

논문 2004-41SD-11-13

# JPEG2000영상압축을 위한 라인 기반의 리프팅 DWT 구조 설계

(Architecture Design of Line based Lifting-DWT  
for JPEG2000 Image Compression)

정 갑 천\*, 박 성 모\*\*

(Gab Cheon Jung and Seong Mo Park)

## 요 약

본 논문은 JPEG2000의 손실 압축 또는 무손실 압축에 사용되어지는 9-7/5-3 리프팅 DWT필터에 대한 효율적인 VLSI 구조를 제안한다. 제안된 구조는 리프팅 DWT 연산을 위해 내부 라인 메모리만을 사용하며, 내부 처리 유닛은 1개의 곱셈기와 1개의 덧셈기의 임계경로를 갖는다. 특히 본 논문에서는 처리유닛의 수를 감소하기 위해 1레벨의 열방향을 담당하는 필터로 하여금 2레벨 이상의 행방향과 열방향 연산 모두를 처리하도록 하였다. 결과적으로 제안된 구조는 기존의 구조에 비해 작은 하드웨어 크기를 갖는다. 제안된 리프팅 DWT구조는 RTL 수준에서 VHDL로 모델링되었으며, 기능 검증 후 Altera APEX 20K FPGA로 구현되었다.

## Abstract

This paper proposes an efficient VLSI architecture of 9-7/5-3 Lifting DWT filters that is used by lossy or lossless compression of JPEG2000. The proposed architecture uses only internal line memories to compute Lifting-DWT operations and its PE(Processing Element) has critical path with 1 multiplier and 1 adder. To reduce the number of PE, we make the vertical filter that is responsible for the column operations of the first level perform both the row and column operations of the second and following levels. As a result, the architecture has smaller hardware cost compared to that of other architectures. It was modeled in RTL level using VHDL and implemented on Altera APEX 20K FPGA.

**Keywords :** Lifting DWT, VLSI architecture, hardware cost, line memory, JPEG2000

## I. 서 론

멀티미디어의 발달에 따라 영상, 음성 등의 데이터 압축/신장 등은 이전보다 새롭고 보다 나은 성능을 필요로 한다. 이러한 요구에 따라 정치 영상 분야에서는

기존 JPEG(Joint Photographic Expert Group)의 손실/무손실 압축의 비단일화, 저비율에서의 성능 열화 등의 단점을 극복하기 위해 JPEG2000 표준이 개발되었다<sup>[1-2]</sup>. DWT(Discrete Wavelet Transform)를 기반으로 압축을 수행하는 JPEG2000은 낮은 비트율에서의 향상된 압축, 단일 시스템에서의 손실 및 무손실 압축 가능, 비트에러에 대한 견고함, 점진적 전송 등 이전 JPEG에 비해 많은 장점들을 제공한다. 이와 같이 JPEG2000은 이전의 DCT(Discrete Cosine Transform)를 기반으로 압축을 수행하는 JPEG에 비해 보다 향상된 기능 및 효율성을 제공하는 반면 실행 시간은 증가되었다.

DWT는 공간과 주파수 특성들을 효율적으로 수집할 수 있는 장점으로 인해 최근 영상 압축에 많이 이용되고 있으나 DWT는 필터뱅크들을 사용하여 구현되기 때

\* 정희원, 전남대학교 전자공학과  
(Dept. of Electronics Eng., Chonnam National Univ.)

\*\* 정희원, 전남대학교 컴퓨터공학과  
(Dept. of Computer Eng., Chonnam National Univ.)

※ 본 논문은 한국과학재단 지정 전남대학교 고품질 전기전자부품 및 시스템 연구센터의 연구비 지원에 의해 연구되었음.

※ This paper was partially supported by IDEC(IC Design Education Center)

접수일자: 2004년6월16일, 수정완료일: 2004년10월23일

문에 방대한 연산들을 요구한다. 이와 같은 연산량을 감소하기 위해 리프팅 DWT방법이 제안되었다<sup>[3]</sup>. 리프팅 연산은 빠른 계산, in-place-calculation, 정수 대 정수, 쉬운 역 구현 등 많은 장점들을 포함한다. 이러한 장점들로 인해 JPEG2000은 DWT 연산을 위해 컨볼루션 방법 이외에 리프팅 필터링 모드를 지원하며, 비가역 변환은 biothogonal (9-7) 필터를 이용하고 가역 변환을 위해서는 (5-3) 필터를 이용한다.

최근 몇 년간 Biothogonal (9-7) 또는 (5-3) 필터에 대한 2차원 리프팅 DWT를 위한 몇몇의 VLSI 구조들<sup>[4-6]</sup>이 제안되었는데, 그 중 [4], [5] 의 구조는 정규적인 데이터 플로우를 가지며 낮은 제어 복잡성을 가진다. 그러나 이 구조들은 완전한 영상에 대해 한 레벨 연산이 모두 끝난 후 다음 레벨 연산들을 수행하기 때문에  $N \times N$  영상에 대해  $N^2/4$  크기의 외부 메모리 모듈을 필요로 한다. [6]의 구조는 RPA(Recursive Pyramid Algorithm)<sup>[7]</sup> 알고리즘을 이용하여 외부 메모리 없이 단지 라인 메모리들만을 사용하지만 낮은 하드웨어 유용성으로 인해 [4], [5]의 구조에 비해 2배의 처리 유닛의 수를 가진다.

