Distortion free image-in-image communication with implementation in FPGA

Santi P. Maity[1] , Malay K. Kundu[2]

[1]*Dept. of Information Technology, Bengal Engineering and Science University, Shibpur P.O. Botanic Garden, Howrah -711 103, India*
[2]*Machine Intelligence Unit, Indian Statistical Institute, 203, B. T. Road, Kolkata 700 108, India*

santipmaity@it.becs.ac.in, malay@isical.ac.in

Abstract

The proliferation of the digitized media (audio, image and video) introduces a challenging problem for data transmission in the network environment. In this paper, a novel, simple and low cost algorithm that serves the purpose of distortion free covert image-in-image communication is proposed. Its very large scale integration (VLSI) implementation using field programmable gate array (FPGA) is also developed. A binary equivalent message signal is developed first from the combination of the auxiliary gray scale image information and the carrier gray scale image (original) using channel coding and spatial bi-phase modulation scheme. The auxiliary image information is then decoded from the distorted/distortion free version of the original image using binary message under certain noise constraint. Implementation of the proposed low cost algorithm can be speeded up significantly by hardware realization. The developed hardware design allows data transmission at the rate of 4.706 Mbits/Sec at 80 MHz clock frequency.

Keywords: Channel coding, spatial bi-phase modulation, image-in-image communication, FPGA, VLSI design.

1 Introduction

Progress in digital technique leads to an ease and efficient distribution and transmission of different media such as text, image, video, and audio etc. to the distant places using communication network and world wide web. At the same time, a class of problems are also emerged to maintain authenticity, integrity and security in digital data transmission [1]. Data encryption and data hiding are the two different techniques that are now being used widely to

[1] Author for correspondence, Santi P. Maity, Dept. of Information Technology, Bengal Engineering and Science University, Shibpur, P.O. Botanic Garden, Howrah - 711 103, India, Tel.: (+91) (+33)2668-4561/62/63 Extn;846 fax:(+91)(33)2668-2916.

serve this security purpose. Data encryption or cryptographic technique offers security by preventing an eavesdropper from accessing the original media. On the other hand, data hiding (watermarking) scheme provides security through an imperceptible embedding of auxiliary message into the digital multimedia signal without preventing the access of the latter to the common users [2]. However, some typical data hiding application, for example, access control is intended to enjoy full quality of the image or video data to the authorized users only [3].

In all practical data hiding methods, the original data, say an image is inevitably distorted by some small amount of embedding distortion that cannot be removed completely due to quantization, bit-replacement or truncation at the gray scales 0 and 255 [4][5]. Although the distortion is often quite small, it may not be acceptable for medical imagery (for legal reasons) or for military images inspected under unusual visual conditions (after extreme zoom)[6][7]. Moreover, embedding of auxiliary information in original data changes the statistical distribution of the latter that can easily be detected in sophisticated steganalysis test and is not permitted to reach the desired user [8]. It would then be effective to develop a scheme that differs from both data hiding and encryption principles in the sense that it neither embeds any data in the original media nor it prevents the access like conventional encryption technique compels to unauthorized users.

A novel data (image)-in-data (image) communication scheme may be thought as a viable alternative for other way secured data transmission [9]. To accomplish this image-in-image transmission, the concept of widely used carrier modulated data transmission may be used. The original (host) image plays the role of carrier signal and is expected to be available at both the transmitter and receiver end (as in Internet service the image is expected to be available anywhere). Message signal and the host (original data) would generate a modulated signal which would be transmitted (i) through a secured channel (using cryptography or data encryption) or (ii) through an insecure channel subject to some distortion constraint (due to the contribution of transmission channel or attack channel). The goal is to keep the original data absolutely unchanged, hence, neither prevents its full access to any user nor distorts the signal due to embedding as occurs in data hiding/watermarking methods. At the receiver, the message signal is demodulated from the original data (carrier) and the modulated signal. An important point is mentioned here that the term security does not imply the degree of computational complexity that an unauthorized user is to face to break or decode the encrypted message, as the term is used in case of conventional encryption technique. The term security implies here the ability to resist hostile attacks or common signal processing operations applied on the original data and also the degree of noise perturbation withstand by the modulated signal in the process of faithful decoding of message.

The various data hiding and encryption algorithms are implemented with either software or hardware. Hardware implementation offers advantages over software realization in terms of less area, low execution time, low power, real-time performance, high reliability and also ease of integration with existing consumer electronic devices [10]. The software designer does not have direct control over the way random access memory (RAM) and processors interact, posing a limit on speed. A software designer must try to limit the total amount of RAM required, while a hardware designer has full control over timing operations into the RAM and direct control over the usage of expensive hardware resources [11]. In any data hiding or data-in-data communication problem, if a chip is fitted in the digital devices, the stego data/modulated signal can be obtained from the output video or images right at the origin.

A hardware based implementation can be designed on a field programmable gate array (FPGA) board [12], trimedia processor board [13], or custom integrated circuit (IC) [14]. The choice between FPGA and cell based IC is a trade-off between cost, power consumption and performance. Hardware implementation using FPGA offers advantages of low investment cost, simpler design cycle, field programmability/re-configurability and desktop testing with moderate processing speed [15]. On the other hand, due to lower unit cost, full custom capability and from an integration point of view custom based application specific integrated circuit (ASIC) design may be more useful. The FPGA design flow eliminates the complex and time-consuming floor planning, place and route, timing analysis, and mask/respin stages of the project since the design logic is already synthesized to be placed onto an already verified and characterized FPGA device. During recent past, FPGAs were used to be selected for lower speed/complexity/volume designs, but today's FPGAs easily push the 500 MHz performance barrier. With unprecedented logic density increases and a host of the features, such as embedded processors, digital signal processing (DSP) blocks, clocking, and high-speed serial at ever lower price points, FPGAs are a compelling proposition for almost any type of design [15].

It would be mentioned here again that the present work is different to that of digital watermarking method. However, since digital watermarking scheme is one form of secured data transmission without restricting the access of original data to common users and recent literature discuss many hardware based realizations, we make a brief review of such implementations as related works. This would highlight the limitation of the existing works and the scope of the present work. As a spatial domain approach, Strycker et al. [16] develop a watermark embedder and detector on a Trimedia TM-1000 VLIW processor. The authors in [17] propose a watermark-based protocol for the document management in large enterprises. Fan et al. [18] propose a visible watermarking design based on an adaptive discrete wavelet transform. Two-path parallel processing architecture is exploited to reduce processing time and the signal is sent to different processing elements by the demultiplexers. The authors in [11] propose the video watermarking algorithms through the hardware implementations of a well-known algorithm called just another watermarking scheme (JAWS) with 0.18 μm CMOS technology. Tsai and Lu [19] propose a discrete cosine transform (DCT) domain invisible watermarking chip with TSMC 0.35 μm technology and has a die size of (3.064×3.064) mm^2. Garimella et al.[20] propose a VLSI architecture for invisible fragile watermarking in the spatial domain. The ASIC is implemented using 0.13 μm technology. The area of the chip is (3453×3453) μm^2, and the chip consumes 37.6 μW of power when operated at 1.2V. The critical path delay of the circuit is 5.89 ns.

