

저전력/유연성 Uniform 필터뱅크 구현을 위한 블록 필터 구조

정회원 장영범*, 준회원 양세정*

Block Filter Structure for Low-power/Reconfigurable Uniform Filter Banks Implementation

Young-Beom Jang* *Regular Member*, Se-jung Yang* *Associate Member*

요약

본 논문은 필터뱅크의 저전력 구조와 유연성 구조를 위한 구현 방법을 제안한다. 이와 같은 특성을 갖도록 하기 위하여 블록 필터를 필터뱅크에 사용하였다. 블록 필터를 데시메이션 필터에 적용함으로써, 블록 필터의 병렬-직렬 변환기와 다운 샘플러가 상쇄됨을 보인다. 인터폴레이션 필터에서도 마찬가지로, 업 샘플러와 블록 필터의 직렬-병렬 변환기가 상쇄되어 효율적인 구조가 만들어짐을 보인다. 더 나아가, 블록 필터를 Uniform 필터뱅크에 적용함으로써 분석 단과 합성 단의 첫 번째 채널 필터가 모든 채널에 공유될 수 있음을 보인다. 이와 같은 공유를 통하여 계산량이 현저히 감소된 필터뱅크를 구현할 수 있었다. 또한 블록 필터뱅크는 필터뱅크의 차수 변환에 따른 하드웨어의 가감이 매우 용이하여 유연성을 갖는 구조임을 보인다.

ABSTRACT

Block filter implementation technique for uniform filter banks is proposed in this paper. By applying block filters into decimation and interpolation filters, it is shown that down and up samplers are cancelled out in respective filters. Furthermore, by applying block filters into uniform filter banks, significant reduction for computational complexity is achieved since prototype filter can be shared in each channel implementation. Also, it is shown that proposed implementation is a reconfigurable structure in term of filter order variation.

I. 서론

필터뱅크를 사용하는 통신 시스템에서는 필터뱅크 계산의 복잡도와 유연성이 동시에 고려되어 연구되어야 한다. 필터뱅크의 유연성이란, 필터의 차수가 변화하는 경우에 적용할 수 있는 구조를 의미한다. 기존의 Uniform DFT 필터뱅크의 경우, Polyphase 구조가 가장 많이 사용된다. 왜냐하면 첫 번째 채널에 사용되는 필터(Prototype 필터라고 함)가 다른 모든 채널에서도 공용으로 사용되어 계산량을 줄일 수가 있기 때문이다. 또한 DFT의 구현은

FFT를 사용하여 역시 계산량을 줄일 수 있다.^{[1][2]}

필터뱅크의 응용인 다채널 DDC(Digital Down Converters)와 DUC(Digital Up Converters)에서는 필터의 유연성이 요구된다. 즉, 소프트웨어 라디오와 같은 시스템에서는 다채널 DDC와 DUC의 필터 유연성을 필요로 한다. 필터뱅크의 저전력 구현과 유연성을 가진 구현을 위하여, Polyphase 구조와 같은 필터 공유 기술과 또한 규칙적인 구조가 요구된다. 본 논문에서는 이러한 두 가지 요구를 만족하는 필터뱅크를 블록 필터를 사용하여 구현하는 방법을 제안한다. 논문의 구성은, II장에서는 데시메이션

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

논문번호 : 010012-0220, 접수일자 : 2001년 2월 20일

※ 본 연구는 교육부 BK21 지원에 의해 수행되었습니다.

필터와 인터플레이션의 블록 필터 구조를 제안한다. 블록 필터의 적용을 통하여 샘플러들이 상쇄됨을 보인다. III장에서는 필터뱅크에 블록 필터를 적용하는 방법을 제안한다. IV장에서는 예제를 통하여 블록 필터뱅크의 저전력 구현과 유연성을 알아본다.

II. 데시메이션과 인터플레이션의 블록 필터 구조

2.1 데시메이션의 블록 Direct form 구조

원래의 블록 필터 구조는 병렬 처리를 통한 필터의 연산 속도를 높이기 위하여 개발되었다.^{[3][4][5]} 그림 1(a)와 같은 데시메이션 필터와 다운 샘플러에 블록 Direct form 필터를 적용하면 그림 1(b)와 같다. 그림 1(b)에서 보듯이, 블록 필터는 SPC(Serial to Parallel Converters, 직렬-병렬 변환기), 블록 필터, 그리고 PSC(Parallel to Serial Converters, 병렬-직렬 변환기)로 구성된다. 이 때, 블록 필터의 크기는 뒤에 붙는 다운 샘플러의 크기와 같도록 해야 한다. 그림 1(b)에서 점선의 원으로 둘러싸인 PSC와 다운 샘플러는 등가적으로 그림 1(c)와 같이 표현될 수 있다. 즉, PSC는 병렬 신호들을 업 샘플러에 의해 제로들을 삽입시키고, 지연소자들을 사용하여 차례로 더함으로서 그림 1(c)의 $p[n]$ 의 직렬신호를 얻을 수 있다. 먼저, 그림 1(c)의 k 번째 업 샘플러 출력은 다음과 같이 표현된다.

$$g_k[n] = \begin{cases} f_k[n/L], & n=0, L, 2L, \dots \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

