

균일 전력 밀도의 엔벨로프 발생기와 변환 부호화 방식의 정보량 축소를 이용한 음원 전용DSP 설계에 관한 연구

A Study on the Design of Digital Sound Processor for Music using Equal Power Density Envelope Generator and Transform Coder

구재을*, 방효창**, 김종한***, 김원후**

(Jae Ul Koo*, Hyo Chang Pang**, Jong Han Kim***, Won Hoo Kim**)

요약

본 논문은 ADPCM에 MPEG(Moving Picture Expert Group)에서 사용하는 변환 부호화 방식을 이용한 양자화 잡음의 축소와 균등 전력 밀도의 엔벨로프 재생 방식을 이용하여 악기의 특색에 따라 서로 다른 형태의 정보량 축소 방식을 채택한 디지털 음원 DSP에 관하여 기술한다. 이를 검증하기 위하여 32개의 악기 소리를 동시에 구현할 수 있는 일종의 RISC인 음 발생 전용 DSP를 설계하였고 1MByte의 메모리에 200여가지의 악기음을 코딩하여 알고리즘의 정확성을 입증하였다.

ABSTRACT

This paper presents the digital music sound DSP by using ADPCM and Perceptual Transform Coder in MPEG to compress sound data and minimize the quantization noise for musical instrument. These method are utilized to develop algorithm of equal power density envelope. And these results are applied to examine the specific characteristics of musical instrument and determine the compression method. The design of new RISC DSP which generates 32 voices of musical instrument simultaneously and the coding of 200 musical instrument sound data in 1MByte memory shows that these algorithm is very useful to regenerate musical sound by using the minimum size of memory.

1. 서론

전기적인 방법에 의한 소리와 이를 이용한 신디사이저는 오랜 세월 동안 연구되어 왔고 전자 기술과 컴

퓨터 기술의 발달로 많은 발전이 있었다. 이러한 전기적인 방법에 의한 악음은 1903년 Thaddius Cahill에 의해 만들어 졌다^[1]. 1950년대 초기에는 처음으로 전기적인 합성법에 의해 소리를 내는 신디사이저가 나타나기 시작했고 비로소 1960년대 중반에 들어 디지털 컴퓨터 프로그램에 의해 제어가 가능한 소리를 합성시키는데 필요한 중간적인 소자가 없는 신디사이

* (주) 화음소

** 한국항공대학교 항공전자공학과

*** 삼성전자(주) 통신 개발실

접수일자: 1994년 8월 20일

저가 나타나기 시작했다. 이 방법은 매우 유연하고 제한된 주파수 영역내에서 악음을 합성 할 수 있으며 임의의 악음 정보(Sound Information)를 적당한 수학적 Modeling과 컴퓨터 프로그램에 의해 나타낼 수 있고 이러한 Modeling에 의하여 파형을 합성하여 악음을 낸다^[14]. 그러나 이를 실제 multimedia 시스템에 적용시킬 때에는 최소한 24개 이상의 악기음들을 동시에 실시간으로 재생하는 것과 메모리 사이즈의 한계로 인하여 많은 문제가 발생한다.

따라서 본 논문에서는 이러한 악기음들의 정확한 모델링과 재생을 위해 균일 전력 밀도의 엔벨로프 재생 방식과 변환 부호화 방식의 정보량 압축 방식에 대한 연구 결과를 기술한다. 2장에서는 정보량 축소와 수많은 음원 정보의 저장 과정에서 생기는 양자화 잡음의 축소를 위해 MPEG(moving picture expert group)에서 사용하는 변환 부호화 방식을 이용하였으며 음원 정보의 인코딩을 위한 변수 추출을 위해 코딩정보의 적용 계수 추출에 대해서 고찰한다. 3장에서는 2장에서 제안된 방식에 의해 코딩된 정보의 복원을 위한 하드웨어적인 관점에서 Decoder에 대한 원리를 기술함과 아울러 하드웨어의 시간별 구성도를 나타내었다. 그리고 4장에서는 3장에서 디코딩된 신호를 정확히 원래음으로 복원하기 위해 균일 전력 밀도의 엔벨로프 생성 과정을 Partition Crossfading Looping 방식의 적용으로 구현하였다. 5장에서는 실제 최대 50:1까지의 정보량을 악기의 특색에 따라 줄일 수 있는 실험 결과와 이를 일종의 RISC Processor인^[15] sound전용 DSP를 설계하여 적은 양의 메모리의 사용으로 수백가지의 악기소리를 구현하여 평가 함으로써 본 알고리즘의 정확성을 보였고 이를 실제 시스템에 적용 함으로써 실용 가능성을 입증하였다.

II. Adaptive Coefficient 추출

본 논문에서는 저마다 다른 악기들의 공통점을 찾아 Decoder를 만들기 위하여 Perceptual Transform Coder^[16]의 Masking Threshold를 이용하였다.

첫째, 원래의 Sampling음에 1/16길이의 식(1)과 같은 Hanning윈도우 function을 취하였다.

$$\text{즉, } W(n) = 0.5 + (1-0.5) \cos\left(\frac{2\pi n}{N}\right) \quad (1)$$

둘째, 신호의 임계 주파수 분석을 하기 위하여 FFT

를 통해 신호의 스펙트럼 $Re(w)$, $Im(w)$ 를 얻은 후 식(3)에 의해 power spectrum을 구한다.

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk} ; k = 0, 1, 2, \dots, N-1 \quad (2)$$

$$\text{여기서 } W_N = e^{-j\frac{2\pi}{N}}$$

$x(n)$ 은 window를 취한 입력신호:

$$P(w) = Re^2(w) + Im^2 X(w) \quad (3)$$

$$B_i = \sum_{n=bn_i}^{bn_{i+1}} P(w) \quad (4)$$

i : 임계 대역 순서

bn_i : 임계 대역 i 의 고역주파수 경계

bl_i : 임계 대역 i 의 저역주파수 경계

B_i : 임계 대역 i 의 Energy

이 과정에서 임계 대역은 중이의 등자매와 달팽이관을 연결하는 전정창으로부터의 거리로 나타나며 보통 24~26개로 나누어지나, 본 논문에서 임계 대역의 선택은 general MIDI에서 사용하는 악기음의 주파수 특성을 분석한 결과 95% 이상의 악기주파수 분포가 그림 1과 같이 21Hz~12KHz의 대역에 걸쳐 분포하고 있음을 보였다. 이러한 근거에 기초하여 표 1과 같이 임계대역을 23개 대역으로 정하였다. 임계 대역은 귀의 특성에 따라 낮은 주파수에서 세밀하고 높은 주파수로 갈수록 넓어지는 로그함수적인 특성을 갖는다.

