VLSI Design of Spread Spectrum Image Watermarking

Santi P. Maity, Ayan Banerjee, Abu Abhijit

Bengal Engg. and Sc. University, Shibpur,
Howrah, India, 711 103
spmaity@yahoo.com, ayan@telecom.becs.ac.in

Malay K. Kundu

Indian Statistical Institute, Kolkata,
203 B. T. Road, Kolkata-700108
malay@isical.ac.in

Abstract

Spread spectrum watermarking for multimedia signal becomes appealing due to its high robustness attribute and is used widely for various applications. Some of these applications demand development of low cost algorithms so that they can be used for real time services such as broadcast monitoring, security in communication. In recent time one similar 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. However, the major shortcomings of the existing SS watermarking schemes are high computation cost and complexity of the algorithm that limits their use for stated purposes. The paper proposes SS image watermarking algorithm using Fast Walsh transform that offers low cost and ease of hardware realization. VLSI implementation using Field Programmable Gate Array (FPGA) has been developed for the algorithms and circuit can be integrated into the existing digital still camera framework.

1. Introduction

Spread spectrum (SS) modulation principle can be used for watermarking in digital media as it is widely accepted to the watermarking research community due to its robustness attribute. SS is accomplished by spreading a narrow band watermark into wide spectrum of the cover so that watermark energy for each frequency bin become less and could hardly be detectable. Several SS watermarking schemes for multimedia signals are developed using DCT [1], Fourier-Mellin [2], wavelet [3] transforms. However, the techniques are not suitable for broadcast monitoring due to the high computation cost, complexity that makes their hardware realization complicated. Hardware implementation of digital watermarking techniques offer advantages of real time processing of data [4][5]. If a chip is £tted 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 will highlight the signi£cance where digital media is to be marked in real time and hardware is the only solution. Campisi et al [6] investigated the scope of the usage of digital watermarking for QoS assessment of multimedia signal.

In this paper, we propose a Walsh transformation based SS watermarking scheme in digital image, for application to blind assessment of QoS, which may also be applicable in video frames and other multimedia signals. The complexity and computation cost of the algorithm is low and hardware implementation is easy. Hardware implementation using Field Programmable Gate Array (FPGA) has advantages of real time application, low investment cost and desktop testing with moderate processing speed [7].

The paper is organized as follows: Section 2 describes brie¤y the proposed algorithm. Sections 3 and 4 describe VLSI design of the embedding unit and the decoding unit respectively using FPGA. Sections 5 and 6 present results and conclusion respectively.

2. Proposed algorithm

The cover image of size $(M_c \times N_c)$ is partitioned into (8×8) non-overlapping blocks. Each image block is then decomposed using Fast Walsh transform [8]. The size of the image block is considered (8×8) in order to make the scheme compatible with JPEG compression operation. The widely used code pattern for SS modulation technique is pseudo noise (PN) sequence and is generated using LFSR (Linear feedback shift register)[9]. The size of the PN sequence is identical to the size of the Walsh coef£cient matrix. Thus a set of PN matrices denoted by (P_i) of number $(M_m.N_m)$ are generated. 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 Walsh coef£cient of the cover image, X^e is the Walsh coef£cient after watermark embedding, k is the modulation index, P is the PN matrix. Two dimensional block based discrete inverse Walsh transform

