BES

Proceedings

Bangladesh Electronics Society Conference 2010

2-3 June 2010
Dhaka, Bangladesh

National Seminar on
Electronics and Telecommunications for Digital Bangladesh

Organized By

Bangladesh Electronics Society

Conference on Electronics and Telecommunication 2010
Bangladesh Electronic Society

Implementation of a FPGA based Architecture of Prewitt Edge detection Algorithm using Verilog HDL

Mohammad Nazmul Haque
Department of Computer Science & Engineering
Daffodil International University, Dhaka, Bangladesh
e-mail: nazmul@daffodilvarsity.edu.bd

This paper presents FPGA based architecture for Prewitt edge detection operator. Typical use of Edge detection is in the computer vision, robotics and artificial intelligence system. Currently the image processing algorithms has been limited to software implementation which is slower due to the limited processor speed. So a dedicated processor for edge detection has been required which was not possible until advancement in VLSI technology. Now more complex system can be integrated on a single chip providing a platform to process realtime algorithms on hardware. This architecture is implemented using Verilog HDL language and verified by Simulation using ModelSim.

Keywords: Edge detection algorithms, hardware implementation, prewitt operators, VLSI, Verilog HDL.

I. Introduction

An edge can be defined as an abrupt change in brightness as we move from one pixel to its neighbor in an image. In digital image processing, each image is quantized into pixels. In gray-scale each pixel *i(x,y)* in the image indicates the amplitude of intensity *i* of the image in a particular spatial coordinate *(x,y)*. The intensity value 0 represents black, and with 8-bit pixels, 255 represent white. An edge is an abrupt change in the intensity (gray scale level) of the pixels. Detecting edges is an important task in boundary detection, motion detection/estimation, texture analysis, segmentation, and object identification. Human eye can detect edges very quickly. But it is most difficult for machine such as computer vision, robotics etc.

II. Edge Detection

Edge information for a particular pixel is obtained by exploring the intensity of pixels in the neighborhood of that pixel. If all of the pixels in the neighborhood have almost the same brightness, then there is probably no edge at that point. However, if some of the neighbors are much brighter than the others, then there is a probably an edge at that point. Measuring the relative brightness of pixels in a neighborhood is mathematically analogous to calculating the derivative of brightness. The image illustrates an example of Hard and Soft Edges on an image. Brightness values are discrete, not continuous, so we approximate the derivative function. Different edge detection methods (Prewitt, Laplacian, Roberts, Sobel and Canny) use different discrete approximations of the derivative function [1–6].

Figure 1: Types of Edges

III. FPGA Primer

In early processes, the edge detection was mainly performed on software due to its large hardware equipment and also the application specific integrated circuits have not gain much advancement. But present researches on programmable devices make it possible to implement edge detection algorithms on these devices whose design turnaround time varies from few hours to few days.

During the recent years field programmable gate arrays (FPGA's) have become the prevailing form of programmable logic [8-12] in comparison to previous programmable devices like complex programmable logic devices (CPLD's) and programmable array logic (PAL). FPGA's supports sufficient logic to implement complete systems and subsystems. FPGA exploit the increasing capacity of integrated circuits to provide designers with reconfigurable logic that can be programmed on application specific basis. This reconfigurable architecture provides high level of parallelism. This radically increases flexibility in both the design process and the final artifact by permitting one board level design to perform many functions, or to be upgraded in the field.

IV. Prewitt Edge Detection Algorithm

Prewitt is gradient based edge detection algorithm. The gradient method looks the edges by finding maximum and minimum in the first derivative of the image. Prewitt algorithm performs a 2-D spatial gradient measurement on an image. It uses a pair of 3X3 convolution masks, one estimating gradient in x-direction and other in y-direction. Then resultant magnitude is computed from the above two gradients. The gradient of the 2-D image *i(x,y)* at spatial coordinate *(x,y)* is defined as the vector

$$\overline{\nabla I} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial I(x,y)}{\partial x} \\ \dfrac{\partial I(x,y)}{\partial y} \end{bmatrix}$$

