International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Clock-Gated VLSI Architecture For Low-Power Sobel Edge Detection In Real-Time Systems

Anant Shankar Ellapalli
Department of ECE,
Sri Venkateswara College of Engineering (Autonomous),
Tirupati, AP.,India

Guru Venkata Sai Ekambaram
Department of ECE,
Sri Venkateswara College of Engineering (Autonomous),
Tirupati, AP.,India

Sowjanya Mude
Department of ECE,
Sri Venkateswara College of Engineering (Autonomous),
Tirupati, AP.,India

Chandana Gopal
Department of ECE,
Sri Venkateswara College of Engineering (Autonomous),
Tirupati, AP.,India

Vijay Poorna Chandra Sai Pathipati Department of ECE,
Sri Venkateswara College of Engineering (Autonomous),
Tirupati, AP.,India

Guru Prasad Chemuru
Department of ECE,
Sri Venkateswara College of Engineering (Autonomous),
Tirupati, AP.,India

***Abstract*— Edge detection is an important procedure in image analysis applications such as object recognition or medical imaging, in which the Sobel operator is popular because of its simplicity and capability to image edges by calculating gradient. Nevertheless, conventional Sobel filters are vulnerable to noise and they fail to identify fine edges. The proposed project applies a better Sobel edge detector algorithm with VLSI technology to increase the accuracy as well as to reduce noise and noises via modified gradient. The architecture, written in Verilog HDL and running on an fpga, is capable of providing a low-latency, parallel process, and power-efficient processing architecture, which is available in real-time embedded vision systems. It is demonstrated with simulations in tools such as Xilinx Vivado that this method raises edge detection accuracy and reduces the computational complexity making it possible to do edge detection quickly and reliably with hardware.**

Keywords: Sobel edge detector , sobel, FPGA-based designs, Vivado

I. INTRODUCTION

A. Background

Edge detection refers to an essential operation in image processing, computer vision, since it is a key component of detecting meaningful image details in terms of object boundaries, contours, and texture differences. Correct edge detection is a requirement in medical imaging, automated inspection, surveillance systems, object recognition, and applications with precision in edge detection [1]. One of the most popular edge detecting techniques is the gradient-based techniques since it is simple and is effective in detecting intensity transitions within images [2].One such gradient-based edge detector is the Sobel operator due to its simple convolution masks and ease in determining the horizontal and vertical intensity gradient [3]. The fact that it has low computational complexity of the algorithm makes it applicable in real time and embedded image processing applications where quick response and low resource consumption are needed [4].

The original Sobel operator is limited even though widely used. It is susceptible to noise in images and may lead to false edges and poor quality of edges in a noisy environment [5]. Also, it can usually fail to detect fine and weak edges which is important in the high-precision of medical diagnostics and industrial quality inspection [6].To overcome this, various advanced edge detection and image restoration methods have been put forward that aim at noise reduction and edge preservation [7]. It has been demonstrated that adaptive filtering, thresholding and hybrid image processing methodologies enhance accuracy of edge detection with noisy and blurred images [8], [9]. Many of these sophisticated techniques are however computationally complex and therefore they are not applicable in real time hardware implementation [10].The growing need of fast and real time image processing system in the current systems has contributed to the application of the hardware based system with the use of the Very Large Scale Integration (VLSI) technology [11]. Field Programmable Gate Arrays (FPGAs) have gained popularity as a platform of implementing image processing algorithms because they are reconfigurable, scalable, and can process multiple images in parallel [13].VLSI implementations are also stated to be ideal in real-time edge detection as they are reconfigurable, scalable and can process several images simultaneously [12]. Convolution-based algorithms like edge detection can be implemented with FPGA-based designs in an efficient way, and resource usage and performance optimized [14].

A better Sobel edge detection algorithm is suggested and implemented in this work with the help of a FPGA architecture on the basis of VLSI. The improved method alters the traditional Sobel operator and incorporates the method of thresholding in order to refine fine edges and eliminate noise. The design is represented in Verilog Hardware Description Language (HDL) and tested with simulators based on industry-standard tools. The system proposed has better accuracy in edge detection, less computational complexity and it is applicable in real-time image processing systems [15], [16].

