FPGA Hardware Based Implementation of an Image Watermarking System

Rohollah Mazrae Khoshki[1], Sami Oweis[2], Shumei Wang[3], George Pappas[4], Subramaniam Ganesan[5]

Electrical and Computer Engineering Department, Oakland University, Rochester, MI, USA[1,2,4,5]

Computer Science and Technology Department, Jiangsu Normal University, Xuzhou 2216 P.R.China[3]

Abstract: The objective of this paper is to implement an algorithm of the digital water marking technique applied on hardware platform, to perform faster processing and provide portability. Several softwares based encryption and watermarking techniques exist and are used currently in the market. However, hardware implementations are not commonly used due to their complexity and cost. The design used in this paper implements both watermarking and encryption techniques on A Field Programmable Gate Array (FPGA) Altera hardware and Quartus II software, which has been proven to be reliable and robust. The development of this system utilized MATLAB and SIMULINK for the simulation environment and ALTERA DSP Builder integrated with Simulink Embedded coder for the Auto-Code generation.

Keywords: FPGA, Watermarking Algorithm, DCT, DSP, Embedded Coder, Code generation, SIMULINK, MATLAB, ALTERA, Quartus, DSP Builder

I. INTRODUCTION

The digital watermarking is a method used to hide or embed unique information into a digital image that can be uniquely identified later for its content and the authenticity [1, 2]. Digital image is an image that can be constructed and rendered by the use of electronic device such as a computer. Watermarking key or data is the information being added or embedded into the original image to be used for identification purpose. Encryption is the method of altering the data to prevent any unauthorized interception and modification while transmission. Therefore, both watermarking and encryption are needed [3].

The digital watermarking process consists of a watermark encoder and decoder [4, 5]. The encoder adds or embeds watermark onto the host object and the decoder reconstructs and identifies the content and authenticity of the original digital image.

This process is done by using a unique key. The key is private and known to only authorized parties, eliminating the possibility of illegal usage of the digital content.

Watermarking can be applied in the spatial domain or the Frequency domain. The Discrete Cosine Transformation (DCT) theory was used by us in the water marking technique.

The hardware based watermarking system was designed on a (FPGA) board.

Figure 1 illustrates the flow of water marking process.

A. Types of watermarking

There are mainly three types of watermarking including visible watermarking, invisible watermarking and dual watermarking.

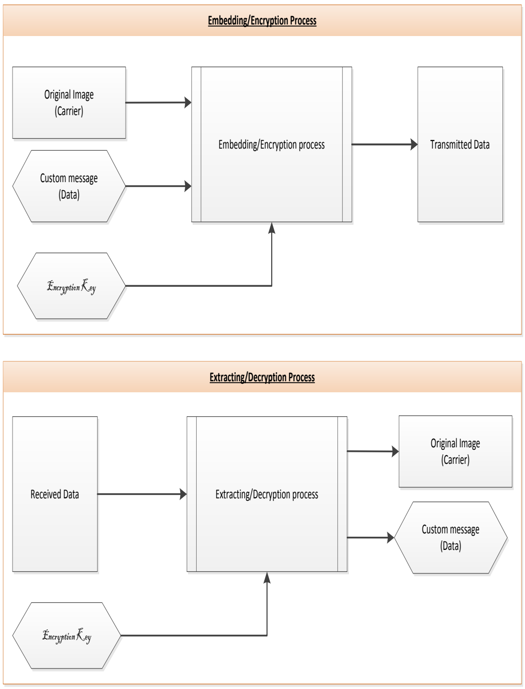


Fig. 1. A Block diagram of Image watermarking embedding and extraction

Visible watermarking is the type that used in art declaration (signing ownership). Invisible watermarking is the type used in exchange information such as secret messages during war. Dual watermarking is the type used in email communications where there is a public and private key.