본 논문에서는 RPA 알고리즘을 사용하면서 보다 적은 처리 유닛의 수를 가지는 라인 기반의 리프팅 DWT 구조를 제안하였다. II장에서 JPEG2000에서 사용되는 리프팅 DWT 알고리즘에 대하여 기술하고, III장에서는 리프팅 DWT에 대한 VLSI 구조를 제안하였다. 그리고 IV장에서는 제안된 VLSI 구조에 대한 VHDL 모델링과 기존의 구조들과의 성능 비교를 기술하며, V장에서 결론을 맺었다.

## II. 리프팅 DWT 알고리즘

Mallet<sup>[8]</sup>은 1차원 및 2차원 웨이브렛 변환 알고리즘을 개발했는데, 웨이브렛 변환은 원신호를 고주파 대역과 저주파 대역으로 분할하고 저주파 대역을 다시 고주파와 저주파 성분으로 분할하는 과정을 반복하는 방법으로 부호화한다. 그림1은 2차원 DWT의 분할 알고리즘을 나타낸다.

첫번째 분할 레벨에서 입력 영상은 행방향으로의 필터링에 의해 2개의 부밴드 L(저주파), H(고주파)로 분할 되어지며, 이 2개의 부밴드들은 다시 열방향으로의 필터링에 의해 4개의 부밴드 LL(저주파-저주파), LH(저주파-고주파), HL(고주파-저주파), HH(고주파-고주파)로 분할되어진다. 다중 레벨 분할은 입력 영상 대신 저주파-저주파 밴드를 가지고 수행된다. 결과로서 입력

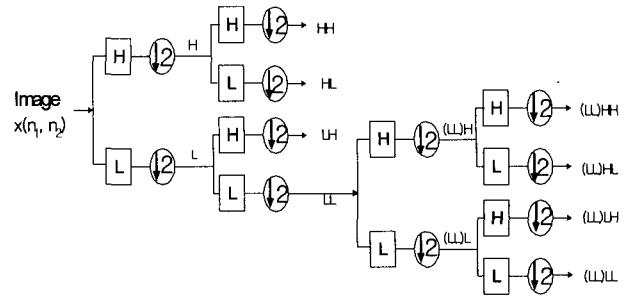


그림 1. 2차원 DWT의 분할 알고리즘

Fig. 1. Decomposition algorithm of 2-D DWT.

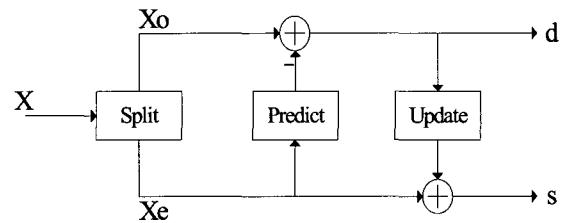


그림 2. 리프팅 스텝

Fig. 2. Lifting step.

영상은 J 레벨에서  $3J + 1$  부밴드로 분할되어지고, 각 밴드의 사이즈는  $N \times N$  입력 영상에 대해  $(N \times N)/4J$ 크기로 감소한다.

리프팅 이론은 이와 같은 필터 뱅크들을 효율적으로 계산하기 위해서 I. Daubechies and W. Sweldens에 의해 제안되었으며, DWT 동작들을 간단한 필터링 스텝들로 분할한다<sup>[3]</sup>. 리프팅 스텝들은 그림 2와 같은 분할(splitting), 예측(prediction), 캐신(update) 세 가지의 스텝들로 수행된다. 그림 2에서 d는 고주파 계수, s는 저주파 계수를 나타낸다.

분할 스텝에서는 입력신호들이 짹수 샘플과 홀수 샘플들로 분할되어진다. 예측 스텝에서는 짹수 샘플들이 홀수 샘플들을 예측하기위해 사용되며, 홀수 샘플들과 예측값들 사이의 차를 계산하여 고주파 계수를 계산한다. 캐신 스텝에서는 고주파 계수들을 이용하여 저주파 계수들을 계산한다. JPEG2000의 biothogonal (9,7)/(5,3)에 대한 리프팅 DWT 계산은 식(1)과 같이 나타낼 수 있다.

$$\begin{aligned}
 d^0(j) &= x(2j+1), & s^0(j) &= x(2j) \\
 d(j) &= d^0(j) + \alpha(s^0(j) + s^0(j+1)), & s(j) &= s^0(j) + \beta(d(j-1) + d(j)) \\
 d^1(j) &= d(j) + \gamma(s(j) + s(j+1)), & s^1(j) &= s(j) + \delta(d^1(j-1) + d^1(j)) \\
 H(j) &= d(j) = d^1(j) & I(j) &= s(j) = s^1(j) \quad \text{for (5,3) filter} \\
 &= (1/\sqrt{2}) d^1(j) & & = \xi s^1(j) \quad \text{for (9,7) filter}
 \end{aligned} \tag{1}$$

$$\begin{aligned} \alpha &= -0.5 & \beta &= 0.25 & \text{for (5,3) filter} \\ &= -1.586134342 & &= -0.05298.11854 & \text{for (9,7) filter} \\ \gamma &= 0.8829110762, & \delta &= 0.4435068522, & \xi &= 1.149604398 \end{aligned}$$

