EasyChair Preprint

№ 11609

Hardware Design for Medical Image Enhancement

Amel Baba Hamed, Hassane Bechar and Houaria Bendouma

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

December 23, 2023

Hardware Design for Medical Image Enhancement

Dr. BABA HAMED Amel

Department of Electrical Engineering, LAAS laboratory
National Polytechnic School of Oran Maurice Audin (ENPO-MA)
Oran, Algeria
amel.bensemain@enp-oran.dz

Dr. BECHAR Hassane

Department of Biomedical Engineering, Faculty of technology
University of Tlemcen
Tlemcen, Algeria
hassane.bechar@univ-tlemcen.dz

Dr. BENDOUMA Houaria

Department of Biomedical Engineering, Faculty of technology
University of Tlemcen
Tlemcen, Algeria
houaria.bendouma@univ-tlemcen.dz

***Abstract*—**In medicine, the images have become essential elements for establishing a good diagnostic. Therefore, the reconstruction and visualization of these medical images refers to a set of techniques, which make it possible to obtain information on the internal and external structure of an object against the background of an image. However, the large amount of data in medical image requires real-time processing. The aim of this article is to initiate a new research theme concerning the implementation of a few blocks for the reconstruction and visualization of medical images on a dedicated architecture FPGA (Field Programmable Gate array).

A high-level hardware description language VHDL (VHSIC: Hardware Description Language (Very High Speed Integrated Circuits)) describes the different blocks of the hardware architecture. For this, we used CAD tools (Computer-Aided Design).

Keywords—Medical Image; Binarization; Dynamic Inversion; Luminance; FPGA; VHDL; CAD tools.

I. Introduction

The beginning of medical imaging began with the discovery of X-rays in 1895 by the German Wilhelm Rõntgen [1]. In recent years, the acquisition techniques have become increasingly sophisticate: X-rays, Ultrasound and Nuclear Magnetic Resonance [2].

In many cases, the raw filmed image does not provide all the details necessary for proper interpretation or analysis. In other cases, it is useful to see only certain points or specific elements of an image while neglecting the others. To do this, the image must be retouches in various ways to meet our needs (this is image processing).

For most methods of these processing tools (spot, local and global operations), the number of operations to be performed depends on the size of the images. Some methods have a high algorithmic complexity, which has the consequence of increasing the execution times on the computer (sequential calculation) [3], [4]. In order to speed up these processes, FPGA (Field Programmable Gate array) reconfigurable circuits were chose because they offer real-time calculation (parallel calculation) [5], [6]. The materialization of these operations opens another aspect in the reconstruction, visualization and analysis of medical images because it allows medical interpretation in real time [7], [8], [9].

The notion of real-time processing is define that information must be processed at the same rate as its acquisition. In this context, this article proposes a hardware implementation of some point operations tools, allowing reconstruction, transformation and visualization of medical images in real time.

This article is organize as follows. Section 2 presents a study on selected spot operations for the treatment of medical images. Section 3 is the essence of this work, as it includes the proposed hardware architecture, in order to establish the reconstruction, transformation and visualization of medical images in real time.

The fourth section is devoted to the study of performance results and the synthesis of implementation of these point operations on the reconfigurable FPGA circuit.

Finally, a conclusion given to summarize the work done in this project and provide new guidance for future work.

II. Proposed method

The purpose of processing an image is to extract the information it contains to solve a problem by eliminating unnecessary information. Processing takes place in several steps:

- Processing to transform the image (binarization, dynamic inversion ...).
- Preprocessing to improve image quality (Filtering...).
- Image decomposition to extract characteristics (Contour, segmentation).

Therefore, the equipment used should be improve to reduce the delays of reconstruction, transformation, processing and display of images. To do this, we propose the use of FPGA circuits “Fig. 1”, which are increasingly used and offer embedded solutions in image processing applications. We propose to implement, in VHDL language (VHSIC: Hardware Description Language (Very High Speed Integrated Circuits)) and using the Vivado CAD tool (Computer-Aided Design), treatments to offer quick and effective solutions.

