MODELING & IMPLEMENTATION OF DIGITAL IMAGE WATERMARKING CHIP BASED ON DCT IN VHDL

ATUL KUMAR[1], SHIVANI CHAUHAN[2]

ABSTRACT-*The success of the Internet, cost-effective and popular digital recording and storage devices, and the promise of higher bandwidth and quality of service (QoS) for both wired and wireless networks has made it possible to create, replicate, transmit, and distribute digital content in an effortless way. The protection and enforcement of intellectual property rights for digital media has become an important issue. On the other hand, this situation has brought about the possibility of duplicating and/or manipulating the data. To keep on with the transmission of data over the internet the reliability and originality of the transmitted data should be verifiable. It is necessary that multimedia data should be protected and secured. One way to address this problem involves embedding an invisible data into the original data to mark ownership of them. There are many techniques for information hiding, which can be divided into different categories such as convert channels, steganography, anonymity, and watermarking. Convert channels techniques were defined in the context of multilevel secure systems. Convert channels usually handle properties of the communication channels in an unexpected and unforeseen way in order to transfer data through the medium without detection by anyone other than the entities operating the covert channel. Digital watermarking plays an important role for protecting digital contents from unauthorized copying. My research work proposes a new digital watermarking method based on Discrete Cosine Transformation (DCT) for copyright protection. In our proposed watermarking method, the original audio is transformed into DCT domain. The absolute values of DCT coefficients are divided into an arbitrary number of segments and the energy of each segment is calculated. Our research describes all these aspects of VHDL design as applied to the design of a simple FPGA that was designed as an individual project in a VLSI class. The size of the FPGA was restricted to the one that could fit into the MOSIS 40-pin tiny chip pad frame. The FPGA consisted of 3-input LUTs as configurable logic blocks, and a chain of shift registers to hold the configuration bits. In addition, output flip-flops were also provided so that a state machine could be implemented in the FPGA. The design process served as a very useful tool to learn about VLSI design since it encompassed all possible aspects of a complex VLSI design. In my proposed research, I have taken the discrete cosine transform (DCT) to configure the watermarks. We developed the hardware chip in Xilinx 13.2 VHDL software, and functional simulation in Modelsim 10.1 b, student edition. After the review of various research papers on DCT and its chip implementation, I concluded that there are many versions of DCT such as DCT-I, DCT-II, DCT-III, multidimensional DCT, but all are slower. There is another version of the DCT which is Inverse modified discrete cosine transform (IMDCT). IMDCT is faster in comparison of simple DCT. IMDCT has been proved the best transformation for image compression and watermarks. Design, modeling and VHDL chip implementation is done for digital watermarking using faster DCT.*

Key Words: **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.

1. INTRODUCTION TO WATERMARK TECHNOLOGY

Because of the widespread of the internet, digital image watermarking became popular for proof of ownership (copyrights and IP protection), copying prevention, broadcast monitoring, authentication and data hiding. Digital image watermarking technique involves, adding undetectable copyright information or data or message to the original image, which identifies the ownership. After adding the watermark in the original image, there should be no image degradation watermark should not be removable and should be robust against different types of attacks. 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. Some of the desired characteristics of visible water-marks are listed below.

- The watermark should be visible yet must not significantly obscure the image details beneath it.
- The watermark must be difficult to remove; removing a watermark should be more costly and labor intensive than purchasing the image from the owner.
- The information carried by the watermark is robust to content manipulations, compression, and soon.

- The watermark should be applied automatically with little human intervention and labor.