(5,3) 필터의 경우 한번의 예측스텝과 갱신 스텝으로 저주파와 고주파 결과를 계산할 수 있는데 반해, (9,7) 필터의 경우 예측 스텝과 갱신 스텝을 한번 더 수행함으로써 고주파 결과와 저주파 결과를 계산한다. 식(1)에서  $\xi$ 는 스케일링 인자(Scaling factor)를 나타내며 (5,3) 필터의 경우 1이다. 식(1)의 각 리프팅 스텝 계산은 상태변수  $q(i)$ 를 이용하여 식(2)와 같이 분해 될 수 있다.

$$\begin{aligned} d^0(i) &= x(2i+1), & s^0(i) &= x(2i) \\ d(i) &= \alpha s^0(i+1) + q(i-1), & s^1(i) &= \beta d^0(i) + q(i-1) \\ q(i) &= \alpha s^0(i+1) + d^0(i+1), & q(i) &= \beta d^0(i) + s^0(i+1) \\ d^1(i) &= \gamma s^1(i+1) + q(i-1), & s^2(i) &= \delta d^1(i) + q(i-1) \\ q(i) &= \gamma s^1(i+1) + d^1(i+1), & q(i) &= \delta d^1(i) + s^1(i+1) \\ H(i) &= d(i) = d^0(i) & I(i) &= s(i) = s^0(i) \quad \text{for (5,3) filter} \\ &= (1/\xi) d^2(i) & &= \xi s^2(i) \quad \text{for (9,7) filter} \end{aligned} \tag{2}$$

### III. 제안된 구조

본 장에서는 JPEG2000의 biothogonal 리프팅 DWT의 실시간 처리를 위한 VLSI 구조를 기술한다. 제안된 구조는 그림3과 같이 수평 프로세서(HOR)와 수직 프로세서(VER)로 구성된다.

#### 1. 수평 프로세서

수평 프로세서는 첫번째 레벨의 행방향으로의 필터링 연산을 수행하며 스플릿터(Splitter)와 수평 필터로 구성된다. 수평 프로세서는 식(2)에서와 같이 영상의 두 열을 가지고서 필터링 연산을 수행하고, 래스트(Raster) 스캔의 경우 영상은 순차적으로 들어오기 때문에 영상의 순차적 데이터를 2개의 병렬 데이터로 변환해주는 변환기가 필요하다. 수평 프로세서내 스플릿터(Splitter)는  $f_i$  비율로 영상 입력들을 직렬로 받아 2개의 병렬 데이터를  $f_i/2$  으로 수평 필터로 보내주는 역할을 한다.

스플릿터로부터의 2개의 병렬 데이터는 프로세서 주파수의 2사이클마다 갱신 되어지고, 수평 필터는 스플릿터로부터 영상의 두 열을 가지고서 식(2) 연산을 수행한다. DWT 연산은 고주파 결과와 저주파 결과 모두 요구되기 때문에 수평 필터는 매 사이클마다 저주파 결과 또는 고주파 결과를 계산한다. 수평 필터는 그림 5

와 같은 파이프라인 구조를 가진다.

영상의 경계처리를 위해 본 논문에서는 주기적인 대칭성 확장(PSE: Periodic Symmetric Extension) 방법을 사용한다. 리프팅 연산의 대칭성으로 인해 PSE 방법은 MUX 들만을 사용하여 쉽게 구현 가능하다<sup>[5][9]</sup>. 그럼 6은 biothogonal(9,7) 필터에 대해 영상 크기가 짹수인 8

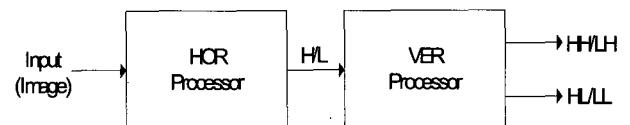


그림 3. 제안된 DWT 구조

Fig. 3. Block diagram of proposed DWT architecture.

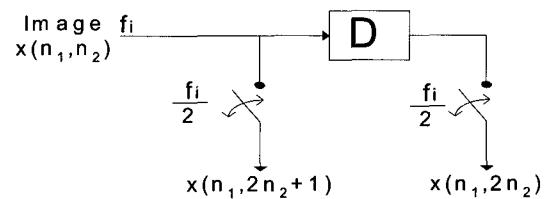


그림 4. 스플릿터의 블록도

Fig. 4. Block diagram of Splitter.

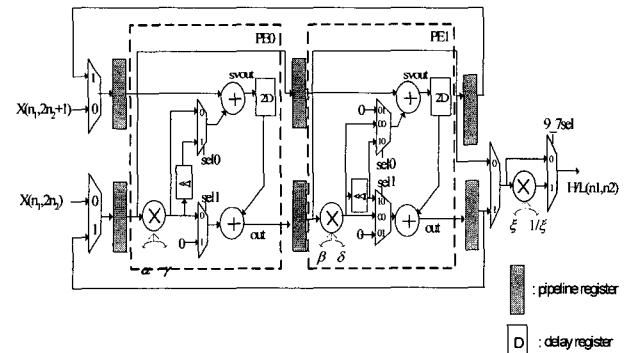


그림 5. 수평 필터의 파이프라인 구조

Fig. 5. Pipelined structure of horizontal filter.

