# Dual Purpose FWT Domain Spread Spectrum Image Watermarking in Real-Time

Santi P. Maity[a,*], Malay K. Kundu[b], Seba Maity[c]

[a]*Dept. of Information Technology, Bengal Engineering and Science University, Shibpur P.O. Botanic Garden, Howrah -711 103, India*

[b]*Center for Soft Computing Research and Machine Intelligence Unit, Indian Statistical Institute, 203, B. T. Road, Kolkata 700 108, India*

[c]*Dept. of EI & ECE, College of Engineering & Management, Kolaghat, P.O. Mecheda, Midnapur East-721 171, India*

## Abstract

Spread spectrum (SS) watermarking for multimedia signal becomes appealing due to its high robustness attribute and is used widely for various applications. Some of these applications essentially demand development of low cost algorithms so that they can be used for real time services such as broadcast monitoring, security in communication etc. In recent time one popular non-conventional application of digital watermarking becomes promising that assesses blindly the QoS (quality of services) of the multimedia services which is expected to be offered by the future generation mobile radio network. Majority of the existing SS watermarking schemes suffer from high computation cost and complexity leading to the difficulty for realtime implementation and limits their usage for the above mentioned applications. This paper proposes Fast Walsh Transform (FWT) based SS image watermarking scheme that serves the dual purposes of authentication in data transmission as well as QoS assessment for digital media through dynamic estimation of the wireless channel condition. Fast Walsh transform offers low computation cost for implementation, smaller change in image (multimedia signal) information due to data embedding and ease of hardware realization. VLSI implementation using Field Programmable Gate Array (FPGA) has been developed to make it suitable for real time implementation.

*Key words:* Authentication; digital watermarking; fast Walsh transform; FPGA; QoS; VLSI

[*] Corresponding author

*Email addresses:* `santipmaity@it.becs.ac.in` (Santi P. Maity), `malay@isical.ac.in` (Malay K. Kundu), `sebamaity@eie.cemk.ac.in` (Seba Maity).

## 1 Introduction

Recent years have witnessed a prolific growth in digital techniques as well as in wireless communication system. Two fold advantages namely (i) the wide use, ease of copying, manipulation and distribution of multimedia signals over Internet and (ii) worldwide mobility between the transmission and reception system have now been achieved. Today various wireless mobile communication services offer data transmission along with voice based applications[1]. Two classes of problems have also been emerged namely (i) how to protect the ownership, authenticity, integrity and security of the transmitted digital data, and (ii) how to ensure end-to-end quality of the offered multimedia services in third or future generation mobile communication system [IMT2000/Universal Mobile Telecommunication System (UMTS)][2]. Digital watermarking scheme, though originally developed as a potential solution for copyright protection and authentication of digital data [3–5] has also been attempted in recent time for non-conventional use like blind assessment of the quality of services (QoS) [6,7] for multimedia signals. Reference watermark pattern (already available to the end user) is embedded into the multimedia host data (called watermarked data after embedding) and is transmitted through the channel. Like a tracing signal, the watermark tracks the host data, since both the host and the watermark follow the same communication link and suffer the same channel degradation. The alteration in watermark is used to estimate wireless channel condition dynamically which in turn assess the quality of the offered services.

The critical difference between these two classes of applications of digital watermarking lie in their mode of implementation. While many of the conventional applications (first type) may be implemented offline, the latter type essentially demands real time realization. Similarly while design of robust watermarking is important for ownership and copy protection of digital media, a preferable solution might be to design a fragile watermark for authentication. Fragile watermarking is also a preferable choice while digital watermarking is used for QoS assessment of multimedia signal [6,7]. However, as the reference watermark is available to the end user in QoS assessment application, the very purpose of digital watermarking for authenticating the transmitted message or sender information is not fulfilled by this type of watermarking application. So an important question arises how to design a digital watermarking algorithm which can simultaneously meet these two different requirements. Intuitively, embedding of two different watermarks in a cover signal may be a viable solution. Spread spectrum (SS) modulation based watermarking scheme allows insertion of multiple watermark information by exploiting orthogonality among the code patterns.

Spread spectrum is accomplished by spreading a narrow band watermark into wide spectrum of the cover so that watermark energy for each frequency bin becomes less and could hardly be detectable [8]. Several SS watermarking schemes for multimedia signals are developed using DCT [9], Fourier-Mellin [10] and Wavelet [11] transforms. However, these algorithms are not suitable for broadcast monitoring and in many other realtime applications due to high computation cost and complexity which makes their hardware realization complicated. Hardware implementation of digital watermarking technique offers advantages of real time processing of data [12,13]. The overall advantage is that hardware consumes less area and less power [14]. If a chip is fitted in the digital devices, the output video or images can be marked right at the origin although the same can be done using software after those videos or images downloaded to the computer. But, in this case embedding software will take more time compared to hardware. The example of TV broadcast highlights the significance where digital media is to be marked in real time and hardware is the only solution [15].

This paper proposes Fast Walsh Transform (FWT) domain SS image watermarking scheme that serves the dual purpose of authentication in data transmission as well as dynamic estimation of the wireless channel condition. The algorithm can be extended to video signal applications by embedding watermark information in different frames. FWT becomes attractive choice for embedding domain due to its low computation cost of implementation, ease of hardware realization, low processing noise arising out of lower quality compression, and smaller change in image (multimedia signal) information due to data embedding. Spread spectrum (SS) methodology is used as this has proven to be efficient, robust and cryptographically secured. Circuits for watermark embedding and decoding are developed that ensures the suitability of the algorithm for such interesting applications in real time environment. The performance of the algorithm is tested for authentication of host data and to mitigate multipath effect of Rayleigh fading environment coupled with corruption of additive noise followed by JPEG and JPEG 2000 (EZW or SPIHT) compression.

The paper is organized as follows: Section 2 introduces with the purpose of the work, review of the related previous works with limitations and scope of the work. Section 3 describes change in image information due to embedding in Walsh coefficients. Section 4 briefly describes the mathematical model of SS watermark embedding and decoding. Section 5 describes the proposed low cost SS watermarking algorithm while section 6 presents its VLSI architecture. Section 7 presents performance evaluation with discussion on QoS and hardware realization. Conclusions are drawn in section 8 along with future scope of work.

## 2 Purpose of the work, review of related works and scope of the work

In this section, we first highlight the purpose of the digital watermarking applications considered in this paper, discuss few related works along with their limitations and lastly the scope of the present work.

### *2.1 Purpose of the work*

Digital watermarking principle has been attempted *separately* for authenticating the host data or sender information [3,4,17] and blind assessment of QoS for multimedia signal in mobile radio channel [6,7,16]. The objective of the present work is to integrate both type of diverse applications in a single digital watermarking framework. This can be accomplished by an algorithm which allows embedding of multiple watermark bits in a particular cover region followed by faithful decoding in absence of signal distortion. The watermarking algorithm should not change cover image information much, must be robust to low quality compression operation and can be implemented in real time.

### *2.2 Review of related works & limitations*

The present work focuses on three different aspects namely authentication, QoS assessment and hardware architecture of digital image watermarking technique. That is why this section discusses only the related digital watermarking works.

Several watermarking algorithms have been developed for authentication of digital documents [4,5,17]. Majority of the works identify whether the original data has been modified or not but fail to identify how the data has been tampered with. The problem can be solved if, for example of an image, is divided into blocks and each block has its own authentication mark embedded in it. The modification process will provide a rough idea to distinguish between parts of the image lying intact and parts tampered with.

Campisi et al [6,7] developed digital watermarking algorithm for blind assessment of QoS for multimedia signal in mobile radio channel. DCT (discrete cosine transform) has been chosen as signal decomposition tool in the work (in many image watermarking algorithms) as the most common compression tools for digital images and videos are DCT based JPEG and MPEG respectively. But it is reported in the digital image watermarking literature that most DCT (wavelet-based) domain embedding schemes are very robust to

JPEG (JPEG 2000) compression, but are not very much robust to JPEG 2000 (JPEG) [18,19]. It is reported in [18] that DHT (discrete Hadamard Transform) domain watermarking techniques offer higher robustness against both JPEG and JPEG 2000 compression operation at low quality factor. The authors have argued that higher robustness is possible as the standard deviations of DCT/ wavelet coefficients are higher for the processing noise at low quality compression compared to DHT coefficients. Ho et al [20] have pointed out some advantages of FHT (fast Hadamard transform) in terms of shorter processing time and ease of hardware implementation than most orthogonal transform techniques such as DCT and discrete wavelet transform. They proposed robust watermarking algorithm for the copyright protection of digital images. Image characteristics such as edges and textures are used to determine the watermarking strength factor in order to make a good trade off between robustness and imperceptibility. It is expected that Walsh transform domain embedding would also offer similar robustness performance against both type of compression operations as Walsh and Hadamard transforms have similar kernel nature.

Over the past decade, several watermarking algorithms for the multimedia signals have been proposed for software implementation. However, only a few hardware implementations are presented in the literature. A hardware based watermarking system can be designed on a field programmable gate array (FPGA) board, Trimedia processor board [21], or custom integrated circuit (IC) [22]. Strycker et. al [23] proposed the implementation of a real-time spatial domain watermark embedder and detector on a Trimedia TM-1000 VLIW processor. The authors in [24] proposed a watermark-based protocol for the document management in large enterprises. Fan et al. [25] have proposed a visible watermarking design based on an adaptive discrete wavelet transform. The authors in [26] proposed the video watermarking algorithms through the hardware implementations of a well-known algorithm called Just Another Watermarking Scheme (JAWS). Tasi and Lu [27] have proposed a DCT domain invisible watermarking chip with TSMC 0.35 $\mu$m technology and has a die size of $3.064 \times 3.064 mm^2$[28]. Garimella et al. [29] have proposed a VLSI architecture for invisible fragile watermarking in the spatial domain. The application specific integrated circuit (ASIC) is implemented using 0.13$\mu$m technology. The critical path delay of the circuit is 5.89 ns. Mohanty et al.[30] have proposed watermarking hardware architecture that can insert two visible watermarks in images in the spatial domain. This architecture can insert either of the two watermarks depending on the requirements of the user. Mohanty et al. [31] have also proposed another VLSI architecture that can insert invisible or visible watermarks in images in the DCT domain. A prototype VLSI chip has been designed and verified using various Cadence and Synopsis tools based on TSMC 0.25 $\mu$m technology with 1.4M transistors and 0.3mW of average dynamic power.

