

# 데이터 재배열을 이용한 계위성 있는 1차원 이산 웨이블릿 필터 설계

정회원 박태근\*, 준회원 송창주\*

## Scalable One-dimensional Discrete Wavelet Filter Design Using Data Reordering

Tae-geun Park\* *Regular Member*, Chang-joo Song\* *Associate Member*

### 요약

본 논문에서는 데이터 재배열을 이용하여 계위성 있는 1차원 이산 웨이블릿 필터를 설계하였다. 각 레벨마다 입력이 간축되어 데이터의 속도가 반씩 감소하므로, 각 레벨에서 필요한 곱셈기와 덧셈기를 공유함으로써 하드웨어 효율과 복잡도를 개선하였다. 제안된 구조에서는 각 레벨에서 입력의 속도를 고려하여 하드웨어를 구성함으로써 모든 레벨에서 100%의 하드웨어 효율을 나타내었다. 또한, 미리 필터의 특성을 이용하여 고역필터와 저역필터의 계수들을 공유함으로써 하나의 필터만으로 구현하였다. 이 때에 필요한 필터 입력을 공급하기 위해 저장된 데이터를 효율적으로 재배열하는 규칙적인 데이터 재배열 구조를 제안하였다. 제안된 구조는 임의의 필터 길이  $M$ 과 레벨  $J$ 에서도 확장이 용이하며 규칙적인 구조를 가지므로 VLSI로 구현하기에 적당하다. 다른 이산 웨이블릿 필터 구조와 비교하였을 때 상대적으로 적은 하드웨어를 이용하여 효율적인 성능과 하드웨어 효율을 보여주었다.

### ABSTRACT

In this paper, we design a scalable one-dimensional discrete wavelet filter using data reordering. At each level, the required hardware is optimized by sharing multipliers and adders because the input rate is reduced by a factor of two at each level due to decimation. The proposed architecture shows 100% hardware utilization by balancing the hardware with input rate. Furthermore, sharing the coefficients of the high-pass and the low-pass filters using the mirror filter property reduces the number of multipliers and adders in half. We designed a regular data reordering structure that reorders and feeds inputs to high and low pass filters efficiently. The proposed data reordering based architecture is quite regular to implement in VLSI and can be easily extended to an arbitrary 1D DWT structure with  $M$  taps and  $J$  levels. Compared with other architectures, the proposed discrete wavelet filter shows efficiency in performance with relatively less hardware.

### 1. 서론

21세기의 멀티미디어 정보통신분야에 있어서 영상 및 음성 정보는 중요한 역할을 차지하며, 영상 및 음성정보에 대한 표현은 멀티미디어 정보통신 시스템에서 필수적인 부분으로 인식되고 있다. 그러나 현대의 멀티미디어 영상처리 장치들은 일반적으로 실시간 처리를 요구함으로써, 이로 인해 처리해

야 할 영상 데이터의 양은 점차로 증가되어 가고 있는 실정이다.

최근 이산 웨이블릿 변환(DWT, discrete wavelet transform)<sup>[1]</sup>은 영상 압축 및 분석, 프랙털 분석, 패턴 인식 등의 여러 분야에서 그 우수한 성능으로 인해 점차 응용분야를 넓혀가고 있다. DWT는 한 레벨에서 입력신호를 분해하고 다음 레벨에서는 이전 레벨의 출력을 분해하는 과정을 반복한다. DWT

\* 가톨릭대학교 컴퓨터전자공학부 VLSI 설계 연구실 (parktg@catholic.ac.kr)  
논문번호 : 020238-0517, 접수일자 : 2002년 5월 17일

의 장점은 한번에 전체 영상 혹은 각 프레임을 필터 처리하여 블록 효과를 방지하고, 압축률의 보다 정확한 제어를 통해 영상에 대한 비트율이나 화질 조정이 가능하다는 점이다. 또한, DWT는 시간과 주파수에 대해 국부성을 가지고 신호를 표현할 수 있어 비정상 성질을 갖는 신호를 해석함에 유리하다. 즉, 푸리에 방법은 주파수 성분을 알려면 영역 전체의 정보가 필요한 반면, 이런 DWT의 국부성은 단구간 푸리에 변환을 기반으로 하는 방법보다 영상의 국소 성질을 표현하는데 탁월하다. 그래서 영상 재구성 분야에서도 DWT 방법이 사용되고 있으며 이를 이용하여 표현된 영상은 인간 시각 특성과 비슷한 것이 특징이다. 그러나 DWT는 많은 양의 연산이 필요하므로 소프트웨어적으로 실시간 처리하기에는 어려움이 많으며, DWT를 이용한 영상 처리 시스템이 실제 휴대 가능한 영상 통신 기기에서 사용된다면 그 처리속도는 매우 빨라야 하므로 알고리즘의 하드웨어 구현이 필요하다.