Mohanty et al. [21] propose 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. The chip is implemented with 0.35μm technology and occupies an area of $(3.34 \times 2.89) mm^2$ and consumes power of 6.9286mW when operated at 3.3V and 292.27MHz. Mohanty et al. [22] also propose another VLSI architecture that can insert invisible or visible watermarks in images in the DCT domain. A prototype VLSI chip is designed and is verified using various Cadence and Synopsis tools based on TSMC 0.25 μm technology with 1.4M transistors and 0.3 mW of average dynamic power. Mohanty et al. [23] develop low-power, high-performance, real-time, reliable and secure watermarking systems, which can be achieved through hardware implementations. They prototyped the watermarking chips in two ways: (i) by using a Xilinx FPGA and (ii) by building a custom integrated circuit. Maity et al [24] propose fast Walsh transform (FWT) based spread spectrum (SS) image watermarking scheme that serves the dual purposes of authentication in data transmission as well as quality of services (QoS) assessment for digital media through dynamic estimation of the wireless channel condition. Maity et al [25] also propose discrete Hadamard transform (DHT) domain watermarking scheme with loss in image information which can be easily mapped to hardware.

1.1 Scope of the work

On summarization, it appears that data (image)-in-data (image) communication algorithm in real time scenario demands the need of an algorithm with low computation cost and complexity for message encoding and decoding. Moreover, implementation of algorithm in hardware is desirable and at the same time faithful recovery of the message information under certain distortion and noise constraints is also essential. For image data, spatial domain approach is preferred to develop real time implementation due to its advantage of low computation cost. This real time realization may be done using FPGA. Spatial domain image-in-image communication algorithm must also be capable of recovering the message signal in the face of various signal manipulations including noise corruption and high degree of lossy compression for the carrier and the modulated signal. To build up a simple yet effective architecture, we attempt to exploit strong spatial correlation that exists among the neighboring pixel values of any natural image. This correlation is reflected by the successive binary run of larger lengths formed by the most significant bit (MSB) plane of the gray values. MSB plan offers relative stability i.e. each run length does not change frequently in the event of various image processing operations, as long as its visual quality remains maintained.

In brief, this paper proposes a novel, simple and cost effective distortion free covert image-in-image communication algorithm and its VLSI implementation using FPGA. In this message transmission scheme, the run length of MSB plane is used for message encoding. Gray scale image is used as message signal so that the decoded message preserves its recognizability when transmitted through the noisy channel. Improvement in data transmission reliability is further increased using channel coding scheme in the form of variable redundancy that is incorporated among the different bit planes based on their relative significance. Spatial bi-phase modulation scheme reduces transmission overhead i.e. the amount of data transmission that is increased due to channel coding scheme.

The paper is organized as follows: Section 2 describes proposed algorithm, while Section 3 presents VLSI implementation using FPGA. Section 4 presents performance evaluation and results of hardware design. Conclusions are drawn in Section 5 along with the scope of future work.

2 Proposed Algorithm

The proposed algorithm is analogous to that of a digital modulation scheme with synchronous detection for decoding of message. The gray scale image-like message signal is considered as information bearing (modulating) signal. Such type of signal not only conveys unique information but also shows a good degree of recognizability after various forms of image distortions. The original image is considered as carrier and a binary equivalent image, called here as modulated signal, is generated using channel coding and spatial bi-phase modulation scheme. The modulated signal is transmitted either through a secured channel or through a noisy channel subject to a distortion constraint. Message is extracted at the receiver from this modulated signal using the original image (carrier) or its noisy version (drift in carrier). Accordingly, the proposed algorithm consists of two main modules, namely (a) message encoding done at transmitter and (b) message decoding done at receiver. Fig. 1(a) shows block diagram representation of message encoding, while Fig. 1(b) shows possible channel model and message decoding.

2.1 Message encoding at transmitter

The inputs to the message encoding process are the original image and the message signal. On the other hand, the output is the binary modulated signal. A common data modulation process involves two signals, namely carrier and message and one liner/non-linear operation called modulation. To accomplish generation of both signals as well as data modulation, the encoding process consists of three different steps.

Step 1: Formation of data sequence 1:Carrier signal generation

The 2-D pixel values of the original image is converted to an 1-D signal. To implement covert message encoding, a string of binary data is formed from the MSB plane of the pixel values and a binary data sequence, called sequence ‘1‘ is thus formed. This process generates a binary string-like carrier signal.

Step 2: Formation of data sequence 2:channel coded message data

The gray scale message signal is first mapped to a 1-D signal and is converted into a binary sequence. An extended binary data sequence (sequence 2) is formed by incorporating variable redundancy (repeating each bit by suitable odd number of times) onto the different bit planes of the message. Higher redundancy is assigned to the higher bit plane since they contain visually significant data and less or no redundancy for lower order bit planes that contribute more subtle details in the image [26]. Incorporation of variable redundancies among the various bit planes make a balance between bandwidth preservation and detection reliability like the way error control codes do in data transmission over noisy channel.

Step 3: Formation of binary modulated data:spatial biphase modulation

Data sequence 1 (carrier signal) and sequence 2 (channel coded data) are partitioned into subsequences having equal and fixed number of digits or symbols. If the symbols in the two respective subsequences match in more than 50% position, a bit ‘1’ is assigned for the subsequence, otherwise a bit ‘0’ is assigned. Bit ‘1’ indicates in-phase condition of two subsequences, while out of phase condition is represented by bit ‘0’. Assigning a binary digit, based on the number of match in symbols between two subsequences, is called here as spatial bi-phase modulation technique. The process converts a gray scale message into a binary equivalent modulated signal.

To quantify the reliability of data compaction using the combined effect of channel coding and spatial bi-phase modulation, a new sequence (sequence 3) is formed. This is done by keeping each subsequence of sequence 2 unchanged or is complemented bitwise based on the bit value ‘1’ or ‘0’ of the newly obtained binary sequence. The number of positional mismatch in the symbols between the sequence 1 and sequence 3 are counted. This is divided by the total number of symbols in the sequence in order to calculate the probability of error denoted by p(e). If a subsequence consists of 'r' number (an odd number) of symbols and the sequence consists of total 'k' number of such subsequences, P(e) that denotes the probability of making wrong decision for all the subsequences can be expressed as follows:

$$P(e) = (\sum_{n}^{r} \binom{r}{n} {p_e}^{n} (1 - p_e)^{r-n})^{k} \tag{1}$$

where $n = (r+1)/2$. All 'k' number subsequences are assumed to be independent among each other. Lower the value of $P(e)$, lower is the message encoding loss. The value of $P(e)$ is related with the length of the subsequence 'r' which is again related with the size of the binary message.

2.2 Message decoding at receiver

The message decoding process is just reverse to that of the encoding process. The inputs are the binary modulated signal or its noisy version and the original signal or its possibly distorted version. The output of the decoding process is the gray scale image like message signal. The various steps for decoding process are described below.

Step 1: Conversion from 2-D to 1-D

The 2-D pixel values of the original image/its noisy version is converted to 1-D subsequence.

Step 2: Formation of sequence using MSB plane

The MSB plane of this 1-D signal is picked up and is partitioned into subsequence of predefined fixed and equal number of symbols that was used during message encoding.

