LSP 계수를 복호화하는 LSP decoder, 복호화된 LSP계수들을 총10차의 LPC 계수로 변환하는 LSP inverse 필터, 그리고 구해진 LPC 계수를 이용하여 복호화된 유무성음을 합성하는 LPC 합성필터로 이루어져 있다. 복호화에 쓰이는 codebook은 작은 메모리 크기에 많은 수의 계수를 표현하기 위한 Split VQ로 구성되어있고, 복호화가 끝난 LSP 계수들은 LPC 필터의 안정성을 위하여 안정화 과정이 수행되어진다[1].

복호화된 LSP 계수($lsp(i)$ $i = 1$ to 10)를 이용하여 LSP inverse filter는 식(1),(2)와 같이 재귀적으로 구현하였다[5].

이때 식(1)의 cosine 함수는 디지털 시스템에서 구현하기 어려운 부분이므로 그림 3.과 식(3)와 같은 Table based approximation을 사용하여 구현하였다.

$$\begin{aligned} & \text{for } i=1 \text{ to } 11 \\ & \quad c(i) = -2\cos(2\pi lsp(i)) \quad (1) \\ & \text{end} \end{aligned}$$

$$\begin{aligned} & \text{for } i=1 \text{ to } 4 \\ & \quad \text{for } j=2+2i \text{ downto } 0 \\ & \quad \quad p(j)=p(j-2) + p(j-2)c(2i+1) + p(j) \quad (2) \\ & \quad \quad q(j)=q(j-2) + q(j-2)c(2i+2) + q(j) \\ & \quad \text{end} \\ & \text{end} \end{aligned}$$

$$\alpha(i) = \begin{cases} 0.5(p(i)+q(i)) & i=1, \dots, 5 \\ 0.5(p(11-i)-q(11-i)) & i=6, \dots, 10 \end{cases}$$

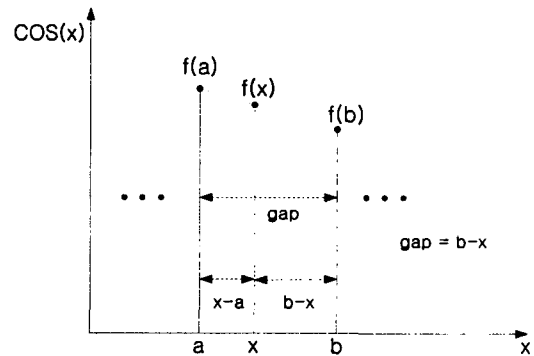


그림 3. cosine 함수를 위한 Linear Interpolation

$$f(x) = f(a) \times \frac{b-x}{gap} + f(b) \times \frac{x-a}{gap} \quad (3)$$

식(3)의 gap은 $0 < x < 0.5\pi$ 사이를 p 등분한 간격이다. $\cos(a)$ 와 $\cos(b)$ 를 나타내는 $f(a)$ 와 $f(b)$ 는 Table에 gap 간격으로 저장된 값이고 $a < x < b$ 의 $\cos(x)$ 은 그림 3.에서와 같이 구해지게 된다. 이때 $\cos(x)$ 값을 나타내는 $f(x)$ 는 식 (3)과 같이 선형 보간에 의해 구해진다[6]. 이때 이 연산은 다음과 같이 간략화 되어질 수 있다.

$$\begin{aligned} & x-a : p \text{ 번째 bit 이하자리만 택한 값} \\ & b-x : x-a의 2의 보수 \\ & 1/gap : p \text{ bit right shift} \end{aligned}$$

따라서 실제 연산에서는 2번의 곱셈과 1번의 덧셈으로 하나의 cosine값을 얻을 수 있다.

그림 4.는 위와 같이 구현된 LSP inverse 필터에서 얻어진 10차 LPC 계수들의 Spectrum 정보를 Verification Model과 비교한 것이다.

3-2 무성음 여기신호 합성부의 구현

무성음 합성에 쓰일 무성음 여기신호는 그림 5.와 같이 Stochastic Codebook에서 복호화된 Time domain samples가 windowing 연산 과정을 거쳐 무성음 합성에 쓰이게 된다.

