

論文95-32B-1-10

LPC 분석 알고리즘의 VHDL 구현

(VHDL Implementation of an LPC Analysis Algorithm)

鮮于 明勳 * , 趙 威 德 **

(Myung Hoon Sunwoo, We Duke Cho)

요 약

본 논문은 북미 및 일본 디지털 이동통신의 표준 음성 부호화기인 Vector Sum-Excited Linear Predictive(VSELP) 알고리즘 중 핵심 부분인 Linear Predictive Coding(LPC) 분석을 위한 Fixed Point Covariance Lattice(FLAT) 알고리즘 및 이와 관련된 4차 고역 통과 Infinite Impulse Response(IIR) 필터, covariance 행렬 계산 및 spectral smoothing technique(SST) 알고리즘들을 VHSIC Hardware Description Language(VHDL) 언어로 구현하였다. 현재 디지털 이동통신 단말기에 사용되는 Digital Signal Processor(DSP) chip들은 기존의 범용 DSP chip들을 개조하여 사용하고 있으며 이들은 디지털 이동통신 알고리즘에는 최적이라 할 수 없기 때문에 알고리즘들을 분석하여 이에 효율적인 architecture를 설계할 필요가 있다. 본 연구는 디지털 이동통신 전용의 Application Specific Integrated Circuit(ASIC) chip 설계의 기초 연구이며 구현한 VHDL 코드(code)는 추후 논리합성을 통하여 LPC용 ASIC chip 및 DSP chip 설계시 사용될 것이다. VHDL 코드의 정확성을 확인하기 위해 C 코드도 함께 구현하여 실제 음성신호에 대해 두 코드의 시뮬레이션 결과가 동일함을 확인하였다.

Abstract

This paper presents the VHSIC Hardware Description Language(VHDL) implementation of the Fixed Point Covariance Lattice(FLAT) algorithm for an Linear Predictive Coding(LPC) analysis and its related algorithms, such as the forth order high pass Infinite Impulse Response(IIR) filter, covariance matrix calculation, and Spectral Smoothing Technique(SST) in the Vector Sum Exited Linear Predictive(VSELP) speech coder that has been Selected as the standard speech coder for the North America and Japanese digital cellular. Existing Digital Signal Processor(DSP) chips used in digital cellular phones are derived from general purpose DSP chips, and thus, these DSP chips may not be optimal and effective architectures are to be designed for the above mentioned algorithms. Then we implemented the VHDL code based on the C code. Finally, we verified that VHDL results are the same as C code results for real speech data. The implemented VHDL code can be used for performing logic synthesis and for designing an LPC Application Specific Integrated Circuit(ASIC) chip and DSP chips. We first developed the C language code to investigate the correctness of algorithms and to compare C code results with VHDL code results block by block.

* 正會員, 亞洲大學校電子工學科
(Dept. of Electronics Eng., Ajou Univ.)

(Korea Electronics Technology Institute)
接受日字 : 1994年 8月 29日

** 正會員, 電子部品綜合技術研究所

I. 서론

디지털 이동통신 시스템은 코딩(coding) 기술을 적용시켜 동일한 대역폭에 통화 회선 수를 증가시킬 수 있고 암호화가 용이할 뿐 아니라 잡음에 강한 장점들 때문에 현재의 아날로그 이동통신 시스템은 디지털 이동통신 시스템으로 대체되고 있다. 음성신호를 코딩하는 방법에는 waveform 양자화, voice 부호화기 및 hybrid 코딩등이 있다.^[1,2] 이들 중 hybrid 코딩 방법은 벡터 양자화(vector quantization)에 기본을 둔 codebook-based 부호화기인데 예를 들면 Code Excited Linear Predictive(CELP) 부호화기^[3], Stochastically Excited Linear Prediction (SELP) 부호화기^[4], VSELP 부호화기^[5] 등을 들 수 있다. 이는 압축률이 낮은 반면 비교적 높은 품질의 음성을 재생할 수 있다.

