IJIRSET
International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

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

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 13, Issue 7, July 2024

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.423

9940 572 462 | 6381 907 438 | ijirset@gmail.com | www.ijirset.com

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.423

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753; p-ISSN: 2347-6710|

Volume 13, Issue 7, July 2024

|DOI: 10.15680/IJIRSET.2024.1307156|

An Efficient VLSI Implementation of Edge Detection Algorithm for Image Processing Application

G.Viji, P.Sindhuja

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

PG Student, Department of Electronics and Communication Engineering, Thiruninravur, India

ABSTRACT: Edge detection is the core research area among different fields such as; image processing.Computer vision, machine learning and pattern recognition. In object detection, the first obligatory step isto determine the edges of an object in better way that is further used for processing. The feature vector comprises of nothing but key point description, which provides the information of edges. Edge detection is the key to success and is somehow or the other dependent on it. The proposed Sobel edge detection algorithm uses approximation methods to replace the complex operations; the pipelining is employed to reduce the latency. Finally, this algorithm is implemented by Verilog HDL. When compared with the previous hardware architecture for Sobel edge detection, the proposed architecture requires fewer Area, Delay & Power Efficient.

I. INTRODUCTION

Edge detection techniques have been successfully used for different applications. In edge detection, the abrupt changesin 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 detectionthe strong edges are determined. Recent research in the fields of Artificial Intelligence computer vision and Pattern Recognition reveals that the edge detection is very important in some way or the other. Key point detection is one major part of the process that majorly deals with image edges not only edges but also true edges. Identified key pointsare then used to describe the feature vectors that are further used in different applications. There is much research going on nowadays on edge detection as it has a key role in almost all upcoming fields.

II. SOBEL EDGE DETECTION

Edge detection algorithms are widely used in various research fields like Image Processing, Video Processing and Artificial Intelligence etc. Edges are most important attribute of image information and a lot of edge detection algorithms are defined in literature. Sobel edge detection algorithm is chosen among of them due to its property of less deterioration in high level of noise. FPGA is becoming the most dominant form of programmable logic over past few years and it has advantages of low investment cost and desktop testing with moderate processing speed and thereby offering itself as suitable one for real time application. This paper describes an efficient architecture for Sobel edge detector which is faster and takes less space than the Existing architecture.

1- First order derivative operator's (gradient method) is contain Robert Detector,Prewitt Detector and Sobel Detector where: Robert Detector : It is gradient based operator. It firstly computes the sum of the squares of the differencebetween diagonally adjacent pixels through discrete differentiation and then calculate approximate gradient of the image.The input image is convolved with the default kernels of operator and gradient magnitude and directions are computed. It uses following 2 x2 two kernels as shown in bellow Table 1.3

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.423

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753; p-ISSN: 2347-6710|

Volume 13, Issue 7, July 2024

|DOI: 10.15680/IJIRSET.2024.1307156|

Gx=

-1	0
0	1

Gy=

0	-1
1	0

Prewitt Detector: The function of Prewitt edge detector is almost same as of Sobel detector but have differentkernels as shown in bellow Table 1.4

Gx=

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

Gy=

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

And Sobel Detector: is one of the most frequently used in edge detection . Sobel edge detection can be implemented by filtering an image with left mask or kernel. Filter the image again with the other mask. After this square of the pixels values of each filtered image. Now add the two results and compute their root. The 3 × 3 convolution masks for the Sobel based operator as shown in Bellow Table 1.5

Gx=

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

Gy=

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

Sobel has two main advantages: it has some smoothing effect to the random noise of the image:

1) Since the introduction of the average factor, it has some smoothing effect to the random noise of the image.
2) Because it is the differential of two rows or two columns, so the element of the edge on both sides ha been enhanced, so that the edge seems thick and bright.

The Sobel operator is used mostly although it is slower than the Roberts cross operator, because its horizontal and vertical kernels smooth the input image and makes operator less sensitive to noise. The reason for using Sobel operator is that it has relatively small masks compare to other operators.

