

論 文

벡터 승산 구조를 이용한 다중채널
FIR 디지털 필터구성

正會員 林 永 道* 正會員 金 明 起**

Multi-Channel FIR Digital Filter Hardware
Implementation Using Vector Multiplication
Structure

Young Do LIM* and Myung Ki KIM**, Regular Members.

要 約 벡터 승산 구조를 이용하여 다중채널 FIR 디지털 필터를 구성하기 위한 한 기법을 제안하였다. 제안된 기법은 하드웨어의 구성을 간소화시키고, 동작속도를 개선할 수 있었다. 위의 기법으로 구성된 4 채널 디지털 필터의 주파수 응답은 Remez 방식으로 시뮬레이션하여 얻어진 주파수 응답과 잘 일치하였다.

ABSTRACT A new method on the hardware implementation of multi-channel Finite Impulse Response(FIR) digital filter using vector multiplication structure is proposed. The proposed method can reduce the complexity of hardware structure and improve execution speed. The frequency response of four channel digital filter implemented by the above method is quite agreeable with the frequency response simulated by Remez method.

1. 서 론

1973년 Croisier⁽¹⁾가 벡터 승산 구조를 이용하여 디지털 필터를 처음 구성하였고, 그 후 Peled와 Liu^{(2), (3)}를 비롯하여 필터 구성에 대한 많은 논문들이 발표되었다^{(4)~(7)}. 그리고 Chang-Fuu Chen⁽⁸⁾은 고정된 계수의 경우 비용절감과 동작

속도의 개선에 관점을 두고 FIR 디지털 필터를 구성한 바 있다. 오늘날 급격한 첨단산업의 발전으로 인해 디지털 통신, 음성, 영상, 그리고 산업 제어 시스템등을 구성할 때 다중채널 디지털 필터의 필요성이 크게 요구된다. 그런데 이러한 방법으로 다중채널 디지털 필터를 구성하려면 하드웨어 구성의 복잡성으로 인한 비용증대와 동작속도의 지연이란 문제점이 대두된다. 그래서 본 논문에서는 이러한 문제점들을 개선할 목적으로 다중채널 디지털 필터를 구성하는 한 기법을 제안하고 예시로서 4 채널 FIR 디지털 필터를 구성하고 실험을 통하여 제안한 기법의 타당성을 확인하였다.

* 東明專門大學電子學科
Dept. of Electronics, Dong Myung Junior College,
Pusan, 608 Korea.
** 東亞大學校電子工學科
Dept. of Electronic Engineering, Dong A University,
Pusan, 602-02 Korea.
論文番號 : 85-39 (接受 : 1985. 11. 9)

2. 벡터 승산 구조

(1) 기본원리와 구성

실시간 응답 FIR 디지털 필터의 차분 방정식 y_n 은

$$y_n = \sum_{k=0}^N a_k \cdot x_{n-k} \quad (1)$$

로 주어진다.

단, N 은 FIR 필터의 차수, a_k 는 계수이다. 여기서 모든 입력정보 및 계수의 최대값은 ± 1 로 한정하고, 부호비트를 갖는 2의 보수로 표현하면 입력정보 x_{n-k} 는

$$x_{n-k} = -x_{n-k}^o + \sum_{j=1}^{B-1} x_{n-k}^j \cdot 2^{-j} \quad (2)$$

로 된다.

단, x_{n-k}^o 은 부호비트(0 혹은 1), x_{n-k}^j 는 x_{n-k} 의 j 번째 비트(0 혹은 1)이다.

식(1)에 식(2)를 대입하면 식(1)은

$$y_n = \sum_{j=1}^{B-1} \left(\sum_{k=0}^N a_k \cdot x_{n-k}^j \right) \cdot 2^{-j} - \left(\sum_{k=0}^N a_k \cdot x_{n-k}^o \right) \quad (3)$$

으로 된다.

