

論 文
36~5~8

# NEC 7720 DSP를 이용한 적응 자기 동조 필터의 실시간 구현

## A real Implementation of an Adaptive Self-tuning Filter Using an NEC 7720 DSP

李 淵 錫\* · 李 商 郁\*\* · 李 章 揆\*\*\*  
(Yeon-Seok Lee · Sang-Uk Lee · Jang-Gyu Lee)

### Abstract

In this paper we have designed and implemented a real time ALE (adaptive line enhancer) using a high speed digital processor, NEC 7720. For the ALE system, we have employed an adaptive LMS (least mean square) algorithm proposed by Widrow and Hoff and a 32-order FIR (finite impulse response) filter. Extensive computer simulations have been performed to investigate the performance of the ALE and to determine necessary parameters for hardware design. The developed software for an NEC 7720 was tested in real time operation using an NEC 7720 hardware emulator. The ALE has been tested by sinusoidal waves and real CW (continuous wave) signals. It was found that the experimental results were well agreed with the computer simulation results. Thus it can be concluded that the ALE is useful for detection and enhancement of a sinusoidal signal which is corrupted by an additive Gaussian noise.

### 1. 서 론

오래전부터 잡음에 의해 손상된 신호의 검출 및 복원은 많은 사람들의 관심사가 되어 왔다. 이에 대한 해결책으로 제시된 것 중 필터를 사용하는 방법이 있는데, 미지의 신호를 예측하고 잡음만을 제거 시키는 필터의 설계는 최적 필터의 영역이며 Wiener, Kolmogoroff, Kalman 등에 의하여 많은 연구가 이루어져 왔다.<sup>1,2)</sup> 신호나 잡음에 대한 사전 지식이 있을 경우에는 Wiener나 Kalman 등이 제시된 이론을 적용하여 시간영역이나 주파수 영역에서

주어진 목적함수를 가장 좋게하는 최적 필터를 설계 할 수 있다. 그러나 신호나 잡음에 대한 사전 지식이 없거나 또는 그 성질이 시간에 따라 변하는 경우에는, 필터의 임출력으로부터 얻은 정보를 이용하여 필터의 특성을 원하는 방향으로 적응시키 나가야 하며 이러한 필터를 적응 필터(adaptive filter)라고 부른다.<sup>3)</sup> 적응 필터의 이론은 Widrow가 FIR 필터를 이용한 LMS (least mean square) 알고리즘을 제안한 이후,<sup>4)</sup> 필터의 형태나 필터의 계수를 조절하는 적응 알고리즘들이 아주 다양하게 발전되어 왔다.<sup>5) ~7)</sup>

적응 필터의 응용은 아주 광범위하나 그 중 한가지가 정현파신호로부터 잡음을 제거하여 정현파 신호의 검출 및 복원을 목적으로 하는 ALE (adaptive line enhancer)이다. ALE에 대한 이론적인 해석은 지금까지 많은 사람들에게 의하여 연구되었으며<sup>8) ~10)</sup> 특히 Widrow의 LMS 알고리즘은 그 효율성과 단순성 때문에 가장 널리 사용되고 있다. ALE를 실제

\*正 會 員 : 서울大 工大 制御計測 工學科 博士課程

\*\*正 會 員 : 서울大 工大 制御計測 工學科 助教授 · 工博

\*\*\*正 會 員 : 서울大 工大 制御計測 工學科 副教授 · 工博

接受日字 : 1986年 8月27日

1次修正 : 1987年 2月12日

2次修正 : 1987年 3月24日

원하는 목적에 사용하기 위해서는 실시간(real time) 동작의 하드웨어가 필요하다, Widrow의 LMS 알고리즘의 실시간 구현(real time implementation)은 아직까지도 많은 어려움이 있다. 그러나 최근 반도체 기술 특히 VLSI(very large scale integration) 기술의 급속한 발전에 힘입어 특수용 신호처리 프로세서들이 소개됨으로써 과거에는 거의 불가능하던 알고리즘들의 실시간 처리가 가능하게 되었다. 이러한 프로세서를 사용하게 되면 대부분의 연산을 하드웨어가 아닌 소프트웨어로 처리할 수 있으므로 간단한 하드웨어의 구성으로 실시간 처리가 가능할 뿐만 아니라 또한 소프트웨어의 수정 만으로 다른 기능들을 첨가할 수 있어 유연성(flexibility) 있는 시스템의 설계가 가능하다.<sup>11, 12)</sup>

따라서 본 논문에서는 현재 많이 사용되고 있는 NEC7720 프로세서<sup>13)</sup>를 사용하여 실시간 ALE의 하드웨어 및 소프트웨어의 구현 및 그 성능을 조사하였다. ALE의 적응 알고리즘은 Widrow의 LMS 알고리즘을 선택하였다. ALE의 설계시 필요한 모든 변수들은 컴퓨터 시뮬레이션을 통해 결정하였으며 실제 제작된 ALE는 잡음에 손상된 CW(continuous wave) 신호를 사용하여 그 성능을 확인하였다. 본 논문의 구성은 다음과 같다.

먼저 2장에서 적응 LMS 알고리즘을 간단히 설명한 후 3장에서 컴퓨터 시뮬레이션 결과 및 하드웨어 설계시 필요한 변수 결정을 서술하였다. 4장에서는 하드웨어에 관한, 5장에서는 소프트웨어에 관한 설명을 한 후 6장에서 실험결과를 논하였다. 끝으로 결론을 7장에 제시하였다.

