International Journal of Engineering Research and Application www.ijera.com ISSN : 2248-9622 Vol. 8, Issue 10 (Part -V) Oct 2018, pp 107-111

RESEARCH ARTICLE OPEN ACCESS

Hardware/Software Co-design Approach for a DCT-Based Watermarking Algorithm

Monalisa Samal [1], Sambit Pattnaik [2]

[1]*AssistantProfessor, Department of Electronics and Communication Engineering, Gandhi Institute For Technology (GIFT), Bhubaneswar*

[2]*Assistant Professor, Department of Electronics and Communication Engineering, Gandhi Engineering College, Bhubaneswar*

ABSTRACT— The rapid increase in the distribution of digital multimedia data over networks creates the need for copyright protection. Watermarking is one of the techniques that can be used for this copyright protection. Many authors have proposed pure software or hardware solutions for the implementation of watermarking algorithms. In this paper we propose a hardware/software co-design approach for the implementation of the watermarking algorithm. Processes that demand high performance are implemented in hardware while those that are not computationally expensive are implemented in software. As a result, power consumption is reduced since only portion of the algorithm is implemented in hardware. In this paper we implement a DCT-based visible watermarking algorithm. Our system is implemented on a Xilinx Virtex-II Pro board. 21% of the slices were utilized with a maximum frequency of 131.092MHz

I. INTRODUCTION

With the development of new technologies, and the advent of the internet, data are now accessible to any person who knows how to use a computer. This development even though beneficial, is accompanied with problems such as the non-authorized use or reproduction of others works, and piracy. To protect their works, authors are nowadays using copyright, which is a form of protection grounded in the U.S. Constitution and granted by law for original works of authorship fixed in a tangible medium of expression. Copyright covers both published and unpublishedworks.

An approach for data protection is the use of watermark. A watermark is a secondary image which is overlaid on a primary image, in order to provide a means of protecting it. Digital watermarking can be divided into four different categories: visible, dual, invisible-robust, and invisible-fragile [3]. Watermarking could also be used for Fingerprinting Broadcast Monitoring and Covert Communication (Steganography).

In this paper, we present a hardware/software watermarking system implemented on an FPGA platform, using a hard-core on the FPGA to run the software part, and a DCT/IDCT hardware block for the hardware part, since implementing the DCT/IDCT functionality in software would have led to a poor performance.
Our system is able to embed visible information such as label and watermark into a multimedia object, in order to protect it.

RELATEDWORK

In the recent years, several multimedia watermarking technologies have been developed [1]. Few deal with hardware approach to digital watermarking. The authorsin[1] and [2] presented watermarking techniques such as a DCT (Discrete Cosine Transform) domain visiblewatermarking for image protection and their implementations. The authors in [2] discussed a dual watermarking technique for images, in which a visible watermark and an invisible watermark are combined in order to increase the robustness of the watermarking. On the other hand, in order to increase the performance of the available watermarking techniques, the authors in [4] are proposing two different methods. The first approach consist of using the Graphics Processing Unit (GPU) available on the modern graphics cards for the complex mathematical computations, whereas the second alternative is to implement a dedicated processor chip, a coprocessor for the GPU, to accomplish the task. There are several factors that encourage designers to opt for a hardware implementation. The authors in [5] explained that hardware watermarking solution is often more economical because adding the watermarking component takes up a small dedicated area of silicon. The authors in [6] and [8] emphasize the advantages of a hardware implementation by introducing a new approach to watermarking on the FPGA board. The use of FPGA as design tool is also emphasized in [7] where, due to its low-cost, its flexibility, adaptability, reprogrammability and speed is preferred over other tools. In [8], a real-time watermarking processor for 2d-DWT (2-

Dimensional Discrete Wavelet Transform)-based video compression is proposed. Watermarking is not only used for copyright protection but also for providing increased security in biometrics data [3]. Introducing an application of wavelet-based watermarking, the authors in [9] provide means to increase the security of fingerprint miniature transmission and the protection of fingerprintimages.

II. SYSTEMOVERVIEW

The watermark algorithm for our system was first

the watermarked image in the time domain is Our system will mainly consist of three different units as represented in Figure 1. The 2d-DCT unit will receive the signal to be watermarked and convert it into the frequency
domain. A watermark, previously transformed into the frequency domain is added to the frequency domain version of the image by the Embedder unit. The obtained image will then be sent to the two-dimensional Inverse DCT unit, which will then convert it back into time domain, where the final watermarked image can be obtained.

