

인공 공통패턴을 사용한 CSD 적용의 선형위상 FIR 필터 구조

정회원 장영범*, 이혜림*

A CSD linear phase FIR filter architecture using artificial common sub-expression

Young-Beom Jang*, Hye-Rim Lee* *Regular Members*

요 약

Digital IF(Intermediate Frequency) 처리단과 같은 고속과 저전력을 요구하는 필터에서 덧셈기만을 사용하여 CSD(Canonical Signed Digit)형의 필터계수들을 구현하는 구조가 널리 연구되고 있다. 본 논문에서는 선형위상 FIR(Finite Impulse Response) 필터의 CSD형 필터계수들을 최소의 덧셈으로 구현할 수 있는 아키텍처를 제안한다. 1과 -1로 이루어진 필터계수 표에서 공통패턴을 공유함으로써 덧셈의 수를 줄이는 방법이 이미 연구되었다. 본 논문은 비트 shift, 비트 add, 비트 반전을 통하여 인공의 공통패턴을 만들어서 이미 존재하는 공통패턴에 합류시킴으로서 덧셈의 수를 더욱 줄일 수 있는 방법을 제안한다. CDMA 이동통신 단말기의 IF단에 사용되는 사양의 디지털 필터를 73탭의 CSD형 계수로 구현하여 9.2%의 덧셈 감소의 효과가 있음을 보였다.

ABSTRACT

Multiplierless CSD(Canonical Signed Digit) filter architectures are widely investigated for the application of high-speed and low-power wireless communication systems like digital IF(Intermediate Frequency) filters. In this paper, we propose a CSD architecture with minimum adders for linear phase FIR(Finite Impulse Response) filters. Reduction of the number of adders through sharing the common sub-expression has been used in fast filter application. We propose new techniques to make artificial common sub-expression through the bit shift, bit add, and bit complement. As an example, we utilize the 73 taps FIR filters with specification in the first version of CDMA cellular. It is shown that the proposed CSD format architecture achieves 9.2% adder reduction comparison to the conventional common sub-expression sharing architecture.

I. 서론

이동통신 단말기 등에서 고속의 선형위상 FIR(Finite Impulse Response) 필터를 현재의 CMOS 기술을 사용하여 저전력으로 수행할 수 있는 필터 구조가 요구되어지고 있다. 이와 같은 고속과 저전력을 요구하는 선형위상 FIR 필터에서, 2진수로 표현되는 필터계수에서 2의 보수형 계수보다 1의 수

가 적게 사용되는 CSD(Canonical Signed Digit)형 계수에 대한 구조가 널리 연구되고 있다. 특히 최근에는 IF(Intermediate Frequency) 대역의 디지털 처리를 위하여 DDC(Digital Down Converter)와 같은 고속의 데시메이션 필터가 상용화되고 있다. 이와같은 고속저전력 필터에서는 덧셈기(뺄셈기포함)와 지연소지만을 사용하여 곱셈을 처리하는 구조가 장점을 갖게 되며, 따라서 1의 수가 적게 사용되는

* 이화여자대학교 정보통신학과(ybjang@mm.ewha.ac.kr)

논문번호 : 00341-0828, 접수일자 : 2000년 8월 28일

* 본 연구는 교육부 BK21 지원에 의해 수행되었습니다.

CSD 구현이 매력적이다. 다음의 식 $\sum_{i=0}^{N-1} a_i 2^{-i}$ 이 나타내는 수 $a_0.a_1a_2 \dots a_{N-1}$ 에서 $a_0=0$ 또는 -1 이고, $i>0$ 에 대하여는 $a_i=0$ 또는 1 이면 2의 보수형이라고 한다. 그리고 수 $b_0.b_1b_2 \dots b_{N-1}$ 이 각각의 b_i 에 대하여 $0, +1, -1$ 이고 두 개의 연속되는 b_i 가 nonzero가 아니면 CSD 형이라고 한다. 모든 n-bit의 2의 보수형의 수는 n-bit의 CSD 형의 수로 나타낼 수 있으며, 변형시키는 방법도 [1][2]에 기술되었다. 2의 보수형의 수에 비하여 CSD 형의 수는 $(N+1)/2$ 이상의 nonzero bit를 갖지 않는 큰 장점을 갖고 있으며, 이는 덧셈기를 사용하여 구현할 때 덧셈기의 수를 줄일 수 있음을 의미한다. 필터계수의 곱셈은 덧셈과 shift를 사용하여 구현될 수 있는데, shift는 Hard wired 로직으로 구현할 때에는 비용이 거의 들지 않는다. 따라서 덧셈의 수를 줄이는데 연구의 초점이 맞추어져 있다. CSD형의 필터계수는 IIR(Infinite Impulse Response)과 FIR 모두에 사용될 수 있으나, 본 논문은 가장 많이 사용되는 선형위상 FIR 필터의 CSD형 계수를 사용한 구조에 관하여 다루기로 한다.

