# JPEG 2000 부동소수점 연산용 Filter의 SoC 설계 및 구현

장 종 권†


## 요 약

JPEG 2000 알고리즘은 기존의 정지영상 압축 JPEG 알고리즘에서 발생하는 블록화 현상의 문제점을 해결하였지만, 압축 율과 이미지 복원율을 높이기 위한 과정이 복잡해 졌고 부동소수점의 연산 시간이 많이 걸리는 단점을 가지고 있다. 이 단점을 보완하기 위하여 본 논문은 JPEG 2000 알고리즘의 부동소수점 연산이 많은 필터 부분을 하드웨어로 구현하였다. 이 DWT Filter[1] 칩은 Daubechies 9/7 filter[6]을 기반으로 설계되었고 성능과 크기(반도체 숫자)를 최적화하기 위해서 3 단계 파이프라인 시스템으로 구성되었다. 본 논문에서 설계한 Filter는 소프트웨어로 구현된 것보다 부동 소수점의 연산에서 7배 정도 성능이 향상되었다.


**키워드 : JPEG, 정지영상 압축, 부동소수점, 압축율, 복원율**

# A SoC design and implementation for JPEG 2000 Floating Point Filter

Jong Kwon Chang†


## ABSTRACT

JPEG 2000 is used as an alternative to solve the blocking artifact problem with the existing still image compression JPEG algorithm. However, it has shortcomings such as longer floating point computation time and more complexity in the procedure of enhancing the image compression rate and decompression rate. To compensate for these we implemented with hardware the JPEG 2000 algorithm's filter part which requires a lot of floating point computation. This DWT Filter[1] chip is designed on the basis of Daubechies 9/7 filter[6] and is composed of 3-stage pipeline system to optimize the performance and chip size. Our implemented Filter was 7 times faster than software based Filter in the floating point computation.


**Key Words : JPEG, Still Image Compression, Floating Point, Compression Rate, Decompression Rate**

## 1. 서 론

JPEG 2000은 기존의 JPEG에서 발생했던 블록화 현상의 문제점을 해결하고 정지 영상의 압축율과 이미지의 복원 율을 크게 향상시켰다. 또한 정지 영상의 압축 율이 개선됨에 따라 메모리 크기에 제한받는 휴대용 기기의 적용에 효율적인 압축방식으로 고려되고 있다. 하지만 JPEG 2000의 압축 방식 알고리즘은 그 과정이 더욱 복잡하고 부동 소수점의 연산도 크게 증가된 단점을 가지고 있다.

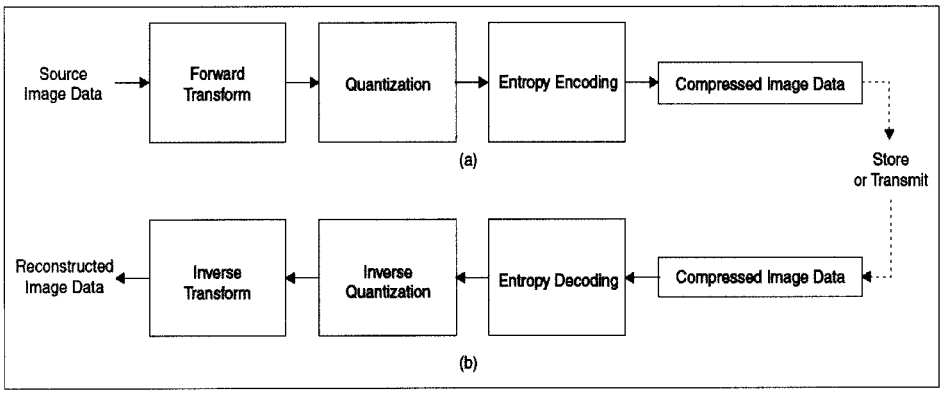
본 논문은 JPEG 2000의 부동소수점 연산을 수행하는 필터를 H/W로 구현함으로써 S/W로 구현된 필터로 수행할 때 발생하는 성능의 저하 문제를 해결하고자 한다. 이 필터의 SoC설계는 Altera사의 Excalibur(H/W 환경)와 Quartus II, Linux, ADS 2.1(S/W 환경)의 기반에서 이루어졌다. 본 논문의 배경지식으로 JPEG 2000에 대한 알고리즘을 소개하고, DWT 모듈, Excalibur Device 및 Quartus II을 설명하고, DWT Filter의 하드웨어 설계를 기술한다. 마지막으로 Filter설계 후 개선된 점과 향후 과제에 대해서 알아본다.

