

# 곱셈기가 제거된 의료 초음파 신호처리용 프로그래머블 FIR 필터 구현을 위한 수정된 SaA 구조

한호산, 송재희, 김학현, 고방영, 송태경

서강대학교 전자공학과

(Received January 22, 2007. Accepted March 26, 2007)

## A Modified SaA Architecture for the Implementation of a Multiplierless Programmable FIR Filter for Medical Ultrasound Signal Processing

Ho-San Han, Jae-Hee Song, Hak-Hyun Kim, Bang-Young Goh, Tai-Kyong Song

Department of Electronics Engineering, Sogang University

### Abstract

Programmable FIR filters are used in various signal processing tasks in medical ultrasound imaging, which are one of the major factors increasing hardware complexity. A widely used method to reduce the hardware complexity of a programmable FIR filter is to encode the filter coefficients in the canonic signed digit (CSD) format to minimize the number of nonzero digits (NZD) so that the multipliers for each filter coefficients can be replaced with fixed shifters and programmable multiplexers (PM).

In this paper, a new structure for programmable FIR filters with a improved frequency response and a reduced hardware complexity compared to the conventional shift-and-add architecture using PM is proposed for implementing a very small portable ultrasound scanner. The CSD codes are optimized such that there exists at least one common nonzero digit between neighboring coefficients. Such common digits are then implemented with the same shifters. For comparison, synthesizable VHDL models for programmable FIR filters are developed based on the proposed and the conventional architectures. When these filters have the same hardware complexity, pass-band and stop-band ripples of the proposed filter are lower than those of the conventional filter by about 0.01~0.19dB and by about 5~10dB, respectively. For the same filter performance, the hardware complexity of the proposed architecture is reduced by more than 20% compared to the conventional SaA architecture.

**Key words :** portable ultrasound scanner programmable FIR filter shift-and-add architecture canonic signed digit

### 1. 서 론

초음파 영상시스템에서 프로그래머블 FIR 필터는 초음파 집속부, 잡음 제거를 위한 동적필터, 혈류 영상을 위한 도플러 신호처리 등 여러 목적을 위하여 사용되고 있다[1-6]. 이러한 프로그래머블 FIR 필터는 일반적으로 이들이 사용되는 부분의 하드웨어 복잡도를 좌우하는 가장 주요한 요소이다. 또한 초음파 영상에서 사용되는 프로그래머블 FIR 필터는 대부분 고속으로 동작하여야 하므로 시스템의 전력소비와 회로의 열 발생을 증가시키는 요

인이기도 하다. 따라서 소형 휴대용 초음파 영상장치 개발을 위해서는 효율적인 프로그래머블 FIR 필터의 설계가 매우 중요하다.

일반적인 FIR 필터는 곱셈기가 큰 하드웨어 면적을 차지하는 단점이 있기 때문에 높은 성능의 고차 필터 설계에 어려움이 있다. 이러한 단점을 개선하여 적은 하드웨어 면적을 차지하는 필터 시스템을 구현하기 위해 곱셈기를 배제한 디지털 필터의 회로 설계의 구현에 대한 연구가 활발히 수행되어 왔다[7-17]. 이를 위한 가장 기본적인 방법은 곱셈 연산을 제한된 수의 시프트 연산과 덧셈 연산 과정으로 대체하는 것이다. 즉, 필터 계수를 제한된 수의 영이 아닌 디지털(NZD: nonzero digit) 들로 표현하고, 이 NZD 들에 대해서만 시프트 연산을 하는 SaA (shift and add) 구조로 곱셈기를 대체하여 필터의 하드웨어 복잡도를 감소시키는 방식이 많이 사용되고 있다[7-11].

Corresponding Author : 송태경  
서울시 마포구 신수동 1번지 서강대학교 전자공학과 Ricci관 806호  
TEL : 02-705-8907 / FAX : 02-707-3008  
E-mail : tksong@sogang.ac.kr

고정된 계수를 가지는 필터의 경우에는 주어진 계수 값에 따른 최적화된 하드웨어 구조를 구현하였다. 그러나, 특정 계수에 적합하게 필터를 구현하면 계수의 변화에 따라 필터의 전체적인 구조를 다시 구현해야하는 단점이 있다. 이러한 시스템의 낮은 유연성을 개선하기 위하여 기존의 프로그래머블 SaA 구조에서는 고정된 시프터와 프로그래머블 멀티플렉서(PM: programmable multiplexer)를 이용하여 하드웨어 복잡도를 줄이는 방법을 사용하였다[9].

