Journal of VLSI circuits and systems
RESEARCH ARTICLE
Journal of VLSI Circuits and Systems, ISSN: 2582-1458 Vol. 7, No. 2, 2025 (pp. 99-110)
WWW.VLSIJOURNAL.COM

FPGA-Based Implementation and Comparative Analysis of Robinson and Kirsch Compass Edge Detectors

Shatha AbuShanab*
Computer Systems Engineering Department, Palestine Technical University - Kadoorie, Palestine.

KEYWORDS:
VHDL,
Digital design,
Edge detector,
Robinson edge detector,
Kirsch edge detector

ARTICLE HISTORY:
Received: 30.09.2025
Revised: 28.10.2025
Accepted: 05.11.2025

DOI:
https://doi.org/10.31838/JVCS/07.02.12

Abstract

This approach aims to develop and implement the Robinson and Kirsch compass edge detector on various complementary metal-oxide-semiconductor (CMOS) technologies. A reconfigurable very-large-scale integration (VLSI) architecture, field-programmable gate arrays (FPGAs), is used as the design platform. In digital image processing, edge detection is a technique used to find edge pixels in an image; these edges form the boundaries of objects. The edge detection system is described using hardware description language (HDL) at an abstract level, and the digital circuits for the Robinson and Kirsch compass edge detector are implemented at a lower (hardware) level. At this hardware level, physical devices are used to obtain real results. The input for the digital system is an RGB color image, which is converted to grayscale, followed by the application of edge detectors and a threshold. ModelSim Intel is used to simulate and verify the functionality of the digital system, while Intel Quartus Prime is used to implement the edge detectors on different CMOS technologies, including Cyclone IV, Cyclone V, and Cyclone LP 10 FPGA devices. Additionally, a comparative analysis based on the CMOS technologies used is conducted, considering accuracy and edge thickness, digital design complexity, resource usage, and power dissipation. The results indicate that the Kirsch edge detector is suitable for accurate applications and can detect weak edges. The Kirsch edge detector, by contrast, dissipates more power because it is more costly and computationally complex. Furthermore, the CMOS technology used affects the overall power dissipation and the digital design requirements.

Authors' e-mail ID: shatha.abushanab@ptuk.edu.ps

Authors' ORCID ID: 0000-0002-8550-5301

How to cite this article: Shatha AbuShanab, FPGA-Based Implementation and Comparative Analysis of Robinson and Kirsch Compass Edge Detectors, Journal of VLSI Circuits and System, Vol. 7, No. 2, 2025 (pp. 99-110).

Introduction

By detecting changes in color or intensity between adjacent pixels, edge detection algorithms are often used in digital image processing to locate edge pixels in an image. Different regions of an image are separated by boundaries made by edge pixels. The edges can be used to extract features, such as an object's size or shape. Throughout history, several edge detection techniques have been developed that are useful in various fields and image processing applications. The Roberts operator, which uses 2 × 2 kernels, is one of the earliest edge detection techniques[1] whose computation and design are simple, but it is sensitive to noise.[2] Then, the Sobel[3] and Prewitt[4] edge detectors were developed around the same time, which use 3 × 3 kernels, and are less sensitive to noise compared to the Robert edge detector. Robinson[5] and Kirsch[6] edge detectors are a set of eight 3 × 3 compass masks to find edges in a specific compass direction.

Numerous research studies have been carried out with image processing applications using computer programs like MATLAB and Python, and the findings were analyzed and compared. Shah et al.[7] used MATLAB and Python programming to compare and analyze several edge detection techniques. Sobel, Prewitt, Robert, Canny, and Log were the edge detection methods that have been studied. According to the experimental findings,

Canny edge detection performed better than other techniques.[7] Mayangky et al. investigated and evaluated several edge detection techniques. These techniques were Prewitt, Robinson, Krisch, and Roberts. The results demonstrated that the Krisch edge detection technique is more effective at detecting edges.[8] Karanwal described and implemented various edge detection techniques with varying scale values. The Sobel operator, Prewitt operator, Robert operator, Kirsch operator, and Robinson operator were edge detection techniques. MATLAB was used for the implementation, and a 200 × 250 image size was used as the input. The study compared the different operators and examined different thresholding and multiscale processing.[9]

Very-large-scale integration (VLSI) design involves integrating millions or billions of transistors into a single chip to create integrated circuits (ICs). Additionally, the development of scaling complementary metal-oxide-semiconductor (CMOS) technology has permitted to implement of a complete digital system on a single chip. Digital systems ranging from simple to huge and complicated designs on a chip can be implemented.[10-12] Recently, a lot of researchers use a reconfigurable VLSI system, field programmable gate array (FPGA) devices for various image processing applications.[13-16] Al-Mukhtar and Hasso developed and implemented the Robinson edge detector on a filed programmable gate array (FPGA). The technique was implemented on an FPGA Xilinx XC3S500E Spartan-3E using MATLAB code and the VHDL language. Peak signal-to-noise ratio (PSNR) and root mean square error (RMSE) were calculated and compared between the MATLAB and VHDL resultant images. The correlation between the two images indicates that they are almost the same.[17] Humaidi et al. developed and implemented a lane detection system using FPGAs. A video camera took pictures, which the system converted to grayscale. Image processing techniques like filtering, edge detection, and thresholding were used on the grayscale images. Sobel, Robinson, Laplacian, and LoG were the edge detection methods that were applied. The Sobel edge detector performed well, according to the results. Furthermore, the number and size of the masks determined how many logic elements the system used; the Laplacian edge detector used fewer than the Sobel, but the Sobel had the strongest performance.[18]

Therefore, this approach aims to investigate, develop, and implement the Robinson and Kirsch compass edge detectors on various CMOS technologies, such as FPGA devices. Additionally, a comparison of the Robinson and Kirsch compass edge detectors is carried out using the same RGB image as the input to the implemented detector, and the results depend on the CMOS technology used. Accuracy and edge thinness, power dissipation, resources utilized at the hardware level, and digital design complexities are all compared. The proposed digital design of the compass edge detector is fully implemented on an FPGA chip, which features a parallel processing technique essential for image processing. Furthermore, provide more details at the hardware level, including an accurate power dissipation estimate. The article is organized into five sections. This section introduces the research that has developed and implemented the Robinson and Kirsch compass edge detection system on various FPGA devices. Robinson and Kirsch Compass Edge Detection section introduces and defines the Robinson and Kirsch compass masks edge detection. In Robinson and Kirsch Edge Compass Edge Detector Implementation section, the implementation of both the compass edge detectors and the methodology used is presented. The second-to-last section discusses and analyzes the results obtained. The conclusion is provided in the last section.