여기서 함수 ϕ_j 를 다음 식으로 정의한다.

$$\phi_j = \sum_{k=0}^N a_k \cdot x_{n-k}^j \quad (4)$$

그러면 식(3)은

$$y_n = \sum_{j=1}^{B-1} \phi_j \cdot 2^{-j} - \phi_o \quad (5)$$

로 주어진다.

만약 ϕ_j 를 안다면 y_n 은 $(B-2)$ 회의 덧셈과 1회의 뺄셈 및 $(B-1)$ 회의 우측 추이 연산을 수행하면 구해진다.

함수 ϕ_j 는 $N+1$ 개의 독립변수인 2진 벡터를 어드레스로 취하는 ROM으로 실현 가능하다. 그러므로 N 차 FIR 디지털 필터의 y_n 은 가산기, 감산기와 추이연산 가산기만으로 구성할 수 있고 이의 논리회로는 그림 1과 같다.

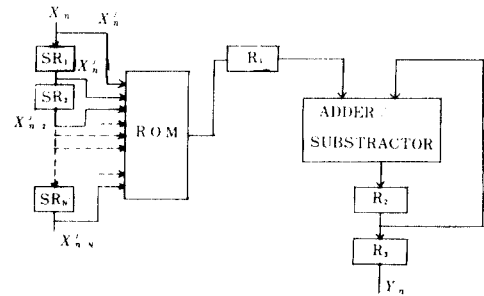


그림 1 N 차 FIR 디지털 필터의 논리회로
Logic circuit of N order FIR digital filter.

입력정보는 쉬프트 레지스터 1 (Shift register 1 : SR 1)에 직접 순차적으로 최하위 비트부터 1비트씩 들어온다.

각각 SR 1, SR 2, ..., SRN에서 나오는 벡터 $(x_n^j, x_{n-1}^j, x_{n-2}^j, \dots, x_{n-N}^j)$ 는 ROM의 입력에 가해져서 함수 ϕ_j 를 실현한다. 예로 2차 FIR 디지털 필터의 계수가 $a_0=0.0152474$, $a_1=0.0021682$, $a_2=0.0166436$ 일때 ROM에 프로그램된 함수 ϕ_j 의 값은 표 1과 같다.

표 1 2차 FIR 필터의 함수 ϕ_j
Function for second order FIR filter.

X_n^j	X_{n-1}^j	X_{n-2}^j	ϕ_j
0	0	0	0000000000000000
0	0	1	0000001000100000
0	1	0	0000000001000111
0	1	0	1111111000100110
1	0	1	0000000111110011
1	0	0	11111111111010010
1	1	1	0000001000111010
1	1	0	0000000000011001

필터 구성을 위해 필요한 가산기에서 두개의 오버랜드를 가산할 경우 오버플로(Overflow)가 발생하면 결과는 정확하지 않으므로 이러한 오

차를 교정하기 위해 오버플로 교정회로가 요구된다. 만일 A 와 D 를 가수와 피가수의 부호비트, B 와 C 를 교정하지 않은 합과 캐리 출력(Carry out)의 부호비트, S 는 교정된 합의 부호비트로 가정하면 오버플로 교정의 진리표는 표 2와 같다.

표 2 오버플로 교정의 진리표
Truth table of the overflow correction.

A	D	C	S
0	0	0	0
0	0	1	d
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	d
1	1	1	1

표 2에서 Karnaugh map에 의해 유도된 논리식 S 는

$$S = A \oplus C \oplus D \quad (6)$$

이고, 이의 논리회로는 그림 2와 같다.

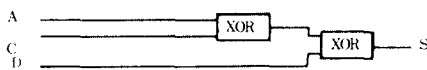


그림 2 오버플로의 교정회로
Circuit of the overflow corrector.

