

이미지 압축을 위한 Lifting Scheme을 이용한 병렬 2D-DWT 하드웨어 구조

Parallel 2D-DWT Hardware Architecture for Image Compression Using the Lifting Scheme

金 鍾 旭*, 鄭 正 和*

Jong-Woog Kim*, Jong-Wha Chong*

요 약

본 논문에서는 2차원 분할을 이용한 병렬 처리가 가능한 리프팅 스킴(lifting scheme) DWT(Discrete Wavelet Transform)를 구현하는 하드웨어 구조를 제안한다. 기존의 DWT 하드웨어 구조는 웨이블릿(Wavelet) 변환이 갖는 특성 때문에 병렬 처리 구조를 구현하는 데 있어서 메모리와 하드웨어 자원이 많이 필요하였다.

제안된 구조는 기존의 구조와 달리 데이터 흐름을 분석하여, 분할 과정을 2차원으로 수행하는 방법을 제안하였다. 이러한 2차원 분할 방법을 파이프라인 구조를 사용하여 병렬 처리의 효율을 증가 시켜 50% 정도의 출력 지연의 감소된 결과를 얻을 수 있었다. 또한 데이터 흐름의 분석과 출력 지연의 감소는 내부 메모리의 사용을 감소 시켰으며, 리프팅 스킴의 특성을 이용하여 외부 메모리의 사용을 감소시키는 결과를 얻을 수 있다.

Abstract

This paper presents a fast hardware architecture to implement a 2-D DWT(Discrete Wavelet Transform) computed by lifting scheme framework. The conventional 2-D DWT hardware architecture has problem in internal memory, hardware resource, and latency. The proposed architecture was based on the 4-way partitioned data set. This architecture is configured with a pipelining parallel architecture for 4-way partitioning method. Due to the use of this architecture, total latency was improved by 50%, and memory size was reduced by using lifting scheme.

Key words : DWT, Hardware Architecture, Lifting Scheme

1. 서 론

최근 몇 년간 동영상상을 비롯한 영상 압축기법에서 웨

이블릿을 이용한 압축 기법들이 소개 되고 있고, MPEG-4와 JPEG2000과 같은 영상 압축 표준안에서 웨이블릿 기반의 압축 기법을 채용하고 있다. 기존의 블록 DCT 기반의 압축 코덱에서는 압축률이 올라 감에 따라서 블록의 경계 부분에 심한 열화 현상이 나타나는 블록킹 현상(blocking artifacts)가 나타나는 단점을 가지고 있었다. 그리고, DCT 기반의 경우는 다양한 전송 환경에 따른 scalability를 구현하는데 많은

* 漢陽大學校 電子電氣컴퓨터工學部
(Department of Electronic & Electrical and Computer
Engineering, Hanyang University)

接受日:2002年 4月30日, 修正完了日:2002年7月25日

약점을 가지고 있었다. 이에 반해 웨이블릿은 하나의 프레임을 블록 단위로 분할하지 않고 전체 영상에 대해 변환을 하고, 변환의 결과가 웨이블릿의 레벨에 따라 밴드 별로 나누어 지기 때문에 전송 레벨의 조정을 통해 scalability를 구현 할 수 있다는 장점을 가지고 있다.

이러한 웨이블릿 기반의 압축 코덱에 있어서 가장 큰 단점은 전체 영상이 처리 기준이 되기 때문에 최종 결과를 얻기 위한지연 시간이 길어지고, 각 밴드에 대한 양자화 결과가 화면 전체의 화질에 영향을 줄 수 있다는 단점이 있었다. 이에 다양한 형태의 웨이블릿 기반의 하드웨어 아키텍처가 제안 되고 있으며, 이러한 하드웨어 아키텍처의 기반이 되는 것은 리프팅 스킴이 많이 사용되고 있다([1]-[4],[6],[7]).

이러한 하드웨어의 구현에 있어서 고려해야 할 점은 지연시간을 최대한 줄이고, 처리 속도의 향상과 메모리의 사용을 최대한 억제하는데 있다고 할 수 있다. 본 논문에서는 리프팅 스킴에서의 예측(prediction)과 보상(update) 과정을 통합하는 방법을 제안하고, 이를 구현하기 위한 병렬 처리 하드웨어 아키텍처와 자원 스케줄링 방법을 제안하였다.