## 2. JPEG 2000 Decoding과 Encoding

JPEG 2000 압축 엔진의 block diagram은 (그림 1)과 같다. (그림 1)은 정지영상 압축 알고리즘인 JPEG의 block diagram처럼 보이지만, 각 block이 수행하는 일은 기존의 JPEG의 것과는 근본적인 차이가 있다. 전체 시스템의 흐름을 잠시 살펴보자.

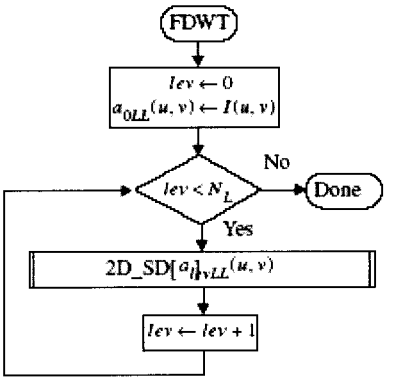
- 이미지를 직사각형 타일로 분해한다. 분해된 타일은 원본 또는 재구성된 이미지의 기본요소로 사용된다.
- 각 타일에 wavelet 변환이 적용된다. 변환된 타일은 각각 서로 다른 분해(resolution) level을 가진다.

※ 이 논문은 2005년도 울산대학교 연구비 지원과 IDEC의 CAD Tool 지원으로 수행되었음.
† 정 회 원 : 울산대학교 컴퓨터 정보통신공학부 교수
논문접수 : 2006년 2월 13일, 심사완료 : 2006년 5월 18일




(그림 1) General block diagram of the JPEG 2000 (a) encoder and (b) decoder.[2]

- 분해 level은 subbands of coefficients로 이루어져 있다. 이것은 타일의 local area의 주파수 특성을 나타낸다.
- subbands of coefficient가 계산되어 "code block"이라고 불리는 직사각형의 배열 값으로 사용된다.
- code block의 계수 값으로 구성된 bit plane은 entropy coding으로 변환된다.
- 정보를 많이 가지고 있는 영역은 주위 배경보다 더 높은 품질의 coding으로 변환된다.
- error resilience(에러 복원력)가 허용되도록 bit stream에 markers가 추가된다.
- main header는 code stream의 시작부분에 있으며 이미지를 재구성하기 위한 여러 가지 정보들을 가지고 있다.



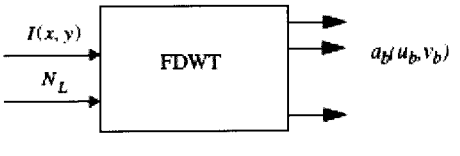

(그림 3) The FDWT procedure

## 3. DWT 소개

ISO/IEC[1]에서 제안하는 JPEG 2000의 표준 DWT 알고리즘을 살펴보기로 하자.

### 3.1 FDWT[1]

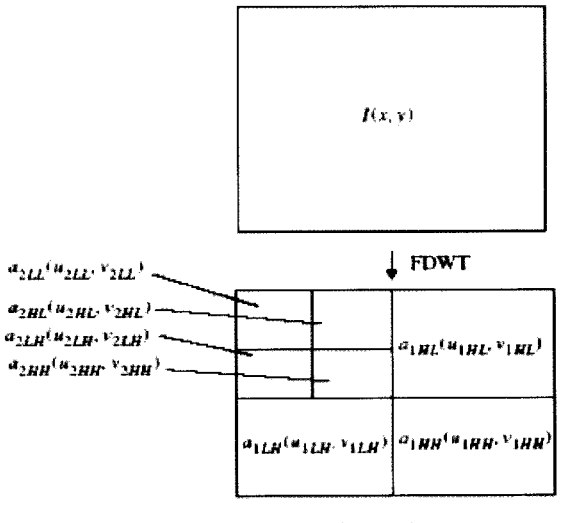
Forward discrete wavelet transformation(FDWT)은 DC-level shift된 tile샘플 I(x,y)을 subband coefficients ab(ub, vb)로 변환 시킨다(그림 2). $N_L$은 decomposition level의 숫자를 나타낸다.