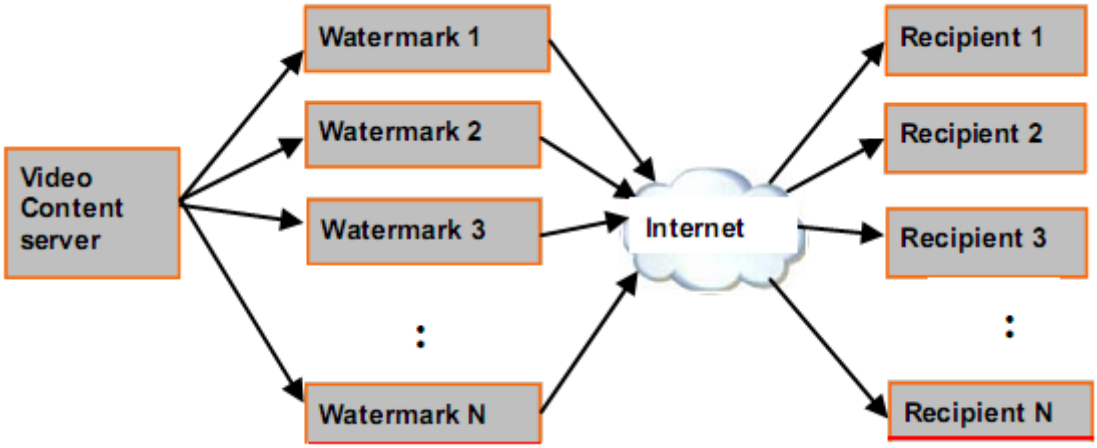

Figure 1.1 Multicast Video Streaming over internet

In multicast application scenario like VoD, a video server sends out a networked high quality visual program to its users is shown in fig 1.1. 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. Watermarking in real time will solve the source authentication issues.

2. MATHEMATICAL MODEL OF DCT

Formally, the discrete cosine transform is a linear, invertible function F: $\mathbf{R}^N$ -> $\mathbf{R}^N$ (where $\mathbf{R}$ 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:

1. DCT-I

$$X_k = \frac{1}{2}(x_0 + (-1)^k x_{N-1}) + \sum_{n=1}^{N-2} x_n \cos\left[\frac{\pi}{N-1} nk\right] \; where \; k = 0, \ldots., N-1$$

Some people further multiply the x_0 and x_{N-1} terms by $\sqrt{2}$, and correspondingly multiply the X_0 and X_{N-1} terms by $1/\sqrt{2}$. This makes the DCT-I matrix orthogonal, if one further multiplies by an overall scale factor of $\sqrt{2/(N-1)}$, but breaks the direct correspondence with a real-even DFT.

The DCT-I is exactly equivalent (up to an overall scale factor of 2), to a DFT of 2N-*2* real numbers with even symmetry. For example, a DCT-I of N=5 real numbers *abcde* is exactly equivalent to a DFT of eight real numbers *abcdedcb* (even symmetry), divided by two. (In contrast, DCT types II-IV involve a half-sample shift in the equivalent DFT.)Note, however, that the DCT-I is not defined for N less than 2. (All other DCT types are defined for any positive N.)

Thus, the DCT-I corresponds to the boundary conditions: x_n is even around n=0 and even around n=N-1; similarly for X_k.

2. DCT-II

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

The DCT-II is probably the most commonly used form, and is often simply referred to as "the DCT".

This transform is exactly equivalent (up to an overall scale factor of 2) to a DFT of $4N$ real inputs of even symmetry where the even-indexed elements are zero. That is, it is half of the DFT of the $4N$ inputs y_n, where $y_{2n}=0$, $y_{2n+1}= x_n$ for $0 \le n < N$, and $y_{4N-n}= y_n$ for $0<n< 2N$. Some people multiply the X_0 term by $1/\sqrt{2}$ and multiply the resulting matrix by an overall scale factor of $\sqrt{2/N}$. This makes the DCT-II matrix orthogonal, but breaks the direct correspondence with a real-even DFT of half-shifted input. The DCT-II implies the boundary conditions: x_n is even around n=-1/2 and even around n=N-1/2; X_k is even around k=0 and odd around k=N.

