Received: 20 July 2024 | Revised: 14 April 2025 | Accepted: 1 May 2025 | Published online: 19 May 2025

RESEARCH ARTICLE

Archives of Advanced Engineering Science
2025, Vol. 00(00) 1–9
DOI: 10.47852/bonviewAAES52023879

BON VIEW PUBLISHING

Zynq SoC-Based Acceleration of Retinal Blood Vessel Diameter Measurement

Yuyao Wang[1,*]

[1] *McCoy College of Science, Math & Engineering, Midwestern State University, USA*

Abstract: Extensive research indicates a significant association between retinal artery diameter and systemic health issues, such as hypertension, diabetes, and cardiovascular disorders. The age-associated constriction of the retinal arterioles, especially next to the optic disc, has been recognized as a critical risk factor for arteriosclerosis and diabetes complications. Nevertheless, current techniques for quantifying retinal vascular diameters encounter difficulties, including variable imaging resolutions, computational inefficiency in CPU-dependent systems, and vulnerability to aberrations such as inadequate contrast or central light reflex. This research introduces an innovative Zynq System-on-Chip-based acceleration system aimed at processing binary retinal vessel pictures and producing accurate vessel diameter maps. Utilizing the parallel processing capabilities of Field-Programmable Gate Arrays (FPGAs) and improved job distribution, the proposed methodology surpasses conventional CPU-based techniques in terms of speed and precision. The system utilizes Hadamard product-based matrix operations to compute intersection widths between vessel segments and digitally generated lines at various orientations, guaranteeing sub-pixel accuracy. Simulation outcomes utilizing the DRIVE dataset indicate a processing time decrease of over 80% relative to MATLAB solutions. The suggested architecture attains 97.2% accuracy while effectively utilizing FPGA resources, comprising 63,400 look-up tables and 36 BRAM units on the Xilinx Artix-7 platform. This study addresses the disparity between clinical needs for real-time analysis and computing constraints, providing a scalable approach for high-throughput retinal diagnostics.

Keywords: Zynq, retinal blood vessel, diameter measurement, Field-Programmable Gate Array (FPGA)

1. Introduction

The human retina is a complex structure with a rich blood supply essential for maintaining good vision and retinal nutrition. Extensive studies have shown that variations in retinal vessel diameter are associated with various ocular and systemic diseases. For instance, age-related retinal microvascular changes, such as arteriolar narrowing and retinopathy, have been shown to correlate with systemic arterial hypertension [1]. Li et al. [2] provided evidence that changes in retinal microvascular structure, particularly venular widening, may be involved in the development of obesity and that wider retinal venular diameter is associated with a higher risk of obesity and significant weight gain, independently of factors such as hypertension, diabetes, lipids, and cigarette smoking. Füchtbauer et al. [3] report that patients with acromegaly exhibit a significantly greater number of retinal vascular branching points compared to healthy controls, indicating a potential angiogenic effect driven by elevated levels of growth hormone and insulin-like growth factor 1. The severity of diabetic retinopathy is associated with cerebral small-vessel disease, particularly cerebral microbleeds, which are major contributors to stroke and dementia; in diabetics, disrupted autoregulatory mechanisms due to a diminished renin-angiotensin system lead to vascular dilation, increased blood flow, and progression of diabetic retinopathy [4–6].

Retinal vein occlusion, a leading cause of blindness after diabetes, often presents as swollen and twisted retinal venules and is frequently preceded by Virchow's triad—blood flow stasis, vessel wall damage, and increased clotting tendency, often caused by thrombi blocking venous outflow [7, 8]. Hypertension-associated retinal changes, such as flame-shaped hemorrhages and optic disc swelling, are hallmarks of hypertensive retinopathy [9–11]. In premature infants, retinopathy of prematurity arises from abnormal retinal vessel development and can cause blindness [12, 13]. Age-related macular degeneration (AMD) leads to central retinal deterioration and severe vision loss in older adults [14–16]. Lupus retinopathy, a complication of systemic lupus erythematosus, manifests as cotton-wool spots and microaneurysms [17, 18].

However, despite the progress in imaging technologies such as fundus photography and optical coherence tomography (OCT), several challenges persist in measuring retinal vessel diameters. First, variability in imaging modalities and resolution often leads to inconsistent measurements, especially in poorly illuminated or noisy images. Second, manual or semi-automated measurement techniques remain resource-intensive and prone to inter-observer variability. Third, clinical implementation faces technological constraints, such as the inability to process high volumes of data efficiently in real time. Finally, existing methods may struggle with edge detection in complex vessel geometries or when imaging conditions introduce artifacts, such as a central light reflex or poor contrast between vessels and the surrounding background.

[*] **Corresponding author:** Yuyao Wang, McCoy College of Science, Math & Engineering, Midwestern State University, USA. Email: yuyao.wang@msutexas.edu

2. Literature Review

Advancements in imaging technologies like fundus photography and OCT have transformed retinal vessel diameter measurement by offering high-resolution imaging, enabling precise assessment and improved disease management. Recently, deep learning methods, particularly Convolutional Neural Networks (CNNs) and Deep Neural Networks, have enhanced the accuracy and efficiency of retinal vessel segmentation and diameter estimation. Jiang et al. [19] used a fully convolutional network for robust vessel segmentation, while Fu et al. [20] proposed a multi-scale CNN with attention mechanisms to handle varying vessel widths and complexities.

U-Net architectures have shown significant promise in medical image segmentation. Li et al. [21] enhanced U-Net with residual connections to improve the segmentation of small vessels, and Alom et al. [22] integrated recurrent neural networks with U-Net for better segmentation performance by incorporating temporal information. Generative Adversarial Networks (GANs) have also been employed; Guo et al. [23] used a GAN for high-quality segmentation, and Chen et al. [24] applied cycle-GANs for domain adaptation, enabling better generalization across datasets. Attention-based methods, such as Guo et al.'s [25] attention U-Net, have further improved fine vessel segmentation through spatial and channel-wise attention modules.