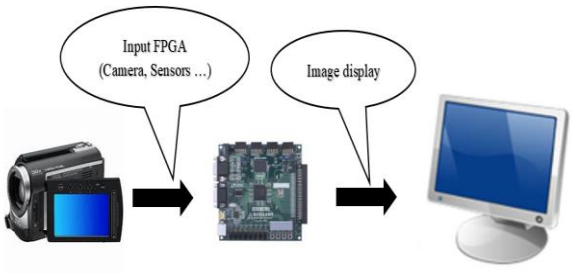

Fig.1. Embedded solutions in image processing applications.

For our work, we use the medical image presented in grayscale "Fig. 2", whose pixels are coded on 8 bits. The size of this image is 600 x 519 pixels.

Fig.2. Medical image selected for treatment.

A. *Simple thresholding*

Simple thresholding is to turn a grayscale image into a binary image (0 and 255). By taking a threshold value (S), the pixel value $I(x,y)$ will be calculated as follows:

$$\boldsymbol{if} \quad I(x,y) < S \quad \boldsymbol{then} \quad T(x,y) = 0 \qquad (1)$$

$$\boldsymbol{else} \quad T(x,y) = 255$$

With: $I(x,y)$ the input pixel values.

$T(x,y)$ output pixel values (processed pixels).

S the value of the threshold

B. *Dynamic Image inversion*

Dynamic inversion is an image transformation, which consists in reversing the black and white extremes. Sometimes we distinguish certain details better in a medical image in white on a black background than in black on a white background. the pixel value $I(x,y)$ will be calculated as follows:

$$T(x,y) = 255 - I(x,y) \qquad (2)$$

With: $I(x,y)$ the input pixel values.

$T(x,y)$ output pixel values (processed pixels).

C. *The luminance*

The luminance of an image is defined as the average of the pixels in the original image:

$$Average = \frac{1}{N.M} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} I(x,y) \qquad (3)$$

With: $I(x,y)$ the input pixel values.

N the number of rows.

M the number of columns.

Adding a constant value to an image causes its overall brightness to increase and subtracting a constant value from an image causes its overall brightness to decrease.

D. *Hardware Architecture*

The hardware architecture of the system "Fig. 3" consists of several blocks, allowing the reconstruction, transformation and visualization of our image:

- Block of the original image (Image original),
- Block that will perform a simple thresholding of our image (Image Binarization),
- Block to perform dynamic image inversion (Dynamic Image Inversion),
- Block to adjust the luminance in the image (Image Luminance).

The pixel values of the image (Paramet_In) arrive successively to the system, whose processing at the level of the four blocks is carried out in parallel. We used a multiplexer at the output of these blocks, which has two command switches, to allow us to display the desired image (Paramet_Out). Both switches are located on the development board containing the FPGA circuit.

The process is synchronized on a clock (Clk), whose cycle number equals the number of pixels in the image (the image size).

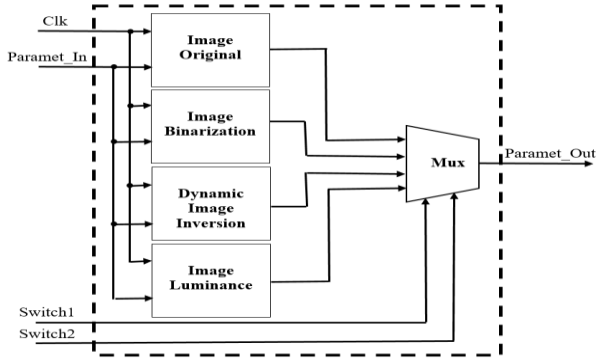

Fig.3. Hardware architecture.

III. Results and discussions

After the implementation of the hardware architecture under the Vivado environment, we perform an RTL analysis on the system file. The system model (design) will be display in a logical view “Fig. 4”.

