# Design of Sobel Operator based Image Edge Detection Algorithm on FPGA

Girish Chaple, R. D. Daruwala, *Member, IEEE*

***Abstract*—Real-time image processing applications requires processing on large data of pixels in a given timing constraints. Reconfigurable device like FPGAs have emerged as promising solutions for reducing execution times by deploying parallelism techniques in image processing algorithms. Implementation of highly parallel system architecture, parallel access of large internal memory banks and optimization of processing element for applications makes FPGA an ideal device for image processing system. Edge detection is basic tool used in many image processing applications for extracting information from image. Sobel edge detection is gradient based edge detection method used to find edge pixels in image. This paper presents a design of a Sobel edge detection algorithm to find edge pixels in gray scale image. Xilinx ISE Design Suite-14 software platforms is used to design a algorithm using VHDL language. MATLAB software platform is used for obtaining pixel data matrix from gray scale image and vice versa. Xilinx FPGAs of family Vertex-5 are more suitable for image processing work than Spartan-3 and Spartan-6.**

***Index Terms*—Edge detection, Sobel operator, FPGA, VHDL, test bench, real-time image processing.**

## I. Introduction

Image processing has applications in intelligent transport system, autonomous vehicles and self guided armaments etc. An image can be represented as two dimensional function $f(x,y)$ where $x$ and $y$ are spatial co-ordinates; magnitude of function $f$ gives intensity or gray level value at that point [1]. At edge pixels intensity of function $f(x,y)$ changes sharply; connected edge pixels are called edges. A process of identifying sharp changes in pixels values with respect to neighboring pixels values is edge detection. Sobel edge detection is first order derivative based method because it is computed using digital gradient of image**.** Sobel and Prewitt operator are used extensively for edge detection in the image processing [2]. In gray scale image each pixel is represented by 8 bit; hence, gray level values vary from 0 to 255 where 0 value stands for black color and 255 value stands for white color. FPGA contains an array of programmable logic element; these elements can be programmed for DSP functions [3]. FPGAs have large number of internal memory banks which can be accessed in parallel that allowed FPGA hardware to execute functions in a few clock cycles whereas sequential operational processor required hundreds to thousands of clock cycles [4].

FPGAs operate on low operational frequency; the lower clock speeds result in low power consumption hence FPGAs are more power efficient [4]. Use of FPGAs in image processing systems enables rapid prototyping, minimizes the time to market cost. The rest of the paper is organized as follow; Section II describes theoretical concepts of edge detection in Image processing. Section III describes operation of two modules Sobel edge detection operation and Valid 3×3 pixel data generation. Section IV describes test design flow to test unit under test. Section V describes experimental results and hardware resources utilization by program for Xilinx FPGA devices of family Spartan-3, Spartan-6 and Vertex-5.

## II. Theoretical Concepts

An image is a spatial representation of an object. Real-time imaging processing system is system that captures images, analyses those images to obtain desired data and then uses that data for controlling purpose. The edges contain high frequency information; high pass filter can be used to detect edges. Edges characterized by a difference in pixel values. In real world image edges are blurred and hence have ramp profile; first order derivative is zero at constant gray level and non zero value along entire ramp region therefore magnitude of first order derivative can be used to detect an edge.

| | | |
|---|---|---|
| Z1 | Z2 | Z3 |
| Z4 | Z5 | Z6 |
| Z7 | Z8 | Z9 |

(a)

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

(b)

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

(c)

Fig. 1. (a) 3x3 region of an image. (b) $0^0$ Sobel kernel. (c) $90^0$ Sobel kernel

Gradient of f(x,y) at co-ordinates $(x,y)$ is defined as two dimensional column vectors pointing to direction of greatest rate of change $f$ at that location [1].

$$\nabla f = grad(f) = \begin{bmatrix} G_x \\ G_y \end{bmatrix} \quad (1)$$

Sobel edge detection is gradient based edge detection method. Sobel operator is linear operator; $0^0$ convolution kernel moved pixel by pixel and line by line across image to detect edge in x-direction. Similarly $90^0$ convolution kernel moved pixel by pixel and line by line across image to detect edge in y-

Girish Chaple is with Department of Electrical Engineering, Veermata Jijabai Technological Institute, Mumbai, India.
(e-mail: gnc.girish@gmail.com).
Dr. R. D. Daruwala is with Electrical Engineering Department, Veermata Jijabai Technological Institute, Mumbai, India.
(e-mail: rddaruwala@vjti.org.in).