최근 DWT의 VLSI 구조에 대한 많은 연구가 진행되어 왔다<sup>[2][3][4][5][7][9]</sup>. 그 중에서 1차원 DWT 처리를 위한 계위성을 갖는 래티스 구조가 제안되었다<sup>[2]</sup>. 래티스 구조가 갖는 효율성으로 하드웨어의 크기가 감소되었으며 이 구조는 래티스의 각 단과 레벨마다 DCU(delay control unit)와 DFC(data format converter)를 삽입하여 효과적으로 스케줄링 하였고 임의의 레벨로 확장할 수 있도록 설계되었으나 2차원으로 확장하기에 어려운 것이 단점이다. Systolic, Semi-systolic, 그리고 RAM-based 등의 세 가지 형태로 제안된 DWT 구조는 우수한 성능과 높은 하드웨어 효율을 보여주지만 복잡한 라우팅 네트워크와 스케줄링이 필요하다<sup>[3]</sup>. 확장이 용이하고 단순한 시스톨릭 구조를 이용한 DWT 구조가 제안되었지만 상위 레벨을 처리하는 블록의 하드웨어 효율이 떨어지는 단점이 있었다<sup>[4]</sup>. 즉, 각 레벨의 하드웨어 효율은 첫 번째 레벨에서 100%이고 두 번째 레벨에서는 50%, 그리고 세 번째 레벨에서는 25%로 감소된다. 최근에 제안된 구조<sup>[5]</sup>에서는 이러한 단점을 보완하였으며 규칙적인 구조의 파이프라인을 적용하였지만 고역필터와 저역필터를 각각 독립적으로 사용함으로써 하드웨어의 크기가 증가한다는 문제점이 있다.

본 논문에서는 미러 필터의 특성<sup>[6]</sup>을 이용하여 고역필터와 저역필터의 계수들을 공유함으로써 하나의 필터만으로 DWT를 구현하였다. 이 때, 고역 및 저역필터의 각 필터 탭에 매 클럭마다 적절한 데이터

를 제공해주기 위하여 임의의 레벨  $J$ 와 필터 탭  $M$ 에서도 적용할 수 있는 계위성 있는 데이터 재배열 구조를 제안하였다. 이는 레지스터들로 구성된 블록으로 입력 다음 단에 위치시킴으로써 데이터 제어 및 하드웨어 복잡도를 감소시킬 수 있다. 일반적으로 DWT 처리 시에 각 레벨에서의 처리 샘플 수는 계속 반으로 줄기 때문에 하드웨어의 효율 또한 감소한다<sup>[4]</sup>. 본 논문에서는 이를 보완하기 위하여 각 레벨에서 들어오는 입력 속도에 따라 하드웨어도 같은 비율로 줄임으로써 100%의 하드웨어 효율을 보여주었다. 이 때 임계 경로를 줄이기 위해 파이프라인 레지스터를 삽입함으로써 성능 개선 효과를 얻을 수 있다.

본 논문의 구성은 다음과 같다. II장에서는 이산 웨이블릿 변환의 이론적 배경과 본 논문에서 사용된 Daubechies 필터의 미러 특성에 대하여 설명한다. III장에서는 본 논문에서 제안된 1차원 DWT와 일반화한 데이터 재배열 구조(data reorder unit, DRU)를 설명하고, IV장에서는 이에 대한 설계와 검증 및 성능 분석에 대하여 설명하며 마지막으로 V장에서는 결론을 맺는다.

## II. 이산 웨이블릿 변환과 Daubechies 필터

### 2.1 이산 웨이블릿 변환

DWT에 사용되는 기저 함수의 집합은 하나의 기본 웨이블릿 기저 함수에 대한 시간 축 방향으로의 확대 및 축소, 그리고 평행 이동을 통해 얻어진다.

$$\Psi^{a,b}(t) = \frac{1}{\sqrt{|a|}} \Psi\left(\frac{t-b}{a}\right) \quad (1)$$

즉, 위의 기저 함수에서  $a$ 는 스케일을 조정하는 성분이고,  $b$ 는 시간 축을 따라 이동하는 성분이다. 다음은 식(1)의 연속 웨이블릿 기저 함수를 이산 웨이블릿 기저 함수로 나타낸 것이다.

$$\Psi^{j,k}(t) = 2^{-j/2} \Psi(2^{-j}t - k) \quad (2)$$

이산 웨이블릿 기저 함수 역시,  $2^j$ 는 스케일을 조정하는 성분이고,  $2^{-j} \cdot k$ 는 시간 축을 따라 이동하는 성분에 대응한다. DWT는 푸리에 변환과 같이 이런 기저 함수들의 집합으로 신호를 분해하여 표현한다. 고전적인 단구간 푸리에 변환은 모든 주파수 대역에 대하여 동일한 크기의 필터 윈도우를 사

