

## 論 文

# DQSM 알고리즘을 이용한 다중채널 FIR 디지털 필터의 구성

正會員 林 永 道\* 正會員 金 明 起\*\*

## Multi-Channel FIR Digital Filter Hardware Implementation using 'DQSM Algorithm

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

**要 約** DQSM 알고리즘을 이용하여 다중채널 FIR 디지털 필터를 구성하기 위한 한 기법을 제안하였다. 본 논문에서는 Double precision 알고리즘과 절대치회로를 이용하여 ROM 용량을 감소시키고 절대치회로의 논리레벨의 단계를 줄이므로 동작속도를 개선할 수 있었다. 위의 기법으로 구성된 4 채널 FIR 디지털 필터의 주파수응답은 Remez exchange 알고리즘으로 시뮬레이션하여 얻어진 주파수응답과 잘 일치하였다.

**ABSTRACT** A method on the hardware implementation of the Multi-channel FIR digital filter using Digital Quarter Square Multiplication(DQSM) algorithm is proposed. This paper describes that ROM requirement can be reduced by using the double precision algorithm and the absolute value circuit, and also execution speed can be improved by reducing logic level steps of absolute value circuit. The frequency response of the four channel FIR digital filter implemented by the above method is quite agreeable with the frequency response simulated by Remez exchange algorithm.

### 1. 서 론

동작속도의 개선과 비용절감을 위한 승산기법에 관한 많은 논문들이 발표되었다<sup>(1), (2), (3), (4), (6)</sup>.

1980년 E. L. Jhonson은 아날로그연산에 이용하는 QSM (Quarter Square Multiplication)

알고리즘을 응용한 DQSM(Digital Quarter Square Multiplication) 알고리즘을 이용하여 2의 보수 승산회로를 제안하였고<sup>(5)</sup>, 그 후 1982년 Chen chang-fuu는 DQSM 알고리즘을 이용하여 FIR 디지털 필터를 구성한 바 있다<sup>(7)</sup>. 본 논문에서는 DQSM 알고리즘의 원리와 구성에 대해 기술하고 double precision 알고리즘과 절대치회로를 이용하여 많은 ROM 용량을 감소시켜 비용절감을 기하고 절대치회로의 논리레벨 단계를 줄이므로 동작속도를 개선할 수 있었다.

위의 기법으로 구성된 4 채널 32차 16비트 FIR 디지털 필터의 주파수응답은 Remez exchange 알고리즘으로 시뮬레이션하여 얻어진 주파수응답

\* 東明專門大學電子科  
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 : 86 - 23 (接受 : 1986. 6. 3)

과 잘 일치하였다<sup>(8)</sup>. 계수를 ROM 대신 RAM 에 기억시키면 계수가 시간적으로 변하는 FIR 디지털 필터를 구성하는 경우에 아주 유용하다.

2. DQSM 알고리즘의 기본원리와 구성

아날로그 컴퓨터에서 주로 이용되는 승산기법은 다음 식으로 주어진다.

$$x \cdot y = \frac{1}{4} \{ (x+y)^2 - (x-y)^2 \} \quad (1)$$

단,  $x$  는 피승수,  $y$  는 승수이다.

식(1)은 두 오퍼랜드(operand)의 합과 차를 발생시켜 합의 제곱에서 차의 제곱을 뺀 값의  $\frac{1}{4}$ 을 취하는 기법이므로 이와같은 승산기법을 QSM이라 한다. B 비트 워어드를 갖는 두개의 오퍼랜드를 승산하는 기법은 그림 1 과 같이  $2^{B+1} \cdot 2B$ -비트 워어드를 갖는 두개의 ROM 과 가산기 및 두개의 감산기로 구성할 수 있는데 이러한 기법을 DQSM 기법이라 한다<sup>(5)</sup>.

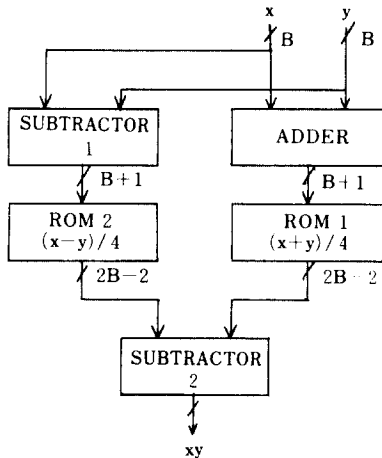


그림 1 DQSM의 기본구성도. Basic block diagram of DQSM.

그림 1에서 ROM 1 은  $(x+y)^2/4$  를 기억시키고 ROM 2 에는  $(x-y)^2/4$  를 기억시키면 감산기 2의 출력에서 두 오퍼랜드의 승산된 결과값을 얻을 수 있다.

$(x \pm y)$  연산을 실행할 때 발생하는 오버플로(overflow)를 교정하기 위한 회로는 그림2에 나타내었다<sup>(9)</sup>.

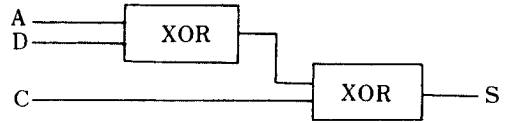


그림 2 오버플로 교정회로. Circuit of overflow corrector.

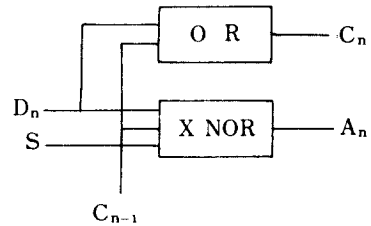


그림 3 절대치회로. Absolute value circuit.