The discussion in the previous subsections indicate that digital watermarking algorithm intended for the stated applications must be of low computation cost and complexity for watermark embedding and decoding, implementable in hardware and capable of faithful assessment of wireless channel condition under fading environment. We propose Fast Walsh Transform (FWT) based SS watermarking scheme for the purpose due to its low computation cost as floating point addition-multiplication is not required when the digital image is convolved with the signed integer valued kernel during the forward and the inverse Walsh transform [32]. In [32], the cover is partitioned into $(8 \times 8)$ block and the blocks are categorized based on variance values. Binary watermark information is embedded redundantly in low and mid variance blocks by substituting the appropriate LSB (least significant bit) plane of the highest AC coefficients. The method shows higher robustness against varieties of signal processing operations but watermark decoding is not blind. Walsh transform shows an ascending of sequency analogous to Fourier transform and unlike random sequency of Hadamard transform [20]. This provides the benefit of Walsh transform computation using fast algorithm which is identical to the FFT leading to the efficient hardware realization. The kernel of Walsh transformation being symmetric matrix with orthogonal rows and columns, the same algorithm can be used for both the 2-D forward and inverse Walsh transforms without modification. Hence, only one hardware block is sufficient to implement both forward and inverse transform which is not possible in DCT based algorithm. It can be shown mathematically that data embedding process causes less change in image information when FWT is used as embedding domain compared to DCT as the former has two valued kernel while the latter has multivalued kernel. Moreover, the wide usage of Walsh codes for implementation of CDMA in wireless communication makes Walsh transform more attractive for the present watermarking application [33].

## 3 Change in image information

We now mathematically prove that image information is changed by less amount in case of Walsh domain embedding compared to DCT (other popular transform) domain embedding. The inverse Walsh transform [34] of an $(N \times N)$ (where $N = 2^n$) image function $f(x, y)$ is written as follows:

$$f(x,y) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} W(u,v) \prod_{i=0}^{n-1} (-1)^{z} \tag{1}$$

where $z = [b_i(x)b_{n-1-i}(u) + b_i(y)b_{n-1-i}(v)]$, $x, y = 0, 1, 2..(N-1)$ and $u, v = 0, 1, 2..(N-1)$. Let the watermark information is embedded in the $(j, k)$-th order Walsh coefficient $W_{j,k}$ where $j, k \neq 0$ and embedding strength is denoted by $\Delta m$. If the watermarked image is denoted by $f_1(x, y)$, where $x, y = 0, 1, 2..(N-1)$, $f_1(x, y)$ can be written as

$$\begin{aligned} f_1(x,y) = [W_{j,k} + \Delta m] \prod_{i=0}^{n-1} (-1)^z \\ + \sum_{u=0,u\neq j}^{N-1} \sum_{v=0,v\neq k}^{N-1} W_{u,v} \prod_{i=0}^{n-1} (-1)^z \end{aligned} \tag{2}$$

The change in the pixel values, due to watermark embedding, can be obtained by subtracting equation (1) from (2) and is expressed as follows:

$$\Delta f(x,y) = \Delta m \prod_{i=0}^{n-1} (-1)^z = \pm \Delta m \tag{3}$$

where, according to the property of Walsh kernel, the product of exponent of (-1) is **0** for half of $x$ and $y$ values and **1** for the remaining $x$ and $y$ values. So the pixel values are increased or decreased by $\Delta m$ respectively and the above relation is true for any $u = l, v = k$ where $l, k \neq 0$. If watermark information is embedded in the coefficient $u = l, v = k$ where $l, k \neq 0$ for other transformation, say DCT, the change in pixel values can be written similar to the equation (3) as follows:

$$\begin{aligned} \Delta f(x,y) = f_1(x,y) - f(x,y) \\ = \Delta m \cos[\frac{(2x+1)l\pi}{2N}] \cos[\frac{(2y+1)k\pi}{2N}] \end{aligned} \tag{4}$$

Equation (4) shows that the amount of changes in pixel values are different for different pixels and the values also depend on the choice of the particular coefficient i.e. $u$ and $v$ values to be used for embedding. The result is also true for other popular transforms such as DFT, Fourier-Mellin, and wavelet etc. So the results in equations (3) and (4) can be summarized as follows:
*If watermark information is added to any $(u, v)$-th Walsh coefficient of the image block, half of the pixel values of the block are incerased by $\Delta m$ and the remaining half of the pixel values are decreased by $\Delta m$. Thus average image information (entropy) is changed by less amount as can be shown due to Shannon [35] or Pal et al [36] compared to DCT domain embedding. On the other hand, for same amount of data embedding using other popular transforms, image information is changed more as in such cases different pixel values are changed by different amount due to the multivalued kernels.*

## 4 SS watermark embedding and decoding

The following subsections describe mathematical model of SS watermark embedding and decoding.

### *4.1 Spread spectrum watermark embedding*

Let the symbol $B$ denotes the binary valued watermark bit string as a sequence of $N$ bits.

$$B = \{b_1, b_2, b_3, ......., b_N\}, b_i \epsilon \{1, 0\} \tag{5}$$

Let the symbol $I$ denotes the image of size $(Q \times Q)$. A binary valued code pattern of length $M$ is used to spread each watermark bit. Thus a set $P$ of $N$ code patterns, each of length $M$, are generated to form watermark sequence $W_Q$ by performing the following operation [37].

$$[W_Q] = \sum_{j=1}^{N} b_j.[P_Q]_j \tag{6}$$

where $[P_Q]_j$ represents a binary valued code pattern matrix of size $M = (Q \times Q)$ corresponding to $j$-th bit of the watermark. The watermarked image $I_W$ can be obtained by embedding watermark information $W$ into the image block $I$. The data embedding can be expressed mathematically as follows:

$$[(I_w)_Q] = [I_Q] + \alpha.[W_Q] \tag{7}$$

where $\alpha$ is the gain factor or modulation index and its proper choice will optimize the maximum amount of allowed distortion i.e. change in structural information of the watermarked image and the minimum watermark energy needed for reliable detection.

### *4.2 Spread spectrum watermark decoding*

In SS watermarking, the detection reliability for the binary valued watermark data depends on the decision variable $t_i$ obtained by evaluating the zero-lag spatial cross-covariance function between the image $I_w$ and each code pattern $P_i$ [38]. The decision variable $t_i$ can be mathematically represented as follows:

$$t_i = < P_i - m_1(p_i), I_w - m_1(I_w) > (0) \tag{8}$$

where $m_1(S)$ represents the average of the sequence $S$. If $s_k$ represents the elements of $S$ with $k$=1,2,3.....M, $m_1(S)$ can then mathematically be expressed as follows:

$$m_1(S) = 1/M\sum_{k=1}^{M} s_k \tag{9}$$

The symbol (0) in Equation (8) indicates the zero-lag cross-correlation and for two sequences $S$ and $R$, the zero-lag cross-correlation is given by

$$< S, R > (0) = 1/M\sum_{k=1}^{M} s_k r_k \tag{10}$$

where the symbols $s_k$ and $r_k$ are the elements of sequences $S$ and $R$ respectively with k=1,2,3......M. If the code patterns $P_i$ are chosen so that $m_1(P_i)$=0 for $\forall$ $i$, the computation of $t_i$ becomes;

$$t_i = < P_i, [I + \alpha.\sum_{j=1}^{N} b_j.P_j - m_1(I)] > \tag{11}$$

$$= < P_i, I > +\alpha.\sum_{j=1}^{N} b_j. < P_i.P_j > - < P_i, m_1(I) > \tag{12}$$

$$= < P_i, I_w > \tag{13}$$

The first and the second term in Equation (12) represents the host signal interference (HSI) and the multiple bit interference (MBI) effect respectively. The i-th embedded bit is detected as follows:

$$b_i = sgn(t_i) = sgn(< P_i.[I + \alpha.\sum_{j=1}^{N} b_j.P_j] > (0)) \tag{14}$$

where *sgn* represents signum function and acts as a hard detector. The bit $b_i$ is detected as **0** if $t_i > 0$ and as **1** otherwise.

With this mathematical model of SS watermark embedding and decoding, we propose a fragile SS watermarking algorithm for digital images. The algorithm is designed to serve a specific purpose of blind assessment of quality of services for the signal transmitted through mobile radio channel as well as authentication of the host data. Low implementation cost for watermark embedding and decoding are essential requirement, that is why the algorithm proposed here is denoted as *low cost spread spectrum watermarking.*

## 5 Proposed low cost SS watermarking algorithm

Like other conventional watermarking methods, the proposed algorithm consists of two parts namely watermark embedding and watermark decoding.

### *5.1 Watermark embedding*

The block diagram representation of watermark embedding process is shown in Fig.1. Different steps for watermark embedding are described as follows:

*Step 1: Image decomposition*

The cover image of size $(M_c \times N_c)$ is partitioned into $(8 \times 8)$ non-overlapping blocks. Each image block is then decomposed using Fast Walsh transform. The size of the image block is considered $(8 \times 8)$ in order to make the scheme compatible with JPEG compression operation.

*Step 2:Formation of message vector*

Let $(M_m \times N_m)$ be the size of the tracing watermark message which is converted into a vector of size $[(M_m.N_m) \times 1]$, called as message vector. Each element of the message vector is either '1' or '0'. The total number of bits of the message vector is $(M_m.N_m)$.

*Step 3:Generation of code patterns*

The widely used code pattern for SS modulation technique is pseudo noise (PN) sequence and is generated using LFSR (Linear feedback shift register)[39]. The size of the PN sequence is identical to the size of the Walsh coefficient matrix. Thus a set of PN matrices denoted by $(P_i)$ of number $(M_m.N_m)$ are generated. It is reported in [8] that cross-correlation values among the code patterns decrease if the later is modulated by Hadamard matrix.

*Step 5:Watermarked image formation*

It is preferable to use antipodal signaling scheme for data embedding in order to increase robustness performance. So the data embedding rule can be expressed as follows:

$$X^e = \begin{cases} X + kP, & \text{if } b = 0 \\ X - kP, & \text{if } b = 1 \end{cases}$$

where $X$ is the Walsh coefficient of the cover image, $X^e$ is the Walsh coefficient after watermark embedding, $k$ is the modulation index, $P$ is the PN matrix.