따라서 그림 1(c)의 $p[n]$ 은 다음과 같이 표현된다.

$$p[n] = g_0[n] + g_1[n-1] + \dots + g_{L-1}[n-L+1] \quad (2)$$

$$= \begin{cases} f_0[-\frac{n}{L}], & n=0, L, 2L, \dots \\ f_1[\frac{n-1}{L}], & n=1, L+1, 2L+1, \dots \\ \vdots \\ f_{L-1}[\frac{n-(L-1)}{L}], & n=L-1, 2L-1, \dots \end{cases}$$

다운 샘플러는 두 단의 동작으로 나누어 표현된다. 첫 단계에서는 L 의 정수 배를 제외한 모든 점을 0으로 바꾸는 동작이므로 다음과 같이 나타내진다.

$$p'[n] = \begin{cases} f_0[\frac{n}{L}], & n=0, L, 2L, \dots \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

두 번째 단계에서는 $p'[n]$ 이 시간 축에서 압축되어

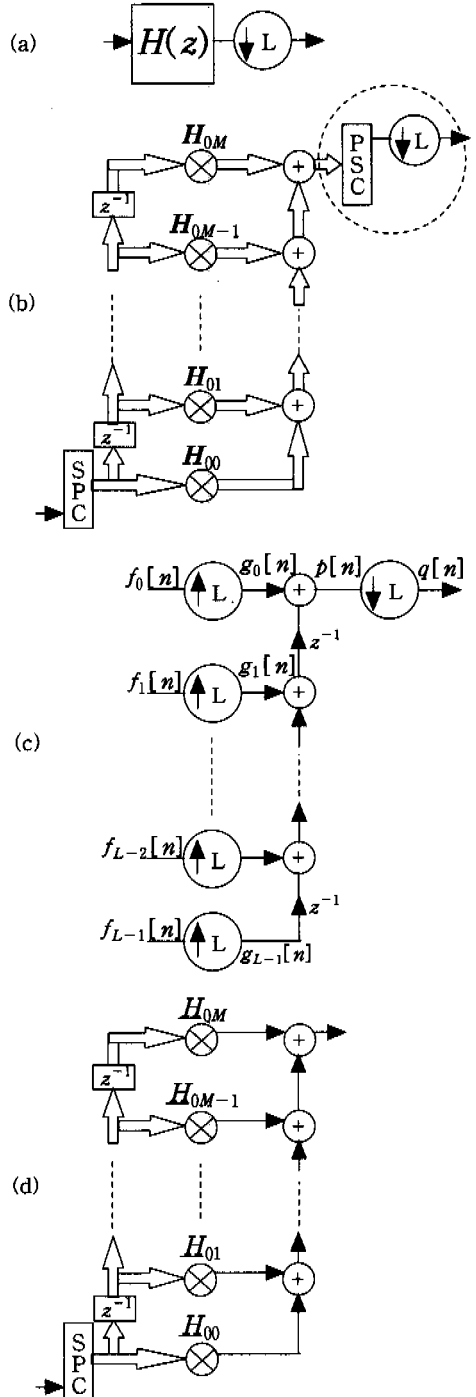


그림 1. 데시메이션 필터의 블록 필터 구조
(a)일반적인 블록도, (b)블록 Direct form 구조, (c)PSC와 다운 샘플러의 등가 구조, (d)데시메이션 필터의 간략화된 블록 필터 구조

0들이 사라지므로 최종적으로 다운 샘플러의 출력은 다음과 같이 표현될 수 있다.

$$q[n] = f_0[n], \quad n = 0, 1, 2, \dots \quad (4)$$

이 식에서 보듯이 다운 샘플러의 출력은 PSC의 병렬 입력 중에서 맨 위의 신호가 됨을 알 수 있다. 즉, PSC의 병렬 입력 중 맨 위의 신호만이 다운 샘플러를 통하여 출력됨을 알 수 있다. 따라서 블록 필터의 병렬 출력 중에서 맨 위의 출력만이 필요하므로 그림 1(d)와 같이 행렬 계수들이 벡터 계수로 간략화 될 수 있다. 본 논문에서는 블록 필터 구조로서 블록 Direct form을 선택하였다. 블록의 크기를 다운 샘플러의 크기와 같이 L 로 정의함으로써 블록 시스템 함수는 다음과 같이 나타내진다.

$$H_0(z) = H_{00} + H_{01}z^{-1} + \dots + H_{0M}z^{-M} \quad (5)$$

이 식에서 행렬 계수들은 다음과 같다.

$$H_{00} = \begin{bmatrix} h_0 & 0 & \dots & 0 \\ h_1 & h_0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ h_{L-1} & h_{L-2} & \dots & h_0 \end{bmatrix}, \quad H_{01} = \begin{bmatrix} h_L & h_{L-1} & \dots & h_1 \\ h_{L+1} & h_L & \dots & h_2 \\ \vdots & \vdots & \ddots & \vdots \\ h_{2L-1} & h_{2L-2} & \dots & h_L \end{bmatrix}, \dots, \quad H_{0M} = \begin{bmatrix} h_N & h_{N-1} & \dots & h_{N-L+1} \\ 0 & h_N & \dots & h_{N-L+2} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & h_N \end{bmatrix} \quad (6)$$

