ISSN XXXX XXXX © 2017 IJESC

Research Article Volume 7 Issue No.6

VHDL Implementation of Adaptive Median Filter

S.Ramesh Babu
Assistant Processor
Department of ECE
RVR&JC College of Engineering, India

Abstract:
This paper introduces, a new intelligent hardware module suitable for the computation of an adaptive median filter is presented for the first time. The function of the proposed circuit is to detect the existence of impulse noise in an image neighborhood and apply the operator of the median filter only when it is necessary. The noise detection procedure can be controlled so that a range of pixel values is considered as impulse noise. In this way, the blurring of the image in process is avoided, and the integrity of edge and detail information is preserved. Experimental results with real images demonstrate the improved performance. The proposed digital hardware structure is capable to process gray-scale images of 8-bit resolution and is fully pipelined, whereas parallel processing is used in order to minimize computational time. In the presented design, a 3x3 or 5x5 pixel image neighborhood can be selected for the computation of the filter output. However, the system can be easily expanded to accommodate windows of larger sizes. The proposed digital structure was designed, compiled and simulated using the Modelsim and Synthesized in Xilinx VERTEX-11. For the implementation of the system the PF10K200SRC240-1 field-programmable gate array device of the FLEX10KE device family is utilized, and it can be used in industrial imaging applications where fast processing is required. The typical clock frequency is 65 MHz.

Indexterms: Field-programmable gate arrays (FPGAs), impulse noise, median filter, real-time filtering.

I. INTRODUCTION

Two applications of great importance in the area of image processing are noise filtering and image enhancement. These tasks are an essential part of any image processor whether the final image is utilized for visual interpretation or for automatic analysis. The aim of noise filtering is to eliminate noise and its effects on the original image, while corrupting the image as little as possible. To this end, nonlinear techniques (like the median and, in general, order statistics filters) have been found to provide more satisfactory results in comparison to linear methods. For this reason, a number of non-linear filters, which utilize correlation among vectors using various distance measures. However, these approaches are typically implemented uniformly across an image, and they also tend to modify pixels that are undisturbed by noise, at the expense of blurred and distorted features. In this paper, an intelligent hardware structure of an adaptive median filter (AMF) suitable for impulse noise suppression for gray-scale images is presented for the first time. The function of the proposed circuit is to detect first the existence of noise in the image window and apply the corresponding median filter only when necessary. The noise detection level procedure can be controlled so that a range of pixel values (and not only the