II. LITERARATURE REVIEW

Edge detection is an essential process in image processing and is a major step towards determining important structural features like object boundaries and object contours. A

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

groundbreaking system of edge detection was proposed by Canny [1], which maximizes the accuracy of detection, localization, and noise reduction. The Canny operator remains to be a standard in assessing more recent edge detection algorithms as it is very robust and has a solid theoretical basis, and it is also important that it is less sensitive to image degradation due to blur and noise. Gota and Min [2] have compared various image restoration algorithms and revealed that effectiveness of the restoration process is dependent on the type of degradation and the algorithm chosen. Image deblurring methods were further surveyed by Mahalakshmi and Shanthini [3] and classified into blind and non-blind methods and the study found that the degradation of a feature extraction and recognition accuracy by blur is severe using a Gaussian blur. Ramya and Christial [5] reviewed blind deconvolution methods of image restoration using unknown blur, and found that most quality algorithms were impractical in real-time because of highly complex computation. The analysis of classical edge detection operators by Ansari et al. [7] revealed that methods based on the gradient like Sobel are computationally efficient and at the same time very sensitive to noise and weak edge gradient.

Syahrian et al. [8] demonstrated the practical significance of edge detection of high order and quality with the pipe monitoring robot that operates on vision to detect cracks. Their effort pointed out the difficulties of the use of edge detection in the real world setting with noise and changes in illumination. Jain and Goswami [9] and Yadav et al. [10] also pointed out that no one of the methods of restoration works best in any blur conditions, which is why adaptable and situation-dependent methods are essential. Verma and Ali [11] compared different types of noise and the methods of its removal, demonstrating that noise plays a major role in lowering the quality of its edges. As Sekehravani et al. [12] have shown, filtering stage-combined edge detection algorithms are more robust to noise in noisy images, whereas Saini et al. [13] identified the trade-off between detection accuracy and computation complexity among various image processing edge detectors. [14], [15].

State-of-the-art restoration algorithms that make use of total variation models as suggested by Prasath and Thanh [16] and Thanh et al. [17] could be characterized by powerful noise-reduction and edge-preservation properties, but entailed the implementation of iterative computations, which are hard to execute efficiently in hardware. Bilateral filtering has also proven to be a useful edge preserving denoising method, with Chen et al. showing it to be an effective method [18] but its computational complexity makes it hard to implement in software in real-time. To overcome this, a series of FPGA based and VLSI based implementations have been suggested.Gabiger-Rose et al. [19] and Dabhade et al. [20] proposed scalable FPGA architectures to perform real-time denoising, and Lien et al. [21] and Jang and Hwang [22] proposed low-cost and noise-aware VLSI based design to implement the proposed edge detection algorithm and show the viability of high-performance hardware based image processing. The architecture is represented in Verilog Hardware Description Language (HDL) allowing the ability to control data flow, timing, and hardware resources with a lot of precision. The implementation will be based on FPGA platforms to verify the adaptability of the suggested architecture on real time applications.

Scope of Work

This work has the scope of designing and implementing a better Sobel edge detection algorithm that would address the shortcomings of the traditional Sobel operator especially its vulnerability to noise and failure to capture fine edges precisely. The algorithm is applicable to real-time image processing systems since it focuses on improving edge detector sensitivity and reducing computational complexity by changing the original Sobel gradient masks and applying the technique of thresholding. With the design, the parallel processing is used to ensure low latency and efficient execution, which are essential to image processing systems with high speeds.

The implementation will be based on FPGA platforms to verify the adaptability of the suggested architecture on real time applications. FPGA-based implementation provides analysis of resource usage, power consumption and processing rate as well as flexibility to modify the design. Simulation and verification are implemented on industry-standard tools like Xilinx Vivado to ensure the functional correctness and performance reliability.Lastly, the performance analysis and comparison with the conventional method of Sobel edge detection are also a part of this work. Measures like accuracy of edge detection, ability to suppress noise, complexity of computation and hardware efficiency are taken into consideration. The findings prove the appropriateness of the intended FPGA enhanced Sobel edge detector to embedded vision system, medical imaging and other real time image processing projects.

III. EXISTING METHOD

The system aims at deploying the improved version of improved Sobel edge detection algorithm with the VLSI technology to attain high speed, precision, and low power image processing. The hardware based solution attempts to overcome the shortcomings of traditional Sobel filters by incorporating algorithmic refinements in the circuit design.The fundamental component of the implementation is the alteration of gradient masks of the standard Sobel operator. The 3x3 classical kernels are modified to enhance sensitivity to fine edges as well as enhance the capabilities of isolating real edges and noise resulting into more accurate edge Fig (1).

