International Journal of Sciences & Applied Research

www.ijsar.in

FPGA implementation of digital watermarking system

Tirunamala Krishnameghana*, I. A. Pasha

Dr. B. V. Raju Institute of Technology, Narsapur, Medak, Telangana State, India.

Correspondence Address: *Tirunamala Krishnameghana, Dr. B. V. Raju Institute of Technology, Narsapur, Medak, Telangana State, India.

Abstract

This paper presents a hardware implementation of a digital watermarking system that can insert invisible, semi fragile watermark information into multimedia content in real time. The watermark embedding is processed in the discrete cosine transform domain. To achieve high performance, the proposed system architecture employs fast discrete cosine transformation algorithm. Hardware implementation using field programmable gate array has been done, and an experiment was carried out using Spartan 3 FPGA for overall performance evaluation. Experimental results show that a hardware-based digital watermarking technique. The proposed water marking system is implemented using the Verilog hardware description language (HDL) synthesized into a field programming gate array (FPGA).

Keywords: Discrete cosine transformation, Watermark generation, Watermark Embedding, Spartan 3 board

Introduction

Digital watermarking is the process of embedding an additional, identifying information within a host multimedia object, such as text, audio, image, or video. 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.

A digital watermark is a digital signal or pattern inserted into a digital image. Since this signal or pattern is present in each unaltered copy of the original image, the digital watermark may also serve as a digital signature for the copies. A given watermark may be unique to each copy (e.g. to identify the intended recipient), or be common to multiple copies (e.g. to identify the document source). In either case, the watermarking of the document involves the transformation of the original into another form. This distinguishes digital watermarking from digital fingerprinting, where the original file remains intact and a new created file 'describes' the original file's content.

Watermarking is not a new phenomenon. In the modern era, providing authenticity is becoming increasingly important as most of the world’s information is stored as readily transferable bits. Digital watermarking is a process whereby arbitrary information is encoded into an image in such a way that the additional payload is imperceptible to the image observer. Watermarking algorithms

are divided into two categories. Spatial-domain techniques work with the pixel values directly. Frequency-domain techniques employ various transforms, either local or global.

Watermarks and watermarking techniques can be divided into various categories in various ways. Watermarking techniques can be divided into four categories according to the type of document to be watermarked such as Text Watermarking, Image Watermarking, Audio Watermarking, and Video Watermarking.

RELATED WORK ON WATERMARKING SYSTEMS

A. Robustness Level of Watermarking

The level of robustness of the watermarking can be categorized into three main divisions:

- Fragile
- Semi fragile
- Robust

A watermark is called fragile if it fails to be detectable after the slightest modification. A semi fragile watermark is the one that is able to withstand certain legitimate modifications, but cannot resist malicious transformations. A watermark is called robust if it resists a designated class of transformations.

In copyright protected applications, the attacker wishes to remove the WM without causing severe damage to the image. This can be done in various ways, including digital-to-analog and analog-to-digital conversions, cropping, scaling, segment removal, and others. Robust WM is used in these applications so that it remains detectable even after these attacks are applied, provided that the host image is not severely damaged. For image integrity applications, fragile watermarks are commonly used so that it can detect even the slightest change in the image. Most of the fragile WM methods perform the embedding of added information in the spatial domain.

Unlike the fragile WM techniques, a semi fragile invisible Watermark, such as that proposed in this paper, is designed to withstand certain legitimate manipulations, i.e., lossy compression, mild geometric changes of images, but is capable of rejecting malicious changes, i.e., cropping, segment removal, and so on. Furthermore, the semi fragile approaches are generally processed in the frequency domain.

Frequency-domain WM methods are more robust than the spatial-domain techniques. In practical video storage and distribution systems, video sequences are stored and transmitted in a compressed format, and during compression the image is transformed from spatial domain to frequency domain. Thus, a watermark that is embedded and detected directly in the compressed video stream can minimize computationally demanding operations. Therefore, working on compressed rather than uncompressed video is beneficial for practical WM applications.

