Journal of Theoretical and Applied Information Technology
31st July 2012. Vol. 41 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

FPGA IMPLEMENTATION OF POINT PROCESSES USING XILINX SYSTEM GENERATOR

[1] V.ELAMARAN, [2] G.RAJKUMAR

[1]Asst Prof., Department of ECE, School of EEE, SASTRA University, Thanjavur

[2]Asst Prof., Department of ECE, School of EEE, SASTRA University, Thanjavur

E-mail: [1]elamaran@ece.sastra.edu, [2]grkumar@ece.sastra.edu

ABSTRACT

Application areas of signal processing have grown dramatically in importance in recent times, in parallel with the growth of powerful and low-cost processing chips. This has led, in turn, to many new applications, including multimedia delivery and hand-held communications delivery. Image processing is one an important application among them, which has a strong mathematical basis. Specifically authors demonstrate point processes which use only the information in individual pixels to produce new images. Arithmetic operations, XOR operations, histograms, contrast stretching and intensity transformations are implemented using Xilinx System Generator (XSG). XSG is a useful tool to understand fundamental Digital Signal Processing (DSP) algorithms for Field Programmable Gate Array (FPGA) implementation. FPGAs provide a better platform for real-time algorithms on application-specific hardware with substantially greater performance than programmable DSPs. The study reveals that the hardware implementation results which are presented here will be extended to perform hardware-in-the-loop simulation.

Keywords: *Image Processing, Point Processing, Xilinx System Generator, FPGA.*

1. INTRODUCTION

The application domain of DSP over the past decade expanded because of the advance in VLSI technology. Application Specific Integrated Circuits (ASIC) and programmable DSP processors were the implementation choices for many DSP applications. But now, reconfigurable computing are being considered for system implementations because of the programmable of software and the functional efficiency of hardware. FPGAs are an attractive choice due to their low energy dissipation per unit computation, high performance, and reconfigurability. The parallel computing power of the FPGA is extremely useful in the modern world of demanding applications 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.

In today's modern computers, media information such as audio, images, and video have come to be necessary for daily business operations and entertainment. In this paper, we study digital images and its processing techniques, specifically point processing algorithms. Digital images are electronics snapshots taken of a scene or scanned from documents, such as photographs, manuscripts, printed texts, and artwork. The digital image is sampled and mapped as a grid of dots or picture elements (pixels). The digital image is picture information in digital form. The image can be filtered to remove noise and obtain enhancement [3]. It can also be transformed to extract features for pattern recognition. The image can be compressed for storage and retrieval, as well as transmitted via a computer network or a communication system. Digital image processing has found application in wide variety of fields of human endeavor. There are number of well defined processes which go to make up a typical image application. Acquisition, Enhancement, Restoration, Segmentation and Analysis are the steps needed by just about every application which involves image processing [4].

Once images are inside the computer system, or more specifically, once they are read inside a program, the images are nothing but matrices.

Journal of Theoretical and Applied Information Technology
31st July 2012. Vol. 41 No.2
© 2005 - 2012 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Hence, all the operations that can be applied to matrices should theoretically be applicable to the images as well. Image arithmetic is the implementation of standard arithmetic operations, such as addition, subtraction, multiplication, and division for images. Image arithmetic has many uses in image processing, both as a preliminary step in more complex operations and by itself [5].

DSP functions are implemented on two primary platforms such as Digital Signal Processors (DSPs) and FPGAs [6]. FPGA is a form of highly configurable hardware while DSPs are specialized form of microprocessors. Most engineers prefer FPGA over DSP because of massive parallel processing capabilities inherent to FPGA and time to market make it the better choice. Since FPGAs can be configured in hardware, FPGAs offer complete hardware customization while implementing various DSP applications.