The input image is then processed pixel-by- pixel whereby the neighbor of each pixel is convolved with the enhanced Sobel masks. The use of pipeline processing stages is used in the design to facilitate real time processing and allow this convolution operation to be performed very fast using the VLSI architecture. Each stage does a portion of the edge detecting job, so that there can be no bottlenecks in passing the pixel data through the system.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

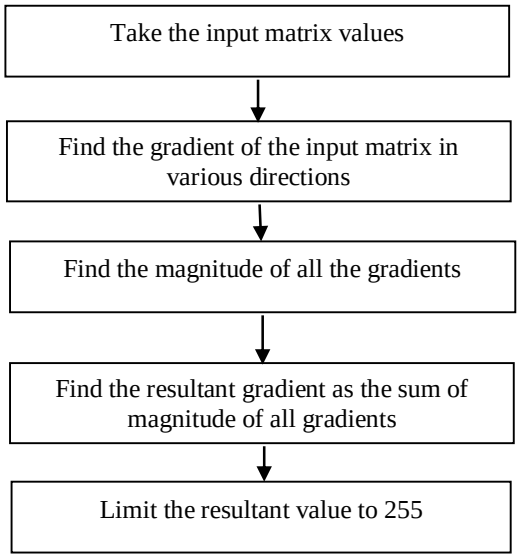

Fig. 1. Algorithm for existing method

Fig (2). architecture takes streaming input images which are received at pixel value in a serial order and at the same time the gradient components are calculated in horizontal and vertical direction. The following gradients are then summed to calculate the magnitude of the edges.The proposed design incorporates a threshold mechanism after calculating the magnitude of the gradients to reduce sensitivity to noise. This thresholding will suppress weak gradients which is most probably due to noise making the edges found to be more clear, the threshold value can be programmed to dynamically change with the image properties or application needs.

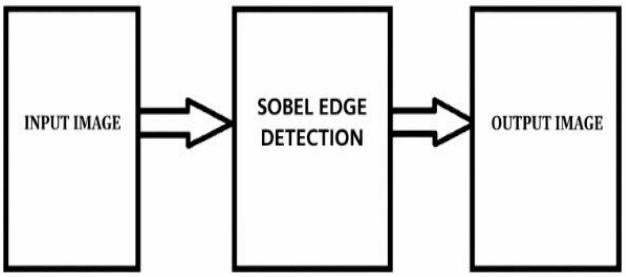

Fig. 2. Block diagram for existing method

All calculations in the system are done using fixed-point arithmetic, to minimize the complexity of the hardware and power use and still achieve sufficiently good precision. The trade-off between the use of resources and the detection accuracy of a specific design is balanced by the design being written in Verilog HDL, with a synthesizable and accurate hardware description that can be optimally exploited by mapping to FPGA or ASIC platform.The design is a key design attribute since it is written in Verilog HDL. The blocks of the system are defined clearly and include input buffering, gradient computation convolution units, magnitude calculation units, thresholding modules and output registers.The input buffering enables the temporary storage of pixel data to enable the convolution window and permit the flow of data to remain constant. This buffering is achieved by use of line buffers and shift registers which are optimized in terms of hardware efficiency.

The parallel multipliers and adders of the convolution units are such that the calculations of the neighbourhood of pixels are carried out simultaneously. This parallelism further reduces throughput by a large margin over sequential software implementations, and the calculation of gradient magnitude is done using an approximation scheme, e.g. sum of absolute values, to avoid expensive square root actions and further minimizing hardware resources.The gradient magnitude computation is compared with the programmable threshold and binary edge/non-edge signals are subsequently produced. This makes further processing of the information easier and also lessens on the bandwidth of the data.

The system produces a binary edge map which is appropriate for further processing or visualization of the input image. The design is also designed to be used in real-time applications with minimal latency between input and output enabling the design to be used in portable and embedded vision systems with appropriate scalability via configuration parameters of image resolution and processing rate. This flexibility can be applied to quite a wide set of devices and performance requirements.

IV. PROPOSED METHOD

Flip flop based clock gating method: It is a technique of transferring the latches to flip flops to designs and vice versa. In the master slave theorem, two latches can be observed by splitting flip flop. In this method, we observe the D flip flop and AND gate Flip-flop-based clock gating is a method that minimally enables the clock consumption of digital circuits when the system is not operating. In contrast to the latch based clock gating which may cause glitches, this method provides a glitch free and stable gated clock since the enable signal (EN) is synchronized with the clock edge via a flip-flop. The method operates by sampling the enable signal on the upstream of the clock with the help of a D flip-flop. The en-ff output is then ANDed with the clock signal to form a gated clock (CLK_GATED). This guarantees that the gated clock transitions can only happen at welldefined clock edges and glitches are avoided and a reliable operation is achieved.