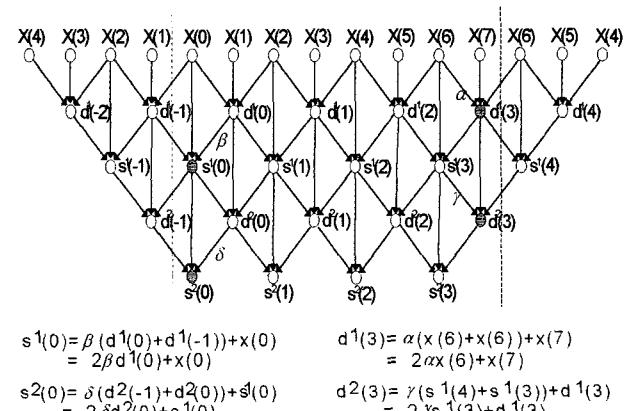


그림 6. 경계영역에서의 리프팅 연산

Fig. 6. Lifting operation at boundary.

표 1. 수평 필터의 연산 순서

Table 1. Computational sequence of the horizontal filter.

Cyc.	PE0 Sel0	PE0 Sel1	PE0 svOut	PE0 Out	PE1 Sel0	PE1 Sel1	PE1 svOut	PE1 Out	Out (5-3)	Out (9-7)
	0	0		d <sup>1</sup> (0,1)	00	00		-	-	L(0,0)
i+2	0	0	vs <sup>1</sup> (0,0)+d <sup>1</sup> (0,0)	-	00	00	$\beta d^1(0,1)+x(0,4)$	s <sup>1</sup> (0,1)	H(0,1)	-
i+3	1	0	2ax(0,6)+x(0,7)	d <sup>1</sup> (0,2)	01	00	s <sup>1</sup> (0,0)	-	L(0,1)	-
i+4	0	0	vs <sup>1</sup> (0,1)+d <sup>1</sup> (0,1)	d <sup>2</sup> (0,0)	00	00	$\beta d^1(0,2)+x(0,6)$	s <sup>1</sup> (0,2)	H(0,2)	-
i+5	0	1	ax(1,0)+x(1,1)	d <sup>1</sup> (0,3)	00	10	$\delta d^2(0,0)+s^1(0,1)$	s <sup>2</sup> (0,0)	L(0,2)	H(0,0)
i+6	0	0	vs <sup>1</sup> (0,2)+d <sup>1</sup> (0,2)	d <sup>2</sup> (0,1)	01	00	x(1,0)	s <sup>1</sup> (0,3)	H(0,3)	L(0,0)
i+7	0	0	ax(1,2)+x(1,3)	d <sup>1</sup> (1,0)	00	00	$\delta d^2(0,1)+s^1(0,2)$	s <sup>2</sup> (0,1)	L(0,3)	H(0,1)
i+8	1	0	2vs <sup>1</sup> (0,3)+d <sup>1</sup> (0,3)	d <sup>2</sup> (0,2)	00	10	$\beta d^1(1,0)+x(1,2)$	s <sup>1</sup> (1,0)	H(1,0)	L(0,1)
i+9	0	0	ax(1,4)+x(1,5)	d <sup>1</sup> (1,1)	00	00	$\delta d^2(0,2)+s^1(0,3)$	s <sup>2</sup> (0,2)	L(1,0)	H(0,2)
i+10	0	1	vs <sup>1</sup> (1,0)+d <sup>1</sup> (1,0)	d <sup>2</sup> (0,3)	00	00	$\beta d^1(1,1)+x(1,4)$	s <sup>1</sup> (1,1)	H(1,1)	L(0,2)
i+11	1	0	2ax(1,6)+x(1,7)	d <sup>1</sup> (1,2)	01	00	s <sup>1</sup> (1,0)	s <sup>2</sup> (0,3)	L(1,2)	H(0,3)

인 경우에 경계영역에서의 리프팅 연산을 나타낸다.

표1은 영상 행의 크기 8에 대한 수평 필터의 연산 순서를 나타내며, 표에서 Sel0, Sel1은 경계 처리를 위한 각 처리 유닛(PE: Processing Element)내 MUX들의 제어신호를 나타낸다.

수평 필터는 고주파 또는 저주파 결과값들을 수직 프로세서로 내어보내고, 계산되어진 상태 변수들은 다음 계산을 위해 레지스터에 저장되어진다. 계산되어진 상태변수는 2사이클 후에 사용되어지기 때문에 각 PE내 각 상태변수는 2개의 레지스터에 저장된다. 수평 필터는 3개의 곱셈기, 4개의 덧셈기, 상태 변수들을 저장하기 위한 4개의 레지스터들을 포함하며, 파이프라인을 위한 6개의 레지스터들을 포함한다.

## 2. 수직 프로세서

행방향 연산은 2열의 데이터를 사용하여 동작하는 반면, 열방향으로의 연산은 2행의 저주파와 고주파 결과들을 이용하여 연산을 수행한다. 즉 첫번째 레벨의 연산의 경우 영상의 짹수번째 행에 대한 저주파와 고주파 결과만으로는 열방향으로의 연산을 수행할 수 없으며, 홀수번째 행의 저주파와 고주파 결과가 유용해야 비로소 열방향으로의 연산을 수행할 수 있다. 이와 같이 첫번째 레벨의 연산이 수행될 수 없는 짹수 번째 행의 주기는 2번째 레벨 이상의 연산을 수행하는데 사용될 수 있다<sup>[7]</sup>. 수직 프로세서는 수평 프로세서와는 달리 매사이클마다 저주파 출력과 고주파 출력을 동시에 계산하며 다음과 같은 RPA 스케줄링을 이용한다.