### 2. 적응 LMS 알고리즘

그림 1은 ALE의 블록선도를 도시하고 있는데 2 채널 잡음제거기의 형태와 유사하다. 즉 기초 입력(primary input)으로는 정현파에 잡음이 섞인  $x_j$ 를

받아들이고 기준입력(reference input)으로는 기초 입력을  $\Delta$ 만큼 지연시켜 사용하고 있다. 여기에서  $\Delta$ 를 비상관 집단지연(decorrelation group delay)이라고 부르는데, 역할은 입력신호에 포함된 주기적 신호는 상관관계를 유지하도록하고 광대역 신호는 서로 비 상관 관계를 갖도록 한다. 즉 광대역 신호(ALE에서는 잡음)는 시간에 따라 상관관계가 저하되어 비 상관화되나 주기적 신호는 시간의 경과에 따라 상관 관계를 유지하는 성질을 이용하여 주기적 신호와 광대역 신호를 분리해 내는 역할을 하게 된다. 시간  $j$ 에서의 입력신호를  $x_j$ 라고 하면 입력신호 벡터  $\bar{x}_j$  및 필터 무게 벡터  $\bar{w}_j$ 를 다음과 같이 정의할 수 있다.

$$\begin{aligned} \bar{x}_j &= (x_{j-\Delta}, x_{j-\Delta-1}, \dots, x_{j-\Delta-n+1}) \\ \bar{w}_j &= (w_{1j}, w_{2j}, \dots, w_{nj}) \end{aligned} \quad (1)$$

또한 필터 출력  $y_j$  및 오차  $\epsilon_j$ 는 다음과 같이 구할 수 있다.

$$\begin{aligned} y_j &= \bar{x}_j \bar{w}_j = \bar{w}_j \bar{x}_j \\ \epsilon_j &= x_j - y_j \end{aligned} \quad (2-a)$$

그러면 LMS 알고리즘은 일종의 steepest descent 방법으로서 오차  $\epsilon_j$ 의 제곱값을 최소화하는 필터 무게 벡터  $\bar{w}_j$ 를 식(2-b)와 같이 반복적으로 찾아가게 된다.<sup>3)</sup>

$$\bar{w}_{j+1} = \bar{w}_j + 2\mu \epsilon_j \bar{x}_j \quad (2-b)$$

윗식에서  $\mu$ 의 역할은 LMS 알고리즘에서 안정성 및 수렴속도를 결정하여 주는 중요한 변수로서 다음과 같은 조건을 만족하여야 한다.<sup>3)</sup>

$$0 < \mu < \frac{1}{\lambda_{max}} \quad (3)$$

여기에서  $\lambda_{max}$ 는 입력 autocorrelation 행렬의 최대 고유치(eigenvalue)를 의미한다.

### 3. 컴퓨터 시뮬레이션

먼저 ALE의 하드웨어를 실제로 설계, 구현하기 앞서 컴퓨터 시뮬레이션을 수행하였다. 시뮬레이션의 목적은 ALE의 동작 특성을 파악하여 하드웨어 설계시 필요한 여러 변수, 즉 필터의 차수,  $\Delta$  및  $\mu$ 의 값을 결정하기 위함이다. 입력 데이터로는 CW 신호에서 가청 tone 신호로 사용되고 있는 600 - 2000Hz의 정현파를 사용하였는데 주파수는 임의로

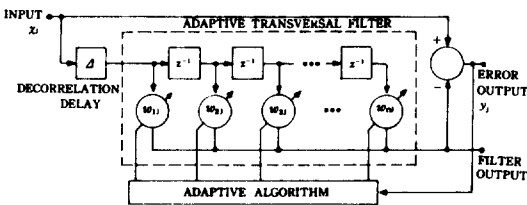


그림 1. FIR 필터를 이용한 ALE의 구조  
Fig.1. ALE structure using a FIR filter.

선정하였다. 잡음원으로는 일반적인 백색 가우시언 잡음과 Poisson분포를 갖는 충격과 잡음을 사용하였는데 충격과 잡음은 ALE의 비가우시언 잡음의 제거 능력을 파악하기 위함이다. 또한 모든 시뮬레이션은 샘플링 주파수를 8000 Hz로 가정하였다.

3.1  $\mu$ 값에 따른 영향

전술한 것 같이 LMS 알고리즘에서  $\mu$ 값의 선택은 안정성 및 수렴속도를 결정하므로 대단히 중요하다. 일반적으로  $\mu$ 값이 크면 수렴속도는 빨라지나 발산하는 경향이 있고, 너무 작으면 잡음의 제거 효과는 크나 수렴속도가 늦어진다. 충분히 수렴되었다고 판단될때  $\mu$ 값의 변화에 따른 S/N이득을 표 1에 제시하였다. 이때 입력 신호는 1004Hz 정현파이고 입력 S/N은 3dB였으며 32차의 필터를 사용하였다. 그림 2는 이때의 출력 파형을 도시한 것인데  $\mu$ 가 클수록 빨리 정상상태에 도달하나 파형의 왜곡이 심함을 알 수 있다 또한  $\mu$ 가  $10^{-2}$ 보다 큰 경우에는 발산하였다. 그림에서 보면 파형에서 진폭의 변화가 발생함을 알 수 있는데, 이는 ALE가 정현파 주변의 잡음에 완전히 제거하지 못함으로써 인한 진폭 변조 효과에 의한 것이다. 이와 같이  $\mu$ 는 수렴속도와 출력의 안정도에 영향을 주게 되는