The suggested approach includes a VLSI-based Sobel edge detector architecture that is capable of running efficient image processing applications at low-power. The system starts with the process of converting the input image to a representation as a text of pixels, which is easily integrated with hardware description languages and gives easy simulation and verification in FPGA settings. This text formatted data serves as the input stream to the Sobel processing module and therefore the design is appropriate to hardware testbenches and real-time embedded systems. The proposed design uses modified gradient computation instead of a fixed gradient mask and is much tolerant to noise compared to the traditional Sobel filters, which are very sensitive to noise and incorrect edge localization. The horizontal and vertical gradient components are calculated simultaneously, which allows the image with low contrast or noisy images to extract the edges correctly.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

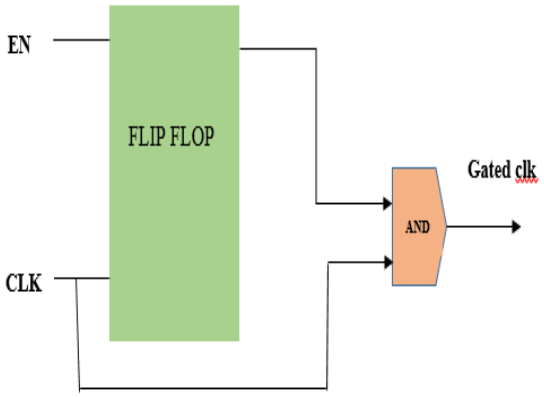

Fig. 3. Flip flop based clock gating technique

Based on the above Fig (3), gated clock will take high value when flip flop output and clock take on high state otherwise gated clock will take high value. This is to say that clock in sleep mode then gated clock also in zero state The benefit is that the enable signal is read at each rising edge of the clock and maintains glitch free and hold time free. The method finds extensive application in low-power processors, digital signal processing designs and FPGA/ASIC designs where power efficiency is of importance. The flip-flop method has superior timing coverage, eliminates race, and is simpler to incorporate into synchronous designs than latch-based clock gating.

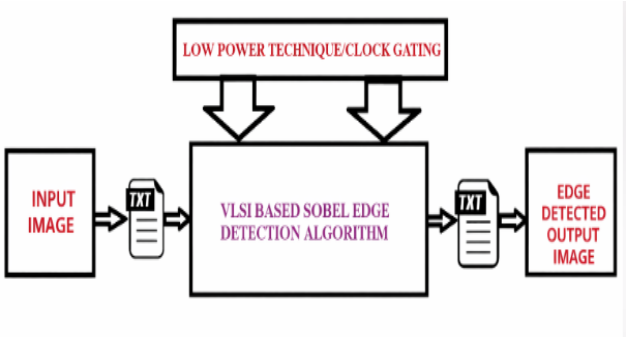

Fig. 4. Proposed block diagram for Sobel Edge Detection

The proposed system Fig (4). takes advantage of parallelism that exists in VLSI architecture to perform at high levels. To minimise processing latency, multiple arithmetic units will be carried out simultaneously to process pixel data, and hence processing latency is minimised at the expense of software-based implementations. One of the major characteristics of the proposed design that suits real-time image processing is that a low-power clock gating option is included. Clock gating is a technique that turns off idle processing blocks when not in use which reduces the switching activity and dynamic power usage unnecessarily. This reduces power consumption in the architecture and is especially useful in battery-powered and portable vision systems where optimization of power consumption is a crucial factor.Once edge detection has been done the processed pixel values are once more stored in text format which makes it easy to compare the input and output images during simulation and debugging. It is based on this data that the final edge-detected image is reconstructed so that the functionality of the hardware design is properly verified. The proposed Sobel edge detector system based on VLSI has been shown to have better edge detection accuracy, less sensitivity to noise and less power consumption than the use of the conventional Sobel architectures. The simulation and synthesis of FPGA shows that design has low computational complexity, high throughput and efficient resource usage. Consequently, the suggested approach is quite appropriate when it comes to real-time embedded vision and low-power image processing systems.

