

Digital Circuit Architecture for a Median Filter of Grayscale Images Based on Sorting Network

Victor Jimenez-Fernandez, Denisse Martinez-Navarrete, Carlos Ventura-Arizmendi,
Zulma Hernandez-Paxtian, Joel Ramirez-Rodriguez,

Abstract— In this paper a digital circuit architecture dedicated to median filtering of grayscale images is presented. The architecture emerges from a sorting network based median algorithm which effectiveness is verified by Matlab programming and its hardware implementation tested on a Spartan-3E FPGA device. The median pixel computation is approached by a sorting network scheme which is constituted by seven three-data comparator modules and hierarchically composed by twenty one switch/compare circuits. The successful operation of the three-data comparator module is demonstrated by transistor level SPICE simulations using 0.5 μ m CMOS technology.

Keywords— Digital architecture, Grayscale image, Median filter, Sorting network, Transistor level

I. INTRODUCTION

One of the filtering techniques, often used to reduce impulsive noise in acquired images, is the so-called median filtering[1]. This kind of filter leads to the best noise reduction of the image when it is corrupted with “salt and pepper” noise [2]. Its main advantage consists in diminishing the lossless of information due to the computed pixel values have correspondence to one of the already presented in the image [3]. However, it presents the inconvenient of requiring a sorting operation what implies a high computational cost. The challenge in this kind of filtering is the requirement of real-time processing [4].

In this sense, an ASIC realization appears like an alternative solution to achieve that goal. The results which are reported in this paper represent a first approach that is oriented to define, as a future perspective, a digital circuit architecture for an ASIC design.

The work is organized as follows: in section II the median filter algorithm used as a reference to define the proposed architecture is presented, section III deals with the description of the digital architecture, in section IV the three-data comparator circuit scheme is described, section V shows the transistor level design of the three-data comparator, in section VI the practical results are depicted and finally in section VII the conclusions are presented.

II. MEDIAN FILTER ALGORITHM

Let us consider a grayscale image I , divided in $(m \times n)$ pixels (squares) and affected by impulsive noise. In order to achieve the filtered image IF which is obtained through the median filtering operation, the value of each output pixel $IF(x,y)$ must be computed by using iteratively a set of 9 pixels from I . These pixels are inside of a 3×3 mask with center in $I(x,y)$. The position of this mask is shifted recursively a long to the entire image until the median filtering process be completed. From a purely numerical point of view, the image I is treated as a matrix array. In this sense, because of the mask operates over the pixel neighbor then it is required to add elements (zeros) around the matrix I , it gives as result the $(m+2) \times (n+2)$ matrix IR . This procedure is summarized in the algorithm described as follows:

Median Filter Algorithm

INPUT: I of $(m \times n)$ with $l = \{1, m\} \uparrow$, $p = \{1, n\} \uparrow$

OUTPUT: IF of $(m \times n)$

```

for  $l$  from 1 to  $m$  do
  for  $p$  from 1 to  $n$  do
     $\alpha = I[l, p]$ 
     $IR[l+1, p+1] = \alpha$ 
  end
end
for  $i$  from 2 to  $(mn)$  do
  for  $j$  from 2 to  $(mn)$  do
     $C1 = \text{sort}(IR[i-1, j-1], IR[i-1, j], IR[i-1, j+1])$ 
     $C2 = \text{sort}(IR[i, j-1], IR[i, j], IR[i, j+1])$ 
     $C3 = \text{sort}(IR[i+1, j-1], IR[i+1, j], IR[i+1, j+1])$ 
     $L1 = \text{sort}(C1[1], C2[1], C3[1])$ 
     $L2 = \text{sort}(C1[2], C2[2], C3[2])$ 
     $L3 = \text{sort}(C1[3], C2[3], C3[3])$ 
     $DIG = \text{sort}(L3[1], L2[2], L1[3])$ 
     $MED = DIG[2]$ 
     $IF(i-1, j-1) = MED$ 
  end
end
end

```

In addressing the elements of the image I , the l , and p indexes are incorporated. The notations: $l=\{1,m\}^\uparrow$, $p=\{1,n\}^\uparrow$ are used as a shorter expression than $l=\{1, 2,\dots, m\}$, and $p=\{1, 2,\dots, n\}$, respectively. Also i and j are indexes used in the IR and IF images. $C1, C2$ and $C3$ denote the sorted columns in the mask, $L1, L2$ and $L3$ the sorted rows, and DIG the sorted diagonal elements. MED is the median of the nine input data.

