ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 3, March 2017

Improving Median Filter Using APC to Handle Signed Integer

Hasna A.E[1], Geethu T George[2]

P.G. Student, Department of Electronics and Communication Engineering, Indira Gandhi Institute of Engineering and Technology for Women, Nellikuzhy, Kerala, India[1]

Associate Professor, Department of Electronics and Communication Engineering, Indira Gandhi Institute of Engineering and Technology for Women, Nellikuzhy, Kerala, India[2]

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 preservationThe time to process each of the W/B processing blocks of a median calculation method on a set of N W-bit integers is improved here by a factor of three compared with literature. The parallelism uncovered in blocks containing B-bit slices is exploited by independent accumulative parallel counters so that the median is calculated faster than any known previous method for any N,W values. The improvements to the method are discussed in the context of calculating the median for a moving set of N integers, for which a pipelined architecture is developed. An extra benefit of a smaller area for the architecture is also reported.

I. INTRODUCTION

The selection of the median, particularly its calculation, arises in many applications in computer science and statistics that extends to a variety of fields. The median, i.e.,M, of a set of integers is such that half the integers in the set is less or equal toM, and the other half of integers is greater or equal to M. For N sorted integers, the median is the integer at position P = _N/2_ or the middle position. The median filter is essentially the computation of the median for a moving set of N values within a window. The median can be calculated by sequential or parallelmethods without performing a full sort [which would take O(n log n) time], with a complexity of O(n) , althoughthese methods are not hardware amenable. Nonsorting-based methods, particularly those designed for hardware architectures, achieve the calculation in a number of steps related to the bit length of unsigned integers, i.e., W, rather than N. Of these, probably the best method compared with previous sortingand nonsorting methods takes W processing steps to find the median. Lately, two architectures have been proposed using the sorting method ; in these, the number of processingblocks is related to N, which intuitively seems area costly for the common case ofN >W. This brief improves our previous method to calculate the median on a set ofN W-bit integers in W/B processing blocks, where B is a parameter of how many bits are blocked for processing . Each block contributes B bits toward finding the median. A designer has to analyze thetradeoffs of parameters N, B, and W in order to produce awinning architecture. For instance, our previous architectureis made faster than the work in only for N >7 whenworking on slices of B = 2, 3, or 4, bits. The improvementhere makes the method faster than our previous work forany N while maintaining blocks of 2 or 3 bits for practical hardware implementations. In fact, an analysis indicates that thearchitecture presented here is faster than previously found, evenin the case of 1-bit slices (B = 1).As each block contributes B bits to the median, the key idea in this brief is to maintain aparallel accumulation to select these B bits within each block,whereas previously, this accumulation was serially computed within a block. The novel approach that led to the improvementin this brief relies on the concept of accumulative parallelcounters (APCs) . This brief starts by applying the APCconcept to a set of N nonnegative integers (or a single window) using a small value of N as an example. An APC is then applied to the case of maintaining the accumulation on a sliding window of size N, from where an architecture for calculating the median follows.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 3, March 2017

II. RELATED WORK

Lowering the dynamic power of a very-large-scale integrated circuit is an effective way to reduce the total power consumption. 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 . The token-ring architecture, adopted in the design of a mixed-timing first-in–first-out (FIFO) interface , offers the potential for low power consumption since data are immobile in the FIFO. In their designs, once data 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, is used to control the dequeuing of old data and queuing of new data at the same time. All the token registers form a token ring, which is a succession of nodes interconnected in a circular manner. The token-ring architecture with exactly one token will be adopted in our design for lowering the power consumption of a onedimension (1-D) median filter. 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 ofarchitectures for hardware design, i.e., word-level architecture and bit-level architecture. In the word-level architecture, the input samples are sequentially processed word by word, and the bits of the sample are processed in parallel the contrary, the bit-level architecture processes the samples inparallel and the bits of the incoming samples are sequentiallyprocessed. In this brief, the word-level architecturewill be adopted in the design of a low-power medianfilter for practical use.The median of a set of samples in the word-level sortingnetwork is often computed by first sorting the input samples andthen selecting the middle value . In their methods, theinput samples are sequentially processed word by word, and theincoming sample is inserted into the correct rank in two steps. In the first step, the oldest sample is removed from the window by moving some of the stored samples to the left. In the secondstep, 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 is that these two steps are separately performed intwo clock cycles in , whereas in , it takes only one cycle.In both of their methods, however, some of the stored sampleshave to be shifted left or right, depending on their values when a new input sample enters the window. For some applications that require a larger sample width, more signal transitions in thecircuit are needed; i.e., more dynamic power will be consumed. To conquer this problem, a new median filter architecture targeting low power consumption is proposed. Instead of sorting the samples physically in the window, the stored samples are kept immobile there. Only the rank of each sample, whichuses fewer bits, has to be updated at each new cycle when an input sample enters the window. Since our architecture is implemented as a two-stage pipeline, the median output, which is the sample with median rank, will also be generated at each cycle.The improvement in power consumption is achieved by utilizing a token ring in our architecture. Since the stored samples in the window are immobile, our architecture is suitable for low-power applications.