V. RESULTS & DUSCUSSION

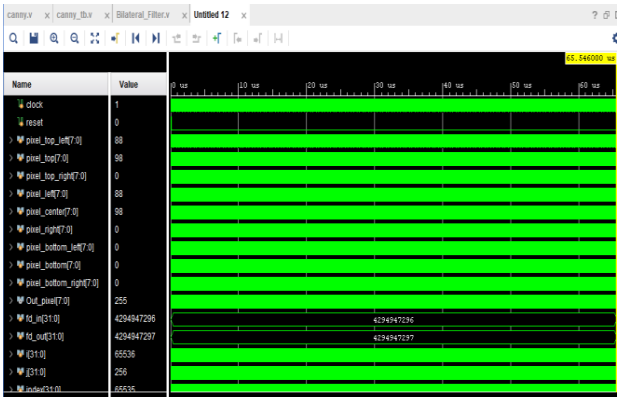

Fig. 5. Simulation Waveform of Proposed Sobel Edge Detection

The VLSI-based Sobel edge detection module functionality verification at the pixel-processing level is illustrated by the presented simulation waveform Fig (5). The signal of a clock makes it work synchronously and reset all the internal registers to some well-known state and then starts to work. After the idea of reset is de-asserted, pixel inputs equivalent to a 3x3 Sobel window- pixel top left, pixel top, pixel top right, pixel left, pixel center, pixel right, pixel bottom left, pixel bottom and pixel bottom right are inputted. The majority of neighboring pixels in this waveform share similar values of intensity whereas the center pixel possesses a contrast difference, forming a possible edge condition. This arrangement simulates a real image neighborhood as being fed sequentially into the equipment.

Sobel operator internally calculates horizontal and vertical gradients, and it is indicated by intermediate signals, i.e., fd in and fd out. These values are very large 32 bit numbers since they are accumulated gradient values prior to normalization or clipping. These values are constant when the pixel inputs are constant, and hence the waveform indicates that the behavior is correct both in combinational and registered operation. The last out_pixel signal is the edge-detected signal of the intensity of the output. In the relevant case, it takes on a high value (e.g., 255), indicating strong edge has been identified at the center pixel location.Generally, the waveform is showing that the logic of Sobel edge detectors has the right timing, data flow and arithmetic operation. The constant gradients and credible output pixel value are explicit that the design is properly working with the 3x3 pixel window and generates a saturated edge response in case a large intensity change occurs. It is checked within this simulation that the design is functional and prepared in terms of additional optimization steps like clock gating and low-power implementation.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

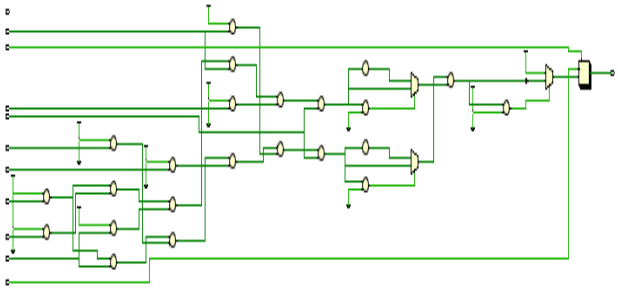

Fig. 6. RTL Schematic of Proposed Sobel Edge Detection

Fig (6) indicates that the logic implementation of the proposed Sobel edge detection hardware at the gate level presents the implementation of both arithmetic and control operations with the use of simple digital components. Interconnected AND, OR and inverter gates are the combinational logic to compute gradient values, compare the magnitudes and data flow control. The parallel signal flow also emphasizes the process of multiple pixel input processing in parallel, which is necessary in order to attain high throughput in real-time image processing application.

Total On-Chip Power:	**7.693 W**
Design Power Budget:	**Not Specified**
Power Budget Margin:	**N/A**
Junction Temperature:	**39.4°C**
Thermal Margin:	45.6°C (24.1 W)

.

Fig. 7. Power of Proposed Sobel Edge Detection

Fig (7). shows the power estimation report of the proposed design based on the generated netlist of the synthesized design. The overall on-chip power consumption is 7.693W, which is determined based on an estimated signal activity based on design constraints, simulation files, or vectors free analysis. As it is an early estimation of the implementation, the tool is a clear indication that the power would change after place-and-route because switching activity and routing effects were more accurately modeled.The power budget margin is also indicated as N/A since no specific design power budget was defined. The measured temperature of the junction is estimated as 39.4 C which indicates that the design is working within a safe thermal temperature of the given conditions.

.

Name	Slice LUTs (134600)	Slice Registers (269200)	DSPs (740)	Bonded IOB (400)	BUFGCTRL (32)
sobelEdge_Detection	39	9	8	75	1
f1 (flipflop_based_clk)	0	1	0	0	0

Fig. 8. Area of Proposed Sobel Edge

Detection The Fig (8). indicates the breakdown of FPGA resource utilization made in the sobelEdge_Detection design, together with the clocking element. The primary Sobel edge detector block uses 39 Slice LUTs and 9 Slice Registers which is incredibly low with regard to the overall resources available on the target platform. Further, the design has 8 DSP blocks to perform gradient and magnitude calculations, 75 bonded IOBs to interface pixel data, and only 1 BUFGCTRL to distribute the global clock, which proves that the logic to generate or include the clock is insignificant in terms of overhead. In general, this report has provided insight into the fact that the proposed Sobel edge detector architecture has been area efficient, scalable, and suitable to the low-power VLSI or FPGA-based image processing application, leaving many resources to be utilized in conjunction with the other processing stages.

