Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2025 SCPE. Volume 26, Issues 2, pp. 599–609, DOI 10.12694/scpe.v26i2.4077

DESIGN AND DEVELOPMENT OF MEMORY PIXEL ARCHITECTURE FOR SOBEL EDGE DETECTION

YELLAMRAJU SRI CHAKRAPANI *, NANDANAVANAM VENKATESWARA RAO †, AND MADDU KAMARAJU ‡

Abstract. Memory is a fundamental hardware structure used in various electronic and multi-media products. The design and development of large-size memory architectures have become increasingly complex due to the growing demands of high-resolution image processing applications. This research work aimes to design a specialized memory architecture module for Sobel edge detection, an essential technique in image processing. The proposed architecture (memory pixel size) consists of memory unit, comprising rows and columns, is determined by the image resolution. For effective Sobel edge detection, the image pixels must be stored in memory, and read operations are performed to access a 3x3 matrix of nine pixels. A critical consideration in developing this memory architecture is power dissipation, which is mitigated through the application of Clock Gating techniques. The proposed pixel memory architecture is implemented using MATLAB and Xilinx ISE software with Verilog HDL. The image pixel memory is developed using Block RAMs (BRAMs) and registers, and the 3x3 pixel matrix required for the Sobel edge detector is generated. Simulation, synthesis, and power analysis are conducted for the image pixel memory across source images with various resolutions, including 10x40, 10x20, 128x128, 320x240, and 512x512. The results indicate reduced power dissipation from 30% to 40% due to Clock Gating. This work demonstrates the effectiveness of the proposed memory architecture in reducing power consumption while maintaining performance. Future work aims to further enhance the performance of image pixel memory by decreasing the number of registers and improving pixel access times. Such module will provide a more efficient and scalable solution for high-resolution image processing applications.

Key words: Memory architecture, Image pixels, Power dissipation, Clock Gating

1. Introduction.

An image is divided into small elements called pixels, each representing a different color. For example, an image with a resolution of m by n has m x n pixels. A 240 by 320 resolution image contains 76,800 pixels. The pixels are arranged in a matrix, where 'm' denotes the number of rows and 'n' denotes the number of columns. The pixel values, which range from 0 to 255, represent their intensities. In a grayscale image, a pixel intensity of 0 represents black, while an intensity of 255 represents white. Each pixel can be stored in an 8-bit register. The memory required to store the pixels is determined by the image resolution, with the address range corresponding to the number of rows. The data retrieved from a specified address represents the pixels in the columns [1].

Designing the hardware for image pixel memory is complex, especially when considering FPGA, due to the numerous registers and block memories required. Additionally, power dissipation is a crucial parameter to consider. This paper focuses on reducing power dissipation in the designed image pixel memory.

Few researchers have explored the development of memory architectures specifically for applications like Sobel edge detection systems. These architectures use input and output buffers, BRAM (Block Random Access Memory), registers, etc. Various methods, such as using external memory, registers with data reuse, BRAMs with data reuse, and combinations of registers and BRAMs with data reuse, are employed in designing image pixel memory for Sobel edge detection. Partitioning image frames into BRAMs utilizes all FPGA resources and reduces power dissipation . Techniques like partitioning, data reuse, loop pipelining, and merging are integrated into an optimized flow for behavioral synthesis . The clock gating technique, synchronized with the global clock, analyzes two clock gating methods.

*Department of ECE, Jawaharlal Nehru Technological University Kakinada, Kakinada, Andhra Pradesh, India (srichakrapani@gmail.com).
†Department of ECE, Bapatla Engineering College, Bapatla, Andhra Pradesh, India (nvr68@gmail.com).,
‡Department of ECE, Seshadri Rao Gudlavalleru Engineering College, Gudlavalleru, Andhra Pradesh, India (profmkr@gmail.com.).

A comparison of clock gating techniques implemented on FPGA and ASIC shows that FPGA implementation reduces dynamic power consumption by 50% to 80% compared to ASIC. Several low power dissipation techniques for System on Chip (SoC), such as clock and power gating, multi-voltage, and voltage scaling, are discussed. The RTL clock gating method with scan techniques is implemented in Synopsys Power Compiler to reduce power consumption and enhance Design For Test (DFT). Energy consumption from computations can be reduced using parallelism in FPGA memory architecture . The concept of gated clocking involves stopping the clock signal when idle, thereby saving power consumed by latches in this state.

In hardware, the Frame Buffer, Line Buffers, and Pixel Window are the most important parts for performing Sobel edge detection. The whole picture is kept in memory by the Frame Buffer, usually in the form of a 2D array with a pixel for each piece. In order to keep the current line and the two lines before it and after it in memory while the picture is being processed, line buffers are necessary. A 3x3 Sobel operator requires a minimum of three line buffers. By getting the correct neighborhood from the line buffers, the 3x3 Pixel Window may traverse the image and apply the Sobel operator to each pixel. Effective and efficient edge identification over the whole image is guaranteed by this organized technique [2].

1.1. Low Power Dissipation. Nowadays, electronic gadgets operate on batteries, which require more battery charge and life. To satisfy these requirements, power dissipation is an important factor in developing an Integrated Circuit (IC) in these electronic gadgets using CMOS VLSI technology. If low power dissipation is achieved, battery consumption will also be reduced. There are three sources of power dissipation, i.e leakage (static) power, short-circuit power, and dynamic (switching) power [8]. 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 dissipation arises whenever both NMOS and PMOS transistors are in active or saturation regions, i.e the power supply is directly connected to the ground. The dynamic power dissipation occurs whenever the system circuit signals change, which depends on switching activity per node (β), Switched Capacitance (C), Frequency (switching events per second, F), and Supply Voltage (VDD).

