

# 임의의 인수를 갖는 Cascaded Integrator-Comb 데시메이션 필터의 Multi-rate Non-recursive 아키텍처

정희원 장영범\*

## Multi-rate Non-recursive Architecture for Cascaded Integrator-Comb Decimation Filters with an Arbitrary Factor

Young-Beom Jang\* *Regular Member*

요 약

본 논문에서는, 임의의 인수를 갖는 CIC(Cascaded Integrator-Comb) 데시메이션 필터의 Multi-rate Non-recursive 아키텍처를 제안한다. CIC 필터는 다운 샘플러와 결합되어 데시메이션을 수행하게되며, 고속처리와 저전력을 요구하는 이동통신시스템의 데시메이션 필터로 상용화되고 있다. 널리 연구되는 CIC 필터의 Non-recursive 구조는 다운 샘플러의 위치를 입력신호의 방향으로 위치바꿈 함으로서 여러 개의 스테이지로 분리하여 스테이지마다 처리 속도를 다르게 하여 저전력의 구현이 가능하도록 하는 구조이다. 그러나, 이러한 아키텍처의 구성은 데시메이션 인수가 2의 승수인 경우로만 제한되었었다. 본 논문에서는 2의 승수가 아닌, 소인수로 분해되는 임의의 데시메이션 인수의 경우에도 스테이지의 분리를 통한 Multi-rate의 Non-recursive CIC 아키텍처가 구성될 수 있음을 보인다. 또한 기존의 아키텍처와의 전력소모를 비교한다.

ABSTRACT

In this paper, multi-rate non-recursive architecture for CIC(Cascaded Integrator-Comb) decimation filters with an arbitrary factor is proposed. The CIC filters are widely used in high speed wireless communication systems since they have multiplier-less and multi-rate low-power structure. Even conventional non-recursive CIC structure is multi-rate, this architecture can be structured only in case of M-th power-of-two decimation factor. This paper proposes that multi-rate non-recursive CIC architecture can be structured with an any decimation factor of product form. Power consumption of the proposed architecture is compared with that of the conventional non-recursive architecture.

### 1. 서론

이동통신 단말기 등에서 점점 더 고속의 데시메이션을 저전력으로 수행할 수 있는 프로세서가 요구되어지고 있으며, 현재의 반도체 집적기술을 사용하여 고속의 데시메이션을 저전력으로 구현하는 필

터 구조가 많이 연구되고 있다. 상용화된 데시메이션 필터들은 몇가지 종류의 필터들의 직렬연결로 구성되어, 각각의 특성에 의해서 전체적인 데시메이션 필터의 특성이 결정된다. 일반적으로 데시메이션의 블록도는 그림 1과 같이 데시메이션 필터와 다운 샘플러로 구성되며, 데시메이션 인수를  $N$ 으로 표시하였다. 최근에는 이동 통신 시스템의 IF(Inter-

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

논문번호: 00156-0505, 접수일자: 2000년 5월 5일

※ 본 연구는 교육부 BK21 지원에 의해 수행되었습니다.

mediate Frequency) 단을 디지털 처리하기 위하여 고속처리와 저전력을 요구하는 데시메이션 필터가 널리 연구되고 있는 추세이다. 이와 같은 요구를 충족하기 위하여 몇 단의 필터가 직렬연결되어 사용되며, 보통 CIC 데시메이션 필터가 첫단에 사용되고 Halfband 데시메이션 필터가 두 번째 단에 사용된다. 그 뒤에 고차의 프로그래머블 필터가 사용되는 경우도 있다. 이 가운데에 CIC 필터는 E. B. Hogenauer에 의하여 제안되었다<sup>[1]</sup>. CIC 필터가 널리 연구되는 까닭은 1)곱셈기가 필요하지 않으며, 2)규칙적인 구조를 갖고 있으며, 3)처리속도의 변환이 가능하기 때문이다. 이러한 장점들 때문에 고속과 저전력을 요구하는 데시메이션을 위하여 CIC 필터가 적합하다. 이러한 CIC 필터의 전달함수는 다음과 같다.

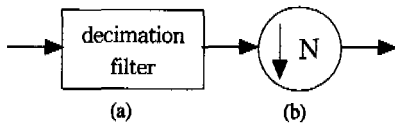


그림 1. 데시메이션의 블록도  
(a)데시메이션 필터, (b)다운 샘플러

$$H(z) = \left\{ \frac{1-z^{-N}}{1-z^{-1}} \right\}^k : \text{Recursive} \quad (1)$$

$$= \left\{ \sum_{i=0}^{N-1} z^{-i} \right\}^k : \text{Non-recursive}$$

위의 식에서 앞의 것은 Recursive 전달함수이며, 뒤의 것은 Non-recursive 전달함수이다. 이 식에서 N은 데시메이션 인수이고 k는 필터의 차수로 정의한다. Recursive 구조와 Non-recursive 구조의 장단점을 먼저 기술하기로 한다.