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay
Path 1	∞	14	15	8	pixel_top_left[7]	Out_pixel_reg[1]/D	15.591	10.783	4.808
Path 2	∞	14	15	8	pixel_top_left[7]	Out_pixel_reg[5]/D	15.591	10.783	4.808
Path 3	∞	14	15	8	pixel_top_left[7]	Out_pixel_reg[6]/D	15.591	10.783	4.808
Path 4	∞	14	15	8	pixel_top_left[7]	Out_pixel_reg[7]/D	15.591	10.783	4.808
Path 5	∞	14	15	8	pixel_top_left[7]	Out_pixel_reg[0]/D	15.572	10.764	4.808
Path 6	∞	14	15	8	pixel_top_left[7]	Out_pixel_reg[2]/D	15.572	10.764	4.808
Path 7	∞	14	15	8	pixel_top_left[7]	Out_pixel_reg[3]/D	15.572	10.764	4.808
Path 8	∞	14	15	8	pixel_top_left[7]	Out_pixel_reg[4]/D	15.572	10.764	4.808
Path 9	∞	2	2	1	Out_pixel_reg[0]/C	Out_pixel[0]	3.557	2.912	0.646
Path 10	∞	2	2	1	Out_pixel_reg[1]/C	Out_pixel[1]	3.557	2.912	0.646

Fig. 9. Delay of Proposed Sobel Edge Detection

Fig (9) of this timing report indicates that there are several critical paths illustrated by the start of pixel top left[7] and the final destination of pixel top left[7] at various Out pixel reg[*]/D registers. Routes 1 to 8 are basically the same: 14 logic levels, 15 routes and fanout of 8 and a total delay of about 15.57-15.59 ns. The largest of that delay is due to logic delay (~10.7610.78 ns), and net delay is constant (~4.81 ns) which suggests both deep combinational logic and significant routing/fanout effect. The infinity slack (∞) indicates that these paths are not presently held up by a clock, or belong to an unbounded timing space.Paths 9 – 10 are far less complicated and avoid all the clutter with a distance of only 2 levels, 1 fanout, and a total delay of about 3.56 nsec. In this case, there is a preponderance of logic delay (~2.91 ns), and little routing (~0.65 ns). The primary issue in general is evidently the long pixel top left [7] Out pixel reg paths which are heavy in logic depth and fanout and would be the first to be considered in the optimizations or pipelining, as soon as appropriate timing constraints are applied.

A. . VLSI-Based Sobel Edge Detection Using Flip-Flop-Based Clock Gating

Edge detection is one of the basic tasks of image processing systems as it is one of the central processes involved in object recognition, medical imaging and surveillance. Of many different edge detection methods, Sobel operator is popular and is preferred by humans because it is simple and resistant to noise. The given paper describes a VLSI-based Sobel edge detector architecture that has been optimized in terms of both low power usage

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

and high performance with the help of a flip-flop-based approach to clock gating.The input grayscale image is first transformed into a text-based pixel representation, which is used in the suggested method. This conversion allows easy interface with hardware description languages like Verilog HDL and allows easy simulation and verification within FPGA environments. The pixel data available in text format is used as the input stream in Sobel processing module and therefore the design is applicable to real-time embedded systems as well as in hardware testbench validation.

The Sobel edge detecting algorithm is based on the calculation of the intensity gradient of the image both horizontally and vertically with the two 3x3 convolution masks. Horizontal gradient focuses more on vertical edges and the vertical gradient on the horizontal edges. The gradient values of every pixel are computed using the preceding pixel intensities in convolving them with the Sobel masks. The edge localization is then carried out by adding up the magnitude of the horizontal and vertical gradients and obtained values will be standard absolute value, which gives the correct edge localization.In order to achieve a high through put, the Sobel operator is executed in a parallel VLSI architecture. Both the horizontal and vertical gradient calculations are done simultaneously, which is a great time-saving process in comparison to the traditional software-based algorithms. This natural parallelism gives the suggested architecture the ability to process real-time image processing functions at a higher level of computational performance.