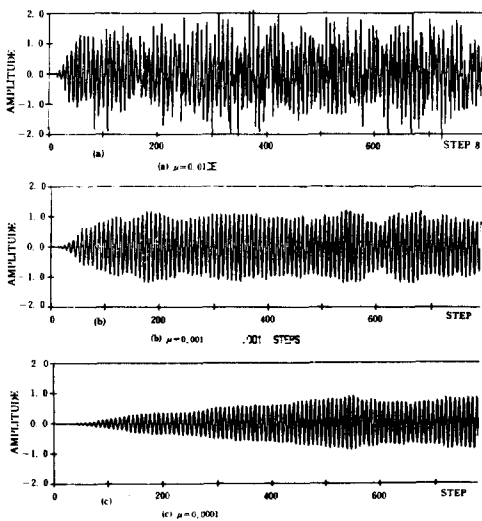


그림 2.  $\mu$ 에 따른 출력 파형 특성 (주파수=1004 Hz)

Fig.2. Output waveform characteristics with a different  $\mu$  (frequency = 1004Hz)

표 1.  $\mu$ 값에 따른 S/N비 이득 (주파수=1004Hz,  $\theta$  필터차수=32)

Table 1. S/N ratio gain with different  $\mu$  (frequency=1004Hz, filter order=32)

$\mu$	이득 (dB)
$10^{-2}$	0.34
$10^{-3}$	9.14
$10^{-4}$	6.88

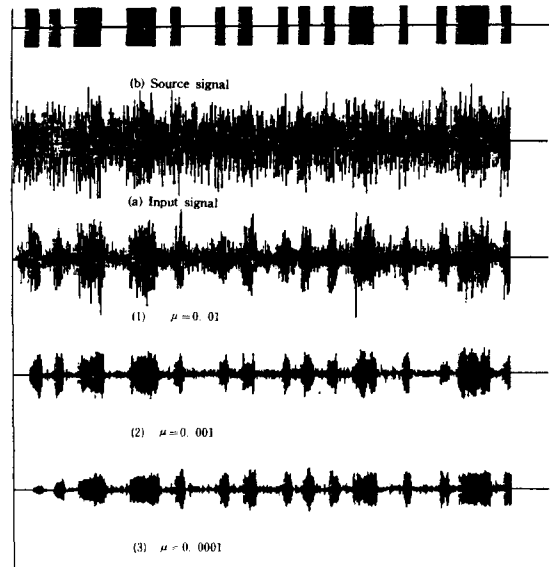


그림 3. CW 신호에 대한 시뮬레이션 결과

Fig.3. Simulation results on the cw signal

데 실제 CW (모오스 부호) 신호에 적용하였을 때의 출력을 그림 3에 도시 하였다. 이때 CW 신호는 매 분당 150 character의 전송 속도를 가지며 각 character들은 평균 20 unit-gap으로 구성되어 있다고 가정하여 얻었으며, 입력 S/N은 3dB로 하였다. 시뮬레이션에서  $\mu$ 를 변화시켜가며 출력을 얻었는데  $\mu$ 가 작으면 수렴속도가 늦어 정확한 모오스 부호의 복원이 어렵고 이와 반대로 너무 크게 하면 오차가 증가함을 알 수 있다. 따라서 본 논문에서는 이상과 같은 시뮬레이션 결과를 종합하여  $\mu = 10^{-3}$ 으로 결정하였다.

3.2 집단 지연 상수

집단 지연상수  $\Delta$ 는 잡음의 성질에 따라 적절히

표 2.  $\Delta$ 에 따른 S/N비 이득

Table 2. S/N ratio gain for the different  $\Delta$

집 단 지 연 $\Delta$	이 득 (dB)
1	8.78
3	9.08
5	9.96
8	9.38
10	9.69
15	10.35
35	8.64

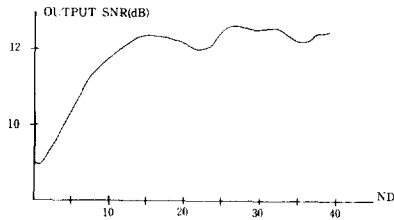


그림 4. 집단지연 상수  $\Delta$ 에 따른 출력 SNR이득  
Fig. 4. Output SNR gain versus a group delay constant  $\Delta$

선택 되어져야 할 값이나 백색 가우시언 잡음의 경우에는 데이터들간의 상관 관계는 0이므로  $\Delta$ 의 크기에 별 영향이 없다. 표2는  $\Delta$ 의 변화에 따른 S/N비의 이득을 나타낸 것으로 표에서 보듯이  $\Delta = 1$  이후로는 이득의 변화가 별로 변하지 않고 있다. 이는 잡음이 백색 잡음이므로  $\Delta$ 의 크기가 1만 되어도 상관 관계가 0이 되기 때문이다. 따라서 비백색 잡음에 대한  $\Delta$ 의 영향을 알아 보기 위해 Poisson분포를 갖는 충격파 잡음에  $\Delta$ 의 크기를 0에서 40까지 변화시켜가며 S/N이득을 계산한 결과를 그림 4에 제시하였다. 그림 4에서 보면  $\Delta$ 가 15까지 증가할 때에는 S/N이득이  $\Delta$ 에 비례하여 증가하지만, 그 이상이 되면 S/N이득이 포화됨을 알 수 있다. 즉 충격파 잡음인 경우  $\Delta$ 는 어느 정도 큰 값을 가져야만 동작을 함을 알 수 있는데, 실제 하드웨어 구현시는 NEC7720의 실행시간 및 내부 구조 등을 고려하여  $\Delta$ 의 크기를 1-16까지 가변할 수 있도록 하였다.

