효과적인 DWT필터의 설계

이동훈, 최덕영, 손승일
한신대학교 정보통신학과

Design of an Efficient DWT Filter

Dong Hun Lee, Dug Young Choi Seung Il Sonh
Dept. of Information and Communication Hanshin University
E-mail : asickorea@hs.ac.kr

요 약

현대에 있어서 영상정보는 아주 큰 비중을 차지하고 있다. 따라서 이러한 영상정보를 얼마나 빨리 그리고 많이 압축 시킬 수 있는가가 핵심적인 관건이다. 본 논문에서는 공간적 압축 방식의 핵심인 DCT와 비교하여 블록킹 효과(Blocking Effect)과 없고, 우수한 압축 성능을 갖는 DWT(Discrete Wavelet Transform)알고리즘을 적용한 2차원 이산 웨이브렛 변환 필터를 설계하였다. 본 논문에서 구현한 DWT 필터는 FIR필터 방법으로 설계하였으며, Daubenchies-4 Tap을 이용하였고, 파이프라인 연산으로 승산기, 가산기를 병렬로 처리하여 고속연산을 수행하였다. 뿐만 아니라 메모리 맵핑 과정과 메모리 컨트롤 어드레스 발생기를 사용하여 메모리와 연산량을 최소화 하여 칩사이즈를 줄여 설계하였다.

1. 서 론

현대에 있어서 영상정보는 아주 큰 비중을 차지하고 있다. 따라서 이러한 영상정보를 얼마나 빨리 그리고 많이 압축 시킬 수 있는가가 핵심적인 관건이다. 이에 많은 부분에서 영상을 압축 시키는 방법들이 연구 되어 지고 있다. 영상을 압축 하는 방법에는 크게 공간적 압축 방법과 시간적 압축 방법이 사용되는데 이 중에서 공간적 압축방식에는 DCT변환을 많이 사용하고 있다. DCT는 기존의 영상 압축 표준안인 MPEG, JPEG에서 많이 사용하는데 8*8 블록 단위로 처리하기 때문에 제한된 압축 범위를 갖게 되어 블록킹 현상이 발생한다. 반면 JPEG2000에서 채택한 웨이브렛(wavelet)기술은 블록킹 현상을 없으며 신축성이 있고 정확하게 제어한다. 그리고 웨이브렛은 영상에 대하여 정확한 비트율이나 화질조절이 가능한데 이러한 과정들은 하드웨어 설계에 있어서 효율적이다. 따라서 DCT와 웨이브렛을 비교할 때 웨이브렛은 비용과 화질면에서 월등히 뛰어나며 높은 압축율에서 저손실을 갖는다.

본 논문에서는 효과적인 영상압축을 위한 웨이브렛 필터를 하드웨어로 구현하였다. 입력되는 영상 데이터는 효과적인 처리를 위해 상위에서 분할하여 입력하도록 처리하였으며 Daubenchies-4 Tap 필터 뱅크의 사용과 블록킹 현상을 없애기 위한 이전 블록의 일부인 컨트롤 데이터를 같이 입력되어 기존 피라미드 데이터의 의존성을 따르지 않는 데이터 처리를 통해 고속의 처리를 수행하였다. 또한 각 단의 필터는 병렬구조로 이루어져 동일 클럭에서 하이패스와 로우패스를 동시해 수행함으로써 속도를 향상시킬 수 있으며 뿐만 아니라 QMF의 특성을 이용하여 DWT계산에 필요한 승산기 수를 절반으로 줄임으로써 하드웨어의 크기를 줄일 수 있다. 제안된 웨이블렛 필터는 하드웨어 기술언어인 VHDL로 코딩 하였으며 ModelSim 5.6으로 시뮬레이션 하여 파형을 분석하여 올바른 결과를 도출하였다.