(2) 분산 ROM 처리

식(4)로부터 N 차 FIR 디지털 필터 구성을 위한 ϕ 를 구하려면 2^{N+1} 어(Word)를 가진 ROM이 필요하다. 그러므로 그림 1과 같은 회로로 고차 FIR 디지털 필터의 구성은 매우 고가인 ROM이 필요하므로 실용적이지 않다. 그래서 식(4)를 분산해 쓰면,

$$\phi = \sum_{k=0}^{m-1} a_k \cdot x_{n-k}^j + \sum_{k=m}^{m-1} a_k \cdot x_{n-k}^j + \sum_{k=m-1}^N a_k \cdot x_{n-k}^j \quad (7)$$

로 된다. 단, i 는 ROM뱅크의 수이다. 부가되는 $i-1$ 개의 가산기는 ROM용량을 감소시키며 함수 ϕ 를 구하는데 필요하다. 고속연산을 유지하도록 $i-1$ 개의 가산기를 그림 3과 같이 $(\log_2 i)$ 레벨로 배열한다. 여기서 (x) 는 x 와 같거나 또는 x 보다 큰 최소정수를 표현한다.

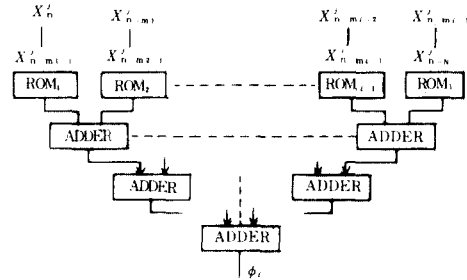


그림 3 메모리 뱅크와 멀티오퍼랜드 가산방법의 구성
Implementation with memory banks and multioperand addition method.

그림 3에서 총 전달지연시간 T 는

$$T = t_1 + t_2 ((\log_2 i) - 1) \quad (8)$$

로 주어진다. 단, t_1 은 한 레벨에서 가산기의 동작시간, t_2 는 가산기에서 합의 전달지연시간이다. 그리고 i 를 교정시켜 $N+1$ 을 i 로 나눈 나머지를 m 이라 하고, 어떤 두개의 ROM뱅크 사이에 어드레스 입력수의 차이가 1보다 크지 않게 하면 그림 3의 ROM최소용량은

$$2^{\lceil \frac{N+1}{i} \rceil} (m \cdot 2^1 + (i-m) \cdot 2^0) = 2^{\lceil \frac{N+1}{i} \rceil} \cdot (i+m) \quad (9)$$

로 주어진다. 단, $\lceil \frac{N+1}{i} \rceil = (Z)$ 는 Z 와 같거나, Z 보다 작은 최대정수를 표현한다. 그러나 어떤 두개의 ROM뱅크 사이에 어드레스 입력수의 차이가 1보다 크면 ROM용량은 식(9)보다 크게 된다. 만약 x_{n-k}^j 를 (여기서 $k=0, 1, 2, \dots, N$) i 그룹으로 나누고 각 그룹의 x_{n-k}^j 수의 차이가 1보다 크지 않으면 필요한 ROM은

$$\frac{2^{(N+1)}}{2^{\lfloor \frac{N+1}{2} \rfloor} (i+m)} = \frac{2^{(N+1)-\lfloor \frac{N+1}{2} \rfloor}}{(i+m)} \quad \text{[배]} \quad (10)$$

만큼 감소된다. 이와같이 ROM을 i 뱅크로 분산할 경우 출력에 대한 전체상대오차 $|E|$ 는

$$|E| \leq \frac{1}{2^{B-1}-1} \left(\sum_{j=(B-1)}^0 (i+1) \cdot 2^{j-1} \right) \cdot 2^{-B} < \frac{i+1}{2^{B-1}-1} \quad (11)$$

이다. 식(11)의 상대오차는 ROM뱅크의 증가에 따라 크게 되나 일반적으로 무시할 수 있다. 예로 $i=5$, $B=16$ 일때 상대오차는 0.0183%보다 작다.

(3) 동작속도를 개선하기 위한 멀티오

퍼랜드가산기에 가산기/감산기의 결합회로. 식(5)에서

$$\sum_{j=1}^{B-1} \phi_j \cdot 2^{-j} = A_{cc} \quad (12)$$

로 두면 y_n 은

$$y_n = \phi_0 + \frac{\dots}{A_{cc}} \quad (13)$$

으로 된다.

