

논문 2010-47SD-12-6

100% 하드웨어 효율을 갖는 블록기반의 이차원 이산 웨이블릿 변환 필터 설계

(Design of a Block-Based 2D Discrete Wavelet Transform Filter with
100% Hardware Efficiency)

김 주 영*, 박 태 근**

(Juyoung Kim and Taeguen Park)

요 약

본 논문에서는 하드웨어 효율이 100%가 되는 2차원 이산 웨이블릿 변환 필터 구조를 제안한다. 전체 구조는 두 채널 QMF PR Lattice 필터로 구성된 1차원 DWT 필터 4개로 구성되었다. 1 레벨부터 J 레벨까지 순차적으로 수행함으로써 메모리 사용을 최소화 하면서도 하드웨어 효율이 100%가 되도록 설계하였으며 필터 입력 데이터를 구성해주는 DFC구조와 DCU구조를 제안하였다. 인접한 4개의 데이터를 동시에 입력 받아 처리함으로써 동시에 행방향과 열방향 DWT를 수행하므로 $N \times N$ 이미지를 처리하는데 $N^2(1-2^{-2J})/3$ 사이클이 소요되며 이 때 필요한 저장공간은 약 $2MN-3N$ 이다. 기존의 2D DWT 구조와 비교해 보았을 때 하드웨어 효율과 동작 속도가 향상되었으며 두 개의 1D DWT를 직렬로 연결하므로 임계경로를 감소시키기 위해서 최대 4 단계까지 파이프라인을 적용하여 임계경로를 향상시킬 수 있다. 제안된 구조는 VerilogHDL로 모델링되고 동부아남 0.18 μ m 표준셀로 합성되어 검증되었다.

Abstract

This paper proposes a fully-utilized block-based 2D DWT architecture, which consists of four 1D DWT filters with two-channel QMF PR Lattice structure. For 100% hardware utilization, we propose a new method which processes four input values at the same time. On the contrary to the image-based 2D DWT which requires large memories, we propose a block-based 2D DWT so that we only need $2MN-3N$ of storages, where M and N stand for filter lengths and width of the image respectively. Furthermore, the proposed architecture processes in horizontal and vertical directions simultaneously so that it computes the DWT for an $N \times N$ image within a period of $N^2(1-2^{-2J})/3$. Compared to existing approaches, the proposed architecture shows 100% of hardware utilization and high throughput rate. However, the proposed architecture may suffer from the long critical path delay due to the cascaded lattices in 1D DWT filters. This problem can be mitigated by applying the pipeline technique with maximum four level. The proposed architecture has been designed with VerilogHDL and synthesized using DongbuAnam 0.18 μ m standard cell.

Keywords : DWT, QMF filter, Lattice structure, pipeline

* 학생회원, ** 정회원(교신저자), 가톨릭대학교 정보통신 전자공학부

(Department of Information, Communication, and Electronic Engineering, The Catholic University of Korea)

※ 본 연구는 2009년도 가톨릭대학교 교비연구비의 지원으로 이루어졌음.

접수일자: 2010년7월21일, 수정완료일: 2010년11월24일

I. 서 론

정보통신 기술이 발전하고 다양한 멀티미디어 기기가 확산됨에 따라 많은 멀티미디어 정보들이 생성되고 활용되어 진다. 디지털 영상 정보를 그대로 파일로 저장할 경우 많은 기억용량을 차지하게 되고 영상을 전송

할 때에는 많은 시간과 비용이 소요된다. 따라서 정보로서의 가치가 떨어지게 되므로 정보량을 압축하는 부호화 기술과 압축된 영상을 원 영상으로 복원하는 복호화 기술의 개발은 정보화 시대의 필수 요소가 되었다.

영상 정보의 압축을 위하여 가장 널리 사용되는 변환 알고리즘은 블록 기반 변환과 이미지 기반 변환의 두 가지 종류로 나누어진다. 대표적인 블록 기반 변환으로는 이산 역현 변환(DCT, Discrete Cosine Transform)이 있는데 일정 크기의 블록 단위로 처리된다. 블록 기반 변환은 요구되는 메모리가 작고 블록 기반 움직임 보상의 오차 데이터 압축과 잘 조화되지만 블록효과(blocking effect)가 발생할 수 있다. 이미지 기반 변환은 이미지 또는 프레임 전체에 대해 수행된다. 가장 널리 알려진 이미지 기반 변환으로는 이산 웨이블릿 변환(DWT, Discrete Wavelet Transform)이 있는데 응용수학에서 처음 소개된 후, 인식 분야에서 연구되어 온 다해상도 표현과 연관성이 있음이 알려졌고, 이산 웨이블릿 이론은 필터 설계 방법과 동일함이 밝혀졌다^[2, 13]. 또한, 정지 이미지 압축에 있어서 블록 기반의 변환보다 성능이 뛰어나다는 것이 입증되었으며 DCT와 달리 블록효과가 없고, 압축률 또한 높으므로 저속의 비트 레이트에서 뛰어난 품질의 영상을 제공한다^[1~2]. 따라서 이미지 압축과 음성 분석, 패턴 인식, 그리고 컴퓨터 비전 등 다양한 분야에서 응용되고 있으며, JPEG2000의 표준으로 채택되었다^[3~4]. 그러나 반복된 계산으로 인해 처리 시간이 많이 소요된다는 단점이 있다. 따라서 90년대부터 이산 웨이블릿 변환 필터의 VLSI 구현에 대한 연구가 활발히 진행되어 왔다^[5~12].

Lattice 구조는 DWT의 다해상도 특성을 효율적으로 지원할 뿐만 아니라 직접 구현 방법(direct method)에 비해서 하드웨어의 복잡도가 줄어들고 규칙적인 모듈로 이루어져 있어 VLSI 구현에 적합하다^[6, 9]. 또한 중첩 스케줄링을 이용하여 QMF(Quadrature Mirror Filter) PR (Perfect Reconstruction) Lattice 필터로 모든 레벨을 수행함으로써 성능을 높이면서도 하드웨어 비용을 낮출 수 있다. 하지만 2차원 영상에 적용 시에 휴면동작 구간으로 인해 하드웨어 효율은 100%를 보여주지 못한다^[9, 12].

