

Implementation of Low Power VLSI Architecture for Sobel Edge Detection

**Yellamraju Sri Chakrapani¹, Nandanavanam Venkateswara Rao²,
Maddu Kamaraju³**

¹Research Scholar, Department of ECE, Jawaharlal Nehru Technological University, Kakinada-533003, India. Email: srichakrapani@gmail.com.

²Professor and Dean (R&D), Department of ECE, Bapatla Engineering College, Bapatla-522102, India. Email: nvr68@gmail.com.

³Professor and Director (AS&A), Department of ECE, Seshadri Rao Gudlavalleru Engineering College, Gudlavalleru-521356, India. Email: profmkr@gmail.com.

Abstract -In image processing, the noticeable variations of the image can be found using the edge detection process. This edge detection process facilitates to identify placement and edges of an object. The Sobel edge detection process is preferred since it is simple and more noise immune. Sobel edge detection calculates gradients of a pixel in horizontal and vertical directions concerning neighboring pixels. The gradients absolute quantities are computed, added, and matched to a threshold for finding whether the pixel is an edge pixel or not. In this paper, the Sobel edge detection system VLSI architecture is designed which consumes less power by using the arithmetic blocks like Brent Kung adders. The source image of different resolutions is taken to perform Sobel edge detection and implemented using various FPGA devices by analyzing power dissipation, delay and device utilization summary. Then, simulation, synthesis processes are performed and finally, power analysis is performed using XPower Analyzer of Xilinx ISE software.

Keywords: Edge Detection, Sobel, Pixels, Power dissipation, FPGA, VLSI.

1 Introduction

The ICs manufactured for several applications are developed using VLSI design flow. In the design flow, there are various levels of design such as system level, algorithmic level, architecture level, gate level, circuit level, layout level, etc. The architecture level is a very important step for designing different modules in the application. The VLSI architectures are developed for obtaining high speed, low power dissipation, less area, and complexity. Edge detection is a significant method in the computer vision field for object detection. In edge detection, the edge pixels are extracted using the difference in the brightness level of the pixels. These edge pixels are the boundaries for the detected object in the image.

The various edge detection methods are Prewitt, Canny, Roberts Cross, Differential, Sobel, etc. (Neoh et al. 2005) Prewitt edge detection uses a discrete differentiation operator to perform the image gradient for detecting the edges. Canny edge detector performs five process algorithm to detect the edges. Roberts Cross edge detection performs the sum of squares to find the difference between neighbor pixels. Differential edge detection performs the differential method of second-order derivative for detecting pixel edges. Sobel edge detection performs convolution of the image pixels with a kernel matrix to compute the gradients in vertical and horizontal directions, absolute gradients, sum of absolute gradients, and then compare with a threshold to find the pixels, which form the edge of an object.

In this paper, the Sobel edge detector is used instead of other edge detection methods, due to its simplicity and being less immune to noise. The edge detection system developed using hardware such as an FPGA, is more suitable for real-time applications since it has high performance, low power dissipation, less area, etc. Nowadays, power dissipation is considered an important factor for developing any real-time system, particularly for battery-operated systems. Low power architecture is proposed for developing a Sobel edge detection system in this paper.

2 Literature Review

For obtaining high performance, many VLSI architectures are specified for computer vision. These architectures are implemented using GPU, ASIC, and FPGA. The ASIC implementation offers high speed and low power but more design time. The GPU implementation achieves high performance with greater clock speed, but consumes

more power and is less hardware flexible. So, nowadays FPGA is preferred for the development of computer vision devices (Lim et al. 2013). Parallel architectures are used for implementing serial algorithms for increasing the speed and since the FPGA has lower clock frequencies, the power consumption is also less. There are a few challenges for designing these architectures in FPGA. The hardware components such as memory, IOBs, LUTs are limited in FPGA, which have to be fit for computer vision applications (Bailey et al. 2011).