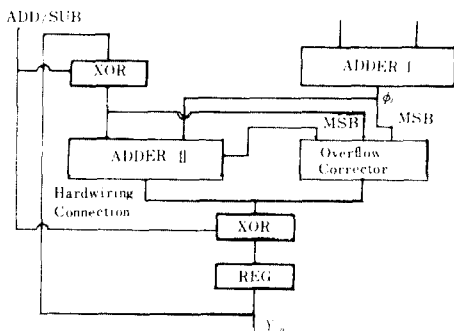


그림 4 멀티오퍼랜드 가산기와 가산기/감산기 결합 구성도. Block diagram of combining adder/subtractor and multioperand adder.

식(13)에 의해 구성된 그림 4는 가산기/감산기를 그림 3에 있는 멀티오퍼랜드 가산기에 논리적으로 결합시켜 동작속도를 개선한 것이다. 그림 4에서 가산기 1과 가산기 2를 하드와이어 연결(hardwire connection)을 행하여서 식(12)의 2^{-j} 연산을 실행할 수 있도록 하였다.

3. 다중채널 디지털 필터의 구성

(1) 입력정보의 기억을 위한 RAM이용.

고차 FIR 디지털 필터의 구성을 위해 벡터 승산 구조를 이용한 그림 1은 입력정보를 기억하는데 많은 쉬프트 레지스터가 필요하게 된다. 그러나 그림 5와 같이 RAM과 버퍼(Buffer)를 이용하면 많은 쉬프트 레지스터의 수를 감소시키면 반면 다중채널 디지털 필터의 실현이 가능하다. 그림 5는 RAM을 메모리 어드레스 레지스터(Mem Memory Address Register: MAR)에 의해 어드레스하고 여분 어드레스로 디코딩하여RAM의 영역을 입력채널의 수만큼 분산시킨다. 이때 두개의 버퍼를 RAM의 R/W신호로 동작시켜 입력정보 x_n 를 읽고 기억하여 메모리 버퍼 레지스터(Memory Buffer Register: MBR)에서 나오는 벡터 $\{x'_{n-1}, x'_{n-2}, x'_{n-3}, \dots, x'_{n-N}\}$ 를ROM 입력에 가한다. ROM도 역시 여분 어드레스로 입력 채널수와 같은 영역으로 분산시켜 각 채널에 해당하는 함수 ϕ 의 값을 얻도록 하였다. 이때 다중채널을 위한 ROM의 디코딩 어드레스는 표 3과 같다. ROM의 디코딩회로는 카운터와 조합 및 순서논리회로도 구성된다.

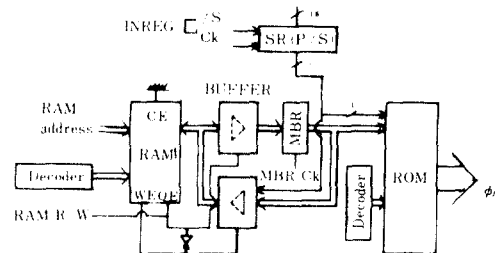


그림 5 쉬프트 레지스터를 대체하기 위한 RAM이용. Using a RAM to replace the shift registers.

표 3 ROM의 디코딩어드레스
Decoding address of ROM.

디코딩어드레스	해당채널	ϕ_i 어드레스
0 0	CH 1	0 0 0 0 0 0 0 ~ 0 0 1 1 1 1 1
0 1	CH 2	0 1 0 0 0 0 0 ~ 0 1 1 1 1 1 1
1 0	CH 3	1 0 0 0 0 0 0 ~ 1 0 1 1 1 1 1
1 1	CH 4	1 1 0 0 0 0 0 ~ 1 1 1 1 1 1 1

이때 최대 입력 채널의 수 (n)은

$$(n) = \left\lfloor \frac{T_s}{ADS_s} \right\rfloor \quad (14)$$

로 된다. 단, T_s 는 한 채널의 표본화 주기(sec), ADC_s 는 AD변환기의 변환시간(Sec)이다. 여기서 $(n) = \left\lfloor \frac{T_s}{ADC_s} \right\rfloor$ 는 n 보다 작은 최대정수의 값을 표현한다.

