International Journal on Recent and Innovation Trends in Computing and Communication
Volume: 5 Issue: 7

ISSN: 2321-8169
599 – 602

FPGA Implementation for Image Edge Detection using Xilinx System Generator

Nikhat Fatema Sheikh
Research Scholar (M.Tech), Dept. of ECE
Anjuman College of Engineering and Technology
Nagpur, India
nikhatsheikh111@gmail.com

Prof. Ruhina Quazi
Assistant Professor, Dept. of ECE
Anjuman College of Engineering and Technology
Nagpur, India
quazi_ruhina@yahoo.co.in

Abstract— Edge detection of an image is the primary and significant step in image processing. Image edge detection plays a vital role in computer vision and image processing. Hardware implementation of image edge detection is essential for real time application and it is used to increase the speed of operation. Field Programmable Gate Array(FPGA) plays a vital role in hardware implementation of image processing application because of its re-programmability and parallelism. The proposed work is FPGA implementation of image edge detection. The hardware implementation of edge detection algorithm is done using the most efficient tool called Xilinx System Generator(XSG).'Xilinx System Generator' (XSG) tool is used for system modeling and FPGA programming. The Xilinx System Generator tool is a new application in image processing, and offers a model based design for processing. The algorithms are designed by blocks and it also supports MATLAB codes through user customizable blocks. This paper aims at developing algorithmic models in MATLAB using Xilinx blockset for specific role then creating workspace in MATLAB to process image pixels and performing hardware implementation on FPGA.

Keywords-*Matlab, Xilinx System Generator, FFPGA, Edge Detection Algorithm.*

*****

I. INTRODUCTION

Edge detection is an image processing technique for finding the boundaries of the object within a image[6]. Edge detection is the name for a set of mathematical methods which aim at identifying points in a digital image at which the image brightness changes sharply or, more formally, has discontinuities.

Edge detection is a necessary tool[5] used in image processing applications to obtain information from the frame. This process detects boundaries between objects and the background in the image. Edge detection is one of the most common preprocessing step in image processing algorithms such as image enhancement , image segmentation, tracking and image/video coding,it provokes great interest in research fraternity. Edge detection changes the image for human interpretation and information extraction; It is useful in various fields such as in biomedical applications, traffic control, satellite imaging etc.

Edge detection process detects outlines of an object. Object and background in image is outlying by edge detection feature. A sharp discontinuity in an image is located by edge detection. The boundary of object in any image is characterized by discontinuity which gives instant change in pixel intensity. This process compress image without losing any important feature of that image. Edge detection is initial stage of image processing. Edge detection is one of the basic characteristics of the image. It is an important basis for the field of image analysis such as: the image segmentation, target area identification, extraction and other regional forms. Edge detection technology not only detect the image gray value of the , but also to determine their exact location

II. EDGE DETECTION ALGORITHMS

Edge detection algorithms are broadly classified as

1) First order derivative (Gradient) Method.
   - A. Sobel Operator
   - B. Prewitt Operator
   - C. Robert Operator
2) Second order derivative(Laplacian) Method.
   - A. Laplacian
   - B. Laplacian of Gaussian
   - C. Difference of Gaussian

A. Sobel edge detection algorithm.

Sobel edge detection gives an averaging and smoothing effect over the image therefore it also takes care of the noise present in the image. This technique extracts all of edges in an image, regardless of its direction. A Sobel edge detection operation provides both a differencing and smoothing effect . This includes a pair of 3×3 convolution masks. One mask is simple and the other rotated by 90°. The masks can be applied to the input image, to produce separate measurements of the gradient component in each direction. These masks are create to respond edges running horizontally and vertically.
The Sobel edge detector uses the masks shown in Fig-1 to approximate digitally the first derivatives Gx and Gy and finds edges using the Sobel approximation to the derivatives.

-1	-2	-1
0	0	0
1	2	1

Gx

-1	0	1
-2	0	2
-1	0	1

Gy

Fig.1: Convolution mask of Sobel edge detection

The masks can be applied differently to the image, to produce separate measurements of the gradient component (call these Gx and Gy).This detectortor works by calculating the gradient of intensity of the image , finding the direction of the change from light to dark and the magnitude corresponds to how sharp the edge is to find out the magnitude of the gradient and the orientation of that gradient. The gradient magnitude is given by:

$|G| = Gx^2 + Gy^2$

An approximate magnitude is computed using:

$|G| = |Gx| + |Gy|$ this is much faster to compute.

B. Prewitt edge detection algorithm.

Prewitt edge detection operators are the one of the oldest and well understood operator for detecting edges in images. Basically, there are two kernels, one for detecting image derivatives in X and another for detecting image derivative in Y. To obtain the maximum responses the Prewitt operator is used which are directly related to the kernel. Prewitt edge detector is an appropriate way to estimate the magnitude and orientation of an edge. The function of Prewitt edge detection is almost same as of sobel edge detection operator but Prewitt has different kernels. The Prewitt edge detection operators include a pair of 3x3 convolution kernel for 8directions. All the eight convolution kernels are calculated. The convolution kernel with the largest module is then selected. The Prewitt edge detector uses the masks in Fig-3 Gx and Gy.