The median computation is performed by the mask at each one of the pixels of matrix IR . It is done in three steps: firstly, the pixels of the mask are sorted in a column by column sequence, then row by row and finally a long to the diagonal elements. After that sorting task is achieved, the central element (median) of the mask is picked out of IR and stored in the matrix IF (the filtered image). An illustrative description for the median algorithm is depicted in Fig. 1.

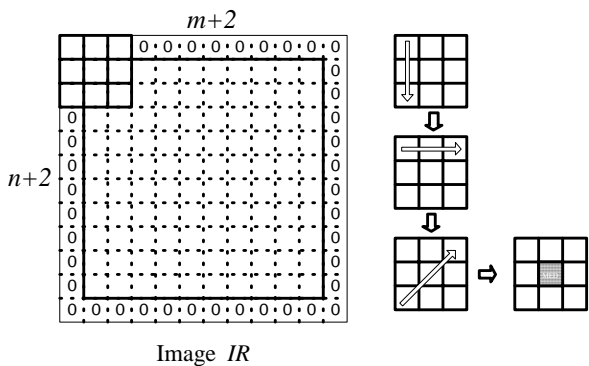


Fig 1: Graphical description for the median algorithm

III. DIGITAL ARCHITECTURE

A simplified scheme of the proposed median filter architecture can be observed in Fig.2. Six main blocks can be distinguished:

1. Serial-in/parallel-out eight-bits **Receiver**
2. Nine-data **Accumulator**
3. Nine-data **Sorting Network** module
4. Parallel-in/serial-out eight-bits **Transmitter**
5. **Baud Rate Generator** block
6. **Control Unit** module

Due to the sorting network needs to receive and transmit data in order to operate as a median filter, a transmitter and a receiver modules based on the RS-232 communication protocol are included.

1. **Receiver:** This block is a state machine with three input signals: CLK, CLKS, and RX. The CLK signal is the 50 MHz FPGA clock, CLKS is an output generated by the baud rate generator (at 9600 bps), and RX is the serial input. The receiver module provides two output signals: DOUT which is the signal that holds the eight-bit input data, and the LD flag which does not rise unless the entire eight-bit word is completely received. The LD signal also works as a clock for the next module.

