

NEC μ PD7720D DSP를 사용한 32Kbps ADPCM CODEC의 실시간 동작 구현
 Real Time Implementation of 32Kbps ADPCM CODEC on NEC μ PD7720D DSP

이강석*
 이상욱

서울공대제어계측공학과

1. 서론

ADPCM(Adaptive Differential Pulse Code Modulation)은 24-48Kbps에서 음성신호를 간단하면서도 효과적으로 코딩할 수 있는 방법이다 [1-4]. 24 Kbps에서는 컴퓨터에 의해서 조절되는 디지털 음성 응답시스템에서 적용할 수 있는 정도의 음성신호를 재생할 수 있으며 [3], 32 Kbps에서는 원음과 거의 구별할 수가 없는 toll-quality 이다 [4].

ADPCM을 마아드웨어로 구현하는 방법에는 여러 가지가 소개되고 있는데 Adaptive step-size 값을 미리 계산하여 ROM에 저장 시켜놓고 디지털 Adder 와 Subtractor, 카운터등을 사용하여 이를 조절하는 Look-up table 방식[4]와 Bit-Slice 에 마이크로 프로그래밍을 하여 codec을 구성하는 방법 [5] 및 디지털 신호 처리용 프로세서(DSP)를 사용하는 방법등이 있다[9]. DSP의 잇점은 기존의 마이크로 프로세서로써는 불가능한 고속의 곱셈이 가능하다는 점이다. 따라서 DSP를 사용한 방식이 처음의 두 방식 보다 마아드웨어 구성을 간단하게 할 수 있는 잇점이 있다.

본 논문에서는 현재 신호처리에 많이 사용되고 있는 NEC μ PD7720D DSP를 사용하여 실시간 동작이 가능한 voice mail용 시스템을 위한 다중 채널 ADPCM codec을 설계 제작하고자 한다.

2. ADPCM 알고리즘

음성신호에 대한 ADPCM 코딩 알고리즘은 널리 알려져 있으며 여러가지 방법이 소개되었는데 본 논문에서는 Jayant가 제안한 방법[6]을 채택하였다. 그림1에 Jayant가 제안한 ADPCM

블록선도를 도시하였다.

ADPCM은 기본적으로 음성신호의 각 샘플 사이에는 높은 상관관계가 존재하기 때문에, 예측기로 부터 입력신호를 예측할 수 있으며, 또한 예측오차신호가 입력신호보다 dynamic range가 작다는 사실을 이용한다 [2]. dynamic range가 작으면 양자기에 할당하는 bit 수를 줄일수 있다.

먼저 예측 오차신호 e_i 는 입력 s_i 와 예측기의 출력 \hat{s}_i 의 차이이다.

$$e_i = s_i - \hat{s}_i \quad (1)$$

예측기의 출력 \hat{s}_i 는 이전의 표본시간 $i-1$ 에서 계산된 s'_{i-1} 와 음성신호 사이의 상관계수 a 로 부터 계산한다.

$$\hat{s}_i = a s'_{i-1} \quad (2)$$

e_i 는 adaptive 양자기로 양자화하여 code word를 만든다. 양자기의 step-size는 식 (3)으로 부터 계산되어진다.

$$\Delta_i = M_i \cdot \Delta_{i-1}^\beta \quad (3)$$

여기서 M_i 는 code word의 함수로서 본 논문에서는 Jayant가 제시한 값을 사용하였다 [6]. M_i 의 역할은 e_i 의 값에 step-size를 적용시킨다. 즉 e_i 가 증가하면 M_i 로써 step size를 증가시키고, e_i 가 감소하면 step size를 감소시킨다. β 는 code word의 전송오차의 영향을 수신측에서 감쇄시켜 주는 역할을 한다 [7]. 보통 β 는 1보다 작은 값을 사용하는데 본 논문에서는 $\beta=63/64$ 를 사용하였다.

3. ADPCM CODEC의 구성

(μ PD7720D 소개)

본 논문에서 사용한 DSP는 NEC PD7720D(이하 7720)으로서 8 MHz로 동작시킬 경우 한 Instruc-

을 수행하는데 250ns 가 요구된다 [8].

7720에서의 모든 연산은 16비트의 고정숫자점으로 실행되며, 한번의 곱하기를 행하여 accumulator 에 저장하는데 필요한 시간은 500ns 에 불과하다. 또한 7720은 28-pin package 내에 512x16의 instruction ROM과 512x13의 data ROM, 128x16의 RAM을 가지고 있다. 7720의 입출력 기능은 Serial I/O, Parallel I/O와 DMA가 있다. 그러므로 7720을 사용함으로써 종래에는 거의 불가능한 실시간 동작 하아드웨어 시스템을 쉽게 실현시킬 수 있다.