일반적으로 FIR 필터의 이웃하는 계수들은 공통된 SPT(signed power of two)항이 자주 발생하는 특징을 가진다. 본 논문에서는 이웃하는 계수 사이의 공통된 SPT 항을 하나의 SaA로 구현함으로써 하드웨어 복잡도를 줄일 수 있는 개선된 SaA 구조를 이용한 고속, 고차 프로그래머블 FIR 필터 설계 방법을 제안하였다. 또한, VHDL을 이용한 필터 설계를 통하여 기존의 SaA 구조와 제안된 SaA 구조를 이용한 프로그래머블 FIR 필터의 성능 및 하드웨어 복잡도를 비교하였다.

결과적으로, 하드웨어 복잡도를 유사하게 구현한 제안된 SaA 구조와 기존의 PM을 이용한 SaA 구조를 갖는 필터에 대해 제안된 SaA 구조의 필터가 주파수 특성에서 통과 대역 리플은 0.01~0.19dB, 저지 대역 감쇠는 약 5~10dB가 개선되었다. 그리고, 성능을 유사하게 구현한 필터에 대해서는 제안된 SaA 구조의 필터가 기존의 PM을 이용한 SaA 구조의 필터에 비해 하드웨어 복잡도가 20%이상 감소되었다.

## II. SPT 공유 방식의 SaA 기반 프로그래머블 FIR 필터 구조

CSD를 이용한 필터 계수  $h(n)$ 은 다음과 같이 표현된다[8].

$$h(n) = \sum_{k=0}^{m-1} s_k \cdot 2^k \quad (1)$$

여기서  $m$ 은 양자화 비트 수이고,  $s_k$ 는 각 비트에 대한 2의 승수의 부호를 결정하는 인자  $\{-1, 0, 1\}$  중 하나의 값을 가지며 인접한 인자들이 모두 NZD가 될 수 없도록 다음과 같은 조건을 만족하여야 한다.

$$s_k \cdot s_{k+1} = 0, \quad \text{when } 0 \leq k \leq m-2 \quad (2)$$

이러한 CSD 표현 방식은 각각의 2의 승수가 다른 부호를 가질 수 있기 때문에 일반적으로 2의 보수를 이용한 표현 방식보다 SPT항의 수가 적게 발생된다[8-9]. 따라서 CSD 방식을 이용하여 계수를 표현하면 2의 보수를 이용한 방식과 비교하여 보다 적은 SaA 연산 과정으로 동일한 곱셈을 수행할 수 있다.

실제 필터 설계에서는 하드웨어 복잡도를 줄이기 위하여 제한된 개수의 NZD 만을 이용하여 각 계수를 표현한다. NZD의 최대 허용 개수를  $J$ 라고 할 때,  $k$ 번째 NZD가 위치할 수 있는 비트영역  $Z_k (0 \leq k \leq J-1)$ 는 다음과 같다.

$$\begin{aligned} Z_k &= \{2^{2k}, 2^{2k+1}, \dots, 2^{(m-1)-2(J-1-k)}\} \\ &= 2^{2k} \cdot \{2^0, 2^1, 2^2, \dots, 2^{(m-1)-2(J-1)}\} \end{aligned} \quad (3)$$

식 (3)에서  $2^{2k}$  인자는  $k$ 값에 따라 결정된다. 그러나  $\{2^0, 2^1, 2^2, \dots, 2^{(m-1)-2(J-1)}\}$ 은  $k$ 에 관계없이 모든  $Z_k$ 에 대해 공통된 시프트 연산을 하는 부분이다. 따라서 그림 1과 같이 모든 NZD에 공통되는 연산은 섀선 박스와 같이 하드웨어를 공유하고, 하위 비트 영역을 시프트시켜 다음 영역을 얻기 위한 시프터는 이 점 섀선 박스와 같이 구현할 수 있다[9].

즉 그림 1은 고정된 시프터와 간단한 디지털 로직의 조합으로 이루어진 프로그래머블 멀티플렉서를 이용하여 시스템의 유연성을 높이고 동시에 하드웨어 면적을 최소화한 SaA 구조이다.