Nowadays, electronic gadgets operate on batteries, which require more battery charge and life. To satisfy these requirements, power dissipation is very important parameter in the design of an Integrated Circuit (IC) in these electronic gadgets using CMOS VLSI technology. If low power dissipation is achieved, then battery consumption will also be reduced. Before discussing the methods of low power dissipation, the cause of power consumption in CMOS technology have to be examined. There are three sources of power dissipation, i.e., leakage (static) power, short-circuit power, and dynamic (switching) power (Chandrakasan et al. 1995). Whenever the system/circuit is in idle mode, then the power dissipated is static, which depends on the leakage current that occurs due to the flow of minority carriers in the sub-threshold region. The short circuit power occurs when current flows directly from the supply to the ground. The dynamic power dissipation occurs whenever the system/circuit signals change.

The power optimization methods at the system level are low-frequency clocks, off-chip components like ROM, RAM integration, etc. At the algorithm level, minimizing the no. of operations, conditions and loop iterations will reduce the power dissipation. Parallel, pipelining, arithmetic architectures are used to minimize power dissipation at the architecture level. To optimize the power in logic or gate level, switching activity reduction, clock, and bus loading optimization are proposed. At the transistor level, the techniques used to minimize the power dissipation are transistor sizing, multi-threshold voltages, etc. If the methods like device scaling, optimization in placement, and routing are applied for device level, then low power dissipation is obtained. In this paper, architecture level optimization of power dissipation is proposed by using low-power arithmetic blocks for implementing a Sobel edge detection system.

Several VLSI architectures are proposed Sobel edge detection by various researchers, for improving the performance. In Abbasi et al. 2007, Sobel edge detection was developed in FPGA, which lacks simplicity and takes more delay to execute. Later in Girish et al. 2014, the architecture has modified, which has less delay when compared to Abbasi et al. 2007. Stochastic computing is proposed for implementing Sobel edge detection (Hounghun et al. 2019). Sobel edge detection has been developed using modified architecture and implemented in FPGA, which has fewer computations, parallelism, etc. (Halder et al. 2012) The reduction of complexity in the Sobel edge detection algorithm is based on parallel architecture in Khalid et al. 2012.

Pipeline architecture was presented in Nausheen et al. 2018, for developing Sobel edge detection. Different FPGAs are used for implementing Sobel edge detection. (Osman et al. 2010) The pipelining architectures are proposed to implement a Sobel edge detection algorithm and then displayed using VGA (Video Graphics Array) (Rajesh et al. 2012). The improved processor architecture is employed in implementing Sobel edge detection (Vanishree et al. 2013). The performance of Sobel edge detection has increased by reusing the parallel architecture of FPGA. (Taslimi et al. 2020) The area and power dissipation are also reduced in this hardware implementation. Most of the architectures are not optimal, since they may improve speed but not reduce power dissipation (Rao, K. M et.al. 2024).

3 Methodology

The source image in pixel matrix form as shown in Fig. 1(a), with a given resolution, is taken and the gradient is determined by considering the neighboring points of an image pixel and 3x3 kernel matrices in the horizontal and vertical direction as shown in Figure 1(b) and 1(c) respectively (Vasimalla, Y et.al 2023).

P0	P1	P2	-1	0	+1	-1	-2	-1
P3	P4	P5	-2	0	+2	0	0	0
P6	P7	P8	-1	0	+1	+1	+2	+1
(a)			(b)			(c)		

Fig. 1 (a) Source image pixel matrix, (b) Kernel matrix in the horizontal direction, (c) 3x3 kernel matrix in the vertical direction

The gradients G_x and G_y are determined by performing convolution operation of source image pixel matrix with the kernel matrices, which are also known as convolution masks, in the x-direction (horizontal) and y-direction (vertical) respectively. G_x and G_y are computed using Equations (1) and (2) respectively.

$$G_x = ((P6 - P0) + 2(P7 - P1) + (P8 - P2)) \quad (1)$$

$$G_y = ((P2 - P0) + 2(P5 - P3) + (P8 - P6)) \quad (2)$$

Where, P0, P1, P2, P3, P5, P6, P6, P7, P8 are the neighboring pixel of the image pixel P4, to which the gradient

has to be calculated. The absolute values of the gradients G_x and G_y are obtained from the calculated gradient of a pixel. The sum of these absolute values of the gradients, abs_G_x and abs_G_y is computed, as shown in Equation (3).

$$\text{Sum} = abs_G_x + abs_G_y \quad (3)$$

Now, the sum gradient is compared with some default threshold value, by which the image pixel P4 is decided whether it is an edge pixel or not. If the threshold value is less than the sum gradient, then the image pixel P4 is an edge pixel and if it is more, then image pixel P4 is not an edge pixel (Ramaiah, V. S et. al 2021).

