A Hardware Efficient Robust Digital Image Watermarking Algorithm Using Integer DCT

Gaurav Gupta[1], Kanika Sharma[2]

[1]M.E. Scholar, [2]Assistant Professor, ECE Department
NITTTR, Chandigarh, India

Abstract — Online watermarking of digital images is successful in avoiding offline attacks. The critical need of real time watermarking is speed and it can be achieved by using a hardware which is dedicated for this very purpose only. The proposed research work presents VLSI architecture for embedding a binary watermark image into an 8-bit gray scale image. The proposed watermarking algorithm uses integer DCT and is robust to various image processing attacks. The watermark is retrieved using the same hardware. The performance of the algorithm is verified using MATLAB by the improved values of MSE, PSNR, NC and TAF. The hardware efficiency of the algorithm is judged and verified in terms of device utilization parameters and delay. The proposed VLSI architecture is implemented on Spartan 3E FPGA and simulated using XILINX ISE9.1.

Keywords — *Real Time Watermarking, VLSI Architecture, Integer DCT, Robust Image Watermarking*

I. INTRODUCTION

The need for real time watermarking was felt with advancement in the field of medical imaging. With the support of information technology, the evolution of medical information system took place. It enables the information to be shared between distant health professionals and manipulated more easily. However, the protection of the patient information is of prime concern. The security issues include authenticity, confidentiality, availability and integrity [1]. In this paper, the major issue of concern is to provide authenticity which involves providing proof that the information belongs to the correct patient and it is issued from the right source. It involves some secret information to be shared between the sender and the receiver. For example, automatic retrieval of patient records based on watermark embedded in an X-ray. The embedded watermark needs to be invisible and should be such that it may alter the details or the information of the original image. Another important application is the real time watermarking of surveillance camera tapes with location and time codes to protect against tampering [2-4].

The image watermarking algorithm needs to satisfy some requirements. First, the watermark should be invisible to normal human eye. This may reduce the chances of easy tampering and copying. Since, a number of distortions occur in the transmission channel, the watermarking algorithm should be robust to various image processing attacks [5, 6]. For implementing the real time watermarking system, the algorithm should be simple and must include less number of complex arithmetic functions like division and multiplication. To further simplify the VLSI architecture of such a system, the floating point arithmetic should be avoided. This will increase the speed of processing but at the cost of accuracy. However, the algorithm may be designed accordingly to compensate for the decrease in accuracy. Lastly, the watermarking algorithm must have high information embedding capacity [7].

In this proposed work, an invisible image watermarking algorithm is designed to embed a binary watermark image in an 8-bit gray scale image. The VLSI architecture for implementing the algorithm is designed using VHDL and simulated on Spartan 3E FPGA using XILINX ISE9.1. Integer DCT is used to transform the image pixels into the DCT coefficients. The reason behind using integer transform is that all the arithmetic operations will be done on integers, thereby reducing the complexity and increasing the speed.

II. RELATED WORK

A lot of research has been carried out to develop a watermarking algorithm for protecting ownership and copyrights. In the field of medical information system, substantial work has been done to embed the patient information in the medical images such as X-rays, ultrasound etc. However, most of the work is done for offline applications. Hardware is needed for embedding watermark in real time. Mohanty et al. [8] developed a low power, reliable, secure and real time watermark encoder. They synthesized an invisible-robust spatial domain encoder chip using Xilinx FPGA using VIRTEX technology operated at 50MHz. Pseudo random numbers are used as watermark information which is generated by a linear feedback shift register (LFSR). The watermarking encoder chip consists of a watermark generator, watermark insertion module, and a controller. The same author developed a chip for invisible robust and fragile watermarking in [9]. He prototyped the watermarking architecture using XILINX FPGA and a custom integrated circuit. For the custom IC design, the author used Cadence tools with 0.35µm CMOS technology. Joshi et al. [10] presented a paper emphasizing the need of hardware in real time watermarking of videos. Software watermarking is comparatively slow which

