VLSI ARCHITECTURE AND FPGA PROTOTYPING OF A SECURE DIGITAL CAMERA FOR BIOMETRIC APPLICATION

Oluwayomi Bamidele Adamo

Thesis Prepared for the Degree of

MASTER OF SCIENCE

UNIVERSITY OF NORTH TEXAS

August 2006

APPROVED:

Saraju P. Mohanty, Major Professor
Murali Varanasi, Co-Major Professor
Elias Kougianos, Minor Professor
Armin R Mikler, Graduate Coordinator
Krishna Kavi, Chair of the Department of
Computer Science and Engineering
Oscar Garcia, Dean of the College of Engineering
Sandra L. Terrell, Dean of the Robert B. Toulouse
School of Graduate Studies

Adamo, Oluwayomi Bamidele, VLSI Architecture and FPGA Prototyping of a Secure Digital Camera for Biometric Application. Master of Science (Computer Engineering), August 2006, 54 pp., 4 tables, 46 illustrations, references, 52 titles.

This thesis presents a secure digital camera (SDC) that inserts biometric data into images found in forms of identification such as the newly proposed electronic passport. However, putting biometric data in passports makes the data vulnerable for theft, causing privacy related issues. An effective solution to combating unauthorized access such as skimming (obtaining data from the passport's owner who did not willingly submit the data) or eavesdropping (intercepting information as it moves from the chip to the reader) could be judicious use of watermarking and encryption at the source end of the biometric process in hardware like digital camera or scanners etc. To address such issues, a novel approach and its architecture in the framework of a digital camera, conceptualized as an SDC is presented. The SDC inserts biometric data into passport image with the aid of watermarking and encryption processes. The VLSI (very large scale integration) architecture of the functional units of the SDC such as watermarking and encryption unit is presented. The result of the hardware implementation of Rijndael advanced encryption standard (AES) and a discrete cosine transform (DCT) based visible and invisible watermarking algorithm is presented. The prototype chip can carry out simultaneous encryption and watermarking, which to our knowledge is the first of its kind. The encryption unit has a throughput of 500 Mbit/s and the visible and invisible watermarking unit has a max frequency of 96.31 MHz and 256 MHz respectively.

ACKNOWLEDGEMENTS

I would like to extend my gratitude to my major professor Dr. Saraju P. Mohanty for his guidance and support throughout my studies. His wealth of knowledge contributed immensely to my research work. I would like to express my appreciation to my co-major professor, Dr. Murali Varanasi for his advice and suggestions. I am also grateful to him for giving me a chance and supporting me financially throughout my studies. I also would like to thank Dr. Elias Kougianos for his advice and spending time to guide me.

I would like to thank my wife, Monica E. Adamo, and my son, Oluwayomi Wesley Bamidele Adamo for their encouragement and patience throughout my research work. I could not have made it without them. They give me a reason to work very hard. Finally, I would like to thank my colleagues in Computer Science and Engineering and Electrical Engineering who encouraged me throughout my research work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Page

CHAPTER 1

INTRODUCTION

1.1. Motivation and Background

In order to improve document and border security, the US Department of State has been testing a pilot program of the first US electronic passport. The electronic passport contains an integrated chip embedded in the cover which stores the same data usually contained in the paper passport. They propose to include biometric data such as fingerprint, iris scans, signatures, etc., in the electronic passport. The key objectives of the initiative are to identify the owner, authenticate the document, and copyright the passport [3]. However, there are continuous risks of unauthorized access and modification to the data contained in the passport. To prevent unlawful access, we believe that there is an urgent need for development of tools and frameworks [2]. An effective solution to combating unauthorized access such as "skimming"; the act of obtaining data from the passport's owner who did not willingly submit the data, or "eavesdropping"; intercepting information as it moves from the chip to the reader, could be judicious use of watermarking and encryption at the source end of the biometric process in hardware like digital camera or scanners, etc while watermarking is the process whereby a host image is embedded with data for the purpose of copyright protection [44], encryption is the transformation of data into secret code with the purpose of protecting the secrecy of the data when sent through an insecure channel.

In the past, several attempts have been made to develop the different units of a digital camera with watermarking capabilities, but few have dealt with the design of the entire digital camera. Only a few of these designs have also incorporated cryptography in the camera design. As a result, we present a design and the architecture of a digital camera system that incorporates watermarking and encryption.

1.2. Digital Watermarking Overview

Digital watermarking is the process whereby a digital signal (watermark) is embedded into a digital data (host data) which could be audio, image, video, and text. The digital signal could be later detected or extracted in order to make assertion about the data [39], [34]. The watermark is hidden in the host data in such a manner that it is resistant to many operations and does not degrade the host image [38]. A watermarking process is usually made up of the watermark, host data, encoder, and decoder as shown in Fig. 1.1.

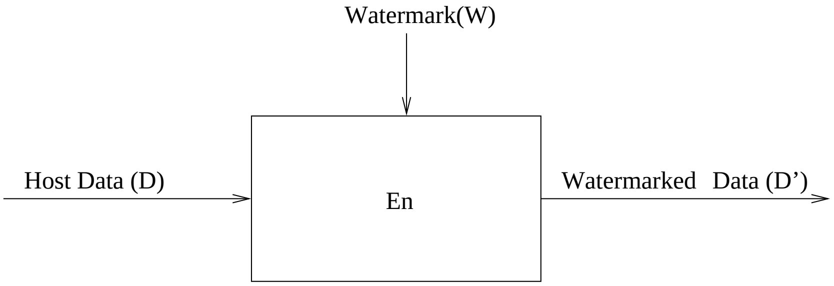

Figure 1.1. Watermark Encoder

If the host data is denoted as D and the watermark as W, then the watermarked data W' is obtained when the encoder function En is applied to the host data D as shown in Eqn.(1). A block diagram illustrating the decoding process is shown in Fig. 1.2.

$$\text{W'}= En(\text{D,W}) \tag{1}$$

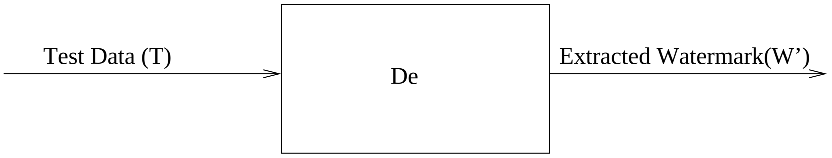

Figure 1.2. Watermark Decoder

The watermark could be extracted if present by applying a watermark decoder function De on a Test data T which could be watermarked or un-watermarked or corrupted data as shown in Eqn.(2).

(2) $$\text{W}= De(\text{T,D})$$

The extracted watermark could be further tested for its originality by using a comparator function as shown in Eqn.(3). if the there is a match between the extracted watermark W' and the watermark W based on a threshold value h, the output is 1 otherwise it is 0.

(3) $$\text{C(W, W')}= \begin{cases} 1, c >= h \\ 0, otherwise \end{cases}$$

Broadly speaking, digital watermarks can be divided into two categories: visible watermark, and invisible watermark [39]. The visible watermark is perceptually visible on careful inspection after the embedding process. For invisible watermark, the modification made to the pixel as a result of the embedding process is not perceptually visible and the watermark can only be recovered by using special image processing techniques. Watermarking techniques can be performed in either spatial or frequency domain [39]. The frequency domain based watermarking schemes have been known to be more robust than spatial domain based watermarking schemes [26].

1.3. Encryption Overview

The art and science of securing data is called cryptography. Cryptography has two distinct steps, encryption and decryption as shown in in Fig. 1.4. Encryption is the transformation of data into secret form with the purpose of protecting the secrecy of the data when sent through an insecure channel. The data is called plaintext and the encrypted data is called ciphertext [49]. The process of changing the ciphertext back into plain text is called decryption. The relationship between encryption and decryption is shown in Fig. 1.4. The art and science of securing data is called cryptography.

If P represents the plaintext which could be text or image data, E is the encryption operator, and C represents the ciphertext, then C could be expressed as shown in Eqn.(4):

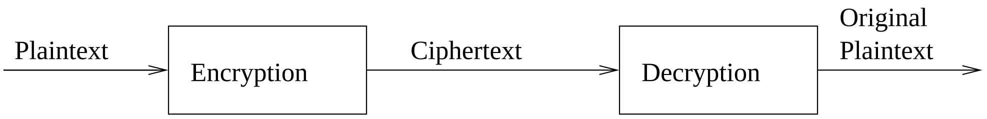

FIGURE 1.3. Encryption and Decryption

$$C = E(P) \tag{4}$$

If D denotes the decryption operator, then P could be represented as shown in Eqn.(5):

$$P = D(C) \tag{5}$$

Based on the description of encryption and decryption, Eqn.(6) is true.

$$P = D(E(P)) \tag{6}$$

Modern day encryption technique incorporates the use of keys, thereby changing the previous equations in Eqn.(4), Eqn.(5), Eqn.(6) into forms shown in Eqn.(7).

$$\begin{aligned} C &= E_k(P) \\ P &= D_k(C) \\ P &= D_k(E_k(P)) \end{aligned} \tag{7}$$

where k represents the encryption and decryption key.

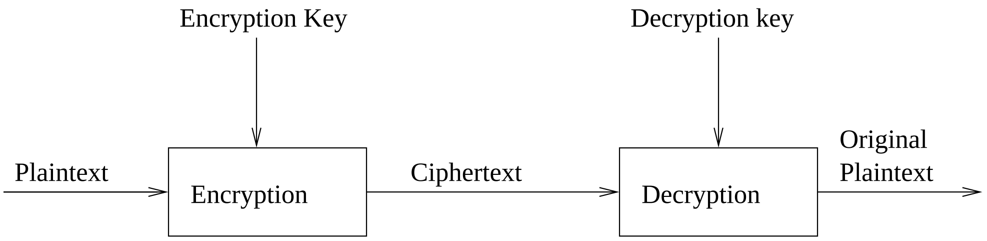

FIGURE 1.4. Encryption and Decryption with key

1.4. FPGA Prototyping Overview

FPGA stands for field programmable gate arrays and use the high densities in modern processes to construct ICs that are completely programmable. An FPGA chip consists of an array of logic cells surrounded by programmable routing resources as presented in Fig. 1.5. There are two basic types of FPGA. The first type uses process option such as fuse and antifuse to permanently program the interconnects and logic and the second type uses static random access memory (SRAM) cells to customize routing and logic functions. The RAM-based FPGA is composed of configurable logic blocks (CLBs) that use lookup tables to compute functions of several variables. The content of the lookup table are contained in the static RAM. The CLBs inputs and outputs (I/Os) are connected to the routing resources. The FPGA is programmed by reading the lookup table stored in RAM and the routing swiches are programmed by the RAM cells [59].

Design flow usually begins at the behavioral level and then moves to the structural level (gates and registers). This step is called Register Transfer Level (RTL) synthesis because the designs are captured using memory element and logic in hardware description language (HDL) [4], [7], [35]. A physical description of the system is then obtained for fabrication as shown in Fig. 1.6.

The system requirements in Fig. 1.6, Fig. 1.7 include both sofware and hardware interfaces. It could also include cost, area and speed requirements. A high level description of the system is done using HDL editor as shown in Fig. 1.7. A behavioral simulation is performed after the high level description. A synthesis of the behavioral description yields a logic level schematic or netlist representation which could be implemented and tested for validation [53],[22].

Validation could be done at different stages of the prototyping. Function simulation could be done at the architectural level and the RTL level. At the implementation level, timing simulation of schematic, netlist or HDL could be done.

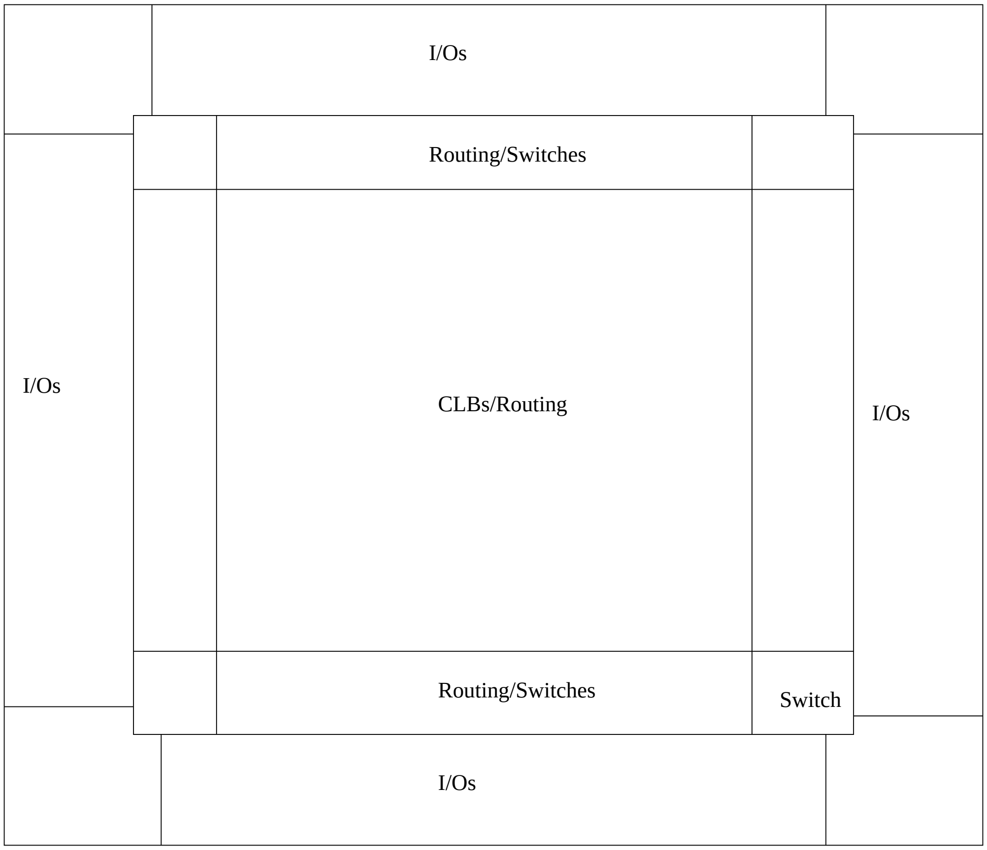

FIGURE 1.5. Generalized FPGA Floorplan

1.5. Still Digital Camera Overview

A digital still camera is a device that is used for capturing and storing digital image files. Fig. 1.8 presents a schematic overview of such a camera. A digital still camera is made up of image sensor that converts light into pixel values. The image sensor could be CCD or CMOS. The amount of detail captured by a camera depends on the number of pixels in the camera [56] [5].

The CMOS sensor converts light into electrons at the photosites. The A/D converter filters, amplifies and converts the analog signal to digital signal. The processor and the controller provides power needed for the imaging algorithm utilized in the camera. The controller coordinates the user interfaces and configurations. The LCD controller controls

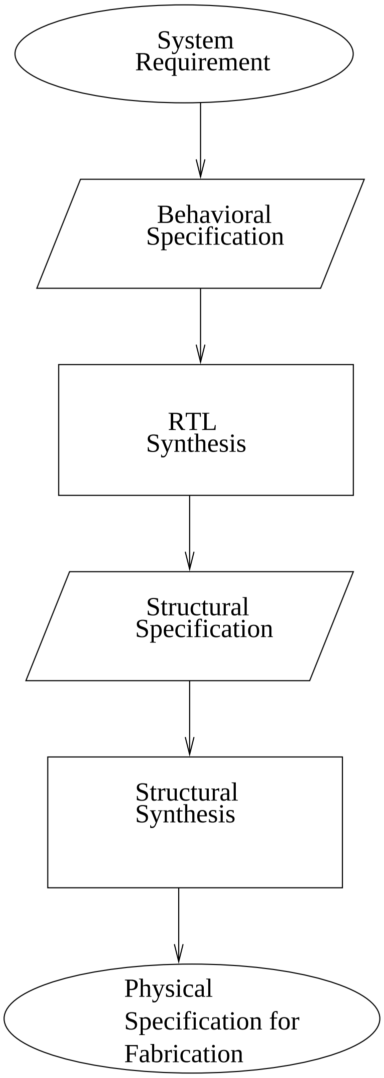

FIGURE 1.6. Generalized Design Flow

the LCD so that the camera image could be displayed. The compression unit compresses the image so that it can be stored in the flash memory [56].

