Area Efficient FPGA Implementation of Sobel Edge Detector for Image Processing Applications

Rajesh Mehra
Associate Professor
Department of Electronics and Comm. Engg
NITTTR, Sector-26, Chandigarh (India)

Rupinder Verma
ME Student
Department of Electronics and Comm. Engg
NITTTR, Sector-26, Chandigarh (India)

ABSTRACT

In this Paper an efficient method of FPGA based design and implementation of area efficient Sobel Edge detection filter is presented using a combination of hardware and software components. The FPGA provides the necessary hardware for image processing algorithms with flexibility to support Sobel edge detection algorithm. A pipelined method is used to implement the edge detection filter. This approach is useful to improve the system performance by taking advantage of available look up tables, routing resources and shift registers available on target device. The proposed 2-D filter is designed using matlab, synthesized with Xilinx ISE 10.1 and implemented on Virtex II Pro based xc2vp30-7-FF896 FPGA device. Results show better performance of proposed design in terms of area utilization.

General Terms

Edge Detection, FPGA, Distributed Algorithm, Gradient Operator

Keywords

Sobel Edge detector, Image Processing Applications, FPGA

1. INTRODUCTION

Real-time video and image processing is used in a wide variety of applications from video surveillance and traffic management to medical imaging applications. These operations often require digital signal processing (DSP) algorithms for several crucial operations [1]. Digital Image Processing is a technique to process 2 dimensional image through computer. A Digital image is obtained from real image through the process of sampling and quantization. Image processing is used in many applications like video surveillance, traffic management and medical imaging. Edge detection is the process of locating an edge of an image. Detection of edges in an image is a very important step towards understanding image features. Edges consist of meaningful features and contained significant information. It's reduce significantly the amount of the image size and filters out information that may be regarded as less relevant, preserving the important structural properties of an image (Yuval, 1996) [2]. So this technique can be used in the field of image processing for object tracking and motion detection. Edges can be categorized based upon their intensity profiles like step edge, Ramp Edge, Ridge Edge, roof edge. Edge detection can be done in four steps smoothing, Enhancement, Detection and localization. There are many edge detection algorithms Prewitt, Robert, Sobel and canny but the proposed work is done on Sobel Edge detector. Field Programmable Gate Array hardware are becoming more and more important for image processing applications because it can be used to implement almost any digital logic function. Logic functions can be implemented by writing VHDL/Verilog code or by schematic diagram. Then schematic or VHDL codes are converted into binary bit file and loaded on target FPGA. Proposed design is implemented on Vertex II Pro Xilinx FPGA because Xilinx provides most flexible devices. FPGAs are fine grain, RAM Based Devices with special routing resources / carry chain to implement efficient arithmetic functions like counters, adders and comparators whereas CPLDs are coarse grain and EEPROM based devices and don't have routing resources or carry chains. General purpose microcontrollers are proved to be less useful when it comes to the implementation of image processing algorithms on embedded scale. In certain instances, image processing algorithms are implemented on a dedicated Application Specific Integrated Circuits (ASIC) and more commonly on Digital Signal processor (DSP). With the advent of Field Programmable Gate Arrays (FPGA), it has become an alternative for the implementation of image processing algorithms on ASIC as it provide speed comparable to an ASIC and is easily reconfigurable [3]. For Serial architecture look up tables (LUTs), Flip flops and registers can be used for area efficient implementation whereas parallel architectures can be implemented by using embedded multiplier and accumulate (MAC) units to achieve high speeds. In case fully parallel implementation is not possible then partial serial approach can be adopted to enhance the system performance. The proposed designs is implemented using serial architecture by taking optimal advantage of look up tables, carry chains and shift registers available on target device. Fully-parallel design cannot share hardware over multiple clock cycles and occupy large amounts of resource. Hence, efficient implementation of such filters is important to minimize hardware requirement.