may introduce delay between capturing and embedding of the watermark in the host image. The algorithm is developed to suit the H.264 video coding standard. The algorithm is implemented on FPGA and its performance is verified using MATLAB. A high speed hardware implementation of watermarking chip is presented by [11]. The author implemented the 5-stage pipelining in DCT/IDCT which is used in digital watermarking algorithm. The 5-stage pipelining gives almost 500% increase in the speed over the conventional methods. XILINX XC3S4000 FPGA is used to implement the algorithm on hardware. Ghosh et al. [12] implemented the reversible contrast mapping based watermarking algorithm on FPGA. The encoder architecture is independent of the clock. Thus, the watermark is embedded as soon as the host image is fetched. This results in increase in the response time, hence, the speed. The proposed architecture is implemented on SPARTAN 3E FPGA family. However, the decoder architecture depends on the clock frequency. Scaria et al. [13] proposed the design and hardware implementation of a fast RGB to YUV converter by standard NTSC conversion and reconstruction formulae by using optimal 2-D systolic arrays for multiplying the matrices. The watermark is embedded in the wavelet domain. The watermark image and the host image are decomposed into sub-bands i.e. LL, LH, HL and HH. SVD is applied on each sub-band and the singular values of the original image are modified using the singular values of the watermark image. Kiran et al. [14] proposed VLSI architecture for real time watermarking based on Haar wavelet transformation. The proposed algorithm uses 2D DWT to decompose the host image into various sub-bands. The watermark is inserted into the middle frequency coefficients.

III. PROPOSED IMAGE WATERMARKING ALGORITHM

The proposed digital image watermarking algorithm includes hiding the information in the DC coefficients of the DCT blocks of the image. NxN DCT transform is applied to the host image which divides it into NxN DCT blocks. The number of DCT blocks in the transformed image is given by (1),

$$No.of\ Blocks(k) = \frac{J * K}{N * N} \quad \dots(1)$$

where k is the number of DCT blocks in the transformed image and JxK is the size of the host image. The process of DCT transformation is shown in fig. 1. $D_{x,y}\left(1 \le x \le \frac{J}{N} and\ 1 \le y \le \frac{K}{N}\right)$ is the resulting DCT block containing NxN number of DCT coefficients. A 4x4 DCT block is shown in fig. 2.

A DCT coefficient of a block is denoted by $D_{x,y}(i,j)$, where the index i and j denotes the row and column respectively of that particular block. The coefficient on the top most left corner of fig. 2 denotes the DC coefficient i.e. $D_{x,y}(1,1)$. The proposed image watermarking algorithm uses integer DCT to avoid floating point arithmetic and to have compatibility with H.264 video coding standard for extending the algorithm for video watermarking. The involvement of integer arithmetic instead of floating point calculations reduces the complexity of the algorithm and makes it suitable for real time applications. The computational complexity of the integer transform is less. The advantages in implementation of integer transform are: (i) Integer transform is calculated without the use of multiplier. It requires only addition and shift operation. (ii) Integer numbers results are faster to compute than the floating point numbers. The discrete cosine transform (DCT) represents an image as a sum of sinusoids of varying magnitudes and frequencies. The DCT has the property that, for a typical image, most of the visually significant information about the image is concentrated in just a few coefficients of the DCT. The two dimensional DCT of a JxK image I is given by (2).