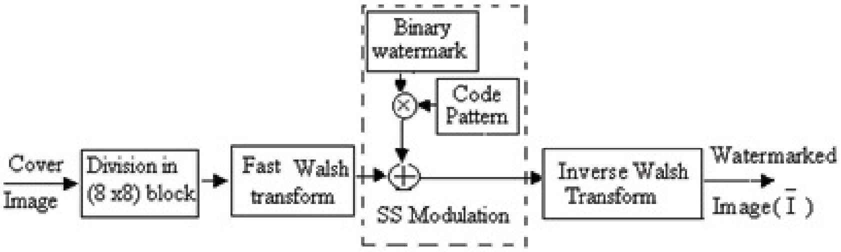

Fig. 1. Block diagram of watermark embedding

Two dimensional block based discrete inverse Walsh transform of the modified coefficients would then generate the watermarked image.

### *5.2 Watermark decoding*

The block diagram representation of watermark decoding process is shown in Fig.2. The watermark recovery process requires the sets of PN matrices $(P_i)$ that were used for data embedding. Different steps for watermark decoding are described as follows:

*Step 1: Watermarked image decomposition*

The received watermarked image is partitioned into $(8 \times 8)$ non overlapping blocks and is decomposed using fast Walsh transform.

*Step 2: Correlation calculation*

Correlation values between Walsh coefficients matrix and each code pattern of the set $(P_i)$ are calculated. We have a total of $(M_m.N_m)$ (equal to the number of watermark bits) correlation values $\mu_i$ where $i = 1, 2, ..M_m.N_m$.

*Step 3: Mean correlation calculation and threshold selection*

We calculate the mean correlation value (T) from these correlation values. This mean correlation value is used as the threshold or decision variable for binary watermark decoding. The decision rule for the decoded watermark bit is as follows:
(i) for $\mu_i \geq T$, the extracted bit is *0*
(ii) for $\mu_i < T$, the extracted bit is *1*

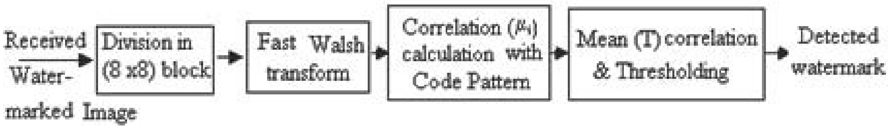
Received Water-marked Image
Division in (8 x8) block
Fast Walsh transform
Correlation ($\mu_i$) calculation with Code Pattern
Mean (T) correlation & Thresholding
Detected watermark

Fig. 2. Block diagram of watermark decoding

## 6 VLSI design

The VLSI architecture of the proposed algorithm is designed using XILINX SPARTAN series FPGA. There are two main subblocks, one is the watermark embedding unit and the other one is the watermark decoding unit. The over all function of the watermark embedding unit is to decompose the image signal using Walsh transform and then embedding the watermark while the decoding unit decodes the embedded watermark. We develop here the architecture for the gray scale cover image of block size ($8 \times 8$), 8-bits/pixels. The watermark consists of 4 bits binary pattern of 0101 and are embedded in this ($8 \times 8$) image block.

### *6.1 Architecture for watermark embedding unit*

The VLSI architecture of the embedding unit for the proposed algorithm is shown in Fig. 3. Hardware design consists of four subblocks or modules namely (1) Walsh transform module, (2) Code generation module, (3) Data embedding module and (4) Inverse Walsh transform module. Data i.e. pixel values of the cover image is fed to the input pin G [15:0] of Walsh transform block with the clock C1. The MUX with control input M4 allows the resultant spreading code to be added with Walsh coefficients at desired time. The output from the adder is fed to the G [15:0] input pin of inverse Walsh transform block. Watermarked output is obtained at the output pin of this block. The other MUXs allow the various signals to flow into the inverse transform block at the desired time. The detailed architecture of each subblock is described below. Brief functional overview of some of the macros used in this hardware design is shown in Table 1 and their output waveforms are shown in Fig. 4. Sample programs of these macros are also shown here.

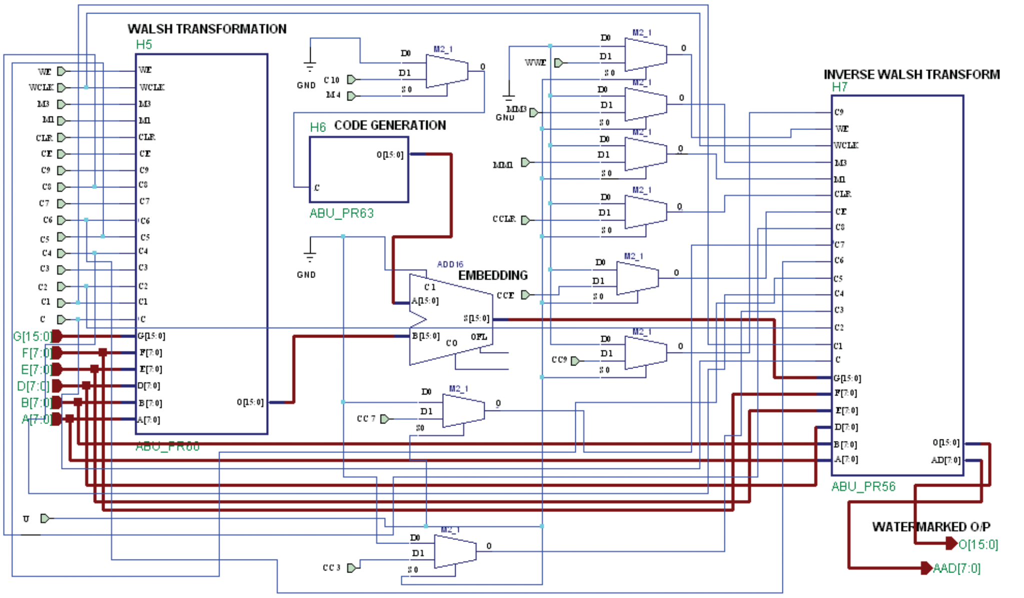

Fig. 3. VLSI architecture of watermark embedding unit

Table 1
Functional overview of different macros

| Unit | Input | Output | Operation |
|---|---|---|---|
| BINARY COUNTER 6 BIT | C,CE | Q[7:0] | Count from 0 to 63 |
| DECODER ( 3 T0 8) | A0,A1,A2 | D[7:0] | Perform $2^N$ where $N$ is input |
| PN SEQU. GENERATOR | C | O | Generate pn sequence |
| BIT REVERSAL MACRO | C,CE | O[7:0] | Generate bit reversed number |

*(1) Sample program for 6 bit binary counter*
(1) clock c 1 0 1 0 1 0 1 0 1 0 1 0 (2) h ce (3) vector q q[7:0] (4) watch q (5) display (6)stepsize 200ns (7) cycle 12
*(2) Sample program for (3 to 8 decoder)*
(1) clock a0 0 1 0 1 0 1 0 1 (2) clock a1 0 0 1 1 0 0 1 1 (3) clock a20 0 0 0 1 1 1 1 (4) vector d d[7:0] (5) watch d (6) display (7)step size 600ns (8) cycle 1
*(3) Sample program for PN sequence generator*
(1) clock c 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 (2) watch o (3) display (4) step size 100ns (5) sim 12800ns
*(4) Sample program for 6 bit reversal macro*

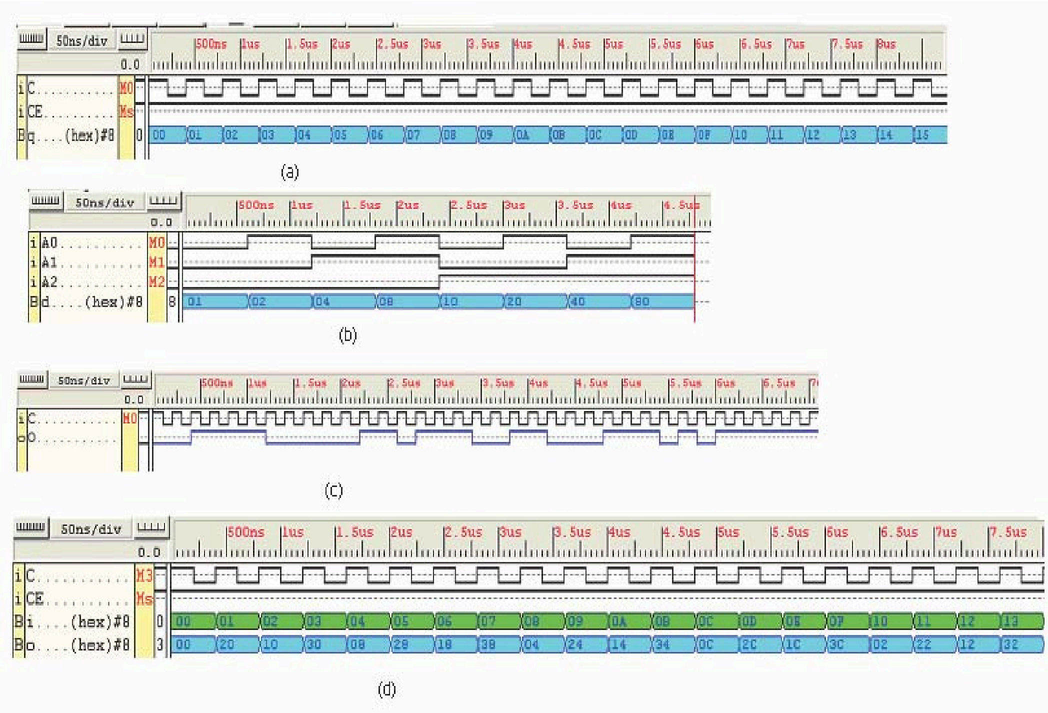
50ns/div
0.0
500ns 1us 1.5us 2us 2.5us 3us 3.5us 4us 4.5us 5us 5.5us 6us 6.5us 7us 7.5us 8us
i C.......... M0
i CE.......... Ms
B q....(hex)#8 0
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15
(a)
i A0.......... M0
i A1.......... M1
i A2.......... M2
B d....(hex)#8 8
01 02 04 08 10 20 40 80
(b)
i C.......... M0
o O..........
(c)
i C.......... M3
i CE.......... Ms
B i....(hex)#8 0
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13
B o....(hex)#8 3
00 20 10 30 08 28 18 38 04 24 14 34 0C 2C 1C 3C 02 22 12 32
(d)

Fig. 4. Various waveform viewers (a) 6-bit binary counter output;(b)Decoder's output (3 to 8); (c)PN sequence generator's output; (d) bit reversal (6-bit) output

(1) clock c 1 0 1 0 1 0 1 0 1 0 1 0 (2) h ce (3) vector i i[7:0] (4) vector o o[7:0] (5) watch i o (6) display (7) step size 200ns (8)cycle 12