2. SOBEL EDGE DETECTION

Most edge detection methods work on the assumption that the edge occurs where there is a discontinuity in the intensity function or a very steep intensity gradient in the image. Using this assumption, if one takes the derivative of the intensity value across the image and finds points where the derivative is higher and then edge could be located [4]. The Sobel edge detection operator is based on the same concept of 2D spatial filtering operation. Spatial filtering operation refers to periodicity with which pixel values change. The algorithms of first difference based operator are the simplest to implement on hardware as compared to other edge detection techniques. In Sobel Algorithm horizontal and vertical masks shown in figure 1 and 2 are used to calculate gradient at each point of image. Convolution is very small as compared to image. So horizontal and vertical masks are slide all over the image starts from upper left side of the image. Masks move to right on image until it reaches the end of row and then again starts from leftmost second row of image. In Sobel algorithm to suppress noise, a certain weight is increased on the centre point.

$$\mathrm{Mx} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Figure 1: Horizontal operator

$$\mathrm{My} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Figure 2: Vertical operator

Equation 1 shows convolution of input image with horizontal mask and Equation 2 shows convolution of image with vertical mask.

$$Gx = \{f(x+1, y-1) + 2f(x+1, y) + f(x+1, y+1)\} - \begin{Bmatrix} f(x-1, y-1) + 2f(x-1, y) \\ + f(x-1, y-1) \end{Bmatrix} \text{.......... ....} \quad (1)$$

$$Gy = \{f(x-1, y-1) + 2f(x, y-1) + f(x+1, y-1)\} - \begin{Bmatrix} f(x-1, y+1) + 2f(x, y+1) \\ + f(x+1, y+1) \end{Bmatrix} \text{.......... ....} \quad (2)$$

Size of gradient can be calculated from gradient vectors as shown in equation 3.

$$G(x, y) = [G_x^2 + G_y^2]^{1/2} \text{.......... ....} (3)$$

Gradient can be written as shown below in equation 4.

$$G(x, y) = | G_x | + | G_y | \quad \text{.......... ....} (4)$$

If we use Sobel operator to detect the edge of image f, then we can use the horizontal template Mx and vertical template My shown in figure 1 and figure 2 to get convolution with image, without taking into account the border conditions. It may get the same size of two gradient Matrix Gx and Gy as the original image. Then the total gradient value G can be obtained by adding the two gradient matrices. Finally, we can get the edge by applying threshold. If G is greater than threshold then pixel should be identified as edge. 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 Sobel masks produces higher output values for edges which are similar as compared to Roberts cross edge detection. The reason for using Sobel operator is that it has relatively small masks compare to other operators. Normally there is a lot of noise in images; therefore, preprocessing step should be used to eliminate noise from the images before edge detection. Sobel edge detection has been implemented using 2D filters for horizontal and vertical convolution.

3. PROPOSED MODEL

Methodology includes designing of 2D filter in matlab. Synthesized and optimized using VHDL Programming using ISE 10.1i and implemented on vertex II pro FPGA as shown in figure 3.

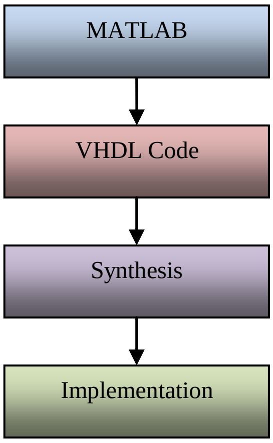

Figure 3: Design Methodology

The model is designed to detect edges by using 2-D image filter. The proposed architecture of Sobel edge detector is shown in figure 4.