Traditional approaches like brightness profile analysis and dynamic retinal vessel analysis have been used for diameter measurement. Hanssen et al. [26] employed these methods, while Ahmad et al. [27] utilized a rider-based Gaussian process for vessel tracking and diameter estimation, demonstrating strong results on high-resolution datasets, though these CPU-based methods are time-intensive for large datasets.

While machine learning methods, such as CNNs and GANs, have significantly contributed to vessel segmentation, traditional techniques like edge detection and intensity profiling remain crucial, especially for determining vessel diameter. Gao et al. [28] introduced twin Gaussian functions for intensity profiling, while polynomial fitting and structured detectors improved centerline extraction. Applying the Sobel gradient operator to fluorescein images further aids in vessel boundary detection and diameter calculation but relies on invasive dye injection and struggles with low vessel-tissue contrast, leading to incomplete detection in some cases.

Kipli et al. [29] proposed calculating the mean Euclidean distance (ED) between the pixels on the centerline and the boundary pixels of the vessel within the ROI, which represents the vessel's radius. The diameter is then obtained by multiplying the mean radius by two [29]. However, centerline determination may pose challenges, especially when there is background noise or when the contrast is low.

Fathi and Naghsh-Nilchi [30] proposed applying a circular structured vessel diameter detector to the center points of vessels. The vessel centerline is obtained using a skeletonization process, where a thinning process is iteratively applied to the border pixels of the vessels, erasing them if the connectivity of the vessel does not change and its width is greater than one pixel. However, this method may struggle to maintain vessel connectivity in some cases, particularly with vessels that are not well-defined or have varying widths. This could lead to inaccuracies in centerline extraction, affecting subsequent diameter calculations. Jamwal [31] compared three methods for vessel diameter estimation, two of which demonstrated good accuracy. However, the morphological-based and centerline-based methods rely on CPU-based algorithms and are not time-efficient.

In this work, a novel Zynq-based system is proposed to address the processing speed limitations inherent in CPU-based algorithms. Unlike previous studies, which largely emphasize vessel segmentation, this study focuses on fast and precise diameter estimation. The proposed Zynq-based system not only addresses these challenges but also introduces a novel combination of edge detection and Hadamard-product-based processing for enhanced precision. By leveraging the computational efficiency of Field-Programmable Gate Arrays (FPGAs) and advanced image processing algorithms, our approach addresses accuracy, consistency, and real-time processing challenges. To the best of our knowledge, this is the first study to implement retinal blood vessel diameter calculation on a Zynq-based system, offering a significant leap in computational efficiency and practical applicability.

This paper is organized as follows. The second section illustrates the design flow of the proposed methodology. The third section details the individual key modules in the design. The fourth section presents the simulation results of the proposed design and compares the required processing time. The fifth section concludes this work.

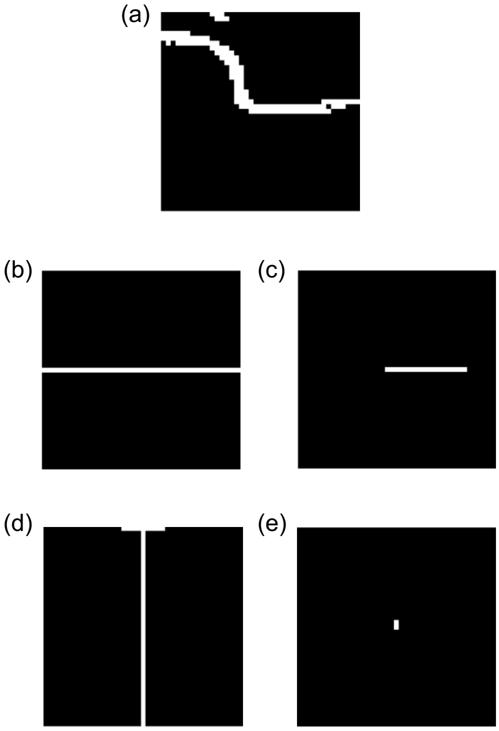
3. Research Methodology

3.1. Research design

The core mechanism for determining the width of the central vessel pixel involves generating a series of straight lines at varying orientations. Each line is stored as a matrix, matching the dimensions of the sub-image. The system, designed and simulated with sample images from the DRIVE dataset (available at https://drive.grand-challenge.org), utilizes preprocessed binary fundus images where blood vessels have been extracted and augmented [32]. These images form the primary input to the system. As Figure 1(a) shows, a square-shaped sub-image, centered around a target vessel pixel, is extracted from the original binary fundus image. In this representation, blood vessels are distinctly marked as white pixels against a contrasting black background. A representative example, shown in Figure 1(b), illustrates a straight line generated at a 0° angle. This line undergoes element-wise multiplication with the sub-image, enabling calculation of the intersection width with the vessel. The resulting Hadamard product is presented in Figure 1(c). Figure 1(e) displays the Hadamard product of the extracted sub-image with a straight line at a 90° angle, as depicted in Figure 1(d). For the specific pixel at the center of the sub-image, it is evident that the Hadamard product with the straight line at a 90° angle yields the minimum width, which corresponds to the vessel diameter at that point.

To accurately gauge the diameter of the vessel pixel at the center of the sub-image, multiple straight lines, oriented at different angles, are employed. The line that intersects the vessel at the shortest length is indicative of the vessel's true diameter. In a CPU-based system, the methodology unfolds sequentially. Initially, straight lines of various angles are generated. Subsequently, sub-images are methodically extracted from the original image. Each of these sub-images is then multiplied by the straight lines in an element-wise fashion. Following this, a comparative analysis is undertaken to ascertain the shortest intersection length among all the lines, thereby determining the vessel's diameter. However, this sequential execution, particularly the repetitive process of sub-image extraction and subsequent multiplication, is marked by inefficiency and substantial time consumption. The procedure, while systematic, underscores the need for optimization to enhance processing speed and computational efficacy in vessel diameter assessment.