(1) *Walsh transform module*

Walsh transform is computed using fast algorithm given below which is nearly identical to the FFT (Fast Fourier Transform).Subroutine for computing FWT is described as follows:

```
  SUBROUTINE FWT (F, LN)......01
  REAL F[64], T.............. 02
  $N = 2^{LN}$ ........03
  $NV2 = N/2$.......04
 $NM1 = N - 1$.......05
  J=1..........06
 DO   3   $I = 1, NM1$ ....07
       IF $(I \geq J)$   GO TO   1....08
       $T = F(J)$......09
       $F(J) = F(I)$....10
      $F(I) = T$.......11
1         $K = NV2$........12
```

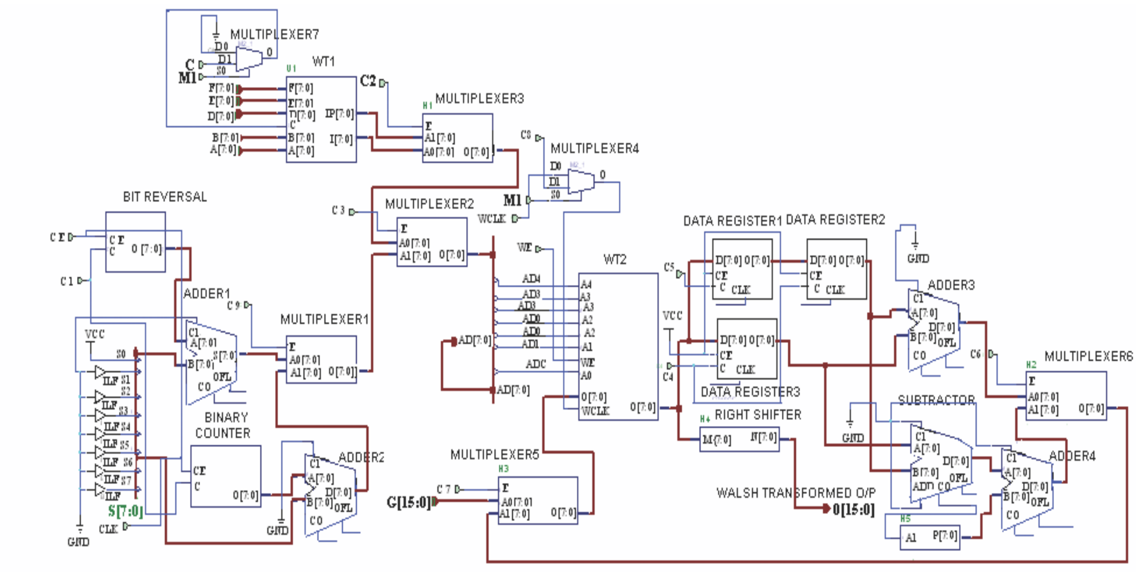

Fig. 5. VLSI architecture of Walsh transform

```
2        IF $(K \geq J)$   GO TO   3....13
       $J = J - K$..........14
       $K = K/2$..........15
      GO TO   2........16
3   J=J+K ....17
  DO   5   L=1, LN......18
       $LE$=$2^L$.....19
      $LE1 = LE/2$.....20
       DO    5    J=1, LE1 .....21
          DO   4   I=J, N, LE ....22
           IP=I+LE1.........23
            T=F(IP)..........24
            F(IP)=F(I)-T.....25
4            F(I)=F(I)+T.........26
5   CONTINUE................27
   DO   6   I=1, N.............28
6   $F(I) = F(I)/FLOAT(N)$.....29
   RETURN .................30
   END .................31
```

Fig. 5 shows the detailed hardware architecture of Walsh transform for an image block of size (8 × 8). In this algorithm statements 03 through 05 are concerned with the initialization of the subroutine. Bit reversal sorting is accomplished by statements 07 through 17. The hardware requirement to imple-

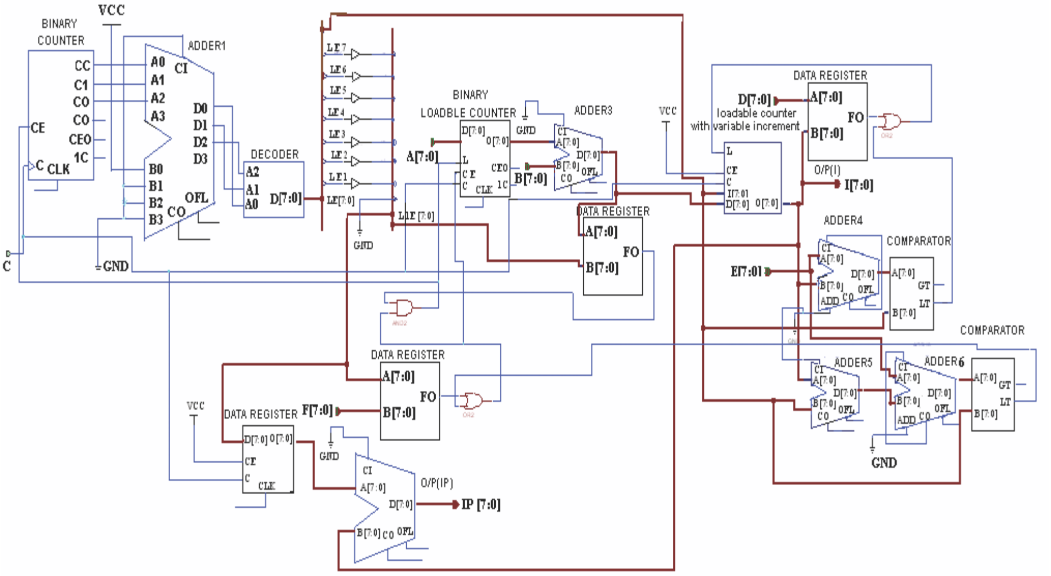

Fig. 6. VLSI architecture of WT1

ment the bit reversal sorting using this algorithm is complex. In this work, the said function is implemented using bit reversal block and WT2 block (RAM). The bit reversal block generates *reversed* addresses. At first C9 input of MUX-1 and C3 input of MUX-2 is kept high and low respectively to allow the bit reversed addresses to be fed to the address pins of the WT2 block. WT2 is a 16 bit RAM with 96 locations out of which 64 locations are used here. The input data is fed to the input pin G[15:0] with the clock C1. C7 input of MUX-5 is kept high to allow the original input data to be fed to the WT2 block. So the data are stored in RAM in a bit reversed order. The WT1 block generates the sequences of I and IP as given in statements 22 and 23. The detailed architecture of WT1 block is shown in Fig. 6.

The outputs of WT1 block are fed into the MUX-3 with control input C2. Proper sequences of IP and I which help to perform the statements 24 through 26 are obtained by applying proper state to the C2 input. C3 input of MUX-2 is kept high to allow these address sequences to be fed into the address pins of RAM. The operations specified by the statements 24 through 26 are performed as follows: RAM is read from two locations specified by the addresses I and IP in two consecutive clocks of WCLK input pin of WT2. The values so obtained are added and subtracted. The results of addition and subtraction are stored back into the RAM in locations specified by I and IP respectively. The complete operations are done using 3 data register, adder 3, subtractor, adder 4, MUX-5 and MUX-6. The read and write operation of RAM is con-

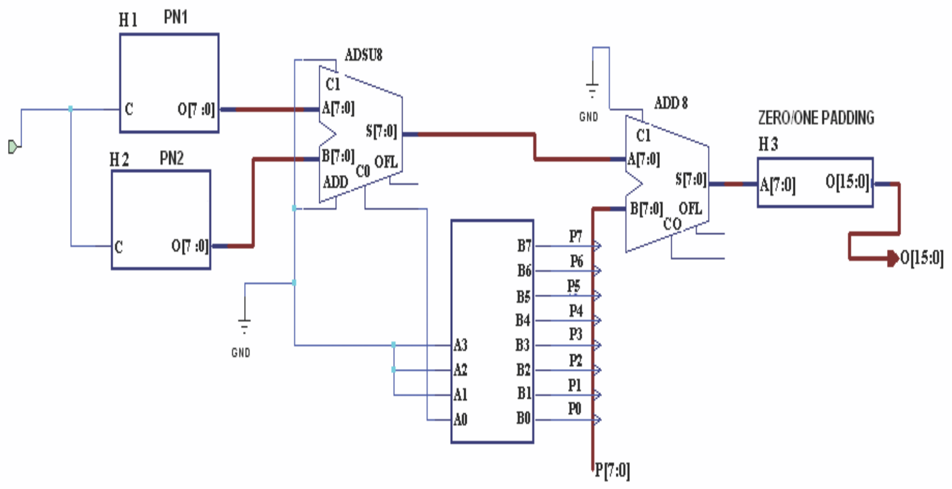

Fig. 7. VLSI architecture of code generation and spread watermark

trolled by WE (write enable) input. Finally the output of the binary counter passes through the MUX-1 and MUX-2 to the address pins of RAM at desired time. The data are read from RAM using these addresses. The output data of RAM is passed through the right shifter to perform the operation of statement 29. Walsh coefficients are obtained at the output pin of the right shifter. The required components for Walsh transform module are two 1-bit MUX (2:1), five 8 bit MUX (2:1), four 8 bit adder, one 8 bit subtractor, one 8 bit binary counter, three 8 bit data register, one right shifter, one bit reversal unit, one WT1 block, one WT2 block.

(2) *Code Generation module*

VLSI architecture of spreading code generation unit consists of the two major sub blocks, PN1 and PN2 blocks. Each block generates two set pseudo noise (PN) sequences of length 64. These PN sequences are added and is obtained at the output of each block. The outputs of PN1 and PN2 blocks are subtracted and the result is passed through a zero/one padding unit. The resultant PN sequence is obtained at the output of padding unit. Fig. 7 shows code generation and spread watermark unit.

(3) *Data embedding module*

The output from code generation unit is added with the output from Walsh transform unit to obtain coefficients of the embedded data.