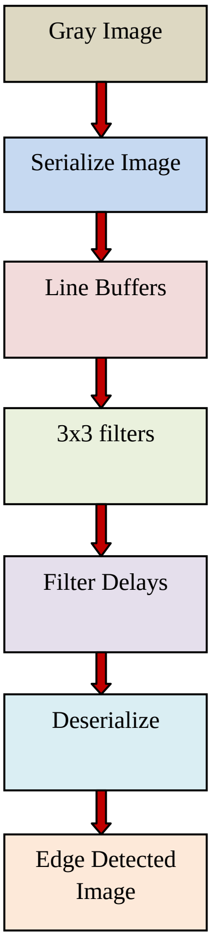

Figure 4: Proposed Model

This design has been implemented using a 2-D filter. 2D filter performs convolution of input matrix using filter coefficient matrix. In the design pipelined Sobel edge detection algorithm is implemented on serialized image. Line buffers are used to create line delay during shifting of incoming pixels. These line buffers add delay to feed pixel values to filters. 3x3 horizontal and vertical filters are realized in matlab using m-files in embedded matlab block in Simulink. Then computed convolutions are stored in shift registers to produce edge detected output image. Filter design is based upon serial sequential Distributed algorithm shown in figure 5. In this algorithm array of multiplicands is multiplied with array of multiplier. All possible combinations of Pre-computed multiplicand sums are stored in look up table. Multiplier A with n-bits data is loaded at the rate of clk / n in parallel in serial out register. This data goes to each shift register. At this time, each serial shift register has a multiplier value. For Edge detection masks are stored in shift registers. Multiplicand B is image input in sequential manner to look up table. The LSB from each serial shift register is used to address the look up table data. Each address bit from shift registers represents a unique multiplicand word B, such that, address bit-0 corresponds to $B_{0,}$ Address bit 1 corresponds to $B_{1,}$ and Address bit 2 corresponds to B2 until kth bit. If the address bit k is 1 then corresponding multiplicand B_k is included in the summation of product terms which is represented by look up table data output. If the address bit is n is 0, then B_n is not included in the total summation. Filter design used in proposed model is shown in figure 5. A, B are the inputs and output is C which is again stored in registers to produce edge detected gray image.

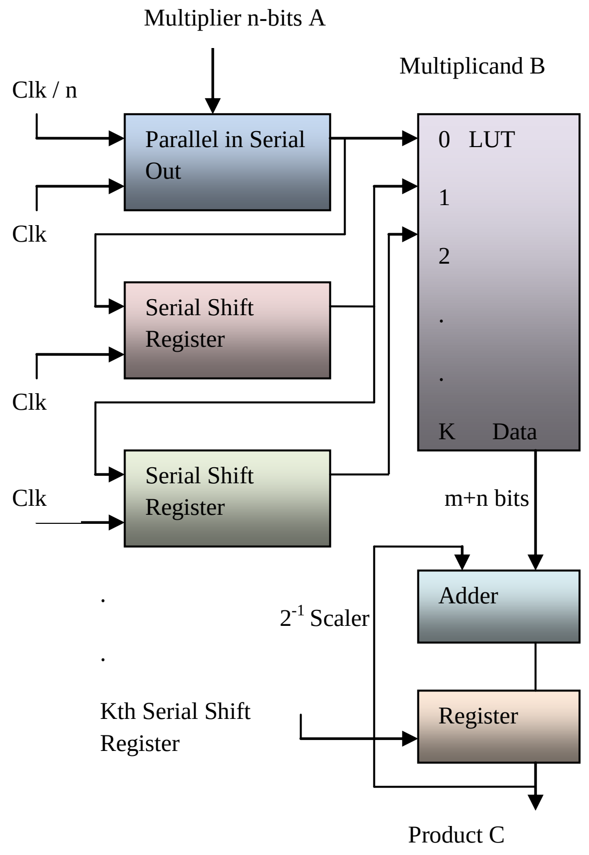

Figure 5: Multiplier design

The proposed design is tested on satellite image shown in figure 6 and edge detection is observed in figure 7.

