MANTECH Publications

Journal of Current Trends in Electrical Engineering
Volume 2, Issue 2, May-August, 2017

Implementation of Edge Detection Filter Using FPGA

Shraddha Y. Swami, Jayashree S. Awati

Department of Electronics and Telecommunication Engineering

Rajarambapu Institute of Technology, Rajaramnagar, Sangli (M.S.), India

Corresponding Authors' Emails: *aswamisy@gmail.com, jayashree.samai@ritindia.edu*

Abstract

Reconfigurable device like FPGA, are capable of reducing execution times by making use of 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 different applications makes FPGA an ideal for image processing applications .Edge detection in an image processing helps to extract information from an image because edges in an image are consisting of meaningful features. So, significant information of an image can be extracted from edges. Sobel edge detection is gradient based edge detection method. Xilinx Spartan 3E FPGA kit is used to realize the Sobel edge detection filter. In VLSI area, power and delay are important design factors. These three factors depend upon each and every operation performing in an FPGA. Arithmetic and logic operations play an important role in digital circuits. A systems performance is generally determined by the performance of the multiplier because the multiplier is generally the slowest element in the system. Furthermore, it is generally the most area consuming. Hence optimizing the speed and area of the multiplier is a major design issue. This paper presents the implementation of Sobel Edge detection filter using Normal multiplier, Shift-Add multiplier and radix-4 Booth Multiplier and comparison of their results.

Keywords: *Edge detection, Field Programmable Gate Array (FPGA), Sobel operator, VHDL, Radix-4 Booth Multiplier, Shift-Add Multiplier*

I. INTRODUCTION

An image is represented as two dimensional function f(x, y) where x and y are called spatial co-ordinates; magnitude of function f gives intensity level value at corresponding point. If intensity of function f(x, y) changes sharply that will be called as an edge pixels and connecting edge pixels forms an edges. Edge detection is a process to identify sharp changes in image pixel values with respect to neighbouring pixels values. Sobel edge detection is first order derivative based method because it computes using digital gradient of image. In gray scale image each pixel is represented by 8 bit which means intensity values vary from 0 to 255 where 0 stands for black color and 255 stands for white color. FPGA consisting of an array of programmable logic elements; these elements can be programmed for DSP functions. FPGAs have large number of internal memory banks which can be accessed parallel that allows FPGA hardware to execute functions in a few clock cycles.

This paper describes implementation of Sobel edge detection algorithm using VHDL language and implementation of two hardware optimization approaches i.e. Shift-Add multiplier and Radix-4 Booth multiplier for the same and their realization on Xilinx Spartan-3E FPGA. Ami J. Shukla, Prof. Vibha Patel, Prof. Nagendra Gajjar provided various algorithms for edge detection. Authors revealed that to carry out extreme computations involved in real time image processing a system which is really fast is required. They addressed that high amount of computation power in limited time can be achieved by using FPGA as a platform. They addressed to use the technology with huge amount of parallelism i.e. FPGA. [1]Aneela Pathan,Tayab D Memon presented an implementation of fixed point finite length 3x3 unsigned integer shift add multiplier. They proposed design which results in less resource utilizations.[2] Girish N. Chapale, R.D. Daruwala presented the design of Sobel Edge detection algorithm to find edge pixels in gray scale image. They designed Image processing algorithm for image edge detection using Sobel operator on target FPGAs. They implemented algorithms for edge detection using VHDL and MATLAB software for obtaining data matrix from gray scale image and vice-versa. [4] Khuraijam Nelson Singh, H. Tarunkumar reviewed study on various multipliers. They addressed that array multiplier is found to have largest delay and large power consumption.

Authors reviewed that delay required for Booth encoded Wallace tree multiplier is least. They revealed that the performance of multiplier can be increased with proper optimization.[5]Manoj Sharma, Richa Verma addressed the issue of disposal of the negative partial products efficiently by computing the 2's complement. They proposed mechanism to support the concept of reducing the partial product. Hence reducing area.[7]

II. THEORETICAL CONCEPTS

An image is a spatial representation of object. Real – time image processing system captures images, analyses those images to obtain desired information. The desired information is then used for controlling purpose. In an image each pixel is having its unique intensity or pixel value. The differences in pixel value with respect to neighbouring pixel value are called edges. Sudden changes of discontinuity in an image pixel values forms edges. Edges consist of meaningful features and contain significant information. In Real world edges of an image can be blurred and hence have ramp profile; as first order derivative is zero at constant gray level and nonzero at entire ramp region therefore magnitude of first order derivative can be used for detecting an edge in images.

