

파이프라인을 이용한 이산 웨이블릿 변환 하드웨어 설계 및 구현

김석*, 이강

한동대학교 전산전자공학부

Taroro30@empal.com* yk@handong.edu

Hardware Design and Implementation of Discrete Wavelet Transform Using Pipelining

Seok Kim and Kang Yi

Computer Science & Electronic Engineering, Handong Global University

요 약

본 논문에서는 최신 정지영상 압축표준인 JPEG2000에 이용되는 알고리즘인 이산 웨이블릿 변환(이하 이산 웨이브릿 변환)을 위한 전용 하드웨어의 파이프라인 설계를 제안한다. 본 연구에서는 3-level 이산 웨이브릿 변환기를 효과적으로 설계하기 위해서 파이프라인 기법으로 데이터의 처리속도를 개선하였다. Xilinx FPGA를 대상으로 한 실험 결과 면적은 약 24%증가된 반면에 throughput은 약 50%정도 향상되었다.

1. 서 론

멀티미디어 기술에 대한 수요가 증가하면서 정지영상 분야에서는 JPEG2000이 새로운 표준으로 소개되어 많은 주목을 끌고 있다. JPEG2000은 JPEG에 비해서 더 높은 압축률을 가지는 장점이 있다[1]. JPEG2000에서 사용되는 영상 압축 알고리즘은 이산 웨이브릿 변환 (이산 웨이브릿 변환)에 근거를 두고 있다. 이산 웨이브릿 변환은 기존의 DCT가 갖는 단점들을 보완하며, 더 높은 성능과 응용력을 가지고 있다.

본 논문에서는 하드웨어 구현시 이산 웨이브릿 변환의 처리속도를 개선하기 위해 기존의 이산 웨이브릿 변환 하드웨어 구조를 개선하여 파이프라이닝 기법을 적용하고자 한다. 실험 결과 본 논문의 제안된 구조

본 논문의 구성은 다음과 같다. 다음 장에서는 이산 웨이브릿 변환의 알고리즘 소개와 일반적 하드웨어 구조를 소개하고, 3장에서는 파이프라인을 이용한 개선된 이산 웨이브릿 변환 구조를 제안한다. 4장에서는 실험결과를 통해 새로운 구조와 기존 구조를 정량적으로 비교한다. 5장에는 결론과 앞으로의 과제를 제시한다.

2. 이산 웨이브릿 변환의 구조

2.1 이산 웨이브릿 변환 처리과정

그림 2.2는 3-level 이산 웨이브릿 변환 처리 순서를

나타낸 것이다.[2] 입력 영상정보가 들어오면, 이것을 가로 방향으로 필터링을 한 후에 1/2로 다운 샘플링(down sampling)을 한다. 이 과정의 결과로 출력되는 값은, low pass (H1)값과 high pass (L1)값으로 나뉘어진다. 게 된다. H1과 L1에 대해서 각각 세로 방향으로 위와 동일한 과정을 거친다. 이로써, 가로와 세로 방향으로 모두 처리가 된 이산 웨이브릿 변환 결과 값이 나오게 되고, 이것은 의 4개의 sub band (LL1, LH1, HL1, HH1)로 나뉘어진다. LL1의 값에 대해서 위와 같은 과정을 반복하여 4개의 sub band HH2, HL2, LH2, LL2,를 얻게 되고, LL2에 대해서 한번 더 이산 웨이브릿 변환을 수행하여 단계 3의 subband HH3, HL3, LH3, LL3를 얻게 된다. 그림 1은 이러한 sub band의 생성 과정을 보여주고 있다. 그림 2는 이산 웨이브릿 처리를 통해서 변환된 이미지가 만들어지고 저장되는 위치를 순서에 따라 보여주고 있다.