2. DWT 개요

2-1. 웨이브렛 이론

일반적으로 신호변환이라 함은 신호가 가지고 있는 특징 정보를 추출하여 신호분석(signal analysis)을 용이하게 하는데 목적이 있다. 여기에는 다양한 신호변환 방법들이 있는데 그 중 널리 사용되고 있는 변환의 방법 중 하나가 푸리에 변환

(Fourier transform)이다. 하지만, 전제조건으로 입력신호의 특성이 시간축에서 변화지 않는 정상 신호라는 제약이 있기 때문에 비정상 신호의 분석에는 적합하지 않다. 웨이브렛 변환은 이런 단점이 없기 때문에 정상 신호뿐만 아니라 비정상 신호의 분석도 가능하다. 웨이브렛의 변환의 기본 개념은 임의의 신호 f(t)를 시간 주파수적으로 지역성을 갖는 웨이브렛 기저함수들의 중첩된 형태로 표현하는 것으로 식(1)에 임의의 신호 f(t)에 대한 이산 웨이브렛 변환(Discrete Wavelet Transform)의 기본식을 나타내었다[1][3]

$$f(t)=\sum_{j,h} a_{jh}\,\Psi_{jk}(t) \qquad \text{(식 1)}$$

여기서 a_{jh}는 웨이브렛 변환 계수를 나타내고 $\Psi_{jk}(t)$는 웨이브렛 (또는 웨이브렛 기저함수)을 나타낸다.

$$a_{jh}=\langle f(t),\ \Psi_{jk}(t)\rangle = INTf(t)\,\Psi_{jk}(t)\,dt \qquad \text{(식 2)}$$

$$\Psi_{jk}(t)=2^{j/2}\Psi(2^{j}t-k) \qquad \text{(식 3)}$$

식(3)에서 나타낸 것과 같이 웨이브렛 기저함수들은 모웨이브렛(mother wavelet) Ψ 를 신축/팽창 그리고 이동 함으로 얻어지는 일련의 집합이다.

2-2. 다해상도 웨이브렛 변환 시스템

이산 웨이브렛 변환을 이용한 다해상도 분석에는 두 개의 기본 함수가 존재하는데 이는 모 웨이브렛 Ψ 와 모 스케일 함수 Φ이며 이를 식(4)에 나타내었다.

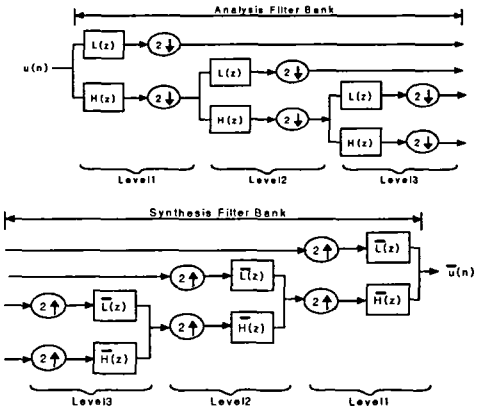
$$f(t)=\sum_k c_j(k)\ \Phi_{j,k}(t)+\sum_k d_i(k)\ \Psi_{jh}(t)$$
$$=\sum_h c_j(k)\ 2^{j/2}\Phi(2^{j}t-k)$$
$$+\sum_k d_j(k)\ 2^{j/2}\Psi(2^{j}t-k) \qquad \text{(식 4)}$$

$c_j(k)$, $d_j(k)$는 각각 스케일 함수와 웨이브렛 함수를 나타내며 식으로 나타내면 다음과 같다[1][3].

$$c_j(k)=\sum_m L(m-2k)\ c_{j-1}(m) \qquad \text{(식 5)}$$

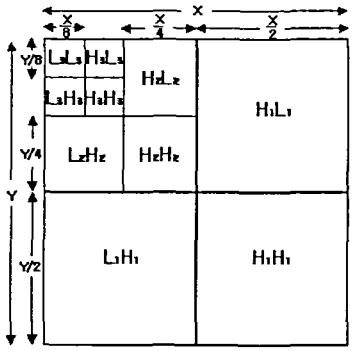
$$d_j(k)=\sum_m H(m-2k)\ d_{j-1}(m) \qquad \text{(식 6)}$$

L(z)와 H(z)는 각각 스케일링 함수 $\Phi(t)$와 웨이브렛 함수 $\Psi(t)$에 일치하는 신장계수로, L는 저역필터, H는 고역필터를 나타낸다.



