FPGA Based Implementation of an Invisible-Robust Image Watermarking Encoder

Saraju P. Mohanty[1], Renuka Kumara C.[2], and Sridhara Nayak[2]

[1] Dept. of Computer Science and Engineering, Univ. of North Texas,
Denton, TX 76203, USA
smohanty@cs.unt.edu
[2] Manipal Centre For Information Science, Manipal Academy of Higher Education,
Manipal – 576104, India
shridhar.n@mcis.manipal.edu

Abstract. Both encryption and digital watermarking techniques need to be incorporated in a digital rights management framework to address different aspects of content management. While encryption transforms original multimedia object into another form, digital watermarking leaves the original object intact and recognizable. The objective is to develop low power, real time, reliable and secure watermarking systems, which can be achieved through hardware implementations. In this paper, we present an FPGA based implementation of an invisible spatial domain watermarking encoder. The watermarking encoder consists of a watermark generator, watermark insertion module, and a controller. Most of the invisible watermarking algorithms available in the literature and also the algorithm implemented in this paper insert pseudorandom numbers to host data. Therefore, we focus on the structural design aspects of watermarking generator using linear feedback shift register. We synthesized the prototype watermarking encoder chip using Xilinx FPGA.

1 Introduction

Owing to the usage of Internet, concerns about protection and enforcement of intellectual property (IP) rights of the digital content involved in the transaction, are mounting. In addition, unauthorized replication and manipulation of digital content is relatively trivial and can be achieved using inexpensive tools. Issues related to ownership rights of digital content are addressed by digital rights management (DRM) systems [1, 2]. Various aspects of content management namely, content identification, storage, representation, and distribution and intellectual property rights management are highlighted in DRM. Besides, unauthorized access of digital content is being prevented by implementing encryption technologies. However, it does not prevent an authorized user from illegally replicating the decrypted content. Hence, encryption alone does not address all the IP issues related to DRM. Digital watermarking is one of the key technologies that can be used for establishing ownership rights, tracking usage, ensuring authorized access, preventing illegal replication and facilitating content authentication. Therefore, a two layer protection mechanism utilizing both watermarking and encryption is needed [3].

G. Das and V.P. Gulati (Eds.): CIT 2004, LNCS 3356, pp. 344–353, 2004.

Table 1. Watermarking Chips Proposed in Current Literature.

Research	Design Type	Watermarking	Multimedia	Domain	Chip Features
Hsiao [6]	Custom IC	Invisible-Robust	Image	Wavelet	NA
Maes [7]	FPGA board/IC	Invisible-Robust	Video	Spatial	$17/14\ kG$ Logic
Tsai [8]	Custom IC-0.35μ	Invisible-Robust	Image	DCT	$3.3V$,$50MHz$
Petitjean [9]	FPGA board	Invisible-Robust	Image	Fractal	$50MHz$
Garimella [10]	Custom IC-0.13μ	Invisible-Fragile	Image	Spatial	$1.2V$
Mathai [11]	Custom IC-0.18μ	Invisible	Video	Wavelet	$1.8V$
Tsai [12]	Custom IC	Invisible-Robust	Video	Spatial	NA
Mohanty [13]	Custom IC-0.35μ	Robust-Fragile	Image	Spatial	$3.3V$, $545MHz$
Seo [14]	FPGA board	Invisible-Robust	Image	Wavelet	$82MHz$
Mohanty [15]	Custom IC-0.35μ	Visible	Image	Spatial	$3.3V$, $292MHz$

Digital watermarking is the process of embedding data called a watermark into a multimedia object such that watermark can be detected whenever necessary for DRM. The digital watermarking system essentially consists of a watermark embedder and a watermark detector [4, 5]. The embedder inserts a watermark onto the host object and the detector detects the presence of the watermark. An entity called watermark key is also used during the process of embedding and detecting the watermark. This watermark key is unique and exhibits a one-to-one correspondence with every watermark. The key is private and known to only authorized parties, eliminating the possibility of illegal usage of digital content.