식(6)에서 N 은 주어진 데시메이션 필터의 차수이며, M 은 블록 필터의 차수이다. 여기서 M 은 N/L 보다 크거나 같은 최소의 정수로 정의된다. (6)의 행렬계수를 사용한 블록 Direct form 구조가 그림 1(b)이다. 앞에서 보인 PSC와 다운 샘플러의 관계를 이용한 그림 1(d)의 벡터 계수들은 다음과 같다.

$$\begin{aligned} H_{00} &= [h_0 \ 0 \ \dots \ 0], \\ H_{01} &= [h_L \ h_{L-1} \ \dots \ h_1], \\ &\vdots \\ H_{0M} &= [h_N \ h_{N-1} \ \dots \ h_{N-L+1}]. \end{aligned} \quad (7)$$

위의 벡터계수들은 식(6)의 행렬계수들의 첫째 행이 됨을 알 수 있다.

2.2 인터플레이션의 블록 Direct form 구조

이 절에서는 블록 필터 구조를 인터플레이션 필터에 적용해 보기로 한다. 개념적으로 데시메이션 필터에 적용하는 것과 같으므로 간단히 설명하기로 한다. 먼저, 블록 시스템 함수는 다음과 같다.

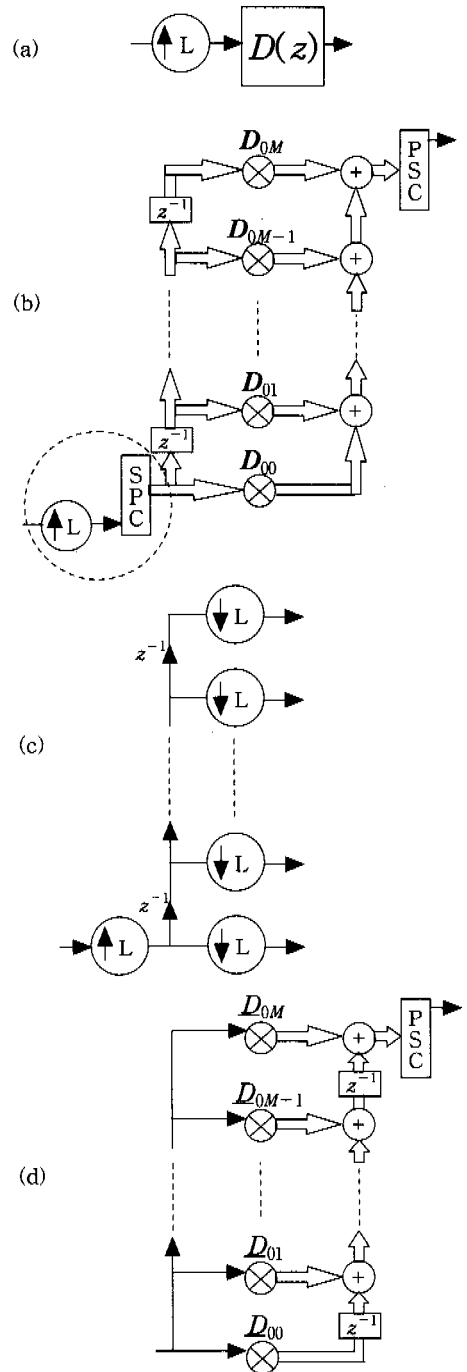


그림 2. 인터플레이션 필터의 블록 필터 구조
(a)일반적인 블록도, (b)블록 Direct form 구조, (c)업 샘플러와 SPC의 등가 블록도, (d)인터플레이션 필터의 간략화된 블록 필터 구조

$$D_0(z) = D_{00} + D_{01}z^{-1} + \dots + D_{0M}z^{-M} \quad (8)$$

식(8)의 행렬 계수는 다음과 같다.

$$\begin{aligned} D_{00} &= \begin{bmatrix} d_0 & 0 & \dots & 0 \\ d_1 & d_0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ d_{L-1} & d_{L-2} & \dots & d_0 \end{bmatrix}, \\ D_{01} &= \begin{bmatrix} d_L & d_{L-1} & \dots & d_1 \\ d_{L+1} & d_L & \dots & d_2 \\ \vdots & \vdots & \ddots & \vdots \\ d_{2L-1} & d_{2L-2} & \dots & d_L \end{bmatrix}, \dots, \\ D_{0M} &= \begin{bmatrix} d_N & d_{N-1} & \dots & d_{N-L+1} \\ 0 & d_N & \dots & d_{N-L+2} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & d_N \end{bmatrix}. \end{aligned} \quad (9)$$

식(9)에서의 N, M, L 은 데시메이션 필터 경우와 똑 같이 정의된다. 위의 식(9)를 사용하여 그림 2(b)의 구조를 얻을 수 있다. 이 구조에 업 샘플러와 SPC의 상쇄를 이용하여 간략화 시키면 그림 2(d)의 구조를 얻는다. 이 그림에서의 벡터 계수들은 다음과 같다.

$$\begin{aligned} D'_{00} &= [d_0 \quad d_1 \quad \dots \quad d_{L-1}], \\ D'_{01} &= [d_L \quad d_{L+1} \quad \dots \quad d_{2L-1}], \\ &\vdots \\ D'_{0M} &= [d_N \quad 0 \quad \dots \quad 0]. \end{aligned} \quad (10)$$

식(10)에서 볼 수 있듯이, 벡터 계수들은 식(9)의 행렬계수들의 첫 번째 열로 구성됨을 알 수 있다.

