ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 1
www.ijisea.org

FPGA Implementation of Approximate Sobel Edge Detection

[1]Bandha. Nagaveni, [2]P. Anjaneya
[1]P.G. Student, Annamacharya Institute of Technology & Science, Kadapa, A.P, India,
bandhanagaveni@gmail.com
[2]Dept. of ECE, Annamacharya Institute of Technology & Science, Kadapa, A.P, India,
anjiaitsk@gmail.com

ABSTRACT

Image processing is an important task in data processing systems for applications such as medical sectors, remote sensing, and microscopy tomography. Edge recognition is a sort of image division method that is used to simplify the image records so as to reduce the amount of data to be processed. Edges are considered the most important in image processing because they are used to characterize the boundaries of an image. The performance of the Canny edge recognition algorithm remarkably surpasses the present edge recognition technology in various computer visualization methods. The main drawback of using Canny edge boundary is that it consumes lot of period due to its complex computation. In order to tackle this problem a hybrid edge recognition method is proposed in block stage to locate edges with no loss. It employs the Sobel operator estimate method to calculate the value and direction of the gradient by substituting complex processes by hardware cost savings, traditional non-maximum suppression adaptive thresholding block organization, and conventional hysteresis thresholding. Pipeline was presented to lessen latency. The planned strategy is simulated using Xilinx ISE Design Suite14.2 running on a Xilinx Spartan-6 FPGA board. The synthesized architecture uses less hardware to detect edges and operates at maximum frequency of 935 MHz.
Keywords: Canny edge detection, Field Programmable gate array, area, power, speed signal processing

1. INTRODUCTION

The electronic device utilization has increased in day to day life. -The idea of digital data computation has made a dramatic effect in our society. In recent computing platforms, the computations are performed in precise way depending on application requirement. In recent years, the design performance and efficiency of computing platforms have an exponential increase in computation. From an application perspective, all computations are not equally important. Figure 1.1 shows two tasks that deal with division between 521 and 32. In the first task, the output is compared to 16.1, and in the second task, it is compared to 10. For human brain, it is simpler and easier to understand the second case than first, compared to computers. In present scenario, computers work harder, and produces efficient results for many applications with trade-off in area, power and delay.