978-1-4799-3357-0

direction. $0^0$ convolution kernel convolved with input image gives gradient component $G_x$, $90^0$ convolution kernel convolved with input image gives gradient component $G_y$ [5].

$$G_x = (Z_7 + 2 * Z_8 + Z_9) - (Z_1 + 2 * Z_2 + Z_3) \quad (2)$$

$$G_y = (Z_3 + 2 * Z_6 + Z_9) - (Z_1 + 2 * Z_4 + Z_7) \quad (3)$$

Magnitude of vector $\nabla f$ can be defined as

$$mag(\nabla f) = \sqrt{{G_x}^2 + {G_y}^2} \quad (4)$$

For faster computation Equation (4) is approximated as

$$mag(\nabla f) \approx |G_x| + |G_y| \quad (5)$$

This expression still preserves relative changes in intensity. window of size $3 \times 3$ is used in this paper because larger the size of window requires more number of FIFO for accumulation of row, number of mathematical stages increases and hence increase in Hardware resources of FPGA.

## III. SOBEL EDGE DETECTION SYSTEM ARCHITECTURE

Xilinx ISE Design Suite-14 software platforms are used for design Sobel edge detection algorithm; VHDL language used for writing algorithm. Gray scale Image of size $m \times n$ is input to system; edge detected gray scale image of same size obtained at output. Fig. 2 illustrates edge detection architecture system which contains two main modules $3 \times 3$ pixel generation and Sobel edge detection operation. 8 bit pixel value of input image is given as input to $3 \times 3$ pixel

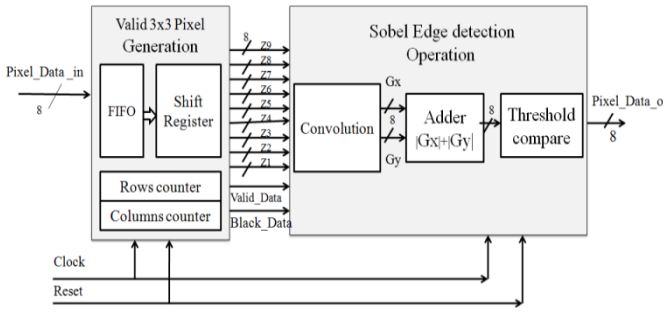


Fig. 2. Sobel edge detection Architecture

generation module. This module contains two FIFO and three shift registers blocks [6]. Sobel edge detection module contains convolution, addition and threshold comparison blocks. Modules are synchronies with input clock to system. When reset is low system go to reset position.

### A. *Valid 3×3 pixel data generation:*

Fig. 3 illustrates memory architecture. Mask of size $3 \times 3$ is used in this paper therefore two FIFO memory array and three shift registers are used for accumulation of valid input pixel data for processing. Design FIFO memory array has first in first out policy; each FIFO memory array accumulates number of pixels in one row of input image [6]. At every rising edge of clock input pixel data is stored, take out from FIFO and pass through shift register.

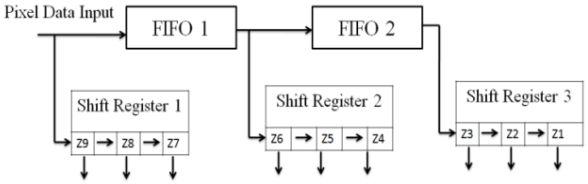


Fig. 3. Memory Architecture

Fig. 4 shows schematic of valid $3 \times 3$ pixel data generation module. Fig. 5 shows simulation of valid $3 \times 3$ pixel data generation module for image of size $8 \times 8$ whose data matrix as shown in Fig. 6. *valid_data* signal goes high when all elements of all shift registers get filled with valid input pixel data otherwise remains low.