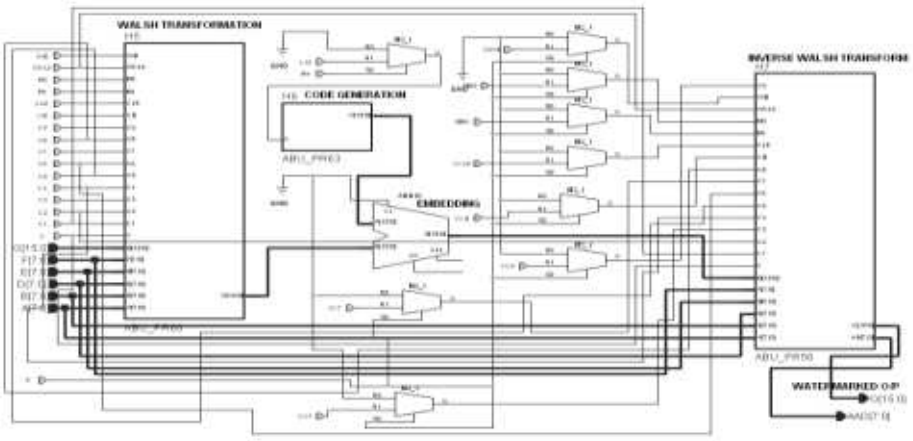

Figure 1: VLSI architecture of transmitter

of the modi£ed coef£cients would then generate watermarked image.

The watermark recovery process requires the sets of PN matrices (P_i) that were used for data embedding. The received watermarked image is partitioned into (8×8) non overlapping blocks and is decomposed using Walsh transform. Correlation values between Walsh coef£cients 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 (μ_i) where i= $1, 2, ..M_m.N_m$. From these correlation values, we calculate mean correlation value (T), 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*.

3. VLSI design

The VLSI architecture of the proposed algorithm is designed using XILINX SPARTAN series FPGA. There are two main sub blocks, one is the transmitter and the other one is the receiver. The over all function of the transmitter unit is to decompose the image signal using Walsh transform and then embedding the watermark while the receiver unit decodes the embedded watermark.

3.1. Transmitter architecture

The VLSI architecture of the transmitter for the proposed algorithm is shown in Fig. 1. Hardware design consists of four sub blocks or module namely (1) Walsh Transform module, (2) Code generation module, (3) Data embedding module and (4) Inverse Walsh transformation module.

Data 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 coef£cients 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 ¤ow into the inverse transform block at the desired time. The detailed architecture of each subblock is described below.

(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:

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
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