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 1
www.ijisea.org

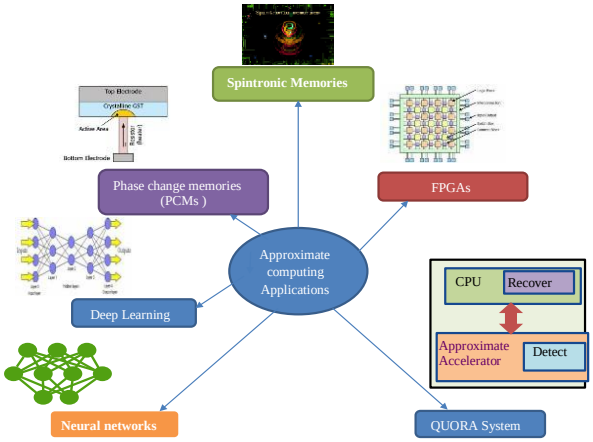


Fig. 1 Approximate computing application

2.LITERATURE SURVEY

Edge detection is a crucial step in image processing and computer vision, serving as the basis for object recognition, feature extraction, and scene understanding. Among various edge detection techniques, the Sobel operator is widely used due to its simplicity and effectiveness in detecting horizontal and vertical gradients in images [1]. In recent years, FPGA-based implementations of Sobel edge detection have received significant attention because FPGAs offer parallel processing, reconfigurability, and low-latency computation, which are ideal for real-time applications. Zhou et al. [1] implemented an improved Sobel operator on FPGA, optimizing the convolution process and leveraging parallelism to enhance edge detection accuracy without sacrificing throughput. Navinkumar et al. [2] proposed a multiplier-free Sobel design to reduce the number of logic elements and DSP blocks required, making it suitable for resource-constrained FPGAs. Raman and Gottipati [3] implemented the Sobel operator using convolution-based architectures on a Zynq-7000 FPGA, demonstrating that parallel computation of gradients can significantly accelerate processing. Earlier work by Anusha et al. [4] and Microprocessors and Microsystems [5] explored hardware-efficient Sobel designs, focusing on minimizing memory access and arithmetic complexity to improve speed and reduce area. Younis et al. [6] applied FPGA-based Sobel edge detection to medical imaging, specifically blood cell analysis, illustrating the practical utility of hardware acceleration in domainspecific applications.
To achieve high throughput, several studies have employed parallelization, pipelining, and accelerator-based architectures. Ravichandran et al. [8] designed a pipelined Sobel operator on FPGA, where multiple stages of gradient computation and thresholding were executed concurrently, allowing real-time processing of high-resolution images. Xu and Zheng [9] proposed a reconfigurable Sobel accelerator that uses pixel-level parallelism and efficient line buffers, enabling simultaneous computation of multiple gradient values while reducing memory bottlenecks. Network-on-Chip (NoC) based designs [10] allow the FPGA to process different image regions concurrently, further improving latency and throughput, which is critical for applications such as video surveillance and industrial inspection.

In addition to parallelism, approximate computing techniques have emerged as an effective way to reduce FPGA resource utilization and power consumption. Exact computation of the gradient magnitude involves square-root operations, which are costly in hardware. By using approximate methods such as the absolute sum $|Gx| + |Gy|$ instead of $\sqrt{(Gx^2 + Gy^2)}$, the hardware can avoid complex arithmetic units while still maintaining acceptable edge detection quality [1], [13]. Prabakaran et al. [11] proposed approximate arithmetic units for FPGA systems, which can be directly applied to gradient computation in Sobel operators. Ebrahimi et al. [12] designed pipelined approximate multipliers and dividers that reduce area and improve

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 1
www.ijisea.org

speed, and Hemanth Krishna et al. [13] introduced sign-focused approximate multipliers specifically optimized for edge detection tasks. These approaches allow designers to make controlled trade-offs between accuracy, throughput, and hardware resources.

Several surveys and comparative analyses highlight the effectiveness of FPGA-based Sobel implementations. Research Gate [14] and the SBC Journal of Embedded Systems [15] provide comprehensive overviews of various edge detection operators implemented on FPGAs, emphasizing the trade-offs between computational complexity, detection accuracy, and throughput. These studies also illustrate how approximate arithmetic, pipelining, and parallelism can be combined to create efficient, real-time edge detection systems suitable for embedded applications. Overall, FPGA-based approximate Sobel edge detection offers a practical solution for applications that require high-speed processing with limited hardware resources.

3. PROPOSED WORK

Edge detection is a fundamental problem of computer vision and image processing. It has been a major issue in image segmentation and for researchers (Gonzalez et al. 2004, Yanga et al. 2008). The purpose of image segmentation is the partition of an image into meaningful regions with respect to particular application, where edges in digital images are areas with strong intensity contrasts and a jump in intensity from one pixel to the next can create major variation in picture quality and image segmentation. For computer vision and image processing systems interpretation and edge detection of an object in images are the major constraint (Marr & Hildreth 1980, Koplowitz 1994). Edge detection is an active area of research as it facilitates higher level image analysis (Gonzalez et al. 2004).

Basically, Prewitt edge detection has two masks, one for detecting image derivatives in x, and another in y. The differential gradient edge detection is a time consuming process in estimating the orientation from the magnitudes in x- and y-directions. This edge detection is limited to 8 possible directions.

The masks of Kirsch technique is defined by considering a single mask and rotating it to eight main compass directions. The edge magnitude is defined as the maximum value found by convolution of each mask with the image. The direction is defined by mask that produces the maximum magnitude.

The Robinson method is similar to Kirsch masks, but is easier to implement because they rely only on coefficients 0, 1 and 2. The masks are symmetrical about their directional axis with zeros. The magnitude of gradient is the maximum value gained by applying all eight masks to the pixel neighbourhood, and the angle of the gradient can be approximated as the angle of line of zeros in the mask and yields maximum response. The Sobel technique computes the gradient by using the discrete differences between rows and columns of 3×3 neighbourhood. Therefore, high spatial frequency corresponds to edges. The disadvantage of Sobel method is the sensitivity towards noise. The primary advantage of Sobel operator is its simplicity and their orientations (Gonzalez et al. 2004). An approximation to gradient magnitude of the Sobel operator detects the edges and their orientations.

