



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

IJIRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 12, Issue 10, October 2023

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.423

9940 572 462

6381 907 438

ijirset@gmail.com

www.ijirset.com



VLSI Design an Efficient Implementation of Edge Detection Algorithm

S. Sailaja¹, G.Sowndharya²

Assistant Professor, Department of Electronics and Communication Engineering, Jaya Engineering College,
Thiruninravur, Tamilnadu, India¹

PG Student, Department of Electronics and Communication, Jaya Engineering College, Thiruninravur,
Tamilnadu, India²

ABSTRACT: Edge detection is an important and growing area in different fields such as image pattern recognition, machine learning and processing, and Computer vision. Edge detection of an image of the object is the main goal for different Edge detection algorithms. In Edge detection, the most important step is to identify the edges of an image of the object and pixel information of the Image In every edge detection algorithm. The effective and efficient performance of a digital system mainly depends on the delay, area, and power consumption. The separameters decide the efficiency of any digital core system..

I.INTRODUCTION

Edge detection techniques have been successfully used for different applications. In edge detection, the abrupt changes in the pixel intensity are determined. These change in pixel intensities are determined by different techniques, in which different parameters are tuned to refine the edges of salient objects while suppressing the redundant objects from image. The edges obtained by different edge detector are broadly classified into two types: correct edges and false edges. Correct edge represent salient object and false edges are produced due to detector sensitiveness. The edge detection algorithms have three steps: filtering, enhancement and detection. Filtering is normally used to remove the noise from image. Enhancement is used to magnify the pixel intensity values in local area of an image and in detection the strong edges are determined.

II. PROPOSED SYSTEM FLOW

2.1Process flow of RGB to binary image conversion (MATLAB Part)

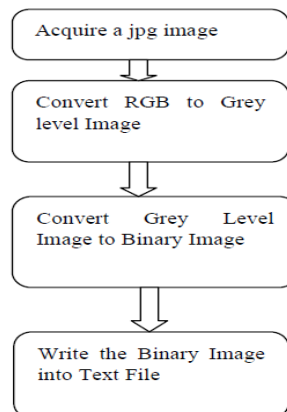


Fig 2.1



2.2 PROPOSED SYSTEM TECHNIQUE:

- Sobel Edge Detection using Binary Operators

2.3 PROPOSED SYSTEM ADVANTAGES:

- Area Efficient
- Delay Efficient

III. MODULE & EXPLANATIONS

3.1MODULES NAME:

- Reading Image
- Compare the gradient magnitude with threshold value and find true edges
- Applying the convolution mask i and j on the input image
- Determine the gradient magnitude by computing
- Compare the gradient magnitude with threshold value and find true edge

3.2 READING IMAGE:

Sobel operator is used to detect edges of the test images used. This procedure is applied on more than one test image as shown in Fig. Firstly the image data is read as an array with the dimension of image size.



Fig 3.1

3.3 APPLYING THE CONVOLUTION MASK I AND J ON THE INPUT IMAGE:

The horizontal template and vertical template shown in Fig. 1.6 above are used to get convolution with input image by using equation (1) and (2). The result matrix after this operation is got the same size of two gradients Matrix Gx and Gy as the original image as shown in Fig



Fig 3.2



IV. SOBEL EDGE DETECTOR OPERATOR

Sobel Edge Detection, there are two masks, one mask identifies the horizontal edges and the other mask identifies the vertical edges. Each of the masks has the effect of calculating the gradient in both vertical and horizontal direction

$$G_i = G_x * F(i,j) \text{ and } G_j = G_y * F(i,j)$$

4.1 SOBEL EDGE DETECTION:

Algorithm for Sobel Edge Detection The Sobel Edge Detection Operator is 3x3 spatial mask. It is based on first derivative based operation. The Sobel masks are defined as:

$$\begin{matrix}
 & -1 & -2 & -1 \\
 H1 = & 0 & 0 & 0 \\
 & 1 & 2 & 1
 \end{matrix}
 \quad (1) \quad
 \begin{matrix}
 & -1 & 0 & 1 \\
 H2 = & -2 & 0 & 2 \\
 & -1 & 0 & 1
 \end{matrix}$$