본 논문의 구성은 2장에서 lifting scheme의 간략한 구조에 대하여 설명하고, 3장에서는 제안하는 예측과 보상 방법을 위한 분할 방법과 제안하는 하드웨어 구조를 기술 하였으며, 4장에서는 비교 결과 및 고찰을 기술하고, 5장에서 결론을 맺는다

II. 리프팅 스킴의 웨이블릿 구조

일반적인 리프팅 스킴을 이용한 웨이블릿을 구현하기 위해 그림 1과 같이 세 단계의 처리 과정을 거치게 되는데, 이것은 입력 데이터에 대하여 split, prediction, update 과정이다[5]. Split은 입력된 데이터를 even/odd의 데이터 집합으로 나누고, 이것은 기존의 웨이블릿에서 행하는 샘플링 과정에 해당하고, 이렇게 분리된 데이터는 고주파 영역의 데이터를 얻기 위한 prediction 과정을 거치게 된다. Prediction은

분리된 두개의 데이터 영역의 상관관계를 구성하는 것으로 결국 이미지에서 고주파 영역에 해당하는 데이터를 갖게 된다. 이렇게 고주파 영역의 데이터를 구한 후에 이미지 전체에 대한 에너지 보존을 위해 update과정을 거치게 된다

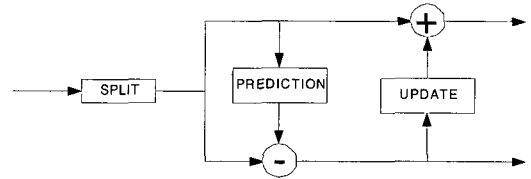


그림 1. 리프팅 스킴 웨이블릿의 블록 다이어그램
Fig. 1 Block diagram of lifting scheme wavelet

2D lifting scheme의 경우 우선 X 방향의 prediction과 update과정을 수행하고, 이것을 다시 Y 방향으로 연산을 하여 최종 결과를 얻게 된다. 2차원의 경우 X 방향의 연산을 수식으로 표현 하면 식(2.1)과 같다.

$$\begin{aligned} X_H^{new}(n, k) &= x_H(n, k) - P_x(x_L(n, k)) \\ X_L^{new}(n, k) &= x_L(n, k) + U_x(X_H^{new}(n, k)) \end{aligned} \quad (1)$$

$$\begin{aligned} \text{이때, } 0 \leq n \leq \frac{N}{2} \quad (N : \text{image width}) \\ 0 \leq k \leq M \quad (M : \text{image height}) \end{aligned}$$

2차원의 경우는 식 (1)을 이용하여 X방향으로 변환을 수행하고, 그 결과값을 다시 Y 방향으로 수행하여 결과를 얻게 된다. 이러한 과정을 이용한 최종 결과는 4개의 밴드로 구성되는 2차원 변환 결과 값을 갖게 되고 각 밴드별 수행을 위한 수식을 보면 식 (2)와 같이 기술된다.

$$\begin{aligned} X_{HH}^{new}(n, k) &= X_H^{new}(n, 2k+1) - P_y(X_H^{new}(n, 2k)) \\ X_{HL}^{new}(n, k) &= X_L^{new}(n, 2k+1) - P_y(X_L^{new}(n, 2k)) \\ X_{LH}^{new}(n, k) &= X_H^{new}(n, 2k) + U_y(X_H^{new}(n, 2k+1) - P_y(X_H^{new}(n, 2k))) \\ X_{LL}^{new}(n, k) &= X_L^{new}(n, 2k) + U_y(X_L^{new}(n, 2k+1) - P_y(X_L^{new}(n, 2k))) \end{aligned} \quad (2)$$

$$\begin{aligned} \text{이때, } 0 \leq n \leq \frac{N}{2} \quad (N : \text{image width}) \\ 0 \leq k \leq \frac{M}{2} \quad (M : \text{image height}) \end{aligned}$$

III. 제안하는 하드웨어 구조

3.1 2차원 분할 방법

제안하는 구조는 식(2)를 바탕으로 하여 구성 되는데, 식 (2)에서 각 첨자들 $n, 2k, 2k+1$ 은 2차원 데이터의 위치에 해당한다. 이 위치를 정리해 보면 다음의 식(3)과 같이 첨자를 정리 할 수 있다.

