

VHDL을 사용한 ADPCM analog/digital변환기의 구현에 관한 연구

김재현⁰, 이준용

홍익대학교

kjh@cs.hongik.ac.kr, jlee@cs.hongik.ac.kr

Implementation of ADPCM analog, digital converter with VHDL

Jae-Hyun Kim⁰ Jun-Yong Lee

Computer Engineering, Hong-Ik University

요 약

컴퓨터의 보급과 통신의 발달에 따라 많은 자료가 모뎀 등의 장비를 통하여 네트워크 상에서 이동하고, 인간의 음성, 혹은 소리가 컴퓨터에 사용되게 되었다. 이에 따라 소리, 네트워크 상의 자료와 같은 아날로그 신호와 컴퓨터가 처리할 수 있는 디지털 신호간의 변환이 필요하게 되었다. 본 연구에서는 CCITT에서 제안한 아날로그, 디지털 변환 알고리즘인 ADPCM 알고리즘, G.721, G.726에 따라 아날로그, 디지털 변환기를 하드웨어 기술언어인 VHDL을 사용하여 실제적으로 구현하였다.

1. 서론

CCITT에서 제안한 G.721, G.726 ADPCM(Adaptive Difference Pulse Code Modulation) 아날로그, 디지털 변환 알고리즘은 입력신호인 아날로그 신호를 그 위상값으로 표현하던 PCM, 이전 샘플과의 차이로 표현한 DPCM을 발전시킨 것으로 이전 샘플들에서부터 추측한 예측값과 실제 입력신호와의 차이를 디지털 신호로 변환하여 표현하는 알고리즘이다.

PCM 알고리즘의 경우 8비트의 디지털 신호로 아날로그 신호를 표현하게 되는데 반해서 ADPCM 알고리즘의 경우, 4비트의 디지털 신호만으로 PCM, DPCM 알고리즘에 따른 표현이상의 정확도를 나타내게 된다.

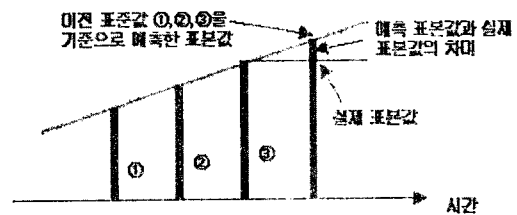
ADPCM을 사용한 아날로그, 디지털 변환의 경우, PCM-ADPCM-PCM의 탄젠트곡을 사용하여 PCM에서 일차적으로 8비트의 디지털값으로 아날로그 신호를 변환하고, 그 변환된 출력값을 ADPCM 로직의 입력으로 사용하여 변환에 사용하여 4비트의 디지털 출력값을 생성한다.

본 논문에서는 이러한 ADPCM 알고리즘에 대해서 설명하고, 하드웨어 표현언어인 VHDL을 사용하여 Altera사의 Max plus-II를 사용하여 구현, 합성한 결과에 대해서 이야기한다.

2. ADPCM 알고리즘

ADPCM 알고리즘의 기본은 <그림 1>에서 보이는 것과 같이 이전 샘플값들의 증가, 감소폭에 따라서 그 다음 샘플의 값을 예측하고, 이 예측신호의 값과 실제 샘플값의 차이인 차분

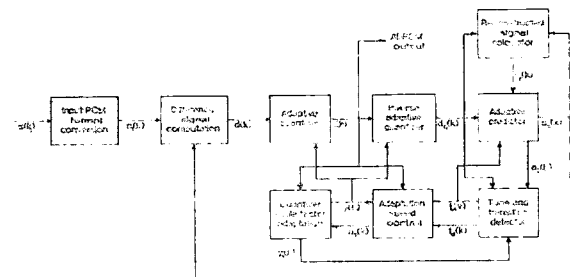
신호(Difference signal)를 양자화하여 아날로그, 디지털 변환을 하는 것이다.



<그림 1> ADPCM 기본원리

변환기의 전체적인 구성은 <그림 2>, <그림 3>과 같은 블록 형태를 가지게 된다. ADPCM 알고리즘은 각 블록에 대한 설명을 통해서 이야기하도록 한다.