본 논문의 II장에서는 선형위상 FIR 필터에서 공통패턴이 얼마나 덧셈의 수를 감소시키는지 기존 공통패턴 공유 방식을 통하여 알아본다. III장에서는 임의의 비트를 shift 시키거나, add 시키거나, 반전시켜서 인공의 공통패턴을 만드는 방법을 제안하고, 덧셈의 수 감소 효과를 알아본다. IV장에서는 1차 버전의 CDMA IF 필터를 CSD형의 계수로 설계하여 본 논문의 아키텍처를 적용함으로써 덧셈의 수 감소효과를 알아본다.

II. 선형위상 FIR 필터에서의 공통패턴에 의한 덧셈의 수 감소효과

이 장에서는 기존의 공통패턴 공유방식을 통하여, 선형위상 FIR 필터에서 덧셈의 수가 얼마나 감소되는지를 알아본다. 앞으로는 표기를 편리하게 하기 위해서 -1 은 *로 표기하기로 한다. 먼저 표 1과 같은 CSD형으로 계수가 표기된 8탭의 선형위상 FIR 필터를 살펴보기로 하자. 표 1에서는 0의 수는 표시하지 않았다. 임의의 필터에서 1 또는 *이 총 m 개가 있을 때, 구현을 위하여 $m-1$ 개의 덧셈이 필요함이 알려져 있다.^[4] 따라서 표 1에서는 $m=24$ 이므로 23개의 덧셈이 필요하다.

이러한 CSD형의 필터구현에서 덧셈의 수를 더욱

표 1. 8탭 선형위상 FIR 필터의 CSD계수

	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}
h_0							1		*
h_1							1		1
h_2					*		1		*
h_3	1		1		1		*		1
h_4	1		1		1		*		1
h_5					*		1		*
h_6							1		1
h_7							1		*

감소시키기 위하여 공통패턴을 공유하여 구현하는 방식이 연구되었다.^{[3][4][5]} 표 1의 선형위상 FIR 필터에서, h_0, h_1, h_2, h_3 의 계수는 h_7, h_6, h_5, h_4 의 계수와 각각 대칭이다. 이와 같이 선형위상 FIR 필터는 자연스럽게 공통패턴이 2개 이상 존재한다. [2]에서 다음과 같은 필터 표기법이 제시되었다.

$$y[n] = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} a_{ij} (x_{n-i} \gg j) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} a_{ij} X_{ij} \quad (1)$$

위의 식에서 a_{ij} 는 1, 0, 또는 -1 이므로 결국 출력은 입력신호 x 의 shift 되고 지연된 여러 개의 합으로 표현될 수 있음을 의미한다. X_{ij} 에서 행은 지연을 나타내고 열은 shift를 나타낸다. 따라서 i 행, j 열의 ± 1 은 $\pm x_1[-i] \gg j$ 로 나타낸다. 그러나 0번째 행과 0번째 열은 i 와 j 를 붙이지 않고 x_1 으로 정의하며, 기준 점으로 정한다. 이와 같은 표기방법을 사용하여 대칭계수들을 공통패턴 x_2, x_3, x_4, x_5 로 정의하면 다음과 같이 표현된다.

$$\begin{aligned} x_2 &= x_1 - x_1 \gg 2 \\ x_3 &= x_1 + x_1 \gg 2 \\ x_4 &= -x_1 + x_1 \gg 2 - x_1 \gg 4 \\ x_5 &= x_1 + x_1 \gg 2 + x_1 \gg 4 - x_1 \gg 6 + x_1 \gg 8 \end{aligned} \quad (2)$$