The power reduction methods are suggested at various abstraction levels of CMOS VLSI design flow such as system level, algorithm level, logic or gate level, transistor level, transistor level, etc. The power optimization methods in system level are low frequency clocks, off-chip components like ROM, RAM integration, etc. In 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 in architecture level.

To optimize the power in logic or gate level, switching activity reduction, clock and bus loading optimization are proposed. In transistor level, the methods such transistor sizing, multi- threshold voltages etc. are employed for low power. If the methods like device scaling, optimization in placement and routing are applied for device level, then low power dissipation in obtained. In this paper, gate or logic level optimization of power dissipation is proposed by using gated clock technique for the implementation of image pixel memory for sobel edge detection system An efficient architecture for image pixel memory usually entails arranging pixel data such that it can be accessed and processed quickly. A 3x3 neighbourhood of pixels must be readily accessible for Sobel edge detection to work. This necessitates giving serious thought to the storage and access of pixels.

2. Literature survey. There has been a lot of research on memory architecture design and optimization for image processing applications, especially Sobel edge detection. This literature review summarises significant findings in optimizing FPGA memory designs, memory allocation, power minimization, and low-power designs using clock gating techniques. Optimal FPGA memory architectures for Sobel edge detection were the primary focus of Harald Devos and Dirk Stroobandt [1]. In this work proved that optimized memory architectures are crucial to making FPGA-based edge detection algorithms work better. This research shows how difficult it is to create FPGA-based solutions while balancing memory size, access speed, and computational efficiency. Deepayam Bhowmik, Robert Stewart, Greg Michaelson, and Andrew Wallace [2] investigated solutions for optimized memory allocation and power minimization for FPGA-based image processing. Their work, tackled the pressing issue of reducing power consumption and memory utilisation in high-resolution image processing jobs. Optimal memory allocation was highlighted as a critical component in their approaches to drastically reduce power consumption without sacrificing performance. In [3], Peng Zhang, Xu Cheng, and Jason Cong presented an integrated and automated memory optimization method for FPGA behavioral synthesis. This

Table 2.1: Comparison of Various Techniques for Sobel Edge Detection Architecture

Details	Technique	Research Gap	Limitations
Joshi, R et.al 2020 [11]	Novel bit-sliced near-memory computing	Complex Design	High power consumption
Osman, Z.E.M et al. 2010 [12]	Optimized processor	High Latency	Critical architecture
Singh, S et.al 2014 [13]	Analysis of hardware architectures	Deep analysis	Future demands
Kumar, S.et.al 2013 [14]	Segmentation with edge detection	FPGA prototype	Latency
Saidani, T et.al 2024 [15]	Model-based design method	Model edge with IoT	Power consumption
Pudi, D et.al 2023 [16]	DRRA and DiMArch architectures	Cannot support IoT models	Operational efficiency is less
Chang, Q et.al 2023 [17]	Multi-directional kernel	Operations possible with GPU processor	High power consumption
Yamini, V et.al 2024 [18]	SoC design with edge	Critical architecture	Less performance
Khalil, A. S et.al 2023 [19]	Enhanced system on chip	Less computational module	High latency
Orthy, M., et.al 2023 [20]	FPGA-based image	Faster performance Sobel edge model	High power consumption

methodology is optimizing memory utilization during behavioral synthesis of FPGA designs . By automating the optimization process, the suggested flow hoped to increase memory efficiency while decreasing design time. Low-power sequential circuit design using clock gating methods was investigated by M. Pedram and Xunwei Wu [4]. By selectively blocking the clock signal to inactive circuit portions, gives Fundamental Theory and Applications—showed how clock gating may drastically cut power usage. To design memory structures that use less power, this method is essential. A new and comparative assessment of clock gating in FPGAs was given by J. Raivainen and A. Mammela [5]. The work was concentrated on how clock gating may be used to reduce power consumption in FLPGA architectures. The study shed light on the advantages and methods of clock gating in systems that rely on field-programmable gate arrays (FPGAs).

Low Power Methodology Manual For System-on-Chip Design, written by David Flynn, Robert Aitken, Alan Gibbons, and Kaijian Shi [6], is an all-inclusive reference on low power design approaches, such as clock gating. An excellent resource for engineers and designers, this document offers practical solutions for attaining low power consumption in system-on-chip (SoC) systems.

Power reduction by RTL clock gating was covered in Mark Biegel's [7] presentation at SNUG, San Jose. The goal of Biegel's research was to find ways to drastically reduce power consumption by using clock gating at the register-transfer level (RTL). By incorporating power-saving strategies early on in the design process, designers may create more efficient goods in the end.

Methods for reducing digital CMOS circuit power consumption were investigated by R. W. Brodersen [8]. His work laid the groundwork for low-power design in digital circuits, including techniques like clock gating.

The effect of memory design and parallelism on FPGA communication energy was studied by A. Dehon and D. Lakata [9]. Their research showed that FPGA designs might minimize communication energy via the use of parallel processing and optimized memory structures. The significance of memory design to FPGA systems' total energy efficiency is highlighted by this study. To reduce power consumption in sequential circuits, methods for synthesizing gated clocks were proposed by P. Siegel and G. D. Micheli [10]. Their work, which was Tested and contributed to the field of low-power electronics by providing realistic approaches for incorporating clock gating in sequential circuit designs.

