Digital Watermarking System using FPGA hardware for Video authentication

Mr. Ashish S. Bhaisare[1], Prof. Atul H. Karode[2]
Electronics and Telecommunication Department
SSBT's COET Bhambori, Jalgaon, India
[1]bhaisare.ashish@gmail.com, [2]atul_karode@rediffmail.com

Abstract — This **paper presents a hardware implementation of a digital watermarking system using Least Significant Bit (LSB) steganography technique in a cyclone II FPGA of the Altera family. The design utilizes the Nios embedded processor as well as specialized logic to perform the steganography steps. Experimented results show that such a hardware-based video authentication system using this watermarking technique features minimum video quality degradation and it can withstand certain potential attacks i.e. cover-up attack, cropping, segment removal on video sequences. To achieve high performance, the proposed system architecture employs pipeline structure and uses parallelism.**

Keyword — ***Least Significant Bit (LSB), Nios embedded processor (Altera), steganography, parallelism, Field Programmable Gate Array (FPGA),HAAR wavelet, Framework, macroblock, Peak Signal to Noise Ratio (PSNR)***

I. INTRODUCTION

Digital Watermarking (WM) is the process of embedding an additional, identifying information within a host multimedia object, such as text, audio, image or video [1]. By adding a transparent watermark to the multimedia content, it is possible to detect hostile alterations as well as to verify the integrity and the ownership of the digital media. There are several techniques to conceal information inside cover-image. The spatial domain techniques manipulate the cover-image pixel bit values to embed the secret information.

The secret bits are written directly to the cover image pixel bytes. Consequently, the spatial domain techniques are simple and easy to implement. The Least Significant Bit (LSB) is one of the main techniques in spatial domain image steganography [2]. The transform domain techniques embed the message in the frequency domain of the cover image.

Typically, spatial domain techniques are easily detectable and have larger capacity [3]. On the other hand, frequency-based steganography has higher peak signal-to-noise ratio (PSNR) and is more secure.

The main objective of this paper is to describe an efficient hardware-based concept of a digital video WM system which features low power consumption, efficient and low cost implementation, high processing speed, reliability and invisible, semi-fragile watermarking in compressed video streams. The system also features minimum video quality degradation and can withstand certain potential attacks i.e. cover-up attack, cropping, segment removal on video

Sequences. The above-mentioned design objectives were achieved via combined parallel hardware architecture with a simplified design approach of each of the components. [4]

II. PAST WORK REVIEW OF VIDEO WATERMARKING SYSTEM

An FPGA prototyping is presented for HAAR-wavelet based real time video watermarking in [5] by Jeong et. al. A real-time video watermarking system using DSP and VLIW processors was presented in that embeds the watermark using fractal approximation by Petitjean et. al. Mohanty et. al. [6] presented a concept of secure digital camera (SDC) with a built-in invisible-robust watermarking and encryption facility. Also another watermarking algorithm and corresponding VLSI architecture that inserts a broadcasters logo (a visible watermark) into video streams in real-time was presented [7] by the same group. The well-known patchwork watermarking in [8] inserted a massage by supporting that two sets of randomly selected pixels are Gaussian distribution with zero mean.

In prior research [9], a real-time watermarking embedder-detector for a broadcast monitoring system is presented in the context of a VLIW DSP processor. The insertion mechanism involves addition of pseudo-random numbers to the incoming video stream based on the luminance value of each frame. The watermark detection process involves the calculation of correlation values. In [10], the graphics processing unit (GPU) is utilized for hardware assisted real-time watermarking. In [11], a watermark detection system is presented for a hardware based video watermark embedder using a Stratix FPGA from Altera®, a PCI controller, and Microsoft Visual C#. In [06], custom IC for video watermarking is presented that performs all operations using floating-point datapath architectures for both the watermarking embedder and detector. The embedder consumes 60 mW, and the detector consumes 100 mW while operating at 75 MHz and 1.8 V. In [03], a video watermarking system called "traceable watermarking" is proposed for digital cinema.