Recursive 전달함수 경우에, 전달함수를 Integrator와 Comb의 직렬연결로 분리시키면 그림 2(a)와 같으며, 각각의 Integrator 필터와 Comb 필터의 전달함수는 다음과 같다.

$$H_I(z) = \left\{ \frac{1}{1-z^{-1}} \right\}^k, H_C(z^N) = (1-z^{-N})^k \quad (2)$$

이 식에서 보여지듯이 Comb 필터의 전달함수가  $z^N$ 의 함수로 나타내지므로 N의 다운샘플러와의 위치 바꿈이 가능해진다.  $z^N$ 의 함수로 표현되는 전달함수와 N의 다운 샘플러의 위치를 바꿀 때에, 뒤의 전달함수는 z의 함수가 된다. 따라서 그림 2(b)와 같이 Integrator 필터는 빠른 샘플링 주파수로 처리

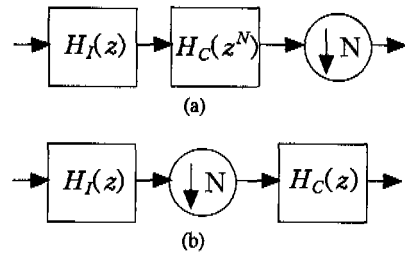


그림 2. Recursive CIC 필터  
(a)필터의 분리(Integrator와 Comb),  
(b)Comb 필터와 다운 샘플러 위치바꿈

되고, Comb 필터는 상대적으로 느린 주파수로 처리되는 구조가 만들어진다. 이와 같은 Recursive의 특성개선과 구조에 대하여 여러 가지의 연구결과가 활발히 발표되고 있다. 특성개선으로서는, 전달대역의 Droop를 감소시켜 필터의 특성을 날카롭게 하는 기술들이 제안되었다<sup>[2][3]</sup>. [2]에서는 여러 개의 CIC 필터의 조합으로 Droop를 개선하였으며, [3]에서는 2차 필터를 CIC 필터의 다음 단에 연결하여 특성을 개선하였다. 고속처리구조에 대한 연구로서는, 전달함수의 분자 다항식과 분모 다항식에 새로운 다항식을 추가하여 인수 분해함으로써 스테이지를 분리하는 방안이 제안되었다<sup>[4]</sup>. 이렇게 함으로써 빠른 샘플링 주파수에 동작하는 첫 번째 스테이지는 Polyphase로 병렬처리하고, 두 번째 스테이지는 일반적인 Recursive 구조로 구성할 수 있게 된다. 지금까지 기술한 내용은 Recursive 구조의 장점이고 단점으로는 레지스터의 크기가 매우 커진다는 점이다. 이에 대한 연구는 [1]에 기술되었다. 즉, Integrator 필터에서의 Recursive 루프 때문에 모든 노드들의 비트 크기를 키워야 Overflow에 의한 데이터의 손실을 방지할 수 있다. 입력신호의 비트 크기를 B라고 하면 모든 레지스터들의 크기  $B_{max}$ 는  $B_{max} = \lceil k \log_2 N + B - 1 \rceil$ 로 계산된다<sup>[1]</sup>. 따라서 입력이 16 비트이고, 필터차수 k가 4이고, 데시메이션 인수 N이 25인 경우에  $B_{max}$ 는 34 비트가 된다. 처리 후에 16 비트의 출력을 필요로 할 경우 19 비트를 버리게 된다. 이와 같이 Recursive 구조에서는 Overflow를 방지하기 위해 처리 비트의 크기를 키워야 하는 단점이 있다.

Non-recursive의 전달함수를 살펴보기로 하자. 식 (1)을 바로 구성할 경우 여러 가지 불리한 점이 있다. 덧셈의 수를 비교해보면, Recursive는 2k개가 필요하며 Non-recursive는 N-1에 비례하여 k(N-1)개가 필요하다. 지연소자의 수를 비교해보면, Recursive