The optimization of power is one of the main goals of the proposed design, which is reached by means of incorporating a clock gating technique based on the flip-flop. Clock switching activity in digital circuit designs is a major source of dynamic power consumption in synchronous designs. Clock gating lowers this power loss by setting the clock signal to inactive blocks in the circuit during idle time.In the flip-flop-based clock gating scheme the enable signal is sampled by a D flip-flop on the upswing of the clock signal. The main clock is logically ANDed with the flip-flop output to form a clock that is gated. This guarantees a glitch-free operation since enable signal only changes at clock edges hence eliminating race conditions and hold-time violations. The gated clock is low when the system goes to the sleep mode, which essentially decreases the needless switching activity.The flip-flop clock gating method has superior timing stability over the latch-based clock gating system and is more appropriate in the ASIC and FPGA implementations. It makes timing closure simple and system stability greater, which is why it is a good choice in low-power digital signal processing.

The general architecture has image input Fig (10) modules, line buffers of creating 3×3 pixel windows, Sobel gradient computation units, magnitude calculation logic, clock gating control circuitry and output storage modules. The gated clock controls the operation of each processing block to ensure that they work efficiently using power consumption without performance loss.

Fig.10. Input

Fig.11. Output

Following detection of edges Fig (11), the processed pixel values are saved in a form of text so that they can easily be compared between the input image and output image in the simulation process. The resulting output data is reconstructed into the final edge-detected image and functional verification of the design is possible with the FPGA synthesis and simulation tools like Xilinx Vivado.The experimental results of the synthesis and simulation of the proposed Sobel edge detection system indicate that the new design has a higher edge detection accuracy, lower noise sensitivity, and lower power consumption. The architecture is designed to offer a low computational complexity and a high throughput but with optimal resource utilization, hence it is highly applicable in real-time embedded vision and low-power image processing application.

TABLE 1. COMPARISION OF EXISTING AND PROPOSED DESIGN

Methods	Area(LUT's)	Power(W)	Delay(ns)
Existing	38	14.424	15.591
Proposed	39	7.693	15.591

Table (1) compares the existing design with the proposed design in terms of area utilization, power consumption and delay. The proposed method only requires an addition of one LUT than the current design, which requires 38 LUTs. This slight increment in the hardware resources is insignificant and does not have a major impact on the efficiency of the overall area. Power consumption has been noted to be significantly improved. The proposed design would use 7.693 W, as compared to the current design that uses 14.424 W. This is a clear indication that the proposed method has high power reduction and therefore more applicable in power sensitive applications. Regarding delay, the two designs have the same timing performance of 15.591 ns. This demonstrates that the power loss is attained without affecting the velocity or timing nature of the system.

VI. CONCLUSION

To sum things up, edge detection has always been a basic process in most image processing programs and although the conventional Sobel operator has been admired due to its simplicity, it can not work well with fine edges as well as with noise. The proposed project has been able to solve these difficulties by applying a better version of Sobel edge detector algorithm with VLSI technology. The design is

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

done by means of gradient computation and hardware optimization using Verilog HDL on an FPGA platform, which allows it to achieve better accuracy, noise resistance, and even efficient parallel operation. The resulting architecture can provide low latency and power efficient performance and is hence suited to support real-time embedded vision applications. The findings of simulation with the help of such tools as Xilinx Vivado confirm the fact that the refined design does not only improve the quality of edge detection but also introduces less computational complexity, which offers a high-quality and high speed hardware implementation of advanced image analysis. The resultant architecture is capable of providing power efficient performance and it can be used in real-time embedded vision applications. The simulation with the help of such tools as Xilinx Vivado confirms that the better design not only improves the quality of the wall detection process but also decreases the computational load, which offers a stable and efficient hardware platform to perform more complex image processing procedures.