Suppose the pixel value of a 3x3 sub-window of an image is as shown as in Table

D0	D1	D2
D3	D4	D5
D6	D7	D8

The pixel intensity values of a 3x3 sub-window of an image

Applying the Sobel mask on the sub-window of Fig. 1 yields

$$\begin{aligned}
 G_x &= (D6 + 2 \times D7 + D8) - (D0 + 2 \times D1 + D2) \\
 &= f1 - f2 \text{ Where, } f1 = (D6 + 2 \times D7 + D8) \text{ and } f2 = (D0 + 2 \times D1 + D2)
 \end{aligned}$$

$$\begin{aligned}
 G_y &= (D2 + 2 \times D5 + D8) - (D0 + 2 \times D3 + D6) \\
 &= f3 - f4 \text{ Where, } f3 = (D2 + 2 \times D5 + D8) \text{ and } f4 = (D0 + 2 \times D3 + D6)
 \end{aligned}$$

5 SOFTWARE REQUIREMENTS

- Verification Tool
 - Modelsim 6.4c
- Synthesis Tool
 - Xilinx ISE 13.2

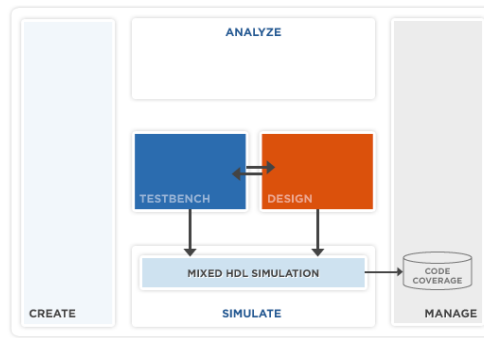
V. INTRODUCTION TO MODELSIM

ModelSim is a useful tool that allows you to stimulate the inputs of your modules and view both outputs and internal signals. It allows you to do both behavioral and timing simulation, however, this document will focus on behavioral simulation. Keep in mind that these simulations are based on models and thus the results are only as accurate as the constituent models.



5.1 MODELSIM - ADVANCED SIMULATION AND DEBUG

ASIC and FPGA design



VI. SOBEL WITH PROPOSED STOCHASTIC LOGIC:

assign gx=((p2-p0)+((p5-p3)<<1)+(p8-p6)); //sobel mask for gradient in horiz. direction

assign gy=((p0-p6)+((p1-p7)<<1)+(p2-p8)); //sobel mask for gradient in vertical direction

assign abs_gx = (gx[10]? ~gx+1 : gx); // to find the absolute value of gx.

assign abs_gy = (gy[10]? ~gy+1 : gy); // to find the absolute value of gy.

assign sum = (abs_gx+abs_gy); // finding the sum

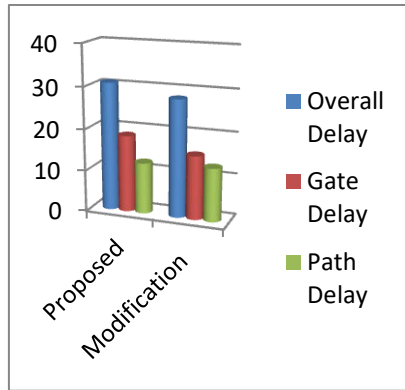
assign dxy = (sum[10:8]?8'hff :8'h00); // to limit the max value to 255

VII. AREA COMPARISON

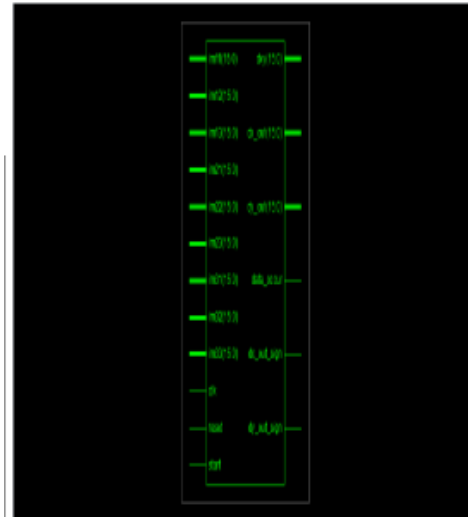
7.1 AREA COMPARISON

Method	Area			Delay		
	LUT	Slices	Gates	Overall Delay	Gate Delay	Path Delay
Spartan 3						
Proposed	302	156	3705	30.745ns	18.485ns	12.260ns
Modification	119	77	1736	28.121ns	15.304ns	12.817ns

7.2 DELAY CALCULATION



7.3 RTL SCHEMATIC



VIII. CONCLUSION

This paper mainly focuses on the design and simulation System of the Sobel edge detection method. This method uses two 3×3 convolution masks to estimate gradient in X and Y-direction and which is easy to implement than other operators. The Sobel edge detection method calculates 2-D spatial gradient of image intensity at each point of an image by convolution with small and integer.