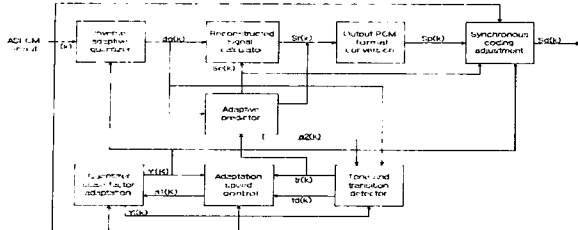
변환기는 입력이 변환방법 A-law와 μ -law에 따라 다르기 때문에 "Input PCM format conversion" 블록을 통해서 정형화



<그림 2> 변환기 중 인코더의 블록 다이어그램

1) 이 논문은 IDEC에서 제공한 소프트웨어를 사용하여 작성되었음

시키고, 이 입력값과 Adaptive predictor에서 예측한 값과의 차분신호를 "Difference signal computation" 블록을 통해서



<그림 3> 변환기 중 디코더의 블록 다이어그램

구하여, "Adaptive quantizer" 블록에서 양자화를 하여 4비트의 출력신호를 얻는다. ADPCM 아날로그, 디지털 변환기의 경우, 일반적인 소리신호와 데이터의 전송에 따라 다른 처리를 해야 한다. 이는 데이터의 전송의 경우, 그 값의 변화폭과 속도면에서 소리신호보다 작기 때문에 예측을 할 때, 고려를 해야하기 때문이다. 이러한 차이를 가지는 소리신호와 데이터 전송은 "Quantizer scale factor adaptation", "Tone and transition detector", "Adaptation speed control" 세개의 블록에서 감지, 조절하게 된다. 예측신호는 두개의 블록 "Adaptive predictor", "Reconstructed signal calculator" 에서 이전 6개의 샘플을 가지고 생성하게 된다.

2.1 적응 양자기(Adaptive quantizer)

적응 양자기는 표현의 용이함을 위하여 차분신호를 베이스 2 로그연산을 취하여 양자화를 하게 된다. 베이스 2 로그연산을 위해서 2의 보수의 형태로 표현된 것을 부호절대값 형태로 변형한 후, 지수와 가수를 추출한다. 지수는 절대값의 범위에서 0 부터 14의 값을 결정해서 추출하고, 가수는 로그의 근사²⁾를 통해서 생성한다.

양자화는 베이스 2 로그연산을 취한 값과 부호를 보고 양자화 테이블로부터 해당값을 구하여 수행하게 된다. 양자화 테이블을 양자화 속도가 40, 32, 24, 16Kbps 중 어떤 속도로 하게되느냐에 따라서 각각 다른 테이블이 존재하게 되는데, 여러 개의 테이블 중에서 하나를 결정하는 것은 양자화 레벨의 결정을 통해서 이루어진다. 하지만, 일반적으로 양자화 레벨은 실행단계에서 결정되기 때문에 임베디드(Embedded) 응용 프로그램 등에는 적합하지 않은 단점과 시그널-에러 비율(SNR, Signal to Noise Ratio)는 낮아지는 장점이 있다.

2.2 적응 예측기(Adaptive predictor)

적응 예측기는 예측계수와 이전 6개의 차분신호, 회귀신호

2) $\text{Log}_2(1 + x) = x$

(Reconstructed signal)의 곱을 합산하여, 이전 신호들의 증감폭을 고려, 다음 입력신호를 예측하게 된다. 곱셈을 위하여, 2의 보수와 부호절대값 형태로 되어있는 신호들을 부동소수점(Floating point)으로 바꾸고, 변형한 부동소수점 형태의 신호를 곱한 후, 다시 2의 보수로 변환하게 된다. 곱셈은 지수와 가수를 추출하여, 지수부끼리는 덧셈을, 가수부끼리는 곱셈, 부호는 배타적 논리합(Exclusive OR) 연산을 통해서 수행한다.

3. 블럭별 구성 및 특징

아날로그, 디지털 변환기의 구현은 <그림 2>의 블럭을 CCITT 권고, G.721과 G.726에서 설명한 것과 같이 세부 블럭으로 나누어 구현하는 구조적 기술(Structural description)로 하였다. 사용된 합성(Synthesize) 프로그램은 Altera사의 Max Plus - II 이며, 합성의 대상이 되는 디바이스는 Flex6000 Family이다.

3.1 Input PCM format conversion

이 블럭은 A-law, μ -law의 두가지 방법에 따라 13, 14비트의 다른 길이를 가지는 입력값을 14비트의 동일한 비트를 가지는 정형화된 PCM 신호를 만들어 주는 역할을 하게된다. 이는 CCITT 권고 G.711의 1a, 1b, 2a, 2b 네개의 테이블을 따르고 있다.

3.2 Difference signal computation

이 블럭은 입력신호와 예측신호의 차이를 계산하는 차분신호를 생성하는 블럭으로 14비트 가산기를 사용하여 구현하였다.

3.3 Adaptive quantizer

양자화를 수행하여, 4비트의 디지털 값을 생성한다. 로그연산은 디코더와 가산기, 쉬프트 연산을 사용하여, 구현하였다. 양자화는 CCITT 권고 G.721의 양자화 테이블을 사용하여 구현하였다³⁾.

3.4 Inverse adaptive quantizer

역양자화는 양자화를 거친 4개의 출력값을 다음 신호의 예측을 위해서 다시 16 비트의 차분신호를 재생하는 역할을 하는 것이다. 역양자화는 CCITT 권고 G.721의 역양자화 테이블을 사용하여 구현하였다⁴⁾. 또한 역양자화를 수행하여 얻은 신호는 베이스 2 로그연산을 취한 것이므로 근사⁵⁾를 통하여, 2의 지수승 연산을 수행한 원래의 차분신호를 얻도록 구현하였다.

