

DSP를 사용한 4 채널용 ADPCM CODEC의 실시간 구현에 관한 연구

(Implementation of a 4-Channel ADPCM CODEC Using a DSP)

李宜宅*, 李康錫**, 李商郁**

(Euh Taek Lee, Kang Suk Lee and Sang Uk Lee)

要約

본 논문에서는 파형 코우딩 방식에 속하는 ADPCM을 NEC 7720 DSP를 사용하여 간단하고 성능이 우수하며 유연성(flexibility)이 뛰어난 ADPCM codec을 실시간 구현을 하였다. ADPCM 알고리즘은 적응 양자기와 1차의 고정예측기를 사용하였으며 프로그램을 최적화하여 하나의 NEC 7720으로 4채널을 동시에 부호화 또는 복호화 할 수 있도록 하였다.

실제 전화음성과 RC 정형 Gaussian 잡음 및 1004Hz 정현파를 사용한 컴퓨터 시뮬레이션으로부터 NEC 7720의 연산 정확도를 조사하였으며, SNR 및 청각조사로 부터 알고리즘 수행에 필요한 값들을 결정하였다.

NEC 7720의 소프트웨어는 real-time hardware emulator로 실시간 동작을 확인하였는데 부호기에서 1샘플을 부호화하는데 최대 23.25 μ s가 걸리고 4채널을 모두 부호화하는데 113.5 μ s가 필요하였으며, 복호기에서는 각각 24.75 μ s와 119.5 μ s가 소요되었다.

Abstract

In this paper we have designed and implemented in real time a simple, efficient and flexible ADPCM codec using a high speed digital processor, NEC 7720. For ADPCM system, we have used an instantaneous adaptive quantizer and a first-order fixed predictor. The software for NEC 7720 has been developed and it was found that the NEC 7720 was capable of performing the entire ADPCM algorithm for 4 channels in real time as optimizing the program. Computer simulation has been made to investigate a computational accuracy of NEC 7720 and to determine necessary parameters for a ADPCM codec. Real telephone speech, RC-shaped Gaussian noise and 1004 Hz tone signal were used for simulation. In simulation, the parameters were optimized from the computed SNR and the informal listening test. The developed software was tested in real time operation using a hardware emulator for NEC 7720. It took a maximum 23.25 μ s to encode one sample and 113.5 μ s, including all the necessary I/O operations, to encode 4 channels. In the case of decoding process, it took 24.75 μ s to decode one sample and 119.5 μ s to decode 4 channels.

*正會員, 韓國電氣通信研究所 音響研究室

(Dept. of Acoustic and Speech, ETRI)

**正會員, 서울大學校 制御計測工學科

(Dept. of Control & Instru., Seoul National Univ.)

接受日字: 1985年 1月 14日

I. 序 論

1960년 초 PCM(pulse code modulation)에 의한 디지털 음성통신이 제안된 이래 디지털 음성통신기술은 지금까지 많은 발전을 하여 왔다. 아날로그 음성신호를 디지털 신호로 변환하여 전송하면 많은 장점이 있

다는 것은 우리에게 널리 알려져 있다. 그러나 디지털 음성통신은 많은 장점을 갖고 있는 대신 큰 단점으로는 전송 대역폭이 커진다는 것이다. 즉 기존 PCM은 200-3400Hz 대역의 음성신호를 8000Hz로 샘플링하여 8bit로 양자화하므로 64kbps의 전송율(bit rate)이 요구된다. 따라서 전송율을 줄이기 위한 여러가지의 음성 코우딩 방식이 지금까지 연구, 제안되고 있다.¹¹⁻¹³ 음성코우딩은 대표적으로 음성신호를 샘플링하여 양자화하는 파형 코우딩(waveform coding)과 음성 신호의 주기 및 음성 발생 장치인 성도(vocal-tract)의 계수등을 음성신호로부터 추출하여 전송하는 소오스 코우딩(source coding)으로 나눌 수 있다. 이중 파형 코우딩 방식은 음성신호의 여러가지 특수한 성질¹¹을 이용하여 코우딩한다. 음성신호의 기본적인 성질은 대역폭이 제한되어 있다는 것이다. 이것은 음성신호를 디지털 신호로 표시하기 위해서 Nyquist 샘플링 이론을 적용할 수 있는 근간이 된다. 또 음성신호의 진폭에 대한 장시간 평균 확률밀도함수(long-time-averaged PDF)는 Gaussian, Laplacian 또는 Gamma 함수로 근사화할 수 있으며 단시간(20ms 정도)은 Gaussian 모델이 적당하다.¹¹ 또한 음성신호의 에너지는 저주파 부근에 모여 있으며 샘플 사이에 높은 상관관계(correlation)가 존재한다. 이것은 음성신호가 인접한 샘플사이에서 완만하게 변한다는 것을 뜻한다. 정보량이 있어서 파형 코우딩 방식은 9.6kbps 이상¹²이고, 소오스코우딩은 0.05-4.8kbps¹² 정도로서 소오스 코우딩 방식이 정보량이 있어서는 유리하나 합성음이므로 음질이 좋지않고 알고리즘도 복잡하다. 반면에 파형코우딩은 정보량은 많으나 음질이 우수하고 알고리즘이 비교적 간단하여 하드웨어를 구성하는데 이점이 있다.