본 논문에서는 기존의 SaA 구조와 비교하였을 때 보다 간단한 하드웨어로 프로그래머블 FIR 필터를 구현하기 위한 개선된 SaA 구조를 제안한다. 일반적으로 FIR 필터의 코딩된 이웃하는 계수는 공통된 SPT항이 적어도 하나 이상 발생하게 된다. 제안하는 방법에서는 이웃하는 두 계수의 공통된 SPT항을 하나의 SaA를 이용하여 구현한다. 즉, 기존에는 두 계수의 연산에 각각  $J$ 개씩  $2J$ 개

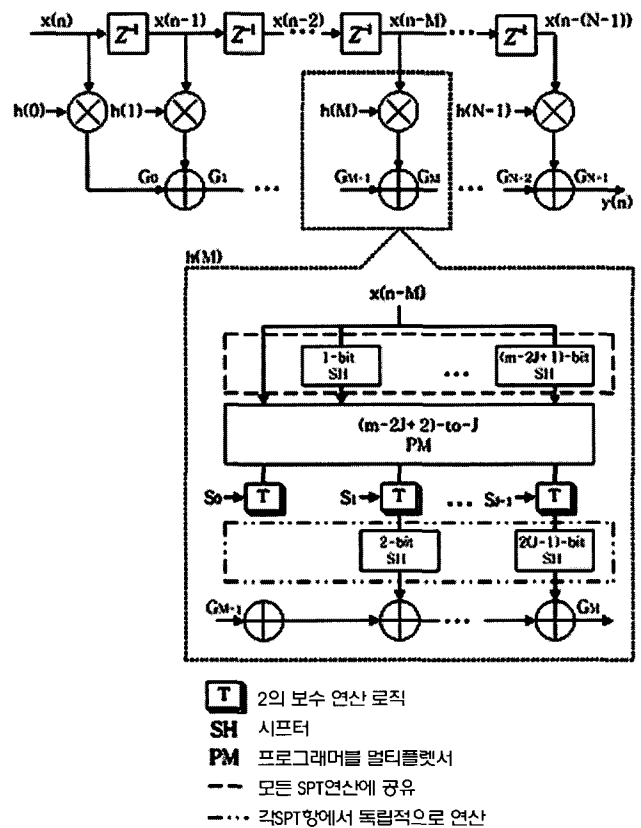


그림 1. 고정된 시프터와 프로그래머블 멀티플렉서를 이용한 SaA 구조  
Fig. 1. The SaA architecture using fixed shifters and programmable multiplexers

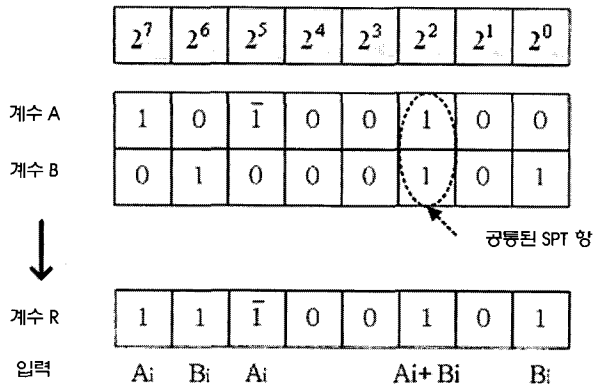


그림 2. 이웃하는 계수 A와 B의 coding 예  
 Fig. 2. A coding example of adjacent coefficients A and B.

의 SPT 항을 연산할 수 있는 하드웨어가 필요했지만, 제안된 방법에서는 이웃하는 계수 간에 공통된 SPT 항의 수에 비례하여 하드웨어 복잡도가 감소한다. 그림 2는 이웃하는 임의의 필터 계수 A, B의 CSD 코딩 예를 보여준다.  $A_i$ 와  $B_i$ 는 각각 계수 A와 B에 대한 입력을 나타낸다.

그림 2에서 R은 제안하는 구조를 적용하기 위하여 A와 B를 결합하여 새롭게 만든 계수이다. 계수 R에서 A에 의해 발생한 SPT 항에 대해서는 입력으로  $A_i$ 를 사용하고, 계수 B에 의해 발생한 SPT 항에 대해서는  $B_i$ 를 입력으로 사용하여 SaA 연산을 수행하게 된다. 이때 A와 B의 공통된 SPT 항에 의해 발생된 R의 SPT 항에 대해서는 입력으로  $A_i+B_i$ 를 사용함으로써 하나의 SaA 연산으로 처리할 수 있게 된다. 따라서 기존의 구조와 비교하여 두 계수의 공통된 SPT 항이 많이 발생할수록 하드웨어 복잡도를 줄일 수 있다.