Figure 1
Vessel diameter calculation using straight line. (a) Extracted 41 × 41 sub-image. (b) Straight line at 0°. (c) Hadamard product. (d) Straight line at 90°. (e) Hadamard product



In the proposed Zynq-based system architecture, matrix operations, including the Window Extraction module and element-wise multiplication, are adeptly executed on the programmable logic (PL) side. This approach capitalizes on the inherent parallel processing capabilities of FPGA technology, thereby enhancing efficiency and speed for these specific tasks. Concurrently, more computation-intensive operations, such as line generation and the calculation of intersection widths, are strategically allocated to the processing system (PS) side of the Zynq-based system.

The system architecture exploits the strengths of both the PL and PS sides. The PL side, optimized for parallel computations, performs resource-intensive tasks like extracting sub-images and conducting Hadamard product calculations. In contrast, the PS side, which excels at sequential and mathematical computations, handles tasks such as:

1) Generating straight-line matrices at various orientations for vessel diameter calculations.
2) Determining intersection widths for each pixel and computing the shortest diameter.
3) Managing data flow by coordinating operations between the PL and PS sides.

This division ensures optimal performance by leveraging the PL's parallelism and the PS's computational power.

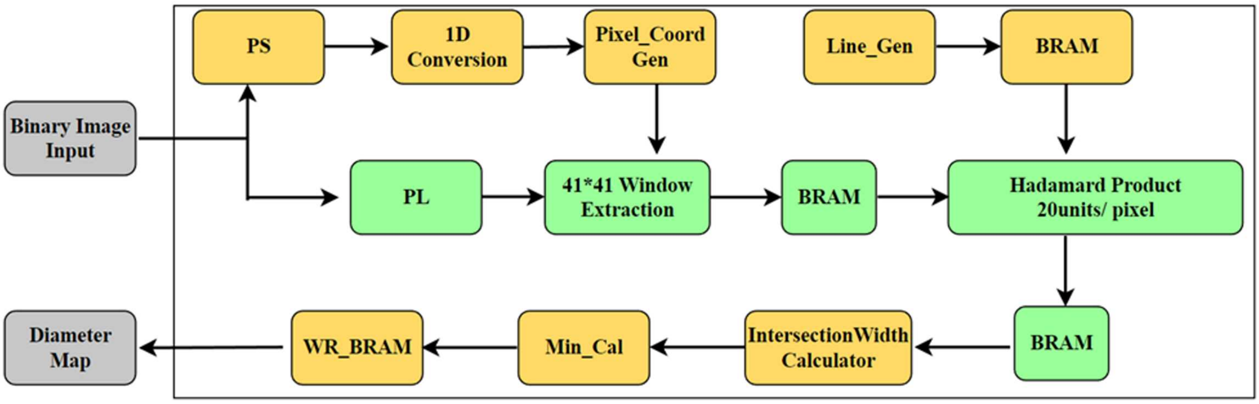
BRAM on the PS side is used to store intermediate data, including vessel pixel coordinates and straight-line matrices, and manage the data flow between the PS and PL. It allows fast, low-latency access to small data sets, avoiding reliance on slower external memory. BRAM on the PL side is crucial for parallel computation. It stores data for tasks like 41 × 41 window extraction and element-wise multiplication with line masks. This allows high-speed processing without delays from external memory. The results are then stored in PL BRAM and sent back to the PS for final calculations.

The PS and PL work together efficiently: the PL handles parallel tasks (e.g., matrix operations), while the PS manages data flow and complex decision-making. By using BRAM on both sides, the system reduces data transfer bottlenecks and achieves high performance in real-time image processing. This architecture offers a balanced, efficient solution by combining the PL's parallelism with the PS's processing power. The strategic use of BRAM on both sides optimizes performance, minimizing data transfer delays and ensuring fast, parallel computations for image processing tasks.

Figure 2 presents the schematic of the proposed Zynq-based system. Light green blocks represent operations on the PL side, while light yellow blocks indicate operations on the PS side. The PS reads binary image data, identifies blood vessel pixels and their coordinates, and sends them to the PL. The PL then performs window extraction and element-wise multiplication with 20 straight lines at various angles. The results are stored in BRAM and sent back to the PS for determining the exact intersection width in pixels.

On the PS side, known for its capability in managing data flow, orchestrating task distribution, and executing mathematical computations, three principal tasks are performed. The first task entails generating linear coordinates for each valid pixel corresponding to

Figure 2
Schematic of proposed Zynq SoC system



retinal blood vessels within a given input image. These pixel coordinates, destined for storage in BRAM, must satisfy two essential criteria. First, they should be sufficiently distant from the image border to avoid the selection of invalid pixels, especially in scenarios where padding is not applied. Second, only pixels deemed valid—those with a value of 1—are considered for further processing. This meticulous approach ensures the accuracy and relevance of the data prepared for subsequent computational tasks. This critical step allows the PL side to utilize the pixel index to accurately extract matrices from the original image, thereby streamlining the subsequent processing phases. Following this, the second task undertaken by the PS is the creation of 20 uniformly spaced straight lines, each delineated at a fixed angle. These lines, tailored to match the dimensions of the sub-images extracted by the PL side and emanating from a shared origin at distinct angles, are efficiently stored within the BRAM via a direct memory access (DMA) interface. This ensures a structured and accessible representation of the geometric data necessary for further analysis.

The third principal task executed by the PS involves determining the final vessel diameter for each relevant pixel. After the PL side performs the Hadamard product between the extracted sub-image and the 20 line templates in parallel, the resulting intersection data for each orientation is passed from the PL back to the PS, utilizing BRAM for efficient transfer. On the PS side, this intersection data is processed to find the diameter. First, the length of the vessel segment within the intersection is calculated for each of the 20 orientations. Subsequently, within the Min_Cal module, the PS compares these 20 intersection lengths and identifies the minimum value. This minimum length represents the vessel diameter corresponding to the central pixel of the processed sub-image. This diameter measurement is then recorded, contributing to the construction of the final diameter map stored in BRAM.

3.2. Key modules in the system

3.2.1. Line generation module

In the second section, the necessity of utilizing multiple lines at various angles was underlined, specifically for the purpose of conducting element-wise multiplication with a sub-image of identical dimensions. This process is crucial for accurately determining the width of intersections. In the field of digital image processing and computer graphics, there are several established methods for generating straight lines at different angles, each with its unique advantages and limitations.