II. EMBEDDING PROCESS

A. Theory of DCT in image watermarking

A discrete cosine transform (DCT) is defined as the summation of cosines function oscillation at different frequencies and it's closely related to the discrete Fourier transform (DFT). DCT is being used in many engineering applications, such as image and audio compressions, where the use of cosine rather than sine functions is critical in these applications. Cosine functions are more efficient for compression of images because fewer functions are needed [6].

Below are the equations describing a two dimensional DCT with input an image A and output image B.

$$B_{pq} = \alpha_p \alpha_q \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} A_{mn} \cos\left[\frac{\pi(2m+1)p}{2M}\right]\cos\left[\frac{\pi(2n+1)q}{2N}\right] \qquad (1)$$

$$0 \le p \le M-1$$
$$0 \le q \le N-1$$

Where

$$\alpha_p = \begin{Bmatrix} \frac{1}{\sqrt{M}}, & p=0 \\ \sqrt{\frac{2}{M}}, & 1 \le p \le M\text{-}1 \end{Bmatrix}, \text{And} \qquad \alpha_q = \begin{Bmatrix} \frac{1}{\sqrt{N}}, & q=0 \\ \sqrt{\frac{2}{N}}, & 1 \le q \le N\text{-}1 \end{Bmatrix}$$

M and N are the row and column size of A, respectively. If you apply the DCT to real data, the result is also real. The DCT tends to concentrate information, making it useful for image compression applications [7].

A_{mn} is the (m, n^{th}) element of the image represented by the matrix A. N is the size of the block that the DCT is done on. The equation calculates one entry (m, n^{th}) of the transformed image from the pixel values of the original image matrix. For the standard 8x8 block that JPEG compression uses, N equals 8 and m and n range from 0 to 7.

Therefore $B(i, j)$ would be as in Equation (2).

$$B_{pq} = \alpha_p \alpha_q \sum_{m=0}^{7}\sum_{n=0}^{7} A_{mn} \cos\left[\frac{\pi(2m+1)p}{2(8)}\right]\cos\left[\frac{\pi(2n+1)q}{2(8)}\right] \qquad (2)$$

$$0 \le p \le 7$$
$$0 \le q \le 7$$

Where

$$\alpha_p = \begin{Bmatrix} \frac{1}{\sqrt{8}}, & p=0 \\ \sqrt{\frac{2}{8}}, & 1 \le p \le 7 \end{Bmatrix}, \text{And} \qquad \alpha_q = \begin{Bmatrix} \frac{1}{\sqrt{8}}, & q=0 \\ \sqrt{\frac{2}{8}}, & 1 \le q \le 7 \end{Bmatrix}$$

Because the DCT uses cosine functions, the resulting matrix depends on the horizontal, diagonal, and vertical frequencies.

B. IDCT Theory (inverse Discrete Cosine Transformation)

The inverse discrete cosine transform (IDCT) is a technique for converting a signal block to its original state and its elementary frequency components intact. Here we utilized some simple functions to compute the IDCT and to extract the compressed images.

Below are the equations describing a two dimensional DCT with input an image A and output image B.

Below are the equations that are used for two dimensional IDCT with input an image B and output image A (original).

$$A_{mn} = \sum_{p=0}^{M-1}\sum_{q=0}^{N-1} \alpha_p \alpha_q B_{pq} \cos\left[\frac{\pi(2m+1)p}{2M}\right]\cos\left[\frac{\pi(2n+1)q}{2N}\right] \qquad (3)$$

$$0 \le p \le M-1$$
$$0 \le q \le N-1$$

Where

$$\alpha_p = \begin{Bmatrix} \frac{1}{\sqrt{M}}, & p=0 \\ \sqrt{\frac{2}{M}}, & 1 \le p \le M\text{-}1 \end{Bmatrix}, \text{And} \qquad \alpha_q = \begin{Bmatrix} \frac{1}{\sqrt{N}}, & q=0 \\ \sqrt{\frac{2}{N}}, & 1 \le q \le N\text{-}1 \end{Bmatrix}$$