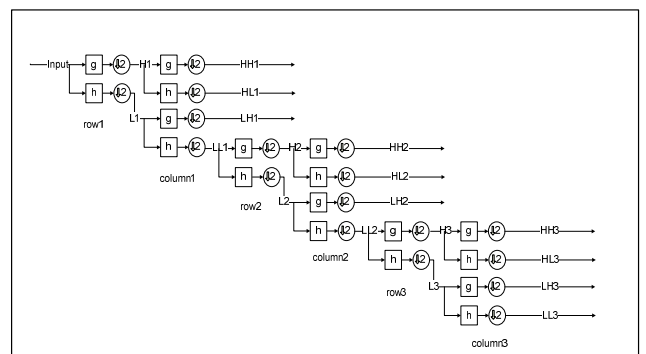


그림 1. 3-level 이산 웨이브릿 변환 처리 과정

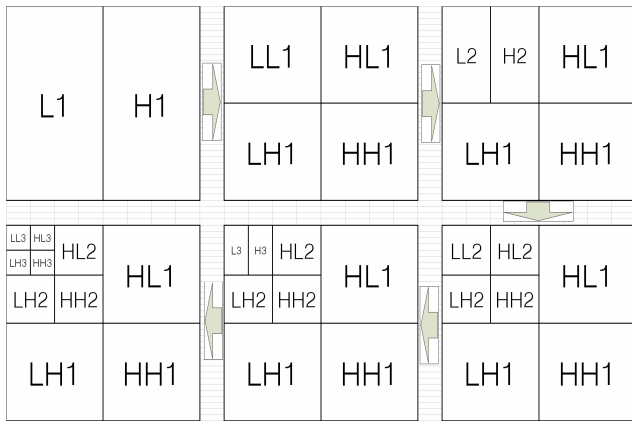


그림 2. 블록화 된 이산 웨이브릿 변환 순서

2.2 이산 웨이브릿 변환의 필터 하드웨어 구조

본 논문의 이산 웨이브릿 변환에서 사용하는 FIR Convolution 필터를 사용한다. 다음 식 1은 이것을 수학적으로 표현 한 것이다[1].

$$y_L(n) = \sum_{i=0}^{\tau_L-1} h(i)x(2n-i), \quad y_H(n) = \sum_{i=0}^{\tau_H-1} g(i)x(2n-i) \quad \text{식 1.}$$

$h(i)$ 는 저역 필터 계수이고, $g(i)$ 는 고역 필터 계수 이다.

표 1. Le Gall spline filter의 계수

	저역필터 $h(i)$	고역필터 $g(i)$
$x(0)$	3/4	1
$x(1), x(-1)$	1/4	-1/2
$x(2), x(-2)$	-1/8	

표 1은 본 논문에서 구현한 이산 웨이브릿 변환에서 사용되는 필터인 Le Gall spline filter[3]의 계수를 나타낸 것이다. Low pass filter는 5의 length를, High pass filter는 3의 length를 갖는다. 위의 filter를 사용하면 무 손실 압축을 할 수 있고, 역 변환이 가능하다는 장점이 있다. 또한 계수의 값이 간단하기 때문에 하드웨어 적으로 computation power가 적게 소모 될 수 있다. 반면에 filter의 length가 더 긴 다른 filter보다 세부적으로 덜 정확한 결과 값이 나올 수 있다는 단점이 있다. 본 논문에서는 하드웨어 구현에 중점이 맞춰져 있으므로 Le Gall spline filter가 적합하다.

이 필터의 하드웨어 구조를 그림 3은 보여주고 있다. Semi-systolic 구조[1]를 사용하여 구현되어 있다. 그림 4는 1차원 이산 웨이브릿 변환을 처리하는 하드웨어의 블록 다이어그램이다. 그림 2.2의 Semi-systolic구조를 본 논문에서 구현하는 필터인 5/3 tap filter의 형태에 맞게 바꾼 것이다.

입력 신호 $X(i)$ 가 들어오면 이 signal을 register에 저장 하였다가 다섯 개의 입력이 $X_{i-2}, X_{i-1}, X_i, X_{i+1},$

X_{i+2} 순서로 누적이 된 후 이 값을 1-D 이산 웨이브릿 변환 Cell로 보낸다. 여기서 1-D 이산 웨이브릿 변환 Cell은 실질적인 계산을 해주는 하드웨어 부분으로서, 필터의 계수가 들어 있다.