위의 codebook-based 부호화기들은 고율의 데이터 압축이 가능하나 codebook을 검색하는데 많은 시간이 소요된다. 또한 디지털 이동통신 단말기에서 모뎀, 무선 표준규격 프로토콜 처리 등도 함께 수행해야 함으로 많은 양의 계산을 실시간으로 처리하기 위해 고속의 DSP chip 뿐만 아니라 Microcontroller, ASIC 등도 함께 사용되어야 한다. 현재 디지털 이동통신에 사용되는 DSP chip들은 처음부터 전용으로 설계된 chip들이 아니라 범용의 DSP chip들을 개조하여 사용함으로써 이동통신용 알고리즘에 최적이라 할 수 없다. 그러므로 디지털 이동통신과 같은 특정된 목적을 위해서는 알고리즘에 적합한 전용의 ASIC을 설계하면 기능, 속도 및 비용면에서 유리하다.

본 논문에서는 북미 및 일본 디지털 이동통신의 표준 음성 부호화기인 VSELP 알고리즘 중 핵심 부분인 LPC 분석을 위한 FLAT 알고리즘, 신호 개선을 위한 4차 고역 통과 IIR 필터, covariance 행렬 계산, SST 알고리즘^[6]을 VHDL^[7]로 구현하여 효율적인 ASIC 구조 설계를 위한 기초 연구를 목적으로 한다. Vantage(TM) VHDL simulator를 사용하였으며 behavior 및 structure 모델을 구현, 실제의 음성신호에 대해 C 코드의 출력과 비교하여 VHDL 시뮬레이션 결과를 확인하였다.

본 논문의 구성은 다음과 같다. II장에서는 VSELP 음성 부호화기 중 VHDL로 구현키 위한 LPC 분석 및 관련 알고리즘에 관해 기술한다. III장에서는 알고리즘의 C 언어 구현에 관해 기술하고, IV장에서는 VHDL behavior 및 structure 모델을 사용한 알고리즘의 구현을 기술한다. V장에서는 C 언어로 구현한 FLAT 알고리즘과 VHDL 시뮬레이션 결과에 대해 논의한다.

마지막으로 VI장에서는 결론을 기술한다.

II. LPC 분석을 위한 FLAT 및 관련 알고리즘

그림 1은 VSELP encoder중 FLAT 알고리즘과 관련된 기능 블럭들을 나타낸다. 아날로그 음성신호는 8 KHz의 샘플링 주파수(sampling frequency)로 동작하는 A/D 변환기(converter)를 거쳐 디지털 음성신호로 변환된다. 이 디지털 신호는 4차 고역 통과 IIR 필터를 거친다. Covariance 행렬은 이 필터를 거친 음성신호 $s(n)$ 으로 부터 구한다^[5]. 이 covariance 행렬로 부터 생성되는 세개의 에러(error) 행렬을 사용하는 FLAT 알고리즘은 LPC 분석을 위한 하나의 방법으로써 반사 계수(reflection coefficient) r_i 를 구한다. LPC 분석 전에 2진 윈도우(window) 함수를 이용한 SST^[6]를 covariance 행렬에 적용한다.

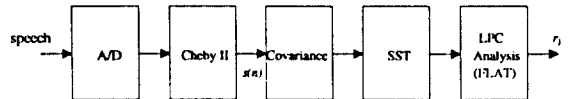


그림 1. VSELP encoder 중 FLAT 알고리즘 부분
Fig. 1. The FLAT algorithm part in the VSELP encoder.

4차 고역 통과 IIR 필터는 A/D 변환기를 거쳐서 나오는 디지털 음성신호의 DC성분을 제거하기 위한 필터로서 그 응답 특성은 60 Hz에서 -40 dB 및 120 Hz에서 -3 dB 감소한다^[8,9]. 전달함수 및 계수들은 다음과 같이 표현되고 그 계수들은 참고문헌에 기술되어 있다.

$$H_{hp} = \frac{\sum_{i=0}^4 a_i z^{-i}}{1 - \sum_{i=1}^4 b_i z^{-i}} \quad (1)$$

LPC 분석을 위하여 입력된 음성 신호를 $s(n)$ 이라 할때, covariance 행렬을 계산하는 식은 다음과 같이 표현된다.

$$\phi(i, k) = \sum_{n=N_s}^{N_s-1} s(n-i)s(n-k) \quad \text{for } 0 \leq i, k \leq N_p \quad (2)$$

여기서 N_a 는 FLAT 분석을 위한 간격으로 샘플 수 170이고, N_p 는 LPC 계수의 갯수인 short-term predictor 차수로서 VSELP에서는 10을 사용한다.