위의 공통패턴을 만드는데 8개의 덧셈이 필요하며, 이를 사용하여 출력신호를 다음과 같이 구할 수 있다.

$$\begin{aligned} y &= x_2 \gg 6 + x_3[-1] \gg 6 + x_4[-2] \gg 4 + x_5[-3] \\ &+ x_5[-4] + x_4[-5] \gg 4 + x_3[-6] \gg 6 + x_2[-7] \gg 6 \end{aligned} \quad (3)$$

위의 식에서 보듯이 8개항의 합을 구하기 위해서 7개의 덧셈이 필요함을 알 수 있다. 선형위상 FIR 필터의 계수대칭 공통패턴을 이용하는 경우에 덧셈

의 수는 다음 식과 같이 된다.

$$N = m/2 - 1 + t/2 \quad (4)$$

위의 식에서 N 은 사용된 덧셈의 수, t 는 탭 수를 나타낸다. 따라서 표 1의 경우에는 $m=24$, $t=8$ 이므로 위의 식을 이용하면 덧셈의 수가 15가 됨을 알 수 있다. 위의 식이 나타내고있는 공통패턴을 공유하기 위하여 Transposed Direct form을 이용하여 구현하면 그림 1과 같다. 그림 1에서 모든 $-n$ 은 2^{-n} 을 의미하며, n 비트 right shift의 동작이다. 식 (2)를 그림 1로 구현할 때 다음을 주의한다. 즉 x_1 자체가 -1 의 shift를 내재하고 있는 기준 점이므로 x_1 을 구현할 때에는 구현 식보다 -1 을 더 shift해야 한다. 앞으로의 모든 그림에서도 같은 표기법을 사용한다.

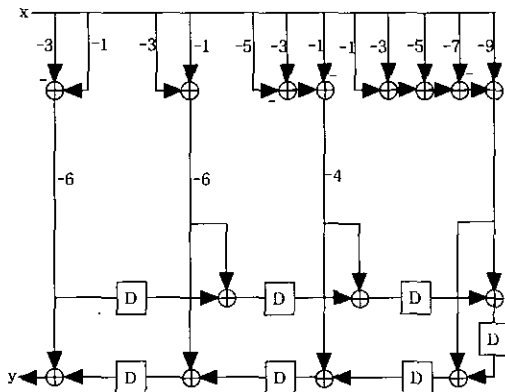


그림 1. 선형위상 대칭만을 이용한 CSD 구조

그림 1에서는 단지 선형위상 FIR 필터계수의 대칭에서 생겨나는 공통패턴만을 이용하였으나, 표 1에서 볼 수 있듯이 계수대칭에서 생겨나는 공통패턴 이외에 더 많은 공통패턴이 있을 수 있다. 표 1에서 $10 * 01$ 의 공통패턴이 4개가 있고, 101 의 공통패턴이 4개가 있다. 여기에서 $10 * 01$ 과 $* 010 *$ 과 같이 보수로 되어있는 패턴은 공통패턴이다. 모든 공통패턴을 구현하는 식은 다음과 같다.

$$\begin{aligned} x_2 &= x_1 - x_1 \gg 2 \\ x_3 &= x_1 + x_1 \gg 2 : \text{공통패턴} \\ x_4 &= -x_1 + x_1 \gg 2 - x_1 \gg 4 : \text{공통패턴} \\ x_5 &= x_3 - x_4 \gg 4 \end{aligned} \quad (5)$$

위의 식과 같이 공통패턴을 만드는데 5개의 덧셈이 필요하고, 출력신호는 식 (3)과 같으므로 7개의 덧셈이 필요하여 총 12개의 덧셈으로 3개가 감소되었음을 알 수 있다. 이를 구현하면 그림 2와 같다. 덧셈 감소의 수를 정량적으로 표시하기 위하여 q 를 공통패턴의 크기, s 를 반의 필터계수(즉, 10탭의 선형위상 FIR 필터에서는 5탭만의 계수)에서 공통패턴의 발생수로 정의하자. 위의 정의에서 크기는 임의의 필터계수에서 0을 제외한 1 또는 -1 의 수를 의미한다. 즉 어떤 패턴이 $10 * 01$ 이면 $q=3$ 이다. q 크기의 공통패턴이 s 개 사용되고 있는 경우에 감소되는 덧셈의 수는 $-(q-1)s + (q-1)$ 로 나타낼 수 있다. $q=3$ 크기의 공통패턴이 6개가 사용되면 $-(q-1)s = -(2)(6) = -12$ 개의 감소가 발생하며, 공통패턴을 만드는데 $q-1=2$ 개의 덧셈이 필요하므로 총 -10 개의 감소가 발생한다.