### 3.3 필터 차수에 따른 이득

필터 차수가 LMS 알고리즘에 미치는 영향을 관찰하기 위해  $\mu=10^{-3}$ ,  $f=1004\text{Hz}$ ,  $\Delta=5$ 일 때 입력 S/N비 및 필터 차수의 변화에 따른 출력 S/N비를 표 3에 나타내었다. 필터 차수가 증가함에 따라

표 3. 필터 차수에 따른 S/N비 이득

Table 3. S/N ratio gain for the different filter order

필터 차수	입력 S/N		
	-3	0	5
8	4.59	6.84	10.84
16	6.03	9.10	13.18
32	6.11	10.48	15.25

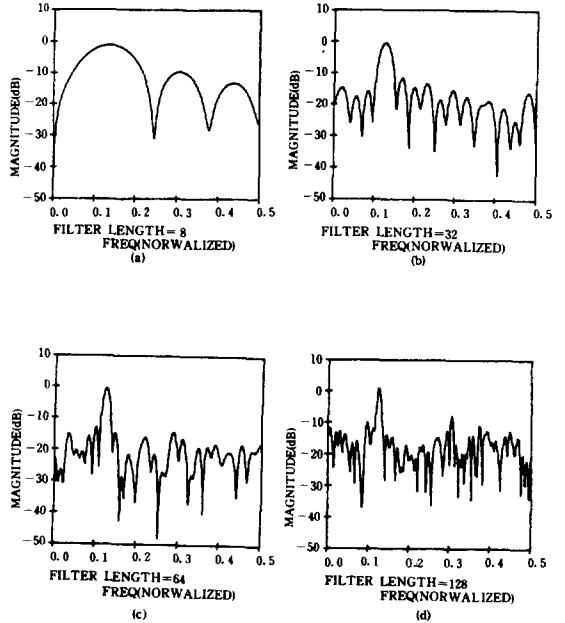


그림 5. 필터 차수에 따른 필터의 주파수 특성

(a)  $N=16$  (b)  $N=32$  (c)  $N=64$  (d)  $N=128$

Fig. 5. Frequency response characteristics with different filter orders

(a)  $N=16$  (b)  $N=32$  (c)  $N=64$  (d)  $N=128$

S/N비가 증가하고 있음은 잡음제거 효과가 향상됨을 알 수 있다. 그림 5는 필터 차수 8, 32, 64 및 128일 때 정상 상태에서의 주파수 특성을 도시한 것이다. 그림에서 보면 필터의 특성이 주어진 정현파 주파수에서 정확히 peak치를 보이고 있어 ALE가 적응자기동조(adaptive self-tuning) 필터로 동작됨을 알 수 있다. 또한 필터의 차수가 증가할수록 대역폭이 감소되고 있어 잡음제거 효과가 향상됨을 알 수 있으나, 64차와 128차를 비교해 보면 대역폭에 별 차이가 없음을 알 수 있다. 즉 필터의 차수를 증가하더라도 잡음제거 효과는 이에 비례하지

않게 된다. 따라서 본 논문에서는 NEC7720 이 128 개의 RAM밖에 가지는 못하는 것을 고려하여 필터의 차수를 32차로 결정하였다.

3.4 입력 S/N비에 따른 이득

그림 6은 필터 차수 32차,  $f=777\text{Hz}$ ,  $\Delta=3$ 일때 입력 S/N비에 따라 S/N이득을 도식한 것이다. 그림에서 보면 입력 S/N비가 약 0dB 이상이면 약 10 dB 정도의 S/N이득을 얻을 수 있으나, 입력 S/N비가 그 이하이면 이득은 이와 비례하여 감소됨을

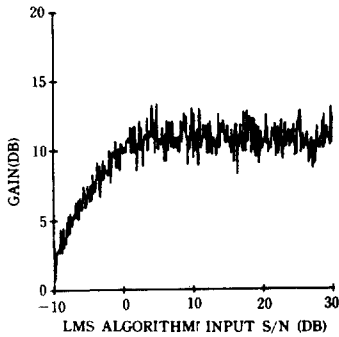


그림 6. 입력 S/N비에 따른 출력 S/N비 이득 곡선  
Fig.6. Curve of output S/N ratio versus input S/N ratio

알 수 있다. 이는 잡음의 전력이 커질수록 정현파를 tracking하는 것이 힘들어지는 것을 의미한다.

3.5 고정 소숫점 연산 시뮬레이션

이상의 모든 시뮬레이션은 유동소숫점(floating-point)연산의 결과인데 실제 하드웨어 구현시는 연산속도의 제한으로 고정 소숫점(fixed-point) 연산을 사용하게 된다. 고정 소숫점 연산시 곱셈에서는 절삭오차(truncation error), 덧셈에서는 overflow 오차가 발생하므로 이에 대한 고찰이 반드시 필요하다.<sup>14)</sup> NEC7720은 내부에 16bit의 승산기(multiplier)가 내장되어 있으므로 16bit 고정 소숫점 연산을 수행할 수 있다. 또한 입력 신호의 A/D 변환은 12bit를 사용하였으므로 이에 대한 고정소숫점 연산 시뮬레이션 결과를 표 4에 제시하였다. 시뮬레