SST^[6]는 대역폭 underestimation으로 인한 문제점을 극복하기 위해 대역폭을 확장하는 방법으로 식은 다음과 같이 표현된다.

$$\phi'(i, k) = \phi(i, k)w(|i-k|) \quad (3)$$

여기서 윈도우 계수 $w(i)$ 의 계산과정 및 실제값은 논문^[8]에 기술되어 있다.

FLAT에서 LPC 필터 계수를 계산하기 위한 lattice 알고리즘은 매 단마다 잔차(residual) 에너지를 최소로 줄이는 역 lattice 필터를 만드는 알고리즘으로 간주할 수 있다. r_j 는 j 번째의 반사계수이며, 순방향 예측 에러(forward prediction error) $f_j(n)$ 과 역방향 예측 에러(backward prediction error) $b_j(n)$ 으로 구성되는 2개의 잔차 에러가 출력된다.

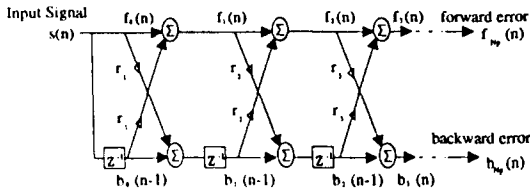


그림 2. 선형 예측 필터의 lattice 구조

Fig. 2. The lattice structure of a linear predictive filter.

그림 2에서 볼 수 있듯이 $f_j(n)$ 과 $b_j(n)$ 은 다음과 같은 재귀적인 관계를 가지고 있다.

$$f_j(n-i) = f_{j-1}(n-i) + r_j b_{j-1}(n-i-1) \quad 0 \leq i \leq N_p \quad (4)$$

$$b_j(n-i) = r_j f_{j-1}(n-i) + b_{j-1}(n-i-1) \quad 0 \leq i \leq N_p \quad (5)$$

위의 $f_j(n)$ 과 $b_j(n)$ 을 사용하여 short-term 상관함수(correlation) 행렬식들을 구할 수 있는데 $f_j(n)$ 의 자기상관함수(autocorrelation) 행렬식은 다음과 같다.

$$F_j(i, k) = \sum_{n=N_p}^{N_p-i} f_j(n-i) f_j(n-k) \quad 0 \leq i \leq N_p \quad (6)$$

$b_j(n-1)$ 의 자기상관함수 행렬식의 표현은

$$B_j(i, k) = \sum_{n=N_p}^{N_p-i} b_j(n-i-1) b_j(n-k-1) \quad 0 \leq i \leq N_p \quad (7)$$

이며, $C_j(n)$ 은 $f_j(n)$ 과 $b_j(n-1)$ 의 상호상관함수(crosscorrelation) 행렬식으로서 다음과 같이 표현된다.

$$C_j(i, k) = \sum_{n=N_p}^{N_p-1} f_j(n-i) b_j(n-k-1) \quad 0 \leq i \leq N_p \quad (8)$$

위의 (6), (7) 및 (8)식을 정리하면 다음의 식들로 나타낼 수 있다.

$$F_j(i, k) = F_{j-1}(i, k) + r_j(C_{j-1}(i, k) + r_j^2 B_{j-1}(k, i)) \quad (9)$$

$$B_j(i, k) = B_{j-1}(i+1, k+1) + r_j(C_{j-1}(i+1, k+1) + C_{j-1}(k+1, i+1)) + r_j^2 F_{j-1}(i+1, k+1) \quad (10)$$

$$C_j(i, k) = C_{j-1}(i, k+1) + r_j(B_{j-1}(i, k+1) + F_{j-1}(i, k+1)) + r_j^2 C_{j-1}(k+1, i) \quad (11)$$

입력 신호가 통계적으로 stationary하고 lattice 필터계수들이 안정 상태에 있다고 가정하면, 반사 계수값은 다음의 식에서 얻어질 수 있다.^[8]

$$r_j = -2 \frac{C_{j-1}(0, 0) + C_{j-1}(N_p-j, N_p-j)}{F_{j-1}(0, 0) + F_{j-1}(N_p-j, N_p-j) + B_{j-1}(0, 0) + B_{j-1}(N_p-j, N_p-j)} \quad (12)$$

이때 구해지는 10개의 반사 계수 r_j 값은 j 의 값에 따라 대응되는 10개의 미리 계산한 look-up 테이블에 의해 독립적으로 양자화된다.^[8]

이와 같은 FLAT 알고리즘을 수행하기 위한 절차는 다음과 같이 요약할 수 있다.