5개의 입력이 들어가서 필터링을 거쳐서 고역 필터의 값과 저역 필터의 값이 출력으로 나온다. 여기서 필터링은 과정은 각각의 입력에 필터의 계수를 곱한 후, 저역 필터는 입력 신호의 (-2) 번째부터 (+2) 까지를 더하고, 고역 필터는 입력 신호의 (-1)부터 (+1) 까지를 더한 값이 나오게 된다. 이러한 과정은 FIR 필터 형태의 Convolution이라고 볼 수 있다.

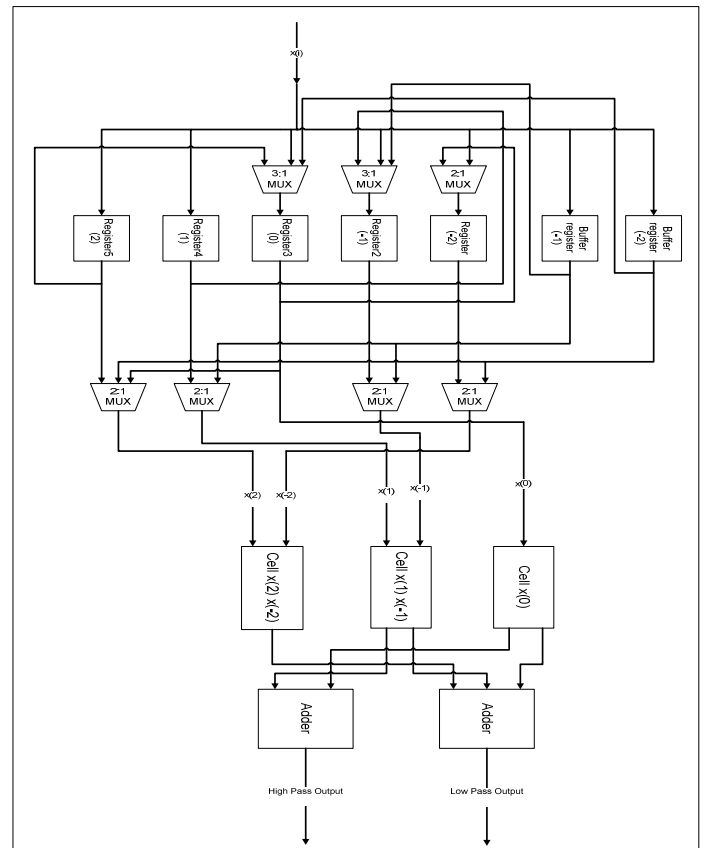


그림 3. 1-D 이산 웨이브릿 변환 블록 다이어그램

3. 제안된 하드웨어의 구조와 처리과정

3.1 파이프라이닝을 통해 개선한 처리과정

그림 5 는 이산 웨이브릿 변환의 처리 과정에 걸리는 시간을 각 단계 별로 나타낸 것이다. 여기서 가장 왼쪽에 있는 단계에 나타나 있는 L1과 H1은 row 방향으로 1-level 이산 웨이브릿 변환을 수행하는 것을 나타내는 것으로, 이산 웨이브릿 변환의 가장 첫 단계이다. 그림 5 에 따르면 가장 처음 단계를 제외한 나머지 단계에서 소요되는 총 시간이 가장 처음 단계를 수행하는데 걸리는 시간의 31/32만큼이라는 것을 알 수 있다. 본 논문의 아이디어는 이 것을 처음 단계와, 나머지 단계의 합의 단계로 나누어 두 단계의

파이프라인 기법을 적용해서 성능을 개선 하는 것 이다. 이에 관한 것을 그림 6으로 확인 할 수 있다.