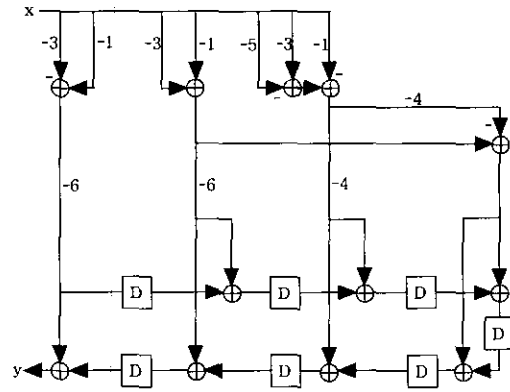


그림 2. 선형위상 대칭과 공통패턴을 이용한 CSD 구조

III. 인공 공통패턴의 생성방법과 덧셈의 감소

3.1 비트 shift에 의한 인공 공통패턴의 생성

II장에서는 필터계수를 나타내는 표 1과 같은 1과 *의 도표에서 공통패턴을 찾아서 공유하는 방법으로 덧셈의 수를 감소시켰다. 이 장에서는 좀 더 능동적인 방법으로서 기존 공통패턴과 비교하여 1비트만 shift시키면 공통패턴과 같아지는 패턴들을 효율적으로 구현하는 방법을 제안한다. 표 2와 같은 14탭의 선형위상 FIR 필터를 살펴보기로 하자. h_7 부터 h_{13} 까지의 계수의 반은 대칭이므로 표에서 생략하였다. 표 2에서 h_0 인 $100 * 001$ 을 공통패턴으로 정의할 경우에 $h_1, h_2, h_3, h_4, h_5, h_6$ 에서 각각

한 비트 만을 좌측이나 우측 방향으로 shift시키면 공통패턴이 됨을 알 수 있다. 즉 h_1 에서는 $100 * 0001$ 에서 맨 우측의 1을 좌로 shift시키면 공통패턴이 된다. 이와 같은 방법으로 $h_1, h_2, h_3, h_4, h_5, h_6$ 은 공통패턴을 사용하여 다음과 같이 나타낼 수 있다.

$$\begin{aligned}
 & 100*0001 \quad 100*010 \quad 1000*01 \\
 & = 100*0010 \quad = 100*001 \quad = 100*001 \\
 & - 0000\ 0001 \quad + 0000\ 001 \quad + 0000\ 100 \\
 & \\
 & 10*0001 \quad 010*001 \quad 1000*001 \\
 & = 100*001 \quad = 100*001 \quad = 0100*001 \\
 & - 0001\ 000 \quad - 0100\ 000 \quad + 01000\ 000
 \end{aligned} \tag{6}$$

위의 식과 같이 인공적으로 공통패턴을 만들어서 공유시키면 덧셈의 수가 얼마나 감소되는지를 보기로 하자. 먼저 기존의 방법을 사용해 보자. 표 3은 $m=42$ 이고 공통패턴이 없으므로 (4)의 식을 사용하여 덧셈의 수를 구해보면 27이 된다.

인공 공통패턴을 만들 경우에는, $100 * 001$ 의 공통패턴 만드는데 2개가 사용되며 공통패턴이 14개의 계수에 모두 적용되므로 14개가 사용되며 식 (6)의 더해주거나 빼주는 양 때문에 6개의 계수에 6개가 사용되므로 $2+14+6=22$ 개의 덧셈이 필요하므로 6개의 감소를 달성하였다. 아키텍처를 만들기 위하여 다음의 식으로 나타낼 수 있다.