The magnitude of this vector is $\theta(x,y) = \tan^{-1}\begin{pmatrix} G_x \\ G_y \end{pmatrix}$ or $\nabla I = |\,\mathrm{Gx}\,| + |\,\mathrm{Gy}\,|$ and the direction of the gradient vector is obtained by:

$$\nabla I = \left\|\overline{\nabla I}\right\| = \sqrt{\left[G_x^2 + G_y^2\right]}$$

The Prewitt Edge filter is use to detect edges based applying a horizontal and vertical filter in sequence. Both filters are applied to the image and summed to form the final result. The two filters are basic convolution filters of the form:

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

a) X-directional convolution mask *Gx*

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

b) Y-directional convolution mask *Gy*

p1	p2	p3
p4	p5	p6
p7	p8	p9

c) Image Neighborhood

Figure 2 : Image Kernel and Prewitt Convolution Masks

From the outputs of these convolution blocks resultant absolute magnitude is computed. This magnitude is the final output of the Prewitt edge detection output. If the X-directional convolution mask is placed on p5 pixel then we get:

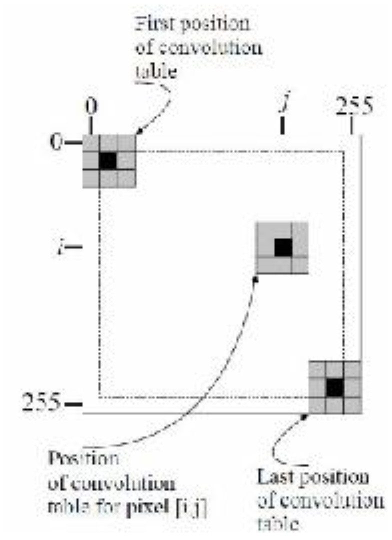
$$Gx = (p1 + p2 + p3) - (p7 + p8 + p9)$$

By placing Y-directional mask on kernel the new value is given by:

$$Gy = (p3 + p6 + p9) - (p1 + p4 + p7)$$

The new value of that kernel's center pixel *p5* will be given by: $G = |\,\mathrm{Gx}\,| + |\,\mathrm{Gy}\,|$

By applying this procedure to all over the image the edges of x and y direction will be gotten.


First position of convolution table
0
j
255
0
i
255
Position of convolution table for pixel [i,j]
Last position of convolution table

$$\underset{Kernel}{\begin{bmatrix} 12 & 167 & 120 \\ 50 & 30 & 23 \\ 1 & 123 & 25 \end{bmatrix}} \otimes \underset{X\text{-}Mask}{\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}}$$

Gx = (12+167+120) – (1+123+25)
Gx=150

$$\underset{Kernel}{\begin{bmatrix} 12 & 167 & 120 \\ 50 & 30 & 23 \\ 1 & 123 & 25 \end{bmatrix}} \otimes \underset{Y\text{-}Mask}{\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}}$$

Gy = (120+23+25) – (12+50+1)
Gy=105

$$G = |150| + |105| \approx 255$$

Figure 3: Convolution Example for Image Pixel

IV. Hardware Architecture of Prewitt Edge Detector Instance

P1, P2, P3, P4, P6, P7, P8 and P9 represent the eight 8bit pixel inputs of the image Kernel to the Prewitt Module. The module consists of signed subtractors, shift registers and modulus operators. The output of the final adder block will be 11 bits (10 bits for the data as the maximum value of the adder output is 4*255 and the 11th bit as the sign bit). The output data is compared to limit the value to a maximum of 255 as the output image is also composed of 8-bit wide pixels. The Prewitt output for one group of pixels calculated as per |Gx| + |Gy|.