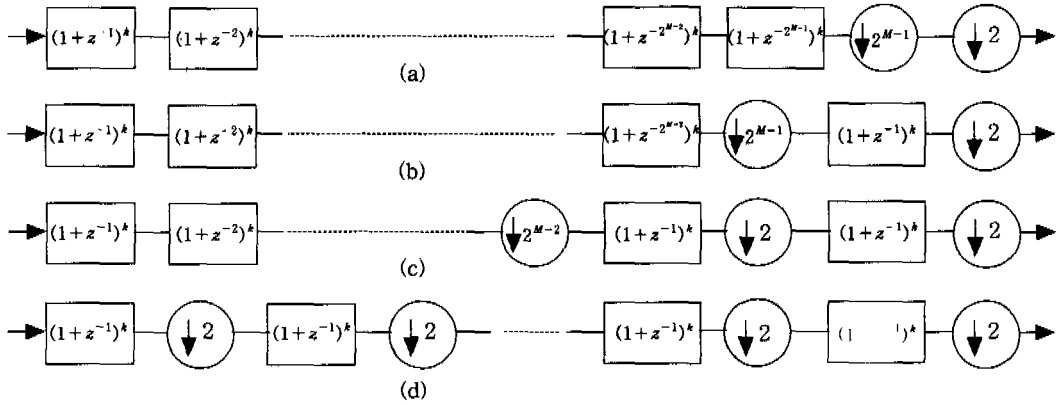


그림 3. 데시메이션 인수가 2의 M제곱인 경우의 FIR CIC 필터 구조과정

의 경우 데시메이션 인수  $N$ 과 무관하며 오직  $2k$ 의 지연소자만이 필요한데 반하여, Non-recursive는 지연소자가  $N-1$ 에 비례하여  $k(N-1)$ 개가 필요하다. 또한 Non-recursive의 전달함수를 바로 구성할 경우 예외 다운 샘플러와의 위치 바꿈이 안되어 모든 블록을 빠른 샘플링 주파수로 처리해야만 한다. 이러한 동작속도의 단점 때문에 식 (1)의 Non-recursive 전달함수를 직접구현하지 않고, 전달함수의 인수분해를 통하여 스테이지 분리와 다운 샘플러의 위치변경이 가능한 구조가 제안되었다<sup>[5]</sup>. 그러나, 이러한 스테이지 분리를 통한 효율적인 Non-recursive 구조는 데시메이션 인수가 2의 승수인 경우에만 아키텍처를 만드는 방법이 개발되었다. 이와 같이 데시메이션 인수가 2의 승수라는 가정 하에, [6]에서는 [5]의 구조에서의 여러 개의 스테이지 중에서 빠른 동작속도를 필요로 하는 앞단의 스테이지들을 Polyphase로 병렬 처리하는 방식이 제안되었다.

본 논문은 Non-recursive 구조의 결함들이 되어왔던 데시메이션 인수의 제한을 완화하기 위하여, 소인수로 분해되는 어떤 데시메이션 인수도 스테이지의 분리와 Multi-rate의 처리가 가능함을 보이고 그 아키텍처를 제안한다. II장에서는 기존의 Non-recursive 아키텍처를 기술하고, III장에서는 임의의 인수를 갖는 Multi-rate Non-recursive 아키텍처를 제안한다. 그리고 IV장에서 아키텍처 구성 예제와 기존의 아키텍처와의 전력소모를 비교한다.

## II. 인수가 $2^M$ 인 CIC 필터의 Muti-rate Non-recursive 아키텍처

데시메이션 인수가  $2^M$ 으로 나타내지는 경우에는

Non-recursive 전달함수가 다음의 식 (3)과 같다.

$$H(z) = \left\{ \sum_{i=0}^{N-1} z^{-i} \right\}^k \tag{3}$$

$$= \left\{ \sum_{i=0}^{2^M-1} z^{-i} \right\}^k = \prod_{i=0}^{M-1} (1+z^{-2^i})^k$$

즉, Non-recursive의 전달함수가 위의 식과 같이 다항식의 곱의 형태로 표현될 수 있다.  $N=32$ 로 데시메이션하는 경우를 예를 들어보면,  $M$ 은 5가되며 5개의 전달함수의 곱으로 인수분해 된다. 이 경우에 첫 번째와 두 번째 전달함수는 각각  $z$ 의 함수와  $z^2$ 의 함수이다. 그리고 세 번째, 네 번째, 다섯 번째의 전달함수는 각각  $z^4$ 의 함수,  $z^8$ 의 함수,  $z^{16}$ 의 함수가 된다. 여기에서 각각의 전달함수는 Multi-rate로 동작하게 되므로 스테이지라고 부르기로 한다. 즉 스테이지가 증가함에 따라 각 스테이지의 전달함수는  $z^{2^i}$ 의 함수로 나타내진다. 이와 같은 관계를 그림 3(a)에 나타내었다. Multi-rate로 스테이지를 분리하기 위하여 먼저 그림 3(a)에서 마지막에 위치한 데시메이션 인수를  $2^{M-1}$ 과 2로 분리한다. 그러면  $M$ 번째(마지막번째) 전달함수가  $z^{2^{M-1}}$ 의 함수이므로  $2^{M-1}$ 의 다운 샘플러와 위치 바꿈이 가능해지므로 (b)와 같이 된다. 이와 같은 방법으로 다시  $2^{M-1}$ 의 다운 샘플러는  $2^{M-2}$ 과 2로 분리시킨 후,  $M-1$ 번째 전달함수가  $z^{2^{M-2}}$ 의 함수이므로 역시  $2^{M-1}$ 의 다운 샘플러와 위치 바꿈이 가능하다. 이와 같은 과정을 통하여 (c)가 얻어진다. 여기에서 보여지듯이 다운 샘플러와 위치를 바꿀 때, 모든 전달함수는  $z$ 의 함수가 됨을 알 수 있다. 따라서 모든 스테이지의 전달함수는  $z$ 의 함수가 된다. 이와 같은 과정을 첫 번째 전달함수까지 반복하면 최종으로 (d)의 그

림이 얻어진다. (d)에서는 첫 번째 스테이지의 샘플링 주파수가 가장 빠르고 각 스테이지마다 처리속도가 2분의 1로 감소되어  $M$  번째 스테이지에서는  $2^{-M}$ 으로 가장 감소된다. 이와 같은 (d)의 구조에서는 입력 단에 가까운 스테이지일수록 빠른 샘플링 주파수로 동작하므로 반도체 등으로 구현할 때에는 병렬처리의 방법이 적용될 수 있다. 지금까지 살펴본 이 구조는 데시메이션 인수 즉 다운 샘플링의 크기가 2의 승수인 경우에만 아키텍처의 구성이 가능하였다. 즉 이 구조로 다운 샘플링이 가능한 크기는 2, 4, 8, 16, 32, 64, 128 등으로서 다운 샘플링의 크기가 커질수록 제한이 커짐을 알 수 있다. 32와 128 사이에서는 오직 64의 다운 샘플링 만이 가능하다.