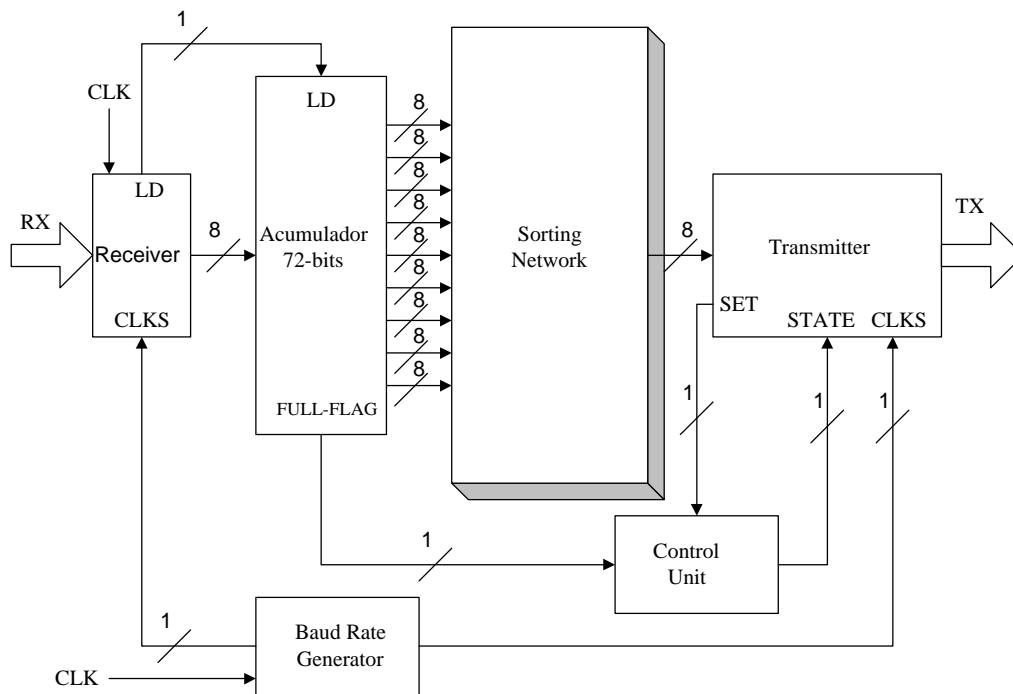


Fig. 2: Scheme of the median filter architecture

2. **Accumulator:** The accumulator is a register in which the received data are temporarily stored. It has one input signal that stores an eight-bit word at every LD pulse and ten output signals: FULL-FLAG and the others nine lines for data. Once the accumulator is full, the FULL-FLAG signal is enabled what releases all data and the capture process remains in stand-by mode until the next LD pulse.

3. **Sorting Network:** The sorting network block is a nine-inputs/one-output combinational module with a data word length of eight bits. It is in fact, the kernel of the median filter architecture and is constituted by an array of seven blocks of three-data comparator modules as illustrates Fig. 3.

The topology of the three-data comparator blocks interconnections is directly related to the median algorithm. As it can be seen, it is a process divided in three stages: the first one is the column sorter, the second is the row sorter and finally the last one is the diagonal sorter.

The connection scheme of Fig. 3 can also be seen in the format of the so-called sorting network diagram shown in Fig. 4.

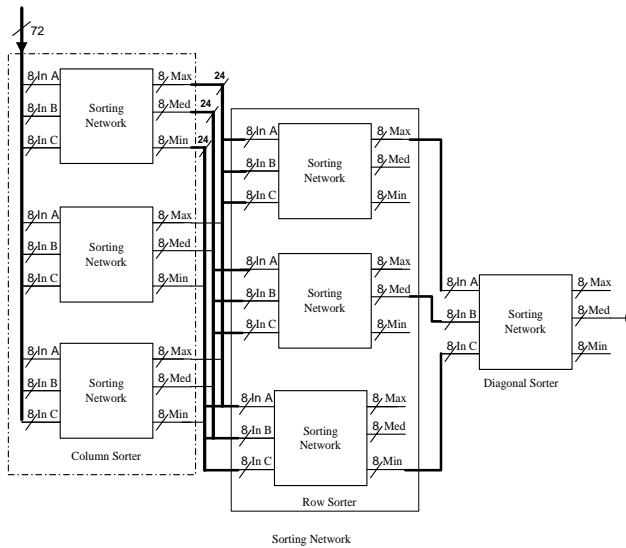


Fig 3: Median filter digital architecture block diagram

Further and detailed information about sorting networks can be found in [5]-[7].

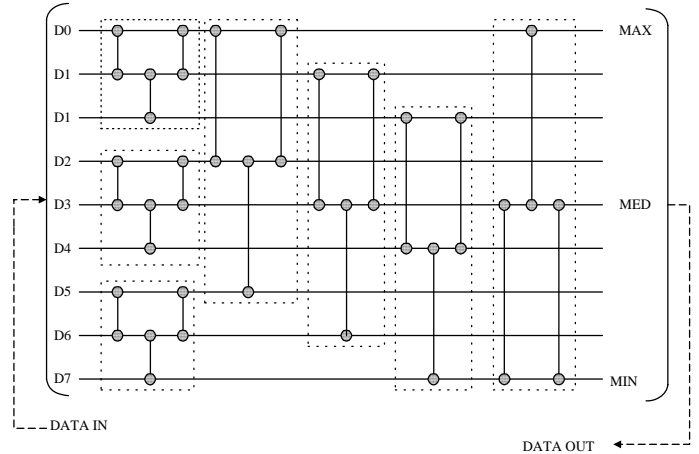


Fig 4: Sorting network used in the median filter architecture

4. **Transmitter:** The transmitter has two input signals: the one-bit CLKS clock and the eight-bits median datum provided from the sorting network. Also two outputs signals, here denoted as TX and SET, are included in this block. It operates in two possible states: "waiting" and "transmitting".