그림 1에서 두 오퍼랜드 X와 Y의 차(감산기 1의 출력) 혹은 합(가산기의 출력)은 부호 비트를 포함한 B+1 비트로써 正 혹은 負의 값을 갖게 되는데 이 값을 +Z 혹은 -Z라 한다.

+Z와 -Z의 차승값이 같으므로 절대치회로를 이용하면 각 ROM의 용량은  $\frac{1}{2}$ 로 감소시킬 수 있다.

단일단계에 대한 절대치회로를 그림 3에 나타내고 이의 진리표는 표 1에 나타낸다.

표 1 절대치 회로의 진리표. Truth table of absolute value circuit.

D <sub>n</sub>	S	C <sub>n-1</sub>	A <sub>n</sub>	C <sub>n</sub>
0	0	0	0	0
0	0	1	d	d
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	d	d
1	1	0	0	0
1	1	1	1	0

그림 3에서  $D_n$ 과  $A_n$ 은 절대치를 취하기 전과 후의 값을 표시하고  $S$ 는  $D_n$ 의 부호비트이다.

이 회로에서  $B+1$  비트 정보의 절대치를 얻기 위해 OR 게이트의  $B-2$ 단 논리레벨과 한단의 XNOR 게이트 논리레벨이 필요하므로 회로의 전달시간이 크게 지연된다. 그러므로 그림 4는 2단의 NAND 게이트 논리레벨과 한단의 XOR 게이트 논리레벨만으로 절대치를 얻을 수 있으므로 동작속도가 개선된다.

동작속도를 개선시킨 절대치 회로를 이용함으로써 2단의 NAND 게이트와 1단의 XOR 게이트가 추가되지만 요구되는 ROM의 용량은 반으로 줄일 수 있다.

$C_{n-1}$ 이 "1"이면  $D_n$ 은 보수로 취해지고 "0"이면 변하지 않는다.  $C_{-1}$ 은 항상 "0"이므로  $D_0$ 는 변하지 않는다.

그림 1에서 ROM 2에  $(x-y)^2/4$ 의 2의 보수

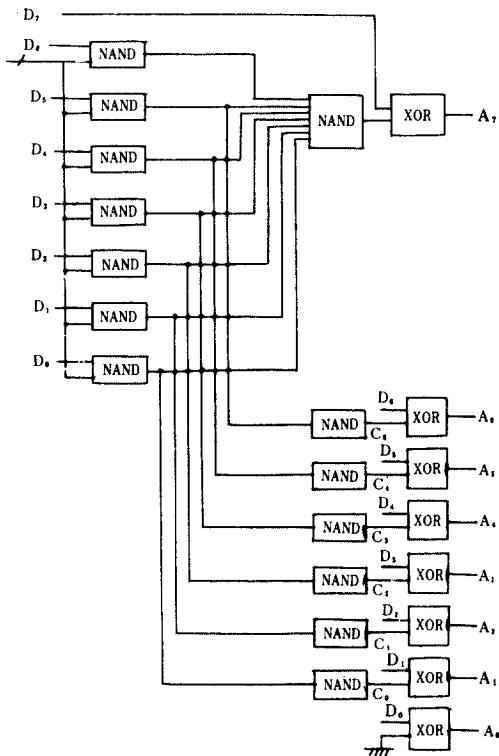


그림 4 고속을 위한 절대치회로  
Absolute value circuit for high speed.

값을 기억시키면 감산기 2 대신에 가산기를 이용하여 회로의 간편성을 기할 수 있다. 이러한 기법들을 이용한 FIR 디지털 필터의 구성도는 그림 5에 나타낸다.

입력정보는 RAM에 기억시키고, 고정계수인 경우는 ROM에 기억시키지만 시변계수인 경우는 ROM 대신 RAM에 기억시켜야 된다.

카운터와 다코우더로 구성된 제어회로는 ROM과 RAM의 어드레스를 지정한다. FIR 디지털 필터의 구성에 있어서 일반적으로 승산 결과의 모든 비트는 ROM의 용량문제점때문에 취하지 않는다. 그러므로 무시될 여분의 비트를 절삭(truncating)하면 최악의 경우 절대치오차는  $2^{-k}$

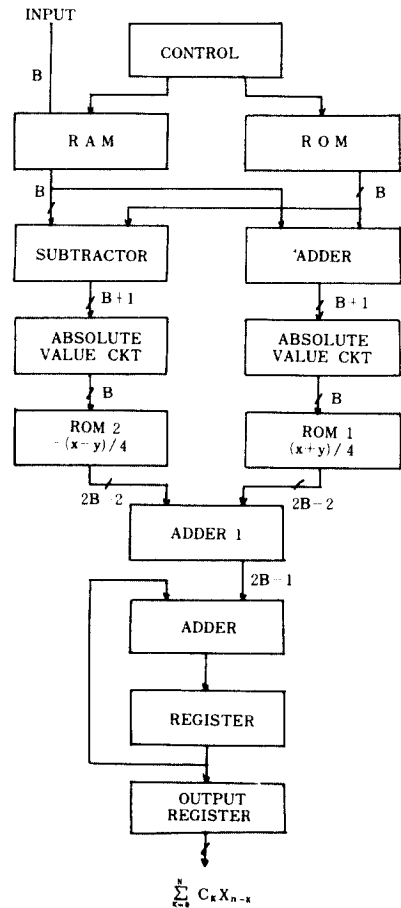


그림 5 DQSM에 의한 FIR 디지털 필터의 구성도  
Block diagram of the implementation of FIR digital filter with the DQSM.

