A new method for designing an efficient switching median filter using VLSI architecture to remove salt and pepper noise

Fatma Sakr[a], Nermin Rafiq Mohamed [b], Suresh Muthusamy[c], Santhiya Pandiyan[d], Chitra Manickam[e], Akilesh Murugesan[f] , Dhanushkar Gnanavel[f], Ibrahim Abd Elatif Gomaa[g], and Diaa Salam Abd Elminaam*[h]

[a] Computer Science Department, Faculty of Computers and Artificial Intelligence, Benha University, 12311, Egypt, fatma.saqr@fci.bu.edu.eg
[b] Sadat City University, Egypt,dr.nerm@gmail.com
[c] Department of Electronics and Communication Engineering, Kongu Engineering College (Autonomous), Perundurai, Erode, Tamil Nadu, India infostosuresh1@gmail.com
[d] Assistant Professor, Department of Computer Science and Engineering, Kongu Engineering College (Autonomous), Perundurai, Erode, Tamil Nadu, India santhiyaashok1722@gmail.com
[e] Department of Electronics and Communication Engineering, Kongu Engineering College (Autonomous), Perundurai, Erode, Tamil Nadu, India, chitrae.ecee@kongu.edu ,: akileshakilesh1102@gmail.com, dhanushseenu555@gmail.com
[g] Computer Science Department, Al-Obour High Institute for Management and Informatics, Cairo 11235, Egypt, ibraheemg@oi.edu.eg
[h] Information Systems Department, Faculty of Computers and Artificial Intelligence, Benha University, 12311, Egypt

*Corresponding Author: Diaa Salam Abd Elminaam [Diaa.salama@fci.bu.edu.eg]

ARTICLE DATA

Article history:
Received 03 Aug 2022
Revised 14 Aug 2022
Accepted 14 Aug 2022
Available online

Keywords:
median filtering, salt and pepper noise, edge preservation, three-cell sorter

ABSTRACT

Any imaging system's images have dynamic intensity value variations, rapid light shifts, and poor contrast. Such visuals are empty of useful information and difficult to comprehend visually. Filtering methods to eliminate noise, improve contrast, and detect edges are used to retrieve secondary information from such photographs. Median filtering is one of the nonlinear ways that remove the range of isolated noise like salt and pepper noise while preserving the edge information of the image. However, median filtering fails to remove the noise when the image is affected by too strong impulsive noise. The switching median filtering technique can be used to eliminate high-intensity impulse noise. This research presents a VLSI design for a novel switching-based median filter to reduce high-density salt and pepper noise in digital images. The absolute difference between the center pixel and the array median obtained from a 3 x 3 sliding window is compared to a predefined threshold value to determine whether a pixel is noisy or not. During the filtering stage, the noisy pixels in the 3 x 3 filtering window are replaced by the median of noise-free pixels. If a pixel value is damaged, it is replaced with the median of the following window. The true pixel intensity value is kept if the pixel is not destroyed. We can tell if a pixel is corrupted or not by using a threshold detector. We may either add salt and pepper noise to an image directly or create it ourselves. This is a step in the image preparation process. The normal image is uploaded and converted it to a salt and pepper noise image using MATLAB's built-in capabilities. We used three tools in our project. We used MATLAB to preprocess the picture and plot data to reconstruct the original image, then ModelSim to produce the median filter, and XILINX to compare the existing bubble sort to the recommended three-cell sorter. The proposed technique outperforms standard median-based filters in simulations and is particularly effective in circumstances where pictures are badly damaged.

1. Introduction

Images are regularly impacted by the external environment and contaminated with impulses during the acquisition and transmission of images via the channel. The salt and pepper noise is caused by fixed impulsive noise that corrupts the genuine intensity value in random positions with corruptive values in the extreme ranges, i.e., '0' (pepper) and '255' (salt). Random valued impulse noise is another sort of impulse noise that corrupts the true intensity value with corruptive values in the range [0, 255], which is also the image's dynamic range. In many signals and image processing applications, it is critical to suppress noisy signals while keeping the required information, i.e., without sacrificing edge information. Linear, average, and median filtering have all been used to smooth noisy signals, but linear filtering smooths noisy signals and edges (high-frequency information) [1-3].