(4) *Inverse Walsh Transform module*

The kernel of forward and inverse Walsh transform is identical. So the hardware requirement for performing both the operations are also same except an

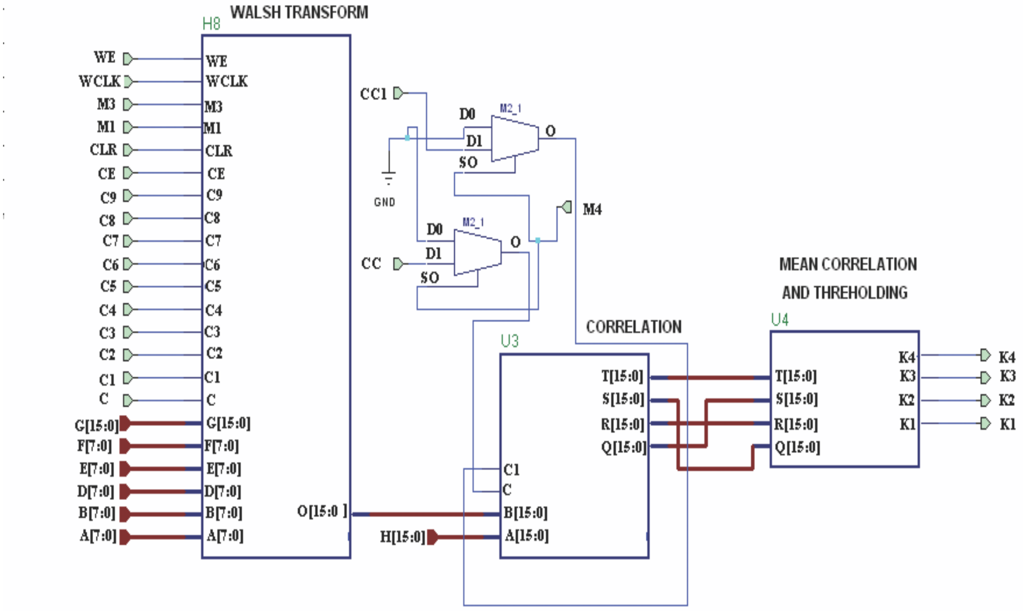

Fig. 8. VLSI architecture of watermark decoding

extra right shifter block that performs the division operation.

## *6.2 Architecture for watermark decoding unit*

The VLSI architecture of watermark decoding is shown in Fig. 8. The major sub blocks are (1) Walsh transform module (2) Correlation calculation module (3) Mean correlation and threshold calculation module.

Watermarked data is fed to the input pin G[15:0] of the Walsh transform block. The output of this block is passed through the correlation calculation block. The function of the correlation calculation block is to calculate the correlation between the spreading functions and Walsh coefficients block. Then the correlation values are passed through a mean correlation and threshold calculation block. At the output of the block, the message bits are detected.
(1) *Walsh transform module*
Walsh transform is applied to the watermarked image block. Theory and hardware architecture of this unit is exactly identical as described in watermark embedding section.
(2) *Correlation calculation module*
The detailed hardware architecture of the correlation calculation block is shown in Fig. 9. The same code generation units PN1 and PN2 used at watermark embedding unit are also used here. Input A[15:0] is set to zero. The Walsh coefficients are applied to the input pin B[15:0] with the clock C. The

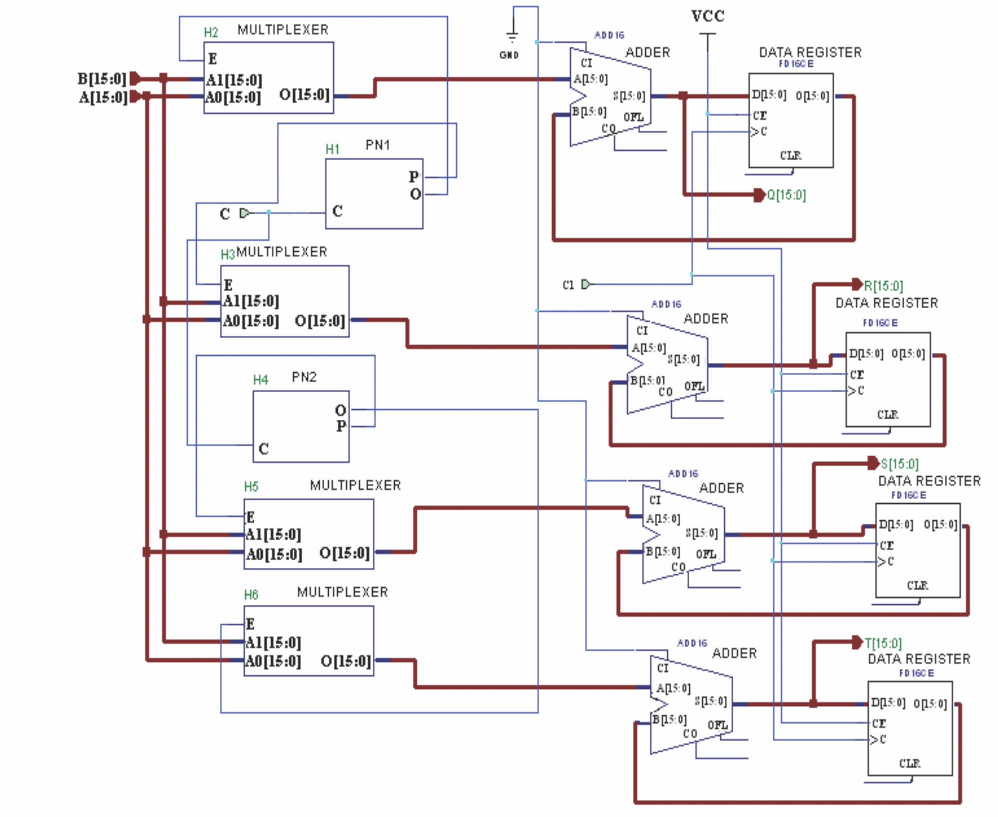

Fig. 9. VLSI architecture of correlation calculation

PN sequences coming from PN1 and PN2 are applied to the control input of MUXs. If the element of the code matrix is ”1”, then it allows the value of B[15:0] to pass through MUXs. On the other hand, if the element of the code matrix is ”0”, it allows the value of A[15:0] to pass through MUXs. The outputs of the MUXs are fed to the one input of the adders unit. The outputs of the adders are fed to the data registers and outputs of the data register are fed back to the other inputs of the adders. Applying proper state sequence to C1, the correlation values Q[15:0], R[15:0], S[15:0] and T[15:0] are calculated. The required components of this unit are four MUXs (16 bit)- 2:1, 4-adders (16 bit), 4 data registers (16 bit), PN1 and PN2 units.

(3)*Mean correlation and threshold calculation module*

The detailed architecture for mean correlation and threshold calculation is shown in Fig. 10. The four correlation values are added using three adders. The result of addition is passed through a right shifter to obtain the mean correlation value. The output of the right shifter block is fed to the one input of each comparator. The other input of the comparators are the correlation values. Message bits are detected at the output of the comparator. The required hardware for this unit are three adders-16 bit, one right shifter, four magnitude comparators -16 bit.

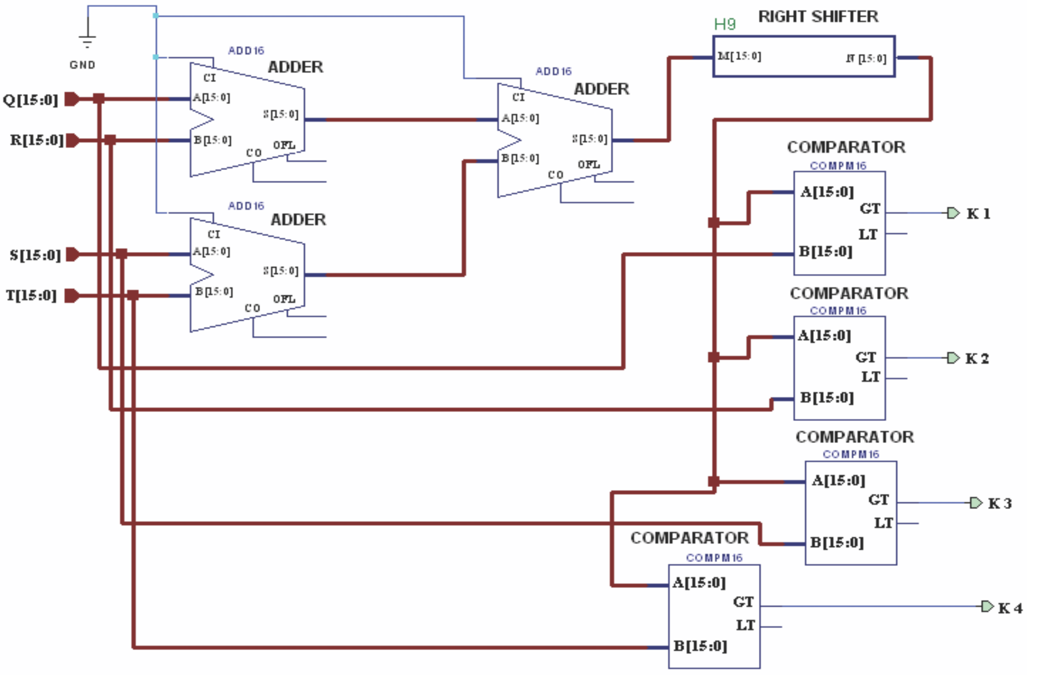

Fig. 10. VLSI architecture of mean correlation and threshold calculation

## 7 Performance evaluation

In this section we report through simulation results (1) advantages of Walsh transform domain embedding compared to DCT/wavelet with respect to data imperceptibility as well as compression resiliency at low quality factor against JPEG and JPEG 2000,(2)results of hardware design in term of number of CLBs (Configurable Logic Blocks) required,(3) performance for authentication and (4) how effective the proposed scheme is to access QoS.