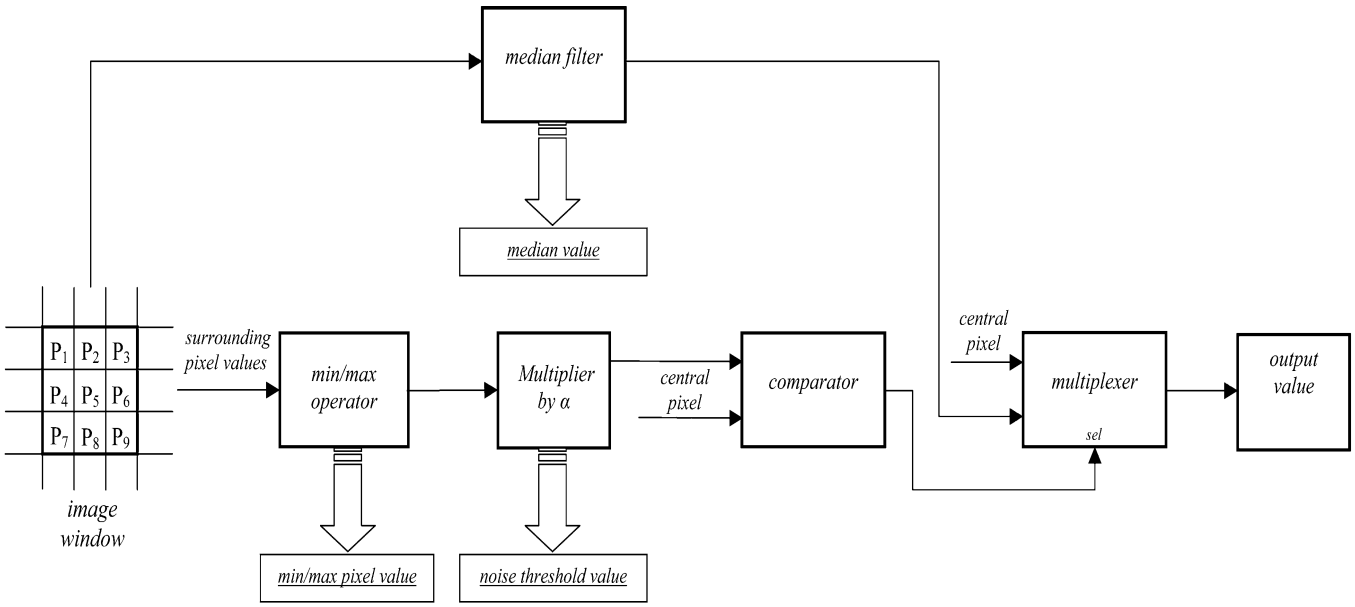


Figure.1. Block diagram of the adaptive filtering method

fixed values 0 and 255, but also salt-and-pepper noise) is considered as impulse noise. The main advantage of this adaptive approach is that the blurring of the image in process is avoided and the integrity of edge and detail information is

preserved. Moreover, the utilization of the median filter is done in a more efficient way. Experimental results demonstrate the improved performance of the AMF.

The proposed digital hardware structure is capable of processing gray-scale images of 8-bit resolution and performs both positive and negative impulse noise removal. A moving window of a 3x3 and 5x5 pixel image neighborhoods can be selected. Furthermore, the system is directly expandable to accommodate larger size gray-scale images. The architecture chosen is based on a sequence of four basic functional pipelined stages and parallel processing is used within each stage. The proposed structure was implemented using field-programmable gate arrays (FPGAs) , which offer an attractive combination of low-cost, high-performance, and apparent flexibility. The presented digital circuit was designed, compiled and simulated using the MAX + PLUS II Programmable Logic Development System by Altera Corporation. For the realization of the system the EPF10K200SRC240-1 FPGA device of the FLEX10KE device family, a device family suitable for designs that require high densities and high I/O count, is utilized. The total number of the system inputs and outputs are 44 and eight pins, respectively and the percentage of the logic cells utilized is (40 inputs for the input data and four in- puts for the clock and the control signals required) and the percentage of the logic cells utilized is 99%. The typical clock frequency is 65 MHz and the system can be used for real-time imaging applications where fast processing is of the utmost importance. As an example, the time required to perform filtering of a grayscale image of 260x244 pixels is approximately 7.6 ms.

Adaptive median filter design

The most common method used for impulse noise suppression for gray-scale and color images is the median filter. Impulse noise exists in many practical applications and can be generated by various sources, including many man-made phenomena such as unprotected switches, industrial machines, and car ignition systems. Images are often corrupted by impulse noise due to a noisy sensor or channel transmission errors. This type of noise is the classical salt-and-pepper noise for grayscale images. The output of a median filter at a point x of an image f depends on the values of the image points in the neighborhood of x.

This neighborhood is determined by a window W that is located at point x of f including n points $x_1, x_2, \ldots, x_n$ of f. the median can be determined when the number of points included in W is odd i.e., when $n = 2k+1$. the n values $f(x_1), f(x_2), \ldots, f(x_n)$ of the n points $x_1, x_2, \ldots, x_n$ are placed in ascending order forming the set of ordered values $\{f_1, f_2, \ldots, f_n\}$, in which $f_1 \leq f_2 \leq \ldots, \leq f_n$. The median is defined as he (k+1)th value of the set $\{f_1, f_2, \ldots, f_n\}$, med = f_{k+1}. The basic disadvantage of the application of the median filter is the blurring of the image in process. In the general case, the filter is applied uniformly across an image, modifying pixels that are not contaminated by noise. In this way, the pixel values of the input image are altered, leading thus to an overall degradation of the image and to blurred or distorted features. The proposed adaptive median filter can be utilized for impulse noise suppression for gray-scale images. Its function is to detect the existence of noise in the image window and apply the corresponding median filter only when necessary. The noise detection scheme for the case of positive (negative) noise is as follows.