용하지만, DWT는 고주파 대역에서는 폭이 좁은 윈도우를, 저주파 대역에서는 폭이 넓은 윈도우를 사용한다는 차이점이 있다. 그러한 이유로 DWT는 뛰어난 시간-주파수 위치성과 비정상적인 신호의 분석에 유리한 특성을 가진다. 기본 웨이블릿 기저 함수는 특별한 형태의 대역 필터로 생각할 수 있으며, DWT의 상대 대역폭 불변성은 기본 웨이블릿 기저 함수에 대한 시간축 방향 축소 및 확대에 의해 충족 되어진다. 이에 따라 DWT에서는 주파수 대역이라는 용어 대신 스케일이라는 용어를 주로 사용하며, 입력 신호에 대한 DWT를 다른 말로 원신호의 시간, 스케일 공간 표현이라 일컫는다. DCT에 커널인 코사인 함수의 원만함파 비교해서 웨이블릿 기저 함수는 시간과 주파수의 지역 편재화가 뛰어나므로, 확대와 축소가 용이하며, 급격한 신호의 변화에도 잘 적응할 수 있는 특성을 보인다. 웨이블릿 기저 함수는 확대 할 때에는 정현파와 여현파의 특징을, 축소 할 때에는 델타 함수의 특징을 접합시켰다.

그림 1은 3단계 웨이블릿 분해와 합성에 대한 흐름도이다.  $H(z)$ 와  $G(z)$ 는 각각 스케일링 함수와 웨이블릿 함수에 일치하는 신장계수로,  $H(z)$ 는 저역필터,  $G(z)$ 는 고역필터를 나타낸다. 분해단에서 각 레벨을 거친 출력은 원신호의 대역폭이 반으로 줄어들기 때문에 샘플링 간격이 두 배 증가하게 되므로 간축(decimation;  $2 \downarrow$ )이 필요하다. 반면에 대역 분할된 신호들로부터 원신호를 재구성할 경우 보간(interpolation;  $2 \uparrow$ )이 필요하다. 다해상도 분석은 그림 1과 같이 저역필터와 고역필터를 번갈아 수행하는데 그 중에서 저역필터의 결과는 다음 단계의 입력으로 들어가 같은 방법으로 분해과정을 반복한다.

일반적인 웨이블릿 필터는 고역필터와 저역필터를 거쳐 DWT가 된다. Daubechies 웨이블릿 필터 계수는 미러 필터의 특성을 가지므로, 본 논문에서는 저역필터와 고역필터 사이의 이런 특성을 이용하여 고역필터와 저역필터를 공유하도록 하였다.

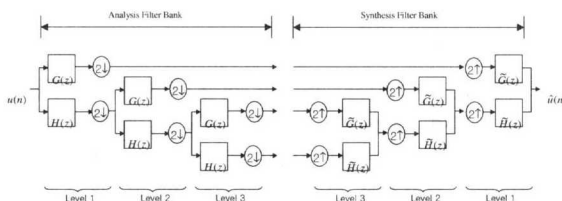


그림 1. 3단계 1차원 웨이블릿 분해와 합성

## 2.2 Daubechies 필터의 미러 특성

본 연구에서 사용된 Daubechies 웨이블릿 필터에서 표1과 같이 저역필터와 고역필터의 계수들간에 다음의 관계식이 성립된다.

표 1. Daubechies 웨이블릿 필터 계수

Daubechies M=6		
m	H(저역필터 계수)	L(고역필터 계수)
0	0.33267055295008	-0.03522629188571
1	0.80689150931109	-0.08544127388203
2	0.45987750211849	0.1350112001025
3	-0.1350112001025	0.45987750211849
4	-0.08544127388203	-0.80689150931109
5	0.03522629188571	0.33267055295008
Daubechies M=4		
0	0.48296291314453	0.12940952255126
1	0.83651630373781	0.2241438680421
2	0.2241438680421	-0.83651630373781
3	-0.12940952255126	0.48296291314453

$$g_{M-1-m} = (-1)^m h_m \quad (M > 2) \tag{3}$$

DWT 필터의 저주파 및 고주파 전달 함수  $H(z)$ 와  $G(z)$ 를 위의 관계식을 이용하여 다시 표현 하면,

$$H(z) = h_0 + h_1 z^{-1} + \dots + h_{M-1} z^{-(M-1)}$$

$$G(z) = (-1)^{M-1} h_{M-1} + (-1)^{M-2} h_{M-2} z^{-1} + \dots + (-1)^0 h_0 z^{-(M-1)} \tag{4}$$