표 1은 임의로 설계된 필터에서 이웃하는 계수들의 코딩 예를 보인 것이다. 두 계수  $h(11)$ ,  $h(12)$ 를 기존의 CSD 방식을 이용하여 표현하면 공통된 SPT 항이 발생되지 않으나, 식 (2)의 제약 없이 새롭게 계수를 표현하면,  $-2^7$ 이 두 계수에 공통적으로 나타난다. 예를 들어,  $J=2$ 일 때, 기존에는  $-191$ 의  $2^0$ 과  $-158$ 의  $2^1$ 을 표현할 수 없어 각각  $-192$ 와  $-160$ 으로 새롭게 양자화 되었다. 그러나, 제안하는 구조를 이용하면  $-2^7$ 이 공통되므로,  $h(11)$ 보다 오차가 크게 발생했던  $h(12)$ 에 SPT 항을 하나 더 할당하여  $-158$ 의 정확한

표 1. 제안된 SaA 구조에 적합한 계수 표현

Table 1. Coefficient representation for the proposed SaA architecture.

계수	CSD 표현 방식	제안된 구조에 적합한 표현 방식
$h(11)$	$-2^8 + 2^6 + 2^0$	$-2^7 - 2^6 + 2^0$
$h(12)$	$-2^7 - 2^5 + 2^1$	$-2^7 - 2^5 + 2^1$
$h(25)$	$-2^5 - 2^3 - 2^1$	$-2^5 - 2^3 - 2^1$
$h(26)$	$-2^5 - 2^3$	$-2^5 - 2^3$

표현이 가능하게 된다. 따라서 필터 계수의 연산 오차가 감소하게 된다. 계수  $h(25)$ 와  $h(26)$ 을 살펴보면 기존 SaA 구조에서는 각각 2개와 3개의 SPT 항을 연산할 수 있는 하드웨어 로직이 필요했지만, 제안하는 SaA 방식을 이용하면 2개의 공통된 SPT 항의 연산 하드웨어를 공유하여 두 계수의 연산에 단지 3개의 SPT 항을 연산할 수 있는 하드웨어가 필요하게 된다. 따라서 하드웨어 복잡도를 감소시킬 수 있다.

본 논문에서는 앞에서 서술한 필터 계수의 특성을 사용하여 그림 3과 같은 SaA 구조를 제안한다. 제안된 SaA 구조는 기존의 PM을 사용한 SaA 구조와 거의 유사하다. 단, 두 계수의 공통된 SPT 항 연산을 동일한 하드웨어로 수행하기 위해 두 계수에 곱해지는 각각의 입력 값을 더하기 위한 하나의 덧셈기만이 추가되었다.

제안된 SaA 구조는 두 계수 값을 동시에 연산하여 출력하므로, 각 계수에 대한 연산 결과는 일정한 경로로 출력되지 않는다. 직접형(Direct-form)으로 필터를 구현하면 각 SPT 항의 출력 경로를 고려하지 않아도 되므로 직접형 구조를 이용하여 그림 3과 같이 제안된 SaA 구조를 구현하였다. 한편, 전치형(transposed form)에서는 입력 데이터가 모두 동일하므로 입력을 더하는 로직은 필요