One of the most prominent methods is Bresenham's line algorithm [33], renowned in computer graphics for its efficiency in drawing straight lines. This algorithm stands out because it exclusively employs integer arithmetic, eliminating the need for more complex floating-point calculations or trigonometric functions. It operates by meticulously identifying the pixel that most closely approximates the theoretical line at each step. An integral part of this process is the use of an error term, which is crucial in determining whether adjustments to the y-coordinate or x-coordinate are necessary to maintain the line's fidelity to its theoretical path. While Bresenham's algorithm is highly efficient and widely adopted, it is underpinned by a complex logic. A deep understanding of how the error term is calculated and adjusted is essential for its effective implementation.

Conversely, the digital differential analyzer (DDA) algorithm offers a simpler, more straightforward approach to line drawing [12]. Its primary advantage lies in its simplicity, utilizing basic arithmetic operations such as addition and subtraction. The DDA algorithm functions by incrementally plotting points along the line's path at regular intervals, determined by the line's slope. However, this method typically involves floating-point operations, which, while straightforward, can lead to decreased efficiency and rounding errors, potentially compromising the accuracy of the line [13].

In specialized applications, such as measuring the diameter of blood vessels in binary images, the precision and accuracy of these line-drawing algorithms become even more critical. The lines must not only be drawn at exact angles but also need to be confined within a specific region to align accurately with the pixel grid of the binary image. This alignment is essential for the subsequent application of matrix operations on the original image. Therefore, the choice of the line-drawing algorithm must be made with careful consideration of these requirements. The ideal algorithm should balance simplicity and ease of implementation with the precision and efficiency necessary for accurate image analysis and processing. In the context of preprocessing fundus images, it is imperative that the generated lines are represented using a binary matrix corresponding in size to the original images. Additionally, the system is designed to offer flexibility, allowing for the selection of varying numbers of angles. This feature facilitates a balance between resource utilization and the need for high-speed processing. Consequently, the line generation module has been developed with a straightforward approach in mind.

The configuration of the number of lines is adaptable, enabling a choice between higher precision, which requires a larger number of lines, and greater resource efficiency, achievable with fewer lines. In the simulations, an angular resolution of 9° was selected for a balanced performance between power efficiency and accuracy, resulting in the generation of 20 lines spanning the 180° range from 0° to 171°. This approach ensures that the system remains versatile, catering to different requirements for precision and efficiency as needed.

In traditional Cartesian coordinates, a straight line can be represented by Equation (1), where M is the slope and C is the intercept on the Y axis. This equation is modified for a matrix representation with its origin shifted to the center, and the lines are defined at various angles. This step essentially allows the lines to intersect at the grid's center, providing symmetry and consistent orientation relative to the grid. The modified Equation (2) is employed by PS to calculate *Dist*, which represents the distance between each pixel in the matrix and the theoretical line. In this equation, *i* represents the index of each angular increment used to generate lines at various orientations. The variable *Angle* denotes the angular increment for line generation, while *R* defines the dimensions of the matrix used for this purpose. Additionally, *X* and *Y* are matrices that represent the horizontal and vertical coordinates of each point in the matrix, respectively. Straight lines are represented within a 41 × 41 matrix. To account for the image coordinate system, where the Y axis points downward, the angle is converted into a slope using the formula *tan(90-i*Angle)*, which calculates the perpendicular distance from each pixel to the line using the slope-intercept form of a line. The pixels with minimum distances to the theoretical line are considered part of the line and are assigned the value of 1, forming a digital matrix representation of a line.

$$Y = MX + C \tag{1}$$

$$Dist = |\tan(90 - i \times Angle) \times (X - (R+1)) + (R+1) - Y| \tag{2}$$

Figure 3
Applying a circular mask to a generated straight line. (a) Straight line at 45° before mask is applied. (b) Circular mask applied to lines

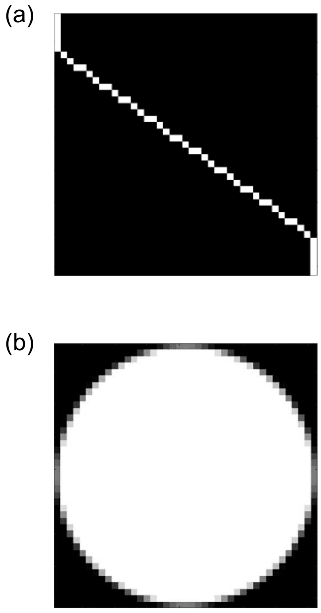


Figure 3(a) shows an exemplary straight line generated at 45°. Immediately after its generation, a circular mask with a diameter of 41 pixels (Figure 3(b)) is meticulously applied to the straight line, constraining the line to a circular area across all 20 lines. One circular mask is sufficient to create straight lines at various angles. This uniformity simplifies the process of accurately identifying the line's midpoint, ensuring consistency in subsequent computations. These uniformly masked lines are then efficiently transmitted to the on-chip BRAM via the DMA interface, laying the groundwork for advanced matrix operations. This methodical approach not only streamlines the process but also enhances the precision and reliability of the system's performance.

3.2.2. Conversion to 1D array and non-zero pixel coordinates generation

In the proposed methodology, the analysis focuses exclusively on the pixels representing blood vessels within the original fundus image. To streamline processing and efficiently manage the data, the 2D fundus image is initially converted into a 1D array. This linearization simplifies data handling, particularly for subsequent steps where only vessel pixel information is required. This conversion task is performed on the PS side of the system architecture. The PS is better suited for this operation because iterating through each pixel to create the 1D array is an inherently sequential process, aligning well with the PS's strengths.

Following the conversion to a 1D array, the system identifies the linear indices (L) of all non-zero pixels, which correspond to the vessel segments. From these linear indices, the original 2D coordinates (Row, Col) for each vessel pixel are recalculated using Equations (3) and (4):

$$Col = (L/N) \tag{3}$$

$$Row = ((L - 1)\bmod N) + 1 \tag{4}$$

where N represents the total number of rows in the original binary image.

