IMAGE PROCESSING AND DATA HIDING FRAMEWORK ON FPGA BASED PLATFORM

[1]SHREEDEEP GANGOPADHYAY, [2]BHASKAR BANERJEE

Techno India, EM 4/1 Salt Lake City, Sector- V, Kolkata
Email: bdeep_mistu@yahoo.com, checkmate872@gmail.com

Abstract— With the introduction of reconfigurable platform such as Field Programmable Gate Arrays (FPGA) and advent of new high level tools to configure them, image processing on FPGA has emerged as practical solutions for most of computer vision and image processing problems. The use rapid prototyping tools such as MATLAB Simulink and Xilinx System Generator becomes increasingly important because of time-to-market constraints. Image processing has very important application in digital domain ranging from medical application through image enhancement to copy right protection through water marking technique .This project presents a methodology for implementing real-time image processing applications on a reconfigurable logic platform using Xilinx System Generator (XSG) for Matlab.

Keywords- Colour watermarking, invisible watermarking, system generator, MATLAB, FPGA, image enhancement

I. INTRODUCTION

Digital image processing is an ever expanding and dynamic area with applications reaching out into our everyday life such as medicine, space exploration, surveillance, authentication, automated industry inspection and many more areas. Applications such as these involve different processes like image enhancement and object detection. Implementing such applications on a general purpose computer can be easier, but not very time efficient due to additional constraints on memory and other peripheral devices. Application specific hardware implementation offers much greater speed than a software implementation. With advances in the VLSI (Very Large Scale Integrated) technology hardware implementation has become an attractive alternative. Implementing complex computation tasks on hardware and by exploiting parallelism and pipelining in algorithms yield significant reduction in execution times. Field Programmable Gate Arrays are reconfigurable devices. Hardware design techniques such as parallelism and pipelining techniques can be developed on a FPGA, which is not possible in dedicated DSP designs. Implementing image processing algorithms on reconfigurable hardware minimizes the time-to-market cost, enables rapid prototyping of complex algorithms and simplifies debugging and verification. Therefore, FPGAs are an ideal choice for implementation of real time image processing algorithms.

An image may be defined as a two-dimensional function,f (x,y) , where *x* and *y* are *spatial* (plane) coordinates, and the amplitude of *f* at any pair of coordinates (*x*, *y*) is called the *intensity* or *gray level* of the image at that point. When *x*, *y*, and the intensity values of *f* are all finite, discrete quantities, we call the image a *digital image*. The field of *digital image processing* refers to processing digital images by means of a digital computer. Note that a digital image is composed of a finite number of elements, each of which has a particular location and value. These elements are called *picture elements*, *image elements*, *pels*, and *pixels*. *Pixel* is the term used most widely to denote the elements of a digital image. Vision is the most advanced of our senses, so it is not surprising that images play the single most important role in human perception. However, unlike humans, who are limited to the visual band of the electromagnetic (EM) spectrum, imaging machines cover almost the entire EM spectrum, ranging from gamma to radio waves. They can operate on images generated by sources that humans are not accustomed to associating with images. These include ultrasound, electron microscopy, and computer-generated images. Thus, digital image processing encompasses a wide and varied field of applications

Steganography and watermarking describe methods to embed information transparently into a carrier signal. Steganography is a method that establishes a covered information channel in point-to-point connections, whereas watermarking does not necessarily hide the fact of secret transmission of information from third persons. Besides preservation of the carrier signal quality, watermarking generally has the additional requirement of robustness against manipulations intended to remove the embedded information from the marked carrier object. This makes watermarking appropriate for applications where the knowledge of a hidden message leads to a potential danger of manipulation. However, even knowledge of an existing hidden message should not be sufficient for the removal of the message without knowledge of additional parameters such as secret keys. A crucial feature of digital watermarking is to hide the additional information not in distinguished locations in a specific media format such as the header of a file—which could be lost during transformation into another presentation format—but directly in the signal to be marked itself. This requires a certain perceptual threshold allowing the insertion of