으로 나타낼 수 있는데, 이것을 다시 일반적인 1 차원 Daubechies 4탭 필터에 적용하면 다음과 같이 정리할 수 있다.

$$v_n = a_2 h_0 + a_{2n-1} h_1 + a_{2n-2} h_2 + a_{2n-3} h_3 \quad :low$$

$$u_n = -a_2 h_3 + a_{2n-1} h_2 - a_{2n-2} h_1 + a_{2n-3} h_0 \quad :high \tag{5}$$

$$v_0 = a_0 h_0 \quad [l_0^j]$$

$$u_0 = -a_0 h_3 \quad [h_0^j]$$

$$v_1 = a_2 h_0 + a_1 h_1 + a_0 h_2 \quad [l_1^j]$$

$$u_1 = -a_2 h_3 + a_1 h_2 - a_0 h_1 \quad [h_1^j]$$

$$v_2 = a_3 h_0 + a_3 h_1 + a_2 h_2 + a_1 h_3 \quad [l_2^j]$$

$$u_2 = -a_4 h_3 + a_3 h_2 - a_2 h_1 + a_1 h_0 \quad [h_2^j] \tag{6}$$

## III. 제안된 DWT 아키텍처

그림 2는 본 논문에서 제안된 DWT구조의 최상위 블록 다이어그램으로 각 레벨이 직렬로 구성되어 있다. 그림에서  $N$ 은 입력샘플의 수를,  $L^j$ 와  $H^j$ 는 각각 레벨  $j$ 에서의 저역필터와 고역필터의 출력을 나타낸다. 또한  $P_j$ 는 레벨  $j$ 에서 요구되는 하드웨어를 의미하는데, 예를 들어, 레벨 1에서는 하나의  $M$ -탭 필터가 필요하다. 각 레벨 내부에는 효율적인 데이터 이동과 하드웨어의 복잡도를 줄이기 위해 규칙적이고 계위성이 있는 DRU 블록이 포함되어 있고, 하드웨어의 효율 저하를 개선하기 위하여 입력 샘플의 수가 반씩 감소되는 레벨 2와 레벨 3은 하드웨어도 반씩 감소시켜 하드웨어 효율을 개선하였다.

그림 3은 레벨1의 블록 다이어그램을 나타낸다. 순차적으로 들어온 입력은 DRU를 거쳐 곱셈을 한 후 덧셈기를 통하여 저역 및 고역필터 출력을 생성한다. 입력들은 짝수와 홀수로 나뉘어져 짝수 클럭에 DRU에 저장된다.

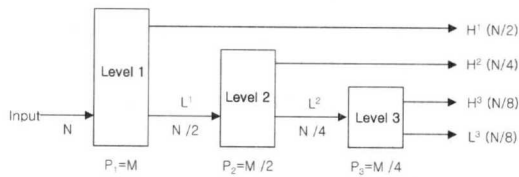


그림 2. 제안된 1 차원 DWT구조 (분석단)

레벨 1 블록에서 홀수 클럭에는 저역필터의 결과를, 짝수 클럭에는 고역필터의 결과를 각각 출력한다. 이를 효과적으로 수행하기 위하여 DRU 블록은 저장된 데이터를 적절한 시간에 재배열하여 하나의 필터 블록에 전달한다. DRU는 다음과 같은 과정으로 설계되었다. 먼저 식 (6)을 보면 입력된 데이터의 흐름이 일정한 규칙을 갖고 있다는 것을 알 수 있다. 입력된 데이터와 미리 특성을 사용한 Daubechies 웨이블릿 필터 계수는 고역필터에서는 서로 순차적으로 곱하여지고 저역필터에서는 서로 역순으로 곱하여지므로 이와 같이 적절하게 데이터를 유지하는 DRU 구조를 설계하였다.

표2는 레벨 1에서 계산되는 저역 및 고역필터 출력을 보여주는 데이터 흐름도이다. 이 표에서 파이프라인으로 인한 지연 시간은 고려하지 않았다. DRU 블록은 규칙적이며 임의의  $J$ -레벨,  $M$ -탭 필터의 경우에도 확장이 용이하다. 레벨 2와 레벨 3에서는 데이터 흐름은 동일하며 단지 데이터의 속도만이 반으로 감소하므로 레벨 1과 유사한 DRU 구조