Median filtering has the following properties: (a) it smooths transient signals, (b) it removes impulse noise from signals, and (c) it maintains edge information in filtered signals (images). Filtering has the goal of restoring the original picture from a noise-corrupted image. In general, linear filters can reduce impulsive noise, but the picture is blurred. As a result, the standard median filter, the most well-known nonlinear filter, is frequently utilized due to its simplicity and computational efficiency. However, at greater noise densities, it displays a blurring effect for bigger window sizes and less noise suppression for lower window sizes. So, in this case, we employed a switching median filter, which effectively removes high-density noise while preserving picture edge information. The temporal complexity and blurring impact is minimized when employing a switching median filter since it moves from one window to another based on the qualities of the pixel, whether it is damaged or not. This project proposes an improved sorting architecture for switching median filters to analyze images rapidly [4-7].

2. Background

There are variant types of filters that will be described as follow:

Median Filter

The median filter is one of the order-statistic filters because of its great performance for noise types such as gaussian noise, salt and pepper noise, and other random noises. The median filter substitutes the norm value of the linked 3 x 3 window for the middle pixel of a M X M matrix. Noise pixels, it should be noted, are regarded as significantly diverse from the middle pixel. By this inkling, the median filter may be used to eliminate this type of noise. Before binarization, this filter minimizes noisy pixels in protein crystal images. As a result, even at low levels of corruption, impulsive noise can degrade the image's appearance and quality significantly[8-10]. Order-statistic filters, better known as rank-order filters, are a prevalent method to deal with impulsive noise. This filter acts in the spatial domain and is nonlinear. It uses a sliding window approach, in which the pixel values closest to the window's center are changed once every iteration. The ordered intensity values of the pixels inside the filtering window's selected region are used to determine this value. The most prevalent techniques for reducing bipolar and unipolar impulse noise are median filters, among the rank-order filters. The median filter, in general, filters based on the median value. The median value of a sample is given in (1),

$$\tilde{X} = \begin{cases} \mathrm{X}(\mathrm{n}+1)/2 \\ 0.5(\mathrm{X}(\mathrm{n})/2 + \mathrm{X}(\mathrm{n}+1)/2) \end{cases}$$

where X1, X2...X_n are the intensity values, ordered in ascending or descending order, and ns is the sample size. There are, however, numerous median filter modifications. As a result, this study examines some median filter frameworks such as Standard Median Filter (SMF) and Switching Median Filter (SWMF).

Weighted Median Filter

Weighted Median filters have the heftiness and edge-preserving capabilities that are similar to the traditional median filters and share certain qualities with linear FIR filters. Furthermore, WM filters are part of the stack filter family of nonlinear filters. This allows the latter class's tools to describe and study the deeds and attributes of the weighted median filters, such as noise diminution capabilities. Because the weighted median filters are the threshold functions, neural network methods may be used to create adaptive weighted median filters. We follow the evolution of the theory of weighted median filtering from its origins in the median filter to the discovered theory of optimum weighted median filtering in other articles. In the application of speech processing, the idempotent weighted filter is used. In case of image sequence restoration, the optimal weighted median filters are used. The weighted median filter is used in scan rate conversion in normal Television system. One of the median filter branches is the weighted median filter[11-14]. The weighted median filter was first presented in 1981. The only difference between the weighted median filter and standard median filter is that each filter element in the weighted median filter has a weight associated with it. To calculate the median value, these weights correspond to the number of replications of pixel values. In (2), the filtered picture from WMF is presented,

$$F(i, j) = \text{median}_{(k, l) \in W_{h, w}} \{W_{h, w}(k, l) \otimes D(i + k \ldots)\}$$

Where operator denotes a repeating operation. The median value is obtained using the preceding equation, where ns is the sum of filter weights. Usually, the filter weight is configured to decrease as it moves out from the center of the n x n window. As a result, it is envisaged that the filter would focus more on the core pixel, improving noise suppression while preserving picture features. The success of the WMF in conserving picture detail largely relies on the weighting factors of the input image. In practice, finding adequate weighting factors for this filter is challenging, and this filter demands a long computing time when the weights are considerable[15-18].

Iterative Median Filter

The use of an iterative filtering technique is required in several impulse noise filtering systems. The iterative technique demands several iterations of the same operation. A ni iteration iterative median filter, in general, necessitates temporary images. This method allows the median filter to employ a small filter size and minimize computing time while still upholding the quality of the picture. The user can specify the number of iterations, or the iteration process can be terminated after the output picture has converged (the current output image equals the previous output image). The number of repetitions required in practice is determined by the extent of corruption as well as the type of the input picture[19-22].