Sobel edge detection method is first order derivative based method and it calculates edges in both horizontal and vertical directions. Sobel operator detects edges in an image using two kernels. One is horizontal mask and other is vertical mask they can also be called 0° convolution kernel and 90° convolution kernel respectively.

Z1	Z2	Z3
Z4	Z5	Z6
Z7	Z8	Z9

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

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

Fig.1 (a) 3x3 region of an image (b) 0° Sobel Kernel (c) 90° 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 is,

$$\nabla f = \begin{bmatrix} Gx \\ Gy \end{bmatrix} = \begin{bmatrix} \frac{df}{dx} \\ \frac{df}{dy} \end{bmatrix} \quad (1)$$

$$Gx = (Z7 + 2 * Z8 + Z9) - (Z1 + 2 * Z2 + Z3) \quad (2)$$

$$Gy = (Z3 + 2 * Z6 + Z9) - (Z1 + 2 * Z4 + Z7) \quad (3)$$

The magnitude of this vector is given by,

$$\nabla f = \text{Mag}(\nabla f) = [\, Gx^2 + Gy^2 \,]^{1/2} \quad (4)$$

$$= \left[\left(\frac{df}{dx}\right)^2 + \left(\frac{df}{dy}\right)^2\right] \quad (5)$$

The computation of implementing Eq. (4) over an entire image is not trivial, and it is common practice to approximate the magnitude of the gradient by using absolute values instead of squares and square roots:

$$\text{Mag}(\nabla f) \approx |Gx| + |Gy| \quad (6)$$

These are the steps involved in Sobel Edge Detection algorithm.

III. HARDWARE IMPLEMENTATION

A. Simple Implementation: The fig. 2 describes entire block diagram for the realization of Sobel edge detection algorithm on Spartan-3E FPGA. An input image is taken and transferred to FPGA serially using serial cable. MATLAB software and serial cable is used for transmission and reception of input and output image data. LCD Display is used to display transmitting and receiving values. Sobel edge detection algorithm is implemented in VHDL language and FPGA is programmed using JTAG programmer.

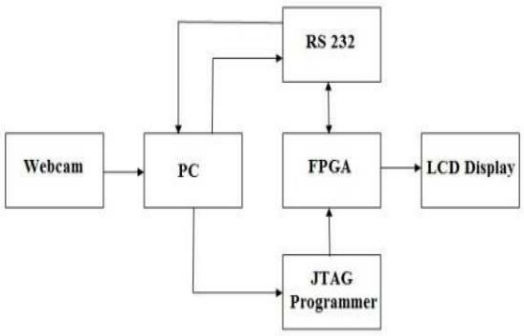

Fig.2: Block Diagram of Proposed Work

Implemented algorithm is applied on input image data and output image data is transmitted serially from FPGA to PC and displayed using MATLAB software.

Xilinx integrated system environment design suite software platforms are used for writing edge detection algorithm in VHDL language. Edge detection system architecture is mainly divided into two module first is accumulation of data in

first in first out register memory. Second module is edge detection operation module and used for performing convolution, addition, threshold comparing operation on gathered pixel data. Sobel edge detection module is responsible for convolution, addition, threshold comparison.

Convolution operation uses multipliers, adders, dividers to calculate output. For real time applications system should be fast as more number of mathematical calculations makes an FPGA slower. Hence, while developing algorithm more attention is required on mathematical computation. Gx and Gy obtained by convolving 0º, 90º Sobel Kernel with input image. Output of convolution may be positive or negative. To reduce mathematical computations eq. 5 is approximated to eq. 6.

B. Hardware optimization using Shift – Add Multiplier:

Important optimizations of the speed, area, and power consumption of circuits can be achieved by using dedicated operators instead of general ones whenever possible. Multiplication is an important operation in arithmetic and logical operations. In case of normal multiplication number of resources required is more because number of partial products is more.