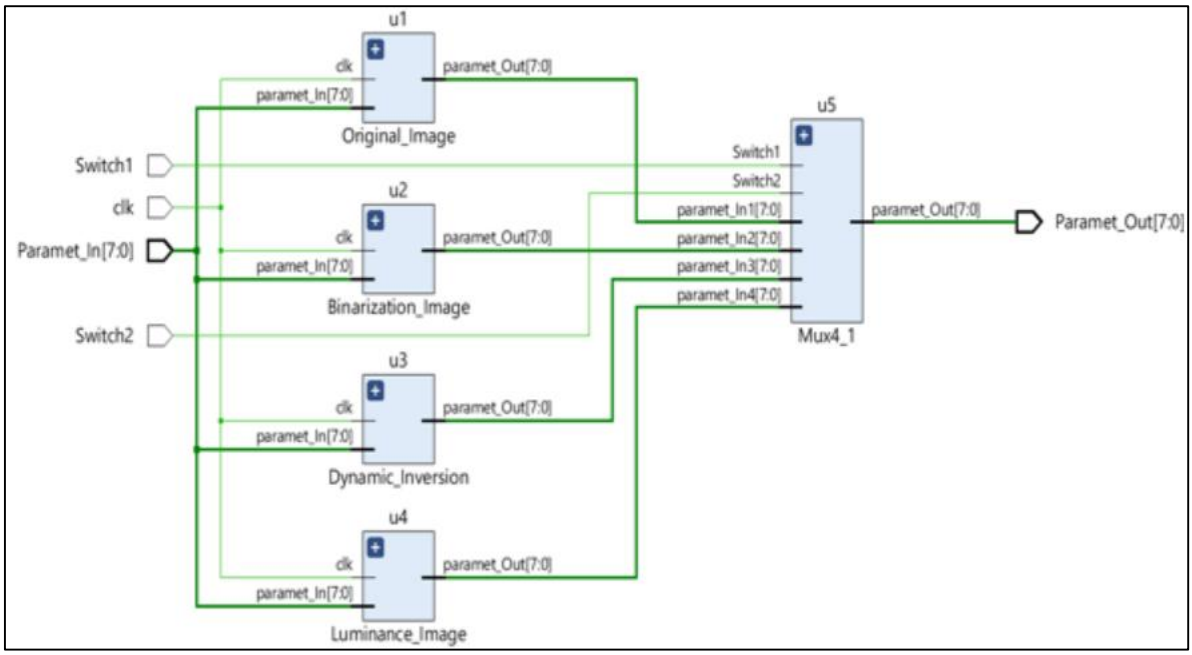

Fig.4. A logic view of the design

As we recorded the functional simulation (test bench) of each block of the system under the Vivado environment, to verify our calculations made on the input pixel values.

A. *Image original block*

This block allows the reconstruction and visualization of the original image “Fig. 14.a”, of which we use a register with parallel loading of 8 bits, which memorizes the values of the input pixels (Paramet_In) successively “Fig. 5”.

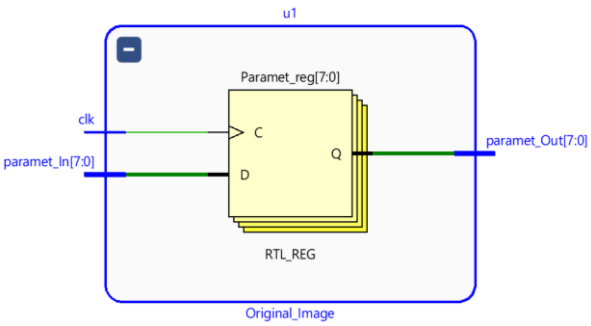

Fig.5. A logical view of the Original Image block.