1) For a neighborhood window W that is located at point x of the image f, the maximum (minimum) pixel value of the n-1 surrounding points of the neighborhood is computed, denoted as $f_{max}(x)$ ($f_{min}(x)$), excluding the value of the central pixel at point x.

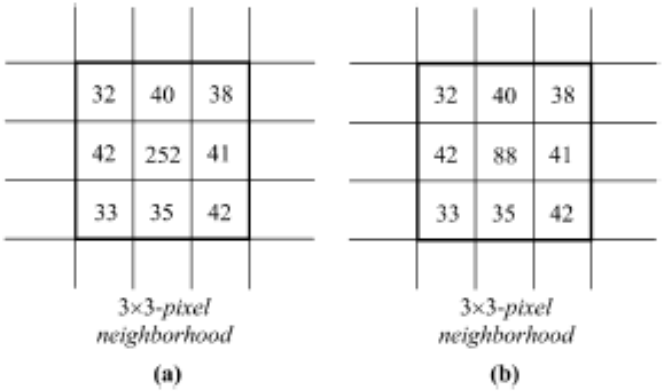


Figure.2. Noise detection algorithm (a) Impulse noise (b) Signal-dependent noise

2) The value $f_{max}(x)$ ($f_{min}(x)$) is multiplied by a parameter α which is a real number and an be modified. The result is the threshold value for the detection of a noise pixel, denoted as $f_{threshold}(x)$ and is limited to a positive (negative) integer threshold value.

3) The value of the central pixel is compared to $f_{threshold}(x)$, and the central pixel is considered to be noise, when its value is greater (less) than the threshold value $f_{threshold}(x)$.

4) When the central pixel is considered to be noise, it is substituted by the median value of the neighborhood, f_{k+1}, which is the normal operation of the median filter. In the opposite case, the value of the central pixel is not altered and the procedure is repeated for the next neighborhood window.

A block diagram of the adaptive filtering procedure previously described is depicted in Fig. 1. An example of the application of the noise detection algorithm for the cases of impulse and signal dependent noise is illustrated in Fig. 2(a) and (b), respectively.

In Fig. 2(a), a typical 3x3 pixel neighborhood window of a gray-scale image is depicted. The central pixel of the window occupies an extreme value (pixel value = 252) compared to the values of the surrounding image points (pixel values ranging from 32 to 42). For this reason, the central pixel is considered to be impulse noise and it should be eliminated. In the specific example, the noise detection scheme is applied as follows.

In the first step, we find the maximum value of the surrounding pixels, $f_{max}(x) = 42$. If we consider the parameter $\alpha = 5$, then threshold value $f_{threshold}(x) = f_{max}(x) * 5 = 210$. The central pixel value is $252 > f_{threshold}(x)$ and, therefore, is considered to be a noisy pixel. Finally, it is substituted by the median value of the neighborhood. The same discussion applies to the example of Fig. 2(b), which is the case of signal-dependent noise. The same procedure is followed, and the noisy pixel is successfully detected for a parameter value $\alpha = 2$.