셋째, 임계 대역 주파수 스펙트럼 분석과 임의의 주파수 대역에 큰 에너지가 있을 때 주변의 작은 에너지 성분을 가진 주파수 성분은 귀에 들리지 않는 Masking원리를 이용하여 이를 filter bank역할을 하는 기저막의 Spreading Function에 중첩함으로써 Spreading 임계 대역 스펙트럼을 구한다. 이는 실제로 귀에서 주파수를 인식할 때 임의의 주파수 대역이 그 대역에 해당하는 청신경만을 자극하는 것이 아니라 주위의 신경까지 자극하게 되어 주파수의 마스킹 현상이 일어난다. 이때 Spreading Function은 주위의 임의의 주파수 대역의 소리가 주변 대역에 어떤 영향을 미치는지를 정량적으로 나타내준다^[7].

$$10 \log s_{ij} = 15.81 + 7.5(i-j + 0.474)$$

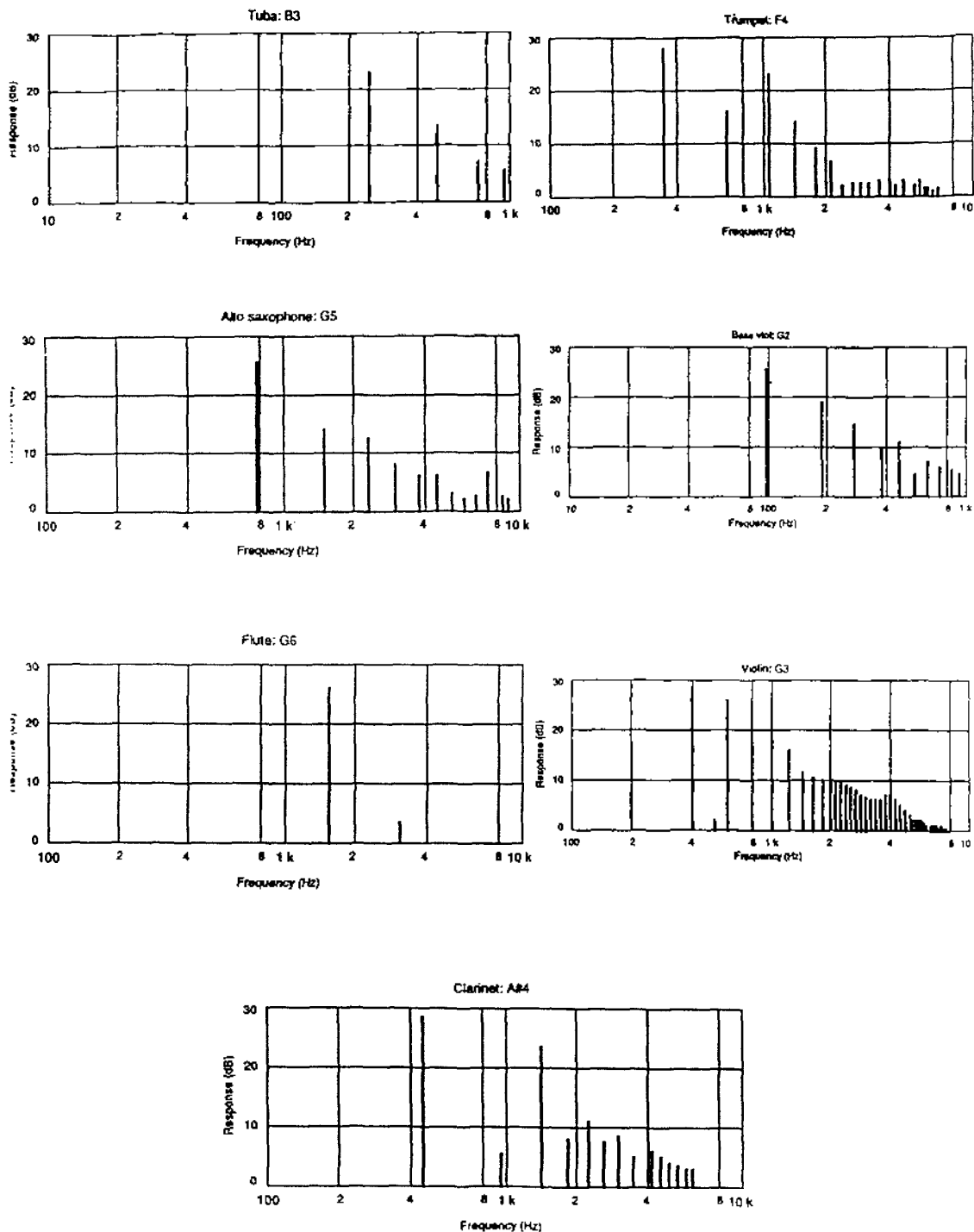


그림 1. 몇가지 미디 악기음별 주파수 성분도.

Fig 1. Frequency characteristics for some MIDI musical instrument.

표 1. 임계 대역

Table 1. Critical Band.

대역순서	하한 주파수	중앙 주파수	상한 주파수
1	0	50	100
2	100	150	200
3	200	250	300
4	300	350	400
5	400	450	510
6	510	570	630
7	630	700	770
8	770	840	920
9	920	1000	1080
10	1080	1170	1270
11	1270	1370	1480
12	1480	1600	1720
13	1720	1850	2000
14	2000	2150	2320
15	2320	2500	2700
16	2700	2900	3150
17	3150	3400	3700
18	3700	4000	4400
19	4400	4800	5300
20	5300	5800	6400
21	6400	7000	7700
22	7700	8500	9500
23	9500	10500	12000

$$-17.5(1+(i-j-0.474)^2)^{\frac{1}{2}} \text{ dB} \quad (5)$$

여기서 i 는 마스크 되는 주파수 대역 순서
 j 는 마스크 시키는 주파수 대역 순서
 이때 중첩 Spread 임계 대역 스펙트럼은

$$C_i = S_{ij} [] \times B_i [] \quad (6)$$

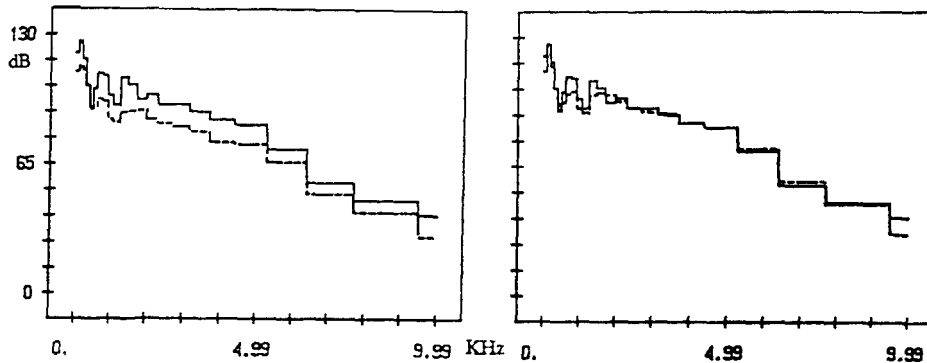
로 표시된다. 이는 각 대역간의 마스크 영향을 나타내주기도 한다.