VII. REFERENCES:

[1]. Canny, J. (1986). A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence, (6), 679-698.
[2]. Gota, A., & Min, Z. J. (2013). Analysis and Comparison on Image Restoration Algorithms Using MATLAB. International Journal of Engineering Research & Technology (IJERT) Vol, 2, 1350-1360.
[3]. Mahalakshmi, A.,& Shanthini, B. (2016, January). A survey on image deblurring. In 2016 International Conference on Computer Communication and Informatics (ICCCI) (pp. 1-5). IEEE.
[4]. Flusser, J., Farokhi, S., Hoschl, C., Suk, T., Zitov ¨ a, B., ´ & Pedone, M. (2015). Recognition of images degraded by Gaussian blur. IEEE transactions on Image Processing, 25(2), 790-806.
[5]. Ramya, S.,& Christial, T. M. (2011, March). Restoration of blurred images using Blind Deconvolution Algorithm. In 2011 International Conference on Emerging Trends in Electrical and Computer Technology (pp. 496-499). IEEE.
[6]. Sada, M. M., & Mahesh, M. G. (2018). Image deblurring techniques—a detail review. Int. J. Sci. Res. Sci. Eng. Technol, 4(2), 15.
[7]. Ansari, M. A., Kurchaniya, D., & Dixit, M. (2017). A comprehensive analysis of image edge detection techniques. International Journal of Multimedia and Ubiquitous Engineering, 12(11), 1-12.
[8]. Syahrian, N. M., Risma, P., & Dewi, T. (2017). Vision-based pipe monitoring robot for crack detection using canny edge detection method as an image processing technique. Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, 243- 250.
[9]. Jain, S., & Goswami, M. S. (2015). A comparative study of various image restoration techniques with different types of blur. International Journal Of Research In Computer Applications And Robotics.
[10]. Yadav, S., Jain, C., &Chugh, A. (2016). Evaluation of image deblurring techniques. International Journal of Computer Applications, 139(12), 32- 36.
[11]. Verma, R., & Ali, J. (2013). A comparative study of various types of image noise and efficient noise removal techniques. International Journal of advanced research in computer science and software engineering, 3(10).
[12]. Sekehravani, E. A., Babulak, E., & Masoodi, M. (2020). Implementing canny edge detection algorithm for noisy image. Bulletin of Electrical Engineering and Informatics, 9(4), 1404-1410.
[13]. Saini, S., Kasliwal, B., & Bhatia, S. (2013). Comparative study of image edge detection algorithms. arXiv preprint arXiv:1311.4963
[14]. (Jan. 2021). European Commission, Industry 5.0: Towards a Sustainable, Human-Centric and Resilient European Industry. [Online]. Available: https://op.europa.eu/en/publication-detail/-/publication/468a892a-5097- 11eb-b59f-01aa75ed71a1/
[15]. P. K. R. Maddikunta, Q.-V. Pham, B. Prabadevi, N. Deepa, and K. Dev, ‘‘Industry 5.0: A survey on enabling technologies and potential applications,’’ J. Ind. Inf. Integr., vol. 26, pp. 1–31, Mar. 2022, doi: 10.1016/j.jii.2021.100257.
[16]. S. Prasath and D. N. Thanh, ‘‘Structure tensor adaptive total variation for image restoration,’’ Turkish J. Electr. Eng. Comput. Sci., vol. 27, pp. 1147–1156, Mar. 2019, doi: 10.3906/elk-1802-76. 4
[17]. D. N. H. Thanh, V. B. S. Prasath, L. M. Hieu, and S. Dvoenko, ‘‘An adaptive method for image restoration based on high-order total variation and inverse gradient,’’ Signal, Image Video Process., vol. 14, no. 6, pp. 1189–1197, Feb. 2020, doi: 10.1007/s11760-020-01657-9.
[18]. B.-H. Chen, Y.-S. Tseng, and J.-L. Yin, ‘‘Gaussian-adaptive bilateral filter,’’ IEEE Signal Process. Lett., vol. 27, pp. 1670–1674, Sep. 2020, doi: 10.1109/LSP.2020.3024990.
[19]. A. Gabiger-Rose, M. Kube, R. Weigel, and R. Rose, ‘‘An FPGA-based fully synchronized design of a bilateral filter for real-time image denoising,’’ IEEE Trans. Ind. Electron., vol. 61, no. 8, pp. 4093–4104, Aug. 2014, doi: 10.1109/TIE.2013.2284133.
[20]. S. D. Dabhade, G. N. Rathna, and K. N. Chaudhury, ‘‘A reconfigurable and scalable FPGA architecture for bilateral filtering,’’ IEEE Trans. Ind. Electron., vol. 65, no. 2, pp. 1459–1469, Feb. 2018, doi: 10.1109/TIE.2017.2726960.
[21]. C.-Y. Lien, C.-H. Tang, P.-Y. Chen, Y.-T. Kuo, and Y.-L. Deng, ‘‘A low-cost VLSI architecture of the bilateral filter for real-time image denoising,’’ IEEE Access, vol. 8, pp. 64278–64283, 2020, doi: 10.1109/ACCESS.2020.2984688.
[22]. S.-J. Jang and Y. Hwang, ‘‘Noise-aware and light-weight VLSI design of bilateral filter for robust and fast image denoising in mobile systems,’’ Sensors, vol. 20, pp. 1–15, Aug. 2020, doi: 10.3390/s20174722.