$$\begin{aligned} x_L(n, 2k) &\Rightarrow x_{LL}(n, m) \\ x_H(n, k) &\Rightarrow x_{LH}(n, m) \\ x_H(n, k+1) &\Rightarrow x_{HH}(n, m) \\ x_L(n, k+1) &\Rightarrow x_{HL}(n, m) \end{aligned} \quad (3)$$

식(3)과 같이 데이터를 분할하면 다음의 그림2와 같이 입력 이미지 데이터를 4분할 한 것과 같이 된다. 이렇게 함으로써 각 첨자 n, m 의 범위를 $0 \leq n \leq \frac{N}{2}, 0 \leq m \leq \frac{M}{2}$ 로 줄일 수 있다.

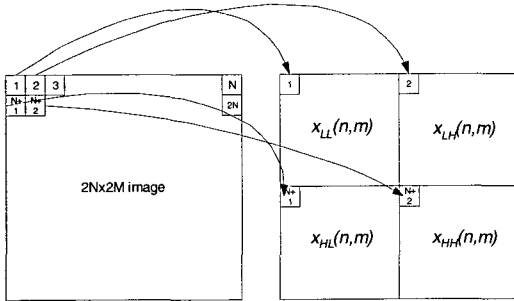


그림 2. 2차원 데이터 분할
Fig. 2 2-D data splitting

식(3)을 이용하여 식(2)를 다시 정리하면 다음의 식(4)와 같이 표현 된다.

$$\begin{aligned} X_{LL}^{***}(n, k) &= x_{LL}(n, k) - U_x P_x(x_{LL}(n, k)) - U_y P_y(x_{LL}(n, k)) + U_x P_x U_y P_y(x_{LL}(n, k)) \\ &\quad + U_y(x_{LH}(n, k)) - U_x P_x(x_{LH}(n, k)) + U_y(x_{HL}(n, k)) - U_x P_x(x_{HL}(n, k)) \\ &\quad + U_y U_y(x_{HH}(n, k)) \\ X_{LH}^{***}(n, k) &= -P_x(x_{LL}(n, k)) + U_x P_x P_y(x_{LL}(n, k)) + x_{LH}(n, k) - U_y P_y(x_{LH}(n, k)) \\ &\quad - U_x P_x(x_{HL}(n, k)) + U_y(x_{HH}(n, k)) \\ X_{HL}^{***}(n, k) &= -P_y(x_{LL}(n, k)) + P_x U_x P_y(x_{LL}(n, k)) - P_x U_x(x_{LH}(n, k)) + x_{HL}(n, k) \\ &\quad - U_x P_x(x_{HL}(n, k)) + U_y(x_{HH}(n, k)) \\ X_{HH}^{***}(n, k) &= P_x P_x(x_{LL}(n, k)) - P_x(x_{LH}(n, k)) - P_x(x_{HL}(n, k)) + x_{HH}(n, k) \end{aligned} \quad (4)$$

이렇게 2차원 분할을 통한 웨이블릿의 수행에 있어서 각각의 prediction을 위한 P 필터와 update를 위한 U 필터가 다항식 형태의 전달 함수를 갖고 있으므로 각 밴드의 웨이블릿 계수를 구하기 위한 입력 값으로 $x_{LL}(n, m), x_{LH}(n, m), x_{HL}(n, m), x_{HH}(n, m)$ 의 2차원 분할된 입력 값을 사용할 수 있게 된다. 이렇게 함으로써 1차원 DWT를 두 번 수행하여 구하는 2차원 DWT를 보다 효율적으로 수행 할 수 있는 파이프라인 구조를 구성할 수 있다.

3.2 병렬 처리 하드웨어 구조

일반적인 웨이블릿 하드웨어는 그림3 (a)와 같은 처리 순서를 갖게 된다. 이러한 순차 구조는 구현에 있어서 하드웨어 복잡도는 낮지만 수행 시간이 많이 걸린다는 단점이 있다.

본 논문에서 제안하는 방법은 3.1에서 분할된 2차원 데이터를 바탕으로 그림3 (b)와 같이 각 밴드 별로 데이터를 병렬 처리 하는 방법을 제안하였다.