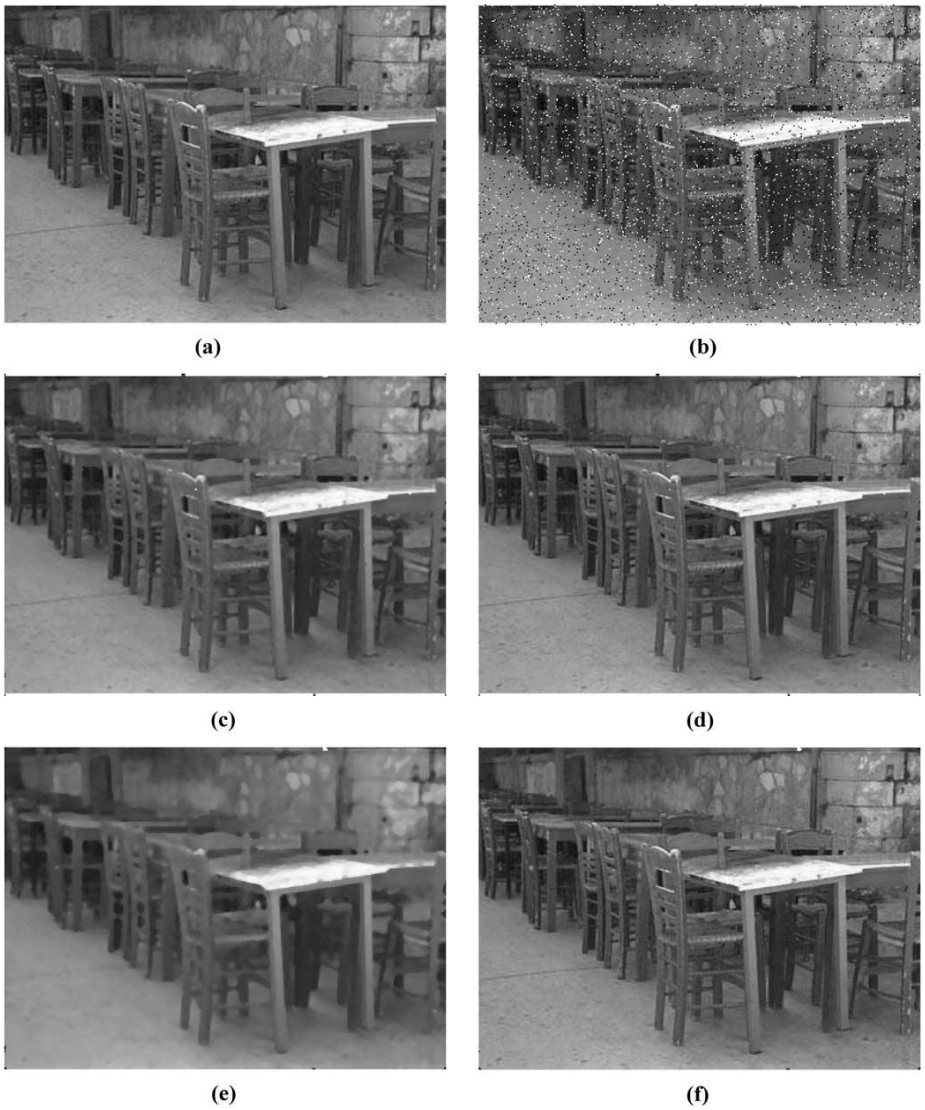


Figure.3. Impulse noise removal. (a) Original image "Café." (b) Image corrupted by 5% positive and negative impulse noise. (c) Median filter result using 3x3 pixel window. (d) AMF result using 3x3 pixel window. (e) Median filter result using 5x5 pixel window. (f) AMF result using 5x5 pixel window.

Two major remarks about the presented adaptive algorithm should be made. First, the value of the parameter α is of great importance, since it controls the operation of the circuit and the result of the overall procedure for different noise cases. Second, an appropriate positive and negative threshold value must be utilized for the case of impulse noise, when $f_{threshold}(x) \geq 255$. For example, in the case of Fig. 2(a), if we consider the parameter $\alpha = 8$, then $f_{threshold}(x) = f_{max}(x) * 8 = 336$ and the $f_{threshold}(x)$ is limited to the value 255, $f_{threshold}(x) = 255$. The central pixel value is $252 < f_{threshold}(x)$ and the central pixel is erroneously not considered to be impulse noise. An adjustable positive threshold value (for example 240) can be used as a limit of $f_{threshold}(x)$. In this way, $f_{threshold}(x) = 240$, whereas the central pixel value is $252 > f_{threshold}(x)$, and the central pixel is successfully detected as impulse noise. The meaning of this normalization procedure is that pixels occupying values between a range of the impulsive values (and not only pixels with values 0 and 255) should be considered as noisy pixels.

Hardware architecture

The proposed architecture is based on a sequence of pipeline stages in order to reduce computational time. Parallel processing has been employed to further accelerate the process. For the computation of the filter output, a 3x3 or 5x5 pixel image neighborhoods can be selected.

The structure of the adaptive filter comprises four basic functional units, the moving window unit, the median computation unit, the noise detection unit, and the output selection unit. The input data of the system are the gray-scale values of the pixels of the image neighborhood, the value of the parameter α, and the positive and negative threshold values. Additionally, two control signals required for the selection of the operation of the system (negative/positive noise suppression) and the neighborhood size (3x3 or 5x5) are also utilized.

