VLSI Implementation of Digital Image Watermarking

Rahate Kunal B.
PG student
Dept. of Electronics and Telecommunication
Govt. College of Engineering.
Aurangabad(M.S.)

Dr. A. S. Bhalchandra
Professor
Dept. of Electronics and Telecommunication
Govt. College of Engineering..
Aurangabad(M.S.)

S. S. Agrawal
Assistant Professor
Dept. of Electronics and Telecommunication
Govt. College of Engineering..
Aurangabad(M.S.)

Abstract

The process of Digital watermark embeds the data called watermark in digital media like image, video audio file etc so that the owner can claim for rights. This paper presents invisible fragile watermarking algorithm and watermark retrieval algorithm along with copyright protection provision. The algorithms are implemented in spatial domain. The pixel wise manipulation of the image to be watermarked (base image) is done in accordance with the pixel values of the watermark. The paper represents the complete software implementation of both the algorithms and the hardware implementation of the same is done on Spartan3 FPGA. Several attempts have been made to achieve low power, low area and high performance. For low and optimized area, the IP cores of block ROM and division generator are used. The task of providing input to the board from PC and getting results from the board to the PC are done using serial communication through UART protocol.

1. Introduction

One of the ways to protect the intellectual property rights of the digital media is digital watermarking. With rapid increase in use of internet and digital media, transmission and reproduction of digital products has become very convenient but it has some drawbacks also. The digital revolution provides tools to unlimited copying without loss in fidelity [1]. The people can easily steal the digital work of others like image, videos, and audio clips and claim their rights on the stolen things. This situation creates the need of copyright protection of digital media. Digital watermarking can be used for content authentication, detection of illegal duplication and alteration, secret communication as watermarking is very helpful in hiding the secret messages or information in the digital media. Using watermarking the people can keep their work copyrighted. The watermarking algorithm incorporates the watermark in the object, whereas the verification algorithm authenticates the object by determining the presence of the watermark and its actual data bits [2] [3].

Watermark has various forms like text value, image, video and audio clip. Watermarking technique has some properties to be defined like robustness, security, complexity, verification etc. Robustness is important property because it defines the survival of watermark in watermarked digital media after going through operations like filtering lossy compression or some kind of geometric modification. The attacks on watermarked media can sometimes lead to removal of watermark. To avoid such things the improved watermark insertion key should be used, such things are taken care by means security. The time required for the algorithm to embed the watermark in to digital media and also its retrieval defines the complexity of watermarking.

The watermarking system can be implemented with either software or hardware. Software implementation of watermarking is large whereas hardware implementation is lacking [4]. Generally, hardware watermarking scheme can be done by using each of the domains (spatial or frequency). Due to the simplicity of spatial domain computational overhead and its easiness for its application if compared to the frequency domain, the spatial domain is usually preferred for hardware implementation [4] [5] [6] [7].

The paper is organized as follows: Section 2 describes the proposed algorithm. Sections 3 and4 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 Scheme for Watermarking

2.1. Algorithm for Watermark Insertion

Digital image watermarking can be done in both spatial domain and frequency domain but the most straight forward fundamental and simplest schemes for digital image watermarking can be implemented in frequency domain because of the less complexity. Such types of techniques deal with modification of luminance value of pixels in spatial domain. In this paper the most common technique of watermarking is presented which involves the manipulation of least significant bits (LSB) of overall pixels of base image or the image to be watermarked. Although the spatial domain watermarking is less complex, but it is not much robust as compared to the frequency domain watermarking. [5] [8]. In this paper the watermarking of 8 bit gray scale image is presented. According to [5] [6] the number of bits 'm' in LSB to be modified can be obtained using:

$$m = \log_2 [M \times M] \quad (1)$$

Where '[M × M]' is the size of the base image.

The reason of particularly modifying only LSBs of the base image is that, the LSBs contains the least information of the pixel value and such manipulation does not hamper the quality of watermarked image. The number of LSBs to be modified can be found out using equation (1). In this paper the base image size is taken as 128X128 and the 3 LSB bits are manipulated of all the pixels of base image.

To embed the watermark in to the base image the LSBs of pixels the base image are replaced by the same number of MSBs of the watermark pixels. In other words the least information holding bits of the base image are replaced by the most information holding bits of the watermark.

