Detection the Traces of Median Filtering using Higher Order Local Ternary Patterns

G. Pranitha, Dr.E. Nagabooshanam, CH. Madhan mohan

Abstract— **In Real time signal and image processing applications, it is often desirable to be able to perform some kind of noise reduction on an image or signal. Impulsive noise replaces the intensities associated to a certain percentage of pixels by the maximum or minimum possible intensity. This kind of noise is called salt and pepper noise. We go for a special kind of filter called Median filter[1] to reduce this noise.**

The median filter is a non-linear filter which is commonly used to remove impulsive noise from images, while preserving edges and other details. The median of a given sequence can be found by sorting all values in the sequence and by choosing the middle value in the sorted sequence. The median filtering methods presented in this paper follow a histogram-based implementation.

In this project, three different directional median filtering methods for FPGA are proposed. For each of the techniques, four directions are processed at the same time. All techniques presented in this project aim to decrease processing time, while also reducing the hardware resource utilization. Three different directional median filtering methods are: a) By saving only the directional codes, b) Without saving directional codes c) By saving all codes of the window. The processing time requirements for the three proposed methods are dependent on the size of the window and number of filtering directions considered.

***Index Terms*—Median filter, impulsive noise, histogram, FPGA.**

I. INTRODUCTION

Image processing is any form of signal processing for which the input is an image, such as a photograph or video frame; the output of image processing may be either an image or a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it. Image processing usually refers to digital image processing [2], but optical and analog also are possible.

Digital image processing is the use of computer algorithms to perform image processing on images. As a subcategory or field of digital signal processing, digital image processing has many advantages over analog image processing. It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise and signal distortion during processing. Since images are defined over two dimensions (perhaps more) digital image processing may be modeled in the form of multidimensional[3].

An image is an array, or a matrix, of square pixels (picture elements) arranged in Columns and rows is shown in figure 1.

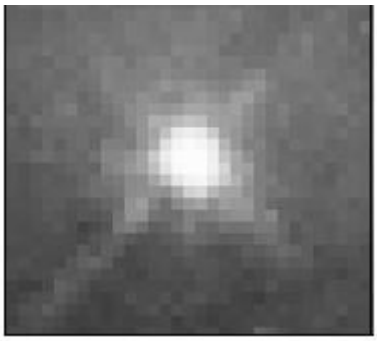

Fig. 1 An image — an array or a matrix of pixels arranged in columns and rows.

In a (8-bit) grayscale image each picture element has an assigned intensity that ranges from 0 to 255. A grey scale image is what people normally call a black and white image, but the name emphasizes that such an image will also include many shades of grey[4].

Real-time signal and image processing applications employ filtering to process, manipulate, or remove noise from data. The median filter is a non-linear filter which is commonly used to remove impulsive noise from images, while preserving edges and other details. Two common types of impulsive noise are salt and pepper noise, and random-valued noise. Impulsive noise replaces the intensities associated to a certain percentage of pixels by the maximum or minimum possible intensity (salt and pepper noise) or by any value between the maximum and minimum intensity (random-valued noise). At the same time, the intensities of the remaining pixels remain intact[5].

II. BLOCK DIAGRAM

The histogram is implemented by instantiating an array of registers is called bin nodes. Bin nodes represent all the possible gray level intensities in an input image (for a gray scale image where each pixel is represented using 8-bits, 2^8= 256 bins are required). A bin node is basically a counter that keeps track of the number of times the bin is incremented.

The main block diagram of median filter which consists of the modules:

1. Pipo shift registers (Accumulator),
2. Sorting Network,

ISSN: 2278 – 7798

3. Comparators,
4. 2-D matrixes,
5. Priority encoder.

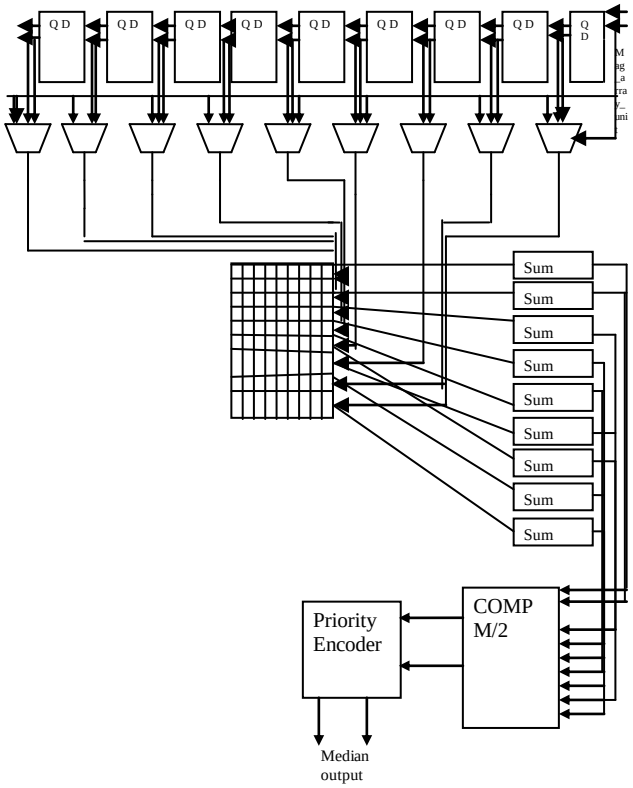

Fig.2 Block diagram of median filter

Each node consists of a register, an incrementer, and a comparator. Each node has two inputs and one output: an enable input, a median index input, and the comparator output. The value of the bin after every increment operation is stored in register, while the comparator checks whether the register value is equal to or greater than the median index input provided. The enable input to bin node determines whether the register value has to be incremented or not. A bin value is incremented if the enable input is "1".

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.

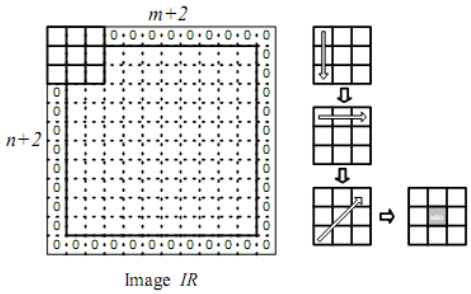

Fig.3 Graphical description for the median algorithm

III. SIMULATION RESULTS

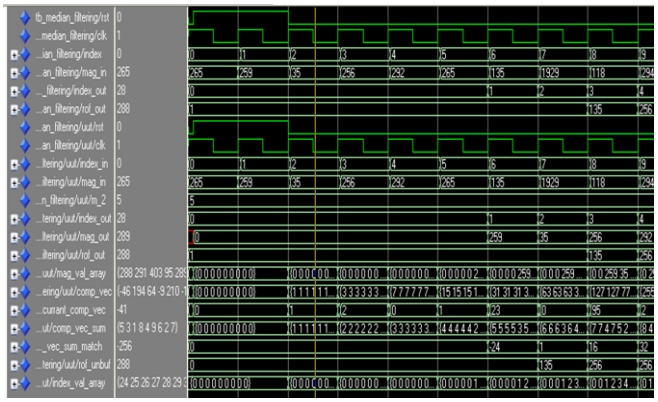

Fig. 4 simulation results of top module

Reset is used to clear data that means data has to start from initial state. All signals starts from next rising edge of the clock. Clock is used for the synchronization. Index input is the input signal passing by using a counter. The counter length is 32 bit. Mag_in is the input signal by passing a counter. The data is randomly passing the image pixel values. An image pixel has the gray levels and having the values. Here we have to find the median of every pixel in the image. Actually we should have to remove the noises like salt and pepper noise.

M_2 is the signal having the 9 bit length but as per median finding we have to set to five only. Index out is the output of the indexes like counter. It has the address like counter of 5 bits length.

Mag_out is the magnitude output which is having the magnitudes of the image pixels. Rof_out is the output signal having final value of the median. Here finding the median as per the removing of the noise like salt and pepper noise by using directional median filtering. Mag_val_array is the signal which is the array of vector storing the signals in vector format. Here the matrix form of the 3x3 is updating. Mag_out is the signal which is the median output of the mag_val_array signal.

