

논문-99-4-2-05

선형 위상 IDWT 필터의 VLSI 구조

김 인 철*, 정 영 모**

A VLSI Architecture for the Linear-Phase IDWT Filter

Rin Chul Kim* and Yong Mo Chung**

요 약

본 논문에서는, IDWT(inverse discrete wavelet transform)를 효율적으로 구현하는 한 방법으로 홀수 탭(tap)의 선형위상 필터의 VLSI 구조를 제안한다. 제안한 필터 구조는 선형 위상 필터의 대칭 특성을 이용하여 대칭적인 위치에 있는 입력을 먼저 합한 다음 필터링을 수행한다. 이때 발생하는 전역 연결을 해결하기 위하여 입력의 흐름을 U자형으로 만듦으로써 국부적인 연결로 필터를 구현한다. 제안한 필터는 지연 소자부, 연산부, 덧셈부, 그리고 후처리부 등으로 이루어진다. 그리고, 각 부분들을 규칙적으로 배열하고, 국부적으로 연결함으로써 제안한 구조를 설계하기 때문에, 단순히 해당 부분들을 추가/삭제함으로써 임의의 선형 위상 IDWT 필터를 구현할 수 있다는 장점이 있다. 그리고, 제안한 필터를 직렬 연결 혹은 반순환적(semi-recursive) 구조로 배열함으로써 M 레벨 IDWT를 구현할 수 있음을 보인다. 본 논문에서 제안한 IDWT 구조는 기존의 구조들에 비해 간단하기 때문에 MPEG-4 등 관련 분야에 효과적으로 적용될 것으로 기대된다

Abstract

In this paper, in order to implement the IDWT(inverse discrete wavelet transform) with relatively low complexity, we propose a VLSI architecture for odd-tap linear-phase IDWT filters. By considering the symmetric property of the linear phase filter, the input is added to the one located at symmetrical position of the filter before filtering. Then, we rearrange the delay line of the filter in a U-shaped fashion, requiring no global interconnection between the components. The proposed architecture for the IDWT filter consists of delay units, operator units, adder units, and postprocessor unit. Since each units are configured regularly and interconnected locally, the proposed architecture can accommodate arbitrary linear phase IDWTs by simply adding/removing the corresponding units. The M-level IDWT can be implemented by interconnecting the proposed architecture in a cascaded or semi-recursive form. It is expected that the proposed architecture for the IDWT can be effectively employed in the related area including MPEG-4, since the proposed architecture is less complex than the conventional architectures.

I. 서 론

최근 멀티미디어 등 다양한 서비스에 대한 관심이 확대

됨에 따라, H.26x, MPEG, JPEG 등 영상 정보를 전송/저장할 수 있는 효과적인 방법에 대한 표준이 정해지고 있다^[1,2]. 이들 표준안에서 일반적으로 채택되고 있는 DCT(discrete cosine transform)는 압축 효과가 뛰어난 반면, 저 전송률에서 구획화 현상(blocking effect)등의 화질 열화 현상이 나타난다는 단점이 있다. 이러한 단점을 해결하는 한 방법으로 DWT(discrete wavelet transform)^[3]가 제시되었다. 특히, Shapiro에 의해 제안된 EZW(embedded zero-tree wavelet) 부호화 방식^[4] 등이 소개됨에 따라

*서울시립대학교 전자전기공학부

School of Electrical Eng., University of Seoul

**한성대학교 정보전산학부 정보통신공학전공

School of Information and Computer Eng., Hansung Univ.

※본 연구는 '97 과학재단 특정기초연구(97-0100-0201-3)의 지원으로 수행되었습니다.

DWT는 효과적인 영상 부호화 방법의 하나로 정착되고 있다.

DWT는 시간과 주파수 영역에서 국부적인 특성을 갖는 기저 함수의 집합을 이용하여 신호를 분해하는 것으로, 영상이나 음성과 같은 비정재적(non-stationary) 성질을 가지는 신호를 표현하는 데 효과적이다. DWT의 역 과정인 IDWT(inverse discrete wavelet transform)는 분해된 신호를 합성하여 원래의 신호를 재생하는 것을 말한다. DWT에서 사용되는 웨이브렛 기저 함수는 대역통과 필터와 같은 특성을 보이는 모 웨이브렛(mother wavelet)을 팽창/수축(dilation), 이동(translation)시켜 얻는다. 이때, 각 웨이브렛 기저 함수는 직교하는(orthogonal) 특성을 가진다.

일반적으로 DWT를 이용한 다해상도 분석에서는 dyadic 구조로 신호를 분해하게 된다. 즉, 모 웨이브렛으로부터 도출되는 고대역 필터(HPF: high-pass filter)와 저대역 필터(LPF: low-pass filter)를 저대역 필터의 출력 신호들에게 반복 적용하여 다해상도 신호를 얻는다. Dyadic 구조에서 m 번 LPF/HPF를 적용한 후 출력되는 신호를 각각 m 번째 레벨(level)에서의 저주파 및 고주파 성분이라고 한다. 이러한 경우, 분석된 신호들의 위상 특성을 원 신호와 동일하게 유지시키고, 분석/합성 후의 신호가 원 신호와 동일한 값을 갖게 하려면, 선형 위상의 쌍직교 필터(bi-orthogonal filter)를 사용하여야 한다^[3]. 쌍직교 필터는 영상 신호처럼 변환 후에도 원 신호의 위상을 보존해야 하는 신호를 DWT 할 때 사용된다. 쌍직교 필터의 대표적인 예로는 Daubeshies의 9-3 필터인데, 이는 MPEG-4에서 기본적인 DWT 필터로 채택되어 있다^[5].

한편, DWT는 LPF/HPF를 반복 적용하여 수행되기 때문에 계산량이 많다. 따라서 DWT를 VLSI 시스템으로 구현하여 실시간 동작시키려는 노력이 많이 이루어져왔다^[6-10]. DWT를 위한 VLSI 시스템 설계시 기본적으로 고려되는 사실은 필터링 후 간축(subsampling)이 이루어진다는 점이다. 즉, 각 레벨에서의 입력 신호는 필터링 후, 1/2 간축된 고주파 및 저주파 성분을 얻는다는 사실을 참조하면, 연속적으로 입력되는 신호에 대하여 매 두 번째 입력되는 신호들만으로 필터링하면 된다. m 번째 레벨에서의 입력 신호는 원 입력 신호에 비해 $(1/2)^{m-1}$ 간축되어 있다는 것을 참조한다면, m 번째 레벨에서의 LPF/HPF 사용 효율은 $100 \times (1/2)^m \%$ 이다. 따라서, dyadic 구조로 DWT를 수행한다면, 하나의 LPF/HPF만으로 전체 DWT를 수행할 수 있다는 것을 알 수 있다. 이때, M 레벨 DWT를 수행한다면, LPF/HPF의 사용 효율은 $100 \times (1 - (1/2)^M) \%$