넷째, 잡음 Masking threshold 추출
 Masking energy에 대한 Offset O_i 를 구해보면

$$O_i = \alpha \times (14.5 + i) + (1 - \alpha) \times 5.5 \quad (7)$$

여기서 α 는 Spectral Flatness Measure

$$SMF_{dB} = 10 \log \frac{N \sqrt{\prod_0^N C_i}}{\sum_0^N C_i} \quad \text{에서} \quad (8)$$



(a) Piano Sound A#3 음에 대한 비교치
 — Noise Masking Threshold
 - - - Critical Band Noise Spectrum

그림 2. Adaptive Table Coefficient 선택도 a) 적절한 계수 선택 b) 부적절한 계수 선택
 Fig 2. Decision Diagram of Adaptive Coefficient a) Correct Selection of Coefficient
 b) Incorrect Selection of Coefficient

$$\alpha = \min\left(\frac{SMF_{dB}}{SMF_{dBmax}}, 1.0\right) \quad (9)$$

에 의해 구해진다.

이때 Spread threshold Estimate $T[i]$ 는

$$T[i] = 10^{\log r_i - \frac{G_i}{10}} \quad (10)$$

으로 구해지며 최종적인 Mask Threshold는

$$div = \sum_{j=1}^M S[i][j] \times \frac{B[j]}{B[i]} \quad (11)$$

$$Th[i] = 10 \log \left[\frac{T[i]}{div} \right] \quad (12)$$

따라서 예측값을 구하기 위한 계수 즉, 디코더 블럭에 있는 계수 h_1, h_2 를 Masking Threshold를 이용하여 Adaptive하게 스케일링 한다. 즉, Decoder를 통과한 출력과 원음과의 양자화 잡음을 구하여 양자화 잡음의 $Th[i]$ 와 원음의 $Th[i]$ 를 비교하여 양자화 잡음의 Threshold가 원음의 Threshold보다 아래에 있게 되는 레벨을 Adaptive한 계수 h_1, h_2 로 선택한다. 이의 비교치를 그림 2에 나타냈으며 (a)는 계수를 적절히 선택한 결과도이고 (b)는 계수를 부적절하게 선택한 결과도이다.

III. 악음의 Digital Decoder

실제의 시스템에서는 코딩 부분은 1주기의 데이터를 감추시킴에만 필요하고 소리를 합성할때는 디코더 부분만 필요하며 그림 3의 디코더 부분에서는 악기의 건반중에서 어떤 건반이 어떤 속도로 두드려졌는가에 관한 정보가 이 부분에 입력되면 해당 건반의 1주기 기본 파형을 만들어서 인접한 데이터를 이용하여 빈 곳의 데이터를 찾아내는 인터플레이션(Interpolation) Block과 공동으로 사용하는 RAM으로 된 공유 메모리(Shared Memory)에 저장한다.

키-on 시에는 그 건반을 할당한 채널 넘버가 VMC (Voice Module Controller)로부터 들어 오게 되며 그 키의 강·중·약, 키의 어택·디케이·서스테인·릴리즈 (Attack·Decay·Sustain·Release)에 따른 디코더 내의 PROM 초기 어드레스(Initial Address)와 예측값 (Predictor)의 계수가 들어있는 어드레스(address), 그리고 샘플수와 32개 채널중 어떠한 채널로 출력시킬 것인가를 공유 메모리에 적고 인터럽트를 건다.

한편 현재 채널이 사용중인가를 나타내는 CHEN이 저 레벨(Low Level)인가를 검사하여 샘플수, h_1, h_2 를 읽어 들이고 ROM 어드레스와 채널 넘버를 카운터에 로딩한다. 다음 CHEN을 하이상태(High Level)로 만든 후에 디코딩 루프를 샘플 수 만큼 수행한다.

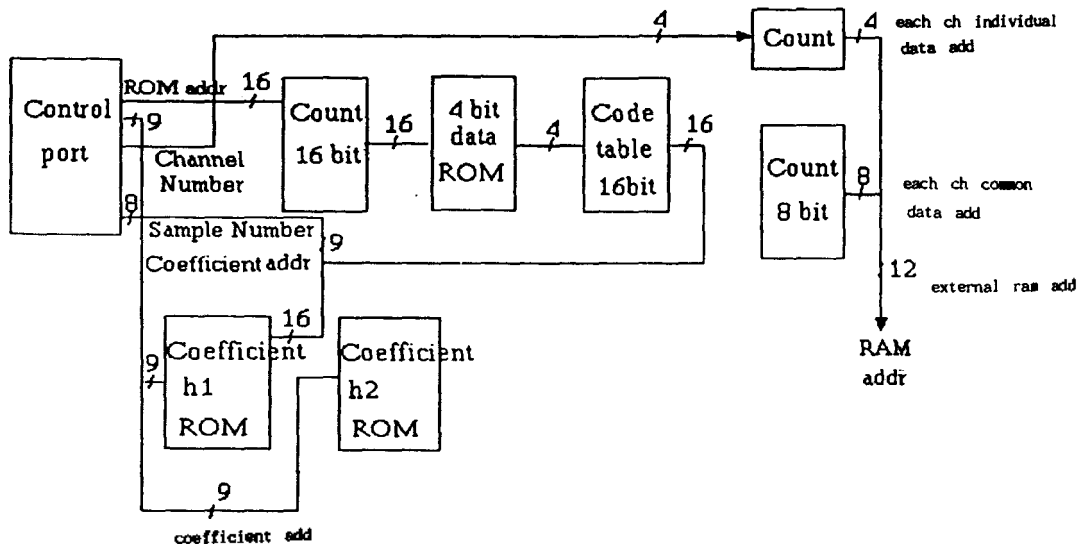
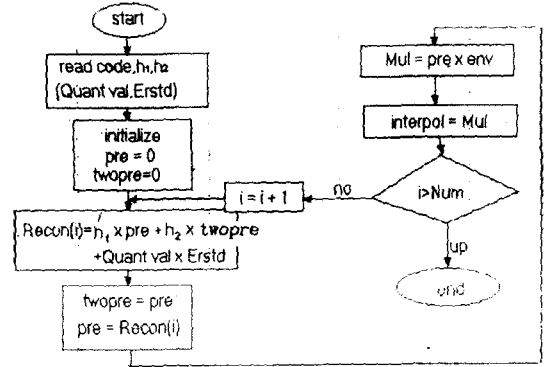


그림 3. 디코더 블럭도
Fig 3. Decoder Block Diagram

여기서 CHEN은 지남 선택한 채널이 사용 가능한지 여부를 나타내주며 CHEN이 high상태로 되어 있으면 저 레벨로 될때까지 계속 검사한다. 디코딩 루프 중에서도 인터플레이션 부분과 상충으로 사용하는 공유 메모리에 데이터를 쓸 경우(Write)에는 인터플레이터가 액세스(Access)하는가를 조사하여 만약 사용할때는 1사이클 기다렸다가 인터플레이터 부분에서 인터럽트 신호를 받음과 동시에 데이터를 쓰면서 인터럽트를 제거하도록 설계하였다.