The following steps summarises the embedding algorithm.

1) The size of the base image is obtained and the 3 LSBs of all the pixels of base image are set to zero.

2) The size of watermark is obtained and if the watermark size is greater, then it is reshaped to the size of base image.

3) Modify the pixels in watermark in such a way that the first 3 MSBs are retained and all other bits are set to zero.

4) Left shift all the pixels in watermark 3 times.

5) Add pixels of base image with the corresponding pixels of watermark.

6) The resultant pixels represent the watermarked image of the base image.

2.2. Watermark Retrieval Algorithm

The watermarked image quality is maintained by only modifying 3LSBs out of 8 bits of base image. To extract watermark, these modified bits are required. Following steps explains the retrieval algorithm.

1) Set every bit of every pixel to zero except 3LSBs.

2) Shift right every pixel by 3 times.

3) The resultant image is the original watermark which was initially used to embed into the base image.

3. VLSI Architecture

3.1. Datapath for Watermark Insertion

The datapath is divided in to four parts. First is UART module. This module is responsible for the PC to FPGA board interface. The second part is watermarking module which mainly consist of two RAMs to store the base image and resultant watermarked image. It also has and two gates and a shifter. Two intellectual property core (IP Core) are also incorporated in datapath. One is for storing the watermark and one is 'Division Generator'.

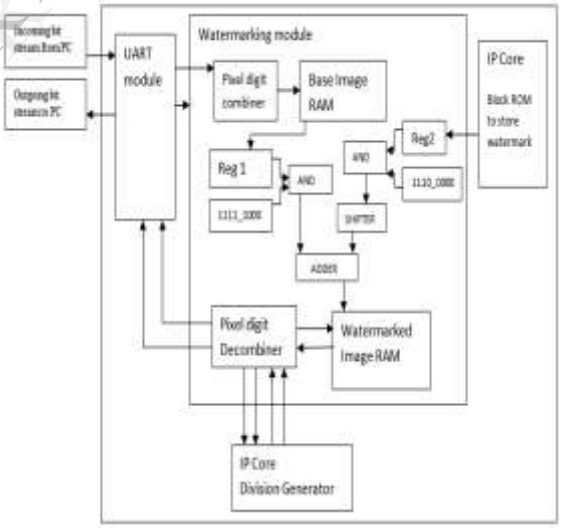


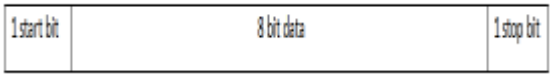
Fig.1: Data path for Invisible Fragile Watermarking

The designing process is carried out in the Xilinx project navigator 13.2 and the hardware description language used is Verilog. The design is simulated and the waveforms are verified in the ISE simulator. Once the designing process is done, the design is ready to implement in the FPGA board. In the simulation, we can provide the image as a input using the Testbench but to provide the input to FPGA board from PC, some sort of interface must be provided. Once the input is given to the FPGA board, the design can be operated and the results are stored in block ram. The RAM

containing the result has to be given to some sort of display like LCD screen or VGA monitor to analyze. In this paper the results are transmitted back to the PC and the displayed in MATLAB. The architecture of watermarking operates by combined action of all the modules.

3.1.1. UART Module:

The base image to be watermarked is first converted to the text format in MATLAB. These text values represent the actual image. The base image is 8 bit image so the pixels values lies between 0000_0000 to 1111_1111 (0-255). For the transmission and the reception, RS 232 port of PC is used UART protocol is used for the serial transmission of these text values. Any terminal software can be used to send the text values to serial port of the PC. UART protocol is asynchronous protocol where the carrier is not used hence the start and end of transmission has to be specified. In this paper the UART module is designed which works with one start bit and one stop bit.



The serial input from PC is received at the FPGA board. The start bit and stop bit are discarded and the 8 bit data (1byte) is accepted. All the pixel values are transmitted and the using the UART module in FPGA the pixel values are received and stored in the RAM. The UART module generates the enable signals which drives the another modules.

The transmission back to PC from FPGA board is done using the same UART module. Start bit and stop bit are attached to the 8 bit data and this 10bit data is serial transmitted back to PC. The respective control signals required for proper operation are generated by watermarking module.