B_{mn} is the (m, n^{th}) element of the output image represented by the matrix B. N is the size of the block that the IDCT is done on. The equation calculates one entry (m, n^{th}) of the transformed image from the pixel values of the original image matrix. For the standard 8x8 block that JPEG compression uses, N equals 8 and m and n range from 0 to 7. Therefore $A(i, j)$ would be as in Equation (2).

$$A_{mn} = \sum_{p=0}^{7}\sum_{q=0}^{7} \alpha_p \alpha_q B_{pq} \cos\left[\frac{\pi(2m+1)p}{2(8)}\right]\cos\left[\frac{\pi(2n+1)q}{2(8)}\right] \qquad (4)$$

$$0 \le p \le 7$$
$$0 \le q \le 7$$

Where

$$\alpha_p = \begin{Bmatrix} \frac{1}{\sqrt{8}}, & p=0 \\ \sqrt{\frac{2}{8}}, & 1 \le p \le 7 \end{Bmatrix}, \text{And} \qquad \alpha_q = \begin{Bmatrix} \frac{1}{\sqrt{8}}, & q=0 \\ \sqrt{\frac{2}{8}}, & 1 \le q \le 7 \end{Bmatrix}$$

C. Theory of encryption message process

In this project we could define three main steps to create and extract message in the host image.

Step 1: Initializing, we take host Image and then divided that to non-overlaying blocks partition In the technique suggested in the Ni method (Ni et al., 2006) [8], the host image with size M×N is divided into a set of m ×n blocks where m and n are even, in this project we choose m=n=8 as block size.

Step2: create the watermarked image according to message data, we take the message image and convert it to black and white image and reshape it to one dimension vector that need to be coded in host image, According to message data(zero or one) we compare two specific elements of DCT block and then swap them. It should be mentioned that the difference between two specific values must be less than threshold K that we defined in the code. In otherwise we adjust the two specific values that the difference is less that threshold by adding and subtracting K/2 values appropriate to DCT elements values. Finally create the watermarked image by applying new blocks according to message data to watermarked image.

Step 3: extracting message data from host image, inversely in the receiver part for decoding message from watermarked image, we apply IDCT to the blocks and by comparing two specific values of blocks we reached the message data (black and white) form watermarked image. Below is an example of the transformation of the 8x8 block from the actual image used in the simulation. S is the representation of the host image, T is the representation of the watermarked image and Z is the recovered image.

Step 1

$$S = \begin{bmatrix} 162 & 162 & 162 & 161 & 161 & 159 & 162 & 163 \\ 162 & 162 & 162 & 161 & 161 & 159 & 162 & 163 \\ 162 & 162 & 162 & 161 & 161 & 159 & 162 & 163 \\ 162 & 162 & 162 & 161 & 161 & 159 & 162 & 163 \\ 163 & 163 & 161 & 160 & 160 & 159 & 161 & 162 \\ 163 & 163 & 159 & 157 & 160 & 160 & 159 & 159 \\ 160 & 160 & 160 & 158 & 159 & 160 & 157 & 158 \\ 157 & 157 & 157 & 157 & 159 & 158 & 157 & 159 \end{bmatrix}$$