1. 식 (2)에 의한 covariance 행렬 계산
2. 초기 행렬식 계산

$$F_0(i, k) = \phi(i, k), \quad 0 \leq i, k \leq N_p - 1$$

$$B_0(i, k) = \phi(i+1, k+1), \quad 0 \leq i, k \leq N_p - 1$$

$$C_0(i, k) = \phi(i, k+1), \quad 0 \leq i, k \leq N_p - 1$$

3. $j = 1$ 설정
4. 식 (12)에 의한 r_j 계산
5. look-up 테이블을 이용한 r_j 양자화
6. $j = N_p$ 이면 완료
7. 다음 반사 계수 계산을 위한 행렬식 계산
 - 식 (9)에 의한 $F_j(i, k)$ 계산, $0 < i, k \leq N_p - j - 1$
 - 식 (10)에 의한 $B_j(i, k)$ 계산, $0 < i, k \leq N_p - j - 1$
 - 식 (11)에 의한 $C_j(i, k)$ 계산, $0 \leq i, k \leq N_p - j - 1$
8. $j = j + 1$ 으로 설정하고 4번째 과정부터 다시 수행

III. LPC 분석 알고리즘의 C 언어 구현

앞에서 설명한 FLAT 및 관련 알고리즘의 전체적인 동작 여부를 확인하고 VHDL 모델 구현 시 각 기능의 출력을 확인할 수 있도록 알고리즘들을 C 언어로 구현

하였다. 고역 통과 필터의 계수, SST 기법을 위한 계수 및 반사 계수를 양자화하기 위한 look-up 테이블들은 모두 헤더(header) 파일로 포함시켜 사용하였다. 그 알고리즘의 순서도는 그림 3과 같다.

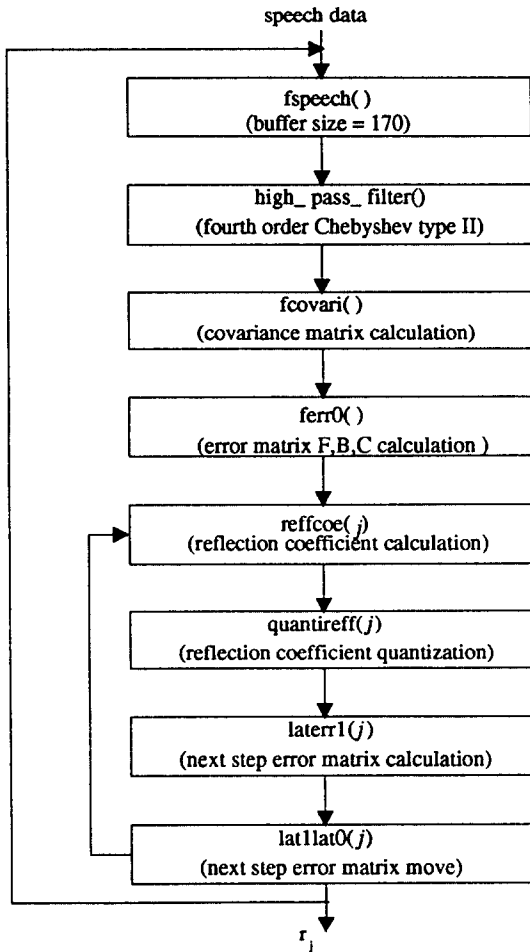


그림 3. C 언어를 위한 시뮬레이션 알고리즘 순서도

Fig. 3. The flowchart of simulation algorithms for C language.

fspeech 함수(function)는 음성 데이터 파일에서 160개씩의 샘플들을 읽어 내부 버퍼(buffer)에 저장한다. 버퍼에서 출력을 시킬 때는 이전 프레임(frame)의 마지막 10개의 데이터를 합하여 170개의 데이터를 갖고 FLAT 알고리즘을 수행하는데 이는 프레임 단위로 데이터가 처리되더라도 각 프레임간의 연관성을 유지하여 음성의 질을 높이기 위함이다. high_pass_filter 함수는 식 (1)의 4차 고역 통과 IIR 필터를 수행한다.