The goal is to develop low power, real time, reliable and, secure watermarking systems [16, 17]. Over the past decade, numerous watermarking algorithms have been invented and their software are available, however recently, hardware implementations are being presented in literature. We have listed most of the watermarking hardwares available in current literature in Table 1, which proves that the VLSI implementation of the watermarking algorithms is not yet significantly explored. A hardware based watermarking system can be designed on a field programmable gate array (FPGA) board, Trimedia processor board [7], or custom IC. The choice between the FPGA and cell based IC is a trade-off between cost, power, and performance [15, 18].

In this paper, we present an FPGA based implementation of an invisible-robust spatial domain watermarking encoder [19]. This algorithm is chosen as it is simple yet robust against geometric attack and is tested using Stirmark benchmark [20]. The watermarking encoder chip consists of a watermark generator, watermark insertion module, and a controller. The invisible watermarking algorithms implemented in this paper insert pseudorandom numbers to host data. Therefore, we focus on the structural design aspects of watermarking generator using linear feedback shift register (LFSR). We synthesized the prototype watermarking encoder chip in a Xilinx FPGA using VIRTEX technology which can be operated at $50MHz$ frequency.

2 Watermarking Algorithm

In this section, we describe the invisible-robust algorithm [19] chosen for VLSI implementation. Let us assume the following notations: I – original gray scale image,

W – binary or ternary watermark image, I^* – watermarked image, (i, j) – pixel location, E_1, E_2 – watermark embedding functions, D – watermark detection function, r – neighborhood radius, I_N – neighborhood image, K - digital watermark key, and α_1, α_2 – scaling constants.

The watermark insertion process consists of the following: First, the watermark W which is a ternary image having pixel values {0,1 or 2} is generated using the digital key K. Then, watermark insertion is performed by altering the pixels of original image using watermark embedding functions.

$$I^*(i,j) = \begin{cases} I(i,j) & \text{if } W(i,j) = 0 \\ E_1\big(I(i,j), I_N(i,j)\big) & \text{if } W(i,j) = 1 \\ E_2\big(I(i,j), I_N(i,j)\big) & \text{if } W(i,j) = 2 \end{cases} \tag{1}$$

The encoding functions E_1 and E_2 are defined as follows.

$$\begin{aligned} E_1(I, I_N) &= (1-\alpha_1)I_N(i,j) + \alpha_1 I(i,j) \\ E_2(I, I_N) &= (1-\alpha_1)I_N(i,j) + \alpha_2 I(i,j) \end{aligned} \tag{2}$$

The signs of α_1 and α_2 are used for the detection function and their actual values determine the watermark strength. The neighborhood image pixel gray value I_N is calculated as the average gray value of the neighboring pixels of the original image for a neighborhood radius r. For example, for neighborhood radius $r = 1$, it is [13]:

$$I_N(i,j) = \frac{\frac{I(i+1,j)+I(i+1,j+1)}{2} + I(i,j+1)}{2} \tag{3}$$

The scaling $(1 - \alpha_1)$ is used to scale I_N to ensure that watermarked image gray value I^* never exceeds the maximum gray value for 8-bit image representation corresponding to pure white pixel. The neighborhood radius determines the upper bound of the watermarked pixels in an image.

The first step of detection process is the generation of watermark W using the watermark key K. Next, the watermark is extracted from the test (watermarked) image using the detection function given below, for $\alpha_1 > 0$ and $\alpha_2 < 0$.

$$W^*(i,j) = \begin{cases} 1 \text{ if } I^*(i,j) - I_N(i,j) > 0 \\ 2 \text{ if } I^*(i,j) - I_N(i,j) < 0 \end{cases} \tag{4}$$

By comparing the original ternary watermark image W and the extracted binary watermark image W^*, the ownership can be established when the detection ratio is larger than a predefined threshold. The value of the threshold determines the minimum acceptable level of watermark detection.