B. Watermark Implementations-Hardware Versus Software

The implementation of watermarking could be on many platforms such as software, hardware, embedded controller, DSP, etc. System performance is a major parameter while designing complex systems. The standard DSP which has Von Neumann style of fetch operate-write back computation fails to exploit the inherent parallelism in the algorithm. For example, a 30 tap FIR filter implemented on a DSP microprocessor would require 30 MAC (Multiply Accumulate) cycles for advancing one unit of real-time. Further, each MAC operation may consist of more than one cycle as it involves a memory fetch, the multiply accumulate operation, and the memory write back. In contrast, a hardware implementation can store the data in registers and perform the 30 MAC operations in parallel over a single cycle. Thus, high throughput requirements of real-

time digital systems often dictate hardware intensive solutions.

FPGAs provide a rapid prototyping platform. They can be reprogrammed to achieve different functionalities without incurring the non-recurring engineering costs typically associated with custom IC fabrication. For commercial applications like movie production, video recording, real on-spot video surveillance, where a real-time response is always required, so a software solution is not recommended due to its long time delay. Since the goal of this research is a high performance encoding watermarking unit in an integrated circuit (IC) for commercial applications, and since FPGAs (field programmable gate arrays) have advantages in both fast processing speed and field programmability, it was determined that an FPGA is the best approach to build a fast prototyping module for verifying design concepts and performance.

Several software implementations of the watermarking algorithms are available, but very few attempts have been made for hardware implementations. Software implementation of watermarking has been implemented because of their ease of use and flexibility. Mostly software based watermarking works on offline where images are captured through camera and stored on computer and the software for watermarking runs and embeds the watermark and then the images are distributed. This approach has the drawback of certain amount of delay, once images are captured and then watermark is embedded. If attackers would attacks the image before the watermark embedded then it creates issues for ownership of the originator. So there is a need of real-time watermarking where watermark embedding unit reside inside the device (as digital camera) and embedding done directly when image is captured. The hardware implementation of watermarking has advantages in terms of reliability and high performance for area, power and speed. This is very much crucial in some applications like real-time broad casting, video authentication and secure camera system for courtroom evidence. The hardware implementation can have advantage of parallel processing. Since watermarking process deals with processing of watermark and pre-processing of original content before embedding watermark. These two processes are independent and can work in parallel to achieve parallelism to achieve high speed for real-time application.

C. Existing Work on Video Watermarking

In the past few years, research effort has been focused on efficient WM systems implementation using hardware platforms. For example, Strycker *et al.* [12] proposed a well known video WM scheme, called just another watermarking system (JAWS), for TV broadcast monitoring and implemented the system on a Philips's Trimedia TM-1000 very long instruction word (VLIW) processor. The experimental results proved the feasibility of WM in a professional TV broadcast monitoring system. Mathai *et al.* [13], [14] presented an application-specific integrated circuits (ASIC) implementation of the JAWS WM algorithm using 1.8 V, 0.18-μm complementary metal oxide semiconductor technology for real-time video stream embedding. With a core area of 3.53 mm2 and an operating frequency of 75 MHz, that chip implemented watermarking of raw digital video streams at a peak pixel rate of over 3 Mpixels/s while consuming only 60mW power.A new real-time WM very large scale integration (VLSI) architecture for spatial and transform domain was presented by Tsai and Wu [15]. Maes *et al.* [16] presented the millennium watermarking system for copyright protection of DVD video and some specific issues, such as watermark detector location and copy generation control, were also addressed in their work. An FPGA prototype was presented for HAAR-wavelet-based real-time video watermarking by Jeong *et al.*

[38]. A real-time video watermarking system using DSP and VLIW processors was presented in [39], which embeds the watermark using fractal approximation by Petitjean *et al.* Mohanty *et al.* [40] presented a concept of secure digital camera 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 [41] by the same group.

PROCEDURE FOR DIGITAL WATERMARKING SYSTEM

A. Overview of Proposed watermarking system

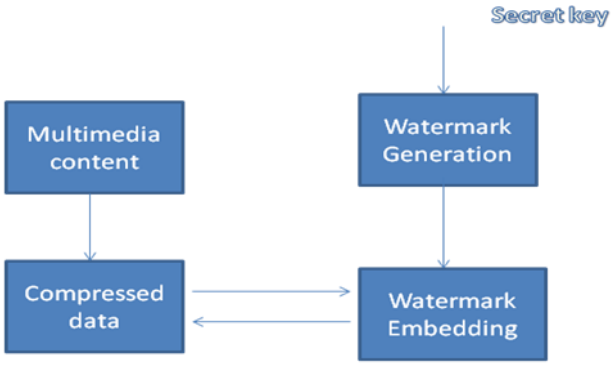

Fig. 1: Overview of Proposed watermarking system.