Robinson and Kirsch Compass Edge Detection

Robinson and Kirsch edge operators are similar and closely related; they both belong to the same family and are used as compass or directional edge detectors. The Robinson and Kirsch compass edge detection algorithms use eight 3 × 3 compass masks. Each compass mask is designed to detect edge pixels in a specific direction. The compass masks are generated by rotating a single mask to create eight different directional masks: east, northeast, north, northwest, west, southwest, south, and southeast. Robinson compass masks have been used in these eight directions as follows[5]:

$$R0(\text{East}) = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad \text{R1(Northeast)} = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}$$

$$\text{R2(North)} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, \quad \text{R3 (Northwest)} = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix}$$

$$\text{R4(West)} = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}, \quad \text{R5(Southwest)} = \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix}$$

$$\text{R6(South)} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, \quad \text{R7(Southeast)} = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

The Robinson compass masks are simple to implement because they only use coefficients 0, 1, and 2, along with their negations around their directional axis, where the zero axis shows the mask's orientation. It is sufficient to compute the results for the four base masks in the Robinson edge detector, since the remaining four masks are simply the opposites of the first four. The eight Kirsch compass masks are listed as follows[5]:

$$K0\,(\text{East}) = \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix},\ K1(\text{Northeast}) = \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}$$

$$K2\,(\text{North}) = \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix},\ K3(\text{Northwest}) = \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$$

$$K4\,(\text{West}) = \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix},\ K5(\text{Southwest}) = \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix}$$

$$K6\,(\text{South}) = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix},\ K7(\text{Southeast}) = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}$$

For any previous compass mask, the sum of all coefficients equals zero. This method involves applying a mask's convolution algorithm on a 3 × 3 subwindow of an image to detect pixel edges in a digital image. Neighboring pixels with different values can detect an edge pixel. Figure 1 displays a digital image, with each square representing a pixel and its coordinates. The convolution formula is applied to the input image (on the left) by multiplying the 3 × 3 subwindow with the 3 × 3 compass mask. The output (on the right) displays changes in pixel intensity within the neighborhood, as seen in Figure 1.

The edge amplitude of each Robinson or Kirsch compass mask is determined. Applying Eq. 1 to determine a maximum magnitude among the eight directional mask results in the amplitude of the combined eight masks.

$$C_t(i,j) = \max_{n=1,2,\ldots 8}\left(\left|C_{(i,j)}\right|\right) \quad (1)$$

where n is the number of the edge direction, $C_{(i,j)}$ is the result of the convolution at the pixel position i, and j coordinate, $C_t(i, j)$ is the amplitude of the combined eight compass masks. Afterward, edge pixels are identified by comparing the absolute value of the result to the threshold. When an edge pixel is assigned a value of zero, it indicates that the image is in a high-frequency region; thus, all low-frequency components will be removed. Edge pixels are assigned a value of 1.

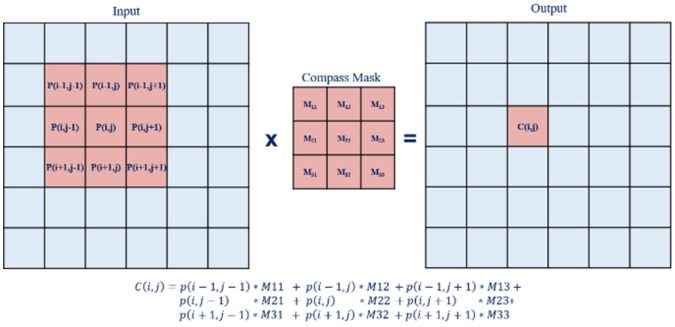


Fig. 1: The convolution formula for a 3 × 3 image sub-window using a compass mask

Robinson and Kirsch Edge Compass Edge Detector Implementation

The main objective of this research is to develop and implement the Robinsonand and Kirsch compass edge detector system across various CMOS technologies. Subsequently, the accuracy, digital design complexity, resource utilization, and power dissipation for each compass mask, based on the CMOS technology used, will be investigated, compared, and evaluated. A two-dimensional (2D) RGB image with 24-bit color depth, for each R, G, and B component is 8-bit depth, will be processed as input into the digital system, resulting in a black-and-white image with black edge pixels as output.

The Robinson and Kirsch compass edge detector systems are described in hardware description language (VHDL). Therefore, the research methodology includes two sequential phases, designed to successfully complete the design process and verify that the functionality of each edge detector meets the requirements. Figure 2 illustrates the phases of the methodology followed:

- The ModelSim Intel software is used to simulate the digital design described using VHDL.

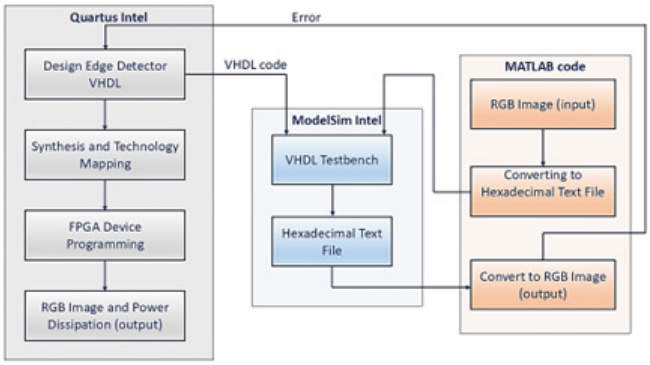


Fig. 2: Methodology used in the study

- The Intel Quartus Prime software is used to synthesize the VHDL digital design, which is then implemented on various CMOS technologies as various FPGA devices.