3 Architectural Design of the Proposed Chip

In this section, the architecture of the invisible-robust watermarking encoder algorithm described in the previous section, is elaborated. We first provide high level description of the encoder, followed by their architectural details.

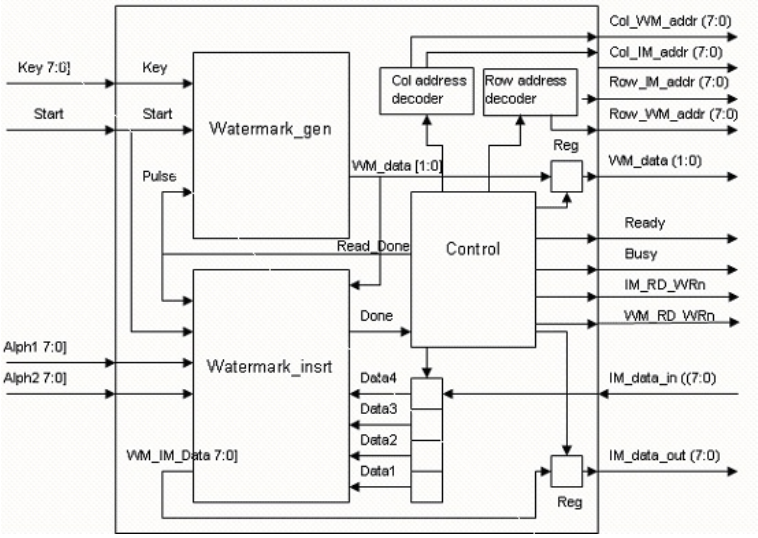


Fig. 1. Datapath and Controller for the Proposed Chip.

3.1 Datapath and Controller

The high-level view of the proposed chip is shown in Fig. 1. The encoder includes the units, such as watermark generation, watermark insertion, control, row and column address decoder, and registers. The generation unit is used to produce the watermark, and insertion unit is used to insert the watermark into the host image as per the described algorithm. The control unit controls the operation of the above two modules and the data flow in encoder. The address decoders are used to decode the memory address where the image and watermark are stored. The registers are used for buffering purpose. We assume that there are two external RAMs, one to store the original image and other to serve as a storage space for watermark data available. The watermarked image is written back to the RAM storing the original image.

3.2 Watermark Generation Unit

The ternary watermark is generated by pseudorandom sequence generator. The watermark generation unit consists of linear feedback shift register (LFSR). LFSR has a multitude of uses in digital system design and is a very crucial unit in watermark security and detection. It is a sequential shift register with combinational feedback logic around it that causes it to cycle pseudo randomly through a sequence of binary values. Therefore, we have studied the difficulties of a LFSR and have taken appropriate measures to ensure quality design [21–23]. The LFSR consists of flip-flops (FFs) as sequential elements with feedback loops. The feedback around a LFSR comes from a selected set of points called taps in the FF chain and these taps are fed back to FFs after either XORing or XNORing.

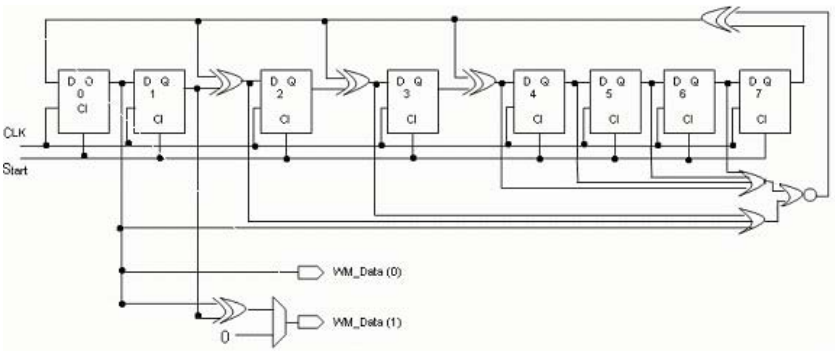


Fig. 2. Watermark Generation Unit: Linear Feedback Shift Register (LFSR).