III.APCS

An APC is defined as an l-bit register that is updated by the sum of the previous contents and its r 1-bit inputs For instance, for a 3-bit register with a current value of 3 and four 1-bit input vector values of [0, 1, 1, 0], the registervalue is incremented by 2 and is thus updated to 5. This can be considered as if the number of ones in the 1-bit input was accumulated. An APC circuit with r 1-bit inputs is arrangedin such a way that the delay to perform its operation in termsof full/half adders using an l-bit ripple-carry adder is given bylog2 r+ *1.* The impact that this result hasfor the median architecture in this brief will be discussed laterin the timing analysis.Our median calculation method slices each W-bit data itemby B bits to arrange for W/B processing blocks. Within eachblock, the accumulation of the slices of bits is kept using anarray of APC registers. Consequently, within a block, a numberof 2B APC registers are maintained, with the first register beingof an r = 1 1-bit input, the second being of r = 2 1-bit inputs,and the last register being of r = 2B 1-bit inputs. In general,given the qi of an r = 2B 1-bit input, an array of 2B APCs, i.e.,Ai= $\_r-1_{i=0}$ q_i, is arranged for processing per block. For N dataitems within a window, each APC register A_i is of l = log2 Nbits. Before an example is presented, it is worth recalling howto generate a 1-bit input vector of length r = 2_B taking B-bit slices from the data items.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 3, March 2017

IV.EXAMPLE

Consider a data set of N =9 integers, i.e., x_j={3, 1, 29, 21,16, 9, 11, 19, 17}, with each of W =5 bits (labeled as [4 : 0]).Note that P =_N/2_=5. Using all the five bits in the representationof each integer (and applying a qubit tensor) requiresa full 5-to-32 binary decoder generating a bit vector of length$r$=32 bits. Performing the 5-to-32 binary decoding foreach integer in the set (and ORing into a bit vector of size 32,With all the 32-bit positions initially in zero), we generate the bitmapping presented in Table I. The binary decoding produces anindirect ordering of the integers in the set, and then, the mediancan be directly taken as 16_{10} since it is the middle position ofthe nine integers in the 32-bit vector q (or the P =5 positionof the nine in the vector). However, the growth of input size$W$ in bits and other nuisances (such as repeated integers in theset) make this full binary decoding approach impractical to beused as a direct method for computing the median, at least forsizes$W$>8 . Nevertheless, the approach can be used as theprinciple of operation when processing slices of bits taken fromthe input integers instead of taking all bits at once.

Table 1.median calculation procedure for nine integers of 5 bits each. Blocks 1 and 2 process 3 and 2 bits, respectively

Window Set		*Block 1 (3 bits)*								*Block 2*			
$(x_j)_{10}$	$(x_j)_2$	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	A_3	A_2	A_1	A_0
3	000.11	1	1	1	1	1	1	1	1	0	0	0	0
1	000.01	2	2	2	2	2	2	2	2	0	0	0	0
29	111.01	3	2	2	2	2	2	2	2	0	0	0	0
21	101.01	4	3	3	2	2	2	2	2	0	0	0	0
16	100.00	5	4	4	3	2	2	2	2	1	1	1	1
9	010.01	6	5	5	4	3	3	2	2	1	1	1	1
11	010.11	7	6	6	5	4	4	2	2	1	1	1	1
19	100.11	8	7	7	6	4	4	2	2	2	1	1	1
17	100.01	9	8	8	7	4	4	2	2	3	2	2	1
		1	1	1	1	0	0	0	0	1	1	1	1
		$P = 5; A_i \geq P$								$P = 1; A_i \geq P$			