In the first state, this module only receives data and waits for the next state. In the second state this module starts to send the data received bit per bit through a shift register at every CLKS pulse. When the stop bit is reached, the transmitter sends a SET signal to the control unit.

5. **Baud Rate Generator:** It is a one-input/one-output sequential block. The CLK input is the FPGA physical clock which operation frequency is 50 MHz and the CLKS output emulates a slower clock timer at 9600 Hz. This block is a frequency divider that is implemented by a 0 to 5208 counter. When it reaches the 5208 top limit, one CLKS pulse is generated and starts again.

6. **Control Unit:** It has two inputs and one output signals. The inputs are denoted as SET and FULL-FLAG and the output as STATE. This block is a counter that starts in the "waiting" state. If control unit receives a pulse from the "FULL-FLAG" it switches to the "transmitting" state and when receives a pulse from "SET" returns to "waiting".

IV. THE THREE-DATA COMPARATOR MODULE STRUCTURE

A more detailed description about the three-data comparator module included in the sorting network of Fig. 3 is depicted in the block diagram of Fig. 5. Three eight-bit word-length inputs described as: In A, In B and In C can be identified. Also, three switch/compare blocks are internally connected what makes possible to collect always the median datum in the internal bus denoted as MED and the minimum and maximum data into the external buses.

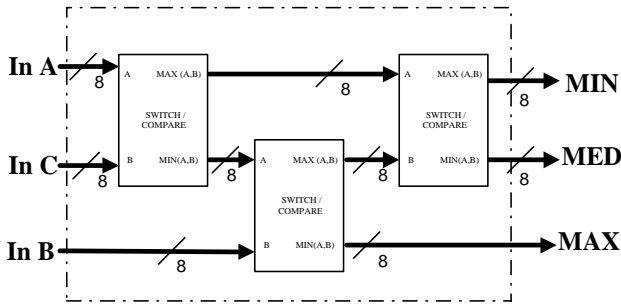


Fig. 5: Block diagram of the three-data comparator module

Each one of the switch/compare blocks operates as follows: firstly, it takes two eight-bit word-length input data, then both data are compared and finally two multiplexed outputs are enabled by directing the maximum-value datum at the upper path MAX(A,B) and the minimum at the lower path MIN(A,B). Fig. 6 illustrates the internal diagram of the switch/compare block. Note that it includes a magnitude detector composed by four full adders (FA) and two eight-bit multiplexor.

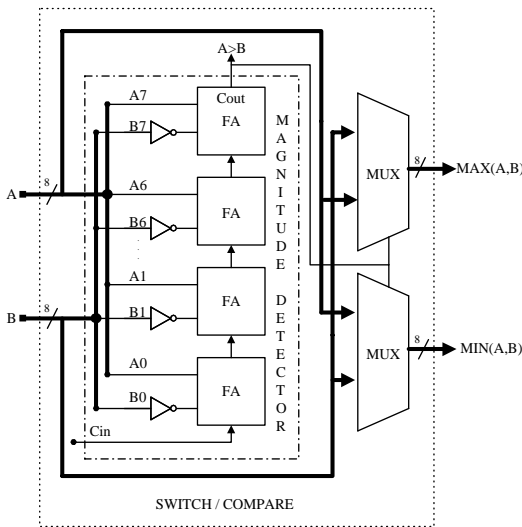


Fig. 6: Switch/compare block diagram

V. TRANSISTOR LEVEL DESIGN OF THE THREE-DATA COMPARATOR MODULE

Specifically, two basic standard circuit cells can be distinguished in the three-data comparator module: the one-bit full adder and the one-bit multiplexor circuits. Fig. 7 shows the schematic diagram, at transistor level, for the full adder used in the switch/compare design [8]. This circuit is composed by 24 MOS transistors topologically connected in a CMOS configuration where the 12 PMOS transistors (M1...M12) belong to the pull up network, and the 12 NMOS transistors (M13...M24) are associated to the pull down network.