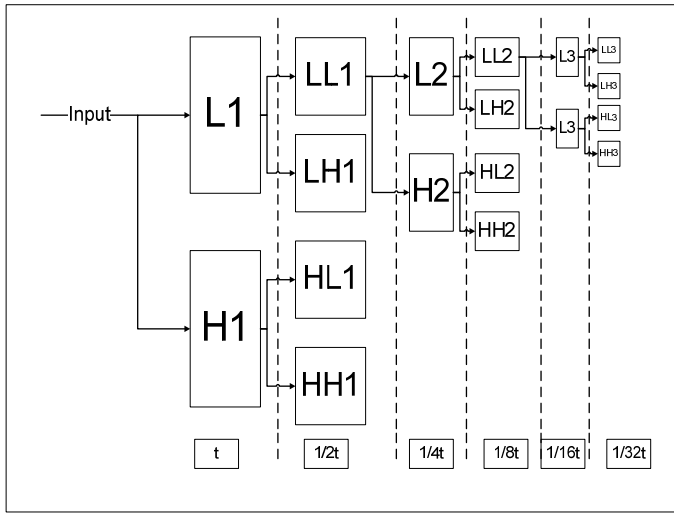


그림 4. 이산 웨이브릿 변환의 단계별 소요시간

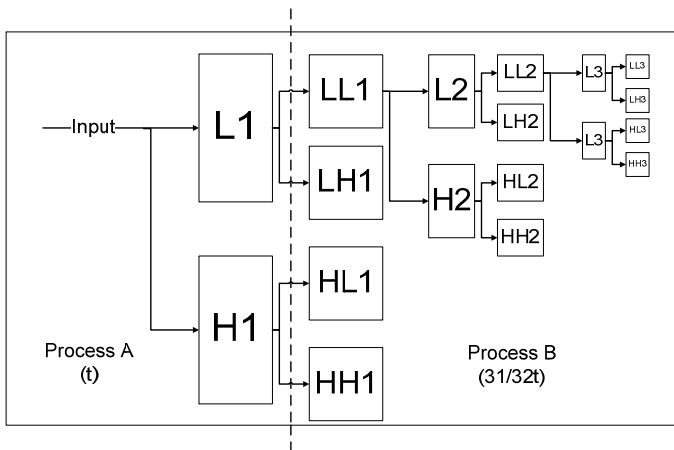


그림 5. 2단 파이프라인을 적용한 이산 웨이브릿 변환

3.2 파이프라인 기법이 적용된 DWT 하드웨어

Convolution의 계산 형태를 갖는 Semi-systolic 구조에서는 반복되는 처리과정이 많지만 계산을 하는 하드웨어 부분은 단순한 구조를 갖는다. 따라서 하드웨어의 개수를 늘려도 전체적으로 봤을 때는 비용이 많이 증가하는 것이 아니라고 할 수 있다. 파이프라이닝을 통한 개선안에서는 하드웨어의 면적을 늘려서 그 이상의 높은 처리속도 항상 효과를 낼 수 있도록 한다. 본 논문에서는 1-D 이산 웨이브릿 변환 하드웨어 개수를 하나 늘려서 효율적인 파이프라이닝이 이루어지도록 설계 하였고, 이에 대한 그림을 그림 6에서 확인 할 수 있다.

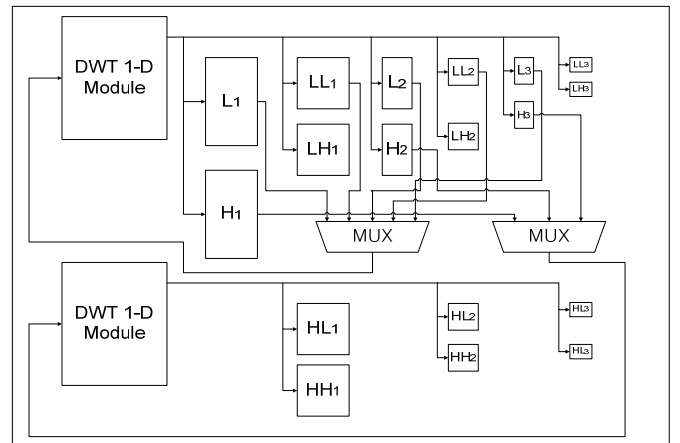
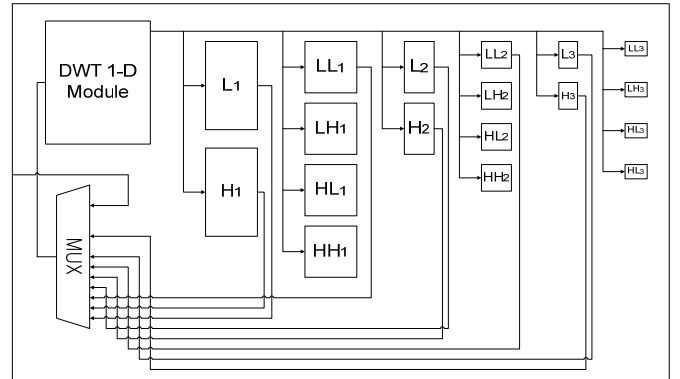