파형 코우딩 방식에는 여러가지의 코우딩 알고리즘이 제안되고 있으나 그중 대표적인 것이 ADPCM(adaptive differential pluse code modulation)이다. ADPCM은 하드웨어 구성이 간단하고 강인성(robustness)이 좋아¹⁴ 음성 및 영상코우딩에 널리 응용되고 있다. ADPCM은 전송율이 24-48kbps인데 32kbps ADPCM 음질은 64kbps PCM 원음과 거의 구별할 수 없다.¹⁴ 따라서 32kbps ADPCM을 사용할 경우 채널 이용율을 2배로 증가시킬 수 있으며 음성저장용으로 사용할 경우 필요한 메모리를 $\frac{1}{2}$ 로 줄일 수 있다. 즉 전화통신에서 음성의 저장은 수화자(receiver)가 부재중일때 통화내용을 메모리에 저장했다가 수화자가 요구할때 그 내용을 알려줄 수 있는 음성 우편 서비스 등에서 사용할 수 있다. 그러므로 음성코우딩 기술의

광범위한 응용을 위해서는 경제적이고, 소형이며 또한 유연성있는 실시간동작(real-time) codec(coder/decoder)의 개발은 필연적이다. ADPCM codec의 하드웨어 구현은 처음에는 아날로그 방식으로 제작되었으나¹⁵, 안정성(stability)에 문제가 있었으며 디지털 방식은 Bates¹⁶가 처음으로 사용하였다. 그는 스택사이즈를 적용시키기 위해서 ROM-based look-up 테이블을 사용하고 하드웨어 승산기의 사용을 피하기 위해서 디지털 연산기(adder, subtracter)를 사용하였다. Bates는 12비트 선형 PCM을 입력으로 사용하였고 사용한 부품은 약 80개의 TTL논리 IC이다.

Adelman¹⁷ 등은 Bates와 비슷한 방법으로 하드웨어를 구성하였는데 입력신호는 64kbps μ -law PCM이다. Dubnowski¹⁸, Agrawal¹⁹ 등은 위의 두 방법과는 달리 ADM 2900 series의 비트 슬라이스(bit-slice) 마이크로 프로세서를 사용하는 새로운 방법을 소개하였다. 그러나 최근 VLSI 기술의 급격한 발전에 힘입어 초고속 신호처리 프로세서(digital signal processor)가 소개되므로써 실시간 동작 codec 개발이 한결 용이해졌다.^{10, 11} DSP는 내부에 ROM과 RAM 및 하드웨어 승산기를 가지고 있어서, 시스템을 구성할 때 부수적인 회로가 거의 필요없어 하드웨어가 간단해지고 내부 프로그램의 수정이 가능하므로 유연성있는 시스템을 구성할 수 있다. 특히 DSP 중 NEC 7720은 31bit(16×16) 곱셈을 8MHz clock 시 250nsec안에 수행할 수 있으므로, 한개의 NEC 7720으로 여러 채널의 ADPCM codec을 설치할 수 있다. 실제 NEC 7720 보다 속도가 느린 Bell Labs.의 DSP로도 가능성을 보인 예가 있다.¹⁰

본 논문에서는 NEC 7720 DSP를 사용하여 전송율 32kbps μ -law PCM/ADPCM 부호변환 codec의 하드웨어 및 소프트웨어의 구현 및 그 성능을 조사하고자 한다. ADPCM 알고리즘은 적응양자기 및 1차 고정예측기를 사용하는 Jayant의 ADPCM 알고리즘¹⁴을 근거로 하며 μ -law PCM 선형 PCM 상호 변환은 CCITT 권고안 G. 711을 채택하였다.¹² ADPCM codec은 PCM과 ADPCM 전송 시스템의 정합 장치로 사용될 수 있으며, 또한 전전자식 디지털 교환기와 ADPCM 전송 시스템간의 정합장치, 디지털 교환기의 자동 응답장치 등 광범위하게 이용될 수 있으나 본 논문에서는 현재 KETRI에서 개발중인 전자우편 서어비스의 codec용으로 개발하였다.

본 논문의 구성은 다음과 같다. 제 2 장에서는 ADPCM에 관한 기본 이론을 소개하고, 제 3 장에서는 NEC 7720으로 수행할 수 있는 ADPCM 알고리즘의 컴퓨터

시뮬레이션 및 그 결과에 대하여 논하며 제 4 장에서는 하아드웨어의 구성을, 제 5 장에서는 소프트웨어 개발에 대해서 설명하였다. 제 6 장에서는 본 논문의 결론을 제시하였다.

2. ADPCD 코우딩 알고리즘

200-3400Hz 대역의 음성 신호를 8000Hz로 샘플링 하면 샘플된 신호 $x(n)$ 은 불규칙 신호(random process)적인 측면에서 볼 때 인접한 샘플들간의 상관계수(correlation)가 매우 높다. 그러므로 $x(n)$ 은 바로 전 샘플값 $x(n-1)$ 로 예측할 수 있다. 또한 $x(n)$ 과 $x(n-1)$ 의 차신호 $d(n)$ 의 variance는 원신호 $x(n)$ 의 variance보다 작으므로 DPCM에서는 $x(n)$ 를 양자화하는 대신 예측차 신호 $d(n)$ 을 양자화해서 보낸다. 위와 같은 ADPCM에 대하여 지난 10여년간 많은 알고리즘이 제안되어 있으나^{13, 14, 15, 16} 이중 가장 보편화 되어 있는 방식은 Jayant¹⁴가 제안한 것이다. Jayant의 알고리즘은 예측기는 고정 1차 예측기를 사용하고 양자기는 $d(n)$ 의 variance에 따라 스텝사이즈를 변화시켜주는 적응 양자기를 사용하고 있다. 그림 1에 그 블록선도를 도시하였다. Jayant가 제안한 스텝사이즈 적응방식은 feedback 적응 방식중의 하나인데 시간 $(n+1)$ 일때의 스텝사이즈 $\Delta(n+1)$ 은

$$\Delta(n+1) = M(n) \cdot \Delta^n(n) \quad (1)$$

으로 주어진다. 여기서 $M(n)$ 은 적응상수로서 시간 n 일때의 출력 $\hat{d}(n)$ 에 의해서 결정된다. $\hat{d}(n)$ 이 큰값이면 입력 신호가 증가하는 것으로 간주하여 $M(n) >$

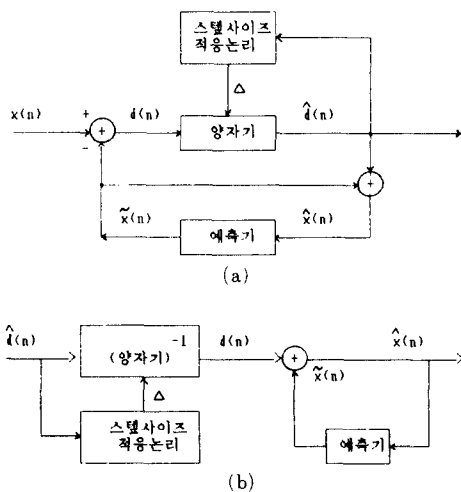


그림 1. ADPCM 블록선도

(a) 부호기 (b) 복호기

Fig. 1. Block diagram of the ADPCM.

a)encoder. b)decoder.