The Laplacian method detects the edges by using second derivative of the images. In general, the basic Laplacian edge detection operators are Laplacian Of Gaussian (LOG) and Canny (Marr & Hildreth 1980, Canny 1986). LOG is a second-order differential algorithm. The first step of this algorithm is pre-smoothing with Gaussian low-pass filter on the image. This identifies image edge, steep with Laplacian algorithm. Finally, it generates closed and connected contours with zero-gray value by excluding all internal points. Canny edge detection is a multistage algorithm to detect a wide range of edges in images and gradient is

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 1
www.ijisea.org

calculated using the derivative of a Gaussian filter. This method uses two thresholds to detect strong and weak edges and includes the weak edges in the output only if they are connected to strong edges (Canny 1986). The disadvantage of canny operator is design complexity.

Among all edge operators, Sobel operator have more perception, simple and approximation of gradient magnitude computation is easier. Therefore, hardware implementation of Sobel edge detection using the appropriate proposed approximate circuits discussed in previous chapters has been used.

APPROACH TO APPROXIMATE SOBEL EDGE DETECTION

Conventional method of edge detection involves operators, and is a two-dimensional filter. Edge is detected in image when gradient is maximum. The main purpose of edge detection is to substantially reduce the amount of data and filters the useless information by preserving the important structural properties in an image. Basically, edge detection involves filtering, edge enhancement, detection, and edge localization as shown in Figure 6.1 (Gonzalez et al. 2004). Filtering is used to remove noise and improves the performance of edge detection in the image. Edge enhancement is related in applying a gradient operator to x and y-direction. Detection involves in selecting true members in the set of points comprising an edge. Localization computes location and orientation of an edge.

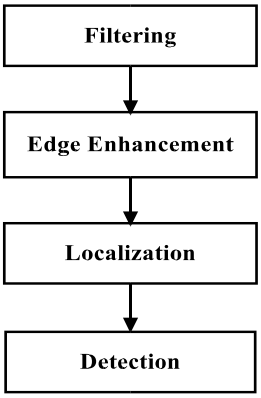


Figure.1 Edge Detection Steps

The pseudo code for conventional Sobel edge detection is given below. In general, the Sobel edge detection is operated in x gradient (Gx) and y gradient direction (Gy). In computing x-gradient and ygradient, normal addition, subtraction, and multiplication operations are involved. After Gx and Gy, compute the magnitude of the gradient. The edge of an image is detected if the magnitude of gradient is greater than the threshold value.

Pseudo Code for Conventional Sobel Edge Detection

Input: Sample Image (f(x,y))

Output: Edge detection of sample image

Step 1: Read the input image

Step 2: Median filtering which has coefficients $\alpha_i = \begin{cases} 1, & i=(N+1)/2 \\ 0, & \text{otherwise} \end{cases}$

Step 3: Edge enhancement - Apply Sobel operator to compute Gx and Gy

Gx = ((2*f(x+2,y+1)+f(y+2,j)+f(x+2,y+2))-(2*I(x,y+1)+f(x,y)+f(x,y+2)));

Gy = ((2*f(x+1y+2)+f(x,y+2)+f(x+2,y+2))-(2*f(x+1,y)+f(x,y)+f(x+2,y)));

Step 4: Localization - Compute the magnitude of gradient G= ׀ Gx+Gy ׀

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 1
www.ijisea.org

Step 5: Fix the threshold (T)
Step 6: Detection - If |Gx| > T then the edge is detected otherwise no edges are detected.
Step 7: Display the detected edges f′(x,y).

In conventional Sobel edge detection, addition, subtraction and multiplication computations are involved during gradient computation in x and y directions. The pseudo code for Sobel edge detection based on proposed approximations discussed in chapter 3, 4 and 5 are given below. The Sobel edge detection based on proposed approximations, approximate adder, approximate subtractor, and approximate multiplier is applied at x and y gradients.

