

Real Time Implementation of Detection of Bacteria in Microscopic Images Using System Generator

Anis Ladgham¹, Fayçal Hamdaoui^{1*}, Anis Sakly² and Abdellatif Mtibaa^{1,2}

¹Laboratory EμE, Faculty of Sciences of Monastir, University of Monastir, Tunisia

²Department of Electrical Engineering, National Engineering School of Monastir, University of Monastir, Tunisia

Abstract

This paper outlines efficient hardware architecture of detection of bacteria and alga in microscopic images, using Xilinx System Generator (XSG). XSG is a high-level design tool based on blocks. It gives bit and cycle accurate simulation. The approach of detection used is the Hough transform. The latter is a very efficient approach of location of parametric curves in an image, especially lines. System was implemented on Virtex-V FPGA. To demonstrate the quality of the system, some experiments on microscopic images are given.

Keywords: FPGA; Microscopic images; XSG; Hough transform

Introduction

In the last few years, there has been an increasing tendency to implement Digital Signal Processor (DSP). DSP functions in Field Programmable Gate Arrays (FPGAs). FPGAs provide an attractive approach in comparison with conventional devices. They offer a very interesting solution that balance high flexibility, cost, time-to-market and performance [1,2]. Although Application-Specific Integrated Circuit (ASICs) and DSP chips have been the traditional solution for high performance applications, now the new technologies are imposing new rules. On one side, high development costs and time-to-market factors associated with ASICs may be prohibitive for certain applications, and on the other side, programmable DSP processors may be unable to achieve a desired performance because of their sequential execution architecture.

In this aim, the hardware level implementation is utilized because we need to process the image in real time. The hardware implementation offers also parallelism and it reduces the processing time. Our algorithm is encoded using XSG, which has been the tool of many researchers in several areas [3,4]. It is a graphical interface added to Matlab Simulink. Its library has a collection of many hardware blocks that can achieve complex functions.

In the digital image processing area, there are several methods of detection of objects in images [5,6]. Each method has different issues and different characteristics. Among them, the Hough transforms is very accurate in detecting parametric curves in an image. It is used in many researches, such as traffic and medical applications [7,8].

In the field of medical image processing, many studies have been carried such as the segmentation, classification and recognition of MRI and microscopic images [9,10]. In this case, our work focuses on the implementation of a real-time system of detection of linear alga and bacteria in microscopic images, using the Hough transform. The platform used is the Virtex-V FPGA. Our method is evaluated and improved by experimental results.

This paper is organized as follows. In Section 1, we present the XSG: its features, strengths. Section 2 addresses the overview of our system of detection. Section 3 gives the details of the hardware architecture. Experimental results are given in Section 4 and section 5 summarizes this paper.

Xilinx System Generator (Section 1)

Design flow

The proposed architecture is developed by the integrated design tool XSG dedicated FPGA. Unlike the standard HDL language, XSG provides an interface based upon a pattern using an extended library of construction blocks [11]. It takes the abstraction level slightly higher. It uses the Mathworks Simulink with a list of specific Xilinx building blocks, which can be used to create designs optimized for Xilinx FPGAs [4]. The Matlab workspace can be used during simulation.

Once the model is designed with XSG, the VHDL or Verilog code may be automatically generated. Each HDL file is related to an XSG building block. After that, the logic synthesis and routing steps should be performed [4,12]. Otherwise, simulation and testbench are not complicated by the generation of test vectors. The upload of this architecture on the FPGA will be possible using Xilinx Chip-Scope Pro tool [12]. The design flow of the XGS development tool is given in figure 1.

Loading digital data on XGS

Unlike software processing, XSG manipulates images as one-dimensional vectors in place of 2-d matrix [13]. By this conversion, the (x, y) pixel in the (m, n) original image will have the $(y-1)*n + x$ position in the output vector. After this, the image is stored in an input ROM to be ready to handle. The image conversion is shown in figure 2.

Overview of our System (Section 2)

As shown in figure 3, the input image must receive some pre-processing stages. It is first converted to grayscale, then filtered and finally thresholded. The filter used is a simple approximation of a discrete derivation, and after the thresholding step, we obtain a binary image in which the bacterium appears clearly.

***Corresponding author:** Fayçal Hamdaoui, Laboratory EμE, Faculty of Sciences of Monastir, University of Monastir, Tunisia, E-mail: faycel_hamdaoui@yahoo.fr

Received September 07, 2012; **Accepted** October 23, 2012; **Published** October 26, 2012

Citation: Ladgham A, Hamdaoui F, Sakly A, Mtibaa A (2012) Real Time Implementation of Detection of Bacteria in Microscopic Images Using System Generator. J Biosens Bioelectron 3:127. doi:10.4172/2155-6210.1000127