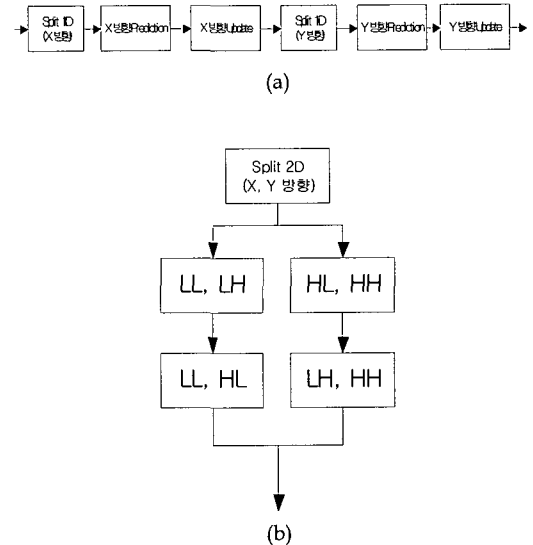


그림 3. 2차원 웨이블릿 처리 과정
(a) 기존의 2차원 웨이블릿 처리 과정,
(b) 제안하는 2차원 웨이블릿 처리 과정
Fig. 3 2D Wavelet transform procedure.
(a) Sequential Wavelet transform procedure
(b) Proposed Wavelet transform procedure

식(1)과 식(3)을 사용하여 정리하면

$$X_{LH}^{new}(n, k) = x_{LH}(n, k) - P_x(x_{LL}(n, k)) \quad (5)$$

$$X_{HH}^{new}(n, k) = x_{HH}(n, k) - P_x(x_{HL}(n, k)) \quad (6)$$

$$X_{LL}^{new}(n, k) = x_{LL}(n, k) + U_x(X_{LH}^{new}(n, k)) \quad (7)$$

$$X_{HL}^{new}(n, k) = x_{HL}(n, k) + U_x(X_{HH}^{new}(n, k)) \quad (8)$$

과 같이 전개 할 수 있다. 그 결과 식(5), (6)을 병렬로 처리하고, 식(7), (8)을 병렬로 처리가 가능하게 된다. 그 결과 기존의 구조에서는 보상과 예측을 각 한 개의 모듈만을 병렬로 처리 할 수 있었는데 제안하는 구조는 각각 2개의 모듈을 병렬로 처리 할 수 있는 구조를 갖게 된다. 그리고, 1차원 결과를 파이프 라인을 이용하여 2차원 결과를 얻을 수 있도록 그림 4와 같은 전체 구조를 제안하였다.

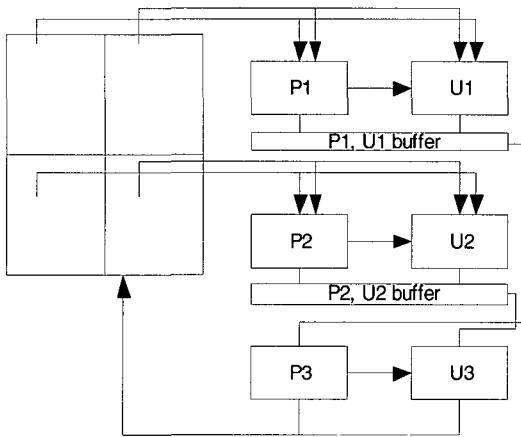


그림 4. 2D-DWT를 위한 구조의 블록 다이어그램
Fig. 4 Functional partitioning of 2-D DWT decomposition with lifting scheme

그림 4의 구조는 $x_{LL}(n, m)$, $x_{LH}(n, m)$ 밴드와 $x_{HL}(n, m)$, $x_{HH}(n, m)$ 밴드를 동시에 처리 할 수 있도록 2개의 P, U 모듈 셋(P1, U1, P2, U2)을 사용하여 1차원 DWT를 구성하였다. 그리고, 이 1차원 DWT의 결과를 이용하여 2차원 DWT를 수행하기 위한 추가 적인 P3, U3 모듈이 사용 되었고, 1차원의

결과를 저장하기 위한 버퍼 메모리를 갖게 된다.

