VLSI Architecture and FPGA Prototyping of a Digital Camera for Image Security and Authentication

Oluwayomi Adamo oba0002@unt.edu
Saraju P. Mohanty smohanty@cs.unt.edu
Elias Kougianos eliask@unt.edu
Murali Varanasi varanasi@unt.edu
Wei Cai wc0072@unt.edu
VLSI Design and CAD Laboratory (http://www.vdcl.cse.unt.edu)
P.O Box 311366, University of North Texas, Denton, TX 76203

Abstract: *Two fundamental operations performed by a digital camera are image capturing and storing. The images are subsequently transmitted in various forms over appropriate media. These images are always vulnerable to various forms of copyright attacks and ownership issues. This paper introduces a digital camera with built-in copyright protection and security mechanism for images produced by it. Since the proposal of the trustworthy digital camera by Friedman [1], significant research has been done in developing algorithms for watermarking and encryption with the aim of using them in digital cameras. However, only few of these efforts are involved with the architectural development of the entire digital camera. Incorporation of encryption and watermarking together in the digital camera will assist in protecting and authenticating image files. In this paper, we present an architecture and a hardware efficient FPGA based watermark module towards the development of the complete digital camera.*

1. Introduction

Watermarking is the process whereby a multimedia object is embedded with data, which could be a label, tag or watermark for the purpose of protecting copyrights. On the other hand, cryptography is the process of encrypting and decrypting a message that could be in the form of text, for the purpose of authenticating such a message. Digital watermarking can be divided into four categories: visible, invisible robust, invisible-fragile, and dual [4], [5]. Rapid advancements in the digital computing world have made manipulation and proliferation of digital media relatively easy and common. Today, every picture appearing in newspapers and magazines has been digitally altered to some degree [1]. This brings about the need to secure, copyright protect and authenticate digital media. Therefore, our system comprising of cryptography and watermarking will enhance security and authentication of digital images. Cryptography will be used to make the image more secure while watermarking will be used mainly for authenticating the image. The difference between watermarking and encryption is that watermarking does not restrict access to the data but encryption does. However, a digital watermark is intended to complement cryptography [2]. As a result, encryption will be used to safeguard the host image in providing a two layers of protection.

2. Digital Still Camera Overview

A block diagram of the proposed digital watermark is shown in Fig. 1.

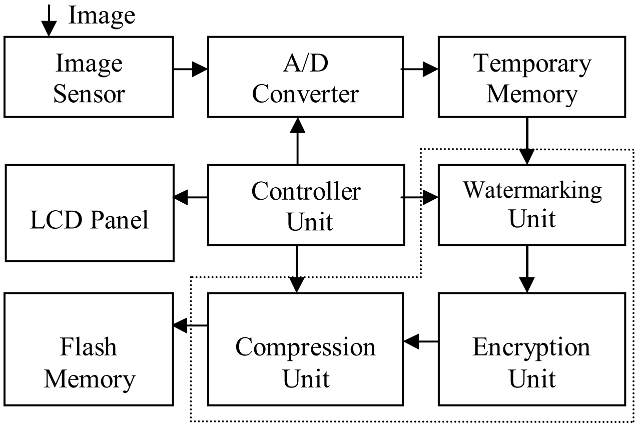


Fig. 1. Secure digital camera for image security and authentication

In the proposed digital camera, the image is captured by an image sensor and converted to a digital signal by the A/D converter. A CMOS image sensor that has an embedded A/D converter will be used. The captured image is stored temporarily in the scratch memory, after which it is displayed on the 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, which 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. We will use both the invisible-robust [2] and visible [3] watermarking algorithms along with encryption [4] and data compression [5]. 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 [6]. We will focus on the structural aspects of the controller, watermarking, encryption, and the JPEG units, to develop the prototype the complete camera. We discuss the implementation of the watermarking unit in this paper and the rest of the units of the camera design are being carried out in our on-going research which will be presented in subsequent publications.