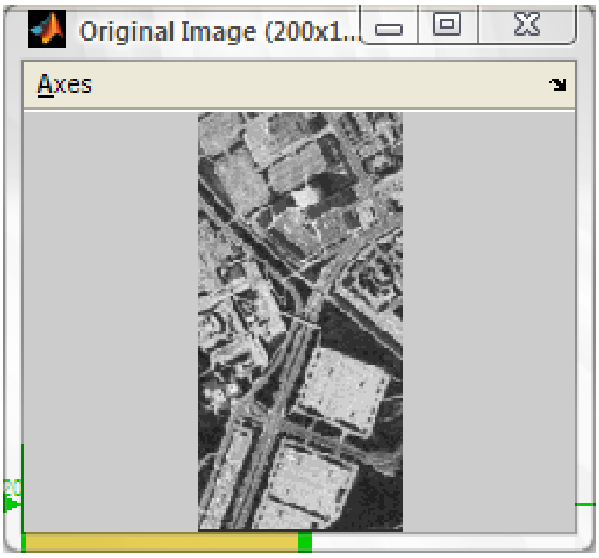

Figure 6: Original Image

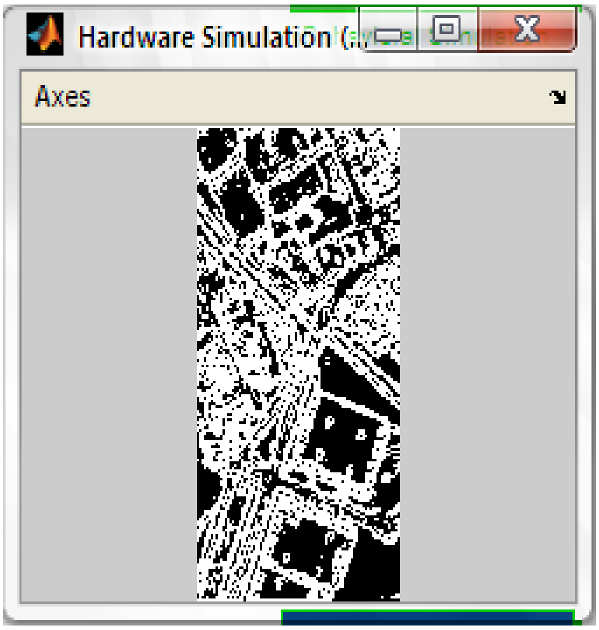

Figure 7: Edge detected output

5. HARDWARE IMPLEMENTATION RESULTS

The proposed sobel edge detection filter is implemented on Virtex II Pro based xc2vp30-7-FF896 FPGA target device. It is observed that the proposed design is working at a maximum speed of 148.33 MHz by consuming only 371 slices. Proposed model is using 353 look up tables, 482 slices flip flops, 126 shift registers and 14 bonded IOBs to implement the whole logic which is less than 5% of available hardware on FPGA. Implementation of proposed logic leads to reduction in number of resources utilization. Hardware used in any design determine the cost of design. The cost of proposed design is reduced due to lesser hardware utilization. The area utilization of the proposed design has been shown in Table 1 with percentage resource utilization.

Table 1: Proposed Design - Resource utilization

Resources	Used	Available	Utilization
Number of Slice Flip Flops	**482**	**27392**	**1%**
Number of occupied Slices	**371**	**13696**	**2%**
Total Number of 4 input LUTs	**353**	**27392**	**1%**
Number of bonded IOBs	**14**	**556**	**2%**
Number of BUFGMUXs	**1**	**16**	**6%**

Table 2 shows the proposed design has minimum period of 6.751ns and maximum frequency of 148.133MHz.

Table 2: Proposed Design - Speed Performance

Parameter	Value
Minimum Period	**6.751ns**
Maximum Frequency	**148.133MHz**

Total power consumption is relatively low to 0.10313 at 25 degree C as shown in table 3.

Table 3: Proposed Design - Power Consumption

Name	Value
Total Power	**0.10313 (W)**
Junction Temp	**25.0 (degree C)**

Proposed Sobel edge Design is compared with [5] in terms of hardware utilization shown in Table 4. Proposed design is using 371 slices which is almost one fifth of the slices used in [5]. Proposed design using 482 flip flops which is half of flip flops used [5]. Similarly proposed design using 353 4 input LUTs which is 10% of LUTs used in [5].

Table 4: Existing [5] Vs. Proposed Design Comparison

Resource Name	Existing [5] Parallel Architecture	Proposed Design
Number of Flip Flops	**836**	**482**
Number of occupied Slices	**1987**	**71**
Total Number of 4 input LUTs	**3901**	**353**

Comparison results show that Proposed model is saving more than five times hardware at the comparable speed of 6.751ns.