One pixel of the 2D image will be read during each clock cycle as part of the hardware-level design process. The Red (R), Blue (B), and Green (G) components of each pixel are 8-bit values ranging from 0 to 255. The edge detection algorithms for each directional compass mask operate separately, and then the eight directional masks are combined as follows:

- The RGB input image is converted into a grayscale image. Compared to an RGB image, converting to grayscale helps save processing time and memory. Equation 2 is used for this conversion[19]:

$$\text{Grayscale} = 0.299R + 0.587G + 0.114\,\text{B} \quad (2)$$

- For Robinson and Kirsch compass edge detection, as described above, the convolution formula for a 3 × 3 sub window of an image and a compass mask is computed for each directional compass mask.
- Edge pixels can be found; the amplitude of the convolution is compared to the given threshold. A value greater than the threshold means that an edge pixel is present, which is assigned a zero value and appears black. If it is below the threshold, the pixel is considered nonedge and is assigned a value of one, appearing white.
- The amplitude of the eight directional masks is combined to find edge pixels, which is calculated as follows:
  - The maximum amplitude for the eight directional compass masks is determined.
  - As mentioned earlier, the threshold is compared to the maximum amplitude to determine whether the pixel is an edge.

ModelSim-Intel Software

The ModelSim-Intel software is used to simulate and verify the digital design before implementing it on an FPGA device. A hardware description language (HDL) called VHDL is used to describe the edge detector. A 3 × 3 subwindow of adjacent pixels from the previous and next readings is required by the approach to perform the convolution with 3 × 3 compass masks. Every clock cycle saves the pixel being read in RAM. All pixel values in a row in an image are stored in one RAM when the last pixel of that row is read, and each subsequent row needing storage is saved in additional RAM. When the system reads the current row, it uses the two previously stored rows in RAM to apply the edge detector. During processing, RAMs replace the oldest pixels with new ones, making them available for the next row. This method effectively reduces the amount of RAM required.

The digital design that has been implemented processes a 2D RGB image as input. M × N, where *M* is the number of rows and *N* is the number of columns, is the matrix representation of the RGB image. In hardware implementation, the system reads the R, G, and B values corresponding to an input pixel, or one pixel each clock cycle. MATLAB is used to convert the image into a hexadecimal text file. This hexadecimal number in the text file represents the R, G, and B values for each pixel. The hexadecimal text file is used as input to the testbench for the ModelSim-Intel simulation. Every pixel in the digital design has a value between 0 and 255 when each component is converted to 8 bits. To produce the signals required for the simulation, a video input control signal is also needed. Horizontal blanking, vertical blanking, vertical synchronization, and horizontal synchronization pulses are the timing control signals of an input image used to read a frame.[20] The text file output by the ModelSim Intel simulation can be converted into an image for observing using MATLAB. The digital design has been successfully verified for functionality and can be effectively implemented into several CMOS technologies, as shown in Figure 3.

Synthesis and Implementation on Different CMOS Technologies

The ultimate goal of this approach is to investigate, compare, and evaluate the hardware-level results obtained across different CMOS technologies. The digital design is developed in accordance with system requirements that can be modified or improved at an abstract level and is independent of the CMOS technology used. Reliability is ensured by a physical system because it provides real data, such as factors affecting the design's power dissipation, which has become a more challenging issue in digital circuits, especially for

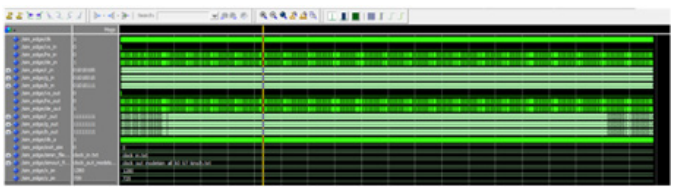


Fig. 3: Verification for the functionality of the Kirsch digital design

portable devices. Consequently, various FPGA devices are used to implement the fully digital design of the Robinson and Kirsch compass edge detectors. The remote lab, selected for this study to evaluate different FPGA technologies, is available as an open educational resource at https://www.h-brs.de/de/fpga-vision-lab. Figure 4 shows the prototype of the remote FPGA laboratory system, designed to support multiple FPGA technologies as a design platform and for image processing applications. The prototype of a remote FPGA system comprises a client that accesses the system via an Internet connection, and a remote server that receives and transmits data between the client and the laboratory equipment required to perform the laboratory experiments.[21-23]

The inputs for the remote lab are an RGB digital image and a binary file, which can be uploaded via an Internet browser. To ensure proper processing, the uploaded image must have a resolution of 720 x 1280 pixels. The input clock is 74.25 MHz, and video streaming is 720p; the throughput is 1 pixel/clock, and the latency is 8 cycles. Intel Quartus Prime software, which synthesizes VHDL-based digital designs, generates a binary file. The remote server will receive the inputs: an RGB image to generate a video stream for the FPGA board, and a binary file containing the edge-detection design to be implemented on the FPGA. As a result of the remote system, a processed RGB digital image and the power dissipation of the digital design are sent to the client via the Internet. Figure 5 illustrates the graphical user interface (GUI) of the remote FPGA laboratory system as displayed on the client screen, where the remote system's inputs are uploaded, and its outputs are delivered via the Internet.[21-23] The remote FPGA-vision-lab system offers different FPGA devices based on CMOS technology that are:

- Cyclone IV FPGA, EP4CE22E22C7, is based on 60 nm semiconductor technology.[24]
- Cyclone V FPGA, 5CEBA2F17C6, is a semiconductor based on 28 nm technology.[25]
- Cyclone 10 LP FPGA, 10CL120ZF484I8G, is a semiconductor with a 60 nm technology.[26]