[그림 1] 3단계 웨이브렛 분해와 합성

그림 1과 같이 저역필터와 고역필터를 번갈아 수행하게 되는데 2차원 영상에 대한 DWT 과정은 영상의 수평/수직방향 데이터를 입력으로하여 고역필터와 저역필터를 통과시키고 그때 나온 결과으로 4개의 주파수 대역(LL, LH, HL, HH)을 얻는다. 가장 효과적인 압축 수행을 위해 3레벨 까지만 DWT을 수행한다. 그림 2는 8x8 이미지에 대한 3단계 레벨 수행에 대한 메모리 맵핑 방법을 보여주고 있다. 그중 LL는 다음단계의 입력으로 사용하며 같은 방법으로 분해과정을 반복한다.그림 2에서 3단계 레벨의 수행하기위한 많은 연산량을 효과적인 메모리 맵핑과 어드레스 처리를 수행하여 연산 해야한다.

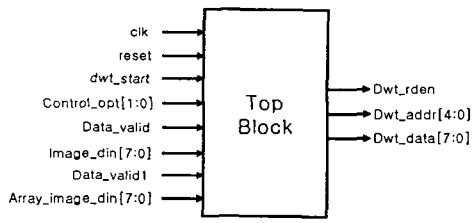


[그림 2] 8x8이미지 3단계 레벨 메모리 맵핑방법

3. DWT 설계

3-1. DWT Top 블록

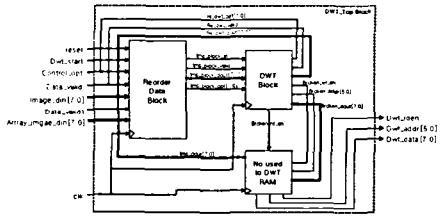
본 논문에서 제안한 DWT 필터는 기존의 데이터 의존성 처리방법 의한 피라미드 알고리즘 사용하지 않고 상위에서 4-tap연산을 수행하기위해 필요한 이전 블록의 데이터를 소프트웨어적으로 처리하여 입력되도록 설계하였다. 기존에 하드웨어적으로 복잡한 데이터를 처리하여 계산하는데 반하여 결과적으로 더 적은 연산량을 수행하는 결과를 낳았다. 그림 3은 전체 DWT처리에 대한 Top_block의 입/출력 신호에 대해 설명하였다[4].



[그림 3] DWT블록의 입/출력 신호

3-2. DWT 전체 블록도

그림 4는 DWT 전체블록으로 3개의 서브 블록 Reorder Data 블록, DWT 블록, No used to DWT Ram블록으로 나누어져 있다. 블록간의 오버랩 데이터를 처리하기 위해서 Reorder 데이터 블록을 이용하여 소프트웨어적 데이터 정렬를 데이터 입력을 처리 한다. 또한 DWT 블록에서는 Dau benches-4 Tap 필터 알고리즘과 동일한 클럭 싸이클에서 로우패스와 하이패스를 동시에 파이프라인 방식을 사용하여 처리하여 수행 속도를 향상시킬 수 있을 뿐만 아니라 QMF의 특성을 이용하여 DWT계산에 필요한 승산기를 절반으로 줄임으로써 하드웨어의 크기를 줄일 수 있어 영상처리 효율을 높일 수 있다.

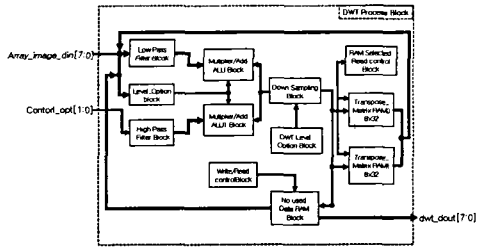


[그림 4] DWT 전체 블록도

DWT에서 가장 효과적인 영상압축을 위해 3단계 레벨을 적용하였고 4개의 분할 영상(LL, LH, HL, HH)으로 나누는 것을 옥타브(Octave)라 하며 각 단계 레벨별로 2차원 DWT수행한 결과에 대한 데이터를 저장하기위해 마지막 No used to DWT RAM블록을 사용하여 저장한 후 IDWT블록에서 다시 읽어 합성하여 처리하도록 설계하였다[5].