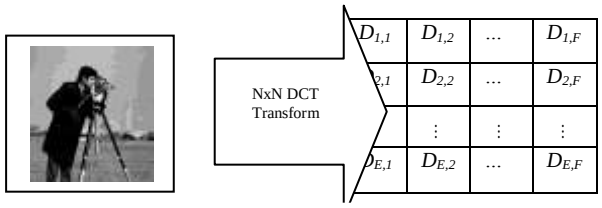


Figure. 1 DCT transform of the host image

(1,1)	(1,2)	(1,3)	(1,4)
(2,1)	(2,2)	(2,3)	(2,4)
(3,1)	(3,2)	(3,3)	(3,4)
(4,1)	(4,2)	(4,3)	(4,4)

$B_{x,y}(i,j)$ DCT coefficient

Figure 2. A 4x4 DCT block

$$B_{pq} = \alpha_p \alpha_q \sum_{m=0}^{J-1} \sum_{n=0}^{K-1} A_{mn} cos \frac{\pi(2m+1)p}{2J} cos \frac{\pi(2n+1)q}{2K}$$
$$(1 \le p \le J\ and\ 1 \le q \le K) \quad \dots(2)$$

$$\alpha_p = \begin{cases} \frac{1}{\sqrt{J}}, & p = 0 \\ \sqrt{\frac{2}{J}}, & 1 \le p \le J-1 \end{cases},$$

$$\alpha_q \begin{cases} \frac{1}{\sqrt{K}}, & q = 0 \\ \sqrt{\frac{2}{K}}, & 1 \le q \le K-1 \end{cases} \quad \ldots (3)$$

The values B_{pq} are called the DCT coefficients of image I. The reverse transform kernel of the DCT transform is same as the forward transform kernel. Equation (2) is applied on every NxN block of the image. In this way, we get the k blocks of the DCT coefficients. 2D Integer DCT is calculated by separable property in two steps by successive 1D operations carried out on each column and row, respectively. The resultant coefficient forms an N × N matrix as shown in (4).

$$A(i,j) = C(i)\cos\left(\frac{(2j+1)i\pi}{2N}\right) \quad \ldots (4)$$

The resultant matrix is defined as Cf, which is derived from A(i,j) for respective values of i and j for N = 8. The resultant matrix is given by (5).

$$C_f = {}^{1}/_{8} * \begin{bmatrix} 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ 12 & 10 & 6 & 3 & -3 & -6 & -10 & -12 \\ 8 & 4 & -4 & -8 & -8 & -4 & 4 & 8 \\ 10 & -3 & -12 & -6 & 6 & 12 & 3 & -10 \\ 8 & -8 & -8 & 8 & 8 & -8 & -8 & 8 \\ 6 & -12 & 3 & 10 & -10 & -3 & 12 & -6 \\ 4 & -8 & 8 & -4 & -4 & 8 & -8 & 4 \\ 3 & -6 & 10 & -12 & 12 & -10 & 6 & -3 \end{bmatrix} \quad \ldots (5)$$

As can be seen from (5), Cf is an 8 × 8 matrix of 1D Integer DCT coefficients. The matrix contains coefficients having integer values. Hardware implementation of DCT for integer values outperforms the same operation carried on floating point values. The algorithm for H.264 encoder is designed to take due advantage of 2D DCT simplicity and compatibility with integer operation [15].

The information content of the image or the energy contents are located in few coefficients rather than being distributed equally among all the coefficients in a block. The frequency distribution of the DCT block is shown in fig. 3.

In the proposed method, only DC coefficient of each block is chosen to hide watermark information. It is because the high frequency contents are easily modified by the noise and other attacks. The low frequency contents are less susceptible and the DC coefficient is least susceptible to attacks. Only one bit is hidden per block. This accounts to less embedding capacity but accounts to greater robustness against attacks.

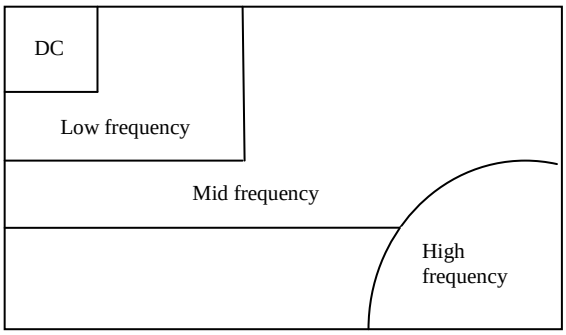


Figure 3. Frequency distribution in a DCT block

Watermark Embedding Algorithm using NxN DCT

The pseudo code for embedding watermark image in the host image is given below:

// I: host image of size JxK, W: watermark image of size XxY
//D: DCT of host image taken NxN block-wise
// I_W: watermarked host image, W_S: watermark bits sequence
// key_lfsr: key to generate pseudo random watermark bits
// LFSR: Linear feedback shift register, W_SS: pseudo randomly generated watermark bits sequence
//$D_{x,y}$: NxN block of DCT transformed host image, $D_{x,y}(i,j)$: coefficient of DCT block

Initialization: μ=15-35, n=1, c=1
I = read (host image)
W = read (watermark image)
[J K] = size (I)
[X Y] = size (W)
D = block_processing(I, DCT, NxN)
For row (1 to X)
  For col (1 to Y)
   W_S(c) = W (row, col)
   c=c+1
  end col loop
end row loop
W_SS = LFSR (key_lfsr, W_S)
For i (1 to J) step N
  For j (1 to K) step N
  $D'_{x,y}(i,j) = D_{x,y}(i,j)/ \mu$
  If W_SS(n) = '1' **then**
   $D''_{x,y}(i,j)$ = odd $(D'_{x,y}(i,j)$
   $D^{new}_{x,y}(i,j) = \mu * D''_{x,y}(i,j)$
  Else
   $D''_{x,y}(i,j)$ = even $(D'_{x,y}(i,j)$
   $D^{new}_{x,y}(i,j) = \mu * D''_{x,y}(i,j)$
  End if
  n = n+1
 End j loop
End i loop
I_W = block_processing ($D^{new}_{x,y}$, IDCT, NxN)

The host image, I, is transformed to D using NxN DCT. The DC coefficient of every block is used to hide one bit of information. The watermark image is converted into bits sequence, W_S. The resulting bit sequence is shuffled to generate an array; W_SS. Linear Feedback Shift Register (LFSR) is used to generate pseudo random numbers using a key, key_lfsr. According to the pseudo random numbers generated by LFSR, the watermark bits sequence (W_S) is shuffled into W_SS. Now, DC coefficient of every block is selected and divided by µ. It results in a modified coefficient $\boldsymbol{D'_{x,y}(i,j)}$. µ is known as the quality factor. It determines the PSNR of the watermarked image. The higher the value of µ, the less will be the PSNR value. The higher the PSNR, the more information is preserved. The value of µ also changes for 4x4, 8x8 or 16x16 DCT.

The shuffled watermark sequence (W_SS) is checked bit by bit. If W_SS (n) is '1' then $\boldsymbol{D'_{x,y}(i,j)}$ is converted into the nearest odd number, resulting in $\boldsymbol{D''_{x,y}(i,j)}$, where n is the index of array W_SS. If W_SS (n) is '0' then $\boldsymbol{D'_{x,y}(i,j)}$ is converted into the nearest even number, resulting in $\boldsymbol{D''_{x,y}(i,j)}$. Now, this modified coefficient is multiplied by quality factor µ, to get a new DC coefficient of block as $D^{new}_{x,y}(i,j)$. After modification of the DC coefficients of every block, an inverse DCT transform (IDCT) of the complete matrix is taken to obtain the watermarked image (I_W).

Watermark Extraction Algorithm using NxN DCT

The pseudo code for extracting watermark image from the watermarked host image is shown below:

// I_W: Watermarked host image, DW: NxN DCT transform of I_W

// $DW_{x,y}$: NxN block of DCT transformed watermarked image, $DW_{x,y}(i,j)$: coefficient of DCT block

// key_lfsr: key to generate pseudo random watermark bits, LFSR: Linear feedback shift register

// W_SSW: Retrieved watermark sequence, W_SW: Watermark sequence arranged using pseudo random sequence

// W_W: Retrieved watermark image

Initialization: µ=15-35, n=1, c=1

I_W = read (watermarked image)

