IJESRT
INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

HARDWARE CHIP IMPLEMENTATION OF DIGITAL WATERMARKING USING IMDCT INVERSE MODIFIED DISCRETE COSINE TRANSFORM (IMDCT) IN HARDWARE DESCRIPTION LANGUAGE (HDL) ENVIRONMENT

Babloo Kumar*, Amit Kumar Chauhan, Sachin Sangal
Assistant Professor, Department of Computer Science & Engineering S.D College of Engineering & Technology, Muzaffarnagar, India.
Assistant Professor, Department of Electronics and Communication Engineering S.D College of Engineering & Technology, Muzaffarnagar, India.
Assistant Professor, Department of Electronics and Communication Engineering S.D College of Engineering & Technology, Muzaffarnagar, India.

ABSTRACT

Digital watermarking is the process of embedding data, called a watermark, into a multimedia object such that the watermark can be detected whenever needed for digital rights management (DRM). The object may be an image, audio, video, text, or graphics. In general, any watermarking algorithm consists of three parts: the water mark, the encoder (insertion algorithm), and the decoder and comparator (verification or extraction or detection algorithm). An entity, called the watermark key, which is unique and exhibits a one-to-one correspondence with every water mark, is also used during the process of embedding and detecting the watermark. The key is private and known only to authorized parties, eliminating the possibility of illegal usage of digital content. Different watermarking techniques have already been evolved in the field of digital image processing. Because of copyright protection, watermarking techniques are often evaluated based on their robustness, recoverability, and invisibility. Field programmable gate arrays (FPGAs) are extensively used in rapid prototyping and verification of a conceptual design and also used in electronic systems when the mask-production of a custom IC becomes prohibitively expensive due to the small quantity. In addition to their usefulness as mentioned above, their internal structure also makes them as a suitable vehicle to learn all aspects of VLSI design because they consist of combinational logic in the form of LUT (look up table), flip-flops as sequential building blocks, and memory for programmability. Speed and size are two important factors while designing any system. It's Speed of operation and flexibility to modify, measures the performance of the system operation. Traffic handling capacity is an important element of service quality and will therefore play a basic role in this choice Microprocessor/microcontroller (MPMC) system can handle sequential operations with high flexibility and use of Field Programmable Gate Array (FPGA) can handle concurrent operations with high speed in small size area. So combined features of both these systems can enhance the performance of the system. Objective is completed in two phases. First phase is designing and second phase is functional simulation and synthesis. Xilinx and Modelsim Simulation tools are proposed to design and verify speed improvement. In our research we considered Inverse Modified Discrete cosine transform (IMDCT) for digital watermarking and its hardware chip implementation with optimized hardware parameters.

KEYWORDS— VHDL- Very High Speed Integrated Circuit Hardware Description language
VLSI-Very Large Scale of Integration
ASIC- Application Specific Integrated Circuits
FPGA- Field Programmable Gate Array
RTL - Register Transfer Logic
IMDCT - Inverse Modified Discrete cosine Transform
DWT - Discrete wavelet Transform

LUT – Look up table
IOB – Input/ Output block.

INTRODUCTION (BAYESIAN TECHNIQUE)

Watermark Insertion involves the process of adding the watermark message. Watermark message WM(x, y) may be a random or pseudo random signal, a binary {-1, +1} or {-1, 0, +1} signal which are added linearly [9]. Watermarking techniques can be broadly classified into different categories based on two criteria's, they are transform domain techniques involves Discrete Fourier Transform (DFT) domain, Discrete Cosine Transform (DCT) domain and Discrete Wavelet Transform Domain (DWT) and spatial domain techniques [1]. Comparatively transform domain watermarking techniques are more robust than spatial domain techniques. My thesis includes only transform domain watermarking techniques. Among all the transform domain watermarking techniques, DCT technique provides more invisibility,[5] fragile and robustness to the retrieved watermark image for copyright protection and data hiding. In DCT each image frequency bands may have property of redundancy but they are redundant in different manner [6, 1]. To remove the redundancy in the image we are going for image compression algorithms, lossy and lossless. In my work we are going for lossy compression algorithm for compressing redundancy by removing irrelevancy [6, 7]. There are several image compression algorithms for both gray images and color images. In this, a bit-plane slicing technique (BPS) involving the bit-level image compression algorithm, to compress gray level image by taking a gray scale image, separating into bit plane then compressing by bit-level image. A significant number of watermarking techniques have been reported in recent years in order to create robust and imperceptible audio watermarks. We proposed a method of embedding watermarks into audio signals in the time domain. The proposed algorithm exploits differential average-of-absolute-amplitude relations within each group of audio samples to represent one-bit information. It also utilizes the low-frequency amplitude modification technique to scale the amplitudes in selected sections of samples so that the time domain waveform envelope can be almost preserved. The strength of the audio signal modifications is limited by the necessity to produce an output signal for watermark detection. The watermark signal is generated using a key, and watermark insertion depends on the amplitude and frequency of audio signal that minimizes the audibility of the watermarked signal [12].