로 근사하고, 만약 반올림(rounding)하면 절대치오차는  $2^{-k-1}$ 이내로 된다. 여기서,  $2^{-k}$ 는 요구하는 결과값의 최하위비트의 웨이트이다.

반올림하여 기억시킨 ROM의 내용을 누산하여 얻어진 최종결과값도 역시 반올림 된다.

그림 6은 전가산기를 이용한 반올림회로를 나타내고 식(2)는 회로동작을 나타낸다.

$$P : A_n \leftarrow B_n + 1 \quad (2)$$

식(2)에서 무시된 비트중의 최상의 비트의 값인 P가 "1"이면 절대치  $A_n$ 값은  $B_n + 1$ 로 되고 "0"이면 절삭된다. ( $i$  비트 무시하면 ROM 용량이 절약되어 비용절감을 기할 수 있다.)

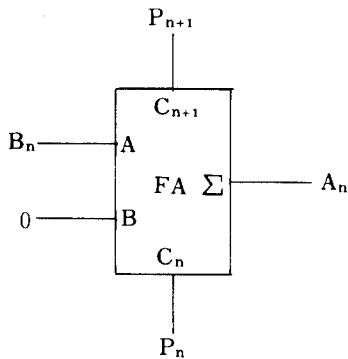


그림 6 반올림회로 Rounding circuit.

만약 ROM의 입력비트를 B로 하면 ROM의 어장은  $2B - 2$ 로 되어 승산 결과값은 거의 정확하다. 이런 경우 N차 FIR 디지털필터의 최종결과값을 절삭 또는 반올림하기 전의 절대치최대오차는 식(3)과 같다.

$$e_1 = (N + 1) \cdot 2^{-(2B-2)} \quad (3)$$

그러나 ROM의 어장을  $2B - 2 - i$ 로 하면 절대치최대오차는 식(4)와 같다.

$$e_i < (N + 1) \cdot 2^{-(2B-2-i)} < e_1 \cdot 2^i; i > 1 \quad (4)$$

이러한 오차의 최대값은 무시된 비트의 수  $i$ 에 지수적으로 증가함을 알 수 있다. 그러나 차수가 낮은 FIR 디지털필터에서는 무시할 정도로 적은 값이다. 예로  $N=26, B=12, i=6$ 이면 절대치 최대오차는  $4.1 \times 10^{-4}$  정도이다.

### 3. Double precision 알고리즘

ROM의 용량은 그림 5와 같이 오퍼랜드의 어장이 증가할수록 지수적으로 증가한다. 예로 B 비트 어장에 대한 정확한 승산 결과값을 얻기 위해 ROM의 용량은  $2 \cdot 2^B \cdot (2B - 2)$  비트이다. 그러므로 B가 크면 실용적이지 아니므로 몇개의 가산기를 부가한 Double precision 알고리즘에 의해 ROM 용량을 감소시킬 수 있다.

부호비트를 갖는 2의 보수시스템에서 두개의 오퍼랜드  $x$ 와  $y$ 의  $j$  번째 비트를  $x^j, y^j$ 로 할때  $x \cdot y$ 는 다음 식으로 표현된다.

$$\begin{aligned} x \cdot y &= (-x^0 + \sum_{j=1}^{B-1} x^j \cdot 2^{-j}) \cdot (-y^0 + \sum_{j=1}^{B-1} y^j \cdot 2^{-j}) \\ &= [(-x^0 + \sum_{j=1}^{K-1} x^j \cdot 2^{-j}) + \sum_{j=K}^{B-1} x^j \cdot 2^{-j}] \\ &\quad \cdot [(-y^0 + \sum_{j=1}^{K-1} y^j \cdot 2^{-j}) + \sum_{j=K}^{B-1} y^j \cdot 2^{-j}] \\ &= (-x^0 + \sum_{j=1}^{K-1} x^j \cdot 2^{-j}) \cdot (-y^0 + \sum_{j=1}^{K-1} y^j \cdot 2^{-j}) \\ &\quad + (-x^0 + \sum_{j=1}^{K-1} x^j \cdot 2^{-j}) \cdot (\sum_{j=K}^{B-K} y^{j+K-1} \cdot 2^{-j}) \cdot 2^{-K+1} \\ &\quad + (-y^0 + \sum_{j=1}^{K-1} y^j \cdot 2^{-j}) \cdot (\sum_{j=1}^{B-K} x^{j+K-1} \cdot 2^{-j}) \cdot 2^{-K+1} \\ &\quad + (\sum_{j=1}^{B-K} x^{j+K-1} \cdot 2^{-j}) \cdot (\sum_{j=1}^{B-K} y^{j+K-1} \cdot 2^{-j}) \cdot 2^{-2K+2} \end{aligned} \quad (5)$$

식(5)는 부호비트를 포함한 K 비트 그룹과 B-K 비트 그룹으로 나뉜다. 식(5)의 마지막 항은 그림 7과 같이 실현될 수 있다.

이때 절대치회로는 그림 8과 같이 1의 보수시스템으로 간단하게 실현되고 감산기는 가산기와 인버터로 구성된다. 감수의 1의 보수를 피감수

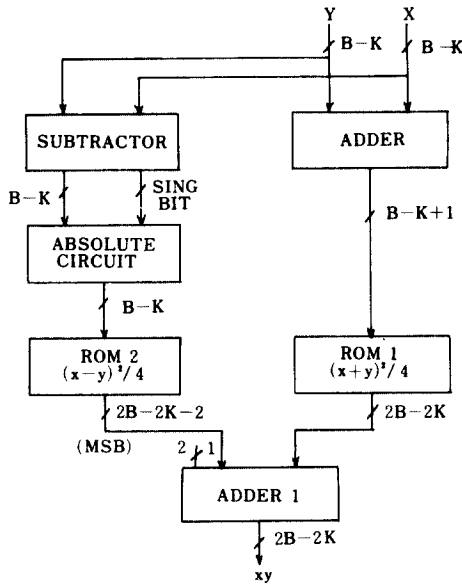


그림 7 부호비트가 없는 두 오퍼랜드의 승산구성도  
Block diagram of implementing a multiplication of operands without sign bit.