Table 2.1 clealry explains about various sobel edge detection models and its limitations.

3. Image Pixel Memory Architecture for Sobel Edge Detection. The proposed image pixel memory architecture for generating sobel edge detection pixel matrix is designed and developed for m x n image resolution, which contains m rows and n columns, as shown in Fig 3.1. The address is used to access the rows

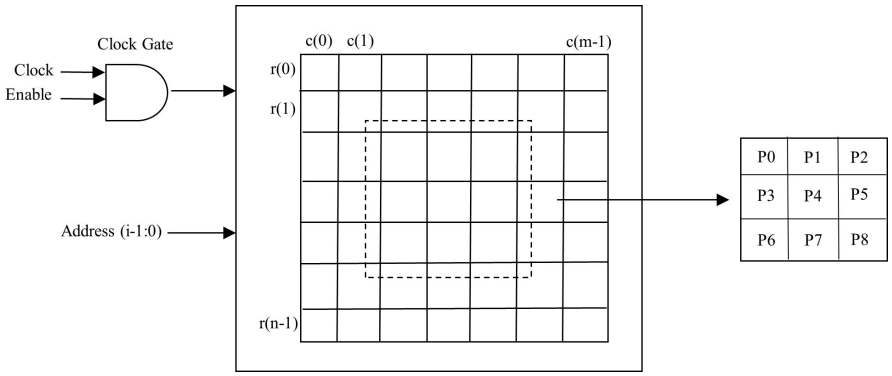

Fig. 3.1: Image pixel memory system and Pixel matrix for Sobel edge detection

from r(0) to r(n-1). Three consecutive rows are accessed to generate P0 to P8 pixels, representing the input matrix for computing sobel edge detection, and each pixel in the memory location comprises 8 bits. The 3x3 pixel matrix will be continuously generated from the image pixel memory from all the successive three rows and all columns i.e. first three rows r(0) to r(2) and columns c(0) to c(m-1), after that rows r(1) to r(3) and columns c(0) to (m-1), in the last rows r(n-3) to r(n-1) and columns c(0) to (m-1). The power dissipated during the implementation of image pixel memory can be minimized by employing clock gating [21]. In this technique, an enable signal is used for controlling the clock signal, i.e., whenever the clock signal is unnecessary, the enable signal is '0', and when it is required, the enable signal is '1'. So, the power consumed by the clock signal is reduced whenever idle or not required [22].

3.1. Image Dataset for experiment. In this section, dataset availability and its testing process have been explained. Many different picture libraries and image-processing tools may be found on SourceForge, an active site for open-source projects [23]. Anyone looking for open-source image libraries may utilize SourceForge's search tool to locate them. Just type in terms like "image library" or "image processing." Finding these projects is also more accessible by perusing the "Software" section and sorting by applicable categories. The OpenCV, ImageMagick, and GIMP open-source image libraries are some of the most prominent ones on SourceForge [24]. OpenCV's many picture and video analysis features have earned it a reputation as a powerful tool in computer vision and image processing. ImageMagick offers a powerful set of tools and libraries for working with bitmap pictures, whether in creation, editing, or conversion. While the main use of GIMP is to edit images, it also has scripts and plugins that may be utilized for a variety of image processing jobs [25].

3.2. Implementation of proposed Architecture. The entire process of implementing image pixel memory and matrix generation is shown in the Fig 3.2. First a colour source image is taken and then it is converted to grey-scale image, which is the source image in this process. The source images with different resolutions are taken, and the pixels are extracted as hexadecimal values by using MATLAB software, which are given as an input text file to Xilinx ISE Software.

The first thing that happens while processing images is reading them from memory. Then, a typical method for drawing attention to borders and transitions, Sobel edge detection, is employed to locate picture edges. Data extraction from random access memory (RAM) is the following step, after which pertinent picture information is retrieved. If you're working with picture pixel data in a line buffer, you may need to convert the values into hexadecimal format. The extraction of a 3x3 matrix from the picture data is an essential step because it is often utilised in many image processing methods. As a last step, a text file is generated from the processed picture data, containing the outcomes of all these actions. To keep everything running smoothly and at the correct time, it may be necessary to use a gated clock signal with clock enable outputs at various points in this process.

In this software, image pixel memory and matrix generation for the Sobel edge detection are developed using Verilog HDL as shown in Algorithm 1. The image pixels memory is generated according to the rows

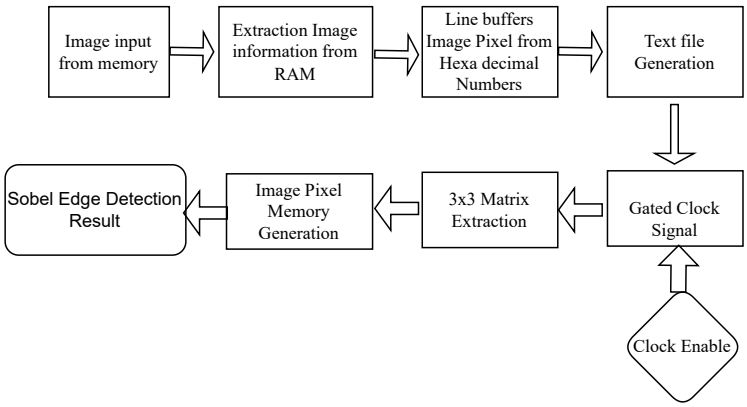

Fig. 3.2: Implementation process of image pixel memory and matrix generation for Sobel edge detection

and columns of image resolution. The 3x3 pixel matrix is formed using accessing the address of corresponding three rows and three columns. Clock gating technique is applied in the processing of image pixel memory and pixel matrix generation for dissipating less power. The clock gating signal is generated by performing AND operation with the clock and enabling signals.