1.6. Research Contribution

The research contributions of this thesis are as follows:

- The problems linked with the inclusion of biometric data in the newly proposed electronic passport are identified.

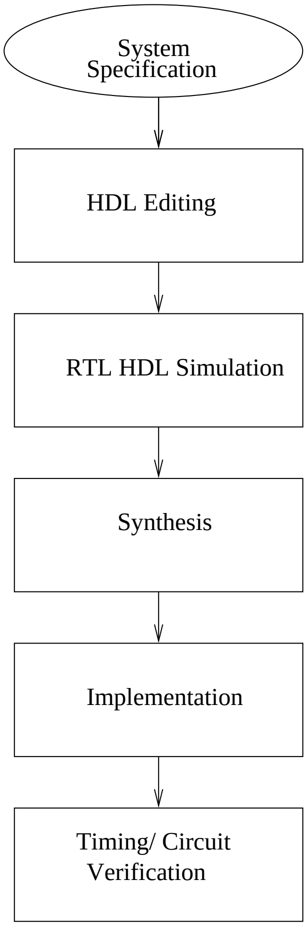

FIGURE 1.7. Generalized Design Flow

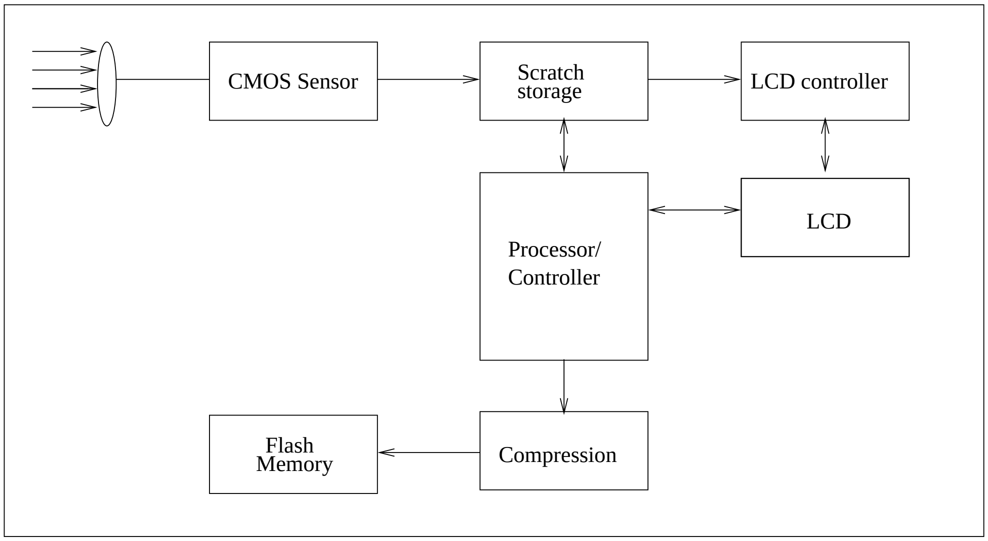

FIGURE 1.8. Simplified Block Diagram of a Digital Still Camera

- A novel approach and architecture in the framework of a Digital Camera conceptualized as a Secure Digital Camera is presented in this thesis with the aim of addressing the unauthorized access and modification of data in passports.
- The very large scale integration (VLSI) architecture of the functional units of the SDC such as watermarking and encryption unit is presented.
- The result of the hardware implementation of Advanced Encryption Standard (AES) and a discrete cosine transform (DCT) based visible and invisible watermarking algorithm is presented.
- The VLSI architecture and the result of the FPGA prototyping of the SDC are presented.

1.7. Our Design Methodologies

A hierarchical approach is used in the design of the secure digital camera. The hierarchical approach involves dividing the system into modules and then in turn dividing the modules into submodules with the aim of reducing the complexity of the system. The use of hierarchical design facilitates design reuse. The approach also aids the understanding of systems when they are viewed at the top level. Hierarchical design involves the concept of abstraction, where at each design level, the internal details of a complex module is abstracted and replaced by a black-box model or box [59]. The abstraction levels in order of increasing abstraction are; the device, circuit, gate, module and system levels as illustrated in Fig. 1.9. A design method that involves the use of an increasing order of abstraction is said to be a "top to bottom" approach. On the other hand, if a decreasing order of abstraction is employed, then it is said to be"bottom to top" approach. For our design, we employ the top to bottom approach. Pipelining and parallelism was employed at different levels of abstraction in our design. Pipelining has to do with breaking important task into smaller operations so that the operations can be done consecutively. There is a level of concurrency involved in pipelining. Parallelism is the process whereby tasks are replicated so that they can be carried concurrently [32].

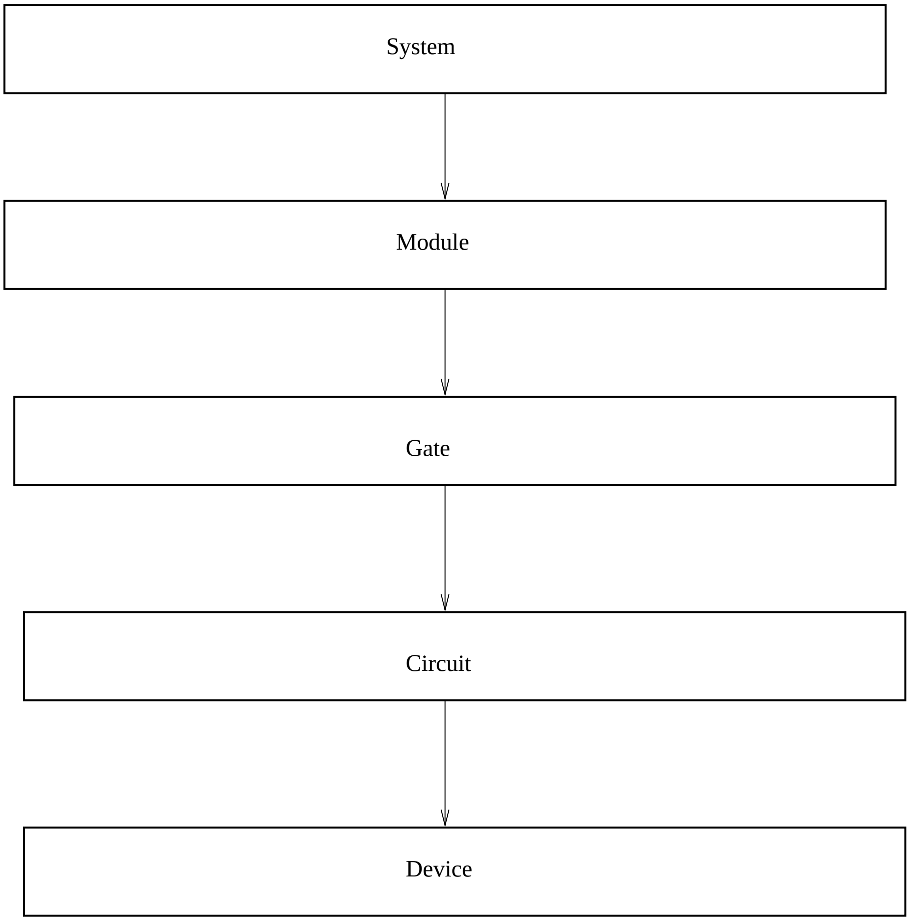

Figure 1.9. Design Abstraction Level in Digital Circuits

- *System Level:* This is the highest level of the design abstraction. The overall structure of the design is considered. The system level abstraction is usually concerned with how modules interconnect. An example of system level abstraction is a processor.

- *Module Level:* In this level, parts of the system are seen to have well defined functions and interfaces. An example is the adder.

- *Gate Level:* The system is viewed as a network of gates and flip-flops. In this level, Logic equations could be used to represent the system. An example is OR gate.

- *circuit Level:* The system is described as a network of transistors. The behavior of the system can be described in terms of the network equations.

- *Device Level* This is the lowest level of abstraction. The individual transistors are considered in this level. An example is PMOS, NMOS

1.8. Outline of the Thesis

The rest of this thesis is organized as follows: Various related works are presented in chapter 2. A detailed design of the secure digital still camera (SDC) is presented in Chapter 3. In Chapter 4, the various algorithms that were used in the design and implementation of the SDC are presented. The architecture of the algorithms chosen for implementation in the SDC is proposed in Chapter 5. The result of prototyping of the SDC and the simulation results are presented in Chapter 6. The Conclusion and future research is presented in Chapter 7.

CHAPTER 2

RELATED RESEARCH AND CONTRIBUTIONS OF THIS THESIS

Though several attempts have been made to present the design and architecture of a digital camera, only few dealt with the design and the architecture of a digital camera with watermarking or encryption capabilities. Most work dealt with the architecture of watermarking unit and encryption unit as a single, separate unit. In this chapter, current research works relating to the architecture and implementation of digital camera, watermarking and encryption are briefly outlined. Section 2.1 deals with research work relating to secure digital camera architecture. Section 2.2 outlines current research works on VLSI architecture for digital watermarking. In Section 2.3, research works relating to VLSI architecture for encryption algorithms are presented.

2.1. Research in Secure Digital Camera Architecture

The trustworthy camera with the aim of restoring credibility to photographic images using encryption is presented in [20]. The camera produces two output files representing the captured image and the digital signature of the captured image. The camera produces an encrypted digital signature produced by applying the camera's unique private key to a hash of the captured image file. A Biometric Authentication System (BAS) for a Secure Digital Camera, is developed in [6], however, a VLSI architecture was not proposed in their work. Authors in [45] designed a CMOS APS imager incorporating circuits using pseudo-random algorithm for watermarking images obtained from the device. A single-chip digital color camera was designed and tested in a 0.8μm CMOS technology by authors in [33]. Authors in [60] compared two architecture for use in a high speed camera. They compare a VLIW processor with a heterogeneous multiprocessor. Authors in [16] present a color camera on a chip with high throughput ADC developed by photobit. A digital still camera system on a chip with 32-bit RISC, CCD signal processor, and high-speed JPEG was developed by authors in [46].

Industries have also produced cameras with watermarking capabilities. For example, Epson released the PhotoPC 3000Z and 800Z model and Kodak also manufactured the DC-200 and DC-260 but were all discontinued [6]. However these camera models were discontinued.

2.2. Research in Watermarking Architecture

Authors in [42] presented a VLSI architecture for implementing two visible digital watermarking scheme. Their work is aimed at integrating the chip in a digital camera framework. The authors also implemented the watermarking algorithms proposed by authors in [55], [5], and [27]. A Dual Voltage-Frequency low power VLSI chip for image watermarking in DCT domain was proposed by authors in [43]. They presented a new VLSI architecture that can insert invisible or visible watermarks in images in the DCT domain. They utilized low power techniques such as dual voltage, dual frequency in their architecture. Authors in [52] proposed the implementation of a real-time spatial domain watermark embedder and detector on a very long instruction word (VLIW) processor. Authors in [36] proposed a hardware architecture for video watermarking algorithm called just another watermarking system (JAWS) taken into consideration the time and area constraints. They carried out a study on the architecture to understand the relationship between algorithm features and implementation. The purpose of their work was to identify some hardware based techniques that will improve performance and to present a VLSI architecture of JAWS algorithm. A VLSI implementation of digital watermarking technique for gray scale images was presented by authors in [19]. Their implementation was for fragile invisible watermarking algorithm in spatial domain which could be used for online watermarking application. Authors in [57] presented a DCT domain invisible watermarking chip. Authors in [40] presented an FPGA based implementation for an invisible spatial based watermarking algorithm. The authors in [14] proposed a hardware efficient architecture of wavelet-based watermarking scheme. They employed discrete wavelet transform (DWT) for their embedding process. They presented an architecture that is based on the algorithm proposed by [24].

2.3. Research in Encryption Architecture

The authors in [23] presented an AES processor that is based on trade off between area and throughput. They presented a fully pipelined architecture for an optical network that requires a throughput of 30 Gbits/s. They reduced the area consumption by 35 percent by pipelining the composite field implementation. The authors in [25] presented a side-channel attack resistant coprocessor IC that is made up of an AES based cryptographic engine. The authors in [29] proposed a high performance, high throughput, and area efficient architecture for Rijndael advanced encryption standard (AES) algorithm. They specifically pointed out that their architecture is suited for low silicon implementation. They used pipelining to achieve high throughput and forward and reverse key scheduling to reduce the area consumption of their implementation.

Two different architectures and VLSI implementations of the Rijndael algorithm was proposed by authors in [51]. They presented an architecture that support both encryption and decryption. Their architecture was optimized with covered area constraints. They also used pipelining in their implementation. Authors in [62] proposed a high-speed VLSI architecture for AES algorithm. They used combinational logic to implement their SubBytes and InvSubBytes transformation as opposed to the use of look-up tables. They utilized subpipelining technique in their implementation. The area required for implementation in their design is reduced by the use of composite field arithmetic. They pointed out that the cost of more subpipelining stages are extra registers and larger counters. The authors in [58] presented the design of an AES encryption chip. Their architecture was designed for best performance in different modes. Some of the earlier implementation of the Rijndael AES before and after it was accepted as a standard were presented by authors in [61], [48], [31], [17], [13], [18], [15].

The architectures proposed by the authors in the related works were for watermarking and encryption as a separate unit. The architecture of encryption and watermarking as a single unit is presented in this thesis. To my knowledge, the chip presented in this thesis is the first to possess simultaneous encryption and watermarking functionalities.

CHAPTER 3

SECURE DIGITAL CAMERA - A CONCEPTUAL HIGH LEVEL DESCRIPTION

The high level description of the SDC is presented in this chapter. The proposed biometric based security system comprising of the “enrollment process” and the “verification process” is presented. The SDC is presented as a tool to implement the proposed biometric based security system. The high level architecture of the SDC is presented in Section 3.1. The enrollment process is discussed in Section 3.2 and the verification process is explained in Section 3.3.