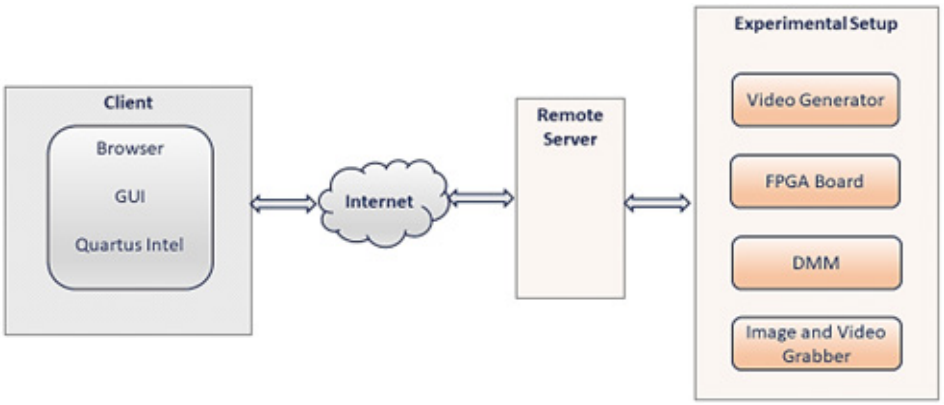


Fig. 4: The system diagram of the remote FPGA lab

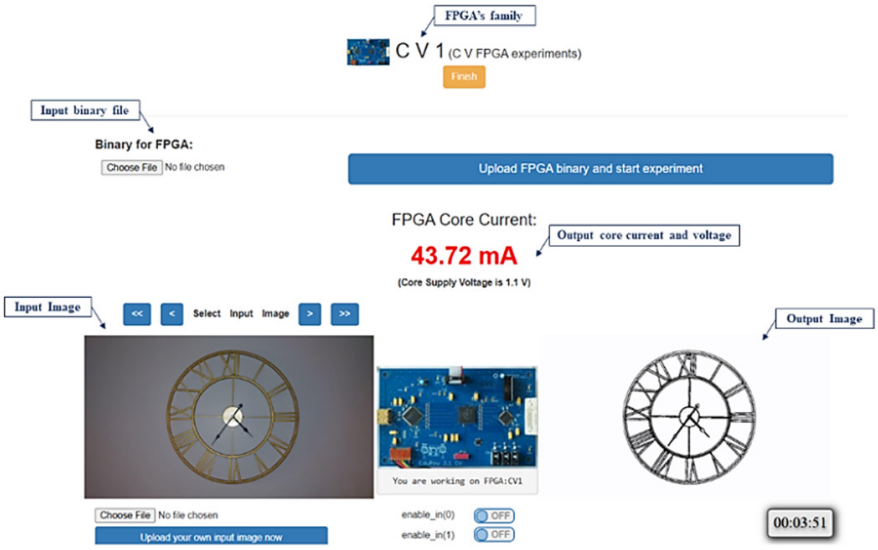


Fig. 5: GUI of the remote FPGA system

Tables 1, 2, and 3 list the FPGA resources utilized from the Intel Quartus Prime software for each of the Robinson and Kirsch compass masks, as well as for the edge detectors of all eight compass masks combined.

Figures 6 and 7 show the register-transfer level (RTL) of the combined eight compass masks for the Robinson and Kirsch edge detector digital design on the selected Cyclone 10 LP FPGA device.

Table 1: Resource Utilization for the Cyclone IV E (EP4CE22E22C7) Device.

Resources Used	Single Compass Mask		All Directional Masks		Four Basic Directional Masks	
	Robinson	Kirsch	Robinson	Kirsch	Robinson	Kirsch
Logic Element	≈671 (3%)	≈914 (4%)	1,997 (9%)	2,520 (11%)	1,207 (5%)	-
Register	≈476	≈527	741	758	617	-
Memory Bits	61,440 (10%)	61,440 (10%)	61,440 (10%)	61,440 (10%)	61,440 (10%)	-

Table 2: Resource Utilization for the Cyclone V FPGA (5CEBA2F17C6) Device.

Resources Used	Single Compass Mask		All Directional Masks		Four Basic Directional Masks	
	Robinson	Kirsch	Robinson	Kirsch	Robinson	Kirsch
Logic utilization (ALMs)	≈128 (1%)	≈194 (2%)	748 (8%)	1,046 (11%)	363 (4%)	-
Register	≈338	≈410	653	712	508	-
Memory Bits	61,440 (3%)	61,440 (3%)	61,440 (3%)	61,440 (3%)	61,440 (3%)	-

Table 3: Resource Utilization for the Cyclone 10 LP FPGA (10CL120ZF484I8G) Device.

Resources Used	Single Compass Mask		All Directional Masks		Four Basic Directional Masks	
	Robinson	Kirsch	Robinson	Kirsch	Robinson	Kirsch
Logic Element	≈552 (<1%)	781 (<1%)	1,873 (2%)	2,380 (2%)	1,090 (<1%)	-
Register	≈328	≈385	599	616	475	-
Memory Bits	61,440 (2%)	61,440 (2%)	61,440 (2%)	61,440 (2%)	61,440 (2%)	-

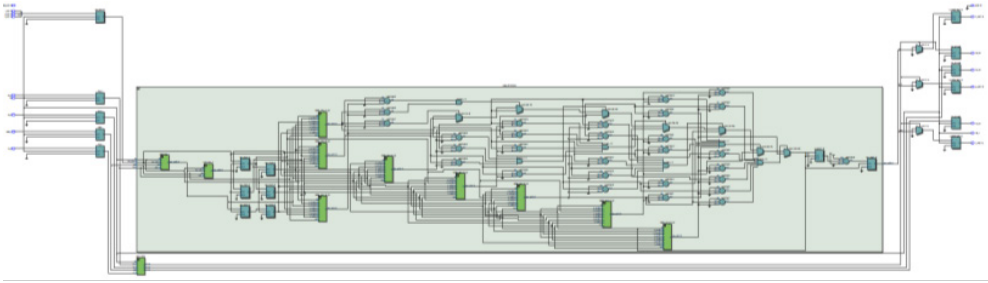


Fig. 6: RTL netlist of the Robinson compass edge detector digital design generated using Intel Quartus Prime