이다. Knowles^[6]는 이러한 개념을 DWT 구현에 처음 소개하였으나, 복잡한 다중화기, 전역 연결 등을 포함하고 있어 실제 하드웨어 복잡도가 높다는 단점이 있다. 이를 개선하는 한 방법으로 Parhi와 Nishitani는 folded 구조^[7], 그리고 Vishwanath등은 RPA(recursive pyramid algorithm)를 이용한 구조^[8] 등으로 발전시켰다. Folded 혹은 RPA 방식은 하나의 LPF/HPF 군을 이용하여 다중 레벨 DWT를 수행시키므로, LPF/HPF 군에 적절히 입력을 제공하여야 한다. 따라서, 기억 소자가 많이 필요하고, 복잡한 라우팅(routing)이 필요하다. 이러한 문제를 해결하기 위하여 Vishwanath등^[8]은 systolic, semi-systolic 혹은 메모리 기반 라우팅(memory-based routing) 등을 제안하여, LPF/HPF 군에 각 레벨에 대한 연산에 필요한 입력을 제공하도록 한다. 그리고, Parhi와 Nishitani^[7]는 life-time 분석을 통하여 최소의 기억 소자로 DWT를 구현하는 방법을 제안하였다. 그러나, Parhi와 Nishitani가 제안한 구조^[7]는 필터의 길이나 레벨의 수가 달라지면 DWT 구조를 새로이 설계해야 하므로, 시스템의 유연성이 낮다는 단점이 있다.

그리고, [9]에서는 LPF와 HPF를 동일한 연산자로 수행하되, 해상도가 낮아도 되는 첫 번째 레벨을 위한 LPF/HPF와 높은 해상도를 요구하는 상위 레벨들을 위한 LPF/HPF로 나누어 처리하는 반순환적(semi-recursive) 구조를 제안하여 실제 하드웨어 복잡도를 감축시켰다. 또한, [10]에서는 Daubechies의 4-탭(tap) 필터를 이용한 DWT를 효율적으로 구현하는 방법을 제안하였으나, 일반적인 필터로 확장하는 데는 한계가 있다. 지금까지 제안된 이러한 방법들은 요구되는 곱셈기가 필터의 탭수의 2배 정도 필요하므로, 하드웨어 복잡도가 여전히 높다는 단점이 있다. 특히 MPEG-4^[5]에서는 최대 15-탭의 필터를 사용할 수 있어야 한다는 점을 고려할 때, 기존의 방법들은 1차원 DWT를 위해 최대 30개의 곱셈기가 필요하여 대단히 복잡한 시스템이 됨을 알 수 있다. 따라서, 실제 DWT 구현을 위해서는, 필터의 특성을 고려하여 하드웨어 복잡도를 크게 감축시킬 수 있는 보다 효율적인 DWT 구조가 필요하다.

본 논문에서는 영상 신호에 적합한 DWT 필터는 홀수 탭(odd-tap) 선형 위상 필터이라는 점을 참조하여 이를 구현하는 효과적인 방법에 대하여 살펴본다. 그리고, 일반적으로 복호기가 널리 사용되므로 IDWT에 대한 구조에 중점을 두었다. 그렇지만 본 논문에서 제시한 방법을 이용한다면 DWT에 대한 구조도 쉽게 얻을 수 있다. 본 논문

에서 제안한 방법에서는, 먼저, 선형 위상 필터의 계산 구조를 이용하여, 대칭적 위치에 있는 입력들을 먼저 더한 다음 곱셈을 수행함으로써 곱셈기의 수를 약 절반으로 줄인다. 이때 발생하는 전역 연결(global interconnection)을 해결하기 위하여 입력의 흐름을 U자형으로 만듦으로써 국부적인 연결로 필터를 구현한다. 제안된 필터는 지연 소자부, 연산부, 덧셈부, 그리고 후처리부 등으로 이루어지며, 필터의 길이 변화에 따라 각 부분을 추가/삭제함으로써 원하는 필터를 구현할 수 있다. 그리고, 제안된 합성 필터를 직렬 연결 혹은 반순환적 연결을 통해 M 레벨 IDWT를 구현할 수 있다. 본 논문에서 설계된 필터를 이용한다면, dyadic 구조의 DWT를 효과적으로 구현할 수 있으며, 곱셈기 등 연산자의 수를 감축시킴으로써 보다 간단한 시스템으로 구현할 수 있음을 보이겠다.

II. DWT

DWT^[3]는 신호를 다해상도로 분석하는 도구로서, 시간 및 주파수 영역에서 적층적으로 신호를 분석할 수 있다. DWT를 이용한 신호 분석에는 대역 통과 필터와 같은 특성을 보이는 모 웨이브렛과 LPF의 특성을 갖는 scaling 함수라는 두 개의 함수를 이용한다. DWT에서의 기저함수인 웨이브렛 함수 $\psi_{m,n}(\cdot)$ 는 다음과 같이 모 웨이브렛 $\Psi(\cdot)$ 을 팽창/수축 및 이동시켜 얻는다.

$$\psi_{m,n}(x) = 2^{-m/2} \Psi(2^{-m}x - n) \tag{1}$$

여기서 m 은 scale 값이고, n 은 이동(translation) 값이다. 마찬가지로 scaling 함수 $\phi(\cdot)$ 도 팽창/수축 및 이동시킨 함수 $\phi_{m,n}(x)$ 를 얻을 수 있다. 주어진 m 에 대하여 $\phi_{m,n}(\cdot)$ 에 의해 정의되는 공간 V_m 은 2^m 의 해상도를 가지며, 이러한 V_m 들은 $\dots \subset V_3 \subset V_2 \subset V_1 \subset \dots$ 과 같은 관계를 가지는 점진적 근사 공간(successive approximation space)으로 설명된다. 주어진 m 에 대하여 $\phi_{m,n}(\cdot)$ 에 의해 정의되는 공간 W_m 은 V_{m-1} 에서 V_m 의 직교 여공간(orthogonal complement space)이 된다. 점진적 근사 특성에 따라 DWT는 그림 1과 같은 dyadic 구조를 가질 수 있으며, LPF $\{h(n)\}$ 과 HPF $\{g(n)\}$ 를 반복적으로 적용함으로써 다해상도로 분석된 신호들을 얻을 수 있다. 그림 1에서는 3-레벨(level) DWT 분석 및 합성의 예를 보인 것이다. 본 논문에서는 해상도 레벨을 M 으로 표시하였는데,