Recursive Median Filter

Several median filtering studies employ a recursive approach in their research. Recursive median filters are theoretically comparable to infinite impulse response filters since their outputs are determined by the input intensities and the computed outputs at earlier places. When exploiting the recursive median filter, the tainted and filtered images usually share a similar array matrix of pixel data. The already processed pixels are now considered noise-free input pixels in this procedure. Consequently, it is alleged that substituting the input pixels with these values will improve the accuracy of the median value computation. The issue may expand to other image sections if the filter fails to eliminate the noise in previous locations. It's also worth noting that the filtering direction affects the output of the recursive median filter[23-25].

Directional Median Filter

The 2-D filter of a directional median filter, better known as a stick median filter, is broken down into many 1-D filter elements. Each filter element or stick represents a certain direction or angle and is represented as a straight line. There will be h+w-2 sticks utilized for a window of size h x w pixels[19-24].

2.1. *Existing Method*

The switching Median filter technique, each pixel is processed and if noise if detected, the filtering will apply; otherwise left unaltered. Assume we're working with a grayscale digital picture whose intensity is represented in an 8-bit integer, giving us 256 grey levels between 0 and 255. The noise can be either positive or negative. In the picture, positive impulses show as white (salt) spots with an intensity of 255 and a probability of pw. Negative impulses, however, show in the picture as black (pepper) dots with intensity 0 and probability pb. Otherwise, the image is corrupted by a moderately high level of salt-and-pepper noise. The switching median filter technique evaluates each and every pixel in the noisy image for noise, then filters it if it is damaged; otherwise, the image is left unmodified. The fixed valued impulse noise corrupted pixels can take either maximum or lowest intensity values in the dynamic range [0, 255]. The processing pixel is noise-free and keeps its original value if it lies within the range between maximum and minimum intensity values. If the noisy intensity value is found to be noisy, as shown in the following algorithm stages, the median value of nearby pixels is used to replace it. The switching median filter's general structure. The Switching Median Filter will compute each pixel of the picture using a window mask (3x3) and an algorithm to detect if the current pixel is damaged or not, then replace the pixel value with the median value if it is contaminated or leave it untouched. This technique will be repeated for the whole image to remove salt and pepper noise. The Switching Median Filter's Noise Detector, Sorting, and Switching stages are used to compute every pixel of the picture for noise reduction. If the Noise detector step detects the presence of a corrupted pixel, the damaged pixel will be subjected to further filtering; otherwise, the pixel will be left untouched. After that, the masked pixel will be sorted in descending order to obtain the median value, which will be used to filter the data further. A Threshold Detector (to evaluate whether the current pixel is altered with the median value or left untouched using MUX) and a Threshold Value Block (to compute the detector's threshold value) make up the switching stage (the value will be the absolute difference of the current pixel value and the median value)[18-25].

2.2. *Proposed Method*

Ranking filters (ROF) are common examples of nonlinear discrete-time signal processors. The content of the current window is a two-step process. First, the samples are arranged in numerical order within the window. Then the value which takes a prescribed position within the order is selected as the output sample. Finally, when a new input sample is available, it scrolls the oldest input sample currently in the window. Then the sorting and selecting process is repeated. If M is odd (the common case), choosing the mean from the sort list makes the ROF a median filter. The first or last sample in the ranking produces a ROF that looks up the maximum or minimum value within the current window. Of course, other sort order choices are also available, although little is currently known about their properties and usefulness in signal processing tasks. Smoothing, noise reduction, edge detection, and other uses of rank-order filters are common in image processing. They're nonlinear filters that rank the input sequence before selecting an output.

The rank-order filters sort the inputs concerning their magnitudes based on merge-sort operations. The odd-even merge-sort method is described in this section and may be used to perform merge-sort operations effectively. The odd-even merge-sort method takes two presorted sequences and merges them to create a single sorted sequence. Odd-merge and even-merge are two merging components. Odd-merge receives all inputs that have odd subscripts. The even-merge takes in all inputs with even subscripts. Except for the first output of odd-merge, the biggest element, and the last output of even-merge, which is the smallest element, each merging unit's outputs are compared. The sorted outputs are obtained by comparing the ith output of the odd merge unit with the output of the even merge unit.