Digital watermarking is a process of embedding data (watermark) into a multimedia object to help to protect the owner's right to that object. The embedded data may be either visible or invisible. In visible watermarking of images, a secondary image (the watermark) is embedded in a primary (host) image such that watermark is intentionally perceptible to a human observer whereas in the case of invisible watermarking the embedded data is not perceptible, but may be extracted/detected by a computer program. Information can be hidden by different ways in images. To hide information, straight message insertion may encode every bit of information in the image or selectively embed the message in "noisy" areas that draw less attention, those areas where there is a great deal of natural color variation. The message may also be scattered randomly throughout the image. A number of ways exist to hide information in digital images. Common approaches include [1]

- Least significant bit insertion,
- Masking and filtering,
- Algorithms and transformations.

Each of these techniques can be applied, with varying degrees of success, to different image files. A large number of watermarking algorithms work with some form of unitary transformation of the image of interest as for example the Discrete Wavelet, or Discrete Fourier, or Discrete Cosine Transform (DWT, DFT, and DCT) [19]. This is because transform domain techniques offer various advantages. For example, by modifying only the spatial frequency bands human beings are not very sensitive to, a watermark embedded in an image can be less visible. Some of the desired characteristics of visible water-marks are listed below.

[1] The watermark should be visible yet must not significantly obscure the image details beneath it.
[2] The watermark must be difficult to remove; removing a watermark should be more costly and labor intensive than purchasing the image.

[3] The information carried by the watermark is robust to content manipulations, compression, and soon.
[4] The watermark should be applied automatically with little human intervention and labor.

In multicast application scenario like VoD, a video server sends out a networked high quality visual program to its users is shown in fig 1. In this case, authenticating all the outgoing video streams with user specific information exhaust the video server. Note that the same video content is watermarked at the source (server side) with their recipient specific information for video finger printing and traitor tracing .Each recipient will receive the embedded video stream having their own information as a watermark. Multimedia fingerprinting and traitor tracing and respective attacks are presented in .By considering the insecure nature of the channel or network, there is highly possible that the video stream under goes counterfeiting or malicious attacks. In such case, it is difficult for the receiver to determine the genuineness and originality of the content without any authentication mechanism between the parties involved in exchanging the streams. Streaming research for authentication has been motivated in finding ways to overcome the limitations.

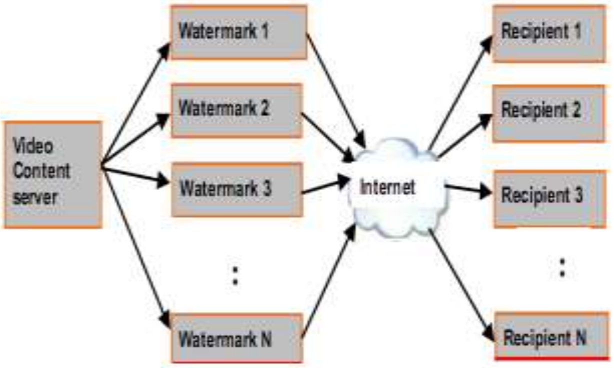

Figure 1 Multicast Video Streaming over internet

Watermarking in real time will solve the source authentication issues. The parties involved in real time stream exchange, checks the authenticity of the data received, by extracting the watermark bits embedded in the stream. This watermark can be introduced into the video stream at source, channel or at the receiver side. In our work, we propose a simple video streaming authentication system using watermarking at the source principle rather than at video delivery or at channel. The proposed system is suitable both for unicast and multicasting application. Unique Device DNA (Identification number of the video encoder chip) or user defined payload of 64-bit length is used as a watermark. A line based watermark embedding technique in which, a single line is embedded with the entire payload. If the payload of the watermark is less, then the same watermark pattern is repeated for the complete line of the video frame.

