FPGA Implementation of Watermarking Scheme Using XSG

Korrapati Rajitha¹, Usha Rani.Nelakuditi¹, Venkata Naresh Mandhala², and
Tai-hoon Kim³

¹*Department of Electronics and Communications Engineering,*

²*Department of Information Technology,*

^{1,2}*VFSTR University,*

^{1,2}*Vadlamudi-522213, Guntur, India*

³*Department of Convergence Security,*

Sungshin Women's University,

249-1, Dongseon-dong 3-ga,

Seoul, 136-742, Korea

¹*usharani.nsai@gmail.com,* ²*mvnaresh.mca@gmail.com,* ³*taihoonn@daum.net*
(Corresponding Author)³

Abstract

Now a day's hardware realization of image processing applications using FPGA is an important area of research due to the speed of implementation, low development costs and less time to market. This is due to the parallelism existing in it. In this paper an efficient architecture for Digital Image Watermarking algorithm using Xilinx System Generator (XSG) is proposed. The objective is, to simulate, synthesize and implement Image Watermarking algorithm on FPGA platform. Hardware implementation of watermarking is realized using the Xilinx Block Sets presented in Simulink. The generated hardware software co-simulation block for Virtex6 FPGA, VHDL/Verilog code and the test-benches are tested on the FPGA development board. Then it is validated in terms of area, power and performance metrics.

Keywords: Watermarking, Xilinx System Generator, FPGA, Simulink, MatLab

1. Introduction

The digital image processing has become an important subject of interest in many different areas such as medical, technological etc. This paper deals with the digital watermarking algorithm for digital images is implemented in hardware FPGA. Now- a- days, Application Specific Integrated Circuits (ASIC) and programmable DSP processors were the other implementation choices for many digital applications. Development cost, time, flexibility, programmable software and the functional efficiency parameters enable us FPGA is considered as the implementation platform. Reconfigurable computing is also being considered as primary concern. FPGA is used for the system implementation in the hardware. Hence FPGAs are an attractive choice due to their low power dissipation per unit computation, high performance and reconfigurability [1]. The parallel computing feature of the FPGA is extremely useful to support the needs of the modern world applications in the areas like DSP, image and video processing etc. To create custom DSP data paths in FPGA System Generator [1-2] is used as a high level well suited design tool. This objective lead to the use of Xilinx System Generator tool with a high level graphical interface *i.e.* Simulink, based on blocks which makes it very easy to handle with respect to other softwares for the hardware design and implementation.

Now-a-days handling of digital data on internets and in multimedia applications is a critical issue. Digital watermarking is applied for copyright protection, content authentication, detection of illegal duplication and alteration, feature tagging and secret communication. Digital Watermarking is the hiding of a secret message within an ordinary message and its extraction at its destination. The secret message embedded as watermark can be anything like plaintext, image *etc.* In general digital watermarking algorithm involves two major operations. 1. Watermark embedding, and 2. Watermark extraction. For both the operations a secret key is needed to secure the watermark [3].

This paper is concentrated on developing algorithmic models in simulink using Xilinx Block set for watermarking algorithm and then hardware implementation on FPGA. It is organized as follows. Section1 explains the importance of hardware implementation of digital images. Section2 is deals with the XSG, Section 3 focuses on hardware implementation and the results are discussed in Section 4.

2. Xilinx System Generator

Xilinx System Generator (XSG) [5] is an Integrated Design Environment (IDE) for FPGAs within the ISE 13.4 development suite, which uses Simulink [5], as a development environment and is presented in the form of model based design. It has an integrated design flow generates the bit stream file (*. bit) from Simulink design environment which is necessary for the programming of FPGA.

XSG has abstraction level operating with fixed point double precision including quantization and overflow characteristics. In contrast Simulink works with numbers of double-precision floating point. The connection between XSG blocks and Simulink blocks are the gateway blocks. XSG automatically generates simulation results, RTL synthesis, VHDL/Verilog code, User Constraint File (UCF) and mapping hardware. It was created primarily to deal with complex Digital Signal Processing (DSP) applications but now it is intensively used for the implementation of many image processing applications.

3. Watermarking

Watermarking algorithm consists of two phases, embedding and extraction. Embedding is like an encoder in hardware applications, where as extraction algorithm resembles a decoder.

3.1. Embedding/Extraction Algorithm

Original image (cover) and key image are considered and key image is embedded into the functional part of the cover data so that it results in watermarked image as shown in Figure 1.

Embedding process :

Figure 1. Embedding of Watermark

Extraction process is performed on watermarked image to extract the original and key images as shown in Figure 2.

Figure 2. Extraction of Cover Image and Watermark

3.2. Implementation in Simulink

Watermarking is implemented using Simulink blocks and Xilinx block set. It is considered in 3 steps.

1. Pre-Processing
2. Embedding/ Extraction as shown in Figure 3 and Figure 4 respectively.
3. Post-Processing

Pre-processing covers conversion of 2-D gray or 3-D color images into 1-D array as per the requirements of target hardware. Embedding/Extraction is nothing but either watermark into the cover image or extraction of key and original on the other side. Post-processing is conversion of 1-D to 2-D/3-D.

Figure 3.1. Subsystem Block for Embedding

Figure 3.2. Watermark Embedding

Embedding Algorithm

Two signals host and key are fed into the logical adder block as input. Then resulting sum and carry out from adder1 are fed to shift block it performs division by 2. The scaling of the image pixel gray values is performed in multiplier by multiplying host image with high order bits. The higher order bits of multiplier are fed to the adder/subtract unit to perform embedding block. Only high order bits are considered low order bits are discarded as shown in Figure 3.