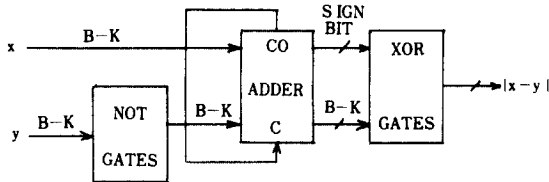


그림 8 부호비트가 없는 두 오퍼랜드의 절대차회로 구성도  
Block diagram of an absolute difference circuit for two operands without sign bit.

에 더하여 EAC(End Around Carry)가 발생하면 두 오퍼랜드의 차는 正이고, EAC가 발생하지 않으면 負의 값이 된다. 負의 값일 경우는 인버터를 이용하여 절대치를 쉽게 얻을 수 있다. 그러므로 전술한 그림 5의 기법에 비해 회로가 간편해지고 동작속도가 개선된다.

식(5)의 제 2항과 제 3항은 부호비트를 갖는 2의 보수 오퍼랜드에 正의 오퍼랜드를 곱하는 경우로서 전술한 그림 5의 회로로 실현된다. 이때 ROM의 위어드는

$K \geq \frac{B}{2} + 1$  인 경우는  $2^{K+1}$ ,  $K \leq \frac{B}{2}$  인 경우는  $2^{B-K+2}$ 가 필요하다.

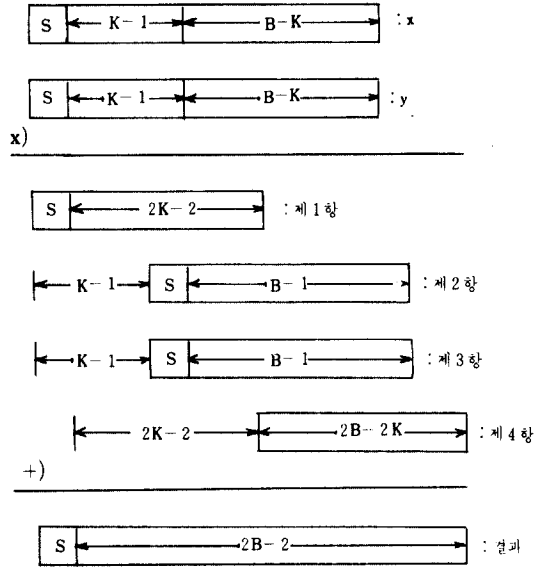


그림 9 이중프레시적승산의 부분 곱  
Partial products of a double precision multiplication.

그림 9에 나타낸 바와같이 식(5)의 제 1항과 마지막 항은 중복(overlap)되지 않으므로 승산 결과를 얻는 과정에서 4개의 오퍼랜드만을 가산할 수 있다. 그림 5와 비교해 보면 한레벨의 가산기가 추가되어 전달시간은 지연되나 실제로 많은 ROM의 용량을 감소시킬 수 있다. 만약 B가 아주 큰 경우에는 Multi-precision 알고리즘이 꼭 필요한 것이다.

#### 4. 다중채널 방식

그림 10과 같이 ROM과 RAM의 여분어드레스로써 입력채널의 수만큼 디코딩하여 메모리영역을 할당한다.

멀티플렉서에서 나오는 해당채널의 입력 정보는 RAM과 버퍼(buffer)를 이용하여 제어신호에 의해 동일번지내에서 전의 정보  $x_{n-k}$ 을 읽어(Buffer  $C_k = "H"$ ,  $R/\bar{W} = "H"$ ) 메모리버퍼 레지스터 MBR(Memory Buffer Resister)로 전송한 다음 현재 입력정보  $x_{n-(k+1)}$ 을 RAM에 기억시킨다. (Buffer  $C_k = "L"$ ,  $R/\bar{W} = "L"$ )

채널별로 할당된 ROM에 기억시키고 ROM의

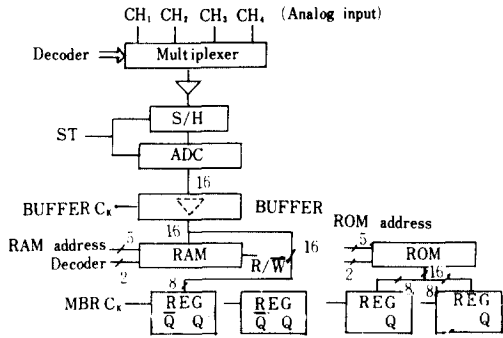


그림 10 다중채널방식을 위한 구성도.  
Block diagram for multi-channel method.

메모리 내용은 표 2에 나타낸다.

표 2 ROM의 메모리 내용  
Memory map of ROM.

메모리어드레스	내 용	비 고
00 - 1F	CH 1의 계수 $a_0 \sim a_{31}$	LPF 1
20 - 3F	CH 2 "	LPF 2
40 - 5F	CH 3 "	BPF 1
60 - 7F	CH 4 "	BPF 2

표 3 필터의 규격  
Specifications of filters.