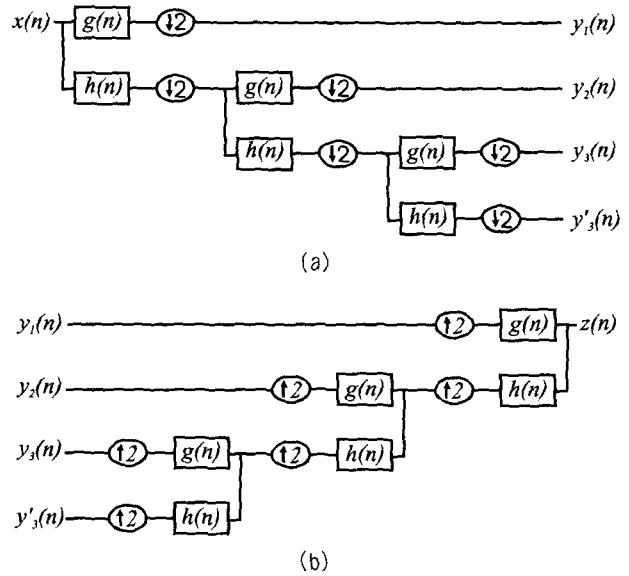


그림 1. 3-레벨 DWT; (a) 분석; (b) 합성.
Fig. 1. 3-level DWT; (a) analysis; (b) synthesis.

그림 1의 경우 $M=3$ 이 된다. 먼저, $x(n)$ 이 입력되면, 첫 번째 레벨에서 저주파 성분 $y'_1(\cdot)$ 과 고주파 성분 $y_1(\cdot)$ 이 생성된다. $y'_1(\cdot)$ 은 다시 두 번째 레벨에서 고주파 성분 $y_2(\cdot)$ 과 저주파 성분 $y'_2(\cdot)$ 으로 분해된다. 이러한 과정을 거쳐 3-레벨 DWT는 입력을 각 레벨에서의 고주파 성분 $y_1(\cdot), y_2(\cdot), y_3(\cdot)$ 과 세 번째 레벨에서의 저주파 성분 $y'_3(\cdot)$ 등 4 성분으로 분해한다. DWT 합성은 분석의 역과정으로, 각 레벨의 고주파 성분과 세 번째 레벨에서의 저주파 성분을 이용하여 원래의 신호를 합성하게 된다.

한편, 영상 신호와 같이 매끄러운(smooth) 특성을 가지는 신호를 Dyadic 구조와 같이 반복적인 필터링으로 DWT를 수행하고자 한다면, 완전 복원 성질을 가진 선형 위상 FIR 필터를 사용하여야 한다. 이러한 필터는 직교성을 완하시킴으로써 얻을 수 있는데, 이때의 필터를 쌍직교 필터(biorthogonal filter)라고 한다^[3]. 쌍직교 필터를 이용할 때, 2^{m-1} 해상도의 저주파 신호 $y'_{m-1}(\cdot)$ 로부터 2^m 해상도의 저주파 성분 $y'_m(\cdot)$ 과 고주파 성분 $y_m(\cdot)$ 은 각각 다음과 같이 얻을 수 있다.

$$y_m(n) = \sum_k g(2n-k)y'_{m-1}(k) \tag{2}$$

$$y'_m(n) = \sum_k h(2n-k)y'_{m-1}(k)$$

그리고, 2^m 해상도의 두 성분 $y'_m(\cdot)$ 과 $y_m(\cdot)$ 로부

터 $y'_{m-1}(\cdot)$ 은 다음과 같이 복원된다.

$$y'_{m-1}(n) = \sum_k [\tilde{h}(2k-n)y'_m(k) + \tilde{g}(2k-n)y_m(n)] \quad (3)$$

여기서 $\{\tilde{h}(\cdot)\}$ 과 $\{\tilde{g}(\cdot)\}$ 은 다음을 만족하는 필터이다.

$$\tilde{g}(n) = (-1)^n h(-n+1), \quad g(n) = (-1)^n \tilde{h}(-n+1) \quad (4)$$

쌍직교 필터의 대표적인 예로는 Daubeshies의 9-3 필터인데, 이는 MPEG-4에서 기본적인 DWT 필터로 채택되어 있다^[5].

III. 제안하는 합성 필터

본 절에서는 본 논문에서 제안하는 DWT 구조에 대하여 설명한다. 본 논문에서 설계하는 DWT는 영상 신호에 적합한 것으로 알려진 쌍직교 성질을 갖는 선형 위상 FIR 필터를 사용하는 것으로 가정한다. 그리고, MPEG-4 등에 쉽게 적용할 수 있도록 필터의 크기나, 웨이브렛 레벨의 수에 대해 유연성을 가질 수 있도록 하였고, 일반적으로 복호기가 많이 사용된다는 점을 참조하여 IDWT를 구현하는 방법에 대하여 설명한다. 그러나, 이는 DWT에도 쉽게 적용할 수 있다.

일반적으로 IDWT는 LPF/HPF로 이루어지기 때문에 일반적인 필터 구현 기법으로 IDWT를 구현할 수 있다. 그러나, IDWT가 up-sampling 후에 필터링한다는 것을 참조한다면, 계산량을 반으로 줄일 수 있다. 필터의 크기가 $L=7$ 일 때, m 번째 레벨에서의 IDWT 연산은 다음과 같다.

$$\begin{aligned} y'_{m-1}(0) &= \frac{g(-2)y_m(-2) + g(0)y_m(0) + g(2)y_m(2)}{h(-2)y'_m(-2) + h(0)y'_m(0) + h(2)y'_m(2)} \\ y'_{m-1}(1) &= \frac{g(-3)y_m(-2) + g(-1)y_m(0) + g(1)y_m(2) + g(3)y_m(4)}{h(-3)y'_m(-2) + h(-1)y'_m(0) + h(1)y'_m(2) + h(3)y'_m(4)} \\ y'_{m-1}(2) &= \frac{g(-2)y_m(0) + g(0)y_m(2) + g(2)y_m(4)}{h(-2)y'_m(0) + h(0)y'_m(2) + h(2)y'_m(4)} \\ y'_{m-1}(3) &= \frac{g(-3)y_m(0) + g(-1)y_m(2) + g(1)y_m(4) + g(3)y_m(6)}{h(-3)y'_m(0) + h(-1)y'_m(2) + h(1)y'_m(4) + h(3)y'_m(6)} \\ y'_{m-1}(4) &= \frac{g(-2)y_m(2) + g(0)y_m(4) + g(2)y_m(6)}{h(-2)y'_m(2) + h(0)y'_m(4) + h(2)y'_m(6)} \end{aligned} \quad (5)$$

