Hardware Design of Pixel Sorter for Rank Order Filters for Applications in Real-time Systems

Ajay Pratap[1], Sanjeev Kumar[2] and Dibyayan Das Sharma[3]

**[1] Dept. of Computer Science & Engineering, JCDM College of Engineering,
Sirsa, Haryana, India**
drajaypratap@gmail.com

**[2, 3] Dept. of Electronics & Communication Engineering, Delhi College of Technology & Management,
Palwal, Haryana, India**
skumardctm@gmail.com, dibyayandsharma@gmail.com

Abstract

Digital Image Processing (DIP) is a technology that has widespread applications. However, the use of images for different applications requires efficient noise removal, image enhancement and image restoration. Rank order filtering is a non-linear method by which these tasks are accomplished. These filters can remove noise from an image while preserving edge information. Rank order filters work by sorting a window of input pixels and selecting the input with a certain rank as output pixel. In this paper, FPGA based hardware design of an efficient pixel sorter is presented which can be used for rank order filters and implemented in real-time for image processing systems. The hardware has been tested with sample grayscale images and its performance has been evaluated. Lastly, an analysis has been done on the use of the proposed hardware for image based biometric authentication systems.

Keywords: *Digital Image Processing, Rank order filters, FPGA, Image Authentication System, Biometrics, Digital System Design, Hardware Software Co-simulation.*

1. Introduction

Manipulation of images using computers is referred to as Digital Image Processing (DIP) and as a technology has widespread applications. Sometimes, the images are also processed using specialized hardware mainly for high speed operation suitable for real-time applications. However, the use of images for different applications requires efficient noise removal, image enhancement and image restoration [1] [2].

Rank order filters are non-linear filters which are used for the above mentioned purposes. It can be used to remove noise from the images while still preserving edge information which is impossible for linear filters to achieve [1]. This is the reason why rank order filters are increasingly being used in speech and image processing applications.

Nevertheless, there is one drawback of rank order filters which is its processing time. The processing time of rank order filters is quite large which is mainly dependent on the complexity and implementation of the sorting algorithm used.

In this paper, we have presented the FPGA based hardware realization of a sorting scheme and have called it Pixel Sorter. The performance of the designed hardware is tested with sample grayscale images for three filters viz. median, maximum and minimum and is compared with software based filters. The hardware has been designed using Impulse C and have been synthesized in Xilinx's Virtex 6 FPGA.

We have also analyzed the possible implementation of the designed hardware within an image based biometric authentication system to highlight the feasibility of the use of this hardware for real-time applications.

2. Review of Rank Order Filters

A device that computes the *n*th order value of a set of signals arranged usually in ascending order is called a rank order filter [3] [4]. However, major interest in signal processing activity lies in finding out the maximum, minimum and median values of a set of signals. Accordingly, the filters are referred to as maximum filter, minimum filter and median filter respectively. An important role of median filters in image processing is in the removal of impulse noise.

In computation of the rank of a signal, sorting of the input signals is a mandatory task. Typically, an odd number of signal values are sorted and the output is the median value, maximum value or the minimum value depending on the application. In Fig. 1, a diagrammatic representation of the working of a rank order filter is shown.

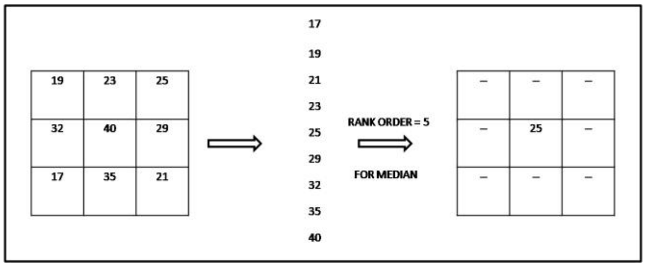

Fig. 1 Working of Rank Order Filter for Rank 5 (Median)

The filter works by analyzing a neighbourhood of pixels around an origin pixel for every valid pixel in an image. Often a 3X3 window is chosen as shown in Fig. 1, although window of any size can be chosen. However, the complexity of computation grows with the size of the window. All the pixels in the window are arranged in ascending order and then the origin pixel (40) is replaced with the pixel value as obtained from the rank order filter (25). This operation is performed for all the pixels in an image.