본 논문에서는 하드웨어 효율이 100%가 되는 2 차원 DWT 필터구조를 제안한다. 전체 구조는 행방향과 열방향에 대한 DWT를 동시에 계산하기 위해서 두 채널 QMF PR Lattice 필터를 사용한 1 차원 DWT 필터 4

개로 구성된다. 간축에 의한 휴면 동작 구간동안 다른 해상도의 레벨을 수행하는 [9]와 [12]의 중첩 스케줄링과 달리 J 레벨까지 수행하는 경우 $2^J \times 2^J$ 의 블록 단위로 지그재그 스캔 하여 블록 내의 짝수 행과 홀수 행의 현재 입력 샘플과 이전 샘플을 동시에 입력 받아 처리함으로써 휴면구간을 없애고 1 레벨부터 J 레벨까지 순차적으로 처리하여 하드웨어 효율이 100%가 되는 2차원 DWT 구조를 설계하였다. 또한 새로운 스케줄링에 맞게 입력 데이터를 적절히 구성해주는 DFC(Data Format Converter)와 중간 데이터의 지연을 제어해주는 DCU(Delay Control Unit) 구조를 제안하였다. 본 논문의 구성은 다음과 같다. 먼저 II장에서는 이차원 이산 웨이블릿 변환의 이론적 배경과 기존에 연구된 구조에 대하여 설명하고, III장에서는 제안한 스케줄링과 구조에 대해서 설명한다. 그리고 IV장에서는 성능을 비교 분석하였으며 마지막으로 V장은 본 논문의 결론이다.

II. 2차원 이산 웨이블릿 변환

1989년 Mallat에 의해서 소개된 웨이블릿은 주파수만을 변화시키며 상관관계를 밝히는 푸리에 변환과 달리 한 파장의 파형을 기본 파형으로 하여 그 크기와 위치를 변화시켜가며 상관관계를 밝히는 기술이다. 웨이블릿 변환은 한 파장의 파형을 크기와 함께 위치도 변화시키므로 주파수 정보와 함께 시간의 정보도 알 수 있다는 장점이 있다.

이산 웨이블릿 변환은 식 (1)과 같이 임의의 신호 $f(t)$ 를 그 신호의 시간과 주파수의 특성을 나타내는 기저 함수 $\Psi_{j,k}(t)$ 와 웨이블릿 계수 $a_{j,k}$ 의 결합된 형태로 표현된다.

$$f(t) = \sum_{j,k} a_{j,k} \Psi_{j,k}(t) \quad (1)$$

웨이블릿을 이용한 다해상도 분석에서는 두 개의 기본 함수, 즉 스케일 함수 $\Phi(t)$ 와 웨이블릿 함수 $\Psi(t)$ 가 존재하며 식 (2)와 같다.

$$\begin{aligned} f(t) &= \sum_{j,k} c_j(k) \Phi_{j,k}(t) + \sum_{j,k} d_j(k) \Psi_{j,k}(t) \\ &= \sum_{j,k} c_j(k) 2^{j/2} \Phi(2^j t - k) + \sum_{j,k} d_j(k) 2^{j/2} \Psi(2^j t - k) \end{aligned} \quad (2)$$

2 차원 이산 웨이블릿 변환은 행방향과 열방향에 대한 1차원 웨이블릿 변환을 수행해야 한다. 따라서 j 레벨을 수행하는 2D DWT 필터의 분해단과 합성단은 그림 1과 같이 구현된다. $L(z)$ 와 $H(z)$ 는 각각 스케일링 함수와 웨이블릿 함수에 일치하는 신장 계수로 저역필터와 고역필터를 의미한다. 이미지를 분해하기 위해서 행방향 데이터 LL_{j-1} 을 저역 필터와 고역 필터에 각각 통과시킨다. 이 때 LL_{j-1} 은 이전 레벨의 저역필터를 연속으로 통과시킨 결과 값이다. 행방향에 대하여 1차원 웨이블릿 변환한 저역 필터 결과 값과 고역 필터 결과 값은 다운 샘플링 과정을 거친 후에, 다시 열방향 웨이블릿 변환을 위한 저역 필터와 고역 필터를 통과시키고 다운 샘플링 함으로써 4개의 주파수대역 LL_j (low, low), LH_j (low, high), HL_j (high, low), HH_j (high, high)을 얻는다.

본 연구와 유사한 Lattice 구조를 갖는 2D DWT 구조^[12]는 그림 2와 같은 2차원 중첩 스케줄링을 적용하였다. 첫 번째 두 개의 H-DWT 필터는 인접한 두 행을 입력 신호로 하여 행방향으로 1D DWT를 계산하고 그 다음 두 개의 Lattice 필터 V-DWT는 행방향 출력 값

을 가지고 바로 열방향 1D DWT를 수행하여 한 레벨의 2D DWT를 수행한다. 이 구조의 특징은 하드웨어 효율과 성능 향상을 최대화하기 위해, 행방향 필터 H-DWT에 대해 동시에 두 개의 입력을 처리한다는 것과 행방향 필터의 출력이 바로 열방향 필터 V-DWT의 입력으로 들어간다는 것이다. V-DWT 필터의 출력 중 LL_j 는 다음 상위 레벨의 분해를 위한 입력으로 다시 H-DWTU와 H-DWTL 필터의 입력으로 피드백 된다.

그림 2에서 행 t_r 과 열 t_c 는 각각 행방향과 열방향의 연산 시간을 의미하는 것으로 열방향 시간에 따라 수행되는 레벨의 스케줄이 다르다. 즉, t_c 가 0일 때는 $j = 1, 2, 1, 3, 1, 2, 1, \dots$ 이고 t_r 이 1 일 때는 $j = 1, \times, 1, \times, \dots$ 순으로 레벨이 수행된다. 점선의 블록에 포함된 흰 도형은 행방향 필터 H-DWT의 출력 값을 의미하고 회색 도형은 흰 도형을 입력으로 하는 열방향 필터 V-DWT의 출력 값을 의미한다. 이 구조는 $N \times N$ 이미지를 처리하는데 $N^2/2$ 사이클이 소요되므로 비교적 고속의 동작이 가능하지만 PE(processing element)가 직렬로 연결되어 있어 긴 임계경로를 갖는다는 점과 중첩 스케줄링을 이용하기 때문에 약 66%의 하드웨어 효율을 갖는다는 단점이 있다.