Copyright: © 2012 Ladgham, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

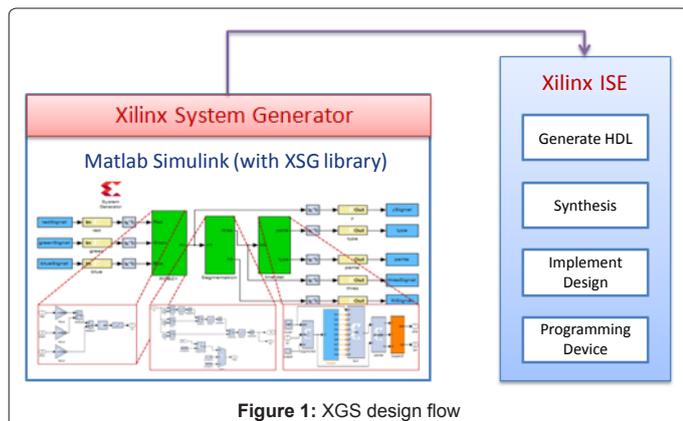


Figure 1: XGS design flow

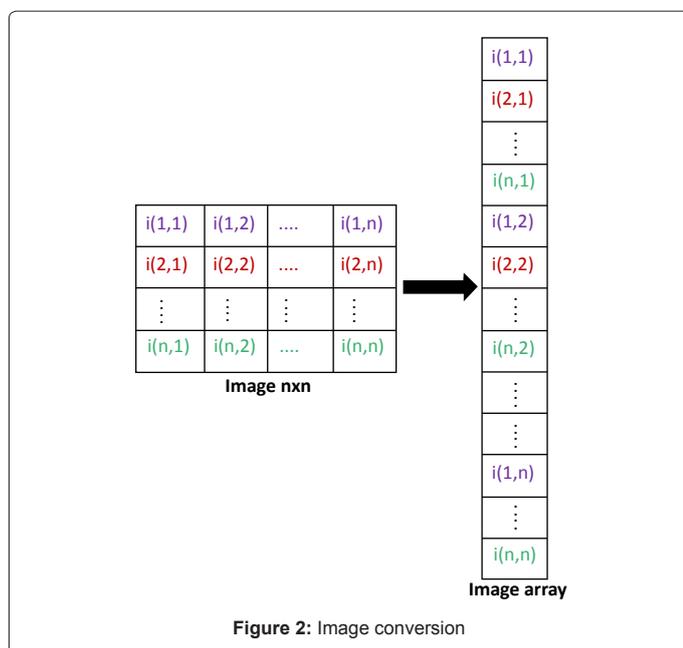


Figure 2: Image conversion

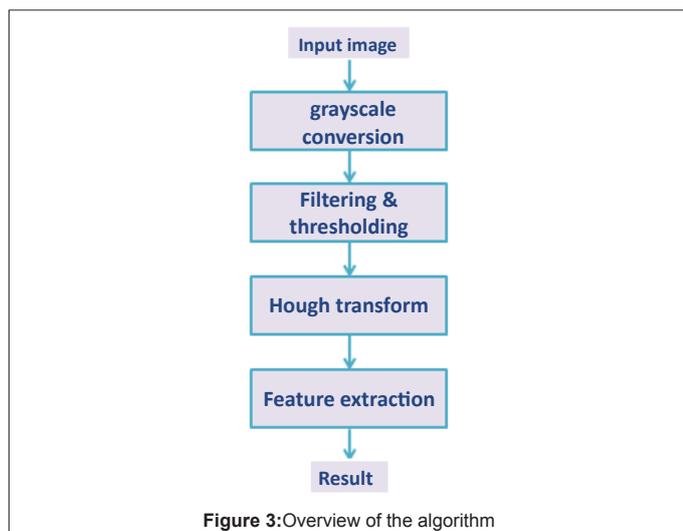


Figure 3: Overview of the algorithm

After applying the pre-processing steps, we applied the Hough transform to the binary image. This transformation is to locate the

linear bacterium. Finally, we attain the features extraction step (slope and length of bacterium).

Architecture of Bacterium Detection Using XSG (Section 3)

The architecture of the linear bacterium detection is shown in figure 4. Data are acquired and transformed to 1-dimensionnal vectors. The blocks of preprocessing are given in the same order listed previously in section 3. The result of each pre-processing step (grayscale conversion, filtering, thresholding) is sent to the Workspace of Matlab. These results are reconverted to 2-d matrix in order to display it as an image. The length and the slope of the linear bacterium are displayed on XSG window.

Grayscale conversion

Images used are in the RGB space. They have three color components R, G and B. To facilitate their manipulation, searchers always try to have one component image that models the original image. Rahatabad et al. [14] used a ratio between the intensities of specific RGB components and the sum of their intensities. This report is used to detect bright colors in images.

In our research, microscopic images are converted to grayscale. We begin first by storing the color components of the input image. The grayscale conversion is the determination of the luminance Y of the YCrCb space. Y is the module that adds the components R, G and B. Each component is assigned by a coefficient using the following formula:

$$Y = 0,3008R + 0,5898G + 0,1094B$$

The hardware architecture of grayscale conversion is shown in figure 5.

Filtering and thresholding

We started by a filtering step to reduce noise and small details. It is a smoothing step, in which we use the first derivative [15] of the image I, as shown in the following formula:

$$\frac{\Delta I}{\Delta x} = \frac{I(x + \Delta x) - I(x)}{\Delta x}$$

For the simple approximation of the discrete derivative, we convolved the image with the following mask:

$$\begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$$

The hardware architecture of filtering is shown in figure 6.

The thresholding operation transforms the filtered image f to a binary one b. Pixels will be arranged according to their superiority at a chosen threshold T, as shown as followed:

$$b(x, y) = \begin{cases} 1 & \text{if } f(x) > T \\ 0 & \text{otherwise} \end{cases}$$

Figure 7 shows the hardware architecture of the thresholding implementation. The desired threshold is given by the input parameter 'Constant2'.