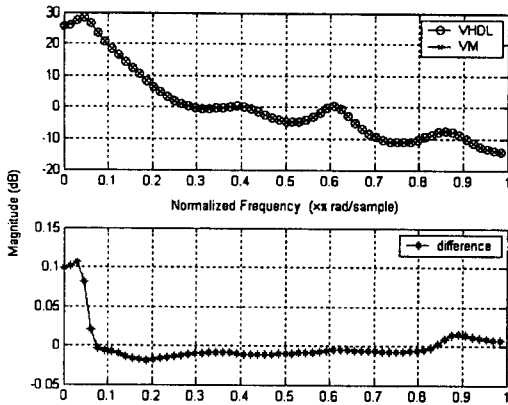


그림 4. 구현된 시스템의 LPC 스펙트럼 결과

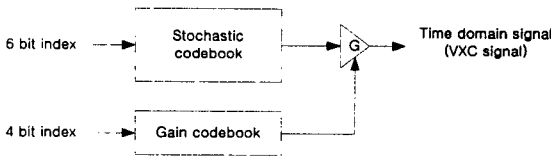


그림 5. VXC decoder의 블록도

한 프레임(160 samples)에 2번 갱신이 되므로 총 $(6\text{bit}+4\text{bit}) \times 2 = 20\text{bit}$ 이 무성음 여기신호 복호화에 쓰이게 된다. 이때 VXC 복호화에 할당된 codebook의 크기(base layer)는 $5120(\text{VXC shape}) + 16(\text{VXC gain}) = 5136$ codewords로 HVXC 복호화에 쓰이는 전체 codebook(7696 codewords)의 약 67%를 차지한다.

이러한 codebook의 크기를 줄이기 위해 다음과 같이 신호의 중복성을 제거하는 Hidden Address in LSB 방법을 사용하여 구현하였다.

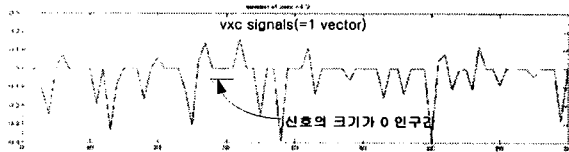


그림 6. VXC codebook samples

Stochastic codebook의 한 벡터를 나타낸 그림 6.에서 볼 수 있듯이 신호의 크기가 "0"인 구간이 존재하게 되는데 이러한 중복된 신호를 제거하여 codebook을 다음과 같이 재구성하였다.

그림 7.과 같이 값이 "0"이 아닌 sample들에 대해서는 하위비트(LSB)에 80 sample들의 위치정보를 표시할 수 있는 최소한의 값인 8bit를 이용하여 address를 은닉시키고 end synch bit을 이용하여 한 vector의 끝을 나타내었다. 이렇게 재구성된 codebook의 크기는 식(4)가 나타내는 바와 같이 약 33%를 줄일 수 있다.

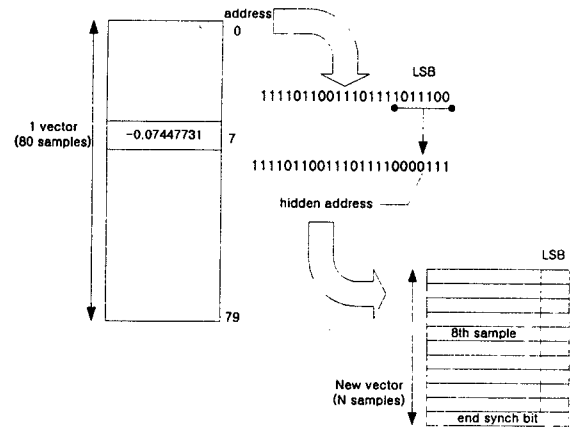


그림 7. Hidden address in LSB 방법

$$\frac{1743(\text{중복된 샘플수}) - 64(\text{추가된 end synch bit})}{5120(\text{원 codebook의 샘플수})} \times 100 \approx 33\%$$

이렇게 재구성된 codebook을 이용하여 다음과 같은 복호화 과정을 거치게 된다. 우선 RAM을 "0"으로 reset을 시킨 후 VXC shape index가 파서부로부터 입력되면 해당 코드북의 start address를 찾게된다. 그 address부터 sample값을 읽어 하위 8bit를 참고하여 원래의 codebook의 sample에 관한 위치정보를 얻어내어 해당 address의 RAM에 쓰게 된다.

위와 같이 복호화된 VXC signal들은 유성음과 부드러운 연결을 위해 windowing 과정을 거쳐 LPC 합성에 쓰일 무성음 여기신호로 변환된다.

그림 8.은 최종 합성된 무성음의 VHDL simulation과 C simulation의 차를 나타낸 것이다.

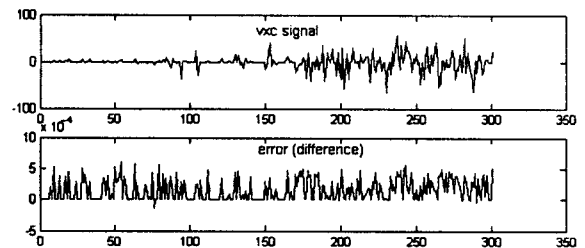


그림 8. 복호화된 VXC 샘플의 오차

3-3 유성음 여기신호 합성부의 구현

유성음 여기신호는 그림 10.와 같이 Harmonic VQ decoder에서 얻어진 Spectral envelop 정보와, Pitch 정보를 이용하여 128-points IFFT를 통해 합성해 낸다. 이때, 복호화된 44차의 Spectral Envelop은 식(5)과 같이 구현된 Dimension converter에 의해 Pitch 정보를 이용하여 해당 기본주파수의 하모닉 성분으로 변환되어진다. 그림 9.는 구현된 결과를 나타낸다.