3. Selected Watermarking Algorithm

In this section, we describe the Discrete Cosine Transform (DCT) based visible watermark algorithm .The visible watermarking algorithm proposed in [7] was chosen for implementation towards the development of the secure digital camera. The insertion algorithm is based on the sensitivity of the human visual system. The original image *I*, and the watermark image W are divided into 8x8 blocks. The DCT coefficient for both *I*, and *W*'s 8x8 blocks are calculated using the DCT module. The DCT coefficients of *n*-th block is represented by $c_{ij}(n)$, where *n* denotes the position of block in image *I*. The mean gray value for each of the blocks of the original image is calculated using Eqn. 1, where c_{00} is the DC coefficient of block *n*.

$$\mu_n = c_{00}(n). \qquad (1)$$

The normalized mean gray value of block n is calculated using Eqn. 2 as follows with where $c_{00\max}$ is the maximum value of $c_{00}(n)$:

$$\mu'_n = c_{00}(n)/c_{00\max} . \qquad (2)$$

The normalized mean gray value of the image *I* is calculated using:

$$\mu' = (1/N)\sum_{n=1}^{N} c_{00}(n), \qquad (3)$$

where *N* is the total number of 8x8 blocks in the image *I*. The variance of AC-DCT coefficients is calculated using Eqn. 4:

$$\sigma_n = (1/64)\sum_i \sum \left(c_{ij} - \mu_{nAC}\right)^2 , \qquad (4)$$

where μ_{nAC} denotes the AC DCT coefficients.

If α_n and β_n are the scaling factors, then the DCT coefficient of original image (c_{ij}) and DCT coefficient of watermark image (w_{ij}) are merged block-wise to obtain the watermarked image as follows:

$$c'_{ij}(n) = \alpha_n c_{ij} + \beta_n w_{ij}(n) \quad n = 1, 2.... \qquad (5)$$

The scaling factors for each block are computed using:

$$\alpha_n = \sigma'_n \exp.(-\mu'_n - \mu')^2)$$
$$\beta_n = (1/\sigma'_n)(1 - \exp.(-(\mu'_n - \mu_n)^2)) \qquad (6)$$

4. Architecture of the Watermarking unit

The architecture of the visible watermark algorithm will be discussed in this section. The block diagram of the architecture of the watermarking unit is shown in Fig. 2.

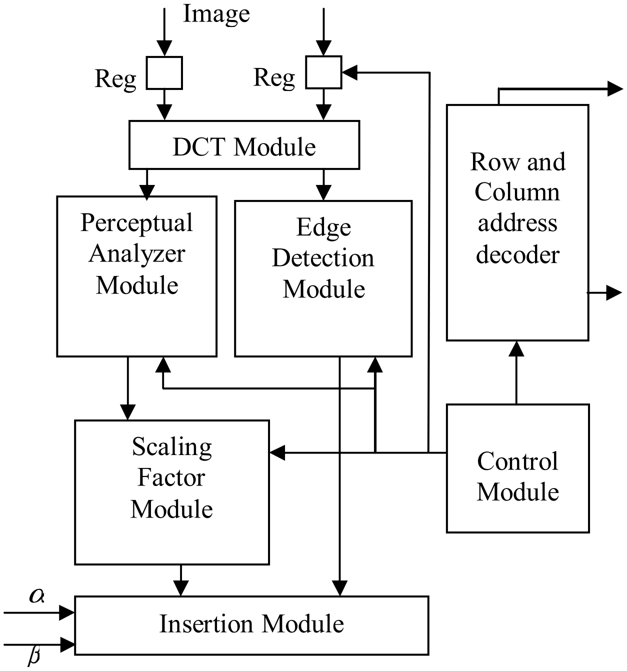


Fig. 2. Architecture of watermarking unit

The watermarking unit is composed of several modules, 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.

4.1 DCT Module

The DCT module calculates the DCT coefficient of the host image and it consists of 2-1D DCT sub-module. The algorithm from [8] was used for our implementation. For the implementation of this module, sixteen multipliers and twelve adders were used. The 1D row DCT of each 8x8 block was first computed. The column DCT of each block is then carried out. A buffer was used to assist in finding the transpose of the 1D row DCT. The final controller for the watermarking unit controls the DCT module. The buffer stores the 1D row DCT coefficient before the column DCT is computed. The block diagram of the DCT module is shown in Fig. 3.