System Generator [7] is a DSP design tool from Xilinx that enables the use of the Mathworks model-based design environment Simulink for FPGA design. It is a system level modeling tool in which designs are captured in the DSP friendly Simulink modeling environment using Xilinx specific Blockset. All of the downstream FPGA implementation steps including synthesis and place and route are automatically performed to generate an FPGA programming file. System Generator provides many features such as System Resource Estimation to take full advantage of the FPGA resources, Hardware Co-Simulation [8] and accelerated simulation through hardware in the loop co-simulation; which give many orders of simulation performance increase [9,10,11].

This paper is organized as follows. Section 2 focuses on the fundamentals of all point processes in image processing and section 3 shows the simulation results using Xilinx System Generator for the models designed. Conclusions are presented in the section 4.

2. POINT PROCESSES

Point processes are the simplest and basic image processing operations. They operate on a pixel bases solely on that pixel’s value. Although point operations are the simplest, they contain some of the most powerful and widely used of all image processing operations. They are especially useful in image pre-processing, where an image is required to be modified before the man job is attempted.

This section contains an important point processing operations such as arithmetic operations, XOR operations, histograms with equalization, contrast stretching and intensity transformations along with the implementations which are done using XSG.

2.1 Arithmetic Operations

The arithmetic operations include adding, subtracting, dividing, and multiplying pixels by a constant value. Addition and subtraction can adjust the brightness of the image [9]. Fig. 1. shows the XSG blocks involved while adding and subtracting 40 from the image. Fig. 2. shows the results of an image which has a size of 512 x 512.

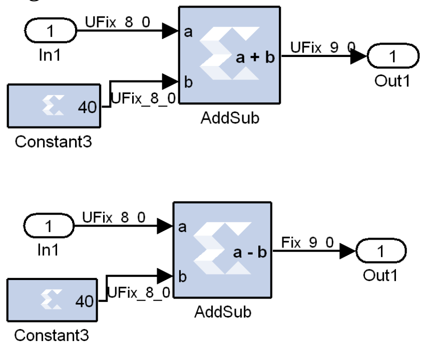


Figure 1: *XSG blocks for addition and subtraction.*

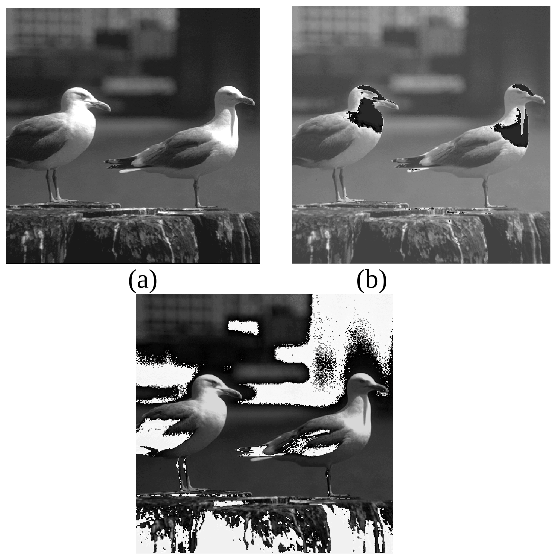


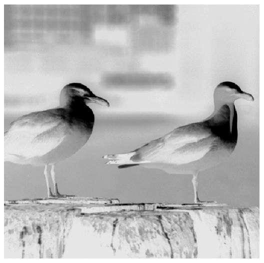
Figure 2: *(a) Original 512x512 image (b) image+40 (c) image -40.*

Similarly, multiplication and division by different values can adjust the contrast of the image.

Journal of Theoretical and Applied Information Technology
31st July 2012. Vol. 41 No.2
© 2005 - 2012 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2.2 XOR Operations

The Exclusive OR function sets bits that are the same in each operand to 0 and bits that are different to 1. All pixels of a certain value can be found by applying XOR function. In computers, a cursor for the mouse can be generated by applying XOR function frequently used on graphic systems. Almost a negative image produced by XOR with 255 to the image. Fig. 3. shows the results of an image with XSG block.