fcovari 함수는 covariance 행렬을 계산하고 SST

를 수행한다. covariance 행렬을 계산하는 부분은 식 (2)에서의 계산과정을 수정하여 계산의 양을 다음과 같이 대폭 줄일 수 있다. 식 (2)를 살펴보면 첫째, $\phi(i, k) = \phi(k, i)$ 임을 알 수 있고 이는 대각선 방향에 대해 대칭이므로 계산량의 절반을 줄일 수 있다. 둘째, covariance 행렬 $\phi(i, k)$ 의 i 가 0인 경우 즉, 첫째행의 원소들을 계산한 후에는 대각선 방향의 $\phi(i, k)$ 의 원소에 대하여는 i 와 k 에 따라서 더해지거나 빠지는 $s(n-i)s(n-k)$ 항 만을 추가로 계산하면 된다. 이와 같은 구조에 의하여 하나의 covariance 행렬 원소를 계산하는데 160개의 음성신호가 전부 필요하지 않도록 하여 계산량을 더욱 줄일 수 있다.

ferr0 함수는 식 (4) 및 (5)에 의하여 최초의 여러 행렬을 계산하는 함수이며, reffcoe(j)는 원하는 10개의 LPC값을 식 (12)에 의하여 계산하는 함수이다.

quantireff(j)는 reffcoe(j)에 의하여 계산된 10개의 반사 계수를 look-up 테이블에 의해 양자화하는 함수이다. look-up 테이블들은 이진 검색(binary search) 알고리즘을 사용하여 신속히 찾을 수 있도록 설계하였다. laterr1(j)과 lat1lat0(j)는 다음 r_j 계산을 위한 여러 행렬을 만드는 함수이다. 10개의 LPC 값이 모두 구해지면 다시 데이터 파일에서 160개의 데이터를 받아들여 위와 같은 계산을 반복한다.

IV. LPC 분석 알고리즘의 VHDL 구현

VHDL 구현시에는 chip 설계 시간을 줄일 수 있고 그 설계 방법이 간단하여 현재 ASIC 설계시 많이 사용되는 top-down 설계방법을 사용하여 먼저 가장 상위 레벨에서 본 구조를 behavior 모델로 구현하여 이의 시뮬레이션을 완료하였다. 다음으로 III장에서 제시한 C 언어의 순서도를 분석하여 각각의 C언어 함수들에 알맞게 상위 레벨 VHDL 모델을 소 블록으로 나누어 각각을 behavior 모델로 구현하였으며, 이의 시뮬레이션을 완료하였다. 마지막으로 완성된 각 블록을 VHDL structure 모델의 positional mapping 방법을 사용하여 그림 4와 같이 구성하였다.

그림 4에서 Text I/O 컴포넌트(component)는 음성 데이터를 파일에서 읽어들이기 위한 컴포넌트이며 buffer 및 high pass filter 컴포넌트는 두개의 제어 신호로 연결되어 있으며 동작 설명은 다음과 같다. buffer 컴포넌트가 음성신호를 받아들여 170개의 버퍼가 모두 차면 high pass filter 컴포넌트에게 buffer_full 신호와 버퍼에 저장해 두었던 데이터를 data_bus에 실는다. buffer_full 신호를 감지한 high

pass filter 컴포넌트는 data_bus에 실린 데이터를 받아들여 고역 통과 필터를 수행한다. data_bus로 들어온 모든 신호에 대하여 high pass filter 컴포넌트의 동작이 끝나면 buffer 컴포넌트에게 종료를 알리는 ack_buffer_full 신호를 주고 새로운 데이터를 받아들일 준비를 한다. 이와 같은 신호를 둔 이유는 입력으로 음성신호를 받아들이는 부분과 반사 계수를 계산하는 부분을 분리시켜 병렬로 동작시키기 위함이다. 반사 계수를 계산하는데 사용되는 클럭은 8 KHz의 표준 음성 샘플링 A/D 변환기의 클럭보다도 빠른 클럭을 사용하였는데 이는 다음 음성 프레임이 들어오기 전에 현재 수행중인 반사 계수의 계산을 완료해야 하기 때문이다. 반사 계수를 계산하는 클럭은 1.8 MHz 이상으로 동작을 하면 한 프레임에 대한 계산을 실시간 처리할 수 있다.