6. CONCLUSION AND FUTURE SCOPE

The proposed design can work by using lesser number of look up tables to produce cost effective solution for edge detection system. Xilinx DSP tool simulation and testing when implemented on Virtex 2P FPGA can work at an estimated frequency of 148.133 MHz and shows an improvement of one fifth hardware utilization over the existing design. Future scope can be extended to achieve more improvement in terms of speed by using partial serial architecture.

7. ACKNOWLEDGMENTS

Authors would like to convey thanks to Dr. SBL Sachan, Professor and Head, Dr. Swapna Devi, Associate Professor ECE Department, NITTTR Chandigarh for their constant encouragement and guidance during this research work.

8. REFERENCES

[1] D.J. Allred, H. Yoo, V. Krishnan, W. Huang, and D. Anderson, “A Novel High Performance Distributed Arithmetic Adaptive Filter Implementation on an FPGA”, Proceeding IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP’04), Vol. 5, pp. 161-164, 2004.

[2] I.Yasri*, N.H.Hamid, V.V.Yap., “Performance Analysis of FPGA Based Sobel Edge Detection Operator”, IEEE International Conference on Electronic Design, pp. 1 – 4, 2008.

[3] KC Sudeep, Dr. Jharna Majumdar, “A Novel Architecture for Real Time Implementation of Edge Detectors on FPGA”, IJCSI International Journal of Computer Science, Volume 8, Issue 1, pp. 193-202, January 2011.

[4] O. R. Vincent, O. Folorunso, A, “A Descriptive Algorithm for Sobel Image Edge Detection”, Proceedings of Informing Science & IT Education Conference, pp. 97-107, 2009.

[5] Varun Sanduja, Rajeev Patial, “Sobel Edge Detection Using Parallel Architecture based on FPGA”, International Journal of Applied Information Systems (IJAIS), Vol. 3, No. 4, pp. 20-24, July 2012.

[6] Pei-Yung Hsiao1, Le-Tien Li1, Chia-Hsiung Chen2, Szi-Wen Chen1, and Sao-Jie Chen2L, “An FPGA Architecture Design of Parameter-Adaptive Real-Time Image Processing”, IEEE, pp. 1-3, 2005.

[7] Arash Nosrat, Yousef S Kavian, “Hardware Description of multi-Directional Fast Sobel Edge Detection Processor by VHDL for implementing on FPGA”,International Journal of Computer Applications, Volume 47-No. 25, pp. 1-7, June 2012.

[8] Zhang Jin-Yu; Chen Yan; Huang Xian-Xiang, “Edge detection of images based on improved Sobel operator and genetic algorithms”, International conference on Image Analysis and Signal Processing, pp. 31 – 35, 2009.

[9] K.N. Macpherson and R.W., “Area efficient FIR filters for high speed FPGA Implementation”, IEEE Proc.-Vis. Image Signal Process, Vol. 153, No 6, pp.711-720, 2006.

[9] T.A Abbasi, M.U Abbasi, “A Novel FPGA-based architecture for Sobel edge detection operator”, International Journal of Electronics, Volume 94, Issue 9, pp. 889-896, September 2007.

[10] Ehsan Nadernejad, “Edge Detection Techniques: Evaluations and Comparisons”, Applied Mathematical Sciences, Vol. 2, pp. 1507-1520, 2008.

[11] Raman Maini, Dr. Himanshu Aggarwal, “Study and Comparison of Various Image Edge Detection Techniques”, International Journal of Image processing (IJIP), Vol 3, Issue 1, pp.1-12, March 2009.

[12] C. Pradabpet, N. Ravinu, S. Chivapreecha, B. Knobnob, K. Dejhan, “An efficient filter structure for multiplierless Sobel edge detection”, Innovative Technologies in Intelligent Systems and Industrial Applications, pp. 40 – 44, 2009.

[13] Patrick Longa and Ali Miri, “Area-Efficient FIR Filter Design on FPGAs using Distributed Arithmetic”, IEEE International Symposium on Signal Processing and Information Technology, pp. 248-252, 2006.

[14] Kolb Jan, Honzik Petr, Danek Martin, “Reconfigurable hardware objects for image processing on FPGAs”, 13th International Symposium on Design and Diagnostics of Electronic Circuits and Systems, pp. 121 – 122, (DDECS), 2010.