3.1. High-Level Architecture

In the proposed SDC, the image is captured by an image sensor (CCD or CMOS) and converted to a digital signal by the A/D converter. The captured digital image is stored temporarily in the scratch memory, after which it is displayed on the liquid crystal display (LCD) panel with the help of the controller. The purpose of the LCD panel is to enable the user to see the image frame before it is watermarked by the watermarking unit and stored in the camera. This can then be further transmitted over the network, or transferred to flash memory, computer hard drive or optical discs. The controller unit is responsible for controlling the entire sequence of events. Our proposed architecture for the camera design will handle both color and monochrome images. In addition, the proposed SDC carries out both invisible robust and visible watermarking scheme [44] along with encryption and data compression [21], [37]. The choice of the operations performed on the image is dependent on the user of the camera. The security of our system will be dependent on the encryption module that will be based on the advanced encryption standards (AES) algorithm [9]. The structural aspects of the controller, watermarking, encryption, and the JPEG units is presented. The JPEG unit is involved in the compression of the image before it is stored in the flash memory [21]. The High-Level architecture of the SDC is shown in Fig. 3.1

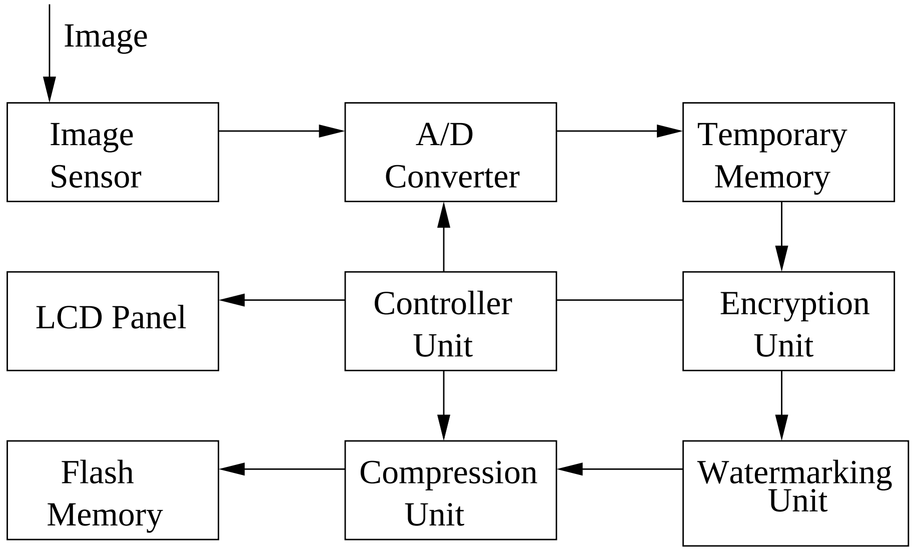

FIGURE 3.1. The High-Level Schematic Architectural Description of the Proposed SDC

3.2. Enrollment Process

The passport image is watermarked with biometric data in this process using our proposed Secure Digital Camera. When an individual applies for passport or ID card, our proposed digital camera is used to invisibly watermark the individual's binary compound biometric data such as "fingerprint", "iris scan", "hand written signature", etc into the applicant's image taken by the camera. A compound biometric image is produced by using a mixer as shown in Fig. 3.1. The original biometric data is acquired externally through a USB interface by connecting the camera with acquisition devices at the time of image capture. The binary compound image is encrypted by the Rijndael advanced encryption standard (AES) [1] [29] [10] unit and then watermarked into the passport image of an individual. The binary watermark assists verifying the identity of the individual and in detecting any form image modification. The camera is designed such that the user could choose to visibly watermark the host image with serial number of the camera in the form of a barcode in

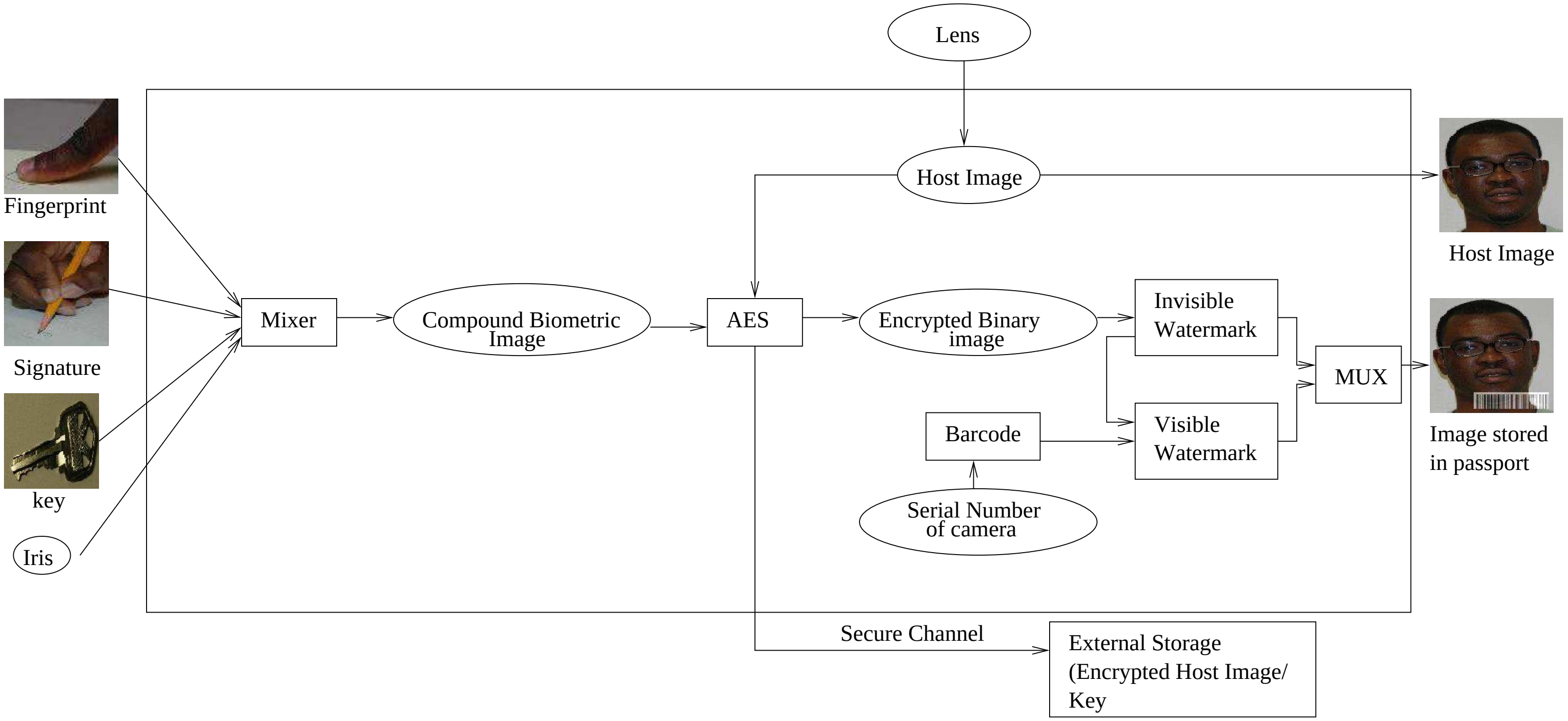

FIGURE 3.2. Flow of the Enrollment Process in SDC

addition to the invisible watermarking as shown in Fig. 3.1 to uniquely identify the image capture source. The individual's un-watermarked image (Host Image) is encrypted and stored through a secure channel in an "external storage" to be later used for verification purposes.

3.3. Verification Process

This process occurs during the verification of the individual person's identity and picture image. It is assumed that the photo image is stored in a smart chip on the cover of the passport. At the checkpoint, the image is acquired by scanning the passport. The encrypted compound biometric data is extracted from the acquired image and decrypted using the original key that was earlier stored in a secure external storage as shown Fig. 3.3. The compound biometric data is then passed through a mixer for separation. The biometric data is then compared with the biometric data of the passport's owner. Our system is able to detect whether the picture image has been modified. The fact that the biometric data is encrypted before it is watermarked invisibly into the image makes it difficult for easy access, thereby reducing the privacy concerns [39].

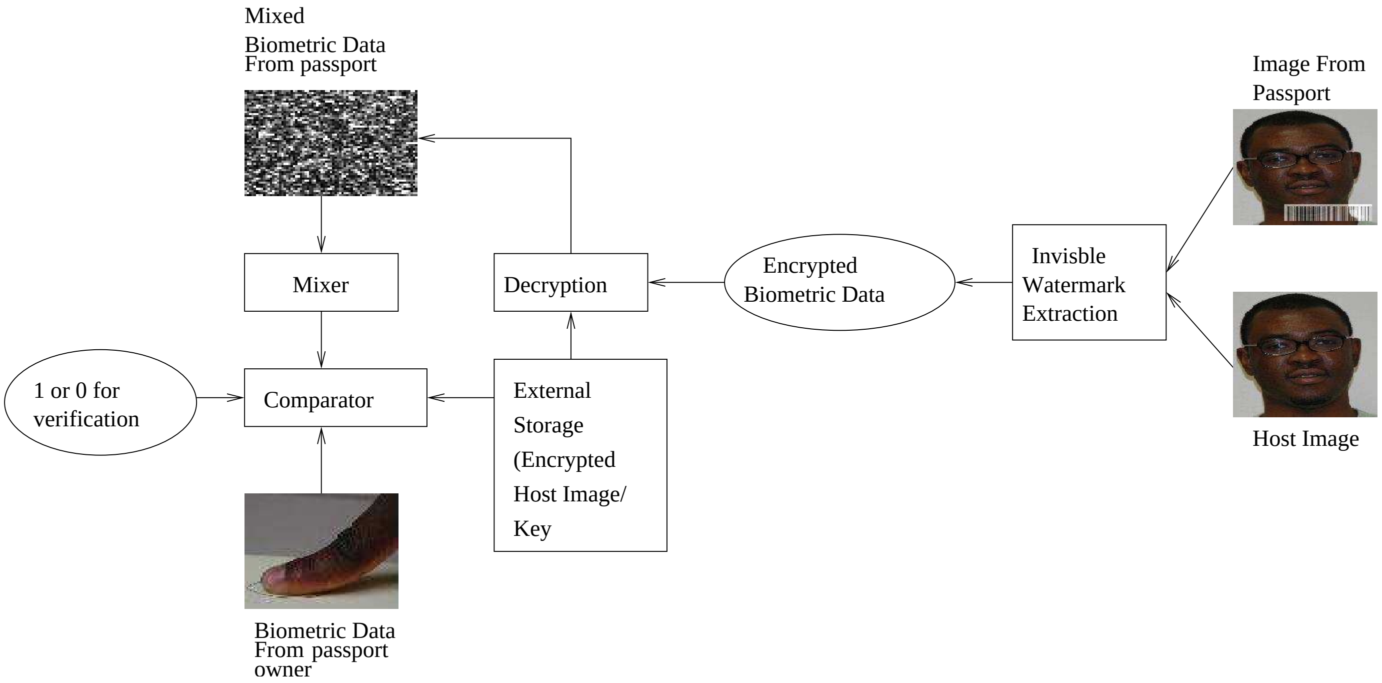

FIGURE 3.3. Flow of the Verification Process in SDC

CHAPTER 4

ALGORITHM SELECTED FOR ARCHITECTURAL IMPLEMENTATION

In this chapter we introduce the algorithms selected for the architecture development and inclusion in the camera. A DCT based visible and invisible watermarking algorithm and the Rijndael AES algorithm was chosen for implementation in the proposed SDC. In Section 4.1, the visible watermarking algorithm whose architecture was developed is presented. The DCT based invisible watermarking algorithm implemented in the SDC is discussed in Section 4.2 . In Section 4.3, we discuss the Rijndael AES algorithm.

4.1. DCT Based Visible Watermarking Algorithm

The DCT based visible watermarking algorithm proposed by [41], [27] was chosen for implementation in the digital camera. The algorithm is based on human visual system (HVS). A statistical analysis of the host image (passport photo) is carried out using a mathematical model to determine the scaling factors of the visible watermarking process. The host image and the watermark image (barcode image) is divided into 8×8 blocks. The DCT coefficient of the 8×8 blocks of host I and watermark image W is then calculated. The DCT coefficient of block n of the host image is represented by $c_{ij}(n)$ and that of the watermarked image is w_{ij}, where n represents the position of block in image I. The DCT coefficient of watermark image (barcode) is combined together blockwise with host image (passport image) to obtain the watermarked image (barcode marked passport photo) using Eqn. (16). The flowchart of the algorithm is shown in Fig. 4.1.

$$c'_{ij} = \alpha_n c_{ij} + \beta_n w_{ij} \tag{8}$$

where α_n and β_n are the scaling factors for a block indexed as n.

The mathematical model for α_n and β_n was derived taken into consideration the points made by authors in [27], [11], [54], [12]. The mathematical model is such that the edge blocks are least altered to minimize the level of distortion of the image. The host image's scaling factor α_n is close to the maximum value of scaling factor and the watermark's scaling factor β_n is close to minimum. Also, the AC DCT coefficients of highly textured blocks are found to possess small variances. As a result, scaling factor α_n of passport photo image is assumed to be directly proportional to the variance σ and β is found to be inversely proportional to the variance σ_n. Based on [41], the scaling factors α_n and β_n for edge blocks are chosen to be α_{max} and β_{min} respectively, while those of non-edge blocks are calculated using Eqn. (9).

$$\begin{aligned} \alpha_n &= \sigma_n^{'} exp.(-(\mu_n^{'} + \mu^{'})^2) \\ \beta_n &= \frac{1}{\sigma_n^{'}}(1 - exp.(-(\mu_n^{'} - \mu^{'})^2)) \end{aligned} \tag{9}$$

Using Taylor series Eqn. (9) could be further simplified to obtain Eqn. (10).

$$\begin{aligned} \alpha_n &= \sigma_n^{'}(1 - (\mu_n^{'} - \mu^{'})^2 + (\mu_n^{'} - \mu^{'})^4) \\ \beta_n &= \frac{1}{\sigma_n^{'}}((\mu_n^{'} - \mu^{'})^2 - (\mu_n^{'} - \mu^{'})^4) \end{aligned} \tag{10}$$

The passport image's scaling factor is directly proportional with the mean gray value of each block (μ_n) if $\mu_n < \mu$ and inversely proportional if $\mu_n > \mu$. The mean gray value for each block of the original image is calculated using Eqn.(11), where $c_{00}(n)$ represents the DC coefficient of block n.

$$\mu_n = c_{00}(n) \tag{11}$$

The gray value of normalized mean of block n is calculated using (12) where c_{00max} represents maximum value of $c_{00}(n)$:

$$\mu_n^{'} = \frac{c_{00}(n)}{c_{00max}} \tag{12}$$

The maximum value of normalized mean of the host image I is calculated with (13)

$$\mu^{'} = \frac{1}{N}\sum \ (c_{00}(n)) \tag{13}$$

The variance of AC-DCT coefficients is calculated using the expression below:

$$\sigma_n^{'} = \frac{1}{64} \sum \sum \; (C_{00} - \mu_{nAC})^2 \tag{14}$$

where μ_{nAC} represents the AC DCT coefficients.

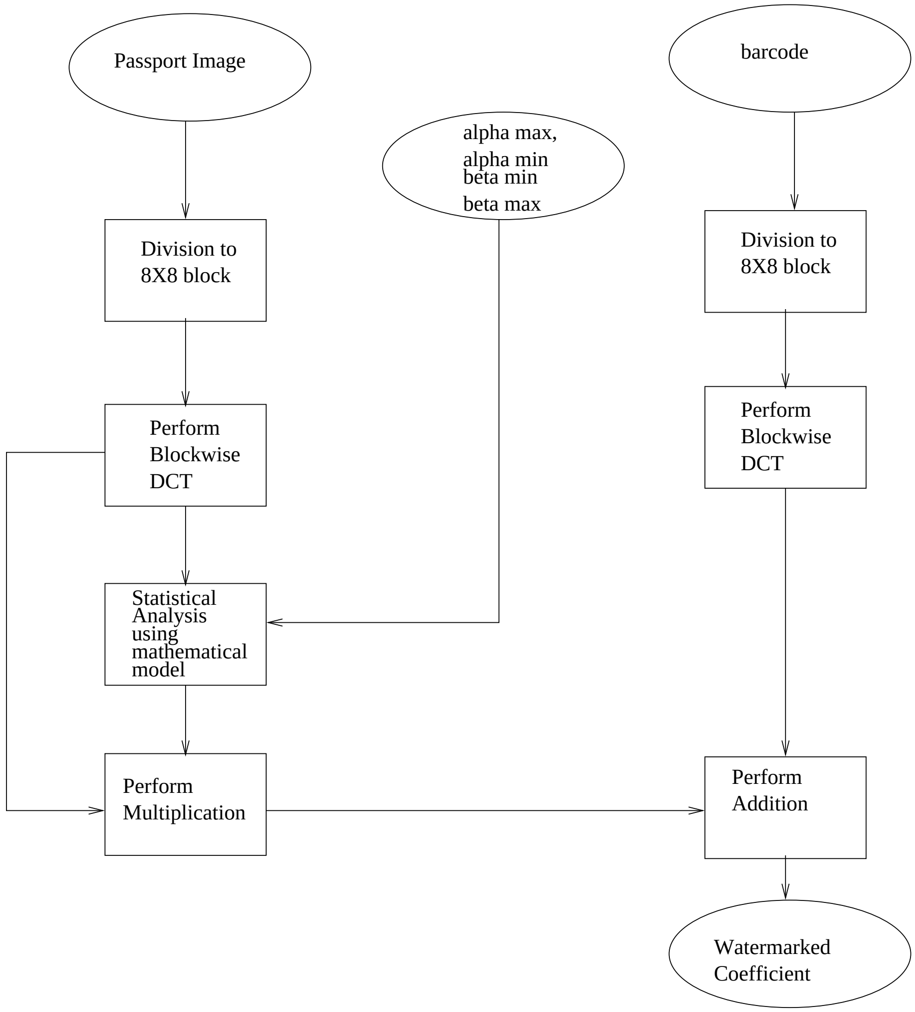

FIGURE 4.1. Flowchart of Visible Watermark algorithm

4.2. DCT Based Invisible Watermarking Algorithm

The invisible watermarking algorithm [39] used in the SDC is discussed in this section. The algorithm inserts a binary image as a watermark into the host image (passport photo). The watermark is inserted in the perceptually significant components so that watermarked image could be robust to common geometric distortions, signal distortions, and malicious attacks.