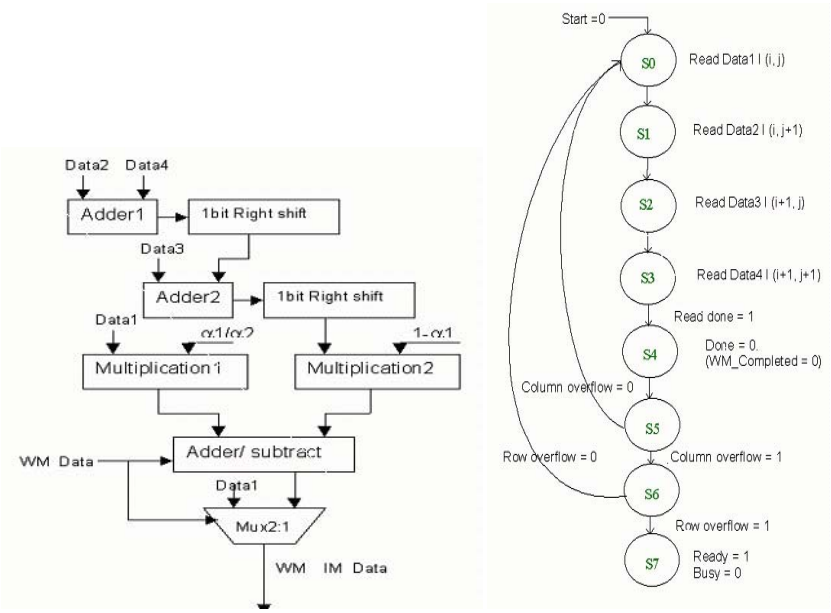
The design aspects considered when modeling LFSRs are as follows [21–23].

- *XOR or XNOR Feed Back Gates*: The feedback path may consist of either all XOR gates or all XNOR gates; LFSR will produce same number of values with different sequence for a particular tap setting.
- *One-to-Many or Many-to-One Feedback Structure*: Both one-to-many or many-to-one feedback structures can be implemented using same number of gates. However, a one-to-many feedback structure will have a shorter worst case delay.
- *Prohibited or Lockup State*: Special care should be placed on the design aspect such that LFSR avoids the prohibited or lockup state. In the case of XOR gates, the LFSR will not sequence through the binary value when all bits are at logic zero. Similarly, for XNOR gates the LFSR will not sequence through the binary values if all bits are at logic one. Thus, the LFSR should bypass these initializations during power up.
- *Ensuring a Sequence of All* $2n$ *Values*: If taps provided for a maximal length sequence are used, the LFSR configurations described so far will sequence through $(2n - 1)$ binary values. The feedback path can be modified with extra circuitry to ensure that all $2n$ binary values are included in the sequence.

Fig. 2 shows the LFSR we designed adopting the above discussed facts. The 8-bit LFSR is modeled so as to use one-to-many feedback structure and has been modified for a $2n$ looping sequence. It calculates and holds the next value of the LFSR which is then assigned to the output signal WM_DATA after each clock edge. The NOR of all LFSR bits minus the most significant bit that is LFSR_REG (6:0) generates the extra circuitry needed for all $2n$ sequence values.

3.3 Watermarking Insertion Unit

Fig. 3(a) shows the architecture of the watermark insertion unit designed to perform the watermarking insertion. The invisible-robust watermarking involves adding or subtracting a constant times the pixel value to be watermarked to or from a constant times the neighborhood function as described in the watermark encoder function in the previous section. The four data lines provide the pixels $I(i, j)$, $I(i + 1, j)$, $I(i, j + 1)$,



(a) Insertion Unit [13] (b) Control Unit

Fig. 3. Watermark Insertion Unit and Control Unit Structural Design.