Hough transform

Principle: The principle of the Hough transform is to associate to each bright pixel in the Cartesian space (x, y) a sine wave in the Polar space (r, θ). The advantage of using the polar space is in alignment

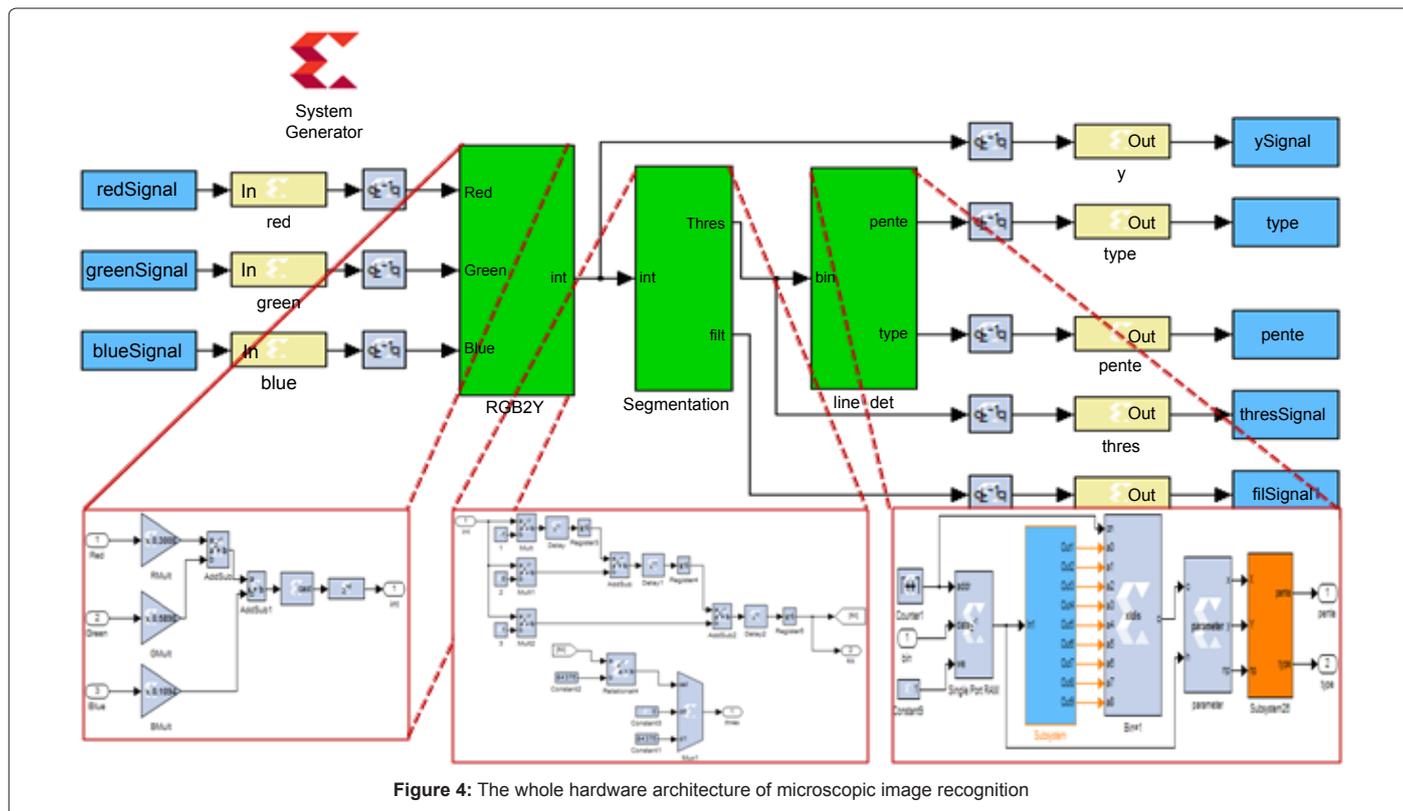


Figure 4: The whole hardware architecture of microscopic image recognition

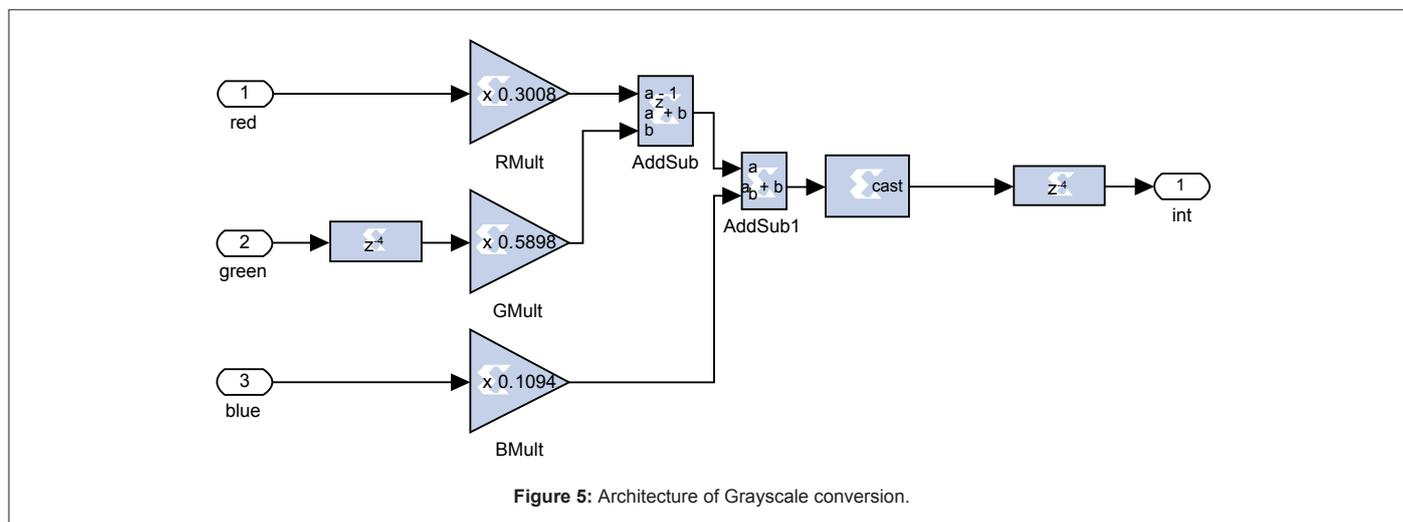


Figure 5: Architecture of Grayscale conversion.