1) 수평 모듈이 영상의 홀수 번째 행에 대한 L, H 출력

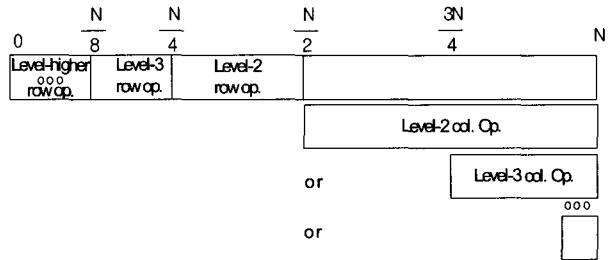


그림 7. 수직 프로세서의 짹수행에서의 연산 타이밍도  
Fig. 7. Timing diagram of vertical processor for even rows.

결과를 내보낼 때, 수직 모듈은 첫번째 레벨의 열방향을 수행한다.

2) 수평 모듈이 영상의 짹수번째 행에 대한 L, H 출력 결과를 내보낼 때, 수직 모듈은 2번째 레벨 이상의 행방향과 열방향 연산을 수행한다.

2a) 수평 모듈이 영상의  $2^{(J-1)} k + (2^{(J-1)} - 2)$  번째 행에 대한 출력 결과를 내보낼 때, 수직 모듈은 J번째 레벨 행방향 연산을 수행한다. ( $J > 1$ )

2b) 수평 모듈이 영상의  $2^J k + (2^{(J-1)} - 2)$  번째 행에 대한 출력 결과를 내보낼 때, 수직 모듈은 J번째 레벨 열방향 연산을 수행한다. ( $J > 1$ )

그림 7은 수평 프로세서가 영상의 짹수번째 행에 대한 출력을 계산할 때 수직 프로세서의 2번째 레벨 이상의 연산 타이밍도를 나타낸다. 수직 프로세서는 라인 메모리 버퍼와 수직 필터로 구성되어진다.

라인 메모리 버퍼는 중간적인 신호 결과들의 저장 및 이 결과들을 수직 필터로 내보내는데 사용되어 진다. 라인 메모리 버퍼는 그림 8과 같이 (LL)L, (LL)H 행 버퍼들과 LL 버퍼로 구성되어진다.

(L/H)e 버퍼는 영상의 짹수번째 행에 대한 수평 프로세서의 고주파와 저주파 결과들을 저장하며, 하나의 레지스터는 영상의 홀수번째 행에 대한 현재 수평 프로세서의 고주파 또는 저주파 결과를 저장한다. (LL)Le, (LL)He 행 버퍼, (LL)Lo, (LL)Ho 행 버퍼는 2번째 레벨의 열방향 연산을 위해 사용되어진다. LL 버퍼는 2번째 레벨 이상의 행방향 연산을 위해 사용되어지며, 저주파-저주파 한 행의 짹수번째 열들과 홀수번째 열들을 분리적으로 저장한다.

만약 수평 프로세서가 홀수번째 행에 대한 고주파, 저주파 결과들을 출력한다면, 수직 필터는 이전 짹수번째 행의 해당 열의 고주파 또는 저주파 결과를 (L/H)e 행 버퍼로부터 읽어 들여 고주파-저주파, 고주파-고주파 또는 저주파-저주파, 저주파-고주파 결과들을 출력한다. 계산되어진 저주파-저주파 결과는 2번째 레벨의 계산을 위해 LL 버퍼에 저장된다. 반면 만약 H 프로세서가 짹수번째 행에 대한 고주파, 저주파 결과들을 출력한다면, 수직 필터는 위의 스케줄링에 따라 2번째 레벨 이상의 행방향과 열방향 연산을 수행한다. 수직 필터는 그림 9와 같은 파이프라인 구조를 가진다.

행방향 연산은 경계영역 처리를 위해 부가적인 전달

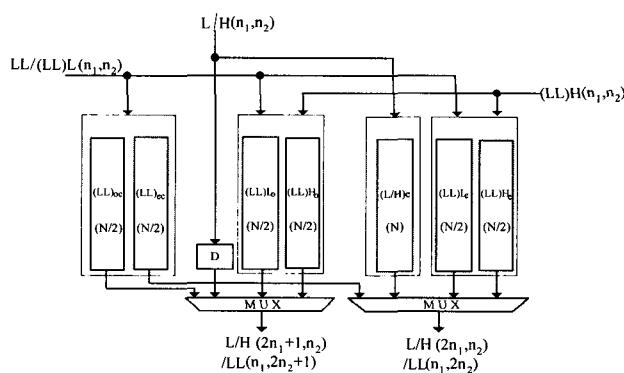


그림 8. 라인 메모리 버퍼의 블록도

Fig. 8. Block diagram of line memory buffer.