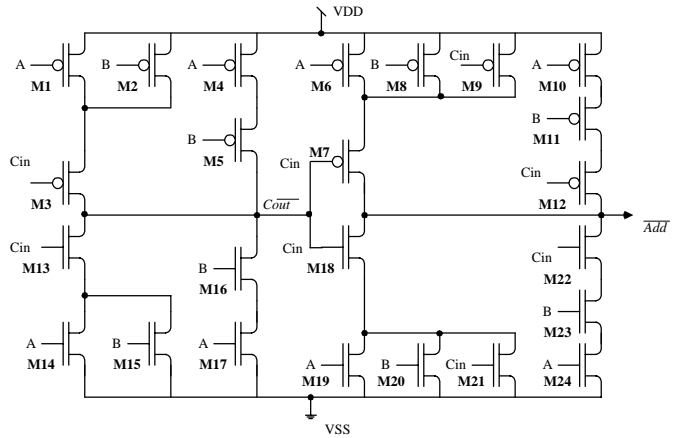


Fig. 7: Schematic diagram of the one-bit full adder

Since only one of these networks may be ON at a time, thus the \overline{Add} output will be electrically connected either to ground reference (VSS) or to voltage source (VDD) but not simultaneously. If both networks are ON, the electrical path from VDD to VSS will cause excessive current what of course must be avoided. Taking as a reference, the logical inputs A, B and carry in (Cin), and their possible outputs: \overline{Add} and \overline{Cout} , Table I summarizes such transistors which are ON at every logical combination.

Table I: MOS transistors in ON state

INPUTS			OUTPUTS		MOS TRANSISTORS IN ON-STATE
A	B	Cin	Add	Cout	
0	0	0	0	0	M1, M2, M3, M4, M5, M6, M7, M8, M9, M10, M11, M12
0	0	1	1	0	M1, M2, M4, M5, M6, M8, M10, M11, M13, M18, M21, M22
0	1	0	1	0	M1, M3, M4, M6, M9, M10, M12, M15, M16, M18, M20, M23
0	1	1	0	1	M1, M4, M6, M7, M10, M13, M15, M16, M20, M21, M22, M23
1	0	0	1	0	M2, M3, M5, M8, M9, M11, M12, M14, M17, M18, M19, M24
1	0	1	0	1	M2, M5, M7, M8, M11, M13, M14, M17, M19, M21, M22, M24
1	1	0	0	1	M3, M7, M9, M12, M14, M15, M16, M17, M19, M20, M23, M24
1	1	1	1	1	M13, M14, M15, M16, M17, M19, M20, M21, M22, M23, M24

VI. RESULTS

Fig. 8 shows the one-bit multiplexor used as a part of the switch/compare module depicted in Fig. 6 at transistor level. It is integrated by two transmission gates and one NOT-gate. Each transmission gate is composed for one PMOS transistor (M25, M27) and one NMOS transistor (M26, M28). It has two inputs (In0, In1), one selector (Sw) and one output (Out).

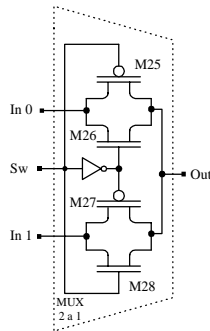


Fig. 8: One-bit multiplexor based on transmission gates

The one-bit multiplexor output depends only of the \overline{Cout} output from the one-bit full adder. \overline{Cout} is assigned to the selector that will activate one pair of transistors. If the selector is '0', the transmission gate at the top integrated by M25 and M26 switches ON, so In0 becomes the output, otherwise the transmission gate at the bottom composed by M27 and M28 is ON, hence In1 is the output..

A $0.5\mu\text{m}$ CMOS technology library and the S-Edit Tanner Tools Pro V15 software were used to perform the SPICE simulation at transistor level for the three-data comparator. The result shown in Fig. 9 corresponds to the four less significant bits of the output MED when the input data is given by In A=00000110, In B=00001110 and In C=00010100.