$$\begin{aligned}
 x_2 &= x_1 - x_1 \gg 3 + x_1 \gg 6 \\
 x_3 &= x_2 - x_1 \gg 7, \quad x_4 = x_2 + x_1 \gg 6 \\
 x_5 &= x_2 + x_1 \gg 4, \quad x_6 = x_2 - x_1 \gg 3 \\
 x_7 &= x_2 - x_1 \gg 1, \quad x_8 = x_2 + x_1
 \end{aligned} \tag{7}$$

$$\begin{aligned}
 y &= x_2 \gg 1 + x_3[-1] \gg 1 + x_4[-2] \gg 1 + x_5[-3] \gg 1 \\
 & + x_6[-4] \gg 1 + x_7[-5] \gg 1 + x_8[-6] \gg 1 \\
 & + \text{symmetry terms}
 \end{aligned}$$

표 2. 14탭 선형위상 FIR 필터의 CSD 계수

	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}
h0		1			*			1	
h1		1			*				1
h2		1			*		1		
h3		1				*		1	
h4		1		*				1	
h5			1		*			1	
h6	1				*			1	

위의 식에서, $x_2, x_3, x_4, x_5, x_6, x_7, x_8$ 의 공통패턴을 만드는데 8개의 덧셈이 필요하고 출력 신호를 만드는데 13개의 덧셈이 필요하므로 총 21개의 덧셈이 요구됨을 식을 통하여 알 수 있다. 식 (7)에서 symmetry terms라고 표기한 것은 대칭의 계수에 대하여 반복하는 것을 생략하기 위해서이다. 즉, h_7 부터는 h_{13} 까지 계수는 대칭이므로 생략하였다. 식 (7)을 Transposed Direct form으로 구성하면 그림 3과 같다.

3.2 비트 add나 비트 반전에 의한 인공 공통패턴의 생성

이 절에서는 임의의 패턴에 1 비트를 add하거나 1 비트를 반전시키면 공통패턴과 같아지는 경우의 효율적인 방법을 제안한다.

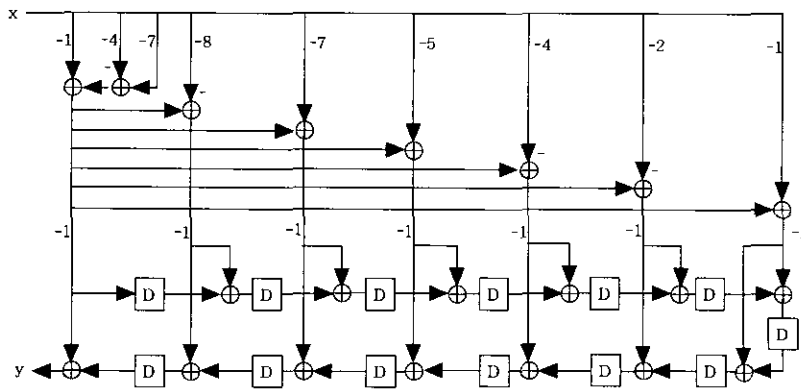


그림 3. 비트 shift 인공패턴을 사용한 CSD 구조

역시 다음 표 3과 같은 10탭의 선형위상 FIR 필터를 살펴보기로 하자. h_5 부터 h_9 까지의 계수는 대칭이므로 표에서 생략하였다. 표 3에서 h_0 를 공통패턴으로 정의하면 h_1, h_2 에서 각각 한 비트만을 더하거나 빼면 공통패턴과 같아지고 h_3, h_4 에서는 각각 한 비트만을 반전시키면 공통패턴과 같아짐을 알 수 있다. 따라서 h_1, h_2, h_3, h_4 는 공통패턴을 사용하여 다음 식과 같이 나타낼 수 있다.

표 3. 10탭 선형위상 FIR 필터의 CSD 계수

	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}
h_0	1		*			1		*	
h_1	()		*			1		*	
h_2	1		()			1		*	
h_3	1		*			(*)		*	
h_4	1		*			1		(1)	

$$\begin{aligned}
 &00*0010* && 100\ 0010* \\
 &= 10*0010* && = 10*0010* \\
 &-100\ 00000 && +001\ 00000 \\
 & && (8) \\
 &10*00*0* && 10*00101 \\
 &= 10*001\ 0* && = 10*0010* \\
 &-000\ 01000 && +000\ 00010
 \end{aligned}$$