여기서 시간 색인 n 은 알아보기 쉽도록 인수 2^{m-1} 을 삭

제하였다. 예를 들어, $y_m(2)$ 은 실제로 $y_m(2 \cdot 2^{m-1})$ 이 되고, 이때, 입력 신호의 주기를 나타내는 클럭(clock)을 2^{m-1} 분주한 것을 m 번째 레벨에서의 기준 클럭으로 정의한다. 즉, m 번째 레벨에서의 주기(cycle)는 입력 주기(input cycle)를 2^{m-1} 배 한 것이다. 본 논문에서는 선형 위상 필터, 즉 대칭 필터(symmetric filter)만을 고려하고 있으므로, $g(n) = g(-n)$, $h(n) = h(-n)$ 이 된다. 따라서, 식 (5)는 다음과 같이 고쳐 쓸 수 있다.

$$\begin{aligned} y'_{m-1}(0) &= \frac{g(2)[y_m(-2) + y_m(2)] + g(0)y_m(0)}{h(2)[y'_m(-2) + y'_m(2)] + h(0)y'_m(0)} \\ y'_{m-1}(1) &= \frac{g(3)[y_m(-2) + y_m(4)] + g(1)[y_m(0) + y_m(2)]}{h(3)[y'_m(-2) + y'_m(4)] + h(1)[y'_m(0) + y'_m(2)]} \\ y'_{m-1}(2) &= \frac{g(2)[y_m(0) + y_m(4)] + g(0)y_m(2)}{h(2)[y'_m(0) + y'_m(4)] + h(0)y'_m(2)} \\ y'_{m-1}(3) &= \frac{g(3)[y_m(0) + y_m(6)] + g(1)[y_m(2) + y_m(4)]}{h(3)[y'_m(0) + y'_m(6)] + h(1)[y'_m(2) + y'_m(4)]} \\ y'_{m-1}(4) &= \frac{g(2)[y_m(2) + y_m(6)] + g(0)y_m(4)}{h(2)[y'_m(2) + y'_m(6)] + h(0)y'_m(4)} \end{aligned} \quad (6)$$

식 (6)을 보면, $y'_{m-1}(2n)$ 과 $y'_{m-1}(2n+1)$ 은 동일한 입력들을 처리하므로, 동시에 계산되어질 수 있음을 알 수 있다. 또한, 짝수번째 입력들만이 계산에 사용되므로, 고주파 입력 $y_m(\cdot)$ 과 저주파 입력 $y'_m(\cdot)$ 을 번갈아 입력시켜 동일한 연산기를 통해 고주파 및 저주파 연산을 매 두 번째 주기마다 수행시킬 수 있다. 이와 같은 사실을 참조하여 식(6)을 블록 선도로 나타내면 그림 2와 같다. 그림 2의 구조에서 기본적인 연산은 두 입력을 더하고, 그 결과를 필터 계수와 곱한 후 합하는 '덧셈-곱셈-덧셈'이다. 매 주기(cycle)마다 고주파 성분 $y_m(\cdot)$ 과 저주파 성분 $y'_m(\cdot)$ 이 번갈아 들어오는 입력들은 그림 2의 좌측에

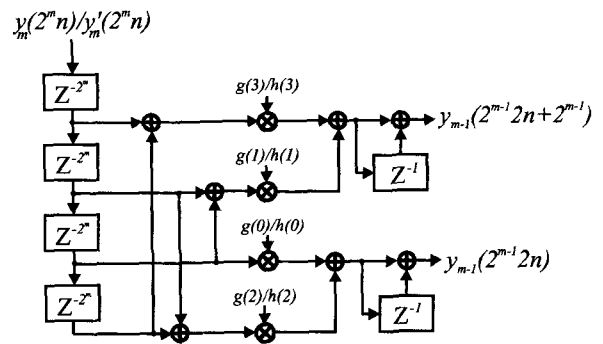


그림 2. 7-탭 선형 위상 필터의 블록선도.
Fig. 2. A block diagram of the 7-tap linear phase filter.

위치한, Z^{-2^m} 지연 소자(delay)로 구성된 지연 소자군(delay line)을 거친다. 만약 m 번째 레벨의 기준 클럭을 사용한다면, 이 지연 소자는 2개의 직렬 연결된 레지스터(register)로 구현될 수 있다. 그림 2의 우측에 있는 지연 소자 Z^{-1} 및 덧셈은 매 주기마다 번갈아 출력되는 고주파 및 저주파 연산 결과를 합하여 $m-1$ 번째 레벨의 저주파 신호 $y'_{m-1}(\cdot)$ 를 만들기 위한 것이다. 그림 2에 제시한 구조는 $\lfloor (L+1)/2 \rfloor$ 개의 곱셈기와, 입력들을 더하는 데 $\lfloor (L-1)/2 \rfloor$ 개의 덧셈기, 곱셈 결과를 합하는 데 $\lfloor (L+1)/2 \rfloor$ 개의 덧셈기만으로 m 번째 레벨의 합성 연산을 구현할 수 있다. 그러나, 입력을 더하는 부분에서 전역 연결(global interconnection)이 필요하다는 단점이 있다.

그림 2의 구조에서 전역 연결이라는 단점을 해결하고, 한 주기동안 '덧셈-곱셈(add and multiply)'만을 수행하도록 시간 조정(time scaling)한 구조를 그림 3에 나타내었다. 그림 3에서는 15-탭 이하의 필터를 이용할 때 ($L \leq 15$), m 번째 레벨의 합성을 수행할 수 있는 합성 필터(synthesis filter)의 구조를 제시한 것이다. 그림 3의 구조는 지연 소자부(delay unit), 연산부(operator unit), 덧셈부(adder unit), 그리고 후처리부(post unit) 등 4부분으로 구성된다. 합성 필터는 1개의 후처리부와, 필터 크기(filter length)을 L 이라고 할 때 각각 $\lfloor L/4 \rfloor$ 개의 나머지 부분들로 구성된다. 먼저, 입력으로 m 번째 레벨의 고주파 성분 $y_m(\cdot)$ 과 저주파 성분 $y'_m(\cdot)$ 이 매 주기마다 번갈아 입력되며, 이 입력은 매 2주기($2 \times 2^{m-1}$ 화소 주기)마다 좌측의 지연 소자군의 지연 소자로 이동된다. 여기서, 지연 소자를 U자 모양으로 배치함으로써, 각 부분들이 국부적으로 연결되도록 하였다. 즉, 지연 소자부 내부는 복잡한 연결로 이루어지지만 동일한 구조의 규칙적 배열로 지연 소자군을 구성할 수 있게 된다. 또한, 인접한 부분간에 규칙적으로 연결되므로 설계가 용이하다는 장점이 있다.

