

# 블라인드 워터마킹을 내장한 실시간 비디오 코덱의 FPGA기반 단일 칩 구조 및 설계

정희원 서영호\*, 김대경\*\*, 유지상\*\*\*, 김동욱\*

## FPGA-based One-Chip Architecture and Design of Real-time Video CODEC with Embedded Blind Watermarking

Young-Ho Seo\*, Dai-Gyoung Kim\*\*, Ji-Sang Yoo\*\*\*, and Dong-Wook Kim\* *Regular Members*

### 요 약

본 논문에서는 입력 영상을 실시간으로 압축 및 복원할 수 있는 하드웨어(hardware, H/W)의 구조를 제안하고 처리되는 영상의 보안 및 보호를 위한 워터마킹 기법(watermarking)을 제안하여 H/W로 내장하고자 한다. 영상압축과 복원과정을 하나의 FPGA 칩 내에서 처리할 수 있도록 요구되는 모든 영상처리 요소를 고려하였고 VHDL(VHSIC Hardware Description Language)을 사용하여 각각을 효율적인 구조의 H/W로 사상하였다. 필터링과 양자화 과정을 거친 다음에 워터마킹을 수행하여 최소의 화질 감소를 가지고 양자화 과정에 의해 워터마크의 소실이 없으면서 실시간으로 동작이 가능하도록 하였다. 구현된 하드웨어는 크게 데이터 패스부(data path part)와 제어부(Main Controller, Memory Controller)로 구분되고 데이터 패스부는 영상처리 블록과 데이터처리 블록으로 나누어진다. H/W 구현을 위해 알고리즘의 기능적인 간략화를 고려하여 H/W의 구조에 반영하였다. 동작은 크게 영상의 압축과 복원과정으로 구분되고 영상의 압축 시 대기시간이 워터마킹이 수행되며 전체 동작은 A/D 변환기에 동기하여 펠드단위의 동작을 수행한다. 구현된 H/W는 ALTERA사의 APEX20KC EP20K600CB652-7 FPGA 칩에서 69%(16980개)의 LAB(Logic Array Block)와 9%(28352개)의 ESB(Embedded System Block)을 사용하였고 최대 약 82MHz의 클럭주파수에서 안정적으로 동작할 수 있어 초당 67필드(33 프레임)의 영상에 대해 워터마킹과 압축을 실시간으로 수행할 수 있었다.

Key Words : DWT, Blind Watermarking, CODEC, Design, Video, FPGA

### ABSTRACT

In this paper, we proposed a hardware(H/W) structure which can compress and reconstruct the input image in real time operation and implemented it into a FPGA platform using VHDL(VHSIC Hardware Description Language). All the image processing element to process both compression and reconstruction in a FPGA were considered each of them was mapped into H/W with the efficient structure for FPGA. We used the DWT(discrete wavelet transform) which transforms the data from spatial domain to the frequency domain, because use considered the motion JPEG2000 as the application. The implemented H/W is separated to both the data path part and the control part. The data path part consisted of the image processing blocks and the data processing blocks. The image pr-

\* 광운대학교 전자재료공학과 Digital Design & Test Lab.(ddntlab.kw.ac.kr, design@kw.ac.kr)

\*\* 한양대학교 응용수학과

\*\*\* 광운대학교 전자공학과

논문번호 : 030408-0917, 접수일자 : 2003년 9월 17일

\*\*이 논문은 한국과학재단 목적기초연구(과제번호 : R01-2001-000-00350-0(2003))의 일부 지원으로 이루어졌음.

rocessing blocks consisted of the *DWT Kernel* for the filtering by DWT, *Quantizer/Huffman Encoder*, *Inverse Adder/Buffer* for adding the low frequency coefficient to the high frequency one in the inverse DWT operation, and *Huffman Decoder*. Also there existed the interface blocks for communicating with the external application environments and the timing blocks for buffering between the internal blocks. The global operations of the designed H/W are the image compression and the reconstruction, and it is operated by the unit of a field synchronized with the A/D converter. The implemented H/W used the 69%(16980) LAB(Logic Array Block) and 9%(28352) ESB(Embedded System Block) in the APEX20KC EP20K600CB652-7 FPGA chip of ALTERA, and stably operated in the 70MHz clock frequency. So we verified the real time operation of 60 fields/sec(30 frames/sec).

## I. 서론

2차원 DWT(Discrete Wavelet Transform)에 대한 연구 중 가장 많은 비중을 차지하는 부분이 DWT를 수행하는 순서나 DWT 연산방법을 변경하고자 하는 것이다. 먼저, 1차원 DWT를 2차원으로 확장하여 분리 가능한(separable) 2차원 DWT(2D DWT) 방법과 행과 열의 변환이 분리 불가능(non-separable)한 방법으로 나누어 볼 수 있다 [1][2][3][4]. 분리 불가능한 알고리즘은 복잡성 때문에 많은 레지스터를 사용하여야 하고 스케줄링이 복잡하여 근본적으로 분리 가능한 방법들에 비해 많은 재원을 필요로 한다. 이러한 레벨 단위의 필터링 방식에 대한 연구들 이외에 필터링 과정에 적응성을 부여한 H/W에 대한 연구[5][6], IP화 하고자 하는 노력[6], 그리고 FPGA에 사상하는 방법[7]들도 연구되었다.

최근 들어서는 리프팅(lifting) 기법을 이용한 H/W구현 연구가 많이 이루어지고 있는데, 리프팅 기법이 컨벌루션에 기반한 필터링 기법에 비해서 소프트웨어적인 메모리량의 감소와 메모리에 대한 접근 횟수의 감소를 가져오고 웨이블릿 변환과 역변환이 동일한 구조로 이루어진다는 장점을 가지고 있다. 그러나 웨이블릿 필터가 가지는 비인과성에 인과성을 부여하는 과정에서 리프팅 기법은 필터링 방식에 비해 부가적인 지연이 발생하고 동기화가 복잡하며 이를 위해 ad-hoc FIFO(First Input First Output)가 요구되는 단점을 가진다. 그리고 주 연산을 수행하는 커널의 확장성과 프로그래머블한 특성을 부여하는 것도 리프팅 방식이 불리하고 경계처리 역시 리프팅 방식이 필터링 방식에 비해서 복잡하다. 마지막으로 H/W 구현 시 외부 메모리로의 접근 횟수는 리프팅 기법이 작지만 내부 메모리의 접근 횟수는 많고 데이터 관리가 복잡한 단점을 가

진다. 따라서 구현하고자 하는 H/W의 특성에 따라서 리프팅 기법과 필터링 기법을 선택하고 적절하게 구현해야 하는데 Mattaveli[9]는 이러한 것들을 고려하여 H/W의 유연성과 프로그래머블 특성을 위해 필터링 기법을 사용하는 H/W구조를 제안하였다.