### *7.1 Walsh Transform domain embedding*

This subsection compares the performance of Walsh transform domain embedding with respect to DCT. Fig. 11(a) shows test image Lena which is a 8 bits/pixel gray scale image of size (256 × 256) while Fig. 11(c) and Fig. 11(d) show the watermarked images using DWT (discrete Walsh transform) and DCT as signal decomposition tools respectively. The watermark image used in both cases is the same and is shown in Fig. 11(b) which is a binary image of size (32 × 32). The size of watermark and the cover indicate that each bit of watermark information is embedded in one (8 × 8) non-overlapping block of the cover images. The proposed algorithm took approximately 1 second for both watermark embedding and extraction process using a MATLAB 6 platform running on a Pentium III 400 MHz PC system while the algorithm in [20] took approximately 2 seconds for embedding process and approximately 1 second for extraction in the same computation platform. The visual quality of

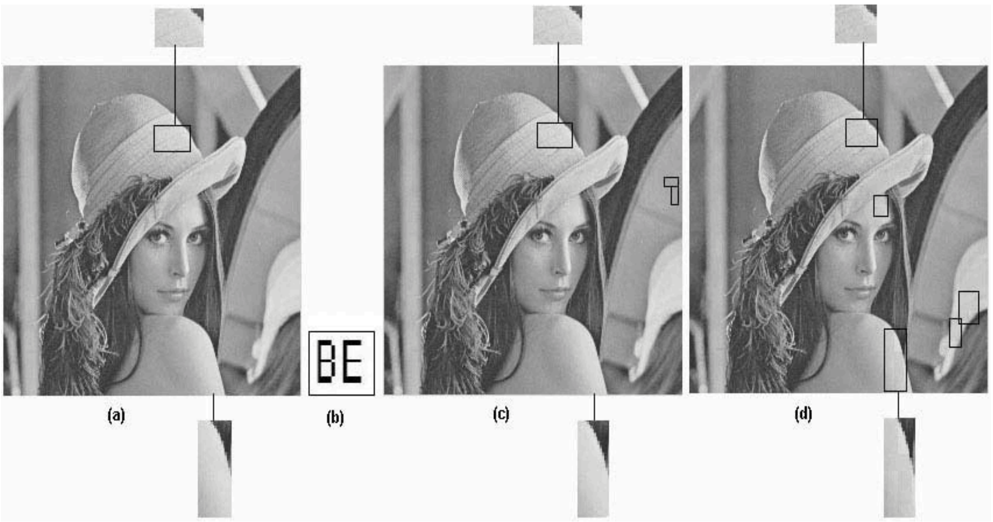

Fig. 11. (a) Cover image,(b)Binary watermark (c) Watermarked image using Walsh Transform (d) Watermarked image using DCT transform

the watermarked image with or without various signal processing operations is denoted by PSNR (peak signal-to-noise ratio) and MSSIM (mean structural similarity index measure)[40] while the visual quality of the extracted watermark is represented by NCC (Normalize cross correlation)[41] and MSE (mean square error)[7]. The rectangular boxes shown in Fig. 11(c) and 11(d) indicate the areas where the visually distinguishable distortions occur. The rectangular zoomed regions in Fig. 11(c) are quiet identical visually to the corresponding regions in Fig. 11(a), however, they have been shown in order to highlight how the corresponding regions in Fig. 11(d) have been degraded severely. On the other hand, there occurs very small distortions in one place on upper middle of right side broader in Fig. 11(c) and this distortion does not occur in Fig. 11(d). But this distortion in Fig. 11(c) is not so much visually prominent. Thus visual distortion shown in Fig. 11(c) is very low and could hardly be perceived while there are noticeable distortions in few places in Fig. 11(d). The distortion occurs possibly due to the contribution of higher change in entropy due to DCT domain embedding. This subjective visual quality of the images are also supported by the objective measures. The PSNR and MSSIM values between the watermarked image and the original image for Walsh transform domain embedding are 41.02 dB and 0.9973 respectively while the respective values for DCT domain embedding are 38.67 dB and 0.9831.

A binary watermark image of size $(64 \times 64)$ is also embedded in a separate context in order to serve the dual purpose of authentication and QoS assessment through 4 bits watermark embedding in each $(8 \times 8)$ block. The PSNR and MSSIM values now become 39.83 dB and 0.9786 respectively which satisfy imperceptibility of the hidden data quite satisfactorily. Out of 4 embedded

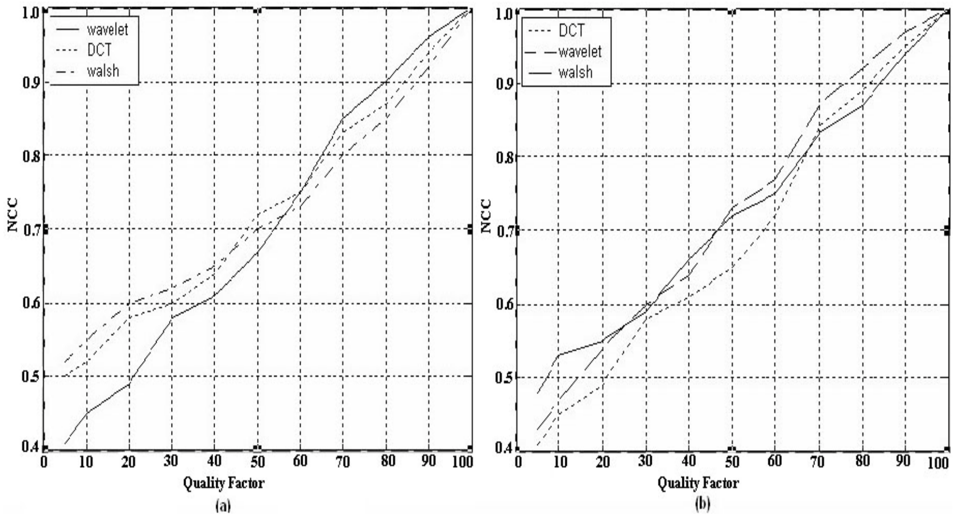

Fig. 12. Robustness performance of the algorithm against(a)JPEG compression operation (b)JPEG 2000 compression operation

bits, decoding reliability of 3 watermark bits indicate relative manipulation of the respective part of the cover image block. The other watermark bit is the part of the reference watermark and its decoding reliability assesses QoS of the digital data. Although performance of the proposed method for authentication and QoS assessment are reported separately in sections 7.3 and 7.4 respectively, the proposed method is efficient to serve the dual purpose through such multiple bit embedding. To show the efficacy of the proposed watermarking algorithm in real time environment, section 7.2 presents hardware design of multiple watermark bit embedding.

We also study the robustness performance of the proposed watermarking method against JPEG and JPEG 2000 (SPIHT) compression operations and the results are reported graphically in Fig 12(a) and Fig. 12 (b). Experiment results support that the selection of Walsh transform as signal decomposition tool shows better performance at low quality compression compared to DCT and wavelet, when both type of compression operations are taken into consideration. Thus the use of FWT, instead to conventional DCT or wavelet, would not limit the applicability of the algorithm.

It is already mentioned that the VLSI design is developed for a gray scale image of size (8 × 8), 8-bits/pixel and the watermark signal is a 4-bit binary pattern 0,1,0,1 i.e. each image block is embedded by multiple watermark bits to serve dual purpose.

Table 2
Specification of Hardware realization for low cost SS watermarking

| System type | Block size | Implementation | CLB count | Clock Freq . (maximum) | Clock cycle |
|---|---|---|---|---|---|
| 4 bit watermark | (8 × 8) | XCS40L | 730 | 80 MHz | 86 cycles |

The running watermarking is updated with arrival of new sample i.e. after the completion of previous watermark decoding and each updating requires 86 clock cycles for (8 × 8) image block. This total clock cycle requirement includes 4 watermark bit embedding and decoding sequentially. However, as these two operations are done at transmitter and receiver separately, the clock cycle requirement for individual operation is significantly less. Moreover, the operation may be done in parallel. The clock cycle requirement for watermark embedding and decoding are reduced significantly if single bit rather than 4 bits are embedded. The maximum clock frequency is 80 MHz and clock cycle 86 cycles/(8 × 8). The data rate can be used is 930.232 Kbits/S. Input specifications of hardware realization is summarized in Table 2. The chip used is XCS40 and XCS40L which contains 784 CLB, out of which 730 CLBs are consumed, 430 for watermark embedding unit and 300 unit for watermark decoding. The design is fully portable and may be integrated into digital still camera framework. Throughput of the proposed architecture may seem to be low, however higher throughput can be achieved if architecture is mapped to higher end FPGA available nowadays such as Virtex PRO etc.

The effect of watermarking using hardware implementation is presented in Table 3 that shows some (half the number of pixel values of the block) sample pixel values, Walsh coefficients of these pixel values in hexadecimal form and the coefficient values (in hexadecimal form) after watermark embedding. It is to be noted that pixel values and Walsh coefficients of a row no way indicate one-to-one correspondence, latter one forms Walsh coefficients matrix which is obtained by applying Walsh kernel over image matrix. Thus it should not be wrongly interpreted that two or more same pixel values show different Walsh Coefficients in hexadecimal form, rather distribution of pixel values are such that when convolved by Walsh matrix produces Walsh coefficients where same pixel values in different locations yield different Walsh Coefficients in hexadecimal form. It is better to consider Walsh coefficients with respect to their positions rather than direct mapping from pixel values. Although it may seem locally too much change by watermarking at glance, the actual change is obtained after applying inverse Walsh transform over the watermarked coefficients. The change in pixel values are quite insignificant and is supported by the Fig. 11(c) (subjective measure) as well as by objective measures of PSNR and MSSIM values of 41.02 dB and 0.9973 respectively. We have performed experiments over large number of test images and results found are satisfactory with same order of objective values.

Table 3
Test results of watermark embedding using hardware realization

| Pixel values | Walsh cof. in Hex. | Walsh Cof. after embed. | Pixel values | Walsh cof. in Hex. | Walsh cof. after embed. |
|---|---|---|---|---|---|
| 57 | 0065 | FFE9 | 58 | FFEF | 0027 |
| 43 | FFFA | 0026 | 39 | FFFE | 0018 |
| 47 | FFFA | 0032 | 76 | 0002 | 003A |
| 166 | FFFF | 00AF | 144 | FFFF | 008C |
| 56 | FFDD | 003D | 57 | FFFB | 0030 |
| 36 | FFFD | 0014 | 43 | 0003 | 0020 |
| 65 | FFFF | 0048 | 102 | 0000 | 005E |
| 138 | FFFE | 008A | 154 | 0002 | 00A5 |
| 53 | FFF4 | 0044 | 52 | FFFE | 0039 |
| 39 | FFFE | 0021 | 59 | FFFB | 001F |
| 93 | FFFE | 0058 | 127 | FFFD | 0072 |
| 149 | FFFE | 00A0 | 152 | FFFC | 0096 |
| 48 | 0000 | 0035 | 47 | 0011 | 002B |
| 52 | 0004 | 002C | 83 | 0002 | 004E |
| 122 | 0004 | 0070 | 146 | 0000 | 0082 |
| 149 | FFFE | 00A2 | 142 | FFFC | 007C |

The four correlation values in hexadecimal obtained are FFDB, FFB6, FFCD, FFA6. The mean correlation value obtained is FFC1. According to step 3 of section 5.2 the detected watermark pattern is 0, 1, 0, 1 which is identical to the embedded watermark pattern.