연산부에서는 '덧셈 후 곱셈'이라는 기본 연산을 수행한다. 후단에 지연 소자를 둬으로써 한 주기동안 기본 연산을 수행하도록 하여 시간 조정이 용이하도록 한다. 여기서 주목할 사항은 연산부에 있는 지연 소자는 지연 소자부에 있는 것과는 달리 Z^{-1} 즉 한 주기(입력 주기 혹은 m 번째 레벨의 주기)만큼 지연시키는 것이라는 점이다. 즉, 이때의 지연 소자는 '덧셈 후 곱셈' 연산을 수행할 수 있는 시간을 확보하기 위한 것이다. 각 연산부는 2개의 곱셈기

와 2개의 덧셈기로 구성되어 있어, 좌측의 지연 소자부로부터 3개의 입력을 받아들여, 짝수 연산(정수 k 에 대하여 $y'_{m-1}(2k)$ 를 얻기 위한 연산) 및 홀수 연산(정수 k 에 대하여 $y'_{m-1}(2k+1)$ 을 얻기 위한 연산)을 동시에 수행한다. 반면에 저주파 필터링과 고주파 필터링은, 동일한 연산자를 이용하여, 매 주기마다 교대로 수행된다.

덧셈부에서는 연산부에서 얻어진 짝수 연산 및 홀수 연산 결과를 각각 합한다. 여기서 홀수 연산의 마지막 덧셈기에 있는 스위치는 $(M+1)/2$ 가 짝수인 경우($L=3,7,11,15,\dots$) ON 되고, 그렇지 않은 경우 OFF되는 것으로 다양한 필터들을 수용할 수 있게 한다. 덧셈부에서는 매 주기마다 저주파 출력 및 고주파 출력이 교대로 계산된다. 후처리부에서는 지연 소자 및 덧셈기로 저주파 출력과 고주파 출력을 합하여 $(m-1)$ 레벨 저주파 성분 $y'_{m-1}(\cdot)$ 을 생성시킨다. 그런 다음, 동시에 생성되는 짝수 출력 및 홀수 출력을 다중화(multiplex)하여 매 2^{m-1} 화소 주기로 $y'_{m-1}(\cdot)$ 이 출력되도록 한다. 여기서 홀수 출력 $y'_{m-1}(2k+1)$ 의 마지막 부분에 설치된 지연 소자 $Z^{-1 \cdot 2^{m-1}}$ 은 홀수 출력을, 동시에 얻는 짝수 출력보다 2^{m-1} 화소 주기만큼 지연시켜 출력시간을 맞추기 위한 것이다.

그림 3에 설치한 지연 소자는 입출력 순서를 맞추기 위한 지연 소자(Z^{-2^m} 과 $Z^{-1 \cdot 2^{m-1}}$)들과 연산을 시간 조정하기 위한 것(Z^{-1})들 등 2가지 종류가 있다. 만약 입력 주기의 클럭으로 합성 필터를 구동시킨다면, 두종류의 지연 소자는 서로 다른 것이다. 따라서 지연 소자부의 Z^{-2^m} 은 2^m 개의 직렬 연결된 레지스터로 구현되며, 연산은 화소율과 같은 속도로 연산부 및 덧셈부의 연산이 수행된다. 그러나, 만약 입력 클럭을 2^{m-1} 분주한 클럭(m 번째 레벨에서의 기준 클럭)으로 합성 필터를 구동시킨다면, $Z^{-2^{m-1}}$ 은 Z^{-1} 과 동일하다. 이때에는 지연 소자부의 Z^{-2^m} 은 두 개의 직렬 연결된 레지스터로 구현되며, 연산부 및 덧셈부의 연산은 m 레벨에서의 주기 단위로 수행된다.

그림 3의 합성 필터에서 입력은 $y_m(\cdot)$ 과 $y'_m(\cdot)$ 이 매 주기마다 번갈아 입력된다. 그리고, 길이가 L 인 필터를 사용한다고 하면, 지연 소자부에서 연산부에 필요한 입력을 제공하기 위해서는 첫 입력이 인가된 후 $(L-1)2^{m-1}$ 입력 주기의 시간이 소요된다. 그리고, 연산부에서의 연산, 덧셈부에서의 연산, 그리고 고주파 필터링과 저주파 필터링 결과를 합하는 데, 각각 1 입력 주기의 시간이 필요하다. 따라서, 합성 필터의 대기시간(latency)은 $(L-1)2^{m-1}+3$

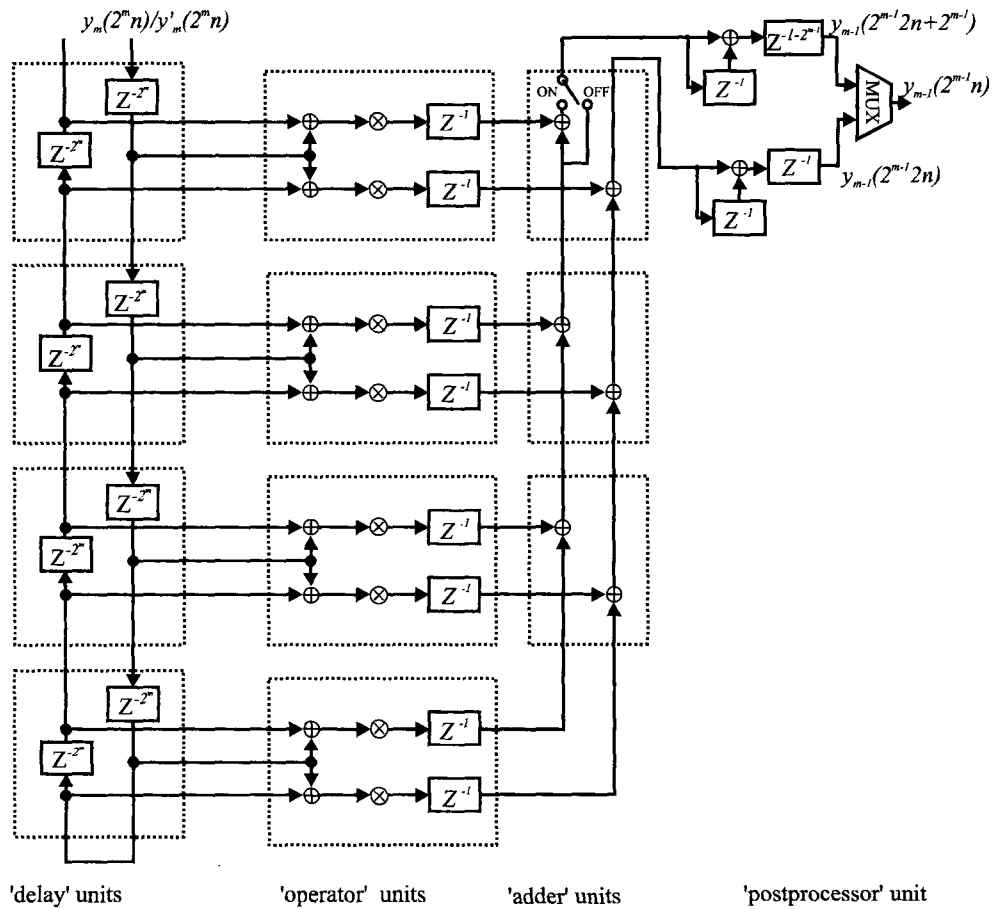


그림 3. IDWT를 위한 15-탭 합성 필터의 구조.
Fig. 3. An architecture for the 15-tap synthesis DWT filter.