3. The Merge Sort Algorithm

As evident from above, sorting the pixels in a window is a key step in the operation of a rank order filter. Generally, rank order filtering has always been used in image processing through software which is run on general purpose hardware such as a PC. But, the process is quite slow and not efficient enough to be used in real-time applications, hence the need of having dedicated hardware implementation. Consequently, the hardware implementation of rank order filters both in analog and digital domain have been researched in the past [5 – 12]. However, the present work focuses on developing the hardware in a manner that makes it suitable for implementation in a real-time environment such as in a digital biometric image based authentication system.

For implementation of a practical sorting scheme, it has been realized that oblivious sorting algorithms are more suitable [13]. Oblivious sorting algorithms always perform the same number of operations in the same order regardless of the data that has to be sorted. Furthermore, the advantage of having oblivious sorting algorithm for hardware implementation lies in reduced circuitry as there is no overhead in the form of control logic.

The algorithm chosen for the Pixel Sorter is the Merge Sort algorithm which is an oblivious sorting scheme. The choice of this algorithm is basically due to its robust time complexity and quite manageable space complexity if properly designed.

In Fig. 2 and 3, the algorithm is represented with a sample array of data.

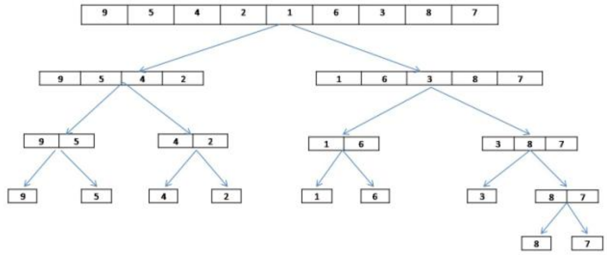

Fig. 2 Splitting of an array of data in groups of two

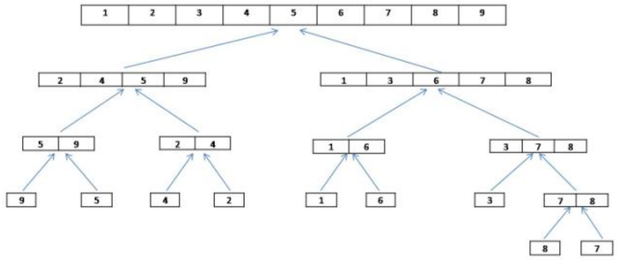

Fig. 3 Merge operation leading to the sorted array

The algorithm works by recursively splitting the array into two parts until only single elements are left which are shown above as the end nodes of the tree. Once the splitting is over, the data elements are merged successively after comparison and in this way sorting is achieved level by level until the final array is fully sorted.

3. Hardware Implementation

The above merge sort algorithm has been implemented in hardware to sort an array of 9 elements which are the contents of a 3X3 window sliding over each pixel in a target image. The pixel values are in the range of 0 to 255 as in a typical grayscale image. The splitting operation as shown in Fig. 2 is carried out by the hardware structure as shown in Fig. 4.

Starting with the original unsorted array consisting of 9 elements, the array is divided into two parts by passing it through the (N+1)/2 splitter. This splitter divides the 9 element array into two arrays one being 5 element wide and the other 4 element wide.