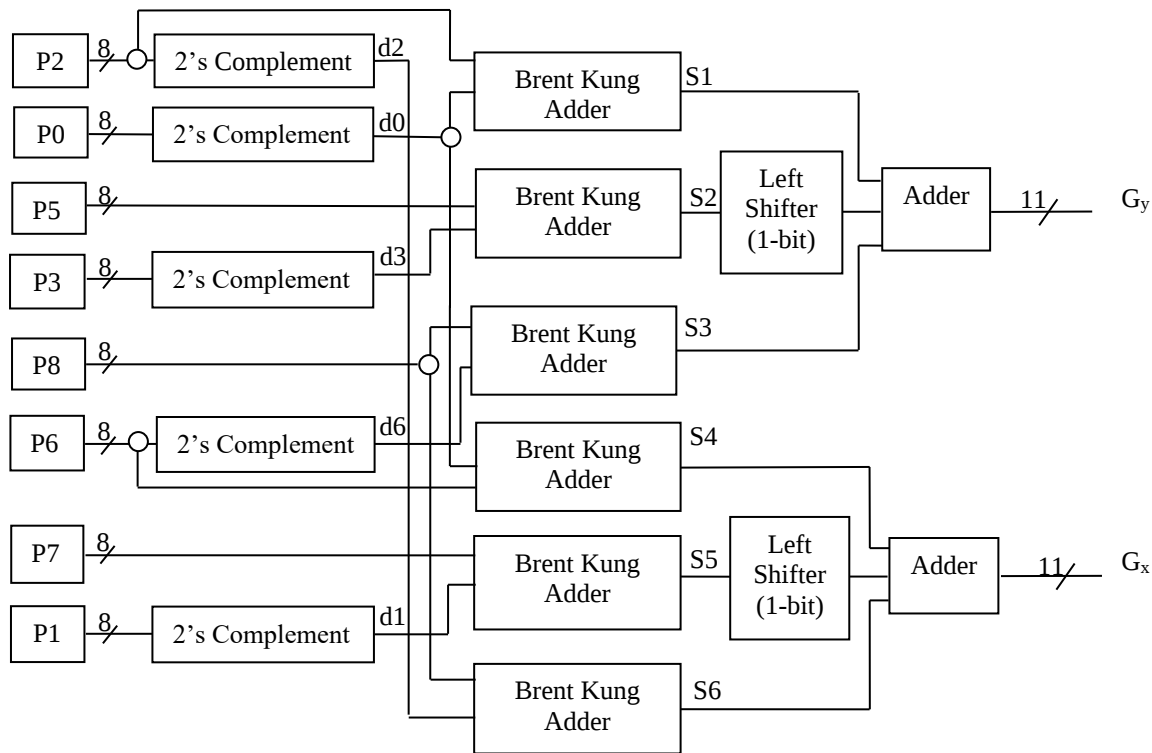


Fig. 2 Gradient calculation architecture using Brent Kung Adders

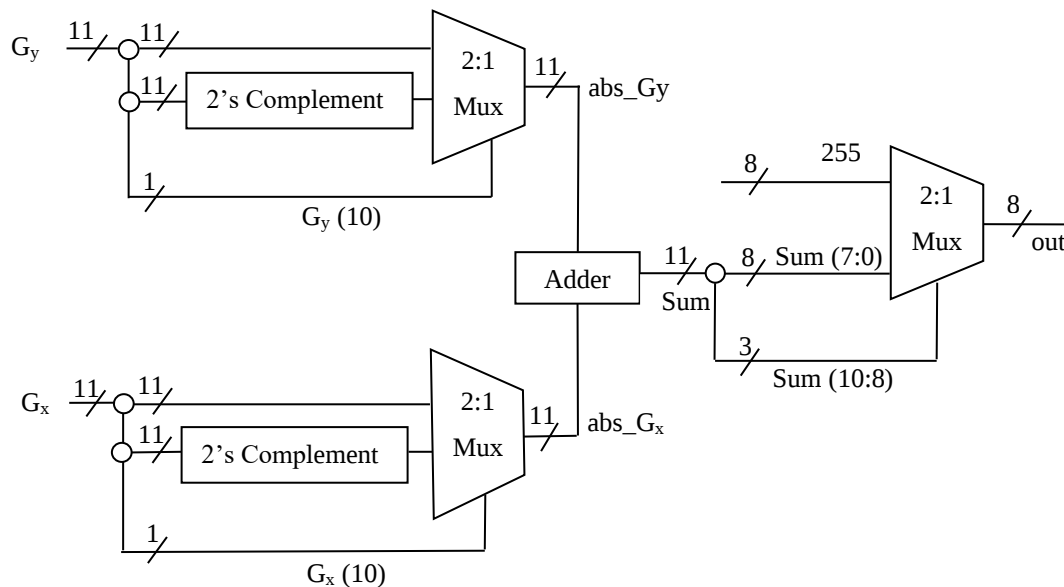


Fig. 3 Edge Pixel detection architecture

In this paper, the low power VLSI architecture of Sobel architecture is presented, which calculates the gradient values, G_x in X-direction, G_y in Y-direction, using the input source image pixels P0 to P8, 2's compliments, 1-bit

shifters and Brent-Kung adders, as shown in Fig. 2.

The intermediate signals d_2 , d_0 , d_3 , d_6 , d_1 are generated by performing 2's complement of source image pixels P_2 , P_0 , P_3 , P_6 , and P_1 respectively. The intermediate signals S_1 to S_6 are the sum outputs of the six Brent–Kung adders (Vadlamudi, M. N et.al 2024). Then, S_2 and S_5 signals are shifted 1-bit left to obtain the signals LS_2 and LS_5 respectively (Saikumar, K et.al 2023).

Finally, the gradient value G_x is computed by adding the signals S_4 , LS_5 , S_6 and gradient value G_y is computed by adding the signals S_1 , LS_2 , S_3 respectively. Fig. 3 shows the Edge Pixel