IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735
PP 26-31
www.iosrjournals.org

A Two Dimensional Median Filter Design Using Low Power Filter Architecture

R.Vinodhini[1], G.Mohanraj [2]
PG Student, VLSI Design, Department of ECE, Maha Barathi Engineering College, India[1].
Assistant Professor, Department of ECE, Maha Barathi Engineering College, India[2]

Abstract: *This brief presents a power architecture for the design of two dimensional median filter .Receiving an input sample and generating a median output at each machine cycle. Sorting method used. The power consumption is reduced by decreasing the number of signal transitions in the circuit. This can be done by keeping the stored samples immobile in the window through the use of token ring in our architecture .Power consumption ,Area, and complexity were successfully reduced. Increased throughput and speed. One first needs to understand what a median filter is and what it does. In many different kinds of digital image processing, the basic operation is as follows: at each pixel in a digital image we place a neighborhoodaround that point, analyze the values of all the pixels in the neighborhood according to some algorithm, and then replace the original pixel's value with one based on the analysis performed on the pixels in the neighborhoodThe neighborhood then moves successively over every pixel in the image, repeating the process.*

I. INTRODUCTION

To understand what adaptive median filtering is all about, the one first needs to understand what a median filter is and what it does. In many different kinds of digital image processing, the basic operation is as follows: at each pixel in a digital image we place a neighborhood Around that point, analyze the values of all the pixels in the neighborhood according to some algorithm, and then replace the original pixel's [1] value with one based on the analysis performed on the pixels in the neighborhood. The neighborhood then moves successively over every pixel in the image, repeating the process Median filtering follows this basic prescription. The median filter [2] is normally used to reduce noise in an image, somewhat like the Median filter. However, it often does a better job than the mean filter of preserving useful detail in the image. This class of filter belongs to the class of edge preserving smoothing filters [4] which are non-linear filters. This means that for two images A(x) and B(x): These filters smooths the data while keeping the small and sharp details. The median is just the middle value of all the values of the pixels in the neighborhood. Note that this is not the same as the average (or mean); instead, the median has half the values in the neighborhood larger and half smaller. The median is a stronger "central indicator" than the average. In particular, the median is hardly affected by a small number of discrepant values among the pixels in the neighborhood. Consequently, median filtering is very effective at removing various kinds of noise. Sometimes we are confused by median filter and average filter, thus let's do some comparison between them. The median filter is a non-linear tool, while the average filter is a linear one. In smooth, uniform areas of the image, the median and the average will differ by very little. The median filter removes noise, while the average filter just spreads it around evenly. The performance of median filter is particularly better for removing impulse noise than average filter.

1) Adaptive median filtering:

Therefore the adaptive median filtering [3] has been applied widely as an advanced method compared with standard median filtering. The Adaptive Median Filter performs spatial processing to determine which pixels in an image have been affected by impulse noise[3]. The Adaptive Median Filter classifies pixels as noise by comparing each pixel in the image to its surrounding neighbor pixels. The size of the neighborhood is adjustable, as well as the threshold for the comparison. A pixel that is different from a majority of its neighbors, as well as being not structurally aligned with those pixels to which it is similar, is labeled as impulse noise. These noise pixels are then replaced by the median pixel value of the pixels in the neighborhood that have passed the noise labeling test.

II. RELATED WORK

Here in this method to calculate the median on a set of NW-bit integers in W/B time steps. Blocks containing B-bit slices are used to find B-bits of the median. A median filter is a non-linear digital filter which is able to preserve sharp signal changes and is very effective in removing impulse noise (or salt and pepper noise).