DW= block_processing (I_W, DCT, NxN)

For i (1 to J) step N

For j(1 to K) step N

$DW'_{x,y}(i,j) = DW_{x,y}(i,j)$ / µ

$DW''_{x,y}(i,j)$ = round ($DW'_{x,y}(i,j)$)

If $DW''_{x,y}(i,j)$ is even **then**

W_SSW (n) = '0'

Else

W_SSW (n) = '1'

End if

n = n+1

end j loop

end i loop

W_SW = LFSR (key_lfsr, W_SSW)

For row (1 to X)

For col (1 to Y)

W_W (row, col) = W_SW (c)

c = c + 1

end col loop

end row loop

calculate NC

calculate TAF

The watermarked image I_W is transformed into DW using NxN DCT transform. The information is hidden in DC coefficient of each block. Thus, the loop is set to select the DC coefficient of each block i.e. $DW_{x,y}(i,j)$. This coefficient is divided by the quality factor µ which results in modified coefficient $DW'_{x,y}(i,j)$. A round function is applied to this value to attain the final value of the modified coefficient $DW''_{x,y}(i,j)$. This value is checked for even or odd. If it is found to be even then the watermark bit is '0' otherwise '1'. The sequence retrieved using this process is not the final one. The final watermark sequence is attained by shuffling it by using the LFSR with the same key. The watermark image is retrieved by converting the watermark sequence into the UxV matrix form. The retrieved watermark image can be

verified by calculating the Normalized Correlation and the Tamper Assessment Function. These two terms are described in detail in the next section.

VLSI Architecture for the Proposed Image Watermarking Algorithm

The architecture for the proposed image watermarking algorithm involves the following units:

1. Watermark Embedding Unit

2. Watermark Retrieval Unit

3. Control Unit

4. ROM

The function of watermarking unit is to take selected DCT coefficient and watermark bit as the input and the watermarked DCT coefficient as the output. The selection of DCT coefficient is done by the control unit. DCT coefficients are16 bits wide. The embedding unit involves operations like division, addition and multiplication. In-built multiplier is used which is faster than the Wallace tree multiplier. Carry look ahead adder is used for 16 bit addition. Division of 16 bit numbers is done using subtractor constructed with the carry look ahead adder. Similarly, watermark is extracted using watermark retrieval unit. The input to the retrieval unit is the watermarked DCT coefficient and the output is the retrieved watermark bit. The architecture of the retrieval unit contains divider, adder and comparator. The function of control unit is to select the DC coefficient of every 8x8 DCT block of the DCT transformed image. ROM units are used to store the coefficients of the DCT transformed image, watermark bits and the retrieved watermark bits. Four ROM units are needed in the architecture. The proposed watermarking chip is shown in fig. 5.

IV. EXPERIMENTAL RESULTS & DISCUSSION

In order to test the robustness of the proposed algorithm, two assessment metrics for the watermark logo has been used. Normalized Correlation (NC) metric is given by (6).

$$NC = \frac{\sum_{i=1}^{N} w * w'}{\sum_{i=1}^{N} w^2} \quad \ldots (6)$$

Where w is the original watermark and w' is the retrieved watermark. N is the size of the watermark. For two identical images, the value of NC is equal to one. The value of NC is less than one if the two images have some amount of dissimilarity. However, the two images are said to be identical if NC is greater than or equal to 0.80. For two completely different images, NC is zero. Another parameter used to test the robustness of the algorithm is Tamper Assessment Function (TAF) which is given by (7).