**III. 임의의 인수를 갖는 CIC 필터의 Multi-rate Non-recursive 아키텍처**

이 장에서는 임의의 인수를 갖는 CIC 필터에 대하여 Multi-rate Non-recursive 구조를 제안한다. 먼저 임의의 수는  $2^P 3^Q 5^R 7^S 11^T \dots$ 과 같이 소인수 분해된다. 여기에서 2, 3, 5, 7, 11 등은 소수라 하며 자신 이외의 인수를 갖지 않는 수이다. 예를 들어 180은 2, 2, 3, 3, 5로 소인수 분해된다. 수식을 보기 편하게 하기 위하여 앞으로 데시메이션 인수  $N$ 이  $2^P 3^Q 5^R$ 으로 소인수 분해되는 경우를 사용하여 식을 전개시킬 것이다. 2, 3, 5 이외의 소수가 더 포함된 경우에도 쉽게 식이 확장될 수 있다. 먼저 그림 4와 같이 데시메이션 인수가 2, 3, 5로 소인수 분해되는 경우에 다운 샘플러를  $2^P$ ,  $3^Q$ ,  $5^R$ 으로 분해한다. 그리고 그림 4의 전달함수를 다음과 같이 분해한다.

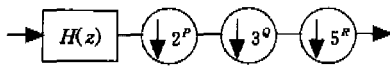


그림 4. 인수가  $N=2^P 3^Q 5^R$ 인 경우의 데시메이션 필터와 다운 샘플러의 블록도

$$\begin{aligned}
 H(z) &= \left\{ \frac{1-z^{-N}}{1-z^{-1}} \right\}^k = \left\{ \frac{1-z^{-2^P 3^Q 5^R}}{1-z^{-1}} \right\}^k \\
 &= \left\{ \frac{1-z^{-2^P}}{1-z^{-1}} \right\}^k \left\{ \frac{1-z^{-2^P 3^Q}}{1-z^{-2^P}} \right\}^k \\
 &= H_1(z) T_1(z^{2^P})
 \end{aligned} \tag{4}$$

위의 식에서  $T_1$ 의 전달함수는  $z^{2^P}$ 의 함수이므로 그림 4의  $2^P$ 의 다운 샘플러와 위치 바꿈이 된다. 이와 같이 위치를 바꾼 후에는  $T_1$ 의 전달함수는  $z$ 의 함수가 되어 그림 5(a)와 같이 된다. 이제  $T_1$ 의 전달함수를 다시 다음과 같이 분해한다.

$$\begin{aligned}
 T_1(z) &= \left\{ \frac{1-z^{-3^Q}}{1-z^{-1}} \right\}^k \\
 &= \left\{ \frac{1-z^{-3^Q}}{1-z^{-1}} \right\}^k \left\{ \frac{1-z^{-3^Q}}{1-z^{-3^Q}} \right\}^k \\
 &= H_2(z) T_2(z^{3^Q})
 \end{aligned} \tag{5}$$

위의 식에서도  $T_2$ 의 전달함수는  $z^{3^Q}$ 의 함수이므로 그림 5(a)의  $3^Q$ 의 다운 샘플러와 위치 바꿈이 되고, 위치 바꿈한 후에는  $z$ 에 대한 함수가 된다. 전달함수  $T_2(z)$ 는 더 이상의 소수로 분해될 수 없으므로 다음과 같이  $H_3(z)$ 로 정의한다.

$$T_2(z) = \left\{ \frac{1-z^{-5^R}}{1-z^{-1}} \right\}^k = H_3(z) \tag{6}$$

1단계로 같은 소수의 집단끼리 다운 샘플러를 위치바꿈한 블록도는 그림 5(b)와 같다. 이 그림과 같이 3개의 소인수 집단으로 다운 샘플러가 표현되는 경우에는 3개의 데시메이션 필터로 먼저 분해한다.

2단계로 이와 같이 1단계에서 분해한 3단의 데시메이션 필터를 각단마다 각각  $P$ ,  $Q$ ,  $R$ 의 스테이지로 분리하여야 최종의 저전력의 구조가 얻어진다. 먼저 소인수  $P$ 의 데시메이션 필터를 스테이지 분리하기 위하여 다음과 같이 전달함수를 분리한다.

$$\begin{aligned}
 H_1(z) &= \left\{ \frac{1-z^{-2^P}}{1-z^{-1}} \right\}^k = \left\{ \frac{1-z^{-2}}{1-z^{-1}} \right\}^k \left\{ \frac{1-z^{-(2)2^{P-1}}}{1-z^{-2}} \right\}^k \\
 &= H_{11}(z) T_{11}(z^2)
 \end{aligned}$$

$$\begin{aligned}
 T_{11}(z) &= \left\{ \frac{1-z^{-2^{P-1}}}{1-z^{-1}} \right\}^k = \left\{ \frac{1-z^{-2}}{1-z^{-1}} \right\}^k \left\{ \frac{1-z^{-(2)2^{P-2}}}{1-z^{-2}} \right\}^k \\
 &= H_{12}(z) T_{12}(z^2)
 \end{aligned}$$

$$\begin{aligned}
 &\vdots \\
 T_{1,P-1}(z) &= \left\{ \frac{1-z^{-2}}{1-z^{-1}} \right\}^k = H_{1P}(z)
 \end{aligned} \tag{7}$$