The appropriate approximate adder, approximate subtractor, and approximate multiplier discussed in previous chapters has been chosen to design and implement the Sobel edge detection on FPGA. This version of Sobel edge detection is named as approximate Sobel edge detection as approximation has been applied during gradient computation in x and y direction. Though the approximations are applied in Sobel edge detection, as accuracy in approximation is mainly focused in this thesis, AA12, APSC4, approximate Dadda multiplier5 based on approximate 4-2 compressor5 designs have been considered for design and implementation due to minimal errors.

Pseudo Code for Proposed Approximate Sobel Edge Detection
Input: Sample Image (f(x,y))
Output: Edge detection of sample image
Step 1: Read the input image
Step 2: Median filtering which has coefficients $\alpha_i = \begin{cases} 1, & \text{i=(N+1)/2} \\ 0, & \text{otherwise} \end{cases}$
Step 3: Edge enhancement - Apply Sobel operator to compute Gx and Gy
Gx = ((2*f(x+2,y+1)+f(y+2,j)+f(x+2,y+2))-(2*I(x,y+1)+f(x,y)+f(x,y+2)));
Gy = ((2*f(x+1y+2)+f(x,y+2)+f(x+2,y+2))-(2*f(x+1,y)+f(x,y)+f(x+2,y)));
During the computation of Gx and Gy, consider AA12, APSC4, and Dadda multiplier5 based on approximate 4-2 compressor5 designs by replacing the normal addition, subtraction, and multiplication.
Step 4: Localization - Compute the magnitude of gradient G= Gx+Gy
Step 5: Fixing the threshold value.
Step 6: Detection - If |Gx| > T then the edge is detected otherwise no edges are detected.
Step 7: Display the detected edges f′(x,y).

ARCHITECTURE FOR SOBEL EDGE DETECTION BASED ON APPROXIMATE COMPUTING CIRCUITS

The architecture for edge detection is shown in Figure 3.2. This architecture uses 3×3 convolution kernels to process the given input image. The architecture of Sobel edge detection includes 3×3 pixel generation, approximate gradient computation, Static Random Access Memory (SRAM) controller and threshold comparator as shown in Figure 3.2. In this architecture, Clk is the clock signal, data input is the pixel signal of gray scale image, result is the output of Sobel edge detection operator signal, generation data and data are the intermediate signal. The Sobel edge detection based on approximate computing circuits shown in Figure 3.2 uses the high accuracy circuit in approximations. The gradient computation in x and y direction uses AA12, APSC4, approximate Dadda multiplier5 based on approximate 4-2 compressor5 designs as accuracy in approximation is mainly considered in this thesis.

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 1
www.ijisea.org

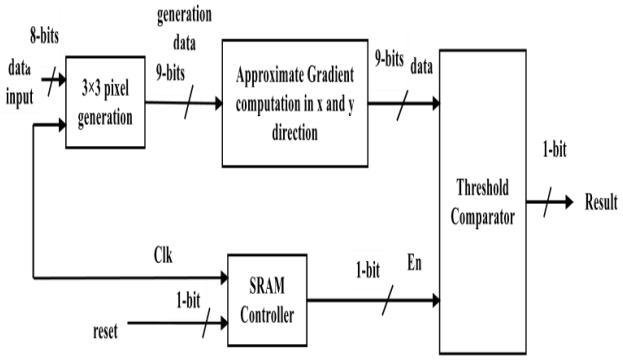


Figure 2. Architecture of Edge Detection

The architecture of approximate gradient computation shown in Figure 3.2 uses approximate adders, subtractor, and multipliers. The gradients in x and y directions are computed using approximate subtraction. The overflow bit flag_x/flag_y indicates the sign of resultant number. If overflow occurs, then it is considered as a positive value else negative value. Negative values are represented using two's complement.