제안하는 구조의 데이터 연관성을 보면 그림 5와 같이 밴드 별로 분할 된 입력 데이터와 1차원 결과를 이용한 2차원 변환에 필요한 데이터와의 관계를 알 수 있다. 즉 1차원 변환을 위해서는 LL, LH 와, HL, HH 밴드의 데이터가 동시에 필요로 하게 된다. 그리고, 2차원 변환을 위해서는 LL, LH의 결과와 HL, HH의 1차원 결과가 영향을 주게 된다. 제안하는 구조에서는 이러한 1차원의 결과를 저장하는데 있어서 1프레임 분량의 메모리가 필요한 것이 아니고, $4L_h N(L_h$: prediction 필터의 탭수, N : 이미지의 가로 픽셀수) 만큼의 임시 버퍼만을 이용하여 2차원 DWT를 수행 할 수 있다. 이것은 그림 5에서와 같이 Y 방향의 변환 과정을 수행하기 위해서는 X 방향의 L_h 개 만큼의 데이터가 필요로 하게 된다. 이렇게 L_h 만큼의 데이터가 모이게 되면 Y 방향의 변환 과정이 수행되게 된다.

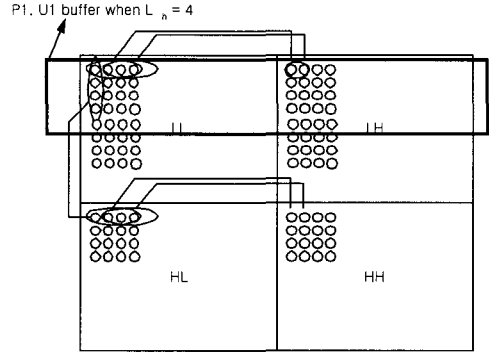


그림 5. 2차원 데이터 연관성 및 버퍼 메모리의 저장 범위
Fig. 5 2-D data dependency and buffer memory covered area

그림 4에서 P, U는 각각 prediction 모듈과 update 모듈을 의미하는 것으로 각 모듈은 필터의 탭 수에 따라 기본적인 필터 구조를 갖게 된다.

그림 6 와 그림 7은 각각 P, U 모듈의 구조를 나타 낸 것으로 각 모듈은 필터 탭수 만큼의 레지스터로 구성되며, 메모리와 연산 모듈간의 데이터 전송을 줄이기 위한 FIFO와 레지스터를 추가 하였다.

그림 6의 REG_{LH} 는 최종 prediction의 값을 구하기 위한 레지스터 이고, U 모듈의FIFO는 통상 update

필터의 탭수는 prediction 필터의 탭수보다 작게 구성 되는데, 이러한 특성에 의해 지연이 발생하게 된다. 이것을 처리 하기 위하여 FIFO를 사용하여 입력으로 들어오는 X_{LL} , X_{HL} 값을 저장해 놓을 수 있도록 구성 하였다.

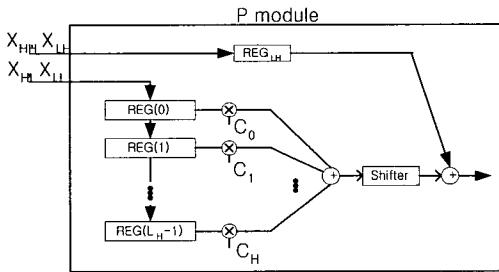


그림 6. P 모듈의 구조 및 데이터 흐름도
Fig. 6 P module architecture and signal flow

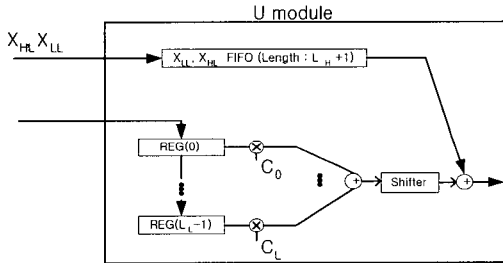


그림 7. U 모듈의 구조 및 데이터 흐름도
Fig. 7 U module architecture and signal flow

3.3 지연 시간 및 수행 흐름

기본 적인 2차원 DWT 구조에서는 최초 2차원 처리 결과를 얻기 위하여 $\frac{NM}{2} + L_h + L_l$ 만큼의 처리 지연 시간을 갖게 되는데 이것은 1차원의 처리를 위한 지연이고, 통상의 병렬 처리 구조에서는 이러한 지연을 줄이기 위해 1차원 결과를 파이프라인을 통하여 처리하는데 이러한 경우라도, 1차원 prediction 필터의 탭수 만큼의 지연($2NL_h + L_l$)이 발생하게 된다. 본 구조에서는 이러한 1차원 지연을 최소화 하기 위하여 P1, P2, U1, U2의 두개의 1차원 처리 모듈을 사용하였고, P3, U3 모듈을 이용하여 최종 지연 시간을 감소 시켰다.