1로 하여 스텝사이즈를 증가시키고, 그 반대이면 $M(n) < 1$ 로 하여 스텝사이즈를 감소시킨다. 또한 β 는 leakage factor로써 0과 1사이의 값을 갖는데 본 연구에서는 β 를 1로 하였다. 이상과 같이 설명한 Jayant 알고리즘을 실제 DSP로 구현할 때는 다소 수정이 필요하다. 그림 2에 실제 구현한 ADPCM 부호기의 블록선도를 보였다. 지금부터 각 주요 블록별로 설명한다.

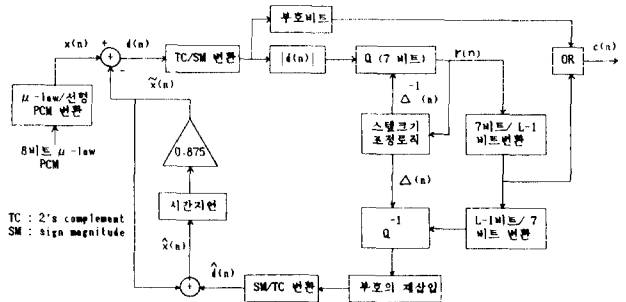


그림 2. ADPCM 구현 블록선도

Fig. 2. Block diagram of the ADPCM implementation.

1) μ -law / 선형 PCM 상호 변환

μ -law PCM을 선형 또는 선형 PCM을 μ -law PCM으로 변환하는 알고리즘은 CCITT 권고안 G.711에 따랐다.¹¹²

(1) μ -law / 선형 PCM 변환

A) μ -law / 선형 PCM 변환

$$|x(n)| = 2 \cdot 2^c (S + 16.5) - 16.5 \quad (2)$$

단, c = 세그먼트 번호 - 1; 0-7

S = 스텝번호 - 1; 0-15

B) 선형 / μ -law PCM 변환

$$|x(n)| \geq 2 \cdot 2^c (S + 17) - 16.5 \quad (3)$$

c 는 식(3)에 $S=15$ 를 대입했을때 다음식을 만족하는 가장 작은 양의 정수이다.

$$x(n) + 33 \leq 2^c \cdot 64, c \geq 0 \quad (4)$$

c 가 구해지면 S 는 다음식으로 부터 구한다.

$$S = \frac{|x(n)| + 33}{2 \cdot 2^c} - 16 \quad (5)$$

2) 스텝사이즈 적응 용

스텝사이즈는 식(1)을 사용하여 적용시키는데 지수의 계산이 NEC 7720에서는 수행하기가 어렵고, 차신호를 양자화할 때 필요한 나눗셈도 시간이 많이 소요된다. 이런 어려움을 해결하는 방법으로는 적응 양자기를 스텝사이즈 간격이 각각 Q 개의 균일 양자기의 집합으로 간주하여 그림 2의 $\Delta(n)$ 과 $\Delta^{-1}(n)$ 의 look-

up 테이블을 작성한 다음 테이블의 주소를 찾아가는 방법으로 스텝사이즈를 적응시킬 수 있다. 스텝사이즈 적응을 위한 look-up 테이블의 작성은 다음과 같이 한다.

식(1)에서의 $\Delta(n)$ 은 컴퓨터가 나타낼 수 있는 유효 숫자의 범위내에서 Δ_{max} 와 Δ_{min} 사이의 모든 값을 가질 수 있지만, 스텝사이즈의 갯수를 Q으로 제한할 경우에는 스텝사이즈 값이 로그분포를 가지게 해야 한다. 음성신호의 dynamic range가 40-50dB이므로 본 논문에서는 양자기의 dynamic range를 $20\log_{10}(\Delta_{max}/\Delta_{min}) = 20\log_{10}(256) = 48\text{dB}$ 로 하였는데 스텝사이즈 갯수가 Q일 경우 스텝사이즈 간격은 $48/Q(\text{dB})$ 이다. 여기에서 Δ_{min} 값은 1로 하였다. 우선 스텝사이즈 간격을 (6)식과 같이 표시하였다.

$$48/Q(\text{dB}) = 20\log 2^b \quad (6)$$

여기에서 상수 b는 식(6)을 가장 근사하게 만족시키는 정수로써 스텝사이즈 간격을 결정하며 Q값에 따라서 변하게 된다. b를 구하게 되면 식(1)의 $\Delta(n)$ 은 다음과 같이 쓸 수 있다.

$$\Delta(n) = \Delta_{min} 2^{\frac{k(n)}{b}}, \quad 0 \leq k(n) \leq Q-1 \quad (7)$$

식(7)에서 보면 최소 스텝사이즈 값은 1이고 최대 스텝사이즈 값은 256임을 알 수 있다. 또한 스텝사이즈 적응상수 M(n)도 같은 방법으로 표시할 수 있다. 본 논문에서는 Jayant가 제안한 5bit ADPCM 적응상수를

표 1. 5bit 스텝사이즈 적응상수

Table 1. Step size adaptation constants for a 5 bit.