Step 3: Spatial Bi-phase demodulation

Each subsequence either remains unchanged or complemented based on the value of the received bit '1' or '0' in the binary modulated signal corresponding to the particular subsequence.

Step 4: Subsequence decoding

Each subsequence obtained after such operation is partitioned into sub subsequence (smaller subsequence) based on the degree of redundancies incorpo-

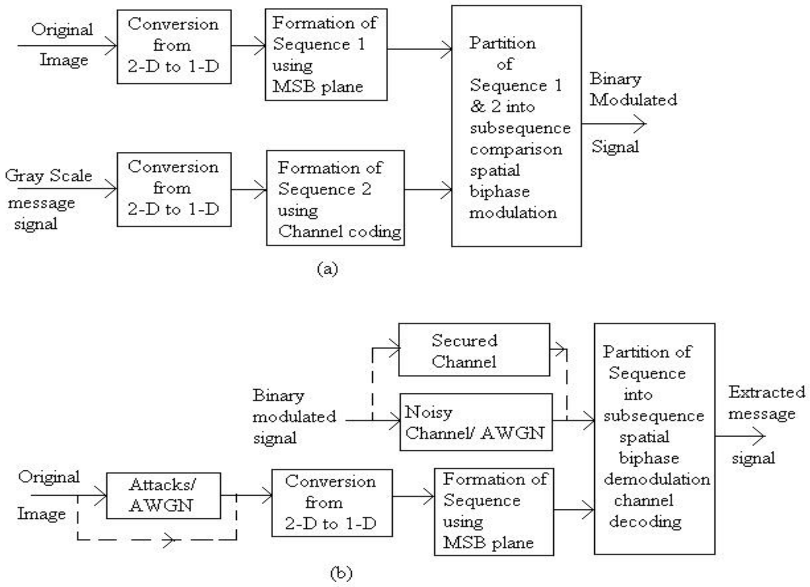
Original Image
Conversion from 2-D to 1-D
Formation of Sequence 1 using MSB plane
Partition of Sequence 1 & 2 into subsequence comparison spatial biphase modulation
Binary Modulated Signal
Gray Scale message signal
Conversion from 2-D to 1-D
Formation of Sequence 2 using Channel coding
(a)
Secured Channel
Binary modulated signal
Noisy Channel/ AWGN
Partition of Sequence into subsequence spatial biphase demodulation channel decoding
Extracted message signal
Original Image
Attacks/ AWGN
Conversion from 2-D to 1-D
Formation of Sequence using MSB plane
(b)

Fig. 1. Block diagram of (a) message encoding and (b) channel model and message decoding

rated on the different message bits. Binary detection is then applied for each sub subsequence based on the majority decision rule i.e. if more than 50% symbols of a sub subsequence are 1, decision for decoding is ‘1’, otherwise ‘0’.

Step 5: Formation of message signal

The binary digits of all the sub subsequences of a subsequence are then converted to the pixel values to decode the message. The similar decoding process is applied for other subsequences and the message decoding is thus completed.

3 VLSI Architecture

The VLSI architecture of the proposed algorithm is designed using XILINX SPARTAN series FPGA. There are two main modules, one is the architecture for message encoding unit and the other one is the same for the message decoding unit. We propose an architecture for the original image of size (256x256), 8-bits/pixels and the message signal is a 4 bits/pixel gray scale image of size (64 x 64). The variable redundancies incorporated in different bit planes of the message image are 9 bits for 4th bit plane (MSB), 5 bits for 3 rd bit plane and no redundancy for the 2nd and 1st bit plane (LSB planes).

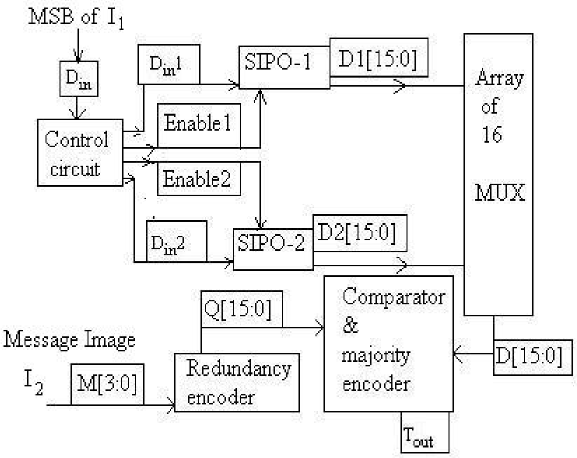

Fig. 2. VLSI architecture of message encoding unit

3.1 Architecture of message encoding

The VLSI architecture of the message encoding unit is shown in Fig. 2. The input data D_{in} is the sequence 1 as sated in Step 1 of Section 2.1. This is already mentioned that this sequence is formed by picking up the MSB of the pixel value of gray scale image I_1 and is carrier-like signal. The control section sends out 4 different signals D_{in}^1, Enable 1, D_{in}^2 and Enable 2. This section along with the array of multiplexers decide which one of the shift registers would send data to the comparator. The redundancy encoder is used to convert each pixel value of 4 bits for the message image I_2 to a 16 bit data. The comparator and the majority encoder compares each substring of 16 bits length of string 1 i.e. D[15:0] with that of string 2 i.e. Q[15:0]. If they match in more than 8 positions, T_{out} becomes '1', otherwise '0'. This T_{out} is the binary modulated signal to be transmitted through secured channel or noisy channel subject to a certain noise constraint.

The major subblocks of message encoding unit are control circuit, serial-in-parallel-out (**SIPO**) shift registers, multiplexer and majority encoder. In order to speed up the process, two **SIPO** operates in parallel. When one **SIPO** takes external input data, the other **SIPO** feeds data to the majority encoder unit and vice versa. Fig. 3(a) shows the circuit of **SIPO**. Two data sets, each of sixteen bit length from the two **SIPO**, are fed to the multiplexer that outputs one data set to the majority encoder block. Circuit realization of majority encoder block is shown in Fig. 3(b). In the majority encoder block, the other input is coming from the extended message obtained after adding redundancy. Fig. 4 shows the detailed circuit of redundancy encoder. Two data sets of 16 bit are fed into an array of 16 XNORs for similarity comparison. The output from the similarity comparator block is then passed through parallel-in-serial-out (**PISO**) shift register and is fed into a 4 bit binary up counter to enable its