For illustration, let us partition each x_j[4 : 0] into two blocks of bits as {x_j[4 : 2], x_j[1 : 0]}, i.e., a first block with B = 3bits (Block 1) and a second block with B = 2 bits (Block 2). For Block 1, eight (2^B = 23 = 8) APC registers A_i are maintained, with i = 0, . . . , 7 (and each wide enough to accumulate a count of up to N = 9). The 3-bit slices on all x_j[4 : 2] are first processed by Block 1, and then, x_j[1 : 0] is processed by Block 2, as shown in Table II (note the dot in $(x_j)_2$). The integers are processed one at a time, and the binary decoding of the integer slice is performed on the fly into a 2^B-bit q vector (e.g., slice "000" of integer 3 in Block 1 is decoded as vector q = [0, 0, 0, 0, 0, 0, 0, 1]). From this decoded bit vector, the input to a given APC is selected as previously stated. For instance, register $A2$ has three 1-bit inputs, taking from the decoding vector bits $q2q1q0$ = "001"; thus, the $A2$ register will update to a count of 1. Therefore, in general, APC register A_i operates on i + 1 1-bit inputs, taking from decoding vector q all bits with indices 0, . . . , i. All APC registers are updated in parallel for each integer slice, as shown in Table II. The running count is shown on each APC after the slice for each input integer is processed. After all nine integers are processed by Block $A7$,$A6$, . . . , $A1$,$A0$ have counts of 9, 8, 8, 7, 4, 4, 2, and 2, respectively. This is the count after the slice for integer 1710 (the last in the input window) is processed.for when $A_i \geq P$; this comparison is parallel. For Block 1, this comparison resolves to the bit vector "11110000" using P = 5 (as shown at the bottom Tabl). Applying priority encoding [9] to this vector (with priority from right to left) gives index i = 4 that corresponds tothe column under APC $A4$. This index corresponds to slices of 3 bits with values of "100." Thus, the threeMSBs of the median are found as M [4:2] = "100." From the nine input integers in the window, only integers 16, 19, and 17 had their processed bit slices with values of "100," indicating that only integers 16, 19, and 17 are still median candidates (highlighted in gray in Table). Integers 3, 1, 29, 21, 9, and 11 need to get nullified so that they cannot update any A_i for Block 2. Next, Block 2 is processed. First, the position for median P is recalculated as P = 5− 4 = 1 (4 being the A value to the right under the $A4$ column,

