공학박사 학위논문

방송용 콘텐츠 저작권 보호를 위한 비디오 워터마킹 및 암호화의 FPGA 구현에 관한 연구

2010년 2월 25일

부경대학교 대학원

전자공학과

정 용 재

공학박사 학위논문

방송용 콘텐츠 저작권 보호를 위한 비디오 워터마킹 및 암호화의 FPGA 구현에 관한 연구

지도교수 문 광 석

이 논문을 공학박사 학위 논문으로 제출함

2010년 2월 25일

부경대학교 대학원

전자공학과

정 용 재

정용재의 공학박사 학위논문을 인준함

2010년 2월 25일

주　심　공학박사　권 기 룡　(인)

위　원　공학박사　정 완 영　(인)

위　원　공학박사　김 종 남　(인)

위　원　공학박사　류 권 열　(인)

위　원　공학박사　문 광 석　(인)

목 차


PUKYONG NATIONAL UNIVERSITY
부경대학교

표 목 차

그 림 목 차

A Study on FPGA Implementation of Video Watermarking and Encryption for Copyright Protection of Broadcasting Contents

Yong Jae Jeong

Department of Electronics Engineering, Graduate School
Pukyong National University

Abstract

The recent development of computing and communication technologies enabled high definition video (HDV) applications. In today's computing environment, the end users can easily perform storing, editing, reproducing, and or distributing HDV scripts. Copyright protection has therefore been increasingly important to protect HDV data against illegal copy and reproduction and copy right infringement.

Digital watermarking technology has been actively investigated to solve the copyright protection problem of HDV media data by hiding injected watermark information into original digital contents. To be effective, a watermarking scheme must satisfy two important technical requirements: invisibility and robustness. To satisfy the invisibility (or transparency) requirement, the watermark should be injected into the original media contents in such a way that human beings cannot visually recognize the watermark. The effective watermarking scheme should also be robust against possible attacks such as translation, rotation, and/or scaling. In addition a watermarking scheme for HDV applications should be robust against natural attacks such as analog/digital conversion and/or MPEG compression.

The applications of HDV broadcast monitoring and on-air live broadcasting need fast real-time watermarking solution. To enable fast real-time watermarking, chip level implementations of watermarking algorithms have been investigated. VLSI implementations of watermarking algorithms, which are robust against analog/digital conversion and JPEG compression, for still digital images were reported in the literature. These research works focused on single chip implementations of watermarking algorithms but did not address system level studies on the effectiveness and efficiency of underlying watermarking schemes.

Digital encryption technology has been investigated to solve the protection of a copy to another devices in the illegal situation. the method of a copy protection using encryption use a standardized encryption method or a personalized encryption method. the former has strong security level but have not flexibility for adapting the copy protection. the letter has lower security level, but it has flexibility. the standardized encryption is DES, AES, SEED and so on.

In this paper, we present FPGA based hardware system that enables fast real-time watermarking and encryption for SD/HD applications. The presented hardware system has an embedded watermarking or encryption processor implemented based on the STRATIX FPGA device family from ALTERA. The embedded watermarking or encryption processor is based on our novel watermarking and encryption algorithm suitable for optimal hardware implementation. Our system can embed user information such as video meta-data to any selected video frames, so that we can extract the injected watermark data at any frames of the watermarked video. Also our system can encrypt a video information such as pixels information of each video frame. The presented hardware watermarking and encryption system for SD/HD real-time applications satisfies invisibility and robustness in

watermarking that are essential requirements for practical watermarking and satisfies visibility of a video contents which is adapted to our encryption. The presented hardware system based on FPGA can not only be used for live broadcasting with real-time SD/HD watermarking but also used for mass production video watermarking such as movies or DVDs.

Ⅰ. 서 론

근래 인터넷 다운로드 속도가 높아지고, 초고속인터넷 보급률이 높아지면서 고화질의 동영상 콘텐츠의 불법 다운로드가 증가하는 상황이다. 디지털 TV와 DMB(digital multimedia broadcasting)와 같은 디지털 방송 서비스는 신호를 수신 할 수 있는 모든 사람에게 무료로 배포되기 때문에 FTA(free-to-air, 무료 방송 채널) TV로 알려져 있다. 이와 같은 방송용 콘텐츠는 다량의 저작권을 포함한 복합 콘텐츠로 원본 품질로 쉽게 복사 할 수 있으며 인터넷에 장착된 컴퓨터를 통하면 순식간에 전 세계적인 대량 배포가 가능하다. 이런 경우 무료 지상파 디지털 방송에 콘텐츠를 제공한 저작권자들에게 돌이킬 수 없는 손실을 초래 할 수 있다.

따라서 저작권 보호가 안 될 경우 지상파 디지털 방송에는 양질의 디지털 콘텐츠 조달이 어려울 것으로 예상되며, 이것은 결국 지상파 디지털 방송을 선택하는 시청자가 손해를 입는 것이다. 이러한 문제점으로 콘텐츠에 대한 보안 및 저작권 보호를 위한 법적 제도 및 실질적 기술의 필요성은 점차 증가되고 있다. 이러한 콘텐츠를 보호 할 수 있는 기술은 콘텐츠의 배포를 사전에 차단 할 수 있는 암호화 기술과 콘텐츠의 불법 배포 후 인증을 통한 저작권을 보호할 수 있는 워터마킹 기술로 구분되어진다. 하지만 이 두 기술 중 하나의 기술만을 사용하여 콘텐츠를 보호하는 방법으로 사용할 경우 취약함을 가질 수 있다. 어떤 불법 사용자가 콘텐츠의 암호를 알고 나면 그 이후의 콘텐츠의 저작권 및 소유권 주장은 어렵되 되며, 또한 워터마킹 기술 하나만으로도 콘텐츠를 보호하여 콘텐츠를 통한 수익모델을 지킬 수가 없게 된다. 콘텐츠를 보호하기 위하여 콘텐츠 배포를 사전에 차단할 수 있는 암호화 기술과 콘텐츠의 불

법 배포 후 인증을 통하여 저작권을 보호할 수 있는 워터마킹 기술을 병합하여 하나로 만들 필요성이 있다.

디지털 워터마킹은 저작권을 인증할 수 있는 의미 있는 데이터를 인간이 시각적으로 인지할 수 없도록 디지털 미디어에 은닉하고, 필요할 경우 내부에 은닉된 데이터를 찾아냄으로써 저작자임을 입증할 수 있는 기술이다. 따라서 비가시성(invisibility)과 강인성(robustness)이 중요한 척도이다[1]. 고화질 방송에서의 워터마킹은 화질열화에 더욱 민감하므로 비가시성은 더욱 중요해지고, 추가적으로 원본 영상 없이 검출이 가능한 블라인드 워터마킹이 가능해야 한다[2]. 또한 HD 방송에서는 ADC(analog to digital conversion)과 같은 자연적인 공격 및 MPEG(moving picture expert group)과 같은 압축 공격에 강인하여야 한다.

비디오 워터마킹에 관한 연구는 비트스트림 영역에서 수행하는 방법, 변환영역에서 수행하는 방법 그리고 공간영역에서 수행하는 방법으로 분류할 수 있다[3,5]. 비트스트림 영역에서 수행하는 방법은 비트스트림의 규격화에 따른 삽입 정보량에 제약이 따르며, 변환 영역에서 수행하는 방법은 DCT(discrete cosine transform) 혹은 양자화과정에서 워터마크를 삽입하므로 부호기가 변형되어야 한다. 공간영역에서 수행하는 방법은 일반적으로 화질열화가 발생하기 쉬우나 실시간 처리에 가장 유리한 방법이다. 워터마킹의 하드웨어 구현에 관한 연구는 동영상 워터마킹과 정지영상 워터마킹으로 구분 지을 수 있다. 동영상 워터마킹으로는 방송 모니터링을 위하여 구현한 것이 있고, 실시간 비디오 워터마킹을 구현한 것이 있다[3,4]. 정지영상 워터마킹으로는 JPEG 부호기에 워터마크를 삽입하여 구현한 Secure JPEG 부호기와 컬러 영상에 워터마크를 삽입하여 구현한 방법이 있다[5,6].

암호화 기술은 불법 배포에 대한 사전 차단 기술로 허가된 사용자만이

접근 가능한 기술로 적극적 보호 방법이라 할 수 있다. 멀티미디어 콘텐츠 보호를 위한 암호화 기술은 전체 암호화와 부분 암호화 기술로 나눌 수 있는데, 전체 암호화 기술은 비압축 영상의 전체 또는 압축된 비트스트림 전체를 암호화 하는 기술로 강력한 압축은 가능하나 계산 비용이 높은 단점이 있다[7,9]. 부분 암호화는 압축과정 또는 비트스트림의 일부분 또는 선택적으로 암호화하는 방식으로 전체 암호화 방식보다 강력한 암호화는 될 수 없지만, 낮은 계산량을 가진다[10,16].

최근 초고화질의 해상도를 가지는 방송용 비디오 콘텐츠의 제작이 증가하고 있지만, 기존의 소프트웨어적인 방법을 사용하여 저작권 정보를 은닉하고 배포 방지를 위하여 암호화 하는 방법으로는 실시간을 처리를 요구하는 방송환경에서는 사용이 어렵다. 또한 이러한 환경에 적용 가능하려면 워터마킹 또는 암호화에 대한 모듈 구현 연구 보다는 실제 적용할 수 있는 시스템화에 대한 연구가 필요하다.

본 논문에서는 HD/SD의 해상도를 가지는 비디오 콘텐츠의 보호를 위하여 실시간 하드웨어 구현을 위한 워터마킹 및 암호화 기법을 제안하고 이를 하드웨어 시스템으로 구현한다. 워터마킹의 경우 HD 해상도에서 실시간으로 동작할 수 있으며 비가시성과 강인성을 가질 수 있는 알고리즘을 제안하고 이를 FPGA를 이용한 하드웨어 기반으로 구현한다. 비디오 콘텐츠에 대한 암호화는 HD 해상도에서 실시간으로 동작할 수 있는 비디오 암호화 응용기술을 제안하고 이를 하드웨어 기반 기술을 통하여 구현하고 시뮬레이션을 통하여 그 결과를 나타낸다. 본 논문에서는 제안한 하드웨어에 최적화된 워터마킹 및 암호화 알고리즘에서 먼저 워터마킹 알고리즘은 하드웨어 구현에서 계산 복잡도를 줄이기 위하여 비트를 많이 차지 할 수 있는 부동 소수점 연산 대신 정수형 연산을 사용하였고, 워터마크 강도 계산을 위하여 주변의 화소 값에 대한 편차를 이용하

여 기존의 복잡한 알고리즘을 대신하여 하드웨어 기반에서 고속으로 처리 할 수 있는 마스킹 방법을 적용하였다. 또한 본 알고리즘의 비가시성과 강인성의 테스트를 위하여 먼저 소프트웨어로 구현한 후 결과를 검증하였다. 암호화의 경우 화소단위의 암호화 방법을 적용할 경우 암호화 적용 후 비디오 콘텐츠 암호화에서 가장 중요한 비디오 콘텐츠의 시각적 인지도 감소가 적기 때문에 암호화 후 원본 비디오 콘텐츠를 시각적으로 유추 가능하게 되어 비디오 암호화에 사용하기에는 부적절하기 때문에 제안한 알고리즘은 고화질의 비디오 콘텐츠의 물체에 대한 시각적 인지도의 감소 및 실시간 암호화를 만족할 수 있도록 하기 위하여 비디오 스크램블링 패턴을 사용하였고, 이 패턴의 고속처리를 위하여 SRAM을 사용하여 스크램블 패턴을 구현하였다. 또한 암호화에 따른 견고한 보안을 위하여 국제 표준화된 암호화 방법인 AES 암호화를 사용하여 패턴을 선택하도록 구현하였다. 제안한 방법 또한 소프트웨어로 구현하여 동작에 대한 검증을 하였고, VHDL을 통하여 하드웨어 구현 결과를 검증하였다. 본 논문에서 제안하는 워터마킹 및 암호화 알고리즘에 대한 FPGA 칩 구현의 검증을 위하여 HD의 고화질을 입력으로 사용할 수 있는 비디오 입력 장치와 알고리즘이 적용된 비디오 콘텐츠와 적용되지 않은 비디오 콘텐츠를 동시에 볼 수 있는 모니터, 비디오 출력 및 모니터의 입력 신호와 FPGA 시스템의 입력 및 출력 사이의 신호변환을 위한 신호 변환기 그리고 FPGA로의 입력 PDI(parallel digital interface) 신호 및 출력 SDI(serial digital interface)신호를 발생 시킬 수 있는 GINNUM사의 SDI/PDI 변환 칩이 내장된 시스템 보드를 사용하여 시각적으로 검증 할 수 있도록 시스템을 구현하였다. 구현 결과 워터마킹의 경우 원본 비디오 콘텐츠와 워터마크가 삽입된 비디오 콘텐츠 사이에 화질 차이를 시각적으로 확인이 어려웠고, 또한 워터마크가 삽입된 비디오에 대한 강인성

실험 결과 H.264와 같은 최신의 비디오 코덱에서도 검출 가능함을 확인하였다. 그리고 비디오 암호화의 경우 비디오 콘텐츠에 암호화를 적용하였을 때 시각적으로 암호화된 비디오 콘텐츠에서 원래 화면을 시각적으로 유추할 수 없을 정도로 비시각적 특성을 가짐을 확인하였다.

본 논문의 구성은 다음과 같다. Ⅱ장에서는 비디오 워터마킹 및 암호화와 관련된 배경 지식과 기존의 다양한 연구 및 기존 연구들의 문제점과 본 논문에서 주장하는 바를 제시하였다. Ⅲ장에서는 제안하는 비디오 워터마킹 및 암호화 알고리즘과 이를 토대로 하드웨어 기술 언어를 이용하여 구현된 알고리즘의 시뮬레이션 결과를 기술하고 있다. Ⅳ장에서는 제안한 방법들의 성능 평가를 위한 실험 환경과 실험 내용 및 실험 결과들을 정리하였다. 마지막으로 Ⅴ장에서는 결론을 맺으면서 논문을 마무리한다.

Ⅱ. 관련연구

본 장에서는 콘텐츠 보호를 위한 워터마킹 및 암호화와 관련된 기존의 개념과 다양한 연구들에 대해서 서술하였다. 먼저 멀티미디어 콘텐츠 보호를 위한 기존의 방법들에 대한 소개를 하고, 이에 사용되는 기술에 대한 설명을 기술하였다. 또한 본 논문에서 사용한 워터마킹 및 암호화의 기존 연구에 대한 소개를 통하여 현재의 기술들에 대한 문제점을 등을 서술하고 제안하는 연구 목적을 서술하였다.

본 장의 구성은 다음과 같다. 1절에서는 멀티미디어 콘텐츠를 보호할 수 있는 기술에 대한 기본 정의와 개념들에 대하여 서술하였고, 2절에서는 워터마킹에 대한 개념 및 종류에 대하여 서술하였고, 3절에서는 암호화에 대한 개념 및 비디오 암호화에 대하여 서술하였고, 4절에서는 하드웨어 구현에 관한 기본 개념 및 알고리즘의 하드웨어 구현으로의 진행과정을 서술하였다. 마지막으로 5절에서는 동영상 데이터에 대한 이해를 위하여 동영상의 구조 및 비디오 신호의 형태에 대하여 기술하였다.

1. 멀티미디어 콘텐츠 보호기술

아날로그에서 디지털시대로의 이동이 빠르게 진행되고 있지만 저자나 콘텐츠 제작자 같은 콘텐츠 소유자는 디지털로 된 멀티미디어 데이터의 배포에 대하여 조심스럽다. 위와 같은 디지털 멀티미디어 콘텐츠는 순식간에 복사되어 인터넷으로 배포되고 있기 때문이다. 그럼에도 불구하고 멀티미디어 콘텐츠가 인터넷을 통해 전자적으로 거래되어야 하는 것은 시대적 흐름이라 할 수 있다. 따라서 디지털 멀티미디어 콘텐츠에 대한 보호기술에 관심이 모아지고 있다. 또한 많은 콘텐츠 소유자들이 콘텐츠 유료화에 관심을 보임에 따라 저작권을 보호할 뿐만 아니라 인터넷을 통하여 안정된 수익 모델을 촉진할 수 있는 보호법과 기술적 장치에 대한 관심이 높아지고 있다. 멀티미디어 콘텐츠에 보호 기술은 사용 권한을 획득하지 못한 사람에게는 콘텐츠를 사용하지 못하게 하는 저작권 관리 기술, 저작권 소유주가 누구였는지를 추적할 수 있게 하는 워터마킹, DOI(digital object identifier) 및 INDECS(interoperability of data e-commerce system)와 같은 저작권 추적 기술들로 대별 될 수 있다. 저작권 관리 기술은 암호화 기술을 응용한 것으로 저작권자가 대가를 지불한 사용자에게만 사용권한을 줄 수 있어 좀 더 적극적인 저작권 보호 기술이다. 하지만 이런 암호화 기술들은 콘텐츠를 보호할 수 있지만, 사용자가 암호화를 알고 나면 그 후 콘텐츠의 저작권 및 소유권을 주장할 수 있는 어렵게 된다. 저작권 추적 기술은 특정마크나 고유번호를 삽입, 콘텐츠를 식별해 법적 분쟁 시 소유권을 입증할 수 있는 기법이다. 멀티미디어 콘텐츠는 복사본이 원본과 동일하기 때문에 대량으로 복사되어도 소유자를 가려낼 수 없는 단점을 보완하기위해 사용된 기술이다.

본 논문에서는 콘텐츠 복제를 원천적으로 방어할 수 있는 적극적인 콘텐츠 보호 기술인 암호화 방법과 불법 사용자가 암호화를 알아내어도 소유권을 주장할 수 있는 저작권 추적 기술 중 하나인 워터마킹 기술을 제안하고 이를 하드웨어로 구현한다.

2. 워터마킹의 개념

워터마킹은 텍스트, 이미지, 비디오, 오디오 등의 데이터에 원래의 소유주만이 아는 마크를 사람의 육안이나 귀로는 구별할 수 없게 삽입하고 이를 검출하여 소유권을 주장하는 기술이다. 만약 불법적으로 멀티미디어 콘텐츠를 사용하는 사용자가 있다면, 콘텐츠에 숨겨져 있는 저작권 정보를 통하여 자신의 소유임을 밝힐 수 있고, 이는 소유권 주장에 결정적인 증거가 될 수 있다.

워터마크라는 용어는 지폐 제작 과정에서 유래된 것으로 지폐 제작 과정에서 젖어있는 상태에서 그림을 인쇄하고 이를 말린 다음 양면을 인쇄하게 된다. 이처럼 젖어있는 상태에서 그림을 넣는 기술을 워터마크라 부른다. 현재 논의되고 있는 기술은 디지털 워터마킹 기술로서 인터넷이 빠르게 보급되고 있고 웹상에서는 책이나 그림, 음악 등의 저작물이 대량으로 복사되고 배포 될 수 있기 때문에 중요한 기술로 떠오른다. 특히 전자 상거래 시대에는 전자책, 인터넷 신문과 잡지, 디지털 방송, DVD 등이 거래될 것으로 예상하고 있다. 하지만 데이터 통신 기술 및 정보 기술의 발달로 언제 어디서든 디지털 멀티미디어 데이터를 원본과 같은 품질로 복사 및 배포가 순식간에 가능하게 되어, 이와 같은 문제에 대한 대책이 필요한 실정이다. 현재 웹하드를 통한 MP3 또는 고화질의 비디오 데이터의 전송이 불법적으로 이루어지고 있어 저작권 소유자들의 피

해는 날로 심각해지고 있다. 이와 같은 저작권에 대한 이력 추적 및 저작권을 검출하여 소유권을 주장 할 수 있는 워터마킹 기술이 개발되고 있다.

2.1 저작권 보호를 위한 워터마킹

디지털 워터마킹의 기본 개념도를 그림 1에 나타내었다. 그림에서와 같이 워터마킹 시스템은 워터마크를 삽입하는 부분과 워터마크 삽입에 따른 강인성 테스트를 위한 공격부분 그리고 삽입된 워터마크를 검출하는 부분으로 나누어져 있다. 그림에서와 같이 비디오 신호에 은닉하기 위한 정보와 비밀 키를 사용하여 워터마크를 생성하고 워터마크는 원본 영상에 은닉하게 된다. 은닉된 영상은 워터마크를 검출하는 시스템에서 비밀 키와 메시지로 분리하여 결과를 얻게 된다.

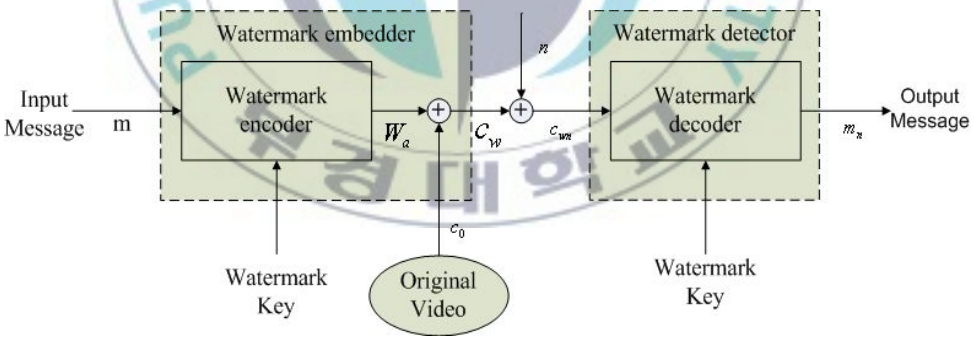


그림 1. 워터마킹 기술 개념도

Figure 1. A block diagram of watermarking technique

각 워터마킹 기술의 응용분야에 따라서 특정한 요구사항을 갖기 때문에 응용분야마다 요구사항이 다르다. 여기서는 저작권 보호라는 기본 취지에서 워터마킹 기술에 필요한 방향에 대해서 언급하고자 한다. 워터마

킹 기술이 콘텐츠 내부에 삽입되기 때문에 원본의 손실이 불가피하고, 따라서 워터마킹 기술로서 활용되기 위해서는 다음의 조건을 만족해야 한다.