Extraction Algorithm

Watermarked image used as input to the extraction algorithm Subsystem block for extraction is as shown in Figure 3 to extract the original host and key image. The output of the XOR operation signal is to perform the thresholding and filtering operation to remove the unwanted noises. The subsystem of extraction process is shown in Figure 4.

Figure 4. Subsystem Block for Extraction

3.3. H/W S/W Co-Simulation

Embedding/Extraction is modeled hardware is generated through system generator, target board Virtex 6 LX240T and JTAG cable. Now it can be verified using hw as well as software. A model can be co-simulated provided it meets the requirements of the underlying hardware board. It generates the JTAG co-simulation block as shown in Figure 5. By using this we can verify the output on the PC or on separate hardware VGS.

Figure 5. System Generator Token

Figure 6. Co-Simulation for Embedded Algorithm

After completing the Co-simulation, design is synthesized and implemented in Xilinx ISE 13.4. The resultant RTL schematic for Embedded/Extraction phases of watermarking algorithm as shown in Figure 7 and Figure 8 respectively.

Figure 7. RTL Schematic for Embedded Algorithm

Figure 8. RTL Schematic for Extraction Algorithm

9. Results

The watermarking algorithm is implemented in 2 phases. The output of the both the embedding and extraction phases are obtained. After completing the hardware-software co-

simulation the resultant outputs are as shown in Figure 9 and Figure10 respectively. It is observed that distortion with the watermarked or extracted images are very less.

Figure 9. Output of Embedding Algorithm

Figure 10. Output of Extraction Algorithm

The resource utilization and power-delay performance of these algorithms are shown in Tables 1 and 2.

Table 1. Resource Utilization Report for Embedding/Extraction Algorithm

Name of Component	Resources Used		Device Usage	
	Embedding	Extraction	Embedding	Extraction
FLIP-FLOPS	156	198	1%	1%
LUTs	42	149	1%	1%
SLICES	16	71	1%	1%
IOBs	50	67	8%	11%

Table 2. Power and Performance Report

Algorithm	Delay(ns)	Power(mW)
Embedding	0.526	1.7
Extraction	0.785	5
Total	1.311	6.7

5. Conclusion

The Watermarking Algorithm is implemented on Virtex-6 target board with the help of Xilinx system Generator and Simulink IDE. It is observed that the proposed algorithm takes 1311ps and consumes 6.7mw power. It uses 156 ffs, 46 LUTs and 50 IOBs. The development time is less. It is less complex and highly flexible for prototyping and modifications. It is implemented with 40nm copper CMOS process technology. It is implemented with MatLab13.2 version Xilinx vi14. It uses less area of 2% of device space. The performance and accuracy of implemented watermarking algorithm is very high as shown in results

table.1. It is easier in implementation and prototyping many applications rather than DSP hardware or PC with Co-Processor.

References

- [1] Mustafa Osman Ali and Rameshwar Rao, "Digital Image Watermarking Basics, and Hardware Implementation", International Journal of Modeling and Optimization, vol. 2, no. 1, **(2012)** February, pp. 19-24.
- [2] S. Gangopadhyay and B. Banerjee, "Image processing and data hiding frame work on FPGA based platform", International journal of electrical, electronics and data communication, vol. 1, no. 1, **(2013)** May.
- [3] P. V. Powar and S. S. Agrawal, "Design of Digital Video Watermarking Scheme Using Matlab Simulink", IJ.RET, vol. 2, no. 5, **(2013)** May, pp. 630-634.
- [4] V. Elamaram and G. Rajkumar, "FPGA Implementation of point processes using Xilinx System Generator", Journal of Theoretical and Applied Information Technology, vol. 41, no. 2, **(2012)** July, pp. 201-206.
- [5] S. Allin Christe, M. Vignesh and A. Kandaswamy, "An Efficient FPGA Implementation of MRI Image Filtering and Tumor Characterization Using Xilinx System Generator", International Journal of VLSI design & Communication Systems (VLSICS), vol. 2, no. 4, **(2011)** December, pp. 95-109 .
- [6] Xilinx System Generator User guide 13.4.
- [7] T. Mahalakshmi and R. Muthaiah, "VLSI Implementation of Edge Detection for Images", Research Journal of Applied Sciences, Engineering and Technology, **(2012)** December, pp. 5474-5479.
- [8] P. Karthigaikumar, K. Jaraline Kirubavathy and K. Baskaran, "FPGA based audio watermarking Covert communication", Microelectronics Journal, vol. 42, no. 5, **(2011)** May, pp. 778-784.
- [9] A. C. Suthar, M. Vayada, C. B. Patel and G. R. Kulkarni, "Hardware Software Co-Simulation for Image Processing Applications", IJCSI International Journal of Computer Science, vol. 9, no. 2, **(2012)** March, pp. 560-562.
- [10] T. Saidani, D. Dia, W. Elhamzi, M. Atri and R. Tourki, "Hardware Co-simulation For Video Processing Using Xilinx System Generator", Proceedings of the World Congress on Engineering (WCE), vol. 1, **(2009)** July, pp. 852-859.
- [11] N. V. Dharwadkar and B. B. Amberker, "Watermarking Scheme for Color Images using Wavelet Transform based Texture Properties and Secret Sharing", International Journal of Information and Communication Engineering , vol. 6, no. 2, **(2010)**, pp. 93-100.
- [12] I. J. Cox, M. L. Miller, J. A. Bloom, J. Fridrich and T. Kalker, "Digital Watermarking and Steganography", Second edition, Morgan Kaufmann Publishers, **(2008)**.