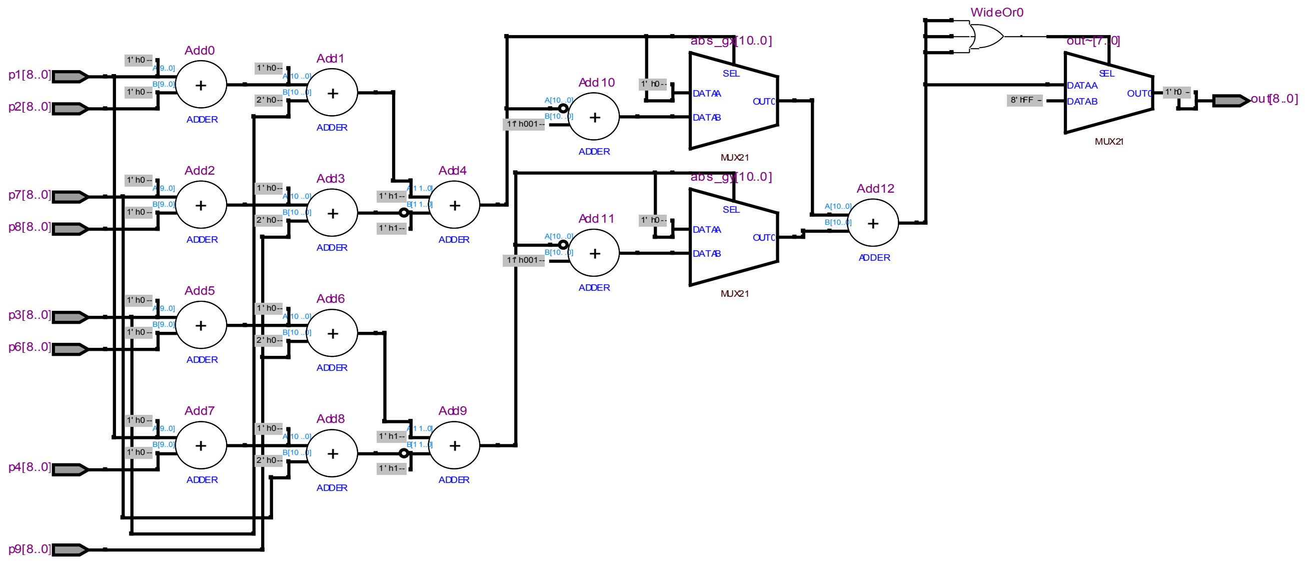
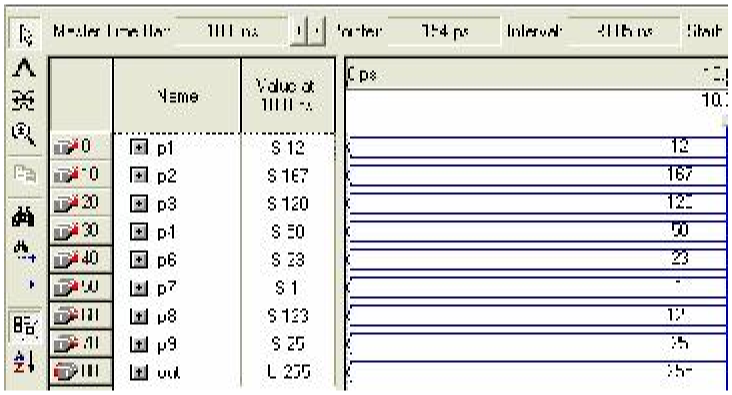


Figure 4: RTL View of Prewitt Module

VI. Results

The architecture for Prewitt edge detection operator was implemented on VerilogHDL [7–10] and simulation on Modelsim Altera 6.4a from Mentor Graphics Corporation. The test images used are maximum 512 pixels in either height or width with 256 gray levels. Figure 5 shows the simulation output shown in the convolution example of Figure 3. Figure 6 shows output for Prewitt Edge Detection Hardware module.


Master Time Bar: 1000 ns | Pointer: 754 ps | Interval: [illegible] ns | Start

	Name	Value at 1000 ns	0 ps
0	p1	S 12	12
10	p2	S 167	167
20	p3	S 120	12[illegible]
30	p4	S 50	50
40	p6	S 23	23
50	p7	S 1	[illegible]
[illegible]	p8	S 123	12[illegible]
[illegible]	p9	S 25	25
[illegible]	out	U 255	25[illegible]