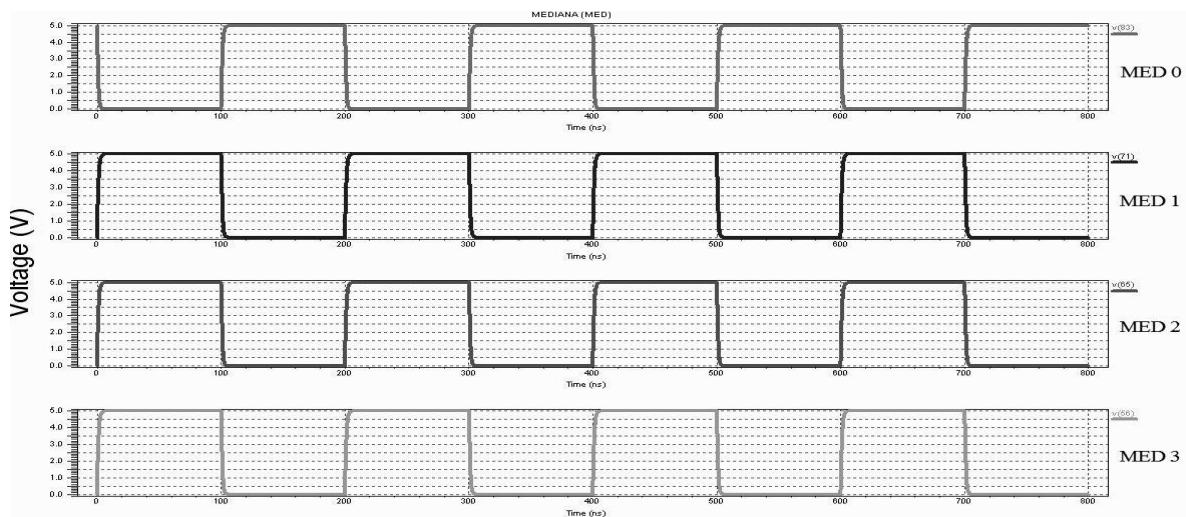


Fig. 9: Spice simulation result in $0.5\mu\text{m}$ CMOS technology

A usual strategy to verify most of digital architectures which are applied to real-time image processing is by using reconfigurable devices like FPGA (Field Programmable Gate Array) [9]-[11]. In order to verify the correct performance of the proposed architecture, results obtained from the FPGA implementation and from the coded algorithm in Matlab 7.3.0.267 are compared. As a reference, the (100×100) grayscale image I , which is depicted in Fig. 10, is considered. The experiment consists in adding impulsive noise, with a density of 0.4, to the reference image as shown in Fig. 11