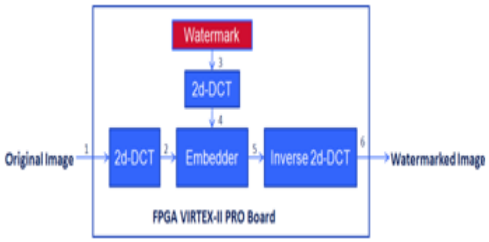


Figure 1: System Architecture

III. WATERMARKINGALGORITHMAN DSOFTWARE DESIGN

The basic color components (i.e. Red, Green and Blue) of the original image and the watermark are obtained. In the software portion of our implementation, the components are changed to "double" or "double precision" in order to obtain accurate results after computation. The original image and the watermark are subdivided into 8 x 8 blocks size. The RGB components serve as an input to the DCT module implemented in hardware for the purpose of obtaining a higher performance. The DCT coefficient of the 8 x 8 block is calculated via the DCT process. The formula for the DCT process is shown in eqn.1. The original image and the watermark image in the frequency domain are added together via the embedding process based on eqn.3.

wmark□□(dct_cov)□□(dct_watermark) eqn. 3

Where, wmarkrepresents the watermarked image, dct_covrepresents the original image, and implemented in MATLAB using two-dimensional Discrete Cosine Transform (2d-DCT) to verify the functionality of the algorithm. This also serves as a model for implementation in C.

The original image to be watermarked and the watermark are converted into frequency domain using the two dimensional discrete cosine transform (2d-DCT). This helps to increase robustness of the watermarked image against signal distortionsand attacks. After the addition of the originalimageandthewatermarkinthefrequencydomain, eqn. 2

obtained through the application a two-dimensional inverse DCT (2d-IDCT).

represents the embedding factor for the watermark and □represents the embedding factor for the original image. The □value is chosen as0.15,

and is 0.85 based on empirical method. The watermarked image in the frequency domain is converted to the watermarked image in the time domain via the two- dimensional inverse DCT (2d-IDCT).

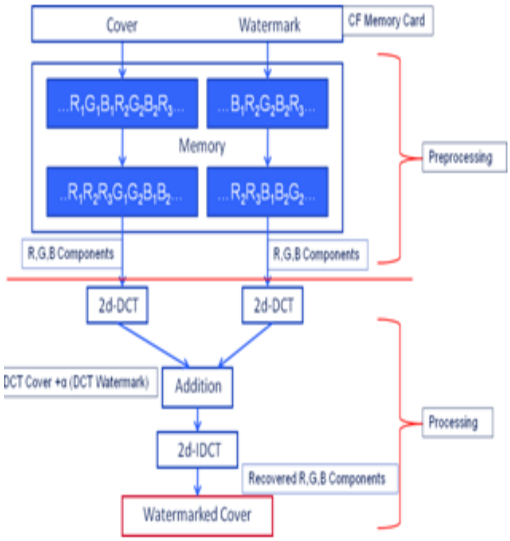


Figure 2: Block Diagram of the Watermarking Algorithm and

Implementation

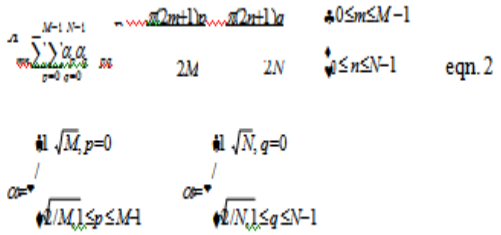
IV. HARDWAREAPPROACH

eqn. 1

Where, 'A' represents the input image and 'B' the two- dimensional discrete cosine transform of 'A'. 'M' and 'N' represents the row and column size of

the input image 'A' respectively. $\alpha_p$and $\alpha_q$are the discrete cosine transform coefficients.

The IDCT_cal module, where the actual two-dimensional inverse cosine transform is computed is also nothing but the direct implementation of the following formula, where 'B' is the two dimensional cosine transform of 'A'.



eqn. 2

For the hardware implementation, we first have to configure the Virtex-II Pro board using the Xilinx Platform Studio (XPS) to create a processor system consisting of the following processor IP: The hardware configuration is shown in Figure 3.