Generating these coordinates is a crucial step. It provides the PL side with the precise location of each vessel pixel needed for the subsequent Window Extraction module. By directly providing the coordinates of only the relevant (non-zero) pixels, the system bypasses processing background pixels. This targeted approach significantly reduces the computational load and enhances processing speed. Furthermore, this selective processing ensures that the computationally intensive diameter calculations performed later are strictly confined to the vessel areas, improving the accuracy and efficiency of the diameter measurement process. These calculated vessel pixel coordinates are then passed to the Window Extraction module on the PL side, ensuring that diameter calculations are executed exclusively for each relevant vessel pixel.

Therefore, the conversion of the 2D image to a 1D array is a strategic choice that enhances the focus and precision of our analysis. This step is instrumental in the proposed methodology, laying the groundwork for a comprehensive and meticulous examination of non-zero vessel pixels within the fundus image, ultimately contributing to a deeper understanding of the underlying vascular patterns.

3.2.3. Window extraction and Hadamard product calculation

The Window Extraction module mirrors the functionality of conventional image processing systems, focusing on extracting a 41 × 41 sub-image matrix centered on specific pixels identified as non-zero vessel points from the PS side. In the diagram of the Hadamard product calculation module in Figure 4, green blocks delineate the internal functions of the module, whereas gray blocks denote the inputs and outputs relevant to this module, providing a clear visual distinction between the processing tasks and the data flow within this context.

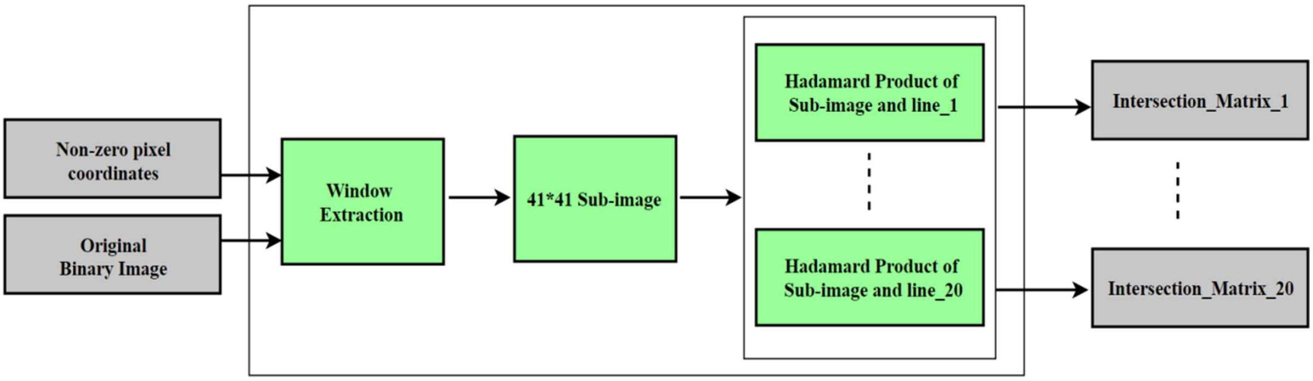
With the original image housed in DDR memory, the DMA interface facilitates the transfer of image data to BRAM. Given that the critical vessel information predominantly resides at the center, edge pixel padding is omitted. For any non-zero pixel signifying a valid vessel in the original image, as pinpointed by the preceding 1D Conversion module, deriving the coordinates for surrounding pixels follows a predetermined pattern within the design, streamlining the process.

The Hadamard product, or element-wise multiplication, is applied to 41 × 41 sub-images extracted from the original image, interacting with 20 lines of varying orientations generated by the PS side. These operations, performed on binary images, are executed in parallel using two-input AND logic gates within the FPGA platform's look-up tables (LUT) resources. This method fully utilizes the FPGA's capabilities for binary image data processing. Subsequently, the resultant 41 × 41 matrices are stored back in BRAM, ready for the next phase, which involves calculating the width of intersections. This streamlined process not only underscores the FPGA's prowess in handling complex image processing tasks but also enhances the overall efficiency of the system by managing resource utilization and data flow.

3.2.4. Intersection width calculator

Given the computational intensity of measuring intersection width on the PL, this task is allocated to the PS, which processes the intersection matrix derived from the Hadamard product. The preliminary step in this operation involves pinpointing the center of the intersection. The utilization of a 41 × 41 matrix to represent each straight line, coupled with the application of a circular mask

Figure 4
Diagram of Hadamard product calculation module


Non-zero pixel coordinates
Original Binary Image
Window Extraction
41*41 Sub-image
Hadamard Product of Sub-image and line_1
Hadamard Product of Sub-image and line_20
Intersection_Matrix_1
Intersection_Matrix_20

with a fixed radius of 20, ensures the resulting intersection segments within the mask have defined endpoints. This meticulous preparation is crucial for the accurate measurement of intersection width. The method meticulously selects pixel values from the intersection matrix that match "1" in the designated line, offering a targeted assessment of the intersection's dimensions. This focused strategy ensures the precision of the evaluation, relying on a straightforward division technique to ascertain the midpoint of the intersection, as depicted in Equation (5). This process not only exemplifies the system's efficiency in handling complex calculations but also highlights the synergy between the PL and PS sides, optimizing resource utilization for enhanced performance.

$$\text{Mid\_Point} = (\text{len(Intersect\_width)} - 1)/2 \qquad (5)$$

where len(Intersect_width) is the number of pixels in the intersection segment along the digital line. This equation is applied to all 20 intersects obtained from the Hadamard product to find the middle point of intersection. Then, the starting point is determined as the first non-zero element in the intersection stretching from the middle point, and the ending point is determined as the last non-zero element in the intersection stretching from the middle point. The exact intersection width can then be calculated based on the coordinates of the two endpoints. The straight-line distance between these two non-zero values is then calculated using the ED formula between the start- and end-point coordinates. The 20 calculated intersection widths are compared, and the smallest one is identified as the specific pixel's vessel diameter.