JPEG2000을 비롯하여 영상 및 비디오 데이터들을 처리하는 다양한 국제 표준들이 사용되고 있는데, 이들은 결국 디지털 영상/비디오 신호들을 정보를 보유할 수 있는 형태로 가공하여 다양한 형태의 영상/비디오 콘텐츠로 제작된다. 이와 같은 디지털 정보, 특히 영상/비디오 정보에 대한 보호 및 보안 방법으로 최근 가장 많은 연구가 되고 있는 기술들이 디지털 워터마킹이다. 워터마킹은 콘텐츠의 불법적인 사용 및 도용이 있는 경우 실 소유자를 판별하는 소유권 인증 등의 분야에서 사용되는 기술로, 영상/비디오가 실제로 사용될 때는 워터마크가 삽입되어 있는지를 알 수 없고, 영상/비디오 콘텐츠에 워터마크를 제거하거나 변경하기 위한 공격이 가해졌을 경우 그 공격에 대한 강인성(robustness)을 가지는 것을 근간으로 하고 있다. 2D DWT를 기본적인 주파수 변환기법으로 사용하는 JPEG2000[10]이라는 새로운 정지영상 압축에 대한 국제 표준안이 제정되면서 2D DWT 기반의 워터마킹의 연구는 더욱더 중요성을 띄게 되었다. JPEG2000을 기반으로 하는 워터마킹 방식의 경우는 영상에 대한 압축과정을 동시에 고려하므로 워터마킹의 삽입에 있어서 다양한 양자화기에 대한 적응성을 가져야 한다. 따라서 최근 들어 양자화를 고려한 워터마킹 기법이 많이 연구되고 있는데 크게 워터마크를 삽입할 계수를 양자화시켜 워터마크를 삽입하는 방식[11]과 양자화기 자체를 고려하여 양자화를 거친 양자화 인덱스 결과를 변조하여 워터마킹을 수행하는 방식 [12] 등이 연구되어왔다. 또한 최근 들어 워터마킹이 다양한 분야에 적용됨에 따라서 소프트웨어로 구현하던 방식에서 나아가서 DSP 혹은 임베디드

프로세서에 내장시키는 방식[13], 그리고 직접 H/W로 구현하여 2D DWT 기반의 영상처리 프로세서 혹은 코덱과 함께 실시간으로 동작시키고자 하는 연구가 진행되었다[14][15]. [23]에서도 비디오를 위한 워터마킹 방식이 제안되었는데 3 레벨의 DWT 후에 최저주파수 대역에 워터마크를 삽입하는 방식으로 다수의 프레임들에 인터리빙 방식의 워터마킹을 수행한다. 1과 0의 워터마크 비트들은 "1100"과 "0100"으로 인코딩되고 최저주파수 대역의 LSB(Least Significant Bit)들과 치환된다. 이 방식은 다른 방식들과 유사하게 DWT를 이용한 압축과정을 직접적으로 고려한 것은 아니고 단지 워터마킹을 위해서 DWT를 이용한 것이다.

본 논문에서는 2차원 DWT를 이용하는 영상압축 및 복원 H/W를 최소의 H/W 자원을 가지면서 FPGA에서 실시간 동작이 가능하도록 설계하고자 한다. 또한 2D DWT 변환을 이용한 영상압축 H/W에 적합한 블라인드 워터마킹 방식을 제안하고 VHDL을 이용해서 H/W로 구현한다.

본 논문의 2장에서는 전체적인 H/W의 구조와 사양에 대해서 설명하고 3장에서는 워터마킹 알고리즘에 대해서 나타낸다. 4장에서 2장에서 언급된 H/W 블록들에 대한 세부적인 구조를 살펴보고 IV장에서는 영상 압축을 위해 어떠한 방식으로 H/W가 동작하는지 설명한다. 그리고 V장에서 실험 결과 및 구현결과를 보이고 마지막으로 VI장에 결론과 향후 연구계획을 밝힌다.

## II. 영상처리 H/W의 전체적인 구조와 사양

그림 1에 전체적인 구조를 나타냈는데 동작적인 역할에 따라서 크게 데이터 패스부와 제어부로 구분되고 데이터 패스부는 영상 처리블록과 데이터처리 블록으로 나누어진다. 영상처리 블록은 2차원 DWT를 위한 필터링을 수행하는 DWT 커널부(DWT Kernel), 웨이블릿 계수의 분포를 이용하여 구현된 양자화기(Quantizer)와 허프만 인코더(Huffman Encoder), 역 DWT 변환 후에 저주파와 고주파 계수를 더하고 시간적인 타이밍을 조절하는 역변환 덧셈기/버퍼(Inverse Adder/Buffer), 그리고 역변환과정에서 허프만 코드를 양자화 계수로 복원하는 허프만 디코더(Huffman Decoder)로 구성된다. 양자화기와 허프만 인코더 사이에 워터마크(Watermarker)와 워터마크 레지스터(Watermark Register)가 존재하여 양자화 인덱스(Index)들에 대

한 워터마킹을 수행한다. 또한 그림 1에서 앞서 설명한 블록들 이외의 블록들은 데이터처리 블록에 해당하고 표준 입출력과 통신하기 위한 입력 인터페이스(Input Interface)와 PCI 인터페이스(PCI Interface), 외부 메모리와의 통신을 위한 메모리 버퍼(Memory Buffer), 커널과 메모리 버퍼사이의 시간적 완충과 데이터 재 정렬을 위한 KtM 버퍼(Kernel-to-Memory Buffer), 그리고 가변길이의 허프만 코드를 일정한 길이로 만드는 출력 버퍼(Output Buffer)들로 구성된다.

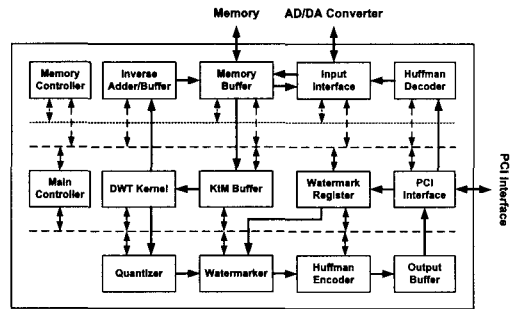


그림 1. H/W의 전체적인 구조  
Fig. 1. Global architecture of the H/W

표 1. 제안된 코덱의 설계 사양  
Table 1. The design specification of the implemented coded

Item	Specification
Image size	640×240/field
Image format	NTSC Y:Cb:Cr(4:2:2)
Compression ratio	about 45:1
PSNR	about 30dB
Data format	DWT coeff. : 16-bit(9,7) Filter coeff. : 12-bit(2,10)
Operation speed	67 field(33 frames)/sec
2D DWT depth	4-Level
DWT filter	Daubechies (9,7)
Quantization	Linear scalar with exception region
Entropy coding	Huffman coding
Memory	External memory (SDRAM)
Image input	Encoding Output of image A/D converter (Bt829b)
	Decoding 32-bit data from PCI interface
Image output	Encoding 32-bit data to PCI interface
	Decoding Input of D/A converter(Bt866)
Output format	32-bit PCI bus (using PLX9054)

제안된 H/W는 A/D 변환기에 맞추어서 NTSC 방식의 영상을 필드단위로 처리하게 되는데 이러한 컬러 영상에 대해서 초당 30프레임의 처리를 통한 실시간 동작을 할 때 약 45대 1의 압축률에서 약

30dB의 화질을 가진다. 내부적으로 웨이블릿 계수는 16비트(정수 9비트, 소수 7비트)를 사용하고 필터 계수는 12비트(정수 2비트, 소수 10비트)를 사용한다. Daubechies (9,7) 필터를 Booth 인코딩된 형태로 내장하여 4 레벨의 DWT를 수행하고 주파수 영역으로 변환한 후에 선형 양자화와 허프만 코딩을 이용하여 영상을 압축한다. 아날로그 신호의 처리를 위해 상용 IC(Bt829b, Bt866)를 사용하며 컴퓨터와 PLX9054을 이용한 PCI 통신을 한다. 이러한 설계 사양을 표 1에 나타냈다.