3. DCT-III

$$X_k = \frac{1}{2}x_0 + \sum_{n=1}^{N-1} x_n \cos\left[\frac{\pi}{N} n \left(k+\frac{1}{2}\right)\right] \; where\; k = 0, \ldots., N-1$$

Because it is the inverse of DCT-II (up to a scale factor, see below), this form is sometimes simply referred to as "the inverse DCT" ("IDCT").Some authors further multiply the x_0 term by $\sqrt{2}$ and multiply the resulting matrix by an overall scale factor of $\sqrt{2/N}$,so that the DCT-II and DCT-III are transposes of one another. This makes the DCT-III matrix orthogonal, but breaks the direct correspondence with a real-even DFT of half-shifted output. The DCT-III implies the boundary conditions: x_n is even around n = 0 and odd around n = N; X_k is even around k = -1/2 and even around k = N-1/2.

4. DCT-IV

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

The DCT-IV matrix becomes orthogonal (and thus, being clearly symmetric, its own inverse) if one further multiplies by an overall scale factor of $\sqrt{2/N}$.A variant of the DCT-IV, where data from different transforms are *overlapped*, is called the modified discrete cosine transform (MDCT) The DCT-IV implies the boundary conditions: x_n is even around n = -1/2 and odd around n = N-1/2; similarly for X_k.

5. DCT V-VIII

DCT types I-IV are equivalent to real-even DFTs of even order (regardless of whether N is even or odd), since the corresponding DFT is of length $2(N-1)$ (for DCT-I) or $4N$ (for DCT-II/III) or $8N$ (for DCT-VIII). In principle, there are actually four additional types of discrete cosine transform, corresponding essentially to real-even DFTs of logically odd order, which have factors of N±½ in the denominators of the cosine arguments.

Equivalently, DCTs of types I-IV imply boundaries that are even/odd around either a data point for both boundaries or halfway between two data points for both boundaries. DCTs of types V-VIII imply boundaries that even/odd around a data point for one boundary and halfway between two data points for the other boundary. However, these variants seem to be rarely used in practice. One reason, perhaps, is that FFT algorithms for odd-length DFTs are generally more complicated than FFT algorithms for even-length DFTs, and this increased intricacy carries over to the DCTs. (The trivial real-even array, a length-one DFT (odd length) of a single number a, corresponds to a DCT-V of length N = 1.)

3. DESIGN METHODOLOGY FOR FASTER DCT

The mathematical equations for various DCT are given as:

DCT-I

$$X_k = \frac{1}{2}(x_0 + (-1)^k x_{N-1}) + \sum_{n=1}^{N-2} x_n \cos\left[\frac{\pi}{N-1} nk\right] \; where\; k = 0, \ldots., N-1$$

DCT-II

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

DCT-III

$$X_k = \frac{1}{2}x_0 + \sum_{n=1}^{N-1} x_n \cos\left[\frac{\pi}{N} n \left(k + \frac{1}{2}\right)\right] \; where \; k = 0, \ldots., N-1$$

DCT-IV

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

Multidimensional DCT

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

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

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. It accepts 18 discrete values. 18-point IMDCT (block size 36) for implementation is given by the following equation.

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

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.