Moving window unit

The pixel values of the input image, denoted as "IMAGE_INPUT [7..0]," are imported into this unit in serial. The value of the parameter is denoted as "MOD_VALUE[7..0]" and the positive and negative threshold values as POS/NEG THRESHOLD respectively. The parameter α is a real number, 5 and 3 bits are used for the representation of the integral and the fractional part, respectively. The "NEG/POS" control signal is used to determine the noise type. When "NEG/POS" is equal to "0" ("1") the circuit operation is negative (positive) noise suppression. For the moving window operation, a 3x3 (5x5) pixel serpentine type memory is used, consisting of 9 (25) registers, illustrated in Fig. 6. In this way, when the window is moved into the next image neighborhood, only 3 or 5 pixel

values stored in the memory are altered. The outputs of this unit are rows of pixel values (3 or 5, respectively), which are the inputs to the median computation unit.

Median computation unit

In this stage, the median value of the image neighborhood is computed in order to substitute the central pixel value, if necessary. In this unit, the min/max value of the neighborhood is also computed, used in the noise detection process. For the computation of both the median and the min/max value a 24-input sorter is utilized, the central pixel value is not included. In this way, the complexity of the design is reduced since no additional min/max modules are utilized. The modules of the sorter used only in the case of the 5x5 pixel neighborhood are enabled by the "en5x5" control signal. A CS block is a max/min module; its first output is the maximum of the inputs and its second output the minimum. The implementation of a CS block includes a comparator and two multiplexers and is depicted in Fig. 4. The outputs of the sorter, denoted as "OUT_0[7..0]"..."OUT_23[7..0],"produce a "sorted list" of the 24 initial pixel values. The output OUT_0[7..0] is the minimum pixel value for both 3x3 and 5x5 pixel image window. The sorter outputs "OUT_3[7..0]" and "OUT_4[7..0]" and the central pixel value are utilized for the computation of the median value for the 3x3 pixel neighborhood, denoted as "MEDIAN_3x3 [7..0]."