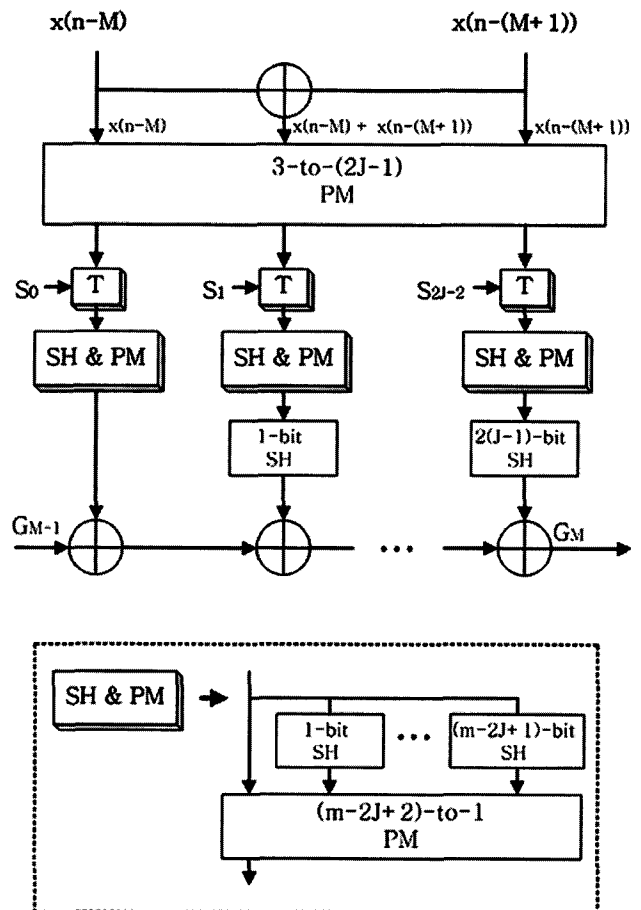


그림 3. 제안된 SaA 구조  
 Fig. 3. The proposed SaA architecture

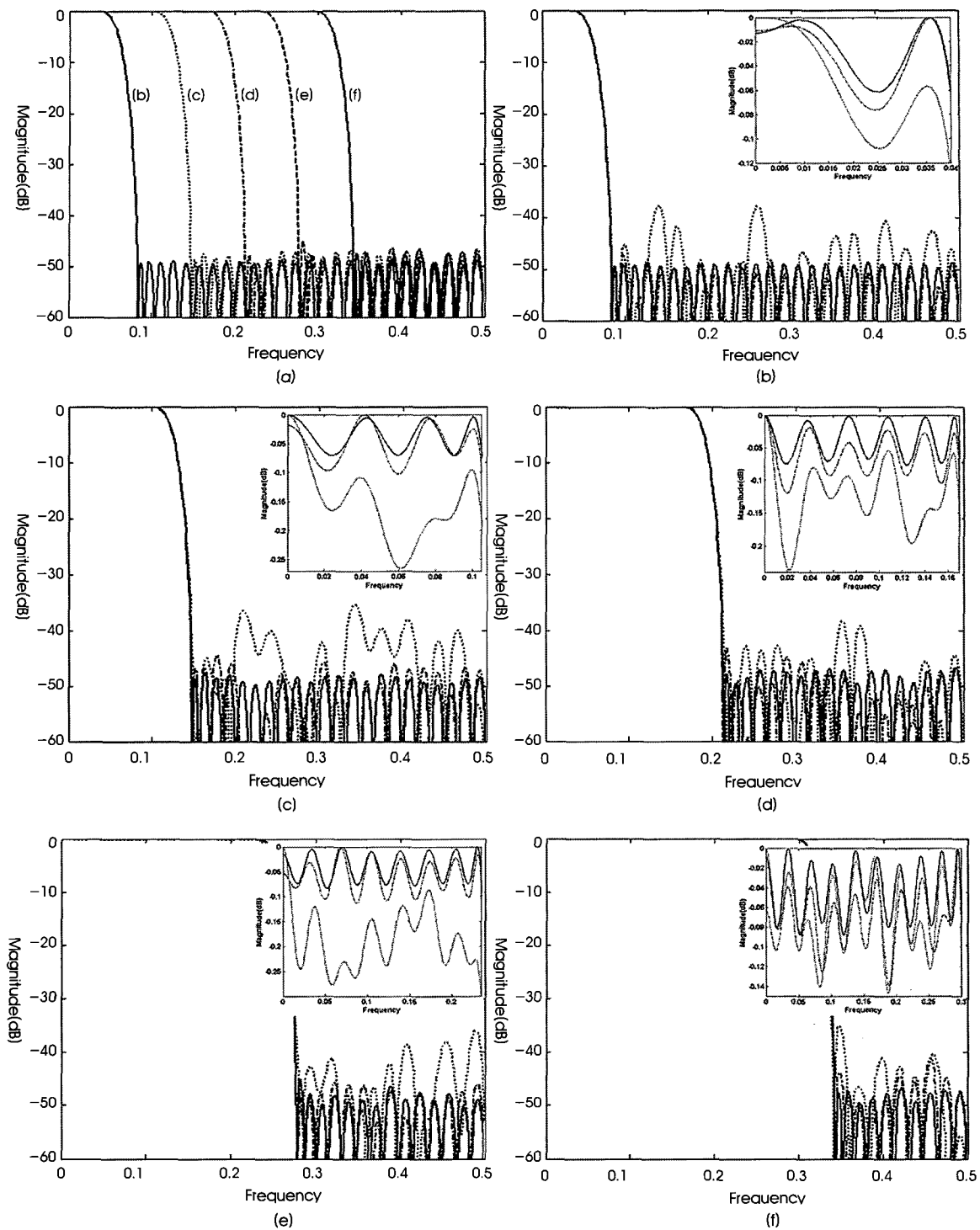


그림 4. 통과 대역폭의 변화에 따른 필터의 주파수 응답 특성 (N=60): (a) (b)-(f)에 사용된 필터의 주파수 특성, (b)  $f_s=0.04$ ,  $f_p=0.0825$ , (c)  $f_s=0.105$ ,  $f_p=0.1475$ , (d)  $f_s=0.17$ ,  $f_p=0.2125$  (e)  $f_s=0.235$ ,  $f_p=0.2775$ , (f)  $f_s=0.3$ ,  $f_p=0.3425$  ((b)-(f): ——— 곱셈기를 사용한 필터, ..... 기존의 PM을 사용한 SaA 구조의 필터, - - - 제안된 SaA 구조의 필터)