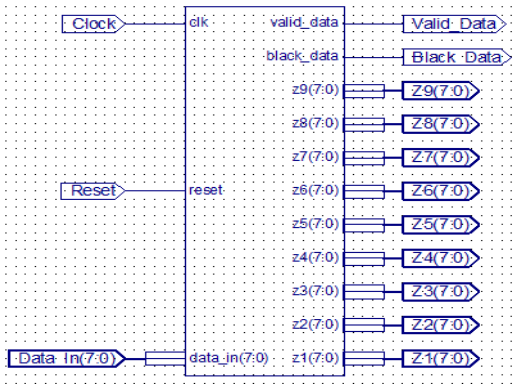


Fig. 4. Schematic of valid 3x3 pixel generation

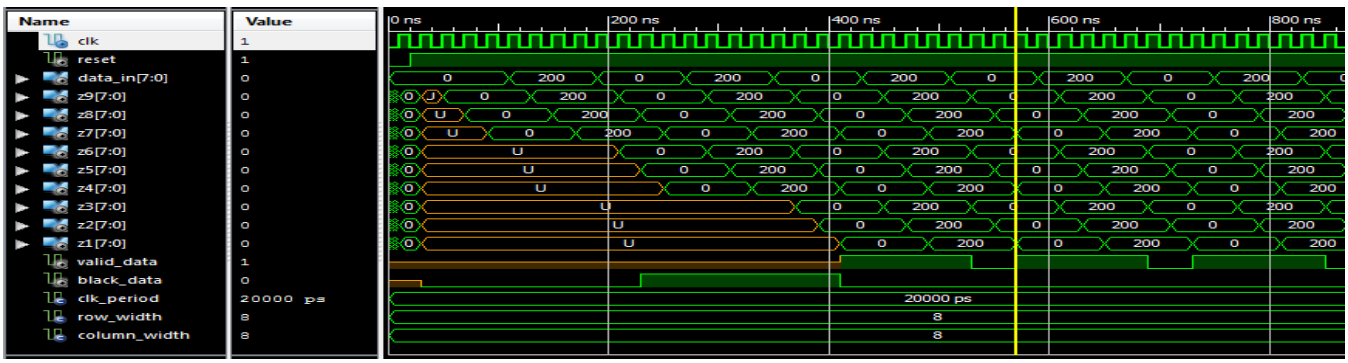


Fig. 5. Simulation result of valid 3x3 pixel generation module

*black_data* signal is responsible for filling 0 gray values at borders of image so that output image is of same size as that of the input image size. Row counter and column counter are used to determine when *valid_data* signal go high or low. It also determines boarders of input image. When *valid_data* signal is high, valid $3 \times 3$ input pixel data is available for convolution operation.

```
0 0 0 0 200 200 200 200
0 0 0 0 200 200 200 200
0 0 0 0 200 200 200 200
0 0 0 0 200 200 200 200
0 0 0 0 200 200 200 200
0 0 0 0 200 200 200 200
0 0 0 0 200 200 200 200
0 0 0 0 200 200 200 200
```

Fig. 6. Gray scale image data matrix of size 8x8.

### *B. Sobel edge detection operation:*

Fig. 7 shows Schematic of Sobel edge detection module. Sobel edge detection module is responsible for convolution, addition, threshold comparison. Convolution operation uses multipliers, adders, dividers to calculate output. For real-time uses system should be fast, more number of mathematical computations tends to slow down FPGA performance. Hence while developing algorithm more attention given to implement mathematical operation in different process statements because all process statements are executed in parallel [7]. Multiplication operation replaced by left shift of a bit vector

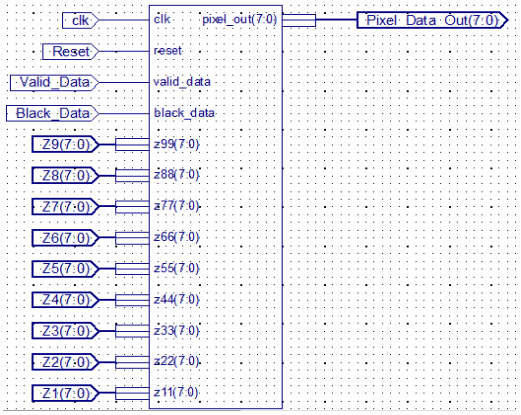


Fig. 7. Schematic of Sobel edge detection module

by one bit. $G_x$, $G_y$ obtained by convolving $0^0$, $90^0$ Sobel kernel with input image. Output of convolution may be positive or negative. Absolute value of negative value bit vector obtained by taking its 2's complement. If *valid_data* signal is high and $|G_x| + |G_y|$ is greater than threshold value then *pixel_data_out* is 255 else *pixel_data_out* is 0. If black_data signal is high then *pixel_data_out* is 0 values. By using the threshold the effect of noise in the extracted features can be reduced [8]. Fig. 8 shows simulation result of Sobel edge detection module for input gray image data matrix as shown in Fig. 6.