The memorization of input pixel values at each clock cycle is show in the following figure “Fig. 6”:

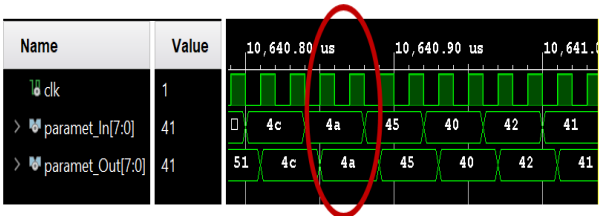

Fig.6. The test bench of the Image Original block.

B. *Image Binarization block*

By choosing the threshold value, we apply the mathematical function “(1)”, which allowed us to binarize the input image “Fig. 14.b”. For this, we use in this block, a register with parallel loading for the memorization of the values of the input pixels (Paramet_In) and a logical comparator, which makes it possible to make the comparison between Paramet_In and the chosen threshold “Fig. 7”.

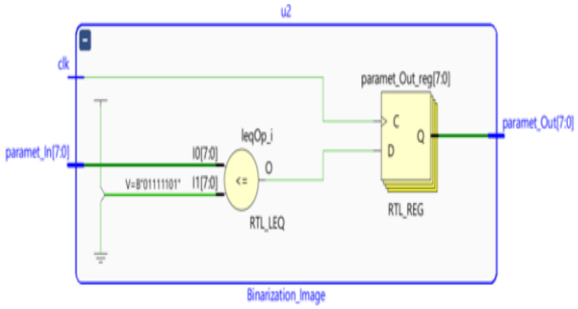

Fig.7. A logical view of the Image Binarization block.

The verification of calculations using functional simulation “Fig. 8”.

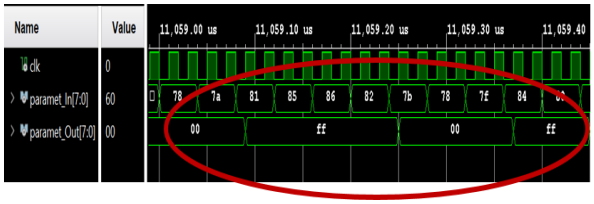

Fig.8. The test bench of the Image Binarization block.

C. *Dynamic Image Inversion block*

After applying the mathematical function “(2)” to all the pixel values of the image (Paramet_In), we reversed the dynamic of the input image “Fig. 14.c”. This block uses a parallel load register for storing input pixel values (Paramet_In) and a reverse logic gate for dynamic inversion “Fig. 9”.

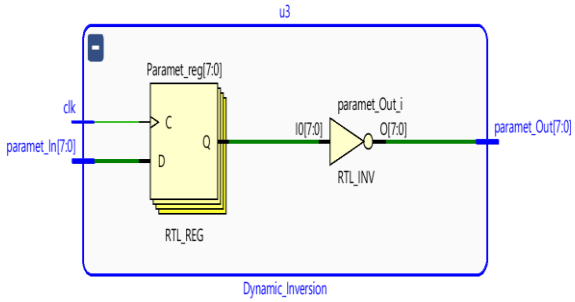

Fig.9. A logical view of the Dynamic Image Inversion block.

The inversion of the pixel values is check in the following figure “Fig. 10” after their memorization by the register.

Fig.10. The test bench of the Dynamic Image Inversion block.

D. *Image Luminance block*

Since the image is perfectly illuminated, we chose to adjust the luminance by subtracting from each input pixel value (Paramet_In) a constant (the choice of the constant is arbitrary), which caused an overall decrease in luminance on the input image “Fig. 14.d”. This block uses a parallel load register that stores input pixel values and an algorithm that performs a binary subtraction between Paramet_In and the chosen constant “Fig. 11”.