그림 4는 1개 채널의 디코더를 하드웨어적인 관점에서 나타내었다. 그림 5는 그림 4의 계통도를 시간 순서에 대해 나타내었다. 여기서 보통 소수점 계산은 소수점 위치가 MSB에서 두번째 점인 Q14 format 곱셈기를 이용하여 곱셈기의 최로 부담을 경감시켰으며 그림 4의 엔벨로프는 균일 전력 밀도 방식에 의해 8 비트만을 사용하여 8*16 곱셈기만의 사용으로 회로를 간략화 하는데 크게 기여하였다. 즉 샘플링 데이터가 16비트이면 이의 엔벨로프도 16비트가 되어야 하나 본 알고리즘에 의하여 16*32개의 레지스터

가 1/2로 되고 곱셈기도 16*16이 8*16으로 줄어들어 그만큼 wafer의 사이즈를 줄일 수 있다.



$pre = x(n-1)$ data, $twopre = x(n-2)$ data

그림 5. 디코딩 순서도
Fig 5. Decoding Flow Chart

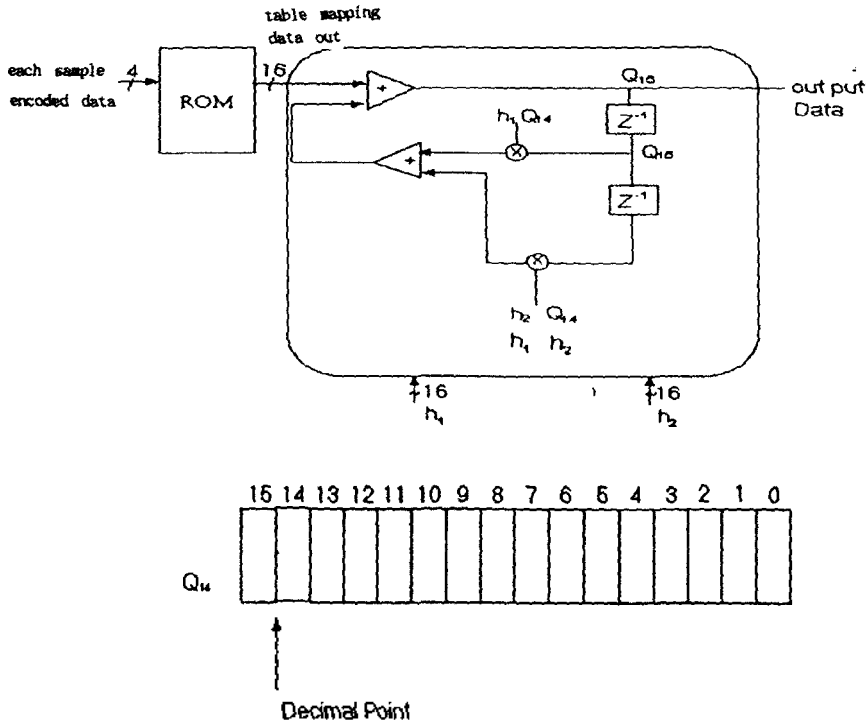


그림 4. 채널 디코더
Fig 4. Channel Decoder

IV. 균일 전력 밀도 엔벨로프 합성법

어떤 물체든 두드리거나 핑기면 자연발생음을 낸다. 음발생 원인에는 여러 형태를 가지나 이는 에너지의 연속적인 공급을 원하며 이를 위하여 악기에서는 두가지 방법을 이용한다. 즉 (1) 현을 활로 켜는 과정에서 소모되는 일, (2) 좁은 구멍을 통하여 일정한 기류를 예리한 날에 입사 시키거나 또는 유연한 lead나 입술을 지나도록 할 때 수행되는 일이며 반일 전기적인 발진기를 이용한다면 연속음을 내는 세번째 방법은 전기적인 에너지의 공급이다. 따라서 이러한 여러 종류의 음발생을 관찰해보면 실질적인 Audio Signal Processing 중 음성 Data와는 달리 악기에서는 규칙적인 음의 진행을 관찰할 수 있으며 이것이 바로 Keyboard 등에서 응용하는 Auto Looping이다.

본 논문에서는 균일 전력 밀도 엔벨로프 발생을 위하여 새로운 looping 방식인 Partition Crossfading Loop 방식을 이용하여 원음을 충실히 재생하였다. Sampling된 audio signal data를 $x(n)$ 이라 하고 loop start data를 $x(s)$, loop end data를 $x(e)$ 라 하면

$$Dif[s, e] = \sum |x(s+i) - x(e+i)|, \quad -\frac{L}{2} < i < \frac{L}{2} \quad (13)$$

여기서 L 은 Window Length, 즉 Partition Length이다.

이와 같이 구한 Dif를 여러 sample 간격으로 구해서 최소값이 나온 곳을 loop로 잡으며 이를 다시 Partition Crossfading하여 loop차를 최소화시킨다. 즉

$$x(s+i)' = \frac{[B(n+i) \times (L-i) + A(n+i) \times i]}{L} \quad (14)$$

$$x(e+i)' = \frac{[B(L-1) \times (L-i+1) + A(L-1) \times (L-1)]}{n} \quad (15)$$

여기서 $x(s+i)'$ 와 $x(e+i)'$ 는 crossfading에 의해서 새로이 생성된 Data이며 이를 그림으로 나타내면 다음과 같다.

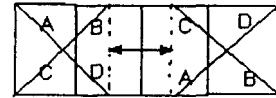
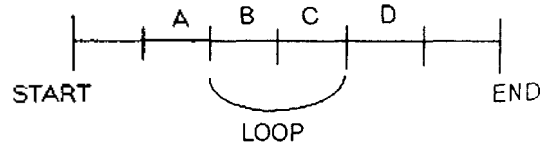


그림 6. 부분 상쇄 루프도
Fig 6. Partition Crossfading Loop Diagram

여기서 A, B, C, D는 시간축상의 연속적인 데이터의 임의의 블록이며 실제 loop는 A와 D, B와 C를 식 (14)에 의해 crossfading하여 구한다. 그림 7은 피아노의 38번째 음의 partition crossfading 하기전의 상태를 보인 것이고 그림 8은 partition crossfading 후의 상태를 보인 것이다. 이 과정은 주파수 성분이 비교적 random 하지 않기 때문에 이와같이 partition crossfading이 수월하다. 그림 9은 snare drum의 파형이고 이는 주파수 성분이 random하고 단발성으로 끝나기 때문에 crossfading이 파형의 끝부분에서 생겨 실제 데이터 축소는 많이 일어나지 않는다.