Here finding the median and middle value is selected that is comparing the comp_vector with the median value. Comp_vector is the signal which is comparator vector. Here we are comparing the input signal. Rof_out is the final output of the median filter here we are removing the maximum and minimum intensity numbers like salt and pepper noises. Finally removed the salt and pepper noise from the image intensity levels.

IV. CHIPSCOPE RESULTS

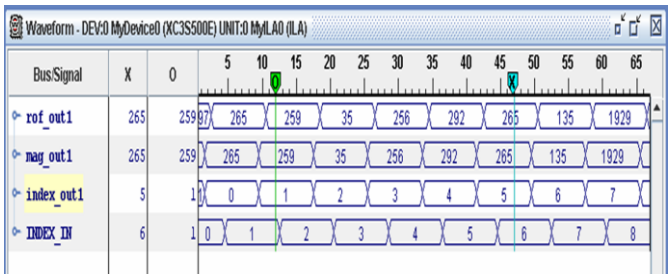

Fig. 5 chipscope results of top module

As per above window which consists of the signals index_in, index_out1, mag_out1 and rof_out1.The length of trigger is 84, in which index_in is 10 bit length, index_out1 is also 10 bit length, mag_out1 is length of 32 and rof_out1 is also 32 bit length.Here index_in is the address counter incrementing every clock cycle which is for 0 to 31 clock pulses.Index_out1 also same as the index_in signal.Mag_out1 is updated by passing image pixel values in the VHDL code and it is updating mag_val_array in array format.Rof_out1 is the final output of the median filter after removing the salt and pepper noise by image levels.

Here we have to observe the signals which are running inside the FPGA hardware. As per the median filtering concept we have to compare and sorting the image values by fixing to m_2 = '5' and storing in rof_unbuf. This is the final Output of the median filtering.

V. TABLE

Logic utilization	used	available	utilization
Number of slice flipflops	698	9312	7%
Number of 4 input LUT's	467	9312	5%
Number of occupied slices	658	4656	14%
Number of bonded IOB's	76	232	32%
Number of block RAM's	6	20	30%

VI. CONCLUSION

In the paper a general, high-speed parallel median filtering architecture was proposed. The parameterised architectural description allowed custom filter realisations, in terms of input sample resolution, filter window width, 1D or 2D implementation.

Compared to other parallel median filter realisations, where stack decomposition was brought to front, this architecture does not involve huge additional matrices, that makes realisations impossible. Compared to solutions carried out on real multiprocessor structures, our architecture contains only one FPGA, that is based on a single board plugged onto a PC.

REFERENCES

1. Olli Vainio, Yrjö Neuvo, Steven E. Butner, *A Signal Processor for Median-Based Algorithms*, IEEE Transactions on Acoustics, Speech, Processing VOL 37. NO. 9, September 1989.
2. V.V. Bapeswara Rao and K. Sankara Rao, *A New Algorithm for Real-Time Median Filtering*, IEEE Transactions on Acoustics, Speech, Processing VOL ASSP-34. NO. 6, December 1986.
3. M. O. Ahmad and D. Sundararajan, *Parallel Implementation of a Median Filtering Algorithm*, Int. Symp. on Signals and Systems, 1988.
4. Dobrowiecki Tadeusz, *Medián Szurok*, Mérés és Automatika, 37. Évf., 1989. 3.szám
5. Xilinx Foundation Series Quick Start Guide, 1991-1997. Xilinx. Inc.

G.Pranitha is presently pursuing final semester M. Tech in VLSI at Sridevi women's engineering College, Gopannapally, Gandipet, Hyderabad**.**

Dr.E.Nagabooshanam is presently working as Head of Department & Professor in the department of Electronics and Communication Engineering in Sridevi women's engineering College, Gopannapally, Gandipet, Hyderabad**.**

CH. Madhan mohan is presently working as Application Engineer in Unistring Tech Solutions Pvt Ltd, Hyderabad, Telangana, India.

ISSN: 2278 – 7798