Algorithm 1 Image Pixel Memory and Matrix Generation for Sobel Edge Detection

Require: Image width W, height H, and pixel data file in hexadecimal format
Ensure: 3x3 pixel matrices for Sobel edge detection
Initialize Parameters:
Set image dimensions $W \times H$
Allocate RAM memory sufficient to store the entire image based on its resolution
Load Pixel Data:
Read the pixels from the hexadecimal text file into RAM memory
Generate Gated Clock Signal:
Perform an AND operation between the clock and enable signals to obtain a gated clock signal
Process Image for Sobel Edge Detection:
while gated clock signal is positive edge **do**
    Step 1: Copy to Line Buffers
    Read pixel values from three consecutive rows in RAM into three line buffers
    Step 2: Extract 8-bit Pixel Matrix
    Extract 8-bit pixel values from the line buffers to form a 3x3 pixel matrix
    Ensure the correct 3x3 neighborhood by shifting the contents of the line buffers
    Step 3: Output Pixel Matrix
    Output the 3x3 pixel matrix for Sobel edge detection
    Step 4: Iterate Through Rows
    Move the 3x3 window across the entire image by iterating through all the rows
    Ensure each pixel is processed for edge detection
end while

Sobel edge detection's architectural design is shown in the block diagram that was supplied. One of the main goals is to use a 3x3 matrix and a threshold to find out whether each pixel is an edge by calculating its gradient magnitude.

Beginning with the 3x3 pixel matrix, every pixel, denoted as P1 through P9, has a width of 8 bits. The

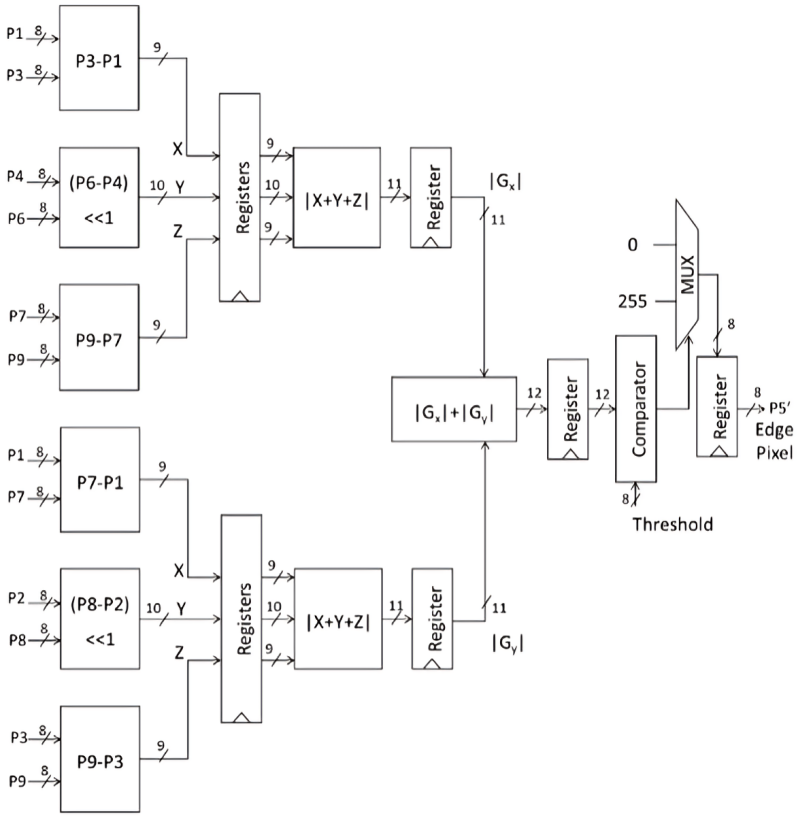

Fig. 3.3: Design Architecture for Sobel Edge Detection Module

horizontal and vertical gradients are computed by performing subtraction operations between certain pairs of pixels. To illustrate, the P3-P1 block takes the value of pixel P3 and subtracts it from pixel P1, producing an output with 9 bits. Other subtractor blocks, such as P6-P4, P9-P7, P7-P1, P8-P2, and P9-P3, carry out similar operations.

With the addition of the shift left operation ($\ll 1$), specific pixel values are effectively multiplied by 2 to highlight their role in the gradient computation, in addition to the subtraction operations. Next, adders labelled |X+Y+Z| are used to total the outcomes of these operations. After each round of subtractions and shifts, these adders take the absolute value of the total, checking that the gradient values are positive.

Combining the outputs of the required subtractor and shift blocks allows one to determine the horizontal gradient (Gx) and the vertical gradient (Gy). The absolute values of these gradients are briefly stored in registers in order to synchronise the processing phases. After that, we get a 12-bit value—the total gradient magnitude – by adding the absolute values of the horizontal and vertical gradients, |Gx| and |Gy|, respectively.

A comparator block is used to compare the magnitude of this gradient to a preset threshold. A pixel's inclusion or exclusion from an edge is determined by the threshold, an 8-bit integer whether the gradient magnitude is greater than the threshold determines the comparator's signal output.