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

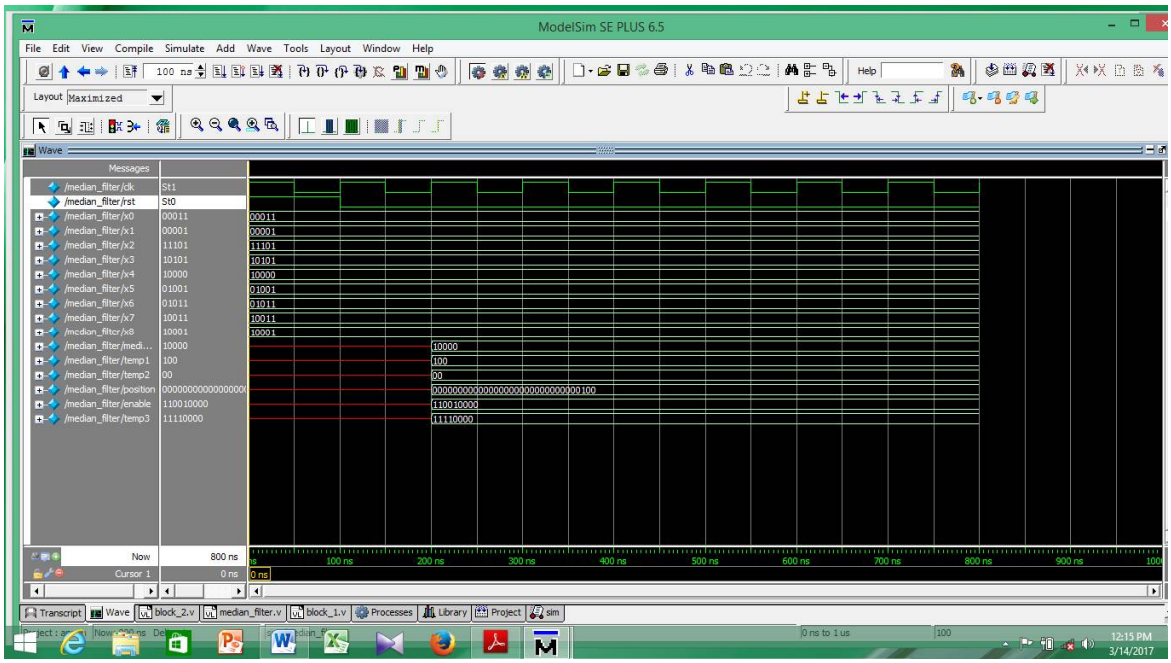
Vol. 6, Issue 3, March 2017

which is underlined in Table for Block 1). Computing *Ai* proceeds as before, i.e., on the remaining 2-bit slice for all *xj*. Condition $A \geq P$ is first satisfied for *A* under $i = 0$. The remaining two bits for the median are thus $M[1:0]$ = "00." Concatenating the results from Blocks 1 and 2 gives the median as $M = 10\ 0002 = 1610$.

V.SIGNED INTEGER EXTENSION

Signed integer extension: Let the number of positive integers be C0 and negativeintegers C1, then N = C0 + C1. If C0 >C1, set P = k + 1 – C1 initially, otherwise set P= k + 1 – C0; the method applies unmodified to the remaining W-1 bits. This method reduces by B the number of steps to calculate the medianon a set of N items compared to previously known method. The method applies to non-negative or signedintegers and also easily extends to the case of rank filtering. Fundamental in median filtering methods for noise reduction in high quality imaging, the method for calculating the median given here makes faster decisions than previous hardware algorithms in the literature. The computation within each processing block is executed faster than before for any size of blocking bits (design parameter *B*).In order to handle signed integers, count the number of negative and positive integers within a window as C_0 and C_1, respectively,

a)Figure 1. simulated output of median filter of signed integers



The figure shows the stimulated output of signed integers where this indicated the median of signed integers.. The method applies to non-negative or signed integers and also easily extends to the case of rank filtering. previous methods of median filtering are very complex methods compared with the signed extension.

VI.CONCLUSION

Fundamental in median filtering methods for noise reduction in high-quality imaging, the method for calculating the median given here makes faster decisions than previous hardware algorithms in literature. The computation within each processing block is executed faster than before for any size of blocking bits (design parameter *B*). The median on a set of *N* integers completes after *K* (typically *W/B*) processing blocks for a serial pipelined stream of*W*-bit integers with a latency of *N +KL*, with *L* being a tuning pipeline parameter for speed. It is also shown that this result holds irrespective

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 3, March 2017

of the actual values of parameter *N* or any combination of *B* and *N*. The use of full APC circuitry is required for extending calculating the median in a parallel approach of accepting more than one integer at a time in a streaming operation. The method is generic following a systematic number of steps from where different architectures and implementations can be derivedWe have implemented an automatic text detection technique from an image for Inpainting. Our algorithm successfully detects the text region from the image which consists of mixed text-picturegraphic regions. We have applied our algorithm on many images and found that it successfully detect the text region.

REFERENCES

[1] S. G. Akl, The Design and Analysis of Parallel Algorithms. Englewood Cliffs, NJ, USA: Prentice-Hall, , pp. 39–58.,1989.
[2] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithms. Cambridge, U.K.: MIT Press, 2003.
[3] D. Prokin and M. Prokin, "Low hardware complexity pipelined rank filter," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 57, no. 6, pp. 446–450, Jun. 2010.
[4] R. D. Chen, P. Y. Chen, and C. H. Yeh, "Design of an area-efficient onedimensional median filter," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 60, no. 10, pp. 662–666, Oct. 2013.
[5] R. D. Chen, P. Y. Chen, and C. H. Yeh, "A low-power architecture for the design of a one-dimensional median filter," IEEE Trans. Circuits Syst. II,Exp. Briefs, vol. 62, no. 3, pp. 266–270, Mar. 2015.
[6] J. Cadenas, G. M. Megson, R. S. Sherratt, and P. Huerta, "Fast median calculation method," Electron. Lett., vol. 48, no. 10, pp. 558–560, May 2012.
[7] B. Parhami and C. H. Yeh, "Accumulative parallel counters," in Proc. 23rd Asilomar Conf. Signals, Syst., Comput., Seattle, WA, USA, pp. 513–516,1999.
[8] J. F. Wakerly, Digital Design: Principles and Practices, 4th ed. Upper Saddle River, NJ, USA: Prentice-Hall, Ch. 6,2006.
[9] Q. Gan, J. M. P. Langlois, and Y. Savaria, "Parallel array histogram architecture for embedded implementations," Electron Lett., vol. 49, no. 2, pp. 99–101, Jan. 2013