Fig. 4. The frequency response of filters with different pass-band widths (N=60): (a) frequency spectrum of filters using (b)-(f), (b)  $f_s=0.04$ ,  $f_p=0.0825$ , (c)  $f_s=0.105$ ,  $f_p=0.1475$ , (d)  $f_s=0.17$ ,  $f_p=0.2125$  (e)  $f_s=0.235$ ,  $f_p=0.2775$ , (f)  $f_s=0.3$ ,  $f_p=0.3425$  ((b)-(f): ——— Filter using multipliers, ..... Filter with the conventional SaA architecture using PM, - - - Filter with the proposed SaA architecture)

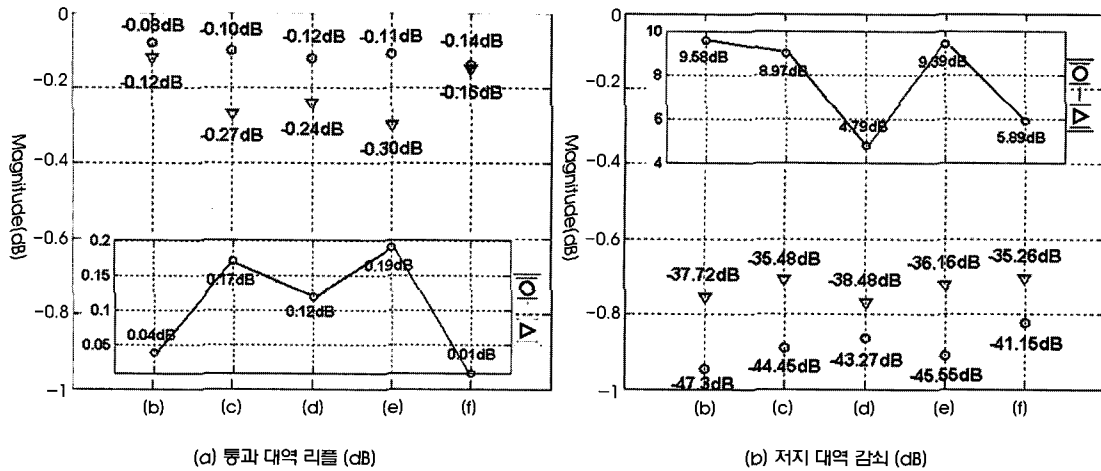


그림 5. 제안된 SaA 구조를 갖는 필터의 주파수 응답 개선  
 ▽: 기존의 PM을 사용한 SaA 구조 ○: 제안된 SaA 구조  
**Fig. 5.** Improvement of frequency response of the filters with the proposed SaA architecture  
 ▽: Conventional SaA architecture using PM. ○: Proposed SaA architecture.

하지 않다. 그러나, 각 SaA 연산 값이 출력되는 위치를 지정하기 위해 SaA 연산 횟수만큼의 PM이 필요하다. 따라서, 직접형 구조의 필터보다 부가 로직이 많이 요구되고 그에 따르는 조절이 복잡하므로 제안된 구조는 전치형으로 구현하기 어렵다.

### III. 설계 및 구현

본 논문에서는 제안하는 SaA 구조의 성능과 하드웨어 효율성을 검증하기 위해 제안하는 구조를 FIR 필터에 적용하여 설계하고, 구현된 필터의 성능을 평가하였다. 제안된 SaA 구조의 특징은 다양한 필터 사양에 모두 적합하다는 것이다. 따라서, 여러 필터 사양에 대해 제안된 SaA 구조의 성능을 검증하였다. 이때, 설계된 모든 필터는 선형 위상 특성을 유지하도록 하였으며, 성능 평가는 두 가지 관점에서 하였다. 먼저 기존의 PM을 기반으로 하는 SaA 구조를 이용한 필터와 제안하는 SaA 구조의 필터가 동일한 개수의 SPT항으로 연산하는 경우에 각 구조에 대한 주파수 응답 특성을 비교한다. 다음에 기존의 PM을 사용한 SaA 구조의 필터와 제안된 SaA 구조를 갖는 필터의 성능이 유사한 경우에 각 구조의 구현을 위해 요구되는 하드웨어 면적을 비교한다.