Algorithm:

- **Step 1:** Initially, the product register (P) of two 8 bit binary numbers is initialized as ‘0’.
- **Step 2:** Check all the positions of 8 bit multiplier from LSB to MSB. If that bit =1 then shift multiplicand by the number of bits left at which position the multiplier has value ‘1’.
- Add the shifted value to the Product register.
- This procedure is repeated until all the bits of multiplier are covered.
- The final accumulated product will be achieved in few steps.

This multiplication method saves area and reduces no. of partial products.

RTL schematic of Shift-Add Multiplier:

Fig.3 RTL schematic of Shift-Add multiplier

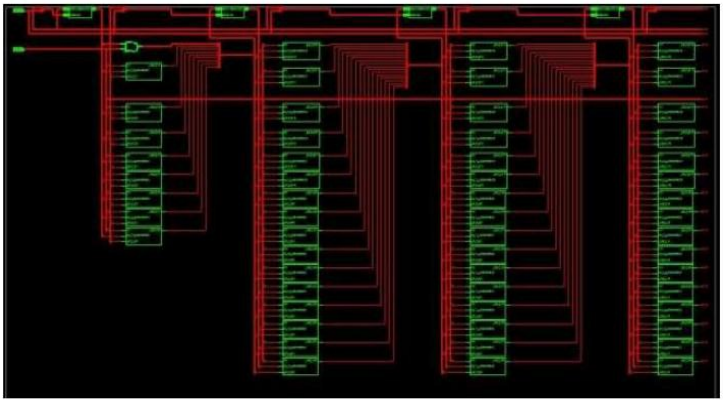

Fig.4 Internal RTL schematic of Shift-Add multiplier part 1

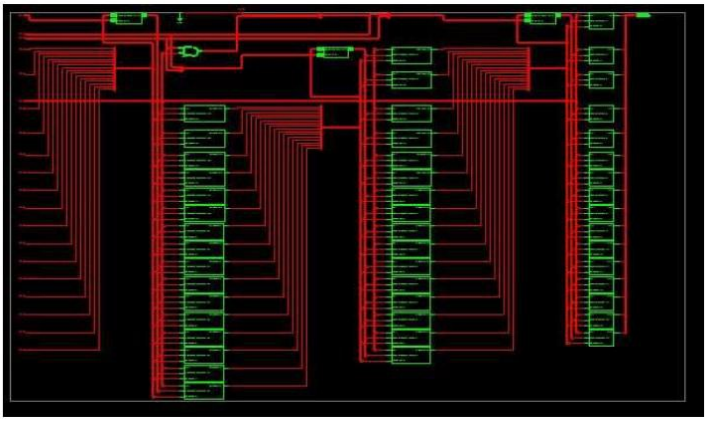

Fig.5 Internal RTL schematic of Shift-Add multiplier part 2

B. Hardware optimization using Radix-4 Booth Multiplier:

Modified Booth multiplier (Radix-4) is used to reduce partial products as it also affects area. It is used to perform high-speed multiplications using modified booth algorithm. It can reduce half the number of partial products. Radix-4 booth algorithm used here increases the speed of the multiplier. And area of the multiplier circuit is reduced using Radix-4 booth multiplier.

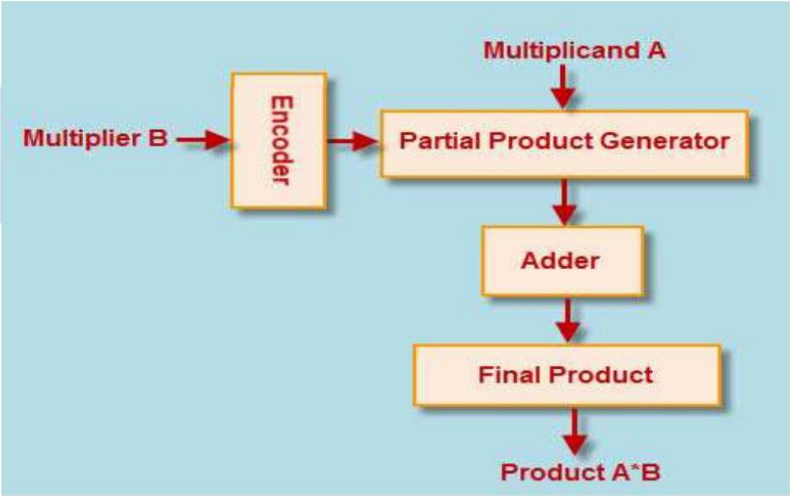

Fig.6 Flow diagram of Booth Algorithm

In this algorithm, every second column is taken and multiplied by 0 or +1or +2 or -1 or -2 instead of multiplying with 0 or 1 after shifting and adding of every column of the multiplier.