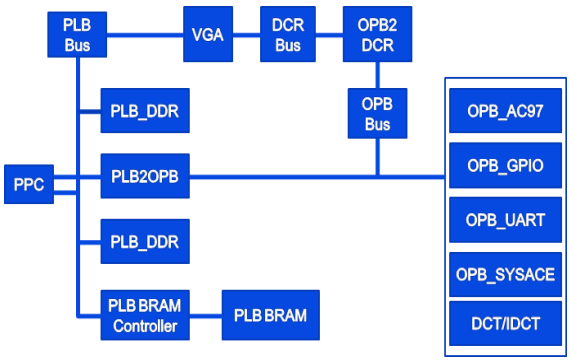


Figure 3: Hardware/Software Co-design Architecture

The PPC405 (Power PC processor) is a 32-bit implementation of the PowerPC embedded environment architecture that is derived from the PowerPC architecture. The various features of the PowerPC architecture are defined at three levels. This layering provides flexibility by allowing degrees of software compatibility across a wide range of implementations. For example, an implementation such as an embedded controller can support the user instruction set, but not the memory management, exception, and cache models where it might be impractical to doso.

The DCM (Digital Clock Manager) generates the various clock frequencies used by the processor, buses, and peripherals. The PPC 405 core accesses high speed and high performance system resources through the Processor Local Bus (PLB) interfaces on the instruction and data cache controllers. The PLB interfaces provide separate 32-bit address and 64-bit data buses for the instruction and data sides. The PLB memory controller connects to the FPGA memory. The BRAM is the FPGA memory. The PLB2OPB constituteabridgebetweenPLBandOPBbuses.ThePPC

405 core accesses low speed and low performance system resources through On-chip Peripheral Bus. The OPB is a fully synchronous bus that functions independently at a separate level of bus hierarchy. It is not intended to connect directly to the processor core. The OPB interfaces provide separate 32-bit address and up to 32-bit data buses. The configuration for our system is shown below:

Processor: PPC 405
Processor clock frequency: 100.000000 MHz # Bus clock frequency: 100.000000 MHz
Debug interface: FPGA JTAG # On Chip Memory: 64 KB
Total Off Chip Memory: 256 MB
DDR_SDRAM_32Mx64 Single Rank = 256 MB

V. HARDWARE/SOFTWARECODESIGN

Our HW/SW system is mainly composed of three parts: a 32-bit PPC, a DCT/IDCT core, and an embedder core as shown in Figure. 4. The host image and the watermark images stored on the compact flash are transferred in the memory for display and computation purposes. These images are then inputted to the DCT/IDCT core, which transforms them into their frequency components. The frequency componentsof both images will then be inputted to the embedder core. Once the watermark is embedded onto the host image, the formed structure is inputted back into the DCT/IDCT core, which converts the composed image from frequency domain into time domain. This output will then be saved into the memory, where it will be accessed by the VGA core in order to display it on a monitor. The DCT/IDCT Core is implemented in hardware because of the high computational cost of the process. A block diagram of our hardware/software architecture is shown in Figure4.

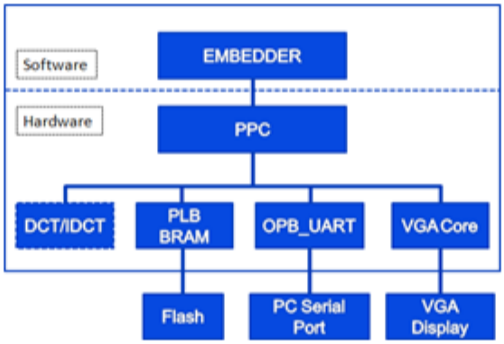


Figure 4: Completed Hardware Design

VI. TEST ANDRESULTS

The watermarking algorithm was first implemented in MATLAB to verify the algorithm used. To ensure the functionality of our system, each hardware/software module was tested through the hyper-terminal. In order to view the matrices values on the hyper-terminal, a portion of code was written and the obtained matrices are presented on the following figures. Figure 5 represents the matrices of the R components of the host image being processed: Process (0). Figure 6 represents the matrices obtained from Process (1&2), where the G and B components are being processed.

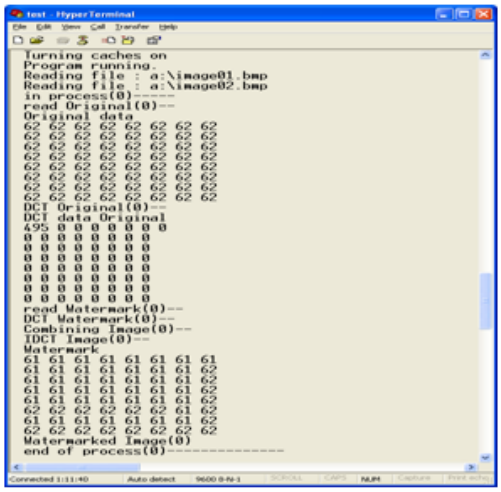


Figure 5: Capture of the hyper terminal-Process (0)

tradeoff between hardware and software implementation which is not present when a pure software or hardware isused.