구 분	통과대역 차단주파수		감쇄대역 차단주파수		차수
	감쇄 [dB]		감쇄 [dB]		
LPF I	0.15	0.2	-0.18	-33.7	32차
LPF II	0.2	0.25	-0.21	-32.7	"
구 분	저역차단주파수		고역차단주파수		차수
	감쇄영역	통과영역	통과영역	감쇄영역	
BPF I	0.08	0.12	0.18	0.22	32차
	-26.2	-0.46	-0.44	-26.2	
BPF II	0.06	0.1	0.2	0.24	"
	-24.9	-0.51	-0.51	-24.9	

이때 최대 입력채널의 수  $n$ 은

$$n = \frac{T_s}{T_c + T_r} \quad (6)$$

표 4 필터의 계수  
Coefficients of filters.

LPF 1

$a_0 = -0.12580726 E - 01 = a_{31}$
$a_1 = 0.68240613 E - 03 = a_{30}$
$a_2 = 0.11493956 E - 01 = a_{29}$
$a_3 = 0.15288784 E - 01 = a_{28}$
$a_4 = 3069177373 E - 02 = a_{27}$
$a_5 = -0.16051199 E - 01 = a_{26}$
$a_6 = -1973429797 E - 01 = a_{25}$
$a_7 = 0.30771027 E - 02 = a_{24}$
$a_8 = 329480390. E - 01 = a_{23}$
$a_9 = 0.32882046 E - 01 = a_{22}$
$a_{10} = -0.11575410 E - 01 = a_{21}$
$a_{11} = -0.63743807 E - 01 = a_{20}$
$a_{12} = -0.56363333 E - 01 = a_{19}$
$a_{13} = 0.47947090 E - 01 = a_{18}$
$a_{14} = 0.21025324 E + 00 = a_{17}$
$a_{15} = 0.33272400 E + 00 = a_{16}$

LPF 2

$a_0 = -0.39445565 E - 03 = a_{31}$
$a_1 = .17375086 E - 01 = a_{30}$
$a_2 = .28881757 E - 02 = a_{29}$
$a_3 = -.13025203 E - 01 = a_{28}$
$a_4 = -.88306228 E - 02 = a_{27}$
$a_5 = .15380838 E - 01 = a_{26}$
$a_6 = .18346490 E - 01 = a_{25}$
$a_7 = -.15094390 E - 01 = a_{24}$
$a_8 = -.32090779 E - 01 = a_{23}$
$a_9 = .98533574 E - 02 = a_{22}$
$a_{10} = .51883899 E - 01 = a_{21}$
$a_{11} = .51658405 E - 02 = a_{20}$
$a_{12} = -.84753782 E - 01 = a_{19}$
$a_{13} = -.47680058 E - 01 = a_{18}$
$a_{14} = .17953773 E + 00 = a_{17}$
$a_{15} = .41309619 E + 00 = a_{16}$

BPF 1

$a_0 = .18644039 E - 01 = a_{31}$
$a_1 = -.18519249 E - 01 = a_{30}$
$a_2 = -.26479645 E - 01 = a_{29}$
$a_3 = -.17918328 E - 01 = a_{28}$
$a_4 = .10704063 E - 03 = a_{27}$
$a_5 = .35720579 E - 02 = a_{26}$
$a_6 = -.90651698 E - 02 = a_{25}$
$a_7 = -.49349219 E - 02 = a_{24}$
$a_8 = .37333585 E - 01 = a_{23}$
$a_9 = .79119043 E - 01 = a_{22}$
$a_{10} = .49747497 E - 01 = a_{21}$
$a_{11} = -.60601674 E - 01 = a_{20}$
$a_{12} = -.15809725 E + 00 = a_{19}$
$a_{13} = -.12831770 E + 00 = a_{18}$
$a_{14} = .30076798 E - 01 = a_{17}$
$a_{15} = .18088952 E + 00 = a_{16}$

BPF 2

$a_0 = -.14033278 E - 01 = a_{31}$
$a_1 = .12828214 E - 01 = a_{30}$
$a_2 = -.13425309 E - 01 = a_{29}$
$a_3 = -.18033192 E - 01 = a_{28}$
$a_4 = .98252892 E - 02 = a_{27}$
$a_5 = .45725916 E - 01 = a_{26}$
$a_6 = .44672649 E - 01 = a_{25}$
$a_7 = .75775599 E - 02 = a_{24}$
$a_8 = -.74270051 E - 02 = a_{23}$
$a_9 = .24698270 E - 01 = a_{22}$
$a_{10} = .33458896 E - 01 = a_{21}$
$a_{11} = -.54259352 E - 01 = a_{20}$
$a_{12} = -.17318834 E + 00 = a_{19}$
$a_{13} = -.15814833 E + 00 = a_{18}$
$a_{14} = .40245365 E - 01 = a_{17}$
$a_{15} = .24782662 E + 00 = a_{16}$

로 된다. 단,  $T_s$ 는 한채널의 표본화주기(sec),  $T_c$ 는 AD변환기의 변환시간(sec),  $T_r$ 은 한채널의 전달지연시간(sec)이다.

여기서  $n$ 은  $\frac{T_s}{T_c + T_r}$  보다 작은 최대정수의 값을 표현한다.

5. 필터의 구성

4 채널 32차 16비트 FIR 디지털 필터를 저역필터 2개(LPF 1, LPF 2)와 대역통과필터 2개(BPF 1, BPF 2)로 구성하였으며 이들 필터의 규격과 계수의 값은 Remez exchange 알고리즘으로 시뮬레이션하여 구하고 표 3과 표 4에 나타내었다.