그림 6. 파이프라인 구조를 적용하기 전과 후의 비교

4. 제안된 하드웨어의 성능분석과 실험 결과

하드웨어의 구현은 VHDL로 표현하고 Xilinx Spartan 3 FPGA를 타겟으로 구현했다. 성능의 비교를 위해서 128 * 128 픽셀의 영상정보 처리를 기준으로 실험했다. 그림 7에서 그림 9는 이산 웨이브릿 변환기의 구현결과를 본 논문에서 제안한 개선안과 파이프라인을 적용하지 않은 것을 서로 비교한 그림이다. 구조 개선 전과 개선 후의 클럭 수와 면적의 비교를 그림 7과 그림 8을 통해 확인할 수 있다. 파이프라인에 의해서 하드웨어가 추가되므로 전체 면적은 24%정도 늘어나지만, 처리속도는 약 50%정도 빨라진다는 것을 알 수 있다.

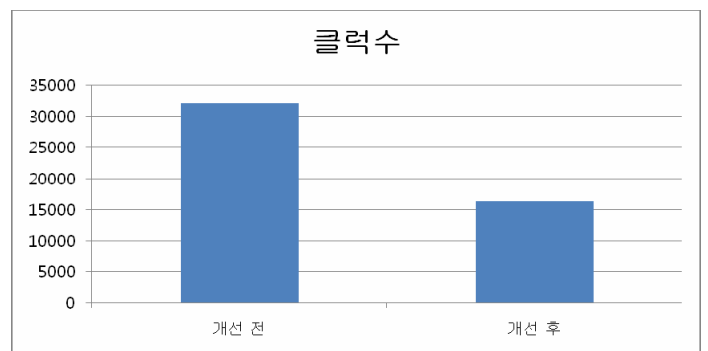


그림 7. 파이프라이닝 구조를 적용하기 전과 후의 클럭수 비교

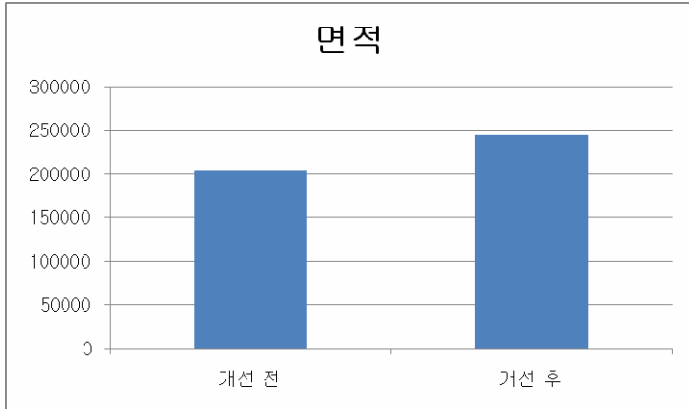


그림 8. 파이프라이닝 구조를 적용하기 전과 후의 면적 비교

그림 9는 개선 전과 개선 후의 Throughput을 비교한 것을 보여 준다. Throughput은 개선 후에 약 50% 정도 증가 함을 알 수 있다. Throughput을 구하는 공식은 식 2와 같다. 개선 전의 클럭 주파수는 80.106 MHz가 나왔고, 개선 후의 클럭 주파수는 80.666 MHz가 나왔다.