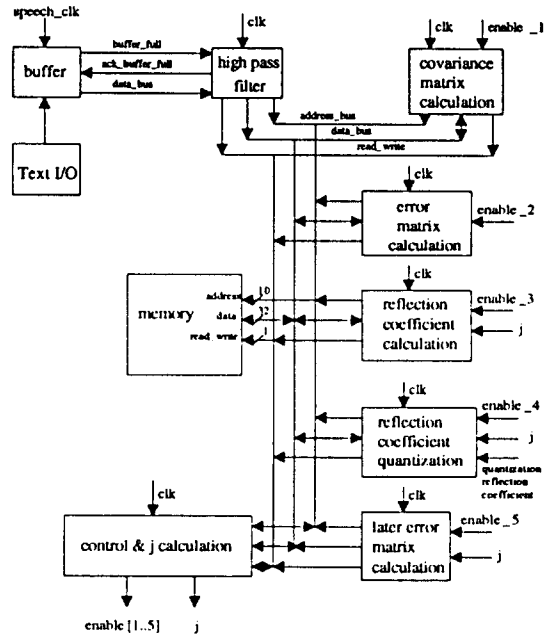


그림 4. LPC 분석을 위한 VHDL structure mapping 모델

Fig. 4. VHDL structure mapping model for an LPC analysis.

control & j calculation 컴포넌트는 일종의 sequencer로서 각 컴포넌트들을 순서적으로 수행시키기 위함이며 각 컴포넌트가 마지막으로 사용하는 기억소자 주소를 감지하면 다음 컴포넌트에게 enable 신호를 주어 다음 컴포넌트가 동작을 시작하게 하는 역할을 한다. covariance matrix calculation 컴포넌트는 high pass filter 컴포넌트가 기억소자에 저장한 데이

타를 읽어 계산을 수행한다. 여기서 C 언어로써 covariance 행렬 계산시 3절에서 설명한 식 (2)의 변형된 계산 절차를 사용하였다. 이 컴포넌트는 enable_1 신호에 의해 동작을 시작하며 완료되면 sequencer에 의하여 정지된다.

error matrix calculation 컴포넌트는 식 (4) 및 (5)의 계산을 수행하며 SST를 수행하는 부분도 포함하고 있다. 이는 covariance matrix calculation 컴포넌트에서 계산하는 것보다 이 컴포넌트에서 기억소자를 읽을 때 일률적으로 SST를 수행하는 것이 더 빠르고 간단한 VHDL 코드를 만들 수 있기 때문이다. 이 컴포넌트는 enable_2 신호에 의해 동작을 시작하며 식 (4) 및 (5)에 의하여 계산이 완료되면 sequencer에 의하여 정지된다. reflection coefficient calculation 컴포넌트는 식 (12)를 구현한 것이며 reflection coefficient quantization 컴포넌트는 식 (12)에 의하여 만들어진 반사 계수를 look-up 테이블에서 이진 검색 알고리즘을 사용하여 가장 근사한 값을 찾아 양자화시킨다. later error matrix calculation 컴포넌트는 식 (9), (10) 및 (11)을 이용하여 다음 r_j 를 구하기 위한 상관함수 행렬들을 계산하기 위한 컴포넌트이다. 위의 세개의 컴포넌트, 즉 reflection coefficient calculation, reflection coefficient quantization 및 later error matrix calculation을 계산하는 컴포넌트들은 sequencer에 의하여 enable 신호와 r_j 의 순서를 나타내는 j의 값을 필요로 한다.

구현한 VHDL 코드에서 필요한 기억소자는 32 비트의 부동소수점 데이터를 저장하는 1차원 배열로 되어 있다. C 언어로 구현할 때 사용한 2차원 array들은 모두 1차원 배열로 기억소자의 주소를 할당하였다. 할당된 기억소자는 모두 601까지의 번지를 갖게 되며 2.4 Kbyte가 필요하다.

V. 시뮬레이션 결과 및 검토

C 언어 및 VHDL 언어 시뮬레이션 시에 사용된 음성 데이터는 동일한 남성의 음성 데이터를 사용 VHDL 시뮬레이션의 결과 값과 C 언어 시뮬레이션의 결과 값이 서로 동일함을 확인하였다. 예로서 C 언어 시뮬레이션에서 나온 두개의 프레임에 대한 양자화된 반사 계수들은 표 1과 같다. 두개의 프레임에 대한 VHDL 시뮬레이션 결과를 보면 그림 5와 같다. 맨 아래의 파형이 우리가 원하는 값으로 j값이 1에서 10까지 변함에 따라 출력되는 10개의 양자화된 반사 계수

이며 이는 위의 C 시뮬레이션에서 얻은 값과 동일함을 알 수 있다.