(소프트웨어 개발)

ADPCM 코우딩 알고리즘은 먼저 PDP 11/44 에서 FORTRAN으로 시뮬레이션을 하였다. 여기서 모든 계산은 7720과 동일한 정확도로 행하여 청각실험에 의해서 round-off의 영향을 조사하였다. 그 결과 고정숫자점 연산에서 word-length가 16 비트면 충분하다는 결론을 얻었다. 이 FORTRAN 프로그램을 바탕으로 하여 CP/M 머신에서 동작하는 어셈블러와 시뮬레이터로 7720 프로그램을 개발하였다.

프로그램은 송신부와 수신부로 나누어진다.

먼저 송신부는 A/D 변환기로 부력 8비트 μ -law PCM을 받아들여서 참조표(Look-up table)를 사용하여 14비트의 선형 PCM으로 변환한다. 이 값으로 ADPCM 코우딩 알고리즘을 수행하여 4비트 code word를 출력한다.

수신부에서는 이 출력을 받아서 14비트의 선형 PCM으로 재생한 다음, 가지 구조 알고리즘(tree-structured algorithm)을 사용하여 μ -law PCM으로 변환한다. 한 샘플을 encode와 decode를 하는데 필요한 명령의 비율은 표1과 같다.

(하아드웨어)

그림 2.는 ADPCM 하드웨어 블록 선도이다. 아날로그 변환부는 현재 국내 전화 교환기에서 사용되고 있는 필터와 CODEC을 사용하여 A/D 와 D/A를 행하여 7720과의 병렬 입출력을 하기 위해서 시프트 레지스터를 사용하였다. 아날로그 신호의 표본화율은 8KHz 이고 8비트 데이터가 시프트되는 클럭은 2.048MHz를 사용하였으므로 프레임당 채널의 수는 32개까지 가능하다.

전체 시스템은 CODEC의 TSx(Time Slot Output-CODEC에서 8비트 데이터가 출력될 동안 LOW 상태가 된다) 신호를 사용하여 8KHz로 동기시켰다. 7720의 내부 ROM과 RAM의 사용은 표2와 같다.

송신부에서 데이터 ROM의 사용이 많은 것은 μ -law PCM을 선형 PCM으로 변환할 때 128개의 ROM을 look-up table로써 사용하기 때문이다.

I/O 콘트롤 부는 CODEC의 time slot을 결정하고 각 채널별로 입출력을 조정한다. 여기는 별개의 제어용 마이크로프로세서(Z-80)가 필요하다.

(실시간 동작조사)

완성된 하드웨어는 실시간 동작 여부를 조사하기 위해서 EVA Kit(7720 emulator)를 사용하였다. 여기서 한 샘플을 처리하는데 필요한 시간은 송신부와 수신부 공히 23 μ sec이하로서 프로그램 크기로 조사한 결과와 일치하였다. 따라서 125 μ sec(샘플링 주파수가 8KHz이므로 샘플 간의 간격은 125 sec이다) 동안 최대 5개의 채널을 동시에 수행할 수 있다.

그림 3은 1KHz의 정현파를 입력할 경우에 수신측에서 재생한 파형이다. 처음의 약 90샘플(11ms)까지는 step-size의 transition 상태를 나타내고 있다. 이때의 시간 지연은 상당히 짧기 때문에 음질에 큰 영향을 주지 못한다.

4. 결론

본 논문에서는 PD7720을 사용하여 실시간 동작 ADPCM Codec을 구성하였다. 8MHz clock으로 8KHz 표본화시 실시간 동작의 약 18% 정도밖에 사용하지 않으므로 1개의 μ PD7720로 5채널 ADPCM Codec을 구성할 수 있음을 알 수 있다. 본 논문에서는 가장 간단한 Jayant 방식 ADPCM을 사용하였으나, 아직도 충분한 ROM 및 RAM이 남아 있으므로 adaptive 예측기 또는 noise-shaping 필터를 사용한 좀더 복잡한 알고리즘을 사용하여 음질의 향상을 얻을 수 있음을 알 수 있다. 이와 같이 DSP를 사용하면 복잡한 코우딩 알고리즘도 쉽게 하아드웨어로 설치할 수 있다. 현대 사회는 음성통신, 즉 전화 통신의 급격한 팽창을 요구하고 있다. 그러므로 음성코우딩의 필요성은 필수적이다. 그러므로 DSP를 사용한 실시간 동작codec의 개발에 앞으로 많은 관심이 집중될 것으로 예상된다.