detection [7]. As shown in figure 8, sinusoids obtained from points belonging to the same line will intersect at the point with the coordinates (r, θ) . These coordinates are the parameters of the line.

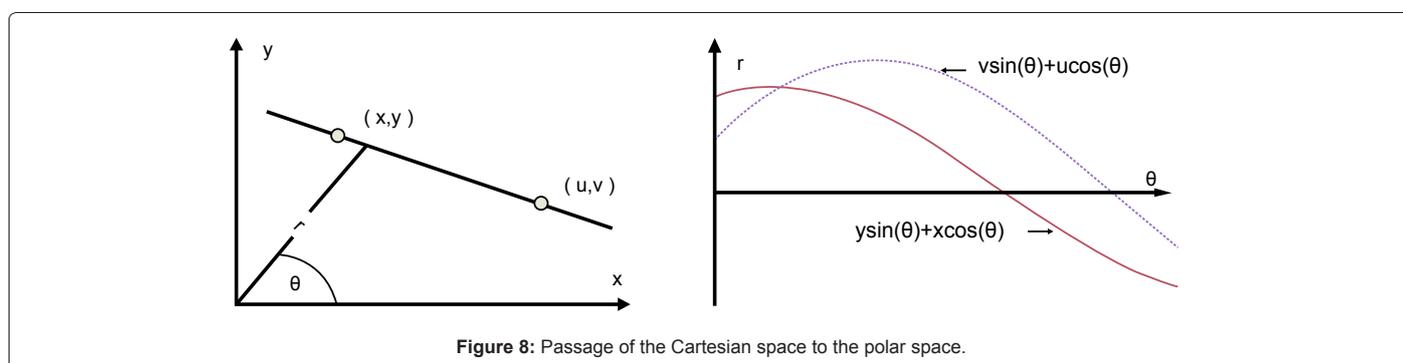
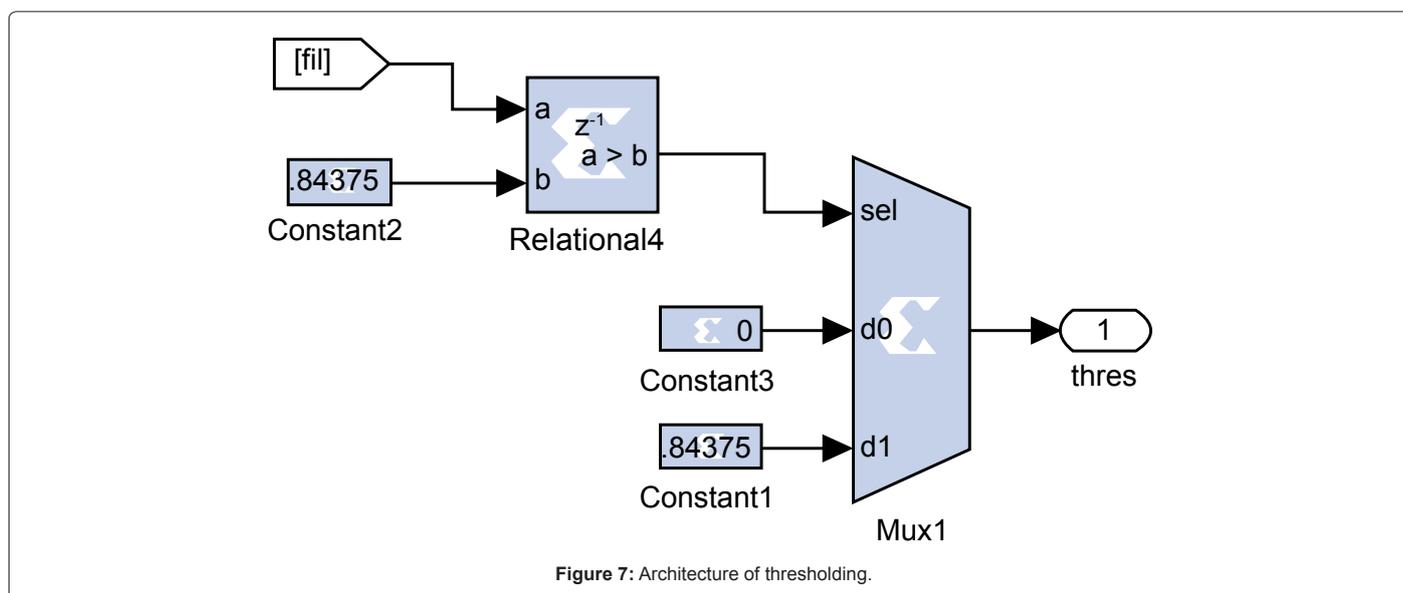
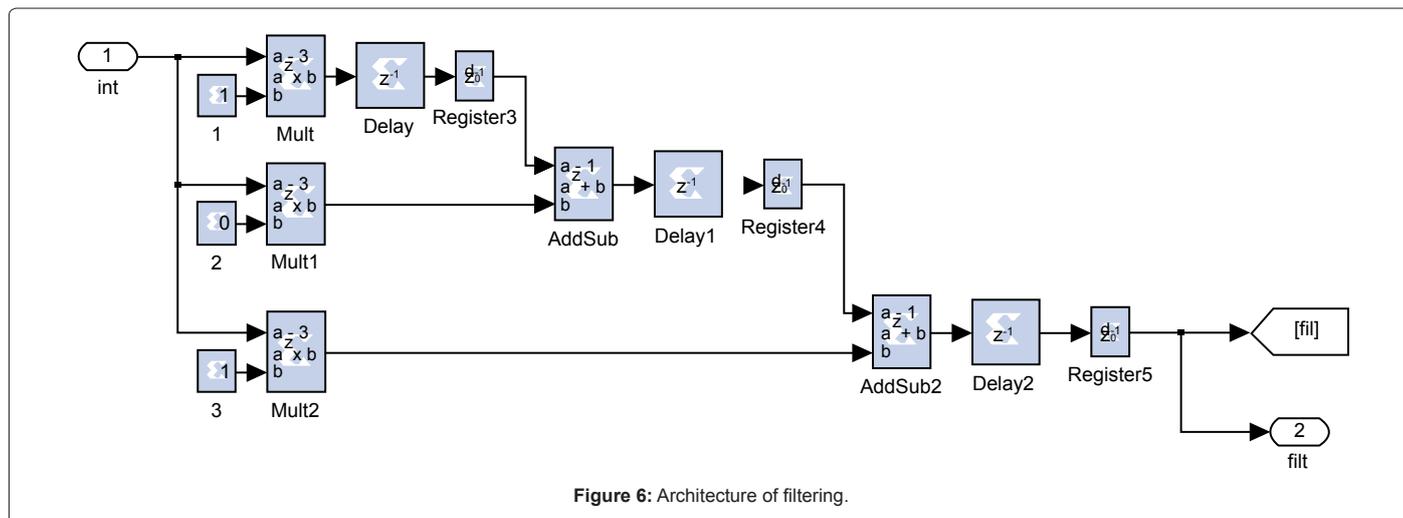
Architecture of hough transform: The whole architecture of Hough transform algorithm and of looking of linear bacterium parameters is given in figure 9. The principle is to use an accumulator with two dimensions in which each cell corresponds to a quantized value of (r, θ) . Before calculating the polar coordinates for each pixel and its neighborhoods, we search if there's enough evidence of an edge. If so, we save its coordinates (x, y) . Then, we calculate its radius r for several angles using the polar equation (1). And we increment the corresponding cells in the accumulator. The peaks of the accumulator are