## IV. TEST BENCH

To test correctness of VHDL code a special type of code called test bench code is used. Xilinx provide feature called simulation which takes valid input and shows how it would work in hardware. To simulate design, both the design under test (DUT) or unit under test (UUT) and the stimulus provided by the test bench are required [9]. The test bench is usually written in the same language (VHDL or Verilog) than entity under test. Test bench has three main purposes to generate stimulus for simulation, to apply this stimulus to the entity under test and to collect output responses. In VHDL test bench it is possible to read data from text file and write data in to text file. To obtained raw image data for processing, MATLAB software is used. Program written in MATLAB writes pixel values of gray scale image into text file. Input text file contains data matrix of size $m \times n$ obtained from $m \times n$ size gray scale image using MATLAB software. Data written in text file is of type integer. Program written in test bench read pixel data from input text file and convert integer values to bit vector of 8 bit length which given as input to unit under test for edge detection process. Output of unit under test i.e. edge detection process is bit vector of length 8 bits, which again converted in to integer type. Output of simulation is collected in output text file in form of edge detected data matrix which is of type integer. For converting data matrix to gray scale image, content from output text file is loaded to MATLAB software to obtained image from edge detected data matrix. Fig. 9 shows a test design flow to check correctness of written VHDL code for edge detection at software platform.