The output of the comparator is then used by a multiplexer (MUX) to determine the final value of the edge pixels. The MUX will output 255 to indicate the existence of an edge if the gradient magnitude is greater than the threshold. Without an edge, the MUX will return a value of 0 if the gradient magnitude is less than the threshold. The value of the edge-detected pixel is represented by the final output, which is labeled as P5'.

```
a4a5a6a6a5a5a4a5a5a5a6a7a7a7a7a7a7a8a9a9aaababa9aa
a4a5a6a6a5a5a4a5a5a6a6a7a7a7a7a7a7a8a8a9a9aaababa8a9
a4a5a6a6a5a5a5a5a6a6a7a8a8a8a8a8a8a8a8a9aaaaababa9aa
a4a5a6a6a6a5a5a5a7a7a8a8a9a9a9a9a8a8a9a9aaaaababaaab
a5a6a7a7a6a6a5a6a7a7a7a8a8a9a9a9a9a8a9a9a9aaaaababaaaa
a5a6a7a7a7a6a6a6a6a6a7a7a8a8a8a8a8a9a9a9aaaaababababab
a6a7a8a8a8a7a7a7a7a7a8a8a9a9a9a9a9a9a9aaaaaaababacacac
a6a7a8a9a8a7a7a7a8a9a9aaaaaaaaaaa9aaaaaaaaababababab
a2a5a8a9a8a7a7a8aaa9a9aaabaaabacaaaaaaaaaaaabacadafac
a3a5a8a9a8a7a7a8a8a8a8aaacabacadaaaaababababacadaeac
a5a7a9a9a9a8a8a8a8a8a9abacacadaea9aaababababacadadac
a7a8a9aaa9a8a8a8aaa9aaacadacabaca8aaabacababababacacac
a7a8a8a8a8a7a7a7aaa9aaacacababaca9abacadacabacadacac
```

(a) (b)

Fig. 4.1: a) Source image of resolution 128x128 b) Image pixels as hexadecimal values in text file

Table 4.1: Error Analysis on various test input images

Input Image	Description	Total Pixels	Error (in Pixels)	Error (%)
Image001	zero	1024	0	0
Image002	one	1024	4	0.16
Image003	two	1024	7	0.45
Image004	three	1024	3	0.21
Image005	four	1024	8	0.72
Image006	five	1024	6	0.58
Image007	six	1024	7	0.56
Image008	seven	1024	5	0.51
Image009	eight	1024	6	0.59
Image010	nine	1024	3	0.79

Finally, this design effectively executes Sobel edge detection on the input pixel data, producing a binary signal of edge existence according to the calculated gradients and the given threshold.

4. Results and Discussion. In this section, a brief discussion of Sobel edge detection for image pixel memory design of testing outcomes has been discussed. The source image of resolution 128x128, shown in Fig. 4.1a, is taken for generating image pixel memory and a 3x3 pixel matrix. This source image is given as input to MATLAB software. Then, image pixels are generated in a text file as hexadecimal values, as shown in Fig. 4.1b. The text file, which consists of image pixels in hexadecimal values, is given as input to Xilinx ISE software. Then, the pixels matrix generates image pixel memory using Verilog HDL. The implemented image pixel memory is made up of BRAMs and registers. The simulation process for the image pixel memory and pixels 'P0' to 'P8', is shown in Fig. 4.1.

Images 001 through 010 make up the test inputs in the table 4.1, which displays an error analysis for each. The analysis checks for pixel accuracy problems in each picture, which has a total of 1024 pixels. Image001, standing for the number zero, has a 0% mistake rate according to the statistics. On the other hand, Image002, which stands for the number one, has four mistakes—a rate of 0.16%. There are seven mistakes in Image003, which represents the number two, resulting in an error rate of 0.45%. Image004 (the third digit) has an error rate of 0.21%, with a total of 3 errors. Image005, which stands for the number four, has 8 mistakes, giving it an error rate of 0.72%. With a total of six mistakes, the error rate for Image006 (the fifth digit) is 0.58%. The error rate for Image007 (digit six) is 0.56%, with seven mistakes. The 0.51% error rate results from 5 mistakes in Image008, which represents the number 7. The error rate for Image009 (digit eight) is 0.59%, with six mistakes. At 0.79%, Image010 (the ninth picture) has the greatest error rate of all the photos due to its three mistakes. By measuring and comparing the error rates across many test pictures, this research sheds light

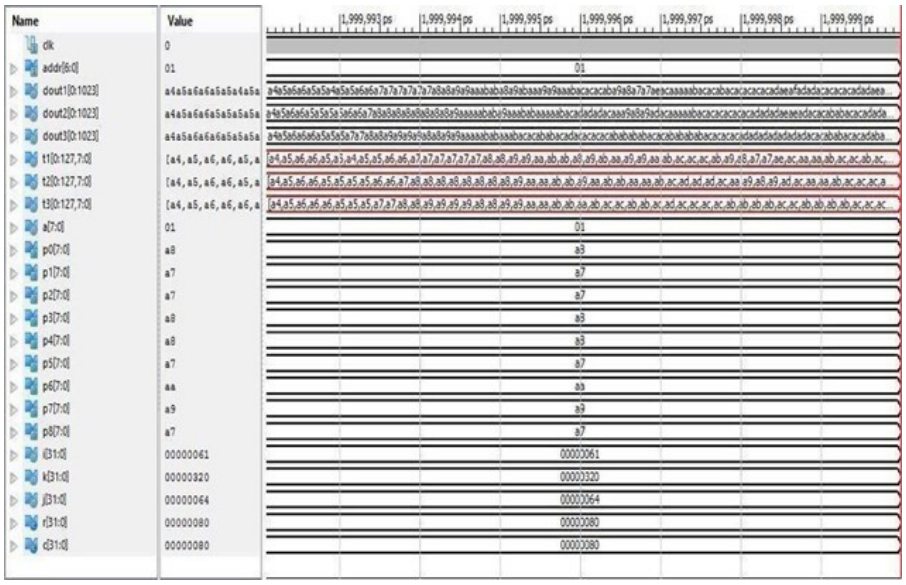

Fig. 4.2: Simulation waveforms for the image pixel memory and output pixels P0 to P8