표 4. 고정 소숫점 연산 결과

Table 4. Results of the fixed-point arithmetic

$\mu$	(SNR) 입력(실수형)	양자화 양자화	(SNR) 출력(유동)	(SNR) 출력(고정)
0.01	0.412	0.425	5.361	5.174
0.001	-0.121	-0.134	10.201	10.704
0.0001	-0.340	-0.127	12.376	12.011
0.00001	-0.017	-0.146	10.984	11.240

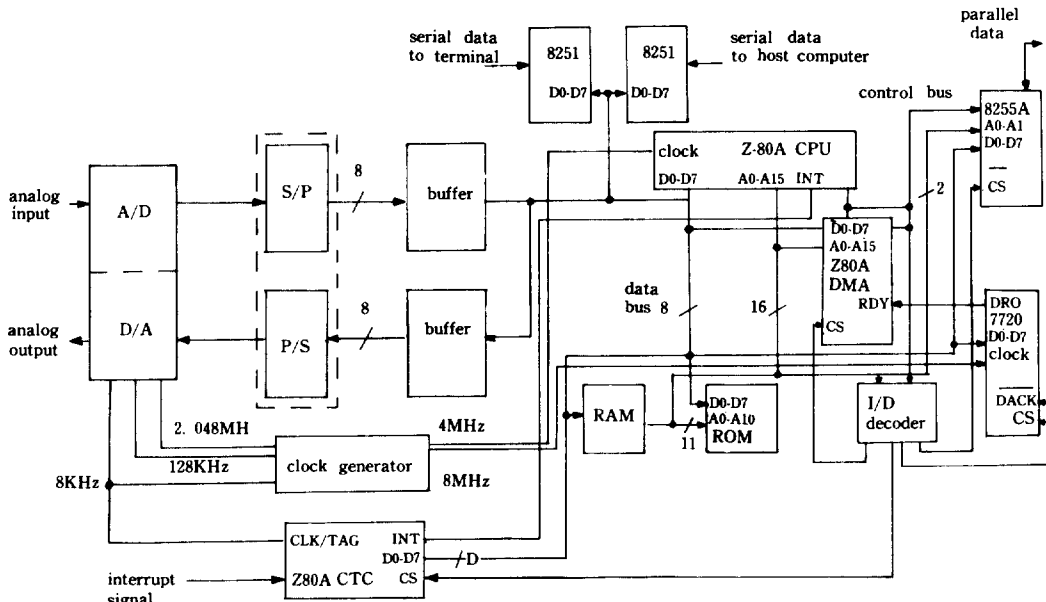


그림 7. 하드웨어 시스템의 블록선도  
Fig.7. Block diagram of hardware system

이선 결과에 의하면, 입력을 12bit 고정 소숫점연산을 하면 그 오차는 유동소숫점 연산과 비교할 때 거의 무시할 수 있음을 알 수 있다.

#### 4. 하드웨어 구현

ALE를 하드웨어로 구현할 때 소형이고 유연성이 있으며 또한 경제성 있는 시스템이 되도록 하였다. 시스템에 유연성을 부여하기 위해서는 모든 연산을 하드웨어로 하기보다 수정이 용이한 소프트웨어적으로 처리하여야 하는데 NEC7720은 내부에 16 bit × 16 bit → 31 bit 승산기가 내장되어 있어 이에 적합하다. 그림 7에 하드웨어 시스템의 블럭선도가 주어 있는데 크게 나누어 아날로그 입출력부, A/D, D/A 변환부, NEC 7720을 사용한 연산부, 전체 시스템을 제어하기 위한 제어부 및 전체 시스템의 동기화 및 필요한 clock을 발생시키는 clock 발생부 등으로 나눌 수 있다. 입출력 신호는 12-bit A/D, D/A 변환기를 사용하였는데 이때 입, 출력에 사용되는 저역 필터와 이득 조정 등은 OP amp. 를 사용하여 구성 하였으며, 입력 이득은 정밀한 triming potentiometer를 사용하여 0-100배의 이득 조정이 가능 하도록 하였다. 저역 필터는 OP amp. 를 이용한 VCVS (voltage controlled voltage source) 회로로 구성하였고 차단 주파수는 3.4kHz로 하였다. A/D 변환기는 ADC-Hz 12BMR 을 D/A 변환기로는 DAC-Hz 12BMR-1을 사용하였다. 한편 전체 시스템을 제어하기 위한 제어부는 Z-80마이크로 프로세서를 사용하여 설계하였는데 NEC7720 DSP와 A/D, D/A 변환기를 제어하고 입출력 신호 및 여러가지 필요한 option 입출력 등을 관장하게 된다.

Z-80제어부의 구성은 Z-80 CPU와 ROM, stack pointer 등의 메모리로 사용되는 RAM, Z-80 CPU를 구동시키고 전체 시스템의 동기화를 위한 clock 발생부, interrupt 처리를 위한 CTC(counter timer controller), address decoder, 그리고 외부와의 interface를 위한 8255 PPI(parallel port interface) 등으로 구성되어 있다. Clock 발생부에서는 NEC7720을 구동하기 위한 8MHz, Z-80 CPU를 위한 4MHz 및 A/D변환기에 필요한 8KHz clock등을 공급하며 8KHz clock은 CTC의 trigger 입력단에 아울러 공급되어 전체 시스템을 sampling 주파수 8KHz에 동기시켜 주었다. 이외에도 외부에서 수렴상수  $\mu$ 를 decimal switch를 통하여 변화시킬 수 있도록하여 8255 PPI를 통하여 Z-80 CPU

가 읽어 들이도록 함으로써 입력에 따라 원활히 동작 되도록 하였다.