III. 필터뱅크의 블록 필터 구조

3.1 분석단의 블록 필터뱅크의 구조

필터뱅크의 분석 단에는 채널 수만큼의 데시메이션 필터가 사용된다. 0부터 $L-1$ 까지 L 개의 채널을 갖는 분석 단을 고려한다. Uniform 필터뱅크의 경우에 l 번째 채널의 데시메이션 필터는 Prototype 필터의 shift 버전이므로 다음과 같이 나타내진다.

$$\begin{aligned} H_l(z) &= H_0(ze^{j\frac{2\pi}{L}l}) \\ &= h_0 + h_1z^{-1}e^{j\frac{2\pi}{L}l} + h_2z^{-2}e^{j\frac{2\pi}{L}2l} + \dots + h_Nz^{-N}e^{j\frac{2\pi}{L}Nl}. \end{aligned} \quad (11)$$

여기에서 $l=0, 1, \dots, L-1$.

이 시스템 함수로부터 식(5)와 같은 블록 시스템 함수가 쉽게 유도될 수 있다.^[5] 이와 같이 유도된 블록 시스템 함수의 행렬 계수들로부터 첫째 행들만을 뽑아서 벡터 계수가 얻어지며, 다음과 같이 나타내진다.

$$\begin{aligned} H_{l0} &= [h_0 \quad 0 \quad \dots \quad 0], \\ H_{l1} &= [h_L e^{j\frac{2\pi}{L}l} \quad h_{L-1}e^{j\frac{2\pi}{L}(L-1)l} \quad \dots \quad h_1 e^{j\frac{2\pi}{L}l}], \\ &\vdots \\ H_{lM} &= [h_N e^{j\frac{2\pi}{L}Nl} \quad h_{N-1}e^{j\frac{2\pi}{L}(N-1)l} \quad \dots \quad h_{N-L+1}e^{j\frac{2\pi}{L}(N-L+1)l}]. \end{aligned} \quad (12)$$

식(12)를 좀더 간단히 표현하기 위해서

$W_L = e^{-j\frac{2\pi}{L}}$ 를 정의하면 식(12) 벡터들의 두 번째 엘리먼트들은 모두 W_L^l 을 사용하여 나타낼 수 있다. 예를 들면 H_{l1} 의 두 번째 엘리먼트는 다음과 같이 표현될 수 있다. 즉,

$$h_{L-1}e^{j\frac{2\pi}{L}(L-1)l} = h_{L-1}e^{-j\frac{2\pi}{L}l} = h_{L-1}W_L^l. \quad (13)$$

이와 같이 같은 번째의 엘리먼트들은 같은 W_L^k 를 사용하여 나타낼 수 있다. 위의 벡터들의 마지막 엘리먼트들은 다음과 같이 표현할 수 있다.

$$h_1 e^{j\frac{2\pi}{L}l} = h_1 e^{j\frac{2\pi}{L}l} e^{-j2\pi l} = h_1 e^{-j\frac{2\pi}{L}l(L-1)} = h_1 W_L^{l(L-1)}. \quad (14)$$

따라서 식(12)의 벡터 계수들은 W_L^k 를 사용하여 다음과 같이 표기할 수 있다.

$$\begin{aligned} H_{l0} &= [h_0 \quad 0 \quad \dots \quad 0], \\ H_{l1} &= [h_L \quad h_{L-1}W_L^l \quad \dots \quad h_1 W_L^{l(L-1)}], \\ &\vdots \\ H_{lM} &= [h_N \quad h_{N-1}W_L^l \quad \dots \quad h_{N-L+1}W_L^{l(L-1)}]. \end{aligned} \quad (15)$$

식(7)과 (15)를 비교해보면, 식(15)에서 W_L^k 들만 제거하면 식(7)과 같음을 알 수 있다. 즉, W_L^k 들만을 분리하여 구현하면 식(7)의 필터를 공유할 수 있게 된다. 첫 번째 채널의 필터를 l 번째 채널에서도 공유할 수 있음을 의미한다. 모든 채널에서 분리된 W_L^k 들의 구현은 DFT가 되므로, 이는 FFT를 이용하여 구현할 수 있다. 전체적인 분석단의 블록 필터 구조는 그림 3(a)와 같다.

3.2 합성단의 블록 필터뱅크의 구조

이 절에서는 필터뱅크의 합성단에 사용되는 l 번째 인터플레이션 필터를 고려해보기로 하자. 역시 Uniform 필터이므로 다음과 같이 Prototype 필터를 shift 하여 표현할 수 있다.