3) CCITT recommendation G.721 P.243 테이블 참조

4) CCITT recommendation G.721 P.245 테이블 참조

5) $2^x = 1 + x$

3.5 Quantizer scale factor adaptation

소리신호의 처리를 위한 빠른 양자화 스케일링인자, 일반적인 데이터 전송의 처리를 위한 느린 양자화 스케일링인자를 생성하여, 두 인자로부터 다음 입력신호의 스케일링을 위한 스케일링인자를 얻는 블록이다. 두개의 필터와 하나의 크기제한을 위한 블록을 사용하여, 두개의 양자화 스케일링인자를 생성하고, 선형조합기를 거쳐 스케일링인자를 생성하도록 구현하였다.

3.6 Adaptation speed control

소리신호와 데이터 전송에 대한 양자화 속도제어는 매계변수의 차이를 통해서 이루어지고, 이러한 매계변수를 생성하는 블록이다. 세계의 필터, 크기제한블록, F(I)함수, 트리거, 감산기로 이루어져 있다. F(I) 함수는 CCITT 권고 G.726에서 설명하였으며, <표 1>과 같다.

II	7	6	5	4	3	2	1	0
F(II)	7	3	1	1	1	0	0	0

<표 1> F(II) 함수의 정의

3.7 Adaptive predictor

차분신호와 회귀신호를 받아서 다음 입력신호를 예측하는 블록으로 두개의 가산기, 부동소수점 변환기, 2개의 크기제한블록, 트리거, 승산기, 10개의 메모리 블록, 예측신호 생성기로 이루어져 있다.

메모리 블록은 예측기가 받아들인 지난 6개의 차분신호를 저장하여, 차분신호와 예측계수의 곱들의 합을 통하여 예측신호 생성기를 통하여 예측신호를 생성한다.

3.8 Tone and transition detector

소리신호와 데이터전송 두개의 아날로그 신호형태 중 어떠한 형태인지를 감지하는 블록으로, 두개의 감지기, 메모리블록, 트리거로 구성된다. 두개의 감지기를 통하여 입력신호의 값의 범위에 따라서 소리신호인지 데이터전송인지에 대한 두개의 1비트 신호를 세팅한다.

3.9 Output PCM format conversion

이 블록은 "Input PCM format conversion" 블록과 반대의 역할을 하는 것으로 정형화시킨 신호를 A-law 혹은 μ -law로 바꾸어준다. 이는 CCITT 권고 G.711의 1a, 1b, 2a, 2b 네개의 테이블에 의거 구현되었다.

3.10 Synchronous coding adjustment

이 블록은 PCM - ADPCM - PCM 탄뎀 로직에서 ADPCM의 출력이 다시 PCM의 입력으로 들어갈때, PCM에서 받아들

일 수 있는 입력의 형태를 가지도록 다시 인코딩을 하며, 두개의 가산기, 로그연산기, 인코더로 구성된다.

4. 구현결과

3장에서 설명한 각각의 블록을 VHDL(VHSIC Hardware Description Language)을 사용하여 구조적 기술을 통하여 구현하였다. 변환기는 "PCM- ADPCM - PCM"의 탄뎀 로직(Tandem logic)형태로 사용되므로 입력값은 PCM의 출력인 8비트 길이의 디지털 신호를 생성하여 사용하였으며, 여기서 나오는 4비트의 결과물을 가지고서 다시 디코더의 입력으로 넣었을 때, 처음 인코더의 입력값과 동일한 값이 출력됨을 확인하였다.

참고문헌

- [1] 임강빈, 정화자, 정기현, "2진 로그생성을 위한 하드웨어 구조" 전자공학회논문집, 제 32권 B편 제 2호, pp. 67 ~70. 1995
- [2] recommendation G.711 : Pulse Code Modulation of voice frequencies, CCITT, Geneva, 1972
- [3] recommendation G.721 ; 32kbit/s Adaptive Differential Pulse Code Modulation, CCITT, Melbourne, 1988
- [4] Comparison of ADPCM algorithms(Rec. G.726 & G.727), CCITT, Geneva, 1994
- [5] Recommendation G.726 : 40, 32, 24, 16kbit/s Adaptive differential Pulse Code Modulation, CCITT, Geneva, 1990
- [6] Recommendation G.727 : 5-, 4-, 3- and 2-bit/sample embedded adaptive differential pulse code modulation, CCITT,
- [7] Zainalabedin Navabi, "VHDL : Analysis and modeling of digital systems", McGraw-Hill, 1998
- [8] 박담홍, "ADPCM 적응예측기 하드웨어 설계에 관한 연구", 원광대 전자공, 1994