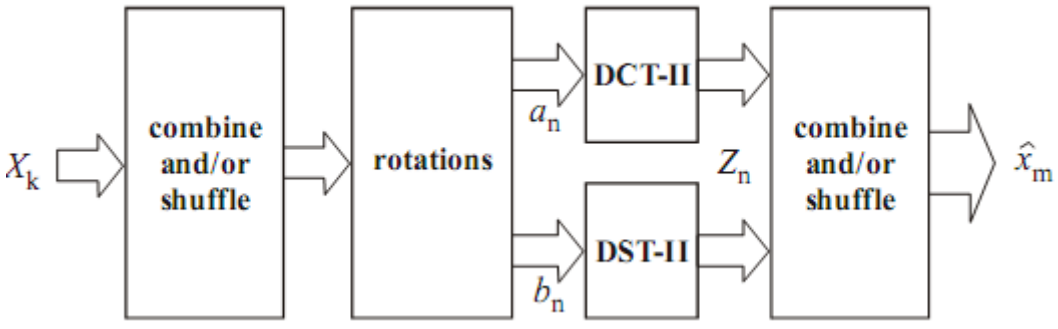

Figure basic set up of IMDCT

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

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

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

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,

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

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

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

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

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

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

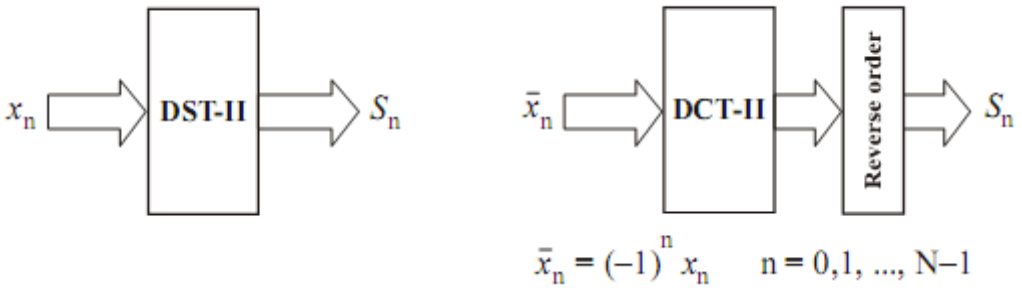

Figure Constructing a DST-II from the DCT-II.

