Available online at www.sciencedirect.com

ScienceDirect

Procedia Engineering 00 (2011) 000–000

Procedia Engineering

www.elsevier.com/locate/procedia

The International Conference on Modeling, Optimization and Computing

(ICMOC 2012)

Hardware Software Co-Design Simulation Modeling For Image Security Concept Using Matlab-Simulink with Xilinx System Generator

Mohammed Vayada[1], A.C. Suthar[2], A.R.Yadav[3,] Dr. G.R.Kulkarni[4]a*

[1]*P.G.Student ,ECE, PIET,Baroda,Gujarat,India*
[2]*Research Scholar, KSV University, Gandhinagar, Gujarat, India & Asst. Prof, ECE, LCIT, Bhandu, Gujarat, India*
[3]*Asst. Prof, ECE, PIET, Limda, Baroda, Gujarat, India*
[3]*Principal, CCET, Wadhwan, Gujarat, India*

Abstract

Nowadays hardware design has become a very complex task because programmable hardware, such as FPGA's, becomes increasingly complex with regard to the number of transistors. In this paper, we will give an overview of some important tools for model based hardware design. Instead of starting from a complex, low-level VHDL or Verilog description, these tools enable to describe an algorithm in a model-based description. They even promise a very automatic translation of this model-based description to HDL. These high-level tools bring hardware design (based on FPGA's) to a wider audience, while increasing designer productivity. In this paper we develop a image security model using matlab-simulink with Xilinx system generator, it use for secure the image, i.e. our secrete image behind of one original image. By observing, everyone can see the original image but behind of that one other image which like a invisible image..

Keywords: Colour Imag; Gray Scale Image; Image Security; Spatial Domain;Xilinx System Generator;

* Corresponding author. Tel.: +0-94281-92520;
vayada_mohammed@yahoo.com, acsuthar@yahoo.co.in, aryme@rediffmail.com,grkulkarni29264@rediffmail.com

Nomenclature

D	Dimension
DSP	Digital Signal Processing
FPGA	Field Programmable Gate Array
HDL	Hardware Discretion Language
PSNR	Peak Signal to Noise Ration
UCF	User Constraint File
XSG	Xilinx System Generator

1. Introduction

The complexity of integrated circuits increases every year. As stated by Moore's law, the number of transistors on integrated circuits doubles every Two year, as shown in figure 1. This offers the designers the possibility to create faster, larger and more complex designs using less area. Because the productivity of design tools increases with a lower rate, a design productivity gap arises. Implementing complex algorithms using low Level hardware languages like VHDL or Verilog, results in a lot of low level Code. Simulation times can become large. To reduce the complexity of the description and the simulation time, a higher level of abstraction needed. Model-based hardware design is a technique in which abstraction used to Control the complexity of new designs.

Several model-based design tools enter the market today and the way each of them implement this new level of abstraction is tool-specific. Some tools Use a graphical interconnection of blocks while other tools limit the length of handwritten code by using a different programming syntax with a higher Level of abstraction. Once the model described and simulated, automatic code generation applied. On one button click, these tools generate HDL code based on the hardware model description. The length, quality And readability of this code are all aspects that will have to be taken in Account when choosing the proper code-generation tool for a specific task.

The handling of digital images has become in recent decades a subject of widespread interest in different areas such as medical and technological application, among others. We may cite many examples where image processing helps to analyze infer and make decision. The main objective of image processing is to improve the quality of the images for human interpretation or the perception of the machines independent of the images for human interpretation or the perception of the machines independently. This paper focuses to embed the information image to the host image using the spatial domain method. The need to process the image in real time, leading to the implementation level hardware, which offers parallelism, thus significantly reduces the processing time, which was why decided to use Xilinx System Generator, a tool with graphical interface under the Matlab-Simulink, based blocks which makes it very easy to handle with respect to other software for hardware description. In addition to offering all the tools for easy graphical simulation level, This article presents architecture of image processing application generator, which is an extension of Simulink and consists of a bookstore called "Blocks Xilinx", which are mapped architectures, entities, signs, ports and attributes, which script file to produce synthesis in FPGAs, HDL simulation and development tools. The tool retains the hierarchy of Simulink when it is converts into VHDL and code can be finally implement using XSG (Xilinx System Generator) on FPGA [3].