4.2.1. *Invisible Watermark Insertion*

The algorithm could be divided into three main phases: preprocessing, DCT and insertion phase. The first phase is the preprocessing phase where the image is decomposed to obtain the required component. For example, if the image is RGB, it is converted to the Y-Cr-Cb. This phase also include any necessary modification to the size of the image to enhance the division of the image to appropriate number of blocks.

After the preprocessing phase, the host image I is divided into 8x8 blocks and the DCT transform of each 8x8 block of the host image I is calculated. The total number of blocks M in the host image is calculated using the number of pixel column-wise ($ncol$) and row-wise ($nrow$) as shown in Eqn.(15).

$$M = \frac{nrow \times ncol}{64} \tag{15}$$

If the image is an RGB image, the DCT coefficient of the Y component is obtained. For the insertion phase, the DC component c_{00} and the three low frequency components c_{00}, c_{10}, and c_{11} are considered for insertion. The watermark (biometric image) is partitioned to the same number of blocks as the host image (passport photo) with the block size as 2×2. The insertion process is carried out using Eqn.(16) where the watermark's(biometric image) binary value in block k is $w_{ij}(k)$.

$$\begin{gathered} \forall i, j, \text{ and } k \\ c'_{ij}(k) = \begin{cases} c_{ij}(k)(1+\alpha_{ij}) & if \quad w_{ij}(k) = 1 \\ c_{ij}(k)(1-\alpha_{ij}) & if \quad w_{ij}(k) = 0 \end{cases} \end{gathered} \tag{16}$$

A combination of both addition and subtraction is done in Eqn.(16) so that a statistical analysis of the passport photo will not reveal the presence of watermark. A value of 0.1 is

TABLE 4.1. Notations used in Describing the Algorithm

I	Original or host image
$I^{'}$	Watermarked image
(i,j)	Location of pixel or coefficients
$nrow$	Number of image pixel row-wise
$ncol$	Number of image pixel column-wise
c_{ij}	(i,j)th frequency or DCT coefficient
α_{dc}	Scaling factor for dc component
α_{ac}	Scaling factor for ac component
M	Total number of 8x8 blocks
k	block number

used for α_{ac} and 0.02 for α_{dc}. These values are the scaling factors for the dc component and the first three AC component after a zig-zag scan. The flowchart of the invisible watermark insertion is shown in Fig. 4.2.

4.2.2. *Invisible Watermark Extraction*

The passport image (Image Under Test) $I^{'}$, live biometric data, key and the original image I, stored in an external drive through a secure channel is needed for extraction of the watermark. The passport image $I^{'}$ and the original image I is divided into 8×8 block and then converted from RGB to YCbCr representation. The DCT transform of both images are calculated and their corresponding blocks are compared. If a dct coefficient in a passport image's block is bigger than the corresponding block in that of the original image, then the watermark be is 1, else it is zero. The flowchart of the extraction process is shown in Fig. 4.3.

4.3. Rijndael AES Algorithm

In this section we discuss the AES encryption algorithm and visible watermarking algorithm whose architectures are presented. The Rijndael algorithm was chosen for implementation in the digital camera. Rijndael is a key-iterated block cipher where the round

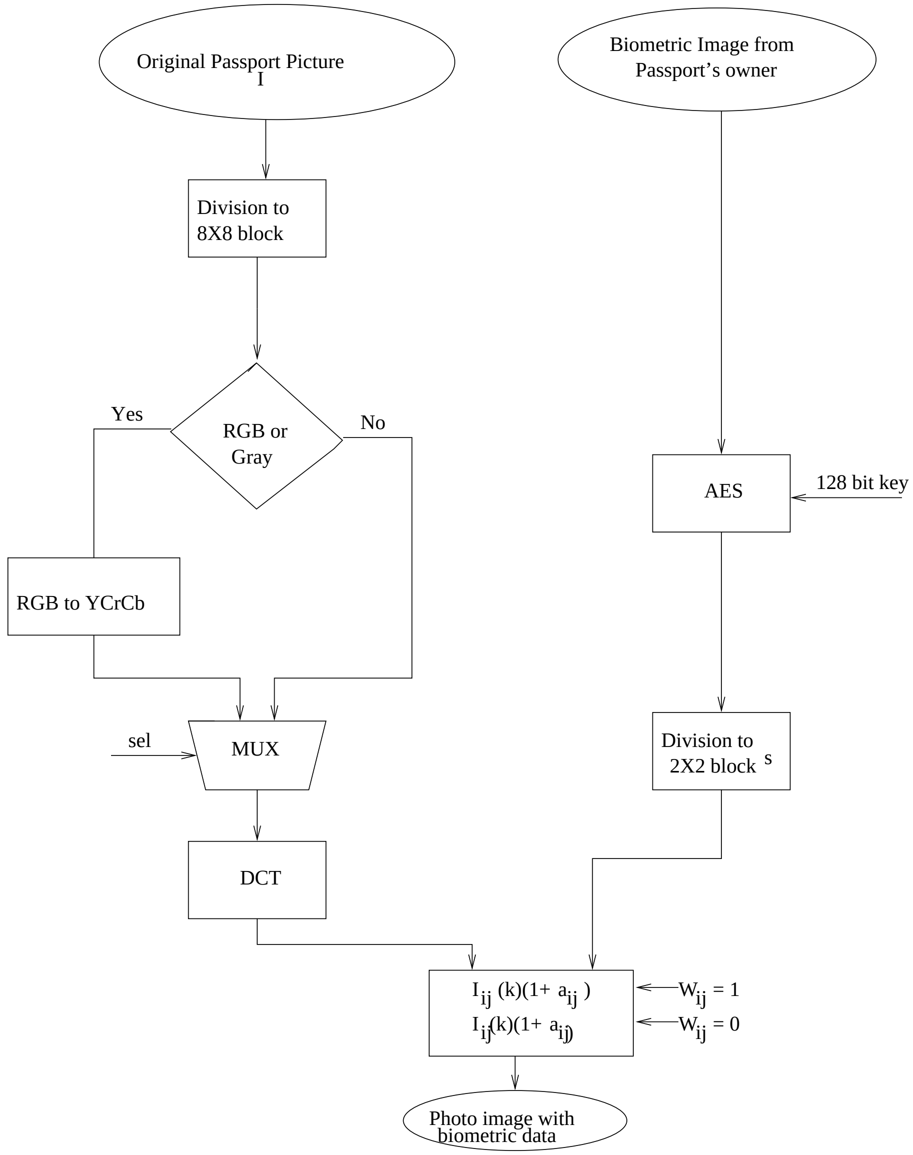

FIGURE 4.2. Flowchart of Invisible Watermark Insertion Algorithm

key needed to produce cipher text from plain text is derived from initial key [10]. The keys derived from the initial key are repeatedly applied for the transformation of the plain text. The algorithm is a linear transformation cipher that is based on S-boxes. The Rijndael algorithm supports fixed block and variable key length of 128, 192 or 256 bits [1] [29]. Data are handled as bytes in the Rijndael algorithm in which each byte forms an element in a polynomial representation of Galois Field $GF(2^8)$[9].

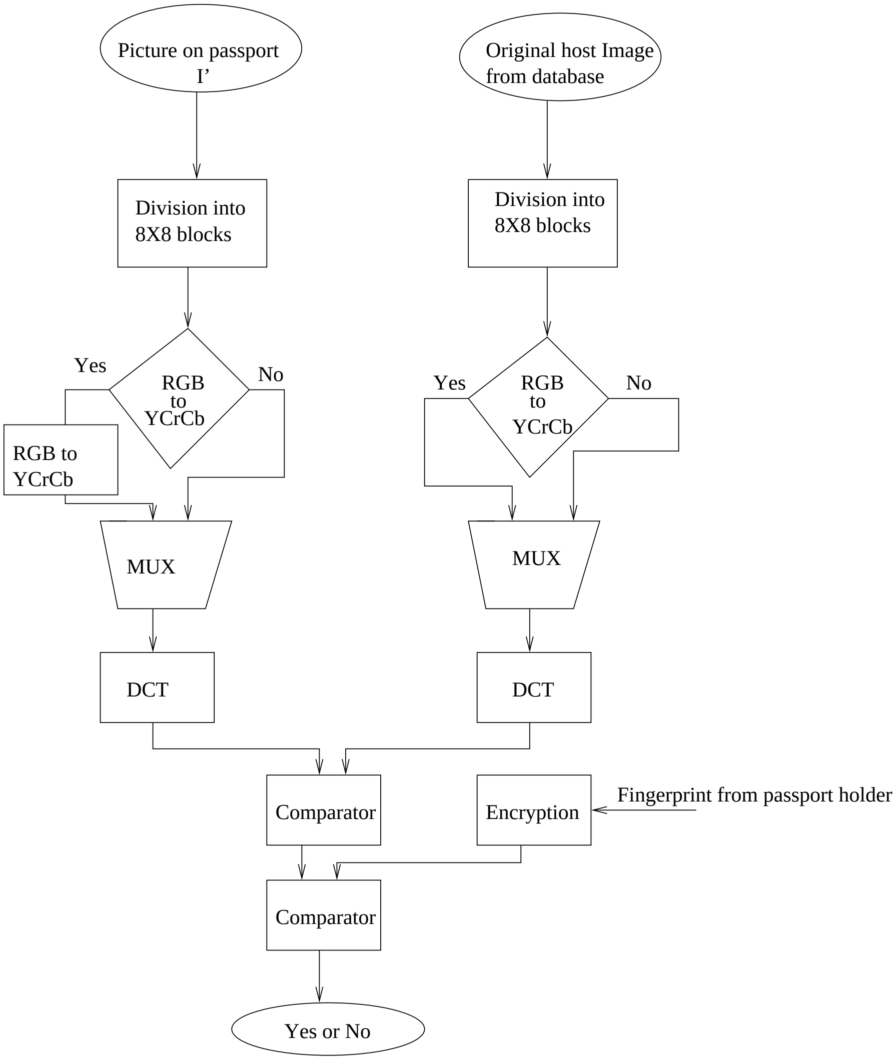

FIGURE 4.3. Flowchart of Invisible Watermark Extraction Algorithm

The encryption process can be broadly divided into "initial", "standard" and "final" round phases as shown in Fig. 4.4. The initial round phase involves initial key and data addition. The standard round is made up of four operations, byte substitution, row shifting, column mixing, and Addition of round keys. These four operations are needed to complete one round. The number of round carried out depends on the key length. The AES with 128 bits key and block length consists of 10 rounds along with an initial adding of the round key. The final round has the same operations as the standard round except the Column Mixing operation.

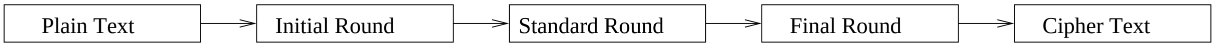

FIGURE 4.4. Block diagram of Rijndael Algorithm

CHAPTER 5

DEVELOPMENT OF THE PROPOSED ARCHITECTURES

In this chapter, we present the architecture of several units and modules of the SDC. The techniques and stages involved in the development of the architecture is described. In Section 5.1, we present the architecture of the visible watermarking module. The architecture of the DCT-based invisible watermarking module is presented in Section 5.2. The watermarking unit comprises of both the visible and the invisible watermarking module. In Section 5.3, the architecture of the encryption unit is presented.

5.1. Visible Watermarking Unit

The high level architecture of the visible watermarking module is shown in Fig. 5.1. The visible watermarking module is composed of several submodules, such as DCT, perceptual analyzer, edge detection, scaling factor, insertion, row and column address decoder, registers and controller. The DCT module calculates the DCT coefficients of host and watermark images before they are stored in the scratch memory. The controller governs the operations of all the other modules and the data flow in the watermarking unit. Address decoders are used to decode the memory address where the image and watermark are stored.

5.1.1. *Perceptual Analyzer*

The mean gray value of host image block is computed from DCT coefficient and it is the same as the DC value of the 8×8 DCT block coefficient. In addition, the mean (α) and variance (σ) of the AC-DCT coefficients for each block is computed using the perceptual analyzer and the architecture is shown in Fig. 5.2. It is made up of two dividers, an accumulator and a subtractor. The accumulator and the divider is used in calculating the mean. The subtractor and the divider is used for calculating the variance.

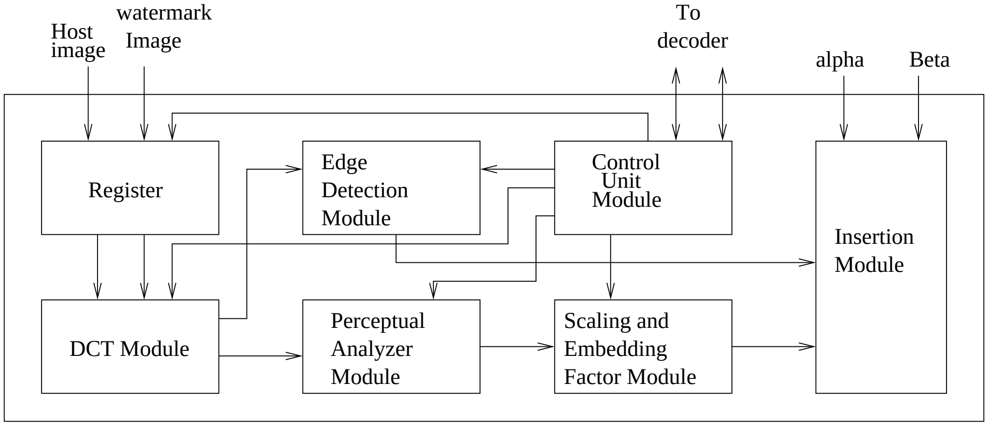

FIGURE 5.1. High Level Architecture of Visible Watermarking Module

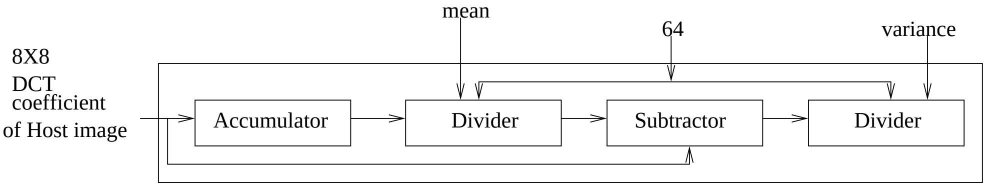

FIGURE 5.2. Architecture of Perceptual Sub-module

5.1.2. *Scaling Sub-module*

The scaling factor determines the ratio of host and watermark image in the final watermarked image and is calculated using the scaling and embedding factor sub-module whose architecture is shown in Fig. 5.3. In order to simplify the architecture, Taylor series approximation was applied to the original equation. The scaling factor is made up of three multipliers, two adders and two subtractor. If there is no area constraints, the module could be duplicated to increase speed.