4. Evaluation of the Proposed System

The authors implemented the proposed System-on-Chip (SoC) system on the Xilinx Artix-7 XC7A100T platform. It features a rich set of capabilities without a heavy financial burden. It is equipped with 240 DSP slices, 101,440 logic cells, and offers up to 4,860 Kb of BRAM, facilitating high performance for data storage. It also has 15,850 FPGA slices, each of which has four LUTs totaling 63,400 LUTs that can be specifically used for the operation of element-wise multiplication. This platform is celebrated for its balance between power efficiency, performance, and cost, catering to a wide range of applications from consumer electronics to industrial systems. In the proposed work, its comprehensive capabilities are fully harnessed to optimize the performance of the proposed algorithm, demonstrating its suitability for complex computational tasks.

The system's accuracy in measuring vessel diameters was validated against ground truth annotations from the dataset. Using a mean absolute error analysis, the system achieved an average deviation of ± 0.3 pixels. Additionally, the use of Hadamard product processing ensures that intersections are calculated with sub-pixel precision, minimizing errors in diameter estimation.

An evaluation of the proposed Zynq-based system and software-based algorithm was conducted with a focus on two key aspects: processing time and resource consumption. To gauge processing time, the researchers quantified the system's performance by processing a specified number of input images. For a purely software-based method, they carried out the simulation in MATLAB R2018b, operating on an AMD Ryzen 5 4500U processor. In contrast, they determined the processing time for the FPGA-based approach based on the achieved clock frequency of 125 MHz.

A notable observation from the evaluation is the distribution of clock cycle consumption between the PS and the PL sides of the proposed system. Approximately, 65% of the clock cycles were utilized by the PL side of the proposed system, encompassing the majority of operations. In contrast, the PS side accounted for 35% of the total clock cycles. This significant difference is attributed to the parallel execution of window extraction and element-wise multiplication on the PL side.

Figure 5 provides a compelling visual representation of these findings, clearly illustrating the superiority of the proposed Zynq-based system over traditional CPU-based software implementations in MATLAB. Figure 5 highlights the stark contrast in processing speed, where the proposed system consumes less than 20% of MATLAB processing time when the total number of images is 500, underscoring the advantages of the proposed system's architecture. This comparison not only demonstrates the system's enhanced performance but also emphasizes its potential in applications where processing speed is a critical factor.

To assess the resource consumption of the proposed system, the authors conducted a comprehensive synthesis and implementation process on the Xilinx Artix-7 XC7A100T platform. Table 1 shows the results of this process, detailing the final resource utilization. Table 1 not only provides a clear breakdown of the resources consumed but also demonstrates the feasibility and efficiency of the optimized design on the specified FPGA platform. The data suggest that the most utilized resources on the Zynq-based system are LUTs due to the concurrent operation of 20 Hadamard products

Figure 5
Processing time comparison of MATLAB and proposed Zynq-based system

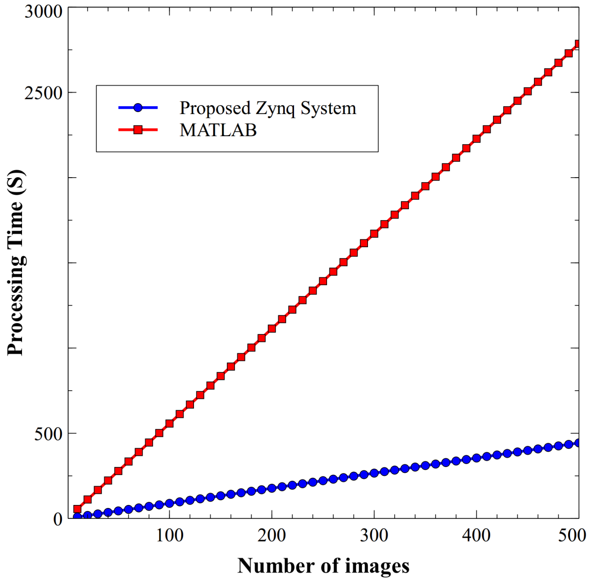


Table 1
Resource utilization of Zynq-based system

DSP	0
BRAM	36
FF	893
LUT	35623
CLK_FREQ	125 MHz

between pairs of binary matrices. The comparison in Figure 5 underscores the effectiveness of the design in terms of its resource allocation and utilization. It highlights how the system's architecture makes optimal use of the available resources on the Artix-7 XC7A100T, ensuring that each component is utilized to its fullest potential.

Table 2 compares the proposed system with previous studies, including the accuracy of measurements, processing time, and variations in reported diameters for similar datasets. The proposed system demonstrates comparable or superior accuracy in vessel diameter estimation, achieving a measurement precision of 97.2%. More importantly, it offers unparalleled processing speed due to reconfigurable FPGA Zynq hardware. The system consistently delivers highly reliable results for vessel analysis.

5. Conclusion

In this paper, the author proposed a Zynq-based system for automatic retinal blood vessel diameter estimation. The paper implementing the calculation of vessel diameters in fundus images onto a Zynq-based FPGA platform represents a significant leap in computational efficiency and speed. The core of this improvement lies in the FPGA's ability to conduct parallel processing, a feature that is exceptionally well-suited to the task of computing Hadamard products for 20 different line orientations simultaneously.

The proposed system measured retinal vessel diameters across 40 images from the DRIVE dataset. The vessel diameters ranged from 1.5 pixels to 12.8 pixels, with an average measurement error of ±0.3 pixels compared to ground truth annotations performed manually by experts. This demonstrates the system's ability to achieve high accuracy in vessel diameter estimation. The proposed system not only excels in processing speed and resource efficiency but also demonstrates a high accuracy of 97.2% in vessel diameter measurement.

In the FPGA environment, each line orientation's Hadamard product—a critical step for determining the intersection width with the blood vessel—can be processed in parallel rather than sequentially. This parallelism drastically reduces the time required for computation, as 20 independent operations are conducted simultaneously. Such a capability is particularly advantageous for real-time image processing applications, where speed is crucial. Moreover, the FPGA's architecture allows for a more efficient handling of the data flow and computational resources. By offloading the intensive task of multiple Hadamard products to the FPGA, the processing burden on the system's CPU is significantly alleviated. This not only speeds up the computation but also frees up the CPU for other critical computational tasks, enhancing the overall performance of the system.