3-3. DWT 처리 블록도

본 논문에서 DWT의 효율적인 처리을 위해 로우필터 패스와 하이필터 패스에 대해 병렬로 처리하여 처리속도를 높였고, 2차원 이산 웨이브렛 변환을 수행하기위해 수평방향으로 로우패스와 하이패스을 통과시키고, 다시 수직방향으로 로우패스와 하이패스을 통과시켜 한번의 레벨을 수행한다. 논문에서는 3단계레벨을 수행하여, 첫 번째 레벨에서의 4개의 분할영상의 데이터(L1L1, L1H1, H1L1, H1H1)를 얻고 다시 L1L1의 데이터를 읽어 두 번째 레벨을 수행하여(L2L2, L2H2, H2L2, H2H2)를 얻어 마지막 레벨은 L2L2 데이터를 읽어 처리한다. 이와같이 샘플 이미지(352*288)에 대해 8x8블록으로 나누어 처리할 경우 수평/수직방향으로 352/8*288/8만큼의 반복 수행한다. 그림 5은 DWT 처리 블록도의 데이터 상세 처리블록을 보여준다.



[그림 5] DWT 처리 상세 블록도

승산기, 가산기 연산수행에 있어 로우필터계수와 하이필터계수는 고정계수을 사용하여 부호있는 11비트로 처리하고 입력 데이터 비트는 8비트로 처리하였다. 승산기에서 나온 출력비트는 19비트인데 부동 실수처리값에 대해서는 소수점값을 버림처리하여 나머지 8비트에 대해서만 사용하였다. 각 DWT 처리 레벨 별로 메모리 맵핑과 어드레스 발생기에 대해서 두개의 8(width)x32(Depth) 메모리를 두어 1차원 수평방향에 대한 처리 DWT계수 결과를 동시에 메모리에 저장하여 클럭사이클 줄였고 2차원 수직방향에서의 두개의 메모리에 대하여 데이터 맵핑과 어드레스 발생에 대한 제어 상태를 레벨 별로 처리하여 다시 읽어 2D DWT를 처리하여 나온 중간 데이터를 또 새로운 8(width) x 32(depth) 메모리 두개를 써서 2D DWT처리에 대한 결과를 로우패스와 하이패스를 동시에 동일 클럭동안 동일한 어드레스 발생하여 저장하였다. 결과적으로 클럭싸이클을 줄였고 속도를 향상시켰다. 두 번째 레벨을 수행하기위해 중간데이터의 L1L1의 데이터 값을 읽어 똑같은 방법으로 방법 수행하여 3단계 레벨까지 수행한다. 결과적으로 두개의 메모리을 사용하여 처음 연산 데이터와 중간데이터의 메모리 각각 어드레스 상태을 동일하게 컨트롤할 수 있으며 클럭싸이클도 줄여 처리속도을 향상시켰고, 효과적인 메모리맵핑과 어드리스 컨트롤을 할 수 있었다[4][6].

3-4. DWT 컴포넌트 테이블

DWT처리에서 승산기의 컴포넌트의 수는 각 레벨 별로 테이블로 정의하였고 동시에 레벨 별로 다운샘플링 처리를 하여 처리 속도를 향상시켰다. 표-1, 표-2, 표-3은 각 레벨의 순서데로 필요한 컴포넌트 수와 짝수 데이터에 대한 다운 샘플링처리을 보여주고 있다.

[표 1] 첫 번째 레벨에서의 데이블 표

컴포넌트 / 타임클럭	MUL0-A	MUL0-B	MUL0-C	MUL0-D
0cycle	x	x	x	x
1cycle	0	x	x	x
2cycle	x	1	x	x
3cycle	0	x	2	x
4cycle	x	1	x	3
5cycle	0	x	2	x
6cycle	x	1	x	3
7cycle	0	x	2	x
8cycle	x	1	x	3
9cycle	x	x	2	x
10cycle	x	x	x	3

[표 2] 두 번째 레벨에서의 데이블 표

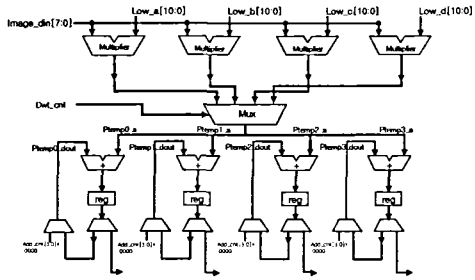
컴포넌트 / 타임클럭	MUL0-A	MUL0-B	MUL0-C	MUL0-D
0cycle	x	x	x	x
1cycle	0	x	x	x
2cycle	x	1	x	x
3cycle	0	x	2	x
4cycle	x	1	x	3
5cycle	x	x	2	x
6cycle	x	x	x	3