Jayant 적응상수	r(n)	M(n)
0.9	-2	0.917
0.9	-2	0.917
0.9	-2	0.917
0.9	-2	0.917
0.95	-1	0.9576
0.95	-1	0.9576
0.95	-1	0.9576
0.95	-1	0.9576
1.2	4	1.189
1.5	10	1.542
1.8	14	1.834
2.1	17	2.089
2.4	20	2.378
2.7	23	2.709
3.0	26	3.084
3.3	28	3.364

사용하였다. 먼저 적응상수 M(n)을 식(8)과 같이 표시한다.

$$M(n) = 2^{\frac{r(n)}{b}} \quad (8)$$

여기서 상수 r(n)은 b가 정해졌을 때 Jayant 적응상수 값들과 가장 근사하게 만족시키는 정수이다. 표 1에 16개의 r(n), M(n) 및 Jayant가 제안한 5bit ADPCM 적응상수 값들을 보였다. 그러면 식(7) 및 (8)을 식(1)에 대입하면 스텝사이즈 계산은 다음과 같다.

$$\begin{aligned} \Delta(n+1) &= 2^{\frac{r(n)}{b}} \cdot [\Delta_{min} 2^{\frac{k(n)}{b}}]^\beta \\ &= 2^{|\frac{r(n)}{b} + \beta \cdot \frac{k(n)}{b}|} \cdot \Delta_{min}^\beta \\ &= 2^{k(n+1)/b} \cdot \Delta_{min} = 1 \end{aligned} \quad (9)$$

식(9)에서

$$k(n+1) = r(n) + \beta k(n) \quad (10)$$

이렇게 되면 식(9)에서 보인 것처럼 어떤 순간 스텝사이즈 값을 결정하는 k(n)을 얻게 되면 다음 순간의 스텝사이즈 값은 식(10)을 이용하여 쉽게 구할 수 있음을 알 수 있다. 실제 k(n)은 $\Delta(n)$ 의 look-up 테이블의 주소이다. 식(7)을 사용하여 $\Delta(n)$ 과 $\Delta^{-1}(n)$ 의 look-up 테이블을 작성할 때 $\Delta^{-1}(n)$ 은 1보다 작으므로 NEC 7720에서 표시하는데 문제가 되지 않으나 $\Delta(n)$ 은 정수와 소수가 함께 있으므로 $\Delta(n)$ 을 1보다 작게 만들어야 한다. 따라서 $\Delta(n)$ 을 최대값인 256으로 나눠서 look-up 테이블을 작성하고 나중에 이를 보상해 주는 방법을 채택하였다. 계산된 $\Delta(n)$ 와 $\Delta^{-1}(n)$ 은 NEC 7720 내부의 데이터 ROM에 저장하였다.

3. 컴퓨터 시뮬레이션 및 검토

컴퓨터 시뮬레이션은 Jayant의 ADPCM 알고리즘⁽¹⁴⁾을 기본으로 하여, 본 논문에서 사용하고자 하는 NEC 7720 DSP⁽¹⁵⁾의 특성에 맞도록 수정한 알고리즘에 대하여 실시하였다. 시뮬레이션에서는 NEC 7720이 16비트의 고정소숫점 연산을 함으로써 발생하는 절삭오차(truncation error)에 대한 영향을 조사하기 위하여 고정소숫점 연산과 유동소숫점연산(floating-point arithmetic)을 병행하였으며, 또한 ADPCM 알고리즘의 수행에 필요한 상수, 예측계수, 스텝사이즈 등도 시뮬레이션에 의해서 NEC 7720에서 사용하는데 가장 알맞는 값으로 선택하였다. 선택된 값은 Q가 128, b는 16이며 예측 계수는 0.875이다. 시뮬레이션에서 사용한 데이터는 대역폭이 200-3400Hz이고 8KHz 샘플링을 하는 PCM codec(TP 3020J)으로 부터 얻은 남자 3명과 여자 1명의 약 20초간의 음성 데이터와 RC 정형 Gaussian 잡음(RC shaped Gaussian noise) 및 1004Hz의 정현파이다. 모든 컴퓨터 시뮬레이션은 PDP

11/44로 실시하였다. 음성데이터는 주로 청각 조사를 하여 알고리즘의 성능을 조사하였는데, 전송율이 32 kbps인 경우 남자음성은 원음과 거의 구별할 수 없었으나 여자음성은 무성음에서 약간의 잡음을 감지할 수 있었다. 이것은 여자의 음성이 남자의 음성보다 주파수가 높고 무성음이므로 이 부분의 상관계수가 낮기 때문이다. 또한 유동 소숫점과 고정소숫점 연산을 사용한 음성들을 청각조사에 의하여 그 차이를 비교하였는데 거의 구별을 할 수 없었다. 이때 고정소숫점연산 시뮬레이션은 모든 연산이 NEC 7720 내부에서 이루어지는 것과 같이 하였다. 그림 3은 음성신호의 에너지와 SNR을 나타낸 것인데, 이때 SNR은 128 샘플

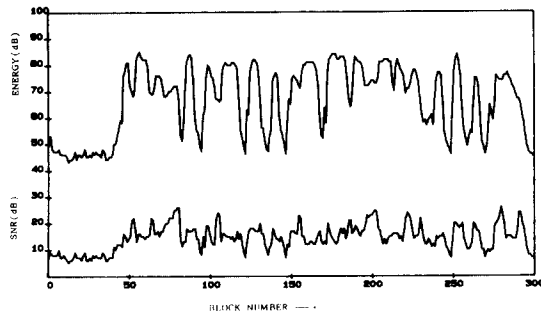
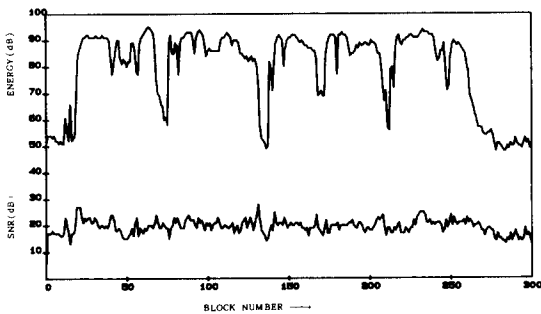


그림 3. 에너지 Vs. SNR (32kbps ADPCM)

(a) 남자음성 (b) 여자음성

Fig. 3. Energy VS SNR (32Kbps ADPCM).

(a) male speech. (b) female speech.

을 1 블록으로 계산한 것이며 고정소숫점 연산을 사용하였다. 이 그림들에서 에너지가 큰 유성음의 부분은 SNR이 높고 에너지가 적은 무성음 부분은 SNR이 낮음을 알 수 있다. 또한 스텝사이즈의 갯수를 128개로 제한하는 영향을 조사하기 위하여 RC 정형 Gaussian 잡음을 입력으로 할때 식(6)의 Q 값의 변화에 따른 SNR을 그림 4에 도시하였다. Q가 64, 128, 256일 경

표 2. NEC 7720의 주요 사항

Table 2. Main specifications of the NEC 7720.