Table 4 shows the performance comparison of some of the watermarking hardware design in current literature. Notable contributions are found from the research works of Mohanty et al [22,28,30,31]. Majority of the works reported are based on custom integrated circuit while this work is FPGA based realization. To the best of our knowledge, this is the first FPGA based architecture for implementing fragile spread spectrum watermarking using Fast Walsh Transform. Moreover, hardware is designed for multiple bit embedding and the work serves the dual purposes unlike the other hardware design of watermarking algorithms. Though clock frequency used is low compared to most of the other hardware designs reported here, the throughput is very high due to novelty of the proposed algorithm as well as the choice of FWT as the signal decomposition tool. The throughput would certainly be further increased if higher end FPGA is used. This high throughput makes the proposed hardware design attractive for the present dual applications compared to other hardware design.

Table 4
Test results of watermark embedding using hardware realization

| Proposed | Types of | Target | Working | Techno- | Gate | Clock |
|---|---|---|---|---|---|---|
| work | watermarking | object | domain | logy | count/CLB | freq. |
| Mathai [26] | Invis.-robust | Video | Wavelet | 0.18 $\mu$m | NR | NR |
| Tsai [27] | Invis.-robust | image | DCT | 0.35$\mu$m | 46374 | 50MHz |
| Garimella[29] | Invis.-fragile | Image | Spatial | 0.13$\mu$m | NR | 100MHz |
| Mohanty [30] | Visible | image | Spatial | 0.35$\mu$m | 28469 | 292 MHz |
| Mohanty [31] | Invis.-robust | Image | DCT | 0.25$\mu$m | NR | 280 & 70MHz |
| This work | Invis.-fragile | Image | FWT | FPGA | 730 CLB | 80 MHz |

### *7.3 Authentication*

The proposed watermarking method has been tested for authentication purpose and results are shown in Fig. 13. Fig. 13 (a) shows tampering of some randomly selected blocks where all pixel values within the blocks are represented by some fixed random values. The fixed values are different for different blocks. PSNR value of the tampered watermarked image shown in Fig. 13(a) is 39.56 dB. Fig. 13(b) shows the extracted binary watermark image with error indicating the tampering of respective blocks. Similarly, Fig. 13(c) shows the watermarked image where ”lips” and ”nose” have been tampered with. The errors in extracted binary watermark shown in Fig. 13(c) conform this tampering. PSNR value of the tampered watermarked image shown in Fig. 13(c) is 40.52 dB. Since the algorithm employs block based embedding rather

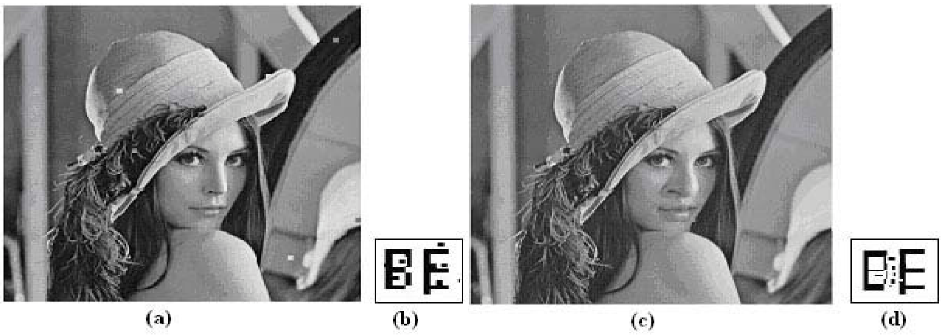

Fig. 13. (a) Watermarked image after deliberate tampering of blocks;(b)Extracted Binary watermark from (a); (c) Watermarked image after tampering ”nose” and ”lips” (d) Extracted binary watermark from (c)

than global embedding in a still image/frame, it is possible to identify relative degradation at different portion within the still image/frame unlike the methods developed in [6],[7]. It is difficult to find proper threshold of allowable image quality degradation for assumed watermarked application to authentication. However, simulation results show that even if there occurs very small change in PSNR value due to tampering, the proposed method can identify the tampered regions. It is needless to mention that higher change in PSNR values due to tampering i.e greater tampering can easily be identified by the proposed method.

### *7.4 Results for QoS assessment*

The proposed watermarking algorithm can also be used for blind assessment of QoS for multimedia signal transmitted through mobile radio channel. Furthermore, we extend this concept to mitigate multipath propagation effect using diversity techniques and study the performance under Rayleigh fading environment.

In UMTS, multimedia signals are compressed first and thus a coded bit stream is obtained. This coded bit stream is then transmitted through noisy channel. Since the original multimedia signal is not available to the MS (mobile station) or end user, the relative quality of the tracing watermark is the indication about the quality of the offered services. The proposed method has been tested for lossy JPEG and JPEG-2000 (SPIHT) coder separately followed by additive

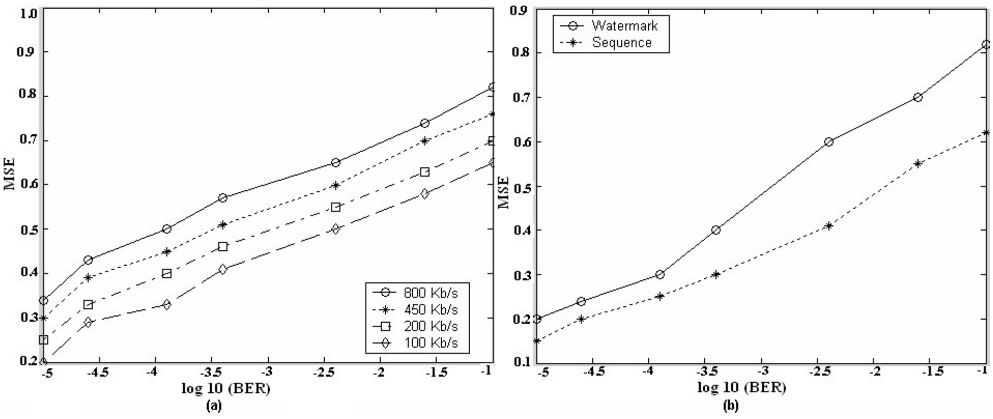

Fig. 14. (a)Watermark MSE (normalized to 1) versus BER for the coded image data for different SPIHT compression ratio;(b)Watermark MSE (normalized to 1) and coded image data MSE versus BER at 100 Kb/s

white Gaussian noise offered by the transmission channel. However, results are reported only for lossy SPIHT compression operation. The relative quality values of the tracing watermarks are represented by Mean Square Error (MSE) between the estimated watermark and the reference watermark. MSE of the extracted watermark, for the i-th transmission channel, can be expressed as follows:

$$MSE_i = \frac{1}{K_1 K_2} \sum_{K_1=1}^{k_1} \sum_{k_2=1}^{K_2} (w_i[k_1, k_2] - w^{'}[k_1, k_2])^2 \tag{15}$$

Let the coded bit stream experiences 'M' number of multiple propagation paths, then the single numerical value that quantifies the quality of the extracted watermark is denoted by

$$MSE = \frac{1}{M} \sum_{i=1}^{M} MSE_i \tag{16}$$

where $M$ are the number of copies for the extracted watermarks.

To show the effectiveness of the proposed algorithm as a means of providing quality measure of the offered services, Fig.14(a) represents the MSE of the extracted watermark (with respect to the original one) versus BER for the received coded bit stream at different compression ratio. It is important to specify how BER in mobile transmission is simulated using compression operation. The watermarked image is first compressed using lossy SPIHT and for each compression ratio a coded bit (binary data) stream is obtained. The binary coded bit stream is digital carrier modulated using BPSK (binary phase shift keying) or QPSK (quadrature phase shift keying) techniques. The n-th bit may be denoted by $a[n]$ where $a_n = \pm 1$ for BPSK scheme. The bit (for BPSK) or symbol (for QPSK) duration is the measure of data rate. Interested readers may go through the work of Maity et al [42] for implementation and performance analysis of mobile transmission channel under Rayleigh fading. The bit rate 100 Kbit/S indicates bit duration of 0.1 microsecond and accordingly bit durations are set for other bit rates. The horizontal axis indicated by BER in Fig. 14(a) and Fig. 14(b) corresponds to different compression ratio with quality factor varying from 90 to 10. It is quite reasonable that as compression ratio increases (low quality images indicated by low quality factor), BER value for the received watermarked image increases. In other words, horizontal axis of Fig. 14 (a) and Fig. 14(b) indicate decrease in quality factor of compression from left to right with higher value (90 quality factor) at left and lower value (10) at right. The graph shows that MSE of the extracted watermark increases as BER i.e compression ratio increases and bit rate increases i.e duration of bit decreases. The result expectedly supports that perceptual degradation of the image data increases with the increase in BER of the watermarked data and increase in transmission bit rate. Fig.14(b) further supports the fact that watermarked data stream (denoted by sequence in graph) and the watermark data are degraded in the similar fashion. This validates our initial hypothesis that the alteration in watermark will indicate the wireless channel condition as well as blind assessment of the quality of the offered services.

It is to be noted that actual BER measurement such as communication loop-back test for mobile phone would be strictly regulated. Moreover, it is not also possible to calculate directly BER values for the received multimedia signal at MS (mobile station). However, it is possible to calculate MSE values for the extracted watermark signal as the reference watermark is available at receiver end. This MSE value would indicate the corresponding BER value of the received or offered multimedia services (quality of the data). It is difficult to specify BER value i.e. compression ratio as requirements of assumed watermark application for assessment of QoS. However, MSE value would indicate relative quality of the offered services. The difference between the regulated BER and the estimated BER is an important point and may act as a feedback information. For example, at a certain high MSE value the mobile station declares a received quality lower than the agreement one. This would imply that the radio mobile channel is not suited for the current bit rate for the given BER and therefore, the bit rate emitted by the base station (BS) is lowered in a few seconds.

In mobile radio channel, multipath propagation effect leads to signal fading. We simulate the effect of Rayleigh fading channel to the watermarked image to characterize the behaviour of multipath channel [42]. We take help of MATLAB *raylrnd (b)* function to simulate the characteristics of fading channel where the mean of the distribution with parameter b is $b\sqrt{\pi/2}$ and variance is $(4-\pi)b^2/2$. MSE values of the extracted watermarks from each multipath can be used as weight factors as the same are determined in maximal ratio combiner (space or antenna diversity) or RAKE receiver (SS time diversity) based on the value of signal voltage to noise power ratio [33]. The higher weight factor is considered for the watermarked signal received from the channel for which MSE value is lower. We achieve quality improvement of the offered services by $\sim 3$ dB at low fading effect by comparing the weight factors calculated from the relative quality measure of the tracing watermarks than the weight factors determined from the $(S+N)/S$ values [33].