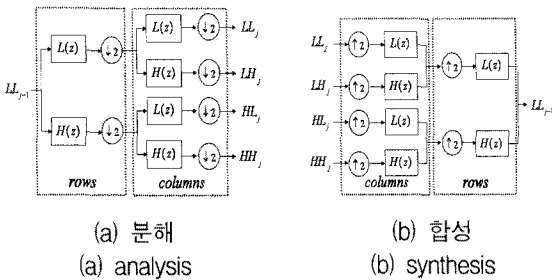


그림 1. 2D DWT 분해와 합성
Fig. 1. Analysis and synthesis of 2D DWT.

III. 제안된 2D DWT 필터 구조

본 논문에서는 하드웨어 효율이 100%가 되는 2차원 DWT를 구현하기 위해서 새로운 DFC 및 DCU 구조를 제안한다.

J 레벨의 해상도를 갖는 2차원 DWT 필터는 $N \times L$ 이미지를 갖는 2차원 데이터를 $2^J \times 2^J$ 블록 단위로 나누어 지그재그 스캔하여 필터에 입력한다. 모두 $X \times Y$ 개의 블록으로 나누어지며 이 때, $X = \lceil L/2^J \rceil$ 이고 $Y = \lceil N/2^J \rceil$ 이다. 블록 내의 데이터도 지그재그 스캔한다. 이때 인접한 4개의 데이터를 동시에 필터에 입력하여 1레벨부터 J 레벨까지 순차적으로 수행함으로써 하드웨어 효율을 떨어뜨리는 휴면 동작 구간을 제거할 수 있다.

외부에서 입력되는 이미지의 값들을 $u^{x,y}(r,c)$ 라고 정의할 때 x 와 y 는 각각 이미지에 대한 블록의 위치를 나타내는 행과 열의 값이고 r 과 c 는 해당 블록의 r 행 c 열에 위치한 외부 입력값을 의미한다. 이 때, x 와 y 는

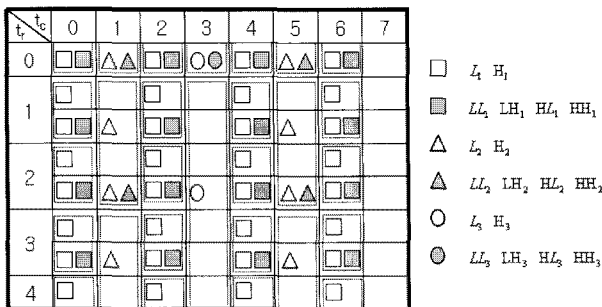


그림 2. 8x8 크기의 영상에 대한 3레벨을 갖는 2D DWT의 스케줄링
Fig. 2. Scheduling of 2D DWT with 3 level in a 8x8 block.

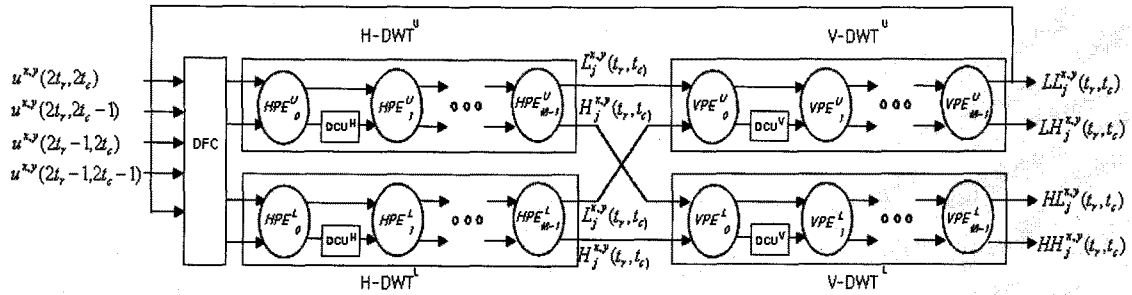


그림 3. 2M의 필터길이를 갖는 제안된 2D DWT Lattice 구조
 Fig. 3. Proposed 2D DWT lattice architecture with 2M filter length.

각각 $x = 1, 2, \dots, X$ 와 $y = 1, 2, \dots, Y$ 의 범위를 가지며 r 과 c 는 0부터 $2^J - 1$ 까지의 값을 갖는다. 또한, j 레벨까지 2차원 DWT를 수행한 결과는 $LL_j^{x,y}(r, c)$, $LH_j^{x,y}(r, c)$, $HL_j^{x,y}(r, c)$ 및 $HH_j^{x,y}(r, c)$ 와 같이 정의한다.

t_r 과 t_c 는 j 레벨에서 수행되는 입력값의 스캔 순서와 출력의 생성 순서를 표시하기 위해 사용되었다. 1레벨을 수행하기 위해서 외부입력 $u^{x,y}(2t_r, 2t_c)$, $u^{x,y}(2t_r, 2t_c - 1)$, $u^{x,y}(2t_r - 1, 2t_c)$ 및 $u^{x,y}(2t_r - 1, 2t_c - 1)$ 을 동시에 필터에 입력하여 $2^{2(J-1)}$ 사이클 동안 $2^{2(J-1)}$ 개의 $LL_1^{x,y}(t_r, t_c)$, $LH_1^{x,y}(t_r, t_c)$, $HL_1^{x,y}(t_r, t_c)$ 및 $HH_1^{x,y}(t_r, t_c)$ 의 1레벨 수행 결과를 얻는다. 이때 t_r 과 t_c 는 0부터 $2^{J-1} - 1$ 까지의 값을 갖는다. 또한, $2 \leq j \leq J$ 의 값을 갖는 임의의 j 레벨을 수행하기 위해서 $LL_{j-1}^{x,y}(2t_r, 2t_c)$, $LL_{j-1}^{x,y}(2t_r, 2t_c - 1)$, $LL_{j-1}^{x,y}(2t_r - 1, 2t_c)$ 과 $LL_{j-1}^{x,y}(2t_r - 1, 2t_c - 1)$ 을 동시에 필터에 입력하여 $2^{2(J-j)}$ 사이클 동안 $2^{2(J-j)}$ 개의 $LL_j^{x,y}(t_r, t_c)$, $LH_j^{x,y}(t_r, t_c)$, $HL_j^{x,y}(t_r, t_c)$, $HH_j^{x,y}(t_r, t_c)$ 의 j 레벨 수행 결과를 얻는다.