[표 3] 세 번째 레벨에서의 데이블 표

컴포넌트 / 타임클럭	MUL0-A	MUL0-B	MUL0-C	MUL0-D
0cycle	x	x	x	x
1cycle	0	x	x	x
2cycle	x	1	x	x
3cycle	x	x	2	x
4cycle	x	x	x	3

3-5. DWT 승산기, 가산기 회로도

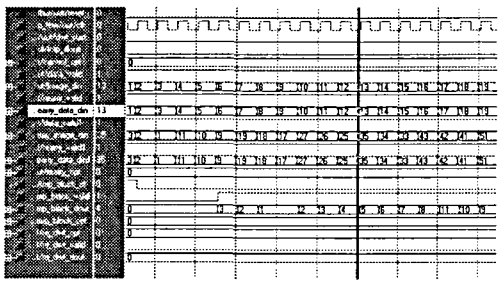
DWT처리에서 승산기와 가산기 연산에서 각 레벨 표을 참조하여 로우패스와 하이패스 계수에 대한 연산을 동시에 수행하고 MUX에 선택된 데이터를 4-tap 가산하여 레벨에 맞은 데이터을 처리한다. 그림 7은 승산기와 가산기의 선택된 데이터의 연산 처리 블록을 보여주고 있다.



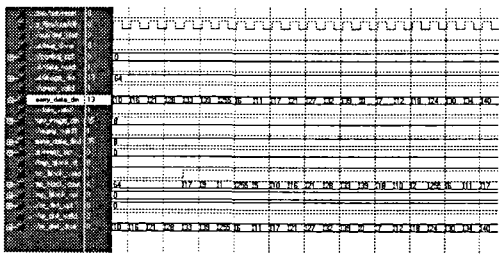
[그림 6] 승산기, 가산기에 따른 데이터 연산 회로도

4. 파형 분석

처음 입력 값과 전블록의 데이터 값이 들어오면 64클럭 이후에 1-레벨의 값이 나오고 이후 16클럭 후 2-레벨 값이 나온다. 이후 4 클럭이 지나면 3-레벨 값을 출력한다.



[그림 7] 입력파형



[그림 8] 출력 파형

6. 결론

본 논문에서는 연산량이 많은 이산 웨이브렛 변환에서 효율적인 메모리 관리와 어드레스 맵핑을 두개의 메모리을 이용하여 클럭싸이클을 줄이고 뿐만아니라 속도를 향상시켰다. 또 승산기, 가산기 연산에서 효과적이고 속도를 향상시키기 위해 승산기, 가산기 연산 수행시 짝수 데이터에 대한 다운 샘플링을 레벨 별로 효과적으로 처리하였다. 그리고 기존 논문에서 적용한 데이터 의존성에 따른 파라미드 알고리즘을 사용하지 않고 상위에서 컨트롤 데이터에 대한 이전처리를 하여 효과적으로 데이터를 처리하였다.

7. 참고문헌

[1]. 김윤홍, 전경일, 방기천, 이우선, 박인정, 이강현, "영상처리를 위한 웨이블렛 변환 디지털 필터의 설계", 전자공학회 논문지 제37권, CI편, 제3호, pp.45-55,2000

[2]. 강봉훈, 이호준, 고형화, "동영상용 웨이블렛 변환 필터의 ASIC설계", 전자공학회 논문지 제36권, S편, 제12호, pp67-75, 1999

[3]. 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

[4]. S. K. Paek, and L. S. Kim, "2D DWT VLSI architecture for wavelet image processing," Electronics Letters, vol.34, no.6, pp.537-538, 1998

[5]. K. K. Parhi, "VLSI architectures for discrete wavelet transform," IEEE Trans. VLSI Systems, vol, pp.191-202, 1993

[6]. I Daubechies, "Orthonormal Bases of Compactlv Supported Wavelets," Comm. PureAppl Math, vol 41, pp 909, 996, 1998