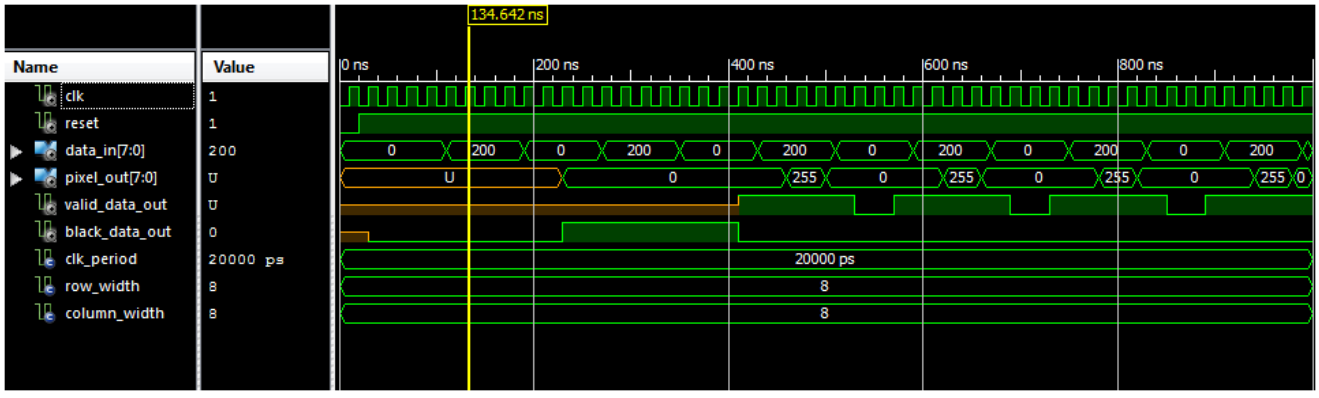


Fig. 8. Simulation result of Sobel edge detection module

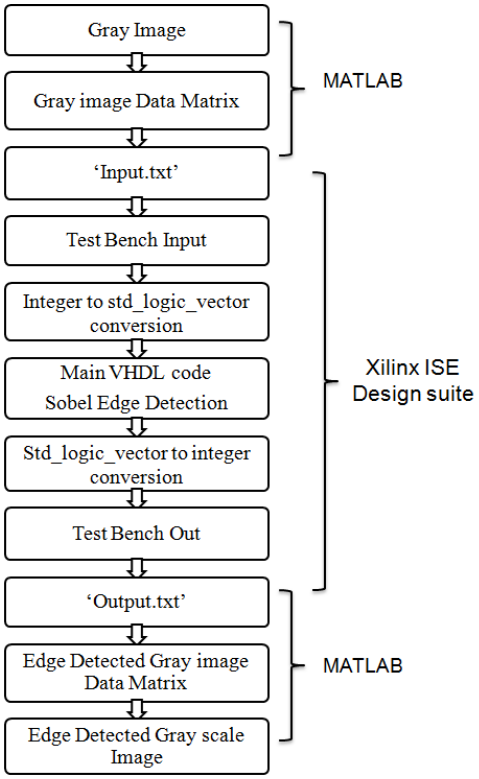


Fig. 9. Test Design Flow.

## V. EXPERIMENTAL RESULT

Gray scale image of size 640×480 is used for experiment where number of rows are 480 and number of columns are 640. Xilinx FPGA device XC5VLX50 of family Vertex-5, XC6SLX25 of family Spartan-6, XC3S400 of family Spartan-3 is used as target device. To meet a real-time requirements; processing on single frame should be completed in desire time. For typical CMOS digital sensor camera maximum frame rate is 30 frame per second and maximum pixel clock frequency for which pixel data outs are valid is 48 MHz [10]. Processing on single frame completed in 7.696 ms, 6.412 ms for pixel clock frequency of 40MHz, 48MHz respectively thus meet real-time requirement. Edge detected image in x-direction is obtained when $0^0$ Sobel kernel convolved with entire input image pixel data and after comparing with threshold value. Similarly edge detected image in y-direction is obtained when $90^0$ Sobel kernel convolved with entire input image pixel data and after comparing with threshold value.

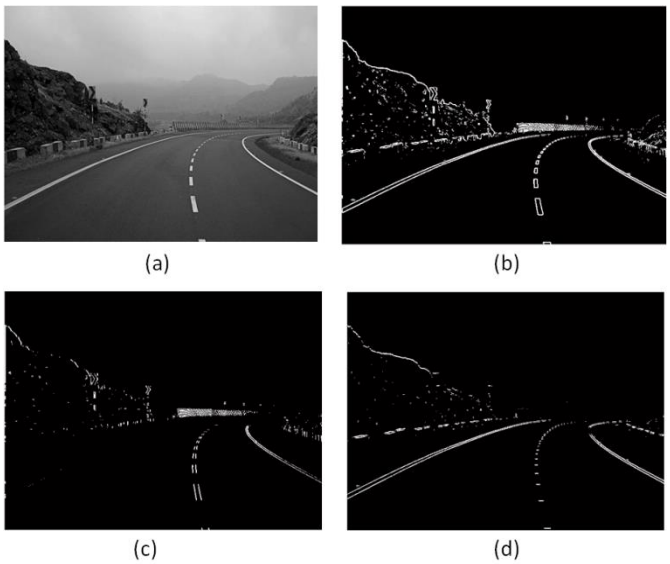


Fig. 10. Experiment Result. (a) Original image. (b) Edge detected image at threshold value 260. (c)Image edge detected in x-direction at threshold value 260. (d) Image edge detected in y-direction at threshold value 260.

Table I, II, III shows hardware resources required for implementing the edge detection algorithm to process over an image of size 640×480. New generation FPGAs such as vertex-5 contains DSP48E slices supports 25-bit × 18-bit multipliers, 3 input 48-bit adders enable efficient adder-chain architectures for implementing high-performance filters and complex math efficiently therefore Xilinx FPGAs from Vertex-5 family are more suitable than Spartan-3 and Spartan-6 for implementing image processing algorithms [11].