detection architecture, in which the absolute values abs_G_x , abs_G_y of horizontal and vertical gradients, are obtained using 2:1 Multiplexer blocks and then “sum” signal is obtained by adding abs_G_x and abs_G_y . Finally, the “sum” signal is compared with threshold (hexadecimal value FF) using multiplexer to find whether it is an edge pixel or not and 8-bit “out” edge detected signal is generated.

The low power arithmetic block such as Brent–Kung adder is presented to implement Sobel edge detection. The Brent–Kung adder can be derived from Parallel Prefix Adders (PPA), where switching activity is minimized by parallel computation of carry signals using generate and propagate signals (Brent et al. 1982), which is shown in Fig. 4.

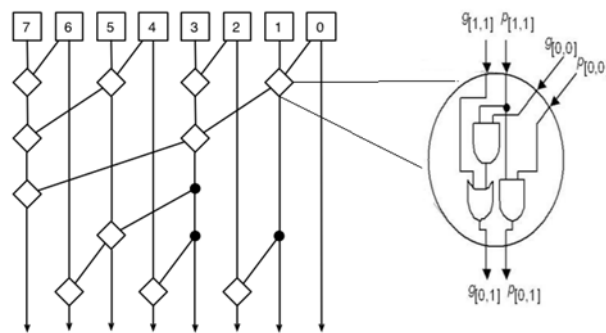


Fig. 4 8-bit Brent–Kung adder

The entire process given in Fig. 5 shows the implementation flow chart of Sobel edge detection. Source images with different resolutions are taken, and the pixels are extracted as hexadecimal values, which are given as an input text file to Xilinx ISE Software.