표 1. 두개의 프레임에 대한 반사 계수들
Table 1. Reflection coefficients for two frames.

프레임 1	프레임 2
r(0) = -0.97110736370	r(0) = -0.95970594883
r(1) = -0.15857782960	r(1) = 0.06527176499
r(2) = -0.32762703300	r(2) = -0.21394510567
r(3) = 0.31384372711	r(3) = 0.23812386394
r(4) = -0.13957321644	r(4) = -0.35039535165
r(5) = 0.26855635643	r(5) = 0.15452256799
r(6) = -0.30078113079	r(6) = -0.17507396638
r(7) = 0.28120946884	r(7) = 0.28120946884
r(8) = -0.02935140580	r(8) = -0.02935140580
r(9) = 0.19580543041	r(9) = 0.19580543041

j가 5 이상의 경우에 표시된 심볼인 '*'는 Vantage VHDL simulator의 window size가 작아서 데이터를 전부 표시하지 못함을 나타낸다.

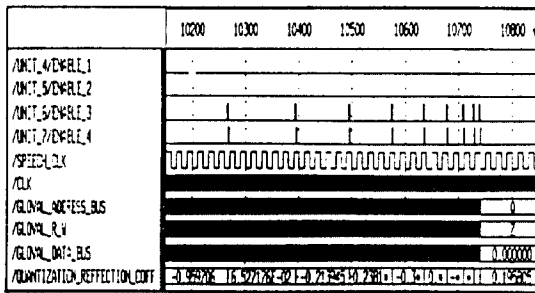
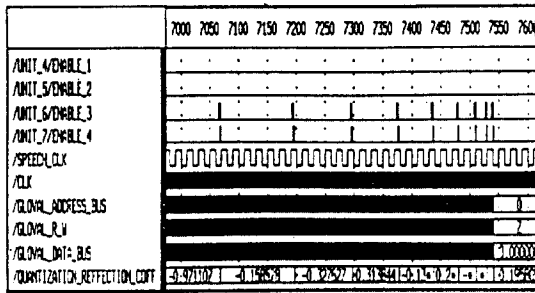


그림 5. 두개의 프레임에 대한 Vantage 시뮬레이션 윈도우

Fig. 5. Vantage simulation windows for two frames.

FLAT 알고리즘의 VHDL 시뮬레이션을 동작시키는 클럭 속도는 앞서도 언급한 바와 같이 1.8 MHz 이상이면 되나 여유를 두기 위해서 1.85 MHz로 시뮬레이션 하였으며 1.8 MHz보다 느린 클럭을 사용하여 동작을 시키면 결과가 나오기 전에 다음 프레임 데이터

가 입력되어 데이터가 혼합되므로 실시간 처리를 할 수 없고 결과 값이 틀리게 된다. 사용된 1.85 MHz 클럭 속도는 현재의 ASIC 및 Field Programmable Gate Array(FPGA)의 클럭 속도보다 훨씬 느려 실시간 처리 FLAT 알고리즘의 ASIC 구현시 충분한 여유가 있다.

VI. 결론

본 논문에서는 VHDL 언어를 사용하여 VSELP 알고리즘 중 핵심부분인 LPC 분석을 위한 FLAT 알고리즘 및 관련 알고리즘들의 behavior 및 structure 모델을 구현하였고 시뮬레이션을 통하여 결과값을 확인하였다. LPC 분석을 위한 다른 알고리즘들에 대해서도 구현할 수 있고 이는 곧 LPC용 ASIC chip 구현 및 더 나아가 하나의 디지털 이동통신용 ASIC chip 구현을 위한 기초 연구이다. 실시간 FLAT 알고리즘을 위한 클럭 동작 속도는 최소 1.8 MHz 이상을 요구하며, 이는 범용의 DSP chip 동작 속도인 40-80 MHz보다 훨씬 낮은 속도에서 동작을 시킬 수 있다. 계산에 필요한 기억소자 용량 역시 2.4 Kbyte로서 충분하고, static 기억소자와 같은 빠른 속도의 기억소자도 필요치 않다. 이 또한 이동통신용 DSP chip은 일반적으로 프로그램 및 데이터 저장을 위하여 약 6 Kword, 즉 12 Kbyte의 내부 static 기억소자를 요구하나 이보다 훨씬 작고 느린 dynamic 기억소자를 사용할 수 있다.