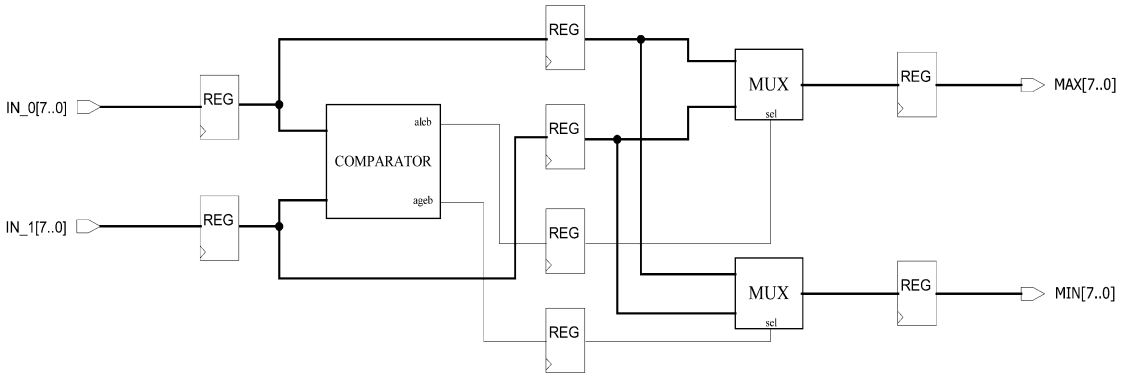


Figure.4. Implementation of CS block

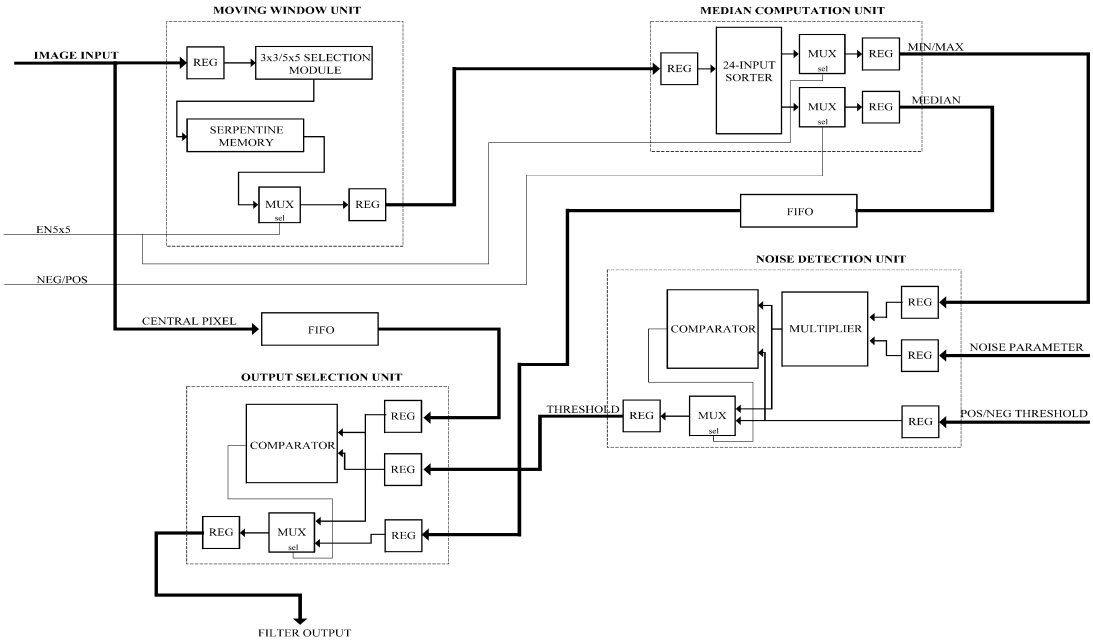


Figure.5. Hardware structure of the figure

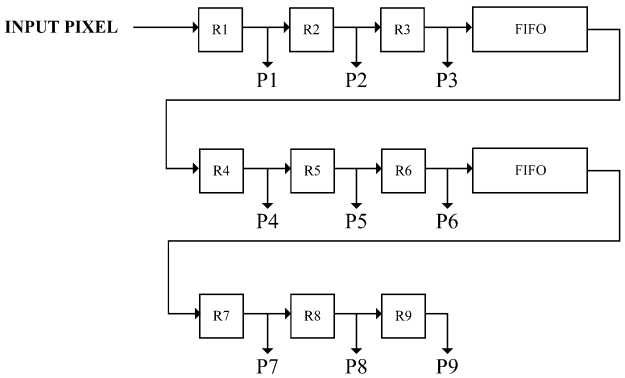


Figure.6. Schematic diagram of a 3x3 pixel serpentine memory

Noise detection unit

The task of the noise detection unit is to compute the threshold value for the detection of a noise pixel, $f_{threshold}(x)$, and to limit this value to the positive (negative) threshold. Initially, the min/max value of the neighborhood is selected, and for that reason, the values "OUT_0[7..0]," "OUT_7[7..0]" and "OUT_23[7..0]" (min and max values, respectively) are imported into a multiplexer. The selection is based on the values of the "NEG/POS" control signals. In the next step, the output of the multiplexer is multiplied by the parameter α (8 bits) using a multiplier module, the resultant 16-bit value is denoted. An additional 2-to-1 multiplexer is utilized to select the positive or negative threshold to which the THRESHOLD_VALUE should be normalized, controlled by

the "NEG/POS" control signal. A comparator is used to compare the THRESHOLD_VALUE to the positive or negative threshold and a multiplexer to select the corresponding output threshold value, denoted as THRESHOLD

Output selection unit

The final stage of the design is the output selection unit. In this unit, the appropriate output value for the performed operation is selected. For the selection of the output value the corresponding threshold value for the image neighborhood, "THRESHOLD," is used. The value "THRESHOLD" is compared to the central pixel value, denoted in the circuit as "CENTRAL_PIXEL." Depending on the result of the comparison, the central pixel is considered to be contaminated by noise or not. For the case of positive (negative) noise, if the central pixel value is greater (less) than the corresponding threshold value, then the