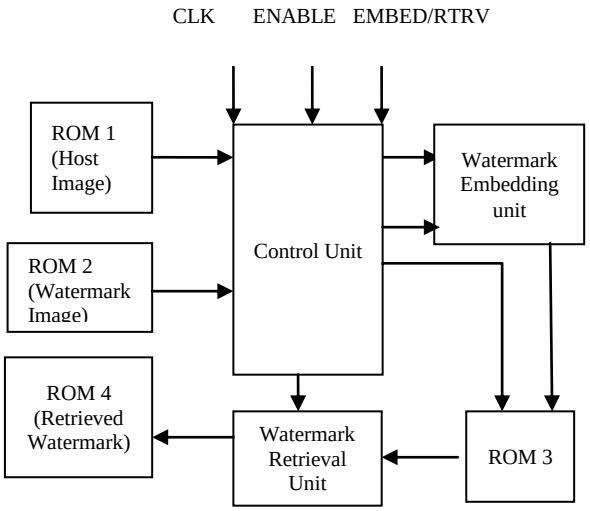


Figure 4. VLSI architecture for the proposed image watermarking algorithm

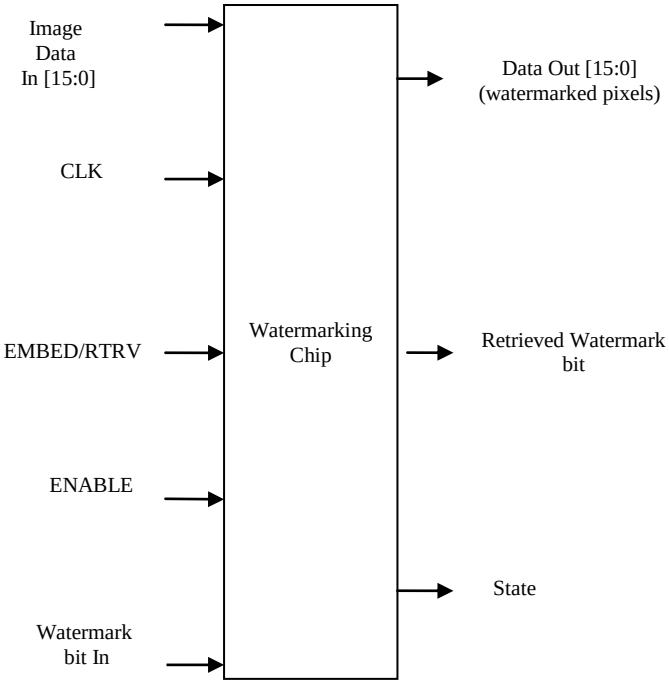


Figure 5. Proposed watermarking chip

$$TAF = \frac{1}{U * V}\left[\sum_{i=1}^{U*V} w(i) \oplus w'(i)\right] * 100 \quad \ldots (7)$$

Where ⊕ is the XOR operation between w (i) and $w'(i)$ bits. For two identical images, the value of TAF is zero. The smaller the TAF, the more similar are the images.

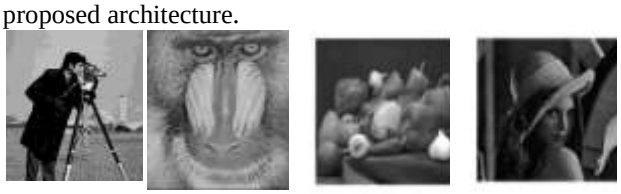
The amount of distortion in the host image caused by the watermarking process is given by Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR). MSE is defined by (8). PSNR for an eight bit gray scale image is given by equation (9).

$$MSE = \frac{1}{J*K}\left[\sum_{x=1}^{J}\sum_{y=1}^{K}(f(x,y) - f'(x,y))^2\right] \quad \ldots(8)$$

$$PSNR\ (dB) = 20\log_{10}\frac{255}{MSE} \quad \ldots(9)$$

Where JxK is the size of the host image, f(x, y) is the original image pixel value and $f'(x,y)$ is the watermarked image pixel value. The value of PSNR above or equal to 40 dB ensures good quality of the image.