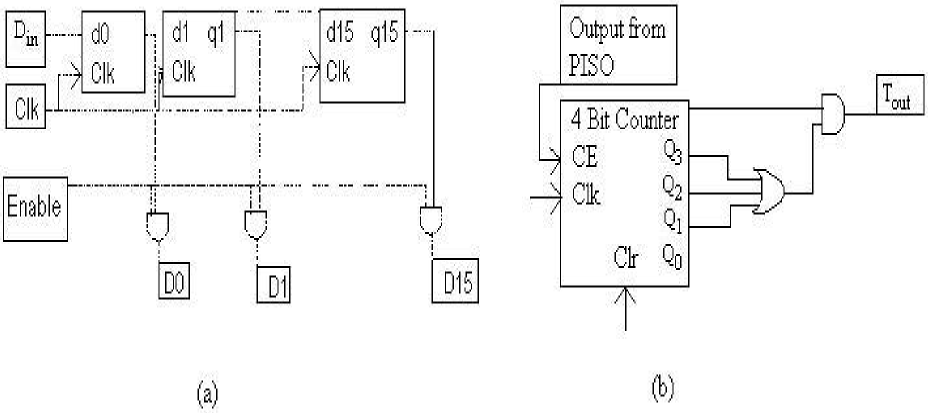

Fig. 3. (a) Serial-in-parallel-out shift register; (b) Majority Encoder block

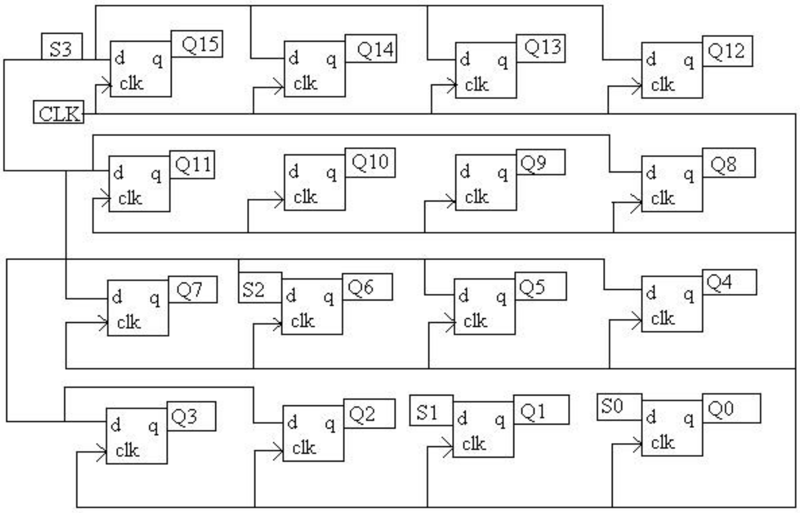

Fig. 4. Redundancy encoder

clock. Fig. 5(a) shows the parallel in serial out (**PISO**) shift registrar circuit. The output of the counter is fed to an encoder. For a string of 16 bits, if the similarity comparator output is '1' in 9 bit positions or more, the counter value would be updated by the same amount. The counter is reset after every 16 bit sequence. The counter value is fed to an encoder block. The encoder is designed in such a way that if its input value is nine or more, it would generate **'1'** at output, otherwise **'0'**. This encoder output is final modulated output from the transmitter.

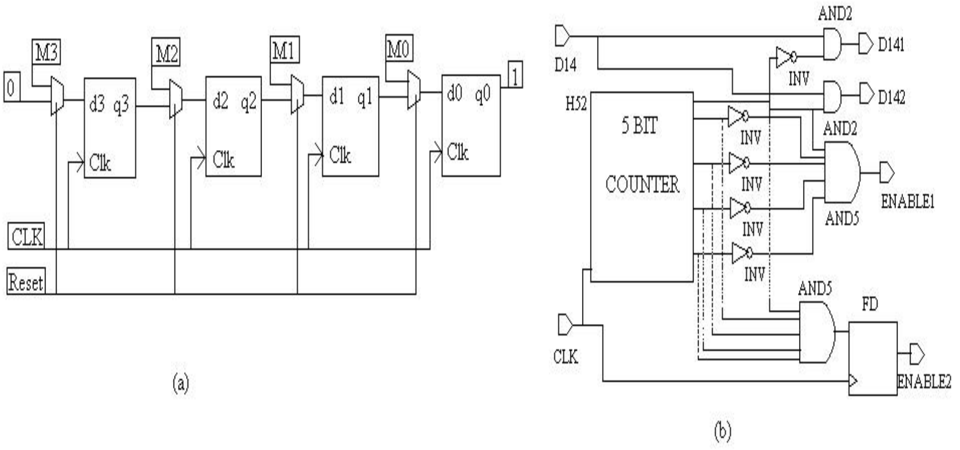

Fig. 5. (a) Parallel-in-serial-out shift register; (b) Control circuit

Fig. 3(b) shows the detailed combinational logic for the majority encoder block. Here the output from the **PISO** (Parallel-in-serial-out) shift register is used as the clock enable (CE) signal for the 4 bit counter. Thus whenever the output from the **PISO** shift register is **'1'**, the output of the counter is incremented by **'1'** value. In order to distinguish between two subsequences of length **16** bits, a **Clr** terminal is used for the counter to reset it after **16** clock cycles. The terminal count of another **4** bit counter after being passed through a D flip-flop is fed to the **Clr** terminal. Thus the message encoder output T_{out} is obtained at an interval of **16** clock cycles with an initial delay of **32** clock cycles.

Fig. 5(b) shows the detailed design of the control circuit. The circuit is basically used to control the flow of data to the two serial-in-parallel-out (SIPO) shift registers. A 5-bit counter is used together with some combinational circuit to generate the necessary control signals. The data channel D_{in} (shown in Fig. 3(a)) is used as the input to the **SIPO-1** for the first 16 clock cycles of the counter. At the 16-th clock cycle, the data channel D_{in} is transferred to the input of the **SIPO-2** and the enable signal intended for **SIPO-1** is made high for one (1) clock cycle only. At the 31-st clock cycle, the enable2 is obtained by passing the terminal count of the counter through a D flip-flop. Thus, as the counter again starts counting from zero, the enable 2 line remains high for 1 clock cycle only and the data channel is now transferred to SIPO-2. One set of data (of size 16 bits) is read from the data file through one SIPO shift register. The previous set of data can be obtained from another SIPO shift register and the combination leads to faster operation.