Step 2

$$T \quad = \quad 1.0e+03 \; *$$

$$\begin{bmatrix} 1.2837 & 0.0025 & 0.0048 & -0.0017 & 0.0010 & -0.0019 & -0.0009 & 0.0020 \\ 0.0096 & -0.0008 & 0.0016 & -0.0035 & 0.0008 & 0.0020 & -0.0017 & 0.0012 \\ -0.0051 & -0.0017 & -0.0021 & 0.0001 & 0.0003 & -0.0003 & 0.0010 & -0.0001 \\ 0.0015 & 0.0027 & 0.0015 & 0.0019 & -0.0011 & -0.0007 & -0.0001 & -0.0004 \\ -0.0002 & -0.0017 & 0.0000 & -0.0014 & 0.0010 & 0.0005 & -0.0006 & 0.0002 \\ 0.0004 & 0.0005 & -0.0010 & -0.0002 & -0.0005 & 0.0002 & 0.0008 & -0.0000 \\ -0.0006 & -0.0002 & 0.0008 & 0.0010 & 0.0001 & -0.0006 & -0.0007 & 0.0002 \\ 0.0003 & 0.0003 & -0.0002 & -0.0008 & -0.0000 & 0.0004 & 0.0003 & -0.0003 \end{bmatrix}$$

Step 3

$$Z = \begin{bmatrix} 162.0000 & 162.0000 & 162.0000 & 161.0000 & 161.0000 & 159.0000 & 162.0000 & 163.0000 \\ 162.0000 & 162.0000 & 162.0000 & 161.0000 & 161.0000 & 159.0000 & 162.0000 & 163.0000 \\ 162.0000 & 162.0000 & 162.0000 & 161.0000 & 161.0000 & 159.0000 & 162.0000 & 163.0000 \\ 162.0000 & 162.0000 & 162.0000 & 161.0000 & 161.0000 & 159.0000 & 162.0000 & 163.0000 \\ 163.0000 & 163.0000 & 161.0000 & 160.0000 & 160.0000 & 159.0000 & 161.0000 & 162.0000 \\ 163.0000 & 163.0000 & 159.0000 & 157.0000 & 160.0000 & 160.0000 & 159.0000 & 159.0000 \\ 160.0000 & 160.0000 & 160.0000 & 158.0000 & 159.0000 & 160.0000 & 157.0000 & 158.0000 \\ 157.0000 & 157.0000 & 157.0000 & 157.0000 & 159.0000 & 158.0000 & 157.0000 & 159.0000 \end{bmatrix}$$

III. MATLAB IMPLEMENTATION OF ENCRYPTION MESSAGE IN TRANSMITTER

We have implemented the encryption/embedding a message/data using the process that is explained in section 2.3 in MATLAB(2012a) Simulink.

The figures below show the watermarking process which is performed by using three steps:

Step 1: Initialize and preparation of the Host image and message image (Figure 2)

In the initializing process both of the host (cover) and the message images were read and loaded to the workspace, the block size was defined as [8, 8], and based on the size of the cover image [M, N] a message pad vector with the size of [M*N, 1] was created, then the message pad was filled with the reshaped data from the host image.

There was an enable status added to the Simulink block that allows this process to run only once, and then it was setup to enable the second processing block.

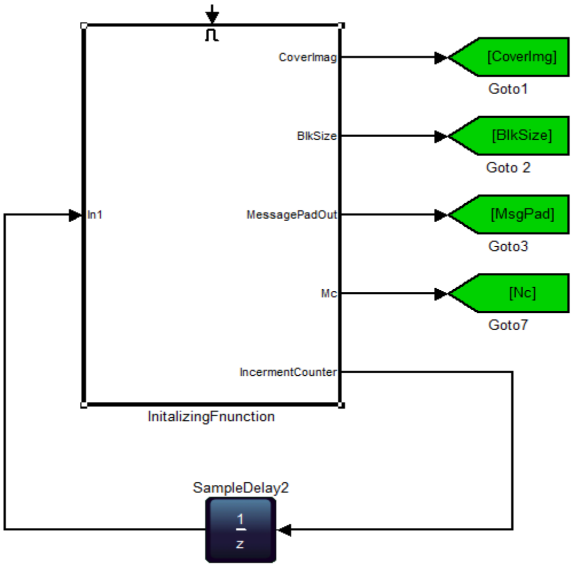


Fig. 2. Initialize Process

Step 2: Embedding the message into the host image are shown in Figure 3, and Figure 3.1.