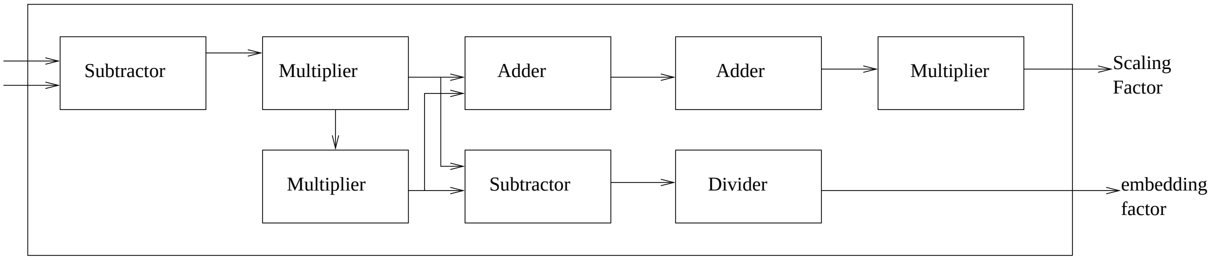

FIGURE 5.3. Architecture of Scaling Sub-module

5.1.3. *Edge Detector Sub-module*

The blocks that are at the edge of the image are determined by the edge detector sub-module shown in Fig. 5.4. A threshold constant is given as an input to the module for edge detection. The edge detection module is made up of three parts that perform accumulation, comparison and detection functions. The absolute values of 8×8 AC-DCT coefficients from the memory are passed into the accumulator and divider sub-module. The result from the sub-module is then compared with a threshold value to determine an edge or non-edge block. The final watermarking process is carried out by the insertion sub-module shown in Fig. 5.5.

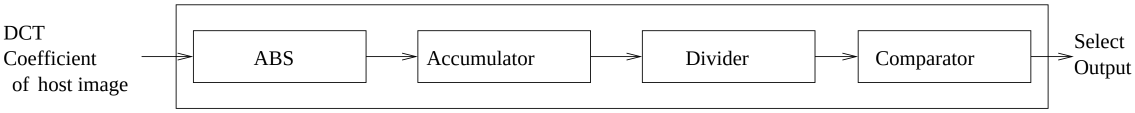

FIGURE 5.4. Architecture of Edge Detector Module

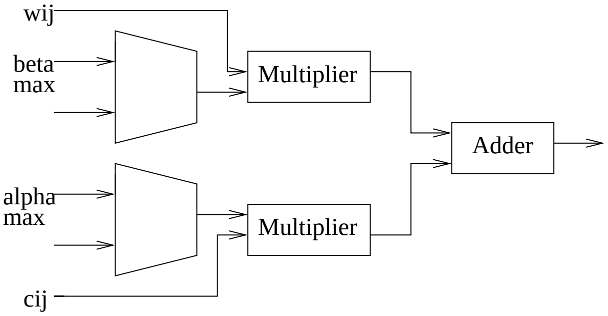

FIGURE 5.5. Architecture of Watermark Unit

5.1.4. *Visible Watermark Insertion Sub-module*

In the insertion sub-module, watermark insertion is carried out using the values provided by the edge detection, and the scaling factor modules. This module is made of two multipliers and an adder for evaluating Eqn.16. The insertion submodule inserts the watermark into the original image. The insertion is carried out using the values provided by the edge detection, and the scaling factor modules. This module is made of two multipliers and an adder. The architecture is made up of two multiplexers that are used to choose between the maximum and AC alpha values from the scaling submodule. The architecture for the insertion submodule is shown in fig. 5.5.

5.1.5. *Visible Watermark Controller*

The controller has 7 states, Init, S0, S1, S2, S3, S4 and S5 as shown in Fig. 5.6. The state "Init" represents the initial state. There is no transition to state S0 unless the start button is pressed. In state S0, the image pixel is written to the appropriate RAM. The image pixels are then read from the RAM for DCT operation to be performed on the pixels in state S1. The resultant DCT coefficients are written back to the RAM in state S2. The DCT coefficients are then read from the RAM in state S3 for the purpose of performing the watermarking operation. In state S4 the watermarked pixels are written back to the RAM waiting for it to be read by the next unit. The state machine has an enable output that becomes one when the watermark operation is complete.

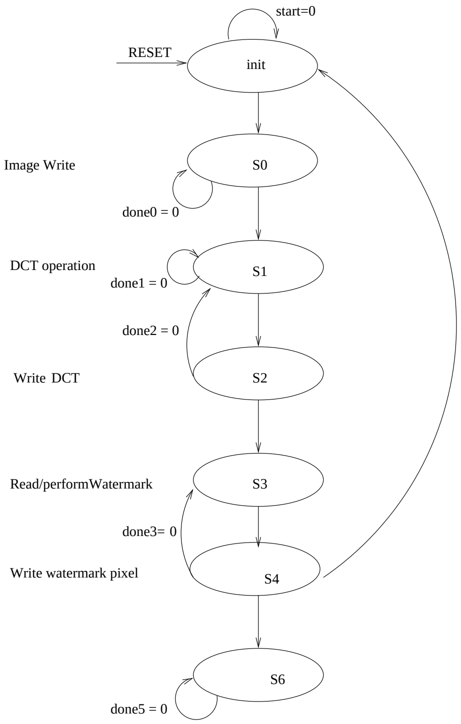

FIGURE 5.6. Architecture of Visible Controller Module

5.1.6. *DCT Sub-module*

The DCT module calculates the DCT coefficient of the host image and it consists of 2-1D DCT sub-module. The architecture is explained in section 5.2.

5.2. Invisible Watermarking Module

The architecture of the invisible watermark algorithm used in our secure camera is presented. The invisible watermark chip is made up of the controller module, DCT module, and the insertion module. The proposed architecture in this section will be discussed.

5.2.1. *DCT Sub-module*

The DCT module is made up of two 1D DCT sub-module as shown Fig. 5.7. The algorithm proposed by [8] was implemented in our secure digital camera. The architecture has sixteen multipliers and twelve adders. The 1D DCT sub-module first computes the row DCT of each 8×8 block, after which the column DCT of each block is computed. The 1D DCT is defined by Eqn. (17)

$$Y = AX \tag{17}$$

where Y is DCT coefficients, X represents the original image to be watermarked. A buffer circuitry is used to assist in finding the transpose. It also serves as a temporary storage for the first 1D DCT coefficients. In order to reduce the latency, a multiplexer is used between the buffer and and the second 1D DCT submodule. Registers as opposed to RAM cells [30] were used to design the transpose buffer in order to reduce the latency and to increase performance. The DCT module does not have a separate controller, it is controlled by the digital camera's controller.

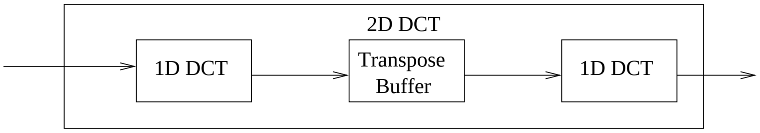

FIGURE 5.7. 2D DCT Architecture

5.2.2. *Invisible Insertion Sub-module*

The insertion process is carried out using the insertion module. The architecture of this module is shown Fig. 5.8. The architecture is made up of one multiplier, two multiplexer, one adder, one subtractor and two latches. The insertion unit only takes the DC component (c_{00}), the first (c_{01}), second ((c_{10}) and third (c_{11}) AC component of each 8×8 block for watermarking. The mux2 is used to select unwatermarked coefficients. The use of parallelism in the implementation Fig. 5.9 of the module gives the capability to watermark a DCT block in two clock cycle instead of four. This improves the performance of our system, however, there is a trade-off between the performance and the area used. Latches were used for temporary storage in the insertion module in order to improve latency.

5.2.3. *Invisible Watermark Control Unit*

The architecture of the controller that drives the datapath is shown in Fig. 5.10. The controller is modeled as a finite state machine with seven states. Transition from the initial state (init) to S0 occurs when the start signal is high. The pixels(I_{ij}) are read from storage to input register for their DCT coefficients to be calculated. The first DCT operation is carried as a pipelined operation. If the DCT coefficient of all the coefficients of a block is not completed, there is a transition from state $S2$ to state $S0$. Transition is made to state $S3$ for the second DCT operation after the completion of the 1D DCT operation on the original image pixel (I) of the block. Due to the use of transpose buffer and the multiplexer discussed in the previous section, the input to the second DCT is done in a parallel fashion. The 2D DCT coefficients (c_{ij}) of the original image is obtained in state $S4$. The watermarking process is performed on (c_{ij}) and then written to RAM or displayed in state $S5$. If all the coefficients of the block are watermarked, a transition occurs to the initial state.

5.3. Encryption Unit

A high level view of encryption unit architecture is presented in Fig. 5.11. Our implementation supports 128 bits of data and key length. The initial round module is carried out by XORing the 128 bits plain text with the 128 bits input key. The plain text input

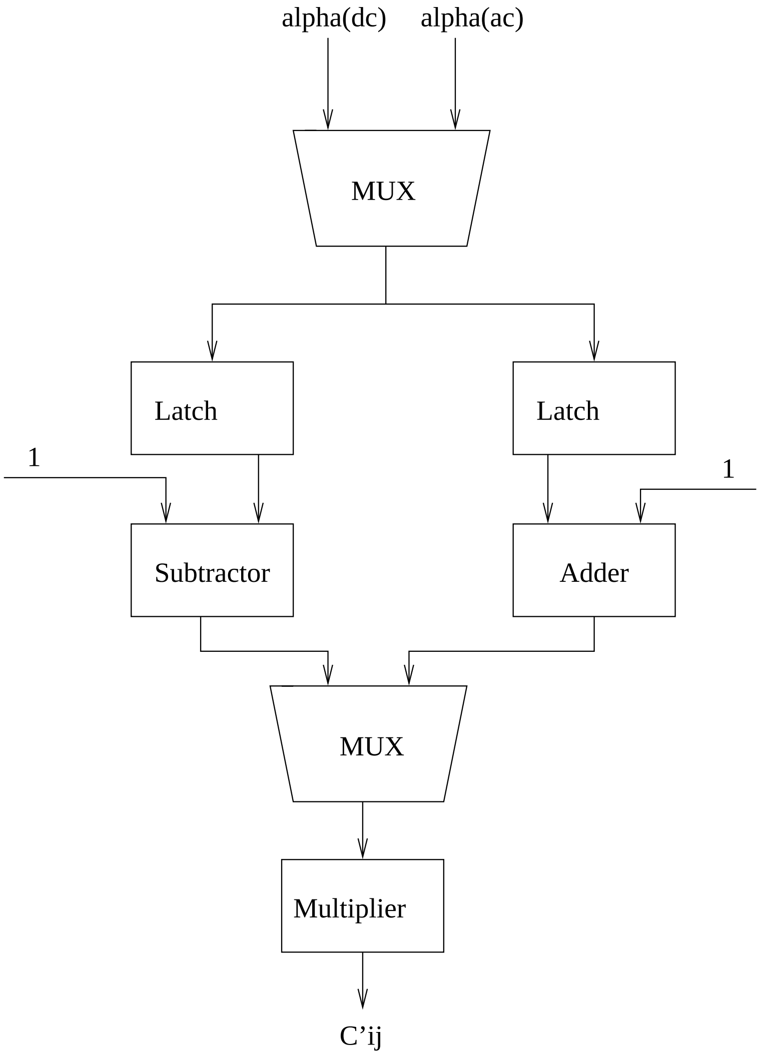

FIGURE 5.8. Architecture of Invisible Insertion Module

and the key input is retrieved from the input register. The output from the initial round is then passed through a 4 to 1 multiplexer to the register for temporary storage, after which it is then passed to the round module. A round key is generated for each round by the key schedule. The output is iterated back into the round module through the muliplexer. The round module is executed nine times. The control module takes care of the sequence of operation of the encryption system.

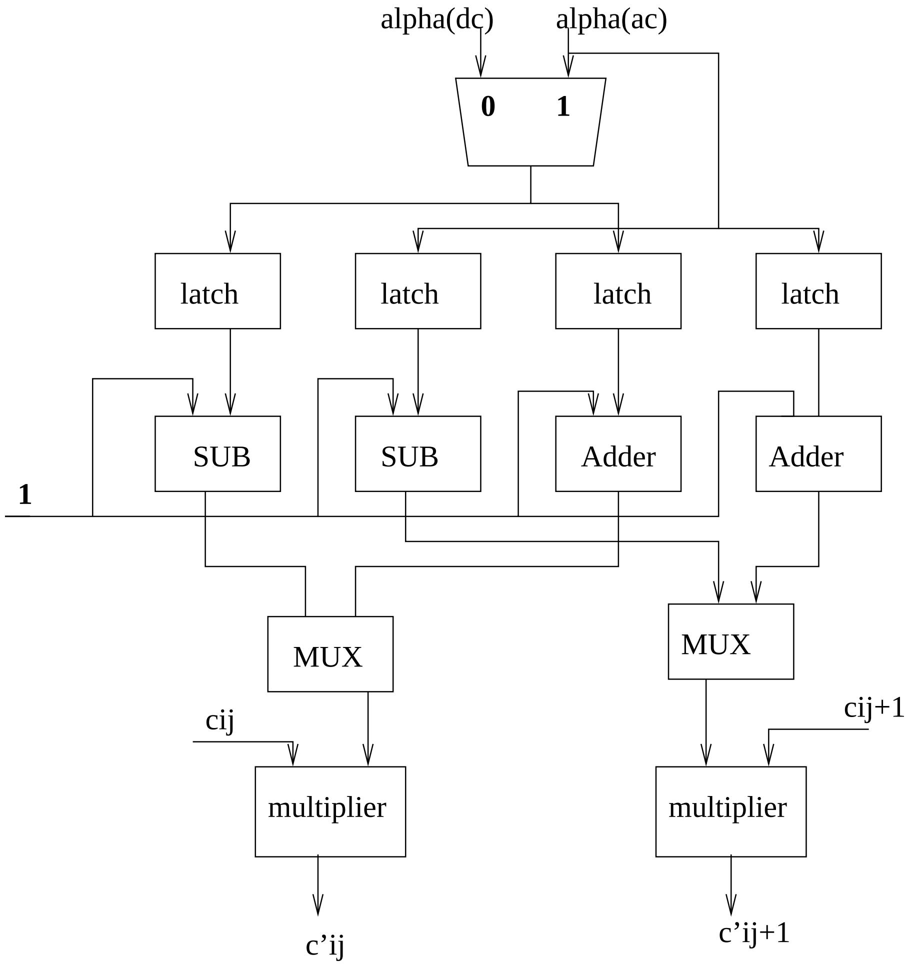

FIGURE 5.9. Modified Architecture of Insertion Module

5.3.1. *Round Module*

As shown in Fig. 5.12, the round module is made up of the Bytesub, Shiftrow, Mixcolumn, and the Addroundkey submodules. The Mixcolumn submodule is not used in the final round.