5. 참고문헌

1 P. Cumiskey, N.S. Jayant, and J.L. Flanagan, "Adaptive Quantization in Differential PCM Coding of Speech, B.S. T.J., 52, No.7

Sept. 1973, pp.1105-18

2 N.S. Jayant, "Digital Coding of Speech Waveforms : PCM, DPCM, and DM Quantizers," Proc. IEEE 62, May 1974, pp.611-32

3 L.H. Rosenthal et.al., "A Multiline Computer Voice Response System Utilizing ADPCM Coded Speech," IEEE Trans. ASSP, ASSP-22, No.5, Oct.1974, pp.339-52

4 H.W.-Adelmann, Y.C. Ching, and B. Gotz, "An ADPCM Approach to Reduce the Bit Rate of μ -law Encoded Speech," B.S. T.J. 58, No.7, Sept. 1979, pp.1659-71

5 J.P. Agrawal, J.B. O'neal, JR., J.S. Cooper, "The Design of an ADPCM/TASI System for PCM Speech Compression," IEEE Trans. on commun., vol.COM-29, No.9, Sept.1981, pp.1393-1398

6 N.S. Jayant, "Adaptive Quantization with a One-Word Memory," B.S.T.J., Spring 1976, pp.159-178

7 D.J. Goodman and R.M. Wilkinson, "A Robust Adaptive Quantizer," IEEE Trans. on commun., vol.COM-23, Nov.1975, pp.1362-1365

8 NEC " PD7720 user's manual "

9 J.R. Bodie, et.al., "Adaptive Differential Pulse-Code-Modulation Coding," B.S.T.J., vol.60, No.7, Sept. 1981, pp.1547-1561

표1. 전체 명령수에 대한 부분별 비율

	ENCODER	DECODER
Initialization	11%	15%
-law to linear	11%	-
linear to -law	-	30%
encode	49%	-
decode	-	38%
step size control	26%	27%
I/O	3%	3%

표2. 내부 RAM과 ROM의 사용비

	ENCODER	DECODER
Instruction ROM	50%	50%
Data ROM	77%	51%
Internal RAM	5%	9%

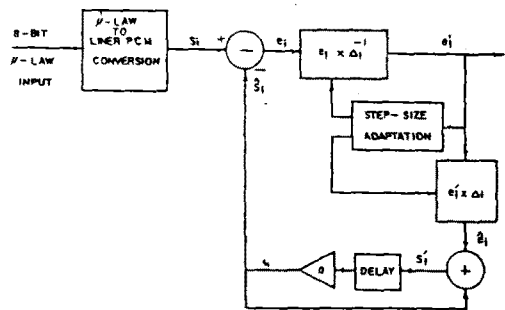


Fig.1 Block diagram of the ADPCM encoder

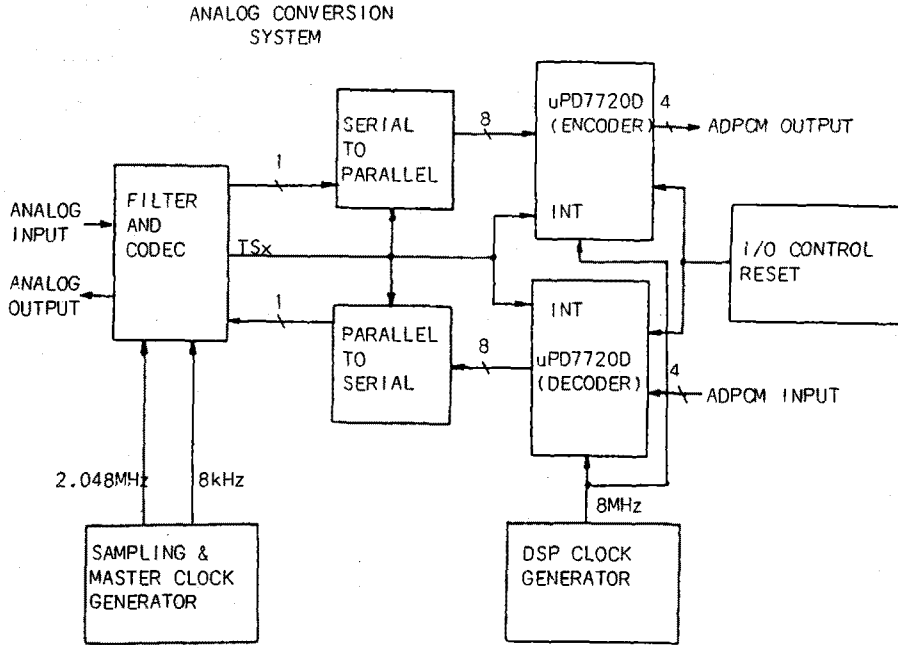


FIG 2. HARDWARE CONFIGURATION OF ADPCM CODEC