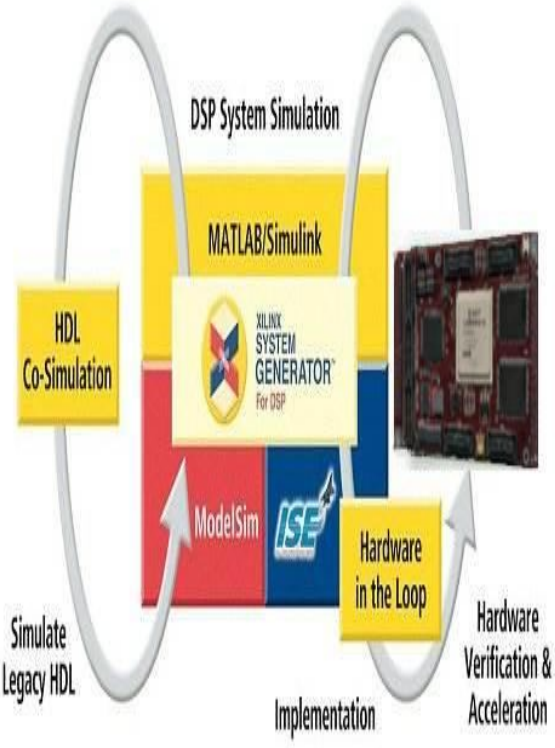
additional information and hence distortions of the carrier signal without incurring unacceptable perceptual degradation of the original carrier signal. This implies that marking of executable program code will be difficult, since any arbitrary modification to the bit stream could destroy the functioning of the program, while changes not affecting the semantics of the program can be removed easily through a normalization process. Watermarking systems are therefore context-specific, that is, the algorithms must be designed with respect to the media type of the data to be watermarked. In this sense, watermarking represents a specific application of steganographic techniques.

It is requirement of an efficient rapid prototyping system to develop an environment targeting the hardware design platform. Although the Xilinx ISE 12.1 foundation software is not directly utilized, it is required due to the fact that it is running in the background when the System Generator blocks are implemented. The System Generator environment allows for the Xilinx line of FPGAs to be interface directly with Simulink. In addition there are several cost effective development boards available on the market that can be utilized for the software design development phase. Xilinx System Generator (XSG) is an integrator design environment (IDE) for FPGAs, which uses Simulink, as a development environment, it is presenting in the form of block set. It has an integrated design flow, to move directly to the configuration file (*.bit) necessary for programming the FPGA. One of the most important features of Xilinx System Generator is possessed abstraction arithmetic, which is working with representation in fixed point with a precision arbitrary, including quantization and overflow. You can also perform simulation both as a fixed-point double precision. XSG automatically generates VHDL code and a draft of the ISE model being develop. Make hierarchical VHDL Synthesis, expansion and mapping hardware, in addition to generating a user constraint file (UCF), simulation and test bench and test vectors among other things. Xilinx System Generator has created primarily to deal with complex Digital signal processing (DSP) applications, but it has other application like the theme of this work . The blocks in Xilinx System Generator operate with Boolean values or arbitrary values in fixed point, for a better approach to hardware implementation. In contrast Simulink works with numbers of double-precision floating point. The connection between blocks, Xilinx system generator and Simulink Blocks are gateway blocks. Figure.1 shows the broad flow design Xilinx System Generator. As already mentioned, you can then move to the configuration file to program the FPGA.

II. OVERVIEW OF WATERMARKING AND DATA HIDING FRAMEWORK

Watermarking is a technique used to hide data or identifying information within digital. The enormous popularity of the World Wide Web in the early 1990's demonstrated the commercial potential of offering multimedia resources through the digital networks. Since commercial interests seek to use the digital networks to offer digital media for profit, they have a strong interest in protecting their ownership rights. Digital watermarking has been proposed as one way to accomplish this.

A digital watermark is a digital signal or pattern inserted into a digital image. Since this signal or pattern is present in each unaltered copy of the original image, the digital watermark may also serve as a digital signature for the copies. A given watermark may be unique to each copy (e.g. to identify the intended recipient), or be serve as a digital signature for the copies. A given watermark may be unique to each copy (e.g. to identify the intended recipient), or be common to multiple copies (e.g. to identify the document source). In either case, the watermarking of the document involves the transformation of the original into another form



.

Fig. II.a System generator design flow

Water marking can also be made invisible to transmit data in the form of audio, image embedded into either of the two to form a data hiding ramework.

Our discussing in this paper will primarily focus on how digital watermarking can be used as a tool to hide data and transmit it through world wide web and use watermarking to protect ownership rights.