3.1.2. Watermarking module:

The proposed system uses he pixel values in decimal which means the maximum 3 digits are required to present the decimal values from 0 to 255. The serial communication can be used to transfer only one digit at a time. The problem of receiving the multidigit decimal value is solved by using the Pixel Value Combiner which combines the multidigit decimal value and store in the RAM. One by one all the pixels are stored in the base image RAM. The base image size is 255X255 so to store these many pixels the RAM size is also kept same as base image size.

The watermark is stored in the Block ROM IP core. The proposed algorithm is this paper works by manipulating the bits of base image pixels according to the respective pixel of watermark. One to one mapping is maintained for the pixels in base image RAM and the Block ROM for watermark. A counter is created which holds the value of address and this address value is given to both the memories. The counter value keeps on incrementing till all the pixels are processed. First pixel value from base image RAM is given to the register REG1. AND operation is performed between REG1 and another register which holds 1111_1000 value in binary. This resets the last 3 LSBs of base image pixel. The first 3 MSBs of the watermark pixel are kept intact and rests of the pixels are set to zero by passing through another and gate. Successive 3 bit left shifter shifts this pixel value, and added with the pixel value of the base image. The resultant pixels is watermarked pixels which is stored in another RAM called Watermarked image RAM. This RAM also has the same size as that of the Base Image RAM (255X255).

3.1.3. IP Core (Block Memory Generator):

The IP Core (Intellectual property Core) refers to preconfigured logic functions that can be used design. Xilinx provides a wide selection of IP that is optimized for Xilinx FPGAs. These can include functions delivered through the Xilinx CORE Generator software, through the Xilinx Architecture Wizard, as standalone archives, from third parties, through Xilinx Platform Studio (XPS), or through System Generator. Xilinx and its partner companies produce IP ranging in complexity from simple arithmetic operators and delay elements to complex system-level building blocks, such as Digital Signal Processing (DSP) filters, multiplexers, transformers and memory. Xilinx IP is delivered through the tools called CORE Generator System which is a design tool that delivers parameterized cores optimized for Xilinx FPGAs. It provides you with a catalogue of ready-made functions ranging in complexity from simple arithmetic operators such as adders, accumulators, and multipliers, to system-level building blocks such as filters, transforms, FIFOs and memories.

The CORE Generator System creates customized cores which deliver high levels of performance and area efficiency. This is accomplished by taking advantage of Xilinx's core friendly FPGA architectures and Xilinx Smart-IP technology.

The CORE Generator System benefits designers by providing the following features:

- Physical layout optimized for high performance.
- Predictable performance and resource utilization.

• Reduced power requirements achieved through compact design and interconnect minimization.

• Performance independent of target device size.

• Ability to use multiple instances of the same core on the same device without deterioration in performance.

• Reduced compile time compared to competing architectures.

One of the IP core used in the proposed architecture is Block Memory generator. The architecture of watermarking requires two images, one is base image and other is the watermark. The base image is obtained and stored in FPGA RAM using serial communication and the watermark is initially inside the FPGA using Block Memory generator IP Core.

One of the problems of memory initialisation in hardware descriptive languages like Verilog and VHDL is that, the constructs used for such initialisations are not synthesisable. The Verilog uses 'initial 'construct which is not synthesisable. To solve this problem the Block Memory Generator is used. The Block memory generator can be initialised with the '.coe' file. The MATLAB software is used to create .coe file of the watermark.

The specification of block memory used is as follows

Table 1: Block memory specifications

Interface type	native
Memory type	single port ROM
Algorithm to concatenate Block RAM primitives	minimum area
Memory size	Read-width-8 bits Read-depth 255x255

3.1.4. IP Core Division Generator:

The Pixel Decombiner module is incorporated with the Division Generator. The pixel value in RAM for watermarked image varies from 1 digit to maximum 3 digits as the pixel value is in decimal (0-255). The UART module is capable of transmitting only one digit at a time. This requires dividing the multi digit pixel value into single digits and transmitting each digit individually. Division operation is most difficult out of four arithmetic operations in VLSI domain as it includes shifting the pixel bits in right side. This means division only with value which is power of 2 is possible and division with other values is only allowed for simulation but not for synthesis.