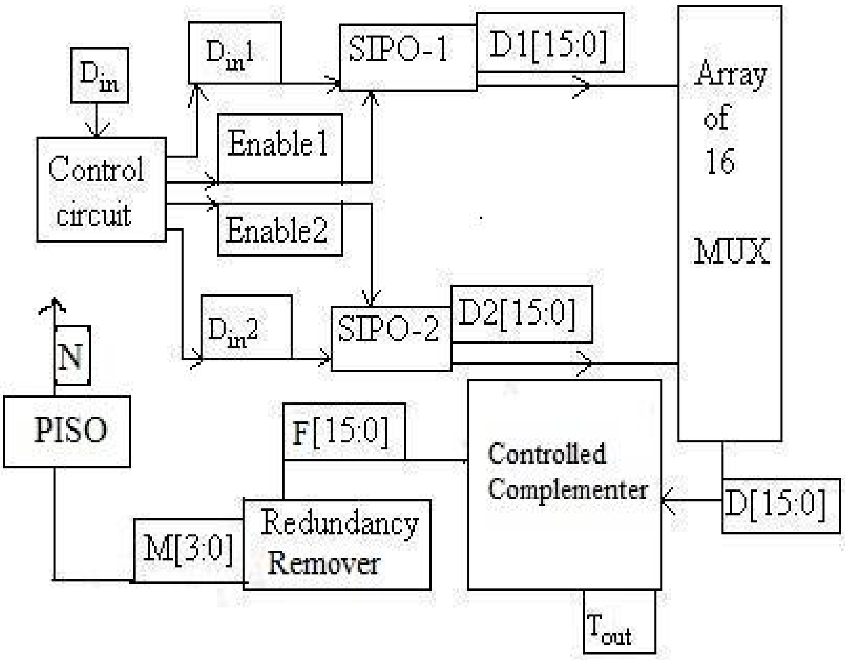

Fig. 6. VLSI architecture of receiver

3.2 Architecture of message decoding unit

The VLSI architecture of the message decoding for the proposed algorithm is shown in Fig. 6. In the decoding unit, the main sub-blocks are (i) control unit, (ii) SIPO-1 and SIPO-2, (iii) controlled complementer, and (iv) redundancy remover. Control unit, SIPO-1 and SIPO-2 work in similar fashion as described in the message encoding unit. The controlled complementer block consists of sixteen 2:l multiplexers and sixteen inverters to generate sixteen image bits. Design of the controlled complementer circuit is shown in Fig. 7. The function of the multiplexer is to select any one out of two available inputs (let **A** and **B**) at a time and send this to the output. The 2-input AND gates (let G1 and G2) and one 2- input OR gate (G3) perform the function. The input **A** is connected to the input terminal of G1 and **B** is connected to the input terminal of G2. One select signal S is connected to the second input terminal of G2 directly and in inverted manner to the second input terminal of **G1**. The output terminals of G1 and G2 are connected to the two input terminals of G3. When S=0, G1 yields **A** and G2 yields 0 due to basic AND operation. The terminal G3 would produce **A** at the circuit output due to basic OR operation. Following the similar logic, the circuit output is **B** when S=l.

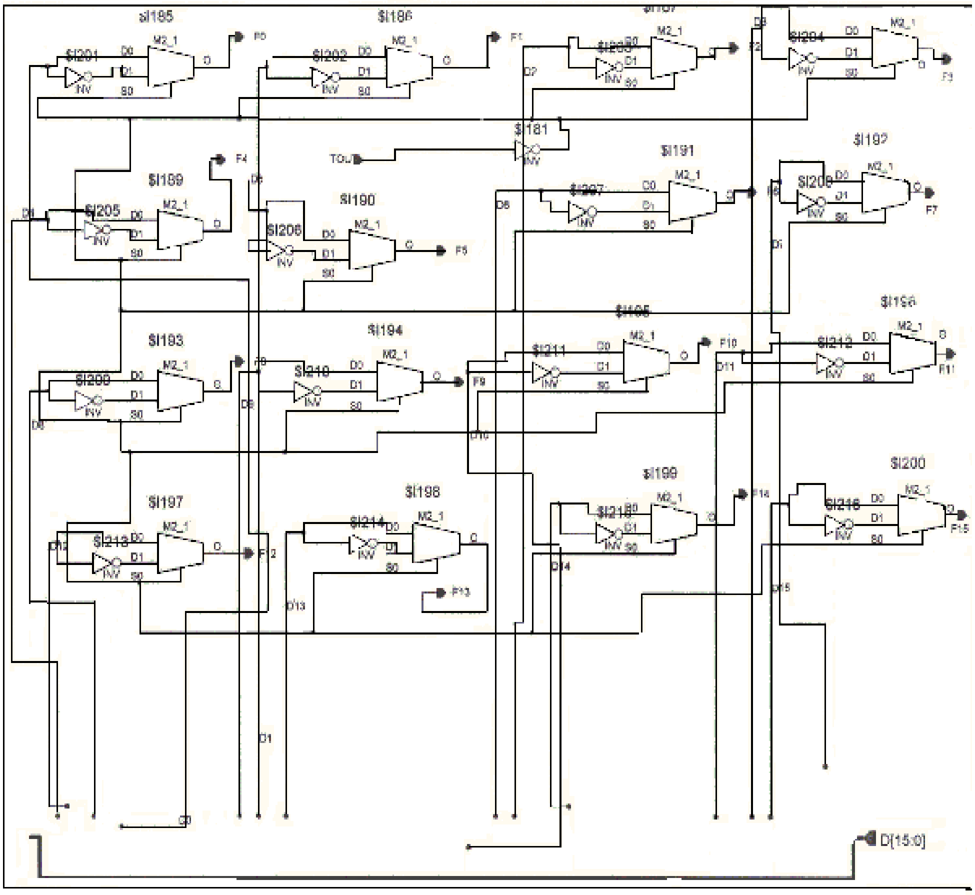

Fig. 7. Controlled complementer circuit

The controlled complementer circuit shown in Fig. 7 receives 16 data bits **D[15:00]** in parallel and coded bit T_{out} as inputs. The 16 data bits D0 ..... D15 are connected in parallel to the first input terminal of the 16 multiplexers. The terminal T_{out} is connected to the select terminal of the MUXs. Second input terminal of each MUX receives the inverted logic level of the first input. Each MUX is designed in such a manner that when select input is 1, the first input would be selected, otherwise the second input would be selected. When **Tout=l**, it implies that message data and carrier image bits are the same, otherwise, opposite to one another. The logic reveals that when **Tout=1**, data bit appears at the output of the circuit as the image bit. Thus 16 image bits Q[15:0] are obtained from 16 multiplexers.

The output from the controlled complementer is fed to the redundancy remover unit. Fig. 8 shows the detailed circuit of the redundancy remover. The 16-bit substring is divided into 4 parts of length 9 bits, 5 bits and two single bits. There are two 4 bit counter, one for removing redundancy from 9 bits and