제안하는 구조의 수행 흐름을 보면 다음의 3개의 스텝으로 구성 될 수 있다.

- 1st: LL, LH 데이터를 이용한 P1, P2의 연산 수행
- 2nd: HL, HH 데이터를 이용한 U1, U2 의 연산 수행
- 3rd: 1st, 2nd 의 결과 데이터를 이용한 P3, U3의 연산 수행

각 스텝의 초기 지연을 보면 1st , 2nd 스텝에서 L_h 만큼의 지연이 생기고, 이것이 2차원을 수행 시키기 위한 3rd스텝의 지연을 보면 $\frac{N(L_h + 1)}{2} + L_l$ 만큼의 지연이 발생하게 된다. 이러한 지연 시간의 감소는 2차원 데이터 분할을 이용하여 1차원의 처리를 병렬로 할 수 있도록 구성한 결과이다.

제안하는 구조는 병렬 처리와 밴드별 분리를 위하여 4개의 메모리 बैं크를 갖는 메모리 구조를 갖게 되는데, 각 메모리 बैं드는 외부 메모리로 구성되고, 초기 입력과 최종 결과를 저장하는 역할을 수행하게 된다. 일반적인 2D-DWT에서는 입력과 출력의 버퍼 메모리를 별도로 구성하지만 제안하는 구조에 입력 버퍼로 사용했던 메모리에 최종 결과를 저장하게 된다. 그림 8에 메모리 구조에 대하여 나타내었고, 그림 8에서와 같이 웨이블릿의 level에 따른 반복 수행을 위해 추가적인 메모리를 갖게 되는데, 추가 메모리를 포함하여 전체 필요한 외부 메모리의 양은 $\frac{5}{4}NM$ (NM : 전체 이미지 크기)으로 기존의 2개의 입력과 출력을 사용하는 메모리 구조에 비하여 37.5% 감소 된 만큼의 메모리를 사용한다.

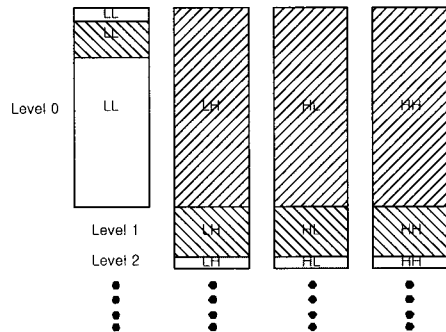


그림 8. 외부 메모리 구성
Fig. 8 The external memory configuration

IV. 비교 결과 및 고찰

제안하는 구조는 기존의 1차원 병렬 처리 구조에 비하여 하드웨어는 증가 하였지만 시스톨릭 어레이를 이용한 구조에 비하여서는 감소된 하드웨어 구조를 얻을 수 있었다. 그리고, 제안하는 구조는 2차원 분할을 이용한 병렬 처리를 이용하여 내부 메모리 버퍼의 용량을 감소 시켰으며, 초기 지연 시간을 50% 정도 감소 시키는 결과를 얻을 수 있었다. 표1은 기존의 구조와 제안하는 구조와의 멀티플라이어 수와 메모리 사이즈, 그리고, 지연시간에 대한 비교표이다. 표에서 알 수 있듯이 연산기의 개수는 증가 하지만 전체 지연시간에 있어서는 감소하는 결과를 얻을 수 있었다. 메모리 사이즈 역시 기존의 결과에 비해 향상된 결과를 얻을 수 있었다.

기존의 병렬 구조는 파이프라인 구조를 사용함에 있어 추가적인 지연 시간 감소를 얻기 힘들지만 제안하는 구조는 2차원 병렬 처리가 가능한 구조로 확장이 가능한 구조를 갖고 있다.