Fig. 1 illustrates the general block diagram of the proposed system that is comprised of four modules. The watermark embedding approach is designed to be performed in the DCT domain. Embedding the watermark after quantization makes the watermark robust to the DCT compression with a quantization of equal or lower degree used during the watermarking process. Then, they are passed to the watermark embedding module. The watermark generation unit produces a specific watermark data for each input, based on initial predefined secret keys. The watermark embedding module inserts the watermark data into the quantized DCT coefficients according to the algorithm. Finally, watermarked DCT coefficients of each input are encoded by the compression unit which outputs the compressed data with embedded authentication watermark data. The proposed watermarking system is implemented using the Verilog hardware description language (HDL) synthesized into a field programming gate array (FPGA).

B. Compression technique

The technique used for compression is fast discrete cosine transformation. In particular, a DCT is a Fourier-related transform similar to the discrete Fourier transform (DFT), but using only real numbers. DCTs are equivalent to DFTs of roughly twice the length, operating on real data with even symmetry (since the Fourier transform of a real and even function is real and even), where in some variants the input and/or output data are shifted by half a sample. There are eight standard DCT variants, of which four are common.

The most common DCT definition of a 1-D sequence of length N is

$$C(u)=\alpha(u)\sum_{x=0}^{N-1} f(x)\cos\left[\frac{\pi(2x+1)u}{2N}\right] \quad (1)$$

For u = 0, 1… N-1 .Similarly inverse transformation can be defined as

$$f(x)=\sum_{u=0}^{N-1}\alpha(u)C(u)\cos\left[\frac{\pi(2x+1)u}{2N}\right] \quad (2)$$

In order to implement fast DCT, equation (1) can be divided into even and odd terms. The following is the flow diagram for 8 point DCT:

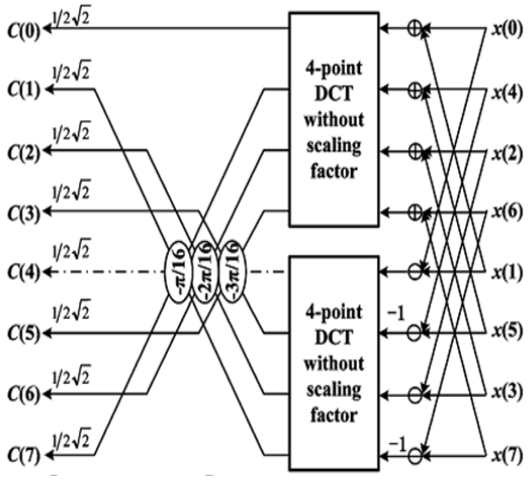

Fig. 2: Flow diagram of 8 point Fast DCT.

C. *Watermark Generation*

Since simple watermark data can be easily cracked, it is essential that the primitive watermark sequence will be encoded by an encipher. This insures that the primitive watermark data are secured before being embedded into each video frame. Currently, there are different approaches to convert a primitive watermark into a secured pattern. Contradictory to existing solution approaches, a novel video watermark generator is proposed. The WM generator generates a secure watermark sequence for each video frame using a meaningful primitive watermark sequence and secret input keys.

According to the recommendation by Dittman in [22] for the feature of a video watermark, a primitive watermark pattern can be defined as a meaningful identifying sequence for each video frame. The unique meaningful watermark data for each video frame contain the time, date, camera ID, and frame serial number (that is related to its creation). This will establish a unique relationship of the video stream frames with the time instant, the specific video camera, and the frame number. Any manipulation, such as frame exchange, cut, and substitution, will be detected by the specific watermark. The corresponding N-bit (8-bit) binary valued pattern, $a_{i,}$, will be used as a primitive watermark sequence. This would generate a different watermark for every frame (time-varying) because of the instantaneously changing serial number and time.

The block diagram of the proposed novel watermark generator is depicted in Fig. 3. A secure watermark pattern is generated by performing expanding, scrambling, and modulation on a primitive watermark sequence. There are two digital secret keys: Key 1 is used for scrambling and Key 2 is used for the random number generator (RNG) module that generates a pseudorandom sequence.