이와 같은 Partition Crossfading Loop에 의해서 생성된 Data를 b 라 하고 원래 Sampling에 의해 얻은 Data를 a 라 하면 균일 전력 밀도를 이용하여 아래와 같이 envelope를 찾기 위하여 Scale Factor를 구한다.

$$A = 10 \log \sum (a^2) \quad : \text{output power} \quad (16)$$

$$B = 10 \log \sum (b^2) + env \quad : \text{crossfade power} + env. \quad (17)$$

env 는 원래 샘플링된 신호의 한주기별 전력을 의미하며 crossfade된 부분과 원신호의 전력을 같게 하기 위하여 임의의 상수 b' 을 도입하면

$$b' = b \times scale \quad (18)$$

$$b'^2 = b^2 \times (scale)^2 \quad (19)$$

이를 각각의 샘플에 대해 합해 보면

$$\sum(b'^2) = scale^2 \times \sum(b^2) \quad (20)$$

이 되고 전력을 같게 놓으면 식(21)이 얻어진다.

$$10 \log \sum(b'^2) + env = A \quad (21)$$

인 scale을 구한다.

$$df = opow - npow = 10 \log \frac{\sum(a^2)}{\sum(b^2)} \quad (22)$$

여기서 df는 각 주기별 전력을 균일하게 만든 정규화된 데이터의 전력과 원신호의 전력차를 나타낸다. opow는 output power를, npow는 normalized power를 의미한다.

$$10 \log \sum(b'^2) = 10 \log (scale^2 * \sum(b^2)) \quad (23)$$

$$= 10 \log scale^2 + 10 \log \sum(b^2) \dots \quad (24)$$

식(21)로부터

$$10 \log \sum(b'^2) = A - env = 10 \log \sum(a^2) - env \dots \quad (25)$$

(24) = (25) 이므로

$$10 \log scale^2 + 10 \log \sum(b^2) = 10 \log \sum(a^2) - env \quad (26)$$

$$10 \log scale^2 = 10 \log \frac{\sum(a^2)}{\sum(b^2)} - env \quad (27)$$

식(22)를 식(27)에 대입하면

$$(27) = df - env \quad (28)$$

$$scale^2 = 10^{\frac{df - env}{10}} \quad (29)$$

$$\therefore scale = 10^{\frac{df - env}{20}} \quad (30)$$

따라서 이와 같이 구해진 scale factor를 partition crossfading loop 방식에 의해 생성된 데이터와 곱하면 loop 시작과 끝에서 생기는 distortion의 감쇄로 인하여 원래의 샘플링 신호와 음질이 동일한 신호를 얻을 수 있다.

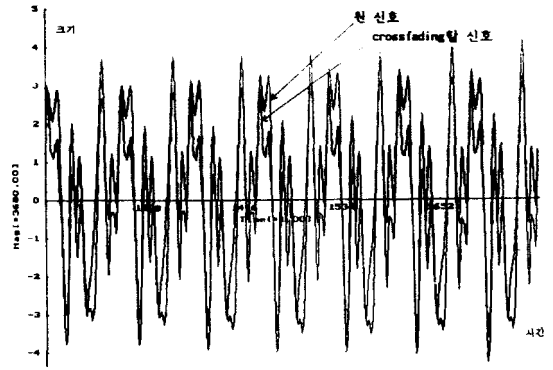


그림 7. partition crossfading하기 전의 피아노 38번째 음
Fig 7. 38 key waveform of piano before partition crossfading.

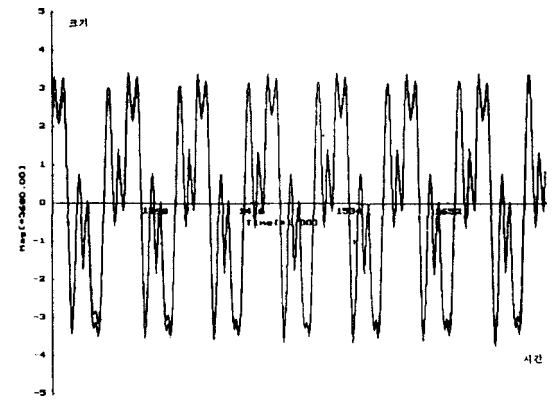


그림 8. partition crossfading후의 피아노 38번째 음.
Fig 8. 38 key waveform of piano after partition crossfading.

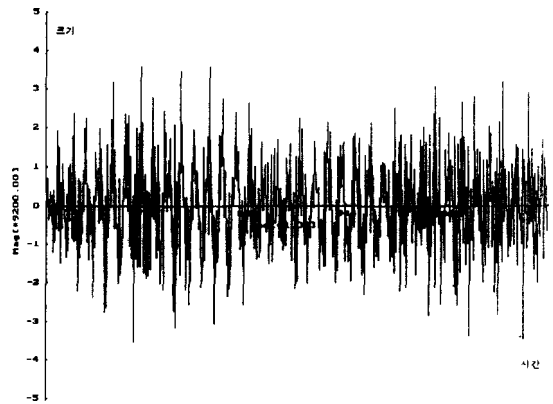


그림 9. 스네어 드럼 파형
Fig 9. waveform of snare drum.

V. COMPUTER SIMULATION 및 결과

본 연구 결과는 sound IC에서 출력되는 data와 원래의 sampling data와의 관계를 규명하여 어떠한 악기음 data라도 정확한 sampling data만 있으면 본 논문에서 구현한 algorithm으로 재생가능함을 보여준다.

그림 10은 디코더 출력에 균일 전력 밀도의 엔벨롭이 더해지고 여기에 음 효과중의 하나인 모듈레이션 정보가 합산해져 최종 출력이 나오는 처리 과정을 보여주는 Mixing단의 Block Diagram이며 그림 11은 본 알고리즘의 실제 구현을 위한 Simulation Timing Diagram으로 그림 3에서 본 control port의 CHEN이 enable 되어 메모리에 저장된 음원 정보가 읽혀져 그림 10의 처리 과정을 거쳐 최종 D/A converter로 나가는 INS[15:0]가 출력됨을 의미한다, 즉 그림 11의 INS[15:0]은 그림 10의 output과 동일하며 여기서 vel, vol, mod는 MIDI(musical instrument digital interface) 표준규격에서 정의된 effect에 관한 제어정보를 의미한다.

따라서 위와 같은 flow에 의해서 새로이 합성된 음파 원음과의 Waveform의 비교치를 그림 12에 나타내었다.

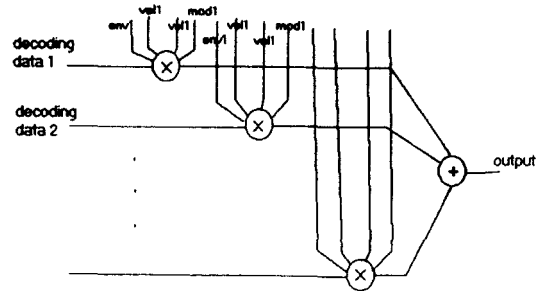


그림 10. 합성단
Fig 10. Mixing Block Diagram

그림 13은 균일 전력 밀도 방식에 의해 만들어진 envelope이며 Partition Crossfade Loop방법을 이용하여 Loop data를 만든 후 envelope와의 합성에 의해서 최종적으로 나온 출력을 같이 보여준 것이다. 따라서 합성출력의 envelope와 원래음의 envelope가 거의 유사함을 알 수 있다. 또한 이를 정량적으로 고찰하기 위하여 128 샘플을 한 구간으로 하여 구간별 신호대 잡음비를 식 (31)과 같이 구하였다. 여기서 S는 segment의 갯수이고 본 논문에서는 512 point의 sample을 하나의 segment로 보았다. ins(n)은 그림