표 3. 여러 구조를 갖는 필터들의 하드웨어 복잡도 비교

**Table 3.** Comparison of hardware complexity between various filters with different architectures.

	디지털 로직 게이트 수	SPT항 개수
곱셈기 사용	131,972	-
기존 방식	109,526	168
제안된 방식	87,242	116
기존방식⇒제안된 방식	20.35%	30.95%
곱셈기 ⇒제안된 방식	33.89%	-

#### A. 제안된 필터의 성능 평가

그림 4(a)와 같이 다른 주파수 응답 특성을 갖는 5개의 필터를 설계하고, 기존의 PM을 사용한 SaA 구조로 구현된 필터의 각 계수에는 1, 2, 3개의 NZD를 각각 24, 24, 12개의 계수에 허용하여 전체 필터 계수 연산에 108개의 NZD를 이용하였다. 그리고, 동일한 연산 하드웨어를 공유하는 방식에서는 2, 3, 4, 5, 6개의 NZD를 각각 8, 6, 6, 8, 2개의 계수 쌍에 허용하여 110개의 NZD를 이용하여 필터를 구현하였다.

그림 4는 각각 곱셈기를 이용한 필터, 기존의 PM을 사용한 SaA 구조 기반의 필터, 제안된 SaA 구조의 필터의 주파수 응답 특성을 비교한 것이다. 각 필터의 통과 대역 리플과 저지 대역 감쇠 수치는 그림 5에 나타내었다.

그림 5에서  $\| \nabla |H| \ominus \circ \|$ 는 두 수치의 차이를 나타낸 것이다. 그림에서 보듯이 연산 하드웨어를 공유하는 구조의 필터는 기존의 PM을 이용한 SaA 구조의 필터에 비해 저지 대역 리플은 최대 9.68dB, 통과 대역의 리플은 0.01~0.19dB 개선된다.

#### B. FIR 필터의 구현 및 하드웨어 면적 비교

본 논문에서 제안하는 SaA 구조의 하드웨어 효율성을 검증하기 위해 VHDL로 설계하여 Synplify pro 7.0(Synplicity co.)을 이용하여 합성하였다. 구현을 위해서는 ISE 6.0(Xilinx)을 이용하여 구현에 필요한 게이트 수를 2-입력 낸드 게이트수로 계산하였다. 입력의 비트 수는 8-비트로 하였다.

성능 평가는 그림 4의 (a)와 같이 대역폭의 변화에 따른 다양한 필터 사양에 대해 수행하였다. 필터 길이는 60이고 양자화 비트 수는 12로 하였다. 유사한 필터 성능을 유지하기 위해 기존의 PM을 이용한 SaA 기반의 필터와 제안된 SaA 구조를 이용한 필터의 저

지 대역 감쇠 최대 값의 차이를 1dB미만으로 허용하였다. 반면, 곱셈기를 이용한 필터와는 1~3dB정도 차이를 허용하였다.

표 3에서 보듯이 제안된 SaA 구조를 이용하여 필터를 구현하면 기존의 PM을 기반으로 하는 SaA 구조를 이용한 필터에 비해 20.35%, 그리고 곱셈기를 사용하는 필터 구조에 비해 33.89% 하드웨어 면적이 감소하게 된다. 따라서, 본 논문에서 제안하는 동일한 연산 하드웨어를 공유하는 SaA 구조를 적용하여 구현한 필터는 기존의 PM을 기반으로 하는 SaA 구조를 이용한 필터와 곱셈기를 사용하는 필터에 비해 성능의 큰 저하없이 하드웨어 면적을 현저히 감소시킬 수 있다.

#### IV. 결 론

본 논문에서는 빠른 데이터 처리와 높은 성능을 기반으로 하는 하드웨어 효율적인 프로그래머블 FIR 필터를 설계하기 위해 효율적인 필터 구조를 제안하고, 제안된 구조의 필터를 설계 및 구현하여 그 성능을 검증하였다.

제안된 SaA 구조는 기존의 PM을 사용한 SaA 구조를 개선한 구조로 이득하는 계수에서 동일한 SPT항의 연산은 하드웨어를 공유할 수 있도록 구현하였다. 기존의 PM을 사용한 SaA 구조와 제안된 SaA 구조의 필터를 동일한 SPT항을 연산할 수 있는 하드웨어를 이용하여 구현했을 때, 하드웨어 공유로 인해 더 많은 개수의 SPT항으로 연산하는 효과를 얻을 수 있었다. 그리고, 동일한 성능을 갖도록 설계된 필터에서는 전체 필터 계수의 연산에 요구되는 NZD 개수를 감소시켜 기존의 PM을 기반으로 하는 SaA 구조보다 하드웨어 면적을 더욱 감소시킬 수 있었다.