### III. 워터마킹 알고리즘

본 논문에서는 두 가지 워터마크 삽입 방식을 제안하는데 그림 3에 워터마크 삽입 방식을 나타내었다. 선택된 비트평면(Bitplane)에 정규적(Regular mode)으로 혹은 몇 개의 비트평면에 대해 무작위적(Random mode)으로 비트평면을 치환하여 워터마크를 삽입하는 것으로 이와 같은 방식을 사용한다면 삽입된 워터마크의 위치정보를 정확히 알고 있기 때문에 워터마크가 삽입된 영상에 대해 조작이 있을 경우에 조작된 위치를 판별할 수 있다.

```

Define :
WCLLa : wavelet coefficient in LLa subband
Mode : random, regular
Bitplane : target bit position for watermarking
CipherKey : parallel input for LFSR
Watermarking_algorithm
(WCLLa[x][y], Mode, Bitplane, CipherKey) {
for(x:0~HeightLLa) {
for(y:0~WidthLLa) {
if (Mode=Regular) {
WCLLa=BitChangeFunction(WCLLa, Bitplane)
}
else if (Mode=Random) {
Bitplane=LFSR(CipherKey)
WCLLa=BitChangeFunction(WCLLa, Bitplane)
}
}
}
}
    
```

그림 2. 비트평면 교환 방식의 워터마킹  
Fig 2. Watermarking by exchange of bitplane

2D DWT 후에 최저주파수 부대역내의 웨이블릿 계수는 모두 양의 값을 가지고 그 값의 범위는 0에서 255이다. 즉, LL4 부대역의 계수는 8비트로 표현가능하고, 일반적인 웨이블릿 기반의 영상압축에서 전체 영상에 대해 80%이상의 에너지 정보를 보유한 LL4 부대역에 대한 주파수 정보를 모두 보존

하기 때문에 그림 3의 비트평면 치환방식이 가능하다. 워터마크 삽입 시 압축된 영상 화질의 저하를 최소로 하면서 강인성을 지닐 수 있는 비트평면조합을 LL4 부대역에서 선택하고 이를 워터마크 삽입 포인트로 결정한다.

표 2. 비트평면 정규적 교환방식에 대한 PSNR 결과  
Table 2. PSNR result by watermarking(W) using bitplane(B) regular changing

B	PSNR(dB) before W	Compression Ratio	PSNR(dB) after W	Error Ratio(%) (Quality 4)
1	32.3	30:1	32.29	6.74
2			31.71	3.22
3			30.06	2.44
4			27.12	0.29
5			24.97	0
6			21.32	0
7			17.55	0

표 3. 비트평면 무작위 교환방식에 대한 PSNR 결과  
Table 3. PSNR result by watermarking(W) using bitplane(B) random changing

B	PSNR(dB) before W	Compression Ratio	PSNR(dB) after W	Error Ratio(%) (Quality 4)
1~2	30.3	30:1	31.99	0.59
1~3			31.24	0.49
1~4			29.76	0.39
2~3			30.77	0.39
2~4			29.17	0.20
3~4			28.48	0.16

최저주파수 부대역에서의 비트평면 치환에 의한 워터마킹으로 영상의 질이 현저히 감소한다면 이 방식은 사용할 수 없기 때문에 적절한 비트평면의 조합을 결정해야한다. 따라서 최저주파수 부대역에서 비트평면의 위치를 변경하며 워터마크를 삽입하고 압축된 영상과의 PSNR과 시각적인 인지도를 고려하여 워터마크의 대상 비트평면으로 결정하였다. 이 결과를 표 2와 3에 나타내었는데 각각 비트평면 정규적 교환방식과 비트평면 무작위 교환방식에 대한 결과를 나타낸다. 표 2와 3의 결과와 시각적인 특성을 고려하여 비트평면 정규교환의 경우는 LSB 비트 평면에서 3번째 비트평면(bitplane 2)을 선택하였고 비트평면 무작위교환의 경우는 2에서 3번째 비트평면(bitplane 1~2)을 선택하였다. 비트평면 정규교환의 실험에서 3번째 비트평면을 교환한 PSNR 결과와 4번째 비트평면을 교환한 PSNR 결과의 정량적인 차이는 1.56dB로 크지 않지만 시각적으로는 많은 차이를 보이기 때문에 3번째 비트평면을 워터

마크 삽입 영역으로 설정하였다. 비트평면 무작위이고 환도 같은 이유에서 앞서 언급한 것과 같이 결정하였다.

#### IV. 영상처리 H/W의 내부 구조

본 장에서는 앞에서 설명한 전체적인 H/W를 구성하는 각각의 블록에 대한 구조와 역할에 대해서 설명한다.

##### IV-1. DWT 커널의 H/W 구조

DWT 커널은 웨이블릿 변환과 역변환을 위한 컨벌루션 형태의 필터링을 수행하는 블록이다. 메모리에 저장된 필드 단위의 영상데이터는 메모리 버퍼와 KiM 버퍼를 거쳐 DWT 커널로 입력되고 필터링이 수행된다. 레벨단위의 필터링 동작과 함께 최종적인 웨이블릿 계수는 양자화기/허프만 인코더 블록을 통해 압축된 비트열이 된다.

그림 4에 나타난 것과 같이 DWT 커널은 커널 셀(Kernel Cell)의 병렬적 확장으로 구성되는데, 커널 셀은 FPGA의 ESB를 이용한 내부 듀얼-포트 램(Dual-port RAM)으로 구성된 RAM 체인(RAM Chain), 32비트 CLA(Carry Look Ahead Adder)로 구성된 선덧셈(Pre-Adder), 그리고 누적 기능을 가진 하이브리드 CSA 트리(Hybrid Carry Save Adder tree)와 Booth 곱셈기 및 CLA로 구성된 MAC(Multiplier and Accumulator, MAC) 열의 구조를 갖고 있다[17]. 또한 커널 셀의 입력을 위한 다중 쉬프트(Multi-Shifter)로 이루어진 프리-버퍼(Pre-buffer)가 입력단을 구성한다.

본 논문에서 구현하고자 하는 H/W의 내부 연산 블록들은 33MHz의 기준 주파수를 사용하는 것을 가정하는데 이 경우 초당 30 프레임(60 필드) 이상의 성능을 보이기 위해서는 1개의 커널 셀로는 불가능하다. 또한 기존 연구에서와 같이 많은 수의 곱셈기와 덧셈기를 사용할 경우 H/W의 양이 과다해질 뿐만 아니라 불필요한 성능(하나의 A/D 변환기에 대해 30 필드 이상의 성능)을 가질 수도 있는데 A/D 변환기의 출력이 25 프레임(PAL)에서 30 프레임(NTSC)임을 감안하면 그 이상의 성능은 불필요하고 H/W의 양만을 증가시켜 비용의 손실을 가져온다. 따라서 최소의 H/W 자원을 사용하면서 실시간성을 가지도록 4개의 커널 셀을 사용하는 DWT 커널구조로 설계하였다. 다중 채널을 위해 더 높은 필터링 성능이 요구된다면 그림 3에 나타난

것과 같이 병렬적으로 커널 셀을 확장하여 선형적으로 성능의 향상을 가져올 수 있다.