#### 5. 소프트웨어

NEC7720의 프로그램은 Z-80 프로그램과 함께

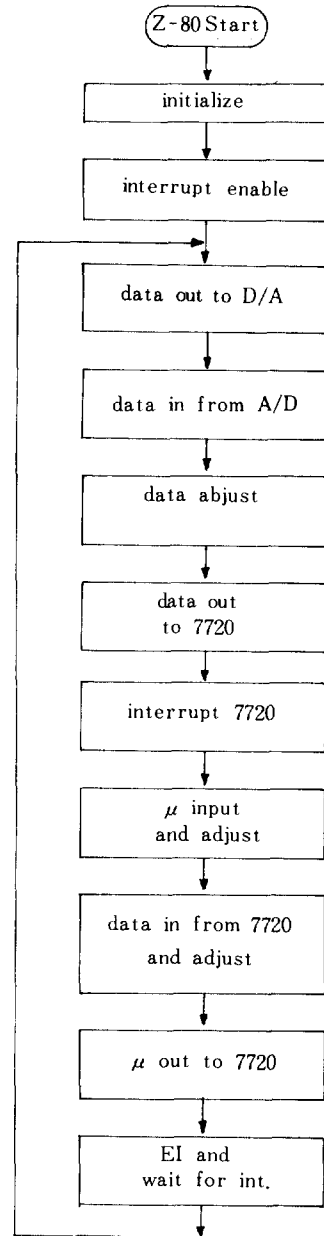


그림 8. Z-80 소프트웨어 흐름도  
Fig. 8. Flowchart for a Z-80 software

구성된 하드웨어에서 real-time emulator인 EVA KIT-7720를 이용하여 그 동작을 확인한 후 NEC 7720에 실제 프로그램을 수록하여 실시간 동작이 이루어짐을 다시 확인하였다.

그림 8은 Z-80 제어프로그램이 흐름도를, 그림 9는 NEC7720 연산프로그램의 흐름도를 각각 보여 주고 있다. 프로그램들 중에서 몇가지 중요한 부분만을 설명하기로하면 먼저 Z-80에서는 NEC 7720으로 데이터를 보내주고나서 interrupt를 인가시키는데, NEC7720에서는 이 interrupt를 인지하고 나서 데이터를 받아들여지게 된다. 또한 Z-80에서 NEC7720으로부터 데이터를 주고나면 NEC7720으로부터 출력데이터를 받아들여지게 되는데 NEC7720은 이 사이에 새로이 입력된 데이터를 처리하고 있으며 출력시킨 데이터는 한 sample 이전의 출력이다. 이렇게 함으로써 NEC7720은 데이터를 주고받는 시간을 절약하게 된다.

앞에서  $\mu$ 의 크기는 매우 중요하다고 하였는데,

$\mu$ 는 0.00006에서 0.00812까지를 128등분하여 그 사이의 값들 중에서 외부에서 선택할 수 있도록 하였고 이를 1초로 8000번 Z-80 CPU가 읽어들이므로 외부에서 선택하기만 하면 그에 따라 즉시 바꾸어 사용하게 된다. 집단지연은 NEC7720의 내부에서 16샘플데이터로 처리하였는데, 내부의 DP레지스터를 이용한 ring카운터 방식으로 처리하였으며, 따라서 프로그램 흐름도에서는 data in의 과정에서부터 data out 과정 사이에 모두 처리된다.

앞의 모든 과정을 처리하기 위하여 NEC7720에서는 16%의 instruction ROM과 64%의 data RAM을 사용하였으며, 표 5는 NEC7720이 수행하는 각 작업들의 소요되는 명령어 수와 시간을 보여주고, 표 6은 NEC7720의 내부 data RAM의 사용 용도를 보여준다. 프로그램의 소요시간은 Z-80 프로그램이 63.5  $\mu$  sec, NEC7720 프로그램이 92.25  $\mu$  sec로 모두 125  $\mu$  sec내에 수행될 수 있어 실시간 처리가 이루어짐을 알 수 있다.

표 5. 소프트웨어의 설명

Table 5. Software description

( $\mu$  sec)

내 용	명령어수	실행시간
Initialization Routine		
초기화 작업	38	97.00
interrupt enable	2	0.50
Interrupt Service Routine		
interrupt 인지작업	2	0.50
data의 입출력 및 집단지역 작업과 $\epsilon_j$ 의 계산	9	2.25
필터의 계수 순환식 및 단위지연 작업	15	56.00
필터의 출력 계산	10	32.00
$\mu$ 의 입력	4	1.00
interrupt enable	2	0.50

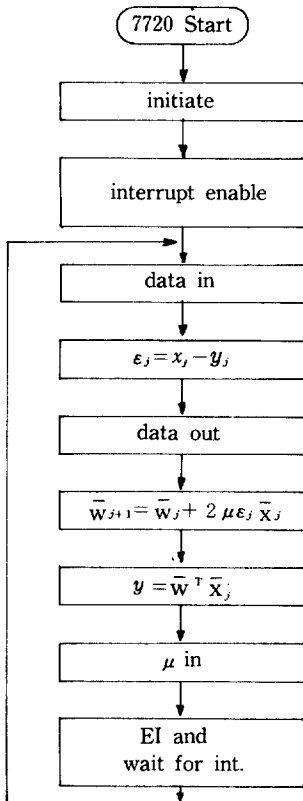


그림 9. NEC 7720 소프트웨어 흐름도  
Fig. 9. Flowchart for a NEC 7720 software

### 6. 실험결과

그림 10에 실제 제작된 ALE의 실험 블럭선도를 보인다. 그림에서 보듯이 입력 신호는 정현파와 백색 가우시언 잡음을 직접 섞어 사용하였는데 입력 주파수는 400-2000Hz에서 적당히 선택하였다. 그림 11에 입출력 파형을 도시 하였는데 입출력을 비교하면 잡음이 많이 감소됨을 알 수 있다. 여기에서 한가지 부언할 것은 실시간 처리 ALE 동작성을 파악하기 위해서는 실제 입출력 신호의 S/N 비