Once this step was enabled, a counter was used to stop the process from overshooting the processing time. While in this process and at every loop, a block size [8, 8] from the host image was watermarked with one pixel of the message data, using the DCT technique. According to the value of message data we change the determined position of two elements of DCT block size [8, 8] and create the watermarked image blocks. At the end of process we have embedded data message to host image.

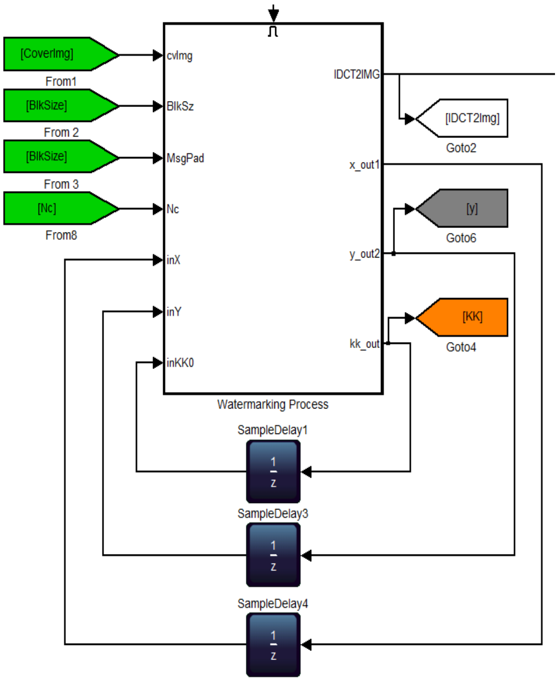


Fig. 3. Embedding Process

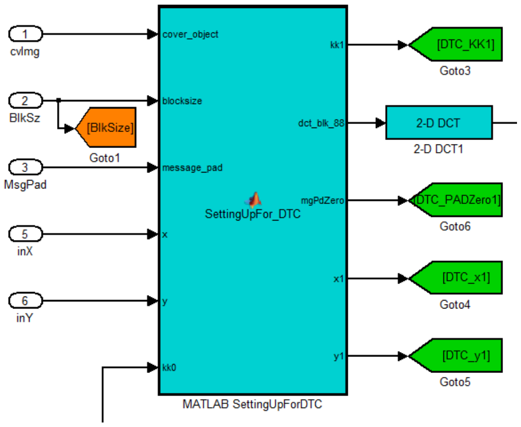


Fig. 3.1. Digested Embedding process

Step 3: Final Stage, create watermarked image according to data message.

At this stage we collected the [8, 8] blocks that were generated in step2 and reconstructed them as the watermarked image. Also we need to know about coding key to regenerate original image and data message at receiver.

IV. MATLAB IMPLEMENTATION OF EXTRACTING MESSAGE IN RECEIVER

The figure 4, shows process for extracting watermarking message.

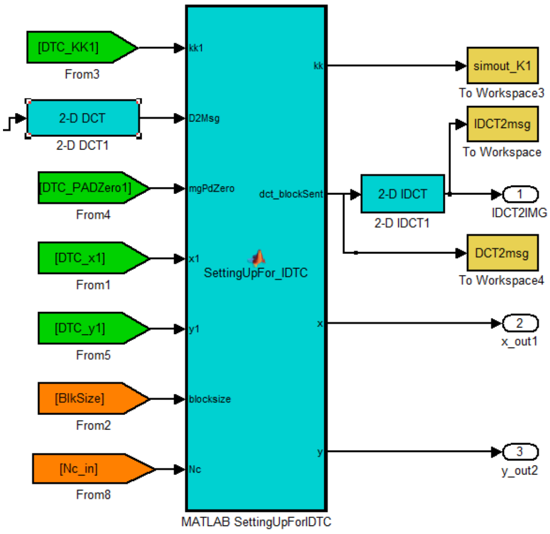


Fig. 4. Extracting watermarked message

The process is performed by using three different steps:
Step1: Initialize the watermarked message
Step2: Extracting the message from the host image
Step 3: Reconstruction of the message.