MAC은 [18]에서 제안한 구조를 변형하여 DWT 필터링을 위한 새로운 구조의 MAC을 구현하였다. 구현된 MAC은 1의 보수 형태로 DWT 필터 계수를 미리 부스 인코딩하여 ROM에 저장하고 있어서 필터링 연산 시 속도를 증가시킬 수 있고 축적 과정을 위한 별도의 덧셈기나 저장공간이 필요없는 CSA 구조를 가지고 있다. CSA는 HA(Half adder), FA(Full adder) 그리고 2-bit CLA(Carry lock-ahead adder)로 구성된 하이브리드 형태[18]로써 고속으로 부분곱과 축적 연산을 수행할 수 있다.

Y, Cb, 그리고 Cr의 컬러 색차성분을 가지는 영상 데이터를 웨이블릿 필터링 할 경우 프리-버퍼는 각각 Y, Cb, 그리고 Cr 화소 성분을 위한 것으로 DWT 필터링의 동작 단계에 따라 해당 화소나 웨이블릿 계수들을 입력받는다. RAM 체인, 선덧셈기, 부분곱 생성기(PP generator) 및 MUX, CSA 트리 그리고 최종 덧셈기까지 5단계의 파이프라인 단계를 거쳐서 하나의 화소에 대한 DWT 계수를 생성한다. 또한 하나의 DWT 계수를 만드는데 5번의 누적과정을 거치기 때문에 첫 DWT 계수가 출력되는데는 9 클럭이 소요되고 다음 계수부터는 5 클럭마다 하나씩 출력되어 전체적으로 DWT 커널은 5클럭마다 4개의 계수가 출력된다.

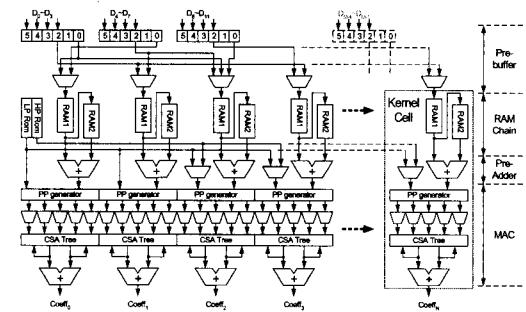


그림 3. DWT 커널과 커널 셀의 H/W 구조  
Fig 3. H/W architectures of DWT kernel and kernel cell

##### IV-2. 양자화기/허프만 인코더의 H/W 구조

본 논문에서는 H/W의 복잡도 및 실시간 동작을 고려하여 양자화기는 선형 스칼라 양자화기를 사용하고 엔트로피 코딩은 허프만 코딩만을 이용한다. 허프만 코딩은 양자화 영역의 발생 빈도수로부터 허프만 코드를 발생시켜 두 과정을 하나의 통합된

구조로 결합하였고, 그 결과의 H/W 구조를 그림 4에 나타내었다. DWT 커널이 출력하는 계수들의 부대역 정보에 따라 양자화기가 두 부분으로 나누어져 있고, FIFO에서 정렬되어 허프만 코드값과 비트 정보 등을 출력한다. 양자화기 및 허프만 코더는 비교기와 ROM 테이블로 이루어져 있는데, 비교기에 의해 ROM의 주소가 결정되면 ROM으로부터 허프만 코드("huff\_code")와 유효 비트를 나타내는 비트 정보("bit\_info")가 출력되고 이들과 함께 DWT 계수("wavelet\_coeff")가 FIFO를 거쳐 출력된다. 5 클럭마다 DWT 커널로부터 4개의 병렬 출력이 발생하므로 이를 순차적으로 처리하기 위해 FIFO가 시간적인 완충작용을 한다.

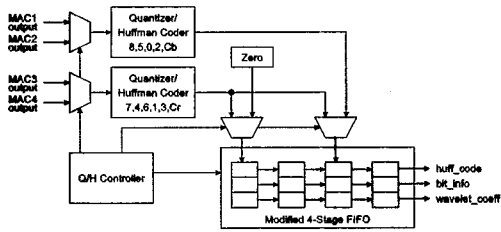


그림 4. 양자화기/허프만 인코더의 H/W 구조  
Fig. 4. H/W architecture of quantizer/Huffman encoder

### IV-3. 역양자화기와 허프만 디코더의 H/W 구조

영상복원의 경우에서도 허프만 디코더는 역양자화기와 결합된 형태를 가지는데 그림 6에 보인 것과 같이 허프만 디코딩 결과로 16비트의 양자화 계수("Coefficient[15:0]")를 출력한다. 16비트 두 개의 입력 포트로부터 32비트 단위의 입력 데이터를 받아 코드경계 검출기(Delimiter detector)에 의해서 부대역 정보를 비롯한 영상 데이터를 추출하고 직렬 쉬프터(HD shifter)를 통해 직렬 데이터를 발생시켜 허프만 디코딩을 수행한다. 디코딩 시 예외 영역에 대한 정보가 검출되면 그 후의 4개의 직렬 데이터를 웨이블릿 계수로 변환하여 출력하는데, 양자화 시 빈도가 낮으면서 큰 값을 가지는 영역을 예외 영역으로 설정하고 허프만 코드와 함께 4비트의 계수 정보를 함께 포함시켰기 때문이다. 최저 주파수 대역의 데이터가 입력되면 허프만 디코딩을 거치지 않고 직렬 데이터들을 곧바로 웨이블릿 계수로 변환하여 출력하게 되는데, 이는 압축 과정에서

최저 주파수 대역의 정보를 정수부만 취하여 거의 무손실 압축을 하였기 때문이다.

허프만 코딩과 같은 VLC의 하드웨어 구조에 대한 연구도 지금까지 계속되어 왔다. 코딩 특성에 따라서 크게 두가지 방식으로 구분되는데 tree-based 방식[19][20]과 group-based 방식이 그것이다. 본 논문에서 구현된 허프만 디코딩 구조는 FSM[21]을 이용한 tree-based 방식으로 허프만 코드 생성 트리와 동일한 순서로 비트를 추적하면서 디코딩을 수행한다. Tree-based 방식은 압축된 데이터에 대해서 고정된 허프만 코드를 사용할 경우에 고속처리가 가능하다. 반면에 tree-based 방식의 VLC는 디코딩 시간이 가변인 단점이 있지만 영상 복원과정이 필드단위의 파이프라인 방식(그림 14 참조)으로 이루어질 경우 문제가 되지 않는다.