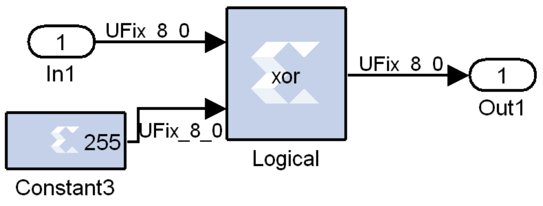


Figure 3*: Image XORed with 255 and XSG blocks.*

2.3 Histogram Stretching (Contrast Stretching)

The contrast of an image is its distribution of light and dark pixels. To stretch a histogram, contrast stretching is applied to an image to fill the full dynamic range of the image. We can stretch out the gray levels in the center of the range by applying piecewise linear function according to the equation.

new pixel = (12/4) (old pixel-5) + 2 (1)

where new pixel is its result after the transformation. Fig. 4. and Fig. 5. shows the XSG blocks for the above contrast stretching to the finger print image and the results respectively.

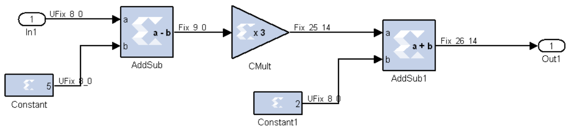


Figure 4*: XSG blocks for contrast stretching.*

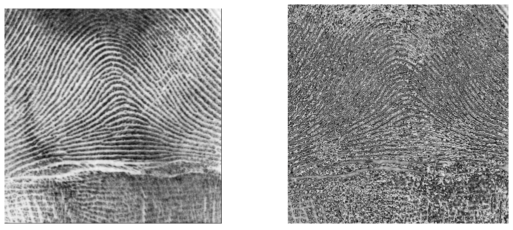


Figure 5*: (a) Original finger print image (b) image after contrast stretching.*

We demonstrate an another piecewise linear function which is as follows:

j = ((255-193) / (255-160)) (i-160) + 192 (2)

where i is the original gray level and j is its result after the transformation. Fig. 6. and Fig. 7. shows the XSG blocks for the above contrast stretching to the finger print image and the results respectively.

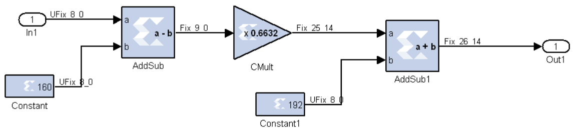


Figure 6*: XSG blocks for contrast stretching*

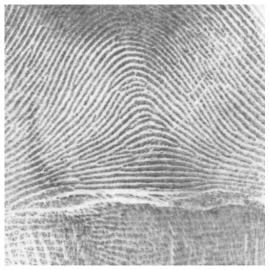


Figure 7*: Image after transformation.*

2.4 Intensity Transformations

Intensity transformations are point processes that convert an old pixel into a new pixel based on some predefined function. These transformations are easily implemented with simple look-up tables.

2.4.1 Negative Transformation

The negative transform exchanges dark values for light values and vice versa. This is the complement of a grayscale image like a photographic negative. The equation is as follows :

new pixel = 255 – old pixel (3)

Fig. 8. and Fig. 9. shows the XSG blocks for the above negative transformation to the finger print image and the results respectively.

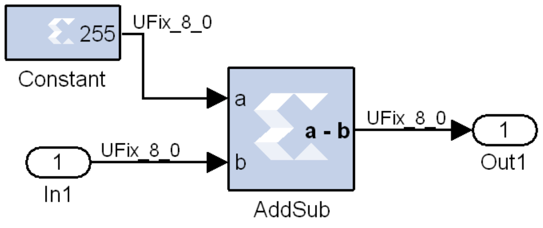


***Figure 8**: XSG blocks for negative transformation.*

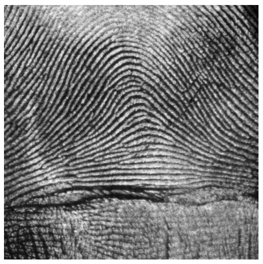


***Figure 9**: Negative image.*

2.4.1 Image Segmentation using Threshold

Image segmentation can be used to separate pixels associated with objects of interest from the image background. This is an important step in many imaging applications of automated analysis and robotics. We demonstrate segmentation on a simple pixel-by-pixel basis using threshold decisions. We use histogram of a image to determine the threshold value . Fig. 10. shows the original cameraman image and histogram result Histogram shows that the object and background are well-separated. We use Mcode block and use 70 as a threshold in our demonstration. Fig. 11. and Fig. 12. shows the matlab code for Mcode block and the image after segmentation results respectively.