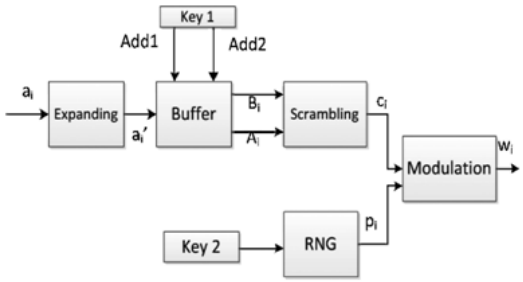

Fig. 3: Block diagram of watermark generator.

D. Watermark Embedding

The watermarking algorithm should be hardware friendly in a way that it can be implemented in hardware with high throughput. For this purpose, one concern for the algorithm development should be that it must support pipelining architecture so that two or more macro blocks inside a single video frame or more than one frame can be watermarked simultaneously. This feature will aid in increasing the processing speed of watermarking.

The watermark embedding approach used in this paper was originally developed by Nelson and Shoshan in [25].This WM algorithm, capable of inserting a semi fragile invisible watermark in a compressed image in the DCT frequency domain, was modified and then applied in watermarking of a video stream. In general, for each DCT block of a

video frame, *N* cells need to be identified as "watermarkable" and modulated by the watermark sequence. The chosen cells contain nonzero DCT coefficient values and are found in the mid-frequency range. This algorithm was detailed by Shoshan in [25]. The algorithm can be described as follows.

1) For each 8 × 8 block (watermark data), perform DCT, quantization, and zigzag scan to generate quantized DCT coefficients.Identify *N* watermarkable cells for each block and calculate the modification value for each selected cell.
2) Modify the identified watermarkable DCT coefficients according to the modification values.
3) Perform inverse DCT and inverse quantization for each 8 × 8 block watermarked coefficient to reconstruct the original I pixel values.
4) Generate compressed and watermark embedded data.

HARDWARE ARCHITECTURE DESIGN

A. Watermark Generator

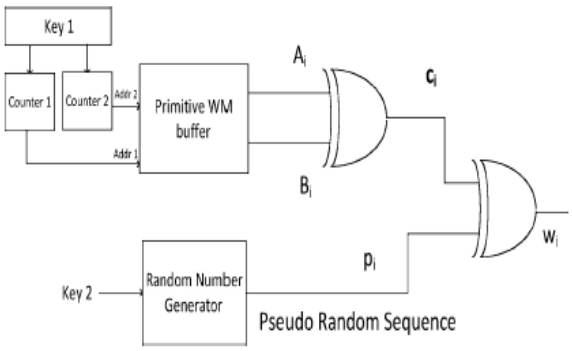

Fig. 4: Hardware architecture of watermark generator module.

Fig. 4 describes the hardware architecture of the novel watermark generator. The expanding procedure is accomplished by providing consecutive clock signal so that an 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 pseudorandom sequence p_i, generated by the proposed Gollman[14] cascade filtered feedback with carry shift register RNG[23], 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.

B. Watermark Embedder

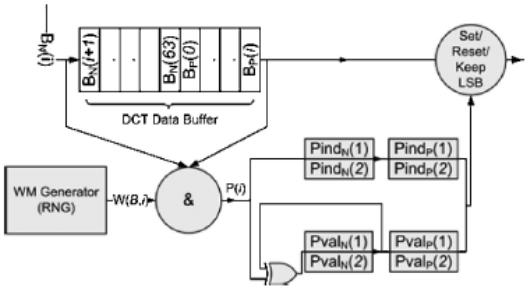

Fig. 5: Hardware architecture of watermark embedded module.

A schematic view of the hardware architecture for the watermark embedder unit is presented in Fig. 5. As described by Shoshan [25], 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 8 × 8 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. 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 [25].

C. Spartan 3 FPGA board

The Spartan-3 family of Field-Programmable Gate Arrays is specifically designed to meet the needs of high volume, cost-sensitive consumer electronic applications. The eight-member family offers densities ranging from 50,000 to five million system gates. The FPGASP3 is a low-priced, compact prototyping module that can be used for rapid proof of concept or for educational environments. The module is based on the Spartan 3 FPGA from Xilinx along with supporting circuitry to ease prototyping efforts. Designers can use the Spartan-3 kit for general FPGA prototyping, experimenting with multiple FPGA configuration techniques, and proving out low cost design methods.