For testing the algorithm, images of Lena, Baboon, Cameraman and Peppers are used. Fig. 6 shows the host images and fig. 7 shows the watermark images. In order to verify the performance of the proposed watermarking algorithm, PSNR, MSE and NC values are calculated and are reported in table I. The value of PSNR is satisfactorily good (above 40 dB) in all cases which implies the validity of the proposed algorithm. NC is excellent (equal to 1 in every case) that shows the successful retrieval of the watermark image without distortion. The original host image is taken to be of 512x512 in size and the watermark is a binary image of size 128x128 and the value of N=8. For testing the robustness of the algorithm, following attacks are considered: Blurring (5 degrees), 3x3 Median filtering, 50% Resizing and Gaussian noise (0.001). The proposed architecture is designed using VHDL and simulated in XILINX ISE 9.1. Spartan 3 family device is used for synthesis. The FPGA used is XCS200-4-FT256. The synthesis report and device utilization for the proposed architecture using XCS200 FPGA is given in table III and IV. The architecture designed using FPGA is suitable for real time applications because it provides less delay in computation which results in higher speed. Maximum combinational path delay in watermark embedding unit is 39.085ns and in watermark retrieval unit is 48.123ns. The results of simulation and synthesis done in XILINX ISE 9.1 for XCS200 FPGA clearly indicates that the architecture for the proposed image watermarking algorithm is area efficient and has high speed of operation. For three adder/subtractor circuits, six multiplexer units and two 8-bit comparators used in the embedding unit, only 24 slices and 42 LUTs are used. One in-built 18x18 multiplier is used. In retrieval unit, 17 slices and 30 LUTs are used. These figures clearly indicate the area efficiency of the proposed architecture.



(a) Cameraman (b) Baboon (d) Peppers (c) Lena

Figure 6. Host images

(a) (b)

Figure 7. Watermark images

TABLE I. PERFORMANCE RESULTS OF THE PROPOSED ALGORITHM

Watermark Image	Host Image	MSE	PSNR	NC
FPGA	Lena	1.3060	46.9715	1
MATLAB	Lena	1.3240	46.9118	1
FPGA	Cameraman	1.3564	46.8070	1
MATLAB	Cameraman	1.3256	46.9068	1
FPGA	Baboon	1.3182	46.9309	1
MATLAB	Baboon	1.3368	46.8701	1
FPGA	Peppers	1.3536	46.8158	1
MATLAB	Peppers	1.3164	46.9369	1

TABLE II. ROBUSTNESS OF THE PROPOSED ALGORITHM AGAINST SEVERAL ATTACKS

Attack	NC	TAF
Blurring (5 degree)	0.9097	9.3750
Median Filtering	0.9806	2.4658
Resizing (50%)	0.9919	0.8057
Gaussian noise (0.001)	0.9368	6.1768

TABLE III. SYNTHESIS REPORT

Resources	Embedding Unit	Retrieval Unit
Adder/Subtractor	3	2
Multiplexer	6	7
8-bit Comparator	2	3
Latches	-	1

TABLE IV. DEVICE UTILIZATION

Resources	Embedding Unit	Retrieval Unit
Slices	24/4656	17/4656
LUTs	42/9312	30/9312
I/Os	45/232	17/232
MULT 18x18	1/20	-
Gate Count	252	257

V. CONCLUSIONS

The paper focuses on real time robust image watermarking algorithm and its hardware architecture. Computational complexity of the algorithm is reduced by using the integer DCT and minimizing the use of complex arithmetic operations like multiplication and division. Multiplication is done using the multiplier in-built in XCS200 FPGA which accounts to higher speed of operation. The performance of the algorithm is verified against various image processing attacks which is shown in table II. The performance of the algorithm is measured in terms of MSE, PSNR and NC, which is shown in table I. The architecture of different modules are synthesized and simulated on FPGA to validate the real time watermarking process. Area efficiency of the proposed architecture is verified through the device utilization and synthesis report given in table III and IV.

Acknowledgment

The authors would like to express their deep expression of gratitude to Director, NITTTR, Chandigarh and HOD, ECE, MNIT, Jaipur for their kind support.