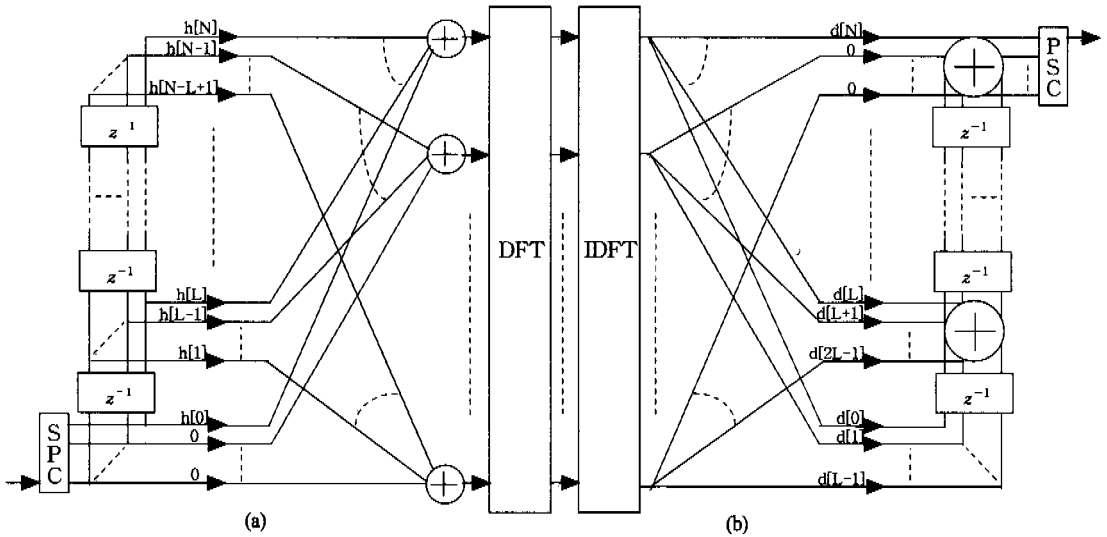


그림 3. Uniform 필터 बैं크의 블록 필터 구조, (a)분석단, (b)합성단

$$D_l(z) = D_0(z e^{j\frac{2\pi}{L}l})$$

$$= d_0 + d_1 z^{-1} e^{j\frac{2\pi}{L}l} + d_2 z^{-2} e^{j\frac{2\pi}{L}2l} + \dots + d_N z^{-N} e^{j\frac{2\pi}{L}Nl}, \quad (16)$$

여기에서 $l=0,1,\dots,L-1$.

분석 단과 같은 방법으로, 위의 시스템 함수로부터 다음과 같은 벡터 계수들을 유도하였다.

$$D'_{l0} = [d_0 \quad d_1 W_L^{-l} \quad \dots \quad d_{L-1} W_L^{-l(L-1)}],$$

$$D'_{l1} = [d_L \quad d_{L+1} W_L^{-l} \quad \dots \quad d_{2L-1} W_L^{-l(L-1)}], \quad (17)$$

$$\vdots$$

$$D'_{lM} = [d_N \quad 0 \quad \dots \quad 0].$$

식(17)에서도 역시 W_L^k 들을 제거하면 식(10)과 같아지므로 W_L^k 들을 분리하면 식(10)의 필터를 l 번째 채널에서 공유할 수 있게 된다. 이는 모든 채널에서 공유할 수 있음을 의미한다. 또한 분리된 W_L^k 들의 구현은 IDFT가 된다. 합성 단은 그림 3(b)와 같이 나타내진다.

IV. 실험 및 고찰

블록 필터 बैं크의 설계 예제로서 채널 수 L 이 4이고 필터 차수 N 이 8인 필터 बैं크를 설계해 보기로 한다. III장에서 제안한 방법을 사용하여 분석 단과 합성 단의 벡터 계수들을 구하면 다음과 같다.

$$H_{00} = [h_0 \quad 0 \quad 0 \quad 0],$$

$$H_{01} = [h_4 \quad h_3 \quad h_2 \quad h_1],$$

$$H_{02} = [h_8 \quad h_7 \quad h_6 \quad h_5], \quad (18)$$

$$D'_{00} = [d_0 \quad d_1 \quad d_2 \quad d_3],$$

$$D'_{01} = [d_4 \quad d_5 \quad d_6 \quad d_7],$$

$$D'_{02} = [d_8 \quad 0 \quad 0 \quad 0].$$

위의 벡터 계수들을 사용하여 블록 필터 बैं크를 설계하면 그림 4(a)와 같다. 이 그림에서 DFT와 IDFT의 크기는 4이다. 두 번째 예제로서 채널 수 L 이 4이고 필터 차수 N 이 12인 필터 बैं크를 설계한다. 분석 단과 합성 단의 벡터 계수들을 구하면 다음과 같다.