- Fast Instruction execution-250 ns/8MHZ clock
- 16bit data word
- Multi-operation instruction for optimizing program execution
- Large memory capacities
- Program ROM 512×23bits
- Data coefficient ROM 512×13bits
- Data RAM 128×16bits
- Fast 16×16 31-bit parallel multiplier
- Dual accumulators
- Four-level sub- μ tine stack for programming efficiency
- Multiple I/O capabilities
- Serial-separate input and output (8 or 16bit)
- Parallel
- DMA
- Compatible with most μ -processors
- +5V power supply
- NMOS Technology
- 28-Pin dip package

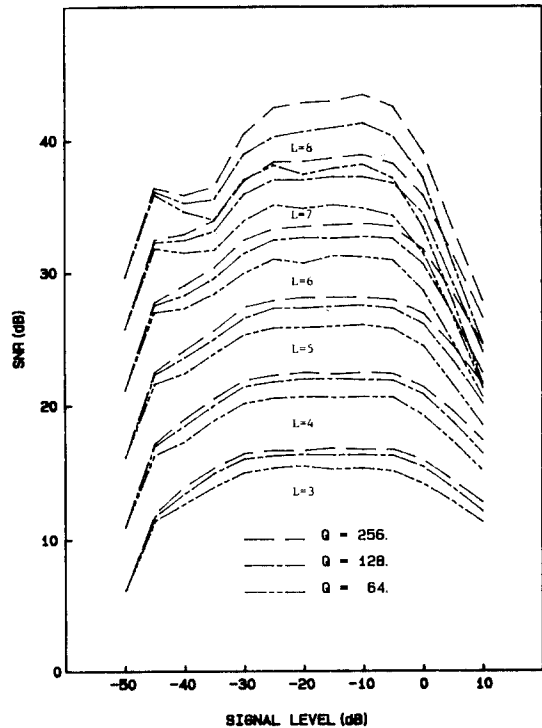


그림 4. 스텝사이즈 갯수 vs SNR

Fig. 4. Number of step size VS SNR.

우를 비교하였을때 Q가 256일 때의 SNR이 가장 높았다. 이것은 Q가 클수록 식(1)에 의한 $\Delta(n)$ 과 가까운 값을 가지기 때문이다. 또한 그림에는 표시하지 않았지만 Q가 256일 때의 SNR은 유동소숫점 연산을 사용하였을 때의 SNR과는 거의 동일하였다. 또한 그림 4에서는 bit 수가 3에서 6으로 증가할때까지는 SNR이 bit당 약 6dB씩 증가하다가 그 이상에서는 증가가 둔화되는 것을 보여주고 있는데, 그 이유는 본 논문에서 bit를 3-8로 가변시키기 위해서 5bit 적응 알고리즘을 사용하였으므로 비트수가 클때는 slope overload noise가 발생하기 때문이다. Q가 작을때 SNR의 감소율이 큰데 이것은 Q가 작을수록 양자오차가 크고 slope overload noise가 첨가되기 때문이다. 그림 5는 원래의 음성신호와 전송율 32kbps일때의 고정소숫점 연산(Q=128, b=16) 시뮬레이션에서 재생한 음성신호의 파형을 보인 것인데 이 두 파형의 차이가 거의 없음을 알 수 있다. 그러므로 고정소숫점 연산을 하더라도 우리가 원하는 정확도를 얻을 수 있음을 알 수 있다.

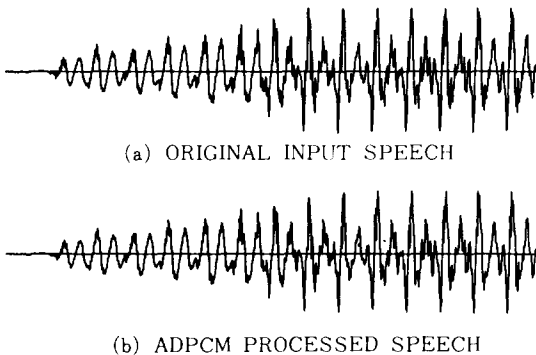


그림 5. 원래의 음성신호와 재생된 음성파형(전송율 32kbps)

Fig. 5. Waveform of the original and decoded speech. (bit rate:32Kbps)

4. 하드웨어의 설계

ADPCM codec을 하드웨어로 구현할때 고려한 사항은 소형이고 시스템에 유연성이 있으며 비싼 하드웨어 승산기의 사용을 피하는 것이었다. 시스템에 유연성을 부여하기 위해서는 많은 작업을 소프트웨어로 처리하여, 단순히 프로그램만의 수정으로 시스템의 성능을 개선할 수 있어야 한다. 이러한 사용으로 사용될 수 있는 디지털 신호처리 프로세서(DSP)가 최근에 많이 개발되었다. 본 논문에서 사용하고자 하는 NEC 7720도 DSP의 일종인데 내부에 31bit(16×16) 연산을 250ns 만에 수행하는 하드웨어 승산기 및

ROM과 RAM을 가지고 있어서 프로그램이 가능하다.

표 3. 선형 PCM과 chord C와의 관계

Table 3. Relationship between a linear PCM and chord.

Linear PCM	Chord C
0 - 30	0
31 - 94	1
95 - 222	2
223 - 478	3
479 - 990	4
991 - 2014	5
2015 - 4062	6
4063 - 8031	7

NEC 7720에 관한 간단한 사항을 표 3에 보였다. NEC 7720은 고속의 연산속도를 가지고 있어서 1채널을 부호화(encoding) 또는 복호화(decoding) 하는데 25μs도 걸리지 않았는데 음성신호를 8KHz로 샘플링할 경우 샘플간의 간격은 125μs이므로 하나의 NEC 7720으로 5채널을 동시에 처리할 수 있다는 것을 알 수 있었다. 그러나 여러 채널을 동시에 처리할때 각 채널에 대한 상태인 스템사이즈, 예측기 출력 등을 저장해 놓아야 하므로 본 논문에서는 4채널을 동시에 처리하도록 시스템을 구성하였다. 또한 부호기와 복호기를 각각 구현하였으므로 full-duplex로 동작하며 각 채널의 동작은 서로 독립적이다. 그림 6에 ADPCM codec의 블록선도가 주어져 있는데 크게 다음의 5부분으로 되어 있다.