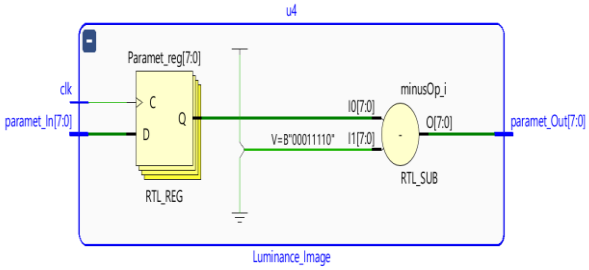

Fig.11. A logical view of the Image Luminance block.

The verification of calculations using functional simulation “Fig. 12”.

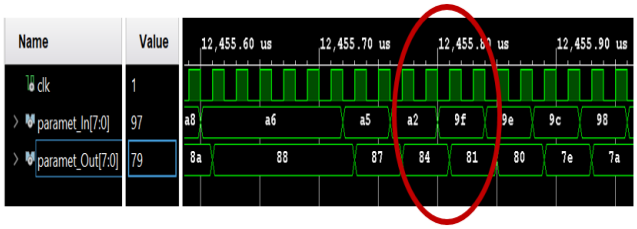

Fig.12. The test bench of the Image Luminance block.

The multiplexer used in this hardware design, for the choice of the processed image to be visualize is define by the following figure “Fig. 13”.

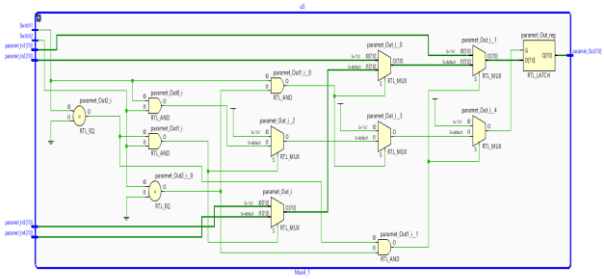

Fig.13. A logical view of the multiplexer.

The result of our treatment on the chosen medical image using the hardware design is present in the following figure “Fig. 14”.