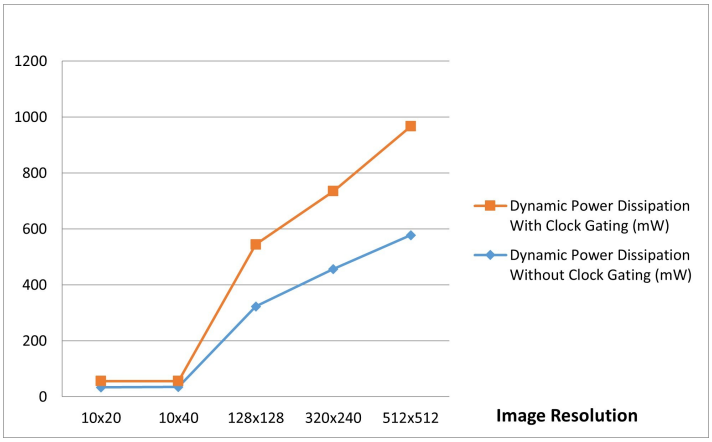

Fig. 4.3: Power dissipation of image pixel memory and 3x3 pixel matrix generation for different image resolutions

on the precision of image processing.

The parameters 'r' and 'c represents the rows and columns of the image pixel memory and 'addr' is the address register which is used to access the rows. The registers 'dout1', 'dout2', 'dout3' represents the combined pixels data stored in three consecutive rows respectively and 't1', 't2', 't3' are the registers that represent the individual pixels values in 8-bit format

The synthesis and power analysis process were performed after simulation. The power analysis of image pixel memory for 3x3 pixel matrix generation, is performed using XPower Analyzer of Xilinx ISE for different source image resolutions with and without clock gating, whose power dissipation values are shown in Fig. 4.2. With reference to Fig. 4.3 and by comparing source images with various resolutions with and without clock gating, the approximate dynamic power dissipation reduces by 31% to 40%. As the resolution of the source image increases the power dissipation reduction decreases. For example, the source image with resolution 10x40 has power reduction of 40%, whereas for source image with resolution 128x128 has power reduction of 31%.

Table 4.2: Comparison of Enhanced Edge Detection Models

Enhanced Edge Detection Model	Power Consumption	Latency Improvement	Performance
SoC Sobel Edge [19]	0.2	0.2	0.15
Multi-directional Sobel Operator Kernel [17]	0.22	0.25	0.13
ML Model-based Design Method [15]	0.25	0.26	0.2
Proposed Method	0.4	0.3	0.26

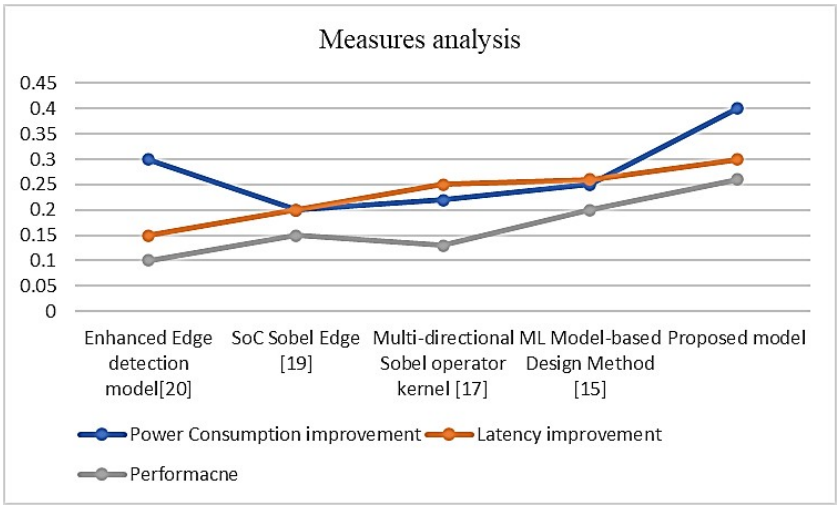

Fig. 4.4: Performance analysis

Table 4.2 and Figure 4.4 compare enhanced edge detection models based on three criteria: power consumption, latency improvement, and performance. The models compared include the SoC Sobel Edge method, the Multi-directional Sobel Operator Kernel, the ML Model-based Design Method, and the Proposed Method.

The SoC Sobel Edge model consumes 0.2 units of power, improves latency by 0.2 units, and has a performance rating of 0.15. The Multi-directional Sobel Operator Kernel consumes slightly more power at 0.22 units, achieves a latency improvement of 0.25 units, and has a performance rating of 0.13. The ML Model-based Design Method shows higher power consumption at 0.25 units, with a latency improvement of 0.26 units and a performance rating of 0.2. The Proposed Method has the highest power consumption at 0.4 units, but it also offers the most significant latency improvement of 0.3 units and the best performance rating of 0.26.