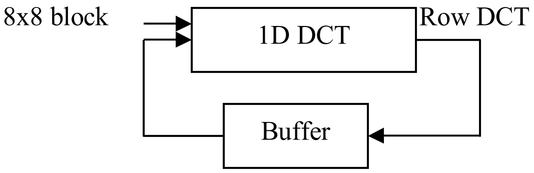


Fig. 3. DCT module

4.2 Perceptual Analyzer Module

The perceptual analyzer module architecture that evaluates Eqn. 2 and Eqn. 4 is presented in Fig. 4. The perceptual analyzer is made of three sub-modules. The mean calculator is the first sub module and it is used to compute the mean of the AC-DCT coefficients. The second sub module called the variance calculator module is used to calculate the variance in the AC-DCT coefficients. The DC-DCT mean calculator is the third sub module and is used to calculate the DC mean. The c_{ij}'s represent the DCT coefficients of the host image and *N* is the number of 8x8 DCT blocks.

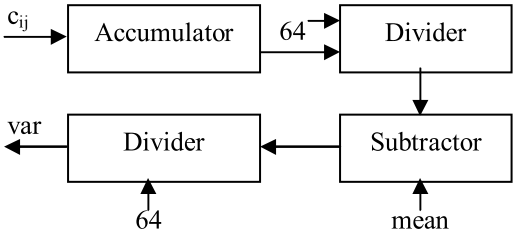


Fig. 4. Perceptual analyzer Module

4.3 Edge Detection Module

The edge blocks in the original image are determined using the edge detection module. 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 as shown Fig. 5. The absolute values of 8x8 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.

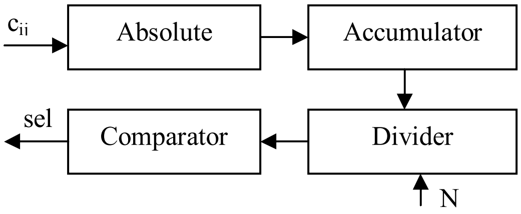


Fig. 5. Edge detection module

4.4 Scaling Factor Module

This module computes the scaling factors using Eqn. 6. A Taylor series approximation is used to evaluate the equation. The values obtained from the evaluation are later scaled to a specific range by the scaling module. The block diagram of the scaling factor module is shown in Fig. 6.

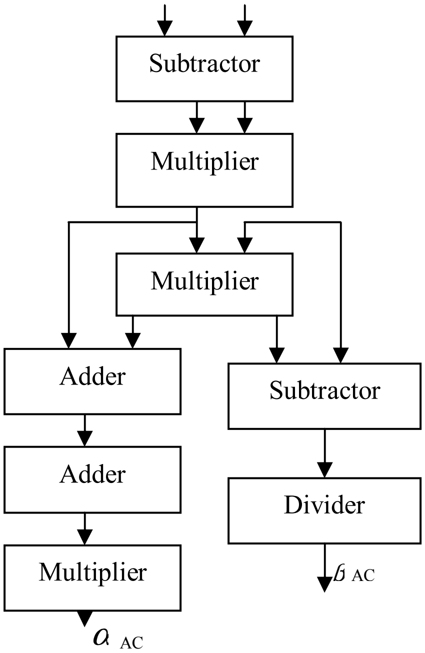


Fig. 6. Scaling factor module

4.5 Insertion Module

The insertion module (Fig. 7) serves the purpose of inserting the watermark into the original image. 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. 5. The architecture for the insertion module is shown below.