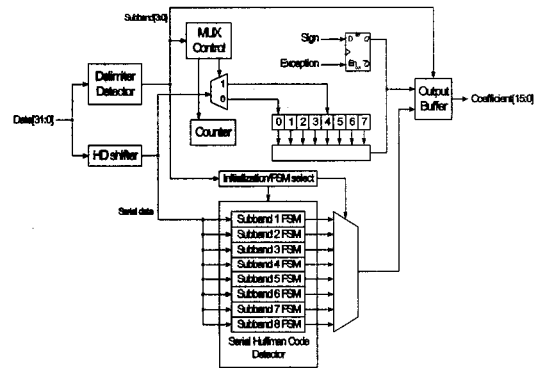


그림 5. 허프만 디코더와 역양자화기의 H/W 구조  
Fig. 5. H/W architecture of Huffman decoder and de-quantizer

### IV-4. 역변환 덧셈기/버퍼의 H/W 구조

영상의 복원과정에서 웨이블릿 역변환을 위한 필터링 수행 시 고주파 성분의 부대역과 저주파 성분의 부대역에 대한 필터링 결과를 더해야 복원된 웨이블릿 계수가 추출된다. 따라서 역변환을 위한 덧셈기와 그에 따른 시간적 완충이 요구되는데 이를 역변환 덧셈기/버퍼에서 수행하며, 그 구조를 그림 6에 나타냈다. 영상 압축과정에서는 이 덧셈과정이 필요 없으므로 그냥 지나친다. 각 변환 레벨("Level[3:0]")에 해당되는 제어신호("sel1", "sel2")에 따라서 "MACx\_output"들의 조합을 이루어 더하거나 그냥 지나치게 하고 각 레지스터(Reg)에 저장하여 결과를 출력한다.

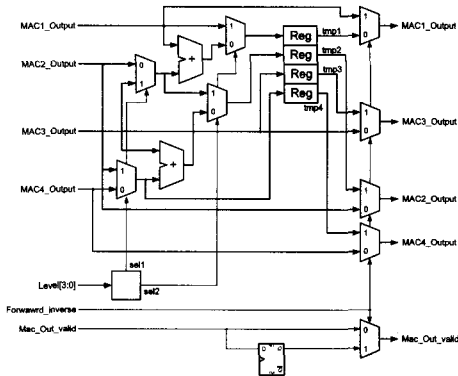


그림 6. 역변환 덧셈기/버퍼의 H/W 구조  
Fig. 6. H/W architecture of inverse adder/buffer

IV-5. 워터마킹 블록의 H/W 구조

그림 7은 그림 1에서 나타낸 워터마킹 프로세서의 자세한 구조를 나타낸다. 먼저 워터마크는 입력 레지스터(Input Register)를 통해서 워터마크 레지스터(Watermark Register)에 저장된다. 워터마크 자체를 보호하기 위해 워터마크를 암호화할 경우에 워터마크는 암호시스템(Cryptosystem)에 의해서 암호화되고 케환 라인(Feedback line)을 통해서 다시 워터마크 레지스터에 재 저장된다. 저장된 워터마크는 계수(Coeff[7:0])와 동기되어 워터마크(Watermarker)에서 워터마킹을 수행한다. 그와 함께 다른 신호들의 시간적인 병렬성을 유지하기 위해 버퍼(Buffer) 블록을 이용해서 파이프라인 동작을 유지한다. 암호화에 사용된 키는 LFSR의 초기값으로도 사용되어 워터마킹 과정에 보안성을 추가한다.

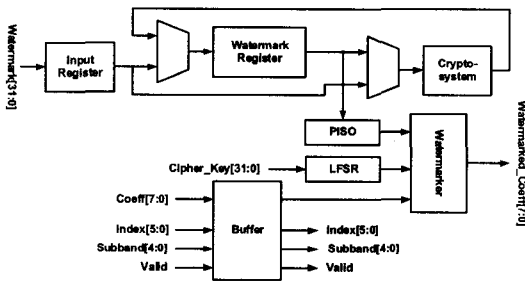


그림 7. 워터마킹 프로세서의 전체적 구조도  
Fig. 7. Detail architecture of the watermarking block

V. 영상처리 H/W의 동작

본 장에서는 구현된 H/W의 압축과 복원을 위해 전체적으로 어떠한 과정을 거치는지를 나타내고 세부적인 동작 흐름에 대해서 설명한다.

제안된 H/W는 A/D 변환기의 필드 표시신호에 동기하여 전체적인 동작이 제어되는데 영상압축 과정과 복원 과정에 대한 동작 순서를 그림 8과 9에 각각 나타냈다. 그림 8과 9에서 보이듯이 15ms 단위의 필드단위로 영상을 처리한다. 영상압축기의 데이터 처리량은 A/D 변환기에서 출력되는 영상 데이터량에 의존하기 때문에 동작의 시작과 처리 시간을 A/D 변환기에 동기시킨다. 그림 8의 영상압축 과정은 크게 필드 단위의 영상을 메모리에 저장하는 과정(Even or Odd Field Store)과 영상을 압축하는 과정(Even or Odd Field Compression), 그리고 압축된 데이터를 출력(Even or Odd Field Q/H Hard disk Store)하는 과정으로 구성되는데, 이 세 가지 과정은 전체적으로 파이프라인 방식의 동작을 통해 실시간으로 영상을 처리한다. 또한 그림 9의 영상복원 과정은 압축된 데이터를 입력받아 허프만 디코딩하는 과정(Even or Odd Field HD and Store)과 복원 과정(Even or Odd Field De-compression), 그리고 복원된 영상을 출력하는 과정(Even or Odd Field D/A Converting)으로 구성되고 영상압축과정과 마찬가지로 전체적인 파이프라인 동작을 이룬다.

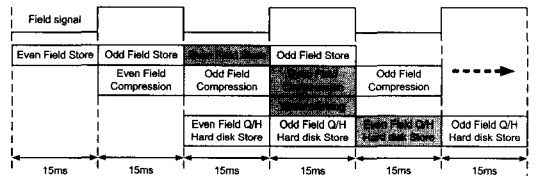


그림 8. 영상 압축과 워터마킹 과정의 파이프라인 동작  
Fig. 8. Pipeline operation of image compression

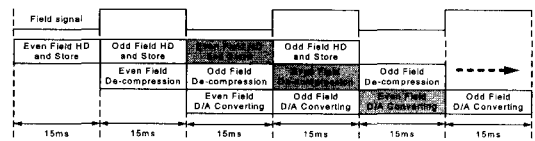


그림 9. 영상 복원과정의 파이프라인 동작  
Fig. 9. Pipeline operation of image reconstruction

VI. 제안된 H/W의 구현결과

본 장에서는 워터마킹에 대한 실험결과와 H/W로 구현 결과를 나타낸다.

VI-1. 워터마킹 실험 결과

본 논문에서는 512×512 화소의 원영상을 사용하였고 4 레벨 DWT를 수행한 결과의 영상을 대상 대상으로 하였기 때문에 최저주파수 대역인 LL4는 32×32(1024개) 개수의 계수를 가진다. 따라서 워터마크도 32×32의 이진 영상으로 사용하였다. 제안된 워터마킹 알고리즘은 일반적인 영상의 조작에 대해 LL4가 가지는 강인성과 거의 유사한 강인성을 가지는데 그 결과를 표 4에 나타냈다. 표 4에서 왼쪽은 비트평면 정규교환 방식에 대해 JPEG 압축, blurring 그리고 sharpening 등과 같은 일반적인 영상 처리에 대한 강인성을 나타내었고 오른쪽은 같은 실험에 대해 비트평면 무작위교환 방식에 적용한 결과이다. JPEG 압축의 경우 두 방식 모두 JPEG quality 4부터 워터마크에 손상이 발생하였고 JPEG 압축은 PhotoshopTM의 기능을 사용하였다.