- ByteSub : This is made up of multiplicative inverse in $GF(2^8)$ and affine mapping over $GF(2)$ transformations as shown in Fig. 5.13. The field $GF(2^8)$ is defined by finding a polynomial that is irreducible over $GF(2)$. The byte substitution operation is carried out in this paper with the aid of S-box. The architecture consists

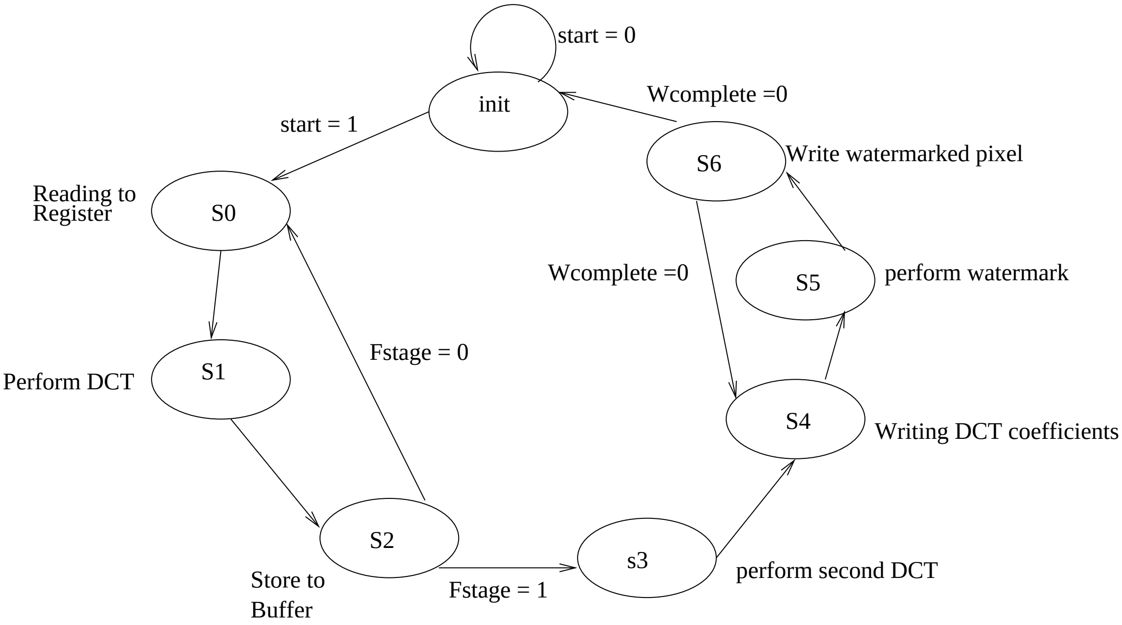

Figure 5.10. State Diagram of Invisible Watermarking Controller

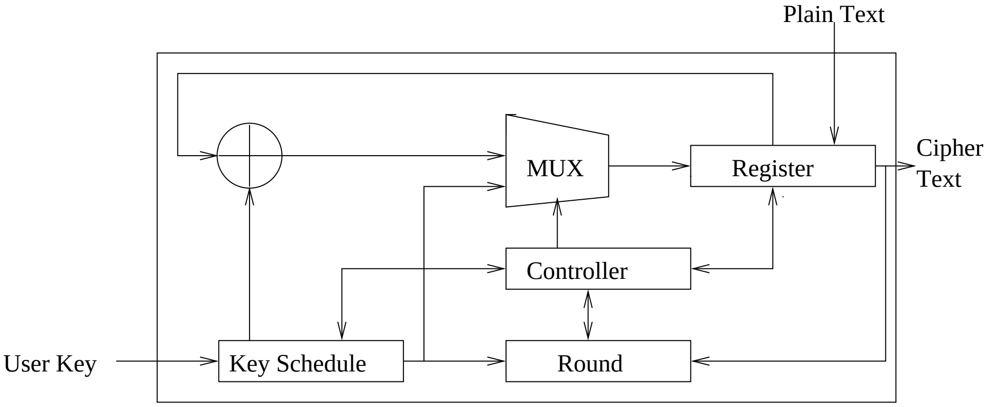

Figure 5.11. High Level Architecture of Encryption Unit

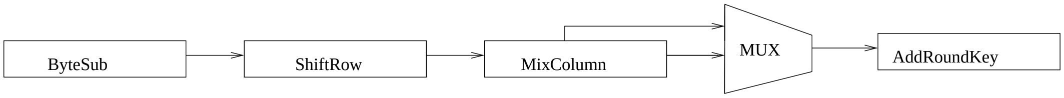

Figure 5.12. Architecture of Round Module

of 16 S-boxes working in parallel. An input byte is replaced by its corresponding value from the S-box.

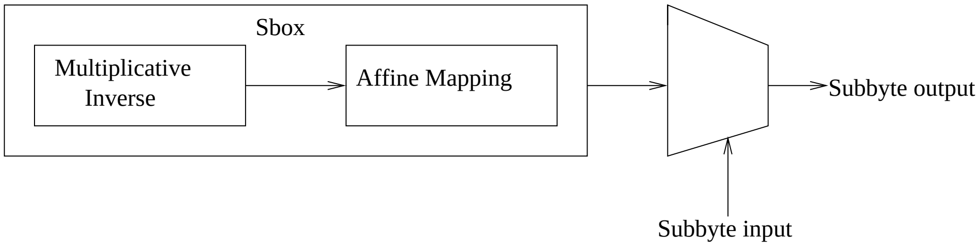

FIGURE 5.13. Architecture of ByteSub Submodule

- ShiftRow : The architecture shown in Fig. 5.14 shifts the position of bytes in the states by offsets in cycle. The first Row is unshifted. The second row is shifted to the left once, the third row is shifted to the left twice, and the fourth is shifted thrice.

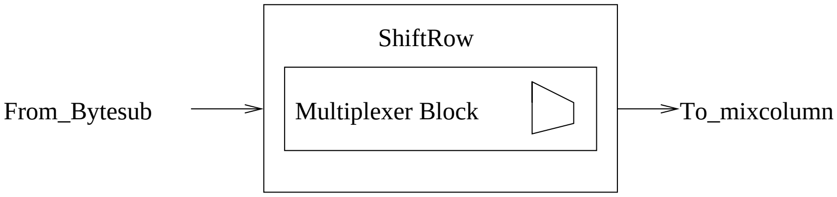

FIGURE 5.14. Architecture of ShiftRow Submodule

- MixColumn : The elements of columns in a state are considered as coefficients of polynomial over Galoid field $GF(2^8)$, where these elements are smaller than three. This polynomial are then multiplied by fixed polynomial $c(x) => (03)X^3 + (01)X^2 + (01)X + (02)$ modulo $X^4 + 1$. The operation could be carried out using matrix. The Column mixing step is basically a matrix multiplication in the galoid field, which is carried out using shift and XOR operations and the architecture is shown in Fig. 5.15.

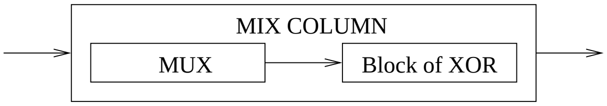

FIGURE 5.15. Top Level Architecture of Mix Column Submodule

- AddRoundkey : The round key from the key generator is XORed with block state obtained from the mixcolumn transformation as Fig. 5.16.

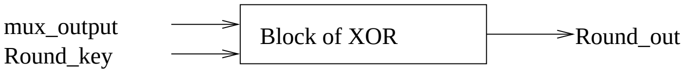

FIGURE 5.16. Architecture of AddRoundkey Submodule

5.3.2. *Key Schedule*

The round key is obtained from the initial key through key expansion using the architecture in fig. 5.17. If it is the first round in the standard round module, the multiplexer outputs the initial key for expansion. The module is able to generate subsequent round key from initial round keys through the register. As a result, round keys are generated at every round thereby reducing area requirement. The bytesub operation in the key schdule was implemented using S-Boxes. Round key computation is completed in one clock cycle.

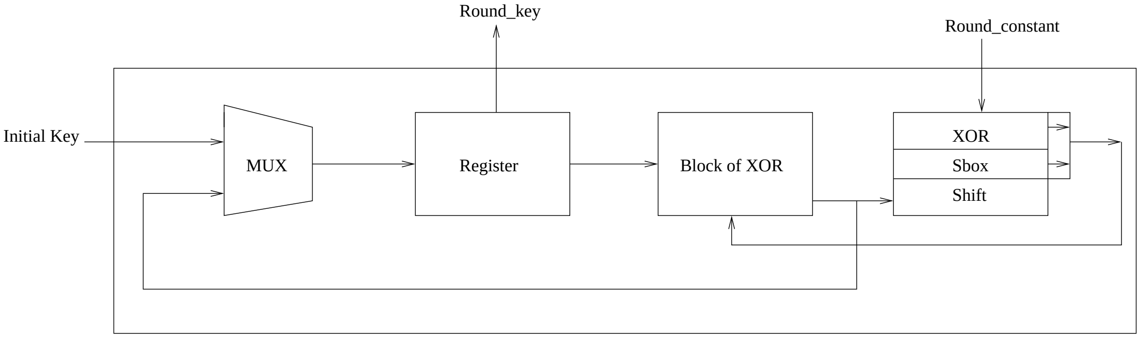

FIGURE 5.17. Architecture of Key Schedule Module

5.3.3. *Controller Module*

The sequence of operation of the system is determined by the control module. Multiplexer select inputs and register load signal are provided by the control module. The controller was implemented as a Finite State Machine(FSM) with twelve states as shown in Fig.5.18.

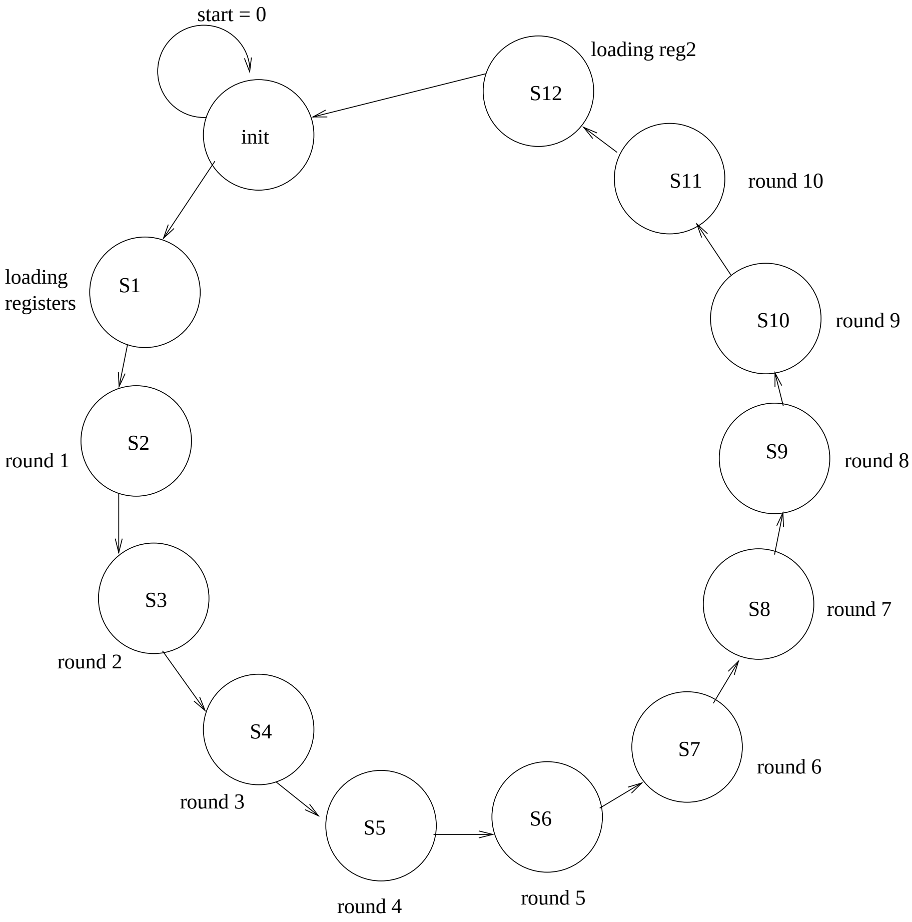

Figure 5.18. State Diagram of the Visible Watermarking Submodule

CHAPTER 6

FPGA PROTOTYPING AND SIMULATION RESULTS

In this chapter, the methodology and steps involved in the prototyping of the SDC are presented. The simulation results of the implementation of the SDC is presented. The description of the FPGA prototyping of the visible and invisible watermarking sub-modules and the encryption unit are presented in Section 6.1. In section 6.2, we present the result of our simulation.

6.1. FPGA Prototyping

For the prototyping, the architecture was modeled using VHDL and the functional simulation was carried out using Modelsim XE III 6.0a tools. The VHDL code was compiled using Xilinx ISE 8.1i Xilinx. The synthesis of the architectures were carried out using VIRTEX -II technology with xc2v500-6fg256 target device. The address generator serves the purpose of address decoder by supplying the appropriate address needed for reading from the scratch memory for watermarking and encryption. Three block RAMs is needed for our implementation; one block RAM is used as a scratch memory for storing individual image and one RAM for storing the biometric data, one RAM is used for storing data that has undergone both encryption and watermarking processes. For the implementation, input buffer was used to temporarily store data. The input buffer was also used for pipeline design so that data could be received while the processing of previous data is being carried out. To aid utilization of pipelining, we also extensively used registers in our DCT submodule. Registers were used instead of RAM to increase the performance of our system. The use of parallelism in the implementation in Fig. 5.9 of the insertion sub-module of the watermarking unit gives the capability to watermark a DCT block in two clock cycle instead of four. For the insertion unit of the invisible watermark module, we employ the concept of *resource sharing* in our implementation. This improves the performance of our system, however, there is a trade-off between the performance and the area used. A multiplication unit was shared between the DCT module and the insertion module by using two multiplexer. The RTL schematic of the visible, invisible watermarking modules and the encryption unit are shown in Fig. 6.1, Fig. 6.2 and Fig. 6.3, respectively.