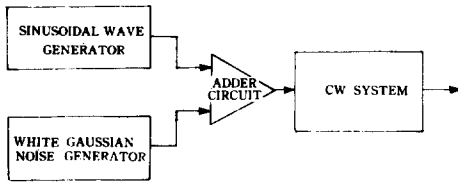


그림 10. 실험 블록선도  
 Fig. 10. Block diagram for experiments

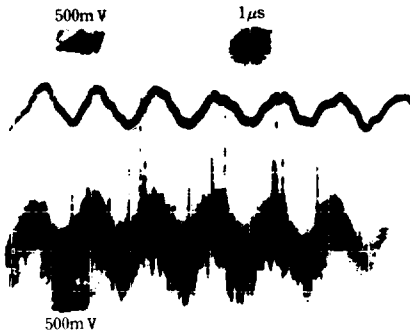


그림 11. 입출력 파형 예(상단 : 출력, 하단 : 입력, x 축 : 시간, y 축 : 전압)  
 Fig. 11. Example of input/output waveforms(top : output, bottom : input, x-axis : time, y-axis : magnitude in volts)

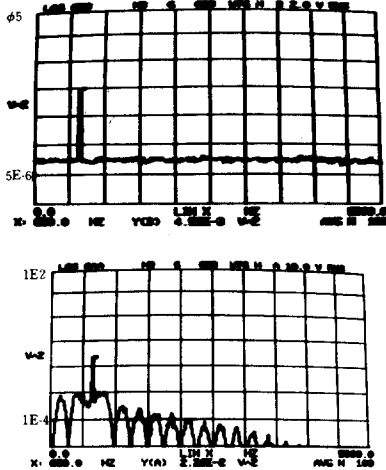


그림 12. 입출력 신호의 스펙트럼(상단 : 입력, 하단 : 출력, x 축 : 주파수, y 축 : 크기)  
 Fig. 12. Spectrum of the input/output signals (top : input, bottom : output, x-axis : frequency in Hz, y-axis : magnitude)

를 측정하는 것이 바람직하다. 그러나 정현파에 잡음이 섞혀 있을때 정현파와 잡음을 분리하여 S/N 비를 직접 측정한다는 것은 대단히 어려운 일이다. 따라서 본 논문에서는 oscilloscope나 spectrum analyzer를 이용하여 시각적으로 ALE의 동작상태를 확인하였다. 그리고 소프트웨어를 debugging 할 때 고정소숫점 연산 시뮬레이션 결과와 실제3장에서 얻은 시뮬레이션 결과가 거의 동일함을 여러번 확인하였으므로 본 ALE도 정상 상태에서의 시뮬레이션 결과처럼 약 9-10dB 정도의 S/N이득을 충분히 얻을 수 있다고 판단된다. 이런 사실은 입, 출력 신호의 스펙트럼을 보면 더욱더 확실하다. 그림 12에 입출력 신호의 스펙트럼을 제시하였는데 입,

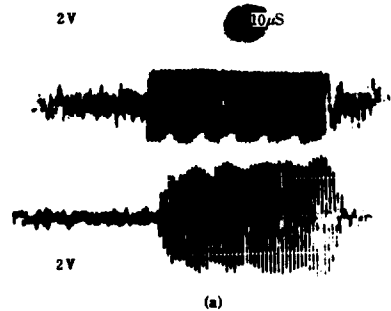


그림 13. CW 신호의 입출력 파형 예(상단 : 입력, 하단 : 출력, x 축 : 시간, y 축 : 전압)  
 (a) CW 신호  
 (b) 확대한 파형

Fig. 13. Example of input/output waveform of the CW signals (top : input, bottom : output, x-axis : time, y-axis : volt)  
 (a) CW signal  
 (b) magnified waveform



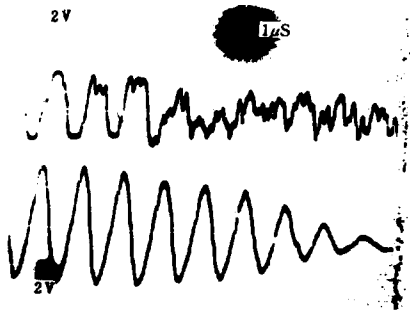


그림 14. CW 신호의 수렴 특성(상단 : 입력, 하단 : 출력)

Fig. 14. Convergence characteristic of the CW signal (top : input, bottom : output)

표 6. 데이터 RAM의 사용

Table 6. Usage of the data RAM

내 용	주 소(HEX)
집단지연작업	00~0F
DP stack	10
$\bar{w}$ 벡터	20~3F
$\mu$ 저장	50
$\bar{x}$ 벡터	60~7F