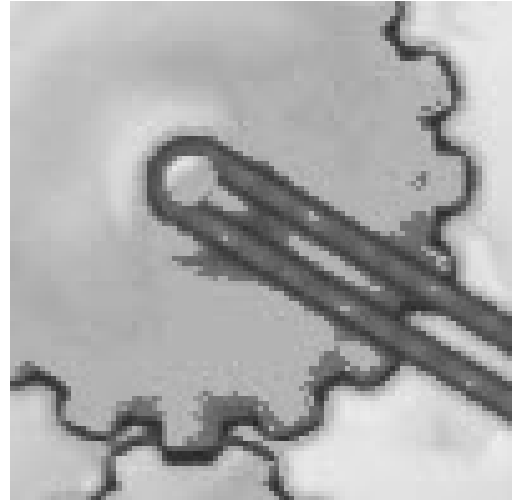


Fig. 10: The (100×100) grayscale reference image

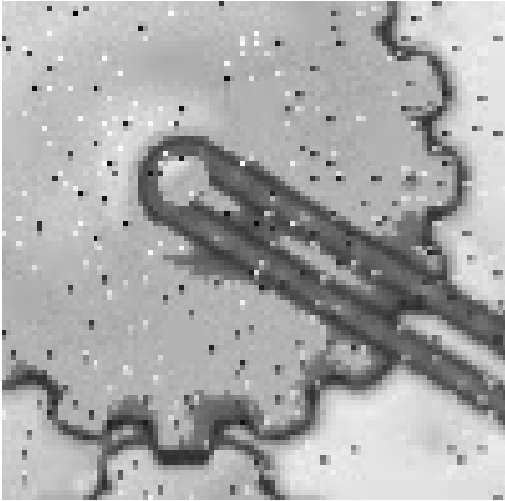


Fig. 11: The (100×100) grayscale corrupted by impulsive noise

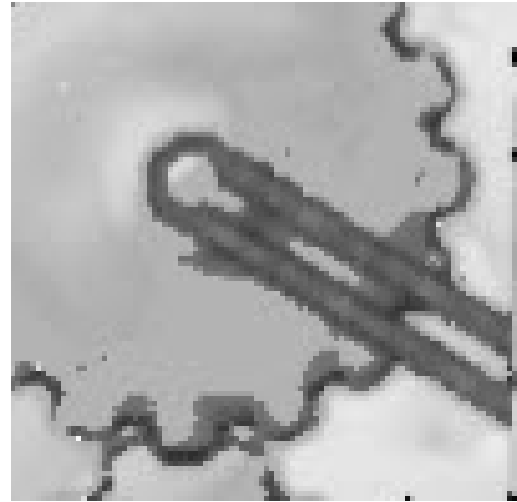


Fig. 13: Filtered image by hardware (FPGA)

The corrupted image is processed through both filtering process, by software with the median filtering algorithm in Matlab, and by the hardware FPGA implementation, respectively. Fig. 12 illustrates the filtering results obtained from software (Matlab).

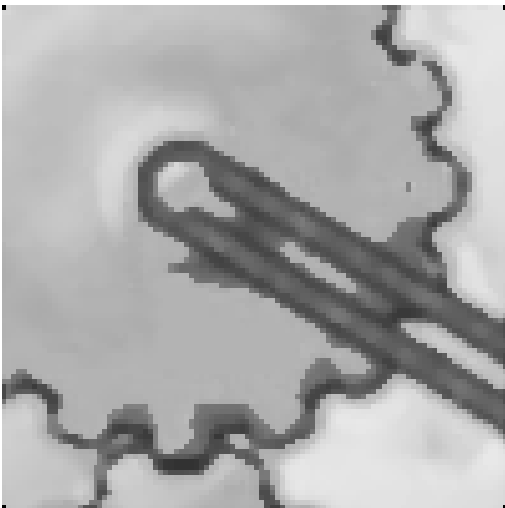


Fig. 12: Filtered image by software (Matlab)

Fig. 13 illustrates the filtering results obtained from hardware (FPGA).

The filtering effectiveness of the proposed circuit architecture is determined by the level of matching between the image that has been corrupted by impulsive noise and the filtered by FPGA. The reconstruction relative error (RRE) is a parameter often used to determine the filtering efficiency [12]. Equation (1) is applied to the reference image depicted in Fig.10 to compute the RRE in the FPGA processed image of Fig. 13.

$$RRE = \frac{\|g - f\|}{\|f\|} \quad (1)$$

The notation $\|g-f\|$ is used to describe the absolute value of the subtraction between the gray levels belonging to the reconstructed image (g) and the original image (f). It is important to note that the RRE is normalized according to the image intensity levels. Fig. 14 shows the relative error plot.

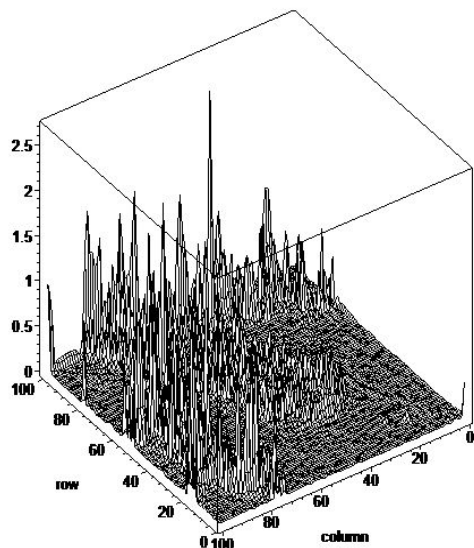


Fig. 14: Relative error plot of reference image and FPGA filtered image

VII. CONCLUSIONS

In this article a proposal of a digital circuit architecture for a median filter algorithm was presented. Its performance was tested by software in Matlab and by hardware through FPGA implementation. A nine-data sorting network was proposed as the kernel of the architecture. A high reduction in the error presented in the image filtered by FPGA with respect of the corrupted by impulsive noise could be appreciated in the RRE plot what confirms the effectiveness of the proposed architecture. The SPICE transistor level simulation, for the three-data comparator confirmed the successful operation of this circuit module.

ACKNOWLEDGMENT

This research is supported by the PROMEP/103.5/09/4482 project of Universidad Veracruzana, Mexico.