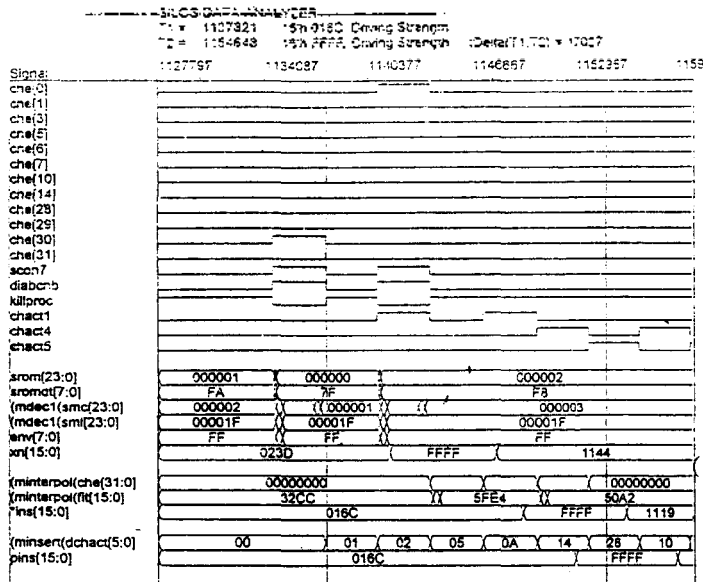


그림 11. 각 신호의 시간별 순서도
Fig 11. Timing Diagram of Each Signal

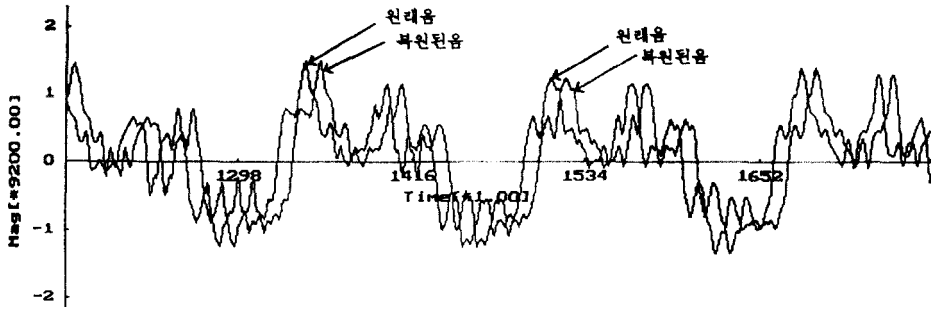


그림 12. 원래 음(앞쪽)과 디코딩에 의해 합성된 음(뒷 부분)
 Fig 12. Original Sound and Synthesis Sound Waveform

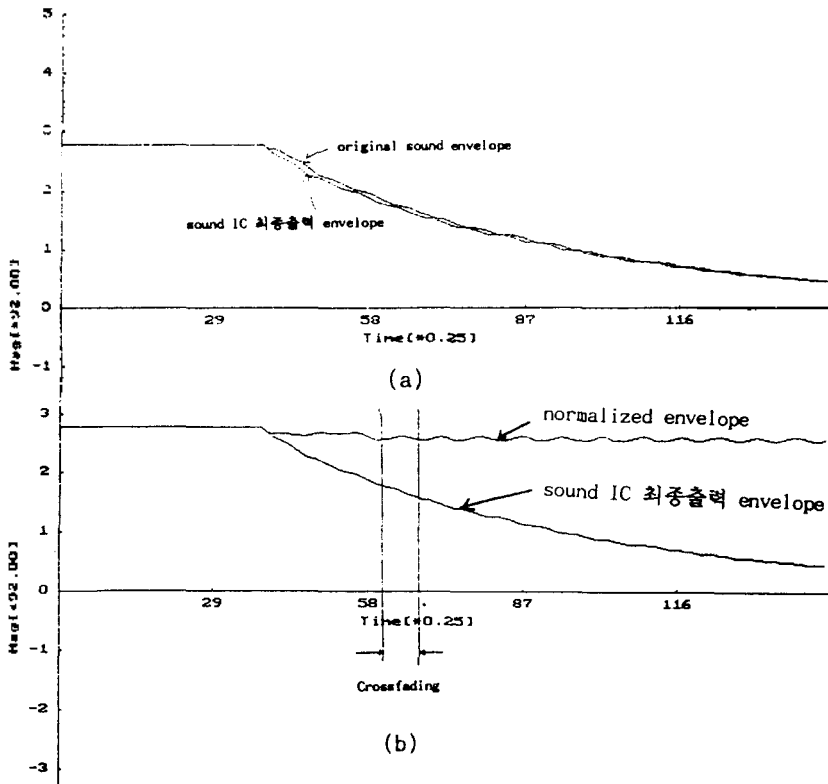


그림 13. a) 원유의 엔벨로프 파형과 균일 전력 밀도를 이용한 Sound IC의 최종 엔벨로프
 b) Partition Crossfading Loop 방식에 의해 구해진 Normalized Waveform의 엔벨로프 파형과 Sound IC의 최종 엔벨로프

Fig 13. a) Envelope of Original Waveform and Output Envelope of Sound IC made by Equal Energy Density
 b) Envelope of Normalized Waveform made by Partition Crossing Loop and Envelope of Sound IC Output

10에서의 최종 출력을 의미하며 $x(n)$ 은 샘플된 data 이다.

$$SNR_{avg} = \frac{1}{S} \frac{\sum_{n=0}^{128} x(n)^2}{\sum_{n=0}^{128} (x(n) - ins(n))^2} \quad (31)$$

이때 SNR이 25dB 이상이면 거의 구별하기 힘들었으나 본 연구에서는 30dB 이상으로 이를 실제 청각 Test를 하여 들어보면 구별할 수 없을 정도로 동일하였다.

다음에는 정보량 축소를 보이기 위해 악기중에서 가장 많이 사용하는 피아노의 파형과 가장 정보의 변화가 심한 snare drum의 데이터를 비교해 본다. 그림 7의 피아노 파형은 약 25초 가량 지속된다. 만일 이를 메모리에 모두 샘플링하여 저장한다면 16비트 44.1 KHz 샘플링시 $44100 * 2 * 25$ 바이트가 필요하다. 그러나 본 알고리즘에 의한 방식을 사용하면 ADPCM

에 의해 1/4이 절약되고 1.5초 까지의 데이터로 이를 loop의 시작과 끝부분을 partition crossfading시켜 이를 Auto looping한 후 scale factor에 의해 구해진 균일 전력 밀도 엔벨로프와 곱에 의하여 원음을 재생해 냄으로써 약 50:1까지의 정보량 축소가 가능하다. 단, 그림 9와 같이 단발성으로 끝나는 악기음은 partition crossfading에 의한 정보량 축소가 거의 끝부분에서 이루어지기 때문에 약 4:1까지의 정보량을 줄일 수 있다.

그림 14는 이와같은 알고리즘에 의해 구성된 음원 전용 DSP의 block diagram을 보였다. 여기서는 wave generator 부분에서 모든 기능이 주어지고 외부 Media와의 통신에는 단순한 8 Bit processor 만으로도 제어가 가능하도록 설계되었다.