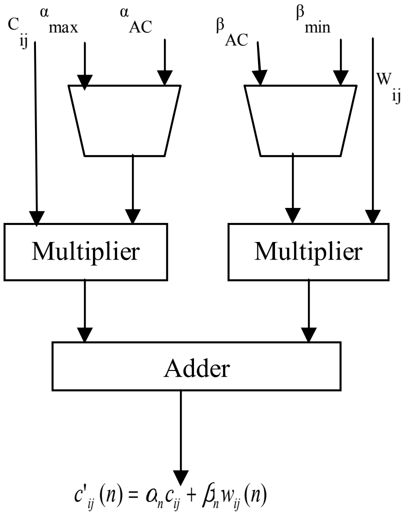


Fig. 7. Insertion module

4.7. Control Unit

The controller has 7 states, Init, S0, S1, S2, S3, S4 and S5 as shown in Fig. 8. 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.

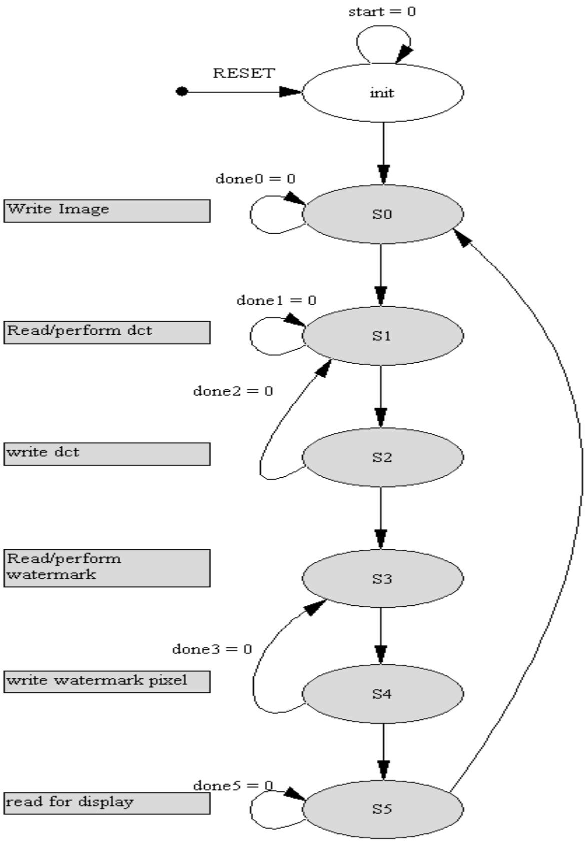


Fig. 8. Finite state machine of the controller

5. FPGA Based Simulation and Prototyping

The chip 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. The synthesis of the chip was carried out using VIRTEX–II technology with xc2v500-6fg256 target device. The RTL schematic that was obtained with the aid of the ISE 8.1i is shown in Fig. 9, and the timing simulation that was obtained with the aid of Modelsim is shown in Fig. 8. The synthesis result and timing report is also presented in Table 1. The cell usage is also shown in Table 1 and it represents 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.

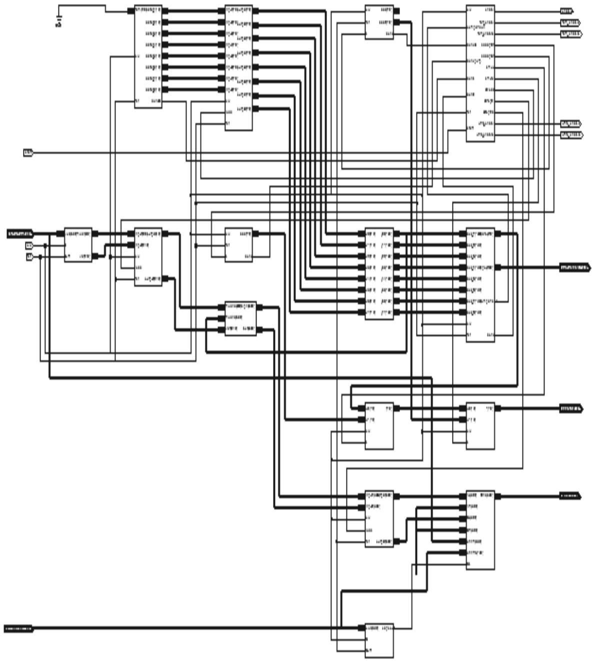
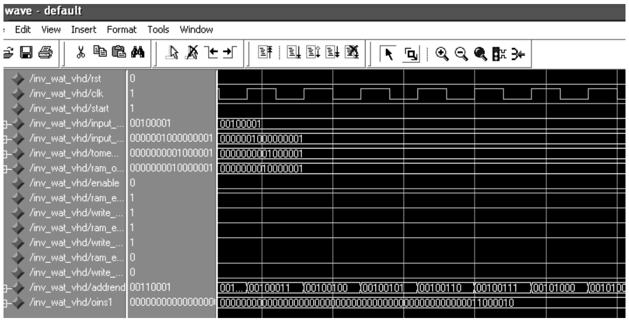