(그림 2) Inputs and outputs of the FDWT procedure

전체 Subband의 개수는 $(3 \times N_L)+1$가 된다. (그림 4)는 $N_L$ = 2일 경우의 모든 subband를 나타낸다.

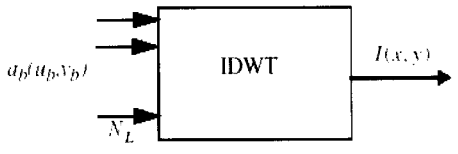
FDWT procedure는 처음에 변수 lev를 0으로, subband a0LL(u,v)를 입력 값 I(u,v)로 초기화 시킨다. 2D_SD procedure가 $N_L$ 값만큼 반복하여 실행된다(그림 3).




(그림 4) The FD WT($N_L$ = 2)

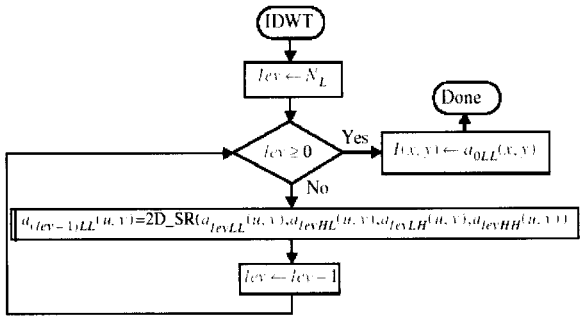
### 3.2 IDWT[1]

Inverse discrete wavelet transformation(IDWT)은 FDWT의 procedure를 역순으로 계산해서 이미지를 재구성한다. IDWT는 coefficients ab(ub,vb)의 subband set를 DC level shift된 tile component samples I(x,y)로 변환시킨다. (그림 5)의 $N_L$은 decomposition level의 숫자를 나타낸다. 여기서 영상속의 subband의 숫자는 $(3 \times N_L)+1$ 이다.




(그림 5) Inputs and outputs of the IDWT procedure

IDWT procedure는 변수 lev를 $N_L$로 초기화한다. 2D_SR procedure가 $N_L$ 값만큼 반복하여 실행되고, lev값은 감소되며 LL subband가 생성된다. Procedure의 끝 부분에서 subband I(x,y)가 결과 값으로 출력된다.(그림 6, 7)




(그림 6) The IDWT procedure

## 4. 하드웨어와 소프트웨어 환경

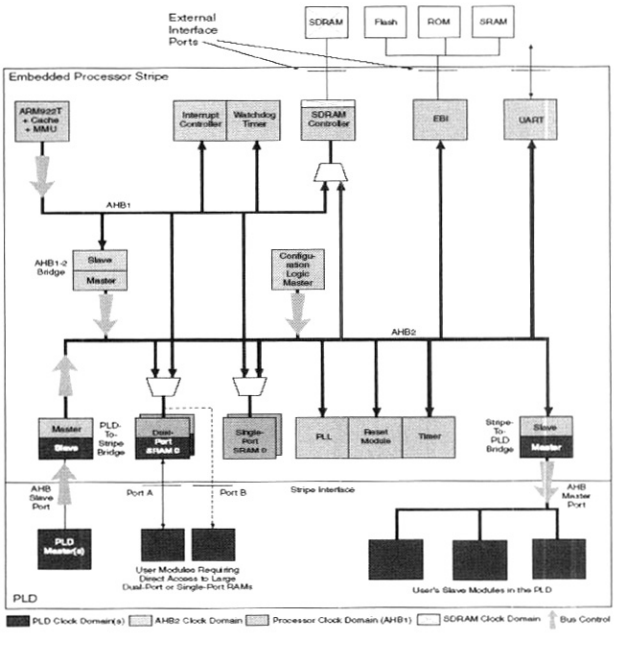
### 4.1 하드웨어 환경

JPEG 2000을 실제 환경에 적용시키고 시험하기 위해서 Altera Embedded Processor인 Excalibur Device에 탑재하였다. (그림 8)에 보인바와 같이 Excalibur칩은 하나의 칩 내부에 ARM9 프로세서, 100~10만 게이트 프로그래머블 로직, 메모리 컨트롤러, 외부 확장버스, 그리고 클럭 로직 등을 모두 포함하고 있다. 따라서 이 칩은 SoC을 설계하고 검증하는데 유용하게 사용되고 있다. 설계된 하드웨어 장치와 소프트웨어를 Excalibur칩에 탑재하여 쉽게 구동하고 관찰할 수 있다.

