VHDL을 이용한 MPEG-4 CELP 부호화기의 구현

이시원, 김수현, 홍민철, 차형태
숭실대학교 전자공학과

Implementation of MPEG-4 CELP Encoder using VHDL

Siwon Lee, Soohyun Kim, Mincheol Hong, Hyuntae Cha
Dept. of Electronics Engineering, Soongsil Univ.

E-mail : winter@nnslab.soongsil.ac.kr

요 약

MPEG-4 CELP의 알고리즘은 전송환경에 따라 선택할 수 있는 다양한 전송율을 지원하며 각종 음성 또는 음성의 추이함으로써 다양한 기능을 부가할 수 있도록 구성되어 있다. 본 논문에서는 MPEG-4 오디오 CELP 부호화기를 VHDL(Very High Speed Integrated Circuit Hardware Description Language)을 이용하여 구현하였다. MPEG-4 CELP의 부호화 과정은 크게 3단계로 나누어 LPC 분석부, 코드북 검색부, 비트율 생성부로 나누어 설계하였으며, 내부 메모리의 크기를 줄이기 위하여 스택적인 성격으로 갖거나 계산과정에서 나오는 중간 결과 값을 저장하지 않도록 알고리즘을 최적화 하였다. 또한 계산 과정의 동적 범위(Dynamic Range)가 크거나 정밀도가 요구되는 부분에 대해서는 베플로(BDouble Precision) 연산을 사용하여 값의 오차를 줄였다.

I. 서론


본 논문에서는 MPEG-4 CELP 시스템의 부호화 알고 리즘을 VHDL로 구현한 VM(Verification Model)을 이용하여 모의 실험하고, 이를 바탕으로 MPEG-4 CELP 부호화기를 ASIC로 구현하기 위하여 합성 가
능한 VHDL 프로그램으로 설계하였다. 또한 부호화기의 알고리즘 구현 과정에서는 필요한 연산과 메모리의 사용이 최소화되도록 최적화 하였다.

본 논문의 구성은 다음과 같다. 2장에서는 MPEG-4 오디오 부호화기에 대해서 살펴보고, 3장에서는 실제로 구현된 MPEG-4 CELP 부호화기에 대한 설명과 구현 결과에 대해 언급하며, 마지막 4장에서 결론을 끝낸다.

II. MPEG-4 오디오 CELP 부호화기의 개요

다른 음성부호화 표준인 G.728(LD-CELP)과 G.729(CS-CELP)가 고정 비트율을 지원하는 것과 달리 MPEG-4 오디오 CELP는 표준화 주파수, 양자화 범위, 보간 여부에 따라 4-24 Kbps 범위의 가변 비트율을 제공한다.

그림 1은 MPEG-4 오디오 CELP 부호화기의 전제적인 구조를 보여주고 있다.

![Diagram 1: MPEG-4 CELP Block Diagram](image)

부호화 하는 과정은 크게 LPC 분석을 하는 부분과 코드북 검색 부분, 비트열을 생성하는 부분으로 나눌 수 있다.

LPC 분석을 하는 부분에서는 입력음성신호를 자기 상관을 이용하여 LP(linear prediction) 계수를 추출하고, 이 LP 계수를 반복적으로 변환하여 양자화 한다.

코드북 검색 과정에서는 가중치를 사용하여 신호를 일치하여 256개로 구성된 작용 코드북과 16개로 구성된 고정 코드북을 검색한다. 작용 코드북 검색은 코드북을 구성하는 256개의 코드워드 중 5개가 먼저 선택되고 이 5개의 코드워드에 포함된 15개를 가진 작용 코드북을 검색한다. 이 15개의 작용 코드워드를 가지고 합성한 음성신호와 입력음성신호의 오차가 최소가 되는 코드워드를 검색하여 그에 작용 코드북 인덱스(Shape Delay)와 이득값(Gain Indices) 0을 전송하게 된다. 식(1)은 작용 코드북 검색의 함수이다.

\[
r_{0}(r) = \frac{(\sum_{n=0}^{N-1} d[n]: v[r][n])^2}{\sum_{n=0}^{N-1} v[r][n]}
\]

고정 코드북 검색은 할스의 위치에 따라 16개의 코드워드를 구성하고 그 코드워드 중 5개를 먼저 선택한다. 고정 코드북 검색을 위한 신호는 작용 코드북에 대한 기여량을 계산함으로써 얻어진다. 그 5개의 고정 코드워드를 합성한 음악 입력음성신호의 오차가 최소가 되는 코드워드를 찾아 그에 작용 코드북 인덱스(Shape Index)와 이득값(Gain Indices 1)을 전송하게 된다. 식 (2)는 고정 코드북 검색을 위한 함수이다.

\[
r_{f}(k) = \frac{(\sum_{n=0}^{N-1} g[k][n]: v[k][n])^2}{\sum_{n=0}^{N-1} v[k][n]}
\]

마지막으로 이렇게 입력음성신호로부터 추출된 LP계수와 코드북 인덱스, 이득값들을 비트열로 만드는 과정이 있다. MPEG-4 CELP의 비트열은 그림 2와 같이 이루어져 있다.