필터의 주파수 응답 특성을 비교해 본 결과, 불필요한 중복 연산을 제거한 제안된 SaA 구조가 기존의 PM을 이용한 SaA 구조보다 필터의 성능을 최대 10dB 정도 향상시켰고, 하드웨어 면적도 20%이상 현저히 감소시키는 개선 효과를 보였다. 따라서, 본 논문에서 제안하는 개선된 SaA 구조를 이용한 필터가 기존의 PM을 기반으로 하는 SaA 구조를 이용한 필터보다 필터 성능과 시스템의 하드웨어 효율성을 향상시킬 수 있음을 확인할 수 있었다.

#### 참고문헌

[1] T. K. Song, "Spectral doppler and 2-D color-flow system," in *Proc. 19th KOSOMBE Spring tutorial*, 1998.  
 [2] B. H. Kim and T. K. Song, "Multiple transmit focusing method with modified orthogonal golay codes for ultrasound imaging," *J. Biomed. Eng. Res.*, vol. 24, no. 3, pp.217-231, 2003.  
 [3] R. G. Pridham and R. A. Mucci, "Digital interpolation

beamforming for low-pass and bandpass signal," in *Proc. IEEE*, 1979, vol. 67, no. 6, pp. 904-919.  
 [4] G. Galati and P. Lombardi, "Design and evaluation of an adaptive MTI filter," *IEEE Trans. on Aerospace and Electronic Systems*, vol. AES-14, no. 6, Nov. 1979.  
 [5] D. E. Bowyer, P.K. Rajasekaran, and W. W. Gebhart, "Adaptive clutter filtering using autogressive spectral estimation," *IEEE Trans. on Aerospace and Electronic Systems*, vol. AES-15, no. 4, July 1979.  
 [6] S. Bjaerum, H. Torp, and K. Kristoffersen, "Clutter filter design for ultrasound color flow imaging," *IEEE Trans. on Ultrasonics, Ferro. and Freq. Control*, vol. 49, Issue 2, pp. 204-216, Feb 2002.  
 [7] L. Dongning, "Minimum number of adders for implementing a multiplier and its application to the design of multiplierless digital filters," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, Vol.42, Issue. 7, July 1995.  
 [8] Q. Zhao and Y. Tadokoro, "A simple design of FIR filters with powers-of-two coefficients," *IEEE Transactions on Circuits and Systems*, Vol. 35, Issue. 5, May 1988.  
 [9] Woo Jin Oh and Yong Hoon Lee, "Implementation of programmable multiplierless FIR filters with powers-of-two coefficients," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, Vol. 42, Issue. 8, Aug. 1995.  
 [10] H. Samuelli, "An improved search algorithm for the design of multiplierless FIR filters with powers-of-two coefficients," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 104&1047, July 1989.  
 [11] K. S. Yeung and S. C. Chan, "Multiplier-less FIR digital filters using programmable sum-of-power-of-two(SOPOT) coefficients," in *Proc. 2002 IEEE Int. Conference on Field-Programmable Technology*, Hong Kong, Dec 2002, pp. 78-84.  
 [12] J. B. Evans, Y. C. Lim, and B. Liu, "A high speed programmable digital FIR filter," in *Proc. 1990 IEEE Int. Conf. Acoust., Speech, Signal Process.*, Albuquerque, NM, Apr. 1990, pp. 969-971.  
 [13] K. Khoo, A. Kwentus, and A. N. Willson, Jr., "An efficient 175 MHz programmable FIR digital filter," in *Proc. 1993 IEEE Int. Symp. Circuits Syst.*, Chicago, IL, May 1993, pp. 72-75.  
 [14] M. Ishikawa et al., "Automatic layout synthesis for FIR filters using a silicon compiler," in *Proc. 1990 IEEE Int. Symp. Circuits Syst.*, May 1990, New Orleans, LA, pp. 2588-2591.  
 [15] T. Yoshino, R. Jain, P. T. Yang, H. Davis, W. Gass, and A. H. Shah, "A 100-MHz @-tap FIR digital filter in 0.8- $\mu$ m BiCMOS gate array," *IEEE J. Solid-State Circuits*, vol. 25, pp. 1494-1501, Dec. 1990.  
 [16] R. Jain, P. T. Yang, and T. Yoshino, "FIRGEN: A computer-aided design system for high performance FIR filter integrated circuits," *IEEE Trans. Signal Process.*, vol. 39, pp. 1655-1668, July 1991.  
 [17] J. B. Evans, "An efficient FIR filter architecture," in *Proc. 1993 IEEE Int. Symp. Circuits Syst.*, Chicago, May 1993, pp. 627-430.