기존의 방법을 사용하여 덧셈의 수를 계산해 보자. 공통패턴이 없고 $m=36$ 이므로 식 (4)를 사용하여 구하면 덧셈의 수가 22가 필요하다. 이번에는 비트 add나 비트 반전의 인공 공통패턴을 다음 식과 같이 정의해 보자.

$$\begin{aligned}
 x_2 &= x_1 - x_1 \gg 2 + x_1 \gg 5 - x_1 \gg 7 \\
 x_3 &= x_2 - x_1, & x_4 &= x_2 + x_1 \gg 2 \\
 x_5 &= x_2 - x_1 \gg 4, & x_6 &= x_2 + x_1 \gg 6 \\
 y &= x_2 + x_3[-1] + x_4[-2] + x_5[-3] + x_6[-4] \\
 &+ \text{symmetry terms}
 \end{aligned}
 \tag{9}$$

위의 식에서, x_2, x_3, x_4, x_5, x_6 의 공통패턴을 만든데 7개의 덧셈이 필요하고, 출력 신호를 만드는 데 10개의 항이 있으므로 9개의 덧셈이 필요하여 총 16개의 덧셈이 필요함을 식을 통하여 알 수 있다. 식 (9)를 Transposed Direct form으로 구성하면 그림 4와 같다.

IV. 실험 및 고찰

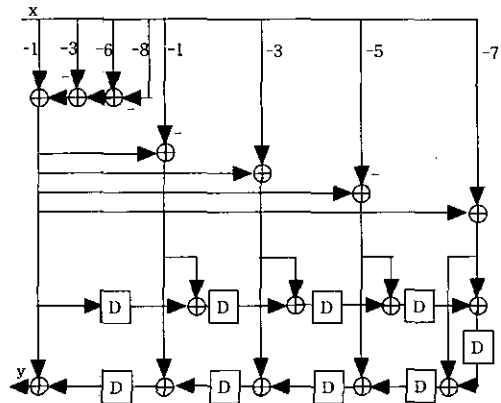


그림 4. 비트 add와 비트 반전의 인공패턴을 사용한 CSD 구조

예제로서 CDMA 이동통신 단말기의 IF단 사양의 디지털 필터를 본 논문이 제안하는 CSD형 구조로 구현한다. 필터의 샘플링 주파수는 19.6608MHz이며, 통과대역의 주파수와 감쇠량은 각각 630KHz와 0.1dB로 하였다. 저지대역의 주파수와 감쇠량은 각각 1.2288MHz와 -40dB로 하였다. 이와 같은 사양을 만족하도록 COSSAP FIR 필터설계 도구를 사용하여 설계한 결과, 73탭의 선형위상 필터계수를 얻었다.

본 시뮬레이션에서는 대칭이 되는 72탭만을 조사하기로 한다. 즉 h_{36} 을 제외하고 대칭의 72개의 필터계수 중에서 36개를 24비트 precision의 CSD형의 계수로 표 4와 같이 나타냈다. 표 4와 5에서는 -1을 n 으로 표기하였다. 표 4를 조사하면 1 또는 n 의 수가 총 $m=458$ 개이므로 공통패턴을 전혀 사용하지 않으면 $m-1=457$ 개의 덧셈이 필요하며, 선형위상의 계수대칭에 의한 공통패턴만을 이용하면 식 (4)에 의해서 $229-1+36=264$ 개의 덧셈이 필요하다.

[3]의 공통패턴 방식으로 덧셈의 수를 구해보자. 표 4에서 맨 좌측의 열은 h_0 부터 h_{35} 의 계수를 나타내며, 맨 위의 행은 2^{-1} 부터 2^{-24} 의 비트 중에서 실제 사용되는 2^{-4} 부터 2^{-24} 을 표시하였다. 이 표에 회색으로 공통패턴을 표시하였다. 길이가 3인 공통패턴의 종류와 사용 빈도는 다음과 같다.

- 공통패턴(1) : n0101(6번)
- 공통패턴(2) : n010n(8번)
- 공통패턴(3) : n0n01(7번)
- 공통패턴(4) : 10101(9번)
- 공통패턴(5) : 100010n(3번)
- 공통패턴(6) : 10n0001(4번)

또한 길이가 4인 공통패턴의 종류와 사용 빈도는 다음과 같다.