REFERENCES

- [1] A. Yang-Keun, P. Young-Choong, C. Kwang-Soon, J. Kwang-Mo and H. Ji-Man, "Boundary-Based Depth Image Improvement", *Proc.3rd WSEAS International Conference on VISUALIZATION, IMAGING and SIMULATION*, University of Algarve, Faro, Portugal, November 3-5, 2010, pp.64-69.
 - [2] P. Bojarczak, S. Osowski, "Denoising of Images – a Comparison of Different Filtering Approaches", *WSEAS Transactions on Computers*, vol. 3, pp. 738, 744, Austria, July, 2004.
 - [3] M. Faúndez Zanuy, "Digital voice and image processing with multimedia application", Ed. Alfaomega Marcombo, Barcelona, Spain, 2001, pp.125-126.
 - [4] M. Vega, J. Sánchez, J. Gómez, "An FPGA-based implementation for median filter meeting the real-time requirements of automated visual inspection systems", *Proc. 10th Mediterranean Conference on Control and Automation-MED2002*, Lisbon, Portugal, 2002.
 - [5] K. E. Batcher, "Sorting networks and their applications", *Proc. AFIPS Spring Joint Computer Conference*, vol.32, April 1962, pp. 187-195.
 - [6] C. Chakrabarti, "Sorting network based architectures for median filters", *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 40, issue 11, November 1993.
 - [7] P. Pirsch, "Architectures for digital signal processing", Ed. John Wiley & Sons, University of Hannover, Germany, 1998, pp. 198.
 - [8] N. H. Weste and K. Eshraghian, "Principles of CMOS VLSI design: A System Perspective", Ed. Addison Wesley, 1988, pp. 310-314.
 - [9] J. Márquez, "Image processing of biplane angiography using a Spartan-3E FPGA device", Thesis-Universidad de los Andes, Mérida, Perú, September 2008.
 - [10] P. Thitimajshima, "FPGA-Based Real Time Digital Image Processing System", *CSCC 2000, MCP 2000, MCME 2000 CD-ROM Proceedings*, Athens 2000, pp. 4091-4093.
 - [11] V. Tadigotla, S. Commuri, "Dynamic Image Filter Selection using Partially Reconfigurable FPGAs for Imaging Operations", *Proc. 5th WSEAS Int. Conf. on CIRCUITS, SYSTEMS, ELECTRONICS, CONTROL & SIGNAL PROCESSING*, Dallas, USA, November 1-3, 2006, pp. 60-65.
 - [12] C. Sing-Long, C. Tejos, P. Irarrazaval, "Reduction of the acquisition time in magnetic resonance imaging using compressed sensing", *Suplemento de Revista Chilena de Radiología*, Octubre, 2009, pp.10-16
- Victor Manuel Jimenez Fernandez** was born in Puebla, Mexico, in 1974. He received the B.S. degree in Electronics Engineering from the Instituto Tecnológico de Veracruz in 1998, the M.Sc. degree from the Universidad de las Américas-Puebla (UDLA) in 2000, and the Ph.D. degree from the Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE) Puebla, Mexico in 2006. From 2006 to 2007 he was a Post-Doctoral student in the Departamento de Ingeniería Eléctrica y de Computadoras at the Universidad Nacional del Sur, Bahía Blanca, Argentina. From 2008 to 2009, he was an assistant researcher at the INAOE-Mexico. He is currently an Assistant Professor in the Facultad de Instrumentación Electrónica y Ciencias Atmosféricas-Universidad Veracruzana, Xalapa, Veracruz, Mexico. His main research interests are integrated circuits design, analog filter synthesis and nonlinear circuit modeling.

Zulma Janet Hernandez Paxtian was born in Veracruz, Mexico, in 1976. She received the B.S. degree in Electronics Engineering from the Benemerita Universidad Autonoma de Puebla (BUAP) in 1998, the M.Sc. degree from the Universidad de las Americas-Puebla (UDLA) in 2000, and the Ph.D. degree in Physiological Sciences from the BUAP, Mexico in 2006. She is currently a Researcher-Profesor in the Universidad de la Cañada, Teotitlan de Flores Magon, Oaxaca, Mexico. Her main research interests are Bioelectronics, Digital Systems, Computer Simulation, Education in Engineering and Fuzzy logic.