2. Matlab with Xilinx System Generator

Xilinx System Generator (XSG) is an integrator design environment (IDE) for FPGAs, which uses Simulink, as a development environment and is presents 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.

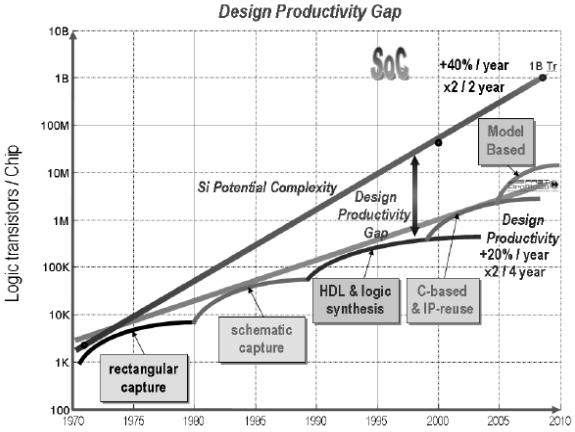


Fig. 1. Design Productivity Gap and Moore's law

We 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 [3,6].

In Fig.2 shows the broad flow design Xilinx System Generator. As already mentioned, you can then move to the configuration file to program the FPGA.

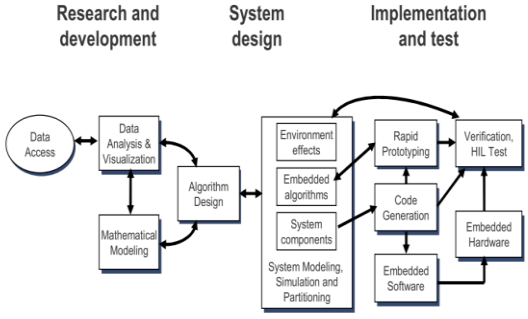


Fig. 2. Xilinx System Generator Design Flow

3. Basic Model Based Design

The basic model based design includes the processes of Executable specification for model, Design with simulation, implementation of automatic code generation and continuous test and verification. We can get the idea with the help of fig.3.

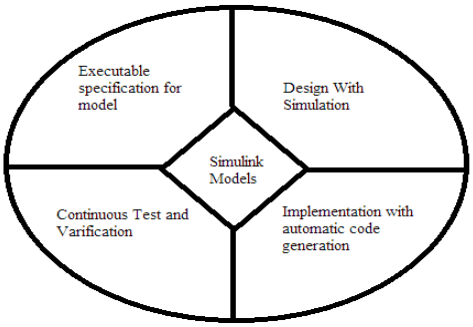


Fig. 3. Model Based Design using Simulink

The value of model-based design includes Innovation, Quality, Cost, and Time to market [6].

4. Basic Image Operating using Xilinx System Generator

The design of reading image is shows in the Fig.4. In this model Xilinx Gateway In/Out blocks expects input image which must be 1-D vector. We use the image read from the memory and processed in Convert 2-D to 1-D, which converts M * N data to MN*1 1-D output, then frame conversion by simply output the stream in 1-D to input of Gateway IN, unbuffer block of Simulink can convert frame based input to sampled based output with frame size = 1. After the Gateway Out, the type conversion will convert uint8, and then buffer the data and the reshape, which is a use to reshape the data from 1-D to 2-D.

The processing time required to simulate this design depends upon the number of pixels operation involved in image. Here Xilinx Gateway In and Out blocks are called Xilinx FPGA Boundaries which generally makes interface of the FPGA environment with Simulink. In a Simulink the input and output can be seen using Simulink block video viewer. The system generator block is add in order to have VHDL code, which can be synthesized in Xilinx ISE design suite [3].

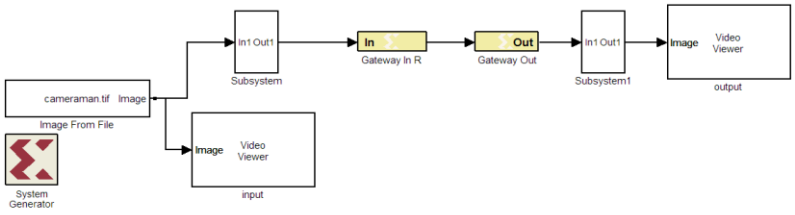


Fig. 4. Reading Image in Simulink with Xilinx System Generator

The processing time required to simulate this design depends upon the number of pixels operation involved in image. Here Xilinx Gateway In and Out blocks are called Xilinx FPGA Boundaries which generally makes interface of the FPGA environment with Simulink. In a Simulink the input and output

can be seen using Simulink block video viewer. The system generator blocks have added in order to have VHDL code, which is synthesize in Xilinx ISE design suite [3].