TABLE I. HARDWARE RESOURCES UTILISATION SUMMARY OF SPARTAN-3 DEVICE XC3S400

| Slice Logic Utilization | Used | Available | Utilization |
|---|---|---|---|
| *Number of occupied slices* | 1,685 | 3,584 | 47% |
| *Number of 4 input LUTs* | 3,166 | 7,168 | 44% |
| *Number of Slice Flip Flops* | 339 | 7,168 | 4% |
| *Number of bonded IOBs* | 97 | 141 | 68% |
| *Number of BUFGMUXs* | 2 | 8 | 25% |

TABLE II. HARDWARE RESOURCES UTILISATION SUMMARY OF SPARTAN-6 DEVICE XC6SLX25

| Slice Logic Utilization | Used | Available | Utilization |
|---|---|---|---|
| *Number of occupied Slices* | 491 | 3,758 | 13% |
| *Number of Slice Registers* | 451 | 30,064 | 1% |
| *Number of Slice LUTs* | 1,680 | 15,032 | 11% |
| *Number used as logic* | 469 | 15,032 | 3% |
| *Number used as Memory* | 1,210 | 3,664 | 33% |

TABLE III. HARDWARE RESOURCES UTILISATION SUMMARY OF VERTEX-5 DEVICE XC5VLX50

| Slice Logic Utilization | Used | Available | Utilization |
|---|---|---|---|
| *Number of occupied slices* | 515 | 7,200 | 7% |
| *Number of Slice Register* | 339 | 28800 | 1% |
| *Number of Slice LUTs* | 1,927 | 28,800 | 6% |
| *Number used as logic* | 623 | 28,800 | 2% |
| *Number used as memory* | 1,303 | 7,680 | 16% |

## VI. CONCLUSIONS

Image processing algorithm for image edge detection using Sobel operator is successfully designed on target FPGAs. Developed algorithm takes advantages of parallelism possible

in FPGA. The Xilinx Vertex-5 device is well suited for implementing imaging functions such as edge detection. The FPGA supports high levels of parallel processing data flow structures that are important for efficient implementation of image processing algorithms. Reconfigurable device like FPGA can be a very useful device to implement real-time image processing algorithm. Uses of complex mathematical operations are avoided to achieve real-time timing constraints. Drawback of proposed method is that detected edges are thicker. The algorithm designed in this paper is basic image processing algorithm, this work can be further extend to developed real-time complex image processing algorithms such as feature detection, feature extraction, target tracking.

## REFERENCES


[1] R. C. Gonzalez, R. E. Woods, *Digital Image Processing*. 3rd ed., Prentice Hall, 2007, pp. 187-190.

[2] B. Kaur, A. Garg, "Mathematical morphological edge detection for remote sensing images," *3rd International Conference on Electronics Computer Technology* (ICECT), vol. 5, pp. 324-327, Apr. 2011.

[3] D. C. M. Bilsby, R. L. Walke, R W M Smith, "Comparison of a programmable DSP and FPGA for real-time multiscale convolution," *IEE Colloquium on High Performance Architectures for Real-Time Image Processing*, pp. 4/1- 4/6, Feb. 1998.

[4] *Accelerating High-Performance Computing With FPGAs*, Altera Corporation, 2007.

[5] D. Alghurair, S. S. Al-Rawi, "Design of Sobel Operator using Field Programmable Gate Array," in *Proc. International Conference on Technological Advances in Electrical, Electronics and Computer Engineering* (TAEECE), pp. 589-594, May 2013.

[6] Z. Guo,W. Xu, Z. Chai,"Image Edge Detection Based on FPGA," in *Proc. Ninth International Symposium on Distributed Computing and Application to Business Engineering and Science* (DCABES), pp. 169-171, Aug. 2010.

[7] J.Bhasker, *A VHDL Primer*. 3rd ed., PHI Learning, 2008, pp. 71-72.

[8] R. Harinarayan, R. Pannerselvam, M. M. Ali, D. K. Tripathi, "Feature Extraction of Digital Aerial Images by FPGA based implementation of edge detection algorithms," in *Proc. International Conference on Emerging Trends in Electrical and computer Technology* (ICETECT), pp. 631-635, Mar. 2011.

[9] *Test benches* [online]. Available: http://www.xilinx.com/itp/xilinx10/isehelp/ise_c_simulation_test_bench.htm

[10] *½-Inch Megapixel CMOS Digital Image Sensor MT9M001C12STM (Monochrome) Data sheet*, Micron Technology, Inc.,2004.

[11] *XtremeDSP 48 Slice* [online]. Available: http://www.xilinx.com/technology/dsp/xtremedsp.htm