위의 식에서와 같이 전달함수를 두 개의 다항식으로 분리하여 인위적으로 두 번째의 다항식이  $z^2$ 의 함수가 되게 함으로서, 2의 다운 샘플러와 위치 바꿈이 된다. 이러한 동작을  $P$  번 반복하면 그림 6(a)와 같은  $P$  개의 스테이지로 분리가 된다. 두 번

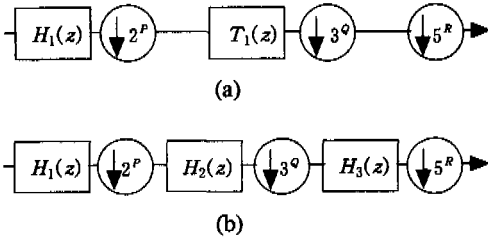


그림 5. (a) 인수  $2^p$ 를 위치바꿈한 후의 블록도  
(b) 인수  $3^Q$ 도 위치바꿈한 후의 최종의 블록도

계 데시메이션 필터도 다음 식과 같이 같은 방법으로 전달함수를 분해한다.

$$\begin{aligned}
 H_2(z) &= \left\{ \frac{1-z^{-3^Q}}{1-z^{-1}} \right\}^k = \left\{ \frac{1-z^{-3}}{1-z^{-1}} \right\}^k \left\{ \frac{1-z^{-(3)3^{Q-1}}}{1-z^{-3}} \right\}^k \\
 &= H_{21}(z) T_{21}(z^3) \\
 T_{21}(z) &= \left\{ \frac{1-z^{-3^{Q-1}}}{1-z^{-1}} \right\}^k = \left\{ \frac{1-z^{-3}}{1-z^{-1}} \right\}^k \left\{ \frac{1-z^{-(3)3^{Q-2}}}{1-z^{-3}} \right\}^k \\
 &= H_{22}(z) T_{22}(z^3) \\
 &\vdots \\
 T_{2,Q-1}(z) &= \left\{ \frac{1-z^{-3}}{1-z^{-1}} \right\}^k = H_{2Q}(z) \quad (8)
 \end{aligned}$$

위의 식에서와 같이 3의 다운 샘플러를 연속적으로 위치 바꿈하면 그림 6(b)가 얻어진다. 마지막 5의 인수들도 같은 방법으로 다음 식과 같이 분리한다.

$$\begin{aligned}
 H_3(z) &= \left\{ \frac{1-z^{-5^R}}{1-z^{-1}} \right\}^k = \left\{ \frac{1-z^{-5}}{1-z^{-1}} \right\}^k \left\{ \frac{1-z^{-(5)5^{R-1}}}{1-z^{-5}} \right\}^k \\
 &= H_{31}(z) T_{31}(z^5) \\
 T_{31}(z) &= \left\{ \frac{1-z^{-5^{R-1}}}{1-z^{-1}} \right\}^k = \left\{ \frac{1-z^{-5}}{1-z^{-1}} \right\}^k \left\{ \frac{1-z^{-(5)5^{R-2}}}{1-z^{-5}} \right\}^k \\
 &= H_{32}(z) T_{32}(z^5) \\
 &\vdots \\
 T_{3,R-1}(z) &= \left\{ \frac{1-z^{-5}}{1-z^{-1}} \right\}^k = H_{3Q}(z) \quad (9)
 \end{aligned}$$

위의 식에서도 5의 인수를 연속적으로 분리해 내어서, 그림 6(c)가 얻어진다. 즉, 첫 단의 데시메이션은  $P$ 개의 스테이지로 분리되고, 두 번째와 세 번째는 각각  $Q$ 와  $R$ 의 스테이지로 분리됨을 알 수 있다. 따라서 총 스테이지의 수는  $P+Q+R$ 이 된다. 이는 데시메이션 인수  $2^p 3^Q 5^R$ 의 소인수의 총수와 같음을 알 수 있다. 소인수의 총수와 스테이지의 총수가 같게 된다.

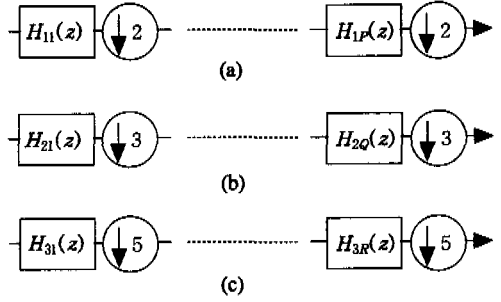


그림 6. (a) 인수  $2^p$ 의 스테이지 분리후 블록도  
(b) 인수  $3^Q$ 의 스테이지 분리후 블록도  
(c) 인수  $5^R$ 의 스테이지 분리후 블록도

이제 마지막 3단계로서 위의 그림 6의 각각의 스테이지를 서브스테이지로 분리한다. 그림 6에서 필터의 이름에 사용된 두 개의 숫자 중에서 뒤의 숫자는 순서만을 나타낼 뿐, 앞의 숫자가 같으면 모두 같은 필터이다. 따라서  $H_{11}(z)$ ,  $H_{21}(z)$ ,  $H_{31}(z)$ 의 3개의 필터 구조만을 구성하면 된다. 이 세 필터는 다음과 같이 Non-recursive 식으로 나타낼 수 있다.