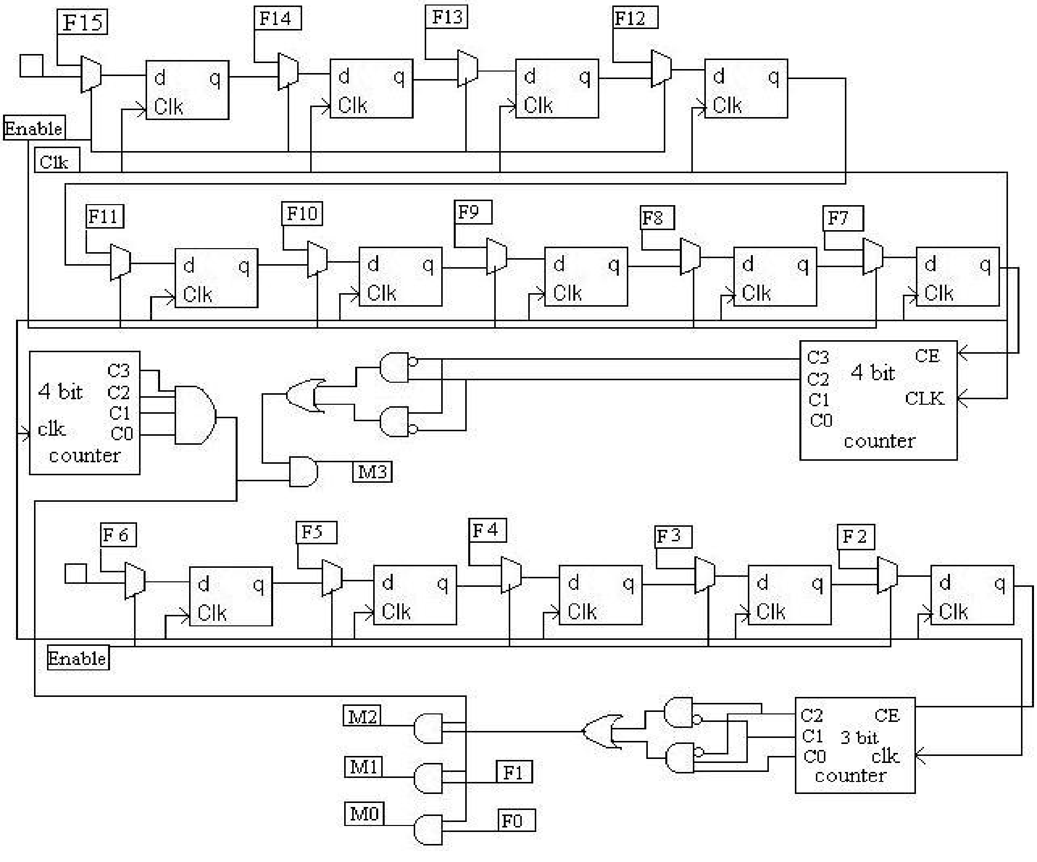

Fig. 8. Redundancy remover

the other one from the next 5 bits. The binary sequence of 9 bits length is fed to a counter to enable its clock. If the counter value is 5 or more, it is further encoded to give an output ‘1’, otherwise ‘0’. Similarly, redundancy is removed from the bit string of five bits. Thus each substring of 16 bit is converted to a 4 bit after redundancy removal. By converting each 16 bit substring to 4 bit string, the message signal is decoded.

4 Performance evaluation and discussion

In this section, we present reliability of message decoding under several forms of distortions of the original image as well by different degree of noise corruption to the binary modulated signal. A brief discussion is also made about the results of hardware design in terms of throughput and number of CLB (configurable logic block) required for implementation.

One important point to be mentioned here is that the proposed algorithm

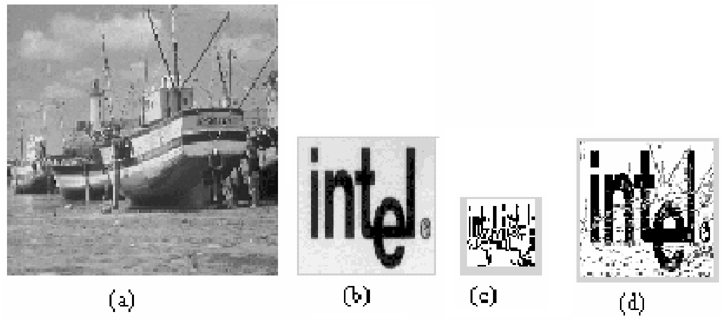

Fig. 9. (a) Original image, (b) message signal (gray scale image) (c) encoded binary image (d) Decoded message signal

keeps the original image completely unchanged except some distortion constraint scenario or any common or deliberate operation applied on it. Hence, the value of structural similarity index measure (SSIM) between the original image at the encoder and its undistorted version available at the decoder is 1, the maximum possible value [25]. Similarly, Kullback-Leibler distance (KLD) used for statistical invisibility would also be the lowest value i.e numerical value '0' that indicates perfect security [27]. Hence quantifying or accessing the visual quality is not meaningful here. On the other sense, accessing or quantifying the reliability of the decoded message after various forms of distortions applied on the original signal is an important issue and is considered here.

4.1 Decoding reliability

We have reported simulation results for the message signal which is a gray scale image of size (64×64), 4 bit/pixel and the original image (carrier) is also a gray-scale image of size (256×256), 8 bit/pixel. Each substring consists of 16 symbols and the size of the binary modulated signal is (64×64). We have tested performance of the proposed algorithm both in terms of the various forms of distortions applied on the original image and noise corruption on the modulated signal. This is analogous to the concept of drift in local oscillator carrier frequency and channel noise in synchronous detection of digital communication, respectively. It is seen that the extracted messages are quite recognizable even after greater depth of degradations like linear and non linear spatial filtering, sharpening, histogram equalization, lossy JPEG and JPEG 2000 compression, noise addition, rescaling etc. occurred on the original image. It is also seen that the decoded message is well recognized even after 30% change in binary modulated signal i.e. even if, due to random noise signal, more than 30% bits in the binary modulated data string are flipped, the extracted message can be recognized faithfully.

The quality of the extracted message is quantified by mutual information value $I(X;Y)$ where random variables X and Y represent the encoded and the decoded message images, respectively. The reason for the consideration of mutual information $I(X;Y)$ as an objective measure stems from the fact that if there were no image impairments (channel noise), the average amount of information received would be $H(X)$ bits (entropy of the source) per received symbol. But because of the channel noise, an average of $H(X/Y)$ bits (called the equivocation of X with respect to Y) of information per symbols is lost. The amount of information the receiver receives is, on the average, $I(X;Y)$ bits per received symbol [27], where

$$I(X;Y) = H(X) - H(X/Y) \tag{2}$$

If $p(x_i)$ represents the probability of occurrence of the i-th pixel value in the transmitted image (gray scale image like information signal) and $p(y_j/x_i)$ represents the channel transition matrix, $I(X;Y)$ that represents the average amount of information received from the signal degradation, can be expressed as follows :

$$I(X;Y) = \sum_i \sum_j p(x_i)p(y_j/x_i) \log \frac{p(y_j/x_i)}{\sum_i p(x_i)p(y_j/x_i)} \tag{3}$$

where i, j represent the spatial location or index of the symbols.

We study the performance of the proposed method over large number of benchmark images [28],[29] and gray scale image like message signals. Fig. 9(a) (fishing boat) shows one such test image and Fig. 9(b), 9(c) and 9(d) represent the message, the binary modulated signal and the decoded message, respectively when there is neither any distortion on the original image nor any form of noise disturbance seen in the binary modulated signal. The entropy $H(X)$ of the message, shown in Fig. 9(b), is 0.867 while the $I(X;Y)$ value for the decoded message (shown in Fig. 9(d)) is 0.624. It is seen that $I(X;Y)$ value is not equal to $H(X)$ value even at ideal situation i.e. when both original image as well as binary modulated signal are not noise corrupted. In other words, $H(X/Y)$ value is nonzero even at noiseless situation. Table 1 shows the message encoding loss i.e. loss in the process of binary message formation based on the size of substring length. The numerical values in Table 1 are computed by averaging the results obtained from the combination of large number of test images and message signals. The results in the table reflects the fact that probability of error for 3-rd bit is higher compared to 4-th bit as amount of redundancy incorporated for the latter is higher compared to the former.

We test the decoding reliability through $I(X;Y)$ values that are obtained from the combination of the proposed channel coding and spatial bi-phase modulation. The mutual information $I(X;Y)$ values may be changed due to the (i) distortion on the original image alone, (ii) noise corruption on modulated binary image alone, as well as (iii) distortion and noise corruption of both orig-

Table 1
Probability of error in single bit, 3rd bit and 4-th bit

Length of substring	Prob. of bit error	Prob. of wrong dec. in 4th bit	Prob. of wrong dec. in 3rd bit	$I(X;Y)$ value
4	0.1534	0.0017	0.0199	0.769
16	0.1787	0.0037	0.0338	0.624
64	0.2594	0.0176	0.0743	0.512
256	0.3530	0.0731	0.1844	0.384

inal and binary modulated signal. Table 2 shows $I(X;Y)$ values for different degree of additive white gaussian noise (AWGN) and consequent recognizability of the decoded message.