According to previous section in this section we do the reverse process to extract message data and host image by using coding key from embedding section.

V. DSP BUILDER COMPILER TO FPGA BOARD

In this part when we create Simulink files according to previous sessions (3 and 4), then watermarking process of encryption data message to host image complete.

Now we need to compile our code for FPGA board as we mentioned before implementation of watermarking process to FPGA board give us these opportunity for fast computing and processing data on FPGA board and also it could be portable because of size and capability of FPGA boards[9,10].

Implementation of MATLAB Simulink file to the embedded system (FPGA board) done with DSP Builder software. When we create the final Simulink model, then we need to convert it to VHDL code for FPGA board [11]. VHDL code is one of the FPGA board languages [12]. By using of DSP Builder software we generate VHDL code for FPGA board then when the compilation was done without any error, we select Specific board; here in this project we used DE2-115 board made by Terasic (figure 5). This board uses Cyclone IV E chip. DE2-115 FPGA board made by Terasic features the Cyclone IV E device and is the low cost, low power and a rich supply of logic, memory and DSP capabilities. It has 114,480 logic elements (LEs).

By downloading the VHDL code to the board we implement our Simulink and algorithm on to the FPGA board [13, 14]. Figure 6 shows screen shot of DSP Builder compilation process.

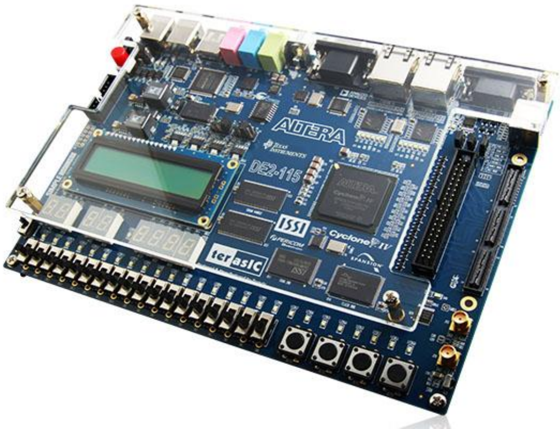
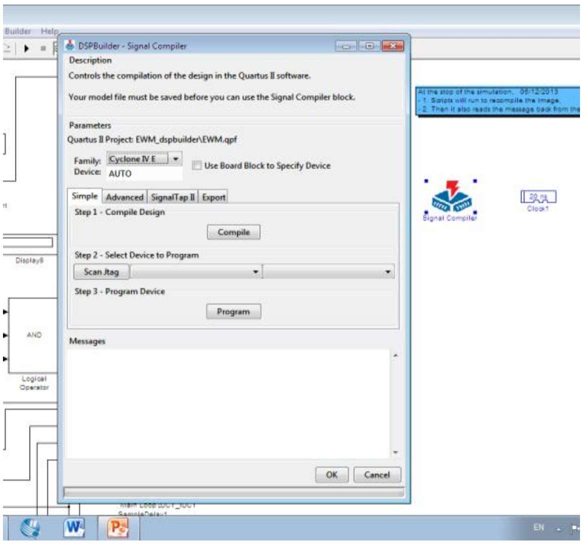


Fig. 5. DE2-115 board (www.Terasic.com)



DSPBuilder - Signal Compiler

Description

Controls the compilation of the design in the Quartus II software.

Your model file must be saved before you can use the Signal Compiler block.

Parameters

Quartus II Project: EWM_dspbuilder\EWM.qpf

Family: Cyclone IV E
Device: AUTO
☐ Use Board Block to Specify Device

Step 1 - Compile Design

Step 2 - Select Device to Program

Step 3 - Program Device

Messages

At the stop of the simulation, 06/12/2013
- 1. Scripts will run to recompile the image.
- 2. Then it also reads the message back from the

Fig. 6. DSP Builder compilation process

VI. SIMULATION RESULTS

From the established model in this paper, it can be learned that Simulink [15,16] have the outstanding advantages as follows, establishment of the model is visual, simple, visual function is powerful, and complicated dynamic analysis program can be embodied by the simplest means in Simulink model. The development of the model is easy and reliable.