워터마크가 삽입된 Lena의 압축 영상을 보이고 있는 그림 10의 (a)와 같이 영상의 국부적 영역에 대한 인위적 공격이 가해졌을 경우 (b)와 같이 검출된 워터마크로부터 공격 위치를 판별할 수 있는데 워터마크가 값의 변조방식에 의해 삽입되는 것이 아니라 특정 위치의 치환을 통해 삽입되기 때문에 쉽게 공격 위치를 판별할 수 있는 장점을 가진다. 즉, 추출과정을 통해 재구성된 워터마크의 손상정보를 이용해서 다른 처리나 부가적인 과정을 거치지 않고 직접적으로 에러 위치를 판별한다.



그림 10. 공격에 대한 위치판별 영상결과 (a) 워터마크삽입 후 인위적으로 조작된 영상 (b) 위치 판별을 위한 검출된 워터마크  
 Fig. 10. Detection of error position after attack (a) fabricated image after watermarking (b) extracted watermark for detection of error position

본 논문에서 제안된 워터마킹 방식과 가장 유사한 형태인 [16]과 결과를 비교하도록 한다. 1장의 서론에서 언급한 것과 같이 [16]에서 제안된 방식은 각 프레임 단위를 개별적인 영상으로 취급하는 방식이 아니다. 즉, 비디오 영상에 대한 압축과정 등의 처리를 고려하지 않고 단지 워터마킹을 위해서만 DWT를 사용하였다는 차이점을 가진다. [16]은 MPEG2 공격에 대해서 워터마크 검출 어려움은 13.8대 1의 압축율일 경우에 1% 미만인데 표 5의 공격강인도(Attack strength) 4에서 6 사이에 해당하므로 본 논문에서 제안된 방식이 실험결과에서 뒤지지 않음을 확인할 수 있다. 이 외에 다른 공격에 대해서는 [16]에서 나타내고 있지 않으므로 비교할 수가 없다. 또한 본 논문에서 제시된 알고리즘은 공격에 대한 위치를 판별할 수 있는 장점을 가지고 있고 영상압축과정과 함께 고려된 알고리즘으로써 H/W로 구현되었다는 특징도 가진다.

표 4. 비트평면(3번째 비트평면) 정규교환 워터마킹에 대한 공격 후 어려움

Table 4. Error ratio by various attacks in watermarking using bitplane regular changing

Attack Strength	Regular Selection (bit-plane 2)			Random Selection (bit-plane 1-2)		
	# of Error Pixels	Error Ratio(%)	PSNR (dB)	# of Error Pixels	Error Ratio(%)	PSNR (dB)
JPEG Compression Attacks						
10	0	0	31.8	0	0	31.9
8	0	0	31.7	0	0	31.9
6	0	0	31.7	0	0	31.9
4	20	2	31.6	25	2.4	31.7
2	32	3.1	31.3	90	8.8	31.5
0	92	9.0	30.9	211	20.6	31.0
Blurring						
Weak	0	0	0	0	0	32.4
Strong	2	0.2	0.2	7	0.7	32.4
Sharpening						
Weak	0	0	0	0	0	32.0
Strong	6	0.6	0.6	10	0.9	31.7

VI-2. H/W 구현 결과

2, 4장을 통해 소개한 H/W구현을 위해 Altera의 Quartus II 환경을 사용하고 VHDL을 이용하여 설계하였다. 타겟 플랫폼으로 이용된 FPGA는 Altera의 APEX20KC AEP20K600CB652-7칩이다. 그림 11에 Quartus II 환경의 그래픽 에디터(GDF)로 설계된 전체 블록도를 나타냈다. 그림 1과의 비교를 용이하게 하고자 블록들을 동일하게 표기하였고 유사하게 배치하고자 하였다. 단, 양자화기(Quantizer)와 허프만 인코더(Huffman Encoder)가 분리되어 있는 것을 볼 수 있는데 이것은 양자화기의 양자화 인덱스를 관찰하기 위해 분리한 것으로 동작 혹은 구조가 변화한 것은 아니다.

구현된 H/W는 33MHz(30ns의 주기)의 기본 주



과수에서 동작하여 초당 약 67필드(33프레임)의 영상을 압축할 수 있기 때문에 실시간 동작이 가능하다. 이 때 메모리는 약 100MHz 클럭(99MHz, 기본 동작주파수의 3배 클럭 속도)에 의해서 동작해야 하는데 두 클럭간의 동기를 위해서 APEX FPGA 칩이 내부적으로 제공하는 클럭 가속(clock boosting) 기능을 이용하였다. 그리고 입력 인터페이스는 A/D 변환기(BT829b)의 동작 주파수인 28.63636MHz에 맞추어서 동작한다.

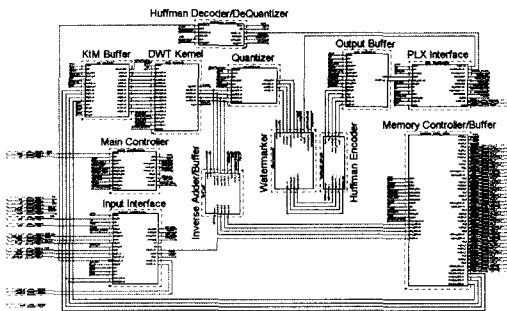


그림 11. Quartus II를 이용한 GDF 설계 결과  
Fig. 11. GDF design result using Quartus II

표 5에 그림 1과 11의 각 기능 블록들이 FPGA 칩에 사상되었을 때 사용된 H/W 자원 사용율을 나타냈다. 전체적으로 타겟 FPGA의 69%(16980개)의 LAB를 사용하고 9%(28352개)의 ESB를 사용한다. 메모리 제어기의 경우 메모리 호출 주소 중 많은 부분을 ROM에 저장하여 사용하므로 다수의 ESB를 사용한다. 또한 커널 내부에서 커널을 제어하는 커널 제어부도 ROM 저장 방식의 제어기를 사용하므로 ESB를 사용한다. 각 기능 블록들이 내부의 제어기를 따로 가지고 있기 때문에 전체 제어부(Main Controller)의 자원 사용률이 상대적으로 낮다. 즉, 전체 제어부는 모든 블록들의 내부 제어부들의 전체적인 동작 순서만을 결정한다. 출력 버퍼 블록은 비교적 많은 H/W 자원을 사용하는데, 이는 허프만 코더에서 불규칙한 크기의 데이터를 축적하였다가 32 비트로 만들어 출력시키는 부분에서 많은 병렬 레지스터와 쉬프트 레지스터를 사용하고 출력 스트림을 일정 길이만큼 저장하기 때문이다.