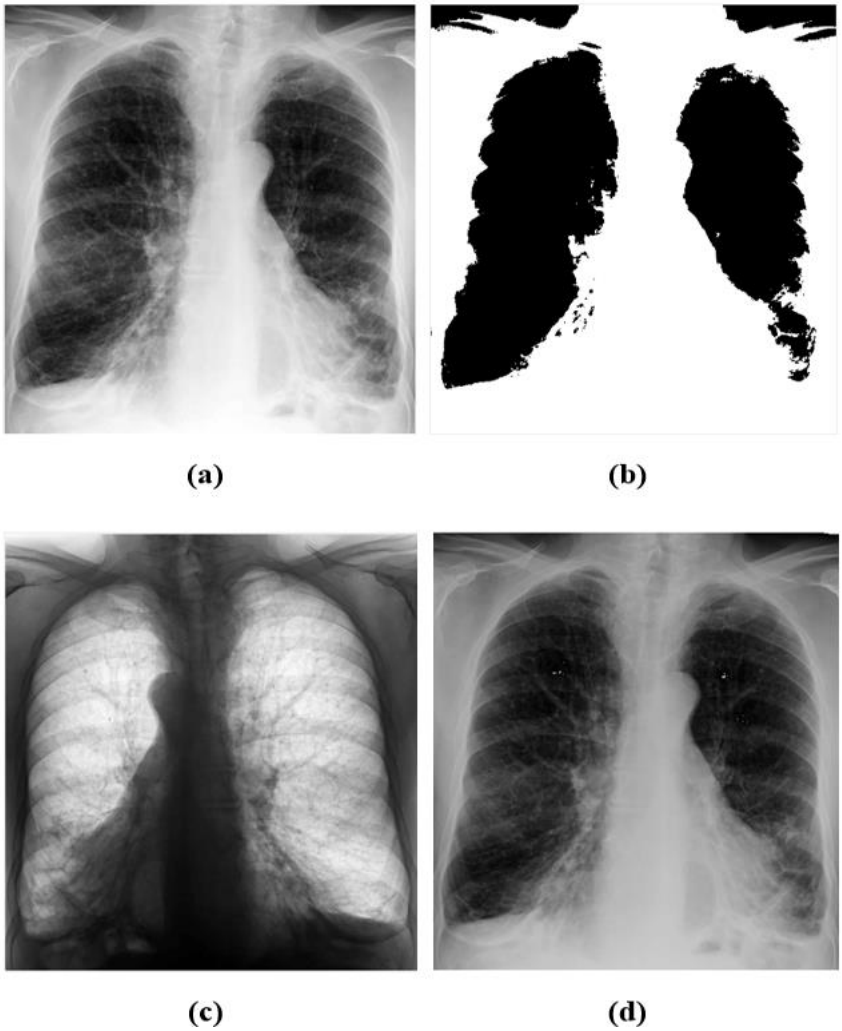

Fig.14. Images obtained after hardware implementation of the system. (a). Restitution of the original image. (b). Binarization of the original image. (c). Dynamic inversion of the original image. (d). Adjusting the luminance in the original image.

IV. Conclusion

Image processing is the new gateway for medical applications to help the physician make a decision. Our application consisted in implementing some medical image processing tools (reconstruction, transformation and visualization) so that the doctor's decision is make in real time.

Our work consisted in processing the medical image in real time (restitution, binarisation, dynamic inversion and luminance adjustment), whose material architecture of the proposed system realized these four transformations at the same time (parallel calculation); processing each pixel value of the image in a clock cycle (the clock cycle is 20 ns). So, the image processing time equal to the size of that image multiplied by the clock cycle.

Our future work will focus on:

- Make an automatic selection of the threshold for thresholding (for example, binarization).
- Realize hardware designs for other image processing tools (filtering, segmentation...).
- Use hardware classifiers for recognition of abnormalities in medical images.

References

[1] "X-rays and the history of their properties", "Les rayons X et l'histoire de leurs propriétés", https://enseignants.lumni.fr/fiche-media

[2] "Nuclear Magnetic Resonance (NMR) Theory and Solution NMR" - Physics LibreTexts, https://phys.libretexts.org/Courses/University_of_California_Davis

[3] Maïtine Bergounioux, "Some mathematical methods for image processing", "Quelques méthodes mathématiques pour le traitement

d'image", HAL Id: cel-00125868 https://cel.hal.science/cel-00125868v4 Submitted on 4 Jan 2009.

[4] Dr. J. Thirumararan, S. Shylaja, "Medical Image Processing – An Introduction", International Journal of Science and Research (IJSR), ISSN (Online): 2319-7064, November 2014.

[5] N.K. Ratha, A.K. Jain and D.T. Rover, "Convolution on Splash 2", Conference: FPGAs for Custom Computing Machines, 1995. Proceedings. IEEE Symposium on. May 1995

[6] B.A. Draper, J. Ross Beveridge, A. P. Willem Böhm, C. Ross, M. Chawathe, Implementing Image Applications on FPGAs, Conference: Pattern Recognition, Proceedings. 16 th International Conference on, Volume 3, 2002.

[7] M. Kraft, Adam Schmidt and Andrzej Kasinski, "HIGH-SPEED IMAGE FEATURE DETECTION USING FPGA IMPLEMENTATION OF FAST ALGORITHM", Conference: VISAPP 2008: Proceedings of the Third International Conference on Computer Vision Theory and Applications, Funchal, Madeira, Portugal, January 22-25, 2008 - Volume 1

[8] F. HAMDAOUI, A. KHALIFA and A. SAKLY, "Real Time Implementation of Medical Images Segmentation Based on PSO", 2013 International Conference on Control, Decision and Information Technologies (CoDIT), May 2013.

[9] H. Bendaoudi, "FLEXIBLE HARDWARE ARCHITECTURES FOR RETINAL IMAGE ANALYSIS", Thesis presented for the graduation of philosophize doctor (computer engineering) April 2017