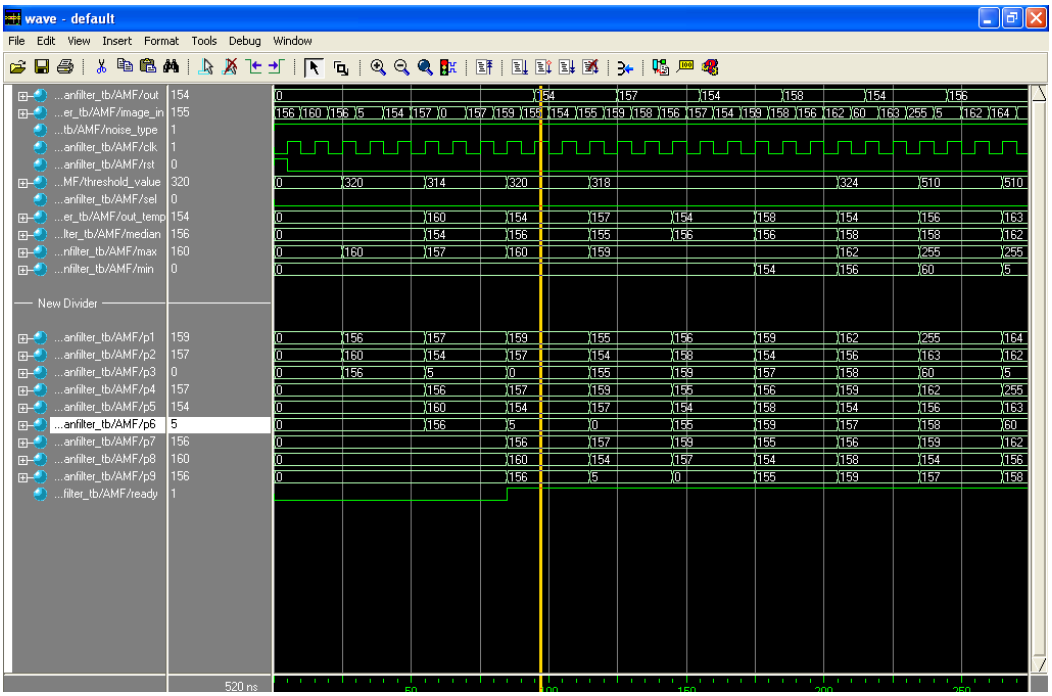
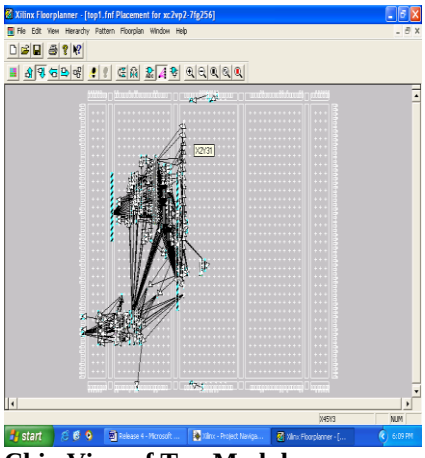


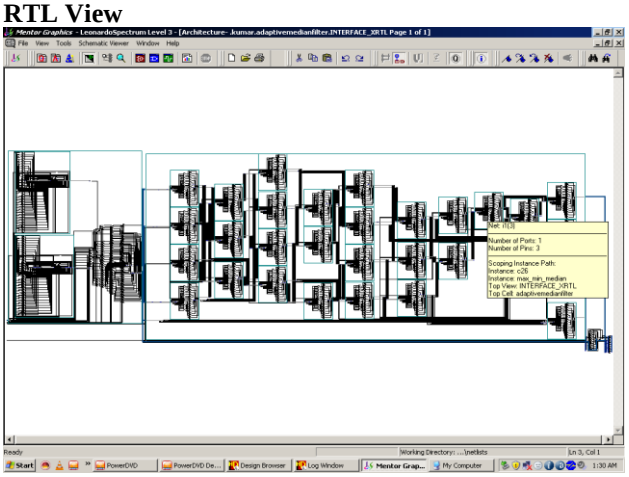
Figure.7. Complete simulation results for the circuit

Central pixel is positive (negative) noise and has to be eliminated. The "FILTER_OUTPUT" is the output of this the adaptive filter. Note the way that the input data is converted to 3-pixel rows every three clock pulses, and that for a sliding of the window to the next image neighborhood only three pixel values of the memory are changed. The modification of the system to accommodate windows of larger sizes is done in a straightforward way, requiring only a small number of changes. More specifically, in the first unit, the size of the serpentine memory used for the serial input of the image data and the corresponding number of multiplexers increase following a square law. In the second unit, the sorter module should be modified, whereas no changes are required in the last two units.