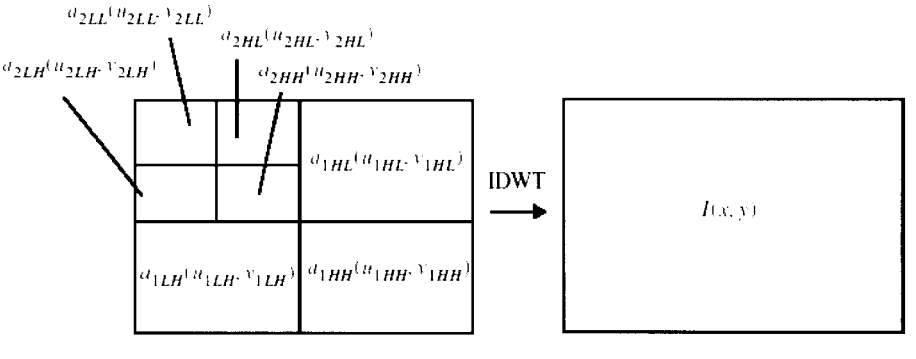
### 4.2 소프트웨어 환경

Altera Quartus II는 Excalibur Device칩 안에 내장되어 있는 ARM 프로세서와 FPGA에 실행프로그램을 탑재하고 구동 환경을 조성하는 CAD tool이다. Quatrus II는 하드웨어 설계 File과 컴파일 된 실행 File을 Excalibur Device칩에 탑재 가능한 File로 생성하여준다. 또한 Quatrus II는 ARM 프로세서의 외부 버스 인터페이스(AMBA BUS), 내부 및 외부 clock 사이클, memory map이 설계자의 목적에 부합되도록 설계환경을 조성하여준다. ARM 프로세서를 동작시키기 위한 소프트웨어는 ADS 2.1(컴파일러)이 컴파일한다. 위의 기능들은 MagaWizard Plug-In Manager를 이용해서 설정된다.

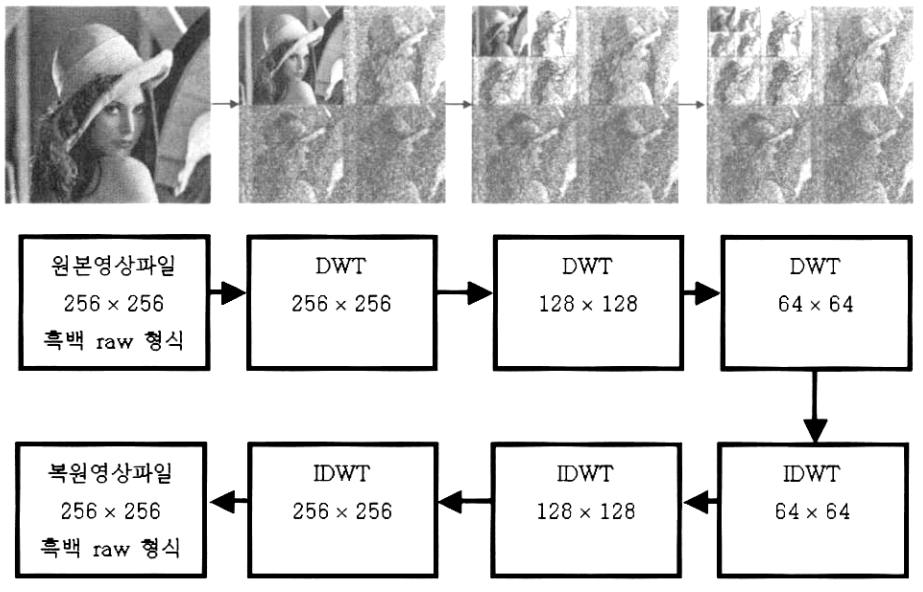
설계된 device를 OS수준에서 시험하기 위해서 저렴한 가격으로 제공되는 embedded linux를 사용하였다. 크로스 컴파일을 이용하여 linux를 설치하고 하드웨어로 설계한 File을 FPGA영역에 탑재하여 제어하고 시험할 수 있었다.