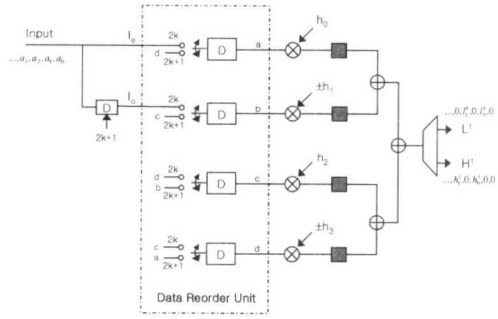


그림 3. 분석단 레벨 1의 블록 다이어그램

표 2. 레벨 1의 데이터 흐름도

clk	0	1	2	3	4	5	6	7	8
$I$	$a_0$	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$	$a_8$
$I_e$	$a_0$	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$	$a_8$
$I_o$	-	-	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$
D R U	a	$a_0$	-	$a_2$	-	$a_4$	$a_1$	$a_6$	$a_3$
	b	-	$a_1$	-	$a_3$	$a_0$	$a_2$	$a_5$	$a_4$
	c	-	-	$a_0$	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$
	d	-	-	$a_0$	$a_2$	$a_1$	$a_4$	$a_3$	$a_6$
$L^1$	0	$l'_0$	0	$l'_1$	0	$l'_2$	0	$l'_3$	0
$H^1$	0	0	$h'_0$	0	$h'_1$	0	$h'_2$	0	$h'_3$

를 갖는다.  $M$ -탭 필터에서 DRU는  $T$ 개의 레지스터들로 이루어져 있으며 DRU 내부의 각 레지스터는 두 값 중 하나를 선택한다. 짝수 클럭에서 처음 두 레지스터는 외부 입력을 저장하며 나머지 레지스터들은 저장된 데이터를 역순으로 재배열한다. 홀수 클럭에서는  $T$ 개의 레지스터에 저장된 데이터를 모두 역순으로 재배열한다. DRU 설계 시에 라이프타임 분석(life-time analysis)<sup>[8]</sup>을 이용하여 레지스터의 수를 최적화하였다. 임계 경로를 곱셈기 하나 정도로 하기 위하여 곱셈기와 덧셈기 사이 회색의 파이프라인 레지스터를 삽입할 수 있다. 표3과 표4는 각각 레벨 2와 레벨 3에 대한 데이터 흐름도이다. 표3의 입력은 그림 4에서와 같이 레벨 1의 저역필터를 통과한 출력  $L^1$ 을 받는다. 같은 데이터 재배열의 방법으로 데이터를 처리하며 단지 각 단의 입력 속도가 이전 단계에 비해 각각 두 배와 네 배로 감소되었으므로, DRU 레지스터의 클럭 주기가 각각 두 배와 네 배로 늘어난다.

그림 4, 그림5와 같이 레벨 2와 레벨 3에서 다운샘플링에 의한 처리 샘플 수가 각각 1/2, 1/4로 감소되기 때문에 그를 위한 하드웨어도 1/2, 1/4로 줄여서 계산이 수행되도록 구성하였으며, 이로 인하여

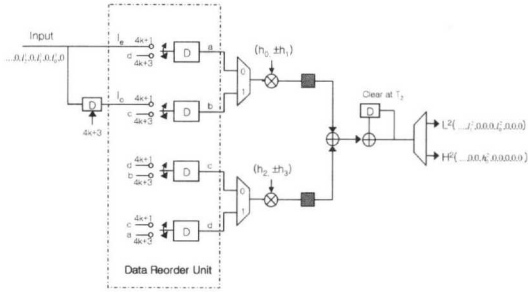


그림 4. 분석단 레벨 2의 블록 다이어그램

표 3. 레벨 2의 데이터 흐름도

clk	0	1	2	3	4	5	6	7	8	9	10
$I$	0	$I'_0$	0	$I'_1$	0	$I'_2$	0	$I'_3$	0	$I'_4$	0
$L_e$	0	$I'_0$	0	$I'_1$	0	$I'_2$	0	$I'_3$	0	$I'_4$	0
$L_o$	-	-	-	$I'_1$	$I'_1$	$I'_1$	$I'_1$	$I'_1$	$I'_1$	$I'_1$	$I'_1$
D R U	a	-	-	$I'_0$	$I'_0$	-	-	$I'_2$	$I'_2$	-	-
	b	-	-	-	-	-	-	$I'_1$	$I'_1$	$I'_0$	$I'_0$
	c	-	-	-	-	-	-	$I'_0$	$I'_0$	$I'_1$	$I'_1$
	d	-	-	-	-	-	-	$I'_2$	$I'_2$	$I'_1$	$I'_1$
$L^2$	0	0	0	$I'_0$	0	0	0	$I'_1$	0	0	0
$H^2$	0	0	0	0	0	0	0	0	0	$I'_1$	0