1. 비인지성 (Imperceptibility)

저작권 보호를 위한 워터마킹 응용분야에서 워터마크를 삽입 할 경우 가능한 원본 데이터의 품질에 영향을 미치지 않도록 해야 한다. 즉 인간의 시각적인 인지 측면에서 원본 영상과 워터마크가 삽입된 영상 사이에 차이를 느끼지 못해야 한다. 원본 영상에 워터마크를 강력하게 삽입하게 되면 원본 영상의 화질 열화를 강하게 가져오기 때문에 워터마크의 삽입 강도를 계산하는 방법은 중요하다.

2. 삽입량(Capacity)

워터마크에 들어갈 수 있는 정보의 량은 워터마킹의 응용분양에 종속적이다. 유럽방송연맹(european broadcasting union, EBU)에서 제안하는 워터마크 삽입 량은 64비트 이상이am로 되어 있다. 일반적으로 지적재산권 보호를 위해서 사용하려면 ISBN은 약 10자리 십진수, ISRC (international standard recoding code)는 12자리 영숫자를 사용하므로 60비트에서 70비트 정도의 정보를 삽입 량으로 가지면 된다.

3. 강인성(Robustness)

강인성은 소유권자를 위해 고려되어야 할 가장 중요한 요소라고 할 수 있는데, 워터마킹 기술을 이용하여 지적 소유권 보호의 목표를 이루기 위해서는 다양한 공격에도 그 워터마크가 소유권자에 의해 검출되어야 한다. 영상에 대한 공격으로는 JPEG와 간은 손실 부호화, 필터링, 크기

변환, 대비 강조, 크리핑(clipping), 재 양자화(re-quantization), 회전, 변위(translation) 등이 있다. 이와 같은 환경에서도 워터마크를 검출할 수 있어야 강인한 워터마킹 기술이라 할 수 있다.

4. 검출률(Detecting ratio)

워터마크가 삽입된 콘텐츠에서 정확하게 워터마크를 검출할 확률을 나타내는 것으로 검출에 성공한 정도를 나타내는 검출률과 부정확한 워터마크를 검출한 정도를 나타내는 오검출률로 나뉜다.

2.2 비 압축 기반 비디오 워터마킹 기술

비 압축 기반 비디오 워터마킹 방법은 현재 알려져 있는 비디오 압축을 적용하기 전단계에서 워터마크를 삽입하는 방법이다. 공간상의 특성을 이용하여 원 비디오의 화소에 직접 워터마크를 삽입하는 방법이 있고, 이극 신호를 이용한 확산 스펙트럼(spread spectrum), 주파수-명암-콘트라스트-에지 마스킹을 통한 HVS(human visual system) 등을 이용하는 방법, DWT(discrete wavelet transform)와 같은 주파수 영역으로 신호를 전환하여 워터마크를 삽입하는 방법 등을 이용하는데, 이들 방법 중 공간상의 화소 특성을 이용하는 방법은 계산 복잡도는 작으나 공간 및 시간상의 압축 기법을 사용하였을 경우 워터마크 검출률이 감소하는 경향을 가지고, 변환 영역의 특성을 이용하여 워터마크를 삽입하는 방법은 검출률을 높일 수 있으나, 구현의 복잡성 때문에 고해상도를 지원하는 응용에서 실시간으로 구현하기 어려운 단점을 가지고 있다.

1. Niu 등이 제안한 방법

공간상의 원 비디오에 3차원 DWT를 이용하여 워터마크를 삽입하는 방법으로, 움직임의 다소에 따라 원 비디오를 영역분류 한 후, 영역에 따라 3차원 DWT및 2차원 DWT를 각각 수행한다[17]. 피라미드 계층구조(pyramid hierarchical structure)를 이용하여 전처리된 워터마크를 DWT 계수에 삽입한다. 그러나 이 방법에서는 워터마크의 견고한 삽입을 위해 오류 정정 부호를 사용함으로써 많은 부가정보가 삽입되어 비가시성이 나빠지며, 계조영상을 비트플레인(bit plane)후 이진화 하여 삽입함으로써 상위 비트에서 에러가 발생할 때 견고성이 떨어진다.

2. Zhang 등이 제안한 방법

인간 시각 특성을 고려하여 움직임이 큰 매크로 블록에 대해 움직임 벡터의 수평 성분과 수직 성분의 크기 비교를 이용하여 워터마크를 삽입하는 방법을 제안하였다. 움직임이 큰 매크로 블록은 움직임 벡터가 설정된 크기 값과 미리 설정된 임계값과의 비교를 통해 구해 질 수 있다[18]. 삽입 과정은 삽입되는 워터마크의 정보에 따라 움직임 벡터의 수평 또는 수직 성분들 중 큰 값을 가지는 성분을 $\mathrm{mod}(x,2)$연산을 수행하여 나온 결과 값과 일치하도록 움직임벡터를 수정하여 구현한다. 만약 움직임 벡터 성분의 크기가 같은 경우는 두 성분 모두를 증가 시켰다. 이 방법은 시각적 특성을 고려하여 워터마크 정보를 삽입할 수 있는 장점이 있으나, 워터마크의 정보에 따라 3가지 분류에 의해 삽입되어 움직임 벡터가 변경될 확률이 1/2 이상이 되는 단점이 있다. 또한 임계값도 임의로 설정되어 워터마크 정보의 삽입을 보장할 수 없다는 문제점이 있다.

3. Kutter가 제안한 방법

인간 시각 특성을 고려하여 공간 영역에서의 워터마킹 방법 중 JPEG

및 MPEG과 같은 손실 압축에 강인한 특성을 가지는 방법이다[19]. Kutter는 화소와 화소 주위의 공간적인 특징을 계산하여 워터마크 삽입에 따른 강인성을 계산하고 워터마크를 화소에 바로 적용하는 것이 아니라 워터마크를 대역확산기법에 사상시켜 삽입되는 정보량의 크기를 증가시키고, 또한 삽입되는 정보를 반복하여 삽입하는 방법을 취하여 검출률을 높였다. 하지만, Kutter의 방법은 공간 특징을 계산하는데 많은 시간 비용을 필요로 하는 단점을 가지고 있다.

2.3 압축 기반 비디오 워터마킹 기술

압축 기반 비디오 워터마킹 기술은 MPEG에서 만들어진 비디오 압축 표준과 같은 비디오 압축에서 워터마크를 삽입하고 검출하는 방법을 말한다. 압축에서 DCT 등과 같은 변환 영역에 워터마크를 삽입하는 방법, 압축 후 비트스트림의 특성을 이용하여 워터마크를 삽입하는 방법, VLC(variable length coding)에서와 같은 부호화 정보를 이용하여 워터마크를 삽입하는 방법 등이 있다. 이러한 방법은 비디오 압축과정 또는 비디오 압축된 결과를 이용하여 워터마크를 삽입하고 변환하는 방법이기 때문에 워터마킹을 독립적으로 구현할 수 없고, 압축 코덱내부에 구현해야하기 때문에 비디오의 압축 형식이 알고리즘이 제시된 압축형식과 일치하지 않을 경우 적용하기 힘들다는 단점이 있다.

1. Hartung 등이 제안한 방법

MPEG-2 비트열에 워터마크를 삽입하고 대역확산 방식을 적용하는 방법을 제안하였다[20]. 워터마크는 원본 비디오 신호화 같은 차원의 의사 잡음 (pseudo noise) 신호를 사용하여 발생된다. 각각의 압축된 비디오

프레임에 대해 해당되는 워터마크 신호 프레임이 8*8 블록 단위로 DCT 변환되고, 이것은 원본 비디오 프레임의 DCT계수에 더해진다. 이 DCT 계수를 양자화 한 후 허프만 부호화(huffman coding)를 수행한다. 워터마크가 삽입된 비디오 시퀀스가 원본 비디오 시퀀스의 비트율을 넘지 않게 하기 위하여 DCT 계수에서의 비트율이 증가하지 않는 경우에만 워터마크 계수가 전송된다. 그러나 이러한 방법은 비트율 제약 때문에 DCT된 워터마크 계수의 일부만이 DCT된 비디오 프레임의 블록에 삽입되며, 심지어 DC 계수만이 삽입되는 경우도 있다. 따라서 비트열 상의 워터마크 삽입 방법은 강인하지 않으며, 다양한 부호화기에 적용할 수 없는 단점이 있다.

2. Checcacci 등이 제안한 방법

랜덤하게 선택된 매크로블록의 휘도성분에 미리 정해진 양자화된 DCT 중간 주파수 계수의 쌍을 이용하여 MPEG4 비트 스트림의 비디오 객체에 워터마킹하는 방법을 제안하였다[21]. 부호화가 MPEG4 비트 스트림에 직접 삽입되므로 비디오 파일이 MPEG2와 같은 다른 압축 표준에 의해 다시 압축될 경우 저작권 정보가 손실된다. 만일 화면의 비디오 오브젝트가 매우 작다면 MPEG4 압축에 강건한 워터마크를 삽입하는 것이 매우 어렵다.

3. Dugad 등이 제안한 방법

MPEG에 의한 동영상 압축시에 I, P, B 프레임에 모두 워터마크를 삽입하는 방법을 제안하였다[22]. 복잡한 이미지가 그렇지 않은 이미지에 비하여 임계값 이상인 DCT 계수가 많다는 원리를 이용하여 미리 정한 임계값 이상인 DCT 계수 값에만 워터마킹하고 검출시에는 삽입된 워터

마크에 의해 DCT 계수의 절대값이 원래 값보다 크다는 것을 이용하여 워터마크 삽입시에 정한 임계값 보다 큰 값을 PN 시퀀스와의 상관관계를 계산하여 워터마크의 유무를 판단한다. 그러나 이 방법은 워터마크 삽입시와 검출시에 정한 임계값이상인 DCT 계수값의 개수가 각 프레임에 따라 다르고 동영상에 따라 다르게 나타난다. 그러므로 저작권 정보로 사용되는 워터마크 검출이 필요한 경우 저작권 정보의 검출 효율이 낮아지는 단점이 있다.

3. 암호화의 개념

암호 알고리즘이란 전통적으로 군사, 외교, 정치 등 소수 특권 계층에서 어떤 목적을 위해 사용되어 온 비밀스러운 문자 코드 체계이다. 그리고 이러한 암호는 고대 시저 암호(caesar encryption)에서 근대의 전치 암호(transposition encryption), 치환 암호(substitution encryption) 와 2차 대전에 독일에서 사용한 회전자 기계(rotor machines) Enigma등 많은 알고리즘이 잘 알려져 있다[23]. 이와 같은 암호 알고리즘을 전통적인 암호 방식 즉 비밀 키(대칭 키, 단일 키) 암호(secret, symmetric, one key)라고 한다. 암호문의 안전성은 암호문 생성에 사용된 비밀 키의 비밀 유지가 핵심적인 사항이다. 그리고 현대 암호는 고전 암호와는 달리 컴퓨터의 빠른 연산 능력을 사용하지 않고서는 암호 및 복호를 수행 할 수 없을 정도의 복잡하고 많은 연산을 요구한다. 이러한 현대 암호 알고리즘의 개발과 연구의 시초는 1977년 미국의 연방표준국(NIST)에서 DES(Data Encryption Standard)[24]를 표준 블록 암호 알고리즘으로 채택과 1976년 공개 키 암호(public key)개념[25]이 발표된 이후 지금까지 현대 암호학이 비약적으로 발전하게 되었다.

현대 암호는 크게 비밀 키 암호와 공개 키 암호로 나눌 수 있다. 그 중 비밀 키의 대표적인 암호인 DES는 적은 키 공간과 획기적인 암호 분석법인 차분 공격과 선형 공격의 개발로 1990년대 후반 암호 알고리즘이 안전하지 않다는 것이 증명 되었다[26,27]. 이후 미국을 비롯한 선진국들은 자국의 정보보호를 위한 새로운 블록 암호 알고리즘 선정을 위해 국제적인 공모 사업을 시작 했다. 특히 미국의 AES(advanced encryption standard)[28]선정 프로젝트, 유럽의 NESSIE(new european schemes for signatures. integrity. and encryption)[29] 프로젝트, 일본의 CRYPTREC(cryptography research and evaluation committees)[30] 프로젝트 등이 대표적이며, 우리나라는 자체 개발한 SEED[31] 와 ARIA[32]를 표준 128비트 블록 암호로 제정하였다. 초기 블록 암호는 소프트웨어로 구현되었지만 최근에는 빠른 정보처리를 위한 하드웨어 구현에도 많은 연구가 진행 중이다.

3.1 비디오 암호화

1. 풀 암호화

풀 암호화는 압축 후에 사용된다. 이는 아주 간단하고 이해하기 쉬운 암호 구조이다. 압축된 코드 스트림을 암호화 하는 것은 전체적으로 단순 암호화와 비슷하다. 이것을 대체하기 위하여, 풀 암호화는 먼저 압축된 비트스트림을 분할하고 포장하여 구조화된 데이터 패킷으로 바꾼다. 그리고 하나의 암호를 이용하여 각각의 독립된 패킷의 데이터에 대해 암호화 하는데 헤더는 암호화 하지 않는다. 이런 방법은 Microsoft의 WMRM과 OMA의 DRM은 모두 이런 방법을 채택하였다. Qao와 Yi는 풀 암호화 방식을 MPEG-4 FGS의 멀티 제어 암호화에 응용하는 것을

제안하였다[33,34].

2. 비트스트림에서의 선택적 암호화

풀 암호화에서는 헤더를 제외한 모든 데이터를 암호화 하는데 반해 선택적 암호화에서는 비트스트림에서 일부 데이터는 중요하지 않거나 다른 비트스트림에 속하여 있다는 압축의 특성을 이용하여 부분적인 데이터에 대해서만 암호화 한다. 그리고 나머지 데이터에 대해서는 암호화 하지 않는다[35,36].

3. 변환 영역에서의 선택적 암호화

변환영역에서의 암호화 방법은 공간역역의 계수 값들을 8*8또는 4*4 크기의 DCT 계수로 만든 후 계수 값을 암호화하거나 양자화 후 계수 값을 암호화한다. 전자의 경우 SEED, AES, DES와 같은 암호화 알고리즘을 계수 값에 직접 적용하여 보안 레벨을 높일 수 있지만, 계수들 간의 연관성을 파괴하므로 압축 효율을 떨어뜨릴 수 있다. 반면에 후자의 경우는 주로 퍼뮤테이션(permutation), 셔플링(shuffling)의 방법을 사용하여 암호화하게 되는데, 계산 복잡도가 낮아 시간 효율성에는 이득이 있으나 보안 레벨이 낮고 압축 효율 또한 떨어진다[37].

4. 공간 영역에서의 선택적 암호화

공간영역에서의 선택적 암호화에서 간단한 방법 중 하나는 압축되기 전이나 압축하지 않은 영상의 비트 판(bit-plane)을 암호화 하는 것이다. 높은 중요도를 가지는 비트 판은 낮은 중요도의 비트 판보다 더 많은 가시 정보를 포함하고 있기 때문에, 선택적 암호화 체계는 자연스럽게 높은 중요도를 가진 비트 판에서 낮은 중요도를 가진 비트 판으로 하향 진

행한다. Maniccam와 Nikolaos는 역방향 순서로, 중요도가 최소의 비트 판부터 중요도가 많은 비트 판을 암호화 하는 암호화를 제안하였다. 이들은 8개의 비트 판을 가진 경우 최소 네 개에서 다섯 개의 비트 판을 암호화해야 가시적인 부분의 감소를 제공할 수 있다고 하였다[38].

공간영역에서 화소별로 암호화 할 경우 비가시성이 낮을 뿐 아니라 계산 복잡도가 너무 크기기 때문에 셔플링 방법을 사용하게 된다.

4. 하드웨어 구현

본 논문에서는 SD/HD 형식의 멀티미디어 콘텐츠를 실시간으로 워터마킹 및 암호화를 구현하기 위하여 하드웨어에 최적화된 알고리즘을 제안하고 하드웨어 시스템을 구성하여 제안한 알고리즘을 구현하였다. 먼저 워터마킹을 하드웨어로 구현한 연구들을 살펴보면 표 1과 같다.

표 1. 워터마킹 알고리즘의 하드웨어 구현 연구

Table 1. A study on hardware implementation of watermarking

Researcher	Design type	Visibility and Robustness	Watermark Domain	Feature of Chip
Strycker[39]	TriMedia (DSP)	Invisible Robust	Spatial	100MHz
Petitjean[40]	FPGA board	Invisible Robust	Fractal	50MHz
Garimella[41]	Custom IC-0.13μ	Invisible Fragile	Spatial	1.2V
Tsai[42]	Custom IC-0.35μ	Invisible Robust	DCT	3.3V 50MHz
Mathai[43]	Custom IC-0.18 μ	Invisible	Video	75MHz
Seo[44]	FPGA board	Invisible Robust	Wavelet	82MHz
Garimella[45]	Custom IC-0.13 μ	Invisible Fragile	Spatial	1.14v 166.6MHz
Mohanty[46]	Custom IC-0.35μ	Visible	Spatial	292MHz
Mohanty[47]	Custom IC-0.25μ	Visible	DCT	1.5v 2.5v Dual

고화질 방송과 실시간 방송에서 실시간 워터마킹은 반드시 필요하다. 실시간 워터마킹을 가능하게 하기 위하여 워터마킹 알고리즘을 칩 수준으로 구현 방법이 있다[39],[42]. 워터마킹 알고리즘의 VLSI(very large scale integrated) 구현은 A/D 변환 그리고 JPEG 압축에 강인하도록 설계되었다[40],[47]. 이와 같은 연구들의 초점은 워터마킹 알고리즘의 단일 칩 구현이었고, 워터마킹 알고리즘을 전용 시스템 수준에서 효과적이고 효율적인 구현에 대하여 연구하지는 않았다.

비디오 암호화 알고리즘에 대한 하드웨어 구현은 화소를 직접적으로 암호화 하지 않고 대부분 압축과 동반하여 비트스트림 상에서 정형화된 블록 암호화방법, 셔플링, Xoring 등을 사용하게 된다. 화소에 직접적으

로 하지 못하는 이유 중에 하나는 계산 량이다. 비디오의 해상도가 HD(1920*1080)크기이고 비디오 포맷이 4:2:0이고 1초에 30장면을 직렬로 전송하게 되면 약 1.45Gbps의 대역폭을 가져야 한다. 이를 20비트의 YUV로 변환하여 데이터를 전송한다 하더라도 약 75Mbps의 데이터 전송속도를 가지게 된다. 만약 블록 암호화중 가장 빠른 AES 128비트 암호화를 적용할 경우를 생각해보면, AES 암호화는 128비트 암호화에 걸리는 시간이 약 2us정도이다 1920*1080의 해상도에 30장면이라면 128비트 암호화를 약 19.44초가 걸리게 된다. 위와 같은 이유에서 화소별 암호화를 할 수 없게 된다.

4.1 하드웨어 구현 방법

하드웨어 구현을 위하여 우선 소프트웨어적 방법으로 알고리즘을 검증하여야 한다. 소프트웨어로 알고리즘을 검증 한 후, 소프트웨어로 개발된 알고리즘을 하드웨어로 구현하기 위하여 데이터 정밀도 조정과 각종 소프트웨어 루틴들을 하드웨어에 적합한 구조로 변경하는 작업이 필요하다. 하드웨어 구현이 어려운 소프트웨어 루틴들은 간소화된 알고리즘으로 변경하여 최적의 하드웨어를 위한 알고리즘으로 변경이 필수적이다. 이렇게 하드웨어에 최적화된 알고리즘을 소프트웨어로 다시 구현하여 처음의 소프트웨어용으로 구현한 알고리즘 성능과 비교하는 작업이 필요하다. 이는 두 가지 목적이 있는데 첫째는 소프트웨어 알고리즘과의 성능 차이 확인과 나중에 하드웨어로 구현했을 때 하드웨어 디버깅의 참조 데이터로 사용될 것이다. 이렇게 함으로써, 소프트웨어로 구현된 알고리즘의 성능과 하드웨어를 고려하여 간소화된 알고리즘의 성능의 차이를 최소화 할 수 있다. 이의 과정을 정리하면 아래 그림 2로 나타낼 수 있다.