표 1 기존 구조와 멀티플라이어 수, 내부 메모리 크기, 지연 시간의 비교표

Table 1. Comparison of the multiplication resource, memory size, and total latency

Resource	Systolic[6]	Previous Lifting[7]	Proposed
Multipliers	$8L_h+4$	$2L_l + 2L_h$	$3L_l + 3L_h$
Memory Size	$8NL_h+4N$	$2N(2L_h-1+ L_l)$	$4NL_h$
Total latency	$O(N^2)$	$O(N^2)$	$O(\frac{N^2}{2})$

V. 결 론

제안하는 병렬 하드웨어 구조는 2차원 분할을 통하여 기존의 웨이블릿 변환기의 처리 속도 향상을 시키고, 파이프라인 하드웨어 구조를 제안하였다. 제안하는 구조는 기존의 구조에 비해 지연 시간(latency)를 감소 시키고, 효율적인 메모리 사용을 이용하여 내부

메모리 사용량을 감소 시키는 결과를 얻을 수 있었다. 그리고, 리프팅 스킴의 가장 큰 장점이라고 할 수 있는 외부 메모리 버퍼의 사용을 줄여 기존의 방법이 2 프레임을 저장하는 메모리 버퍼를 사용하는 것을 1 프레임과 1/4 만큼의 추가 적인 외부 메모리 버퍼만을 이용하여 처리 할 수 있도록 구조를 개선하였다. 그리고, 전체 지연 시간을 기존의 방법에 비해 50% 정도 향상된 결과를 얻을 수 있었다.

제안하는 구조를 기본으로 최소한의 하드웨어 증가로 최대한의 지연 시간 감소를 얻을 수 있도록 세부 하드웨어 구조를 개선하는 것이 향후 연구 과제로 남아 있다.

참 고 문 헌

- [1] Honggang Li, Qiao Wang, and Lenan Wu, "A Novel Design of Lifting Scheme from General Wavelet," *IEEE Trans. on Signal Processing*, vol. 49, no. 8, pp. 1714-1717, Aug. 2001.
- [2] Wenqing Jiang, and Antonio Ortega, "Lifting Factorization-Based Discrete Wavelet Transform Architecture Design," *IEEE Trans. on Circuits and System for Video Technology*, vol. 11, no. 5, pp. 651-657, May 2001.
- [3] Andra, K., Chakrabarti, C., and Acharya, T., "A VLSI architecture for lifting-based wavelet transform," *IEEE Workshop on Signal Processing Systems 2000*, pp. 70 -79, 2000.
- [4] Andra, K., Chakrabarti, C., and Acharya, T., "Efficient implementation of a set of lifting based wavelet filters," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 1101 -1104, 2001.
- [5] W. Sweldens, "The lifting scheme: A new philosophy in biorthogonal wavelet structions," *processings of SPIE*, vol. 2569, pp. 68-79, 1995.
- [6] M. Vishwanath, R.M. Owens, and M.N. Irwin, "VLSI Architecture for the Discrete Wavelet

Transform," *IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 42, No. 5, pp. 305-316, 1995

- [7] M. Ferretti, and D. Rizzo, "A Parallel Architecture for the 2-D Discrete Wavelet Transform with Integer Lifting Scheme," *Journal of VLSI Signal Processing*, vol. 28, pp. 165-185, 2001

저자 소개

金 鍾 旭 (正會員)



1968년 3월 1일생

1992년 한양대학교 전자공학과 (공학사)

1994년 한양대학교 대학원 전자공학과(석사)

1995년 ~ 현재 한양대학교 대학원 전자공학과 박사 과정

1999년 ~ 2001년 주)디지털 펄스 미디어텍 연구소 부소장, 관심분야 : 동영상 압축, 전송 및 하드웨어 설계 기법,

鄭 正 和 (正會員)



1950년 3월 10일생

1975년 2월 한양대학교 전자공학과(공학사)

1977년 2월 한양대학교 대학원 전자공학과(석사)

1981년 3월 일본 와세다대학교 대학원 전자공학과 (박사)

1986년 6월 ~ 1987년 9월 미국 Berkeley 대학 박사후과정, 2002년 1월 ~ 현재 대한전자공학회 학술이사, 2000년 10월 ~ 현재 한양대학교 정보통신 대학장 및 대학원장. 관심분야: HW/SW Co-design, High Speed wireless LAN system, MPEG encoder/decoder chip design