Tools Utilized: Design and implementation includes the following software development tools: Project navigator Application ISE 13.2 of Xilinx Company is a tool to design the IC and to view their RTL (Register Transfer Logic) schematic. Model SimEE 10.1b students edition of Mentor Graphics Company is used for simulation and debugging the functionality. The hardware chip implementation is done using VHDL programming language.

The paper is organized as follows: Section I presents the introduction and the tools utilized. Section II discusses the mathematical model of DCT. Section III presents the implementation of IMDCT. IV describes the result and performance evaluation. Section V presents the Device utilization and timing summary and Conclusions and future scope is presented in Section VI.

MATHEMATICAL MODEL OF DISCRETE COSINE TRANSFORM (DCT)

Formally, the discrete cosine transform is a linear, invertible function $F : \mathrm{R}^N \rightarrow \mathrm{R}^N$ (where R denotes the set of real numbers), or equivalently an invertible $N \times N$ square matrix. There are several variants of the DCT with slightly modified definitions. The N real numbers $x_0, ..., x_{N-1}$ are transformed into the N real numbers $X_0, ..., X_{N-1}$ according to one of the formulas:

DCT-I

$$X_k = \frac{1}{2}\left(x_0 + (-1)^k x_{N-1}\right) + \sum_{n=1}^{N-2} x_n \cos\left[\frac{\pi}{N-1}\, nk\right]$$

$$where\ k = 0, \dots., N-1$$

DCT-II

$$X_k = \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right]$$

$$where\ k = 0, \dots., N-1$$

DCT-III

$$X_k = \frac{1}{2}x_0 + \sum_{n=1}^{N-1} x_n \cos\left[\frac{\pi}{N} n\left(k + \frac{1}{2}\right)\right]$$

$$where\ k = 0, \dots., N-1$$

DCT-IV

$$X_k = \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}\left(n + \frac{1}{2}\right)\left(k + \frac{1}{2}\right)\right]$$

$$where\ k = 0, \dots., N-1$$

Multidimensional DCT

$$X_{k1,k2} = \sum_{n_1=0}^{N_1-1}\left(\sum_{n_2=0}^{N_2-1} \cos\left[\frac{\pi}{N_2}\left(n_2 + \frac{1}{2}\right)k_2\right] x_{n_1,n_2}\right) \cos\left[\frac{\pi}{N_1}\left(n_1 + \frac{1}{2}\right)k_1\right]$$

$$X_{k1,k2} = \sum_{\mathrm{n}_1=0}^{\mathrm{N}_1-1} \sum_{\mathrm{n}_2=0}^{\mathrm{N}_2-1} \mathrm{x}_{\mathrm{n}_1\mathrm{n}_2} \cos\left[\frac{\pi}{N_1}\left(n_1 + \frac{1}{2}\right)k_1\right] \cos\left[\frac{\pi}{N_2}\left(n_2 + \frac{1}{2}\right)k_2\right]$$

IMPLEMENTATION OF IMDCT

The drawback of DCT-I to DCT-IV is that these applicable for 1 D DCT and having slower speed. Multidimensional DCTs can be used for 2D DCT design but it also has slower speed. Chip design for 2D digital watermarking can faster using IMDCT transformation [19]. It accepts 18 discrete values. 18-point IMDCT (block size 36) for implementation is given by the following equation.

$$\widehat{x_m} = \frac{2}{N} \sum_{k=0}^{\left(\frac{N}{2}\right)-1} X_k . \cos\left[\frac{\pi}{2N}(2k+1)\left(2m + 1 + \frac{N}{2}\right)\right], \quad with\ m = 0,1,2, \dots \dots \dots., N-1$$

Generally, since we are dealing with a lapped transform, the recovered data sequence $\{\widehat{x_m}\}$ does not correspond to the original data sequence $\{x_m\}$. To obtain the correct $\{x_m\}$ the outputs of consecutive transforms have to be combined. It can be seen that N/2 (non redundant) input values result in N output values (of course the MDCT reads N input values and results in N/2 output values). Since it is not completely clear whether Equation1 should be called an N-point IMDCT or an N/2-point IMDCT, in the following we shall identify these transforms given the number of inputs. We will be describing an18-point IMDCT that delivers 36 output values, thus length N will be 36.