입력 주기가 된다. 반면에 합성 필터에서는 매 2^{m-1} 입력 주기마다 $y_m(\cdot)$ 과 $y'_m(\cdot)$ 을 교대로 입력시켜, 매 2^{m-1} 입력 주기마다 새로운 출력 $y'_{m-1}(\cdot)$ 이 출력되므로, 그림 3의 합성 필터는 m 레벨 IDWT 연산을 실시간으로 처리할 수 있다.

IV. 합성필터군으로의 확장

그림 3에 보인 합성 필터는 m 번째 레벨의 신호들을 합성하는 구조로, 이를 이용하면 전체 합성 필터군 (synthesis filter bank)을 구현할 수 있다. 이 구현 방법은 크게 직렬 구조와 반순환적 구조로 나눌 수 있다. 이제, 이 두 종류의 구조들을 좀 더 자세히 설명하겠다.

1. 직렬 구조

직렬 구조(cascaded architecture)는 그림 4에 제시한 바와 같이 IDWT의 각 레벨마다 별도의 합성 필터를 사용하여 IDWT를 구현하는 것이다. m 번째 레벨의 합성 필터에는 입력 클럭을 2^{m-1} 분주한 클럭에 의해 구동되며, 이때, 각 지연 소자 Z^{-2^m} 은 2개의 레지스터로 구현된다. 이 구조는 다중화기(MUX) 이외의 부가 로직이 필요 없고, 규격화된 구조를 규칙적으로 배열함으로써 구현할 수 있다는 장점이 있다. 그러나, 레벨 수만큼의 합성 필터가 요구되므로, 복잡하고, 레벨 수가감에 따른 유연성이 저하된다는 단점이 있다. 또한, M 레벨 합성 필터군을 구현할 때 필터의 사용 효율은 다음과 같이 주어지,

$$\sum_{k=1}^M \frac{2^{k+1}}{M} = \frac{2}{M} (1-2^{-M}) \quad (7)$$

레벨 수가 증가함에 따라 필터의 사용 효율이 급격히 저

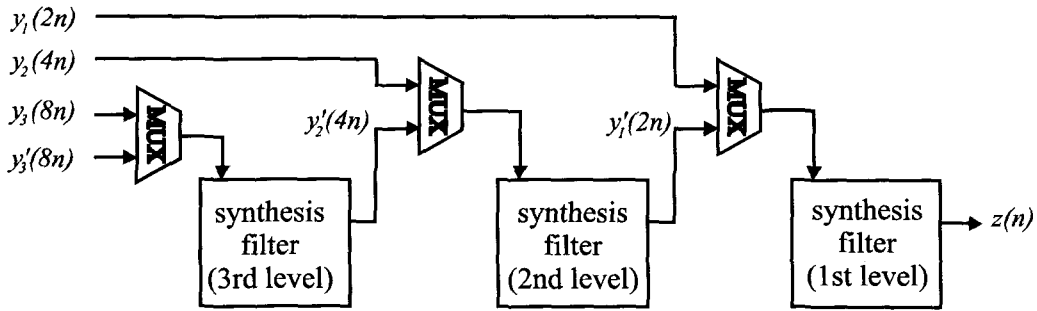


그림 4. 직렬 연결을 통한 3-레벨 IDWT의 구현.
Fig. 4. A cascaded implementation of the 3-level IDWT.

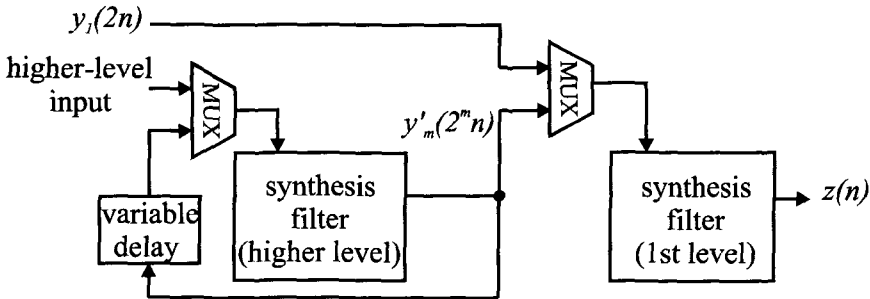


그림 5. 반순환적 형태의 3-레벨 IDWT의 구현.
Fig. 5. Implementation of the 3-level IDWT in the semi-recursive form.

하된다는 단점이 있다.

앞에서 설명한 바와 같이 합성 필터의 대기시간은 $(L-1)2^{m-1}+3$ 입력 주기이고, 직렬 구조는 부가적인 지연 요소가 없으므로, M 레벨 합성 필터군을 구현할 때의 대기시간은

$$\sum_{m=1}^M \{(L-1)2^{m-1}+3\} = (L-1)(2^M-1)+3M \quad (8)$$

로 표시된다. 만약 MPEG-4의 기본 예와 같이 $L=15$ 이고, $M=3$ 이라면 대기시간은 107 화소 주기가 된다.

2. 반순환적 구조

이 구조는 IDWT의 점진적 근사 특성을 이용하여, 하나의 필터를 여러 레벨의 필터링이 시분할하여 사용하도록 하는 것이다. 그림 5에 제시한 바와 같이, 첫 번째 레벨은 하나의 필터에 의해 구현되고, 나머지 상위 레벨은 또 다른 필터에 의해 구현되어 총 2개의 필터로 전체 IDWT를 수행하도록 한다. 2개의 필터는 입력 주기를 나타내는 입

력 클럭에 의해 구동된다. 첫 번째 레벨의 합성 필터에서는 지연 소자 Z^{-2^m} 은 2개의 레지스터로 구성되며, 매 주기마다 번갈아 입력되는 $y_1(\cdot)$ 와 $y_1'(\cdot)$ 으로부터 복원된 신호 $z(\cdot)$ 을 출력한다.

상위 레벨을 위한 합성 필터에서 지연 소자부는 상위 레벨에서 처리해야 하는 각 레벨마다 별도의 지연 소자부를 가져야 한다. 만약 레벨의 개수가 M 이라면, 2, 3, ..., M 레벨에 대한 지연 소자부를 모두 가지고 있어야 한다. 그리고, 각 레벨의 지연 소자부에 각 레벨에 알맞은 기준 클럭이 제공된다고 하면, 지연 소자 Z^{-2^m} 은 2개의 레지스터로 구현될 수 있어, 지연 소자부는 모두 $(L-1)(M-1)$ 개의 레지스터로 구성된다.