In this software, Verilog HDL is used for describing the Sobel edge detection algorithm, whose steps are given below.

Step 1: Initialize the module with pixel values P_0 to P_8 , clock signals as inputs, and “out” signal as output.

Step 2: Declare the reg variables G_x , G_y , abs_G_x , abs_G_y , Sum.

Step 3: Declare the sum and carry signals for Brent-Kung Adder as S_1 , S_2 , S_3 , S_4 , S_5 , S_6 .

Step 4: Calculate the sum and carry signals for Brent-Kung Adder using the standard expressions.

Step 5: Now, the clock signal is used for synchronizing the further steps.

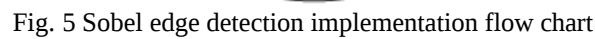
Step 6: The sum and carry signals obtained in Brent-Kung Adders are used to calculate Sobel mask G_x (horizontal direction gradient) and G_y (vertical direction gradient).

Step 7: Find absolute values of G_x and G_y , i.e. abs_G_x and abs_G_y .

Step 8: After adding abs_G_x and abs_G_y , the Sum signal is obtained, which is tested for an edge pixel.

Step 9: The sum signal is compared with a threshold, if it is greater, then that pixel is an edge and else it is not an edge pixel.

The simulation process is performed after the development of Verilog HDL code. The edge and non-edge pixels generated during the simulation is converted into text file and converted to edge detected image using MATLAB. FPGA device utilization summary, delay is analyzed during the synthesis process. The power dissipated is analyzed for various clock frequencies and source image resolutions using XPower tool of Xilinx ISE.



The pixels P0 to P8 represented in hexadecimal values are taken from a source image of different resolutions and are given as input to the Verilog HDL code, described from the algorithm steps given above. After that, the simulation process is performed and resultant waveforms are shown in Fig. 6. “out” signal value determines whether it is edge pixel or not.





Fig. 7 Schematic of Sobel edge detection

Next, synthesis process is performed using the FPGA device 6VLX240TL, in which the RTL (Register Transfer Level) schematic information is shown in Fig. 7.

Also, power analysis, delay and device utilization are performed for the Sobel edge detection implemented using different FPGAs, which is shown in Table 1. The power dissipated using Spartan6 FPGA is less when compared to other FPGAs since it has typical supply voltage of 1.0V. The delay is minimum for ZynQ-7000 and Kintex7 FPGAs since the architecture consists of fastest logic fabric. The device utilization of the FPGAs is almost same, when implemented sobel edge detection, as shown in Table 1.

Table 1 Comparison of power dissipation, delay and device utilization for FPGAs implementing Sobel edge detection

Name of the FPGA	LUT Flip Flop pairs	Slice LUTs	Delay (ns)	Power Consumption (mW)
Virtex6	167	167	5.93	79
Virtex6 low power	167	167	7.206	68
Virtex4	224	127	10.342	62
Virtex5	151	151	9.6	48
Spartan 3A	234	132	17.721	39
ZynQ-7000	167	167	5.791	31
Kintex7	167	167	5.791	31
Spartan6	167	167	13.017	6
Spartan6 low power	167	167	20.484	8.21

Table 2 Comparison of power dissipation results

References	Clock Frequency (MHz)	Image Size	Power Dissipation (mW)
Rajesh et al. (2012)	148.133	----	103.13

Khalid et al. (2012)	300	640x480	27.31
Taslimi et al. (2020)	129	512x512	8.45
Brent Kung Adder (Proposed)	129	512x512	8.21

Table 2 shows the comparison results of power dissipation, in which the proposed architecture of the Sobel edge detection consumes less power (8.21mW) when compared to other architectures of Sobel edge detection with clock frequency 129MHz due to the usage of brent kung adders in the proposed architecture of Sobel edge detection for the source image resolution of 512x512.