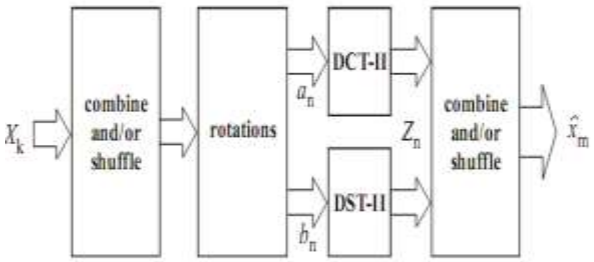

Figure 2 basic set up of IMDCT

Let us define N =36, we start from an18 values input sequence: {X0, X1, ⋯, X17}. The output of rotational block is given by

$$a_n = X_n \cos\left[\frac{\pi}{2N}(2n+1)\right] + X_{N/2-1-n} \sin\left[\frac{\pi}{2N}(2n+1)\right]$$

$$b_n = X_n \sin\left[\frac{\pi}{2N}(2n+1)\right] - X_{N/2-1-n} \cos\left[\frac{\pi}{2N}(2n+1)\right] \qquad n = 0,1,2,\ldots..\frac{N}{4} - 1$$

The left most' combine and shuffle'-block is thus nothing more than a reverse ordering of the second half of the input data.

The cos and sin angles

$$\theta = \frac{\pi}{2N}(2n+1)$$

Involved are,

$$\theta = \frac{\pi}{72}, \frac{3\pi}{72}, \frac{5\pi}{72} \ldots \ldots, \frac{17\pi}{72}$$

Next, we perform a 9-point DCT-II on the a_n vector and a 9-point DST-II on the b_n vector, which delivers us

$$Z_n = DCT - II\_9p(a_n)$$

With the DCT-II=9p given by

$$Z_j = \sum_{n=0}^{8} a_n . \cos\left[9j\pi\left(n+\frac{1}{2}\right)\right], \qquad j = 0,1, \ldots \ldots 8$$

$$Z_{(\frac{N}{4})-1+n} = DCT - II\_9p(b_n)$$

$$-b'_{n=even} = -b_{n=even,} \ \ then$$

$$Z_{(\frac{N}{2})-1+n} = DCT - II\_9p(b_n)$$

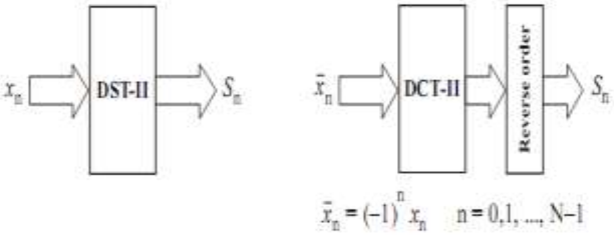

Figure 3 Constructing a DST-II from the DCT-II.

In the right most combine and shuffle'-block, y_k can be derived from these Z's as

$$y_k = \begin{cases} -Z_k, & for\ \ k = 0\ and\ k = \frac{N}{2} - 1 \\ Z_{\frac{N}{4}-1+\frac{k}{2}} - Z_k & for\ k = 2,4,6, \ldots\ldots, \frac{N}{2} - 2 \\ Z_{\frac{N}{4}-1+\frac{k+1}{2}} - Z_{\frac{k+1}{2}} & for\ \ k = 1,3,5, \ldots\ldots, \frac{N}{2} - 3 \end{cases}$$