$$\begin{aligned}
 H_{11}(z) &= \left\{ \frac{1-z^{-2}}{1-z^{-1}} \right\}^k = (1+z^{-1})^k \\
 H_{21}(z) &= \left\{ \frac{1-z^{-3}}{1-z^{-1}} \right\}^k = (1+z^{-1}+z^{-2})^k \\
 H_{31}(z) &= \left\{ \frac{1-z^{-5}}{1-z^{-1}} \right\}^k = (1+z^{-1}+z^{-2}+z^{-3}+z^{-4})^k \quad (10)
 \end{aligned}$$

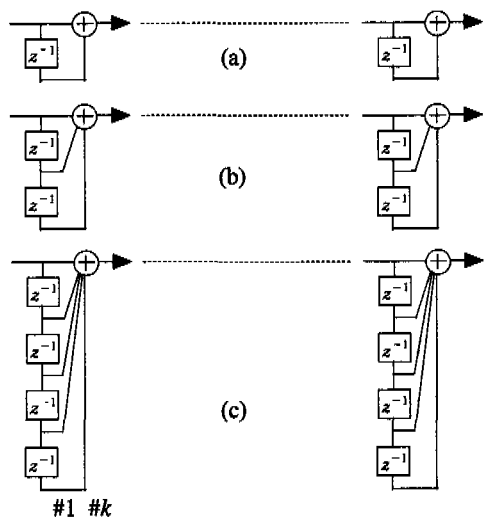


그림 7. (a) 인수 2에 사용되는 필터구조  
(b) 인수 3에 사용되는 필터구조  
(c) 인수 5에 사용되는 필터구조

위의 Non-recursive 식을 구조하면 그림 7과 같다. 이 그림과 같이 각각의 스테이지는 모두  $k$ 개의 필터의 직렬연결로 구성되며,  $H_{11}(z)$ 에서는  $k$ 개의 덧셈기와  $k$ 개의 지연소자가 사용된다. 또한  $H_{21}(z)$ 에서는  $2k$ 개의 덧셈기와  $2k$ 개의 지연소자가 사용되며,  $H_{31}(z)$ 에서는  $4k$ 개의 덧셈기와  $4k$ 개의 지연소자가 사용된다. 즉 서브스테이지에 사용되는 덧셈의 수는 그 서브스테이지의 데시메이션 인수보다 1이 작은 수와 필터 차수  $k$ 와의 곱임을 알 수 있다.

#### IV. 아키텍처 구성 예제와 전력소모 비교

예제로서  $N=180$ 으로 다운 샘플링하고, 필터차수를  $k=3$ 으로 하는 경우를 제안된 구조로 구성해보기로 한다.  $N$ 은 2, 2, 3, 3, 5로 소인수 분해되므로 2와 3과 5의 소수끼리 분리시키기 위하여 1단계로 전달 함수를 다음 식과 같이  $H_1(z), H_2(z), H_3(z)$ 의 전달함수로 분해한다.

$$\begin{aligned}
 H(z) &= \left\{ \frac{1-z^{-2^2 3^2 5}}{1-z^{-1}} \right\}^3 \\
 &= \left\{ \frac{1-z^{-2^2}}{1-z^{-1}} \right\}^3 \left\{ \frac{1-z^{-2^2 3^2}}{1-z^{-2^2}} \right\}^3 \\
 &= H_1(z) T_1(z^{2^2}) \\
 T_1(z) &= \left\{ \frac{1-z^{-3^2 5}}{1-z^{-1}} \right\}^3 = \left\{ \frac{1-z^{-3^2}}{1-z^{-1}} \right\}^3 \left\{ \frac{1-z^{-3^2 5}}{1-z^{-3^2}} \right\}^3 \\
 &= H_2(z) T_2(z^{3^2}) \\
 T_2(z) &= \left\{ \frac{1-z^{-5}}{1-z^{-1}} \right\}^3 = H_3(z)
 \end{aligned} \tag{11}$$

2,3,5의 3가지의 소인수들이므로 위의 식과 같이 먼저 3개의 데시메이션 필터로 분해된다. 2단계로 각 단의 데시메이션 필터를 서브 스테이지로 분리해보자.

$$\begin{aligned}
 H_1(z) &= \left\{ \frac{1-z^{-2^2}}{1-z^{-1}} \right\}^3 = \left\{ \frac{1-z^{-2}}{1-z^{-1}} \right\}^3 \left\{ \frac{1-z^{-(2)^2}}{1-z^{-2}} \right\}^3 \\
 &= H_{11}(z) T_{11}(z^2) \\
 T_{11}(z) &= \left\{ \frac{1-z^{-2}}{1-z^{-1}} \right\}^3 = H_{12}(z) \\
 H_2(z) &= \left\{ \frac{1-z^{-3^2}}{1-z^{-1}} \right\}^3 = \left\{ \frac{1-z^{-3}}{1-z^{-1}} \right\}^3 \left\{ \frac{1-z^{-(3)^2}}{1-z^{-3}} \right\}^3 \\
 &= H_{21}(z) T_{21}(z^3) \\
 T_{21}(z) &= \left\{ \frac{1-z^{-3}}{1-z^{-1}} \right\}^3 = H_{22}(z) \\
 H_3(z) &= \left\{ \frac{1-z^{-5}}{1-z^{-1}} \right\}^3 = H_{31}(z)
 \end{aligned} \tag{12}$$

위의 식들과 같이 2와 3의 인수에 각각 2개의 스테이지가 만들어지며, 5의 인수에 1개의 스테이지가 만들어진다. 마지막 3단계로 각각의 스테이지를 그림 7의 구조를 사용하여 구현하면 그림 8과 같다. 각각의 스테이지별 동작속도는 첫 스테이지를 1이라고 하면 1/2, 1/4, 1/12, 1/36로 동작하게 된다. 지금까지유도한 식에서도 알 수 있듯이 2, 3, 5의 인수가 있을 때, 5의 인수를 사용하는 스테이지가 첫단에 위치하도록 할 수도 있다. 그러나 5의 인수를 사용하는 데시메이션 필터는 2의 인수를 사용하는 스테이지보다 상대적으로 덧셈과 지연소자를 많이 사용하므로 동작속도가 낮은 뒷단에 사용하는 것이 전력소모에 이득이 된다.