Figure 5: Simulation Output of Prewitt Hardware Module

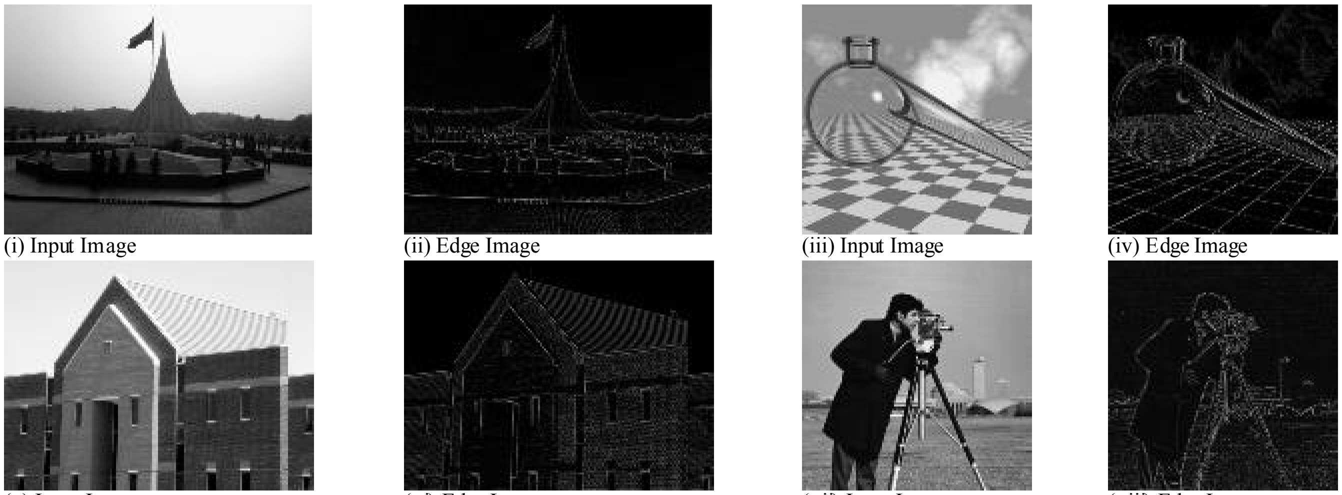

(i) Input Image (ii) Edge Image (iii) Input Image (iv) Edge Image
(v) Input Image (vi) Edge Image (vii) Input Image (viii) Edge Image

Figure 6: Hardware Architecture Validation

VII. Conclusion and Future Scope

The FPGA based architecture for Prewitt edge detection operator is proposed. This architecture is capable of operating at much higher speed than processing images on software platform using high level programming languages like C or C++. This architecture the result of edge detection can be found out for big images (like of size 1024 × 1024 pixels) in just 7.8 ms. Further improvement in speed could be achieved by appending more pipelining stages at decoder and processing blocks level but it may increase silicon area considerably.

References

[1] Jain, Anil K. *Fundamentals of Digital Image Processing*, Prentice-Hall,Inc.
[2] Chanda, B.and Dutta, D. Majumdar. *Digital Image Processing and Analysis*,Prentice Hall of India.
[3] Gonzalez, Rafael C. and Woods, Richard E. *Digital Image Processing*, Pearson Education, Inc.
[4] J. F. Canny. *A computational approach to edge detection*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(6):769–798, November 1986.
[5] Pratt,W. K. *Digital Image Processing*, John Wiley & Sons, Inc.
[6] Heath, Mike, Sarkar, Sudeep, Sanocki, Thomas, and Bowyer, Kevin, *Comparison of Edge Detectors: A Methodology and Initial Study*.
[7] Palnitkar, Samir. *Verilog HDL-A Guide to Digital Design and Synthesis*, Pearson Education.
[8] Chan C.,Mohanakrishnan, Evans, *FPGA Implementation of Digital Filters*, Proc ICSPAT, 1993
[9] Thomas, Donald and Moorby, Phil. *The Verilog Hardware Description Language*, Kluwer Academic Publishers.
[10] Smith, Douglas. *HDL Chip Design: A practical Guide for Designing, Synthesizing and Simulating ASICs and FPGAs using VHDL or Verilog* ,Doone Publications.
[11] Jenkins, Jesse H.*Designing with FPGAs and CPLDs*,Prentice-Hall Publications. F.G.Lorca, L Kessal and D.Demigny. *Efficent ASIC and FPGA implementation of IIR filters for Real time edge detection*. International Conference on image processing (ICIP-97) Volume 2. Oct 1997.
[12]Wakerly, JohnF. *Digital Design: Principles and Practices*, Pearson Education Asia. Muthukumar Venkatesan and Daggu Venkateshwar Rao, *Hardware Acceleration of Edge Detection Algorithm on FPGAs*,