- 1) 아날로그 필터와 PCM codec부
- 2) shift logic부
- 3) NEC 7720 및 컨트롤부
- 4) 채널 선택부
- 5) clock 발생부

각 부분에 대한 간단한 기능별 설명은 다음과 같다.

아날로그 필터와 PCM codec부는 음성신호로부터 μ -law PCM을 발생시키고 또 재생하는 부분으로 TP 3040J 아날로그 필터와 TP 3020J PCM codec으로 구성되어 있다. TP 3020J는 200-3400Hz의 대역필터(BPF)와 저역필터(LPF) 및 증폭기를 내장하고 있다. 음성신호의 샘플링 주파수는 8KHz이고 PCM 데이터는 2.048MHz의 shift clock으로 serial로 입력 또는 출력된다. NEC 7720의 입출력은 DMA 기능을 사용하여 parallel I/O를 하고 있는데 이것은 다중 채널을 수용할 때 입출력 조절이 용이하기 때문이다.

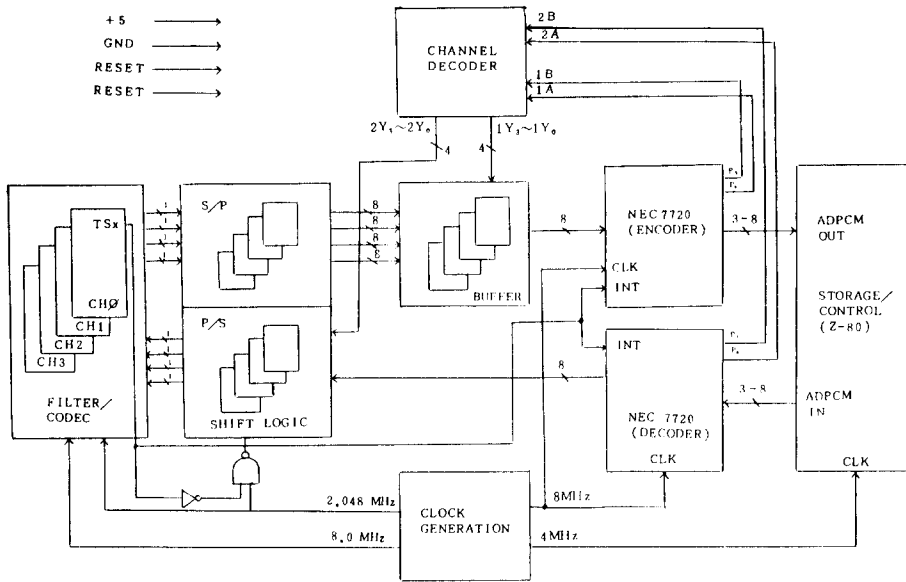


그림 6. Codec 블록선도
Fig. 6. Codec block diagram.

PCM codec의 데이터는 serial로 입출력되므로 NEC 7720과 입출력을 하기 위해서 serial-to-parallel 및 parallel-to-serial shift logic이 필요하다. shift register의 shift clock은 TP 3020J의 \overline{TS}_x 신호를 사용하여 PCM 데이터가 입력 또는 출력될때만 인가되도록 하였다. NEC 7720의 시스템 clock은 8MHz를 사용하는데 하나의 명령을 실행하는데 2개의 clock 즉 250ns가 소요된다. NEC 7720이 4 채널의 데이터를 모두 부호화 또는 복호화하는 시간과 각 프레임이 시작되는 시간이 일치하지 않으므로 동기가 필요하다. 이 동기는 NEC 7720의 interrupt 기능과 PCM codec의 \overline{TS}_x 신호를 이용 하였다. 또한 1개의 NEC 7720으로 4 채널을 동시에 처리하므로 입력과 출력의 multiplexing이 필요한데, 이것은 NEC 7720의 출력 포트 P_0, P_1 과 $\overline{RD}/\overline{WR}$ 신호를 74LS139 decoder에 입력하여 decoding 하므로써 각 채널의 enable 신호를 얻었다. 표 4에 채널 선택에 관한 진리표가 주어졌다.

이상에서 설명한 논리는 부호기와 복호기에서 동일하다. 하아드웨어 실시간 동작은 다음과 같이 testing 하였다. 다음장에서 설명하는 소프트웨어를 real-time emulator인 EVAKIT-7720^[16]으로 동작시키면서 real-time 동작 여부를 조사하였다. 이때 사용한 입력 데이터는 PCM codec으로 얻은 약 250개의 음성데이터이다. 이 데이터를 사용하여 고정소숫점 시뮬레이션 결과와 EVAKIT-7720으로 동작시킨 결과를 비교하여 완전히 일치함을 확인하였다. 또한 부호기의 ADPCM

표 4. 채널선택

Table 4. Channel selection.

\overline{WR}	\overline{RD}	P_1	P_0	채널 및 I/O 선택
0	1	0	0	CH 0, input enable
0	1	0	1	CH 1, input enable
0	1	1	0	CH 2, input enable
0	1	1	1	CH 3, input enable
1	0	0	0	CH 0, output enable
1	0	0	1	CH 1, output enable
1	0	1	0	CH 2, output enable
1	0	1	1	CH 3, output enable

출력을 Intel 8255 PPI를 통해 Z-80 마이크로 컴퓨터 메모리에 저장시켰다가 복호기로 입력하여 음성신호를 재생하여 완전히 그 동작을 확인할 수 있었다. 이것은 음성 우편 서어비스 시스템과 같은 음성 저장에 필요한 곳에서 이용할 수 있다. 각 채널의 부호기 및 복호기의 동작 여부는 내대의 handset을 각 부호기, 부호기에 연결하여 동시에 통화하여 그 동작 여부를 확인할 수 있었다. 이때 부호기의 출력을 복호기의 입력에 직접 연결하였다. 또한 codec을 통해 얻은 음질은 기술한 시뮬레이션을 통해 얻은 음질과 동일하였다.

