A Novel Architecture for the Design of One Dimensional Filter - A Survey

Sharanabasappa
Department of ECE
Don Bosco Institute of Technology,
Bengaluru, India (Affiliated to VTU, Belagavi)

Dr. P Ravibabu
Department of EEE
SNIST, Ghatkesar
Hyderbad, India (Affiliated to JNTU, Hyderbad)

Abstract **- A median filter is a nonlinear filter widely used in digital signal and image processing for the smoothing of signals, suppression of impulse noise, and edge preservation. The median filter replaces a sample with the middle-ranked value among all the samples inside the sample window, centered around the sample in question. Depending on the number of samples processed at the same cycle, there are two types of architectures for hardware design, i.e., word-level architecture and bit-level architecture.**

Keywords – Median, Rank, VLSI

1. INTRODUCTION

The median filter is a canonical image processing operation, best known for its salt and pepper noise removal aptitude. It is also the foundation upon which more advanced image filters like unsharp masking, rank-order processing, and morphological operations are built . Higher-level applications include object segmentation, recognition of speech and writing, and medical imaging.

In many signal and image processing applications, it is essential to suppress the noisy signals while preserving the required necessary information i.e., without losing edge information in this process. The techniques such as linear filtering, average filtering, and median filtering have been used to smoothen the noisy signals but the linear filtering smoothens noisy signals as well as edges i.e., high frequency information. Median filtering techniques have been used to smoothen the impulsive noise without losing high frequency signals i.e., preserving edge information.

The effective way to reduce the total power consumption of a very-large-scale integrated circuit is to lowering the dynamic power. The dynamic power consumption is due to switching activities in the circuit and the best way to minimize the signal transitions. Using first in first out interface in the design and the data in the FIFO is stationary, the switching activities are reduced and this design offers low power. One of the best ways to reduce the dynamic power dissipation is to minimize the switching activities, i.e., the number of signal transitions in the circuit [1], [2]. The token-ring architecture is adopted in the design first in first out interface which offers low power consumption since data is stationary in the FIFO [3],[4]. In their designs, a status is maintained in the token to indicate whether queuing is required or not. When the token =1 indicates that refresh a new data by removing the old data from the queue. A token ring is formed by the token register, which are circular register interconnecting nodes. In the token-ring architecture for lowering the power consumption we adopt one token in one dimension (1-D) median filter.

Median filtering is a nonlinear smooth technology. Each pixel of the gray value of a neighborhood has its own pixel gray value of the median. That means all pixels within the neighborhood sort by gray value, taking the median of the group as a neighborhood center pixel output value. A median filter is used in image processing for suppression of impulse noise, and edge preservation and smoothing of signals [5]. There are word-level architecture and bit-level architecture for hardware design depending on the number of samples processed at the same cycle. In the word-level architecture, the input samples are sequentially processed word by word, and the bits of the sample are processed in parallel [6], [7], [8]. On the contrary, the bit-level architecture processes the samples in parallel and the bits of the incoming samples are sequentially processed [9], [10], [11], [12]. For low power median filter normally we will adopt the word-level architecture for practical use.

The set of input samples in the word level sorting network is computed in two phases 1) sorting the input samples and 2) selecting the middle value [6], [7]. The authors proposed their methods, the input samples are sequentially processed word by word, and the incoming sample is inserted into the correct rank in two steps. The oldest sample is removed from the window by moving some of the stored samples to the left in the first step and in the second step, the incoming sample is compared with the already sorted samples and then inserted in the right place by

moving some of them to the right. The difference between the two architectures in [6] and [7] is that these two steps are separately performed in two clock cycles in [6], whereas in [7], it takes only one cycle.
However in these methods the stored samples are shifted left or right depending on their values when a new sample is arrived in the window. For few applications require a large sample width, there will be a more number of signal transition can occur and this will increase the dynamic power consumption.

2. *Literature Survey*

Paper[11] proposed high speed hardware implementation of run-time and fixed variable window length 1-D median filters. In this paper, the clock frequency does not depend on the window length of the median filter. If the window length is increased, the clock frequency is not affecting. This paper mainly concentrates on area, power, and delay. This paper failed to concentrate on a number of slices, LUT, flip flop and frequency on FPGA hardware implementation.

Paper[12] presented an area-efficient one-dimensional median filter based on the sorting network. The window is sorting in descending order in word level filter. The new sample gest into the window. But, the old sample not following the queue that causes the collision in the sample window.

Paper[13] presented a low complexity reconfigurable hardware architecture for the adaptive median filter. In this paper, just focus on mean square blunder (MSE) and Peak Signal to Noise Ratio (PSNR). Here, the power and region are confined one. So can't diminish beneath from its confined range..

Paper[14] proposed a low-power architecture for thhedesign of a one dimensional median filter. It performsword-leveltwo-phase pipelined channel, which creating the middle channel yield. In this paper, range and power can be lessened however any sort of sorting calculation is not utilizing to locate the middle esteem.

Paper[15] presented impulse noise removal using adaptive switching median filter. This worked at two phase, for example, channel stage and detection stage. In detection stage, the boisterous pixel can be recognized. The filtering stage the defiled picture can be de-noising however we can't ready to get high productivity de-nosing picture.

3. Filter architecture.

In order to take care of the power consumption a new median filter architecture is proposed to take care of low power consumption. In this architecture the samples are kept stationary and the rank of each samples are altered and updated. In order to improve the throughput the architecture has two stage pipelines and the median output will be generated at each clock cycle when an input sample enters in the window. The improvement in power consumption is achieved by utilizing a token ring in the proposed architecture. Since the stored samples in the window are stationary, this architecture is suitable for low-power applications.