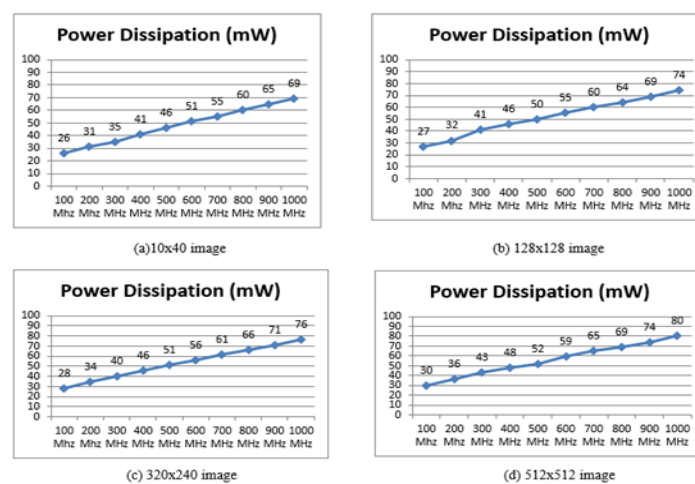


Fig. 8 Comparison between power dissipation and clock frequency for different image resolutions

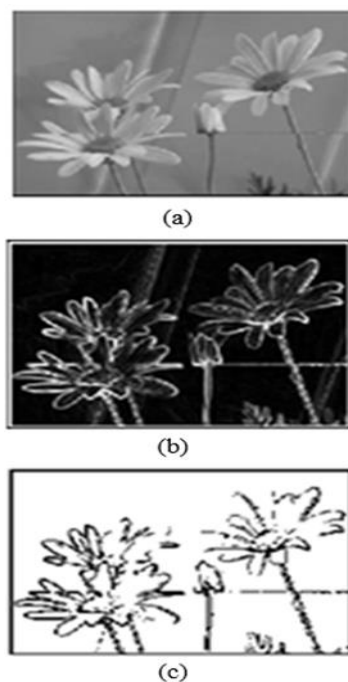


Figure 9: (a) Source image (b) Sobel gradient image (c) Edge detected image

Sobel edge detection is implemented for source images with different resolutions such as 10x40, 128x128, 320x240, 512x512 etc. and graphs are obtained comparing input clock frequency and output power dissipation as shown in Fig. 8 (a), (b), (c), (d).

From the graphs given Fig. 8, it is observed that the power dissipation increases as the clock frequency increases for different resolutions of source images. The power dissipation of 512x512 image resolutions slightly high when compared to other image resolution due to processing of more no. of pixels and hence increased operations.

The image shown in Fig. 9(a) below represents the original image. After applying gradients in horizontal and vertical directions, the original image changes to Sobel gradient image as shown in Fig. 9(b). Finally, the edge detected image shown in Fig. 9(c) is obtained after performing Sobel edge detection by finding the absolute gradient and comparing with the given threshold value to obtain edge detection pixels.

5 Conclusion

Low-power architecture is presented for developing and implementing the Sobel edge detection system in this paper. Brent Kung adders are used in the hardware architecture, which reduces the number of carry calculations in consecutive stages, thereby reducing the switching activity and power dissipation. The Sobel edge detection system was developed in MATLAB, Xilinx Software and in which a power of 8.21mW was dissipated. Future work will include, incorporating low power design along with high-speed techniques and comparing the Sobel edge detection with other edge techniques like Canny, Robert Cross, Prewitt Edge detection.