Merge-sort units of various sizes can be joined to generate bigger ones. A compare-and-swap unit is the smallest merge-sort unit, commonly abbreviated as C&S and referred to as a compare-and-swap unit. Parallel processing decreases power usage by decreasing the supply voltage while maintaining the same sample rate as a sequential system. The supply voltage, or the power reduction factor, is subject to a basic limit. β, i.e., the supply voltage $\beta V0$ cannot be lower than the threshold voltage Vt of the CMOS device. Usually, the supply voltage is maintained at twice the threshold voltage for reliable operation. The same amount of hardware is duplicated L times for trivial block processing. In other words, the area of the parallel system is increased linearly concerning the block size L. Therefore, the power consumption of the parallel system is $Ppar = \beta 2Pseq$, where $Pseq$ denotes the power consumption of the original sequential system. This imposes a lower bound on power reduction. However, by using the substructure sharing to reduce the number of C&S units for parallel rank-order filters, the area (or total capacitance) of the parallel filter grows less than linearly with the block size. Efficient parallel rank-order filters can be designed by taking advantage of shared comparisons within the block structure such that the hardware complexity of the resulting parallel filter is reduced substantially. In digital signal and picture processing applications, the median, maximum, and minimum are the most used ranks. The median filter is well-known for removing impulse noise while preserving the edges. In contrast, the maximum and minimum filters are used to enlarge or contract regions and lines of images in applications such as robot vision and pattern recognition. The sequence of numbers which has to sort is given to the input, and the ROF will sort the number by the odd-even merge sort algorithm, and the output will be in sorter order.

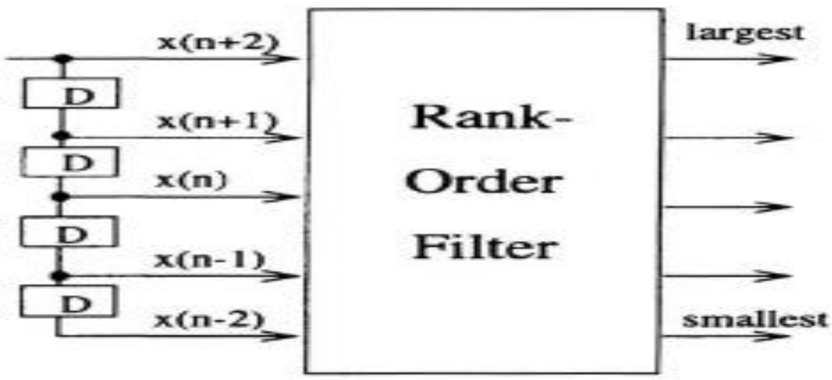

Figure 1. Rank Order Filter

2.2.1 Architecture of Rank Order Filter

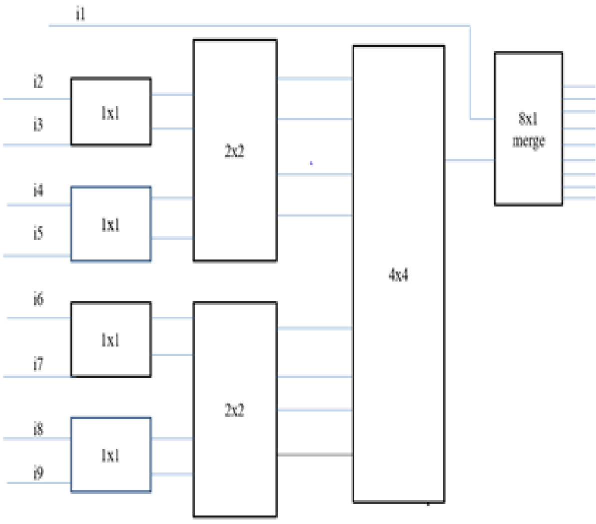

Figure 2. Structure of 9X9 Rank Order Filter

Figure 2 represents the structure of the proposed Rank Order Filter architecture. The Rank Order Filter uses the parallel architecture method for sorting the input based on the number's magnitude and utilizes an odd-even merge sort algorithm. Here the architecture construction contains compare and swap units and merger units. Each merger unit has been compared and swapped based on the number of inputs. Here the architecture has 1x1, 2x2, 4x4, 8x1, and 4x1 2, 4, 8, 9, and 5 inputs. Each merger unit merges and sorts the respective mergers' 2, 4, 8, 9, and 5 inputs. Each merger block is then constructed individually using Verilog. It is merged to get the 3 x 3 window sorted from larger to smaller, and the median value is also acquired to replace the corrupted pixel intensity values, which may be salt (255) or pepper (0) noise.