Edges in an image are critical features that mark regions where there is a sudden change in pixel intensity. In a grayscale image, which varies from black (representing the lowest intensity) to white (the highest intensity), edges appear where the intensity changes abruptly. To detect these changes, we analyze the slope of the intensity curve, which is effectively the first derivative of the intensity function.

The process of detecting edges involves a mathematical operation known as convolution. This operation applies a small matrix, or kernel, to the image to estimate derivatives. One of the most common edge detection techniques is the Sobel Operator, which utilizes two distinct kernels—one for detecting vertical edges and another for horizontal edges.

In vertical edge detection, the Sobel operator uses a 3x3 kernel to compute the gradient in the x-direction, which highlights horizontal edges by convolving this kernel with the image. Similarly, for detecting horizontal edges, another 3x3 kernel is used to calculate the gradient in the y-direction, which reveals vertical edges. By combining the gradients from both directions, we can determine the overall gradient magnitude at each pixel. This combined gradient helps identify edges where the magnitude exceeds a specified threshold.

Sobel Edge Detection is widely applied in various image processing tasks. It is instrumental in object

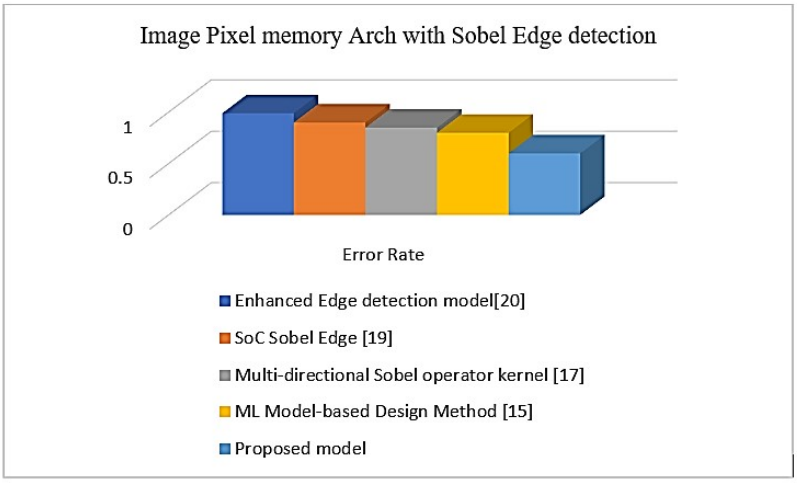

Fig. 4.5: Performance analysis

recognition, where it helps to identify the boundaries of objects within an image. It is also used in image segmentation to separate regions of interest and in pattern recognition to detect specific shapes or features. These practical applications highlight the importance of edge detection in analyzing and interpreting visual information shown in figure 4.5.

5. Conclusion. In this paper, the development and analysis of image pixel memory are thoroughly examined through the use of Block RAMs (BRAMs) and registers. The approach involves generating a 3x3 pixel matrix for a Sobel edge detector, utilizing MATLAB and Xilinx ISE software. The process encompasses various stages, including simulation, synthesis, and power analysis of the image pixel memory system. The analysis highlights a significant aspect of the study—power dissipation. The power consumption of the developed image pixel memory is evaluated across different source image resolutions, including 10x40, 10x20, 128x128, 320x240, and 512x512. The results indicate that the use of clock gating techniques effectively reduces power dissipation by approximately 30% to 40%. Clock gating, which selectively disables the clock signal to inactive parts of the circuit, proves to be a crucial method for enhancing the energy efficiency of the system. Looking forward, there is potential for further improvement in the performance of image pixel memory. Future work can focus on optimizing the system by reducing the number of registers and minimizing the pixel access time. Such improvements would not only enhance the efficiency but also contribute to more responsive and high-performance image processing applications. The ongoing development and optimization of image pixel memory systems are essential for advancing edge detection technologies and other image processing techniques.

REFERENCES

[1] Harald Devos, Dirk Stroobandt, *Optimizing the FPGA Memory Design for a Sobel Edge Detector. International Conference on Engineering of Reconfigurable Systems & Algorithms (ERSA). 2009.*

[2] Deepayam Bhowmik, Robert Stewart Greg Micharlson, Andraw Wallace, *Optimized Memory Allocation and power Minimization for FPGA based Image processing. Journal of Imaging. 2019, vol. 5, iss. 1, DOI: 10.3390/jimaging5010007*

[3] Peng Zhang, Xu Cheng, Jason Cong, *An Integrated and Automated Memory Optimization Flow for FPGA Behavioural Synthesis. In: 17th Asia and South Pacific Design Automation Conference. Sydney, NSW, Australia: IEEE, 2012, pp. 257-262. DOI: 10.1109/ASPDAC.2012.6164955*

[4] M. Pedram, Xunwei Wu, *Clock-gating and its application to low power design of sequential circuits. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications. 2000, vol. 47, iss. 3, pp. 415-420. DOI: 10.1109/81.841927.*

[5] J Raivainen, A Mammela, *Clock Gating in FPGA's: A Novel and Comparative Evaluation. In: Ninth Euromicro Conference on Digital System Design, Cavtat, Croatia: IEEE, 2006, pp. 584-590, DOI: 10.1109/DSD.2006.32.*