and $I(i+1, j+1)$ for the row-column address pair (i, j). First, the $I(i, j+1)$ and $I(i+1, j+1)$ are given to the adder1 as input. Then, the resulting sum and carry out from adder1 are fed to the adder2 alongwith $I(i+1, j)$. The resulting sum of the adder2 is the neighborhood function value. The division by two is performed by shifting the results bit to the right by one bit, consequently discarding the rightmost bit (LSB). The scaling of the neighborhood function is achieved by multiplying it with $(1-\alpha_1)$ using the multiplier2. At the same time, the scaling of the image pixel gray values is performed in multiplier1 by multiplying $I(i, j)$ with α_1 or α_2. The eight high order bits of the multipliers are fed to the adder/subtract unit to perform watermark insertion. Since, we are concerned only with the integer values of the pixels, the lower eight bits of the multiplier results are discarded, which represent the values after the decimal point. The output of the adder/subtract unit(watermarked image pixels) and the original image pixel values are multiplexed based on the watermark values and are driven on to signal WM_IM_Data if the watermark value is "1" or "2" as per watermark encoding function in the previous section.

3.4 The Control Unit as a Finite State Machine (FSM)

Fig. 3(b) shows the control unit implemented as FSM. Following are the control signals: Start – active high signal used to activate all the modules, Alpha1 – 8-bit input scaling constants for watermark insertion algorithm, Alpha2 – 8-bit input scaling constants for watermark insertion algorithm, Key – 8-bit Digital watermark key. Following are the

output control signals: Ready – signal to indicate the insertion process is completed. Busy – signal to indicate the watermarking process is in progress.

The FSM has seven states as defined below. At each state certain events take place and the FSM moves to the next state on the next positive edge of the clock.

- S0: When the signal start is reset the control jumps to the state S0. In this state the $I(i, j)$ is read from the image RAM. The column and row addresses are registered in row_var and col_var.
- S1: The second data $I(i, j + 1)$ is read from the image RAM. In this state the column address is incremented to (Col_IM_addr = col_var + 1).
- S2: The data $I(i + 1, j)$ is read from the image RAM. The row address is increased to (row_IM_addr = row_var +1).
- S3: In this state fourth data $I(i + 1, j + 1)$ is read from the image RAM. The row address is incremented to (row_IM_addr = row_var +1). The column address is incremented to (Col_IM_addr = col_var + 1).
- S4: The signal Read_done is set, indicating that all the four pixels are read from image RAM for an address pair (i, j). The control will be in this state until done signal from the watermark insert module is set. The watermarked image pixel value and watermark pixel value are stored in respective RAM at address (i, j).
- S5: the column address is incremented (col_var = col_var + 1). In this state the control checks for the possibility of column overflow, i.e. the column address reached its right most pixel address or not. If col_var is equal to the right most address then the control moves to state S6 else to state S0.
- S6: the row address is incremented by (Row_var = Row_var + 1)and Col_var =0. In this state the control checks for row overflow, i.e. the row address reached its lower most pixel address or not. If Row_var is equal to the lower most address then the control moves to state S7, else to state S0.
- S7: In state S7, the busy signal is reset and ready signal is set indicating that the input image is watermarked.

4 Implementation, Simulation and Conclusions

The chip was modeled using VHDL and functional simulation was performed. The three modules created are watermark insertion, watermark generator, and watermark encoder. The watermark encoder is the main module which instantiates the other two components.The synthesis of the chip is carried out using Synplify Pro^{TM} tool targeting Xilinx VIRTEX-II technology with XCV50-BG256-6 target device.The simulations are done using the ModelSim. Fig. 4 shows the RTL schematic of the synthesized encoder. The timing simulation is presented in Fig. 5. From the synthesis results we provide the macro statistics and timing report of the units in Table 2. Minimum period is an indicator of the timing path from a clock to any other clock in the design. The minimum period is reported for both generation unit and encoder, whereas the critical path delay is reported for the insertion unit which is fully combinational. The cell usage indicates all the logical cells that are basic elements of the technology.

This paper presented an architecture and FPGA implementation of a watermarking encoder. Its low power high performance implementation is currently under progress.