식(5)의 Double precision 알고리즘에서 K의 값은 8로 해서 전술한 기법으로 구성한 32차 16비트 4 채널 FIR 디지털 필터의 하드웨어 구성도와 제어신호를 그림11, 그림12에 나타내었다.

그림11에서 ROM의 어드레스를 2의 보수 시스템인  $(x \pm y)$  정보로 취하여 4 개의 ROM (+)

에는  $\frac{|x+y|^2}{4}$ , 4 개의 ROM (-)에는  $-\frac{|x-y|^2}{4}$ 의 값을 기억시키고 표 5에 나타내었다.

6. 실험결과 및 고찰

16비트 32차 4 채널 FIR 디지털 필터를 저역필터 2개, 대역통과필터 2개로 구성하였다. 구성된 필터의 입력에 4 채널 0dB의 정현파 발진세력을 가하고 각각의 입력 주파수를 증가시키면서 필터된 4 채널 출력 세력을 dB미터로써 측정하여 주파수 응답을 조사하였다.

예시로 표 6은 LPF 1의 경우 Remez exchange 알고리즘으로 시뮬레이션하여 얻어진 주파수 응답은 구성된 필터의 실험치와 잘 일치함을 보여 준다.

이 회로의 전체 전달지연시간은 레지스터 2단, 전가산기 5단, ROM 2단, 절대치회로의 전달시간 및 레지스터의 설정시간의 합으로 이루어지며 TTL ROM IC를 사용하면 약 2  $\mu$ s, MOS ROM

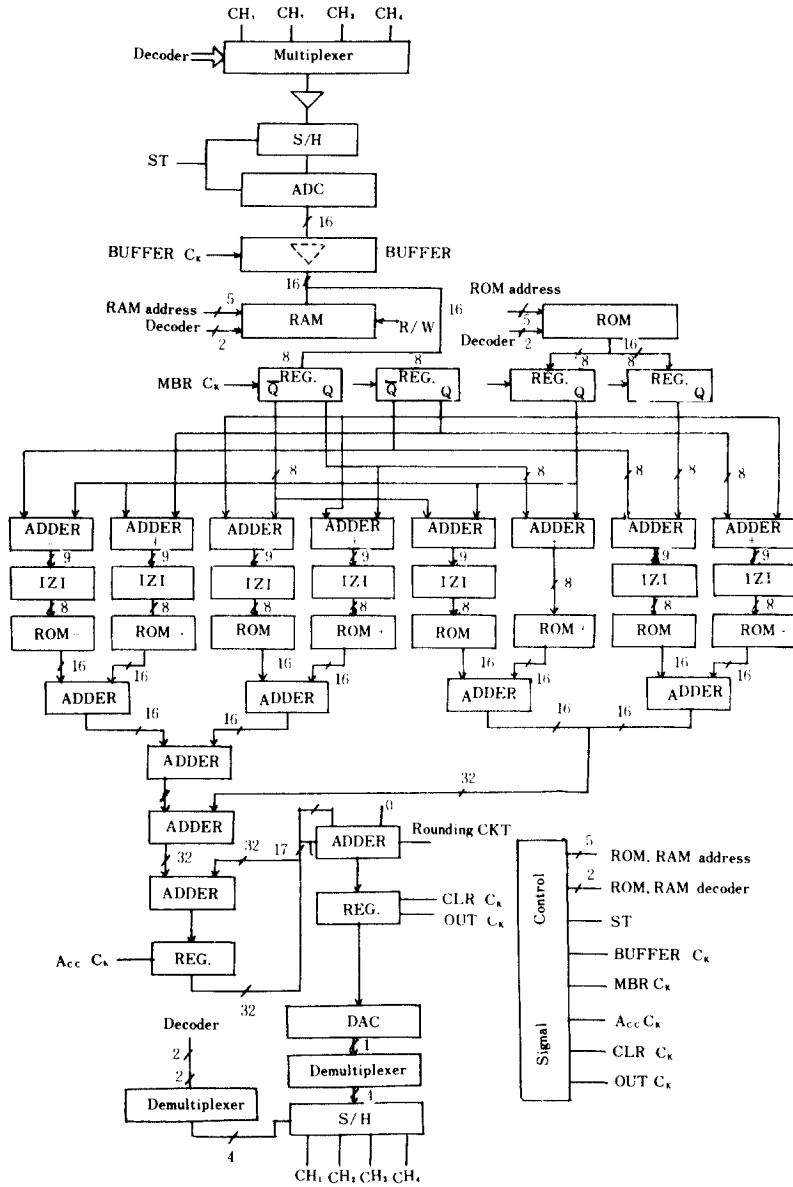


그림 11 하드웨어 구성도. 어  
Block diagram for hardware implementation.

IC를 사용하면 약  $14\mu s$  정도이다.

본 회로의 차수가 32차이므로 실시간 처리에서 사용할 수 있는 최대주파수는 약 500 KHz 정도가 된다. 그리고 Double precision 알고리즘과 절대치회로를 이용하므로 그림 5에 비해 ROM의 용량은  $(B=16, K=8) 2 \cdot 2^B \cdot (2B-2) / 2^3 \cdot 2^{B-K}$ .

$\{2(B-K)-2\} \approx 137$ 배 정도 감소될 수 있다.

### 7. 결 론

본 연구에서는 DQSM 원리를 이용한 다중채널 FIR 디지털필터를 구성하는 한 기법을 제시하고



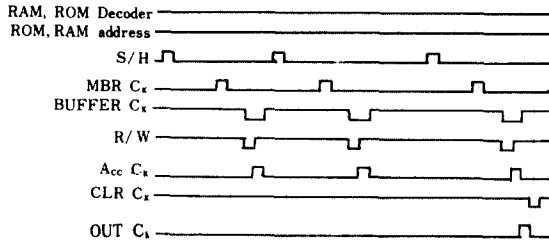


그림12 제어신호  
Control signal.