![Diagram 2: MPEG-4 CELP Bitstream](image)
III. MPEG-4 오디오 CELP 부호화기의 구현

부호화기의 전체적인 구성은 그림 3과 같으며 프로세스로 코어의 구성은 레지스터 클록, RAM, ROM, 연산 플록과 이를 제어하는 제어 모듈로 구성된다. 연산기에 서 공생기와 나뭇잎생기는 24비트로 구성하고 랜덤화 및 랜덤은 연산시 사용되는 오차의 방향을 위하여 48비 토로 구성하였다. 제어모듈은 부호화 정보를 가져지고 입력신호와 각 플록의 입·출력을 클록에 동기하여 프레임별로 처리한다.

\[ H(x) = \frac{1}{A(x)} = \frac{1}{1 - \sum_{k=0}^{n} a_k \cdot x^{-k}} \] (3)

전산된 LP 계수는 적응하기 위한 반사계수로 변화되며, 변환된 반사계수는 테이블(lock-up Table)을 참조 하여 양자화 된다. 양자화 된 반사계수는 각각의 샘플 및 주파수에 사용되는 필터의 차수에 맞게 그룹핑 (Grouping) 과정을 통하여 변환된다. 16kHz 표준화 주파수일 경우에는 20개의 반사계수가 9개의 데이터로 변 환된다.

코드북 검색은 서브프레임 단위로 이루어져 적응 코드북 검색이 이루어진 후 고정 코드북 검색이 이루어진 다. 코드북 검색을 하기 위하여 입력음성신호를 고정된 필터를 통과시켜, 이 가중치 필터를 통과한 신호가 코드북 검색의 입력으로 사용된다. 식(4)는 가중치 함수이다.

\[ W(x) = \frac{1 - \sum_{k=0}^{n} a_k \cdot x^{-k}}{1 - \sum_{k=0}^{n} a_k \cdot x^{-k}} \] (4)

적응 코드북 검색이 고정 코드북 검색은 256개의 16개로 구성된 코드블록 중 1차로 근사화 된 합성필터를 가지고 오차가 최소가 되는 값을 구하게 된다. 식(5)은 근사화 된 합성필터의 함수이다.

\[ S_r(x) = \frac{1}{A_r(x/r)} = \frac{1}{1 - a \cdot r \cdot x^r} \] (5)

최종적으로 검색된 적응 코드블록과 고정 코드블록의 인덱스와 이동값은 비트맵으로 생성하기 위하여 RAM에 저장된다. 코드북 검색과정에서 생성하는 중간 계산결과는 RAM에 따르 저장하지 않고 한 경로를 사용하여 계산 하여 설계의 효율성을 증가 시켰고 불필요한 출력 부분 의 시간이론을 최소화 하였다.

코드북 검색이 이루어진 후에는 검색된 코드블록과 이동값을 가지고 테이블을 생성하여 적응 코드블록을 개선한다. 그림 4는 적응 코드블록의 개선 과정을 보여준다.
그림 4. 코드북 개선 구조.

또 적용 코드북은 식 (6)과 같이 갱신되는데, 이때 메시지 개수를 256번의 이트를 하는 것은 많은 시간과 복잡한 코드를 요구한다. 따라서 큐(Queue)를 이용하여 갱신한 메모리 사용과 연산의 시간을 줄였다.

\[
\begin{align*}
Ac(i) &= Ac(i + j) & i = 0, 1, \ldots, 256 - j \\
Ac(i) &= Ext(i) & i = 0, 1, \ldots, j & j = 24, 30, 40
\end{align*}
\]  

(6)

본 논문에서는 VHDL을 이용하여 MPEG-4 오디오 CELP 부호화기를 구현하였다. 설계된 부호화기는 스파크 양지화 방식과 16kHz의 블록과 주파수를 사용하는 모드를 지원하며 합성 가능한 VHDL 구문만을 이용 ASIC 구현이 가능하도록 설계하였다. 부호화기 구현을 위하여 시스템의 타이밍을 분석하였고 LP 계수의 연산 시간 전환도를 높이기 위하여 디🐇드 연산을 수행하였으며, 프로세서 모드에는 RAM과 ROM의 크기를 그리고 그 부분의 연산에 필요한 계산량을 최적화하기 위하여 알고리즘 효율적으로 적용하였다.

참 고 문헌


표 1. 부호화기의 기능별 수행 분석

<table>
<thead>
<tr>
<th>LPC 분석</th>
<th>적용 코드북</th>
<th>고정 코드북</th>
<th>기타</th>
<th>합계</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clock</td>
<td>8,512</td>
<td>13,304</td>
<td>15,640</td>
<td>9,332</td>
</tr>
<tr>
<td>비율</td>
<td>18%</td>
<td>28%</td>
<td>33%</td>
<td>21%</td>
</tr>
</tbody>
</table>

-564-