The research presented in the current paper will significantly advance the state -of- the art in digital rights management. The algorithm and architecture proposed in this paper will be immensely useful for real-time copyright logo

insertion in emerging applications, such as IP-TV. An embedded system that will allow such operations needs to have embedded copyright protection facilities such as the one

presented in this paper. These watermarking algorithms primarily work off-line, i.e., the multimedia objects are first acquired, and then the watermarks are inserted before the watermarked multimedia objects are made available to the user. In this approach there is a time gap between the multimedia capture and its transmission.

The objective of this paper is to lead to a hardware-based watermarking system to bridge that gap.

III. WATERMARKING IN THE MPEG-4 FRAMEWORK: A HARDWARE COST PERSPECTIVE

The most important phases for video compression are color space conversion and sampling, the Discrete Cosine Transform (DCT) and its inverse (IDCT), Quantization, Zigzag Scanning, Motion Estimation, and Entropy Coding. The watermarking process is implemented with a single embedding step in the video compression framework. In this section the watermarking algorithm in the framework of MPEG-4 is discussed highlighting the design decisions that were made towards real-time performance through a hardware-based solution.

A. Color Space Conversion

The conversion from RGB-color space to YCbCr-color space is performed using the following expression [12,13]:

$$\left.\begin{aligned} Y &= 0{:}299R + 0{:}587G + 0{:}114B; \\ Cb &= 0{:}564(B - Y) + 128; \\ Cr &= 0{:}731(R - Y) + 128{:} \end{aligned}\right\} \quad (1)$$

A total of 6 adders and 5 multipliers are needed to perform the color conversion, and they can be concurrent. The delay introduced does not contribute significantly to the critical path delay as the conversion takes place at the input stage, where the additive terms in the Cb and Cr components guarantee that their values will be positive. The sampling rate is chosen to be 4:2:0 so that in a 4 pixel group, there are four Y pixels, a single Cb pixel and a single Cr pixel to meet digital TV broadcasting standards.

B. Discrete Cosine Transformation (DCT)

DCT is one of the computationally intensive phases of video compression. For ease of hardware implementation, a fast DCT algorithm and its inverse [12,13] is selected. The fast DCT algorithm reduces the number of adders and multipliers so that the evaluation of the DCT coefficients is accelerated. The twodimensional DCT and IDCT algorithms can be implemented by executing the one-dimensional algorithms sequentially, once horizontally (row-wise) and once vertically (column-wise).

C. Quantization

After the DCT, the correlation of pixels of an image or video frame in the spatial domain has been de-correlated into discrete frequencies in the frequency domain. Since human visual system (HVS) perception is more acute to the DC coefficient and low frequencies, a carefully designed scalar quantization approach reduces data redundancy while maintaining good image quality. In the MPEG-4 video compression standard, a uniform scalar quantization is adopted. The feature of the scalar quantization scheme is an adaptive quantized step size according to the DCT coefficients of each macroblock. For computational efficiency and hardware simplification, the scalar quantization step size can be chosen from predefined [11].

D. Zigzag Scanning

Zigzag scanning sorts the matrix of DCT coefficients of video frames in ascending order. For progressive frames and interlaced fields, the zigzag scanning routes are provided by predefined tables as explained in [12,13].

E. Motion Estimation

Prior to performing motion estimation, an image (video frame) is split into smaller pixel groups, called macroblocks, as the basic elements of the image rather than a single pixel. This is driven by a compromise between efficiency and performance to analyze a video's temporal model. A macroblock commonly has a size of 16×16 pixels. With the macroblock in the base frame and its two dimensional motion vector, the current frame can be predicted from the previous frame. In the MPEG-4 standard, the region in which the macroblock is sought for match could be a square, diamond, or of arbitrary shape. For most applications, a square region is considered. For example, if the macroblock has pixel size, the searching region will be a pixel block. The similarity metric for two blocks is the minimized distance between them. For simplicity, the sum of the absolute difference (SAD) is applied as the criterion for matching, as represented in [12,13]:

$$SAD(x,y) = \begin{cases} \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} |c(i,j) - p(i,j)| \text{ for } (x,y) = (0,0), \\ \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} |c(i,j) - p(i+x, j+y)|, \end{cases} \quad (2)$$

where c (i; j) are the pixels of the current block, i; j = 0; 1; ; N − 1; p(m; n) are the pixels of the previous block in the searching region, and m; n = − R; − R + 1…..0; 1; ; R + N − 1, where the size of the macroblock is R pixels. Motion estimation is in the critical path of video compression coding and most time delay will occur at this step. The SAD algorithm will search the square target region exhaustively to find a matching macroblock. The output of this procedure is the prediction error for motion compensation and the motion vector. The hardware implementation of the motion estimation block is sequential and contributes the largest delay to the critical path.