Here we get the foreground portion as a white image.

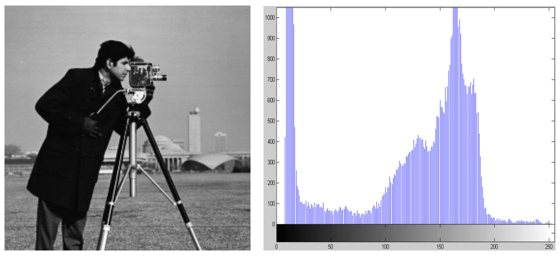


***Figure 10**: Image and histogram.*

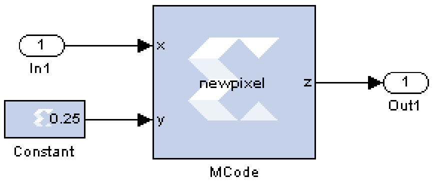


***Figure 11**: XSG blocks for segmentation.*

The Mcode block description is as follows :

```
function z = newpixel(x,y)
    if x>y
        z = x;
   else
        z = 1;
end
```

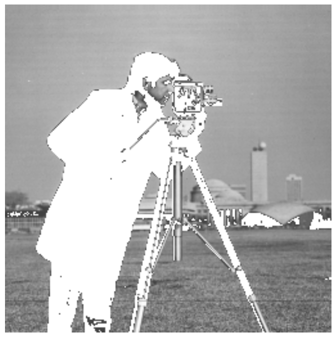
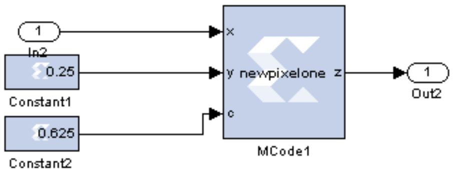


***Figure 12**: Image after segmentation.*

2.4.3 Range highlighting Transformation

An intensity transform can also highlight a range of pixels while keeping others constant. Fig.13. shows the Xilinx blocks implementation and the resulting image.

```
function z = newpixelone(x,y,c)
    if (x > y) & (x < c)
        z = x;
    else
        z = 1;
end
```



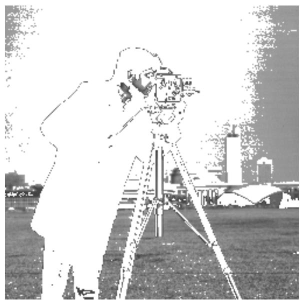


***Figure 13**: Xilinxs blocks and resulting image.*

2.4.4 Parabola Transformation

The two formulas for the parabola transformation are as follows:

$$\text{new pixel} = 255 - 255\,((\text{old pixel}/128) - 1)^2 \quad (4)$$

and

$$\text{new pixel} = 255\,((\text{old pixel}/128) - 1)^2 \quad (5)$$

Xilinx blocks are connected for the above equations and displayed in Fig. 14. and Fig. 15. Both the results are observed and produced in Fig. 16. Similarly, solarize transformation, iso-intensity contouring transformation and bit-clipping transformation can also be implemented [8].