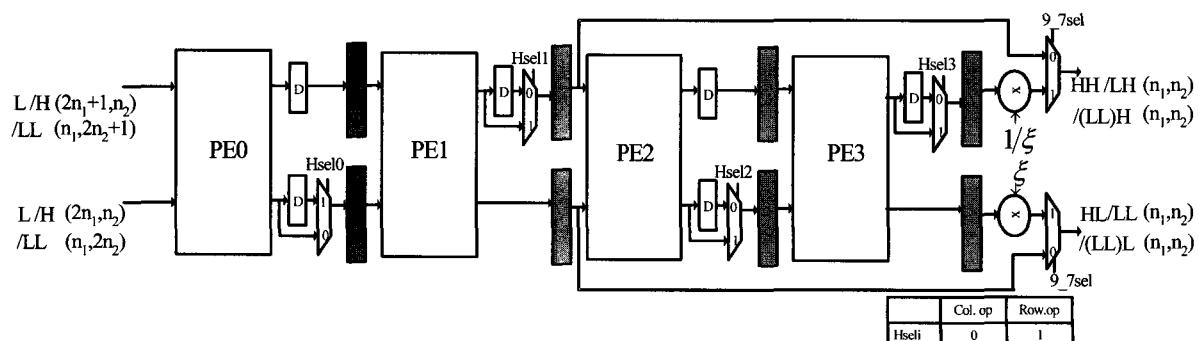


그림 9. 수직 필터의 블록도

Fig. 9. Block diagram of vertical filter.

지연을 요구하기 때문에, 행방향과 열방향 연산을 하나의 필터에서 처리할 때 파이프라인 충돌 또는 stall이 발생할 수 있다. 이와 같이 행방향 연산과 열방향 연산 사이의 파이프라인 충돌을 피하기 위해 본 논문에서는 그림 9와 같이 레지스터들과 MUX들을 추가하였다. 그림 9에서 각 PE는 경계영역 처리를 위한 MUX들을 제외하고는 일치하며, 그림10은 PE0의내부 구조를 나타낸다.

수직 필터내의 각 PE는 행방향의 상태변수들을 위해 1개의 레지스터를 요구하며, 열방향의 상태 변수를 위해  $2N (=N+(N/2)+\dots+(N/2^{J-1}))$  크기의 버퍼를 필요로 한다. 이와 같이 수직 필터는 스케일링 인수를 포함하여 6개의 곱셈기와 8개의 덧셈기, 그리고  $8N$  크기의 버퍼 들을 포함한다.

#### IV. 모델링 및 성능 비교

##### 1. VHDL 모델링

제안된 리프팅 DWT 구조는 VHDL을 사용하여 RTL 수준에서 모델링 되었다. 모델링 된 구조는  $64 \times 64$  영상 크기를 사용하며, 3 분해 레벨까지 수행한다. 내부

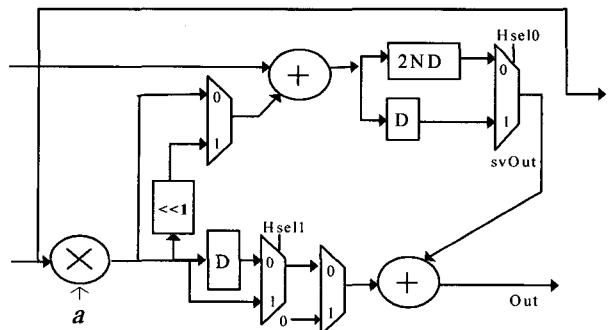
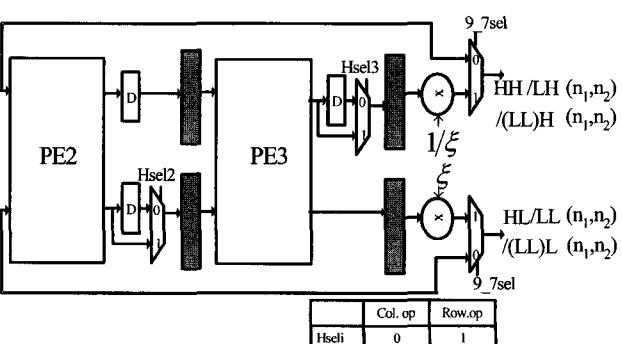


그림 10. 수직 필터내 PE0 구조

Fig. 10. Structure of PE0 in vertical filter.



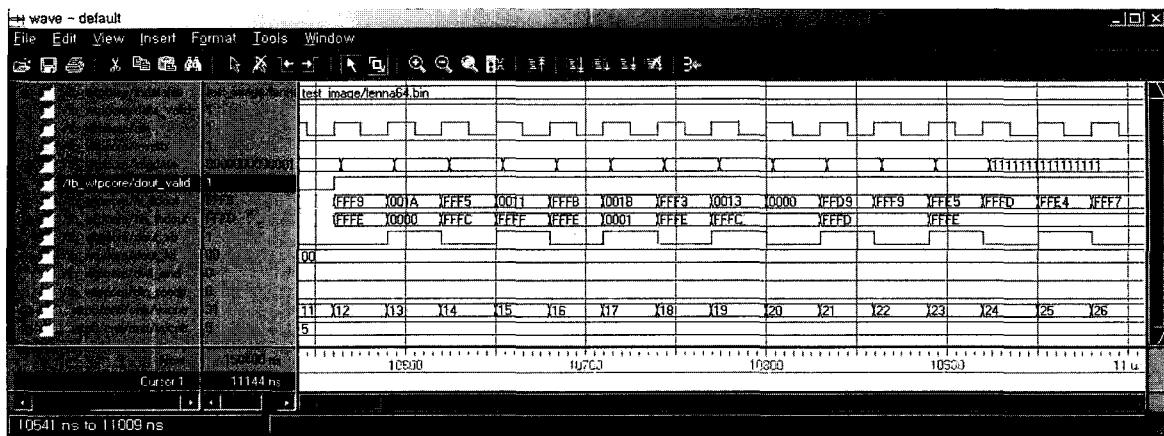


그림 11. 제안된 구조의 1 레벨에 대한 시뮬레이션 결과

Fig. 11. Simulation result of proposed architecture at 1 level.