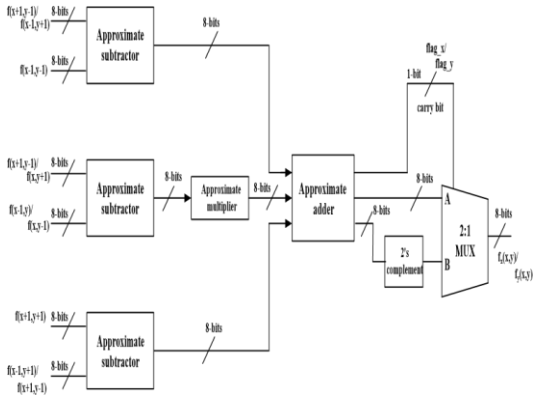


Figure 3. Architecture of Approximate Gradient Computation

The structure of threshold comparator is shown in Figure 3.3. En is an enable signal to control the output. Data is the result of approximate gradient computation block. In threshold comparator, Data and fixed threshold value is compared using En. The final result of edge detection having only two pixel values according to the given threshold value.

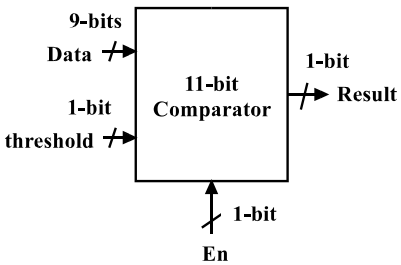


Figure2. Threshold Comparator

Systems designers have become more creative in their use of cache memory, interleaving, burst mode and other high-speed methods for accessing memory. There are many reasons to use SRAM or Dynamic Random Access Memory (DRAM) in edge detection. Compared to DRAM, SRAM is effective. Asynchronous SRAMs respond to changes at the device's address pins by generating clock signal that is used to timing

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 1
www.ijisea.org

control the SRAM's internal circuitry during a read or write operation. This type of design runs into limitations at the high end of performance range. The need of control logic is to configure the logic blocks in FPGA. In the proposed hardware setup, asynchronous SRAM controller is preferred than synchronous SRAM controller, due to less execution time (https:// www.ece.cmu.edu/ ~ece548/ localcpy/ sramop.pdf). The block diagram of SRAM Control logic is shown in Figure 6.5.

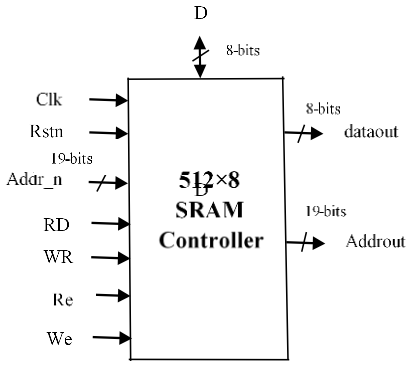


Figure3. SRAM Control Logic

The state diagram of SRAM control logic is illustrated in Figure 3.5. The clock (Clk), reset (Rstn), 19bit address (Addr_n), read enable (Re), write enable (We), read data (RD) and Write data (WR) are the inputs of SRAM Controller, Data (D) of 8-bits is the input and output data of SRAM controller with address (Addrout) of 19-bits. In this control logic, various state of operations such as read 0 (RD0), read 1 (RD1), write 0 (WR0) and write 1 (WR1) are performed based on the control signals. When Reset (Rstn) = 1, SRAM FSM_D goes to an idle state. During the rising edge of clock, when Re=1,

SRAM FSM_D performs the operation RD0. When We=1, SRAM FSM_D switch to WR0 state. When Re=1, and We=0 the state switches to perform RD1 operation. When Re=0 and We=1, WR1 operation has been performed and change the state from SRAM FSM_D to WR1. When Re=1, state changes from WR1 to RD0. When Re=0 and We=0, RD0 and WR0 goes to SRAM FSM_D state respectively.

4.RESULT ANALYSIS

4.1. Implementation Approach of Approximate Sobel Edge Detection on FPGA

Recently, FPGA technology has become a viable target for the implementation of algorithms appropriate to video and image processing applications. The unique architecture of the FPGA has allowed the technology to be used in many such applications encompassing all aspects of video and image processing (Chou et al. 1993), (Benedetti & Perona 1998). FPGAs generally consist of a system of logic blocks (usually look up tables and flip-flops) and some amount of Random Access Memory (RAM), wired together using a vast array of interconnects. FPGA can be rewired, or reconfigured, with a different design as often as the designer likes. ASIC design methods be able to use for FPGA design, by allowing the designer to implement designs at gate level. However, engineers use hardware language such as VHSIC Hardware Description Language (VHDL) or Verilog, for design methodology similar to software design.