The gradient magnitude is given by:

$$|\mathbf{G}| = \boldsymbol{Gx^2} + \boldsymbol{Gy^2}$$

In general an approximate magnitude is computed using:

$|\mathbf{G}| = |\mathbf{Gx}| + |\mathbf{Gy}|$ this is much faster to compute.

-1	-1	-1
0	0	0
1	1	1

Gx

-1	0	1
-1	0	1
-1	0	1

Gy

Fig.2: Convolution mask of Prewitt edge detection

III. PROPOSED SYSTEM

The entire operation of edge detection using Simulink and Xilinx blocks goes through three phases

A) Image Pre-Processing Blocks
B) Edge Detection using XSG
C) Image Post-Processing Blocks

A) Image Pre-Processing Blocksets.

As an image is two dimensional (2D) array size with R*C where R,C represent the row and column of an image respectively. For XSG implementation, image must be converted to one dimensional (1D) vector. Image pre-processing blocks are used to convert 2D image data to 1D array which is shown in Fig.3. Image Pre-processing blocks includes the Transpose, Convert 2-D to 1-D, Frame conversion and Unbuffer block. The transpose block transposes the R*C input image matrix into C*R sized matrix, Convert 2-D to 1-D block reshapes a C*R matrix input to a 1-D vector, Frame conversion block set the output signal to frame based data and provide to unbuffer block which converts this frame to scalar samples output samples at a higher sampling rate.

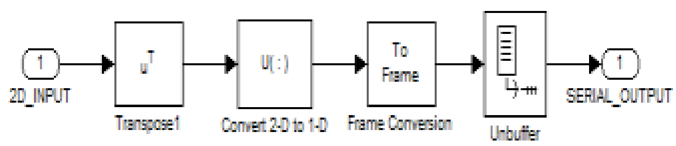

Fig.3: Image Pre-Processing

B) Edge Detection using XSG.

Figure 4 shows the model based design[8] using Xilinx system generator blocks for processing the input image for edge detection. To perform the edge detection, the input image is convolved with the horizontal and vertical mask which is made up of delay block, adder and substractor.

ISSN: 2321-8169
599 – 602

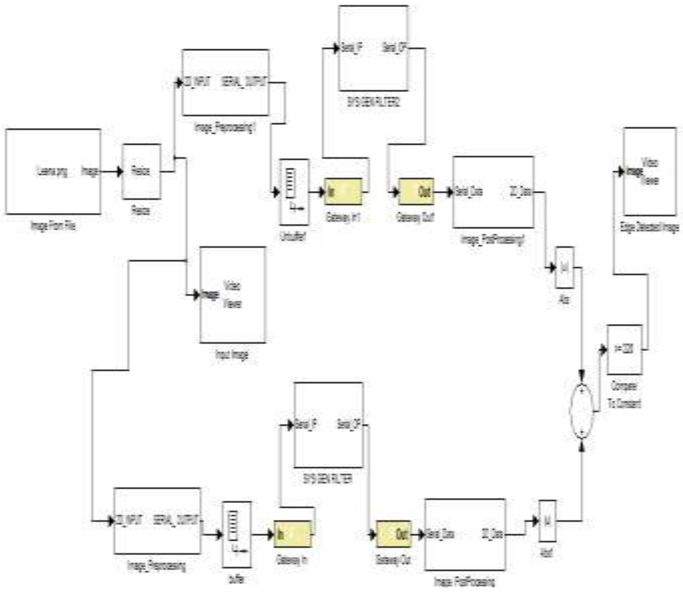

Fig.4: XSG model for edge detection

C) Image Post-Processing Blocksets.

Image post-processing is as shown in figure 5. Image post processing helps recreating image from 1D array. Primary significance of image post-processing is to make processed data available and with suitable data rate. It also includes a buffer block which converts scalar samples to frame output at lower sampling rate, followed by convert 1-D to 2-D, transpose blocks.

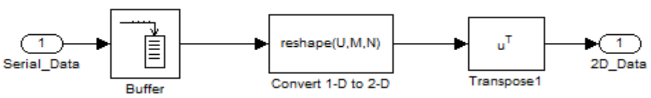

Fig.5: Image Post-Processing

IV. HARDWARE IMPLEMENTATION

For implementation of design in a FPGA board the entire module should be converted to FPGA synthesizable format. For that purpose main module for edge detection is converted to JTAG hardware co-simulation, this is done with the help of System generator block specially its system generator token. This block is configured according to the target platform and a bit stream (*.bit) file is generated. After the bit stream file is generated, hardware co-simulation target is selected and in this work, Spartan 3E starter kit (XC3S500E-FG320) is used for board level implementation. The entire architecture with the hardware and software co-simulation design is shown in figure 6.