$$\text{Throughput} = \text{클럭 주파수} * (\text{처리량/클럭 수}) \text{ 식 2.}$$

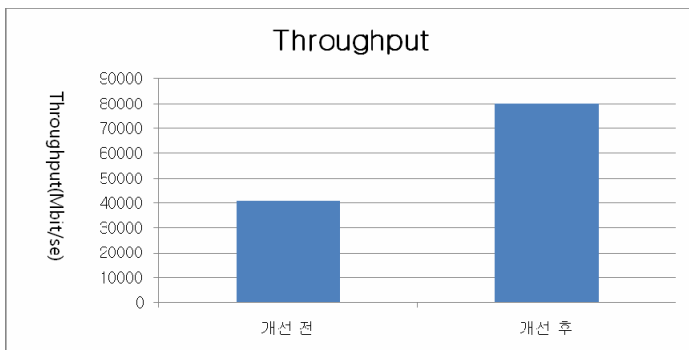


그림 9. 파이프라이닝 구조를 적용하기 전과 후의 클럭수 비교 Throughput 비교

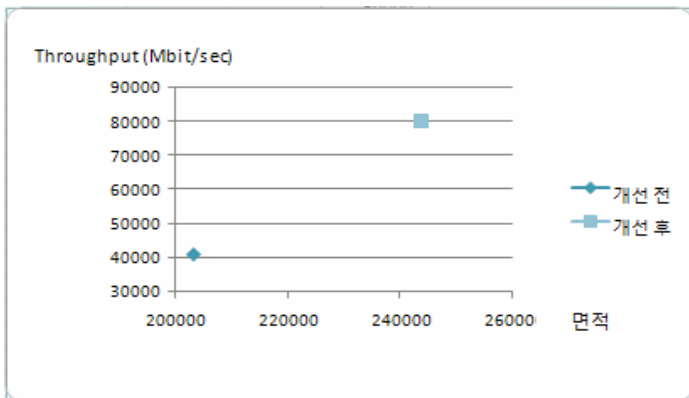


그림 10. 파이프라이닝 구조를 적용하기 전과 후의 면적 대비 Throughput 비교

표 1. 파이프라이닝 구조를 적용하기 전과 후의 성능 비교

	개선 전	개선 후
면적	203,255 Gates	243,750 Gates
클럭수	32,256 Clock	16,384 Clock
Throughput	40,688 Mbit/sec	80,666 Mbit/sec

그림 10은 파이프라이닝 구조를 적용하기 전과 후의 면적 대비 Throughput을 비교한 그래프로서, 면적과 성능의 trade-off 관계를 보여주고 있다. 표 1은 파이프라이닝 구조를 적용하기 전과 후의 성능을 비교한 표이다.

5. 결론

이산 웨이브릿 변환은 현재 정지영상의 압축 중 가장 높은 압축률을 지닌 압축표준인 JPEG2000의 핵심 알고리즘으로 사용되고 있다. 본 논문은 이산 웨이브릿 변환기의 하드웨어 구현을 효과적으로 하기위해서 2단 파이프 라인 구조를 제안하고 이를 FPGA 상에서 검증하였다. 본 논문의 실험 결과에 의하면, 2단 파이프라인 구조를 가지는 웨이브릿 변환기를 사용함으로써 파이프라인을 하지 않은 구조에 비해서 24% 정도의 면적 증가로써 약 50% 정도의 처리 속도 향상이 이루어졌다.

참고 문헌

- [1] Tinku Acharya, Ping-Sing Tsai “JPEG2000 Standard for Image compression in VHDL”, 2004
- [2] David S. Taubman, Michael W. Marcellin “JPEG200: Image Compression Fundamentals, Standards and Practice”, 2002
- [3] Rafael C. Gonzalez, Richard E. Woods “Digital Image Processing 2nd Edition”, 2003