As per generations of Spartan families, designing with Spartan-3E help designers avoid the high initial NREs. Spartan-3E is the logic optimized, low-cost Xilinx FPGA family (https://www.xilinx.com/ support/ documentation/ data_sheets/ ds312.pdf). The Spartan-3E devices have the lowest FPGA unit cost, lowest cost-per-logic, and lowest configuration memory cost. Hence, Spartan-3E FPGAs offer a very large selection

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 1
www.ijisea.org

of low cost IP solutions (https://www.xilinx.com/ support/ documentation/ data_sheets/ ds312.pdf). The target applications of Spartan-3E FPGAs have focused on consumer electronics applications and also benefit for low-cost or high-volume applications such as image displays, video displays, peripherals, and DSP.

The process of hardware prototype for approximate Sobel edge detection mechanism has been explained with the help of a flow diagram shown in Figure 3.8. Initially the image is read from SRAM of Spartan 3E FPGA in Matlab and converted into blocks. The image blocks are send to FPGA using a serial port. Bit file generated based on proposed approximate Sobel edge detection as specified in the previous section is dumped into the FPGA device Xilinx Spartan 3E board (XC3S100E) through Joint Test Action Group (JTAG). After processing the image blocks using the proposed approximate Sobel edge detection, the processed blocks are send back through serial port. The processed blocks read through serial port are displayed using Matlab after receiving all blocks. Finally, the edge detected output image is displayed.

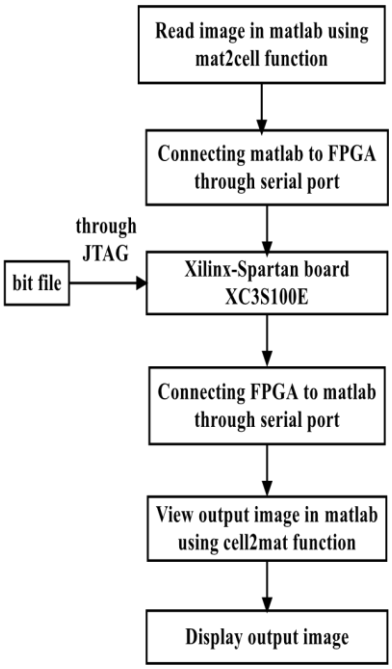


Figure.6 Steps in Hardware Prototype for Sobel Edge Detection

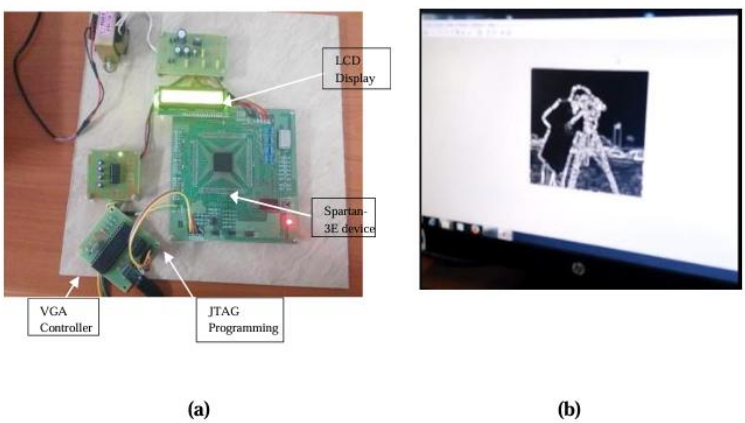


Figure6. Approximate Sobel Edge Detection (a) Hardware Setup (b) Edge Detection

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 1
www.ijisea.org

The software platform used for the approximate Sobel edge detection is Matlab 2013b and Xilinx ISE 14.2. Xilinx Integrated Software Environment (ISE) is a software tool for synthesis and analysis of HDL styles, simulate and configure the target device. Matlab is a high-level language and interactive environment for numerical computation, visualization, and programming. In this section, the approximate Sobel edge detection algorithm has been presented and quantified for various images.