4. 필터의 구성

4 채널 FIR 30차 FIR 디지털 필터를 구성하기 위해 저역필터 2개(LPF1, LPF 2) 대역통과필터 2개(BPF 1, BPF 2)를 이용하였으며 이들 필터의 규격과 계수의 값은 Remez 방식으로 구하여 표 5와 표 6에 나타내었다. 주어진 계수들 2의 보수로 변환하여 각 채널의 ϕ_i 에 해당하는 값을 얻도록 6개의 뱅크로 분산시킨 ROM에 기억시켰다. 예시로 채널 1(LPF 1)에 있어서 6개의 ROM으로 분산된 ϕ_i 값중 ROM 1에 기억된 ϕ_i 의 내용을 표 4에 나타내었다.

표 4 채널 1(ROM 1)의 ϕ_i 내용
Contents of channel 1(ROM 1)

memory address	Contents
$x'_n \ x'_{n-1} \ x'_{n-2} \ x'_{n-3} \ x'_{n-4}$	ϕ_i
0 0 0 0 0	0 000 0000 0000 0000

0 0 0 0 1 1 111 1101 1101 1111
0 0 0 1 0 0 000 0000 0100 0111
0 0 1 1 1 1 111 1110 0010 0110
0 0 1 0 0 0 000 0001 1111 0011
0 0 1 0 1 1 111 1111 1101 0010
0 0 1 1 0 0 000 0010 0011 1010
0 0 0 1 1 0 000 0000 0001 1001
0 1 0 0 0 0 000 0010 0010 1010
0 1 0 0 1 0 000 0000 0000 1001
0 1 0 1 0 0 000 0010 0111 0001
0 1 0 1 1 0 000 0000 0101 0000
0 1 1 0 0 0 000 0100 0001 1101
0 1 1 0 1 0 000 0001 1111 1100
0 1 1 1 0 0 000 0100 0110 0100
0 1 1 1 1 0 000 0010 0100 0011
1 0 0 0 0 0 1 111 1111 0010 1111
1 0 0 0 1 1 1 111 1101 0000 1110
1 0 0 1 0 0 1 111 1111 0111 0110
1 0 0 1 1 1 1 111 1101 0101 0101
1 0 1 0 0 0 0 000 0001 0010 0010
1 0 1 0 1 1 1 111 1111 0000 0001
1 0 1 1 0 0 0 000 0001 0110 1001
1 0 1 1 1 1 1 111 1111 0100 1000
1 1 0 0 0 0 0 000 0001 0101 1001
1 1 0 0 1 1 1 111 1111 0011 1000
1 1 0 1 0 0 0 000 0001 1010 0000
1 1 0 1 1 1 1 111 1111 0111 1111
1 1 1 0 0 0 0 000 0011 0100 1100
1 1 1 0 1 1 0 000 0001 0010 1011
1 1 1 1 0 0 0 000 0011 1001 0011
1 1 1 1 1 1 0 000 0001 0111 0010

표 5 필터의 규격
Specifications of filters.

구분	통과대역차단주파수	감쇄대역차단주파수	차 수
	감쇄(dB)	감쇄(dB)	
LPF I	0.15	0.2	30 차
	-0.25	-30.9	
LPF II	0.2	0.25	"
	-0.21	-32.6	
	저역차단주파수	고역차단주파수	

구 분	감쇄영역	통과영역	통과영역	감쇄영역	차 수
	감쇄 [dB]	감쇄 [dB]	감쇄 [dB]	감쇄 [dB]	
BPF I	0.08	0.12	0.18	0.22	30 차
	-24.3	-0.55	-0.55	-24.3	
BPF II	0.06	0.1	0.2	0.24	"
	-23.8	-0.58	-0.56	-23.8	

표 6 필터의 계수
Coefficients of filters.