An impulse noise has a gray level with higher or lower value that is different from the neighborhood point. An order R filter, for a set of N elements has R elements less or equal to the output and N – R elements greater or equal to the output [5]. For N=2k + 1, the median is a rank filter with R=k. The method in this Letter calculates the median by setting P=R+1 initially. The accumulation in Fig. 1 can be performed left to right as shown, or right to left. Thus, for the method to perform as a rank filter,[4] for left to right accumulation, requires setting P=N – R initially. For right to left it requires setting P=R+1.It consists of N cascaded blocks, one for each window rank. Each cell block ci is composed of a data register (Ri), anm-bit (m = log 2 N) age register (Pi), and a control unit for data transfer. Register Ri stores the value of the sample in cell ci, and register Pi keeps the age of this sample, i.e., the number of clock cycles the sample has experienced in the window. All cells are connected to a global input X, through which they receive the new input sample. At each machine cycle, depending on the value of the input sample, the data transfer control unit determines if each cell will keep its original sample, receive the input sample from the outside, or receive the sample from its adjoining cell on the left or right. The window samples are stored in descending order from left to right, so that at any time, the maximum sample is at the leftmost register (R1) and the minimum sample is at the rightmost register (RN). The median value of the window will beat the center cell (R(N+1)/2),assuming N is an odd number .When an input sample is inserted into the window along with some necessary shift operations, each cell in the window may keep it original sample, receive the input sample, or receive the sample from its adjoining cell on the left or right .Median filter [3] can still degrade image quality somewhat, although the preservation of edges is paramount in the computer version domain. Such filters within image processing are almost uniquely two-dimensional and have small window sizes. The trace transform is a recently introduced algorithm that has been shown to be perform well in a variety of image recognition and categorization tasks. Including image database search Face authentication and distortion correction. It maps a standard image to an alternative domain and while defining the spatial mapping is general in terms of mathematical computation. In order to process the input samples sequentially non-recursive Sorting network architecture is proposed. Can abl to achieve high throughput Rate with minimal latency.

III. SYSTEM ANALYSIS

Median filter:

Median filtering follows this basic prescription. The median filter is normally used to reduce noise in an image, somewhat like the mean filter. However, it often does a better job than the mean filter of preserving useful detail in the image. This class of filter belongs to the class of edge preserving smoothing filters which are non-linear filters.These filters smooth the data while keeping the small and sharp details. The median is just the middle value of all the values of the pixels in the neighborhood. Note that this is not the same as the average (or mean); instead, the median has half the values in the neighborhood larger and half smaller.The median is a stronger "central indicator" than the average. In particular, the median is hardly affected by a small number of discrepant values among the pixels in the neighborhood. Consequently, median filtering is very effective at removing various kinds of noise. Fig 1 illustrates an example of median filtering. Like the mean filter, the median filter considers each pixel in the image in turn and looks at its nearby neighbors to decide whether or not it is representative of its surroundings. Instead of simply replacing the pixel value with the mean of neighboring pixel values, it replaces it with the median of those values. The median is calculated by first sorting all the pixel values from the surrounding neighborhood into numerical order and then replacing the pixel being considered with the middle pixel value. Fig 2 illustrates an example calculation.

Calculating the median value of a pixel neighborhood. As can be seen, the central pixel value of 150 is rather unrepresentative of the surrounding pixels and is replaced with the median value: 124. A 3×3 square neighborhood is used here --- larger neighborhoods will produce more severe smoothing.

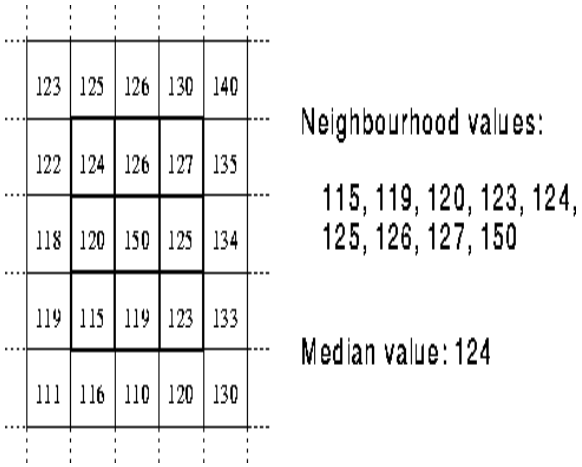


Fig 1 Example Matrix

Low Power Architecture:

Fig. 2 gives an overview of our low-power median filter architecture with window size N. It consists of a circular array of N identical cells and three auxiliary modules: rank calculation (RankCal), rank selection (RankSel), and median selection (MedianSel). All the cells are also connected to a global input register X, through which they receive the incoming sample and the median is stored in the output register Y.