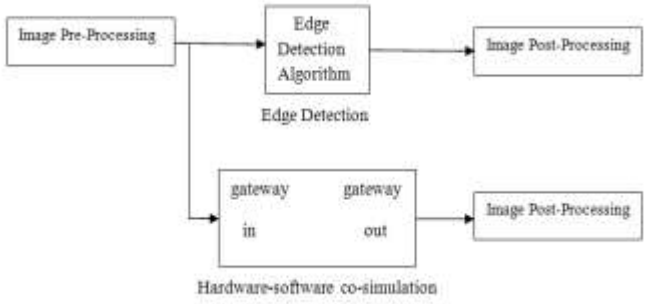

Fig.6: FPGA Hardware Implementation.

V. RESULTS AND COMPARISONS

Peak Signal to Noise Ratio (PSNR) is used to measure the similarity between Hardware/FPGA output and MATLAB/simulink output for a given image. As the PSNR value is high, the quality of image is good. Here, the PSNR value is calculated between the MATLAB/simulink output and FPGA output.

A. Results for edge detection based on Sobel operator:

The image shown in Fig.6 (a) of size 512X512 is given as input to the design and the corresponding software output and the FPGA/Hardware output are shown in Fig.18 (b), (c) respectively. The PSNR value obtained is 22.05. The Verilog code generated for edge detection consists of 38075 lines.

(a) (b) (c)

Fig.6: (a) input image (b) software output (c) FPGA output

B. Results for edge detection based on Prewitt operator:

The image shown in Fig.7 (a) of size 512X512 is given as input to the design and the corresponding software output and the FPGA/Hardware output are shown in Fig.7 (b), (c) respectively. The PSNR value obtained is 21.23. The Verilog code generated for edge detection consists of 19110 lines.

(a) (b) (c)

Fig.7: (a) input image (b) software output (c) FPGA output

C. Device utilization summary

In the design along with system generator token, resource estimator is added to calculate the amount of resources utilized after simulating the design in the FPGA. The amount of resources utilized for the image edge detection which are implemented in this paper are tabulated in the Table I.

TABLE I. DEVICE UTILIZATION SUMMARY

Image Processing Technique	Slices	FFs	LUTs	IOBs
Sobel operator edge detection	945	1357	1604	32
Prewitt operator edge detection	943	1357	1479	32

VI. CONCLUSION

In this paper, Sobel and Prewitt edge detection algorithms are designed and implemented on Spartan 3E FPGA using Xilinx System Generator. It is simpler to generate a stream of bit files rather than writing thousands of code lines for implementing of image processing techniques on FPGA. The PSNR value decreases as the number of pixels of input image increases.

REFERENCES

[1] SK.Areefabegam and T.Narendrakumar, “FPGA Based Design and Implementation of Image Edge Detection Using Xilinx System Generator.” International Journal of New Trends in Electronics and Communication (IJNTEC-ISSN: 2347-7334) Vol. 2, Issue. 8, Nov. 2014

[2] Prof.(Dr.) P.K Dash, Prof. Shashank Pujari, Miss. Sofia Nayak, “Implementation of Edge Detection Using Fpga & Model Based Approach,” ICICES2014 - S.A.Engineering College, Chennai, Tamil Nadu, India, ISBN No.978-1-4799-3834-6/14/$31.00©2014 IEEE

[3] G. Bharadwaja Reddy, K. Anusudha, " Implementation of Image Edge Detection on FPGA using XSG", 2016 International Conference on Circuit, Power and Computing Technologies [ICCPCT] , 978-1-5090-1277-0/16/$31.00 ©2016 IEEE

[4] Qian Xu, Srenivas Varadarajan, Chaitali Chakrabarti, and Lina J. Karam, “A Distributed Canny Edge Detector: Algorithm and FPGA Implementation IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 23, NO. 7, JULY 2014.

[5] Swapnil G. Kavitkar, and Prashant L. Paikraoarn , “FPGA based Image Feature Extraction Using Xilinx System Generator,” International Journal of Computer Science and Information Technologies, Vol. 5 (3) , 2014, 3743-3747

[6] Dharampal and Vikram Mutneja, "Methods of Image Edge Detection: A Review," J Electr Electron Syst Volume 4 • Issue 2 • 1000150 ISSN: 2332-0796 JEES an open access journal.

[7] R.Jayakumar, B.Suresh, "A REVIEW ON EDGE DETECTION METHODS AND TECHN International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, Issue 4, April 2014

[8] Aniket A. Ingle, Vrushali G Raut, " HARDWARE SOFTWARE CO-SIMULATION OF EDGE DETECTION FOR IMAGE PROCESSING SYSTEM USING DELAY BLOCK IN XSG", International Journal of Research in Engineering and Technology Volume: 03 Issue: 05 | May-2014