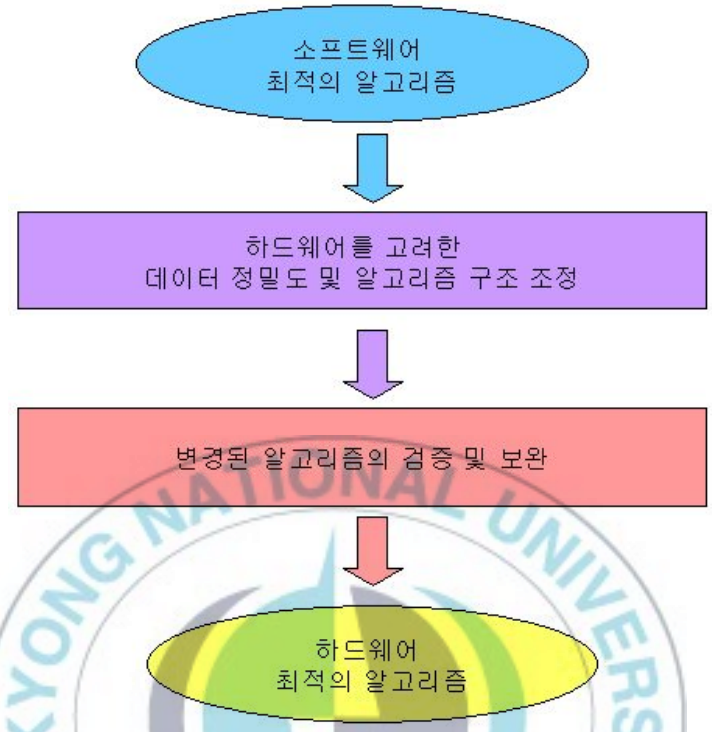


그림 2. 하드웨어를 고려한 최적화된 알고리즘 개발 절차

Figure 2. A block diagram of optimized algorithm development flow with a hardware implementation

다음 단계로서 하드웨어를 고려한 최적화된 알고리즘을 실제 FPGA 칩으로 구현하기 위하여 하드웨어 서술 언어 일종인 VHDL을 이용한 VHDL 코딩 작업을 수행할 것이다. 본 단계에서는 위에서 C언어로 개발한 알고리즘을 VHDL로 재 구현하며, 코드 논리에서의 동작과 게이트 레벨에서의 동작의 오차를 줄이고 최적화된 게이트 레벨 설계와 시뮬레이션 및 테스트를 수행한다. 이 단계가 완료되면 실제로 FPGA칩을 제작하게 된다. 아래 그림 3은 VHDL을 이용하여 칩 설계 과정을 도시한 것이다.

그림 3. HDL을 이용한 칩 설계 과정

Figure 3. A chip design flow using HDL

위의 설계 과정에서 보듯이 VHDL로 코딩을 하게 되면 언어의 문법적인 체크와 함께 컴파일 작업이 수행되며, VHDL코드를 논리 게이트로 합성하는 과정과 논리회로들이 제대로 동작하는지 검사하는 시뮬레이션 과정이 수반된다. 아래 그림 4는 VHDL 코드에 대한 모의실험 결과 화면이다. 시뮬레이션 결과가 원하는 결과를 나타낸다면 FPGA에 그 결과를 입력하여 시스템에서 동작확인을 하게 된다.

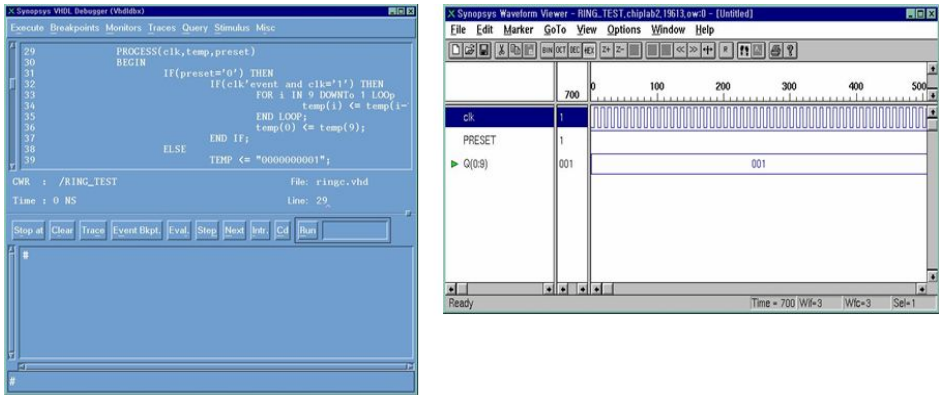


그림 4. VHDL 모의실험 결과

Figure 4. VHDL simulation result

5. 동영상 데이터

5.1 동영상 데이터의 구조

하나의 비디오(video) 데이터는 시퀀스(sequence)들의 집합으로 이루어진다. 시퀀스는 동영상 구조의 최상위 계층으로 신(scene)들의 집합으로 구성된다. 하나의 신은 다수의 샷(shot)으로 이루어지며, 샷은 동영상 구조의 최소 단위인 프레임(frame)의 집합으로 구성된다. 동영상의 계층 구조를 그림 5에서 도식화하여 나타내었다. 그림 5에서와 같은 동영상 데이터는 시퀀스, 신, 샷, 프레임의 4단계의 계층적 구조를 가지는 것을 알 수 있다.

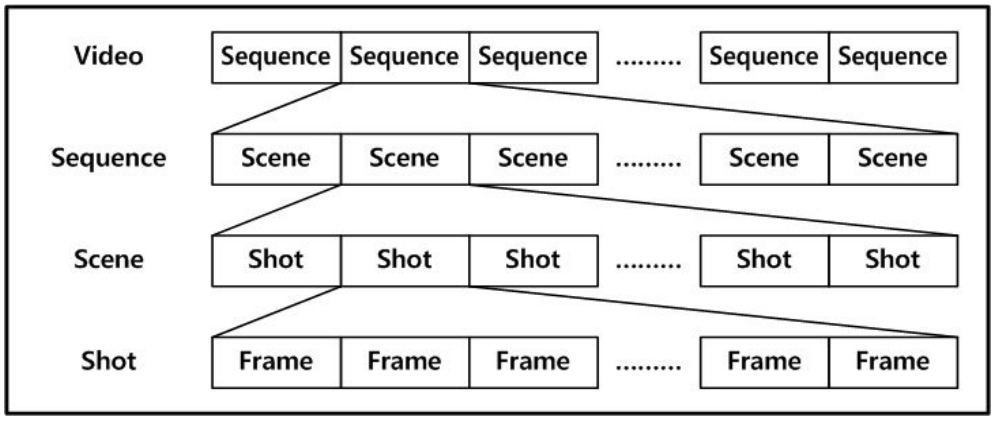


그림 5. 동영상의 계층적 구조

Figure 5. Hierarchical structure of moving picture

Frame : 동영상 데이터를 구성하는 최소단위이며, 카메라 필름 한 장에 해당하는 하나의 정지영상이다.

Shot : 동영상을 구분할 때의 기본단위로 필름이 끊기지 않고 연결된 프레임으로 구성되어있다.

Scene : 하나의 주 대상을 촬영한 연속된 Shot들의 집합이다.

Sequence : Scene과 Scene 사이의 경계에 해당하는 장면 전환점이다.

5.2 비디오 신호의 구조

일반적으로 DVD 재생기나 비디오 재생기와 같은 하드웨어 장치에는 비디오 출력을 가능하게 하는 단자들이 장착되어 있다. 이러한 단자들은 컴포지트(composite), 컴포넌트(component), HDMI, DVI 등이 있는데, 이와 같은 단자를 통하여 전송되는 비디오 신호는 특정 표준에 근거하여 만들어 지게 된다. 이러한 표준 중 하나가 VESA(video interface port

standard) 이다. VESA에서는 비디오의 해상도에 따라 어떻게 전송되는지에 대하여 설명하고 있다. 그림 6은 SD급의 해상도를 가지는 비디오 신호를 표현하고 있다[48].

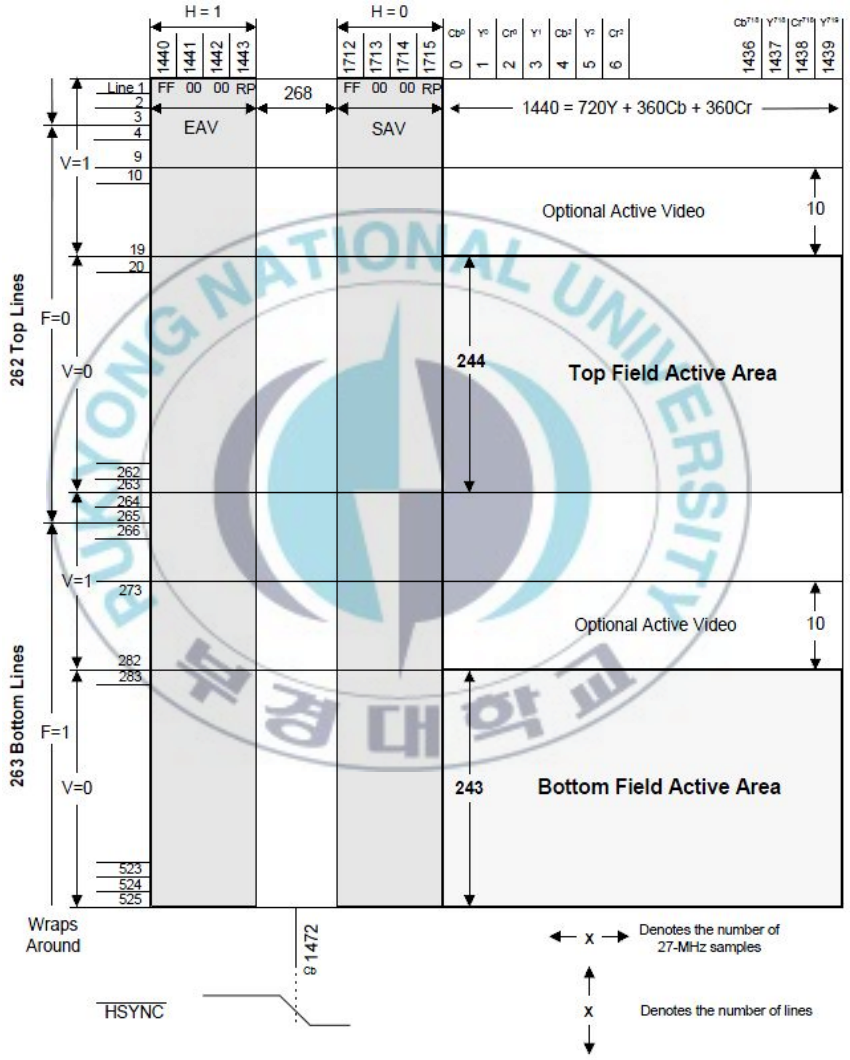


그림 6. SD급 비디오 신호 구조

Figure 6. A video signal structure of SD format

그림 6에서 보면 720*487의 해상도를 가지는 SD급의 비디오를 나타내고 있으며, 두 개의 필드를 가지는 비월 주사방식으로 되어 있다. 또한 비디오 포맷은 4:2:2의 YCbCr로 되어 있다. 그림에서 비디오 신호 중 실제 모니터와 같은 화면에 보이는 부분을 “top field active area” 와 “bottom field active area”로 되어 있고, 나머지 부분은 비디오 신호는 존재하지만, 실제 화면에 나타나지 않는 부분이다. 그림 7은 HD급의 해상도를 가지는 비디오 신호 구조를 나타낸다. HD급의 비디오 신호 구조 또한 SD급의 비디오 신호와 비슷하지만 SD급의 경우 하나의 클록에 따라 Y또는 CbCr 중 하나의 신호를 출력하지만 HD의 경우 하나의 클록에 Y와 CbCr이 동시에 출력되는 특징을 가지고 있다. 이러한 이유로 비디오 신호를 제어하기 위한 하드웨어를 만들기 위하여 SD급의 비디오와 HD급의 비디오 신호를 다른 방법으로 구현하여야 한다.

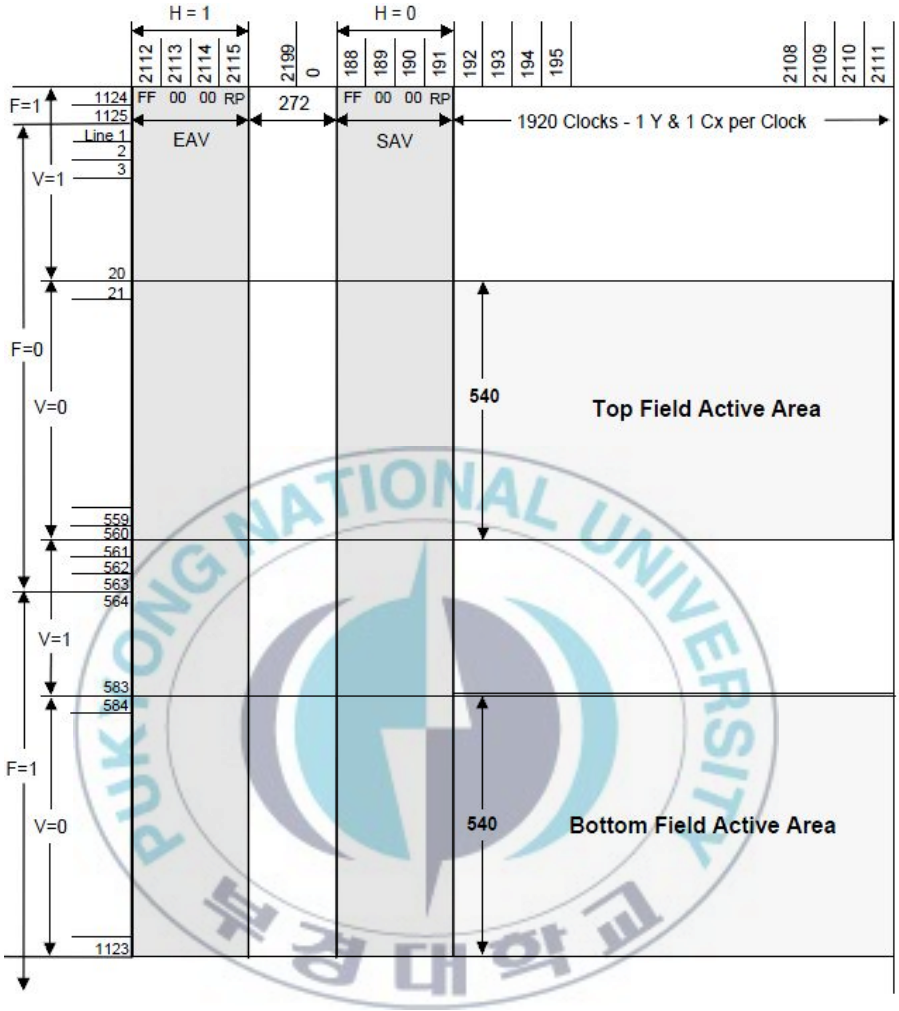


그림 7. HD급 비디오 신호 구조

Figure 7. A video signal structure of HD format

Ⅲ. 제안하는 방법

본 장에서는 제안하는 콘텐츠 저작권 보호를 위한 하드웨어 기반의 워터마킹 및 암호화 구현을 상세하게 기술하고 있다. 본 장에서는 하드웨어 구현에 적합하면서 실시간으로 처리 가능한 워터마크 삽입 알고리즘을 제안한다. 그리고 제안한 알고리즘을 하드웨어 기반의 언어인 VHDL을 이용하여 작성하고 모의실험 결과를 나타낸다. 다음으로 콘텐츠 배포 방지를 위한 비디오 암호화의 하드웨어 구현을 위한 최적 방법을 제안하고 이를 VHDL 작성하여 모의실험 결과를 타나낸다.

본 장의 구성은 다음과 같다. 먼저 1절에서는 하드웨어 구현에 적합한 워터마킹 알고리즘에 대하여 기술하고, 2절에서는 제안하는 워터마크 삽입 알고리즘의 FPGA 구현 방법에 대하여 기술하고, 워터마크 삽입 및 검출 시스템에 대하여 기술한다. 3절에서는 비디오 암호화를 위하여 본 논문에서 제안하는 방법에 대하여 소개하고 이를 하드웨어 기술 언어를 사용하여 시뮬레이션 결과에 대하여 기술하였다.

1. 하드웨어 구현에 적합한 워터마킹 알고리즘

1.1 워터마크 삽입 알고리즘

강인한 워터마킹을 위하여 본 구현에서는 Kutter 등이 제안한 워터마킹 기법을 사용하였다[19]. Kutter는 강인한 워터마크 삽입을 위하여 대역확산 기법과 워터마크를 반복적으로 삽입하는 방법을 사용하였다. 하지만, 워터마크 강도가 소수점 단위이고, 복잡한 방법으로 삽입강도를 계산하였고, 의사 랜덤 신호를 이용하여 워터마크를 구현하였다. Kutter가 제안하는 워터마크 삽입 방법은 식 (1)과 같다. 식 (2)는 워터마크 삽입을 위한 강도 계산 방법이다. 본 논문에서는 하드웨어 구현에서 필수적인 계산복잡도 간소화를 위하여 다음과 같은 방법을 제안한다.

$$\hat{I}(x,y) = I(x,y) + w(x,y) \tag{1}$$

$$w(x,y) = \sum_{i=0}^{N-1} (-1)^{b_i} s_i(x,y)\alpha(x,y) \tag{2}$$

$I(x,y)$: original contents　　$\hat{I}(x,y)$: watermarked contents
$\alpha(x,y)$: watermark strength　　$w(x,y)$: watermark
$s_i(x,y)$: modulation function　　b^i : binary signature

첫 번째, 워터마크 강도 계산을 위하여 기존 방법은 로그스케일과 자승의 조합으로 영상 내부의 국부적인 특징을 추출하였지만, 로그스케일

과 자승의 조합은 하드웨어 구현에서 계산 복잡도를 매우 크게 하기 때문에 영상내 국부적인 특징을 얻기 위하여 지역 분산 값의 크기를 이용하였다. 두 번째, 메시지를 워터마크로 변환 할 때 곱셈과 나눗셈을 배제하고 단순 사상방법을 사용하여 계산량을 줄였다. 마지막으로, 워터마크 삽입되는 위치에 따라서 워터마크 강도가 조정되어야 하는데 일반적으로 복잡한 계산을 통하여 해당 위치에 대한 강도 크기를 소수점 단위로 계산하게 되지만, 정수형으로 워터마크 강도 크기를 결정한다. 또한 화질열화와 워터마크 강도 개선을 위하여 영상의 특징을 이용한 단계별 워터마크를 삽입하여 시각적으로 민감할 수 있는 부분에는 워터마크를 약하게 은닉하고 그렇지 않은 부분을 6단계로 나누어 워터마크를 은닉하여 화질열화를 최소화하였다. 그림 8은 워터마크 강도를 계산하기 위한 계산 과정을 나타내었다. 그림과 같이 전처리 여과기를 사용하여 원본영상에서 고주파와 저주파 특성을 분리한다. 그 다음 중심화소와 그 주변 화소간의 상관도의 크기에 따라 6단계로 나누어 워터마크 강도를 계산하게 된다.

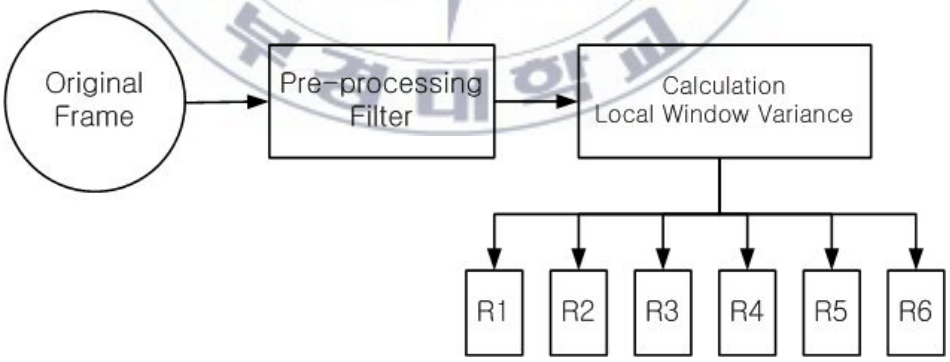


그림 8. 워터마크 강도 계산 과정

Figure 8. A flow for calculation of watermark strength

그림 9는 워터마크 강도를 계산을 위한 전처리 효과를 나타내었다.

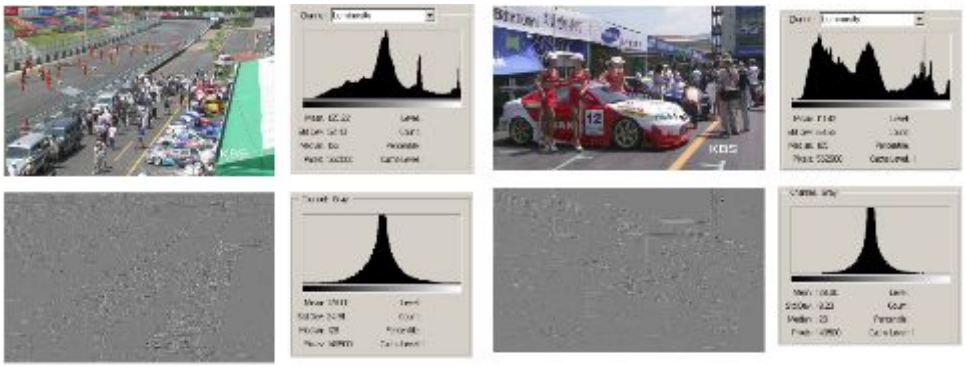


그림 9. 전처리 효과

Figure 9. A effect of pre-processing

그림에서와 같이 전처리 후 히스토그램은 가우시안 분포를 가지게 된다. 전처리를 위하여 사용한 방법은 고주파 여과기인 CSF(cross shaped filter)를 사용하였다. CSF는 다른 고역 통과 필터보다 작은 계산량으로 비슷한 효과를 가지게 된다. 그림 10은 5*5의 크기를 가지는 CSF를 나타내고 있다.