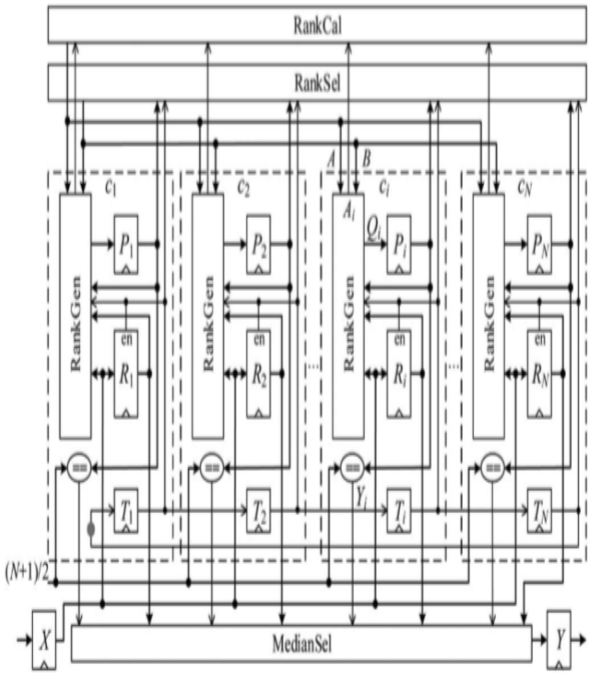


Fig.2 Low-power filter architecture

The architecture is implemented as a two-stage pipeline, where the registers in all the cells serve as the internal pipeline registers.All the registers in the architecture are synchronized by the rising edge of a global clock. Each cell block ci is composed of a rank generation (RankGen) module, a comparator module "==," and three registers: an m-bit (m = log 2 N) rank register (Pi), a data register (Ri), and a 1-bit token register (Ti). Register Ri stores the value of the sample in cell ci, register Pi keeps the rank of this sample, and the enable signal (en) of Ri is stored in register Ti. All the samples in the window are ranked according to their values, regardless of their physical locations in the window. In our design, a cell with a greater sample value will be associated with a greater rank. However, for two cells ci and cj, whose sample values are equal, ci will be given a greater rank if Ri is newer than Rj (or Rj is older than Ri); i.e., the sample in cj enters the window earlier than the sample in ci. The rank is hence unique for each cell. For a window with size N, the rank starts from 1 for a cell with the least sample value, and ends with N for a cell with the greatest sample value. The median of the window can then be obtained from the sample value Ri of a cell ci whose rank Pi is equal to (N + 1)/2, assuming

N is an odd number. In the architecture, the input sample enters the window in a FIFO manner. After it is queued, it will not be moved and is simply dequeued in place. A token, which is represented as logic state 1 in the token register of some cell, is used to control the dequeuing of old sample and queuing of new input sample at the same time. After the token is used, it will be passed to the next cell at a new cycle. All the token registers form a token ring with exactly one token. It serves as a state machine in the circuit, where the location of the token defines the state of the machine. Since the data in each register Ri is immobile, our architecture has the potential for low-power applications. At the initial state of the architecture, to make the first incoming sample be stored in the first cell c1, the token is assumed to exist in the last cell cN, shown as the shadowed circle at the output of register TN.

Circuit Behavior:

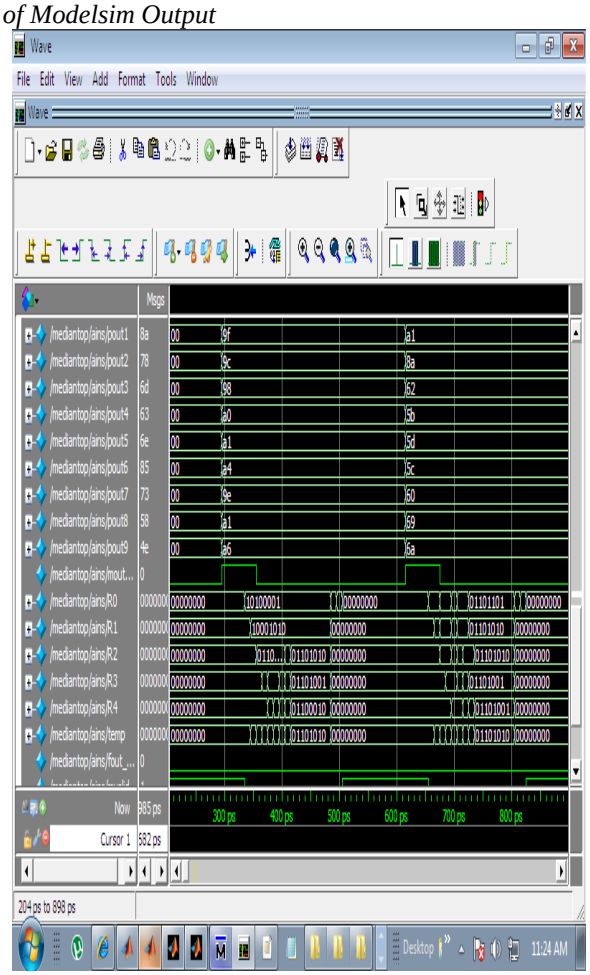
At each machine cycle ti, the first stage of our two-stage pipelined filter performs the following operations for the input sample X: calculate the new rank of each cell, insert X in a cell that contains the token, and pass the token to the next cell. This means that for all Pi, Ri,and Ti registers,their new values will be calculated and determined at this stage so that they can be updated at the next cycle ti+1. At the same time, the second pipeline stage calculates the median value for the input sample that enters the window at the previous cycle ti−1. That is, for the output register Y , its new value will be calculated at this stage so that it can also be updated at the next cycle ti+1.

Cell with the token ring:

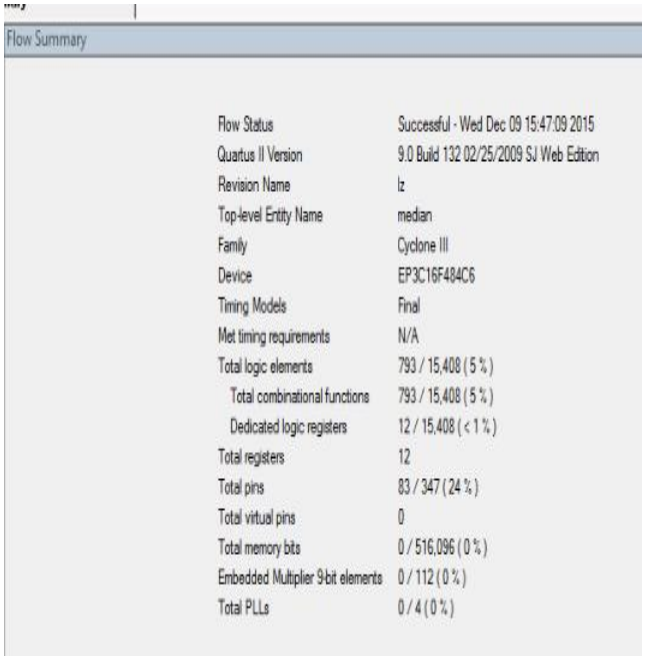
For a cell ci with the token, its sample value Ri will be replaced by the input sample X, and its rank Pi has to be re calculated. The new value of Pi can be obtained by comparing X with the sample values of all the other N −1 cells that do not contain the token. For these cells,if K is the number ofc ells whose sample value is less than or equal to X, the new value of Pi will be K +1. At cyclet6 of Fig. 2, for example, the new value of rank P1 will be calculated as 2+1at the next cycle t7.