Table 2
$I(X;Y)$ values due to the combination of AWGN effect with different variance

Original image	Binary mod. signal	$I(X;Y)$ value of Decoded message	Recognizability of message
No AWGN	No AWGN	0.624	yes
AWGN with 0.05	No AWGN	0.576	yes
No AWGN	AWGN with 0.03	0.493	yes
AWGN with 0.05	AWGN with 0.03	0.412	yes

Decoding reliability of the message signal for the proposed algorithm is also studied for various possible signal processing operations applied on the original image. However, the distorted images are not shown due to space but the respective decoded messages are shown in Figs. 10(a)- (t). The signal processing operations include common manipulation to image dithering, shearing, warping and wiener filtering etc. available in checkmark package [30]. The I (X;Y) values for the decoded messages are 0.51 and 0.56 when the original image is mean and median filtered with PSNR (peak signal to noise ratio) values 22.56 dB and 25.36 dB, respectively. The $I(X;Y)$ values for the decoded message signals are 0.48 and 0.41, respectively when the original image is compressed by JPEG and JPEG 2000 operations at quality factor 50. Comparison of the chekmark package results with the other existing works Cox et al [1], Maity et al [25] and Chang et al [6] are also reported in Table 3. Numerical values show that our proposed image-in-image communication algorithm offers the best decoded message ($I(X;Y)$ values are significantly high compared to other three methods considered here) even after various forms of common and deliberate signal processing operations available in the checkmark package.

Finally, we study the probability of miss-decoding (P_M) when a binary mod-

Table 3
Test results of checkmark package for the proposed, Cox et al [4], Maity et al [25] and Chang et al [6]

Name of attack	$I(X;Y)$ value Prop. algo.	$I(X;Y)$ value Cox et al [1]	$I(X;Y)$ value Maity et al [25]	$I(X;Y)$ value Chang et al [6]
Wiener fil.	0.6014	0.3756	0.3678	0.3872
dpr	0.524	0.3942	0.3834	0.3012
dprcorr	0.5783	0.2564	0.3143	0.2840
Midpoint	0.5735	0.3432	0.3345	0.3418
Threshold	0.5627	0.3276	0.3156	0.3257
Hard threshold	0.5679	0.3245	0.3356	0.3124
Soft threshold	0.5424	0.3106	0.3014	0.3152
Sample downup	0.5564	0.3256	0.3434	0.3054
dither	0.5965	0.3263	0.3078	0.3134
Trimed mean	0.5665	0.3131	0.3242	0.3143
Copy-collage	0.5764	0.3189	0.3056	0.3224
Projective	0.5832	0.3223	0.3164	0.3275
Ratio	0.6145	0.3342	0.3228	0.3243
Rowcol	0.6076	0.3127	0.3023	0.2912
Shearing	0.6057	0.3175	0.3234	0.3176
Warping	0.5987	0.3156	0.3034	0.3148

ulated signal obtained from a particular combination of message signal and the original image is used for message decoding from the other original signal i.e. inter channel/carrier interference (ICI) in synchronous detection. Simulation was carried out for the combination of large number of original images and message signals and visual inspection was used to test recognizability of the decoded message. To quantify the visual recognizability of the decoded message, we consider a threshold value of $I(X;Y)$. Although, it is difficult to correlate between the visual recognizability and threshold $I(X;Y)$ value, after simulation over large number of combination of original and message signals of the above types, we consider 0.32 as threshold $I(X;Y)$. Simulation results show that probability of miss-decoding (P_M) is of the order of 10^{-4} which in turn indicates high security of message decoding.

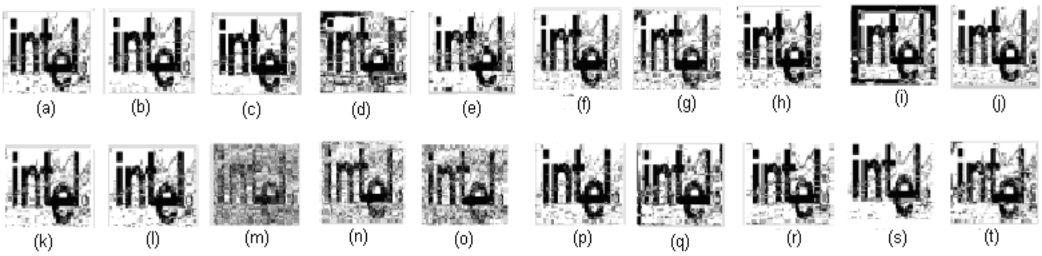

Fig. 10. (a) Decoded message after three times mean filtering of the original image (carrier) using window size (3×3), (b) Decoded message after three times median filtering of the original image using window size (3×3), (c) Decoded message after five times gaussian filtering of the original image with variance 1, window size (9×9), (d) Decoded message after histogram equalization of the original image, (e) Decoded message after image sharpening, (f) Decoded message after noise addition, (g) Decoded message after change in dynamic range from 252- 4 to 200-50, (h) Decoded message after image rescaling, (i) Decoded message after image cropping operation, (j) Decoded message after least significant bits manipulation, (k) Decoded message after JPEG compression at quality factor 30, (l) Decoded message after JPEG 2000 compression at quality factor 30, (m) Decoded message after image dithering, (n) Decoded message after additive gaussian noise with variance 0.01, (o) Decoded message after speckle noise with variance 0.04, (p) Decoded message after sample down up operation, (q) Decoded message after image shearing operation, (r) Decoded message after stirmark operation, (s) Decoded message after wiener filtering, (t) Decoded message after image warping operation.

4.2 Result of hardware design

The running message encoding is updated with the arrival of new sample i.e. after the completion of previous message sample decoding. Each updating requires 17 clock cycles for 4 bits/pixel gray scale image-like message signal. This total clock cycle requirement includes 4 bits/pixel gray scale image signal encoding 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 maximum clock frequency is 80 MHz and clock cycle 17cycles/message pixel value. The data transmission rate i.e. throughput can be achieved at 4.706 Mbits/sec. Input specifications of hardware realization is summarized in Table 4. The chip used is XCS05 or XCS05XL which contains 100 CLB (configurable logic block), out of which 40 CLBs are used for message encoding and 45 CLBs are used for message decoding.

Table 5 shows the performance comparison of some of the hardware design for watermarking in current literature. Notable contributions are found from the research works of Mohanty et al [21,22, 23]. Majority of the works reported

Table 4
Specification of hardware realization

Original image	Message signal	Implemen- tation	CLB count	Clock freq.	Clock cycle	Through- put
(256×256) 8 bits/pixel	(64×64) 4bits/pixel	XCS05 XCS05 XL	85	80	17/ message sample	4.706 Mbits/s

are based on custom integrated circuit, while this work is FPGA based realization. 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. The throughput would certainly be further increased if higher end FPGA such as Virtex PRO etc. is used.