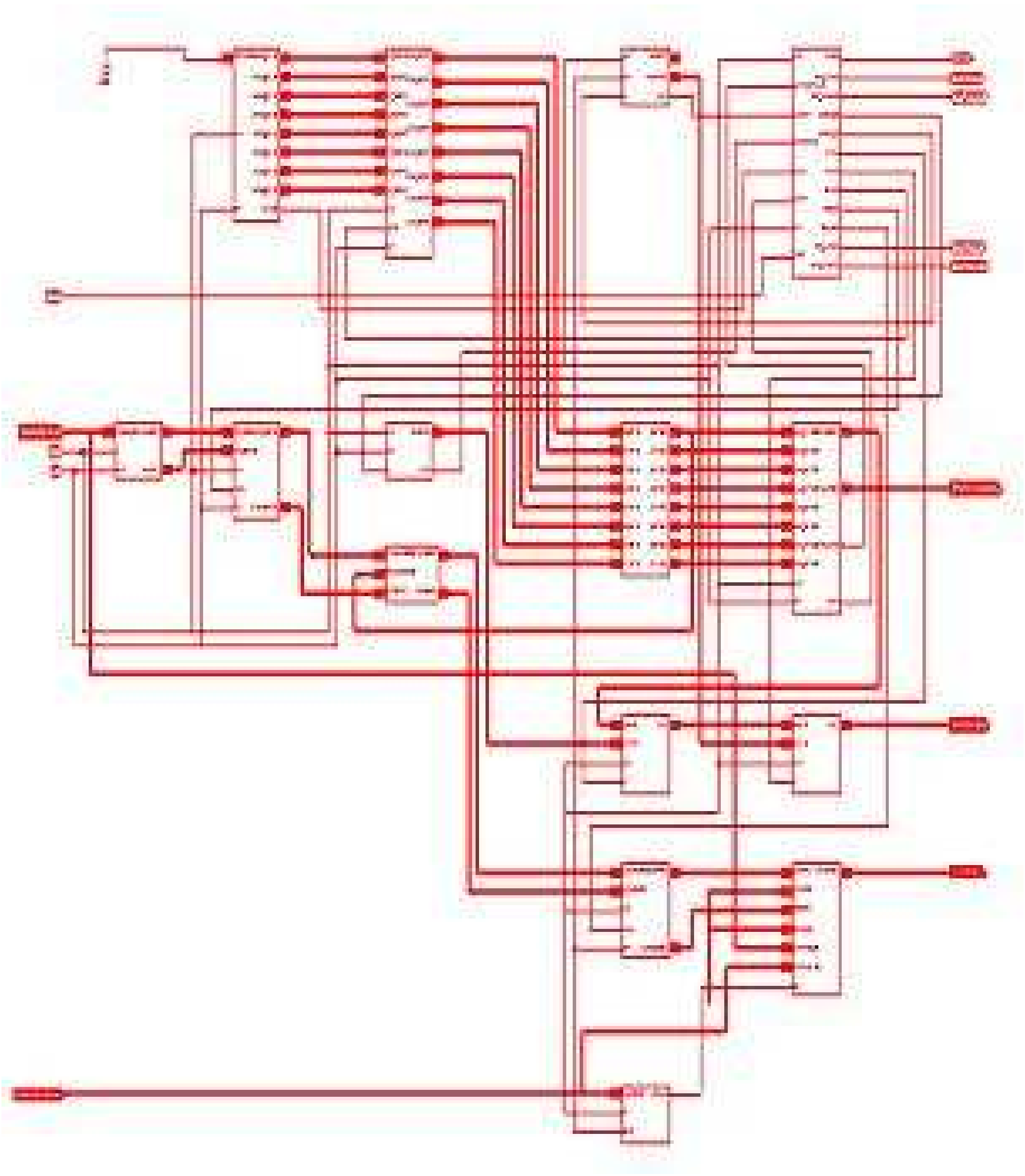

Figure 6.1. RTL Schematic of Visible Watermarking Module

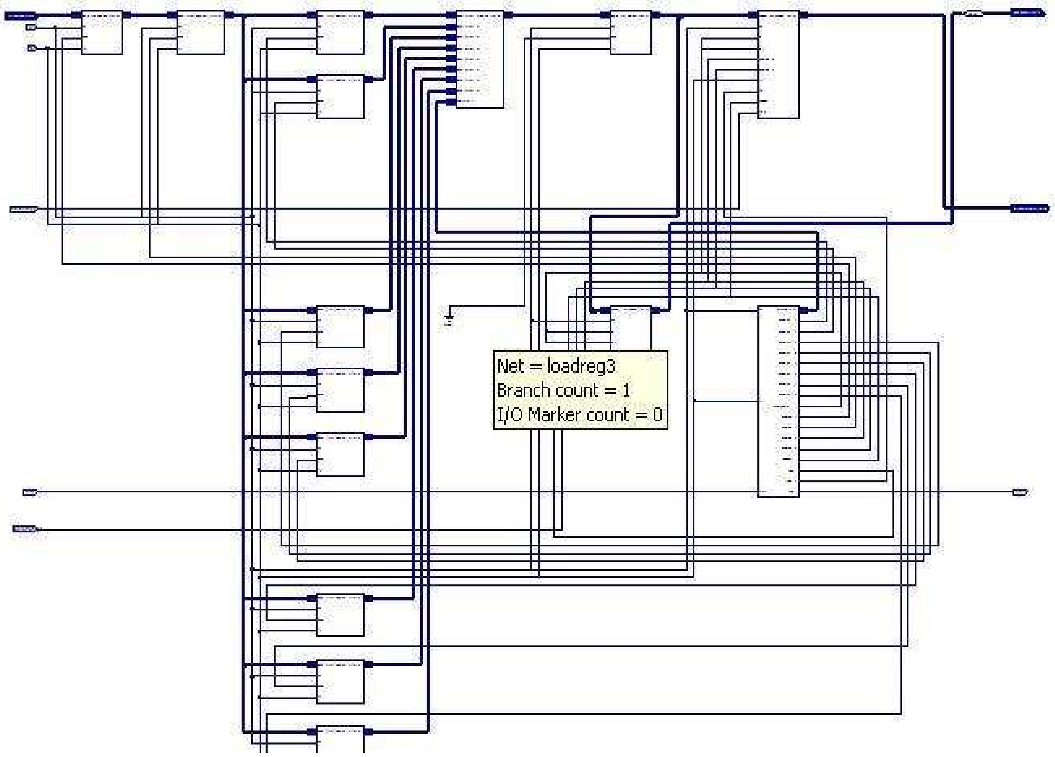

FIGURE 6.2. RTL Schematic of Invisible Watermarking Module

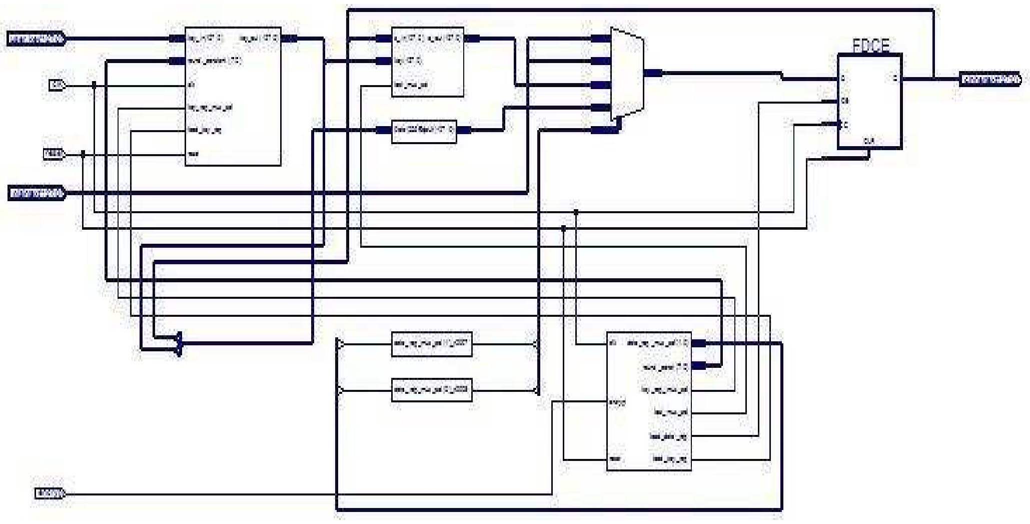

FIGURE 6.3. RTL Schematic of Encryption Unit

6.2. Simulation Results

The timing simulation result for the insertion sub-module for the visible and invisible watermarking module are shown in Fig. 6.4, and Fig. 6.5 respectively. The simulation waveform of the DCT sub-module used in the visible and the invisible watermarking module is shown in Fig. 6.6. The simulation waveform of the encryption unit is shown in Fig. 6.7. The simulation waveform of the visible and invisible watermarking unit are shown in Fig. 6.8, and Fig. 6.9. All the values obtained from simulation is the same as the expected values. The synthesis result and timing report is presented in Table 6.2. The cell usage for the different modules and submodules are also presented in Table 6.2 which is all the logical cells that are basic elements of the technology. The minimum period is the timing path from a clock to another clock in the design. The encryption unit's rate based on simulation is determined to be 0.5 Giga bits/s. Exhaustive testing of the algorithms in the camera was performed on several test images and three samples are shown in Fig. 6.10, Fig. 6.11 and Fig. 6.12 .

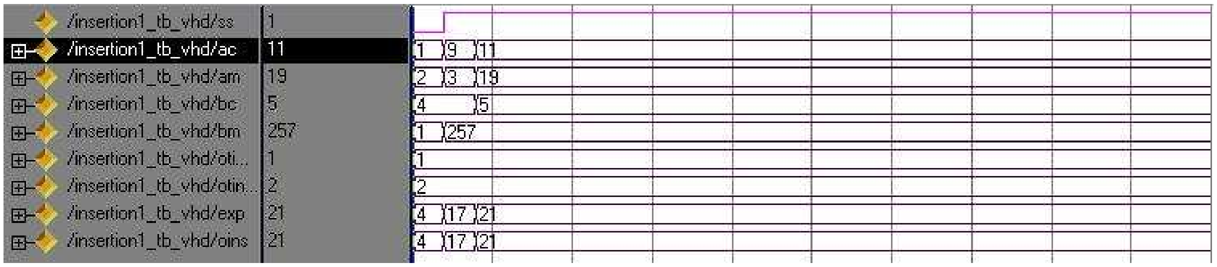
/insertion1_tb_vhd/ss 1
/insertion1_tb_vhd/ac 11
/insertion1_tb_vhd/am 19
/insertion1_tb_vhd/bc 5
/insertion1_tb_vhd/bm 257
/insertion1_tb_vhd/oti... 1
/insertion1_tb_vhd/otin... 2
/insertion1_tb_vhd/exp 21
/insertion1_tb_vhd/oins 21
1 9 11
2 3 19
4 5
1 257
1
2
4 17 21
4 17 21

Figure 6.4. Simulation Waveform of Visible Watermarking Insertion Sub-module

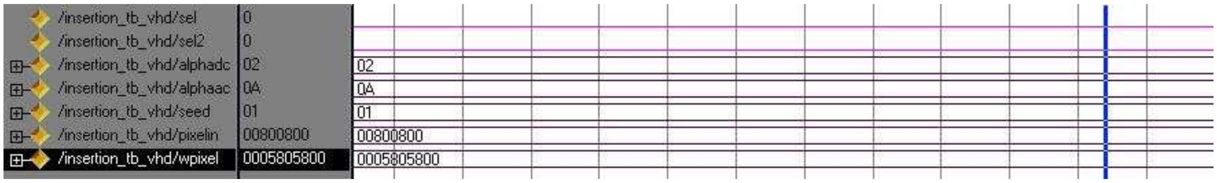
/insertion_tb_vhd/sel 0
/insertion_tb_vhd/sel2 0
/insertion_tb_vhd/alphadc 02
/insertion_tb_vhd/alphaac 0A
/insertion_tb_vhd/seed 01
/insertion_tb_vhd/pixelin 00800800
/insertion_tb_vhd/wpixel 0005805800
02
0A
01
00800800
0005805800

Figure 6.5. Simulation Waveform of Visible Watermarking Insertion Sub-module

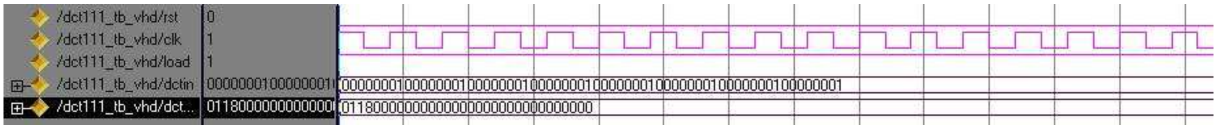

Figure 6.6. Simulation Waveform of DCT Sub-module

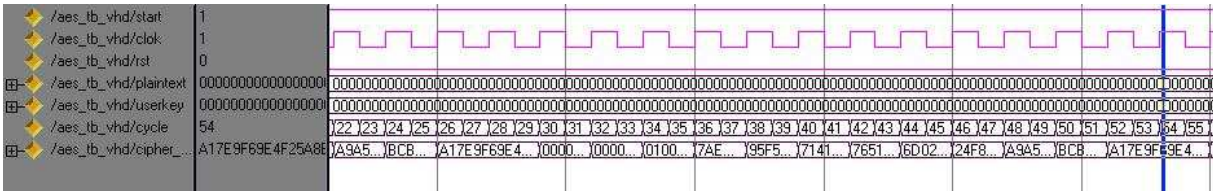
/aes_tb_vhd/start 1
/aes_tb_vhd/clok 1
/aes_tb_vhd/rst 0
/aes_tb_vhd/plaintext 0000000000000000
/aes_tb_vhd/userkey 0000000000000000
/aes_tb_vhd/cycle 54
/aes_tb_vhd/cipher_... A17E9F69E4F25A8E
22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55
A9A5... BCB... A17E9F69E4... 0000... 0000... 0100... 7AE... 95F5... 7141... 7651... 6D02... 24F8... A9A5... BCB... A17E9F69E4...

Figure 6.7. Simulation Waveform of DCT Encryption Unit

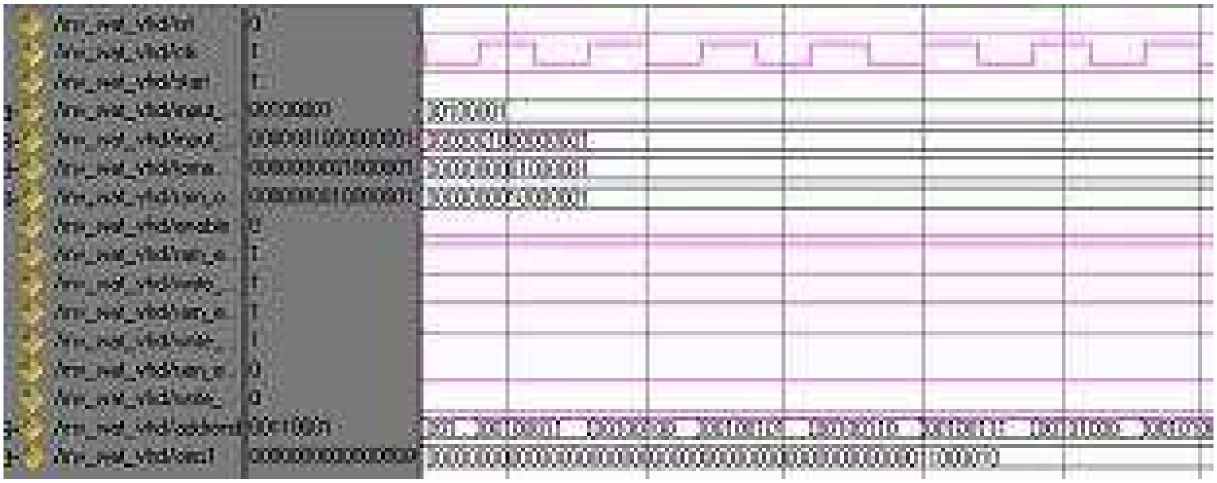

Figure 6.8. Simulation Waveform of Visible Watermarking Module

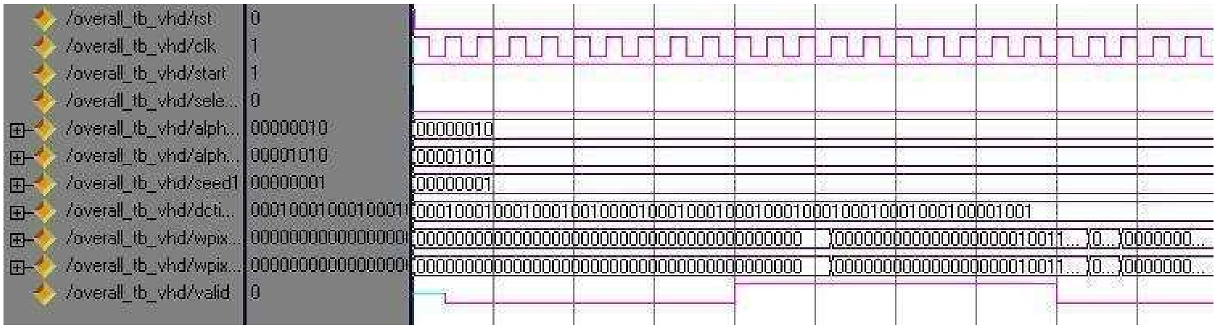

Figure 6.9. Simulation Waveform of Invisible Watermarking Chip

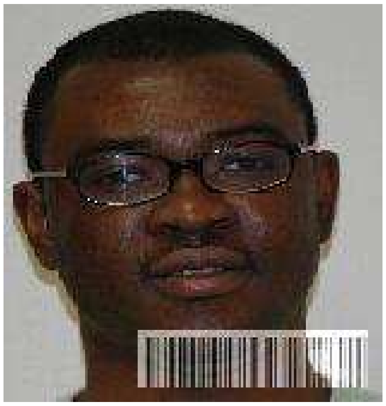

Figure 6.10. Experimental Result on Sample Image 1

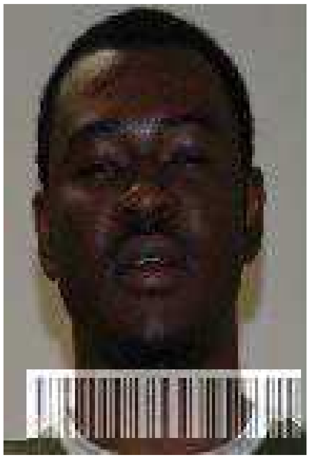

Figure 6.11. Experimental Result on Sample Image 2

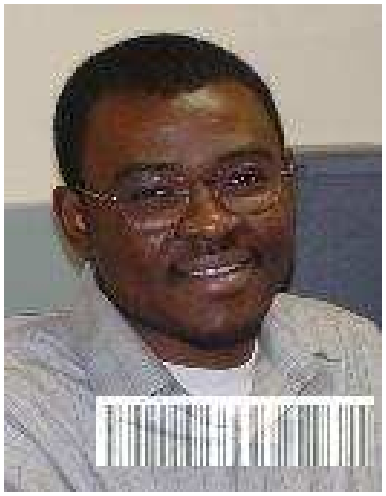

Figure 6.12. Experimental Result on Sample Image 3

Table 6.1. Encryption and Visible Watermarking Simulation Results

Parameter	Encryption Unit	Visible Watermark Module
Cells Usage (Bels)	6117	639
Maximum Frequency	217 MHz	96.318
Critical Path Delay	4.383ns	28.485ns
Minimum Time period	4.969ns	10.318ns