LPF 1.

$$\begin{aligned}
 a_0 &= -0.63974322 E - 02 = a_{29} \\
 a_1 &= 0.16931076 E - 01 = a_{28} \\
 a_2 &= 0.15247434 E - 01 = a_{27} \\
 a_3 &= 0.21682018 E - 02 = a_{26} \\
 a_4 &= -0.16643593 E - 01 = a_{25} \\
 a_5 &= -0.20264052 E - 01 = a_{24} \\
 a_6 &= 0.21395399 E - 02 = a_{23} \\
 a_7 &= 0.31772356 E - 01 = a_{22} \\
 a_8 &= 0.31942353 E - 01 = a_{21} \\
 a_9 &= -0.11977901 E - 01 = a_{20} \\
 a_{10} &= -0.63863650 E - 01 = a_{19} \\
 a_{11} &= -0.56597322 E - 01 = a_{18} \\
 a_{12} &= 0.47508702 E - 01 = a_{17} \\
 a_{13} &= 0.20983319 E + 00 = a_{16} \\
 a_{14} &= 0.33246157 E + 00 = a_{15}
 \end{aligned}$$

LPF 2

$$\begin{aligned}
 a_0 &= 0.16611932 E - 01 = a_{29} \\
 a_1 &= 0.39063655 E - 02 = a_{28} \\
 a_2 &= -0.14135711 E - 01 = a_{27} \\
 a_3 &= -0.77827009 E - 02 = a_{26} \\
 a_4 &= 0.14410681 E - 01 = a_{25} \\
 a_5 &= 0.19209864 E - 01 = a_{24} \\
 a_6 &= -0.15869658 E - 01 = a_{23} \\
 a_7 &= -0.31398699 E - 01 = a_{22} \\
 a_8 &= 0.92630908 E - 02 = a_{21} \\
 a_9 &= 0.52391376 E - 01 = a_{20} \\
 a_{10} &= 0.47413148 E - 02 = a_{19}
 \end{aligned}$$

$$\begin{aligned}
 a_{11} &= -0.84429458 E - 01 = a_{18} \\
 a_{12} &= -0.47905643 E - 01 = a_{17} \\
 a_{13} &= 0.17968154 E + 00 = a_{16} \\
 a_{14} &= 0.41304219 E + 00 = a_{15}
 \end{aligned}$$

BPF 1

$$\begin{aligned}
 a_0 &= 0.11224436 E - 02 = a_{29} \\
 a_1 &= -0.39693195 E - 01 = a_{28} \\
 a_2 &= -0.12469279 E - 01 = a_{27} \\
 a_3 &= 0.19336008 E - 02 = a_{26} \\
 a_4 &= 0.28095618 E - 02 = a_{25} \\
 a_5 &= -0.10042617 E - 01 = a_{24} \\
 a_6 &= -0.47989525 E - 02 = a_{23} \\
 a_7 &= 0.38253594 E - 01 = a_{22} \\
 a_8 &= 0.79625458 E - 01 = a_{21} \\
 a_9 &= 0.49225695 E - 01 = a_{20} \\
 a_{10} &= -0.61783273 E - 01 = a_{19} \\
 a_{11} &= -0.15908252 E + 00 = a_{18} \\
 a_{12} &= -0.22852947 E + 00 = a_{17} \\
 a_{13} &= 0.30797208 E - 01 = a_{16} \\
 a_{14} &= 0.18209115 E + 00 = a_{15}
 \end{aligned}$$

BPF 2

$$\begin{aligned}
 a_0 &= -0.74124928 E - 02 = a_{29} \\
 a_1 &= 0.53727343 E - 02 = a_{28} \\
 a_2 &= -0.27217191 E - 01 = a_{27} \\
 a_3 &= 0.10429103 E - 01 = a_{26} \\
 a_4 &= 0.48102085 E - 01 = a_{25} \\
 a_5 &= 0.43767709 E - 01 = a_{24} \\
 a_6 &= 0.66545047 E - 02 = a_{23} \\
 a_7 &= -0.66623352 E - 02 = a_{22} \\
 a_8 &= 0.25547735 E - 01 = a_{21} \\
 a_9 &= 0.33022583 E - 01 = a_{20} \\
 a_{10} &= -0.55318512 E - 01 = a_{19} \\
 a_{11} &= -0.17350549 E + 00 = a_{18} \\
 a_{12} &= -0.15768115 E + 00 = a_{17} \\
 a_{13} &= 0.40253326 E - 01 = a_{16} \\
 a_{14} &= 0.24679385 E + 00 = a_{15}
 \end{aligned}$$