5. 소프트웨어

NEC 7720 프로그램은 cross-assembler와 non-real-

time simulator를 사용하여 개발하였다. 125 μ s의 제한된 시간안에 4 채널의 처리를 마쳐야 하므로 in-line operation을 하여 서브루틴 CALL과 RETURN에 의한 시간을 절약하도록 하였다. 완성된 프로그램은 real-time emulator인 EVAKIT-7720¹¹⁾으로 4장에서 구성한 하드웨어를 조사하였고, NEC 7720에 프로그램을 load하여 실시간 동작이 이루어짐을 직접 확인하였다. 또한 본 연구에서는 전송율을 32kbps로 하였으나 약간의 소프트웨어 수정만으로 전송율을 24-64kbps로 변환 가능하게 하였다. 그림 7과 그림 8에 부호기 알고리즘과 복호기 알고리즘의 흐름도가 각각 주어져 있다.

알고리즘의 흐름도에서 중요한 몇부분에 대해서 간단히 설명하고자 한다.

먼저 interrupt를 기다리는 부분은 시스템을 8KHz로 동기시키기 위한 부분으로서 NEC 7720의 INT핀에 interrupt가 인가될 경우에 프로그램은 100번지에서 restart를 한다. μ -law/선형 PCM 변환부는 식(2)에

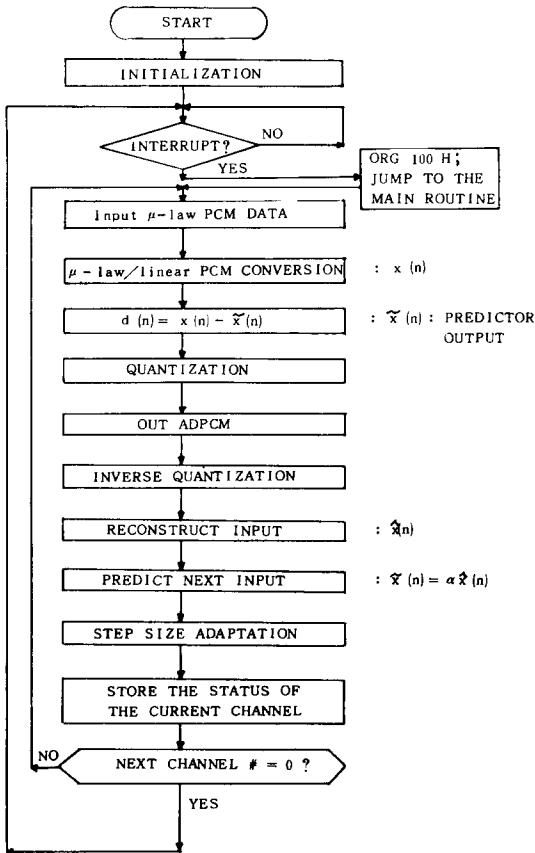


그림 7. 부호기 알고리즘의 흐름도
Fig. 7. Flow chart for the encoder.

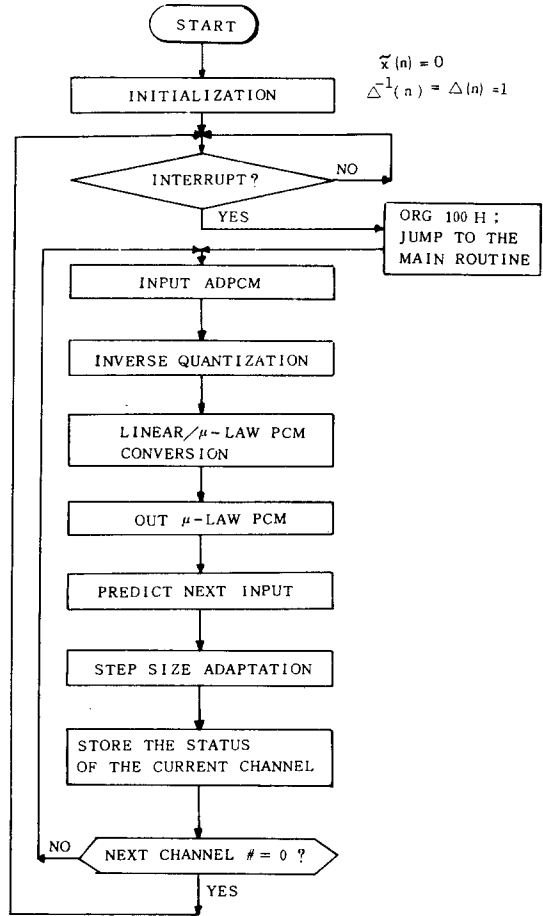


그림 8. 복호기 알고리즘의 흐름도
Fig. 8. Flow chart for the decoder.

의한 직접 계산을 하지 않고 data ROM에 저장한 테이블을 이용한다. 선형/ μ -law PCM의 변환도 식(3)을 사용하여 μ -law PCM의 chord C를 구하기가 어려우므로 표 3에 주어진 선형 PCM과 chord C와의 관계를 이용하여 chord C를 구하고 식(4)를 사용하여 S를 구하는 방법을 사용하였다. 표 5는 부호기와 복호기 알고리즘을 수행하는데 사용한 instruction ROM의 부분별 사용갯수를 보여주고 있다. Instruction ROM은 모두 512개의 word로 되어 있는데 부호기에서 181 word, 복호기에서 196word를 사용하고 있다. 복호기가 부호기보다 많은 instruction ROM을 사용하고 있는 것은 복호기의 선형 μ -law PCM 변환에서 49word를 사용하기 때문이다. 부호기와 복호기에서 소요되는 시간은 다음과 같다. 부호기에서 하나의 샘플을 부호화하는데 최대 23.25 μ s가 걸리고 4 채널을 모두 처리하는데는 최대 113.5 μ s가 필요하다. 여기에는 채널을

표 5. Instruction ROM 사용표
Table 5. Description of the instruction ROM.