This problem is solved using Division Generator, which accepts two input as 'dividend' and 'divisor', and produces two output as quotient and remainder. The width for all the four parameters is set to 8 bits.

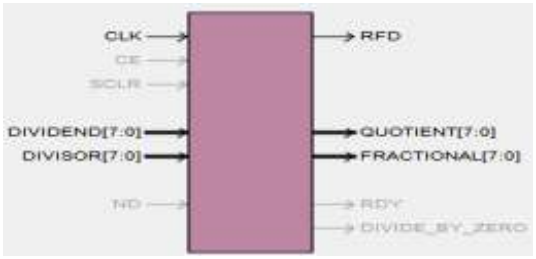


Fig. 2 Pin diagram for Division Generator

3.2. Datapath to Retrieve Watermark

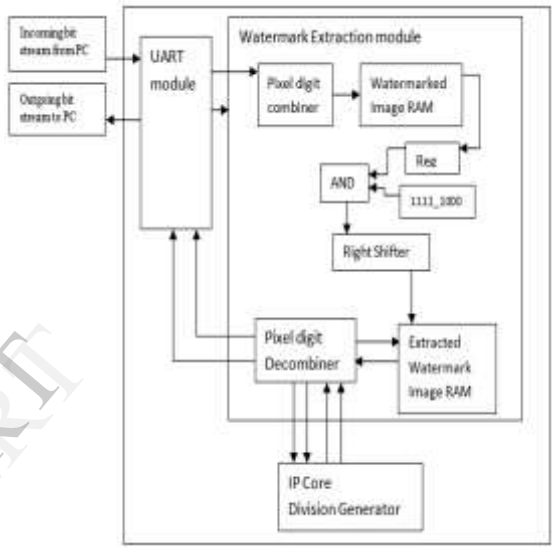


Fig.3: Datapath for Watermark Extraction

The watermark retrievals architecture is presented in figure 3. The watermarked image which is obtained using watermarking algorithm is transmitted using the same serial communication standard and UART protocol. The Block memory generator is removed as no need to store any watermark. Only two RAMs are implemented using block ram inference. The architecture for watermark extraction lacks one and operation and only one right shifter is required. Rest of the operation (providing input and taking output out) are same.

4. Synthesis and Implementation

The complete RTL schematic of the embedding the watermark top module is presented in figure 4. The datapath for image watermarking module shown in fig 3 consist of mainly four blocks. Each block is designed separately and all the four blocks are brought in single top module to work together. The RTL schematic shows all the four blocks in a single top module. The chip is modelled using a

Verilog and functional simulation was performed. The code was synthesized on Spartan 3 technology on xc3s200-4tq144 device using ISE Project Navigator (0.61xd) from Xilinx. The clock 50MHz was given to the Spartan 3 board. The device utilization summary is given in Table 2.

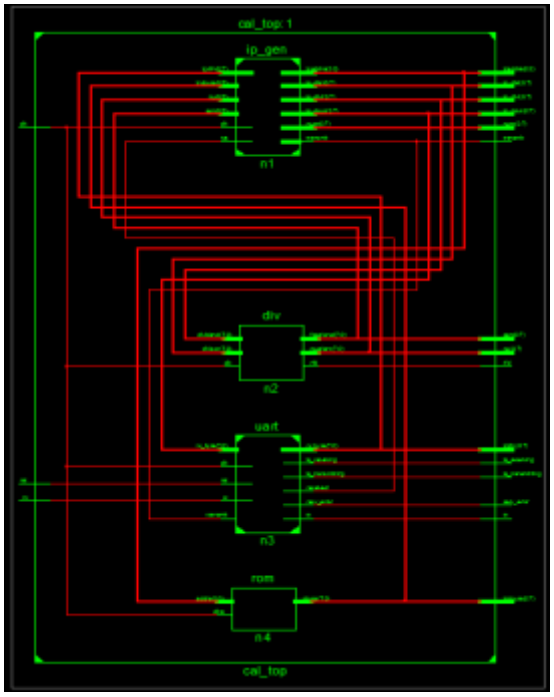


Fig. 4: RTL schematic of top module for watermark insertion

Table 2: Device Utilization Summary

Logic Utilization	Used	Available	Utilization
Number of Slices	690	1920	35%
Number of Slice Flip Flops	552	3840	14%
Number of 4 input LUTs	1140	3840	29%
Number of bonded IOBs	44	97	45%
Number of BRAMs	2	12	16%
Number of MULT18X18s	1	12	8%
Number of GCLKs	1	8	12%