t_r 또는 t_c 가 0인 경우 입력값 중 $2t_r - 1$ 또는 $2t_c - 1$ 의 값이 -1이 되는 경우가 발생하는데 t_c 가 0인 경우는 현재 블록의 왼쪽에 위치한 인접 블록의 $2^{J-(j-1)}$ 번째 값을 의미하며 t_r 이 0인 경우는 현재 블록의 상위에 위치한 인접 블록의 $2^{J-(j-1)}$ 번째 값을 의미한다.

그림 3은 4개의 1D DWT 필터와 DFC 모듈 및 DCU 모듈로 구성된 제안된 2D DWT 구조이다. 제안된 구조는 행방향과 열방향 1D DWT를 동시에 처리하기 위해

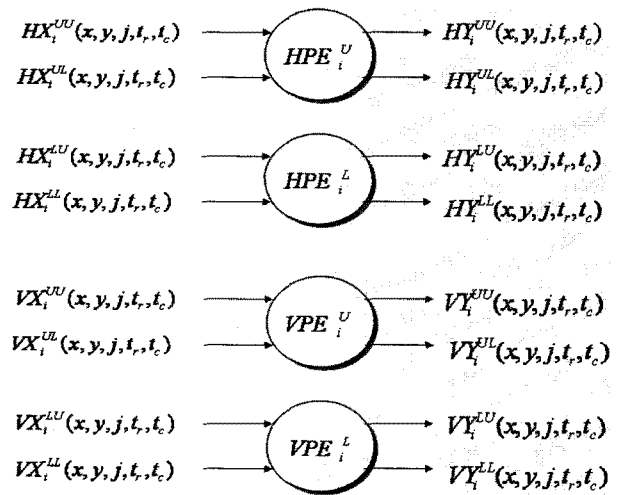


그림 4. Lattice 필터 뱅크의 입출력 데이터 표기법
 Fig. 4. I/O data notation of lattice filter bank.

서 2M의 필터 길이를 갖는 4개의 1D DWT Lattice 필터로 구성되어 있다. 입출력 구조를 그림 4와 같이 정의 하였을 경우 제안한 스케줄링을 수식으로 표현할 수 있으며 $VY_{M-1}^{UU}(x, j, j, t_r, t_c) = LL_j^{x,y}(t_r, t_c)$ 의 관계를 갖는다. 이 때 각각의 PE를 $dPE_i^{p_0 p_1}(t_r, t_c)$ 로 표기하는데 아래 첨자 i 는 하나의 1D DWT 필터의 i 번째 PE를 의미하고, 위 첨자 p_0 는 Upper 필터인지 Lower 필터인지를 나타내고 p_1 은 PE에서 위쪽 입력력인지 아래쪽 입력력인지를 구별하는 인자이다. d 는 행방향 필터의 PE이면 H(horizontal), 열방향 필터의 PE이면 V(vertical)를 갖는 인자이다.

식 3에서 식 6까지는 H-DWT 필터의 첫 번째 Lattice 필터 뱅크에 대한 입력 관계식으로 DFC 설계에 사용되며 식 7에서 식 10까지는 H-DWT 필터의 첫 번째 Lattice 필터 뱅크를 제외한 필터 뱅크의 입력 관계식으로 DCU^H 설계에 사용된다. 즉 i 의 범위는

$$HX_0^{UU}(x, y, j, t_r, t_c) = \begin{cases} u^{x,y}(2t_r, 2t_c) & ;j = 1, t_r = 0, \dots, 2^{J-1} - 1, t_c = 0, \dots, 2^{J-1} - 1 \\ LL_{j-1}^{x,y}(2t_r, 2t_c) & ;j \geq 2, t_r = 0, \dots, 2^{J-1} - 1, t_c = 0, \dots, 2^{J-1} - 1 \end{cases} \quad (3)$$

$$HX_0^{UL}(x, y, j, t_r, t_c) = \begin{cases} u^{x,y}(2t_r, 2t_c - 1) & ;j = 1, t_r = 0, \dots, 2^{J-1} - 1, t_c = 1, \dots, 2^{J-1} - 1 \\ u^{x,y-1}(2t_r, 2^J - 1) & ;j = 1, t_r = 0, \dots, 2^{J-1} - 1, t_c = 0 \\ LL_{j-1}^{x,y}(2t_r, 2t_c - 1) & ;j \geq 2, t_r = 0, \dots, 2^{J-1} - 1, t_c = 1, \dots, 2^{J-1} - 1 \\ LL_{j-1}^{x,y-1}(2t_r, 2^{J-(j-1)} - 1) & ;j \geq 2, t_r = 0, \dots, 2^{J-1} - 1, t_c = 0 \end{cases} \quad (4)$$

$$HX_0^{LU}(x, y, j, t_r, t_c) = \begin{cases} u^{x,y}(2t_r - 1, 2t_c) & ;j = 1, t_r = 1, \dots, 2^{J-1} - 1, t_c = 0, \dots, 2^{J-1} - 1 \\ u^{x,y-1}(2^J - 1, 2t_c) & ;j = 1, t_r = 0, t_c = 0, \dots, 2^{J-1} - 1 \\ LL_{j-1}^{x,y}(2t_r - 1, 2t_c) & ;j \geq 2, t_r = 1, \dots, 2^{J-1} - 1, t_c = 0, \dots, 2^{J-1} - 1 \\ LL_{j-1}^{x,y-1}(2^{J-(j-1)} - 1, 2t_c) & ;j \geq 2, t_r = 0, t_c = 0, \dots, 2^{J-1} - 1 \end{cases} \quad (5)$$