- 공통패턴(7) : 10n010n(2번)
- 공통패턴(8) : n00n0010n(2번)

길이가 3인 공통패턴을 만드는데 2개의 덧셈이 필요하며 6가지 종류가 있으므로 이 공통패턴들을 만드는데 12개의 덧셈이 필요하다. 길이가 4인 공통패턴을 만드는데 3개의 덧셈이 필요하고 2가지의 종류가 있으므로 6개의 덧셈이 필요하다. 따라서 공통패턴을 만드는 데에는 총 18개의 덧셈이 사용된다. 표 4의 맨 우측 열에 그 계수를 구현하는데 필요한 덧셈의 수를 표시하였으며 그 합은 179이다. 따라서 총 18+179-1=196개의 덧셈이 사용된다.

본 논문에서 제안한 인공 공통패턴을 표 5에 표시하였다. 표 4에서 사용된 것과 같은 공통패턴은 그대로 회색으로 나타내었다. 각각의 계수마다 본 논문의 방식을 적용하여 새로 인공 공통패턴이 된 것

들을 ()로 표기하였으며, 적용해도 덧셈이 감소되지 않는 필터계수에는 적용하지 않았다. 각각의 계수에 적용된 방식은 구체적으로 다음과 같다.

- h1 : n00101= 공통패턴(1)의 비트 shift
- h8 : 100100n0n= 공통패턴(8)의 비트 반전
- h12 : n000n0n= 공통패턴(5)의 비트 반전
- h13 : 101001= 공통패턴(4)비트 shift
- h15 : n00n01= 공통패턴(3)의 비트 shift
- h18 : n0n001= 공통패턴(3)의 비트 shift
- h19 : 10n000n= 공통패턴(7)의 비트 add
- h20 : n010001= 공통패턴(7)의 비트 add
- h21 : n0010n= 공통패턴(2)의 비트 shift
- h23 : n000101= 공통패턴(5)의 비트 반전
- h26 : n0n00n= 공통패턴(4)의 비트 shift
- h29 : 100100n0n= 공통패턴(8)의 비트 반전
- h34 : 10n00n= 공통패턴(1)의 비트 shift

새로 공통패턴으로 정의하여 인공 공통패턴에 합

표 4. 기존 방식에 의한 공통패턴 (73탭 FIR 필터)

	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	#	
0						n	n																5
1						n		1	1										n				5
2																							3
3						n	n			n	1									1	6		6
4										n									n				4
5																							3
6																							3
7						1	n		n	n					n	n							7
8						1			1	1						1		n	n				7
9																					n		4
10																							4
11											n	1											5
12																			n	n			5
13						1	1				1	1	1										7
14																							4
15																		n	n			1	5
16																							4
17																							3
18						n	n			1		1											6
19													1	n					n				5
20						n				1										1			6
21						n				n	1												6
22																							4
23						n				1										1	1		6
24																				n			5
25																					n		3
26																							5
27						1	n													n	n		6
28																					1	n	5
29																							6
30																							6
31						1															1	n	6
32																							5
33																					n		6
34						1	n																5
35						1	n																5

표 5. 제안된 방식에 의한 공통패턴(73탭 FIR 필터)

	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	#		
0						n	n																5	
1							(n		1	1)									n				-1	
2																							3	
3										n	n				(n	1						1)	-1	
4																n			n				4	
5																							3	
6																							3	
7										(1	n		n	n)				n	n				7	
8										1		(1		1		1		n	n				-2	
9																							4	
10																							4	
11																							5	
12																			(n		n	n)	-1	
13																				(1	1		1)	-1
14																				n			4	
15																				(n	n		1)	-1
16																							4	
17																							3	
18										(n	n		1)		1								-1	
19															(1	n			n)				-1	
20																				(n	1		1)	-1
21												(n	1	n)									-1	
22																							4	
23																							6	
24																							5	
25																							3	
26																							5	
27																							6	
28																							5	
29																							6	
30																							6	
31																							5	
32																							6	
33																							5	
34																							5	
35																							5	

류시킨 계수들은 다음과 같다.