(그림 8) Excalibur Device System Architecture[7]




(그림 7) The IDWT($N_L$ = 2)




(그림 9) 전체 흐름도 및 각 모듈별 출력 영상

## 5. JPEG 2000 설계 및 구현

### 5.1 소프트웨어 설계 및 구현

ISO에서 제안하는 JPEG 2000의 사양과 설계 단계별 구현의 일치 여부는 C 언어로 프로그램하여 검증하였다. JPEG 2000의 알고리즘을 구현하기 위한 입력과 출력은 *.raw 형식의 흑백영상으로 나타냈고, tile size는 256X256 pixel로 정하였다.

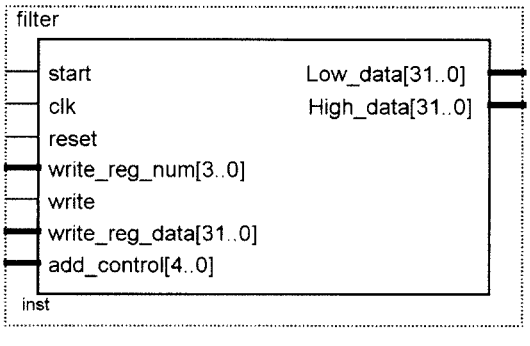
원본 영상이 흑백 raw 형식으로 입력되어 DWT(256X256) 모듈을 통과하면 4등분된 영상이 출력 된다. 주파수 분해 된 영상 중, 왼쪽 상단에 위치한 저주파수 영역인 LL(128X128) 영상을 DWT(128X128) 모듈에 입력시키면 다시 주파수 분해가 된다.

이 과정을 3번 반복시키면 (그림 9)에 보이는 영상의 출력이 나온다. 왼쪽 상단 sub-band 영상으로 갈수록 저 주파수 영상이 나타나고 오른쪽 하단 sub-band 영상으로 갈수록 고 주파수 영상이 나타난다. 저 주파수 영상은 원본 영상의 모습을 대부분 가지고 있지만, 고 주파수 영상은 원본 영상의 데이터를 대부분 가지고 있지 않다. 이것은 손실압축기법의 중요한 요소 중 하나다. 따라서 데이터의 양이 많은 영상은 압축 율을 줄이고 데이터의 양이 적은 영상은 압축 율을 높이는 것이 손실압축 기법의 핵심이다.

### 5.2 하드웨어 설계 및 구조

JPEG 2000의 DWT 필터를 소프트웨어로 구현하고 시간 소요가 많은 부동 소수점 연산 기능을 하드웨어로 구현하였다. 하드웨어와 소프트웨어의 연결은 Excalibur 내부에 있는 AMBA 버스에 의해서 만들어 진다. AMBA 버스의 제어 신호, 데이터 값 및 address값을 분석해서 해당 부품에 데이터를 보내는 Address Decoder는 하드웨어로 설계하였다.

(그림 10)은 Quartus II의 기반에서 설계한 DWT Filter의 입력과 출력을 보여준다. DWT Filter는 Address Decoder에서 9개의 데이터 값(Pixel 값), write 신호, 데이터가 저장될 4비트 주소 값 write_reg_num을 받아서 Register File에 저장한다. Lena의 9개 데이터 값이 저장된 다음에 start 신호가 Low에서 High로 변경되면 DWT 필터가 부동소수점의 계산을 시작한다. 계산 결과로 Low_data에 low pass filter를 통과한 값이 저장되고, High_data에 high pass filter를 통과한 값이 저장되어 진다. 여기서 Filter의 동작 clock은 40MHz이다.