2.2.2 The 1 X 1 Merger unit

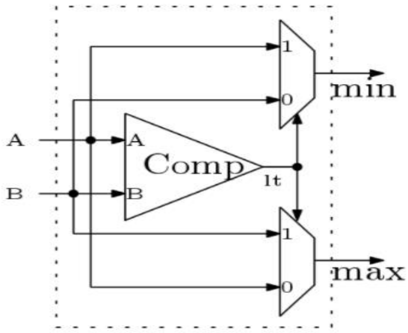

Figure 3. Structure of 1X1 Merger Unit

Figure 3 represents the block of the 1x1 merger. It compares given two individual numbers using a comparator and the output will be the minimum and maximum of the given number by selecting the multiplexer. 1 x 1 merger comprises a magnitude comparator and multiplexer.

2.2.3 The 2 x2 Merger Unit

Figure 4 represents the structure of the 2x2 merger unit, and it is constructed using 1x1 merger units to sort 4 inputs from two sets of different numbers. The input of the merger is pre-assumed as sorted one. Each block compares two inputs and gives maximum and minimum values. It has 3 C&S blocks, and the first block compares odd indexed numbers to give the largest value of inputs, another will compare even indexed numbers and give the smallest of the inputs, another block will compare the smallest of 1st block and highest of 2nd block and give the highest, smallest of the input. The overall output of the 2X2 merger will have the sorted output of four sequences.

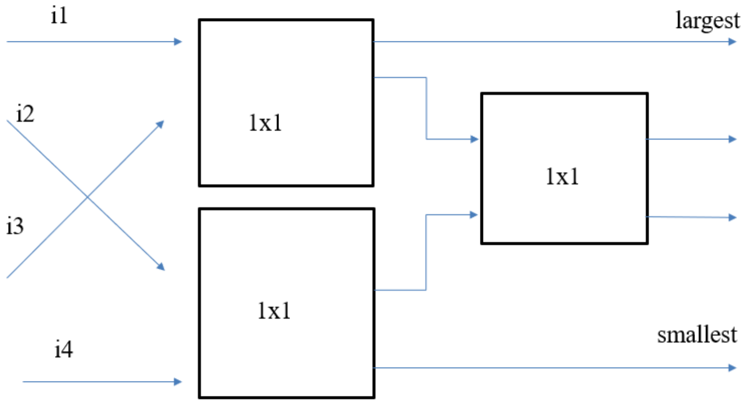

Figure 4. Structure of 2X2 Merger Unit

2.2.4 The 4X4 Merger Unit

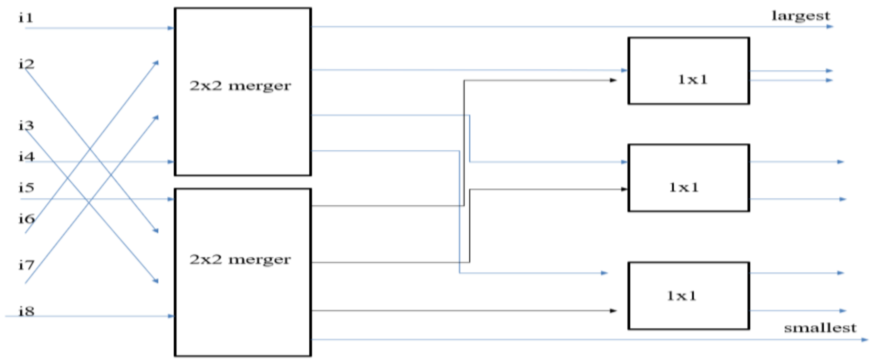

Figure 5. Structure of 4X4 Merger Unit

Figure 5 represents the structure of 4x4 merger block. It is structured using 2x2 merger and 1x1 merger to sort and merge two sorted sequence of 4 values and merge them as 8 sequence values. The input of the merger is pre-assumed as the sorted one. One of the 2x2 merger sort and merge the odd indexed values and give the largest sequence; another 2x2 merger sort and merge the even indexed values and give the smallest sequence. The 1x1 merger units will sort the remaining values that are output of 2x2 merger and the output of the entire 4x4 block will be the sorted ordered of eight numbers.

2.2.5 The 4X1 Merger Unit

Figure 6 represents the structure of the 4x1 merger block. It is constructed using 1x1 merger units. The input of the merger is pre-assumed as sorted inputs. Each 1x1 unit will give the maximum and minimum of the input. The overall output of the 4x1 merger will be the sorted ordered of 5 outputs.