Fig. 6: FPGA spartan hardware kit.

The following are the key features of Spartan 3 FPGA:

1) 8 Nos. Point LEDs (Logic Output)
2) 8 Nos. Digital Input (DIP Switch)
3) One UART (RS232)
4) VGA Interface
5) Reset Button | Power-on Indication
6) PS/2 (Keyboard Interface)
7) 40-Pin Expansion Connector
8) JTAG (Program/Debug)
9) 3 Nos. 20pin- I/O Expansion Connector
10) On-Board Voltage regulators +5V | +3V3 | 2V5 | 1V2

EXPERIMENTAL RESULTS

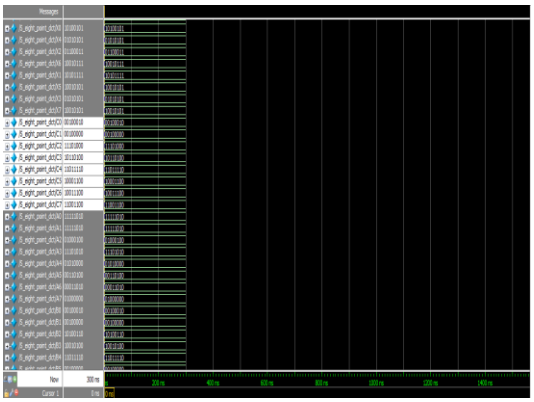

Fig. 6: DCT result.

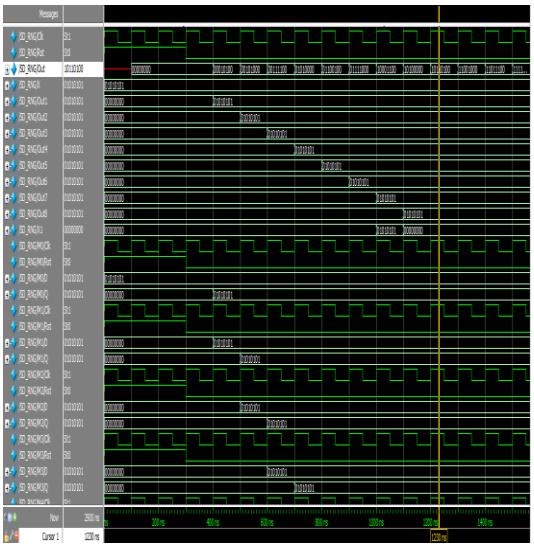

Fig. 7: RNG Output.

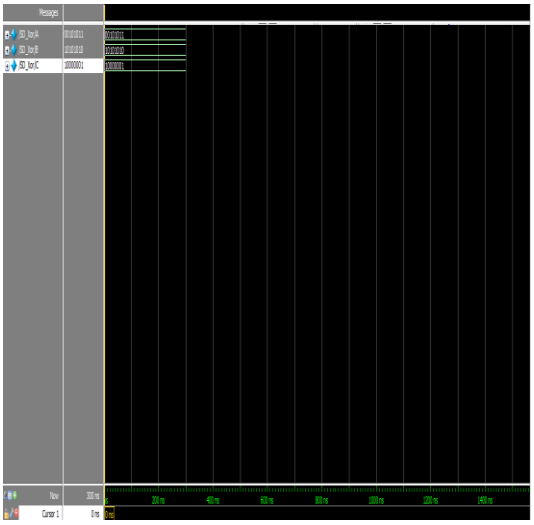

Fig. 8: XOR result.

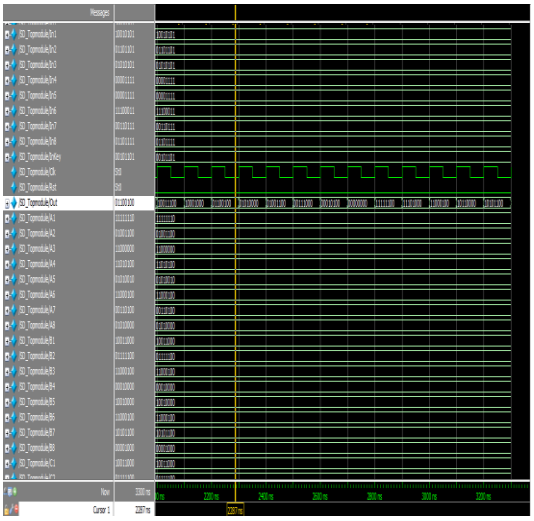

Fig. 9: Watermark Generation module result.