Chip View of Top Module

RTL View



Floor plan placement of the top module

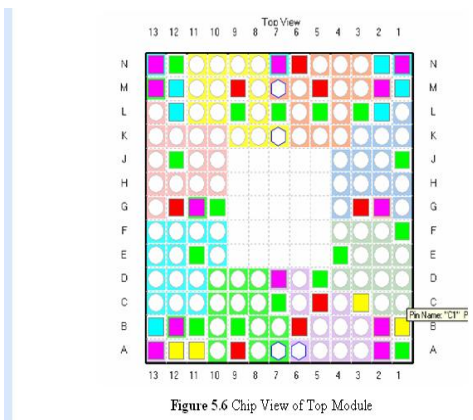


Figure 5.6 Chip View of Top Module

Synthesized top level report

Device Utilization for 2V80cs144
**********Resource
Used Avail Utilization

IOs 19 92 20.65%
Function Generators 608 1024 59.38%
CLB Slices 304 512 59.38%
Dffs or Latches 161 1300 12.38%

Using wire table: xcv2-80-6_wc
Clock Frequency Report
Clock : Frequency
---
clk : 44.6 MHz

II. CONCLUSION

This paper presents a new design of an adaptive median filter, which is capable of performing impulse noise suppression for 8-bit grayscale images using a 3x3 or 5x5 pixel neighborhoods. The proposed circuit detects the existence of noise in the image neighborhood and applies the corresponding median filter only when it is necessary. The noise detection procedure is controllable, and, thus, pixel values other than the two extreme ones can be considered as impulse noise, provided that they are significantly different from the central pixel value. In this way, the blurring of the image is avoided. The system is suitable for real-time imaging applications where fast processing is required. Moreover, the design of the circuit can be easily modified to accommodate larger size windows. In this case, only small modifications are required in the first two units, mostly regarding the size of the serpentine memory and the sorter module. The proposed digital hardware structure was designed, compiled and successfully simulated in FPGAs. The typical system clock frequency is 65 MHz.

III. REFERENCES

[1].W. K. Pratt, Digital Image Processing.New York: Wiley, 1991.

[2]. G. R. Arce, N. C. Gallagher, and T. Nodes, "Median filters: Theory and applications," in Advances in Computer Vision and Image Processing, T. Huang, Ed. Greenwich, CT: JAI, 1986.

[3]. T. A. Nodes and N. C. Gallagher Jr., "The output distribution of median type filters," IEEE Trans. Commun., vol. COM-32, pp. 532–541, May 1984.

[4]. T. Sun and Y. Neuvo, "Detail-preserving median based filters in image processing," Pattern Recognit. Lett., vol. 15, pp. 341–347, Apr. 1994.

[5]. E. Abreau, M. Lightstone, S. K. Mitra, and K. Arakawa, "A new effici-cient approach for the removal of impulse noise from highly corrupted images," IEEE Trans. Image Processing, vol. 5, pp. 1012–1025, June 1996.

[6]. K. S. Ali, "Digital circuit design using FPGAs," in Proc. 19th ICCIE, 1996, pp. 127–129.

[7]. K. E. Batcher, "Sorting networks and their applications," in Proc. AFIPS-SJCC, 1968, pp. 307–314.

[8]. I. Pitas and A.N. Venetsanopoulos Nonlinear Digital Filters: Principles and Applications. Boston, MA: Kluwer, 1990.

[9]. R.M Hodgson, D.G. Bailey, M. J. Naylor, A.L. M. Ng, and S.J. McCneil, "Properties, Implementations and applications of rank filters", Image Vis. Comput., vol.3,pp. 3-14,1985.

[10]. E. R. Dougherty and P. Laplante, Introduction to Real time Imaging. Bellingham, WA:SPIE,1995.