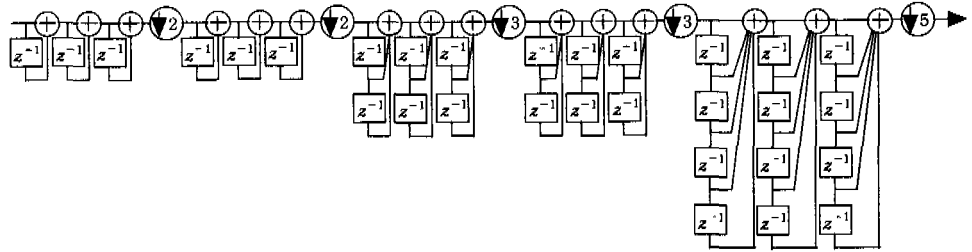
이제 2부터 32까지의 인수에 대한 전력소모를 기존의 아키텍처와 비교해보자. 전력소모는 다음의 식을 사용하였다.

$$P_{dyna} = P_t \cdot C_L \cdot V_{dd}^2 \cdot f_{clk} \tag{13}$$

위의 식에서 dynamic 전력소모,  $P_{dyna}$ 은 CMOS 디지털회로의 총 전력소모 가운데 가장 큰 비중을 차지하는 전력소모이다. 이식에서  $P_t$ 는 전력이 소모되는 transition의 확률이고,  $C_L$ 은 부하 커패시턴스,  $V_{dd}$ 는 인가전압, 그리고  $f_{clk}$ 는 클럭의 주파수이다. 본 논문의 아키텍처와 기존의 것과의 상대 dynamic 전력소모를 구하기 위하여 식 (13)을 다음과 같이 변형하여 사용한다.

$$P_{dyna} = \sum_{\Delta \text{스테이지}} (\text{동작속도} \times \text{면적}) \tag{14}$$

위의 식에서 동작속도는 첫 번째 스테이지를 항상 1로 하고 두 번째 스테이지부터는 다운 샘플러의 크기만큼 줄어들도록 하여 사용한다. 일반적으로 FIR 필터에서는 위의 식의 면적으로서 탭수를 사용한다. 즉 한 개의 탭이 곱셈, 덧셈, 지연소자를 대략 1개씩 사용하여 구성되므로 면적 대신에 탭수를 사용한다. 즉 100탭의 필터의 경우 100개의 곱셈, 99개의 덧셈, 그리고 99개의 지연소자로 구성된다. 본 논문의 아키텍처에서는 곱셈은 사용되지 않으며, 덧셈의 수와 지연소자의 수가 같으므로 둘 중에서 어느 것을 면적으로 사용해도 된다. 본 논문에서는 덧셈의 수를 사용하기로 한다. 따라서 식 (14)를 계산할 때, 덧셈의 수만을 세어서 각각의 스테이지 동작속도와 곱하였다. 각각의 스테이지에서 계산된 곱을 모두 더하면 전력소모가 된다. 이와 같은 식을 사용



스테이지 :	1	2	3	4	5
동작속도 :	1	1/2	1/4	1/12	1/36
덧셈기수 :	3	3	6	6	12
전력소모 :	3	1.5	1.5	0.5	0.33