F. Entropy Coding

After DCT and quantization compression, additional compression can be achieved via entropy coding, which includes Huffman coding, Arithmetic coding, etc. Unlike lossy compression, as in the color space, DCT and quantization procedures, the entropy coding compression is lossless. The entropy coding efficiency depends on the precision of calculating the probability of occurrence of each coefficient. However, calculating probabilities of all the coefficients is impossible in real-time MPEG-4 coding and watermarking. The approach we followed is to utilize precalculated Huffman code tables for generic images [09]. The robustness of DCT watermarking arises from the fact that if an attack tries to remove watermarking at mid-frequencies,

it will risk degrading the fidelity of the image because some perceptive details are at mid-frequencies

IV. HARDWARE ARCHITECTURE DESIGN

There exists a wide range of available techniques to implement the peripheral blocks of the proposed video WM system. Here, the focus is on simplifying the process as much as possible, thus making it fit easily within existing video processing circuitry. At the same time, the security level and video frame quality are kept high. An overall view of the hardware implementation for the video WM system is depicted. Basically, the proposed system architecture includes six modules: video camera, video compressor, watermark generator, watermark embedder, control unit and memory. The parts implemented by the FPGA are shown in Fig. No. 1 shaded blocks.

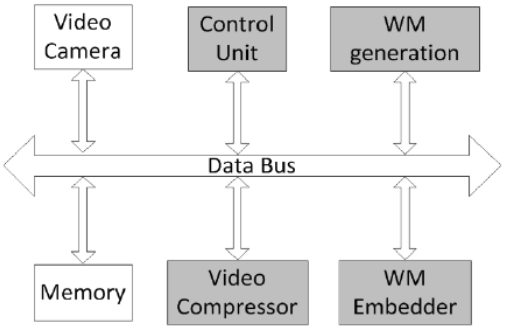


Fig. 1 Block diagram of the hardware system architecture.

The hardware implementation for the complete design is developed using Verilog HDL. As previously mentioned, all the processing in the implementation is assumed to be done on a block basis (such as 8 × 8 pixels). First, the captured video frame is temporarily stored in a memory buffer and then each block of the frame data is continuously processed by the video compressor unit using DCT and quantization cores. The watermark embedder block inserts an identifying message, generated using the watermark generator unit, in the selected block data within the video frame and sends it to memory for storage. The control unit is responsible for driving the operations of the modules and the data flow in the whole system.

A. Video Compressor

For hardware implementation of the video compressor module, the Motion JPEG (MJPEG) video encoding technique rather than the MPEG-2 standard was chosen as because MJPEG offers several significant advantages over

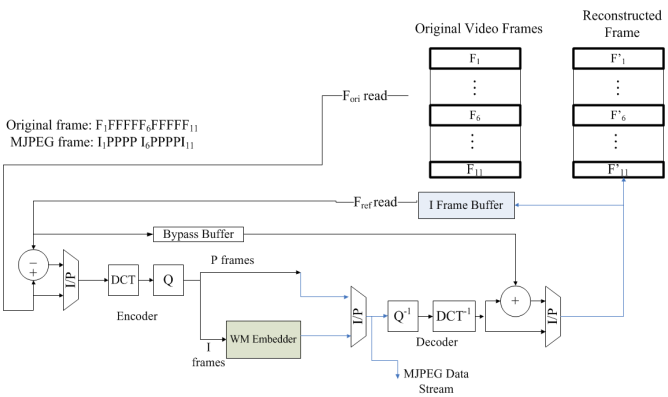


Fig.2 Hardware architecture of MJPEG video compressor module with Watermarking unit (Fori is the original frame, Fref is the reference frame).