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

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

Finally $\widehat{x_m}$ is given by

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

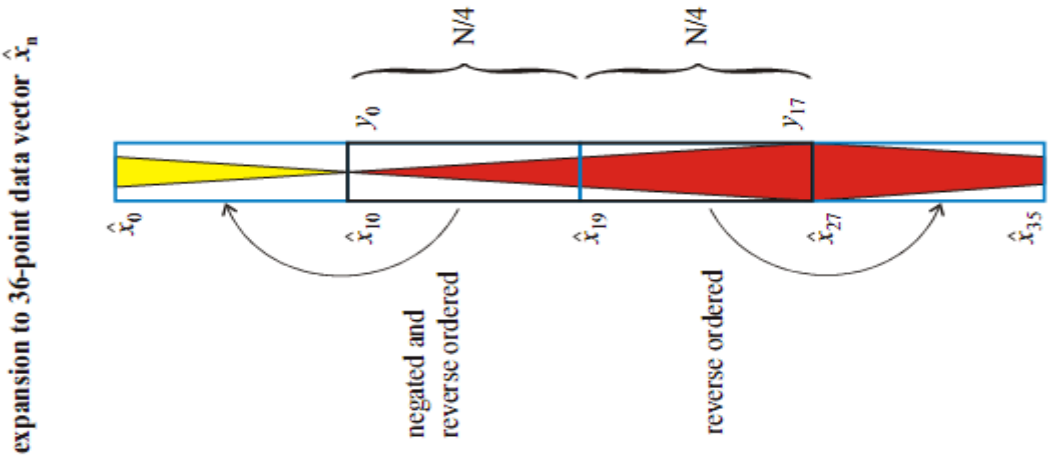

Figure Rotation of data

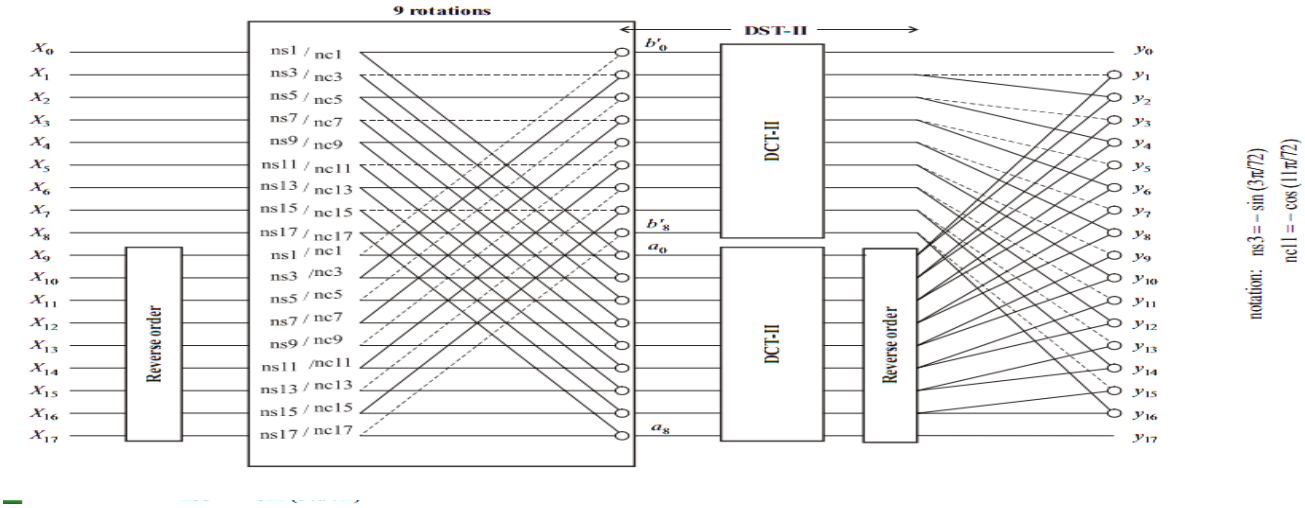

Figure Computation of IMDCT using butterfly method based on the proposed Methodology