표5 ROM의 내용  
Contents of ROM.

번지	ROM (-) 내용 $\left( -\frac{ x-y ^2}{4} : \text{High Byte} \right)$
00	00 00 00 FF FF FF FF FF FF FF FF FF FF FF FF FF
10	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
20	FF FF FF FF F7 FF FF FF FF FF FF FF FF FF FE FE
30	FE FE FE FE FF FE FE FE FE FE FE FE FE FE FE FE
40	FE FD FD FD FD FD FD FD FD FD FD FD FD FD FD FC
50	FC FC FC FC FC FC FC FC FC FC FC FB FB FB FB FB
60	FB FB FB FB FB FA FA FA FA FA FA FA FA FA FA F9
70	F9 F9 F9 F9 F9 F9 F9 F8 F8 F8 F8 F8 F8 F8 F8 F8
80	F8 F7 F7 F7 F7 F7 F7 F7 F6 F6 F6 F6 F6 F6 F6 F6
90	F5 F5 F5 F5 F5 F5 F5 F4 F4 F4 F4 F4 F4 F3 F3 F3
A0	F3 F3 F3 F3 F2 F2 F2 F2 F2 F2 F1 F1 F1 F1 F1 F1
B0	F0 F0 F0 F0 F0 F0 FF EF FF EF EF EE EE FE EE FE
C0	EE ED ED ED ED ED EC EC EC EC EC EB EB EB EB EB
D0	EA EA EA EA EA E9 E9 E9 E9 E9 E8 E8 E8 E8 E7 E7
E0	E7 E7 E7 E6 E6 E6 E6 E5 E5 E5 E5 E4 E4 E4 E4 E4
F0	E3 E3 E3 E3 E2 E2 E2 E2 E1 E1 E1 E1 E0 E0 E0 E0

이 기법에 따라 4 채널 FIR 디지털 필터를 구성하였으며 실험을 통해 다음과 같은 결론은 얻었다.

- (1) Double precision 알고리즘과 절대치 회로를 이용하여 많은 ROM 용량을 감소시켜 비용절감을 기하였다.
- (2) 절대치회로의 논리레벨을 줄이고 식(5)의 마지막 항의 승산기법을 제시한 그림 8 과 같은 구성도를 이용하여서 동작속도의 개선을 기하였다.
- (3) 다중채널 필터의 구성법을 제안하여 필터의 개수가 많은 디지털시스템의 구성을 용이하게 하

번지	ROM (-) 내용 $\left( -\frac{ x-y ^2}{4} : \text{Low Byte} \right)$
00	00 00 00 FF FE FD FC FA F8 F6 F4 F1 EE EB E8 E4
10	E0 DC D8 D3 CE C9 C4 BE B8 B2 AC A5 9E 97 90 88
20	80 78 70 67 5E 55 4C 42 38 2E 24 19 0E 03 F8 EC
30	E0 D4 C8 BBAE A1 94 86 78 6A 5C 4D 3E 2F 20 10
40	00 F0 E0 CF BE AD 9C 8A 78 66 54 41 2E 1B 08 F4
50	E0 CC B8 A3 8E 79 64 4E 38 22 0C F5 DE C7 B0 98
60	80 68 50 37 1E 05 EC D2 B8 9E 84 69 4E 33 18 FC
70	E0 C4 A8 8B 6E 51 34 16 F8 DA BC 9D 7E 5F 40 20
80	00 E0 C0 9F 7E 5D 3C 1A F8 D6 B4 91 6E 4B 28 04
90	E0 BC 98 73 4E 29 04 DE B8 92 6C 45 1E F7 D0 A8
A0	80 58 30 07 DE B5 8C 62 38 0E E4 B9 8E 63 38 0C
B0	E0 B4 88 5B 2E 01 D4 A6 78 4A 1C ED BE 8F 60 30
C0	00 D0 A0 6F 3E 0D DC AA 78 46 14 E1 AE 7B 48 14
D0	E0 AC 78 43 0E D9 A4 6E 38 02 CC 95 5E 27 F0 B8
E0	80 48 10 D7 9E 65 2C F2 B8 7E 44 09 CE 93 58 1C
F0	E0 A4 68 2B EE B1 74 36 F8 BA 7C 3D FE BF 80 40

번지	ROM (+) 내용 $\left( \frac{ x+y ^2}{4} : \text{High Byte} \right)$
00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
10	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
20	00 00 00 00 00 00 00 00 00 00 00 00 00 01 01
30	01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
40	02 02 02 02 02 02 02 02 02 02 02 02 02 02 03
50	03 03 03 03 03 03 03 03 03 03 04 04 04 04 04
60	04 04 04 04 04 04 05 05 05 05 05 05 05 05 06
70	06 06 06 06 06 06 06 06 06 07 07 07 07 07 07
80	07 08 08 08 08 08 08 08 08 09 09 09 09 09 09
90	0A 0A 0A 0A 0A 0A 0A 0B 0B 0B 0B 0B 0C 0C 0C
A0	0C 0C 0C 0D 0D 0D 0D 0D 0D 0E 0E 0E 0E 0E 0E
B0	0F 0F 0F 0F 0F 10 10 10 10 10 11 11 11 11 11
C0	11 12 12 12 12 12 13 13 13 13 14 14 14 14 14
D0	15 15 15 15 15 16 16 16 16 16 17 17 17 18 18
E0	18 18 18 17 19 19 19 1A 1A 1A 1A 1B 1B 1B 1B
F0	1C 1C 1C 1C 1D 1D 1D 1E 1E 1E 1E 1F 1F 1F 1F