Fig. 9. RTL schematic generated by synthesis

TABLE 1
Synthesis Summary

Minimum period	10.318 ns
Maximum frequency	96.318
Total BELS	639
N0 I/O	128
Scaling factor path delay	10.318 ns
Edge detection path delay	3.618 ns
Insertion path delay	14.549



```
/inv_wat_vhd/rst        0
/inv_wat_vhd/clk        1
/inv_wat_vhd/start      1
/inv_wat_vhd/input_...  00100001          00100001
/inv_wat_vhd/input_...  0000001000000001  0000001000000001
/inv_wat_vhd/tome...    0000000001000001  0000000001000001
/inv_wat_vhd/ram_o...   0000000010000001  0000000010000001
/inv_wat_vhd/enable     0
/inv_wat_vhd/ram_e...   1
/inv_wat_vhd/write_...  1
/inv_wat_vhd/ram_e...   1
/inv_wat_vhd/write_...  1
/inv_wat_vhd/ram_e...   0
/inv_wat_vhd/write_...  0
/inv_wat_vhd/addrend    00110001          001... 00100011 00100100 00100101 00100110 00100111 00101000 0010100
/inv_wat_vhd/oins1      0000000000000000  000000000000000000000000000000000000000000000000000011000010
```

Fig. 8. Simulation waveform

6. Conclusions and Future research

This paper introduced the concept of a digital camera for image security and authentication. The architecture and the FPGA implementation of the watermark unit were presented towards the implementation of the digital camera. The design and implementation of the remaining components of the camera is being conducted as on-going research in our laboratory. We plan to implement the AES for the cryptosystem, to be incorporated with the watermark unit.

References

[1] G. L. Friedman, “The Trustworthy Digital Camera: Restoring Credibility to the Photographic Image,” *IEEE Transactions on Image Processing.* Vol. 39, no. 4 pp. 905-910, Nov. 1993

[2] I. J. Cox, J. Kilian, F.T. Leighton, and T. Shamoon, “Secure Spread Spectrum Watermarking for Multimedia,” *IEEE Transactions on Image Processing*, vol. 6, no. 12, pp. 1673-1687, Dec. 1997.

[3] S. P. Mohanty, N. Ranganathan, and R. K. Namballa, "A VLSI Architecture for Visible Watermarking in a Secure Still Digital Camera (S^2DC) Design", *IEEE Transactions on VLSI Systems, Vol. 13, No. 7, July 2005, pp. 808-818.*

[4] S. P. Mohanty, “Watermarking of Digital Images,” M. S. Thesis, Dept. of Electrical Engineering, Indian Institute of Science, India, 1999.

[5] N. Memon and P. W. Wong, “Protecting Digital Media Content,” *Communications of the ACM*, vol. 41, no. 7, pp. 34–43, July 1998.

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

[7] S. P. Mohanty, K. R. Ramakrishnan and M. S. Kanakanhalli, "A DCT Domain Visible Watermarking Technique for Images", in *Proceedings of the IEEE International Conference on Multimedia and Expo*, pp. 1029-1032, 2000.

[8] W. H. Chen, C. H. Smith, and S. C. Fralick, “A Fast Computational Algorithm for the Discrete Cosine Transform,” *IEEE Transactions on Communications,* Vol. COM-25, No. 9, pp. 1004-1009, Sep. 1977.