	부 호 기	복 호 기
초기치 설정	39	41
μ -law / 선형 PCM 변환	19	-
선형 / μ -law PCM 변환	-	048
양자화	19	-
역양자화	20	27
예측기	7	7
스텝사이즈적응	30	27
입출력	5	5
채널선택	42	42
합계	181 (35.4%)	197 (38.5%)

선택하는데 소요된 20.5 μ s도 포함되어 있다. 복호기에서는 하나의 샘플을 복호화하는데 최대 24.75 μ s, 전체적으로는 최대 119.5 μ s가 걸렸다. 표6은 데이터 ROM의 사용을 보여주고 있는데 부호기에서 복호기보다 μ -law/PCM 변환 테이블과 $\Delta^{-1}(n)$ 의 테이블 만큼 더 많은 메모리를 사용하고 있다. 데이터 ROM에서 scale factor와 inverse scale factor는 ADPCM 출력 bit를 조절하는데 사용되며 $r(n)$ 은 스텝사이즈를 적응시킬때 사용된다. ADPCM 알고리즘에서 시간지연은 예측기에서의 하나밖에 없기 때문에 내부 RAM은 많이 사용하지 않았다. RAM은 알고리즘 수행에 필요한 값들을 저장하는 곳으로 사용되고 있다. 부호

표 6. 데이터 ROM 사용표
Table 6. Description of the data ROM.

A) 부 호 기

주소(HEX)	내 용
002-081	μ -law / 선형 PCM 변환 테이블
082-101	역 스텝사이즈 테이블
102-181	스텝사이즈 테이블
182	scale factor
183	inverse scale factor
184-18B	16개의 γ 값

B) 복 호 기

주소(HEX)	내 용
002-081	스텝사이즈 테이블
082	scale factor
083	inverse scale factor
084-08A	16개의 γ 값
100-107	2의 누승($2^i - 2^*$)

기에서는 전체 128word중 24word(18.8%)를 복호기에서는 25word(19.4%)를 사용하고 있다.

Ⅶ. 結 論

본 논문에서는 디지털 음성 코우딩 방식중에서 파형 코우딩 방식에 속하는 32kbps ADPCM을 NEC 7720 DSP를 사용하여 실시간 구현하였다 또한 프로그램을 최적화 하여 하나의 NEC 7720으로 4 채널을 동시에 부호화 또는 복호화할 수 있도록 하였으며, 약간의 프로그램의 수정만으로 전송율을 24-64kbps로 변화시킬 수 있도록 하였다. ADPCM 알고리즘을 수행하는데 소요된 시간은 한개의 샘플을 처리하는데 부호기에서는 23.75 μ s, 복호기에서는 24.75 μ s가 각각 필요하였고 4 채널을 모두 처리하는데는 각각 113.5 μ s 및 119.5 μ s가 필요하였다. 본 논문에서는 알고리즘이 비교적 간단한 Jayant의 ADPCM 코우딩 알고리즘을 실시간 구현하였으나 현재 CCITT에서 제정한 표준 ADPCM 알고리즘도 NEC 7720으로 쉽게 구현할 수 있다. 음성통신의 광범위한 응용을 위해서는 실시간 동작 codec의 개발은 필연적이다. 그러므로 NEC 7720이나 TMS 320 같은 DSP로써 소형이고 유연성 있는 RELP나 LPC vocoder 같은 codec의 개발도 아울러 국내에서 기대되며 또한 가능한 것으로 생각된다.

參 考 文 獻

- [1] J.L. Flanagan, et al., "Speech Coding," *IEEE Trans. Commun.*, vol. COM-27, no. 4, pp. 710-736, April 1979.
- [2] L.R. Rabiner, et al., *Digital Processing of Speech Signals*, Prentice-Hall Inc., Englewood Cliffs, New Jersey 1978.
- [3] N.S. Jayant, "Digital Coding of Speech Waveforms: PCM, DPCM, and DM Quantizers," *Proc. IEEE*, vol. 62, pp. 611-632, May 1974.
- [4] N.S. Jayant, et al., "Adaptive Quantization in Differential PCM Coding of Speech," *BSTJ*, vol. 52, no. 7, pp. 1105-1118, September, 1973.
- [5] P. Cummiskey, "Adaptive Quantization in Differential PCM Coding of Speech," *BSTJ*, vol. 52, no. 7, pp. 1105-1118, Sept. 1973.
- [6] S. Bates, *A Hardware Realization of a PCM-ADPCM Code Converter*. S.M. Thesis, Dept. Electrical of Engineering, Massachusetts Institute of Technology, 1976.

- [7] H.W. Adelman, et al., "An ADPCM Approach to Reduce the Bit Rate of μ -Law Encoded Speech," *BSTJ*, vol. 58, no. 7, pp. 1659-1671, Sept. 1979.
- [8] J.J. Dubnowski, "A Microprocessor Log PCM/ADPCM Code Converter," *IEEE Trans. Commun.*, vol. COM-26, no. 5, pp. 660-664, May 1978.
- [9] J.P. Agrawal et al., "The Design of an ADPCM/TASI Systems for PCM Speech Compression," *IEEE Trans. Commun.*, vol. COM-29, pp. 1393-1398, Sept. 1981.
- [10] J.R. Boddie, "Adaptive Differential Pulse-Code-Modulation Coding," *BSTJ*, vol. 60, no. 7, pp. 1547-1561, Sept. 1981.
- [11] "Joint Special Issue on Integrated Circuits for Speech," *IEEE Trans. ASSP* vol. ASSP-31 (part II), whole issue, Feb. 1983.
- [12] CCITT Yellow Book, *Volume III-Fascimile 111.3 Digital Net works-Transmission Systems and Multiplexing Equipment*, Geneva, Nov. 1980.
- [13] J.D. Gibson, "Sequentially Adaptive Backward Prediction in ADPCM Speech Coders," *IEEE Trans. Commun.*, vol. COM-26, no. 1 pp. 145-150, Jan. 1978.
- [14] C.S. Xydeas et al., "Sequential Adaptive Predictors for ADPCM Speech Encoders," *IEEE Trans. Commun.*, vol. COM-30, no. 8, pp. 1942-1954, Aug. 1982.
- [15] NEC Microcomputer, μ PD7720 Signal Processing Interface (SPI) User's Manual.
- [16] NEC Microcomputer, EVAKIT-7720 Operational Manual.
-