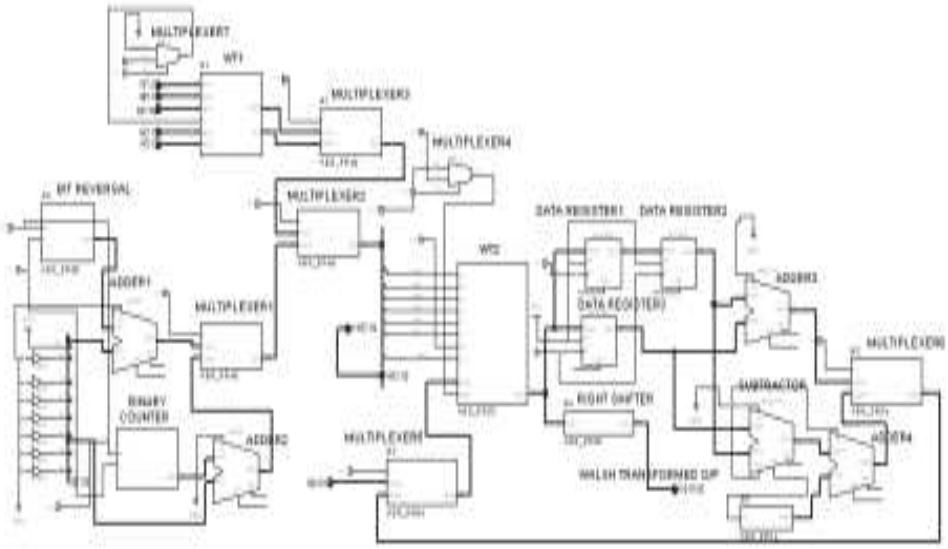

Figure 2: VLSI architecture of Walsh transformation

```
                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. 2 shows the detailed hardware architecture of Walsh transform for an image block of size (8). It is to be noted that digital image considered here is a gray scale image of 8 bits/pixel. 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 implement 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 £rst 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 not shown due to space limitation. 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 speci£ed by the statements 24 through 26 are performed as follows: RAM is read from two locations speci£ed 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 speci£ed 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 controlled by WE (write enable) input. Finally the output of the binary counter passes through the MUX1 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 coef£cients are obtained at the output pin of the right shifter. The required components for Walsh transform module are two 1-bit MUX (2:1), £ve 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.

(3) *Data embedding module*: The output from code generation unit is added with the output from Walsh transform unit to obtain coef£cient 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 extra right shifter block that performs the division operation.

3.2. Receiver architecture

The VLSI architecture of receiver design is shown in Fig. 3.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 coef£cients 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 transmitter section.

(2) *Correlation calculation module*: The detailed hardware architecture of the correlation calculation block is shown in Fig. 4.

The same code generation units PN1 and PN2 used at transmitter are also used here. Input A[15:0] is set to zero. The Walsh coef£cients are applied to the input pin B[15:0]with the clock C. The 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],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. 5. 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 comparators. 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.

Table 1: Speci£cation of hardware realization

Chip	CLB count	Clock Freq.	Clock cycle
XCS40	730	80 MHz 1	344 cycles/ (8×8)

4. Results and applications

The VLSI design is implemented for a gray scale image of size (8×8) and a 4 bit binary watermark with element value **1** and **0**. The choice of (8×8) block size is to make the scheme compatible with DCT based JPEG compression operation.

The hardware design can be easily extended for large image size, say (256×256) or (512×512) 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. The hardware design is implemented using XILINX SPARTAN series FPGA. The chip used is XCS05 which contains 784 CLB, out of which 730 CLBs are consumed, 430 for transmitter unit and 300 for the receiver. Speci£cation of hardware realization is shown in Table 1.