Finally $\widehat{x_m}$ is given by

$$\widehat{x_m} = \begin{cases} -y_{\frac{N}{4}-1-m} \quad , & for\ \ m = 0,1, \ldots\ldots.. \frac{N}{4} - 1 \\ y_{m-\frac{N}{4}} & for\ m = \frac{N}{4} \ldots\ldots., \frac{3N}{4} - 1 \\ -y_{\frac{5N}{4}-1-m} & for\ m = \frac{3N}{4} \ldots\ldots., N - 1 \end{cases}$$

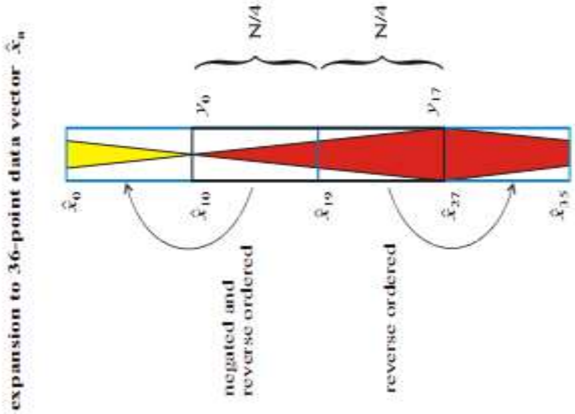

Figure 4 (a) Rotation of data

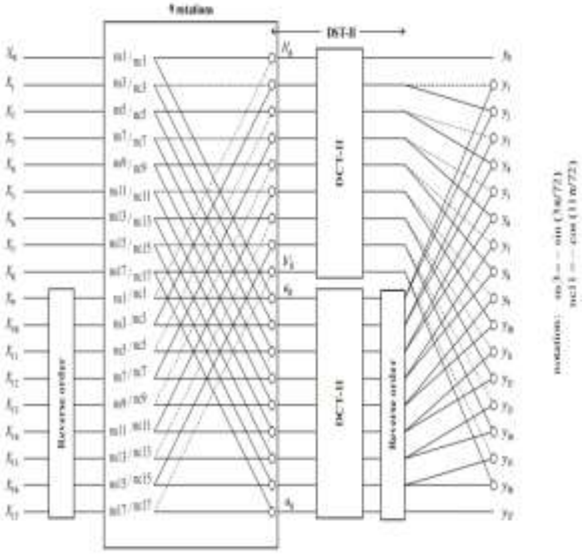

Figure 4 (b) Computation of IMDCT using butterfly method based on the proposed Methodology

Given an input data sequence {x0, x1, ⋯, x8}, the output data sequence $\{C^{II0}, C^{II1}, \cdots, C^{II8}\}$ can be computed with

$a_1 = x_3 + x_5$
$a_2 = x_3 - x_5$
$a_3 = x_6 + x_2$
$a_4 = x_6 - x_2$
$a_5 = x_1 + x_7$
$a_6 = x_1 - x_7$
$a_7 = x_8 + x_0$
$a_8 = x_8 - x_0$
$a_9 = x_4 + a_5$
$a_{10} = a_1 + a_3$

$a_{11} = a_{10} + a_7$
$a_{12} = a_3 - a_7$
$a_{13} = a_1 - a_7$
$a_{14} = a_1 - a_3$
$a_{15} = a_2 - a_4$
$a_{16} = a_{15} + a_8$
$a_{17} = a_4 + a_8$
$a_{18} = a_2 - a_8$
$a_{19} = a_2 + a_4$

$m_1 = -d_1 \cdot a_6$
$m_2 = d_2 \cdot a_5$
$m_3 = d_2 \cdot a_{11}$
$m_4 = -d_3 \cdot a_{12}$
$m_5 = -d_4 \cdot a_{13}$

$m_6 = -d_5 \cdot a_{14}$
$m_7 = -d_1 \cdot a_{16}$
$m_8 = -d_6 \cdot a_{17}$
$m_9 = -d_7 \cdot a_{18}$
$m_{10} = -d_8 \cdot a_{19}$

$a_{20} = x_4 - m_2$
$a_{21} = a_{20} + m_4$
$a_{22} = a_{20} - m_4$
$a_{23} = a_{20} + m_5$