Table 5
Test results of data hiding using hardware realization

Reported work	Target object	Working domain	Techno- logy	Gate count/CLB	Clock freq.
Mathai [11]	Video	Wavelet	0.18 μm	NR	NR
Tsai [19]	image	DCT	0.35μm	46374	50MHz
Garimella[20]	Image	Spatial	0.13μm	NR	100MHz
Mohanty [21]	image	Spatial	0.35μm	28469	292 MHz
Mohanty [22]	Image	DCT	0.25μm	NR	280 and 70MHz
Maity [24]	Image	FWT	FPGA	730 CLB	80 MHz
This work	Image	spatial	FPGA	85 CLB	80 MHz

5 Conclusions and Scope of Future Works

An algorithm for distortion free covert image-in-image communication and its VLSI realization using FPGA is proposed in this work. Faithful decoding of a message is possible provided that the binary modulated signal is available at the receiver end by transmitting either through a secured channel or through a channel subject to a certain noise constraint for the modulated data. The algorithm can be applied to unchangeable image, multi-owner original image sharing, medical imagery, low power verification systems, military images inspected under unusual visual conditions etc. Algorithm requires few simple computations 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 Transaction on Image Processing, 6 (12)(1997) 1673-1687.
[2] D.-C. Lou and J.-L. Liu, Steganographic method for secure communications, Computers Security, 21(5)(2002) 449-460.
[3]A. Phadikar and S. P. Maity, Quality access control of compressed color images using data hiding,AEU Journal of Electronics and Communication Engineering, 64(2010)833-843.
[4] J.J. Fridrich, M. Golijan, and R. Du, Lossless data embedding:new paradigm in digital watermarking, EURASIP Journal of Signal Processing, 2, pp.185-196, 2002.
[5]P. Tsai, Y.C. Hu, and H. L. Yeh, Reversible image hiding scheme using predictive coding and histogram shifting, Signal Processing,89(6),(2009) 1129-1143.
[6]C.C. Chang, J.C. Chuang, and H. M. Hand, An image intelligent property protection scheme for gray-level images using visual secret sharing strategy, Pattern Recognition Letters, 23(2002) 931-941.
[7] S.W. Weng, Y. Zaho and J.S. Pan, Reversible watermarking resistant to cropping attack, IET Information Security, 1 (2007) 91-95.
[8] I. Absivas, N.Menon and B. Sankur, Steganalysis based image quality metrics-Differentiating between techniques, IEEE workshop on multimedia, Cannes, France, October 2001.
[9] S. P.Maity, A. Banerjee and M. K. Kundu, An image-in-image communication scheme and VLSI implementation using FPGA, Proceedings Of IEEE Indian Annual Conference (INDICON 2004), IIT Kharagpur, 2004,pp. 6-11.
[10] E.Kougianos, S. P.Mohanty, and R. N. Mahapatra, Hardware assisted watermarking for multimedia, Special Issue on Circuits and Systems for Real-Time Security and Copyright Protection of Multimedia, International Journal on Computers and Electrical Engineering (IJCEE) 35(2)(2009) 339-358.
[11]N.J. Mathai, D.Kundur, A. Sheikholeslami, Hardware implementation perspectives of digital video watermarking algorithms,IEEE Transaction on Signal Processing, 51(2003) 925-938.
[12] S. P. Maity, A. Banerjee, A. Abhijit and M. K. Kundu, VLSI design of spread spectrum watermarking, Proceedings of 13th National Conference on Communications, IIT Kanpur, India, 2007, p. 251-257.
[13] M.Maes, T.Kalker, J. P. M. G. Linnartz, J. Talstra, G. F. G. Depovere and J. Haitsma, Digital Watermarking for DVD Video Copyright Protection, IEEE Signal Processing Magazine, 17(2000), 47-57.
[14] S. P.Mohanty, E. Kougianos and N. Ranganathan, VLSI architecture and chip for combined invisible robust and fragile watermarking, IET Computer and Digital Technique, 1(2007) 600-611.
[15]S. P. Maity, Spread spectrum watermarking:implementation in FPGA,

Advanced techniques in multimedia watermarking, A. M. Al-haj (Ed.), IGI Global, Information Science Reference, Hershey, New York, pp. 455-485, 2010.

[16] L. D. Strycker, P. Termont, J. Vandewege, J. Haitsma, A. Kalker, M. Maes and G. Depovere, Implementation of a real-time digital watermarking process for broadcast monitoring on trimedia VLIW processor, IEEE Proceedings on vision, image and signal processing,147(2000) 371-376.

[17] S. C.Cheung and D. K. W. Chiu, A watermarking infrastructure for enterpries document management. In Proc. of 36th annual Hawaii international conference on system sciences; 2003, p. 105-14.

[18] Y. C. Fan, L. D Van, C. M. Huang, H. W. Tsao, Hardware-efficient architecture design of wavelet-based adaptive visible watermarking, In Proceedings of the 9th IEEE international symposium on consumer electronics; 2005, p. 399-403.

[19] T. H. Tsai, C. Y. Lu, A systems level design for embedded watermark technique using DSC systems. In Proceedings of the IEEE International Workshop on Intelligent Signal Processing and Communication Systems; 2001.

[20] A. Garimella, M. V. V Satyanarayan, R. S. Kumar, P. S. Murugesh, U.C. Niranjan VLSI implementation of online digital watermarking techniques with difference encoding for the 8-bit gray scale images. In Proceedings of the international conference on VLSI design, 2003. p. 28388.

[21]S. P. Mohanty, N. Ranganathan and R. K. Nambala, A VLSI architecture for visible watermarking in a secure still digital camera (S2DC), IEEE Transaction on Very Large Scale Integration System (TVLSI), 13(2005) 1002-12.

[22] S. P. Mohanty, N. Ranganathan, K.Balakrishnan, A dual voltage-frequency VLSI chip for image watermarking in DCT domain,IEEE Trans Circ Syst II (TCAS- 639 II) 53(2006)3948.

[23] S. P. Mohanty, E. Kougianos, N. Ranganathan, VLSI architecture and chip for combined invisible robust and fragile watermarking, IET Comput Digital Tech 1(2007)60011.

[24] S. P. Maity, M. K. Kundu, S. Maity, Dual purpose FWT domain spread spectrum image watermarking in real time, Special issue circuits and systems for real-time security and copyright protection of multimedia, International journal of Computers and Electrical Engineering, Elsevier, 35(2009) 415-433.

[25] S. P. Maity and M. K. Kundu, DHT domain digital watermarking with low loss in image information, AEU Journal of Electronics and Communication, 64(3)(2010)243-257.

[26]R. C. Gonzalez and R. E. Woods, Digital Image Processing Using Matlab, 2nd ed. Prentice Hall, Upper Saddle River, NJ., 2005.

[27]R.H. Hamming, Coding and Information Theory. Prentice-Hall, Inc., Englewoods Cliffs, New Jersey, 1980.

[28]http:// www.cl.cam.ac.uk/ fapp2/watermarking.

[29]http://www.petitcolas.net/fabien/watermarking/image$_d atabase/index.html.$

[30]$http://watermarking.unige.ch/checkmark$