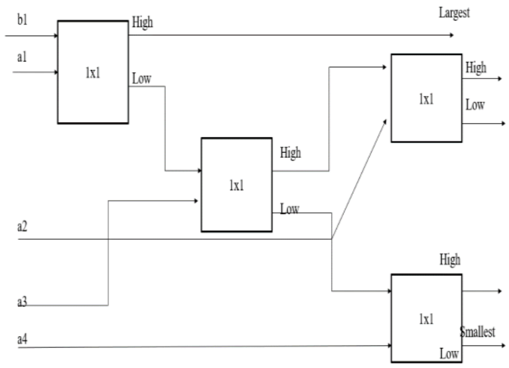

Figure 6. Structure of 4X1 Merger Unit

2.2.6 The 8X1 Merger Unit

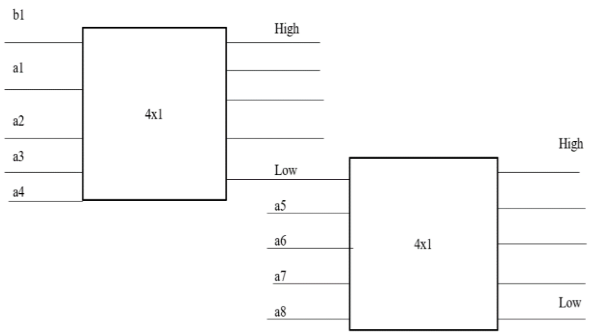

Figure 7. Structure of 8X1 Merger Unit

One of the units will sort the input with odd indexed values and give the highest value of the sequence, and remaining order, one of the units will sort the input with odd indexed values and give the highest value of the sequence and remaining order, the output of 8x1 one will be in sorted ordered.

3. RESULTS AND DISCUSSION

3.1. Output of Rank Order Filter

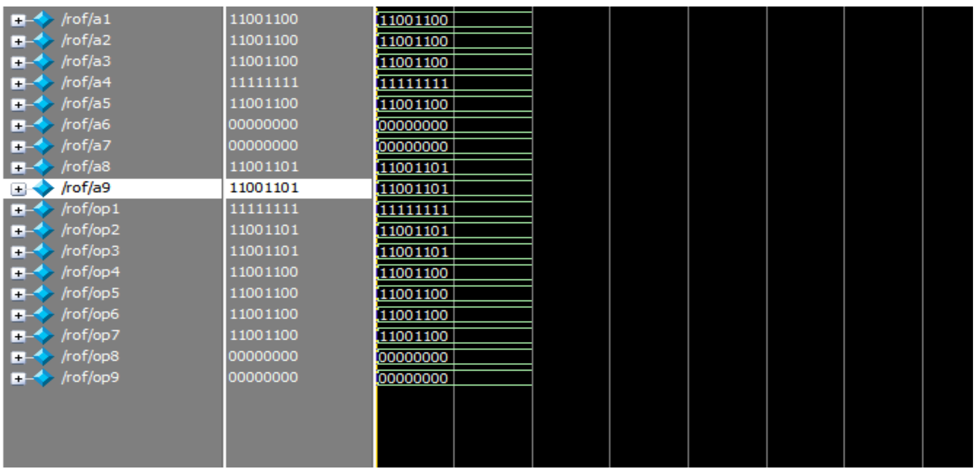

Figure 8. The output of the Rank Order Filter

Rank Order Filter sorts the given input number in parallel architecture based on the magnitude of the value. The first value of ROF will be the largest value of the given input, and the last output will be the smallest number of the given number. The output's center value will be the given sequence's median value.