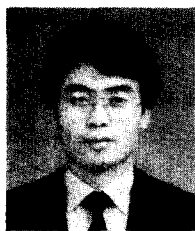
추후 계속 연구해야 할 분야는 현재 부동소수점 시뮬레이션을 수행하였으나 내부 버스 크기 및 기억소자 용량을 줄여 전체 chip 면적을 줄이기 위하여 정수 시뮬레이션을 위한 VHDL 코드를 개발하고 논리 합성을 통하여 LPC용 및 이동통신 알고리즘에 적합한 ASIC chip 설계에 있다. 본 논문에서 선택한 방법론이나 structure 모델은 최적이라 할 수 없으며 이를 위한 연구도 계속 수행할 예정이다.

참고 문헌

[1] B. S. Atal and L. R. Rabiner, "Speech research directions," *AT&T Technical Journal*, vol. 65, issue 5, Sep./Oct. 1986.
[2] N. S. Jayant, "High-quality coding of telephone speech and wideband audio."

- IEEE Commun. Magazine, vol. 28, pp. 10-20, Jan. 1990.
- [3] M. R. Schroeder and B. S. Atal, "Code-excited linear prediction (CELP): high-quality speech at very low bit rates," in Proc. Int. Conf. Acoust., Speech, Signal Processing, pp. 937-940, 1985.
- [4] W. B. Kleijn, D. J. Krasinski, and R. H. Ketchum, "Improved speech quality and efficient vector quantization in SELP," in Proc. Int. Conf. Acoust., Speech, Signal processing, pp. 155-158, 1988.
- [5] I. A. Gerson and M. A. Jasiuk, "Vector sum excited linear prediction (VSELP) speech coding at 8kbps," in Proc. Int. Conf. Acoust., Speech, Signal Processing, pp. 461-464, Apr. 1990.
- [6] Y. Tohkra, F. Itakura, and S. Hashimoto, "Spectral Smoothing Technique in PAR COR Speech Analysis-Synthesis," IEEE Trans. ASSP, vol. ASSP-26, no. 6, pp. 587-595, Dec. 1978.
- [7] ANSI/IEEE Std 1076-1993 IEEE Standard VHDL Language Reference Manual, IEEE, Jun. 1994.
- [8] Myung H. Sunwoo and Sangil Park, "Real-time implementation of the vselp on a 16-bit DSP chip," IEEE Trans. on Consumer Electronics, vol. 37, no. 4, pp. 772-782, Nov. 1991.
- [9] Cellular System Dual-Mode Mobile Station-Base Station Compatibility, Standard EIA/TIA-PN 2759(IS-54-8), Aug., 1991.

 저 자 소 개



鮮于明勳(正會員)

1980년 2월 서강대학교 전자공학과 졸업. (공학사) 1982년 2월 한국과학기술원 전기 및 전자공학과 졸업. (공학석사) 1982년 3월 ~ 1985년 8월 한국전자통신연구소 연구원. 1990년 8월 Univ. of Texas at Austin. (공학박사) 1990년 8월 ~ 1992년 8월 Motorola DSP Chip Operation 연구원. 1992년 8월 ~ 현재 아주대학교 전자공학과 조교수. 주관심분야 통신 및 신호처리 ASIC 설계, Parallel Architecture, VLSI 구조 및 설계, 디지털 이동통신



趙威德(正會員)

1958년 11월 17일생. 1981년 2월 서강대학교 전자공학과 졸업. (공학사) 1983년 2월 한국과학기술원 전기 및 전자공학과 졸업. (공학석사) 1987년 2월 한국과학기술원 전기 및 전자공학과 졸업. (공학박사) 1983년 ~ 1990년 3월 금성전기 (주) 기술연구소. 1990년 4월 ~ 1991년 1월 생산기술연구원 HDTV 사업단. 1991년 1월 ~ 현재 전자부품종합기술연구소 ATEL 사업단 디지털 이동통신 개발팀장. (수석연구원) 주관심분야 데이터 통신 프로토콜 설계, 디지털 이동통신 시스템 설계, 통신 신호처리 알고리즘 설계