2- Second order derivative operator's is contain Laplacian of Gaussian:

The Laplacian is a 2-D isotropic measure of the 2nd spatial derivative of an image. The Laplacian of an image highlights regions of rapid intensity change and is therefore often used for edge detection. The operator normally takes a single gray level image as input and produces another gray level image as output. The Laplacian L(x,y) of an image with pixel intensity values I(x,y) is given below:

$$L(x,y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

Since the input image is represented as a set of discrete pixels, we have to find a discrete convolution kernel that can approximate the second derivatives in the definition of the Laplacian. Three commonly used small kernels are shown in bellow Table 1.6

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.423

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753; p-ISSN: 2347-6710|

Volume 13, Issue 7, July 2024

|DOI: 10.15680/IJIRSET.2024.1307156|

0	1	0
1	-4	1
0	1	0

1	1	1
1	-8	1
1	1	1

-1	2	-1
2	-4	2
-1	2	-1

Three commonly used discrete approximations to the Laplacian

Edges characterize boundaries and are therefore considered for prime importance in image processing. An edgeis seen at a place where an image has a strong intensity contrast. Edges could also be represented by a difference in color, without any difference in intensity. Edges include large amount of important information about the image. Thechanges in pixel intensity describe the boundaries of objects in an image.

III. SOBEL EDGE DETECTION

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

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

Suppose the pixel value of a 3×3 sub-window of an image is as shown as in Table 1.7

D0	D1	D2
D3	D4	D5
D6	D7	D8

The pixel intensity values of a 3×3 sub-window of an image Applying the Sobel mask on the sub-window of Fig. 1 yields

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

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

IV. SOFTWARE REQUIREMENTS

- Verification Tool
  - Modelsim 6.4c
- Synthesis Tool
  - Xilinx ISE 13.

Xilinx SRAM-based FPGAs

The basic structure of Xilinx FPGAs is array-based, meaning that each chip comprises a two dimensional array of logic blocks that can be interconnected via horizontal and vertical routing channels. An illustration of this type of architecture was shown in Figure. Xilinx introduced the first FPGA family, called the XC2000 series, in about1985 and now offers

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.423

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753; p-ISSN: 2347-6710|

Volume 13, Issue 7, July 2024

|DOI: 10.15680/IJIRSET.2024.1307156|

three more generations: XC3000, XC4000, and XC5000. Although the XC3000 devices are still widely used, we will focus on the more recent and more popular XC4000 family. We note that XC5000 is similar to XC4000, but has been engineered to offer similar features at a more attractive price, with some penalty in speed. We should also note that Xilinx has recently introduced an FPGA family based on anti-fuses, called the XC8100. The XC8100 has many interesting features, but since it is not yet in widespread use, we will not discuss it here. The Xilinx4000 family devices range in capacity from about 2000 to more than 15,000 equivalent gates. The XC4000 features a logic block (called a Configurable Logic Block (CLB) by Xilinx) that is based on look-up tables (LUTs). A LUT is a small one bit wide memory array, where the address lines for the memory are inputs of the logic block and the one bit output from the memory is the LUT output. A LUT with K inputs would then correspond to a 2K x 1 bit memory, and can realize any logic function of its K inputs by programming the logic function's truth table directly into the memory. The XC4000 CLB contains three separate LUTs, in the configuration . There are two 4-input LUTS that are fed by CLB inputs, and the third LUT can be used in combination with the other two. This arrangement allows the CLB to implement a wide range of logic functions of up to nine inputs, two separate functions of four inputs or other possibilities. Each CLB also contains two flip-flops.