그림 8. 제안된 아키텍처로 예제 구성

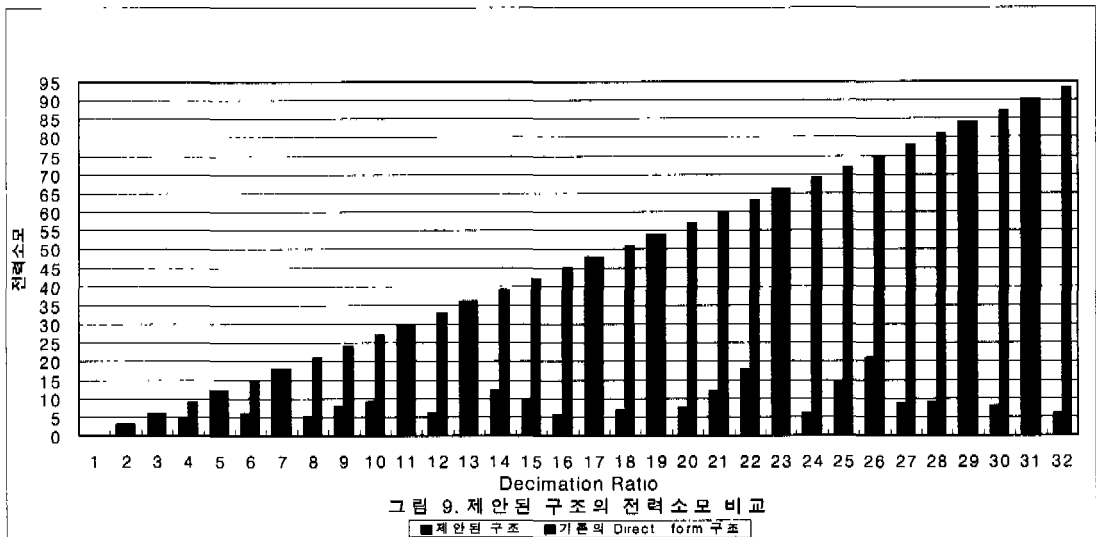


그림 9. 제안된 구조의 전력소모 비교

하여 상대 전력소모를 구한 결과는 그림 9와 같다. 30으로 데시메이션하는 경우에 본 논문이 제안하는 구조의 전력소모를 구해보자. 스테이지는 우선 2, 3, 5로 분리된다. 각 스테이지의 동작속도는 1, 1/2, 1/6이 되고 스테이지별 덧셈의 수는 3개, 6개, 12개가 된다. 따라서 총 전력소모는  $1 \times 3 + 1/2 \times 6 + 1/6 \times 12 = 8$ 이 된다.

기존 구조의 전력소모를 계산하기 위하여 다음의 식을 Direct form으로 구성하여 계산하였다.

$$H(z) = \left\{ \sum_{i=0}^{N-1} z^{-i} \right\}^3 \quad (15)$$

그림 9에서 보여지듯이 위의 식을 Direct form으로 구조할 경우 데시메이션 인수에 비례하여 전력

소모가 증가함을 알 수 있다. 이는 Direct form 구조는 스테이지 분리가 안되므로 모든 동작속도는 1이며, 사용되는 덧셈의 수가 데시메이션 인수에 비례하기 때문이다. 이와 비교하여 제안된 구조는 데시메이션 인수가 소인수로 분해되는 경우에 전력소모가 그림 9에서 보듯이 21이하임을 알 수 있다. 그림 9에서 보인 31개의 데시메이션 인수 중에서, 소인수 분해가 안되는 1개 스테이지 구조의 평균 전력소모는 40.64이며 2개 스테이지 구조의 평균 전력소모는 11.49이다. 또한 3개와 4개는 각각 7.345와 5.8125로서 소인수 분해의 수가 많을수록 스테이지 분리가 많아져서 전력소모가 감소함을 알 수 있다. 본 논문이 제안한 구조의 경우에도, 소인수 분해가 안 되는 2, 3, 5, 7, 11, 13, 19, 23, 29,

31의 데시메이션 인수에 대한 전력소모는 기존의 Direct form 구조와 같음을 알 수 있다. 즉 소인수로 분해되는 경우에만 스테이지 분리가 되어 전력 소모가 감소함을 알 수 있다.

### V. 결론

CIC 데시메이션 필터의 Non-recursive 구조는 지금까지는 데시메이션의 크기가  $2^M$ 으로 나타내지는 경우의 스테이지 분리에 대하여만 구조가 개발되었다. 본 논문은 소인수로 분해되는 임의의 인수에 대하여 스테이지의 분리가 가능함을 보이고 스테이지 분리를 통한 스테이지별 동작속도의 감소를 이룰 수 있음을 보였다. 또한 이러한 스테이지별 Multi-rate의 동작이 전력소모를 감소시킬 수 있음을 보였다. 따라서 Non-recursive 구조에서의 데시메이션 인수에 대한 제약이 완화되어 제안된 Non-recursive 구조는 고속의 데시메이션에 널리 사용될 수 있다. 또한 제안된 구조는 스테이지 별로 동작속도가 감소하므로 가장 빠른 속도의 첫 번째 스테이지부터 병렬처리 등의 방법을 적용하면 전체적인 필터의 속도를 향상시킬 수 있는 아키텍처이기도하다.

### 참고 문헌

[1] E. B. Hogenauer, "An economical class of digital filters for decimation and interpolation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, no. 2, pp. 155-162, April 1981.

[2] A. Y. Kwentus, Z. Jiang, and A. N. Willson, Jr., "Application of filter sharpening to cascaded integer-comb decimation filters," *IEEE Trans. Signal Processing*, vol. 45, no. 2, pp. 457-467, Feb. 1997.

[3] H. J. Oh, S. Kim, G. Choi, and Y. H. Lee, "On the use of interpolated second-order polynomials for efficient filter design in programmable downconversion," *IEEE Journal on selected areas in communications*, vol. 17, no. 4, pp. 551 - 560, April 1999.

[4] H. K. Yang and W. M. Snelgrove, "High speed polyphase CIC decimation filters," *IEEE International Symposium on Circuits and*

*Systems, Volume 2*, pp. 229-232, 1996.

[5] Y. Gao, L. Jia, and H. Tenhunen, "An improved architecture and implementation of cascaded integrator-comb decimation filters," *IEEE International ASIC / SOC Conference*, pp. 391-395, 1999.

[6] Y. Gao, L. Jia, and H. Tenhunen, "A partial-polyphase VLSI architecture for very high speed CIC decimation filters," *IEEE Pacific Rim Conference on communications, computers and signal processing*, pp. 317-320, 1999.

장 영 범(Young-Beom Jang)

정회원



1981년 2월 : 연세대학교 전기공학과 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