Conclusion

A. Summary and Conclusion

Current work implementation is the Digital Watermarking system with fast discrete cosine transformation algorithm using Spartan3 kit. Overall, this project is all about implementing digital water marking system on hardware of low cost and consumes low power and more reliable.

B. Future Scope

This work can be extended by using different FPGA and usage of modern algorithms for compression to make the Watermarking technique robust.

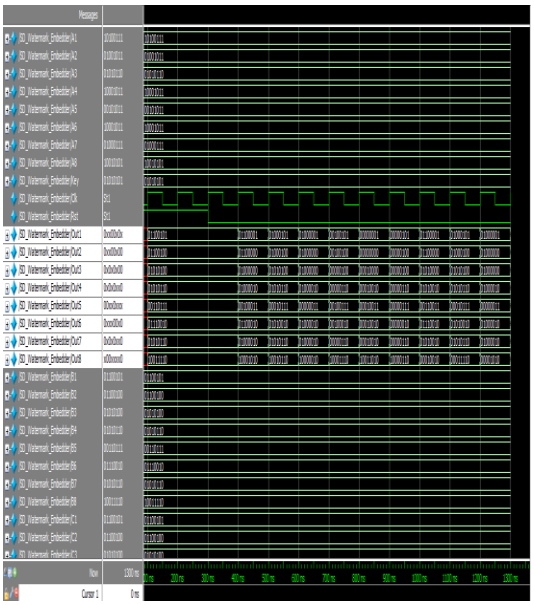

Fig. 10: Watermark Embedded module result.

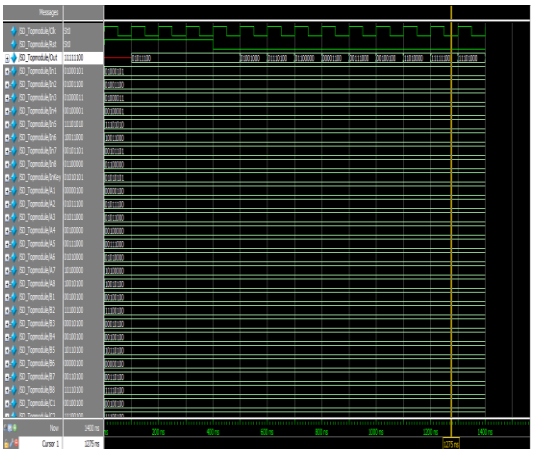

Fig. 11: FPGA result.

Table 1: Spartan 3 (XC3S200) FPGA Attributes.

Device	System Gates	Logic Cells	CLB Array (OneCLB = Four Slices)			Distributed RAM (bits1)	Block RAM (bits1)	Dedicated Multiplies	DCMs	Maximum User I/O	Maximum Differential I/O Pairs
			Rows	Columns	Total CLBs						
XC3S200	200K	4,320	24	20	480	30K	216K	12	4	173	76

Table 2: Ordering Information of XC3S200.

Device	Speed Grade		Package Type / Number of Pins		Temperature Range (TJ)	
XC3S200	-5	High Performance	TQ144	144-pin Thin Quad Flat Pack (TQFP)	I	Industrial (–40°C to 100°C)

Table 3: FPGA Synthesis Report.

Research Works	Design Type	Type of WM	Processing Domain/Method	Logic cells (Kilo gates)	Power Consumption
Proposed implementation	FPGA(Spartan 3)	Semi Fragile	8 point FAST DCT Architecture	10.431	56mW

References

[1] V. M. Potdar, S. Han, and E. Chang, "A survey of digital image watermarking techniques," in *Proc. IEEE Int. Conf. Ind. Informatics*, Aug. 2005, pp. 709–716.

[2] 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.

[3] A. Piva, F. Bartolini, and M. Barni, "Managing copyright in open networks," *IEEE Trans. Internet Comput.*, vol. 6, no. 3, pp. 18–26, May–Jun. 2002.