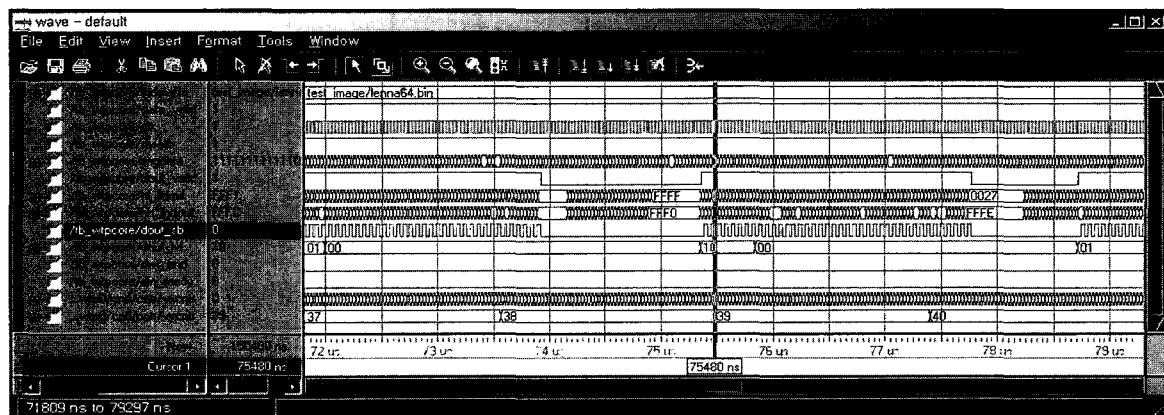


그림 12. 제안된 구조의 다중 레벨에 대한 시뮬레이션 결과

Fig. 12. Simulation result of proposed architecture at multi-level.

정밀도는 일반적인 영상 응용에서 주로 사용하는 입력 8비트, 출력 16 비트를 사용하였고, 필터계수의 정밀도는 부호비트를 포함 12비트를 사용하였다.

그림 11, 12는 모델링된 구조의 64×64 Lenna 영상에 대한 VHDL 시뮬레이션 결과를 나타낸다. 그림 11은 첫번째 레벨(dout\_lvl="00")에 대한 제안된 구조의 입, 출력을 나타내며 출력 enable 신호(dout\_valid)가 '1' 일 때 출력 데이터(hl\_lldout, hh\_lhdout)들은 각 서브밴드 출력을 의미한다. 이때 dout\_sb 가 '0'이면 HL, HH 밴드 출력을 지시하며, '1'일 때는 LL, LH 출력을 지시한다. 그림 12는 2 레벨이상(dout\_lvl= "01" or dout\_lvl="10") 대한 실험 결과를 나타낸다.

기능 검증 후, 제안된 리프팅 DWT 구조는 Altera APEX 20K FPGA (EP20K300EQC)로 구현되었다. 리프팅 DWT 구조는 9개의 12×16 Modified Booth 곱셈기와 12개의 CLA를 포함하여 2입력 nand 게이트기준으로 36,548 게이트 크기를 가지며 내부적인 저장을 위해 dual-port SRAM 모듈을 사용한다. 구현된 FPGA는

5,328 LE(Logic Element)와 12,288 ESB(Embedded System Block) 비트들을 포함하고 동작 주파수는 37MHz이다.

## 2. 성능 비교

본 절에서는 제안된 구조와 DWT의 모든 레벨 연산에 대해 평균  $N^2$  컴퓨팅 시간을 가지는 이전 2-D DWT 구조들 과의 성능 비교를 수행하였다. 비교에 사용되는 필터는 biothogonal (9,7) 필터이며, 곱셈기 수, 덧셈기 수, 메모리크기, 임계경로에 대해 비교하였다. 성능 비교는 표2에 요약하여 나타내었으며, 표에서 스케일링 인수들은 곱셈기 수 계산에 고려되었으며, 임계 경로 계산에는 파이프 라인은 적용하지 않았다.

표 2에서 [10], [11]의 구조들은 컨볼루션 방법을 사용한 구조들이며, [6]의 구조와 본 구조는 리프팅 방법을 사용한다. 표2에서 알수 있듯이 리프팅 방법을 사용한 구조(제안된 구조, [6]의 구조)는 컨볼루션 방법을 사용한 구조에 비해 적은 하드웨어 양을 갖는다. [6]의

표 2. Biothogonal (9,7) 필터에 대한 기존 구조들과의 비교(Tm:곱셈 전달 지연, Ta: 덧셈 전달 지연)

Table 2. Comparison of various architectures for biothogonal (9,7) filter(Tm: latency of multiplication, Ta: latency of addition).