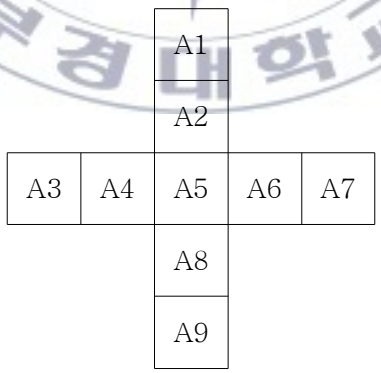


그림 10. 십자 형태 여과기

Figure 10. A window of cross shaped filter

CSF는 중심 화소 크기가 주변 화소에 비하여 상대적으로 크다면 중심 화소의 크기를 더 키우고, 중심 화소의 크기가 주변 화소의 크기와 비슷하다면 중심 화소의 크기를 더 낮게 만들어 전체 적으로 가우시안 분포를 만들게 한다. CSF식은 식 (3)과 같다.

$$CSF = 8 \times A5 - (A1 + A2 + A3 + A4 + A6 + A7 + A8 + A9) \tag{3}$$

이러한 분포와 실험을 통하여 식 (4)과 같은 간략화된 HVS 식을 나타내었다.

$$W_{st}(n,m) = \begin{cases} R_1 & Th_1 \le I_{vari}(n,m) < Th_2 \\ R_2 & Th_2 \le I_{vari}(n,m) < Th_3 \\ R_3 & Th_3 \le I_{vari}(n,m) < Th_4 \\ R_4 & Th_4 \le I_{vari}(n,m) < Th_5 \\ R_5 & Th_5 \le I_{vari}(n,m) < Th_6 \\ R_6 & Th_6 \le I_{vari}(n,m) \end{cases} \tag{4}$$

그림 11은 본 논문에서 제안하는 알고리즘에 따른 워터마크 삽입을 도식화 하였다. 그림에서와 같이 워터마크를 삽입하기 위하여 워터마크를 생성하고 원본영상의 특징을 이용하여 워터마크 강도를 계산하고 워터마크 강도와 워터마크를 곱한 결과 값과 원본영상을 더함으로써 워터마크 삽입이 된다.

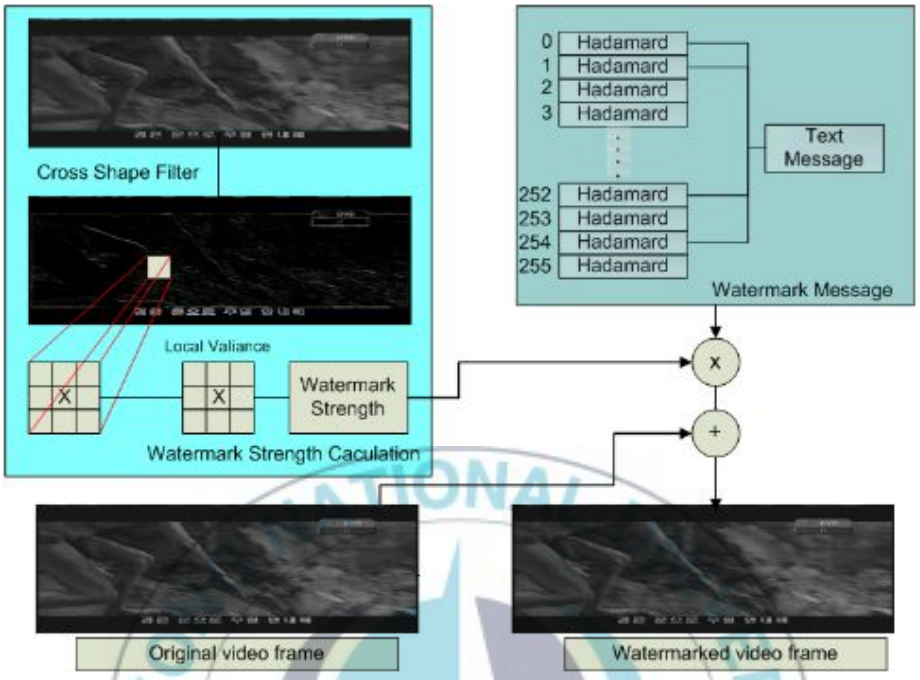


그림 11. 워터마크 삽입도

Figure 11. A block diagram of watermark embedding

워터마크 삽입을 크게 두 부분으로 나누면 워터마크 메시지를 만드는 부분과 워터마크 강도를 계산하는 부분으로 나눌 수 있다. 워터마크 메시지를 만들기 위해 대역확산 기법을 사용하였다. 워터마크 메시지를 만들기 위한 입력 메시지는 저작권 정보이다. 또한 저작권 보호에 필요한 정보를 식 (2)와 같은 이진 신호로 만들기는 어렵기 때문에, 본 논문에서는 텍스트를 사용하여 저작권 정보를 포함 할 수 있는 메시지를 만들었다. 식 (2)와 같이 워터마크 메시지를 메시지와 의사랜덤 신호와의 곱을 통하여 워터마크를 생성하면 곱연산에 의한 계산 복잡도가 증가하게 되므로 메시지로 사용되는 텍스트 정보를 ASCII로 변환하고 그 결과와 의사랜덤신호와의 단순 사상을 통하여 워터마크 메시지를 만들게 된다. 본 논문에서는 의사랜덤신호로 하다마드 신호를 사용하였다. 하다마드 신호

는 각 요소들이 이진인 1, -1로 구성되며 각 행과 각 열이 직교성(orthogonal)을 가지고 있으며, 하나의 심벌로 사용할 수 있다[49]. 이런 특성으로 워터마크 삽입 반복횟수를 비약적으로 높여 검출 효율을 개선할 수 있고, 하드웨어 구현하기에 간단한 구조적 특성을 가지게 된다. 본 논문에서는 영상의 특성에 따른 적응적인 워터마크 강도를 만들기 위해서 영상의 국부영역의 분산 값으로 질감정보를 구한 후 양자화 하여 워터마크 강도를 구하였다. 우선 영상의 질감정보를 구하기 전에 전처리 필터를 사용한다. 국부 영역의 분산 값만을 이용하여 워터마크 삽입 강도를 결정할 경우엔 비가시성과 강인성의 상관관계를 적절하게 맞추기 어려우므로 전처리 필터를 통하여 대략적인 영역을 추출하였다. 추출 되어진 영상에서 국부영역의 분산 값들을 구한 후 양자화 하여 소수의 양자화 값이 아닌 정수의 양자화 인덱스 값을 얻었다. 이 인덱스 값이 워터마크 강도이다. 이 방법을 사용하면 워터마크 강도를 소수 값이 아닌 정수 값으로 구현하여 하드웨어적인 복잡도를 줄여 처리속도의 향상을 가져 올 수 있게 된다.

1.2 워터마크 검출 알고리즘

삽입된 워터마크를 검출하는 방법은 워터마크를 삽입하는 방법에 따라서 틀리게 된다. 본 논문에서는 워터마크 삽입 시 삽입할 메시지를 아스키 코드화하여 의사 랜덤 함수인 하다마드 신호에 대역확산 하였다. 그리고 대역 확산된 신호는 영상의 국부적인 특성을 이용하여 만들어진 워터마크 강도와 곱한 후 원 비디오 신호화 더하여 워터마크가 삽입된 신호로 만들어 진다. 여기서 원 신호원에 어떤 신호가 더해진다는 것은 중첩의 원리에 따라서 워터마크 신호원에서 워터마크 신호를 뺄 수도 있게 된다는 의미가 된다. 워터마크가 삽입된 비디오에서 워터마크를 검출하

기 위하여 Kutter가 제안한 식 (5)를 사용하였다. 이 알고리즘은 워터마크를 영상에 첨가된 잡음이라고 생각하고 먼저 원영상과 잡음을 분리 한 다음 잡음과 변조함수 사이의 자기상관의 차이에 의해서 삽입한 정보를 검출한다.

$$r_i = \langle f(\hat{I}), s_i \rangle \tag{5}$$

여기서 r_i는 검출된 삽입 신호, f는 워터마크 삽입된 신호에서 워터마크를 분리하기 위한 처리 함수, $\langle\ \rangle$는 자기상관함수이다.

그림 12는 워터마크 검출 알고리즘을 도식화하였다. 그림에서와 같이 워터마크가 삽입된 영상에서 워터마크를 추정하기 위하여 워터마크를 삽입하는 과정의 역순으로 진행된다. 워터마크는 원본영상에서 봤을 때는 특정 잡음이 되기 때문에 잡음과 영상을 분리하는 과정을 거치고 분리된 잡음을 키 행렬과 유사성을 검토하여 가장 유사성이 가장 높은 키를 워터마크로 인식하게 된다.

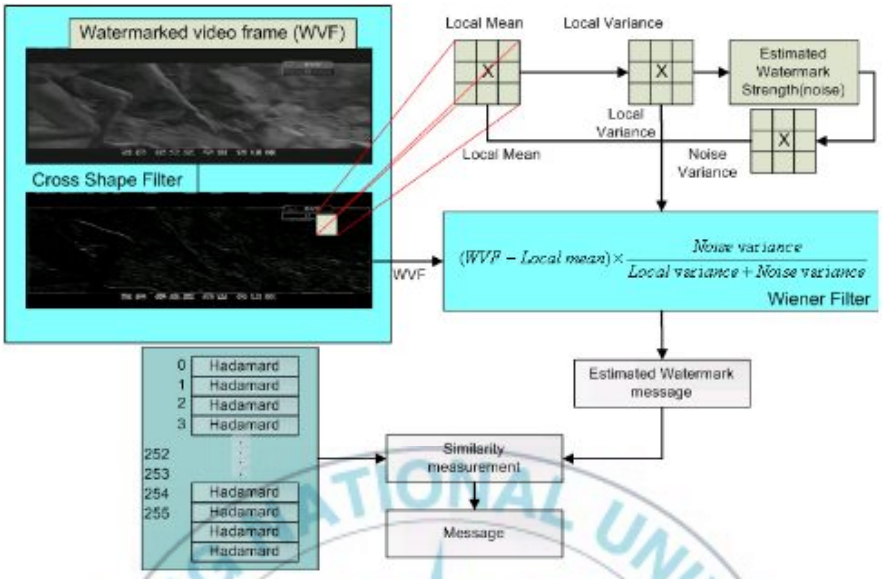


그림 12. 워터마크 검출도

Figure 12. A block diagram of watermark detection

워터마크를 검출하기 위하여 크게 두 과정으로 볼 수 있는데 워터마크를 추정하는 부분과 추정된 워터마크를 통하여 메시지를 검출하는 과정이다. 워터마크를 추정하는 과정은 전술한 것과 같이 워터마크 삽입 과정에서 원 신호에 워터마크를 더하는 방법이기 때문에 중첩의 원리에 의해 분리가 가능하다. 여기서는 분리를 효율적으로 하기 위하여 위너 필터(winner filter)를 사용한다. 위너 필터는 신호와 잡음을 효과적으로 분리 할 수 있는 필터로 워터마크가 삽입된 비디오에서 워터마크를 효율적으로 예측할 수 있는 장점이 있다. 워터마크를 예측하는 과정에서의 결과는 의사 랜덤 신호인 하다마드 신호와의 상관도를 계산한다. 하다마드 신호는 열과 열사이의 직교성이 매우 좋은 신호로 자기 상관도는 매우 높고 교차 상관은 매우 낮은 특징을 가진다. 만약 추정된 워터마크와 하다마드 신호의 각 열과의 상관관계를 계산하여 가장 높은 상관도를 가지

는 하다마드 열이 메시지와의 사상된 하다마드 열로 추정 할 수 있다.

2. 워터마킹 시스템 구현

2.1 워터마크 삽입 알고리즘의 FPGA 구현

워터마크 삽입기를 구현하기 위하여 FPGA의 특성이 고려되었다. 그림 13은 워터마크 삽입 알고리듬의 FPGA 구현에 대한 흐름을 도식화 하였다. 워터마크 삽입 칩은 ALTERA사의 FPGA인 STRATIX를 사용하여 구현하였다. FPGA에 입력되는 비디오 신호는 ANSI/SMPTE 125M 표준화 문서에 정의된 병렬형태의 YCbCr신호이다[48]. YCbCr은 Y와 C의 두 개의 신호선으로 프레임, 필드를 구별하고 수평과 수직을 구분 할 수 있는 제어 신호인 TRS(time reference signal)가 포함되어 있다. 입력 비디오 신호의 Y와 C 데이터에서 TRS신호를 제외한 데이터가 실제 모니터에 재생되고, 모니터에 재생되는 부분의 Y성분에 워터마크를 삽입하였다. TRS를 제외한 Y성분에 워터마크를 삽입하는 이유는 입력되는 전체 비디오 신호에 워터마크를 삽입할 경우 동기신호를 잃어 버려 재생이 되지 않는 문제점이 있기 때문이다. 워터마크 삽입 알고리듬을 FPGA로 구현하기 위하여 전체 구성을 전처리부(CSF), 워터마크 강도 생성부(generate watermark strength), 워터마크 생성부(watermark generator) 그리고 워터마크 삽입부(watermark insert)로 나누어서 구현하였다. FPGA로 입력되는 신호는 고주파 통과 여과기인 CSF에 의해서 여과되는데, 이는 원 비디오 신호에서 에지를 검출하기 위하여 사용한다. 에지를 검출하기 위하여 N*N 윈도우를 가져야 하고, 윈도우는 N^2 바이트의

레지스터이다. FPGA의 입력으로 들어오는 신호는 먼저 비디오 입력 버퍼에 저장되는데, 이 버퍼는 N열의 영상 화소들을 저장한다. HD 비디오를 영상 신호 처리 할 경우 1920*N byte의 버퍼 크기가 요구된다. 버퍼는 쉬프트 레지스터를 이용하여 FIFO(first in first out)형태로 저장되고, 버퍼 레지스터와 CSF를 위한 산술 논리부는 STRATIX FPGA 디바이스에 의해서 구현된다. CSF의 출력 신호는 워터마크 강도 생성부의 입력에 연결된다. 워터마크 강도 생성부는 영상의 국부적인 특성을 이용하게 되는데 저주파 성분과 고주파 성분이 확실하게 구분되는 영상에서 각 화소의 분산 값을 계산하여 주변 보다 고주파 성분이 큰 화소에는 높은 강도를 생성하고 저주파 성분이 큰 화소에는 낮은 강도를 생성하도록 하였다.

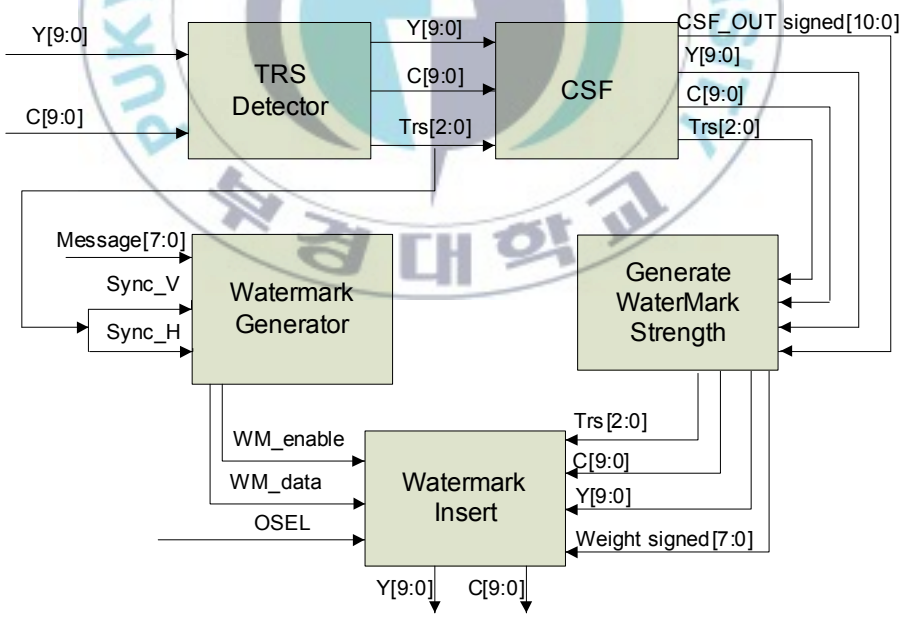


그림 13. 워터마크 삽입 칩 도식화

Figure 13. A block diagram of watermark embedding chip

워터마크 생성부에서는 저작권정보를 텍스트형태로 만들어 입력메시지로 사용하고 하다마드 행렬은 ROM(read only memory)형태로 FPGA에 구현한다. 입력되는 텍스트 형태의 메시지를 ASCII형태로 변환하여 ROM에 저장된 하다마드 행렬의 번지 값과 ASCII값을 대조하여 해당하는 번지의 하다마드 값들을 워터마크 메시지로 사용한다. 워터마크 메시지와 영상의 국부적인 질감 특성을 이용하여 만든 워터마크 강도를 곱하여 워터마크를 만든다. 워터마크 메시지는 TRS신호 수평, 수직 동기 신호를 이용하여 실제 모니터에 재생되는 화면의 화소에만 더해지도록 구현하였다.

SMPTE에 정의되어 있는 BPDI(bit-parallel digital interface)의 YCbCr신호는 SAV(start of active video), EAV(end of active video), V(vertical flag), H(horizontal flag), F(field flag)로 구성되어 있는 TRS신호를 포함하고 있고, 이 신호를 사용하여 입력되는 비디오 신호에서 블랭크신호(blank signal)를 제외한 화면에 표시되는 비디오 데이터를 찾을 수 있다. BPDI 신호에 포함된TRS를 이용하면 모니터에 재생되는 비디오 영역을 찾을 수 있고, 이 부분의 화소에 워터마크를 삽입 한다.

그림 14는 워터마크 삽입기에 대한 RTL이다. 그리고 그림 15는 Quartus II에서의 시뮬레이션 결과를 보여준다. 입력 비디오 신호인 YCbCr은 Y와 CbCr이 각각 10bits의 크기를 가진다.

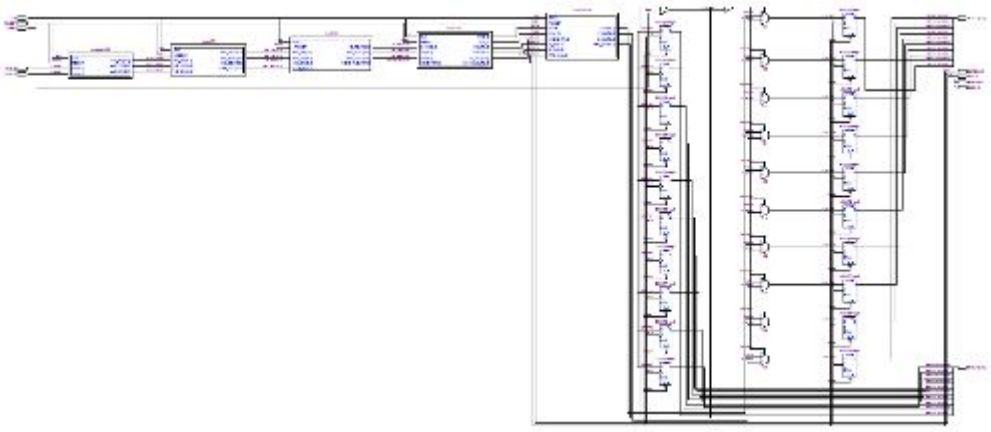


그림 14. 워터마크 삽입기 RTL
Figure 14. RTL of watermark embedder

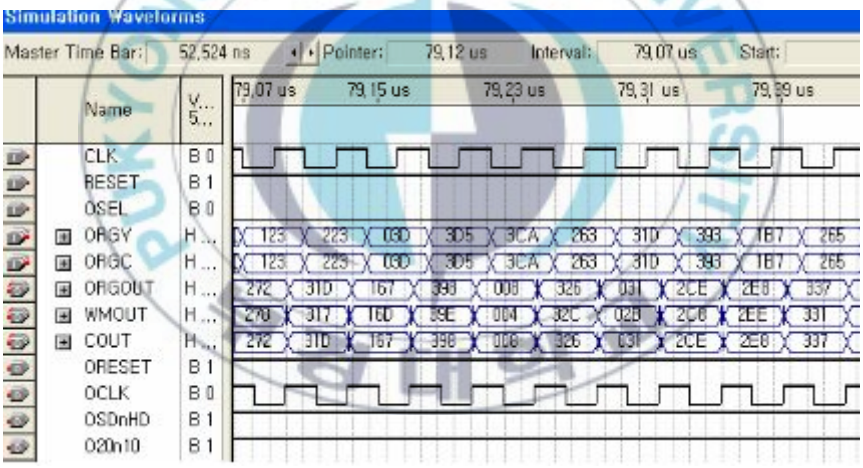


그림 15. Quartus II에서의 모의실험 결과
Figure 15. Simulation result with Quartus II

10비트중 상위 8비트만이 데이터를 가지고 하위 2비트는 다른 용도로 사용하기 위하여 예약되어 있다. ORGOUT은 워터마크가 삽입되지 않은 영상이고, WMOUT은 워터마크 삽입된 영상을 나타낸다. 워터마크가 삽입되지 않은 부분과 삽입된 부분을 비교했을 때 값의 차이를 알 수 있

다. 영상에 삽입되는 워터마크의 크기는 워터마크 강도에 따라서 -6에서 +6까지의 차이를 가진다. 시뮬레이션 결과에서 79.15us근처의 ORGOUT와 WMOUT을 비교하면 ORGOUT은 값은 16진수 167을 가지고 WMOUT은 16진수 16D값을 가지며 그 차이는 6이다.

2.2 워터마크 삽입 시스템 구현

그림 16는 워터마크 삽입 시스템을 나타내었다. 그림에서와 같이 시스템은 입력 신호를 처리하는 부분과 워터마크 삽입 알고리즘을 구현한 FPGA 그리고 출력 신호를 처리할 수 있는 부분으로 구성되어 있다.