Fig.7 Bit pairing as per Booth recoding

Based on the multiplier bits, the process of encoding the multiplicand is performed by radix-4 booth encoder. The overlapping is used for comparing 3 bits at a time. The grouping is started from LSB least Significant Bit, in which only 2 bits are used by the first block and a zero is assumed as third bit as shown in the figure 7.

Table I describes the recoding table for Radix-4 Booth Multiplier.

Table I Radix -4 Booth Recoding

Multiplier bits			Recoded multiplicand, X
Y_2	Y_1	Y_0	
0	0	0	0*X
0	0	1	1*X
0	1	0	1*X
0	1	1	2*X
1	0	0	-2*X
1	0	1	-1*X
1	1	0	-1*X
1	1	1	0*X

RTL schematic of Radix-4 booth multiplier:

Fig.8 RTL schematic of Radix-4 Booth Multiplier

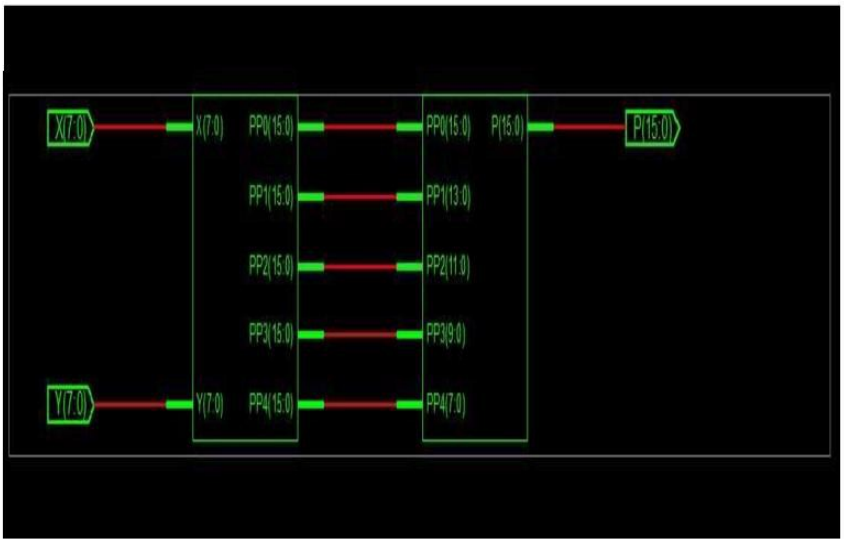

Fig.9 Internal RTL schematic of Radix-4 Booth Multiplier

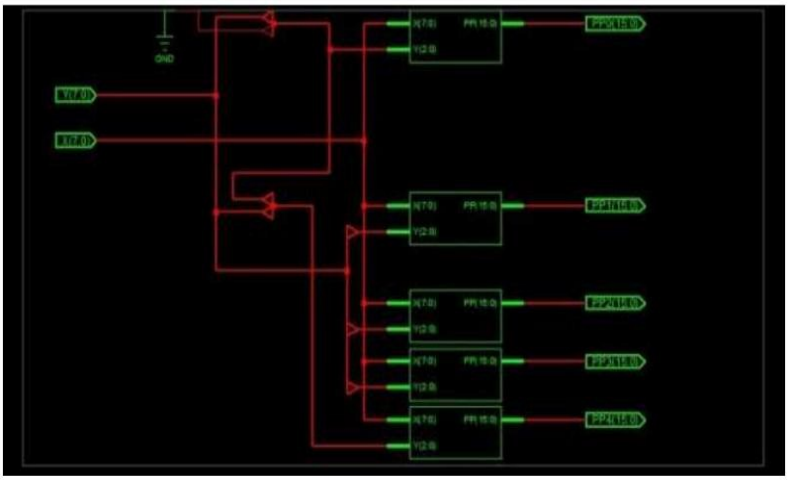

Fig.10 RTL schematic of Radix-4 Booth Encoding

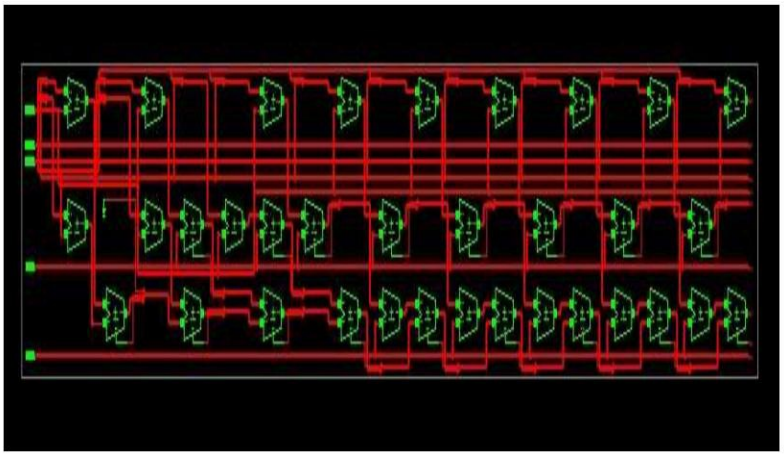

Fig.11 RTL schematic of Radix-4 Adder Tree part 1

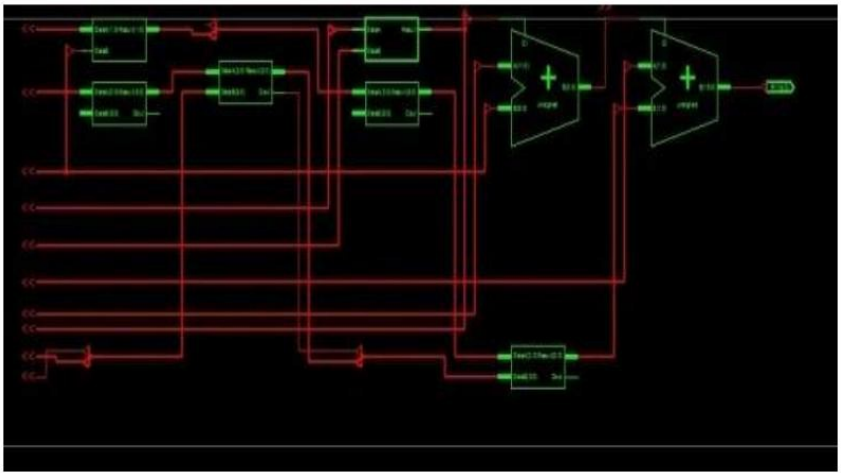

Fig.12 RTL schematic of Radix-4 Adder Tree part 2

V. RESULTS

A. Simulation Results:

The design of Sobel edge detection algorithm is described using VHDL/Verilog. Simulation results of Sobel edge detection algorithm are observed using MATLAB and Modelsim software.