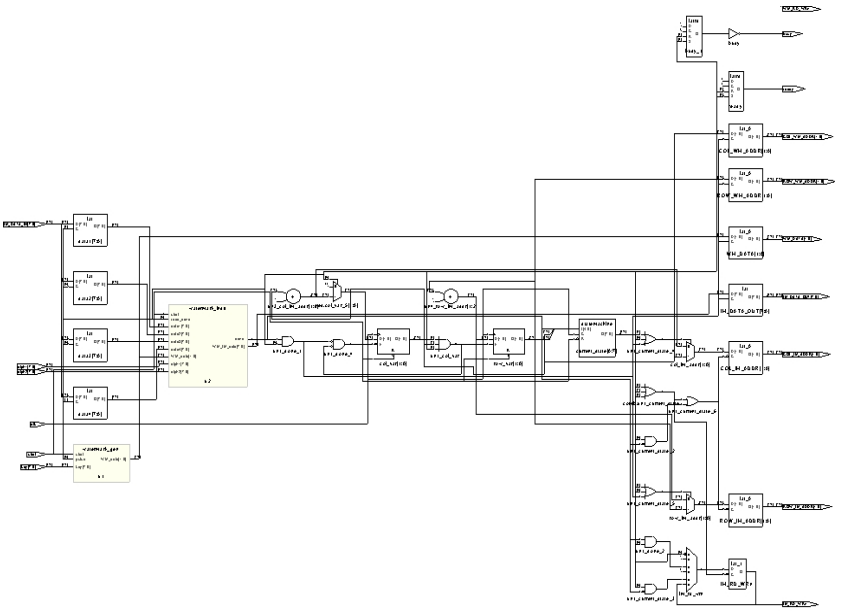


Fig. 4. Synthesis of the Encoder.

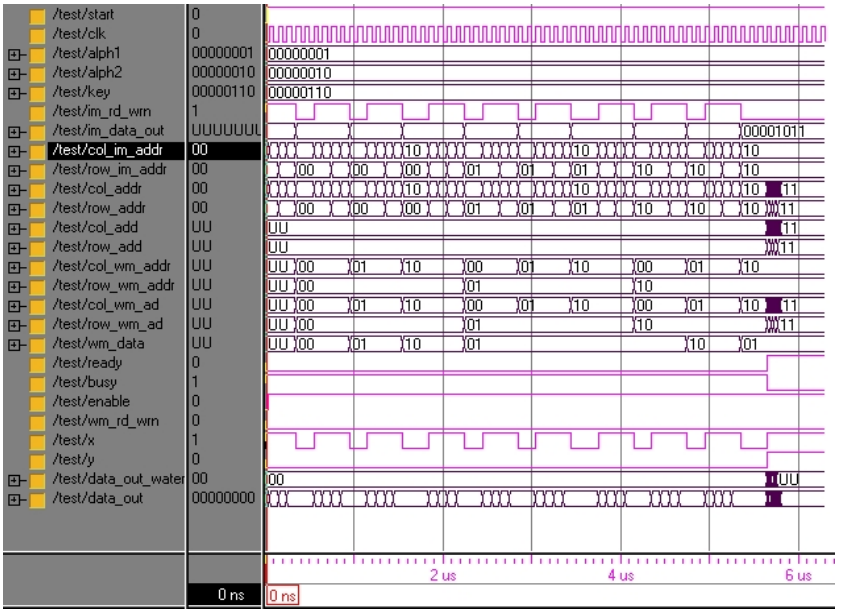


Fig. 5. Simulation Waveforms of the Encoder.

Table 2. Summary of Synthesis Report.

Units	Period/Delay (ns)	Cells Usage(BELS)
Watermark Generation	4.916	43
Watermark Insertion	15.526	122
Overall Encoder	19.842	838

The disadvantage of the watermarking algorithms implemented is that the processing needs to be done pixel-by-pixel. In future, we are aiming to investigate block-by-block processing. Since DRM systems need both encryption and watermarking, we think that combining both the hardware alongwith data compression hardware would be beneficial. Moreover, the on-chip encryptor can be used in storing the watermarking generator key in encrypted form, thus enhancing the watermark security.