$$H_{00} = [h_0 \quad 0 \quad 0 \quad 0],$$

$$H_{01} = [h_4 \quad h_3 \quad h_2 \quad h_1],$$

$$H_{02} = [h_8 \quad h_7 \quad h_6 \quad h_5],$$

$$H_{03} = [h_{12} \quad h_{11} \quad h_{10} \quad h_9], \quad (19)$$

$$D'_{00} = [d_0 \quad d_1 \quad d_2 \quad d_3],$$

$$D'_{01} = [d_4 \quad d_5 \quad d_6 \quad d_7],$$

$$D'_{02} = [d_8 \quad d_9 \quad d_{10} \quad d_{11}],$$

$$D'_{03} = [d_{12} \quad 0 \quad 0 \quad 0].$$

위의 벡터 계수들을 사용하여 블록 필터 बैं크를 설계하면 그림 4(b)와 같다. 이 예제를 통하여 알 수 있듯이, 필터의 차수가 4만큼 증가할 때마다 한 개의 블록이 분석 단과 합성 단의 맨 위에 추가된다. 필터 बैं크의 응용에서 채널의 수는 고정되어 있고, 필터의 차수 만 변화하는 시스템에 본 논문이 제안하는 구조가 장점을 갖는다. 반도체 LSI로 구현할

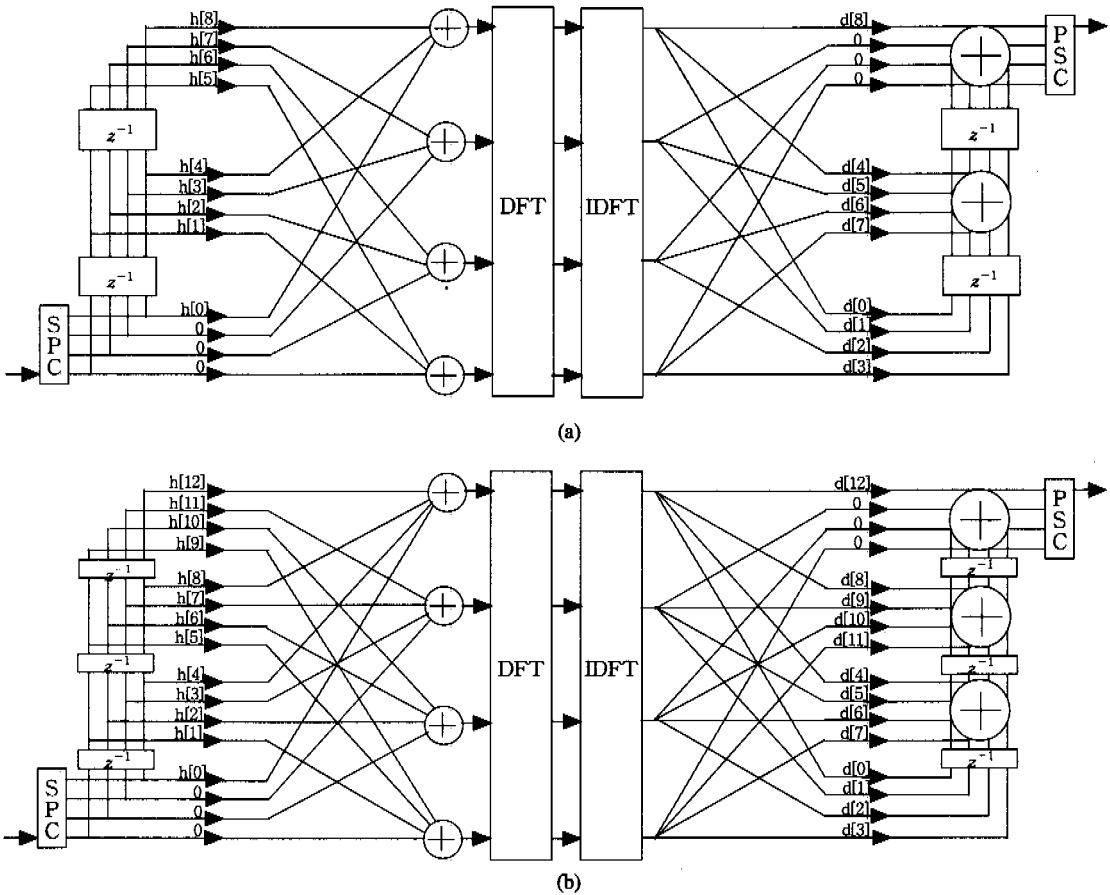


그림 4. 예제의 블록 필터 뱅크 구조, (a)N=8, L=4, (b)N=12, L=4

때에, 필터의 차수가 4가 증가할 때마다 한 개의 블록을 sleep mode에서 active mode로 전환시키도록 설계함으로써 유연성을 갖는 구조가 될 수 있다.

이제 그림 4(a)의 실제 필터 계수를 구하여 각 채널별 주파수 응답을 구하면 다음과 같다. 샘플링 주파수 f_s 는 19.6608MHz를 사용하였고, 필터의 차수를 8로 미로 정했으므로 그것에 맞추기 위해 통과대역의 주파수와 리플은 각각 1.260MHz와 4dB로 하고 저지대역의 주파수와 어테뉴에이션은 각각 2.4576MHz와 -20dB로 하였다. FIR 필터 설계도구인 REMEZ 알고리즘을 사용하여 사양에 맞는 필터 계수를 구한 후, III장에서 제안한 방법을 적용하여 각 채널의 벡터 계수와 시스템 함수를 구하면 다음과 같다.