REFERENCES

- [1] A. Alaghi and J. P. Hayes, "Survey of stochastic computing," *ACM Trans. Embedded Comput. Syst.*, vol. 12, no. 2s, pp. 92:1–92:19, May 2013.
- [2] J. P. Hayes, "Introduction to stochastic computing and its challenges," in *Proc. 52nd ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, Jun. 2015, pp. 1–3.
- [3] B. R. Gaines, "Stochastic computing systems," in *Advances in Information Systems Science*, J. T. Tou, Ed. New York, NY, USA: Springer, 1969, pp. 37–172.
- [4] P. Li, D. J. Lilja, W. Qian, K. Bazargan, and M. D. Riedel, "Computation on stochastic bit streams digital image processing case studies," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 22, no. 3, pp. 449–462, Mar. 2014.
- [5] M. H. Najafi and M. E. Salehi, "A fast fault-tolerant architecture for sauvola local image thresholding algorithm using stochastic computing," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 2, pp. 808–812, Feb. 2016.
- [6] A. Alaghi, C. Li, and J. P. Hayes, "Stochastic circuits for real-time image-processing applications," in *Proc. 50th ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, May 2013, pp. 1–6.
- [7] D. Fick, G. Kim, A. Wang, D. Blaauw, and D. Sylvester, "Mixed-signal stochastic computation demonstrated in an image sensor with integrated 2D edge detection and noise filtering," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Sep. 2014, pp. 1–4.
- [8] S. S. Tehrani, W. J. Gross, and S. Mannor, "Stochastic decoding of LDPC codes," *IEEE Commun. Lett.*, vol. 10, no. 10, pp. 716–718, Oct. 2006.
- [9] N. Onizawa, W. J. Gross, T. Hanyu, and V. C. Gaudet, "Asynchronous stochastic decoding of LDPC codes: Algorithm and simulation model," *IEICE Trans. Inf. Syst.*, vol. 97, no. 9, pp. 2286–2295, 2014.
- [10] B. D. Brown and H. C. Card, "Stochastic neural computation. I. Computational elements," *IEEE Trans. Comput.*, vol. 50, no. 9, pp. 891–905, Sep. 2001.



|| Volume 12, Issue 10, October 2023 ||

| DOI:10.15680/IJIRSET.2023.1210020 |

- [11] K. Kim, J. Kim, J. Yu, J. Seo, J. Lee, and K. Choi, "Dynamic energyaccuracy trade-off using stochastic computing in deep neural networks," in Proc. 53rd Annu. Design Autom. Conf. (DAC), New York, NY, USA, 2016, pp. 124:1–124:6.
- [12] B. Li, M. H. Najafi, and D. J. Lilja, "Using stochastic computing to reduce the hardware requirements for a restricted Boltzmann machine classifier," in Proc. ACM/SIGDA Int. Symp. Field-Programmable Gate Arrays (FPGA), New York, NY, USA, 2016, pp. 36–41.
- [13] Y. Liu, H. Venkataraman, Z. Zhang, and K. K. Parhi, "Machine learning classifiers using stochastic logic," in Proc. IEEE 34th Int. Conf. Comput. Design (ICCD), Oct. 2016, pp. 408–411.
- [14] W. Qian, X. Li, M. D. Riedel, K. Bazargan, and D. J. Lilja, "An architecture for fault-tolerant computation with stochastic logic," *IEEE Trans. Comput.*, vol. 60, no. 1, pp. 93–105, Jan. 2011.
- [15] Q. Tang, B. Kim, Y. Lao, K. K. Parhi, and C. H. Kim, "True random number generator circuits based on single- and multi-phase beat frequency detection," in Proc. IEEE Custom Integr. Circuits Conf. (CICC), Sep. 2014, pp. 1–4.
- [16] W. H. Choi et al., "A magnetic tunnel junction based true random number generator with conditional perturb and real-time output probability tracking," in Proc. IEEE Int. Electron Devices Meeting (IEDM), Dec. 2014, pp. 12.5.1–12.5.4.



Impact Factor: 8.423



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details