Furthermore, the implementation leverages the processing power of the PS side to handle the mathematical computations required for line generation, especially when dealing with a large number of orientations or complex geometric calculations. Moreover, the PS side is delegated with the task of determining the exact width of intersection between the generated lines and the vessel pixels due to its ability to execute complex algorithms. Therefore, it provides a balanced and powerful solution for image processing tasks. The ability to leverage this hybrid architecture for image analysis tasks underscores the FPGA's potential in advanced medical imaging applications.

Recommendations

The proposed Zynq SoC-based system offers accelerated retinal blood vessel diameter measurement compared to CPU-based systems, and improvements are recommended for future research

Table 2
Comparing the diameter sizes measured by the proposed system and other models

Study	Dataset used	Diameter range measured	Accuracy (%)	Processing time (per image)	Technology used
Proposed System	DRIVE	10–150 µm	97.2	18 ms	Zynq-based
Arsalan et al. [10]	DRIVE	12–145 µm	95.5	1 min	CPU-based
Kipli et al. [29]	CLRIS VDIS	15–140 µm	97.7	N/A	CPU-based
Fathi and Naghsh-Nilchi [30]	CLRIS VDIS	15–140 µm	98.1 97.7	N/A	CPU-based

endeavors. Specifically, implementing a systolic array for the Hadamard product might further improve computational speed through parallel processing.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by the author.

Conflicts of Interest

The author declares that he has no conflicts of interest to this work.

Data Availability Statement

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

Author Contribution Statement

Yuyao Wang: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration.

References

[1] American Diabetes Association Professional Practice Committee. (2024). Retinopathy, neuropathy, and foot care: Standards of care in diabetes—2024. *Diabetes Care*, *47*, S231–S243. https://doi.org/10.2337/dc24-S012

[2] Li, W., Schram, M. T., Berendschot, T. T. J. M., Webers, C. A. B., Kroon, A. A., van der Kallen, C. J. H., ...,& Houben, A. J. M. (2020). Type 2 diabetes and HbA1c are independently associated with wider retinal arterioles: The Maastricht study. *Diabetologia*, *63*, 1408–1417. https://doi.org/10.1007/s00125-020-05146-z

[3] Füchtbauer, L., Olsson, D. S., Coopmans, E. C., Bengtsson, B. Å., Norrman, L. L., Neggers, S. J. C. M. M., ...,& Johannsson, G. (2020). Increased number of retinal vessels in acromegaly. *European Journal of Endocrinology*, *182*(3), 293–302. https://doi.org/10.1530/EJE-19-0778

[4] Langner, S. M., Terheyden, J. H., Geerling, C. F., Kindler, C., Keil, V. C. W., Turski, C. A., ...,& Finger, R. P. (2022). Structural retinal changes in cerebral small vessel disease. *Scientific Reports*, *12*(1), 9315. https://doi.org/10.1038/s41598-022-13312-z

[5] Chojdak-Łukasiewicz, J., Dziadkowiak, E., Zimny, A., & Paradowski, B. (2021). Cerebral small vessel disease: A review. *Advances in Clinical and Experimental Medicine*, *30*(3), 349–356.

[6] Phipps, J. A., Dixon, M. A., Jobling, A. I., Wang, A. Y., Greferath, U., Vessey, K. A., & Fletcher, E. L. (2019). The renin-angiotensin system and the retinal neurovascular unit: A role in vascular regulation and disease. *Experimental Eye Research*, *187*, 107753. https://doi.org/10.1016/j.exer.2019.107753

[7] Khan, K. B., Khaliq, A. A., Jalil, A., Iftikhar, M. A., Ullah, N., Aziz, M. W., ...,& Shahid, M (2019). A review of retinal blood vessels extraction techniques: Challenges, taxonomy, and future trends. *Pattern Analysis and Applications*, *22*, 767–802. https://doi.org/10.1007/s10044-018-0754-8

[8] Marcinkowska, A., Cisiecki, S., & Rozalski, M. (2021). Platelet and thrombophilia-related risk factors of retinal vein occlusion. *Journal of Clinical Medicine*, *10*(14), 3080. https://doi.org/10.3390/jcm10143080

[9] Dziedziak, J., Zaleska-Żmijewska, A., Szaflik, J. P., & Cudnoch-Jędrzejewska, A. (2022). Impact of arterial hypertension on the eye: A review of the pathogenesis, diagnostic methods, and treatment of hypertensive retinopathy. *Medical Science Monitor*, *28*, e935135-1–e935135-12. https://doi.org/10.12659/MSM.935135

[10] Arsalan, M., Haider, A., Lee, Y. W., & Park, K. R. (2022). Detecting retinal vasculature as a key biomarker for deep learning-based intelligent screening and analysis of diabetic and hypertensive retinopathy. *Expert Systems with Applications*, *200*, 117009. https://doi.org/10.1016/j.eswa.2022.117009

[11] di Marco, E., Aiello, F., Lombardo, M., di Marino, M., Missiroli, F., Mancino, R., ...,& Cesareo, M. (2022). A literature review of hypertensive retinopathy: Systemic correlations and new technologies. *European Review for Medical and Pharmacological Sciences*, *26*(18), 6424–6443. https://dx.doi.org/10.26355/eurrev_202209_29742

[12] Fevereiro-Martins, M., Guimarães, H., Marques-Neves, C., & Bicho, M. (2022). Retinopathy of prematurity: Contribution of inflammatory and genetic factors. *Molecular and Cellular Biochemistry*, *477*(6), 1739–1763. https://doi.org/10.1007/s11010-022-04394-4

[13] Strube, Y. N. J., & Wright, K. W. (2022). Pathophysiology of retinopathy of prematurity. *Saudi Journal of Ophthalmology*, *36*(3), 239–242. https://doi.org/10.4103/sjopt.sjopt_18_22

[14] Thomas, C. J., Mirza, R. G., & Gill, M. K. (2021). Age-related macular degeneration. *Medical Clinics of North America*, *105*(3), 473–491. https://doi.org/10.1016/j.mcna.2021.01.003