the MPEG-2 technique for hardware implementation [14]. Even though MJPEG provides less compression than MPEG-2, it is easy to implement MJPEG format in hardware with relatively low computational complexity and low power dissipation. Moreover, it is available for free usage and modification, and it is now supported by many video data players like VLC Media Player [15], MPlayer [16]

Furthermore, using MJPEG video encoding technique has no effect on the watermark embedding procedure, since the intra-frames (chosen as watermarkable) in both compression standards have the same formats [17]. This is due to the same encoding process that is performed on the raw data. The difference is motion-compensated prediction, which is used to encode the inter-frames. However, as described, only the intra-frames are identified to be watermarked, thus the procedure does not affect the watermark embedding process. Therefore, for our evaluation purposes, the MJPEG compression standard was found to be a better alternative for hardware implementation of the video compressor module than the MPEG-2 standard. Fig. 1 depicts the hardware architecture of the MJPEG video compressor. Depending on the type of the frames, raw video data is encoded either directly (intra-frame) or after the reference frame is subtracted (inter-frame). The encoded Inter-frames (P) are passed to the decoder directly and Intra-frames (I) are fed to watermark embedder and after WM embedding they are also passed to the decoder.

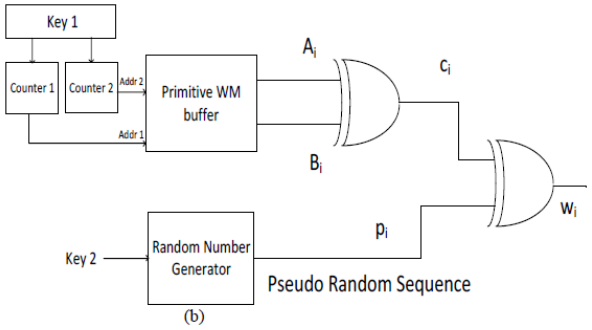
The output of the decoder is combined with the reference frame data delayed by the bypass buffer so that it arrives at the same time as the processed one and is fed to the multiplexer. The multiplexer selects the inverse DCT output (Decoded Intra frames) or the sum of the inverse DCT output and previous reference frame (Decoded Inter frames). The multiplexer output is stored in the SRAM as reconstructed frames. Furthermore, the decoded Inter frames are stored in a memory buffer to be utilized as reference frame for the next frames which are yet to be encoded. The second branch of the video data from the encoder block is fed to the watermark embedder module.

B. Watermark Generator

Fig. 3 describes the hardware architecture of the novel watermark generator. The expanding procedure is accomplished by providing consecutive clock signal so that expanded watermark sequence can be generated by reading out the primitive watermark sequence (a_i) for c_r times. Expanded sequence (a_i') is stored in memory buffer.
Scrambling is done by using the secret digital key *Key1*, which has two parts. The two different parts initiate two different counters. At each state of the counters two readings (addressed by *Add1* and *Add2*) from the buffer occur for having the XOR operation between them. Thus the scrambled watermark sequence, c_i is generated. Furthermore, different digital keys can make the counters start running with different states and generate different corresponding addresses, so that we can get different patterns of c_i.

A secure pseudo-random sequence p_i, generated by the proposed Gollman Cascade Filtered Feedback with Carry Shift Register (FFCSR) RNG [16], seeded with secret key *Key2*, is used to modulate the expanded and scrambled

watermark sequence c_i. Finally, the generated secure watermark data w_i is embedded into the video stream by the



watermark embedder.

Fig 3 Hardware architecture of the watermark generator module.