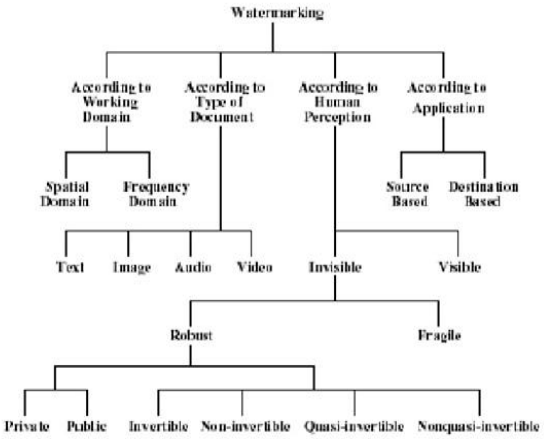


Fig.II.b Types of watermarking

Watermarking requirements.

- **Security:** The security requirement of a watermarking system can differ slightly depending on the application. Watermarking security implies that the watermark should be difficult to remove or alter without damaging the host signal. As all watermarking systems seek to protect watermark information, without loss of generality, watermarking security can be regarded as the ability to assure secrecy and integrity of the watermark information, and resist malicious attacks.
- **Imperceptibility:** The imperceptibility refers to the perceptual transparency of the watermark. Ideally, no perceptible difference between the watermarked and original signal should exist . A straightforward way to reduce distortion during watermarking process is embedding the watermark into the perceptually insignificant portion of the host signal . However, this makes it easy for an attacker to alter the watermark information without being noticed.
- **Capacity:** Watermarking capacity normally refers to the amount of information that can be embedded into a host signal. Generally speaking, capacity requirement always struggle against two other important requirements, that is, imperceptibility and robustness . A higher capacity is usually obtained at the expense of either robustness strength or imperceptibility, or both.

III. XILINX SYSTEM GENERATOR AND MATLAB

A. OVERVIEW OF XILINX VIRTEX-5 FPGA ML506

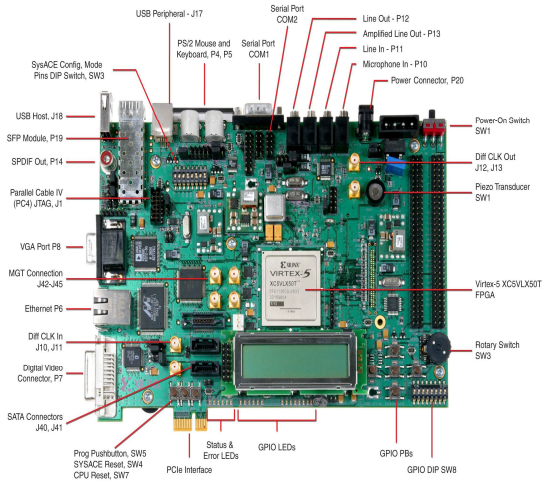


Fig.III.a Vitrex ML506 FPGA board

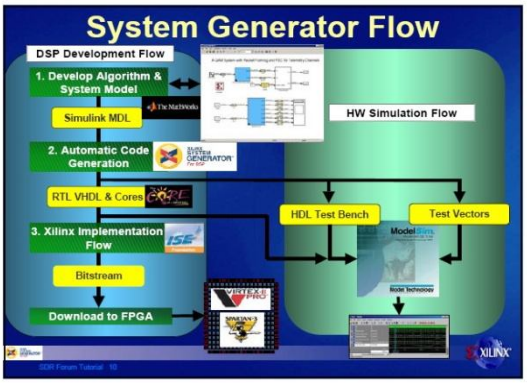
We are using Vitrex ML506 FPGA board in our project. The XtremeDSP™ development platform — Virtex®-5 ML506 FPGA edition is a feature-rich DSP general purpose evaluation and development platform. Though economically priced, the ML506 offers users the ability to create DSP based and high speed serial designs utilizing the Virtex-5 FPGA DSP48E slices and RocketIO™ GTP transceivers. A variety of on-board memories and industry standard connectivity interfaces add to the ML506's ability to serve as a versatile development platform for embedded applications.

B.Xilinx System Generator Software

System Generator allows device-specific hardware designs to be constructed directly in a flexible high-level system modeling environment. In a System Generator design, signals are not just bits. They can be signed and unsigned fixed-point numbers, and changes to the design automatically translate into appropriate changes in signal types. Blocks are not just stand-ins for hardware. They respond to their surroundings, automatically adjusting the results they produce and the hardware they become.

System Generator allows designs to be composed from a variety of ingredients. Data flow models, traditional hardware design languages (VHDL, Verilog, and EDIF), and functions derived from the MATLAB programming language, can be used side-by-side, simulated together, and synthesized into working hardware. System Generator simulation results are bit and cycle-accurate. This means results seen in simulation exactly match the results that are seen in hardware. System Generator simulations are considerably faster than those from traditional HDL simulators, and results are easier to analyze.