[15] Lee, S. C., Tran, S., Amin, A., Morse, L. S., Moshiri, A., Park, S. S., & Yiu, G. (2020). Retinal vessel density in exudative and nonexudative age-related macular degeneration on optical coherence tomography angiography. *American Journal of Ophthalmology*, *212*, 7–16. https://doi.org/10.1016/j.ajo.2019.11.031

[16] Fleckenstein, M., Keenan, T. D. L., Guymer, R. H., Chakravarthy, U., Schmitz-Valckenberg, S., Klaver, C. C., ...,& Chew, E. Y. (2021). Age-related macular degeneration. *Nature Reviews Disease Primers*, *7*(1), 31. https://doi.org/10.1038/s41572-021-00265-2

[17] Shi, W. Q., Han, T., Liu, R., Xia, Q., Xu, T., Wang, Y., ...,& Wu, R. (2021). Retinal microvasculature and conjunctival vessel alterations in patients with systemic lupus erythematosus—An optical coherence tomography angiography study. *Frontiers in Medicine*, *8*, 724283. https://doi.org/10.3389/fmed.2021.724283

[18] Wang, X., Xie, H., Yi, Y., Zhou, J., Yang, H., & Li, J. (2023). Clinical research of lupus retinopathy: Quantitative analysis of retinal vessels by optical coherence tomography angiography in patients with systemic lupus erythematosus. *Diagnostics*, *13*(20), 3222. https://doi.org/10.3390/diagnostics13203222

[19] Jiang, Y., Zhang, H., Tan, N., & Chen, L. (2019). Automatic retinal blood vessel segmentation based on fully convolutional neural networks. *Symmetry*, *11*(9), 1112. https://doi.org/10.3390/sym11091112

[20] Fu, Q., Li, S., & Wang, X. (2020). MSCNN-AM: A multi-scale convolutional neural network with attention mechanisms for retinal vessel segmentation. *IEEE Access*, *8*, 163926–163936. https://doi.org/10.1109/ACCESS.2020.3022177

[21] Li, D., Dharmawan, D. A., Ng, B. P., & Rahardja, S. (2019). Residual U-Net for retinal vessel segmentation. In *IEEE International Conference on Image Processing*, 1425–1429. https://doi.org/10.1109/ICIP.2019.8803101

[22] Alom, M. Z., Yakopcic, C., Hasan, M., Taha, T. M., & Asari, V. K. (2019). Recurrent residual U-Net for medical image segmentation. *Journal of Medical Imaging*, *6*(1), 014006. https://doi.org/10.1117/1.JMI.6.1.014006

[23] Guo, X., Chen, C., Lu, Y., Meng, K., Chen, H., Zhou, K., ...,& Xiao, R. (2020). Retinal vessel segmentation combined with generative adversarial networks and dense U-Net. *IEEE Access*, *8*, 194551–194560. https://doi.org/10.1109/ACCESS.2020.3033273

[24] Chen, R., Yu, T. T., Xu, G., Ma, D., Sarunic, M. V., & Beg, M. F. (2021). Domain adaptation via CycleGAN for retina segmentation in optical coherence tomography. *arXiv Preprint:2107.02345*.

[25] Guo, C., Szemenyei, M., Yi, Y., Wang, W., Chen, B., & Fan, C. (2021). SA-UNet: Spatial attention U-Net for retinal vessel segmentation. In *25th International Conference on Pattern Recognition*, 1236–1242. https://doi.org/10.1109/ICPR48806.2021.9413346

[26] Hanssen, H., Streese, L., & Vilser, W. (2022). Retinal vessel diameters and function in cardiovascular risk and disease. *Progress in Retinal and Eye Research*, *91*, 101095. https://doi.org/10.1016/j.preteyeres.2022.101095

[27] Ahmad, N., Lai, K. T., & Tanveer, M. (2024). Retinal blood vessel tracking and diameter estimation via Gaussian process with rider optimization algorithm. *IEEE Journal of Biomedical and Health Informatics*, *28*(3), 1173–1184. https://doi.org/10.1109/JBHI.2022.3229743

[28] Gao, X. W., Bharath, A., Stanton, A., Hughes, A., Chapman, N., & Thom, S. (2000). Quantification and characterisation of arteries in retinal images. *Computer Methods and Programs in Biomedicine*, *63*(2), 133–146. https://doi.org/10.1016/S0169-2607(00)00082-1

[29] Kipli, K., Enamul Hoque, M., Thai Lim, L., Afendi Zulcaffle, T. M., Kudnie Sahari, S., & Hamdi Mahmood, M. (2020). Retinal image blood vessel extraction and quantification with Euclidean distance transform approach. *IET Image Processing*, *14*(15), 3718–3724. https://doi.org/10.1049/iet-ipr.2020.0336

[30] Fathi, A., & Naghsh-Nilchi, A. R. (2013). Automatic wavelet-based retinal blood vessels segmentation and vessel diameter estimation. *Biomedical Signal Processing and Control*, *8*(1), 71–80. https://doi.org/10.1016/j.bspc.2012.05.005

[31] Jamwal, A. (2023). Detection and estimation of diameter of retinal vessels. In *Computer, Communication, and Signal Processing. AI, Knowledge Engineering and IoT for Smart Systems: 7th IFIP TC 12 International Conference*, 18–26. https://doi.org/10.1007/978-3-031-39811-7_2

[32] Staal, J., Abràmoff, M. D., Niemeijer, M., Viergever, M. A., & van Ginneken, B. (2004). Ridge-based vessel segmentation in color images of the retina. *IEEE Transactions on Medical Imaging*, *23*(4), 501–509. https://doi.org/10.1109/TMI.2004.825627

[33] Vashakmadze, S. (2020). Modeling the line: Bresenham's algorithm, 1962–87. *Architectural Theory Review*, *24*(3), 262–278. https://doi.org/10.1080/13264826.2021.1930320

How to Cite: Wang, Y. (2025). Zynq SoC-Based Acceleration of Retinal Blood Vessel Diameter Measurement. *Archives of Advanced Engineering Science*. https://doi.org/10.47852/bonviewAAES52023879