5. Model Based Gray Scale Image Security

The Model based design is build-using Simulink and Xilinx block sets. The image can easily combine by having right shift or left shift of one of the image, the result is then bit shifting with another image, and PSNR of the input image and output image is calculate to check its concealing property. For the different right and left shift we can have different PSNR values.

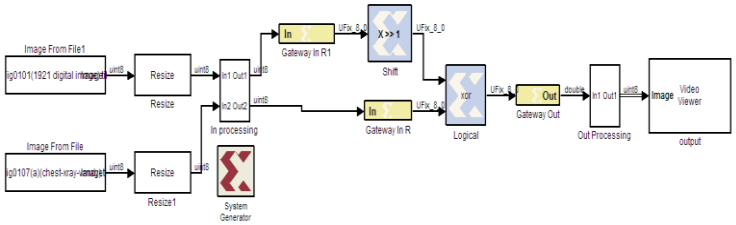
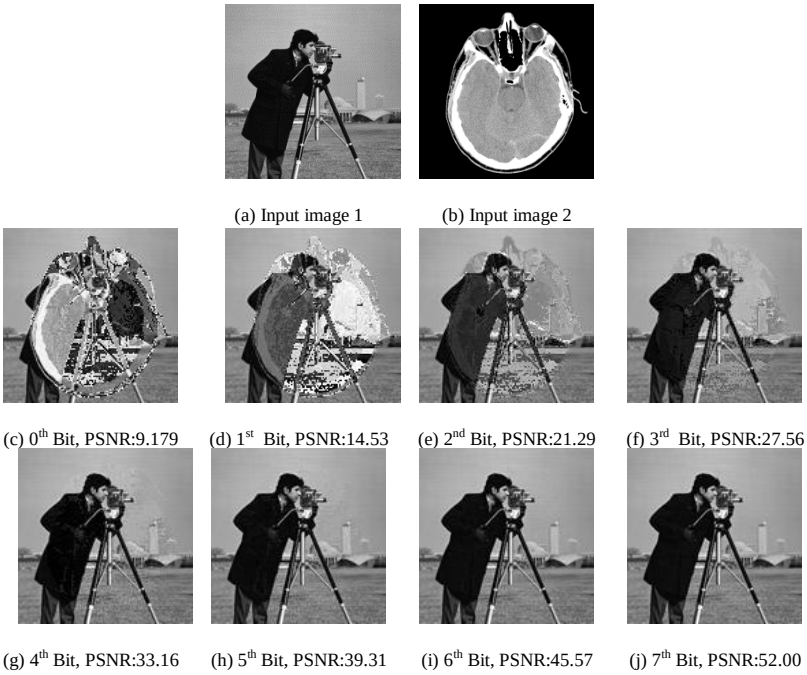


Fig. 5 Model of Gray Scale Image Security

Fig. 5 shows the model of secure of image, as we have used shift and logical Xilinx blocks that help the image to shift in either direction and is logically operates with logical Xilinx block set. With the right and left bit shift, we got different PSNR values. Further shifting bit 1 to 7 gave different PSNR values.

The image taken as two inputs and their result for the different right shifts of bits is shows in Fig.6(a) to (j).



(a) Input image 1 (b) Input image 2

(c) 0^{th} Bit, PSNR:9.179 (d) 1^{st} Bit, PSNR:14.53 (e) 2^{nd} Bit, PSNR:21.29 (f) 3^{rd} Bit, PSNR:27.56

(g) 4^{th} Bit, PSNR:33.16 (h) 5^{th} Bit, PSNR:39.31 (i) 6^{th} Bit, PSNR:45.57 (j) 7^{th} Bit, PSNR:52.00

Fig. 6 Secure Image with respective number of bits and its PSNR

6. Model Based Colour Image Security

The colour image secure design model is shows in Fig.7.