VIII. ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support provided by National Science Foundation (NSF) grant number NSF – 0431818 "A Design- and Project- Oriented Innovative Electrical Engineering Program".

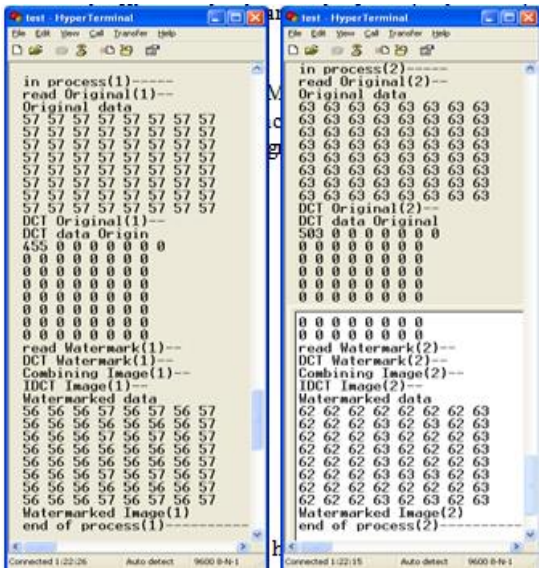


Figure 6: Capture of the hyper terminal-Process (1&2)

The output image for our hardware/software co-design is shown in Figure 7.

Figure 7: Watermarking Result

The synthesis result for the hardware portion of our system is shown in Table 1. The table shows that 21 % of the slices were utilized and the maximum frequency for our systemis 131.095 MHZ

Table 1: FPGA Resource Usage

Slices	2952 (21%)
Flip Flops	3008 (10%)
4-input LUTs	3601 (13%)
RAMs	476
BRAMs	35 (25%)
Minimum Period	7.628 ns
Maximum Frequency	131.095 MHz

VII. CONCLUSION

We have successfully implemented the watermarking algorithm using hardware/software co-design approach. Our system was implemented on the Xilinx Virtex-II Pro Board, using the Xilinx Platform Studio (XPS). Our design allows

REFERENCES

[1] F. Hartung, M. Kutter, "Multimedia Watermarking Techniques". In Proceedings

of the IEEE. Volume87,Issue 7, July 1999. Pages: 1079-1107

[2] 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 (ICME) (Vol.2), pp. 1029-1032,2000

[3] O. B. Adamo, S. P. Mohanty, E. Kougianos, and M. Varanasi, "VLSI Architecture for Encryption and Watermarking Units Towards the Making of a Secure Digital Camera", in Proceedings of the IEEE International SOC Conference (SOCC), pp. 141-144, 2006.

[4] Mohanty, S.P, Pati, N. and Kougianos, E "A Watermarking Co-Processor for New Generation Graphics Processing Units".In proceedings of the ICCE 2007 International Conference on Consumer Electronics, Pages: 1-2, 2007.

[5] Nebu J Mathai, DeepaKundur, and Ali Sheikholeslami "Hardware Implementation Perspectives of Digital Video Watermarking Algorithms" In Proceedings of the IEEE Conference on Signal Processing Vol.51, Pages: 925-938, 2003.

[6] Hesham A. El Zouka''FPGA Based Implementation of Robust Watermarking System'' In Proceedings of the Fifth International Conference on Information Technology. Pages: 1274-1278,2008.

[7] E.Ayeh, K. Agbedanu, Y. Morita, O. Adamo, and P.Guturu "FPGA Implemetation of an 8-bit Simple Processor" In Proceedings of the IEEE Region 5 P-BASICS^2 Technical Conference, March , 2008.

[8] Seo Young-Ho and Kim Dong-Wook''FPGA Design of Real-Time Watermarking Processor for 2DDWT-Based Video Compression'' International Technical Conference on Circuits/Systems, Computers and Communications, 2003.

[9] Khalil Zebbiche, LahouariGhouti, FouadKhelifi and Ahmed Bouridane ''Protecting Fingerprint Data using Watermarking''InproceedingsoftheFirstNASA/ESA ConferenceonAdaptiveHardwareandSystems .Pages: 451-456,2006.

[10] Xilinx, "Virtex-II Pro X Platform FPGA UserGuide" UG012 (v4.2) November2007