$$\begin{aligned} H_{00} &= [0.0897 \quad 0 \quad 0 \quad 0], \\ H_{01} &= [0.1822 \quad 0.1768 \quad 0.1438 \quad 0.1095], \\ H_{02} &= [0.0897 \quad 0.1095 \quad 0.1438 \quad 0.1768]. \end{aligned}$$

$$\begin{aligned} H_0(z) &= 0.0897 + 0.1095 z^{-1} + 0.1438 z^{-2} \\ &\quad + 0.1768 z^{-3} + 0.1822 z^{-4} + 0.1768 z^{-5} \\ &\quad + 0.1438 z^{-6} + 0.1095 z^{-7} + 0.0897 z^{-8} \end{aligned}$$

$$\begin{aligned} H_{10} &= [0.0897 \quad 0 \quad 0 \quad 0], \\ H_{11} &= [0.1822 \quad 0.1768 W_4^1 \quad 0.1438 W_4^2 \quad 0.1095 W_4^3], \\ H_{12} &= [0.0897 \quad 0.1095 W_4^1 \quad 0.1438 W_4^2 \quad 0.1768 W_4^3]. \end{aligned}$$

$$\begin{aligned} H_1(z) &= 0.0897 + 0.1095 W_4^3 z^{-1} + 0.1438 W_4^2 z^{-2} \\ &\quad + 0.1768 W_4^1 z^{-3} + 0.1822 z^{-4} + 0.1768 W_4^3 z^{-5} \\ &\quad + 0.1438 W_4^2 z^{-6} + 0.1095 W_4^1 z^{-7} + 0.0897 z^{-8} \end{aligned}$$

$$\begin{aligned} H_{20} &= [0.0897 \quad 0 \quad 0 \quad 0], \\ H_{21} &= [0.1822 \quad 0.1768 W_4^2 \quad 0.1438 W_4^4 \quad 0.1095 W_4^6], \\ H_{22} &= [0.0897 \quad 0.1095 W_4^2 \quad 0.1438 W_4^4 \quad 0.1768 W_4^6]. \end{aligned}$$

$$\begin{aligned} H_2(z) &= 0.0897 + 0.1095 W_4^6 z^{-1} + 0.1438 W_4^4 z^{-2} \\ &\quad + 0.1768 W_4^2 z^{-3} + 0.1822 z^{-4} + 0.1768 W_4^6 z^{-5} \\ &\quad + 0.1438 W_4^4 z^{-6} + 0.1095 W_4^2 z^{-7} + 0.0897 z^{-8} \end{aligned}$$

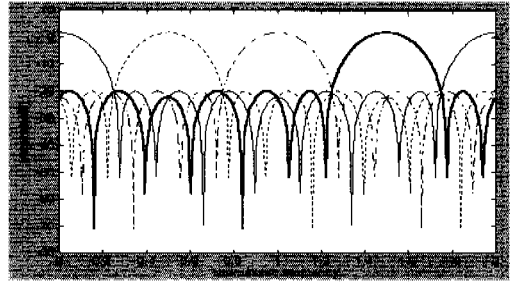
$$\begin{aligned} H_{30} &= [0.0897 \quad 0 \quad 0 \quad 0], \\ H_{31} &= [0.1822 \quad 0.1768 W_4^3 \quad 0.1438 W_4^6 \quad 0.1095 W_4^9], \\ H_{32} &= [0.0897 \quad 0.1095 W_4^3 \quad 0.1438 W_4^6 \quad 0.1768 W_4^9]. \end{aligned}$$

$$H_3(z) = 0.0897 + 0.1095 W_4^9 z^{-1} + 0.1438 W_4^6 z^{-2} + 0.1768 W_4^3 z^{-3} + 0.1822 z^{-4} + 0.1768 W_4^9 z^{-5} + 0.1438 W_4^6 z^{-6} + 0.1095 W_4^3 z^{-7} + 0.0897 z^{-8}$$

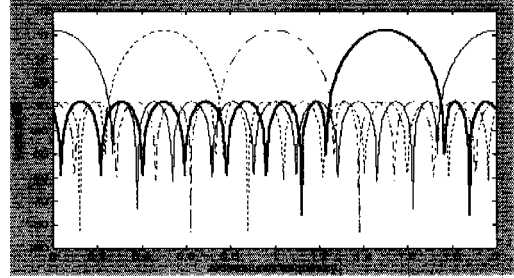
그림 4(b)의 각 채널별 시스템 함수도 마찬가지로 구할 수 있다. 이 때 저지대역의 감쇠만 -30dB로 하였으며 다른 사양과 전개과정은 그림4(a)의 경우와 동일하다. 위와 같이 설계한 그림4(a),(b)의 각 채널별 주파수 응답을 그림5(a),(b)에 제시하였다.