References

1. Emmanuel, S., Kankanhalli, M.S.: A Digital Rights Management Scheme for Broadcast Video. ACM-Springer Verlag Multimedia Systems Journal **8** (2003) 444–458
2. Kundur, D., Karthik, K.: Digital Fingerprinting and Encryption Principles for Digital Rights Management. Proceedings of the IEEE **52** (2004)
3. Eskicioglu, A.M., Delp, E.J.: An Overview of Multimedia Content Protection in Consumer Electronics Devices. Elsevier Signal Processing: Image Comm. **16** (2001) 681–699
4. Memon, N., Wong, P.W.: Protecting Digital Media Content. Communications of the ACM **41** (1998) 35–43
5. Mohanty, S.P.: Digital Watamerking of Images. Master's thesis, Department of Electrical Engineering, Indian Institute of Science, Bangalore, India (1999)
6. Hsiao, S.F., Tai, Y.C., Chang, K.H.: VLSI Design of an Efficient Embedded Zerotree Wavelet Coder with Function of Digital Watermarking. IEEE Transactions on Consumer Electronics **46** (2000) 628–636
7. Maes, M., Kalker, T., Linnartz, J.P.M.G., Talstra, J., Depovere, G.F.G., Haitsma, J.: Digital Watamarking for DVD Video Copyright Protection. IEEE Signal Processing Magazine **17** (2000) 47–57
8. 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)
9. 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
10. 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
11. Mathai, N.J., Sheikholeslami, A., Kundur, D.: VLSI Implementation of a Real-Time Video Watermark Embedder and Detector. In: Proceedings of the IEEE International Symposisum on Circuits and Systems (Vol. 2). (2003) 772–775
12. Tsai, T.H., Wu, C.Y.: An Implementation of Configurable Digital Watermarking Systems in MPEG Video Encoder. In: Proc. of Intl. Conf. on Consumer Electronics. (2003) 216–217
13. Mohanty, S.P., Ranganathan, N., Namballa, R.K.: VLSI Implementation of Invisible Digital Watermarking Algorithms Towards the Developement of a Secure JPEG Encoder. In: Proceedings of the IEEE Workshop on Signal Processing Systems. (2003) 183–188

14. Seo, Y.H., Kim, D.W.: Real-Time Blind Watermarking Algorithm and its Hardware Implementation for Motion JPEG2000 Image Codec. In: Proceedings of the 1st Workshop on Embedded Systems for Real-Time Multimedia. (2003) 88–93
15. Mohanty, S.P., Rangnathan, N., Namballa, R.K.: VLSI Implementation of Visible Watermarking for a Secure Digital Still Camera Design. In: Proceedings of the 17th International Conference on VLSI Design. (2004) 1063–1068
16. Petitcolas, F.A.P., Anderson, R.J., Kuhn, M.G.: Information Hiding - A Survey. Proceedings of the IEEE **87** (1999) 1062–1078
17. Voloshynovskiy, S., Pereira, S., Pun, T., Eggers, J., Su, J.: Attacks on Digital Watermarks: Classification, Estimation-based Attacks and Benchmarks. IEEE Communications Magazine **39** (2001) 118–126
18. Mathai, N.J., Kundur, D., Sheikholeslami, A.: Hardware Implementation Perspectives of Digital Video Watermarking Algortithms. IEEE Transanctions on Signal Processing **51** (2003) 925–938
19. Tefas, A., Pitas, I.: Robust Spatial Image Watermarking Using Progressive Detection. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (Vol. 3). (2001) 1973–1976
20. Petitcolas, F.A.P.: Watermarking Schemes Evaluation. IEEE Signal Processing **17** (2000) 58–64
21. Nelson, V.P., Nagle, H.T., Irwin, 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)
23. Smith, M.J.S.: Application-Specific Integrated Circuits. Addison-Wesley Publishing Company, MA 01867, USA (1997)