the line parameters. Figure 10 show the architecture of test of neighbors and figure 11 shows the implementation of the polar equation.

$$r = x * \cos\theta + y * \sin\theta; 0 < \theta < 180^\circ \quad (1)$$

Experimental Results (Section 4)

The hardware architecture of detection of linear bacterium in microscopic images consists on the association of the blocks presented above. Firstly, we start by the acquisition of images from the compact flash to the FPGA registers. Then, images are converted from the RGB space to the Grayscale. Afterwards, filtering and thresholding stages are done. We end with the Hough transform application. Figure 12 shows the complete RTL schematic of our architecture.



The architecture has been implemented on Virtex 5 (ML507 target, XC5VSXT platform). The number of FLIP-FLOPs of FPGA used is the most important parameter that provides the amount of the used resources. The second parameter is the amount of memory used. Our architecture uses only 1% of the FPGA logic (flip flop) and 4% of the amount of memory, i.e. it uses 691 CLBs slices and 90 KB of memory.

In figure 13 and 14 below, we give some experimental results on 512 x 512 microscopic images. After Grayscale conversion and filtering, the linear bacteria appear clearly in the thresholded images given in figure 13 (d) and 14 (d).

To validate operatively our design, we use the ISIM simulator. Firstly, it is necessary to define the stimuli (the test vectors and the

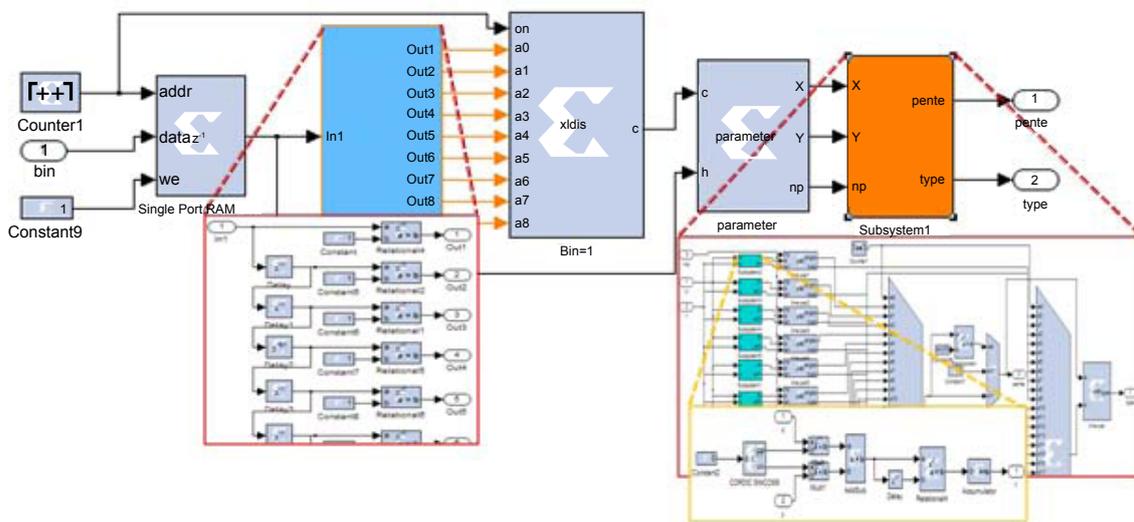


Figure 9: The whole architecture of Hough transform.

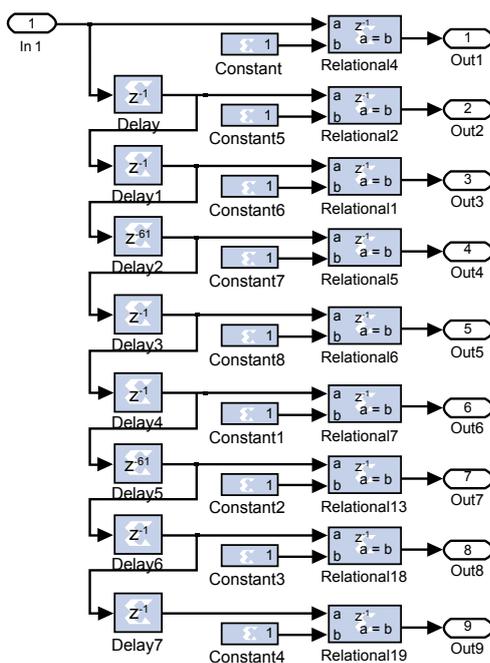


Figure 10: Architecture of test of neighbors.

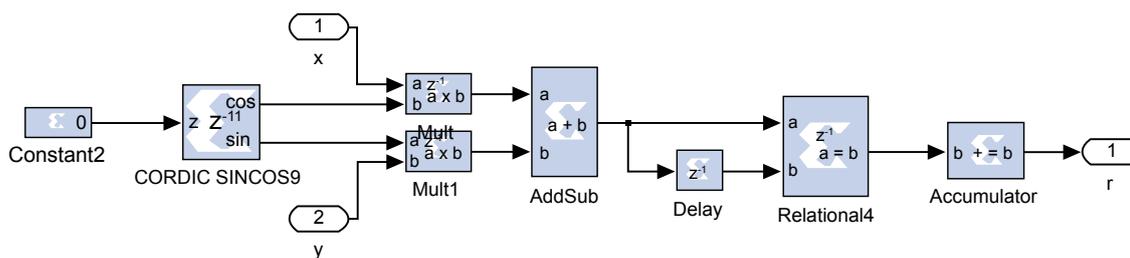


Figure 11: The implementation of the formula.