C. Watermark Embedder

A schematic view of the hardware architecture for the watermark embedder unit is presented in Fig. 4. As described by Shoshan et al. [18], the watermark embedder works in two phases (calculation and embedding). When considering a cycle of embedding the watermark in one 8 × 8 block of pixels, each phase takes one block cycle. Two blocks are processed simultaneously in a pipelined manner so that the embedding process only requires one block cycle. As the number of cells to be watermarked (N) in an 8x8 block increases, the security robustness of the algorithm also increases. But such an increase reduces the video frame quality because of the reduction in the originality of the video frame. Simulation results show that even for *N* as low as 2, the performance like detection ratio or Peak Signal to Noise Ratio (PSNR) is satisfactory. A block that produces less than *N* cells is considered to be unmarked and disregarded. Only blocks that are distinctively homogeneous and have low values for high-frequency coefficients are problematic. The details of the architecture for the watermark embedder module were presented by Shoshan et al. [18].

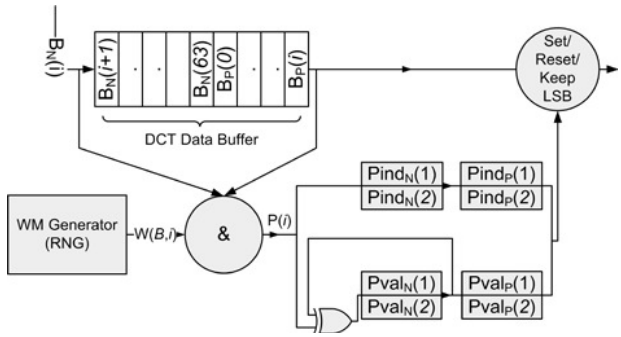


Fig. 4 Hardware architecture of the watermark embedder module designed by Shoshan et al. [18].

D. Control Unit

The control unit generates control signals to synchronize the overall process in the whole system. As shown in Fig. 5, it is implemented as a finite state machine (FSM) with four main states.

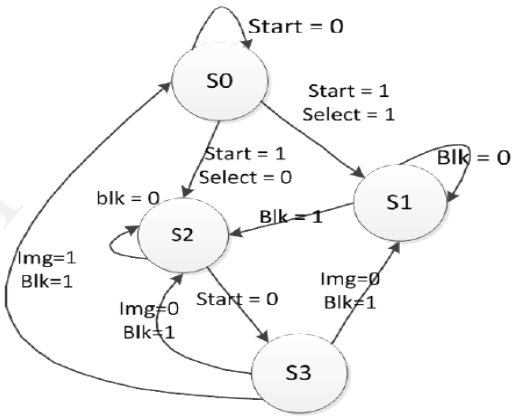
• **S0:** In this initial state, all the modules stay idle till the signal **Start** is set to high. Depending on the **Select** signal, different processing steps are applied to the video frames. When **Select** is 1, the control state will move to state S1 for intra-frames or directly jump to the state S2 for inter-frames if **Select** is 0.

• **S1:** In this state, watermark embedding is performed. The intra-frame blocks read from memory and the generated watermark sequence are added together by activating the watermark embedder module in this state. Once the watermarking is completed for the block, **Blk** signal will be '1' and the FSM moves to state S2.

S2: In the state S2, the watermarked intra-frame data from the watermark embedder module or the unmodified inter-frame sequences from the memory are encoded. The signal **Blk** remains '0' until the encoding of the current block is completed. When finished, the encoded and watermarked blocks are fed to the next state S3.

• **S3:** In this stage, the watermarked and compressed video frame blocks are written back to the memory. When all the blocks of a frame are encoded the control signal **Img** changes to '1' and the FSM goes back to the state S0 for considering the next frame. If encoding of all the blocks of the current frame is not finished, the system goes back to the state S1 or S2 depending on the type of the current frame (inter-frame or intra-frame).

Fig. 5 State diagram of the controller for WM system.



V. SUMMARY AND CONCLUSIONS

First design of the hardware architecture of a novel digital video watermarking system to authenticate video stream in real time is presented in this paper. FPGA -based prototyping for the hardware architecture has been developed. The proposed system is suitable for implementation using an FPGA and can be used as a part of an ASIC. In the current implementation, FPGA is the simple and available way of the proof-of-concept. The implementation makes integration to peripheral video (such as surveillance cameras) to achieve real-time image data protection.

REFERENCES

[1] V. M. Potdar, S. Han, and E. Chang, "A survey of digital image watermarking techniques", in Proc. of the IEEE International Conference on Industrial Informatics, pp. 709–716, Perth, Australia, Aug. 2005.