- h30 : 10n0010n= 새로 공통패턴(9)로 정의
- h3 : n0100001= 공통패턴(9)의 비트 add
- h7 : 10n00n0n= 공통패턴(9)의 비트 반전

h30의 공통패턴을 새로 정의하는데 3개의 덧셈이 사용되므로 총 공통패턴을 만드는데 기존의 18개와 합하면 21개의 덧셈이 사용된다. 36개의 계수 중에서 위와 같이 16개의 계수에서 21개의 덧셈 감소를 이룰 수 있었다. 표 5의 맨 우측 열에 표 4에 비하여 감소된 덧셈의 수만을 기록하였으며 이 열의 합은 179-21=158이다. 따라서 본 논문의 방식을 사용하는 경우 총 21+158-1=178의 덧셈이 필요하다. 기존의 방식에^[3] 비하여 총 18개의 덧셈을 감소시켰음을 알 수 있다. 지금까지의 결과를 표 6에 요약하였다. 예제의 73탭의 선형위상 FIR 필터의 경우, 기존의 방식들 중에서 가장 덧셈을 적게 사용하는 공통패턴 방식에 비하여 본 논문의 구조는 9.2%의 덧셈 감소 효과를 얻었다.

표 6. 기존의 방식들과의 덧셈의 수 비교

구분	덧셈의 수	%
직접구현 (m=458)	457	233.2
계수대칭만을 이용한 방식	264	134.7
기존의 공통패턴 방식 ^[4]	196	100
인공의 공통패턴 방식	178	90.8

V. 결론

선형위상 필터의 계수들을 CSD형의 계수로 구현할 때, 인공의 공통패턴을 만들어서 덧셈의 수를 최소화하는 구조를 제안하였다. 인공의 공통패턴을 만드는 방법으로서 비트 shift, 비트 add, 그리고 비트 반전의 방법을 구체적으로 제시하였다. 본 논문이 제안하는 인공의 공통패턴 구조는 선형위상 FIR 필터뿐만 아니라 일반 필터에서도 사용될 수 있다. CDMA 이동통신 단말기의 IF단의 필터를 본 논문이 제안하는 구조로 구현한 결과, 기존의 공통패턴을 사용한 CSD 구조보다 9.2%의 덧셈 감소의 효과를 얻을 수 있었다.

참고 문헌

[1] R. W. Reitwiesner, "Binary arithmetic," in

Advances in Computers, New York: Academic, vol. 1, pp. 231-308, 1966.

- [2] K. Hwang, *Computer Arithmetic: Principles, Architecture, and Design*, New York: Wiley, 1979.
- [3] R. I. Hartley, "Subexpression sharing in filters using canonic signed digit multipliers," *IEEE Trans. Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 43, No. 10, pp. 677-688, Oct. 1996.
- [4] M. Yagyu, A. Nishihara, and N. Fujii, "Fast FIR digital filter structures using minimal number of adders and its application to filter design," *IEICE Trans. Fundamentals of Electronics Communications & Computer Sciences*, vol. E79-A No. 8, pp. 1120-1129, Aug. 1996.
- [5] H. Samuelli, "An improved search algorithm for the design of multiplierless FIR filters with powers-of two coefficients," *IEEE Trans. Circuits and Systems*, vol. 36, No. 7, Jul. 1989.
- [6] Y. W. Kim, Y. M. Yang, J. T. Yoo, and S. W. Kim, "Approximate processing for low-power digital filtering using variable canonic signed digit coefficients," *IEE Electronics Letters* vol. 36, No. 1, pp. 11-13, 6th Jan. 2000.
- [7] R. Hawley, T. Lin, and H. Samuelli, "A silicon compiler for high-speed CMOS multirate FIR digital filters," *Proc. IEEE International Symposium on Circuits and Systems*, San Diego, CA, pp. 1348-1351, May 1992.

장 영 범(Young-Beom Jang)

정회원



1981년 2월 : 연세대학교 전기 공학과 졸업, 공학사

1990년 1월 : Polytechnic University 전기공학과 졸업, 공학석사

1994년 1월 : Polytechnic University 전기공학과 졸업, 공학박사

1981년 2월~1999년 12월 : 삼성전자 System LSI 사업부 수석연구원

1999년 12월~현재 : 이화여자대학교 정보통신학과

조교수

<주관심 분야> 통신신호처리, 음성/오디오 신호처리

이 혜 림(Hye-Rim Lee)

1997년 3월~현재 : 이화여자대학교 정보통신학과