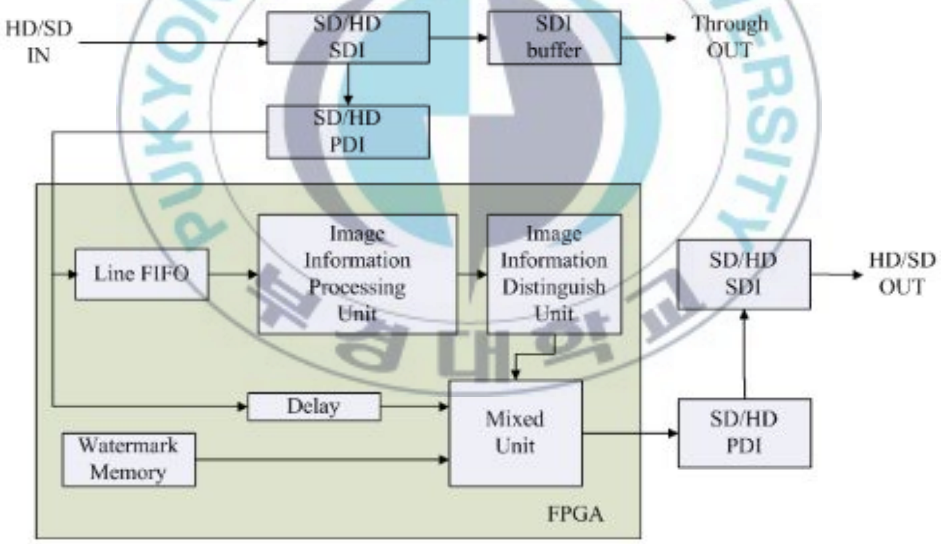


그림 16. 워터마크 삽입 시스템
Figure 16. Watermark embedding system

워터마크 삽입 시스템을 구현하기 위하여 HD급의 고화질 비디오 신호에 대하여 고려하였다. HD/SD 비디오 소스는 SDI 형태의 신호로 입력

되지만, FPGA에서 비디오 신호처리를 위하여 PDI(parallel digital interface)신호 형태로 변환하여 사용한다. 비디오 신호를 PDI 형태의 신호로 변환하여 FPGA의 입력으로 전달할 수 있도록 GINNUM사의 GS1560A 디바이스를 사용하였고, 워터마킹 처리가 완료된 PDI신호를 SDI신호로 만들어 출력하기 위하여 GS1532 디바이스를 사용하였다. GINNUM사의 GS15360A와 GS1532 디바이스는 SD급 및 HD급의 신호에 대한 신호 변환이 가능하여 HD급 실시간 시스템을 만드는데 적당한 디바이스이다. FPGA의 Clock과 Reset은 입력으로 들어오는 비디오 신호의 데이터와 동기화 하기위하여 GS1560A의 Clock과 Reset을 사용하고, FPGA에서 처리가 완료된 비디오 데이터를 출력하여 GS1532 디바이스의 입력으로 전달 할 경우 역시 GS1560A의 Clock과 Reset을 사용하여 동기화 하였다.

2.3 워터마크 검출 시스템 구현

그림 17은 본 논문에서 구현한 워터마크 검출 시스템을 나타낸다. 워터마크가 삽입된 비디오 데이터는 PDI 신호로 변환되고, FPGA를 이용하여 입력되는 워터마크 신호와 PCI 인터페이스 신호와의 동기를 맞추었고, PCI 인터페이스를 사용하여 워터마크가 삽입된 비디오 신호를 PC로 보내었다. PC에서는 입력으로 들어오는 워터마크 삽입된 비디오 신호에서 워터마크를 검출 할 수 있도록 프로그램이 제작되어 있다.

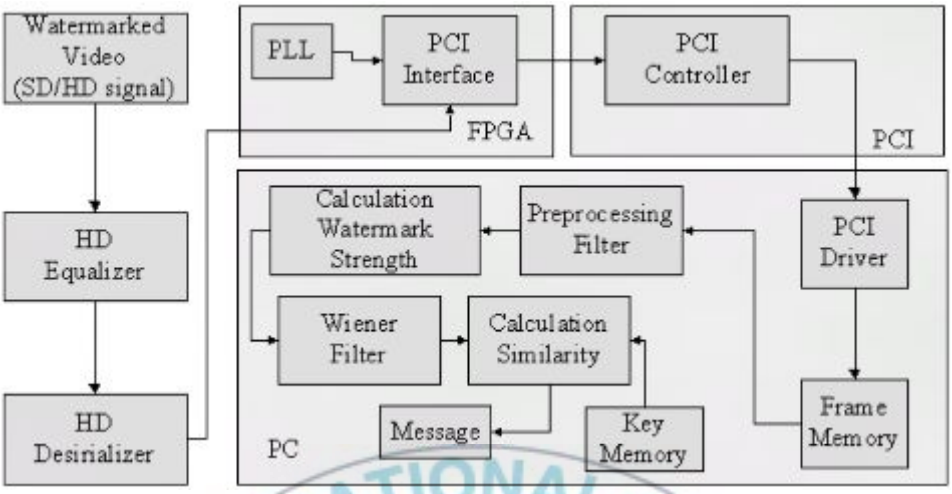


그림 17. 워터마크 검출 시스템

Figure 17. Watermark detection system

워터마크가 삽입된 비디오 신호는 HD 이퀄라이즈와 디이퀄라이즈를 통하여 YCbCr의 PDI 신호로 변환되고 FPGA에서 이를 처리 할 수 있도록 만들어져 있다. FPGA 칩은 PCI를 제어할 수 있는 신호 발생과 워터마크가 삽입되는 신호를 처리하기 위한 신호 발생 두 부분을 위하여 두 개의 신호발생기를 사용하게 된다. PLL(phase locked loop)을 통하여 들어오는 신호를 PCI 제어 신호와 같은 크기의 동작을 위하여 66MHz의 클록과 본 시스템에서 사용한 DVD의 SD급 신호를 위한 27.5MHz의 클록을 만들어서 사용한다. FPGA에서는 이렇게 두 가지 클록을 만들어서 입력으로 들어오는 워터마크가 삽입된 신호를 PCI 제어 칩으로 전달하는 역할을 하게 된다. PCI 칩은 전달 받은 SDI 신호를 컴퓨터로 전달하는 역할을 하게 된다. 컴퓨터에서는 PCI 제어기에서 들어오는 신호를 효과적으로 처리하기 위하여 PLX에서 제공되는 PCI 제어기 라이브러리를 이용하여 설정할 수 있는 프로그램을 만들고, 워터마크 신호를 캡처하여 소프트웨어적인 방법을 통하여 워터마크 검출을 하게 된다.

3. 비디오 암호화 구현

3.1 실시간 비디오 암호화 구현 방법

본 논문에서는 비디오 암호화를 구현하기 위하여 다음과 같은 방법을 사용하였다. 본 논문에서의 비디오 암호화는 비가시성을 만족해야 하며 암호화에 따른 비디오 프레임의 건너뜀 없이 실시간 암호화가 가능하여야 한다. 전술한 비디오 암호화에서와 같이 실시간 암호화를 위하여 비디오 전체를 암호화 하는 방법을 공간 전체 화소에 적용하면 많은 계산량에 따른 시간 소모가 크기 때문에 실시간이 어렵고 또한 각 픽셀을 암호화 하는 것으로 비가시성이 크지 않다. 비디오 부분 암호화는 화소에 직접 적용할 수 있고, 또는 비트스트림에 적용하는 방법이 주로 쓰이는데 본 논문에서 적용할 비디오는 DVD 또는 Bru-ray에서 출력인 YUV 신호이고 이 신호는 비디오 압축 기법을 사용하여 압축된 것이 아니라 비압축 비디오이다. 이와 같이 비압축 비디오를 화소 단위로 암호화 하면 비가시성이 떨어지기 때문에 화소를 직접 암호화하는 것은 본 논문의 목적에 맞지 않다. 암호화 후 비가시성을 높이고, 프레임 건너뜀 없이 실시간 암호화를 위하여 스크램블 방식을 사용하였다. 스크램블링 기법을 사용한 암호화 방법은 영상 화소의 계조값을 변화시키지 않고 위치 정보를 변화시킴으로서 비디오를 암호화하는 방법이다. 스크램블링 방식은 영상 데이터에 대한 위치를 스크램블링하는 것과 키를 암호화하는 기능으로 이루어져 있다. 영상 데이터의 스크램블링은 line rotation 및 line permutation을 이용하게 된다. 하지만 강인한 스크램블을 위하여 필요한

것은 키의 보안인데, 본 논문에서는 키의 보안을 AES 암호화 알고리즘을 사용하여 보안을 높였다. 또한 프레임의 건너뜀 없이 실시간으로 구현하기 위하여 16가지 스크램블 패턴을 메모리에 저장 할 수 있도록 하였다.

그림 18은 본 논문에서 제안하는 방법을 도식화 하였다. 입력 비디오 시퀀스에서 하나의 라인을 저장하고, 이 라인을 스크램블링하게 된다. 스크램블링에 필요한 키 정보는 AES알고리즘을 통하여 생성된 암호화 데이터를 16*4의 행렬 형태로 만들어 사용하고, 스크램블 패턴은 16가지를 만들어 사용하였다.

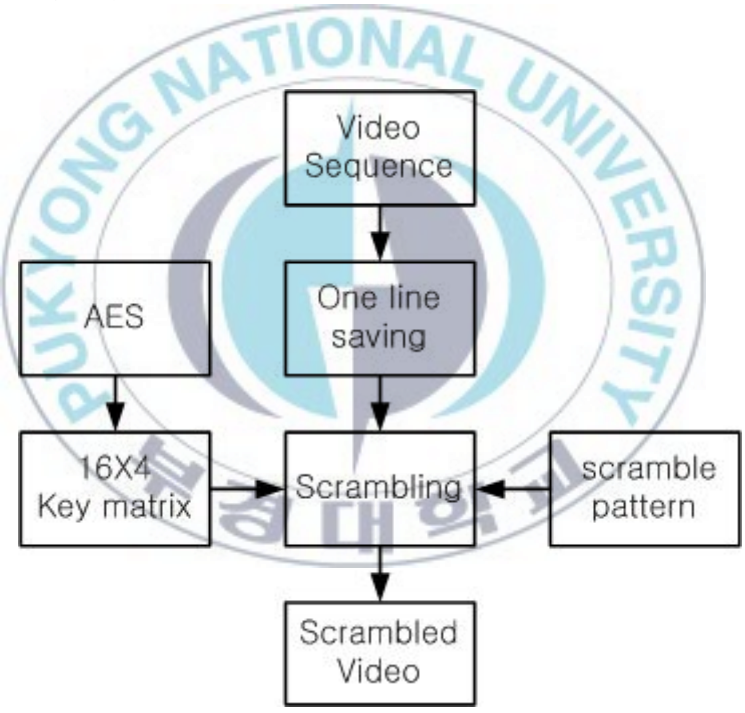


그림 18. 비디오 암호화 블록도

Figure 18. A block diagram of video encryption

그림 19는 키 정보를 생성하는 방법에 대하여 나타내었다. 그림에서와 같이 강력한 키를 얻기 위하여 AES암호화 알고리즘을 사용하여 키

를 생성하였다. AES 암호화 알고리즘은 128비트, 192비트, 256비트의 암호화된 출력을 가진다. AES 암호화 알고리즘은 현재까지 가장 강력한 암호화 알고리즘으로 알려져 있고, 소프트웨어 방법으로 구현하였을 경우 128비트 암호화에 걸리는 시간은 펜티엄프로 CPU의 계산 기준으로 2μs정도의 속도를 가지기 때문에 키 생성에서 소모되는 계산량은 거의 무시될 수 있다. 그림에서는 AES 결과 128비트 중 64비트를 사용하고, 64비트 결과를 4비트씩 나누어 16X4의 행렬을 만들게 된다. AES의 강력한 암호화 결과를 키로 사용하기 때문에 스크램블을 역으로 추정하기 어렵게 된다.

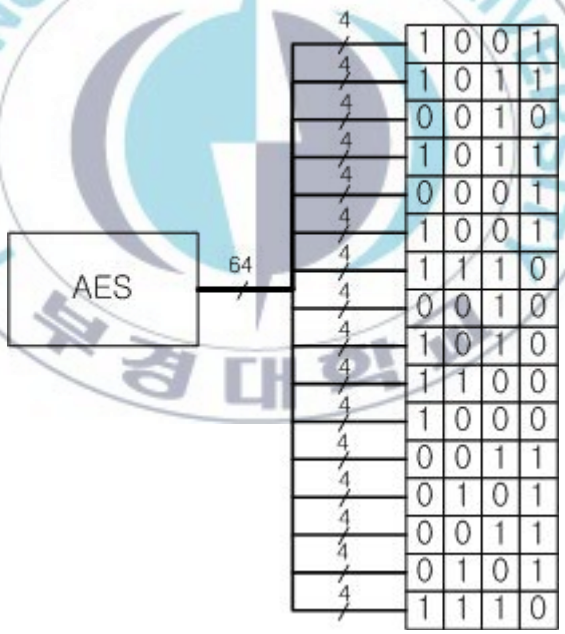


그림 19. 키 생성 방법

Figure 19. A block diagram of key generation method

그림 20은 스크램블링 패턴을 생성에 사용된 방법을 나타내었다. 패턴

생성 방법은 비가시성을 높이기 위하여 하나의 라인을 16가지 방법으로 나누게 된다. 나누는 방법은 첫 번째는 라인을 반으로 나누어 앞뒤를 썩고, 다음은 이전의 나누어진 것을 다시 반으로 나누어 앞뒤로 썩는다. 이렇게 16번을 하여 만들게 된다.

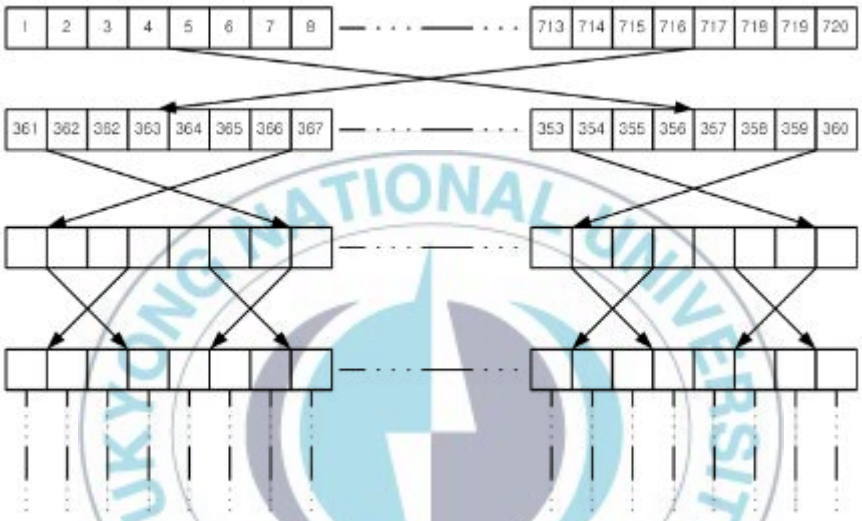


그림 20. 패턴 생성 방법

Figure 20. A block diagram of pattern generation method

3.2 하드웨어 구현 방법

본 논문에서 하드웨어 구현을 위한 방법은 전술한 워터마크 삽입기 하드웨어 구현과 유사하게 진행된다. 시스템 구현을 위하여 사용된 FPGA는 ALTERA사의 STRATIX이다. STRATIX의 가장 큰 특징 중 하나가 내부 TriMatix 메모리를 가지는데 내부 TriMatrix 메모리의 종류는 총 3가지(M512, M4K, M-RAM)이고, 그 크기가 1,669,248비트이다. 외부 메모리를 사용하여 HD 해상도 크기를 실시간으로 암호화를 하게 되면 프

레임의 건너뜀을 하여야 한다. 본 논문에서와 같이 실시간으로 라인별 스크램블링을 하기 위하여 고속의 내부 메모리를 사용하여야 한다. 그림 21은 본 논문에서의 비디오 암호화 하드웨어 구현에 대한 블록도를 나타내었다.

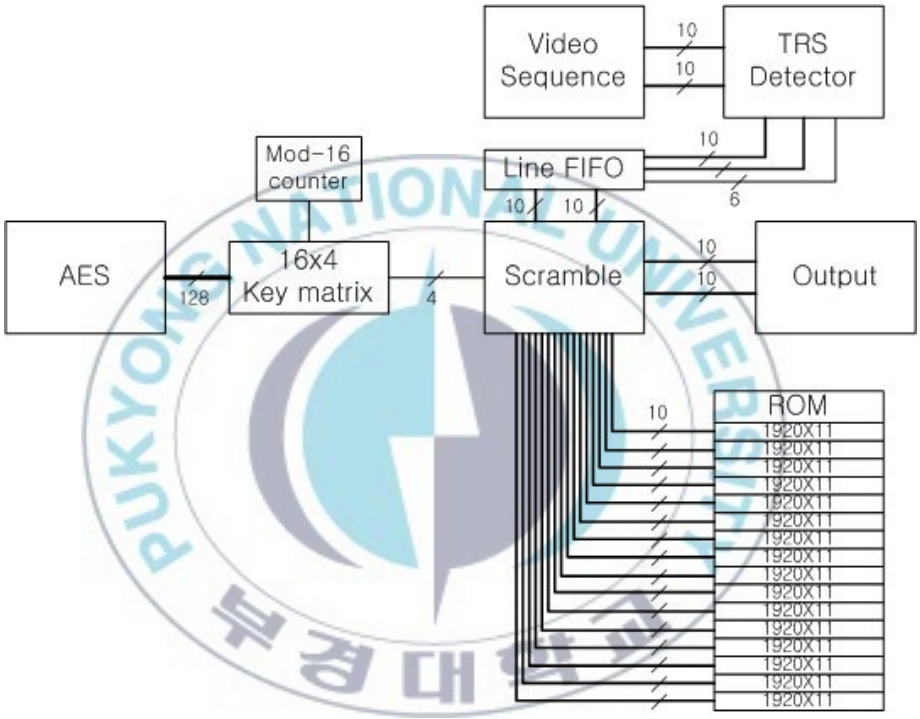


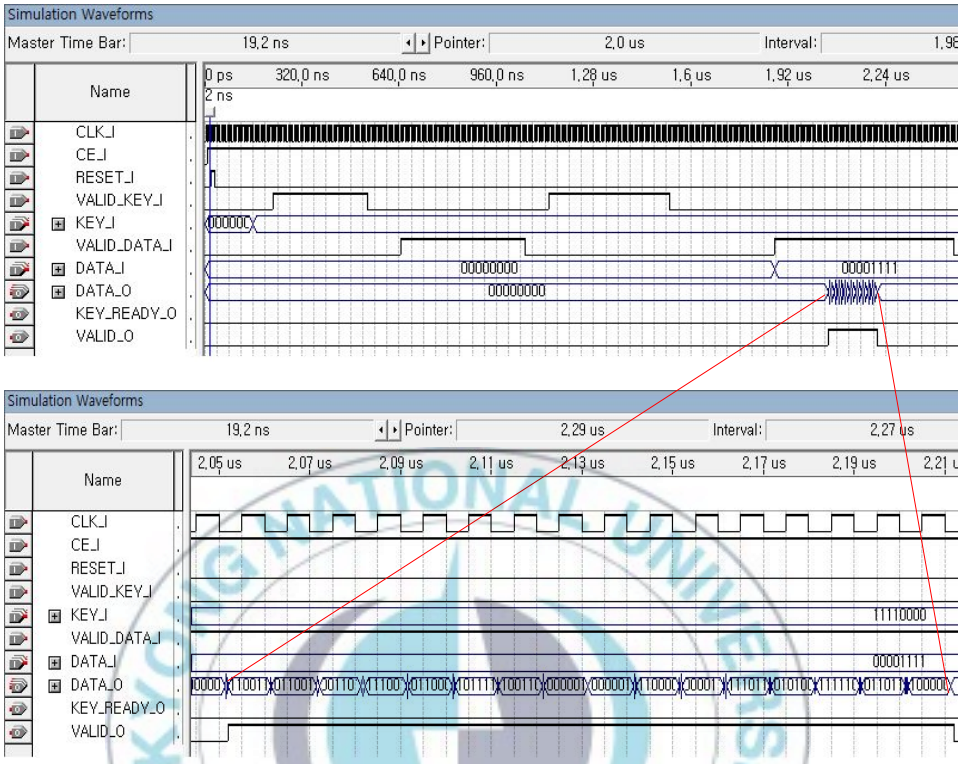
그림 21. 비디오 암호화 하드웨어 구현을 위한 블록도
Figure 21. A block diagram for hardware implementation of video encryption

비디오 암화를 위하여 입력되는 비디오 시퀀스는 YCbCr의 20비트로 되어 있고, Y가 10비트이고 CbCr이 10비트를 가지게 된다. Y 또는 CbCr정보를 이용하여 TRS 신호를 얻게 되고 TRS신호 및 YCbCr 신호는 하나의 라인을 저장할 수 있는 FIFO를 통하여 저장되게 된다. TRS

를 검출하는 시간과 동시에 AES에서는 키 행렬을 만들게 된다. AES를 통하여 키를 만드는 시간동안 비디오 신호는 계속 들어오게 되지만, 이때 들어오는 비디오 신호는 “active video region”이 아니기 때문에 암호화에는 영향을 미치지 않게 된다.