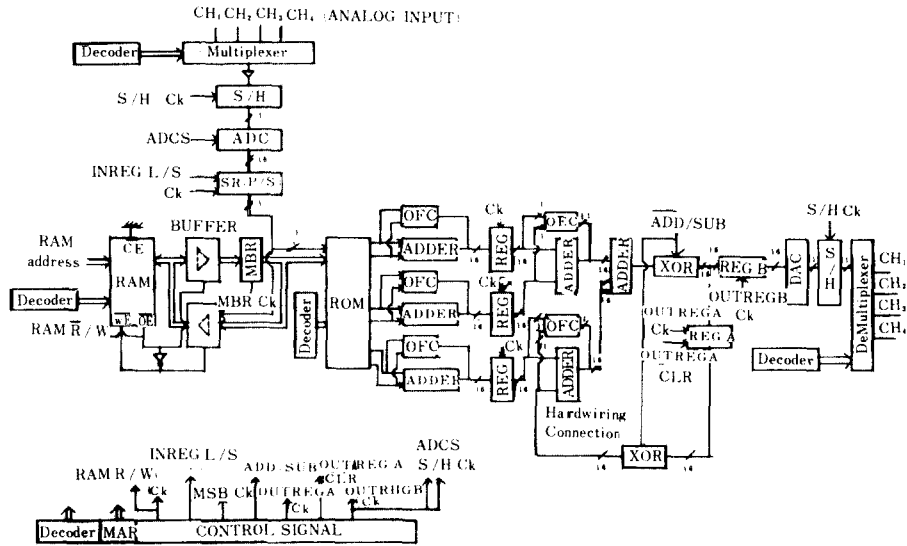


그림 6 하드웨어 구성도
Block diagram for hardware implementation.

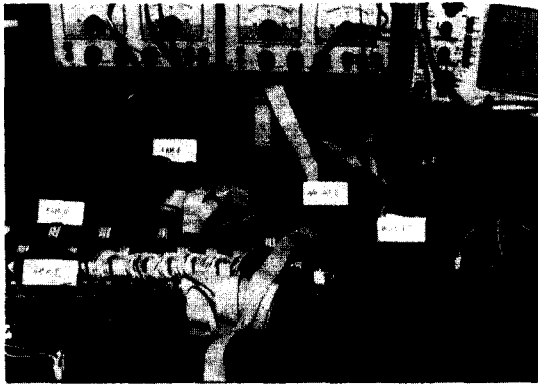


그림 7 구성된 디지털 필터의 외관도
Overall view of implemented digital filter.

5. 실험결과 및 고찰

저역필터 2개, 대역통과필터 2개를 이용한 4 채널 필터를 구성하였고, 모든 필터의 차수는 30차로 제한하였다.

표 7은 LPF 1의 경우, Remez 방식으로 시뮬레이션하여 얻어진 주파수 응답이 실험치와 잘 일치함을 보여준다. 변환시간이 짧은 AD 변환기를 이용하면 채널의 수를 증가시킬 수 있고, 필터의 개수가 많이 요구되는 디지털 시스템 구성

표 7 Remez 방식으로 시뮬레이션한 주파수 응답과 구성된 LPF 1의 주파수 응답의 비교
Comparison between the frequency response simulated by Remez method and that of LPF/implemented.