The work reported in this paper can £nd applications in authentication and broadcast monitoring for real time video transmission through wireless channel. As mentioned earlier, the extracted watermark quality can represent the status of the channel condition and necessary feedback information may be sent to the transmitter to control data transmission rate adaptively in order to suit the characteristics of the transmission channel. Thus end-to-end quality assurance is possible for multimedia signal transmission in mobile radio channel. FWT has been chosen as signal decomposition tool as it offers better robustness of the embedded data at low quality compression considering both JPEG and JPEG 2000 compression [10]. Reference watermark pattern is embedded in the multimedia host data and transmitted through the channel. Host data and watermark suffer the channel degradation, where alteration in watermark is used to assess the quality of offered services. The algorithm is tested against JPEG and JPEG 2000 coder followed by additive white Gaussian noise and performance is superior to [6] (Results not shown due to space limitation).

5. Conclusions

An algorithm for low cost SS image watermarking using Walsh transformation is proposed in the paper and its VLSI realization using FPGA is also reported. Algorithm requires few simple computation and VLSI implementation using FPGA allows it application for real time

multimedia data transmission. Current work is going on to develop the dedicated digital system using this FPGA chip.

References

[1] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. on Image Proc.*, vol. 6, pp. 1673-1687, 1997.

[2] J. O. Ruanaidh and T. Pun, "Rotation, scale and translation invariant spread spectrum digital image watermarking,"*Signal Proc.*, vol. 66, pp. 303-317, 1998.

[3] S. P. Maity and M. K. Kundu, "A blind CDMA image watermarking scheme in wavelet domain," *Proc. of Int. Conf. on Image Proc., Singapore*, pp. 2633-2636, 2004.

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

[5] S. P. Maity, P. K. Nandi, M. K. Kundu, and A. Banerjee, "Low cost data authentication scheme and hardware realization," *11th National Conf. on Comm.*, pp. 574-577, 2005.

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

[7] W. K. Chen, *The VLSI Handbook*, CRC Press, August 2000.

[8]R. C. Gonzalez, and R. E. Woods, "Digital Image Processing," Addison-Wesley, New York, 1992.

[9] B. Sklar, *Digital Communication*, Prentice Hall, Englewood Cliffs, NJ, 1988.

[10] M. Ramkumar, A. N. Akansu, "Capacity estimates for data hiding in compressed images," *IEEE Transactions on Image Processing*, vol. 10, pp. 1252-1263, 2001.

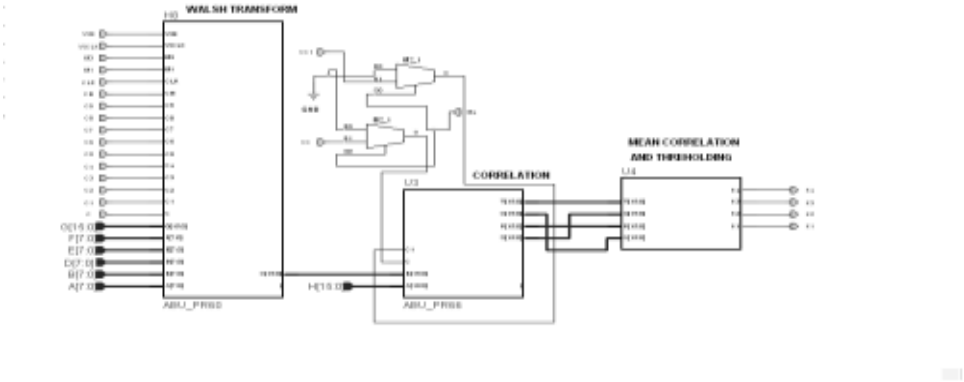

Figure 3: VLSI architecture of receiver

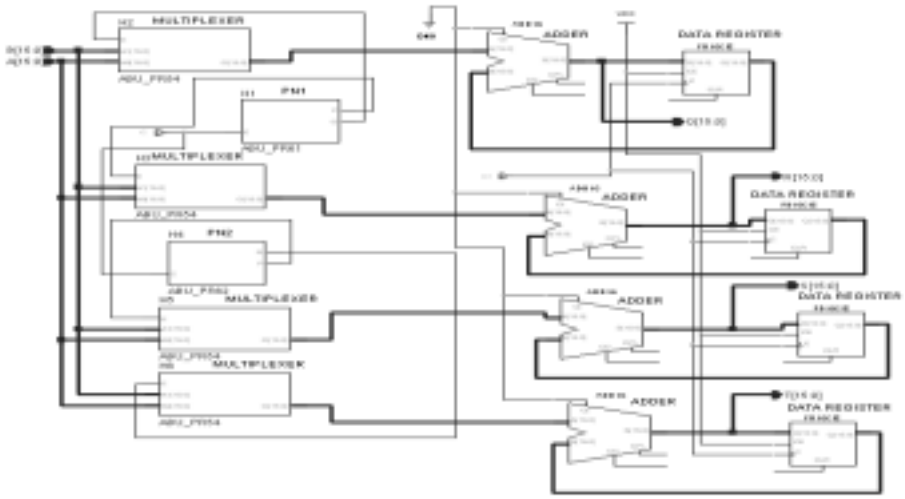

Figure 4: VLSI architecture of correlation calculation

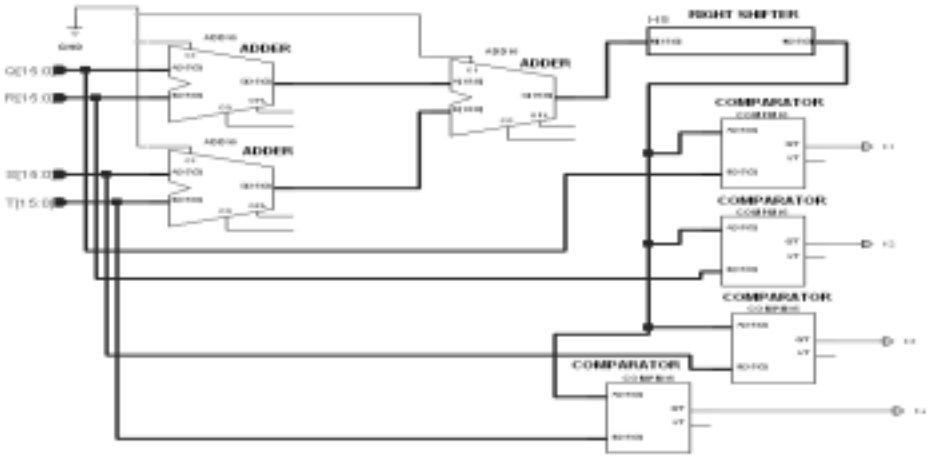

Figure 5: VLSI architecture of mean correlation and threshold calculation