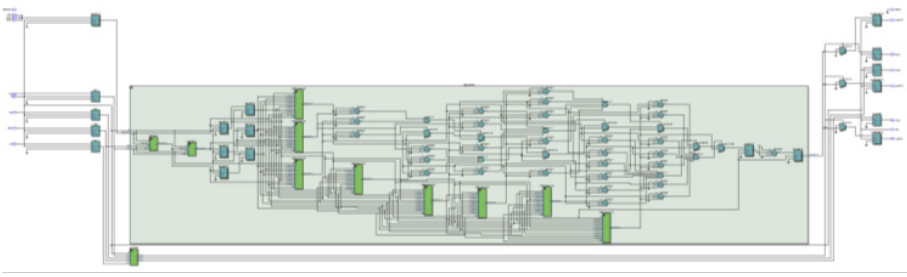


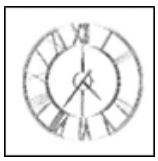
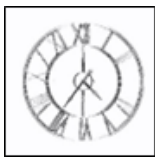
Fig. 7: RTL netlist of the Kirsch compass edge detector digital design generated using Intel Quartus Prime

Results and Discussion

When the two successive phases are completed, the results from ModelSim Intel and from implementing the compass edge detector on different FPGA devices in the remote lab are compared for the same input image.

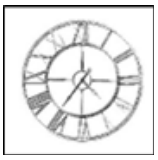
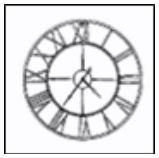
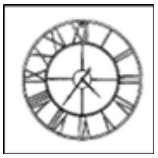
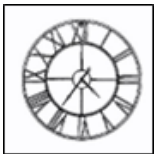
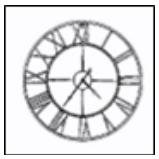
Table 4 presents the processed RGB images produced by the Robinson and Kirsch compass edge detector system, using ModelSim and an Intel FPGA board in a remote environment. The results of each mask compass indicate that the digital system has been successfully designed

Table 4: The input and output images processed by the Robinson and Kirsch edge detectors from ModelSim Intel simulations and the remote FPGA system.

The Input RGB Image				
				
Output Processed Image				
Directional Mask	**Robinson**		**Kirsch**	
	ModelSim	**FPGA System**	**ModelSim**	**FPGA System**
East				
Northeast				
North				
Northwest				
West				
Southwest				
South				

(*continues*)

Table 4: (Continued)

Southeast				
Eight masks				
Four Basic Masks				

and effectively detects edges across all directional masks.

The results obtained are discussed and compared in terms of:

- Accuracy, edge thickness, and threshold. Compared to the Robinson compass masks, Table 4 shows that the Kirsch compass masks produce stronger edge responses, resulting in thicker edges. Since Kirsch compass masks use coefficients of 0, +5, and −3, while Robinson employs simpler coefficients of 0, +1, +2, −1, and −2. This results in the Kirsch compass mask coefficients exhibiting greater convolution amplitude at higher intensities. Additionally, to achieve comparable results, the threshold is set to be 2–5 times higher than that used with the Robinson edge operator for the same edges.[27-29] The threshold of the Kirsch is three times higher than that of the Robinson, according to the results shown in Table 4. Additionally, PSNR and structural similarity index for measuring image quality between: the outputs from ModelSim Intel simulation and FPGA board, outputs from ModelSim Intel simulation without noise added, and ModelSim Intel simulation with noise salt and pepper, the outputs from FPGA board without added noise and FPGA board with noise salt and pepper are computed. The results indicate that the Kirsch method outperforms Robinson, as shown in Figures 8 and 9.
- Computational cost and complexity. The resources utilized by the Kirsch and Robinson compass edge detector design implemented on Cyclone IV, Cyclone V, and Cyclone LP 10 FPGA devices are compared with the relative maximum values in Tables 5, 6, and 7. The Kirsch edge detectors require more logic elements and registers than the Robinson detectors, as shown in the comparison (Table 5). Moreover, using base-four masks and their negatives in the Robinson compass edge detector design reduces the need to implement

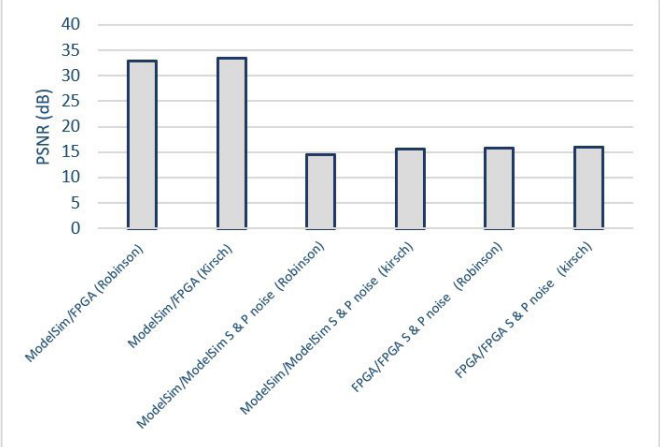


Fig. 8: Peak signal-to-noise ratio (PSNR)