4. RESULTS AND EVALUTION

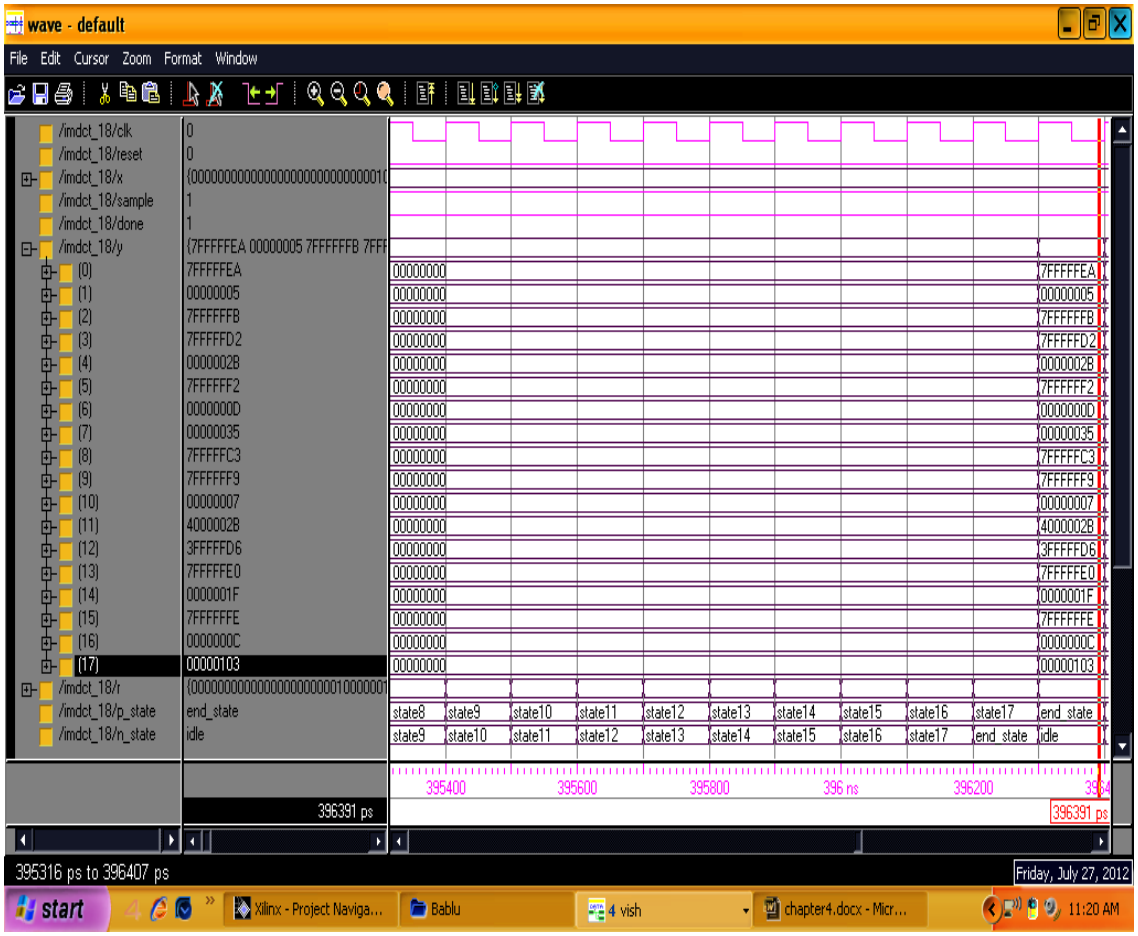

Figure ModelSim waveform with watermarked output

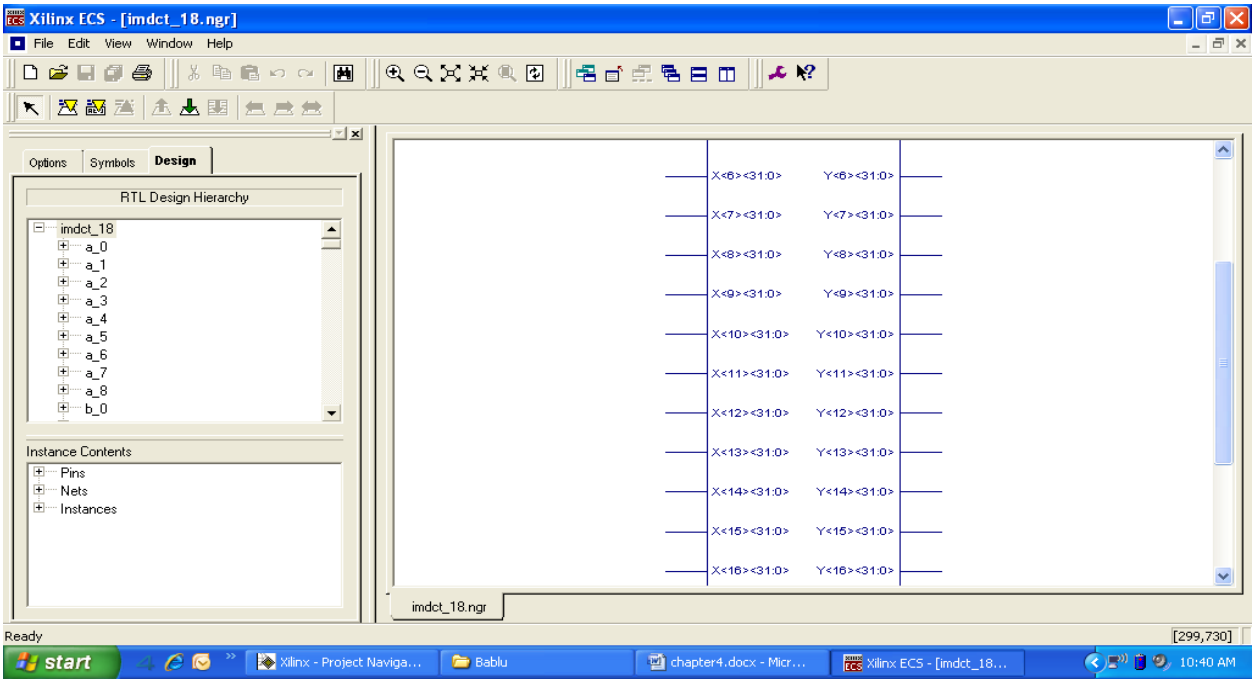

Figure RTL view

5. CONCLUSION

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.

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] M. Eskiciogluand 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.