출력을 비교해보면 잡음의 제거 효과가 뚜렷이 나타나고 있다. 다음은 실제 모오스 코오드 신호 즉 CW신호를 ALE에 입력시켜 그 파형을 그림 13에 도시하였다. 그림 13은 CW신호이며 상단부가 입력 신호 하단부가 출력신호인데 잡음이 상당히 감소되어 ALE의 잡음제거 효과가 있음을 알 수 있다. 그림 14는 입력에서 정현파가 없어졌을때(즉 모오스 코오드에서 "space"에 해당되는 부분) 출력에서 그 상태로 수렴하는 과정을 잘 보여주고 있다. 그림 14에서 보면 CW신호에서 정현파가 존재하는 시간은 약  $50 \mu\text{sec}$ 인데 수렴속도는 약  $5 \mu\text{sec}$  정도로서 제작된 ALE는 시뮬레이션 결과에서 구해진 특성에 따라 동작되며, CW 신호의 실시간 잡음 제거기로 사용되고 있음을 보여 준다. 실제 입, 출력을 증폭하여 speaker를 통해 들어보면 청각적으로 잡음 제거 효과는 매우 좋음을 알 수 있었고, 장시간 가동해도 전혀 무리가 없었다.

### 7. 결 론

본 논문에서는 잡음에 손상된 정현파 신호의 복

원 및 검출을 위하여 실시간 ALE를 고속 신호처리 프로세서인 NEC7720을 사용하여 설계 제작하였다. DSP를 사용함으로써 ALE의 적응LMS 알고리즘에 필요한 대부분의 연산을 소프트웨어로 처리할 수 있어 하드웨어의 복잡성을 줄임은 물론 시스템의 유연성을 강조할 수 있었다. 광범위한 컴퓨터 시뮬레이션을 수행하여 실제 제작되는 ALE의 동작을 파악할 수 있었으며, 하드웨어 설계시 필요한 여러 변수들을 결정하였다. 실제 제작된 ALE는 32차 FIR필터를 사용하였으며 비 상관 집단지연 상수는 16샘플, 수렴상수  $\mu$ 는 시뮬레이션 결과  $\mu=10^{-3}$  정도의 값이 적당하였으나 외부에서 임의로 조정할 수 있게 하였다. 한편 실제 제작된 ALE는 정현파 및 CW신호를 백색 가우시언 잡음을 사용하여 실시간 동작을 확인하였는데 정상상태에서 약 9-10dB 정도의 S/N이득을 얻을 수 있을 것으로 믿어진다. 따라서 본 ALE는 잡음에 손상된 정현파 신호의 복원 및 검출에 충분히 이용될 수 있다. 또한 본 논문에서 구현한 적응LMS 알고리즘은 적응 안테나 시스템, 시스템 인식, 채널 등화(channel equalization), 반향제거(echo cancellation), 스펙트럼 추정, 시간지연 추정 등에 응용 될 수 있어 앞으로의 연구 결과에 따라 다양하게 이용될 수 있을 것으로 추정된다.

### 참 고 문 헌

- 1) T. Kailath, "A view of three decade of linear filtering theory," IEEE Trans. IT, Vol. IT-20, No. 3, pp. 145-181, March 1974.
- 2) R. E. Kalman, "A now approach to linear filtering and prediction problems," Trans. ASME, J. of Basic Engineering, pp. 35-45, March 1960.
- 3) B. Widrow and S. D. Stearns, Adaptive Signal Processing, Prentice Hall, Inc., Englewood Cliffs, 1985.
- 4) B. Widrow and et. al., "Adaptive noise cancelling: principle and applicattions," IEEE Proc., Vol. 63, No. 12, pp. 1692-1716, Dec. 1975.
- 5) J. R. Treichler, "Transient and convergent behavior of the adaptive line enhancer," IEEE Trans. ASSP, Vol. ASSP-27, No. 1, pp. 53-62, Feb. 1979.
- 6) D. V. Bhaskar and S. Y. Kung, "Adaptive notch filtering for the retrieval of sinusoids in noise," IEEE Rrans. ASSP, Vol. ASSP, No. G, pp. 791-802, Aug. 1984.

- 7) M. L. Honig and D. G. Messerschmitt, Adaptive filters, structure, algorithms and applications, Kluwer Academic Publishers, 1984.
- 8) N. J. Bershad and et. al., "Tracking characteristics of the LMS adaptive line enhancer—response to a linear chirp signal in noise," IEEE Trans. ASSP, Vol. ASSP-28, No. 5, pp. 506-516, Oct. 1980.
- 9) P. L. Feintuch, "An adaptive recursive LMS filter," Proc. IEEE, Vol. 64, No. 11, pp. 1622-1624, Nov. 1976.
- 10) B. Friedlander, "Lattice filters for adaptive processing," Proc. IEEE, Vol. 70, No. 8, pp. 829-867, Aug. 1982.
- 11) E. T. Lee, K. S. Lee and S. U. Lee, "Implementation of a 4-channel ADPCM codec using a DSP," KIEE, Vol. 22, No. 5, pp. 448-47, Sep. 1985.
- 12) J. S. Lee and S. U. Lee, "Implementation of a real-time SIFT pitch detector," KIEE, Vol. 23, No. 1, pp. 101-103, Jan. 1986.
- 13) NEC microcomputer,  $\mu$ PD 7720 Signal Processing Interface (SPI) User's Manual.
- 14) A. V. Oppenheim and R. W. Schaffer, Digital Signal Processing, Prentice Hall Inc., Englewood Cliffs, 1975.