3.2. Output of the Switching Median Filter

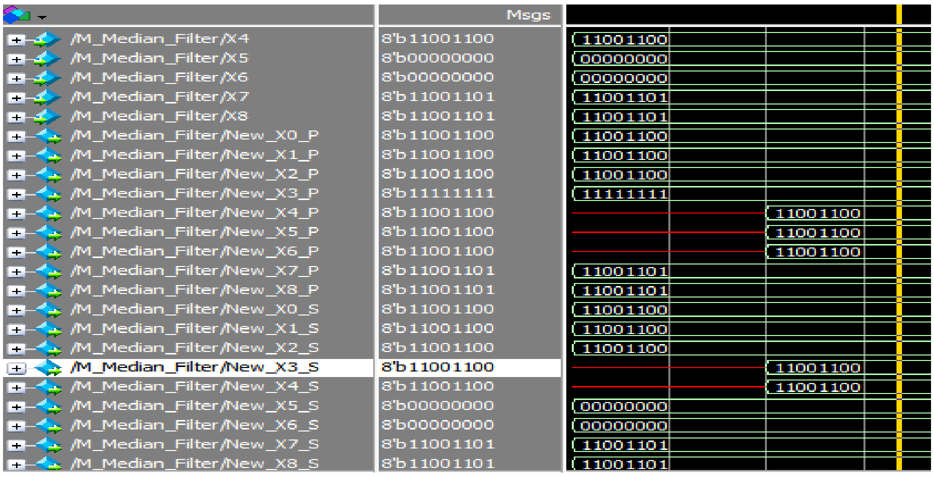

Figure 9. The output of Switching Median Filter

Figure 9 represents the final output of the switching median filter. Here the nine input pixels values are given as input, then after it detects salt and pepper in the noise detector stage. After this, it sorts all pixel values in rank order filtering stage, computes the median value, and replaces it with the salt and pepper values to remove noise in the image.

3.3. Comparison

The number of compare and select units required for three cell sorter is 84 (maximum- 4CS, minimum- 4CS, middle-4CS). The number of compare and select units required for the Rank Order Filter is 27 (1x1 merger -4, 2x2 merger -6, 4x4 merger -9, 8x1 merger-8).

Table 1 represents the comparison of 3-cell sorter and ROF, which are used in the sorting stage of the filter. Here the number of slices utilized, the number of LUT's utilized and the number of comparators utilized by ROF is much lesser than 3-Cell Sorter. At this moment the area occupied by the ROF will be less than the Three-Cell Sorter.

TABLE 1. Comparison between three cell sorter and rank order filter

Parameters	Three cell Sorter	Rank Order Filter
No.of.Slices (out of 2448)	363	356
No.of. LUT's (out of 4896)	648	636
No.of.Comparators	84	27

4. Conclusion and Future Work

Regarding comparators, the three-cell sorting has a much larger number of them, whereas the Rank Order Filter has a smaller number of them compared to the three-cell sorting. It is one of the most significant benefits. The number of compare and select units required for three cell sorter is 84(maximum- 4CS, minimum- 4CS, middlle-4CS.The number of compare and select units required for Rank Order Filter is 27 (1x1 merger -4, 2x2 merger -6, 4x4 merger -9, 8x1 merger-8).As a result, it aids in the reduction of delay. The key advantage of the Rank Order Filter is that it runs in parallel with the architecture, reducing the time it takes. By using other types of comparators, we can reduce the delay, area, or power.

The best one is to use the Carry Select logic data comparator. It consists of a half subtractor, seven numbers of complete subtractors, two numbers of multiplexers, and a not gate. The least significant bit of both inputs is sent into the half subtractor, which then feeds the remaining bits of the inputs to a sequence of seven complete subtractors. The minimum value is found by the multiplexer that gets the borrow of the full subtractor, and the maximum value is found by the multiplexer that receives the complement of the borrow.

As an extension or future work of this work, the complete subtractor can be updated by a full subtractor utilizing mux. Except for the least significant bit full adder, which receives input from the NOT gate, a two complement-based data comparator comprises eight full adders whose inputs come from eight NOT gates and B inputs come from seven XOR gates. To save computational time, the architecture's full adder can be replaced with a full adder that uses multiplexers.

References

[1] V. R. Vijaykumar, G. Santhanamari, S. Elango (2016), 'VLSI Architecture of Switching Median Filter for Salt and Pepper Noise Removal', IAENG International Journal of Computer Science, Vol.43, No. 1.

[2] Chi-Yuan Lien Pei-Yin Chen, Yi- Ming Lin and Po-Kai Chang (2010), 'An Efficient Denoising Chip for The Removal of Impulse Noise', IEEE Transactions on Circuit and Systems – I, Vol.47 pp. 500-678.

[3] Dargana Prokin and Milan Prokin (2010), 'Low Hardware Complexity Pipelined Rank Filter', IEEE Transactions on circuits and systems- II; Express Briefs, Vol.51, No.6. pp. 700-889.

[4] Florencio D. A and Schafer R. W (2002), 'Decision-based median filter using local signal statistics', Visual Communications and Image Processing, International Society for Optics and Photonics, Vol.51, pp.268–275.

[5] Eng H.-L. and Ma K.-K. (2001), 'Noise Adaptive Soft-switching Median Filter', IEEE Transactions on image processing, Vol. 10, No. 2, pp. 242–251.