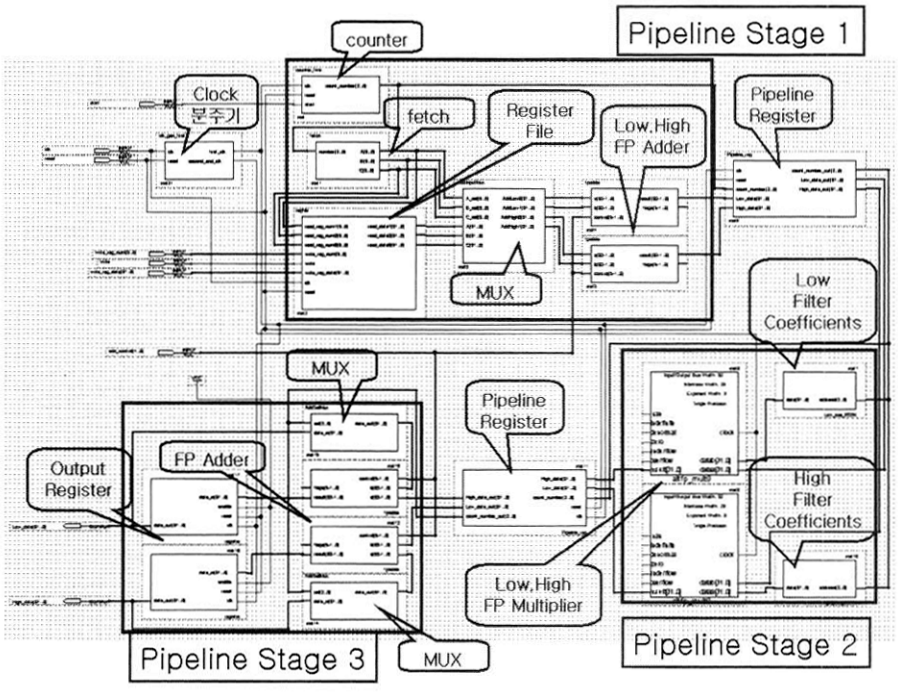



(그림 10) DWT Filter 입력과 출력

(그림 11)는 Quartus II의 기반에서 DWT Filter를 schematic으로 구현한 화면을 보여준다. 각 pipeline stage의 실행 시간이 <표 1>에 나타나있다.

〈표 1〉 Pipeline stage간 필요 clock수와 실행시간

| | clock number | need clock cycle | 실행시간 |
|---|---|---|---|
| Stage 1 | 1 | first clock | 129 ns |
| Stage 2 | 6 | 1st clock, 6th clock | 138 ns |
| Stage 3 | 1 | first clock | 109 ns |




(그림 11) DWT FilterSchematic 설계도

〈표 2〉 하드웨어 구현에 따른 delay

| | Delay |
|---|---|
| 싱글 사이클 | 1.9 μs |
| 파이프라인 | 1.068 μs |

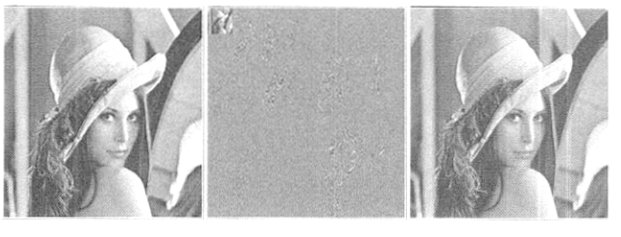
<표 2>는 파이프라인으로 설계된 DWT Filter가 싱글 사이클 시스템과 비교하여 약 2배의 성능이 향상된 것을 보여준다.

## 6. 실행 결과

Convolution based filter는 부동소수점 연산이 많기 때문에 부동소수점 연산의 처리 속도에 따라서 filter의 성능이 좌우 된다. 필터를 통과한 값을 down-sampling한 값과 down-sampling한 값을 필터에 통과시킨 값은 동일하다. down-sampling한 값을 필터에 통과시킴으로서 convolution 계산이 반으로 줄어든다. 이 방법을 사용하여 filter의 영상 압축에 요구되는 부동 소수점 계산 량이 절반으로 줄어들었다.

(그림 12)는 설계된 filter를 사용해서 lena 이미지를 압축하고 재구성한 것을 보여준다. 외관상 원본 이미지와 차이가 없고, 블록화 현상도 개선되었다. 소프트웨어와 하드웨어로 설계한 것 둘 다 모두 같은 결과를 보였다. 하지만, <표 3>에서 보인바와 같이 부동소수점 크기를 32bit long 형으로 고정 시켰기 때문에 재구성한 영상은 64bit double 형의 것보다 PSNR이 작은 것으로 나왔다.

<표 4>는 하드웨어로 구현한 DWT Filter의 실행 clock 수를 소프트웨어로 구현한 것과 비교하여 나타냈다. Clock 수는 7배 이상 효율적이었지만 clock 주기의 큰 차이로 인해서 실질적인 수행시간은 약 2.8배 정도 효율이 떨어진 것으로 나타나 있다. 하드웨어로 구현된 칩은 소프트웨어의 방식보다 clock 수가 적게 요구됨으로 휴대용과 같은 저 전력 소비를 요구하는 장비에 적합할 것으로 생각된다.