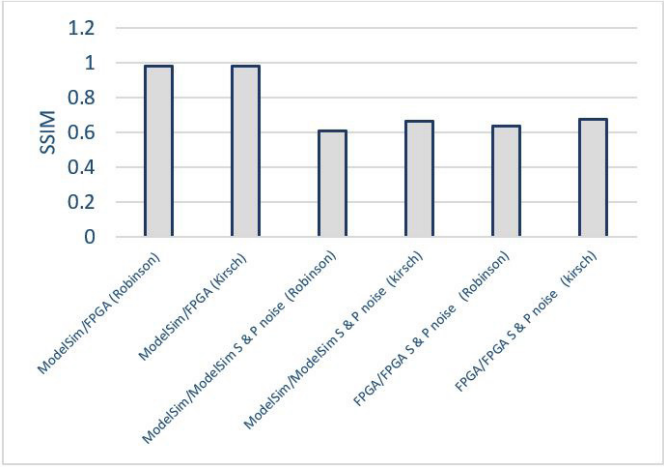


Fig. 9: Structural similarity (SSIM) index

all eight masks, thereby lowering computational complexity and cost. Depending on the FPGA technology, logic element usage decreases by 40–51% and register usage by 17–22%, whereas the Kirsch edge operator must perform all eight convolutions. Furthermore, when pixel reading and storage are complete by the specified time for a single mask convolution, the FPGA's parallel processing capability enables simultaneous computation of eight convolution masks, keeping memory bit usage constant for the single or combined computation of all eight compass masks.

- Power dissipation at the hardware level: In Table 8, as well as in Figures 10 and 11, the power dissipations of the Robinson and Kirsch compass edge detectors implemented on Cyclone IV, Cyclone V, and Cyclone 10 FPGA devices during operation are shown. The Kirsch's digital design requires more resources than Robinson edge detectors, resulting in higher power consumption. More logic elements, ALMs, and registers will increase power dissipation during switching in the digital circuits and require more routing capacitance between the resources. The same digital design is implemented across several CMOS technologies for the Kirsch or Robinson edge detector; comparing the Cyclone 10 FPGA with the Cyclone V and IV, the former consumes less power. Additionally, Cyclone IV dissipates less power than Cyclone V, despite using all eight compass masks for the Kirsch edge detector, a newer technology. Several factors influence this: Cyclone IV FPGA is built on 60 nm semiconductor technology, and older semiconductor technologies typically result in higher leakage currents and increased power

Table 5. Kirsch and Robinson Edge Detectors Implemented on the Cyclone IV E (EP4CE22E22C7) Device are Compared.

Item	Logic element (%)	Register (%)	Memory Bits (%)
Kirsch compared to Robinson for a single compass mask	27	10	0
Kirsch compared to Robinson for all eight compass masks	21	2	0
Robinson compared to Robinson for four basic and eight compass masks	40	17	0
Kirsch for all eight compasses compared to Robinson for four basic compasses masks	52	19	0

Table 6. Kirsch and Robinson Edge Detectors Implemented on the Cyclone V FPGA (5CEBA2F17C6) Device Are Compared.

Item	Logic utilization (ALMs) (%)	Register (%)	Memory Bits (%)
Kirsch compared to Robinson for a single compass mask	34	17	0
Kirsch compared to Robinson for all eight compass masks	28	8	0
Robinson compared to Robinson for four basic and eight compass masks	51	22	0
Kirsch for all eight compasses compared to Robinson for four basic compasses masks	65	28	0

Table 7. Kirsch and Robinson Edge Detectors Implemented on Cyclone 10 LP FPGA (10CL120ZF484I8G) Device Are Compared.

Item	Logic Element (%)	Register (%)	Memory Bits (%)
Kirsch compared to Robinson for a single compass mask	29	14	0
Kirsch compared to Robinson for all eight compass masks	21	3	0
Robinson compared to Robinson for four basic and eight compass masks	42	21	0
Kirsch for all eight compasses compared to Robinson for four basic compasses masks	54	23	0

Table 8. The power dissipation of the Robinson and Kirsch edge detectors implemented on FPGA devices.

Compass mask used	Power Dissipation (mW)					
	Cyclone IV		Cyclone V		Cyclone 10	
	Robinson	Kirsch	Robinson	Kirsch	Robinson	Kirsch
Single compass mask	≈30.54	≈33.32	35.44	38.51	16.9	17.9
All eight compass masks	44.18	53.78	48.88	52.47	26.63	31.61
Four basic compass masks	36.14	-	41.05	-	20.5	-

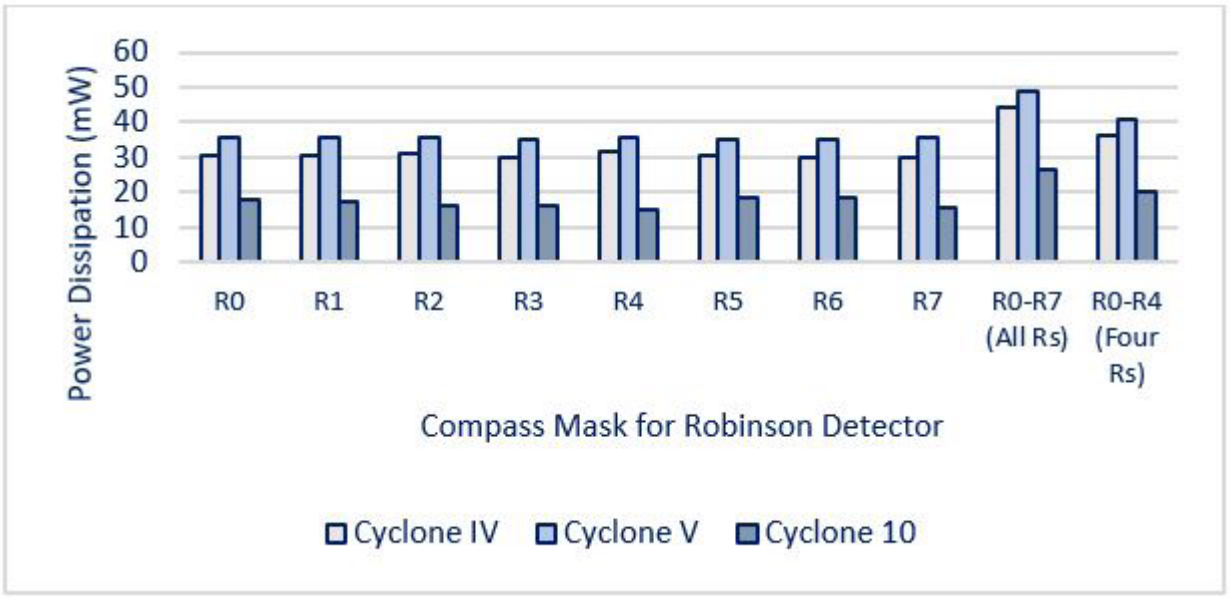


Fig. 10: The power dissipation of the Robinson detector's compass mask implemented on different FPGA devices when operating