for $i=0$ to $Harmonics-1$
 index=near_integer($i*stepSize$) (5)
 Converted[i]=Fixed[index]
 end for
 where, $Harmonics = \frac{0.95 \times pch}{2}$, $stepSize = \frac{44}{Harmonics}$

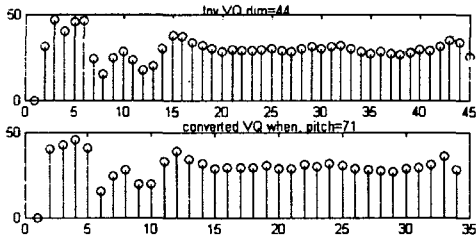


그림 9. Harmonics의 Dimension conversion

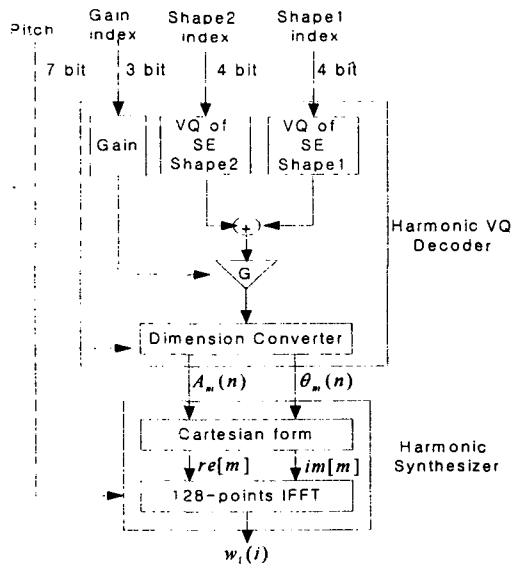


그림 10. 유성음 여기신호 합성부의 블록도

이렇게 복호화된 하모닉 성분은 실수성분과 허수성분으로 바뀌어 128-point IFFT의 입력이 되고 현재와 다음 frame의 합성신호 $w_1(i)$, $w_2(i)$ 를 각각 얻게 되는데 이 신호를 sampling conversion하여 최종 유성음 여기신호로 사용이 된다[2].

3-4 LPC 합성필터의 구현

생성된 무성음(3-2)과 유성음(3-3)의 여기신호는 LSP inverse filter에서 구해진 10차의 LPC 계수(3-1)를 사용한 식 (6)과 같은 LPC 합성필터를 통과하여 최종의 음성신호를 얻게 된다.

$$H(z) = \frac{1}{1 + \sum_{i=1}^{10} a(i) Z^{-i}} \quad (6)$$

구현된 LPC 필터를 통한 합성음성의 신호 대 잡음비는 90.6dB 이고 이때 SIR 값은 식(7)과 같이 계산하였다[6].

$$SNR[dB] = 20 \log_{10} \frac{|PCM_c|}{|PCM_c - PCM_{VHDL}|} \quad (7)$$

4. 결론

본 논문에서는 MPEG-4 HVXC 복호화기를 하드웨어 수준에서 최적화를 수행하여 VHDL을 이용하여 구현하였다. LSP inverse filter에 사용되는 cosine값을 위해 Table based approximation을 사용하였고 VXC codebook의 크기를 줄이기 위하여 Hidden Address in LSB를 사용하였다. 유성음 여기신호 생성부에서는 128-points IFFT를 사용하여 구현하였다.

표 1.는 MPEG-4 HVXC 복호화기에서 사용된 메모리 내용을 나타낸다.

	내용	codewords(24 bits)
ROM	LSP codebook	1120
	VXC codebook	3457
	SE codebook	1450
	cosine table	256
RAM	LSP 연산, 저장	256
	VXC 연산, 저장	256
	IFFT 연산	512

표 1. HVXC 복호화기의 메모리 사용(2.0Kbps모드)

참고 문헌

1. ISO/JTC1/SC29/WG1 N2203 "Committee Draft International standard of ISO/IEC 14496-3" 1998.
2. M. Nishiguchi, K. Iijima "Harmonic Vector Excitation Coding of Speech at 2.0 kbps" IEEE Workshop on Speech Coding, Sep. 1997.
3. Fang Liu, Jitae Shin, Kuo, C.-C.J. Tescher, A.G. "Streaming of MPEG-4 speech/audio over Internet" Consumer Electronics, 1999.
4. Nishiguchi, Iijima, K., Matsumoto, J. "Harmonic vector excitation coding of speech at 2.0 kbps Speech Coding For Telecommunications" Proceeding, IEEE Workshop on 1997.
5. Yun-Keun Lee "On Improving the Performance of Bit-Rate CELP-Type Speech Coder" Thesis of Ph.D KAIST 1998.
6. 김구용, 홍민철, 차형태 "VHDL을 이용한 MPEG-4 HVXC LSP 합성필터 구현" 2001 하계음향학회