In this part we show the result of Simulink files and implementation on FPGA board. Figure 7 shows the results, in this figure the host image (left side) and watermarked image (right side) and message image in black and white (middle) are shown.

According to Figure 7, we could use any picture by determined size of [M, N] and data image by size [m, n] as message that M and N must be multiple of m and n respectively. We changed the data image format to black and white image and according to message data bits we embedding data message to host image to create the watermarking image. After we complete the Simulink step we could create VHDL code by using DSP_Builder software and by selecting appropriate FPGA board, download our code to the board and processing data will be done on the board. As a result we could have watermarked image as an output, in the receiver we could extract data message from host image according to extracting data Simulink model.

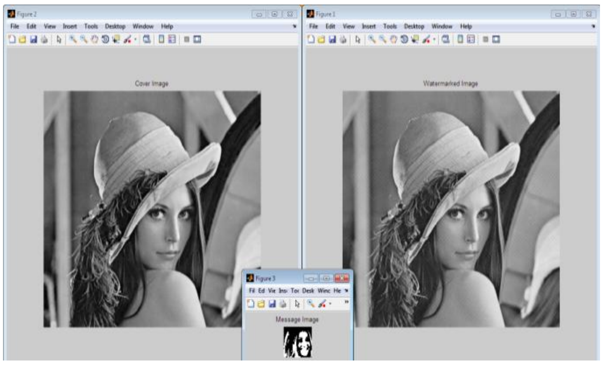


Fig. 7. Watermarking process Simulation result

VII. CONCLUSION

A Watermarking is a very active research field with a lot of applications. The increasing amount of research on watermarking over the past decade has been largely driven by its important applications in digital copyrights management and protection, our contribution was to study and analyze image watermarking process and implemented on hardware FPGA board.

Image includes of huge amount of data for processing then we need Fast and reliable processor. FPGAs are great fits for video and image processing applications, such as broadcast infrastructure, medical imaging, HD video conferencing, video surveillance, and military imaging. Video and image processing solutions for Altera FPGAs include optimized development tools. These solutions can improve cost, performance, and productivity for many video and imaging applications. Also by implementing watermarking image processing in FPGA board, it could be portable and available in each place.

REFERENCES

[1] G. Coatrieux, C. L. Guillou, J. M. Cauvin, and C. Roux, “Reversible watermarking for knowledge digest embedding and reliability control in medical images,” IEEE Trans. Inf. Technol. Biomed., vol. 13, no. 2, pp. 158–165, Mar. 2009.

[2] J. Fridrich, M. Goljan, and R. Du, “Lossless data embedding-new paradigm in digital watermarking,” Eur. Assoc. Signal Process. J. Appl. Signal Process., vol. 2002, no. 2, pp. 185–196, Feb. 2002.

[3] International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 5, May 2013, “Watermarking of Images in Discrete Cosine transform “ , Prof. Priya Pise, Prof. R M Goudar

[4] A Color Image Watermarking Scheme in the Associated Domain of DWT and DCT Domains Based on Multi-channel Watermarking Framework, Jiang-bin Zheng, Sha Feng in Digital Watermarking (2009)

[5] S. K. Lee, Y. H. Suh, and Y. S. Ho, “Reversible image authentication based on watermarking,” in Proc. IEEE Int. Conf. Multimedia Expo, Jul. 2006, pp. 1321–1324.