번지	ROM (+) 내용 $\left( \frac{ x+y ^2}{4} : \text{Low Byte} \right)$
00	00 00 00 01 02 03 04 06 08 0A 0C 0F 12 15 18 1C
10	20 24 28 2D 32 37 3C 42 48 4E 54 5B 62 69 70 78
20	80 88 90 99 A2 AB B4 BE C8 D2 DC E7 F2 FD 08 14
30	20 2C 38 45 52 5F 6C 7A 88 96 A4 B3 C2 D1 E0 F0
40	00 10 20 31 42 53 64 76 88 9A AC BF D2 E5 F8 0C
50	20 34 48 5D 72 87 9C B2 C8 9E F4 0B 22 39 50 68
60	80 98 B0 C9 E2 FB 14 2E 48 D 7C 97 B2 CD E8 04
70	20 3C 58 75 92 AF CC EA 08 62 44 63 82 A1 C0 E0
80	00 20 40 61 82 A3 C4 E6 08 2A 4C 6F 92 B5 D8 FC
90	20 44 68 8D B2 D7 FC 22 48 2E 94 BB E2 09 30 58
A0	80 A8 D0 F9 22 4B 74 9E C8 62 1C 47 72 9D C8 F4
B0	20 4C 78 A5 D2 FF 2C 5A 88 F6 E4 13 42 71 A0 D0
C0	00 30 60 91 C2 F3 24 56 88 BA EC 1F 52 85 B8 EC
D0	20 54 88 BD F2 27 5C 92 C8 BE 34 6BA 2 D9 10 48
E0	80 B8 F0 29 62 9B D4 0E 48 F2 BC F7 32 6D A8 E4
F0	20 5C 98 D5 12 4F 8C CA 08 8B 84 C3 02 41 80 C0

표 6 Remez exchange 알고리즘으로 시뮬레이션 한 주파수 응답과 구성된 필터의 주파수 응답의 비교  
 Comparison between the frequency response simulated by Remez exchange algorithm and that of filter implemented.

디 지 탈 주 파 수	시뮬레이션 값(dB)				구성회로의 값(dB)			
	LPF 1	LPF 2	BPF 1	BPF 2	LPF 1	LPF 2	BPF 1	BPF 2
0.00	0.18	0.20	-26.21	-24.93	0.15	0.18	-26.4	-25.2
0.05	-0.18	0.00	-42.84	-28.11	-0.15	-0.01	-41.9	-27.6
0.10	-0.01	-0.21	-5.89	-0.51	-0.0	-0.15	-6.3	-0.48
0.15	-0.18	-0.05	0.30	-0.50	-0.1	-0.05	0.2	-0.52
0.20	-33.71	-0.21	-5.45	-0.51	-34.0	-0.1	-4.8	-0.56
0.25	-53.59	-32.65	-33.30	-26.95	-54.5	-31.8	-32.9	-27.4
0.30	-35.3	-45.55	-30.51	-44.67	-35.5	-46.4	-30.4	-45.2
0.35	-35.87	-32.64	-27.10	-24.94	-36.9	-31.4	-28.2	-25.3
0.40	-41.15	-59.62	-37.23	-50.24	-40.8	-58.7	-36.5	-51.4
0.45	-33.91	-32.64	-26.26	-24.92	-34.5	-31.4	-27.3	-25.3
0.49	-35.27	-34.51	-27.88	-26.71	-36.5	-34.2	-28.2	-27.3

였다.

(4) 구성된 4 채널 디지털필터중 LPF 1 의 주파수응답은 Remez exchange 알고리즘으로 시뮬레이션하여 얻어진 주파수응답과 잘 일치하였다.

參 考 文 獻

(1) H. Ling, "High-speed computer multiplication using a multiple-bit decoding algorithm", IEEE Transactions on computers, vol. c-19, no. 8, pp706-709, Aug., 1970.

(2) Tien Chi Chen, "A binary multiplication schime based on squaring", IEEE Transaction on Computers, pp.678-680, June 1971.

(3) J. Robert Logan, "A Design Technique for Digital squaring Networks", Computer Design, pp84-88, February 1970.

(4) J. Robert Logan, "A Square-Summing High Speed Multiplier", Computer Design, pp. 67-70, June 1971.

(5) Evertt L. Johnson, "A digital quarter square multiplier", IEEE Transaction on Computers, Vol. c-29, No.3, pp. 258-261, March 1980.

(6) M. J. Narasimha, "Implementation of FIR filters with random access memories", presented at the ELECTRO'79, New York, NY, April 24-26, paper 8/2, 1979.

(7) Chen Chang-Fuu, "Astudy of FIR digital filter algorithms and hardware implementations", Dissertation, Ph D. Stanford Univ., pp. 84-103, 1981.

(8) 임영도, 김명기 "벡터 승산 구조를 이용한 다중채널 FIR 디지털 필터구상", 한국통신학회지 제10권 제 6 호, pp.327-334, 1985. 12.

(9) Kai-Ping Yiu, "On sign bit assignment for a vector Multiplier. Proceedings of the IEEE, pp.372-373, March 1976.



林 永 道 (Young Do LIM) 正會員  
 1949年 9月 20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



金 明 起 (Myung Ki KIM) 正會員  
 1930年 1月 25日 生  
 1958年 6月 : 美國 海軍工科學大學 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 工科學大學 電子工學科 教授 本學會 釜山-慶南 支部長