[4] Y. Shoshan, A. Fish, X. Li, G. A. Jullien, and O. Yadid-Pecht, "VLSI watermark implementations and applications," *Int. J. Information Technol. Knowl.*, vol. 2, no. 4 pp. 379–386, Jun. 2008.

[5] X. Li, Y. Shoshan, A. Fish, G. A. Jullien, and O. Yadid-Pecht, "Hardware implementations of video watermarking," in *International Book Series on Information Science and Computing*, no. 5. Sofia, Bulgaria: Inst. Inform. Theories Applicat. FOI ITHEA, Jun. 2008, pp. 9–16 (supplement to the *Int. J. Inform. Technol. Knowledge*, vol. 2, 2008).

[6] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Process.*, vol. 6, no. 12, pp. 1673–1687, Dec. 1997.

[7] S. P. Mohanty. (1999). *Digital Watermarking: A Tutorial Review* [Online]. Available: http://www.linkpdf.com/download/dl/digital-watermarking-a-tutorial-review-.pdf

[8] F. Hartung and B. Girod, "Watermarking of uncompressed and compressed video," *IEEE Trans. Signal Process.*, vol. 66, no. 3, pp. 283–302, May 1998.

[9] T. L. Wu and S. F. Wu, "Selective encryption and watermarking of MPEG video," in *Proc. Int. Conf. Image Sci. Syst. Technol. (CISST)*, Jun. 1997, pp. 0–9.

[10] A. Shan and E. Salari, "Real-time digital video watermarking," in *Proc. Dig. Tech. Papers: Int. Conf. Consumer Electron.*, Jun. 2002, pp. 12–13.

[11] L. Qiao and K. Nahrstedt, "Watermarking methods for MPEG encoded video: Toward resolving rightful ownership," in *Proc. IEEE Int. Conf. Multimedia Comput. Syst.*, Jun. 1998, pp. 276–285.

[12] L. D. Strycker, P. Termont, J. Vandewege, J. Haitsma, A. Kalker, M. Maes, and G. Depovere, "Implementation of a real-time digital watermarking process for broadcast monitoring on Trimedia VLIW processor," *Proc. Inst. Elect. Eng. Vision, Image Signal Process.*, vol. 147, no. 4, pp. 371–376, Aug. 2000.

[13] N. J. Mathai, A. Sheikholesami, and D. Kundur, "Hardware implementation perspectives of digital video watermarking algorithms," *IEEE Trans. Signal Process.*, vol. 51, no. 4, pp. 925–938, Apr. 2003.

[14] N. J. Mathai, A. Sheikholesami, and D. Kundur, "VLSI implementation of a real-time video watermark embedder and detector," in *Proc. Int. Symp. Circuits Syst.*, vol. 2. May 2003, pp. 772–775.

[15] T. H. Tsai and C. Y. Wu, "An implementation of configurable digital watermarking systems in MPEG video encoder," in *Proc. Int. Conf. Consumer Electron.*, Jun. 2003, pp. 216–217.

[16] M. Maes, T. Kalker, J. P. Linnartz, J. Talstra, G. Depoyere, and J. Haitsma, "Digital watermarking for DVD video copy protection," *IEEE Signal Process. Mag.*, vol. 17, no. 5, pp. 47–57, Sep. 2000.

[17] X. Wu, J. Hu, Z. Gu, and J. Huang, "A secure semifragile watermarking for image authentication based on integer wavelet transform with parameters," in *Proc. Australian Workshop Grid Comput. E-Research*, vol. 44. 2005, pp. 75–80.

[18] *Information Technology: Generic Coding of Moving Pictures and Associated Audio Information*, ISO/IEC

13818-2:1996(E), Video International Standard, 1996.

[19] K. Jack, *Video Demystified: A Handbook for the Digital Engineer*, 2nd ed. Eagle Rock, VA: LLH Technology Publishing, 2001.

[20] I. E. G. Richardson, *H.264 and MPEG-4 Video Compression*. Chichester, U.K.: Wiley, 2003.

[21] F. Bartolini, M. Barni, A. Tefas, and I. Pitas, "Image authentication techniques for surveillance applications," *Proc. IEEE*, vol. 89, no. 10, pp. 1403–1418, Oct. 2001.