C. Typical Design Flow for System Generator Software



IV. PROPOSED ARCHITECTURE SCHEME OF IMAGE PROCESSING ALGORITHM

A. *Data hiding framework*

Our proposed work is building up a framework through which data in the form of image file can be hidden in another image file using LSB coding and a simple encryption algorithm that will encrypt data at the sending end and a simple decryption algorithm that will decrypt data at the receiving end with the help of a key .

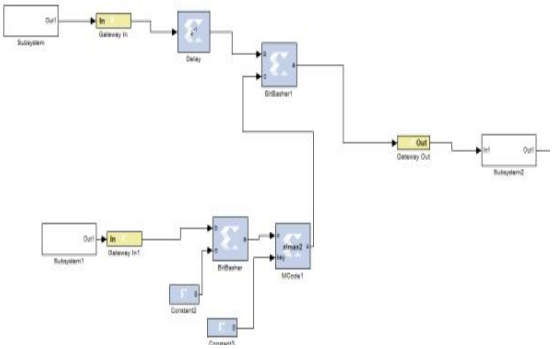


Fig.IV.a Invisible watermarking sending end architecture

The above architecture receives each individual pixels of the data to be hidden, extracts the msb bits and encrypts it using an encoder block with a key and embeds the encrypted bits into the least significant two bits of the carrier image

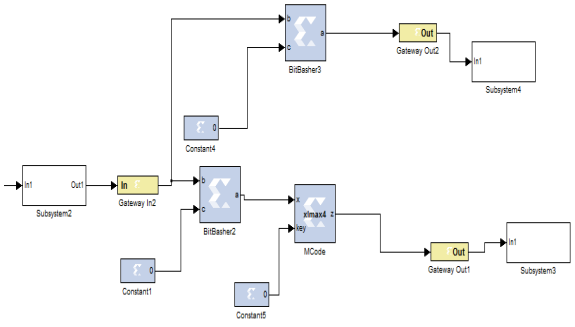


Fig. .IV.b Invisible watermarking receiving end architecture

At the receiver end the same key is used to decrypt using a decoder block and both the original carrier image and the hidden image are extracted.

B. *Visible watermarking for protection of ownership rights*

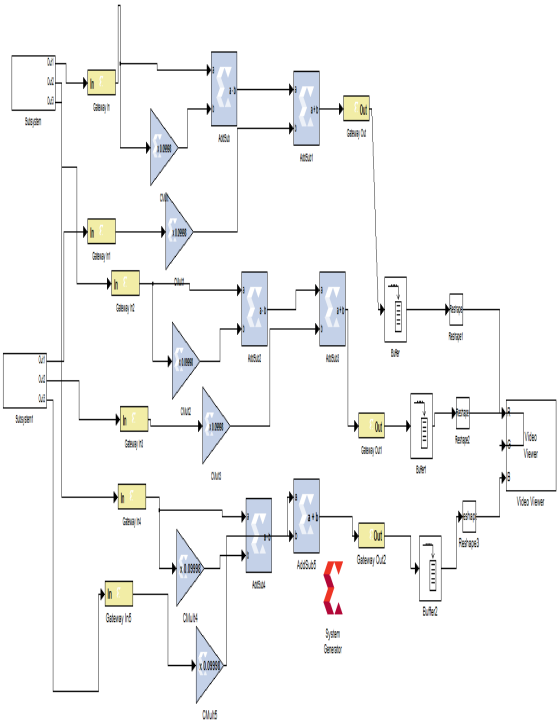


Fig. .IV.cVisible watermark architecture

This architecture is proposed to develop a visible watermarking framework of colored image using spatial domain watermarking algorithms for copyright protection. The above system generator block takes a color RGB image and a watermark image to produces a watermarked image using the following algorithm

$$fw = (1-\alpha)\, f + \alpha\, w$$

where f is the intensity value of the original image and w is the intensity value of watermark image. The constant α controls the relative visibility of the watermark. If α = 1, the watermark is opaque and the underlying image is completely obscured. As α approaches 0, more of the underlying image and less of the watermark is seen. In general: $0 < \alpha \leq 1$ In the previous example, α = 0.1. The underlying image is clearly visible through the "semitransparent" watermark.

C. *Image enhancement algorithm*

In this proposed architecture we take a cube of the image pixel values and then take the eight msb values to represent the resulting enhanced image which is much more crisp and sharp than the previous.