REFERENCES

[1] Coatrieux, Gouenou, Catherine Quantin, Julien Montagner, Maniane Fassa, François-André Allaert, and Christian Roux, "Watermarking medical images with anonymous patient identification to verify authenticity", Studies in health technology and informatics, Vol. 136, pp. 667-672, 2008.

[2] Rajendra Acharya, U., Debiprasad Acharya, P. Subbanna Bhat, and U. C. Niranjan, "Compact storage of medical images with patient information", , IEEE Transactions on Information Technology in Biomedicine, Vol. 5, Issue 4, pp.320-323, 2001.

[3] Nayak, Jagadish, P. Subbanna Bhat, M. Sathish Kumar, and U. Rajendra Acharya, "Reliable transmission and storage of medical images with patient information using error control codes", Proc. of IEEE INDICON India Annual Conference, pp. 147-150, IEEE, 2004.

[4] Korrai, P. K., and K. Deergha Rao, "Robust transmission of watermarked medical images over wireless channels", Proc. of Global Humanitarian Technology Conference, pp. 242-246, IEEE, 2012.

[5] Umaamaheshvari, A., and K. Thanushkodi, "Robust image watermarking based on block based error correction code" , International Conference on Current Trends in Engineering and Technology, pp. 34-40, IEEE, 2013.

[6] Kamran, Muhammad, and Muddassar Farooq, "An information-preserving watermarking scheme for right protection of EMR systems", IEEE Transactions on Knowledge and Data Engineering, Vol. 24, Issue 11, pp. 1950-1962, 2011.

[7] Richardson, Iain E. The H. 264 advanced video compression standard. John Wiley & Sons, 2011.

[8] Mohanty, Saraju P., C. Renuka Kumara, and Sridhara Nayak, "FPGA Based Implementation of an Invisible-Robust Image Watermarking Encoder", In CIT, pp. 344-353, Springer-Verlag Berlin Heidelberg, 2004.

[9] Mohanty, Saraju P., Elias Kougianos, and Nagarajan Ranganathan, "VLSI architecture and chip for combined invisible robust and fragile watermarking", Computers & Digital Techniques, IET Vol. 1,Issue 5, pp. 600-611, 2007.

[10] Joshi, Amit M., Vivekanand Mishra, and R. M. Patrikar, "FPGA prototyping of video watermarking for ownership verification based on H. 264/AVC", Multimedia Tools and Applications, pp. 1-24, 2015.

[11] Megalingam, Rajesh Kannan, K. B. Venkat, S. V. Vineeth, M. Mithun, and Rahul Srikumar, "Hardware implementation of low power, high speed DCT/IDCT based digital image watermarking", Proc. of International Conference on Computer Technology and Development, 2009, vol. 1, pp. 535-539, IEEE, 2009.

[12] Ghosh, Sudip, Bijoy Kundu, Debopam Datta, Santi P. Maity, and Hafizur Rahaman, "Design and implementation of fast FPGA based architecture for reversible watermarking" Proc. of International Conference on Electrical Information and Communication Technology, pp. 1-6, IEEE, 2014.

[13] Scaria, Arun, D. Badari Nath, M. Nirmala Devi, and N. Mohankumar, "Hardware Implementation of SVD Based Colour Image Watermarking in Wavelet Domain", Proc. of International Conference on Process Automation, Control and Computing, pp. 1-5, IEEE, 2011.

[14] Kiran, S., K. V. Nadhini Sri, and J. Jaya, "Design and implementation of FPGA based invisible image watermarking encoder using wavelet transformation", Proc. of International Conference on Current Trends in Engineering and Technology, pp. 323-325, IEEE, 2013.

[15] Joshi, Amit M., Vivekanand Mishra and R. M. Patrikar, "Design of real-time video watermarking based on Integer DCT for H. 264 encoder", International Journal of Electronics, Vol. 102, Issue 1, pp. 141-155, 2015.