(a) (b) (c)

(그림 12) (a) 원 영상, (b) 필터를 3번 통과한 결과, (c) 그림 (b)을 다시 재구성한 영상

〈표 3〉 부동소수점 데이터 크기별 PSNR 결과

| | 32 bit (PSNR) | 64 bit (PSNR) |
|---|---|---|
| S/W | 38.743 | 46.209 |
| H/W | 38.743 | N/A |

〈표 4〉 DWT Filter의 하드웨어 설계와 소프트웨어 설계 비교

| | Clock 수 | 수행환경 (chip 종류) | Clock주기 (MHz) | 수행 시간 |
|---|---|---|---|---|
| S/W | 300 | Intel Pentium 3 | 800 | 0.375μs |
| H/W | 42 | Altera Excalibur EPXA4F1020C2 | 40 | 1.05μs |

## 7. 결론 및 향후 과제

JPEG 2000은 현재 가장 많이 사용하고 있는 정지영상 압축 알고리즘인 JPEG에서 발생했던 블록화 현상을 해결하고, 좀 더 향상된 압축율과 이미지 복원 율을 보이고 있다. 하지만, JPEG 2000은 압축 알고리즘이 더욱 복잡할 뿐만 아니라 수행 시간이 긴 부동소수점 연산이 많다는 단점도 가지고 있다. 이 단점을 보완하기 위해서 JPEG 2000 알고리즘에서 부하가 많이 걸리는 필터 부분을 하드웨어로 구현하였다. Daubechies 9/7 filter를 기본으로 필터를 설계하였다. 하드웨어로 구현된 DWT Filter 칩은 속도와 크기(반도체 숫자)를 최적화하기 위해서 3 단계 파이프라인 시스템으로 설계되었다. 하나의 파이프라인 단계는 6 clock이 요구되며 동작 속도는 40 MHz이다. 하드웨어로 구현된 DWT Filter 칩은 소프트웨어로 구현된 필터 보다 clock 수에서 7배 이상의 효율을 보이지만 동작 clock 주기에서 효율이 떨어진다. 왜냐하면 하드웨어를 설계하기 위한 target device(Altera Excalibur)의 수행 속도가 상용화된 프로세서(Intel 계열 칩)에 비교하여 느리기 때문이다. 보다 좋은 결과를 위해서 고성능 target device가 요구 되고, 반도체 단계에서 최적화된 ASIC 설계가 요구된다. 또한 더욱 정교한 파이프라인 구조의 설계가 요구된다.

## 참 고 문 헌

[1] ISO/IEC FCD15444-1 : 2000 (V1.0, 16 March 2000)

[2] Athanassios Skodras, Charilaos Christopoulos, and Touradj Ebrahimi, "The JPEG 2000 Still Image Compression Standard", IEEE signal processing magazine, 2001.

[3] 홍성훈, "JPEG 2000정지영상 압축부호화 표준의 이해", IDEC 2002, 2002.

[4] David S. Taubman, Michael W. Marcellin, "JPEG 2000 Image Compression Fundamentals, Standards and Practice", Kluwer Academic Publishers, 2002.

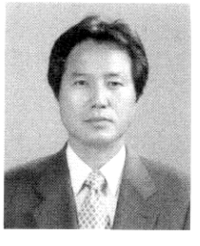
[5] Steve Hollasch. "IEEE Standard 754 Floating Point Numbers", IEEE, 2004.

[6] Marc Antonini, Michel Barlaud, Member, IEEE, Pierre Mathieu, and Ingrid Daubechies, Member, IEEE, "Image Coding Using Wavelet Transform", IEEE transaction on image processing, 1992.

[7] Altera, http://www.altera.com, "Excalibur Devices Hardware Reference Manual", Altera Homepage, November 2002.

[8] ARM, "AMBA Specification", ARM, 2001

[9] ARM, "ARM922T(Rev 0) Technical Reference Manual", ARM Limited, 2001.



장 종 권

e-mail : jkchang@ulsan.ac.kr

1990년 The University of Texas at Ausitin, 전기공학(공학박사)

1991년~현재 울산대학교 컴퓨터정보통신공학부 교수

관심분야 : VLSI 설계 및 시험 등