5. Simulation Results

The design is implemented on Spartan 3 FPGA board for the 8 bit gray scale image considering the memory capability of the hardware design, the base image size is kept 64X64 and so the size of watermark is also kept same. For the bigger image sizes, like 256X256 and 512X512 design works properly but with increased memory size and extended hardware requirements. The simulation results include the complete waveforms representation of all the signals included in top module for embedding watermark. The input for UART module is 'r_x' which receives the bit stream of the image pixels serially. Two other signals namely 'is_receiving' and 'recv_error' works as a status signals for reception of image pixels from PC. The transmission of the image pixels from PC to FPGA board continues till the inferred RAM dedicated to store the base image is full. Once the base image RAM is fully stored the appropriate control signals are generated which initializes the watermarking process. The watermarking process involves interaction of image processing block and the IP core which stores the watermark. Appropriate address is placed on the address of IP core block ROM to get pixel value of watermark. The processed pixel is sent back to UART module for transmission and the respective control signals are activated for transmission. The transmission of the watermarked pixel from FPGA to PC is done through the hardware pin named 't_x'. The other signal which is associated with transmission is 'is_transmitting' which represents the status of transmission. The digits in the pixel value are broken in to individual digits and each digit is transmitted individually. This is happening because of the use of Pixel Digit Decombiner. The simulation results are shown in fig4.

6. Conclusion

In this paper, watermarking encoder and the decoder that can perform invisible fragile watermarking in spatial domain is presented along with VLSI realisation using FPGA with developed memory efficient hardware architecture. The

experimental results showed that the proposed watermarking scheme is imperceptible and robust against geometric attacks but fragile against the filtering and image compressions. Great advantages are gained due to using IP core hardware based implementation of watermarking algorithms, such as block ROM and division generator to reduce hardware scheme area, decrease power consumption and increase speed of performance. Therefore a hardware watermarking solution is often more reliable and economical.

References

[1]Er-Hsinen, "Literature Survey on Digital Image Watermarking Watermarking,"EE381K Multidimentional signal Processing, 8/19/98.

[2] C.C.Chang and J. C. Chuan, "An image intellectual property protection scheme for gray images using visual secret sharing strategy," Pattern Recognition Letters, vol. 23, June 2002, pp. 931- 941.

[3] Pankaj U. Lande,Sanjay N.Talbar and G.N. Shinde "ROBUST IMAGE ADAPTIVE WATERMARKING USING FUZZY LOGIC AN FPGA APPROACH" International Journal of Signal Processing, Image Processing and Pattern Recognition Vol. 3, No. 4, December, 2010

[4]S. P. Mohanty, N. Ranganathan, and R. K. Namballa, "VLSI Implementation of Invisible Digital Watermarking algorithms Towards the Developement of a Secure JPEG Encoder," in Proc. Of the IEEE Workshop on Signal Processing Systems, pp. 183-188. 2003.

[5] D. Samanta, A. Basu, T. S. Das, V. H. Mankar, A. Ghosh, M. Das, and S. K Sarkar. SET Based Logic Realization of a Robust Spatial Domain Image Watermarking. In IEEE (ICECE). in. Proc. of 5th International Conference on Electrical and Computer Engineering. Dhaka, Bangladesh. 2008. pp. 986-993.

[6]A. Basu, T. S. Das, S. Maiti, N. Islam, and S. K. Sarkar. FPGA Based Implementation of Robust Spatial Domain Image Watermarking Algorithm. in Proc. in International Conference on Computers and Devices for Communication. 2009.

[7] A. Basu, T. Das, S. Sarkar, A. Roy, and N. Islam. FPGA Prototype of Visual Information Hiding. IEEE. 2010.

[8]J. Pan, H. C. Huang, and L. C. Jain. Intelligent Watermarking Techniques. World Scientific, 2004.