디지털주파수	시뮬레이션값 [dB]	구성회로의 값 [dB]
0.00	0.24	-0.23
0.05	-0.12	-0.13
0.10	-0.18	-0.18
0.15	-0.25	-0.24
0.20	-30.9	-30.3
0.25	-35.1	-34.9
0.30	-33.1	-32.7
0.35	-33.2	-32.8
0.40	-37.7	-36.5
0.45	-31.1	-30.3
0.49	-33.0	-32.4

에 적합하다고 사료된다. 제안된 기법에서 가장 전달시간이 지연되는 부분은 그림 4의 회로이다. 이 회로에서 전체 전달지연시간은 플립플롭 (7473), 레지스터 (74174), XOR 게이트의 두레벨

(7486), 가산기의 두 레벨(74283)의 전달시간 및 레지스터의 설정시간(setup time)의 합으로 이루어지며 약 78ns 정도가 된다.

본 회로의 어장(word length)이 16비트이므로 실시간처리에서 사용할 수 있는 최대 주파수는 약 800KHz 정도가 된다.

6. 결 론

본 연구에서는 벡터 승산 구조를 이용한 다중 채널 FIR 디지털 필터를 구성하는 한 기법을 제안하고, 이 기법에 따라 4 채널 디지털 필터를 구성하였으며 실험을 통해 다음과 같은 결론을 얻었다.

(1) ROM과 RAM을 이용하여 하드웨어의 구성을 간소화시켜 비용절감을 기하였다.

(2) 멀티오퍼랜드 가산기에 가/감산기를 결합한 회로를 구성하여서 동작속도를 개선하였다.

(3) 구성된 4 채널 디지털 필터중 LPF 1의 주파수 응답은 Remez 방식으로 시뮬레이션하여 얻어진 주파수 응답과 잘 일치하였다.

참 고 문 헌

- (1) A. Croisier, D. J. Esteban, M. E. Levilion, and V. Rizo, "Digital filter for PCM encoded signals," U. S. Patent. 3777130, December 4, 1973.
- (2) Abraham peled and Bede Liu, "A new hardware realization of digital filter," IEEE Transactions on Accoustics, speech, and signal processing, pp. 456~462, December 1974.
- (3) Abraham peled and Bede Liu, "Digital signal processing theory design and implementation," John Wiley & Sons, New York, pp. 212-227 & pp. 239-247, 1976.
- (4) J. L. Schmalzel, D. N. Hein, and N. Ahmed, "Some pedagogical considerations of digital filter hardware implementation," IEEE Circuits and System Magazine, pp. 4-13, March 1980.
- (5) M. Arjmand and R. A. Roberts, "On comparing hardware implementations of fixed point digital filters," IEEE Circuits and System Magazine, pp. 2-8, June 1981.
- (6) Stanley A. White, "On mechanization of vector multiplication," proceedings of the IEEE, pp. 730-731, march 1976.
- (7) Manfred Büttner and Hans-Wilhelm schübler, "On structures for the implementation of the distributed arithmetic," Nachrichtentech. z. Communications Journal, Vol. 6, No. 29, pp. 472-477, June 1976.
- (8) Chang-Fuu Chen, "A study of FIR digital filter algorithms and hardware implementation," Dissertation. Ph D. Stanford Univ, pp. 6-29, 1981.



金 明 起(Myung Ki KIM) 正會員
1930年 1月25日生
1958年 6月: 美國 海軍工科學 卒業
1966年 2月: 서울大學校 大學院 電子工學科卒業(工學碩士)
1976年 2月: 東亞大學校 大學院 電子工學專攻(工學博士)
1954年 2月-1969年 3月: 海軍士官學校 教授部勤務

1969年 4月-1972年 2月: IMEC電子株式會社 勤務

1972年 3月-現在: 東亞大學校 工科學 電子工學科教授



林 永 道(Young Do LIM) 正會員
1949年 9月12日生
1969. 3-1973. 2: 東亞大學校 電子工學科 卒業
1976. 3-1978. 2: 東亞大學校大學院 電子工學科(工學碩士)
1981. 3-1984. 2: 東亞大學校大學院 電子工學科博士過程修了

1978. 3-1980. 2: 盛智工業專門大學通信科 專任講師

1980. 3-現在: 東明專門大學電子科 助教授