IV. SOFTWARE IMPLEMENTATION RESULTS

a) Functional Verification of Modelsim Output



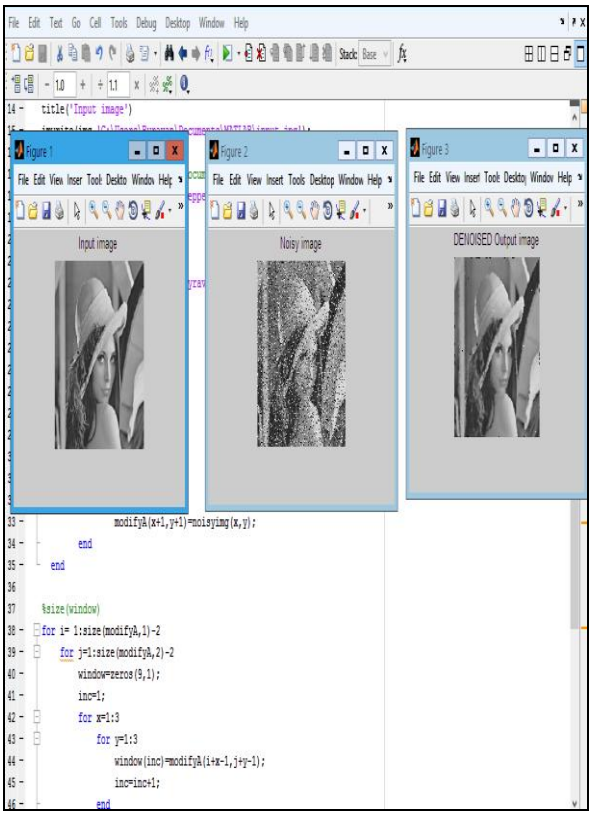
b) Performance Area



Flow Summary

Flow Status	Successful - Wed Dec 09 15:47:09 2015
Quartus II Version	9.0 Build 132 02/25/2009 SJ Web Edition
Revision Name	lz
Top-level Entity Name	median
Family	Cyclone III
Device	EP3C16F484C6
Timing Models	Final
Met timing requirements	N/A
Total logic elements	793 / 15,408 (5 %)
Total combinational functions	793 / 15,408 (5 %)
Dedicated logic registers	12 / 15,408 (< 1 %)
Total registers	12
Total pins	83 / 347 (24 %)
Total virtual pins	0
Total memory bits	0 / 516,096 (0 %)
Embedded Multiplier 9-bit elements	0 / 112 (0 %)
Total PLLs	0 / 4 (0 %)

C) Input Vs Denoised Output



V. CONCULSION

A method to remover salt-and-pepper noise is proposed. This increases the efficiency of the system. The algorithm removes noise by finding median for each macro blocks with minimum number of transitions. The performance of the proposed median filter is better when compared to the other filter of this type. In proposed method to reduce the complexity median is computed with minimum number of registers.

Here we implement median filter on a noisy image blocks. The proposed method is successfully synthesized using QUARTUS II EDA tool.

REFERENCES

[1]. Cadenas J. Megson G.M. Sherratt R.S. and Huerta P. (2012), 'Fast median calculation', Electron. Lett., vol. 48, no. 10, pp. 558–560.

[2]. Chelcea T. and Nowick S.(2004), 'Robust interfaces for mixed-timing systems', IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 12, no. 8, pp. 857–873.

[3]. Chen O. Wang S. and Y.W.wu.(2003), 'Minimization of switching activities of partial products for designing low-power multipliers', IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 11, no. 3, pp. 418–433.

[4]. Chen R.D. Chen P.Y, and Yeh C.H.(2013), 'Design of an area-efficient onedimensional median filter', IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 60, no. 10, pp. 662–666.

[5]. Choo C. and Verma P.(2008), 'A real-time bit-serial rank filter implementation using Xilinx FPGA' in Proc. SPIE Real-Time Image Process., vol. 6811, pp. 68110F-1–68110F-8.

[6]. Fahmy S.A. Cheun P.Y.K. and Luk W.(2009), 'High-throughput onedimensional medianandweightedmedianfilters onFPGA'IET Comput. Digit. Tech., vol. 3, no. 4, pp. 384–394.

[7]. Greiner A. and Panades I.M. (2007), 'Bi-synchronous FIFO for synchronous circuit communication well suited for network-on-chip in GALS architectures' in Proc. 1st Int. Symp.