그림 22는 키 생성을 위한 AES 암호화 알고리즘의 VHDL로 구현한 결과를 나타내었다. 그림에서 (a)는 AES 암호화에 대한 RTL을 나타낸 것이고, (b)는 모의실험 결과를 나타낸 것이다.

(a) RTL 그림

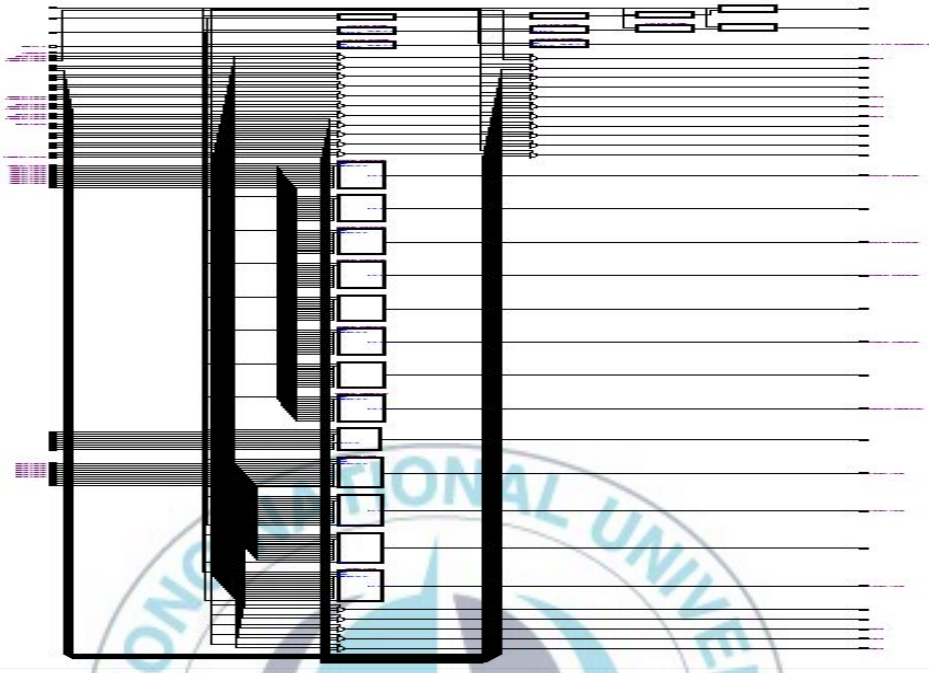


(b) 시뮬레이션 결과

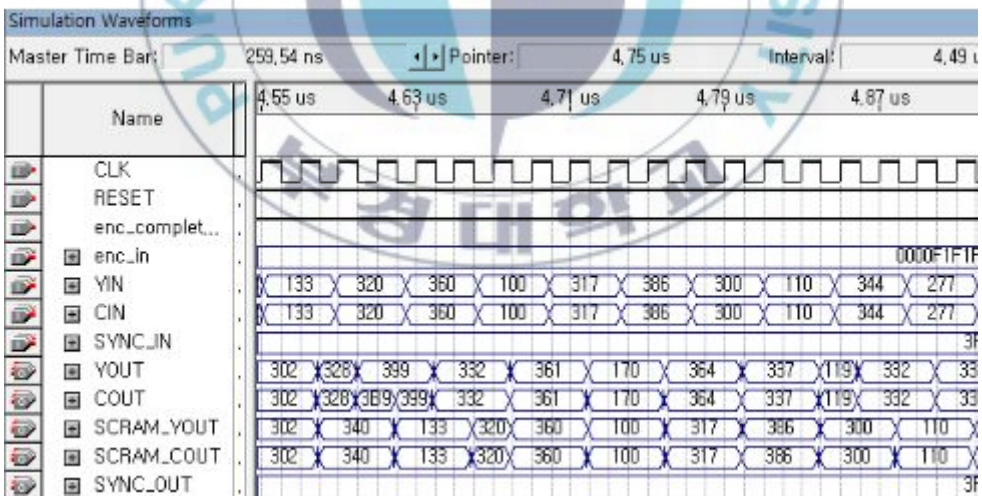
그림 22. AES 암호화 VHDL 구현 결과

Figure 22. Result of VHDL implementation for AES encryption

모의실험 결과 약 0.2μs안에 8비트의 결과가 16개가 출력되는 것을 볼 수 있다. 즉 약 0.2μs내에 128비트를 출력하는 것을 볼 수 있다. 그림 23은 암호화 결과를 나타내었다. 그림에서 (a)는 VHDL로 만든 결과를 RTL(register transfer level)로 보여진 것이고, (b)는 모의실험 결과를 나타낸 것이다. (b)에서 enc_in은 AES로 암호화 된 데이터를 입력으로 받은 것이고, enc_in에 의한 데이터는 16*4크기의 SRAM에 저장 된다.



(a) RTL 그림



(b) 시뮬레이션 결과

그림 23. 비디오 암호화 VHDL 구현 결과

Figure 23. A result of VHDL implementation of video encryption

저장된 데이터는 스크램블을 위한 행렬로 쓰이게 되고, 사전에 생성한 스크램블 패턴을 ROM에 저장하여 SRAM에 저장된 16*4에 의해서 ROM에 저장된 패턴을 인식하여 패턴의 순서대로 화소 순서를 변경하여 출력으로 내 보내게 된다. 출력으로 보내진 데이터는 SCRAM_YOUT, SCRAM_COUT이다. SCRAM_YOUT은 밝기 성분을 스크램블 한 것이고, SCRAM_OUT은 색차 성분을 스크램블 한 결과이다.

Ⅳ. 실험 결과 및 분석

본 장에서는 제안한 방법에 대한 실험 결과와 분석을 기술하고 있다. 제안한 방법의 성능을 평가하기 위한 실험 방법과 내용, 실험의 과정, 실험의 결과와 의의 등을 기술하였다.

본 장의 구성은 다음과 같다. 먼저 1절에서는 실험 환경에 대해서 설명한다. 실험에 사용된 비디오 시퀀스에 대하여 설명하고 실험을 위한 시스템 구성 및 환경에 대하여 기술하고 있다. 2절에서는 실험 결과와 분석을 기술하였다. 알고리즘의 검증을 위한 소프트웨어 방법의 실험 및 하드웨어 구현을 통한 결과를 나타내고 있다.

1. 실험 환경

본 절에서는 제안한 방법의 실험을 위한 환경에 대해서 알아본다. 사전 정의되는 사항들을 알아보고 실험을 수행한 환경과 실험 방법의 내용들을 정리하였다.

1.1 워터마킹 소프트웨어 실험 환경

제안한 방법들의 성능 평가를 위한 실험 환경은 다음과 같다. 제안한 워터마킹 알고리즘의 소프트웨어적 검증을 위하여 사용한 PC는 Core Duo 2.4GHz의 CPU와 2G의 RAM으로 구성되었고, 워터마킹 프로그램은 Microsoft Visual C++ .net 8.0을 이용해서 구현하였다. 제안한 알고리즘의 강인성 테스트를 위하여 워터마크 삽입된 영상을 H.264압축 표준에 근거한 압축 방법인 X264를 사용하였고, 삽입된 워너마크를 검출하기 위하여 X264의 디코더는 JM15의 디코더를 사용하였다. 실험을 위해 사용된 비디오 데이터는 3가지 시퀀스이고 각각의 시퀀스의 포맷은 YUV 4:2:0이고 300프레임으로 구성되어있다. 또한 실험 영상의 크기는 CIF(352*288)를 사용하였다. 표 2는 실험에 사용된 영상에 대한 요약을 나타내었다. 그림 24는 표2에서 설명된 비디오 시퀀스의 정지 영상을 나타내고 있다.

표 2. 워터마킹 알고리즘 테스트를 위한 비디오
Table 2. Video titles for a test of watermarking algorithm

Title	포맷	크기	프레임수	특징
Akiyo	4:2:0	CIF	300	움직임이 적고, 약한 질감
Coastguard	4:2:0	CIF	300	횡움직임이 크고, 강한 질감
Foreman	4:2:0	CIF	300	객체 움직임이 크고, 강한 질감

(a) akiyo (b) coastguard (c) foreman

그림 24. 비디오 표본
Figure 24. Video samples

제안한 워터마킹 알고리즘의 비가시성과 강인성을 평가하기 위하여 워터마크를 비디오에 삽입 후 H.264를 통한 인코딩/디코딩을 통하여 강인성 테스트를 하였고, 이와 병행하여 화질 측정을 통한 비가시성을 측정하였다.

1.2 워터마킹 및 암호화 하드웨어 시스템 환경

본 논문에서 제안한 워터마킹 알고리즘을 구현하기 위한 하드웨어 구성을 두 가지로 하였다. 첫 번째는 SD급에서의 워터마크 삽입 및 검출을 위한 시스템 구성이고, 두 번째는 HD급의 워터마크 삽입 및 검출을 위한 시스템 구성이다. 실시간 비디오 암호화는 후자의 시스템을 사용하여

실험하였다. SD 및 HD에 따른 구조의 차이는 없지만 사용되는 장비가 다르다. 먼저 그림 25와 26은 SD 및 HD급 워터마킹 시스템의 구성을 나타내었다. SD급의 경우 워터마크가 삽입된 영상과 워터마크가 삽입되지 않은 영상 사이의 주관적인 화질을 비교하기 우하여 17인치 LCD를 사용하였고, 비디오 신호원은 SD급(720*480)의 신호를 발생 시킬 수 있는 DVD 재생기를 사용하였다. DVD 재생기의 출력은 YPbPr의 세 가지 신호원으로 구성되어 있는데 이를 디지털 신호로 변환하기 위하여 AD/DA 비디오 변환기를 사용하였다. 변환된 신호는 SDI(serial digital interface)신호이며 YCbCr로 표현된다. 워터마크 검출은 PCI 컨트롤러를 이용하여 비디오 신호를 캡쳐 한 후 PC에서 구현된 워터마크 검출기를 통하여 검출 결과를 확인할 수 있도록 구성하였다. 또한 HD급의 워터마킹 시스템의 경우 Full HD를 표현 할 수 있는 26인치 1920*1200해상도의 LCD를 사용하여 워터마크가 삽입된 비디오와 워터마크가 삽입되지 않은 비디오를 인간 시각적인 즉, 주관적인 화질 비교를 할 수 있도록 구성하였고, 비디오 소스는 Sony사의 Playstation3을 사용하여 Bru-ray를 재생하였다. Bru-ray의 경우 Full HD 해상도인 1920*1080을 지원한다. HD의 경우 SD보다 4배 빠르기 때문에 고속의 비디오 데이터 전송이 가능한 HDMI(high definition multimedia interface)형식의 데이터를 입력으로 사용하였다. 워터마크 삽입기는 SDI형태의 신호를 처리 할 수 있도록 설계되어 있기 때문에 HDMI/HD-SDI 변환기를 사용하여 HDMI 신호를 HD-SDI신호로 변환하였다. HDMI/HD-SDI 변환기는 AJA사의 HA5를 사용하였고, HD-SDI/HDMI는 AJA사의 Hi5를 사용하였다. 워터마크 검출을 위하여 Blackmagic사의 PCI-E형태의 비디오 저장 보드인 DeckLink HD Extreme을 사용하였다.

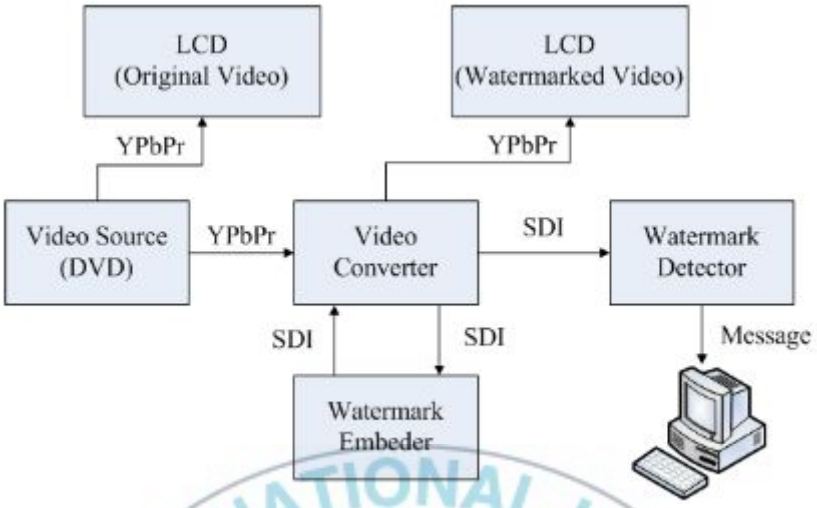


그림 25. SD급 워터마킹 시스템 구성도

Figure 25. A block diagram of watermarking system of SD format

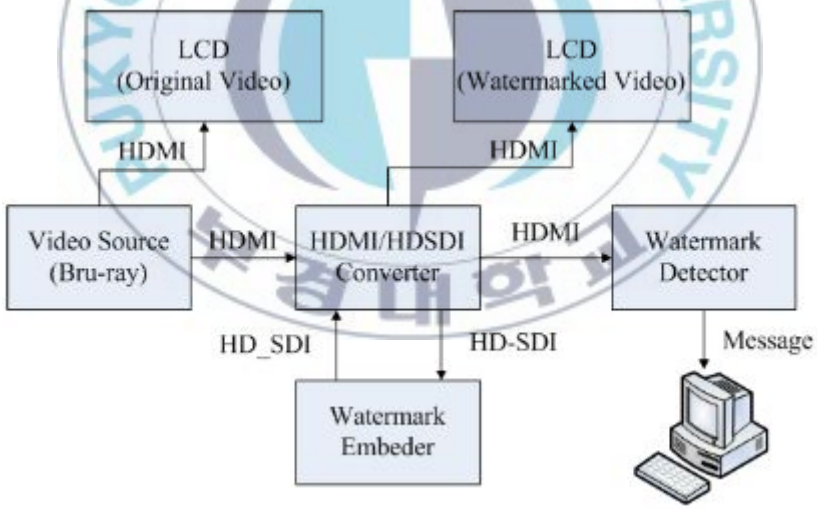


그림 26. HD급 워터마킹 시스템 구성도

Figure 26. A block diagram of watermarking system of HD format

그림 27은 본 논문에서 제안하는 워터마킹 및 암호화를 위한 하드웨어 보드를 나타내었다.

그림 27. 비디오 처리 보드
Figure 27. Video processing board

그림에서 ALTERA로 표시된 부분이 FPGA(field programmable gate array)이고 ALTERA사의 STRATIX를 사용하였다. ALTERA사의 STRATIX는 내부에 세 가지 형태의 SRAM(static random access memory)을 가지고 있어 FPGA 내부의 메모리를 통하여 고속으로 데이터 처리가 가능하고, 100MHz의 고속 처리가 가능한 FPGA이다. HD-SDI 입력을 위하여 GINNUM사의 최신 칩 세트를 사용하였다. 이 칩 세트는 SD 입력도 가능하다. 이퀄라이저(equalizer) 부는 GS1524, 디시리얼라이저(deserializer) 부는 GS1560을 사용하였다. 액티브 루프 쓰루(active loop through)는 GS1560의 출력단을 별도의 드라이버 칩 없이 사용하였다.

디시리얼라이저를 거친 HD-SDI 신호는 Y/C 각각 10비트의 병렬데이터로 바뀌어 FPGA에 입력된다. 그 외 필드 비트(field bit), 수직 블랭킹(vertical blanking), 수평 블랭킹(horizontal blanking) 신호가 FPGA로 입력된다.

디시리얼라이저로부터 추출되는 74.25MHz의 클록 신호 역시 FPGA로 입력되어 메모리 라이트 클록 및 컨트롤 클록으로 사용된다. FPGA로부터 비디오처리가 완료된 비디오가 Y/C 20비트로 출력되면 National CLC030 시리얼라이저/케이블 드라이버를 통해 HD-SDI 신호가 만들어진다. 이때 사용되는 클럭이 위에서 설명한 외부 레퍼런스에 젠록(gen lock)된 클록이며, 이 클록의 품질이 좋아야 SMPTE 규격의 지터 범위를 만족하는 SDI 신호가 만들어진다.

2. 실험 및 구현 결과

1.1 워터마킹 알고리즘 소프트웨어 실험 결과

제안한 워터마킹 알고리즘의 성능을 평가하기 위하여 앞에 항에서 언급한 실험 비디오 영상 세 가지를 사용하여 다음과 같은 방법으로 실험을 진행하였다. 먼저 제안한 알고리즘을 통하여 워터마크 삽입 및 검출이 가능한지에 대한 실험, 두 번째 삽입된 워터마크 정보에 대한 강인성 실험, 마지막으로 원본 영상과 워터마크 삽입 영상간의 객관적인 화질 차이에 대한 실험이다. 그림 28은 워터마크 삽입 및 검출을 위한 프로그램을 나타내었다.

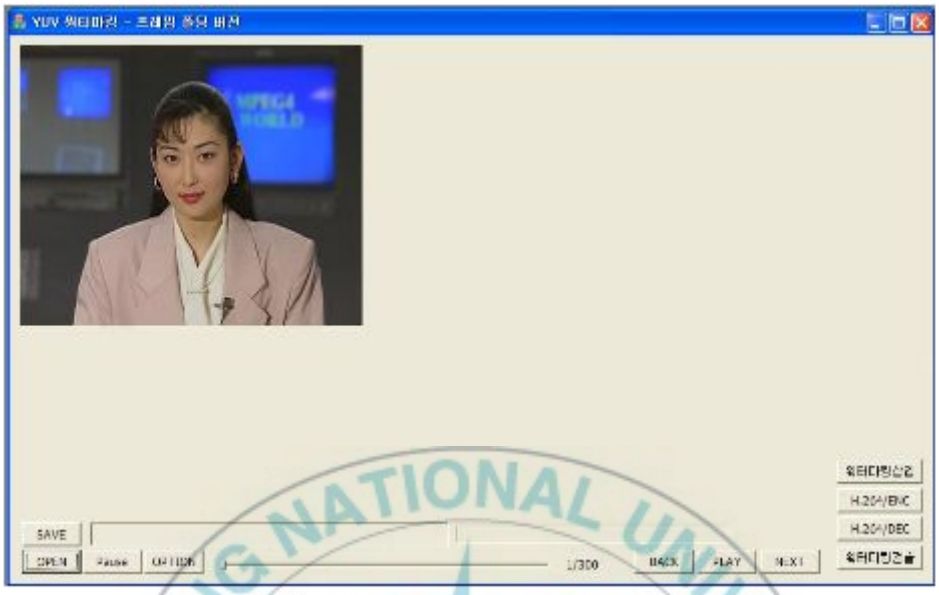


그림 28. 워터마킹 실험 프로그램

Figure 28. A program for watermarking test

위의 프로그램을 통하여 워터마크 정보를 삽입하고 워터마크가 삽입된 비디오 데이터에서 워터마크를 추출할 수 있도록 구현되었다. 그림 29는 강인성 검증을 하지 않고 제안한 알고리즘만을 사용하여 워터마크를 삽입하고 검출한 결과를 나타내었다.

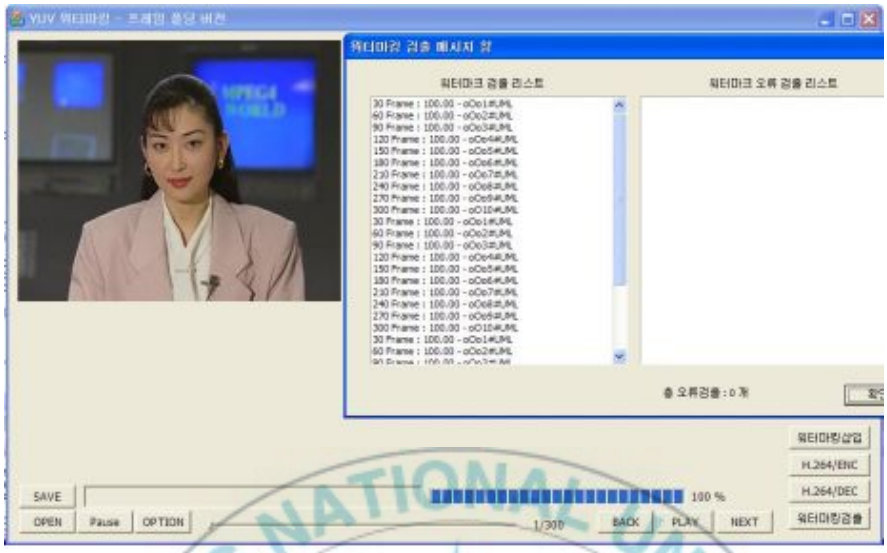


그림 29. 워터마크 검출 결과

Figure 29. A result of watermark detecting

위의 그림에서와 같이 제안한 방법을 사용하여 워터마크를 삽입하고 검출결과 모두 검출됨을 알 수 있다. 그림 28에서는 30 프레임 마다 정보를 달리하여 워터마크를 삽입하고 30 프레임을 누적하여 유사도를 측정한다. 유사도 측청에서 가장 유사한 형태의 값이 삽입된 워터마크로 인식하여 그 결과를 나타내게 된다.

표 3에서 표5은 워터마크 강인성에 대한 실험 결과이다. 실험 결과 'akiyo' 영상의 경우 움직임이 적고 잡음이 적기 때문에 낮은 양자화 파라미터(quantization parameter) 크기에서도 고압축이 가능하고 이에 따라 워터마크 검출률이 상대적으로 좋게 나타난다. 'coastguard'는 움직임이 크지만 작은 객체의 움직임에 의해 움직임 벡터 크기가 크지 않기 때문에 'akiyo'보다는 압축률이 떨어지지만 높은 압축률에서 검출이 가능하다. 하지만, 'foreman'의 경우 객체의 크기가 매우 크고 움직임이 많기 때문에 양자화 파라미터 값을 높여도 압축률이 상대적으로 떨어지며, 다

른 두 비디오 시퀀스에 비해 상대적으로 워터마크 검출률이 떨어지는 경향을 보인다. 하지만, 100:1 정도의 높은 압축률에서 'akiyo'의 경우 82.5%, 'coastguard'의 경우 71.25%의 검출률을 보이므로 실제응용에서 사용하기에 충분한 검출률을 나타내고 있다.