References

- [1] Abbasi TA, Abbasi MU (2007) A novel FPGA-based architecture for Sobel edge detection operator. *Int J of Electronics* 94(9): 889-896
- [2] Bailey DG. (2011) Adapting algorithms for hardware implementation. *Proc IEEE CVPR Workshop* 177–184
- [3] Brent RP, Kung HT (1982) A Regular Layout for Parallel Adders. *IEEE Trans on Comput* 31(3):260-264
- [4] Chandrakasan AP, Brodersen RW (1995) Minimizing Power Consumption in CMOS Circuits. *Proc IEEE* 83(4):498-523
- [5] Girish C, Daruwala RD (2014) Design of Sobel Operator based Image Edge Detection Algorithm on FPGA. *Proc IEEE Int Conf on Comm and Signal Processing*: 788-792
- [6] Halder S, Bhattacharjee D, Nasipuri M, Dipak Kumar B (2012) A fast FPGA based architecture for Sobel edge detection. *Proc 16th Int Conf on Progress in VLSI Design and Test, Lecture Notes in Computer Science, Springer-Verlag Berlin, Heidelberg* 7373: 300-306
- [7] Hounghun J, Youngmin K (2019) Novel Stochastic Computing for Energy-Efficient Image Processors. *Electron Open Access J* 8(6):72
- [8] Khalid AR, Paily R (2012) FPGA IMPLEMENTATION OF HIGH SPEED AND LOW POWER ARCHITECTURES FOR IMAGE SEGMENTATION USING SOBEL OPERATORS. *J of Circuits, Syst and Comput* 21(7):1250050-14
- [9] Lim YK, Kleeman L, Drummond T (2013) Algorithmic methodologies for FPGA-based vision. *Machine Vision and Applications* 24(6): 1197-1211
- [10] Nausheen N, Seal A, Khanna P, Santanu H (2018) A FPGA based Implementation of Sobel Edge Detection. *Microprocessors and Microsystems* 56: 84-91
- [11] Neoh HS, Hazanchuk A (2005) Adaptive Edge Detection for Real-Time Video Processing using FPGAs.
- [12] Osman ZEM, Hussin FA, Zain Ali NB (2010) Hardware Implementation of an Optimized Processor Architecture for Sobel Image Edge Detection Operator. *Proc of IEEE Int Conf on Intelligent and Advance Syst*: 1-4
- [13] Rajesh M, Verma R (2012) Area Efficient FPGA Implementation of Sobel Edge Detector for Image Processing Applications. *Int J of Comput Applications* 56(16): 7-11
- [14] Taslimi S, Faraji R, Aghasi A, Reza Naji H (2020) Adaptive Edge Detection Technique Implemented on FPGA. *Iran J of Science and Technology, Tran of Electrical Engineering* 44:1571–1582
- [15] Vanishree, Ramana Reddy KV (2013) Implementation of Pipelined Sobel Edge Detection Algorithm on FPGA for High-Speed Applications. *Proc. of Int Conf on Emerging Trends in Communication, Control, Signal Processing and Comput Applications (C2SPCA)*: 1-5.

- [16] Rao, K. M., Kumar, P. S., Reddy, T. V., Nilima, D., Saikumar, K., & Khlaif, A. M. (2024, April). Ultra Low Power High Speed DFT Implementation For ASIC SoC. In 2024 IEEE 9th International Conference for Convergence in Technology (I2CT) (pp. 1-6). IEEE.
- [17] Vasimalla, Y., Pradhan, H. S., Pandya, R. J., Saikumar, K., Anwer, T. M. K., Rashed, A. N. Z., & Hossain, M. A. (2023). Titanium dioxide-2d nanomaterial based on the surface plasmon resonance (SPR) biosensor performance signature for infected red cells detection. *Plasmonics*, 18(5), 1725-1734.
- [18] Ramaiah, V. S., Singh, B., Raju, A. R., Reddy, G. N., Saikumar, K., & Ratnayake, D. (2021, March). Teaching and Learning based 5G cognitive radio application for future application. In *2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)* (pp. 31-36). IEEE.
- [19] Vadlamudi, M. N., Jayanthi, N., Swetha, G., Nishitha, P., Al-Salman, G. A., & Saikumar, K. (2024, April). IoT Empowered GNSS Tracking in Real-time via Cloud Infrastructure. In *2024 IEEE 9th International Conference for Convergence in Technology (I2CT)* (pp. 1-6). IEEE.
- [20] Saikumar, K., Ahammad, S. H., Vani, K. S., Anwer, T. M. K., Hadjouni, M., Menzli, L. J., ... & Hossain, M. A. (2023). Improvising and enhancing the patterned surface performance of MIMO antenna parameters and emphasizing the efficiency using tapered miniature sizes and layers. *Plasmonics*, 18(5), 1771-1786.