Table 6.2. Encryption and Invisible Watermarking Simulation Results

Parameter	Invisible Watermark Module
Cells Usage (Bels)	218
Maximum Frequency	256.148 MHz
Critical Path Delay	2.164 ns
Minimum Time Period	3.904 ns

CHAPTER 7

CONCLUSION AND FUTURE RESEARCH

A possible solution that alleviates the problems linked with the inclusion of biometric data in forms of identification such as the newly proposed electronic passport is presented. In order to address the unauthorized access and modification of data in forms of identification, a novel architecture of a secure digital camera that invisibly watermarks an encrypted biometric data into an image is presented. The top level system architecture of the secure camera was presented.

The VLSI architecture of the visible, invisible watermarking and encryption scheme that was employed in our secure digital camera was presented.

The result of the FPGA implementation of the watermarking scheme and the techniques that were employed were also presented. The result of the FPGA prototyping of the encryption unit of the SDC was presented. Our implementation of encryption unit yielded a higher throughput of 500 Mbit/s compared to other implementation in Table 7.1.

There are plans to optimize the SDC and to fabricate the first VLSI chip that carries out simultaneous encryption and watermarking processes.

Table 7.1. Comparison of Encryption Implementations

Architecture	Design Type	Throughput
Kosaraju et. al. [29]	ASIC	232 Mbit/s
Sever et. al. [50]	ASIC	2.41 Gbit/s (pipeline)
K. Gaj et. al. [47]	FPGA	414 Mbit/s
Sklavos et al. [51]	FPGA	259 Mbit/s
R. Karri et. al. [28]	FPGA	137 Mbit/s
This Thesis	FPGA	500 Mbit/s

BIBLIOGRAPHY

[1] *Advanced Encryption standard Development Effort*, http://www.nist.gov/aes.

[2] *PC World magazine*, http://www.pcworld.com/news/article/0,aid,119324,00.asp.

[3] *U.S Department of State*, http://www.state.gov/r/pa/prs/ps/2006/61538.htm.

[4] P. J. Ashenden, *The Designers Guide to VHDL*, Morgan Kaufmann Publishers, San Francisco, CA, 1995.

[5] F. Bartolini, M. Barni, A. Tefas, and I. Pitas, *Image authentication techniques for surveillance applications*, Proceedings of the IEEE, vol. 89, October 2001, p. 14031418.

[6] P. Blythe and J. fridrich, *Secure Digital camera*, Proceedings of Digital forensic Reseach Workshop (DFRWS). Linthicum, MD, August 2004, pp. 17–19.

[7] S. Brown and Z. Vranesic, *Fundamentals of Digital Logic Design with VHDL*, McGraw Hill, 2004.

[8] W. H. Chen, C. H. Smith, and S. C. Fralick, *A Fast Computational Algorithm for the Discrete Cosine Transform*, IEEE Transactions and Communications COM-25 (1977), no. 9, 1004–1009.

[9] C. Chitu, D.Chien, C. Chien, I. Verbauwhede, and F. Chang, *A hardware Implientation in FPGA of the Rijndael Algorithm*, Proceedings of 39th ACM/IEEE Design Automation Conference (Heidelberg, Newyork), June 2002, pp. 399–404.

[10] J. Daemen and V. Rijmen, *The Design of Rijndael*, Springer-Verlag Berlin, heidelberg, Newyork, 2002.

[11] D.J.Granrath, *The Role of Human Visual Models in Image Processing*, Proceedings of IEEE, vol. 69, May 1981, pp. 552–561.

[12] M. Kankanhalli et al., *Content Based Watermarking for Images*, Proc. of 6th ACM International Multimedia Conference, ACM-MM 98, September 1998, pp. 61–70.

[13] T. Ichikawa et al, *Hardware Evaluation of the AES Finalists*, Proc. 3th AES Candidate Conference, 2000.

[14] Y. Fan, L. Van, C. Huang, and H. Tsao, *Hardware-Efficient Architecture Design of Adaptive Visible Watermarking*, Proceedings of the Ninth International symposium of Consumer Electronics, June 2005, pp. 399–404.

[15] V. Fischer and M. Drutarovsky, *Two Methods of Rijndael implementation in Reconfigurable Hardware*, Proceedings of CHESS 2001, May 2001.

[16] E. Fossum, *Digital Camera System on a Chip*, IEEE Micro, 8–15.

[17] K. Gaj and P. Chodowiec, *Fast Implementation and Fair Comparision of the Final Candidates for Advanced Encryption Standard Using Field Programable Gate Arrays*, Proceedings of RSA Security Conference, April 2001.

[18] K. Gaj and P.Chodowiec, *Comparision of the Hardware Performance of the AES Candidates using Reconfigurable Hardware*, Proceedings of Third Advanced Encryption Standard (AES) Candidate Conference, April 2000.

[19] Annajirao Garimella, M. V. V. Satyanarayana, R. Satish Kumar, P. S. Murugesh, and U.C. Niranjan, *VLSI Implementation of Online Digital Watermarking Technique with Difference Encoding for 8-Bit Gray Scale Images*, Proceedings of the 16th International Conference on VLSI Design, vol. 51, April 2003, pp. 283–288.

[20] G.L.Friedman, *The Trustworthy Digital Camera:Restoring Credibility to the Photographic Image*, IEEE Transactions on Image Processing 39 (1993), no. 4, 905–910.

[21] R. C. Gonzalez and R. E woods, *Digital Image Processing*, Prentice-Hall, 2005.

[22] S. Govindarajan, I. Ouaiss, M. Kaul, V.Srinivasan, and R. Vemuri, *An Effective Design Systems for Dynamically Reconfigurable Architectures*, Proceedings of the Sixth Annual IEEE Symposium on Field-Programmable Custom Computing Machines, 1998, p. 312313.

[23] A. Hodjat and I. Verbauwhede, *Area-Throughput Trade-Offs for Fully Pipelined 30 to 70 Gbits/s AES Processors*, IEEE Transactions On Computers 55 (2006), no. 4, 569–572.

[24] Y. Hu and S. Kwong, *Wavelet domain adaptive Visible Watermarking*, Electronics Letters, vol. 37, 2001, pp. 1219–1220.

[25] D. Hwang, K. Tiri, A. Hodjat, B. Lai, S. Yang, P. Schaumont, and I. Verbauwhede, *AES-Based Security Coprocessor IC in 0.18-μm CMOS With Resistance to Differential Power Analysis Side-Channel Attacks*, IEEE Journal of Solid-State Circuits, 41 (2006), no. 4, 781–792.

[26] I.J.Cox, J.Kilian, F.T.Leighton, and T.Shamoon, *Secure Spread Spectrum Watermarking for Multimedia*, IEEE Transactions on Image Processing 6 (1997), no. 12, 1673–1687.

[27] M. Kankanhalli, *Adaptive Visible Watermarking of Images*, Proc. ICMCS99,Centro Affari, Florence, Italy, June 1999.

[28] R. Karri, K. Wu, P. Mishra, and Y. Kim, *Concurrent Error Detection Schemes for Fault-Based Side-Channel Cryptanalysis of Symmetric Block Ciphers,*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 21 (2002), no. 12.

[29] N. M. Kosaraju, M. Varanasi, and S. P. Mohanty, *A High Performance VLSI Architecture for Advanced Encryption Standard (AES) Algorithm*, Proceedings of the 19th IEEE International Conference on VLSI Design, 2006, pp. 481–484.

[30] M. Kovac and N. Raganathan, *A fully Pipeline VLSI Architecture for JPEG Image Compression Standard*, Proc. of IEEE, 1995, pp. 247–258.

[31] T. F. Lin, C. P. Su, C. T. Huang, and C. W. Wu, *A High-Throughput Low-Cost AES Cipher Chip*, Proc. IEEE Asia-Pacific Conference ASIC, 2002, pp. 85–88.

[32] H. J. Liu and H. Yao, *High-Performance VLSI Signal Processing Innovative Architectures and Algorithms*, IEEE Press, NY, USA, 1998.

[33] M. J. Loinaz, K. J. Singh, A. J. Blanksby, D. A. Inglis, K. Azadet, and B. D. Ackland, *A 200-mW, 3.3-V, CMOS Color Camera IC Producing 352 288 24-b Video at 30 Frames/s*, IEEE Journal of Solid-State Circuits, 33 (1998), no. 12, 781–792.

[34] C. Lu, *Multimedia Security*, IDEA Group Publishing, PA, USA, 2004.

[35] M. Mano and C. R. Kime, *Logic and Computer Design Fundamentals*, Pearson and Prentice Hall, NY, USA, 2004.

[36] N.J Mathai, D. Kundur, and A. Sheikoleslami, *Hardware implementation Perspectives of Digital Videao Watermarking Algorithms*, IEEE Transactions on Signal Processing (2003), 925–937.

[37] N. Memon and P. W. Wong, *Protecting Digital Media Content*, Communications of the ACM 41 (1998), no. 7, 34–43.

[38] F. Mintzer, G. Braudaway, and M. Yeung, *Effective and Ineffective Digital Watermarks*, Proceedings of the IEEE International Conference on Image Processing, vol. 3, 1997, p. 912.

[39] S. P. Mohanty, *Watermarking of Digital Images*, Master's thesis, Dept of Electrical Engineering, Indian Institute of Science, India, 1998.

[40] S. P. Mohanty, R. Kumara C., and S. Nayak, *FPGA Based Implementation of an Invisible-Robust Image Watermarking Encoder*, Lecture Notes in Computer Science (LNCS), CIT 2004,Springer-Verlag 3356 (2004), 344–353.

[41] S. P. Mohanty, K. R. Ramakrishnan, and M. S. Kanakahalli, *A DCT Domain Visible Watermarking Technique for Images*, Proceedings of IEEE International Conference on Multimedia and expo, 2000, pp. 1029–1032.

[42] S. P. Mohanty, E. Ranganathan, and R. K. Namballa, *VLSI Implementation of Visible Watermarking for Secure Digital Still Camera Design*, Proceedings of the 17th IEEE International Conference on VLSI Design (VLSID), 2004, pp. 1063–1068.

[43] S. P. 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 (TCAS-II) 53 (2006), no. 11, 394–398.

[44] S.P. Mohanty, N.Ranganathan, and R.K. Namballa, *A VLSI Architecture for Visible Watermarking in a Secure Still Digital Camera Design*, IEEE Transactions on VLSI Systems 13 (2005), no. 7, 808–818.

[45] G. R. Nelson, G. A. Jullien, and O. Y. Pecht, *CMOS Image Sensor with Watermarking Capabilities*, Proceedings of the IEEE international Symposium on Circuits and Systems(ISCAS), 2005, pp. 5326–5329.

[46] S. Okada, S.I. Okada, Y. Matsuda, T. Yamada, and A. Kobayashi, *System On A Chip For Digital Still Camera*, IEEE Transactions on Consumcr Electronics 45, no. 3, 9–12.

[47] P.Chodowiec, P. Khuon, and K. Gaj, *Fast Implementations of Secret-Key Block Ciphers Using Mixed Inner- and Outer-Round Pipelining,*, Proceedings of ACM/SIGDA int. Symposium on Field Programmable Gate Arrays, Monterey, CA, 2001.

[48] A. Satoh, S. Morioka, K. Takano, and S. Munetoh, *A compact Rijndael Hardware Architecture with S-Box Optimization*, Proc. ASIACRYPT, 2001, pp. 239–254.

[49] B. Schneier, *Applied Cryptography*, John Wiley and Sons, Inc., 1996.

[50] R. Sever, N. Ismailoglu, M. Askar, and Y. C. Tekmen, *A High Speed Asic Implementation of Rijndael Algorithm,*, IEEE International Symposium on Circuits and Systems, May 2004.

[51] N. Sklavos and O. Koufopavlou, *Architectures and VLSI Implementations of the AES-Proposal Rijndael*, IEEE Transactions on Computers, 51 (2002), no. 12, 1454–1459.

[52] L. D. 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*, IEEE Proceedings on Vision, Image and Signal Processing, vol. 147, August 2000, p. 371376.

[53] J. B. Sulistyo and D. S. Ha, *Developing Standard Cells for TSMC 0.25 μm Technology under MOSIS DEEP Rules*, Technical Report,Electrical and Computer Engineering, Virginia Tech, 2002.

[54] B .Tao and B .Dickinson, *Adaptive Watermarking in DCT Domain*, Proc. IEEE International Conference on Acoustics, ICASSP-97, vol. 4, 1997, pp. 1985–2988.

[55] A. Tefas and I. Pitas, *Robust Spatial Image Watermarking Using Progressive Detection*, Proceedings of the IEEE International Conference on Accoustics, Speech, and Signal Processing, vol. 3, 2001, pp. 1973–1976.

[56] TI, *Digital Camera System Block Diagram*, http://focus.ti.com/docs/solution/folders/print/80.html.

[57] T. H. Tsai and C. Y Lu, *A System Level Design for Embedded Watermark Technique using DSC System*, IEEE InternationalWorkshop on Intelligent Signal Processing and Communication System, 2001.

[58] I. Verbauwhede, P. Schaumont, and H. Kuo, *Design and Performance Testing of a 2.29-GB/s Rijndael Processor*, IEEE Journal of Solid-State Circuits 38 (2003), no. 3, 569–572.

[59] N. H. E. Weste and D. Harris, *CMOS VLSI Design : A Circuit and Systems Perspective*, Addison Wesley, 2005.

[60] W. Wolf, T. Lv, and I. B. Ozer, *An architectural design study for a high-speed smart camera*, Proceedings of IEEE 4th Workshop on Media and Streaming Processors, 2002.

[61] J. Wolkerstorfer, E. Oswald, and M. Lamberger, *An ASIC Implementation of the AES Sboxes*, Proceedings of the The Cryptographer's Track at the RSA Conference on Topics in Cryptology, 2002, pp. 67–78.

[62] X. Zhang and K. K. Parhi, *High-Speed VLSI Architectures for the AES Algorithm*, IEEE Transactions on Very Large Scale Integration(VLSI) Systems 12 (2004), no. 9, 957–967.