[6] Govardha, M Senthamizh Selvi, Srinivasn K. S (2012), 'VLSI Architecture for Nonlinear Filter To Remove Impulse Noise', International Journal of Environmental Engineering and Management ISSN 2231-1319, Vol. 3, No.1, pp.119-123.

[7] Gonzalez R. C and Woods R. E (2002), 'Digital Image Processing', 2nd ed. Upper Saddle River, NJ: Prentice-Hall, Vol.5, No.3, pp.330-377.

[8] Herruzo E, Ruiz G, Benavides J. I, and O. Plata (2007), 'A new parallel sorting algorithm based on odd-even merge sort in Parallel', Distributed and Network-Based Processing. PDP'07. 15th International Conference on EUROMICRO, pp. 18–22.

[9] Ko S.-J and Lee Y. H (2011), 'Centre weighted median filters and their applications to image enhancement', IEEE transactions on circuits and systems, Vol. 38, No. 9, pp. 984–993.

[10] Kuo-Liang Chung, Yih-Kai Lin (1997), 'A Generalized Pipelined Median Filter Network', Elsevier Signal Processing 63, pp. 101-106.

[11] Mueller R, Teubner J, and Alonso G (2012), 'Sorting networks on FPGAS', The VLDB The International Journal on Very Large Data Bases, vol. 21, No. 1, pp. 1–23.

[12] Peixuan Zhang and Fang Li (2014), 'A New Adaptive Weighted Median Filter for Removing Salt and Pepper Noise', IEEE Signal Process. Lett., vol.21, No. 10, pp. 1280–1283.

[13] Pei-Yin Chen, Member, Chih-Yuan Lien, Hsu-Ming Chuang (2010), 'A Low-Cost VLSI Implementation for Efficient Removal of Impulse Noise', IEEE Transactions on Very Large-Scale Integration (VLSI) Systems, Vol.18, No. 3. pp. 202-210.

[14] Ravi Teja V.V., Ray K.C., Chakrabarti I., Dhar A.S (2008), 'High Throughput VLSI Architecture for One Dimensional Median Filter', IEEE-International Conference on Signal processing, Communications and Networking (ICSCN), pp. 339-344.

[15] Sadangi S, Baraha B, Satpathy D. B, and P. K. Biswal (2017), 'FPGA implementation of spatial filtering techniques for 2D images', Recent Trends in Electronics, Information & Communication Technology (RTEICT), pp. 1213–1217.

[16] Srinivasan K. S and Ebenezer D (2007), 'A New Fast and Efficient Decision-Based Algorithm for Removal of High-Density Impulse Noise', IEEE – Signal Processing Letters, Vol.14, No. 3, pp.555-621.

[18] Sun T and Neuvo Y (1994), 'Detail Preserving Median Based filters in image processing', Pattern Recognition Letters, Vol. 15, No. 4, pp. 341–347.

[19] Swenson R.L. and Dimond K.R (1999), 'A Hardware FPGA Implementation of a 2-D Median Filter Using a Novel Rank Adjustment Technique', 7th IEEE Int.conf.proc.on Image Processing and its Applications, Vol. 1, pp. 103-106.

[20] Takeaki Matsubara, Vasily G.Moshnyaga and Koji Hashimato (2010), 'A Low Complexity Noise removal technique and its hardware implementation', pp. 4244-6890.

[21] Toh K. K. V, Ibrahim H, and Mahyuddin M. N (2008), 'Salt-and-pepper noise detection and reduction using fuzzy switching median filter', IEEE Trans. Consumer Electron., Vol. 54, No. 4, pp. 956–961.

[22] Vasanth K., Nirmal raj S., Karthik S., and Preetha Mol P (2010),' FPGA Implementation of Optimized Sorting Network Algorithm for Median Filters", IEEE International Conference on Emerging trends in Robotics and communication Technologies, pp 224-229.

[23] Wang Z, Bovik A. C, Sheikh H. R, and Simoncelli E. P (2004), 'Image quality assessment rom error visibility to structural similarity', IEEE transactions on image processing, Vol. 13, No. 4, pp. 600–612.

[24] Wang G, Li D, D. Pan D, and Zang Z (2010), 'Modified switching median filter for impulse noise removal', Signal Processing, vol. 90, no. 12, pp. 3213–3218.

[25] Zhang S and Karim M. A (2002), 'A new impulse detector for switching median filters," IEEE Signal processing letters', Vol. 9, No. 11, pp. 360–363.

.