$a_{24} = m_1 + m_8$
$a_{25} = m_1 - m_8$
$a_{26} = m_1 + m_9$

$C^{II}0 = a_9 + a_{11}$
$C^{II}1 = m_{10} - a_{26}$
$C^{II}2 = m_6 - a_{21}$
$C^{II}3 = m_7$
$C^{II}4 = a_{22} - m_5$

$C^{II}5 = a_{25} - m_9$
$C^{II}6 = m_3 - a_9$
$C^{II}7 = a_{24} + m_{10}$
$C^{II}8 = a_{23} + m_6$

$$Where \quad d_1 = \frac{\sqrt{3}}{2},$$
$$d_2 = 0.5,$$
$$d_3 = \cos\left(\frac{8\pi}{9}\right),$$

$$d_4 = \cos\left(\frac{4\pi}{9}\right),$$
$$d_5 = \cos\left(\frac{2\pi}{9}\right),$$
$$d_6 = \sin\left(\frac{8\pi}{9}\right),$$
$$d_7 = \sin\left(\frac{4\pi}{9}\right),$$
$$d_8 = \sin\left(\frac{2\pi}{9}\right),$$

Hence we can calculate the values of intermediate coefficients supporting the equation of IMDCT.

METHODOLOGY & TOOLS

Project navigator Application Version 6.1i or ISE 13.0 of Xilinx Company: Xilinx has been a semiconductor industry leader at the forefront of technology, market and business achievement. It is a tool to design the IC and to view their RTL (Register Transfer Logic) schematic .It is a tool to test the code on FPGA environment and we can get the all parameters details required to implement the Chip.

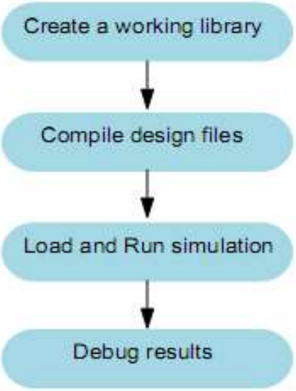

Figure 5: Chip Design Process Flow

Model SimEE 5.4a or 10.0 D of Mentor Graphics Company: Mentor Graphics was the first to combine single kernel simulator (SKS) technology with a unified debug environment for Verilog, VHDL, and SystemC. The combination of industry-leading, native SKS performance with the best integrated debug and analysis environment make ModelSim the simulator of choice for both ASIC and FPGA design. The best standards and platform support in the industry make it easy to adopt in the majority of process and tool flows.

Simulation and Design Steps: The following diagram shows the basic steps for simulating a design in ModelSim**.**
Creating the Working Library: In ModelSim, all designs are compiled into a library. Typically start a new simulation in ModelSim by creating a working library called "work," which is the default library name used by the compiler as the default destination for compiled design units.Compiling Design: After creating the working library, design is being compiled into it. The ModelSim library format is compatible across all supported platforms. Loading and Running the Simulator with the Design: With the design compiled, we load the simulator with design by invoking the simulator on a top-level module (Verilog) or a configuration or entity/architecture pair (VHDL). Assuming the design loads successfully, the simulation time is set to zero, and you enter a run command to begin simulation.

RESULTS AND EVALUTION

The snapshot is taken from the modelsim 10.1 software. *CLK and Reset* are the inputs. *X* is the input of sample and y is the sample output.

Step input1: *reset =1, clk* is applied for synchronization and then run.

Step input 2: *reset =0, clk* is applied for synchronization, *X* samples are taken and then run.