구현된 DWT 커널은 [3]에서 제안된 구조와 가장 유사한데 차이점을 표 3에 나타냈다. 제안된 MAC의 구조는 내부적인 피드백 구조를 통해서 필터링 연산을 수행하기 때문에 [3]과 달리 필터 길이에 따른 곱셈기와 덧셈기의 숫자가 무관하다. 그림 3의

구조를 바탕으로 [3]에서 제안된 구조와 동일한 조건에서 비교할 경우의 결과를 표 7에 나타내었다. 여기서는 Daubechies의 (9,7) 필터를 사용하는 것으로 기준을 삼았다. 동일 데이터율과 레벨 처리방식이 적용될 경우에 본 논문에서 제안된 커널의 경우 총 12개의 곱셈기와 24개의 덧셈기를 사용하는 것에 해당하므로 [3]의 방식에 비해서 비교적 적은 H/W 자원을 사용한다. 모든 연산 메모리(working memory)를 외부 메모리를 사용하기 때문에 [3]과 달리 [4]에서 제안된 구조와 동일하게 N2의 storage가 요구되고 N2의 연산 시간(computing time)이 요구된다. 최소화된 H/W 구조를 보상하기 위한 입력 데이터 재 사용율을 높이는 RAM 체인의 제어 복잡성을 가진다. 그러나 [6]과 마찬가지로 최적의 하드웨어 사용율을 보인다.

표 5. 구현된 H/W의 FPGA 자원 사용률  
Table 5. FPGA resource usage of the implemented H/W

Block Name	LAB (24320:100%)	ESB (311296:100%)
DWT kernel	3129 (13%)	9472 (3%)
Quantizer	975 (4%)	0 (0%)
Huffman encoder	283 (1%)	0 (0%)
Output buffer	2405 (10%)	0 (0%)
PCI interface	1433 (6%)	0 (0%)
KitM buffer	1242 (5%)	0 (0%)
Main controller	168 (<1%)	0 (0%)
Huffman decoder/Dequantizer	1154 (5%)	0 (0%)
Watermarker	4037 (16%)	0 (0%)
Input interface	391 (2%)	0 (0%)
Memory controller/Buffer	1501 (6%)	18880 (6%)
Inverse adder/buffer	262 (1%)	0 (0%)
Total	16980 (69%)	28352 (9%)

표 6. 제안된 DWT 커널과 [3]의 성능 비교  
Table 6. Performance comparison of the proposed DWT kernel and [3]

Arch.	Multiplier	Adder	Storage Size	Computing Time	Control Complexity	Hardware Utilization
Ours	12	24	$N^2$	$N^2$	Complex	100%
[6]	36	36	$N^2/4 + KN+K$	$0.5N^2 \sim 0.67N^2$	Simple	100%

본 논문을 통해 구현된 커널을 비롯한 코덱은 워터마킹 프로세서를 비롯하여 영상압축 및 복원과정을 위한 전체적인 H/W를 포함한다. 2D DWT를 기반으로 한 영상압축 코덱 중에서 가장 널리 알려진 것은 아날로그 디바이스사의 ADV611/612[22] 칩으

로 DVR 시스템과 웹캠 등의 다양한 영상 시스템에 사용되고 있다. 제안된 전체적인 영상압축 코덱은 ADV611/612의 구조와 유사성을 가지고 있는데 차이점을 표 7에 나타냈다. 두 H/W 간에 성능면에서 유사성을 가지고 있지만 메모리 사용에 있어서는 상당한 차이를 보인다. 제안된 H/W는 FPGA를 대상으로 하기 때문에 칩내부 메모리를 사용하고 있지 않고 프레임 버퍼와 연산을 위한 메모리, 그리고 출력 버퍼 등을 모두 외부 메모리로 처리한다. 또한 워터마킹이라는 보안 기능을 내장한 것은 큰 특징으로 볼 수 있다.

표 7. 구현된 H/W와 ADV611/612와의 비교  
Table 7. Difference between ours and ADV611/612

Item	ADV611/612	Ours
Performance	> 30 frame/sec	> 30 frame/sec
Maximum Image Size	720×288 (field)	640×240 (field)
Operation Frequency	27MHz	33MHz (Max. 82MHz)
On-Chip Memory	Yes	No
External Memory	256K 16-bit (DRAM)	256K 16-bit×3 (SDRAM)
Internal Output FIFO	Yes (512 Deep 32-bit)	No
Security Facility	No	Yes (Watermarking)

### VII. 결론

본 논문에서는 입력되는 영상 혹은 압축 데이터를 실시간으로 압축하거나 복원할 수 있는 H/W의 구조와 압축되는 영상에 대해 실시간으로 워터마킹을 삽입할 수 있는 워터마킹 프로세서를 제안하고 FPGA로 구현하였다. Motion JPEG2000을 고려하여 공간영역의 데이터를 주파수 영역으로 변환하는 도구로서 DWT를 채택하여 컨벌루션 방식의 확장이 용이한 필터링 구조를 제안하였다. 시각적 특성을 토대로 통계적으로 구성된 양자화와 허프만 코딩과정을 통합된 형태의 H/W로 모두 내장시켰고 영상의 표준 입출력을 처리할 수 있는 입출력 장치들도 내장하였다. 또한 영상압축과 동시에 워터마킹이 가능하고 영상에 대한 소유권과 공격에 대한 위치 판별이 원영상없이 가능한 워터마킹 프로세서를 내장시켰다.

구현된 H/W는 Altera의 APEX20KC

EP20K600CB652-7 FPGA 칩에서 69%(16980개)의 LAB와 9%(28352개)의 ESB를 사용하면서 문제없이 사상되었다. 그리고 최대 82MHz의 클럭주파수에서 안정적으로 동작할 수 있어 기존 클럭으로 가정하였던 33MHz의 속도를 충분히 만족시켰다. 따라서 초당 60필드(30 프레임) 이상으로 영상을 처리할 수 있어 실시간 처리가 가능함을 확인하였다.

JPEG2000을 비롯하여 웨이블릿 변환 기반 제품의 저변이 확대되고 있고 모든 영상관련 시스템에 대한 보안성이 대두되고 있는 시점에서 구현된 하드웨어는 독립적으로 하나의 영상처리 제품이 될 수 있고 SOC(system-on-a-chip)를 구성하는 IP(intellectual property)로도 사용될 수 있을 것으로 사료된다. 또한 향후 본 연구는 단일 형태의 Motion JPEG2000 칩셋으로 더욱 접근하기 위해 EBCOT(Embedded Block Coding with Optimized Truncation), ROI(region of interest), 그리고 적응적인 선형 양자화 알고리즘들을 H/W에 적합하게 개발하여 구현된 H/W의 기능 블록들로 추가할 계획이다.