	곱셈기수	덧셈기수	내부 메모리	임계 경로
제안된 구조	9	12	12N	$4T_m+4T_a$
리프팅 <sup>[6]</sup>	12	16	14N	$4T_m+8T_a$
Folded <sup>[10]</sup>	36	36	$18N+(3/2)N$	$T_m+4T_a$
시스템 병렬 <sup>[11]</sup>	36	36	22N	$T_m+4T_a$

구조와 본 구조를 비교했을 시는 [6]의 구조는 행방향 연산을 위한 처리 유닛 4개, 열방향 연산을 위한 처리 유닛 4개를 사용하는 반면 제안된 구조는 2번째 레벨 이상의 행방향 처리를 열방향 처리 유닛에서 수행하게 함으로써 행방향 연산을 위한 처리 유닛 2개, 열방향 연산을 위한 처리 유닛 4개를 사용하여 곱셈기 수와 덧셈기 수들과 같은 하드웨어 양을 적게 가진다.

## V. 결 론

본 논문에서는 JPEG2000의 리프팅 DWT의 라인 기반의 하드웨어 구조를 제안하였다. 제안된 구조의 각 리프팅 스텝을 계산하는 처리 유닛은 1개의 곱셈기와 1개의 덧셈기의 임계경로를 가지며, 전체 구조는 9개의 곱셈기, 12개의 덧셈기, 그리고  $N \times N$  영상에 대해 12N의 내부 메모리를 포함한다.

성능 비교에서도 알 수 있듯이 제안된 구조는 같은 성능을 가지는 구조들에 비해 하드웨어 양이 적으며, 라인 메모리만을 사용하기 때문에 단일 칩 VLSI 구현에 적합하다.

## 참 고 문 헌

- [1] "JPEG2000 Image Coding System", JPEG2000 final committee draft version 1.0, March 2000.
- [2] C. Christopoulos, A. Skodras and T. Rbrahimi, "The JPEG2000 Still Image coding System: An Overview", IEEE Trans. On consumer Electronics, Vol. 46, No. 4, pp. 1103-1127, November 2000.
- [3] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting schemes", The Journal of Fourier Analysis and Applications, Vol. 4, pp.247-269, 1998.

- [4] K. Andra, C. Chakrabarti and T. Acharya, "A VLSI Architecture for Lifting-Based Forward and Inverse Wavelet Transform", IEEE Trans. Signal Processing, Vol. 50, No. 4, pp. 966-977, April 2002.
- [5] G. Dillen, B. Georis, J.D. Legat and O. Cantineau, "Combined Line-Based Architecture for the 5-3 and 9-7 Wavelet Transform of JPEG2000", IEEE Trans. Circuits and Systems for Video Technology, Vol. 13, No. 9, pp. 944-950, September 2003.
- [6] C. T. Huang, P. C. Tseng, and L.G. Chen, "Efficient VLSI architectures of Lifting-Based Discrete Wavelet Transform by Systematic Design Method", Proc. Of IEEE Int. Sym. on Circuits and Systems(ISCAS), vol. 5, pp. 26-29, May 2002.
- [7] C. Chakrabarti and M. Vishwanath, "Efficient Realizations of the Discrete and Continuous Wavelet Transforms: from Single Chip Implementations to Mappings on SIMD array Computers", IEEE Trans. Signal Processing, Vol. 43, No. 3, pp.759-771, Mar 1995.
- [8] S. G. Mallat, "A theory of multiresolution signal decomposition: the wavelet representation", IEEE Trans. on Pattern Recognition and Machine Intelligence, Vol. 11, No. 7, pp. 674-693, July 1989.
- [9] K. C. B. Tan and T. Arslan, "Low power embedded extension algorithm for the lifting based discrete wavelet transform in JPEG2000", Electronic Letters, Vol. 37, pp. 1328-1330, October 2001.
- [10] C. Chakrabarti, C. Mumford, "Efficient Realizations of encoders and decoders based on the 2-D Discrete Wavelet Transform", IEEE Trans. VLSI Systems, pp.289-298, September 1999.
- [11] M. Vishwanath, R. M. Owens, and M. J. Irwin, "VLSI architectures for the discrete wavelet transform", IEEE Trans. Circuits and Systems II, Anlog and digital Signal Processing, Vol. 42, No. 5, pp.305-316, May 1995.

---

저자소개

---



**정 갑 천(정희원)**  
 1996년 전남대학교 컴퓨터공학과  
 학사.  
 1998년 전남대학교 전자공학과  
 석사.  
 1999년 ~ 현재 전남대학교 고품질  
 전기전자부품 및 시스템  
 연구센터 연구원.

1998년 ~ 현재 전남대학교 전자공학과 박사과정.  
 <주관심분야: 저전력 프로세서 구조, 영상압축,  
 영상통신용 ASIC 설계, DSP 설계, VLSI 설계 및  
 CAD 등>



**박 성 모(정희원)**  
 1977년 서울대학교 전자공학과  
 학사.  
 1979년 한국과학기술원 전기 및  
 전자공학과 석사.  
 1988년 노스캐롤라이나 주립대학  
 전기 및 컴퓨터공학과 공  
 학박사.  
 1979년 ~ 1984년 한국전자기술연구소 설계개발부  
 선임연구원.  
 1988년 ~ 1992년 올드도미니언 대학교 전기 및  
 컴퓨터공학과 조교수.  
 1992년 ~ 현재 전남대학교 컴퓨터공학과 교수.  
 2002년 ~ 2004년 전남대학교 정보전산원 원장 겸임.  
 <주관심분야: 멀티미디어 프로세서 구조, VLSI  
 시스템 설계, 신호처리용 ASIC 설계, 영상압축, 임  
 베디드 시스템 등>