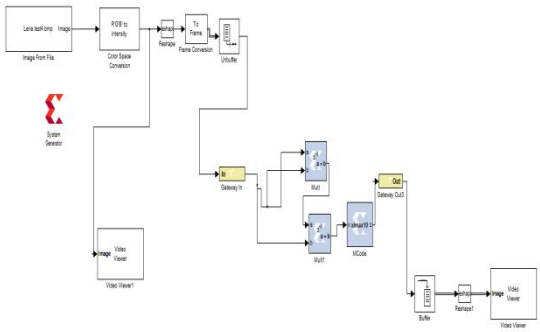


Fig. .IV.d architecture of image enhancement

RESULT AND DISCUSSIONS

A. *Data hiding framework*

Fig. V.a carrier image

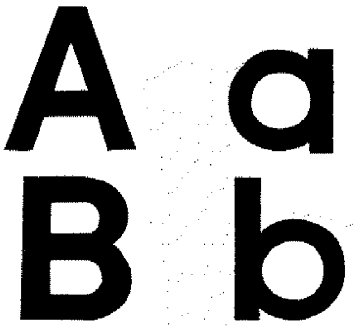


Fig. . V.b Data to be hidden

Fig.9 Invisible watermarked image

Fig. . V.c Extracted carrier image

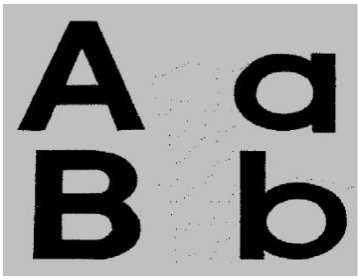


Fig. . V.d Extracted hidden data

B.Visible watermarking for protection of ownership rights

Fig. . V.e Color original image

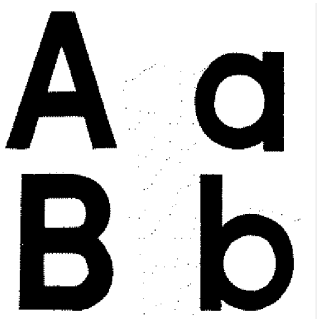


Fig. . V.f Watermark image

Fig. . V.g Watermarked image

C.Image enhancement algorithm

Fig. . V.h Test image

Fig. . V.i Enhanced image

FUTURE SCOPE & CONCLUSION

Image processing algorithms we have used thus far can be used to process a video stream to improve its quality and pass on hidden information in a video stream which certainly have important application in defence and internet applications.

Image processing, though dates back to year 1920, is a relatively young field. Space for explorations in this field is far reaching and applications are diverse. The protection of intellectual property through technical means was presumably one of the primary motivations for applying well-known watermarking techniquesIn general my work thus far deals with explorations of the application of image processing particularly in the field of hidden data transmission, copyright protection.

REFERENCES

1. IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 2, No 2, March 2012, A.C.Suthar1, Mohammed Vayada2, C.B.Patel3, G.R.Kulkarni4 1Research Scholar, Dept. of E.C.E., KSV University, Gandhinagar, Gujarat, India 1,3Department of E.C.E., L. C. Institute of Technology, GTU, Mehsana, Gujarat, India 2Department of E.C., PIET, Limda, Baroda, Gujarat, India Principal, CCET, Wadhwan, Gujarat, India
2. International Journal of Emerging Technology and Advanced Engineering Website: www.ijetae.com (ISSN 2250-2459, Volume 2, Issue 3, March 2012) Abhishek Basu1, Tirtha Sankar Das2, Subir Kumar Sarkar3 1&2RCC Institute of Information Technology, Kolkata-700015, West Bengal 3Jadavpur University, Kolkata-700032, West Bengal
3. Proceedings of the World Congress on Engineering 2009 Vol I WCE 2009, July 1 - 3, 2009, London, U.K. T. Saidani , D. Dia, W. Elhamzi, M. Atri and R. Tourki
4. IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 1, January 2011 193 ISSN (Online): 1694-0814 by Sudeep K C [1] and Dr. Jharna Majumdar [2]
5. IJRRAS 6 (1) ● January 2011, Hebah H.O. Nasereddin Middle East University, P.O. Box: 144378, Code 11814, Amman-Jordan
6. 2nd International Conference on Autonomous Robots and Agents December 13-15, 2004 Palmerston North, New Zealand by Abdullah AlMuhit, Md. Shabiul Islam and Masuri Othman
7. A FRACTAL WATERMARKING SCHEME FOR IMAGE IN DWT DOMAIN

★★★