### Reference

- [1] Trieu-Kien Truong, et al., "A New Architecture for the 2-D Discrete Wavelet Transform", IEEE Int'l Conf. of Communications Computers and Signal Processing, pp. 481-484, 1997.
- [2] Chu Yu and Sao-Jie Chen, "Design of an Efficient VLSI Architecture for 2-D Discrete Wavelet Transform", IEEE Trans. on Consumer Electronics, Vol. 45, No. 1, pp. 135-140, Feb. 1999.
- [3] Ming-Hwa Sheu, Ming-Der Shieh and Sheng-Wet Liu, "A VLSI Architecture Design with Lower Hardware Cost and Less Memory for Separable 2-D Discrete Wavelet Transform", IEEE ISCAS'98, Vol. 5, pp. 457-460, 1998.
- [4] Mohan Vishwanath, Robert Michael and Mary Jane Irwin, "BSLI 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, May 1995.
- [5] Iijin Chen and Magdy A. Bayoumi, "A Scalable Systolic Array Architecture for 2-D Discrete Wavelet Transforms", IEEE Proc. of Midwest Symp. on Circuits and Systems, Vol. 2, pp. 303-312, 1996.
- [6] Kolin Paul, D Roy Chowdhury and P. Pal Chaudhuri, "Scalable Pipelined Micro-Architecture for Wavelet Transform", IEEE Int'l Conf. of VLSI Design, pp. 144-147, 2000.
- [7] Shahid Masud and John V. McCanny, "Wavelet Packet Transform for System-on-Chip Application", IEEE Proc. on ICASSP, Vol. 6, pp. 3287-3290, 2000.
- [8] Ali M. Reza and Robert D. Turney, "FPGA Implementation of 2D Wavelet Transform", IEEE Conf. of Signals, Systems and Computers, pp. 584-588, 1999.
- [9] M. Ravasi, L. Tenze, and M. Mattavelli, "A scalable and programmable architecture for 2-D DWT decoding," IEEE Trans. Circuits Syst. Video Technol., vol. 12, no. 8, Aug. 2002
- [10] Martin Boliek, et al., JPEG 2000 Part I Fina11191 Draft International Standard, ISO/IEC JTC1/SC29 WG1, 24 Aug. 2000.
- [11] Joachim J. Eggers, Jonathan K. Su, and Bernd Girod, "Robustness of a Blind Image Watermarking Scheme", ICIP 2000, Special session on WM, Sep. 2000
- [12] Brian Chen and Gregory W. Wornell, "Quantization Index Modulation : A Class of Provably Good Methods for Digital Watermarking and Information Embedding", IEEE Transaction on Information Theory, vol. 47, no. 4, pp. 1423~1443, May. 2001
- [13] Petitjean, G, Dugelay, A., Gabriele, S., Rey, C., Nicolai, J., "Towards real-time video watermarking for system-on-chip", Multimedia and Expo, 2002. Proceedings. 2002 IEEE International Conference on , Volume: 1, pp.: 597-600, 2002
- [14] Shen-Fu Hsiao, Yor-Chin Tai, Kai-Hsiang Chang, "VLSI design of an efficient embedded zerotree wavelet coder with function of digital watermarking", Consumer Electronics, 2000. ICCE. 2000 Digest of Technical Papers. International Conference on, pp. 186-187, 2000
- [15] Mathai, N.J., Kundur, D., Sheikholeslami, A., "Hardware implementation perspectives of digital video watermarking algorithms", Signal Processing, IEEE Transactions on , Volume: 51 Issue: 4, pp. 925-938, Apr. 2003
- [16] Hongmei Liu, et al., "A Robust DWT-Based Video Watermarking Algorithm", IEEE ISCAS 2002, pp. 631-634, 2002.
- [17] Behrooz Parhami, "Computer Arithmetic : Algorithm and Hardware Desig", Oxford University Press, 2000.
- [18] Fayez Elguibaly, "A Fast Multiplier-Accumulator Using the Modified Booth Algorithm", IEEE Trans. On Circuit and Systems II:Analog and Digital Signal Processing, Vol. 47, No. 9. Sept. 2000
- [19] Y. Ooi, A. Taniguchi, and S. Demura, "A 162Mbit/s variable length decoding circuit using an adaptive tree search technique," in Proc. IEEE 1994 Custom Integrated Circuits Conf., pp. 107-110, May 1994
- [20] R. Hashemian, "Design and hardware implementation of a memory efficient Huffman decoding," IEEE Trans. Consumer Electron., vol. 40, pp. 345-352, Aug. 1994
- [21] V. Iyenger and K. Chakrabarty, "An Efficient Finite-State Machine Implementation of Huffman Decoders", Information Proceeding Letters, Vol. 64, pp. 271-275, 1997
- [22] [http://www.analog.com/Analog\\_Root/productPage/productHome/0%2C2121%2CADV611%2C00.html](http://www.analog.com/Analog_Root/productPage/productHome/0%2C2121%2CADV611%2C00.html)

서 영 호(Young-Ho Seo)

정회원

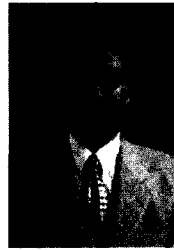


1999년 2월 : 광운대학교  
전자재료공학과 졸업(공학사).  
2001년 2월 : 광운대학교  
대학원졸업(공학석사).  
2000년 3월~2001년 12월 :  
인티스닷컴(주) 연구원.  
2001년 8월 : 광운대학교  
전자재료공학과 졸업  
(공학박사)

2003년 6월~2004년 6월 한국전기연구원 연구원.  
<관심분야> 2D/3D 영상처리/압축, 워터마킹, 암호  
학, FPGA/ASIC 설계  
e-mail : design@kw.ac.kr

유 지 상(Ji-Sang Yoo)

정회원



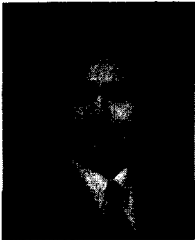
1985년 2월 : 서울대학교  
전자공학과 졸업(공학사)  
1987년 2월 : 서울대학교  
대학원 졸업(공학석사)  
1993년 5월 : Purdue 대학교  
전기공학과 졸업(Ph.D.)  
1993년 9월~1994년 8월 :  
현대전자산업(주) 산전연구소

선임연구원

1994년 9월~1997년 8월 : 한림대학교 전자공학과  
조교수  
1997년 9월~현재 : 광운대학교 전자공학과 부교수  
<관심분야> 웨이블릿 기반 영상처리, 영상압축, 영  
상인식, 비선형 신호처리

김 대 경(Dai-Gyoung Kim)

정회원



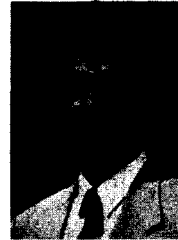
1983년 2월 : 한양대학교  
수학과 졸업(이학사).  
1986년 2월 : 한양대학교  
수학과 대학원졸업(이학석사)  
1994년 8월 : Purdue 대학교  
수학과 졸업(Ph. D)  
1994년 9월~1995년 8월 : 미  
국 Minnesota 대학 응용수

학센터(IMA) Post-doctor

1995년 9월~현재 : 한양대학교 응용수학과 조교수  
<관심분야> 웨이블릿 이론 및 응용, 수치해석학, 편  
미분 방정식

김 동 욱(Dong-Wook Kim)

정회원



1983년 2월 : 한양대학교  
전자공학과 졸업(공학사).  
1985년 2월 : 한양대학교  
대학원 졸업(공학석사).  
1991년 9월 : Georgia 공과대학  
전기공학과 졸업(공학박사).  
1992년 3월~현재 : 광운대학교  
전자재료공학과 정교수. 광운

대학교 신기술 연구소 연구원.

1997년 12월~현재 : 광운대학교 IDEC 운영위원.  
2000년 3월~현재 : 인티스닷컴(주) 연구원.  
<관심분야> 디지털 VLSI Testability, VLSI CAD,  
DSP 설계, Wireless Communication  
e-mail : dwkim@kw.ac.kr