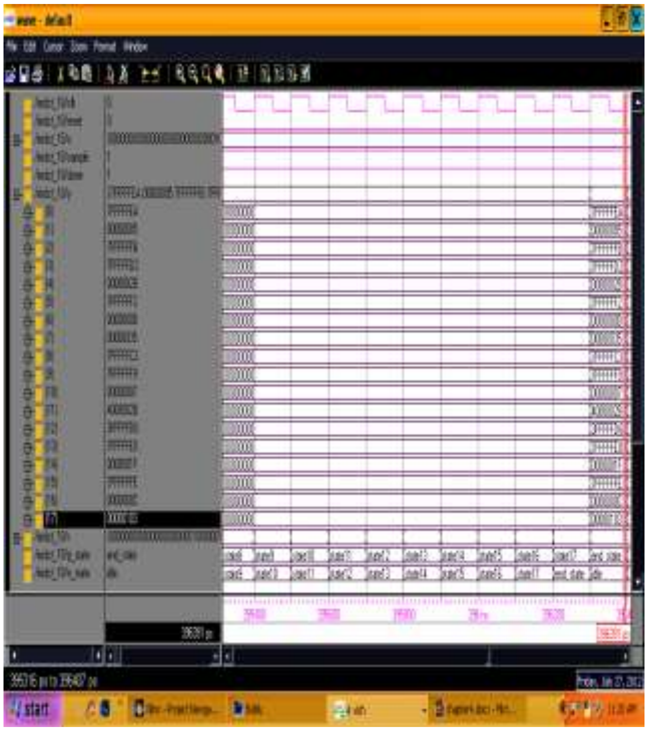

Figure 6 Modelsim waveform with watermarked output

DEVICE UTILIZATION AND TIMING SUMMARY

Device utilization report is the report of used device hardware in the implementation of the chip and timing report is the minimum and maximum time to reach the output.

Selected Device: 3s200pq208-4

Table 1 device utilization in IMDCT structure

Device part	Utilization
Number of Slices	904 out of 1920 47.08%
Number of Flip Flops	627 out of 3840 16.32%
Number of 4 input LUTs	440 out of 3840 11.45%
Number of bonded IOBs	153 out of 141 108.51%
Number of MULT18X18s	2 out of 6 33.33%
Number of GCLKs	1 out of 8 12%

Timing Summary:

Speed Grade: -4

Minimum period: 2.223ns (Maximum Frequency: 449.843MHz)

Minimum input arrival time before clock: 19.322ns
Maximum output required time after clock: 6.379ns

CONCLUSION AND GUTURE WORK

Different hardware architectures for implementing secure watermarking algorithms proposed by different authors has discussed through this paper. These architectures have implemented by using different tools of VLSI technologies and they have achieved positive results. Great advantages are gained due to using hardware based implementation of watermarking algorithms, such as reduce hardware scheme area, decrease power consumption and increase speed of performance. Therefore a hardware watermarking solution is often more reliable and economical. We have validated the hardware parameters in the chip design of faster DCT. Hardware parameters are optimized using IMDCT. We also have analyzed that chip design with IMDCT is faster than simple DCT. Timing analysis and synthesis report shows the device functionality and proves the optimized results.

To the best of our knowledge, this is the best watermarking architecture having both functionalities. The chip can be easily integrated in any existing JPEG encoder to watermark images right at the source end. The implementation of a low-power, high-performance version is currently in progress .Low-power VLSI features, such as multiple supply voltages, dynamic clocking, and clock gating will be considered. High performance architectural implementations, such as pipeline or parallelism, are under investigation. The disadvantage of the watermarking algorithms implemented is that the processing needs to be performed pixel by pixel. In the future, we plan to investigate block-by-block processing. A low power, high-performance watermarking decoder is also in the implementation stage. We can also plan to modify the architectures to handle color images. Because DRM systems need both encryption and watermarking, we believe that combining both the hardware and the data compression architectures will be necessary. Moreover, the on-chip encrypter can be used in storing the watermarking generator key in encrypted form, thus enhancing watermark security. We can enhance the security of data with correct reorganization of each character at receiving end. Chip implementation is done with dual port block RAM and FIFO logic which decides the priority of characters based on encrypted text at transmitting end. FIFO logic assigns the priority on the receiving end and saves data in the blocked RAM memory.

REFERENCES

[1] S. Emmanueland M.S. Kankanhalli, “A Digital Rights Management Scheme for Broadcast Video,” ACM-Springer Verlag Multimedia Systems Journal, vol.8, no.6, pp.444–458, June 2003.

[2] D. Kundur and K. Karthik, “Digital Finger printing and Encryption Principles for Digital Rights Management,” IEEE Signal Processing, vol.52, 2004.

[3] A. M. Eskicioglu and E. J. Delp, “An Overview of Multimedia Content Protection in Consumer Electronics Devices,” Elsevier Signal Processing: Image Communication, vol.1 6, pp.681–699, 2001.