필터 차수의 변화에 대한 구조의 유연성을 블록 필터 बैं크와 Polyphase 필터 बैं크를 비교해보기로 한다. 필터 차수는 8차에서 20차까지 4차씩 증가되는 4가지의 필터를 지원하는 필터 बैं크를 만들도록 한다. 즉 8, 12, 16, 20차의 차수 변화가 있는 경우를 비교한다. 각각의 아키텍처에서 한 개의 모듈 당 한 개의 곱셈기를 사용하도록 한다. 한 개의 곱셈, 즉 1 MAC (Multiplication and Accumulation)에 필요한 계산시간을 10nsec라고 가정한다. 먼저 블록 필터 बैं크 분석 단의 경우에, 8차 필터이면 3개의 모듈이 필요하며 각 모듈 당 40nsec의 계산시간이 소요되므로 총 소요시간도 역시 40nsec가 된다. 블록 필터 बैं크는 병렬처리와 Pipeline 처리가 동시에 수행되는 Systolic 구조임을 알 수 있다. 차수가 12차로 증가되면 분석 단의 모듈은 4개가 되며 역시 각 모듈 당 40nsec의 계산시간이 소요되므로 총 소요시간도 40nsec는 변하지 않는다. 즉, 필터 차수의 증가에 대하여 모듈의 수만 변하고 모듈내의 계산시간은 변하지 않는 구조임을 알 수 있다. 이와 같이 블록 필터 बैं크는 Pipeline 처리는 변하지 않고, 병렬 처리의 수가 증가하는 Systolic 프로세서이다.

Polyphase 필터 बैं크를 살펴보자. 8차인 경우에는 분석 단에 4개의 모듈이 필요하며 각 모듈 당 30nsec의 계산시간이 소요되므로 총 소요시간은 30nsec가 된다. 차수가 12차로 증가되면 모듈은 4개이지만 각 모듈 당 4번의 곱셈이 필요하므로 40nsec의 계산시간이 소요되므로 총 소요시간은 40nsec이다. 즉 분석 단의 모듈은 4개로 고정이나, 각 모듈의 계산시간이 변화한다. 이를 종합하면 표 1과 같다. 이와 같이 Polyphase 필터 बैं크는 Pipeline 처리시간은 증가하고, 병렬 처리의 수는 변하지 않는 Systolic 프로세서이다. 표 1에서 모듈수가 칩으로 구현할 때에는 면적이 된다. 따라서 8차에서 20차까지 지원되는 분석 단을 설계할 경우에 블록 필터 बैं크는 면적이 6이고, Polyphase 필터 बैं크



(a)



(b)

그림 5. 블록 필터 बैं크 구조의 주파수 응답, (a)N=8, L=4, (b)N=12, L=4

크는 4로서 고정이다. Polyphase 필터 बैं크는 구현 면적에 최적인 Systolic 프로세서이고 제안된 블록 필터 बैं크는 처리시간에 최적인 Systolic 프로세서를 알 수 있다.

IV. 결론

본 논문은 필터 बैं크의 효율적인 블록 필터 구조를 제안하였다. 블록 필터를 데시메이션 필터나 인터플레이션 필터에 적용할 경우 샘플러들과의 상쇄에 의하여 계산량이 감소됨을 보였다. 또한 블록 필터를 Uniform 필터 बैं크에 적용할 경우에는 첫째 채널의 필터가 모든 채널에서 공유할 수 있음을 보였다. 따라서 제안된 필터 बैं크는 계산량이 감소된

표 1. 차수변화에 따른 필터 बैं크 계산시간 비교

필터 차수	블록 필터 बैं크			Polyphase 필터 बैं크		
	분석단 모듈수	곱셈수 /모듈	계산 시간 (nsec)	분석단 모듈수	곱셈수 /모듈	계산 시간 (nsec)
8	3	4	40	4	3	30
12	4	4	40	4	4	40
16	5	4	40	4	5	50
20	6	4	40	4	6	60

저전력 구현이 가능하다. 또한 제안된 구조는 필터 차수의 증가에 대하여 Pipeline 처리는 변하지 않고 병렬처리의 수만 증가하는 Systolic 구조가 되어 유연성을 갖는 구조임을 보였다.

참고 문헌

[1] M. Bellanger, G. Bonnerot, and M. Coudreuse, "Digital filtering by polyphase network: Application to sample rate alteration and filter banks," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-24, pp. 109-114, Apr. 1976.

[2] P. P. Vaidyanathan, *Multirate systems and filter banks*, New Jersey: Prentice Hall, pp.120-133, 1993.

[3] C. S. Burrus, "Block Implementation of Digital Filter," *IEEE Trans. Circuit Theory*, vol. CT-18, No. 6, pp. 697-701 Nov. 1971.

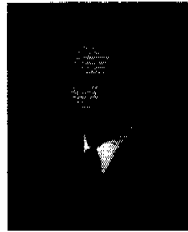
[4] C. W. Barnes, and S. Shinnaka, "Block-Shift Invariance and Block Implementation of Discrete-Time Filters," *IEEE Trans. Circuits and Systems*, vol. CAS-27, No. 8, pp. 667-672, Aug. 1980.

[5] Y. Jang, and S. P. Kim, "Block digital filter structures and their finite precision responses," *IEEE Trans. Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 43, No. 7, pp. 495-506, July 1996.

[6] S. J. Jou, S. Y. Wu, and C. K. Wang, "Low-power multirate architecture for IF digital frequency down converter," *IEEE Trans. Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 45, No. 11, pp. 1487-1494, Nov. 1998.

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

양 세 정(Se-Jung Yang)

준회원



2001년 2월 : 이화여자대학교 정보통신학과 졸업, 공학사
 2001년 3월~현재 : 이화여자대학교 정보통신학과 석사1학기

<주관심 분야> 통신신호처리