$$HX_0^{LL} = \begin{cases} u^{x-1,y-1}(2^J - 1, 2^J - 1) & ;j = 1, t_r = 0, t_c = 0 \\ u^{x-1,y}(2^J - 1, 2t_c - 1) & ;j = 1, t_r = 0, t_c = 1, \dots, 2^{J-1} - 1 \\ u^{x,y-1}(2t_r - 1, 2^J - 1) & ;j = 1, t_r = 1, \dots, 2^{J-1} - 1, t_c = 0 \\ u^{x,y}(2t_r - 1, 2t_c - 1) & ;j = 1, t_r = 1, \dots, 2^{J-1} - 1, t_c = 1, \dots, 2^{J-1} - 1 \\ LL_{j-1}^{x-1,y-1}(2^{J-(j-1)} - 1, 2^{J-(j-1)} - 1) & ;j \geq 2, t_r = 0, t_c = 0 \\ LL_{j-1}^{x-1,y}(2^{J-(j-1)} - 1, 2t_c - 1) & ;j \geq 2, t_r = 0, t_c = 1, \dots, 2^{J-1} - 1 \\ LL_{j-1}^{x,y-1}(2t_r - 1, 2^{J-(j-1)} - 1) & ;j \geq 2, t_r = 1, \dots, 2^{J-1} - 1, t_c = 0 \\ LL_{j-1}^{x,y}(2t_r - 1, 2t_c - 1) & ;j \geq 2, t_r = 1, \dots, 2^{J-1} - 1, t_c = 1, \dots, 2^{J-1} - 1 \end{cases} \quad (6)$$

$$HX_i^{UU}(x, y, j, t_r, t_c) = HY_{i-1}^{UU}(x, y, j, t_r, t_c) \quad (7)$$

$$HX_i^{UL}(x, y, j, t_r, t_c) = \begin{cases} HY_{i-1}^{UU}(x, y, j, t_r, t_c - 1) & ;t_c = 1, \dots, 2^{J-j} - 1 \\ HY_{i-1}^{UL}(x, y - 1, j, t_r, 2^{J-j} - 1) & ;t_c = 0 \end{cases} \quad (8)$$

$$HX_i^{LU}(x, y, j, t_r, t_c) = HY_{i-1}^{LU}(x, y, j, t_r, t_c) \quad (9)$$

$$HX_i^{LL}(x, y, j, t_r, t_c) = \begin{cases} HY_{i-1}^{LL}(x, y, j, t_r, t_c - 1) & ;t_c = 1, \dots, 2^{J-j} - 1 \\ HY_{i-1}^{LL}(x, y - 1, j, t_r, 2^{J-j} - 1) & ;t_c = 0 \end{cases} \quad (10)$$

$$VX_i^{UU}(x, y, j, t_r, t_c) = VY_{i-1}^{UU}(x, y, j, t_r, t_c) \quad (11)$$

$$VX_i^{UL}(x, y, j, t_r, t_c) = \begin{cases} VY_{i-1}^{UL}(x, y, j, t_r - 1, t_c) & ;t_r = 1, \dots, 2^{J-j} - 1 \\ VY_{i-1}^{UL}(x - 1, y, j, 2^{J-j} - 1, t_c) & ;t_r = 0 \end{cases} \quad (12)$$

$$VX_i^{LU}(x, y, j, t_r, t_c) = VY_{i-1}^{LU}(x, y, j, t_r, t_c) \quad (13)$$

$$VX_i^{LL}(x, y, j, t_r, t_c) = \begin{cases} VY_{i-1}^{LL}(x, y, j, t_r - 1, t_c) & ;t_r = 1, \dots, 2^{J-j} - 1 \\ VY_{i-1}^{LL}(x - 1, y, j, 2^{J-j} - 1, t_c) & ;t_r = 0 \end{cases} \quad (14)$$

1 ≤ i ≤ M-1이다.

식 11부터 식 14까지는 V-DWT 필터의 첫 번째 Lattice 필터 बैं크를 제외한 필터 बैं크의 입력 관계식으로 DCU^V 설계에 사용된다.

3.1 DFC 구조 설계

제안된 스케줄링에 맞게 외부 입력 데이터와 이전 레벨의 분해 결과인 $VY_{M-1}^{UU}(j, t_r, t_c)$ 를 2D DWT 필터에 입력하기 위해서 새로운 DFC 구조를 제안한다.

현재 x행 y열 블록에 대한 2D DWT를 수행한다고

가정하자. 식 3에 의해서 외부 입력 $u(2t_r, 2t_c)$ 와 레지스터 $UU_{k,l}$ 가 멀티플렉서를 통해 $HX_0^{UU}(j, t_r, t_c)$ 로 출력된다. 레지스터 $UU_{k,l}$ 에 저장되는 값들은 j 레벨의 수행 결과 값이 동일 블록의 $j+1$ 레벨 수행 시에 바로 사용되므로 레지스터를 공유할 수 있다. 따라서 레지스터 $UU_{k,l}$ 는 모두 $2^{2(J-2)}$ 개의 레지스터로 구성된다.

식 4에 의해서 $HX_0^{UL}(j, t_r, t_c)$ 에 외부입력 $u(2t_r, 2t_c - 1)$ 과 레지스터 $UL_{k,l}^j$ 가 멀티플렉서에 의해 연결된다. $l \neq 0$ 인 경우 동일 블록에서 생성된 j 레벨의 결과 값이 $j+1$ 레벨 수행 시 바로 사용되므로 레지스터를 공유할 수 있지만 $l = 0$ 인 경우 x 행 $y-1$ 열의 블록에 의해 생성된 결과 값을 출력해야하므로 현재 블록의 수행 결과를 저장하는 $UL_{k,l}^j$ 의 이전 블록의 결과 값을 저장하는 $UL_{k,0}^{j-1}$ 레지스터가 추가되다. 따라서 레지스터 $UL_{k,l}^j$ 는 모두 $2^{J-2}(2^{J-2}-1) + 2^J - 2$ 개의 레지스터로 구성된다.