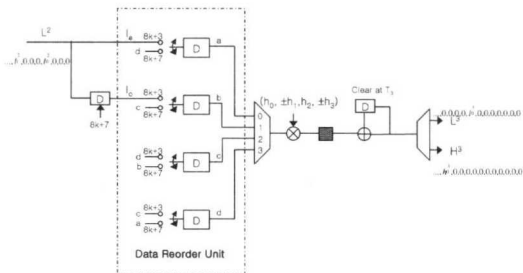


그림 5. 분석단 레벨 3의 블록 다이어그램

표 4. 레벨 3의 데이터 흐름도

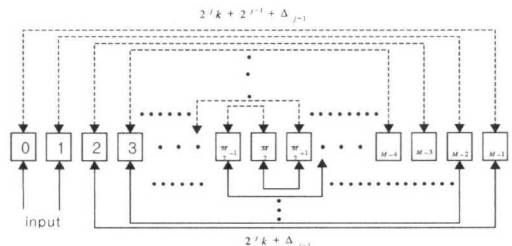
clk	0	1	2	3	4	5	6	7	8	9	10
$I$	0	0	0	$I'_0$	0	0	0	$I'_1$	0	0	0
$L_e$	0	0	0	$I'_0$	0	0	0	$I'_1$	0	0	0
$L_o$	-	-	-	-	-	-	-	$I'_1$	$I'_1$	$I'_1$	-
D R U	a	-	-	-	$I'_0$	$I'_0$	$I'_0$	$I'_0$	-	-	-
	b	-	-	-	-	-	-	-	-	-	-
	c	-	-	-	-	-	-	-	-	-	-
	d	-	-	-	-	-	-	-	$I'_0$	$I'_0$	$I'_0$
$L^3$	0	0	0	0	0	0	0	$I'_0$	0	0	0
$H^3$	0	0	0	0	0	0	0	0	0	0	0

제안된 구조에서의 하드웨어 효율은 100%를 보여 준다. 레벨 2와 3에서 DRU 클럭의 오프셋(offset)은

이전 레벨에서의 지연시간이 고려되었다. 만일 파이프 라인 레지스터를 무시한다면 레벨  $j$ 에서의 지연시간  $\Delta_j = \Delta_{j-1} + 2^{j-1}$  ( $j \geq 1, \Delta_0 = 0$ )로 정의된다. 이 경우, 각 클럭마다 생성되는 부분 합을 올바르게 계산하기 위하여 레벨  $j$  ( $j \geq 2$ )의 출력 단에 있는 레지스터는,  $T_j = \Delta_j + 2^{j-1}k$  ( $k=0,1,2,3,\dots$ )마다 리셋 되어야 한다.

$J$ -레벨,  $M$ -탭을 갖는 1차원 DWT에서의 일반화한 DRU 구조는 짝수와 홀수의 탭에 따라 다음 그림 6의 (a)와 (b) 같이 나타낼 수 있다. 레벨에 따른 레지스터간 데이터 흐름은  $2^{j-1}k + \Delta_{j-1}$  시간에 두 레지스터의 입력과 나머지 짝수 탭은  $M/2$ 과  $(M/2)+1$ 을 중심으로, 홀수 탭에서는  $[M/2]+1$ 을 중심으로 대칭적으로 그림 6의 실선과 같이 데이터 교환이 이루어진다. 또한,  $2^{j-1}k + 2^{j-1} + \Delta_{j-1}$  시간에는 짝수 탭에서는  $(M/2)-1$ 과  $M/2$ 을 중심으로, 홀수 탭에서는  $[M/2]$ 를 중심으로 그림에서 점선으로 나타낸 것과 같이 대칭적으로 데이터의 교환이 일어난다. 이런 DRU의 특성은 그림 6에서와 같이 필터의 레벨과 탭에 대하여 규칙적이고 확장이 용이하다는 것이다.

제안된 구조는 합성단에서도 동일하게 적용될 수 있다. 레벨 3의 출력인  $L^3$ 와  $H^3$ 가  $\dots 0h_1^3 000l_1^3 000h_0^3 000l_0^3$ 와 같이 입력되면 그림 5와 같은 구조에서  $8k$  시간마다 DRU에 입력되어 재배열된다. DRU에 저장된 데이터는 네 사이클마다 한번씩  $L^2$  값을  $\dots l_2^2 000l_1^2 000l_0^2 0000$ 와 같이 출력하며 출력단의 디멀티플렉서는 필요 없다. 이 출력 값들은 레벨 2에 입력되고  $H^2$ 와 결합하여  $\dots h_1^2 0l_1^2 0h_0^2 0l_0^2 0000$ 의 시퀀스를 구성하며, 그림 4와 같은 구조에서  $4k$  시간마다 DRU에서 재배열되어 두 사이클에 한번씩  $L^1$  값을  $\dots 0l_1^1 0l_0^1 0l_1^1 0l_0^1 00000000$ 와 같이 출력한다. 최종적으로, 레벨 1에서도  $\dots h_1^1 l_1^1 h_0^1 l_0^1 h_1^1 l_1^1 h_0^1 l_0^1 00000000$ 의 시퀀스를 구성하며 그림 3과 같은 구조에서  $2k$  시간마다 DRU에서 재배열되어 최종 출력 값을  $\dots a_4 a_3 a_2 a_1 a_0 00000000$ 와 같이 출력한다. 따라서 제안된 구조는 동일하게 분석 및 합성에서 이용될 수 있다.