[22] J. Dittmann, T. Fiebig, R. Steinmetz, S. Fischer, and I. Rimac, "Combined video and audio watermarking: Embedding content information in multimedia data," in *Proc. SPIE Security Watermarking Multimedia Contents II*, vol. 3971. Jan. 2000, pp. 455–464.

[23] X. Li, Y. Shoshan, A. Fish, and G. A. Jullien, "A simplified approach for designing secure random number generators in HW," in *Proc. IEEE Int. Conf. Electron. Circuits Syst.*, Aug. 2008, pp. 372–375.

[24] G. R. Nelson, G. A. Jullien, and O. Yadid-Pecht, "CMOS image sensor with watermarking capabilities," in *Proc. IEEE Int. Symp. Circuits Syst.* vol. 5. May 2005, pp. 5326–5329.

[25] Y. Shoshan, A. Fish, G. A. Jullien, and O. Yadid-Pecht, "Hardware implementation of a DCT watermark for CMOS image sensors," in *Proc. IEEE Int. Conf. Electron. Circuits Syst.*, Aug. 2008, pp. 368–371.

[26] Theora.org. (2012, Mar. 8) [Online]. Available: http://www.theora.org/

[27] A. Filippov, "Encoding high-resolution Ogg/Theora video with reconfigurable FPGAs," *Xcell J.* (Plugging into High-Volume Consumer Products), no. 53, pp. 19–21, 2005 [Online]. Available: http://www.xilinx.com/publications/archives/xcell/Xcell53.pdf

[28] (2012, Mar. 8) [Online]. Available: http://www.videolan.org/vlc/features.php?cat=video

[29] (2011, Apr. 25) [Online]. Available: http://www.mplayerhq.hu/design7/info.html

[30] (2012, Jan. 13) [Online]. Available: http://wiki.xiph.org/index.php/TheoraSoftwarePlayers

[31] B. Schneier, *Applied Cryptography*, 2nd ed. New York: Wiley, 1996.

[32] V. M. Potdar, S. Han, and E. Chang, "A survey of digital image watermarking techniques," in *Proc. 3rd IEEE Int. Conf. Ind. Informatics*, 2005, pp. 709–716.

[33] F. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding: A survey," *Proc. IEEE*, vol. 87, no. 7, pp. 1062–1078, 1999.

[34] F. Arnault, T. Berger, and A. Necer, "A new class of stream ciphers combining LFSR and FCSR architectures," in *Proc. Adv. Cryptology INDOCRYPT*, LNCS 2551. 2002, pp. 22–33.

[35] NIST. (2012, Jun. 19). *A Statistical Test Suite for the Validation of Random Number Generators and Pseudo Random Number Generators for Cryptographic Applications* [Online]. Available: http://csrc.nist.gov/groups/ST/toolkit/rng/documentation−software.html

[36] (2012, Jul. 28) [Online]. Available: http://en.wikipedia.org/wiki/Digital−watermarking

[37] (2012, Apr. 16) [Online]. Available: http://en.wikipedia.org/wiki/Motion−JPEG

[38] Y.-J. Jeong, K.-S. Moon, and J.-N. Kim, "Implementation of real time video watermark embedder based on Haar wavelet transform using FPGA," in *Proc. 2nd Int. Conf. Future Generation Commun. Networking Symp.*, 2008, pp. 63–66.

[39] G. Petitjean, J. L. Dugelay, S. Gabriele, C. Rey, and J. Nicolai, "Towards realtime video watermarking for systems-on-chip," in *Proc. IEEE Int. Conf.*

Multimedia Expo, vol. 1. 2002, pp. 597–600.

[40] S. P. Mohanty, "A secure digital camera architecture for integrated realtime digital rights management," *J. Syst. Architecture*, vol. 55, nos. 10–12, pp. 468–480, Oct.–Dec. 2009.

[41] S. P. Mohanty and E. Kougianos, "Real-time perceptual watermarking architectures for video broadcasting," *J. Syst. Softw.*, vol. 84, no. 5, pp. 724–738, May 2011.

[42] S. Saha, D. Bhattacharyya, and S. K. Bandyopadhyay, "Security on fragile and semifragile watermarks authentication," *Int. J. Comput. Applicat.*, vol. 3, no. 4, pp. 23–27, Jun. 2010.