식 5에 의해서 $HX_0^{LU}(j, t_r, t_c)$ 에 외부입력 $u(2t_r - 1, t_c)$ 와 레지스터 $LU_{k,l}^j$ 가 멀티플렉서를 통해서 출력된다. $k \neq 0$ 인 경우 역시 레지스터를 공유할 수 있으며 $k = 0$ 인 경우는 $x-1$ 행 y 열에서 수행된 결과 값을 2D DWT에 전달해야 하므로 Y 개의 블록에서 발생하는 각 레벨의 데이터를 저장하기 위한 레지스터가 필요하다.

따라서 $LU_{k,l}^j$ 는 모두 $(2^{J-2}-1)2^{J-2} + Y(2^{J-1}-1)$ 개의 레지스터로 구성된다.

식 6에 의해 $HX_0^{LL}(j, t_r, t_c)$ 는 멀티플렉서를 통해서 외부입력 $u(2t_r - 1, 2t_c - 1)$ 과 레지스터 $LL_{k,l}^j$ 로부터 받은 데이터를 2D DWT에 출력한다. $k \neq 0$ 이고 $l = 0$ 인 경우 x 행 $y+1$ 열의 블록에 대한 2D DWT 수행 시 사용하기 위해서 레지스터 $LL_{k,0}^j$ 에 저장되고 $k = 0$ 이고 $l \neq 0$ 인 경우 $x+1$ 행 y 열 블록의 2D DWT 수행 시 사용하기 위해서 $LL_{0,l}^j$ 에 저장된다. $k = 0$ 이고 $l = 0$ 인 경우에는 $x+1$ 행 $y+1$ 열의 블록에 대한 2D DWT 수행 시 사용하기 위해서 레지스터 $LL_{0,0}^j$ 에 저장되며 $k \neq 0$ 이고 $l \neq 0$ 인 경우에는 레지스터를 공유할 수 있다. 따라서 레지스터 $LL_{k,l}^j$ 는 모두 $2^{2(J-2)} + Y(2^{J-1}-1)$ 개로 구성된다. 따라서, 전체 DFC

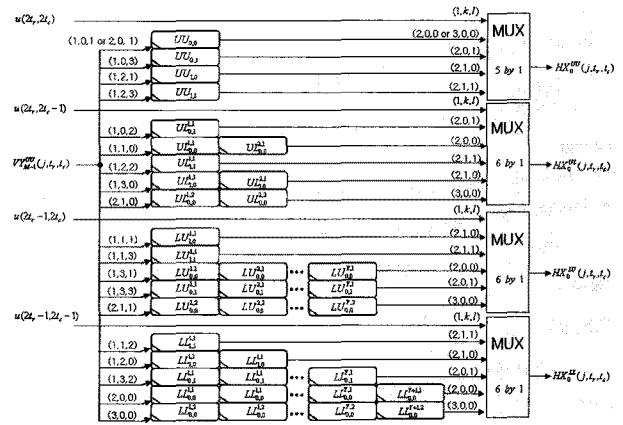


그림 5. 3 레벨 DFC 구조
Fig. 5. DFC structure with 3 level.

는 4개의 멀티플렉서와 $2^{2(J-1)} - 2^{J-1} + (Y+1)(2^J - 2)$ 개의 레지스터로 구성된다.

그림 5는 $J = 3$ 인 경우의 DFC 구조이다. 이 때 k 와 l 은 레지스터의 인덱스를 표시하며 $k = 0, \dots, 2^{J-j} - 1$ 와 $l = 0, \dots, 2^{J-j} - 1$ 의 범위를 갖는다.

3.2 DCU 구조 설계

Lattice 필터 뱅크 간 지연을 제어해주는 DCU는 PE들 사이에 위치하며, 해상도 레벨 j 에 따라 지연 z^{-2^j} 를 제어하는 모듈이다. H-DWT^U 필터가 j 레벨을 k, l 시간에 수행할 때 레지스터 $R_{j,l}$ 로부터 $HX_{i+1}^{UL}(j, t_r, t_c)$ 로 값을 출력한다. $k \neq 0$ 인 경우 레지스터 $R_{j,l-1}$ 에 입력 값 $HY_i^{UL}(j, t_r, t_c)$ 를 저장하고 $k = 0$ 인 경우 인접한 블록으로부터 발생한 $HY_i^{UL}(x, y-1, j, t_r, 2^{J-j}-1)$ 값을 출력하기 위해서 레지스터 $R_{j-1, 2^{J-j}-1}$ 로부터 출력된다. 파라미터 x 와 y 는 그림에서 생략했으며 H-DWT^L의 DCU^H도 동일하게 동작한다. 식 8에 의해서 행방향 Lattice 필터 H-DWT^U를 위한 $J = 3$ 의 경우 DCU^H의 구조를 그림 6과 같이 구현할 수 있다.

식 10에 의해서 열방향 Lattice 필터 V-DWT^U를 위

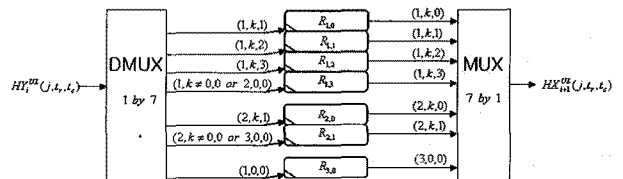


그림 6. 3레벨 DCU^H의 구조
Fig. 6. DCU^H structure with 3 level.

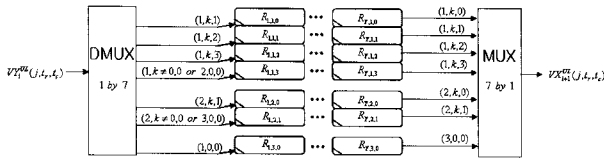


그림 7. 3 레벨 DCU^V의 구조
Fig. 7. DCU^V structure with 3 level.

한 $J=3$ 경우의 DCU^V 구조를 그림 7과 같이 구현할 수 있다. V-DWT^U 필터가 j 레벨을 k, l 시간에 수행할 때 레지스터 $R_{y,j,l}$ 의 값을 $VX_{i+1}^{UL}(j, t_r, t_c)$ 에 출력한다. 만약 $k \neq 0$ 인 경우 레지스터 $R_{y,j,l-1}$ 에 입력 값 $VY_i^{UL}(j, t_r, t_c)$ 을 저장하고 $k=0$ 인 경우 인접한 상위 블록에서 발생한 $VY_i^{UL}(x-1, y, j, 2^{J-j}-1, t_c)$ 의 값을 저장하기 위해서 레지스터 $R_{y,j-1,2^{J-j}-1}$ 에 저장한다.