[6] A High Robust Blind Watermarking Algorithm in DCT Domain, Ching-Tang Hsieh, Min-Yen Hsieh in Knowledge-Based Intelligent Information anengineering systems(2005)

[7] Comparisons and Analysis of DCT-based Image Watermarking Algorithms, Ihab Amer, Pierre Hishmat, Wael Badawy… in Advanced Techniques in Computing Sciences and software engineering(2010)

[8] Z. Ni, Y. Q. Shi, N. Ansari, and W. Su, “Reversible data hiding,” IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 3, pp. 354–362, Mar.2006.

[9] Tsai, T.H., Lu, C.Y.: A Systems Level Design for Embedded Watermark Technique using DSC Systems. In: Proceedings of the IEEE International Workshop on Intelligent Signal Processing and Communication Systems. (2001)

[10] Petitjean, G., Dugelay, J.L., Gabriele, S., Rey, C., Nicolai, J.: Towards Real-time Video Watermarking for Systems-On-Chip. In: Proceedings of the IEEE International Conference on Multimedia and Expo (Vol. 1). (2002) 597–600

[11] Garimella, A., Satyanarayan, M.V.V., Kumar, R.S., Murugesh, P.S., Niranjan, U.C.: VLSI Impementation of Online Digital Watermarking Techniques with Difference Encoding for the 8-bit Gray Scale Images. In: Proc. of the Intl. Conf. on VLSI Design. (2003) 283–288

[12] Mathai, N.J., Sheikholeslami, A., Kundur, D.: VLSI Implementation of a Real-Time VideoWatermark Embedder and Detector. In: Proceedings of the IEEE International Symposium on Circuits and Systems (Vol. 2). (2003) 772–775

[13] J.D., Caroll, B.D.: Digial Logic Analysis and Design. Prentice Hall, Upper Saddle River, New Jersey, USA (1995) 22. Smith, D.J.: HDL Chip Design: A Practical Guide for Designing, Synthesizing and Simulating ASICs and FPGAs Using VHDL or Verilog. Doone Publications, USA (1998)

[14] http://www.altera.com/technology/dsp/dsp-builder/dsp-simulink.html DSP Builder information

[15] http://www.mathworks.com/products/slhdlcoder/description1.html

[16] http://www.mathworks.com/company/events/webinars/wbnr51985.html -- Webinar on Simulink HDL coder

BIOGRAPHIES

Rohollah Mazrae Khoshki, PhD student in Oakland University, Rochester, MI, USA, in Embedded System Electrical Eng., He is working as a teaching assistant for the electrical and computer engineering department at Oakland University. He investigated and designed of Interferometric Fiber Optic Gyroscope (IFOG) as his Master Thesis, designed and implemented of Heater Health Monitoring (HHM) project for Nexthermal

Company. He is interested in Embedded Systems, FPGAs and Microcontrollers, Robotics and control, Image processing and Optoelectronic in research and practical applications.

Sami Oweis (seweis@oakland.edu) received his Masters in Science (M.S.) degree in Electrical and Computer Engineering, from Oakland University. He is currently pursuing his Ph.D in artificial intelligence and controls Engineering from Oakland University. He has been working as a teaching and research assistant for the Electrical and Computer Engineering department at Oakland University during this time. He has taught and mentored students in the areas of digital signal and imaging processing, microcontrollers, micro-based computer systems, embedded application, artificial intelligent and control theory as well as using MATLAB/Simulink. More information is at: https://www.linkedin.com/in/samioweis & https://www.researchgate.net/profile/Sami_Oweis

Dr. Subramaniam Ganesan is a Professor of Electrical and Computer Engineering, Oakland University, Rochester, MI, 48309, USA. He has over 25 years of teaching and research experience in Digital systems. He served as the chair of the CSE department from 1991 to 98. He is with Electrical and Computer Engineering Department since 2008. He received his masters and Ph.D. from Indian Institute of Science (IISc) Bangalore, India. He worked at National Aeronautical Laboratory (NAL) India, Ruhr University, Germany, Concordia University Canada, and Western Michigan University before joining Oakland University in 1986.