The rationale behind such improvement is due to the better accuracy of the assigned weight factor as they are calculated from the comparison of tracing watermarks with reference signals. On the other hand, in the conventional method of [33], weight factors are calculated from random signal analysis and possibly less accurate to represent variable nature of wireless channel condition.

## 8 Conclusions and scope of future work

A low cost SS watermarking scheme along with hardware design is proposed and tested for blind assessment of QoS for digital images. The novelty of the scheme lies in low loss of structural information due to watermark embedding, high resiliency to compression operations and ease of hardware realization that makes it suitable for real time multimedia mobile communication applications. The estimation of the tracing watermark at MS will provide detailed information about the quality of services due to watermark embedding, status of the link, information relating to billing purpose etc. Furthermore, the quality of the tracing watermarks may be explored in diversity techniques for cancelation of the fading effect arising out of multipath propagation. The hardware design of the algorithm is reported for $(8 \times 8)$ and the same can be easily extended for large image size, say $(256 \times 256)$ or $(512 \times 512)$ or even larger for real life application using parallel processing of many such modules. The parallel processing offers simultaneous execution of several hardware units and total time of execution remains unchanged but hardware requirement will be increased. Current work is going on to develop the dedicated digital system using this FPGA chip.

## References

[1] Hanzo, L., Cherriman, P. J., & Streit, J., Wireless video communication: second to third generation systems and beyond, IEEE Series on Digital and Mobile Communication, New York: IEEE, 2001.

[2] Li, F. Y., Stol, N. , T. T. Pham,& Andresen, S., A priority-oriented QoS management framework for multimedia services in UMTS, In Proc. fourth Int. IEEE Symp. Wireless Pers. Multimedia Commun., September 9-12, 2001.

[3] Voyatzis, G.,& Pitas, I., Protecting digital-image copyrights: A framework, IEEE Computer Graph. Applicat., **19**,1999,18-24.

[4] Dittman, J., & Steinmetz, A., Content-based digital signatures for motion pictures authentication and content-fragile watermarking, Proc. IEEE ICMCS, Florence, 1999, 574-579.

[5] Fridrich, J., Goljan, M., & Baldoza, A. C., New fragile authentication watermark for images, Proc. of IEEE Int. Conf. of Image Processing (ICIP), 2000,446-449.

[6] Campisi, P., Carli, M., Giunta, G., & Neri, T., Tracing watermarking for multimedia communication quality assessment, In Proc. IEEE Int. Conf. Commun., Apr.-May 28-2, 2002.

[7] Campisi, P., Carli, M., Giunta, G., & Neri, T., Blind quality assessment for multimedia communications using tracing watermarking, IEEE Transaction on Signal Processing, **51**, 2003, 996-1002.

[8] Maity, S. P., Kundu, M. K., & Das, T. S., Robust wSS watermarking with improved capacity, Pattern Recognition Letters (Advances in Visual Information Processing), Elsevier Science, **28**, 2007, 350-356.

[9] Cox, I. J., Kilian, J., Leighton, T., & Shamoon, T., Secure spread spectrum watermarking for multimedia, IEEE Transaction on Image Processing , **6**, 12, 1997, 1673-1687.

[10] Ruanaidh, J. O., & Pun, T., Rotation, scale and translation invariant spread spectrum digital image watermarking, Signal Processing, **66**, 1998, 303-317.

[11] Grobois, R. & Ebrahimi, T., Watermarking in JPEG 2000 domain, Proc. of the IEEE workshop on multimedia signal proc., 2001, 3-5.

[12] Mohanty, S. P., Ranganathan, N. & Namballa, R. K., VLSI implementation of invisible digital watermarking algorithms towards the development of a secure JPEG encoder,Proc. IEEE Workshop on Signal Processing Systems, 2003,183-188.

[13] Maity, S. P., Banerjee, A., Abhijit, A., & Kundu, M. K., VLSI design of Spread Spectrum watermarking, 13th National Conf. on Comm., IIT Kanpur, India, 2007, 251-257.

[14] Maity, S. P., Banerje, A., & Kundu, M. K., An image-in-image communication scheme and VLSI implementation using FPGA, IEEE Indian Annual Conf. (INDICON 2004), IIT Kharagpur, India, 2004, 6-11.

[15] Cox, I. J., Miller, M. L., & Bloom, J. A., Digital Watermarking, Morgan Kaufmann (San Francisco, 2001).

[16] Maity, S. P., Kundu, M. K., & Maity, S., An efficient digital watermarking scheme for dynamic estimation of wireless channel condition,Int. Conf. on Computing: Theory and Applications. Indian Statistical Institute, Kolkata, India, IEEE CS Press, 2007, 671-675.

[17] Fridrich, J., Image watermarking for tamper detection,Proc. of IEEE Int. Conf. of Image Processing (ICIP), 1998, 404-408.

[18] Ramkumar, M. & Akansu, A. N., Capacity estimates for data hiding in compressed images, IEEE Trans. on Image Proc., **10**, 2001, 1252-1263.

[19] Maity, S. P., & Kundu, M. K., An image watermarking scheme using HVS characteristics and spread transform, Porc. of 17th Int. Con. of Pattern Recognition (ICPR), IEEE CS Press, Cambridge, UK, **4**, 2004, 869-872.

[20] Ho, A. T. S., Shen, J., Chow, A. K. K. & Woon, J., Robust digital image-in-image watermarking using the fast Hadamard transform, Proc. of Int. Symposium on Circuits and Systems (ISCAS 2003), 2003, 826-829.

[21] Maes, M., Kalker, T., Linnartz, J. P. M. G., Talstra, J., Depovere, G. F. G., & Haitsma, J., Digital watermarking for DVD video copyright protection, IEEE Signal Processing Magazine, **17**, 2000, 47-57.

[22] Mohanty, S. P., Ranganathan, N., & Namballa, R. K., VLSI implementation of invisible digital watermarking algorithms towards the development of a secure JPEG encoder, Proc. of the IEEE Workshop on Signal Processing Systems, 2003, 183-188.

[23] Strycker, L. D., Termont, P., Vandewege, J., Haitsma, J., Kalker, A., Maes, M., & Depovere, G. , Implementation of a Real-Time Digital Watermarking Process for Broadcast Monitoring on Trimedia VLIW Processor, IEE Proceedings on Vision, Image and Signal Processing, **147**, 2000, 371-376.

[24] Cheung, S. C., Chiu, D. K. W., A watermarking infrastructure for enterpries document management, 36th Annual Hawaii Int. Conf. on System Sciences, 2003, 105-114.

[25] Fan, Y. C., Van, L. D., Huang, C. M., & Tsao, H. W., Hardware-efficient architecture design of wavelet-based adaptive visible watermarking, Proceedings of 9th IEEE International Symposium on Consumer Electronics, 2005, 399-403.

[26] Mathai, N. J., Kundur, D.,& Sheikholeslami, A., Hardware implementation perspectives of digital video watermarking algorithms, IEEE Transaction on Signal Processing, **51**, 2003, 925- 938.

[27] Tsai, T. H., & Lu, C. Y., A Systems Level Design for Embedded Watermark Technique using DSC Systems, Proceedings of the IEEE International Workshop on Intelligent Signal Processing and Communication Systems, 2001.

[28] Mohanty, S. P., Kougianos, E., & Ranganathan, N., VLSI architecture and chip for combined invisible robust and fragile watermarking, IET Computers and Digital Techniques, **1**, 2007, 600-611.

[29] Garimella, A., Satyanarayan, M. V. V., Kumar, R. S., Murugesh, P. S., & Niranjan, U. C., VLSI implementation of online digital watermarking techniques with difference encoding for the 8-bit gray scale images, Proceedings of the Int. Conf. on VLSI Design, 2003, 283-288.

[30] Mohanty, S. P., Ranganathan, N.,& Namballa, R. K., A VLSI architecture for visible watermarking in a secure still digital camera (S2DC) design, IEEE Trans. on Very Large Scale Integration Systems (TVLSI), **13**, 2005, 1002-1012.

[31] Mohanty, S. P., Ranganathan, N., & Balakrishnan, K., A dual voltage-frequency VLSI chip for image watermarking in DCT domain, IEEE Trans. on Circuits and Systems II (TCAS-II), **53**, 2006, 394-398.

[32] Bhattacharya, B. B., Kundu, M. K., Maity, S. P., Murthy, C. A., & T. Acharya, Robust digital image watermarking utilizing a Walsh transform algorithm, US Patent application field on 11th December 2003, application no. 20050144456.

[33] Prasad, R., CDMA for Wireless Personal Communications, Artech House (Boston, 1996).

[34] R. C. Gonzalez and R. E. Woods, Digital Image Processing, Addison-Wesley (New York, 1992).

[35] Shannon, C. E., A mathematical theory of communication, Bell Syst. Tech. Journal, **27**, 1948, 379-423.

[36] Pal, N. R., & Pal, S. K., Object-back ground segmentation using new definitions of entropy, IEE Proceedings, **136**, 1989, 284-295.

[37] Langelaar, G. C., Setyawan, I., & Lagendijk, R. L., Watermarking digital image and video data: A state-of-the-art overview,IEEE Signal Processing Magazine, **17**, 2000, 20-46.

[38] Depovere, G., Kalker, T.,& Linnartz, J. P., Improved watermark detection reliability using filtering before correlation, Proc. of IEEE Int. Conference on Image Processing (ICIP), **1**, 1998,430-434.

[39] Peterson, R. L., Ziemer, R. E., & Borth, D. E., Introduction to Spread Spectrum Communications, Prentice Hall International Editions, 1995.

[40] Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P., Image quality assessment: From error measurement to structural similarity, IEEE Trans. on Image Proc., **13**, 2004, 1-14.

[41] Hsu, C. T., & Wu, J. L., Hidden digital watermarks in images, IEEE Transaction on Image Processing, **8**, 1999, 58-68.

[42] Maity, S. P., Hati, S., & Maity, S., Efficient Channel Estimation of Diversity Assisted Synchronous CI/MC-CDMA Uplink System Using Genetic Algorithm, 10th Int. Symp. on Wireless Personal Multimedia Communications, Jaipur, India, December 3-6, 2007, 870-874.