[2] T. Morkel, J. Eloff and M. Olivier, "An Overview of Image Steganography," The Fifth Annual Information Security South Africa Conference (ISSA2005), Sandton, South Africa, July 2005

[3] A. D. Gwenael, and J. L. Dugelay, "A guide tour of video watermarking," Signal Process. Image Commun. , vol. 18, no. 4, pp. 263–282, Apr. 2003.

[4] Cox J.,Miller L., Bloom A.,Digital Watermarking,Morgan Kaufmann Publishiers,USA, 2002.

[5] Y.-J. Jeong, K.-S. Moon, J.-N. Kim, "Implementation of Real Time Video Watermark Embedder Based on Haar Wavelet Transform Using FPGA", in Proceedings of the Second International Conference on Future Generation Communication and Networking Symposia (FGCNS), 2008,
pp. 63 – 66.

[6]S. P. Mohanty, "A Secure Digital Camera Architecture for Integrated Real-Time Digital Rights Management", Journal of Systems Architecture, Volume 55, Issues 10-12,October-December 2009, pp. 468-480.

[7] S. P. Mohanty and E. Kougianos, "Real-Time Perceptual Watermarking Architectures for Video Broadcasting", Journal of Systems and Software,
Vol. 84, No. 5, May 2011, pp. 724-738.

[8] I. K. Yeo and H. J. Kim, "Generalized patchwork algorithm for video watermarking," Multimedia System, vol. 9, no. 3, pp. 261-265, 2003

[9]L. D. Strycker, P. Termont, J. Vandewege, J. Haitsma, A. Kalker, M. Maes, G. Depovere, An Implementation of a Real-time Digital Watermarking Process for Broadcast Monitoring on a Trimedia VLIW Processor, in: Proceedings of the IEE International Conference on Image Precessing and Its Applications
(Vol. 2), 1999, pp. 775–779.

[10] A. Brunton, J. Zhao, Real-time Video Watermarking on Programmable Graphics Hardware, in: Proceedings of Canadian Conference on Electrical and Computer Engineering, 2005, pp. 1312–1315.

[11] Y.-J. Jeong, W.-H. Kim, K.-S. Moon, J.-N. Kim, Implementation of Watermark Detection System for Hardware Based Video Watermark Embedder,in: Proceedings of the Third International Conference on Convergence and Hybrid Information Technology (ICCIT), 2008, pp. 450 – 453.

[12] J. Chen, et al., Design of Digital Video Coding Systems - A Complete Compressed Domain Approach, Marcel Dekker, New York, 2002.

[13] W. Cai, FPGA Prototyping of aWatermarking Algorithm for MPEG-4, Master's thesis, Department of Engineering Technology, University of North Texas (Spring 2007).

[14] A. Filippov, "Encoding high-resolution Ogg/Theora video with reconfigurable FPGAs," in Xcell Journal, Second Quarter 2005.Available:http://www.xilinx.com/publications/xcellonline/xcell_53 /xc_pdf/xc_video53.pdf

[15] http://www.videolan.org/vlc/features.php?cat=video

[16] http://www.mplayerhq.hu/design7/info.html

[17] http://wiki.xiph.org/index.php/TheoraSoftwarePlayers

[18] Y. Shoshan, A. Fish, G. A. Jullien, and O. Yadid-Pecht, "Hardware implementation of a DCT watermark for CMOS image sensors," IEEE International Conference on Electronics, Circuits, and Systems, pp. 368-371, Aug. 2008.

BIOGRAPHIES

Mr. Ashish Sheshrao Bhaisare received B.E degree in Electronics from Mumbai University, India in 2009. Currently he is pursuing his M.E at SSBT's COET in the department of Electronics and Telecommunication, India and specialization in Digital Electronics, his research involves watermarking system for video authentication using FPGA.

Prof. Atul Hiralal Karode completed M.E in Electronics and Telecommunication and specialization in Image Processing. Currently working as Associate Professor in
EXTC Department of SSBT's COET, Bhambori Jalgaon Maharashtra.