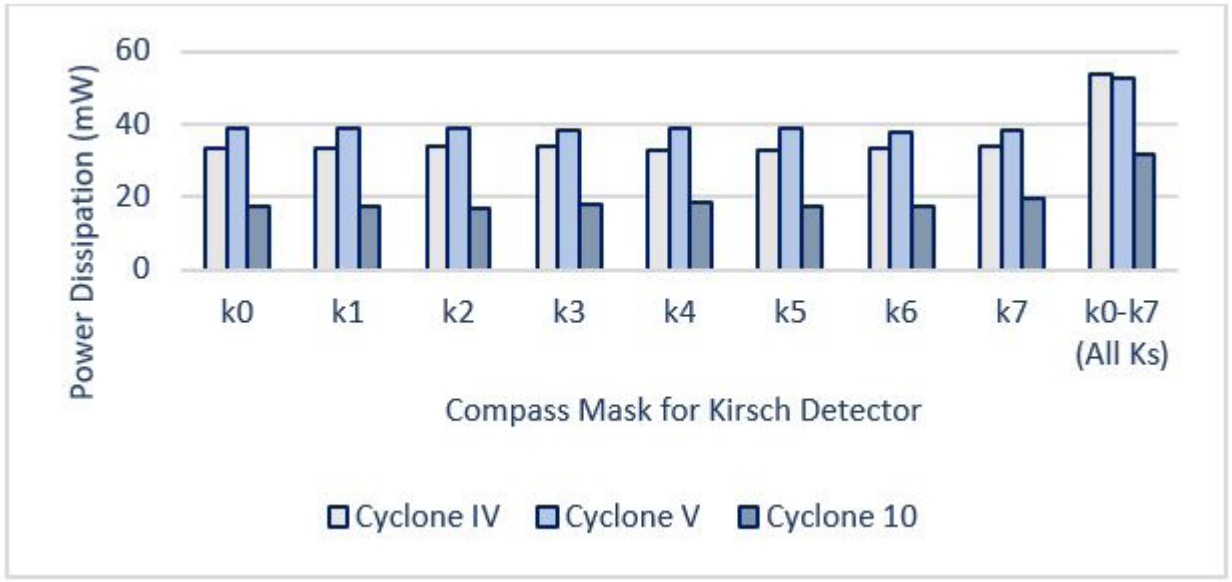


Fig. 11: The power dissipation of the Kirsch detector's compass mask implemented on different FPGA devices when operating

dissipation. In contrast, the Cyclone V FPGA is based on 28 nm semiconductor technology, offering more logic resources and lower power dissipation than previous generations when fully utilized. However, the Cyclone V logic utilization (ALMs) required to implement all eight compass masks of the Kirsch edge detector is approximately 11%, which is lower than that of Cyclone IV. Because Cyclone IV has fewer features, resources, and tasks, it consumes less power in systems with low utilization.[24-26] Furthermore, converting an RGB image to grayscale, where an integer data type represents each pixel value, addresses computational complexity and cost, as well as allowing a scaling threshold to be compared to the computed value instead of performing a division operation. Therefore, power dissipation depends on the design specifications and the CMOS technology used.

Conclusion

This research aims to develop and implement compass edge detectors on different CMOS technologies using a reconfigurable VLSI system and FPGA devices. Based on the FPGA devices used, the results of the compass

masks for the Robinson and Kirsch are analyzed and compared with respect to accuracy, edge thickness and threshold, computational cost and complexity, and power dissipation. VHDL is the HDL used to describe the digital design of both Robinson and Kirsch compass edge detectors. ModelSim, an Intel software, is used to verify the functionality of the digital design. After verification, the digital design was successfully synthesized and implemented on Cyclone IV, Cyclone V, and Cyclone 10 FPGA devices using Intel Quartus Prime.

The hardware-level results confirm the following: Kirsch compass masks use coefficients of 0, +5, and −3, whereas Robinson used simple coefficients of 0, +1, +2, −1, and −2. The Kirsch compass mask assigns greater weight to pixel intensities near pixel edges than the Robinson operator does. This causes the Kirsch masks to better detect weak edges and produce thicker edges than the Robinson masks. The threshold for detecting edges with the Kirsch design is higher because its coefficients are larger than those of the Robinson detector. Implementing the Kirsch compass edge detector design on an FPGA requires more logic elements and registers than Robinson detectors. However, using base-four masks with negative values in the Robinson edge detector reduces computational complexity and cost while dissipating less power. In contrast, the Kirsch operator must perform all eight convolutions. Additionally, digital design implemented on Cyclone V and Cyclone IV dissipates more power than on Cyclone 10. Cyclone V also dissipates more power than Cyclone IV at low utilization.

The Kirsch edge detector detects weak edges better than Robinson and is suitable for accurate applications. Thus, the optimal digital design for an edge detector, in terms of power dissipation, depends on the selected CMOS technology and design requirements.

Data Availability

The data supporting the findings of this research are available within the article.

Acknowledgements

The author would like to thank Palestine Technical University- Kadoorie (PTUK) for supporting this research.

Conflicts of Interest Statement

The authors declare no conflict of interest.