This algorithm is implemented in Spartan XC3S100E kit by writing VHDL code for approximate Sobel edge detection using the ISE Style Suite. VHDL cannot handle the standard image formats, so the images are converted to block data using Matlab 2013b. Conversion from image file to block data and again block data to the image file is carried out using Matlab. Table 4.1 shows the analysis of various images on Sobel edge detection and proposed approximate Sobel edge detection. The approximate Sobel edge detection is compared with conventional Sobel edge detection and measures the accuracy in terms of Structural Similarity Index Measurement (SSIM) for various images. On the basis of visual perception and edge counts of various gray scale images, it is proved that proposed method is able to detect majority pixels in images. The thin edges without distorting the shape of images and less complex design makes us to choose Sobel edge detection.

5. CONCLUSION

Approximate computing provides an effective strategy for improving the efficiency of modern digital and image processing systems by exploiting application-level error resilience. By designing approximate arithmetic circuits, such as adders and multipliers, it is possible to achieve significant reductions in power, area, and delay while maintaining acceptable accuracy. These techniques address the computational bottlenecks inherent in FPGA, DSP, and VLSI-based systems, enabling high-performance and energy-efficient processing. Overall, approximate computing represents a promising approach for future computing architectures, particularly in domains where exact precision is not critical, such as image and signal processing applications.

REFERENCES:

[1] G. Zhou, S. Guo, and Z. Chen, “FPGA-Based Improved Sobel Operator Edge Detection,” Future Computing and Informatics Journal, 2023.
[2] K. Navinkumar, R. Logesh, P. VishnuBabu, and A. V. Ananthalakshmi, “FPGA Implementation of Sobel Edge Detection Algorithm,” EAI Endorsed Transactions on Internet of Things, 2024.
[3] S. S. Raman and R. Gottipati, “FPGA Based Implementation of Edge Detection Using Sobel Mask,” Asian Journal of Science and Technology, 2018.
[4] G. Anusha, T. JayaChandra Prasad, and D. Satya Narayana, “Implementation of Sobel Edge Detection on FPGA,” International Journal of Computer Technology and Applications, vol. 3, no. 3, 2012.
[5] S. Microprocessors and Microsystems, “A FPGA Based Implementation of Sobel Edge Detection,” Microprocessors and Microsystems, 2018.
[6] A. K. Younis et al., “Hardware Implementation of Sobel Edge Detection System for Blood Cell Images on FPGA,” International Journal of Electrical and Computer Engineering Systems, 2025.
[8] Ravindraiah, R., et al. "Deep Learning Classification of Diabetic Retinopathy Using ResNet-101 Convolutional Neural Networks." Convergence of Internet of Medical Things (IoMT) and Generative AI. IGI Global Scientific Publishing, 2025. 417-438.

[9] Gayathri, Bukke, and Grande Naga Jyothi. "Design and Analysis of a Super Source Follower-Based Multipath Fully Differential Operational Trans Conductance Amplifier." 2025 IEEE International Conference on Advanced Computing Technologies (ICACT). IEEE, 2025.
[10] "Design and Implementation of Sobel Edge Detection on FPGA Using NoC Concepts," International Journal for Scientific Research & Development, vol. 4, no. 2, 2016.
[11] B. S. Prabakaran et al., "ApproxFPGAs: Embracing ASIC-Based Approximate Arithmetic Components for FPGA-Based Systems," arXiv preprint arXiv:2004.10502, 2020.
[12] Z. Ebrahimi et al., "RAPID: Approximate Pipelined Soft Multipliers and Dividers for High Throughput and Energy Efficiency," arXiv preprint arXiv:2206.13970, 2022.
[13] Jyothi, Grande Naga, et al. "Design of FINFET based DRAM cell for low power applications." Computer-Aided Developments: Electronics and Communication (2019): 35-43.
[14] "FPGA-Based Improved Sobel Operator Edge Detection," ResearchGate, 2023.
[15] "Survey on FPGA Edge Detection Operators: Comparative Analysis of Sobel Implementations," SBC Journal of Embedded Systems, 2025.