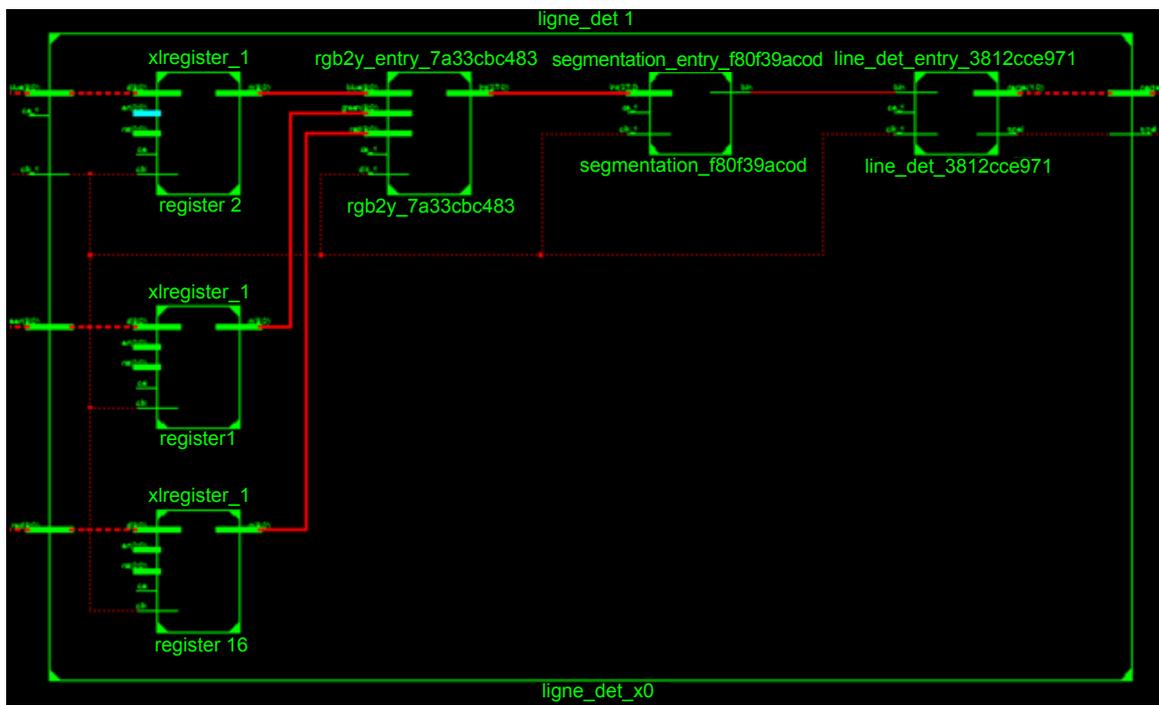


Figure 12: The internal component RTL blocks.

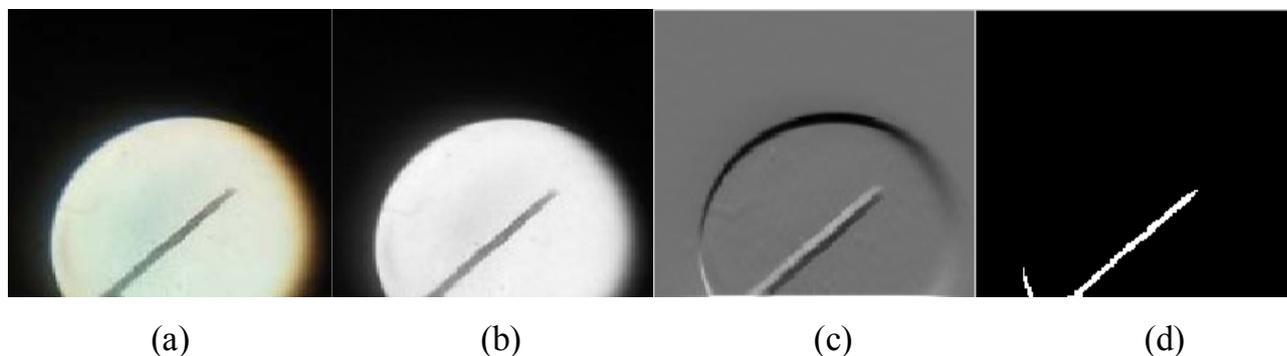


Figure 13: Experimental results, (a) original acquisition: microscopic bacterium, (b) Grayscale conversion, (c) filtering step, (d) thresholding step.

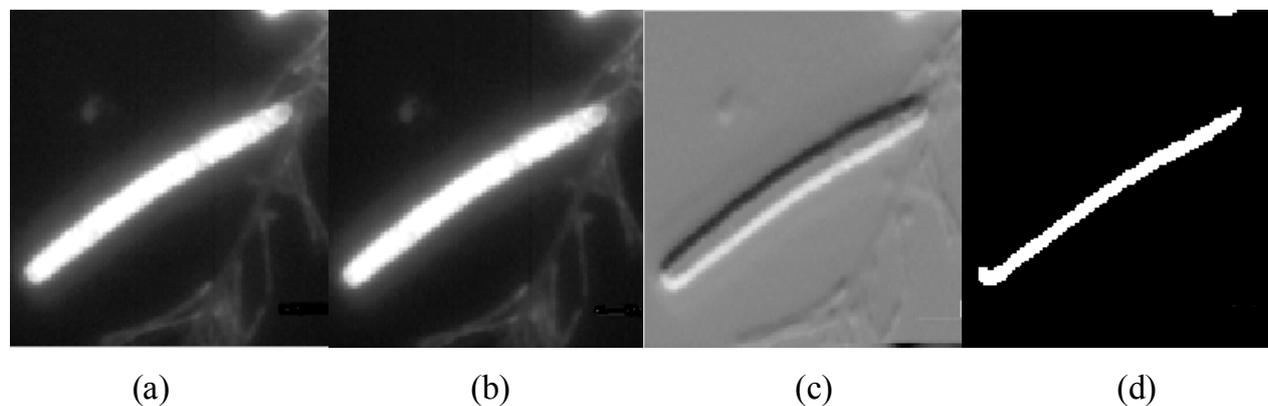


Figure 14: Experimental results, (a) original acquisition: filamentous bacterium, (b) Grayscale conversion, (c) filtering step, (d) thresholding step.