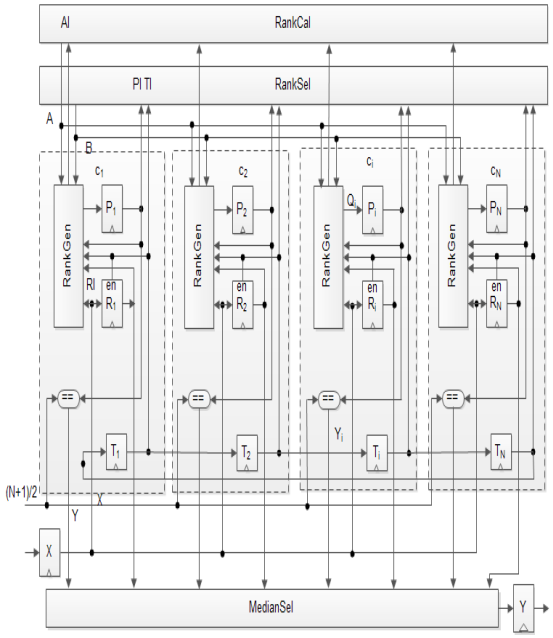

Fig1: Median filter architecture

Fig. 1 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* . 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* = log2 *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 locatio ns 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 wndow 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 Whenever an input sample enters the window at a new cycle, the rank of each cell has to be updated. It may have to be recalculated, or may be the old rank decremented by 1, incremented by 1, or kept unchanged. The new rank of each cell, which is denoted by signal *Qi* in the cell, is generated by the RankGen module. Each RankGen module receives signal *A* from the RankCal module and signal *B* from the RankSel module. Signal *A* is the recalculated rank of a cell *ci* that contains the token and signal *B* is the old rank of *ci*. Moreover, signal *Yi* is the output of a comparator module "==," which compares the value of rank *Pi* with a constant value $(N + 1)/2$ so that $Yi = 1$ if *Pi* is equal to $(N + 1)/2$, else $Yi = 0$. This signal is used to indicate if the corresponding cell *ci* contains the median in *Ri*. Similar to the RankSel module, the MedianSel module transfers the value of *Ri* to the output register *Y* if $Yi = 1$; i.e., if the median is stored in *Ri*.

REFERENCES

[1] Donge, Yiqiuu, and ShufengXu. "A new directional weighted median filter for removal of random-valued impulse noise." *IEEE Signal Processing Letters* 14.3 (2007): 193-196.

[2] Perreeult, Simon, and PatrickeHébart. "Median filtering in constant time." *IEEE transactions on image processing* 16.9 (2007): 2389-2394.

[3] Teja, V. R., Ray, K. C., Chakrabarti, I. &Dhar, A.S.Janu., "High throughput VLSI architecture for one dimensional median filter", In Signal Processing, Communications and Networking, vol.21, no. 6, pp. 339-344, 2008.

[4] Moshnyaga, V. G., & Hashimoto, K., "An efficient implementation of 1-D median filter", IEEE International Midwest Symposium on Circuits and Systems, vol.12, no. 6, pp. 451-454, 2009.

[5] Proekin, Draegana, and Milan Prodkin., "Low hardware complexity pipelined rank filter", IEEE Transactions on Circuits and Systems II: Express Briefs, vol.57, no. 6, pp. 446-450, 2010.

[6] Fange, Q., Zhang, W., Pang, Z., Chen, D., & Wang, Z., "A proposed fast word level sequential scheme and parallel architecture for bit plane coding of EBCOT used in JPEG2000", In Multimedia Technology (ICMT),vol.7, no. 2, pp. 1-4, 2010.

[7] Blaed, A., &Guestaffsson, O., "Bit-level optimized FIR filter architectures for high-speed decimation applications", IEEE International Symposium on Circuits and Systems, vol.16, no. 12, pp. 1914-1917, 2008.

[8] Sanniy, Andreaa, and Viktor K. Prasana., "Energy-efficient median filter on FPGA", International Conference on Reconfigurable Computing and FPGAs (ReConFig), IEEE, vol.15, no. 8, pp. 1-8, 2013.

[9] Leeab, S. B., Ortiaz, A., Leperd, R. F., Shaw, S. R., &Kirtley, J. L. (1997). Applications of real-time median filtering with fast digital and analog sorters. *IEEE/ASME Transactions on Mechatronics*, *2*(2), 136-143.

[10] Oflazear, Kemeal., "Design and implementation of a single-chip 1-D median filter", *IEEE* transactions on acoustics, speech, and signal processing, vol.31, no. 5, pp. 1164-1168, 1983.

[11] Nikahad, Eesha, PaymaenBehnam, and Reza Sameni. "High-Speed Hardware Implementation of Fixed and Runtime Variable Window Length 1-D Median Filters." *IEEE Transactions on Circuits and Systems II: Express Briefs* 63.5 (2016): 478-482.

[12] Chean, R.D., Chean, P.Y. and Yeh, C.H., "Design of an area-efficient one-dimensional median filter", IEEE Transactions on Circuits and Systems II: Express Briefs, vol.60, no. 10, pp. 662-666, 2013.

[13] Mukkherjee, M. and Maaitra, M., 2015, "Reconfigurable architecture of adaptive median filter—An FPGA based approach for impulse noise suppression", In Computer, Communication, Control and Information Technology (C3IT), vol.5, no. 9, pp. 1-6, 2015.

[14] Chean, Ren-Der, Pei-Yin Chean, and Chun-HesienYeh., "A low-power architecture for the design of a one-dimensional median filter", IEEE Transactions on Circuits and Systems II:vol.62, no. 3, pp. 266-270, 2015.

[15] Thirilogaasundari, V., and S. Agattha Janet., "Fuzzy based Salt and Pepper noise removal using adaptive switching median filter", Procedia Engineering, vol.38, no. 4, pp. 2858-2865, 2012.