References

1. Roberts, L. (1963). Machine perception of three-dimensional solids. Ph.D. thesis, Massachusetts Institute of Technology.
2. Davis, L. S. (1975). A survey of edge detection techniques. Computer Graphics and Image Processing, 4(3), 248–270. doi:10.1016/0146-664x(75)90012-x
3. Sobel, I., and Feldman, G. (1968). A 3x3 isotropic gradient operator for image processing. a talk at the Stanford Artificial Project, 271-272.
4. Prewitt, J. (1970). Object enhancement and extraction. Picture Processing and Psychopictorics, 10(1), 15–19.
5. Robinson, G. S. (1977). Edge detection by compass gradient masks. Computer Graphics and Image Processing, 6(5), 492–501.
6. Kirsch, R. A. (1971). Computer determination of the constituent structure of biological images. Computers and Biomedical Research, 4(3), 315–328.
7. Shah, B. K., Kedia, V., Raut, R., Ansari, S., and Shroff, A. (2020). Evaluation and comparative study of edge detection techniques. IOSR Journal of Computer Engineering, 22(5), 6–15.
8. Mayangky, N. A., Merlina, N., Prasetyo, A., Amelia, D., Irsictia, M., and Putri, M. (2024). Analyzing the comparative methods of Prewitt, Robinson, Krisch and Roberts in detecting the edges of rice leaves. Jurnal Techno Nusa Mandiri, 21(1), 37–43.
9. Karanwal, S. (2021). Implementation of edge detection at multiple scales. International Journal of Engineering and Manufacturing, 11(1), 1-10. doi:10.5815/ijem.2021.01.01
10. Ohinok, S., Kulyniak, I., Kohut, M., Gerlach, I., and Voronko, O. (2025). Impact of Digital Technologies on Economic and Marketing Innovation and Competitiveness within VLSI System Employment. Journal of VLSI Circuits and Systems, 7(1), 1-10.
11. Kigarura, M., and Rshour, L. (2025). A scalable parallel processing architecture for real-time and energy-efficient biomedical signal analysis in edge-enabled health monitoring systems. Journal of Integrated VLSI, Embedded and Computing Technologies, 2(3), 56–62.
12. Punam, S. R. (2025). On-device learning-assisted predictive control for real-time trajectory planning in pervasive autonomous systems. Archives of Electronics, Communication and Emerging Technologies, 1(1), 1–9.
13. Guo, L., and Wu, S. (2023). FPGA Implementation of a real-time edge detection system based on an improved Canny algorithm. Applied Sciences, 13(2), 870. doi: 10.3390/app13020870
14. Ghodhbani, R., Saidani, T., Alhomoud, A., Alshammari, A., and Ahmed, R. (2023). Real time FPGA implementation of an efficient high speed Harris Corner detection algorithm based on high-level synthesis. Engineering, Technology and Applied Science Research, 13(6), 12169-12174. doi: 10.48084/etasr.6406
15. Narasimhamurthy, C. G., and Kulkarni, S. (2021). Fast architecture for low level vision and image enhancement for reconfigurable platform. In 2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), pp. 1–4. IEEE. doi:10.1109/icaect49130.2021.9392425

16. Noria, F., and Matharine, L. (2025). Zero-shot learning for remote sensing image segmentation using cross-domain transfer and self-training. National Journal of Signal and Image Processing, 1(2), 52–59.
17. Al-Mukhtar, F. S., and Hasso, M.A.-R. (2013). Robinson edge detector based on FPGA. International Journal of Computer Science and Information Security, 11(77), 77–81.
18. Humaidi, A. J., Fadhel, M. A., and Ajel, A. R. (2019). Lane detection system for day vision using altera DE2. TELKOMNIKA(Telecommunication Computing Electronics and Control), 17(1), 349-361. doi: 10.12928/telkomnika.v17i1.10071
19. Khudhair, Z. N., Khdiar, A. N., El Abbadi, N. K., Mohamed, F., Saba, T., Alamri, F.S., and Rehman, A. (2023). Color to grayscale image conversion based on singular value decomposition. IEEE Access, 11, 54629–54638.
20. Bailey, D.G. (2023). Design for embedded image processing on FPGAs. John Wiley & Sons.
21. Winzker, M., and Schwandt, A. (2019). Open education teaching unit for low-power design and FPGA image processing. In: 2019 IEEE Frontiers in Education Conference (FIE), pp. 1–9. IEEE. doi: 10.1109/fie43999.2019.9028694
22. AbuShanab, S., Winzker, M., and Brück, R. (2019). Development and implementation of remote laboratory as an innovative tool for practicing low-power digital design concepts and its impact on student learning, pp. 175–185. Cham: Springer International Publishing. doi: 10.1007/978-3-319-95678-7_20
23. AbuShanab, S. (2018). Remote and on-site laboratory system for low-power digital circuit design. Ph.D. thesis, Siegen. https://nbn-resolving.org/urn:nbn:de:hbz:467-13459 URN: urn:nbn:de:hbz:467-13459
24. Intel: Cyclone® IV Featured Documentation - Quick Links. (2025). https://www.intel.com/content/www/us/en/docs/programmable/767845/current/cyclone-iv-featured-documentation-quick.htm
25. 5CEBA2F17C6 DataSheet - Altera Corporation - Datasheets360.com (2025). https://www.datasheets360.com/part/detail/5ceba2f17c6/4718592257345144063/
26. Intel: Intel® Cyclone® 10 LP Device Overview (2025). https://www.intel.com/content/www/us/en/docs/programmable/683879/current/device-overview.html
27. Gonzalez, R. C., and Woods, R. E. (2018). Digital image processing. 4th Edition, Pearson Education, New York.
28. Maini, R., and Aggarwal, H. (2009). Study and comparison of various image edge detection techniques. International Journal of Image Processing, 3(1), 1–11.
29. Orhei, C., Bogdan, V., Bonchis, C., and Vasiu, R. (2021). Dilated filters for edge-detection algorithms. Applied Sciences, 11(22), 10716.