(a)  $M$ 이 짝수일 때

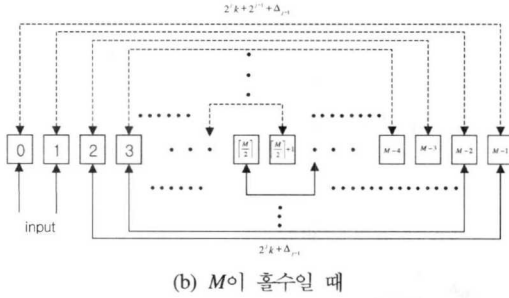


그림 6. M-탭, J-레벨에서 일반화한 DRU 구조

#### IV. 실험 및 결과분석

제안된 1차원 DWT구조는 VHDL로 모델링 되었으며 Modelsim 환경에서 시뮬레이션 되어 그의 동작이 검증되었다. 그림 7은 앞에서 제안된 1차원 DWT 구조의 타이밍 다이어그램이다.  $(H^1, L^1)$ ,  $(H^2, L^2)$ , 그리고  $(H^3, L^3)$ 들은 각각 레벨 1과 레벨 2, 그리고 레벨 3의 출력을 나타낸다. 각 레벨에서 고역필터 출력과 저역필터 출력은 서로 반 주기동안 중첩되어 엇갈려 출력되며  $(H^1, L^1)$ 은 2 클럭 사이클,  $(H^2, L^2)$ 는 4 클럭 사이클, 그리고  $(H^3, L^3)$ 는 8 클럭 사이클 주기로 출력이 나타남을 볼 수 있다. 설계 시에 데이터의 비트 할당은 각각 그림 8과 같이 하였다. 4탭의 Daubechies 계수 값들이 모두 정수 1보다 작으므로, 처리되는 데이터 16비트에서 하위 5비트는 소수점 이하 부분으로 할당하며, 나머지 상위 11비트는 부호와 정수부분으로 할당하였다.

표 5는 일반적으로 3레벨까지의 4탭 필터에 대해서 하드웨어 및 성능을 비교한 것이다. 시스톨릭 구조를 이용한 방법<sup>[3]</sup>은 DWT를 연산하기 위하여 2N의 주기가 필요하며, 데이터를 필터에 제공하기 위하여 복잡한 라우팅 네트워크가 필요하다. 또 다른 시스톨릭 구조<sup>[4]</sup>에서는 4 탭일 때의 계수 특성을 이용하여 곱셈기의 수를 6개까지 줄일 수 있으나 일반적인 경우 12개가 필요하며 레벨이 증가됨에 따라 하드웨어의 효율이 떨어지는 단점이 있다. 병렬 구조<sup>[9]</sup>를 이용한 방법에서는 비슷한 하드웨어를 필요로 하지만 상위 레벨을 연산하기 위하여  $J \cdot M$ 크기의 메모리 블록이 부가적으로 필요하다. 부하가 적절히 배분된 파이프라인 구조<sup>[5]</sup>에서는 고속 처리가 가능하지만, 저역필터와 고역필터를 동시에 처리하기 위하여 두 배의 하드웨어가 필요하다. 중첩된 구조를 적용한 방법<sup>[7]</sup>에서는 제어가 복잡하며 최적의 하드웨어 효율을 보이지 않는다. 제안된 구조는

하드웨어의 복잡도를 나타내는 곱셈기의 수는 7개로 구현 할 수 있고, DRU의 확장 가능한 구조로 인해 다른 구조보다 좀더 일반화되고 하드웨어 효율을 높은 결과를 얻을 수 있다. 제안된 구조를 이용한 J-레벨, M-탭의 1 차원 DWT 구조에서의 곱셈기의 총수는  $\sum_{k=1}^J [M / 2^{k-1}]$ 와 같이 나타낼 수 있다.