3.3 제안된 2D DWT의 파이프라인 구조

본 논문에서 제안한 스케줄링과 구조를 통해서 100%의 하드웨어 효율을 갖는 2D DWT를 구현할 수 있었지만 직렬로 연결된 1D DWT 필터로 인해 긴 임계 경로를 갖게 된다. 그 결과 동작 속도가 느려지는 단점을 가지고 있으며 임계 경로를 줄이기 위한 개선된 구조가 요구된다.

본 논문에서는 임계 경로를 줄이기 위해서 파이프라인을 적용하였으며 제안하는 2D DWT의 스케줄링의 경우 $LL_{j-1}^{x,y}(0,0)$ 을 계산하기 위해서 요구되는 $LL_{j-1}^{x,y}(0,0)$ 값이 4 사이클 전에 발생한다. 따라서 1D DWT의 필터 길이를 M이라고 했을 때 PE 마다 P_d 만큼의 파이프라인을 적용하면 $MP_d > 4$ 인 경우에 100%의 하드웨어 효율을 유지할 수 없다. 따라서 본 논문에서 제안하는 2D DWT 구조는 100%의 하드웨어 효율을 유지하기 위해서 그림 8과 같이 파이프라인을 최대 4단까지 삽입할 수 있다. P_d 가 0인 경우는 파이프라인을 적용하지 않은 것과 같고 1 이상인 경우 2D DWT

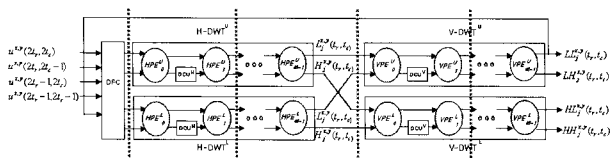


그림 8. 2D DWT의 파이프라인 구조($P_d=4$)
Fig. 8. Pipelined architecture of 2D DWT.

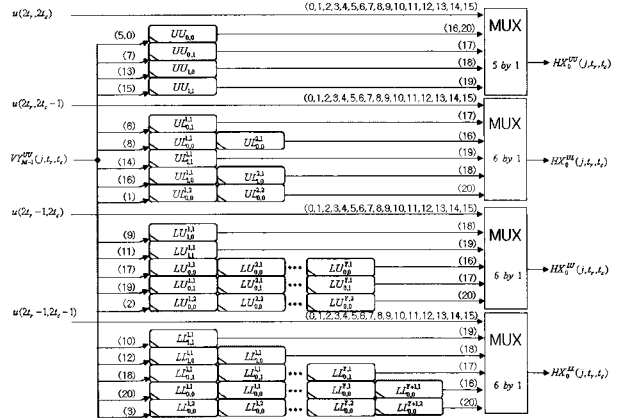


그림 9. 3레벨 DFC 구조($P_d=4$)
Fig. 9. DFC structure with 3 level ($P_d=4$).

는 P_d 사이클의 Latency를 갖는다. 이 때, P_d 사이클만큼의 Latency만 발생한 것이므로 DCU의 구조와 스케줄링은 변하지 않지만 파이프라인 적용에 의한 2D DWT 입출력 스케줄링의 변화로 인해 DFC 구조를 변경해야 한다. 그림 9는 $P_d=4$ 이고 $J=3$ 인 경우의 스케줄링을 나타낸다.

IV. 설계 및 성능 분석

제안된 구조는 VerilogHDL를 이용하여 모델링 되었고, 시뮬레이션을 통해 검증되었으며 동부아남 0.18 μ m 표준셀을 사용하여 합성 및 검증되었다. [12]에서 제안된 필터계수의 양자화와 비트 폭을 그대로 사용하므로 PSNR 성능은 동일하다.

기존의 구조와 본 논문에서 제안하는 구조의 성능을 표 1에서 비교하였다. 동일한 비교를 위해서 영상 크기는 $N \times N$ 으로 하였다. 1D DWT의 필터 길이는 M이고 J레벨까지 분해하는 것으로 가정하였다. 직접구현 방식^[7]은 한 개의 1D DWT를 행방향 분해와 열방향 분해를 위해서 반복 사용하므로 최소의 하드웨어를 사용하지만 매우 성능이 낮고, 병렬 처리방법^[5, 8, 10]은 직접구현 방식에 비해서 우수한 성능을 보여주지만 다수의 필터 뱅크를 사용하므로 VLSI 설계 시 면적 복잡도가 증가한다. Semi-recursive 방법^[12]은 짧은 워드 길이와 최적화된 데이터 버스 효율을 높이기 위해 네 개의 분리된 필터 뱅크를 사용하기 때문에 하드웨어가 많이 사용된다. Systolic, Semi-systolic과 RAM 기반의 3가지 형태로 구현된 재귀 DWT 구조^[11]는 우수한 성능과 하드웨어 효율을 보이지만, 복잡한 라우팅 네트워크와 스케줄링

표 1. 2D DWT 구조의 성능 비교

Table 1. Comparison with various 2D DWT architectures.

Architectures	Mult	Adder	Storage	Period	Util.
Direct[7]	M	M	N^2	$4N^2$	100%
Parallel[10]	3M	3(M-1)	2N(M-1)	N^2	low
Systolic-parallel[5]	4M	4M	2N(M+2)	N^2-N	high
Semi-recursive[8]	4M	4(M-1)	$2N^2$	$4N^2/3$	low
2D Lattice[12]	4M	4M	M(N+J)	$N^2/2$	high
proposed	4M	4M	2MN-3N	$N^2(1-2^{-2J})/3$	100%

(N^2 : Image size, M : Filter length, J : Decomposition level)

표 2. 파이프라인 구조 성능비교

Table 2. Performance of the pipelined architectures.