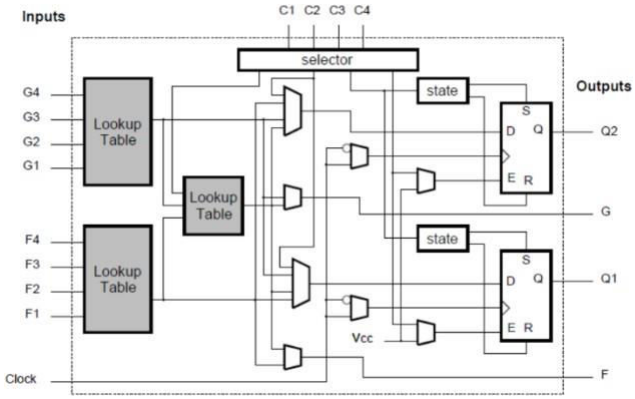


FIG Xilinx XC4000 Configurable Logic Block (CLB)

V. CODE IMPLEMENTATION

SOBEL WITH STOCHASTIC LOGIC

```
sobeldx sobeldx_inst(clk,rese
start, im11, im21, im31, im12,
im22,

im32,im13,im23,im33,dxy
                                                                    );

initial begin clk=0; reset=0; start=0;
end always
#20 clk=~clk;initial
                                                                    begin

$readmemh("image_textfile.txt",mem);file_id=$fopen("edgefile.txt");
endinitial begin #200;
for (k=0;k<(`MAX_LEN);k=k+9)begin

@(negedge clk)begin
reset=1; start=1; im11=mem[k]; im21=mem[k+1]; im31=mem[k+2]; im12=mem[k+3]; im22=mem[k+4];
im32=mem[k+5]; im13=mem[k+6]; im23=mem[k+7]; im33=mem[k+8];
endend
@(negedge clk) start=0; $fclose(file_id);
```

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.423

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753; p-ISSN: 2347-6710|

Volume 13, Issue 7, July 2024

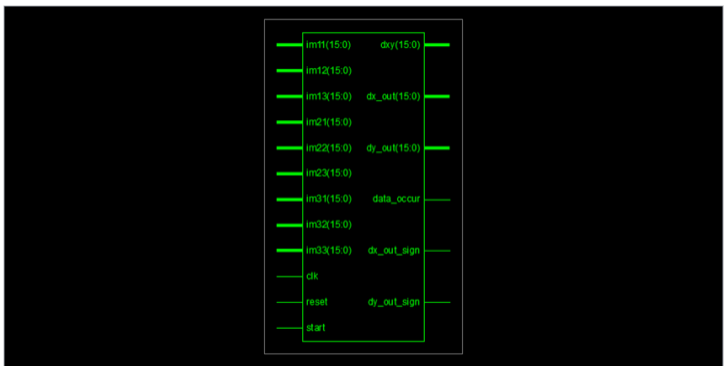
|DOI: 10.15680/IJIRSET.2024.1307156|

```
$finish;

End

always @(negedge clk)begin
if(reset)begin
$fdisplay(file_id,"%d",dxy);

$display("%d",dxy);End
```

DEVICE UTILIZATION SUMMARY

Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of 4 input LUTs	119	55,296	1%	
Logic Distribution				
Number of occupied Slices	77	27,648	1%	
Number of Slices containing only related logic	77	77	100%	
Number of Slices containing unrelated logic	0	77	0%	
Total Number of 4 input LUTs	121	55,296	1%	
Number used as logic	119			
Number used as a route-thru	2			
Number of bonded IOBs	75	633	11%	
IOB Flip Flops	64			
Number of GCLKs	1	8	12%	
Total equivalent gate count for design	1,736			
Additional JTAG gate count for IOBs	3,600			

RTL SCHEMATIC



VI. 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 pointof an image by convolution with small and integer valued filters therefore relatively less expensive in terms of computations. Application of Sobel edge detection algorithm to an image may considerably decrease the amount of details to be processed and with pulse width modulation, time-encoded signals corresponding to specific values are generated by adjusting the frequency and duty cycles of signals. With this approach, the latency, area consumption is all greatly reduced.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.423

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753; p-ISSN: 2347-6710|

Volume 13, Issue 7, July 2024

|DOI: 10.15680/IJIRSET.2024.1307156|

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.
11. K. Kim, J. Kim, J. Yu, J. Seo, J. Lee, and K. Choi, "Dynamic energy accuracy 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-Program. Gate Arrays (FPGA), NewYork, 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.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

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