각 레벨의 저주파 및 고주파 신호가 연산부에 입력되는 순서는 그림 6의 순환적 연산 순서를 따른다. 그림 6에서 H 및 L은 각각 해당 레벨의 고주파 및 저주파 성분을 나타낸다. 제안한 합성 필터에서는 고주파 필터링 및 저주파 필터링 결과를 합하여 출력을 만들기 때문에, 해당 레벨의 고주파 및 저주파 입력을 연속적으로 인가하여야 한다. 그리고, 하나의 LPF/HPF만으로 모든 상위 레벨에서의 연산들을 실행시키므로, 특정 주기에서는 한 레벨에 대한 입

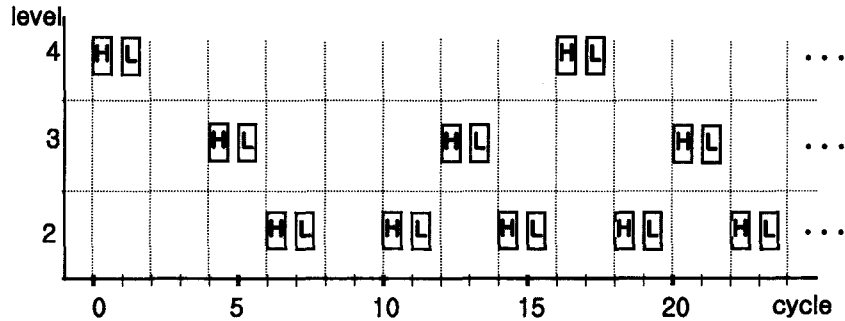


그림 6. 상위 레벨 합성 필터에서 순환적 연산 순서.
 Fig. 6. Schedule of the recursive computation at the higher order synthesis filter.

력만이 제공되어야 한다. 이러한 사실을 참조하여, 그림 6에 제시된 바와 같이, 매 $16n$ 및 $16n+1$ 주기(cycle)에서 네 번째 레벨의 고주파 및 저주파 성분이 각각 입력되고, 매 $8n+4$ 및 $8n+5$ 주기에서는 세 번째 레벨의 고주파 성분 및 네 번째 레벨에서 산출되는 출력인 저주파 성분이 각각 입력된다. 그리고, 매 $4n+6$ 및 $4n+7$ 주기에서는 두 번째 레벨의 고주파 성분 및 세 번째 레벨에서 산출되는 출력인 저주파 성분이 각각 입력된다.

m 번째 레벨 입력이 인가될 때, 연산부 및 덧셈부에서는 m 번째 레벨에 대한 연산을 수행한다. 연속적으로 계산되어지는 고주파 및 저주파 필터링 결과를 합하여 m 번째 레벨에서의 연산 결과로서 2개의 $(m-1)$ 번째 레벨 저주파 성분을 얻는다. 이 연산 결과는 $m=2$ 인 경우 첫 번째 레벨을 위한 합성 필터로 제공되며, $m>2$ 인 경우에는 다시 차하위 레벨을 위한 합성 필터로 인가된다. 이때, 각 레벨의 출력은 그림 6의 순환적 연산 순서에 따라 입력되어야 하므로 가변 지연 소자(variable delay)를 설치하여 입력 순서를 맞춘다. 예를 들면, 4번째 레벨 연산 결과 $y'_3(\cdot)$ 는 4번째 고주파 성분 $y_4(\cdot)$ 이 입력된 지 3주기 후에 출력되지만, 3번째 레벨 저주파 연산은 $y_4(\cdot)$ 이 입력된 지 5주기 후에 수행되므로, $y'_3(\cdot)$ 는 2주기만큼 지연된 다음 입력되어야 한다. 반면에, 3번째 레벨 연산 결과 $y'_2(\cdot)$ 는 2번째 레벨 연산에 바로 사용되므로, 별도의 지연 없이 입력으로 인가된다.

그림 5에 제시한 반순환적 구조는 첫 번째 레벨을 위한 합성 필터는 매 화소 주기에서 연산이 수행되고, 상위 레벨을 위한 합성 필터는 레벨의 개수 M 이 2, 3, 4일 때 각각 50%, 75%, 그리고 87.5%의 계산 점유율을 보이므로, 합성 필터의 사용 효율은 점근적으로 100%가 된다. 반순환적 구조에서의 대기시간은 식 (8)에 제시된 것과 거의 유사하다. 차이점은 반순환적 구조인 경우 순환적으

로 연산이 수행되기 때문에 가변 지연 소자에 의해 발생하는 지연만큼 추가적인 시간이 소요된다. 이 추가 시간은 위에서 살펴본 바와 같이 4번째 레벨 연산에서는 2 화소 주기, 3번째 레벨 연산에서는 0 화소 주기 등이 된다. 여기서 3번째 레벨 이하에서는 가변 지연 소자에 의한 지연이 0이므로, $M=3$ 일 때 반순환적 구조에서의 대기시간은 직렬 구조에서의 대기시간과 동일하다. 또한, 상위 레벨을 위한 합성 필터는 임의의 M 을 처리할 수 있으므로, 보다 유연한 구조를 가지는 것으로 볼 수 있다. 그러나, 지연 소자부가 $(M-1)$ 개의 개별적인 지연 소자부의 조합으로 이루어져 있고, 가변 지연부가 추가되는 등 추가적인 회로가 필요하다는 단점이 있다.

V. 성능 평가

본 논문에서 제안한 합성 필터를 사용한 직렬 및 반순환적 구조의 성능을 H/W 복잡도 측면에서 평가한 결과를 표 1에 제시하였다. 표 1은 L-탭의 LPF/HPF를 이용하여 M 레벨로 분해된 신호들을 합성하는 경우에 각 구조에서 요구되는 곱셈기 및 덧셈기의 개수를 비교한 것이다. 이때, IDWT 후 연산 오차에 의한 영향을 최소화할 수 있도록, 곱셈 및 덧셈은 16-bit의 해상도로 연산되는 것을 가정하였다^[9]. 표 1에 나타난 바와 같이 제안한 합성 필터를 사용한 반순환적 구조는 기존의 방법에 비해 덧셈기는 유사한 개수가 필요하지만, 약 반 정도의 곱셈기로 구현될 수 있다. 예를 들어 $L=15$ 이고, $M=3$ 인 경우, Parhi의 구조 및 Vishwanath의 구조에서는 각각 30개의 곱셈기와 덧셈기가 필요하나, 제안한 합성 필터를 사용한 반순환적 구조에서는 16개의 곱셈기와 30개의 덧셈기만으로 구현된다. 또한, 제안한 합성 필터는 각 부분이 규칙적으로 배열

표 1 하드웨어 복잡도 비교

Table 1. Comparison of hardware complexities

	multiplier	adder	routing
cascaded	$M \cdot \lfloor \frac{L+1}{2} \rfloor$	$M \cdot (\lfloor \frac{L-1}{2} \rfloor + \lfloor \frac{L+1}{2} \rfloor)$	simple
semi-recursive	$2 \cdot \lfloor \frac{L+1}{2} \rfloor$	$2 \cdot (\lfloor \frac{L-1}{2} \rfloor + \lfloor \frac{L+1}{2} \rfloor)$	simple
Parhi	$2L$	$2L$	complex
Vishwanath	$2L$	$2L$	simple

되고, 국부적인 연결로 이루어져 있다. 따라서, 라우팅이 간단하고, 필터의 길이 및 레벨 수가 변하더라도 그 구조를 쉽게 조정할 수 있게 된다.