[4] N. M. Kosaraju, M. Varanasi, and S.P. Mohanty, “A High Performance VLSI Architecture for Advanced Encryption Standard (AES) Algorithm, ”in Proceedings of 19th IEEE International Conference on VLSI Design, 2006,pp .481–484.

[5] S. Katzen beisser and F.A.P. Petitcolas, Information Hiding techniques for steganography and digital watermarking, Artech House, Inc.,MA, USA, 2000.

[6] S.P.Mohanty, “Digital Watamerking of Images,” M.S. thesis, Department of Electrical Engineering, Indian Institute of Science, Bangalore, India,1999.

[7] N. Memonand P.W. Wong, “Protecting Digital Media Content,” Communications of the ACM, vol. 41, no.7, pp.35–43, July 1998.

[8] F. Mintzer, G.Braudaway, and M.Yeung, “Effective and Ineffective Digital Watermarks,”in IEEE International Conference on Image Processing (ICIP-97), 1997, vol.3, pp.9–12.

[9] IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 4, No 1, July 2010 ISSN (Online): 1694-0784 ISSN (Print): 1694-0814 Watermarking of Digital Video Stream for Source Authentication Kesavan Gopal, Dr. M. Madhavi Latha, Team Leader, Infotech Enterprises Limited Bangalore, Karnataka, India (Page 1,6)

[10] White paper on "IMPLEMENTATION OF THE 2D DCT USING A XILINX XC6264 FPGA", D.W. Trainor J.P. Heron* and R.F. Woods, Department of Electrical Integrated Silicon Systems Ltd and Electronic Engineering Chlorine Gdn's The Queen's University of Belfast N. Ireland(Page 9)

[11] International Journal of Computer Science & Security (IJCSS), Volume (4): Issue (2) FPGA Prototype of Robust Image Watermarking for JPEG 2000 With Dual Detection: The watermark detection algorithm is implemented using MatLab (Page 1,4)

[12] I.J.Cox, J. Kilian, T. Leighton, and T. Shamoon, "Secure Spread Spectrum Watermarking for Multimedia," IEEE Transactions on Image Processing, vol.6, no.12, pp.1673–1687, Dec1997.

[13] H. Berghel, "Watermarking Cyber space," Communications of the ACM, vol.40, no.11, pp.19–24, November1997.

[14] M.Swanson, M.Kobayashi, and A.Tewfik, "Multimedia Data Embedding and Watermarking Technologies," Proceedings of the IEEE, vol.86, no.6, pp.1064–1087, June1998.

[15] S.Craver, N.Memon, B.L.Yeo, and M.M.Yeung, "Resolving Rightful Ownerships with Invisible Watermarking Techniques: Limitations, Attacks and Implications,"IEEE Journal on Selected Areas in Communications, vol.16, no.4, pp.573–586, May1998.

[16] S. P. Mohanty, K.R. Ramakrishnan, and M. S. Kankanhalli, "A DCT Domain Visible Watermarking Technique for Images," in Proceedings of the IEEE International Conference on Multimedia and Expo, 2000, pp.1029–1032.

[17] S. P. Mohanty, K. R. Ramakrishnan, and M.S.Kankanhalli, "An Adaptive DCT Domain Visible Watermarking Technique for Protection of Publicly Available Images," in Proceedings of the International Conference on Multimedia Processing and Systems, 2000, pp.195–198.

[18] C.T.Li, "Digital fragile watermarking scheme for authentication of JPEG images," IEEE Proceedings-Vision, Image and Signal Processing, vol.151, no.6, pp.460–466, Dec2004.

[19] IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL 5, NO. 8, AUGUST 1996: An FFT-Based Technique for Translation, Rotation, and Scale-Invariant Image Registration B. Srinivasa Reddy and B. N. Chatterji (Page 8)

[20] Semiconductor Complex Limited, Internet PDF: Data sheets of XC 95 series CPLD [Online].

[21] www.wikipedia.com

[22] www.altera.com

[23] www.actel.com

[24] www.xilinx.com

[25] M. Morris Mano Digital Design, Second Edition by Pearson Education.