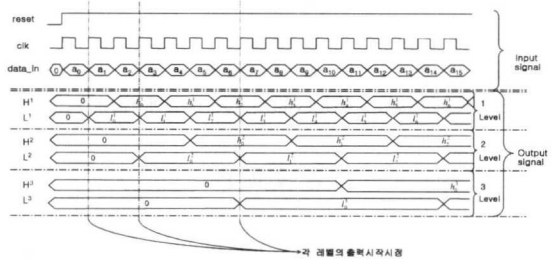


그림 7. 제안된 1차원 DWT의 타이밍 다이어그램

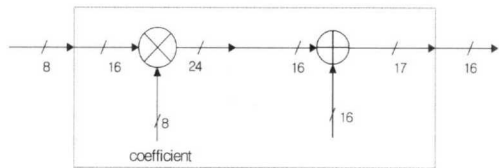


그림 8. 제안된 구조에서의 비트할당 방법

표 5. 여러 가지 기존 구조와 1 차원 DWT 성능 비교

Architecture	Multiplier	Adder	Period	H/W util.	Scheduling
Systolic [3]	8	8	2N	100%	complex
Systolic [4]	12	9	N	58.3%	simple
Parallel [9]	8	6	N	100%	complex
Pipelined [5]	14	14	N/2	100%	simple
Folded [7]	8	6	N	87.5%	complex
Proposed	7	6	N	100%	simple

#### V. 결론

본 논문에서는 데이터 재배열을 이용한 일반화된 1차원 DWT 필터의 구조를 제안하였다. 각 레벨에는 효율적인 데이터 이동과 하드웨어의 복잡도를 줄이기 위해 규칙적이고 계위성 있는 데이터 재배열 구조가 제안되었다. 먼저 입력을 데이터 재배열 구조에 저장하여 적절한 시간에 곱셈기에 입력할 수 있도록 설계하였고 최소의 레지스터로 임의의 레벨과 탭에서도 적용할 수 있는 계위성 있고 규칙적인 구조를 제안하였다. 또한 각 레벨마다 입력이 다른 샘플링 되기 때문에 그에 해당하는 각 레벨의

하드웨어도 이전 레벨의 반으로 구현함으로써 시스템의 복잡도가 개선되었다. 그리고 저역필터와 고역필터 계수의 미리 특성을 이용하여 하나의 필터를 공유함으로 하드웨어를 반으로 감소시켰다. 따라서 제안된 구조의 각 레벨은 입력 속도를 고려해서 설계하여 100%의 하드웨어 효율을 나타내었다.

### 감사의 글

저자들은 본 연구를 위하여 설계 환경을 제공하여준 IDEC(IC Design Education Center)에 감사드립니다

### 참 고 문 헌

[1] S. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. And Machine Intell.*, vol.11, no.7, pp.674-693, 1989.

[2] J. T. Kim, Y. H. Lee, T. Ishiki, and H. Kunieda, "Scalable VLSI architectures for lattice structure-based discrete wavelet transform," *IEEE Trans. Circuits Syst. II*, vol.45, no.8, pp.1031-1043, 1998.

[3] M. Vishwanath, R. M. Owens, and M. J. Irwin, "VLSI architectures for the discrete wavelet transform," *IEEE Trans. Circuits Syst. II*, vol.42, no.5, pp.305-316, 1995

[4] 반성범, 박래홍, "이산 웨이블릿 변환을 위한 효율적인 VLSI구조," *전자공학회논문집*, 제36권, S편, 제6호, pp.96-103, 1999.

[5] F. Marino, D. Guevorkian, and J. T. Astola "Highly efficient high-speed/low-power architectures for the 1-D discrete wavelet transform," *IEEE Trans. Circuits Syst. II*, vol.47, no.12, pp.1492-1502, 2000.

[6] K. R. Castleman, "Digital Image Processing", *Prentice Hall*, pp.329-330, 1996.

[7] K.K. Parhi and T. Nishitani, "VLSI architectures for discrete wavelet transform," *IEEE Trans. VLSI Systems*, vol.1, no.2, pp.191-202, 1993.

[8] K.K. Parhi, "VLSI digital signal processing systems: design and implementation," *Wiley*, 2000.

[9] C. Chakrabarti and M. Vishwanath, "Efficient realization of the discrete and continuous

wavelet transforms: from single chip implementations to mappings on SIMD array computers," *IEEE Trans. Signal Proc.*, vol. 43, no.3, pp.759-771, 1995.

박 태 근(Tae-geun Park)

정회원



1985년 : 연세대학교  
전자공학사.  
1988년 : Syracuse Univ.  
Computer공학석사.  
1993년 : Syracuse Univ.  
Computer공학박사.

1991년~1993년 : Coherent Research Inc. VLSI 설계 엔지니어.  
1994년~1998년 : 현대전자 System IC 연구소 책임 연구원.  
1998년~현재 : 가톨릭 대학교 컴퓨터 전자 공학부 부 교수.  
<주관심 분야> VLSI, CAD, 병렬처리

송 창 주(Chang-joo Song)

준회원



2001년 : 가톨릭대학교  
컴퓨터 통신학과 학사  
2001년~현재 : 가톨릭 대학교  
컴퓨터 전자공학부  
석사 과정  
<주관심 분야> VLSI, 디지털  
신호처리, 디지털통신