[6] David Flynn, Robert Aitken, Alan Gibbons, Kajian Shi, *Low Power Methodology Manual For System-on-Chip Design. Springer, 2007. ISBN 978-0-387-71818-7.*
[7] Mark Biegel, *Power Reduction through RTL Clock Gating. SNUG, San Jose, 2000, pp.1-11.*
[8] R. W. Brodersen, *Minimizing power consumption in digital CMOS circuits In: Proceedings of the IEEE, vol. 83, no. 4, pp. 498-523, 1995, DOI: 10.1109/5.371964.*
[9] D Lakata, A Dehon, *Impact of Parallelism and Memory Architecture on FPGA Communication Energy. ACM Transactions on Reconfigurable Technology and Systems. 2016. vol. 9, iss. 4, pp. 1-23. DOI:10.1145/2857057*
[10] P. Siegel, G. D. Micheli, *Saving power by synthesizing gated clocks for sequential circuits. IEEE Design & Test.1994. pp 32-41.*
[11] R. Joshi, M. A. Zaman, and S. Katkoori, *Novel bit-sliced near-memory computing based VLSI architecture for fast Sobel edge detection in IoT edge devices. In 2020 IEEE International Symposium on Smart Electronic Systems (iSES)(Formerly iNiS) (pp. 291-296). IEEE, December 2020.*
[12] Z. E. M. Osman, F. A. Hussin, and N. B. Z. Ali, *Hardware implementation of an optimized processor architecture for SOBEL image edge detection operator. In 2010 International Conference on Intelligent and Advanced Systems (pp. 1-4). IEEE, June 2010.*
[13] S. Singh, S. Saurav, R. Saini, A. K. Saini, C. Shekhar, and A. Vohra, *Comprehensive review and comparative analysis of hardware architectures for Sobel edge detector. International Scholarly Research Notices, 2014(1), 857912, 2014.*
[14] S. Kumar, and P. Pandey, *FPGA implementation of image segmentation by using edge detection based on Sobel edge operator. International Journal of Research in Engineering and Technology, 2(10), 198-203, 2013.*
[15] T. Saidani, R. Ghodhbani, M. Ben Ammar, M. Kouki, M. H. Algarni, Y. Said, ... and E. H. Abd-Elkawy, *Design and Implementation of a Real-Time Image Processing System Based on Sobel Edge Detection using Model-based Design Methods. International Journal of Advanced Computer Science & Applications, 15(3), 2024.*
[16] D. Pudi, R. Ryansh, V. Goudu, S. Boppu, and A. Hemani, *Implementation of Sobel Edge Detection on DRRA and DiMArch Architectures. In 2023 26th Euromicro Conference on Digital System Design (DSD) (pp. 16-23). IEEE, September 2023.*
[17] Q. Chang, X. Li, Y. Li, and J. Miyazaki, *Multi-directional Sobel operator kernel on GPUs. Journal of Parallel and Distributed Computing, 177, 160-170, 2023.*
[18] V. Yamini, S. A. Hussain, G. Chandra Sekhar, P. Avinash Kumar, P. Lehitha, B. Sree Venkata Teja, ... and P. K. Sanki, *An SoC System for Real-Time Edge Detection. Journal of Electronic Materials, 1-8, 2024.*
[19] A. S. Khalil, M. Shalaby, and E. Hegazi, *An Enhanced System on Chip-Based Sobel Edge Detector. In 2023 International Telecommunications Conference (ITC-Egypt) (pp. 179-183). IEEE, July 2023.*
[20] M. Orthy, S. M. R. Islam, F. Rashid, and M. A. Hasan, *Implementation of Image Enhancement and Edge Detection Algorithm on Diabetic Retinopathy (DR) Image Using FPGA. IET Circuits, Devices & Systems, 2023(1), 8820773, 2023.*
[21] K. M. Rao, P. S. Kumar, T. V. Reddy, D. Nilima, K. Saikumar, and A. M. Khlaif, *"Ultra Low Power High Speed DFT Implementation For ASIC SoC,"* in 2024 IEEE 9th International Conference for Convergence in Technology (I2CT), IEEE, April 2024, pp. 1-6.
[22] R. Revathi, R. Vatambeti, K. Saikumar, M. A. Alkhafaji, U. R. Khairy, and S. Noori, *"An advanced online mobile charge calculation using artificial intelligence,"* in AIP Conference Proceedings, vol. 2845, no. 1, AIP Publishing, September 2023.
[23] K. Mannepalli, K. B. Raju, J. Sirisha, K. Saikumar, and K. S. Reddy, *"LOW complex OFDM channel design using underwater-acoustic-communication using machine learning techniques,"* in 2021 5th International Conference on Electronics, Communication and Aerospace Technology (ICECA), IEEE, December 2021, pp. 1505-1513.
[24] D. S. Kumar, C. S. Kumar, S. Ragamayi, P. S. Kumar, K. Saikumar, and S. H. Ahammad, *"A test architecture design for SoCs using atam method,"* International Journal of Electrical and Computer Engineering, vol. 10, no. 1, pp. 719, 2020.
[25] N. Soumya, K. S. Kumar, K. R. Rao, S. Rooban, P. S. Kumar, and G. N. S. Kumar, *"4-bit multiplier design using cmos gates in electric VLSI,"* International Journal of Recent Technology and Engineering, vol. 8, no. 2, pp. 1172-1177, 2019.

Edited by: Dhilip Kumar V
Special issue on: Unleashing the power of Edge AI for Scalable Image and Video Processing
Received: May 31, 2024
Accepted: Aug 24, 2024