표 3. 'akiyo'에 대한 검출 결과

Table 3. Results of watermark detection about 'akiyo'

원본 파일 크기(A) (Byte)	압축후 파일크기(B) (Byte)	압축비 (B/A*100) (%)	양자화크기	검출률 (%)
45,619,200	35,0252	0.76777322	26	60
45,619,200	46,7878	1.02561641	24	82.5
45,619,200	64,9971	1.42477509	22	85
45,619,200	85,8194	1.8812123	20	100

표 4. 'coastguard'에 대한 검출 결과

Table 4. Results of watermark detection about 'coastguard'

원본 파일 크기(A) (Byte)	압축후 파일크기(B) (Byte)	압축비 (B/A*100) (%)	양자화크기	검출률 (%)
45,619,200	460,855	1.010222	36	71.25
45,619,200	689,069	1.51048	34	86.25
45,619,200	981,080	2.150586	32	100
45,619,200	1,434,913	3.145415	30	100

표 6. ‘foreman’에 대한 검출 결과

Table 6. Results of watermark detection about ‘foreman’

원본 파일 크기(A) (Byte)	압축후 파일크기(B) (Byte)	압축비 (B/A*100) (%)	양자화크기	검출률 (%)
45,619,200	1,325,241	2.905007	28	100
45,619,200	1122596	2.490797	30	87.5
45,619,200	767626	1.682682	32	60
45,619,200	542694	1.189618	34	36.25

그림 30은 원본 영상(a)과 워터마크 삽입 영상(b) 그리고 워터마크 삽입된 영상을 X264 코덱을 통하여 압축 후 H.264 디코드를 통하여 디코딩 한 결과 영상(c)이다.

(a) 원본 비디오 (b) 워터마크 삽입된 비디오 (c) H.264로 압축된 비디오

그림 30. 원영상과 워터마크 삽입된 영상 비교

Figure 30. Comparison of a orignal video with a watermarked video

위의 그림에서와 같이 원 영상과 워터마크가 삽입된 영상과의 화질 차이를 시각적으로 인지하여 구별하기 어려움을 알 수 있고 세 개의 영상의 차이가 거의 나지 않음을 시각적으로 확인 가능하다.

1.2 비디오 암호화 소프트웨어 실험 결과

제안한 비디오 암호화의 결과를 나타내기 위하여 SD급 비디오 다섯 가지를 이용하여 그 결과를 평가하였다. 그림 31은 실험 결과를 나타내기 위하여 만들어진 프로그램이다. 그림에서와 같이 프로그램은 세 단계

의 과정을 거치게 되는데 첫 번째 단계는 AES 암호화 알고리즘을 통한 4비트 16개의 행렬과 길이 720을 가지는 16가지 패턴을 생성하게 되고, 두 번째 단계는 스크램블을 위한 대상을 불러오는 것이고, 다음 단계는 AES 알고리즘에 의해서 만들어진 16개의 행렬과 16가지 패턴을 이용한 스크램블하게 된다.

그림 31. 비디오 암호화를 위한 프로그램
Figure 31. A program for video encryption

그림 32는 제안한 방법에 대한 실험 결과를 나타내었다. 그림에서 (a)는 원본 영상이고, (b)는 제안한 방법에 의하여 암호화된 결과 영상이다. 그림에서 위에서 아래로의 순서로 각각 'saving private ryan', 'the matrix 3', 'forrest gump', 'star wars 3', 'the load of the ring 3'에 대한 원본 영상과 결과영상을 나타내었다. 그림에서와 같이 제안한 방법을 사용한 결과 원본 영상과 제안한 영상 사이의 시각적으로 구별이 확연이 되고, 제안한 방법에 따라 암호화된 결과 영상만을 가지고 원본영상의 형태를 알아보기가 힘들다는 것을 알 수 있다.

(a) 원 비디오 (b) 암호화된 비디오

그림 32. 비디오 암호화를 결과

Figure 32. Results of video encryption

1.3 하드웨어 워터마크 삽입 시스템 구현 결과

본 논문에서는 실시간으로 워터마크를 삽입하기 위한 시스템을 두 가지로 구현하였고 각각을 그림 33과 36에 나타내었다.

그림 33. SD급 워터마크 삽입 시스템
Figure 33. A system of watermark embedding for SD format

그림 33은 SD급을 위한 워터마크 삽입 시스템의 구성을 나타내었고, 그림에서 왼쪽의 LCD 모니터가 워터마크가 삽입된 비디오이고, 오른쪽의 LCD는 워터마크가 삽입되지 않은 원본 영상을 나타내고 있다. 왼쪽 LCD 아래의 장비는 워터마크 삽입을 위한 FPGA가 내장되어 있고, 오른쪽 아래의 장비는 비디오 소스로 사용한 DVD 재생기이다. 본 시스템에서의 실험을 위하여 다섯 가지 DVD 타이틀을 가지고 진행하였다. DVD 타이틀은 각각 SD(720*480) 크기를 가진다. 실험 결과를 평가하기 위하여 시각적인 검증 및 강인성 검증을 진행하였다. 그림 34는 실험 DVD

콘텐츠 중 ‘saving private ryan’에 대한 원본 영상과 워터마크가 그리고 H.264로 압축 후 복원한 영상의 10번째, 30번째, 50번째 영상을 나타내었다. 그림에서와 같이 시각적으로 영상의 화질 차이는 거의 없음을 알 수 있다.

(a) 원 비디오

(b) 워터마크 삽입된 비디오

(c) H.264압축 후 복원 비디오

그림 34. 워터마크 삽입 결과

Figure 34. Results of watermark embedding

시각 테스트의 결과를 표 7에 나타내었다. 주관적 화질 비교 실험을 위하여 피실험인이 두 대의 모니터를 번갈아 보면서 모니터에 재생되는 화면의 이상 유무를 살펴보게 하였다. 실험 결과 두 모니터에 재생되는 화면의 시간차이를 전혀 느낄 수 없는 실시간 워터마킹이 구현되었음을 확인 할 수 있었고, 모니터에서 30cm, 70cm, 1m, 2m의 거리를 두고 워

터마크 삽입 정도에 대한 의견에 대한 결과는 70cm, 1m, 2m정도의 거리에서는 시각적으로 인지 할 수 없는 정도였고, 30cm에서도 시각적으로 거의 인지할 수 없을 정도였다.

표 6. 워터마크 삽입 실험 결과표

Table 6. Results of watermark inserting test

Distance / Video sequence	30(cm)	70(cm)	100(cm)	200(cm)
The lord of the rings 3	2	1	1	1
Forrest gump	2	1	1	1
Star wars 3	2	1	1	1
Saving private ryan	2	1	1	1
The Matrix 3	2	1	1	1

1. 인지 안 됨, 2. 거의 인지 안 됨, 3. 인지됨

본 논문에서의 실시간 구현은 비디오 프로세싱이 비디오 재생 시간에 거의 영향을 주지 않는 범위를 말한다. 즉, 초당 30프레임을 재생할 경우 비디오 프로세싱이 적용되더라도 초당 30프레임의 재생이 가능하여야 함을 말한다. 표 8은 소프트웨어로 워터마크 삽입 프로그램을 구현하여 100 프레임의 크기를 가지는 샘플 비디오에 대하여 테스트 한 결과이다. 테스트 환경은 펜티엄 듀어 코어 2.6GHz, Memory 4GBytes이다. 실험 결과 100프레임을 재생하는데 소요되는 시간은 약 3.3초 이지만, 구현된 프로그램은 워터마크를 삽입하는데 평균 9.24초가 소요됨을 알 수 있고, 이는 실시간 처리가 되지 않고 있음을 보여주고 있다. 하드웨어 구현에서는 지역 분산을 구하기 위한 라인 버퍼에 지연 시간만이 존재하고, 소수 연산에 필요한 비트를 제거하였기 때문에 곱셈기, 덧셈기의 수가 절

반으로 줄어 상대적인 계산 감소가 있었고, 비디오 재생 중 워터마크를 실시간으로 삽입할 수 있었다.

표 7. 워터마크 삽입 시간 결과

Table 7. Results of watermark inserting time

Saving private ryan	Forrest gump	The lord of the rings 3	The Matrix 3	Star wars 3
8.351	9.502	9.462	9.296	9.593

원본 비디오에 삽입되어 있는 워터마크를 검출하기 위하여 본 논문에서 제안한 방법을 PC에 구현하였다. 검출 결과는 그림 35와 같다. 본 논문에서 제안한 방법과 구현에 의하여 삽입된 워터마크는 워터마크 검출 프로그램을 통하여 정확하게 검출 된 것을 볼 수 있다.

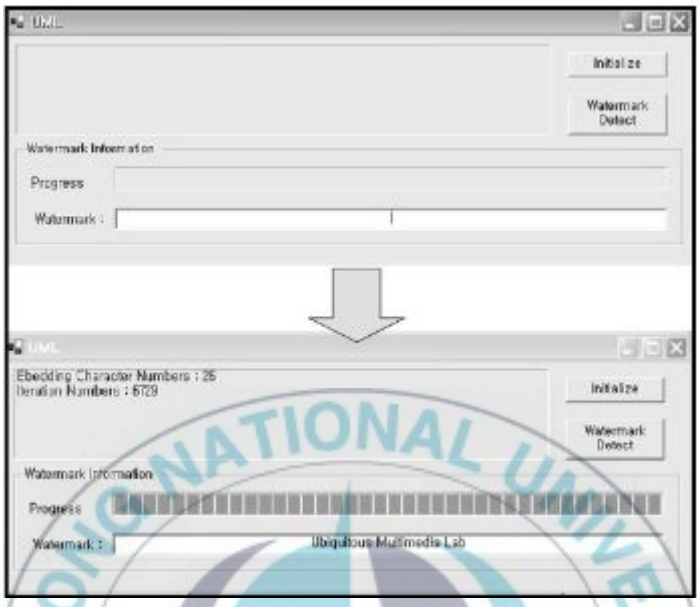


그림 35. 워터마크 검출 결과

Figure 35. A result of watermark detection

표 8은 워터마크 삽입 및 검출에 따른 객관적인 화질 및 압축에 대한 강인성을 표로 나타내었다. 실험결과 720x480 크기를 가지는 비디오에서 워터마크 삽입하였을 경우 평균 44.29dB의 PSNR을 가지고 이는 시각적으로 워터마크 삽입한 경우와 그렇지 않은 경우 사이의 화질 차이를 구분하기 힘든 정도이고, 현재 압축효율이 가장 좋은 H.264로 압축하였을 경우 평균 92.5%의 검출률을 보여 압축에 강인한 함을 보여준다.

표 8. 워터마크 강인성 실험 결과표

Table 8. Results of watermark robustness test

	PSNR(dB)		검출률(%)		압축률(%)
	비압축	H.264	비압축	H.264	
Saving private ryan	46.93	42.64	100	87.5	450
Forrest gump	42.63	35.19	100	100	370
The lord of the rings 3	46.76	42.52	100	87.5	556
The Matrix 3	41.52	37.70	100	100	376
Star wars 3	43.62	40.70	100	87.5	322

그림 36은 HD급에 대한 실시간 워터마킹을 위한 시스템을 나타내었다.

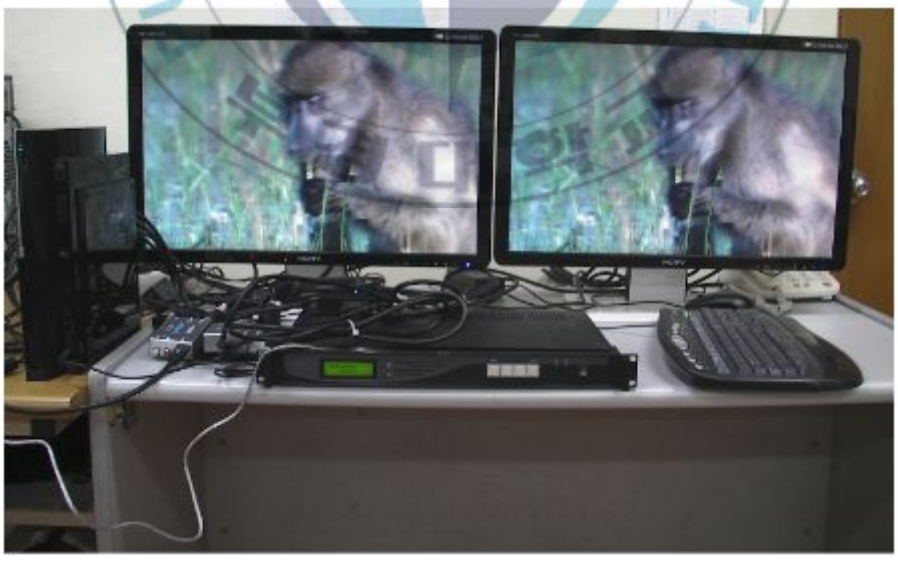
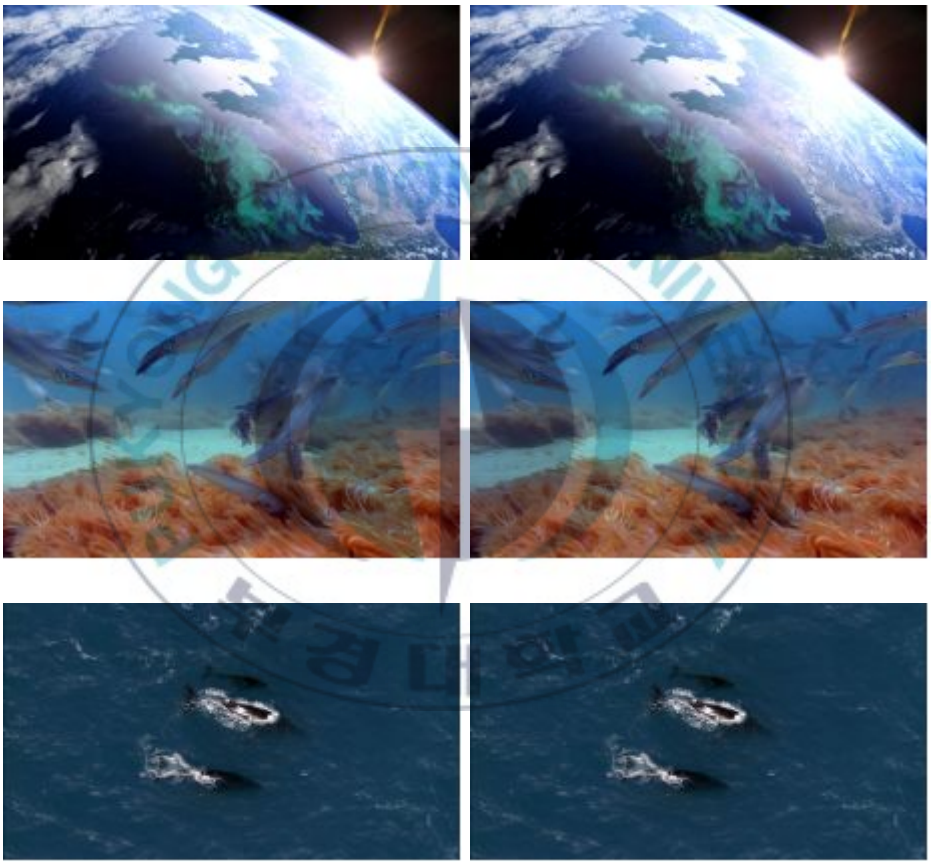


그림 36. HD급 실시간 워터마킹 시스템

Figure 36. A real-time watermarking system with HD format

위의 그림에서 HD 신호의 발생을 위하여 Sony사의 Playstation 3가 왼쪽 LCD 옆에 있고, 워터마크가 삽입된 영상이 왼쪽 LCD이고, 워터마크가 삽입되지 않은 원 영상을 오른쪽 LCD에 나타내었다. 그림 37은 HD원본 영상과 워터마크가 삽입된 영상을 각각 나타내었다.



(a) 원 비디오 (b) 워터마크 삽입된 비디오

그림 37. 워터마크 삽입 결과

Figure 37. Results of watermark embedding

위의 그림에서와 같이 워터마크가 삽입된 영상과 워터마크가 삽입되지

않은 영상에서의 시각적인 화질 차이는 거의 없음을 알 수 있다. 실험을 통하여 제안한 워터마크 삽입 알고리즘은 HD(1920*1080) 30Hz 비디오에 대하여 실시간 삽입 되는 것을 화면으로 확인 할 수 있었다. 소프트웨어적인 방법으로는 SD급 720*480크기의 비디오를 실시간으로 처리하기 어려웠지만, 제안한 알고리즘을 바탕으로 구현된 하드웨어 기반 워터마킹은 SD급의 네 배 크기인 HD에서도 프레임 손실 없이 실시간으로 처리됨을 보였다.

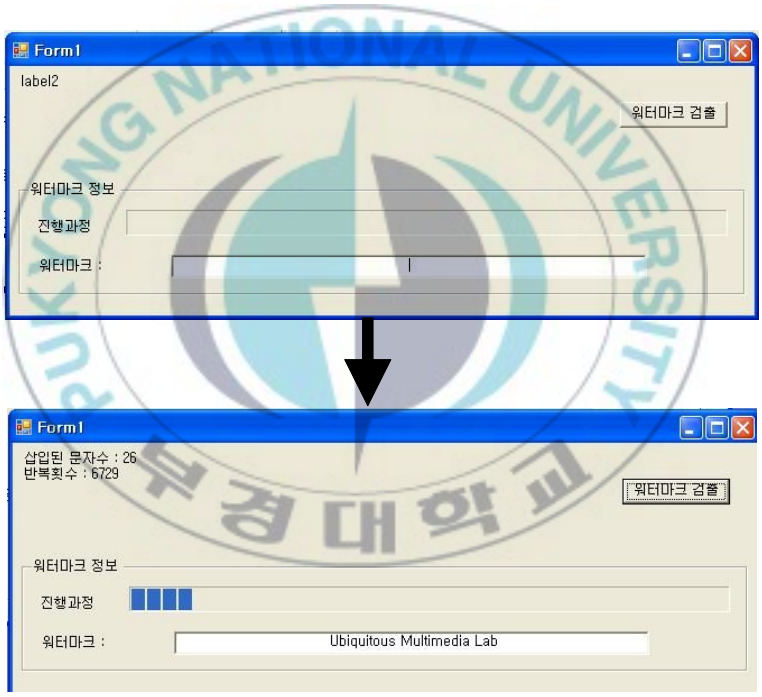


그림 38. 워터마크 검출 결과

Figure 38. A result of watermark detection

그림 38은 HD 비디오에서 워터마크를 삽입하고 이를 Blackmagic사의 Declink HD extreme을 통하여 캡처한 비디오에 대하여 워터마크 검출을 시도하였을 때 검출됨을 보여주고 있다.