제안한 합성 필터를 사용한 반순환적 구조는 VHDL 레벨에서 그 동작을 검증하여 IDWT 연산을 적절히 수행함을 확인하였다. 만약 화소 주기동안에 '덧셈과 곱셈' 연산을 수행한다면, 제안한 방법은 M 레벨 IDWT 연산을 실시간으로 수행할 수 있다. 이때, 대기시간은 가변 지연에 의한 영향을 무시한다면, 식 (8)에 제시한 바와 같이 $(L-1)(2^M-1)+3M$ 입력 주기가 된다. 예를 들어 $L=15$, $M=3$ 인 경우, 가변 지연이 0이므로 대기시간은 107 화소 주기가 된다. 이는 기존의 Parhi 및 Vishwanath의 구조와 동일한 크기의 대기시간이다. 이러한 점을 살펴볼 때, 제안한 합성 필터를 사용한 반순환적 구조는 곱셈기의 수를 반으로 줄이면서 기존의 방법과 동일한 계산 성능을 보임을 알 수 있다.

VI. 결론

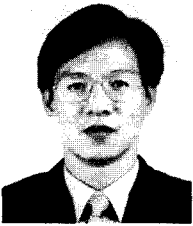
본 논문에서는 영상 신호 처리에 적합한 홀수 탭의 선형 위상 필터를 사용하는 IDWT의 VLSI 구조를 제시하였다. 제안한 구조의 합성 필터는 필터의 대칭적 관계를 이용하고, 지연 소자들을 U자형으로 배열함으로써 전역 연결을 국부적 연결로 변환한 것으로 기존의 구조에 비해 약 반 정도의 곱셈기만으로 IDWT를 구현한다. 합성 필터는 지연 소자부, 연산부, 덧셈부, 그리고 후처리부등으로 이루어져 있으며, 필터의 길이 변화에 따라 각 부분을 추가/삭제함으로써 원하는 필터를 구현할 수 있다. 제안한 합성 필터를 직렬 연결 혹은 반순환적 연결을 통해 M 레벨 IDWT를 구현할 수 있다. 이 가운데 반순환적 구조는

합성 필터의 지연 소자부를 M개 설치함으로써 IDWT를 구현할 수 있어 임의의 레벨 수의 IDWT 구현이 용이할 뿐만 아니라 기존의 방법에 비해 약 반 정도의 곱셈기만으로 동일한 계산 성능을 얻을 수 있다. 따라서, 본 논문에서 제안한 IDWT 구조는 MPEG-4 등 영상 신호 처리에 효과적으로 적용시킬 수 있을 것으로 기대된다.

참 고 문 헌

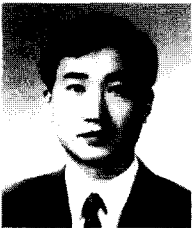
- [1] R. Steinmetz and K. Nahrstedt, *Multimedia: Computing, Communications and Applications*, Prentice-Hall International, Inc., 1997.
- [2] P. Pirsch, N. Demassieux, and W. Gehrke, "VLSI architectures for video compression-A survey," *Proc. of IEEE*, vol. 83, no. 2, pp. 220-246, Feb. 1995.
- [3] M. Antonini, et al, "Image coding using wavelet transform," *IEEE Trans. on Image Processing*, vol. 1, no. 2, pp. 205-220, Apr. 1992.
- [4] J. M. Sapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. on Signal Processing*, vol. 41, no. 12, pp. 3445-3462, Dec. 1993.
- [5] ISO/IEC JTC1/SC29/WG11 MPEG97/N1796, "MPEG4 Video Verification Model Version 8.0," Jul. 1997.
- [6] G. Knowles, "VLSI architecture for the discrete wavelet transform," *Electronic Letters*, vol. 26, no. 15, pp. 1184-1185, 19th Jul. 1990.
- [7] K. K. Parhi and T. Nishitani, "VLSI architectures for discrete wavelet transforms," *IEEE Trans. on VLSI Systems*, vol. 1, no. 2, pp. 191-202, June 1993.
- [8] M. Vishwanath, R. M. Owens, and M. J. Irwin, "VLSI architectures for the discrete wavelet transform," *IEEE Trans. on CAS-II: Analog and Digital Signal Processing*, vol. 42, no. 5, pp. 305-316, May. 1995.
- [9] S.-K. Paek and L.-S. Kim, "2D DWT VLSI architecture for wavelet image processing," *Electronics Letters*, vol. 34, no. 6, pp. 537-538, 19th Mar. 1998.
- [10] 장시중 등, "정규직교 이산 웨이블릿을 위한 효율적인 VLSI 구조," *한국통신학회 논문지*, 제23권, 제1호, pp. 239-252, 1998년 1월.

저 자 소 개



김 인 철

1985년 : 서울대학교 제어계측공학과(공학사)
1987년 : 서울대학교 제어계측공학과(공학석사)
1992년 : 서울대학교 제어계측공학과(공학박사)
1992년 8월~1994년 3월 : 대우전자 영상 연구소
1994년 3월~1999년 8월 : 한성대학교 정보전산학부 조교수
1999년 9월~현재 : 서울시립대학교 전자전기공학부 조교수
주관심 분야 : 신호처리, 영상 압축, 실시간 신호처리, ATR 등



정 영 모

1986년 2월 : 서울대학교 제어계측공학과 공학사
1988년 2월 : 서울대학교 대학원 제어계측공학과 석사
1993년 2월 : 서울대학교 대학원 제어계측공학과 박사
1992년 4월~1994년 1월 : 서울대학교 공과대학 유급조교
1994년 1월~1995년 2월 : 서울대학교 제어계측신기술 연구센터 연구원
1998년 8월~1999년 7월 : Georgia Institute of Tech. Post doc.
1995년 3월~현재 : 한성대학교 정보통신공학과 부교수
주관심 분야 : 디지털 이동통신 시스템, 디지털 방송 시스템