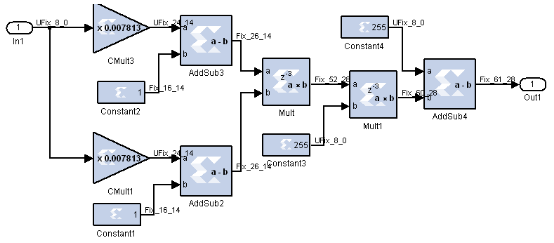


***Figure 14**: XSG blocks for Equation (4).*

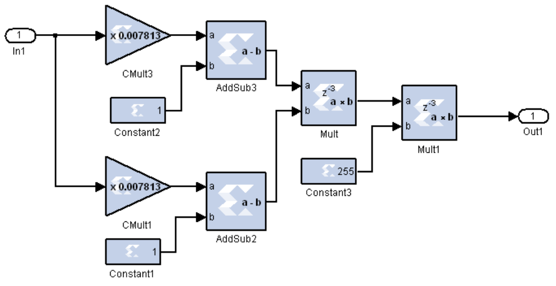


***Figure 15**: XSG blocks for Equation (5).*

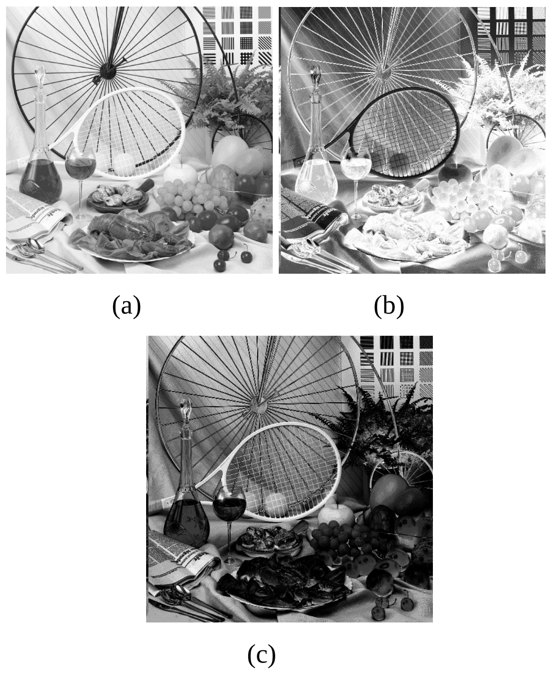
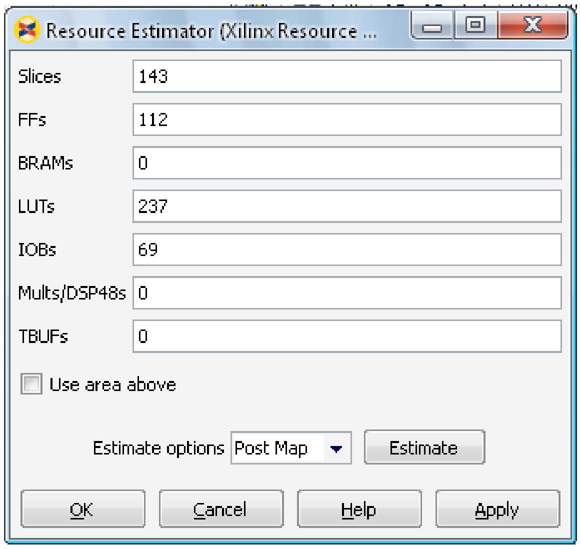


***Figure 16**: (a) Original image (b) Parabola – Equation (4) (c) Parabola – Equation (5).*

3. SIMULATION RESULTS

The hardware implementation results are produced using Xilinx Spartan 3E FPGA. The resource utilization summary is obtained for the parabola transformation (Fig.16 and Fig.17) results of Equation 4 and 5 respectively in Fig.17 and Fig.18.



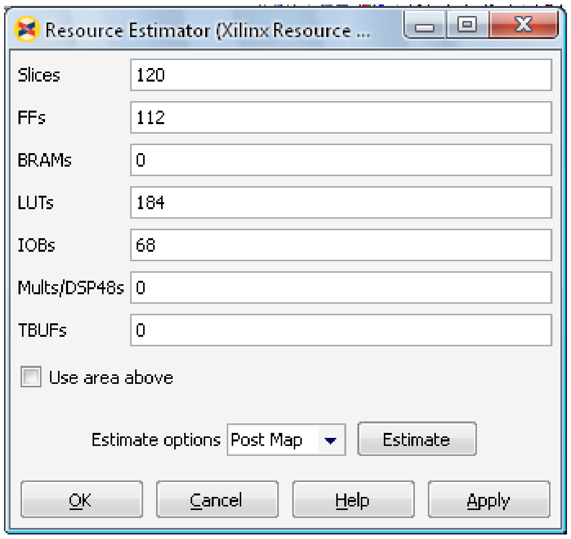
Resource Estimator (Xilinx Resource ...