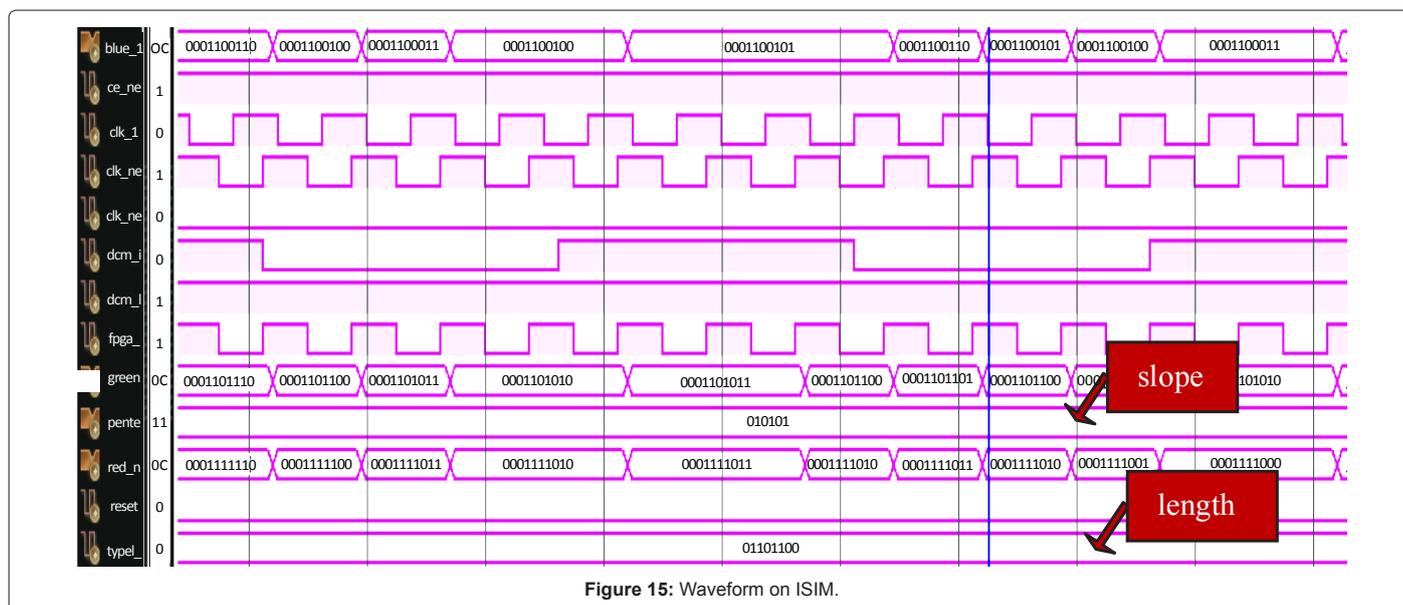


Figure 15: Waveform on ISIM.

frequency of the clock signal). Figure 15 below shows the waveform result made on the image of figure 13. This waveform shows the frames of the image, the slope of the detected bacterium and its length.

Conclusion (Section 5)

For summary, Xilinx System Generator is a really valuable tool for developing computer vision algorithms. It could be described as an advantageous option for development in a way more comfortable than that permitted by VHDL or Verilog Hardware Description Language. The aim of this paper is to prove the use of System Generator in designing a hardware system for image recognition. This concept is successfully implemented in the device Virtex 5 ML507 (XC5VSXT platform).

References

- Zemcik P (2002) Hardware acceleration of graphics and imaging algorithms using FPGAs. Proceedings of the 18th Spring Conference On Computer Graphics 25-32.
- Fons M, Fons F, Cantó E (2010) Fingerprint Image Processing Acceleration Through Run-Time Reconfigurable Hardware. IEEE Transactions on Circuits and Systems II: Express Briefs 57: 991-995.
- Moctezuma JC, Sanchez S, Alvarez R, Sanchez A (2008) Architecture for filtering images using Xilinx system generator. CEA'08 Proceedings of the 2nd WSEAS International Conference on Computer Engineering and Applications 284-289.
- Saidani T, Atri M, Dia D, Tourki R (2010) Using Xilinx System Generator for Real Time Hardware Co-simulation of Video Processing System. Electronic Engineering and Computing Technology 60: 227-236.
- Soetedjo A, Yamada K (2005) Traffic sign classification using ring partitioned method. IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences 88A: 2419-2426.
- Fleyeh H, Gilani SO, Dougherty M (2006) Road sign detection and recognition using Fuzzy ARTMAP: a case study swedish speed-limit signs. Proceedings of Artificial Intelligence and Soft Computing, Acta press.
- Hechri A, Hamdaoui F, Ladgham A, Mtibaa A (2011) Using Fuzzy Logic Path Tracking for an Autonomous Robot. International Review of Automatic Control, IRECO 4: 115-123.
- Xu X, Zhou Y, Cheng X, Song E, Li G (2011) Ultrasound intima-media segmentation using Hough transform and dual snake model. Computerized Medical Imaging and Graphics 36: 248-258.
- Lee ME, Kim SH, Cho WH, Park SY, Lim JS (2009) Segmentation of Brain MR Images using an Ant Colony Optimization Algorithm. IEEE International Conference on Bioinformatics and Bioengineering 9: 366-369.
- Wang Q, Niemi J, Tan CM, You L, West M (2010) Image Segmentation and dynamic lineage analysis in single-cell fluorescence microscopy. Cytometry A 77: 101-110.
- The MathWorks Inc. Embedded Matlab Language User Guide (2007).
- Xilinx System Generator User's Guide.
- Alba M, Sánchez G, Alvarez GR, Sánchez GS (2003) Architecture for filtering images using Xilinx System Generator. International Journal of Mathematics and Computers in Simulation.
- Rahatabad FN, Moradi MH, Nafisi VR (2007) A Multi Steps Algorithm for Sperm Segmentation in Microscopic Image. World Acad Sci Eng Technol 12: 516-518.
- Panda CS, Patnaik S (2009) Filtering Corrupted Image and Edge Detection in Restored Grayscale Image Using Derivative Filters. International Journal of Image Processing 3: 105-119.