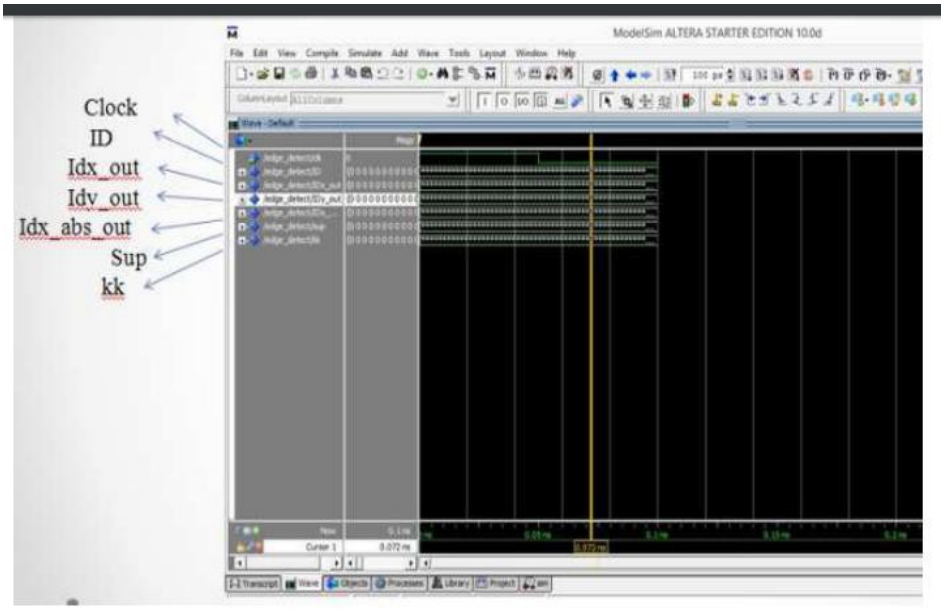

Fig.13 Simulation waveform

Fig.13 shows the simulation result of Sobel edge detection algorithm in Modelsim. In simulation waveform ID, Idx_out, Idy_out, Idx_abs_out, sup, kk are signals used in vhdl program designed for Sobel filter. Input image data is stored in an array ID. Idx_out and Idy_out are used to store xgradient and y-gradient outputs respectively.

The signals sup and kk are used for suppression and thresholding purpose. Magnitude of gradient is stored in Idx_abs_out.

Fig 14 shows the edge detected image using MATLAB.

B. Hardware Results:

The design for Sobel edge detection filter is implemented in VHDL language and is realized using Xilinx Spartan-3E XC3S100E FPGA device. VHDL programs implemented for hardware realization of Sobel filter are synthesized in Xilinx ISE tool. Fig. 15 shows the Hardware setup for realization of Sobel Edge Detection Filter on FPGA.

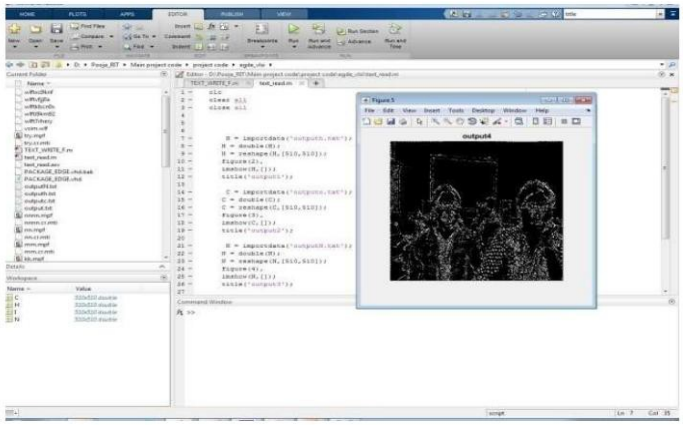

Fig. 14 Edge detected (output) image

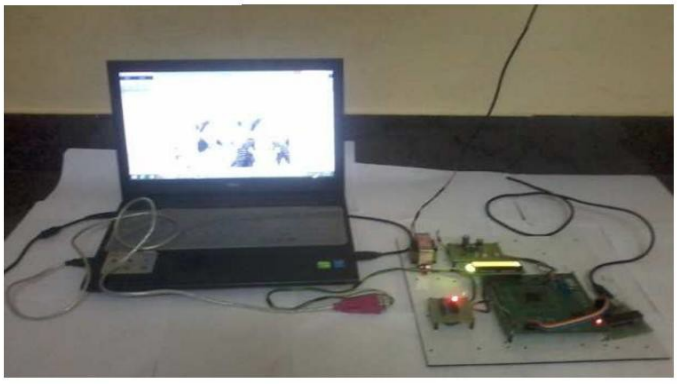

Fig.15 Hardware setup

Occupied CLB's, maximum frequency and Period etc parameters for Sobel edge detection algorithm are observed in synthesis report. Table II and Table III shows Device utilization summary and Timing Summary for simple implementation of Sobel edge detection filter.

Table II - Device Utilization Summary

Logic Utilization	**Used for simple implementation**	**Available**
No. of slice Flip flops	275	1920
No. of 4 Input LUTs	463	1920
No. of Occupied Slices	352	960
No. of CLB's	88	240
Total Equivalent gate count	6,610	100k

Table III- Timing summary

	Simple Implementation
Minimum Time period	15.629 ns
Maximum frequency	63.984 MHz
Total Estimated Power Consumption	34 mW
Setup Time	1.829ns
Hold Time	6.024ns

Table IV- Comparison with Shift –Add multiplication Device Utilization Summary

Logic Utilization	Available	Simple implementation	Optimization by Shift Add multiplier
No. of slice Flip flops	1920	275	229
No. of 4 Input LUTs	1920	463	385
No. of Occupied Slices	960	352	294
No. of CLB's	240	88	74
Total Equivalent gate count	100k	6,610	5,540