Pipeline P_d	Area (gates)	frequency (MHz)
0	726k	16.6
4	932k	100

이 요구된다.

본 논문에서 제안하는 구조는 필터 길이 M 을 갖는 1D DWT를 사용하는 경우 $2M$ 개의 Lattice 필터뱅크로 구성되고 필터뱅크마다 2개의 곱셈기와 덧셈기로 구성된다. 따라서 $4M$ 개의 곱셈기와 $4M$ 개의 덧셈기를 사용한다. 메모리 복잡도는 약 $2MN-3N$ 으로 [12]에 비해서 증가하였지만 하드웨어 효율에서 100%를 나타낸다. 100%의 하드웨어 효율을 갖는 직접구현 방법^[7]과 비교했을 때 수행시간이 짧고 요구하는 레지스터의 수가 적다. 따라서 제안된 2D DWT의 새로운 스케줄링을 적용함으로써 100%의 효율을 갖는 고속의 2D DWT 구조를 구현할 수 있다.

하지만 표 2에서 알 수 있듯이 파이프라인을 적용하지 않은 경우($P_d = 0$) 긴 임계경로를 가지고 있어서 동작 주파수가 매우 낮다. 이를 극복하기 위해서 파이프라인을 적용하는 경우 스케줄링의 특성상 최대 4단까지 파이프라인을 적용이 가능하며 이 때 약 100MHz의 동작 주파수를 보여준다.

V. 결 론

본 논문에서는 영상 정보의 효과적인 압축을 위해서 두 채널 QMF Lattice 필터 기반의 2차원 이산 웨이블릿 변환 구조에 대해서 연구하였다.

제안된 구조는 100%의 하드웨어 효율을 얻기 위해서 인접한 4개의 입력 값을 2D DWT 필터에 동시에 입력함으로써 휴면 동작 구간을 제거하였으며 4개의 1D DWT를 사용하므로 1 사이클에 행방향 분해와 열방향 분해를 동시에 처리할 수 있다. 본 연구에서는 $2^J \times 2^J$ 크기의 블록으로 $N \times N$ 이미지를 나누고 블록 단위로 1레벨부터 J 레벨까지 순차적으로 처리함으로써 $O(MN)$ 의 메모리 복잡도를 갖는다. 제안하는 2D DWT의 처리 방법을 지원하기 위해 최소한의 레지스터를 사용하는 DFC와 DCU 구조를 설계하였다. 긴 임계경로로 인해 동작 속도가 느리다는 단점을 해결하기 위해 파이프라인 구조를 제안하였다.

감사의 글

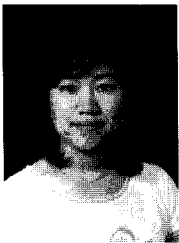
저자들은 본 연구를 위하여 설계 환경을 제공하여준 IDEC에 감사드린다.

참고 문헌

- [1] Iain E. G. Richardson, "H.264 and MPEG-4 Video Compression," John Wiley & Sons, 2003.
- [2] 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.
- [3] T. Ryan, L. Sanders, H. Fisher, and A. Iverson, "Image compression by texture modeling in the wavelet domain," IEEE Trans. Image Process., vol.5, pp.26-36, 1996.
- [4] A. Skodras, C. Christopoulos, and T. Ebrahimi, "JPEG2000: The Upcoming Still Image Compression Standard," Proceedings of 11th conf. of Pattern Recognition, pp.359-366, 2000.
- [5] M. Vishwanath, R. Owens, and M. Irwin, "VLSI architectures for the discrete wavelet transform," IEEE Trans. CAS-II, vol.42, no.5, pp.305-316, 1995.
- [6] T. Denk and K. Pahari, "VLSI architectures for Lattice structure based orthogonal discrete wavelet transform," IEEE Trans. CAS-II, vol.44, no.2, pp.129-132, 1997.
- [7] A. Lewis and G. Knowles, "VLSI architecture for 2D Daubechies wavelet transform without multipliers," Electron. Lett., vol.27, no.2, pp.171-173, 1997.

- [8] S. Paek and L. Kim, "2D DWT VLSI architecture for wavelet image processing," *Electron.Lett.*, vol.34, no.6, pp.537-538, 1998.
- [9] J. Kim, Y. Lee, T. Isshiki, and H. Kunieda, "Scalable VLSI architecture for lattice structure-based discrete wavelet transform," *IEEE Trans. CAS-II*, vol.45, no.8, pp.1031-1043, 1998.
- [10] C. Yu and S. Chen, "Design of an efficient VLSI architecture for 2D discrete wavelet transforms," *IEEE Trans. Consumer Elect.*, vol.45, no.1, pp.135-140, 1999.
- [11] F. Marino, "A Double-Face Bit-serial Architecture for the 1D Discrete Wavelet Transform," *IEEE Trans. on Circuits & Systems II-Analog & Digital Signal Preocessing*, vol.47, no.1, pp.65-71, 2000.
- [12] T. Park and S. Jung, "High Speed Lattice Based VLSI Architecture of 2D Discrete Wavelet Transform for Real-Time video Signal Processing," *IEEE Trans. on Consumer Electronics*, vol.48, no.4, pp.1026-1032, 2002.
- [13] P. Vaidyanathan, "Multirate systems and filter banks", Prentice-Hall, 1993.

 저 자 소 개



김 주 영(정회원)
 2005년 가톨릭대학교
 정보통신공학과 졸업.
 2005년~2007년 가톨릭대학교
 컴퓨터공학과 공학석사
 2007년~현재 (주)넥서스칩스
 주임연구원

<주관심분야 : VLSI 설계, 영상처리 및 암호시스
 템 등>



박 태 근(정회원)-교신저자
 1985년 연세대학교 전자공학과
 졸업.
 1988년 Syracuse Univ.
 Computer 공학석사 졸업.
 1993년 Syracuse Univ.
 Computer 공학박사 졸업.

1991년~1993년 Coherent Research Inc. USA
 VLSI 엔지니어.
 1994년~1998년 현대전자 System IC 연구소
 책임연구원
 1998년~현재 가톨릭대학교 정보통신전자공학부
 교수

<주관심분야 : VLSI 설계, CAD, 컴퓨터 구조>