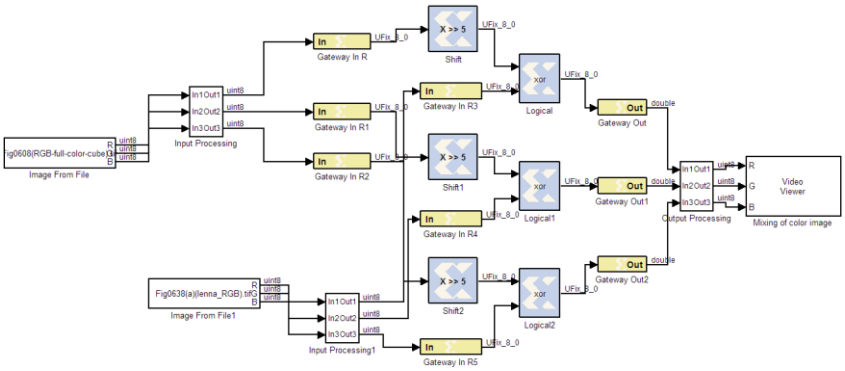
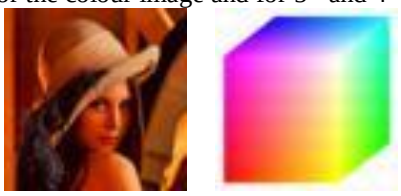


Fig. 7. Colour Image Secure Model

In this design, the three components of colour image of both inputs are processed separately and mixed individually with respective R, G, B components of another image.

The different bit shift mixing output are shown in fig.8, from figure easily observed that, the 6th bit right shift gives the better PSNR for the colour image and for 5th and 4th bit, PSNR is relatively low.



(a) Input image 1 (b) Input image 2

(c) 4th Bit,PSNR:29.77 (d) 5th Bit, PSNR: 34.40 (e) 6th Bit, PSNR:43.00

Fig. 8 Colour image Mixing outputs with respective PSNR

The above designs for the image mixing for colour as well as gray scale image can be implement on FPGA Devices. As the Xilinx system generator is a tool, required for converting this model in to the compact VHDL code. This VHDL code can individually be optimize by neglecting certain parameters; this will reduce the processing time and increase the code performance. After synthesizing the code in Xilinx ISE domain and creating programming file, the code can be downloaded to one of the FPGA Board and the results can be verified [5].

The FPGA resources Estimator for both the designs are shows in Table 1.

Table 1. Resource Estimator for Colour and Grayscale Image Models

Resource	Slices	BRAMs	IOBs
Gray Scale Image Model	8	8	24
Colour Image Model	24	24	72

The resource estimator shows the total no of slices, I/O Blocks and BRAMs required for both the models.

7. Conclusion

The Xilinx System Generator tool is a new application in image processing, and offers a friendly Design environment for the processing, because processing units are designed by using blocks. This tool supports software simulation, but the most important part is that it can be synthesize on FPGAs hardware, with the parallelism, robustness and speed, these features are essentials in image processing. From this work, we determine the peak signal to noise ratio, which is a parameter for observing the quality of image. Using the spatial domain combine two images one image is visible, other is invisible, and the peak signal to noise ratio depends upon the bit position of particular image. From the observation knows the right shift greater than third bit in gray scale image model shows better PSNR, which is more than 30. In colour images, mixing models right shift greater than fourth gives better PSNR. The same model implemented in FPGA devices like Xilinx Spartan 3E using Xilinx System Generator (XSG).

Acknowledgements

Authors are very much thankful to KSV University, Parul Institute and LC Institute of Technology for proving a good platform for doing the research work.

References

[1] R. Gonzalaz, R. Woods. Digital Image Processing. New Jersey: *Prentice-Hall*; 2002.

[2] DSP System Generator User guide release 10.1, March 2008.

[3] Xilinx System Generator, User's Guide, www.xilinx.com.

[4] White paper: *Using System Generator for Systematic HDL Design, Verification and Validation*.WP283 (v1.0) January 17, 2008.

[5] J.C.Moctezuma, S.Sanchez, R.Alvarez, A. Sánchez. *Architecture for filtering images using Xilinx system generator*" World Scientific Advanced Series In Electrical And Computer Engineering, Proceedings of the 2nd WSEAS International Conference on Computer Engineering and Applications; 2008, pp. 284-289.

[6] Sofia Mosesson. *Using MATLAB® and Simulink® for Image and Video Processing*, Application Engineer; 2007.