Table IV and Table V shows Comparison of simple (initial) implementation with Device utilization summary and Timing Summary of Shift Add multiplication

Table V- Comparison with Shift –Add multiplication Timing Summary

	Simple Implementation	Optimization by Shift Add multiplier
Minimum Time period	15.629 ns	14.567 ns
Maximum frequency	63.984 MHz	68.649 MHz
Total Estimated Power Consumption	34 mW	34mW
Setup Time	1.829ns	1.829ns
Hold Time	6.024ns	5.982ns

Table VI and Table VII shows Comparison of simple (initial) implementation with Device utilization summary and Timing Summary of Radix-4 Booth multiplication

Table VI- Comparison with Radix-4 Booth multiplier Device Utilization Summary

Logic Utilization	**Availabl e**	**Simple impleme ntation**	**Optimization by Radix-4 Booth multiplier**
No. of slice Flip flops	1920	275	229
No. of 4 Input LUTs	1920	463	387
No. of Occupied Slices	960	352	299
No. of CLB's	240	88	74
Total Equivalent gate count	100k	6,610	5,654

Table VII- Comparison with Radix-4 Booth multiplier Timing Summary

	Simple Implementation	**Optimization by Radix-4 multiplier**
Minimum Time period	15.629 ns	15.560 ns
Maximum frequency	63.984 MHz	64.268 MHz
Total Estimated Power Consumption	34 mW	34mW
Setup Time	1.829ns	1.829ns
Hold Time	6.024ns	6.024ns

The Sobel edge detection filter is applied on different images and it is realized using FPGA hardware. Following are the results of Sobel edge detection realization on Xilinx FPGA.

Fig. 16 A) **B)**

Fig. 16 C) **D)**

Fig. 16 E) **F)**

Different input images and their edge detected images are shown in Fig.16. Fig A, C, E is input images and B, D, F is their corresponding edge detected images.

CONCLUSION

This paper presents implementation of Sobel edge detection algorithm using VHDL language and its realization on Xilinx Spartan 3E XC3S100E FPGA and comparison with the result of two hardware optimization approaches i.e. Shift add multiplication and Radix-4 Booth multiplication. The minimum period required using simple, shift- add and Radiux-4 booth implementation is 15.629 ns, 14.567ns and 15.560 ns respectively. The maximum frequency required using simple, shift-add and Radix-4 booth implementation is 63.984 MHz,68.649 MHz and 64.268 MHz respectively .

REFERENCES

1) Ami J. Shukla, Prof. Vibha Patel, Prof. Nagendra Gajjar,"Implementing Edge Detection Algorithms in Real Time on FPGA", IEEE 2015.

2) Aneela Pathan, Tayab D Memon, " An Optimized 3x3 Shift and Add Multiplier on FPGA", IEEE , pp-346-350, 2017.

3) Girish N. Chaple, R.D. Daruwala , Manoj S. Gofane , "Comparisons of Robert,Prewitt,Sobel Operator Based edge detection methods for real time uses on FPGA",IEEE 2015.

4) Girish Chaple, R. D. Daruwala, "Design of Sobel Operator based Image Edge Detection Algorithm on FPGA", IEEE pp- 788-792, 2014.

5) Khuraijam Nelson singh,H. Tarunkumar, " A Review on Various Multipliers Designs in VLSI", IEEE 2015.

6) Lavanya K B, K. V Ramana Reddy , Siva S Yellampalli, "Comparative analysis of different optimization technique for Sobel edge detection on FPGA", ICAECC 2014.

7) Manoj Sharma, Richa Verma, " Disposition(reduction) of (negative) partial product for Radix-4 Booth's Algorithm" , IEEE 2011.

8) Mario Garrido, Fahad Qureshi, Oscar Gustafsson, "LowComplexity Multiplierless Constant Rotators Based on Combined Coefficient selection and Shift-and Add Implementation(CCSSI)" IEEE transactions on on circuits and systems, Vol.61, pp 2002-2012, July 2014.

9) Niharika, Dilip Kumar, "Analysis of Booth Multiplier using Radix-4 Technique using VHDL", IJRDASE, Volume 8 , Issue 2, January 2016.

10) Vanishree, K.V. Ramanna Reddy, "Implementation of Pipelined Sobel Edge Detection Algorithm on FPGA for High Speed Applications".