그림 15는 MIDI에서 정의된 PAN, VOLUME, POTAMENTO, MODULATION, PITCH BENDER, AFTER TOUCH 등의 명령 체계로 구현된 RISC processor를 이용한 회로 diagram을 그렸으며

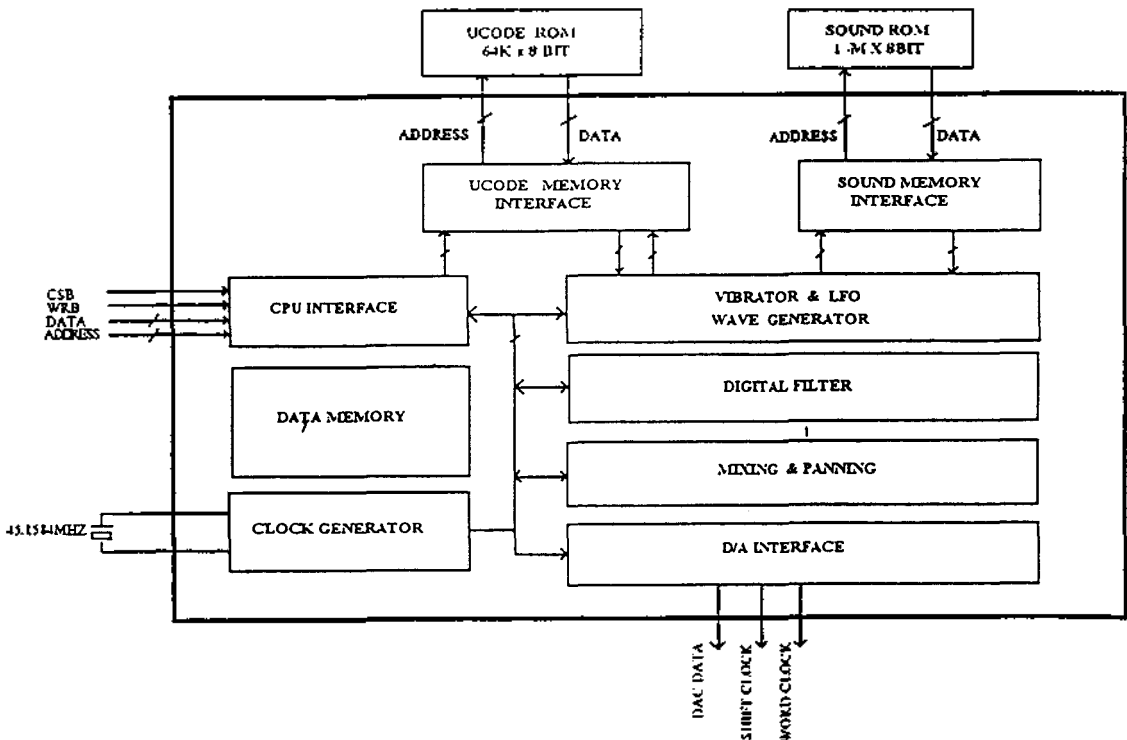


그림 14. 음원 전용 DSP 블록 도
Fig 14. Block Diagram of sound DSP

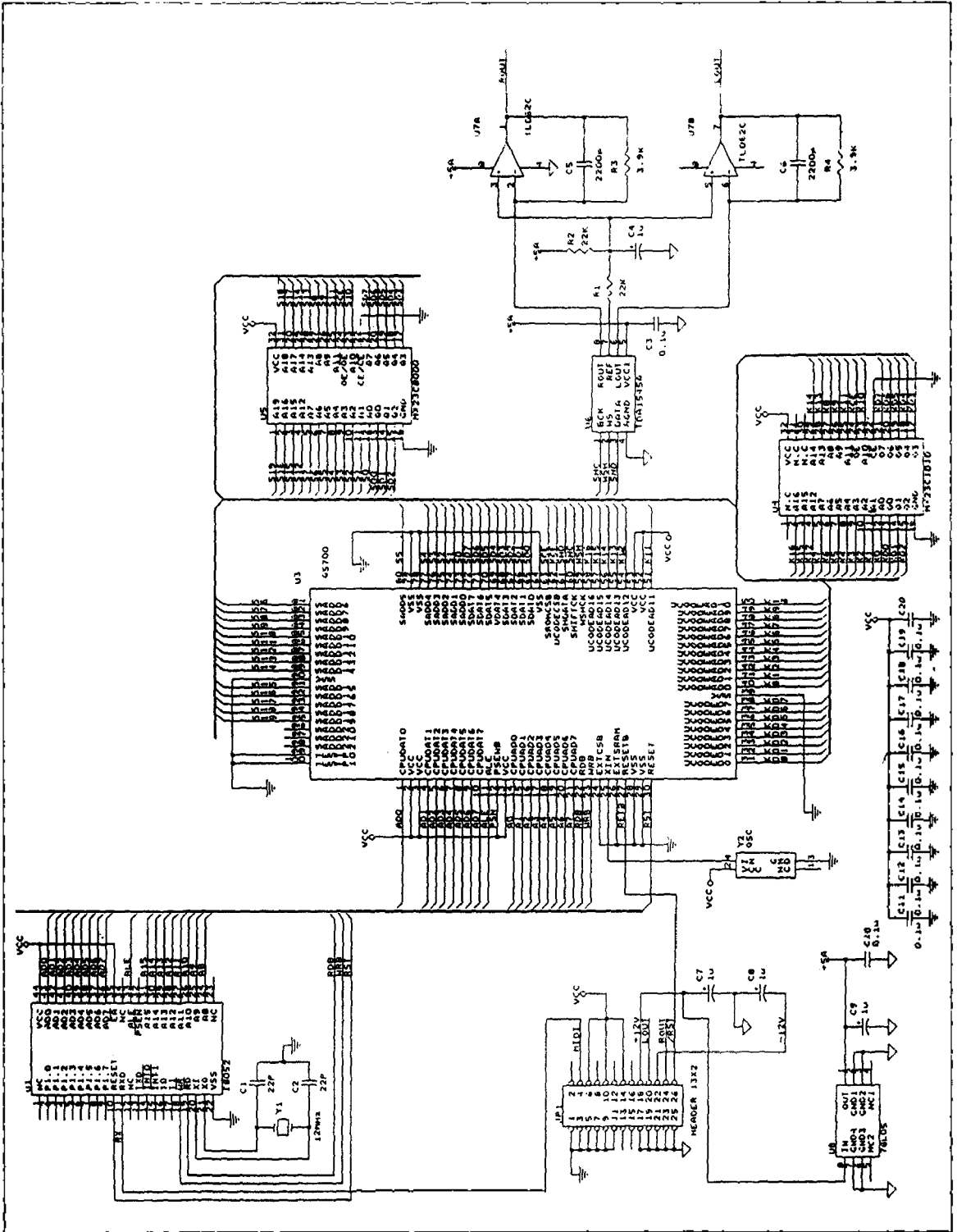


그림 15. 실험 회로 부리보드

Fig 15. test circuit diagram.