Slices	143
FFs	112
BRAMs	0
LUTs	237
IOBs	69
Mults/DSP48s	0
TBUFs	0

☐ Use area above

Estimate options: Post Map

***Figure 17**: XSG blocks for Equation (5).*


Resource Estimator (Xilinx Resource ...

Slices	120
FFs	112
BRAMs	0
LUTs	184
IOBs	68
Mults/DSP48s	0
TBUFs	0

Use area above

Estimate options Post Map

Figure 18: *XSG blocks for Equation (5).*

4. CONCLUSIONS

In this paper, a real-time image processing algorithms are implemented on FPGA. Implementation of these algorithms on a FPGA is having advantage of using large memory and embedded multipliers. Advances in FPGA technology with the development of sophisticated and efficient tools for modeling, simulation and synthesis have made FPGA a highly useful platform [7].

The study reveals that implementation of point processing algorithms using Xilinx System Generator can be extended to applications like background estimation in video, image filtering both in spatial and frequency domains and digital image watermarking applications, etc. The implementation further can be extended to area processes and frame processes in the field of digital image processing. The above experiments may also be performed with hardware-in-the loop verification and co-simulation approaches.

REFRENCES

[1] Ana Toledo Moreo, Pedro Navarro Lorente, F.Soto Valles, Juan Suardiaz Muro, Carlos Fernandez Andres, “Experiences on developing computer vision hardware algorithms using Xilinx system generator,” Microprocessors and Microsystems, vol.29(8-9), pp.411-419, Nov 2005.

[2] Alba M.Sanchez G, Richardo Alvarez G, Sully Sanchez G, FCC and FCE BUAP, “ Architecture for filtering images using Xilinx System Generator,” International Journal on Mathematics and Computers in Simulation, vol.1(2), pp.101-107, May 2007.

[3] Li Tan, “ Digital Signal Processing – Fundamentals and Applications, ” Elsevier, 2008.

[4] Rhys Lewis, “Practical Digital Image Processing”, Ellis Horwood Series in Digital and Signal Processing, 1991.

[5] C.H.Chen, Uvais Quidwai , “Digital Image Processing : An Algorithmic Approach with MATLAB, CRC Press, 2009.

[6] Ownby,M., Mahmoud,W.H., “A design methodology for implementing DSP with Xilinx System Generator for Matlab,” IEEE International Symposium on System Theory, pp.404-408, 2003.

[7] Xilinx Inc., “System Generator for Digital Signal Processing:, http://www.xilinx.com / tools / dsp.htm.

[8] T.Saidani, D.Dia, W.Elhamzi, M.Atri, R.Tourki, , “Hardware Co-simulation for Video Processing Using Xilinx System Generator,” Proceedings of the World Congress on Engineering, vol.1, Jun 2009.

[9] Randy Crane, “A Simplified Approach to Image Processing – Classical and Modern Techniques in C”, Prentice Hall, 1997.

[10] P. Karthigaikumar, K.Jaraline Kirubavathy, K.Baskaran, “FPGA based audio watermarking – Covert communication,” Microelectronics Journal, vol.42, pp.778-784, Feb 2011.

[11] Ana Toledo, Cristina Vicente-Chicote, Juan Suradiaz and Sergio Cuenca, “Xilinx System Generator Based HW Components for Rapid Prototyping of Computer Vision SW/HW Systems,” Lecture Notes in Computer Science, vol. 3522, pp.667-674, 2005.