Ⅴ. 결론

본 논문에서는 방송용 콘텐츠의 저작권 보호를 위한 실시간 비디오 워터마킹 및 암호화 알고리즘을 하드웨어 구현에 적합하도록 개선하고 이를 구현하였다. 제안한 실시간 워터마킹 시스템은 실시간 처리를 위하여 공간 영역에 워터마크를 삽입하였고, 강인성을 높이기 위하여 대역확산 기법을 사용하여 프레임 내에서 워터마크를 반복적으로 삽입하고, 또한 프레임과 프레임 사이에 같은 워터마크를 삽입하는 방법을 사용하였다. 또한 워터마크 강도를 HSV에 기반한 6단계 강도조정을 하도록 하여 비가시성과 강인성이 뛰어나도록 설계하였다. 암호화 방법에서는 스크램블링을 이용하여 화소의 위치정보를 바꾸는 기법을 사용하였고 스크램블링 기법의 약점인 암호화 안전성을 높이기 위하여 스크램블링을 위한 패턴 선택에 대하여 AES 암호화 방식을 적용하였다. FPGA 기반의 하드웨어 구현에 적합한 알고리즘을 제안하기 위하여 본 논문에서는 두 가지의 방법으로 접근하였다. 첫 번째는 하드웨어 구현에서의 계산 복잡도를 줄여 실시간으로 고화질의 영상을 처리할 수 있는 방법이고 두 번째는 FPGA에 내부 블록을 고려하여 워터마킹 및 암호화 알고리즘을 제안하는 것이다. 전자를 위하여 워터마킹에서는 워터마크 강도를 소수가 아닌 정수형태의 값을 사용하였고, 계산량이 많은 변환 영역이 아닌 계산 복잡도가 작은 공간영역에서의 워터마킹 기술을 적용하였다. 또한 은닉하려고 하는 정보를 워터마크 정보로 바꾸기 위하여 단순한 사상법을 사용하여

복잡한 곱셈 및 나눗셈 연산을 없애어 속도의 향상을 가지는 알고리즘을 제안하였다. 암호화의 경우 화소 단위의 암호화를 배제하고 공간상의 위치 정로를 변경하는 기법을 사용하였다. 후자의 경우 워터마킹에서는 FPGA 내부의 제한된 크기의 고속 쉬프트 레지스터를 최대한 활용 할 수 있도록 비디오 프레임에서 최소량의 값만을 받을 수 있도록 알고리즘을 고안하였다. 암호화의 경우 FPGA 내부에 제한되어 있는 고속의 SRAM을 최대한 사용하기 위하여 암호화 패턴 테이블을 최소화 하였다. 이렇게 최소화 되는 패턴의 안정성을 위하여 패턴의 선택은 AES 암호화 알고리즘을 통하여 나온 결과를 사용하였다. 제안한 워터마킹 및 암호화와의 하드웨어 구현을 위하여 사용된 칩은 ALTERA사의 FPGA인 STRATIX를 사용하였고, 워터마크 검출을 위하여 STRATIX FPGA와 PCI제어기 그리고 소프트웨어 프로그램을 사용하여 구현하였다. 제안한 방법에 대한 검증을 위하여 소프트웨어적인 방법과 하드웨어 적인 방법으로 구현하여 실험하였다. 소프트웨어적인 방법으로 알고리즘을 검증한 결과 워터마킹의 경우 워터마크가 삽입된 비디오를 H.264와 같은 고압축 비디오 코덱을 사용하여 부호화/복호화 후 워터마크를 검출한 결과 SD급 영상에서 압축률 400%의 고압축에서도 워터마크 검출률이 87.5%로 높게 나옴을 확인 하였다. 또한 암호화의 경우 SD급의 화질에서 화면의 내용을 알아 볼 수 없을 정도의 시각적 인지도를 보여주었다. 하드웨어 검증을 위하여 본 논문에서는 워터마킹 및 암호화 통합 시스템을 구현하였다. 통합시스템의 구성은 SD 및 HD 비디오를 발생 시킬 수 있는 비디오 소스 장치, 비디오 소스의 신호를 FPGA에서 처리할 수 있는 SDI 신호로 변환하는 장치, SDI신호를 FPGA에 맞도록 병렬 신호로 변환 가능한 칩이 내장된 FPGA 시스템 그리고 FPGA를 통하여 비디오 처리 결과를 직접 눈으로 볼 수 있고, 비디오 처리 결과와 원본 비디오 신호를

동시에 볼 수 있도록 두 대의 모니터를 사용하여 시스템을 구성하였다. 이렇게 구성한 시스템을 통하여 알고리즘을 검증한 결과 프레임의 끊김 없이 워터마크가 삽입됨을 확인할 수 있었고, 워터마크가 삽입된 비디오 콘텐츠와 워터마크가 삽입되어 있지 않은 원본 비디오 콘텐츠간의 화질 차이가 거의 없음을 확인 할 수 있었다. 워터마크 검출 결과 삽입된 워터마크는 전부 검출됨을 확인할 수 있었다. 암호화의 경우 AES 암호화 알고리즘의 VHDL 구현을 통하여 암호화가 적용됨을 확인 하였고, 암호화 전체 시스템 또한 VHDL 모의실험을 통하여 그 결과 스크램블링이 정확히 적용됨을 확인하였다. 그리고 제안한 시스템은 방송 또는 영화산업과 같은 콘텐츠 보호를 필요로 하는 곳에 적용 가능하다[50,69].

참 고 문 헌

[1] I. Cox, M. Miller, and J. Bloom, *Digital Watermarking,* Press of Morgan Faukmann, 2002.

[2] S. Sun, P. Qiu, Y. Wang, and L. Yao, "Blind Watermarking Algorithm based on General Gaussian Model," *in Proc. IEEE International Conference on Signal Processing*, Vol. 3, pp. 2302-2305, Aug. 2004.

[3] M. Marcinak and B. Mobasseri, "Digital Video Watermarking for Metadata Embedding in Uav Video," *IEEE Military Communications Conference*, pp. 1-5, Oct. 2005.

[4] Y. Wang and A. Pearmain, "Blind MPEG-2 Video Watermarking Robust Against Geometric Attacks: A set of Approaches in DCT Domain," *IEEE Transactions on Image Processing,* Vol. 15, No. 6, pp. 1536-1543, Jun. 2006.

[5] K. Su, D. Kundur, and D. Hatzinakos, "Spatially Localized Image-dependent Watermarking for Statistical Invisibility and Collusion Resistance," *IEEE Transactions on Multimedia*, Vol. 7, No. 1, pp. 52-66, Feb. 2005.

[6] L. Strycker, P. Termont, J. Vandewege, J. Haitsma, A. Kalker, M. Maes, and G. Depovere, "Implementation of a Real-Time Digital Watermarking Process for Broadcast Monitoring on Trimedia VLIW Processor," *IEE Proceedings on Vision, Image and Signal Processing,* Vol. 147, No. 4, pp. 371-376, Aug. 2000.

[7] N. Mathai, A. Sheikholeslami, and D. Kundur, "VLSI Implementation of a Real-Time Video Watermark Embedder and Detector," *in Proc. IEEE International Symposium on Circuits and Systems*, Vol. 2, pp. II-772 - II-775, May 2003.

[8] S. Mohanty, N. Ranganathan, and R. Namballa, "VLSI Implementation of Invisible Digital Watermarking Algorithms towards the Development of a Secure JPEG Encoder," *IEEE Workshop on Signal Processing Systems*, pp. 183-188, Aug. 2003.

[9] A. Garimella, M. Satyanarayana, P. Murugesh, and U. Niranjan, "ASIC for Digital Color Image Watermarking," *IEEE Signal Processing Education Workshop*, pp. 292-296, Aug. 2004.

[10] A. Kudelski. *Method for Scrambling and Unscrambling a Video Signal.* United States Patent 5375168, Dec. 1994.

[11] J. Liu, L. Zou, C. Xie, and H. Huang, "A two-way Selective Encryption Algorithm for MPEG Video,"

International Workshop on Networking, Architecture, and Storages, pp. 5, Aug. 2006.

[12] S. Maniccam and G. Nikolaos, "Image and Video Encryption using SCAN Patterns," *Pattern Recognition*, Vol. 37, No. 4, pp. 725-737, 2004.

[13] X. Yi, C. Tan, C. Siew, and S. Rahman, "Fast Encryption for Multimedia," *IEEE Transactions on Consumer Electronics,* Vol. 47, No. 1, pp. 101-107, Feb. 2001.

[14] L. Qao and K. Nahrstedt, "New Algorithm for MPEG Video Encryption," *International Conference on Imaging Science, Systems and Technology (CISST'7),* pp. 21-29, Jul. 1997.

[15] A. Romeo, G. Romdotti, M. Mattavelli, and D. Mlynek, "Ryptosystem Architectures for very high throughput Multimedia Encryption: the RPK Solution," *International Conference on Electronics, Circuits and Systems (ICECS),* pp. 261-264, 1999.

[16] C. Shi, S. Wang, and B. Bhargava, "PEG Video Encryption in Real-time using Secret Key Cryptography," *In Proc. of PDPTA'9,(Las Vegas,Nevada),* pp. 2822-2828, 1999.

[17] X. Niu, S. Sun, and W. Xiang, "Multiresolution Watermarking for Video based on Gray-level Digital Watermark," *IEEE Trans. Consumer Electronics*, Vol. 46, No. 2, pp. 375-384, May 2000.

[18] J. Zhang, J. Li, and L. Zhang, “Video Watermark Technique in Motion Vector,” *XIV Brazilian symposium on Computer Graphics and Image Processing,* pp. 179-182, Oct. 2001.

[19] M. Kutter and S. Winkler, “A Vision-Based Masking Model for Spread-Spectrum Image Watermarking,” *IEEE Trans. Image Processing,* Vol. 11, pp. 16-25, Jan. 2002.

[20] F. Hartung and B. Girod, “Digital Watermarking of Uncompressed and Compressed Video,” *Signal Processing (Special Issue on Copyright Protection and Access Control for Multimedia Service)*, Vol. 66, No. 3, pp. 283-301, 1998.

[21] M. Barni, F. Bartolini, V. Cappellini, and N. Checcacci, “Object Watermarking for MPEG4 Video Streams Copyright Protection,” *Proc. of the SPIE In Security and Multimedia Contents II*, Vol. 3971, pp. 465-476, 2000.

[22] R. Dugad and N. Ahuja, “A Scheme for Joint Watermarking and Compression of Video”, *Proc. of IEEE ICIP,* pp. 80-83, 2000.

[23] A.J. Menezes, P.C. ven Oorschot, and S.A. Vanstone, *Handbook of Applied Cryptography,* CRC Press, ISBN 0-8493-8523-7, http://www.cacr.math.uwaterloo.ca/hac/, 1996.

[24] National Bureau of Standards, *Data Encryption Standard*, FIPS-Pub 46, 1977.

[25] R.L. Rivest, A. Shamir, and L. Adleman, “A Method for Obtaining Digital Signatures and Public Key Cryptosystems,” *Communications of the ACM*, Vol. 21, No. 2, pp. 120-126, 1978.

[26] E. Biham and A. Shamir, “Differential Cryptanalysis of the Full 16-Round DES,” *LNCS 537*, pp. 2-21, 1990.

[27] M. Matsui, “Linear Cryptanalysis Method for DES,” *LNCS 765*, pp. 386-397, 1994.

[28] *Report on the Development of the Advanced Encryption Standard(AES)*, http://www.csrc.nist.gov/encryption/aes/.

[29] *New European Schemes for Signatures Integrity and Encryption(NESSIE)*, http://cryptonessie.org/.

[30] *Cryptography Research and Evaluation Committees(CRYPTREC)*, http://www.cryptrec.go.jp/.

[31] “128비트 블록 암호알고리즘(SEED) 개발 및 분석 보고서,” 한국정보보호진흥원, 10월, 2003.

[32] *ARIA*, http://www.nsri.re.kr/ARIA/.

[33] L. Qao, and K. Nahrstedt, “New Algorithm for MPEG Video Encryption,” *International Conference on Imaging Science, Systems and Technology (CISST'7)*, pp. 21-29, Jul. 1997.

[34] X. Yi, C. Tan, C. Siew, and S. Rahman, “Fast Encryption for Multimedia,” *IEEE Transactions on Consumer Electronics*, Vol. 47, No. 1, pp. 101-107, Feb. 2001.

[35] C. Shi, S. Wang, and B. Bhargava, “PEG Video Encryption in Real-time using Secret Key Cryptography,” *In Proc. of PDPTA'9,(Las Vegas,Nevada),* pp. 2822-2828, 1999.

[36] Y. Li, L. Liang, Z. Su, and J. Jiang, “A New Video Encryption Algorithm for H.264,” *International Conference on Information, Communications and Signal Processing,* pp. 1121-1124, Dec. 2005.

[37] Y. Wang, M. Cai, and F. Tang, "Design of a New Selective Video Encryption Scheme based on H.264," *International Conference on Computational Intelligence and Security*, pp. 883-882, Dec. 2007.

[38] S. Maniccam and G. Nikolaos, “Image and Video Encryption using SCAN Patterns,” *Pattern Recognition*, Vol. 37, No. 4, pp. 725-737, 2004.

[39] L. Strycker, P. Termont, J. Vandewege, J. Haitsma, A. Kalker, M. Maes, and G. Depovere, “Implementation of a Real-Time Digital Watermarking Process for Broadcast Monitoring on Trimedia VLIW Processor,” *IEE Proc. on Vision, Image and Signal Processing,* Vol. 147, No. 4, pp. 371 - 376, Aug. 2000.

[40] G. Petitjean, J. Dugelay, S. Gabriele, C. Rey, and J. Nicolai, “Towards Real-time Video Watermarking for Systems-On-Chip,” *Proc. IEEE International Conference on Multimedia and Expo*, Vol. 1, pp. 597-600, 2002.

[41] A. Garimella, M. Satyanarayan, R. Kumar, P. Murugesh, and U. Niranjan, "VLSI Impementation of Online Digital Watermarking Techniques with Difference Encoding for the 8-bit Gray Scale Images," *Proc. IEEE International Conference on VLSI Design.* pp. 283-288, 2003.

[42] T. Tsai and C. Wu, "An Implementation of Configurable Digital Watermarking Systems in MPEG Video Encoder," *Proc. IEEE International Conference on Consumer Electronics*. pp. 216-217, Jun. 2003.

[43] N. Mathai, A. Sheikholeslami, and D. Kundur, "VLSI Implementation of a Real-Time Video Watermark Embedder and Detector," *Proc. IEEE International Symposium on Circuits and Systems,* Vol. 2, pp. II-772 - II-775, May 2003.

[44] Y. Seo and D. Kim, "Real-Time Blind Watermarking Algorithm and its Hardware Implementation for Motion JPEG2000 Image Codec," *Proc. 1st Workshop on Embedded Systems for Real-Time Multimedia,* pp. 88-93 Oct. 2003.

[45] A. Garimella, M. Satyanarayana, P. Murugesh, and U. Niranjan, "ASIC for digital color image watermarking," *IEEE Signal Processing Education Workshop,* pp. 292-296 Aug. 2004.

[46] S. Mohanty, N. Ranganathan, and K. Balakrishnan, "A

Dual Voltage-Frequency VLSI Chip for Image Watermarking in DCT Domain," *IEEE Transactions on Circuits and Systems-II,* Vol. 53, pp. 394-398, May, 2006.

[47] S. Mohanty, N. Ranganathan, and R. Namball, "A VLSI Architecture for Visible Watermarking in a Secure Still Digital Camera (S2DC) Design," *IEEE Transactions on VLSI Systems,* Vol. 13, No. 8, pp. 1002-1012, Aug. 2005.

[48] *Component Video Signal 4:2:2 Bit-Parallel Digital Interface,* American National Standard, ANSI/SMPTE 125M, Sep. 1995.

[49] P. Shlichta, "Higher-dimensional Hadamard Matrices," *IEEE Trans. on Information Theory,* 1979.

[50] Y. J. Jeong, K. S. Moon, and J. N. Kim, "FPGA based Implementation of Real-Time Video Watermarking Chip," *LNCS (Lecture Note in Computer Science) 4523,* pp. 133-141, May, 2007.

[51] Y. J Jeong, K. Y. Ryu, T. I. Jeong, K. S. Moon, and J. N. Kim, "FPGA Implementation of Video Watermark Embedding System," *CCIS (Communications in Computer and Information Science) 2,* pp. 868-877, Aug. 2007.

[52] 정용재, 문광석, 김종남, "워터마킹과 암호화를 이용한 DMB 콘텐츠의 재생 및 복사 제어를 PMP에 구현," 대한전자공학회 논문지, 제 46권, CI편, 제2호, pp. 52-57, 2009년 3월.

[53] 정용재, 박성모, 문광석, 김종남, "디지털 방송 콘텐츠 저작권

보호를 위한 실시간 워터마크 삽입 시스템 구현," 신호처리시스템학회논문지, 제10권, 제2호, pp. 100-105, 2009년 4월.

[54] 정용재, 김종남, 문광석, "디지털 방송에서 콘텐츠의 저작권 보호를 위한 실시간 워터마킹 하드웨어 시스템 구현," 한국콘텐츠학회 논문지, 제9권, 제9호, pp. 51-59, 2009년 8월.

[55] Y. J Jeong, W. H. Kim, K. S. Moon, and J. N. Kim, "Implementation of Watermark Detection System for Hardware Based Video Watermark Embedder," *International Conference on Convergence and hybrid Information Technology 2008,* Vol. 2, pp. 450-453, Nov. 2008.

[56] Y. J. Jeong, K. S. Moon, and J. N. Kim, "Implementation of Real Time Video Watermark Embedder Based on Haar Wavelet Transform," *Future Generation Communication and Networking(FGCN) 2008,* Vol. 1, pp. 63-66, Dec. 2008.

[57] Y. J. Jeong, K. S. Moon, and J. N. Kim, "Implementation of Real Time DMB Encryption on Mobile Environment for Contents Protection," *Proc. International conference on advanced communication technology* , pp. 2191-2194, Feb. 2009.

[58] Y. J. Jeong, K. S. Moon, and J. N. Kim, "Implementation of Play Control for Illegal Distribution Blocking T-DMB Contents on Portable Multimedia Player," *Proc. International Symposium on Advanced Engineering* , pp.

55-58, Mar. 2009.

[59] 정용재, 강경원, 문광석, 김종남, “개선된 대역 확산 기법을 이용한 실시간 비디오 워터마크 삽입기의 설계 및 구현,” 한국 멀티미디어학회 춘계학술발표대회, 제9권, 제1호, pp. 127-131, 2006년 5월.

[60] 정용재, 유태경, 김원희, 김종남, 문광석, “저작권 보호를 위한 실시간 비디오 워터마킹 시스템 구현,” 한국신호처리시스템학회 추계학술발표대회논문집, pp. 246-249, 2007년 11월.

[61] 정용재, 문광석, 김종남, “디지털 웨이브릿 변환 기반 실시간 비디오 워터마킹 FPGA구현,” 한국멀티미디어학회 춘계학술발표대회논문집, 제11권, 제1호, pp. 217-220, 2008년 5월.

[62] 정용재, 문광석, 김종남, “Haar 웨이블릿 기반 실시간 워터마크 삽입기 FPGA 구현,” 한국신호처리시스템학회 춘계학술발표대회논문집, 제9권, 제1호, pp. 107-110, 2008년 1월.

[63] 정용재, 정태일, 김종남, 문광석, “다이렉트쇼 환경 기반에서 고압축과 저작권 보호를 위한 비디오 트랜스 코딩과 워터마킹 구현,” 제30회 한국정보처리학회 추계학술발표대회 논문집, 제15권, 제2호, pp. 1500-1503, 2008년 11월.

[64] 정용재, 정태일, 문광석, 김종남, “모바일 환경에서 콘텐츠 보호를 위한 실시간 DMB 암호화 구현,” 한국멀티미디어학회 추계학술발표대회 논문집, 제11권, 제2호, pp. 460-463, 2008년 11월.

[65] 이성연, 정용재, 문광석, 김종남, “저작권 보호를 위한 워터마킹 시스템과 H.264 비트스트림의 선택적 암호화 구현,” 한국

멀티미디어학회 추계학술발표대회 논문집, 제11권, 제2호, pp 709-712, 2008년 11월.

[66] 정용재, 정태일, 문광석, 김종남, “DMB 콘텐츠 보호를 위한 실시간 암호화 및 재생제어를 PMP에 구현,” 한국신호처리시스템학회 추계학술발표대회 논문집, 제9권, 제2호, pp. 421-424, 2008년 11월.

[67] 정용재, 문광석, 김종남, “워터마킹 및 암호화 기법을 사용한 실시간 DMB 콘텐츠 보호를 PMP에 구현,” 대한전자공학회 학술심포지움 논문집, pp. 131-132, 2008년 12월.

[68] 정용재, 문광석, 김종남, “포터블 멀티미디어 재생기에서 실시간 DMB 콘텐츠 암호화 및 재생 제어 구현,” 영상처리 및 이해에 관한 워크샵, pp. 640-644, 2009년 2월.

[69] 정용재, 문광석, 김종남, “H.264 비디오 압축에서 GOP내의 프레임특성을 이용한 강인한 워터마킹,” 한국콘텐츠학회 춘계종합학술대회논문집 pp. 861-864, 2009년 5월.

감사의 글

이 논문이 완성되기까지 정성어린 지도와 편달을 아끼지 않으시고 용기를 주신 문광석 교수님께 진심으로 감사드립니다. 박사 학위 과정 동안 많은 지도와 가야 할 길을 제시해 주시고, 항상 웃음으로 저를 이끌어주신 김종남 교수님께 깊은 감사를 드리며, 여러 가지 부족한 본 논문의 심사를 맡아 세심한 조언을 해주신 권기룡 교수님, 류권열 교수님, 정완영 교수님 그리고 학위과정동안 많은 가르침을 주셨던 학과의 여러 교수님들께도 감사드립니다.

박사 과정 동안 같은 연구실에서 실험실 생활에 물심양면으로 도움을 주신 류권열 교수님, 허영대 교수님, 정태일 박사님께 감사드립니다. 지난 기간 동안 같은 연구실에서 동고동락하며 저에게 많은 조언과 진심어린 관심을 보여주신 경원 형에게 감사드리고, 대학원 생활에서 항상 옆에 있어준 든든한 친구 태경, 대학원 초기에 매일 같이 밤을 보낸 진호, 몇 달을 제안서에 매달리면서 힘들어하는 기색 없이 항상 옆에서 부지런하게 연구하는 원희, 연구실 생활을 너무 힘들어하는 영웅, 항상 밝은 웃음으로 잘 따라 준 성연, 필요할 때마다 항상 저를 도와준 재환, 대학원 생활 초기에 같이 생활한 두경, 형택 그리고 논문을 준비하면서 도와준 나머지 연구실 후배들에게도 감사드립니다. 학교생활에서 서로 도움주고 도움 받는 사이에서 이제는 같이 회사에 근무하게 된 윤귀씨, 항상 연구실에 가면 박사 커피라면서 커피 한잔을 주던 택영, 가끔 이런저런 이야기하고 같이 졸업하게 된 수완씨, 술 그만 마시라는 충고를 해주고 싶은 봉주에게도 고마움을 전합니다.

저에게 아날로그 IC 설계를 시작하게 해주시고, 여러 가지 부족한 본

논문의 완성을 위해 많은 관심을 가져주신 김기호 사장님, 많은 일에도 항상 웃음을 잃지 않는 김재희 전무님, 늘 격려해 주시는 최원재 기술위원님 그리고, 모든 에임즈 식구들에게도 감사드립니다.

늘 부족한 아들 뒷바라지를 해주신 어머니와 늘 자식 걱정이 앞서는 아버지께 오늘의 영광을 돌리고 싶습니다. 그리고 동생 용범에게도 감사의 말을 전합니다.