이의 실제동작의 입증을 위해 악기간의 통신규격인 MIDI 신호를 이용하여 실험하였다. 따라서 이러한 과정을 거쳐 최종적으로 코딩한 정보의 크기를 표 2에 나타내었으며 각 번호는 MIDI에서 말하는 프로그램 번호를 의미하고 크기는 partition crossfade loop 방식에 의해 구해서 memory에 저장된 악기음의 auto loop size이다.

VI. 결 론

본 논문에서 제작한 음원 Algorithm은 전자 음악 합성기로서 전기적인 방법에 의한 악기음의 재생 방법으로 다음과 같은 결론을 얻었다. 즉, 충실한 악기음의 샘플링 데이터만 있으면 거의 완벽한 원음을 재생할 수 있고 적당한 샘플링 기구만 있으면 가청주파

표 2. 부분 크로스 페이딩에 의한 각 악기별 루프 길이.

Table 2. loop length of each musical instrument by partition crossfading.

키 이름	길 이	키 이름	길 이	키 이름	길 이
11-C5	R 379	MI-TOM1	R 336	125	R 2328
105-C3	R 337	NO	R 2	126	R 4593
106-C3	R 253	O-HI-HAT	R 7916	127	R 11194
108-C3	R 253	O-TRIAN	R 722	13-C3	R 28
57-C2	R 169	OP-SURDO	R 351	13-C4	R 21
57-C3	R 84	RI-SYMB2	R 11696	15-F3	R 1311
57-C4	R 84	RID-BELL	R 1504	15-F4	R 1304
58-C3-22	R 169	S-H-WHIS	R 205	16-F3	R 95
57-C5	R 84	SPLA-SYM	R 8128	16-F4	R 79
61-F5	R 95	VIB-SLAP	R 500	74-C5	R 83
10-C4	R 253	77-C3	R 450	74-F2	R 63
61-F3	R 127	77-C4	R 505	49-F5	R 9358
61-F4	R 95	77-C5	R 337	23-F3	R 253
61-F2	R 84	80-F2	R 84	23-F4	R 284
110-C3	R 225	80-F4	R 95	23-F5	R 63
10-C5	R 296	80-F3	R 127	24-E3	R 2679
49-F4	R 19509	80-F5	R 95	24-G4	R 3564
112-C3	R 224	81-C3	R 9699	25-D4	R 75
113-C3	R 506	81-C5	R 3654	25-E3	R 67
29-3	R 63	82-C3	R 8237	25-G4	R 75
29-4	R 95	82-C5	R 2916	26-F3	R 379
30-3	R 85	89-C3	R 12785	26-F4	R 284
30-4	R 84	46-C4	R 21	26-F5	R 126
74-C4	R 169	47-F3	R 158	28-F3	R 63
60-C4	R 337	47-F4	R 142	28-F4	R 284
41-3	R 84	48-F3	R 2404	28-F5	R 63
41-4	R 84	5-F2	R 253	31-D3	R 50
41-5	R 63	5-F3	R 337	31-E4	R 156
44-3	R 84	5-F4	R 142	33-C2	R 169
44-4	R 84	57-A2	R 101	33-C3	R 232
44-5	R 84	60-C3	R 422	34-C2	R 338
65-C2	R 170	27-F3	R 126	34-C3	R 211
65-C4	R 84	27-F4	R 126	34-C4	R 253
65-C5	R 84	60-C5	R 21	35-C2	R 169
66-3	R 84	8-E3	R 101	35-C3	R 225
66-4	R 84	8-E4	R 111	35-C4	R 281
66-5	R 84	83-C4	R 7376	36-C2	R 169
68-F2	R 84	9-C3	R 169	36-C3	R 232
68-C3	R 84	9-C4	R 309	37-C3	R 253
68-C4	R 84	9-C5	R 63	37-C4	R 295
67-3	R 84	BELL-TRE	R 1980	39-C3	R 225
67-4	R 84	CHIN-SYM	R 3360	39-C4	R 28
67-5	R 84	CRA-SYM1	R 14562	42-F3	R 42
68-C5	R 84	JINGLE	R 1180	46-C3	R 42

수(20Hz~20000Hz)내의 거의 모든 원음을 샘플링하여 저장 가능하므로 모든 악기의 파형을 디지털화 할 수 있다. 그리고 또한 이미 샘플링된 데이터가 존재하는 경우 적당한 시그널 프로세싱 과정을 통하여 다양한 종류의 파형 변화가 가능하므로 실질적인 음의 합성 또는 창조라 할 수 있는 새로운 효과 음향을 만들 수 있으며 최대 50:1 까지 최소 4:1 까지의 데이터 축소가 가능하다.

이러한 결과는 궁극적으로 Multimedia 및 전자 악기용으로 악기음을 합성하는데 있어 막대한 양의 데이터를 감소시키는 물론 고속의 메모리가 아닌 범용 메모리로서도 실시간으로 저렴하게 시스템을 구현할 수 있음을 보여준다.

참 고 문 헌

1. D.F.Steve, *THE SECRETE OF ANALOGE AND DIGITAL SYNTHESIS*. Third Earth Production, NJ, 1986.
2. D.F.Steve. *THE MIDI BOOK*. Third Earth Production, NJ, 1986.
3. Haward Massey, Alex Noyes, And Daniel Shklair, *ASYNTHETIC'S GUIDE TO ACOUSTIC*. Ameco Publication, NY, 1987.
4. D.R.Thomas, *The Science of Sound*. Addition, Wesley, 1990.
5. F.Catthoor, J.vansas, L.Inze, and H.De Man, "Testability Strategy for Multiprocessor Architecture," *IEEE Design and Test Computer*, pp.18-24, Apr. 1989.
6. N.S.JAYANT, PETER NOLL, *Digital Coding of Waveform*, Prentice-Hall, INC, 1984.
7. J.Johnston, "Transform coding of audio signals using perceptual noise criteria," *IEEE J. Selected Areas Comm*. pp.314-323, 1988.

▲김 종 한(Jong Han Kim)

현재 : 삼성전자 통신연구소 수석연구원

▲김 원 후(Won Hoo Kim)

현재 : 한국항공대학교 항공전자공학과 교수

▲구 재 울(Jae Ul Koo)

정회원

1983년 2월 : 한국항공대학교 통신공학과 졸업
(공학사)

1986년 3월 : 공군장교 예편

1986년~1987년 : 큐닉스 컴퓨터 연구소

1988년 2월 : 연세대학교 대학원 전자과 졸업(공학 석사)

1987년~1990년 : 한국과학기술원 전자음악 합성기술에 관한 연구 참여

1990년~1993년 : 삼익악기 연구소

1991년~1993년 : 미국 Admos사, 프랑스 DREAM사

현재 : 한국항공대학 박사과정 중
(주)H.W.A Sound Source

※주관심분야 : Digital Audio singal Processing 및 멀티미디어 음원 전용 Processor설계

▲방 효 창(Hyo Chang Pang)

정회원

현재 : 한국항공대학교 대학원
항공전자공학과 박사과정