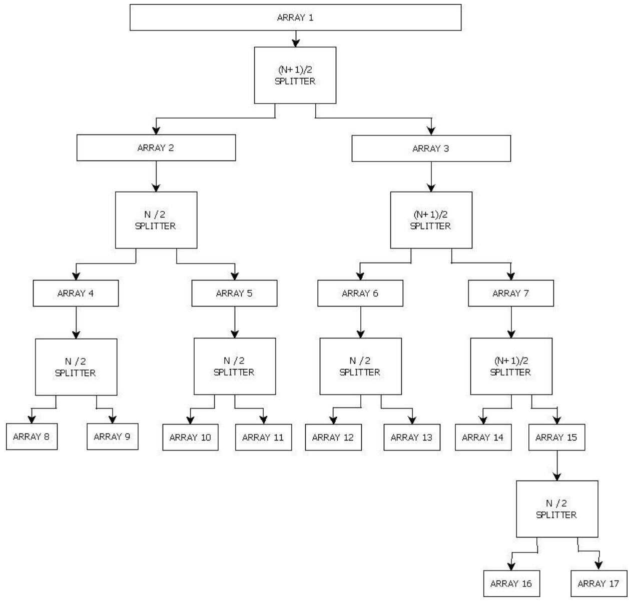

Fig. 4 Hardware structure of the splitter

These two sub-arrays are then again passed through (N/2) and (N+1)/2 splitters to obtain smaller sub-arrays and in this manner the process is continued until we have the individual elements of the original array which by definition are already sorted.

The individual elements are then merged together repeatedly after comparison to form sorted arrays until finally we have the final sorted array with us. The hardware structure of the merging operation is shown in Fig. 6.

4. Simulation Results

The proposed hardware has been designed using Impulse C within the CoDeveloper 3.60 IDE and implemented in Xilinx Virtex6 FPGA. Impulse C is a subset of standard ANSI C language which is specifically used for Hardware Software co-design. It is basically used for embedded and high performance computing applications. The greatest advantage in designing DSP hardware using Impulse C is the fine grain parallelism that it provides by virtue of system level pipelining and instruction scheduling which saves the number of clock cycles for computation and thus accelerates performance [14]. Once the Impulse C code for the hardware is written, one can directly obtain the equivalent VHDL code which can be used for synthesis or use it within the CoDeveloper IDE for hardware software co-simulation. The complete design flow using Impulse C is shown in Fig. 5

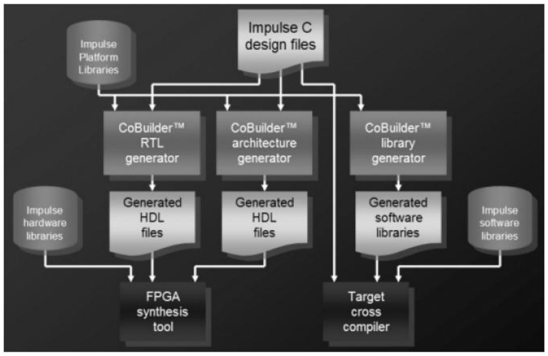
Impulse C design files
Impulse Platform Libraries
CoBuilder™ RTL generator
CoBuilder™ architecture generator
CoBuilder™ library generator
Impulse hardware libraries
Generated HDL files
Generated HDL files
Generated software libraries
Impulse software libraries
FPGA synthesis tool
Target cross compiler

Fig. 5 Design flow using Impulse C

The output of the designed pixel sorter was used in a software based rank order filter to process images and the result was compared with completely software based filters. The Pixel Sorter takes about 0.209 seconds to output the sorted array for a 100 MHz clock. Depending on the chosen rank, the target pixel is then replaced with the rank pixel in the original image by simply overwriting it.

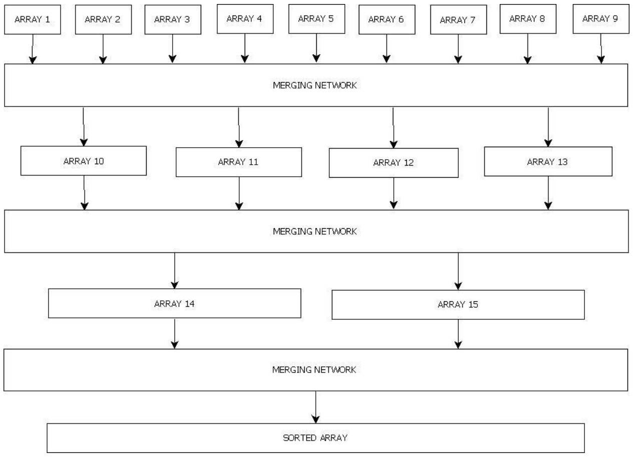

Fig. 6 Hardware structure of the merge operation

As can be seen in Fig. 7 the original image having salt and pepper noise has been processed with a median filter.

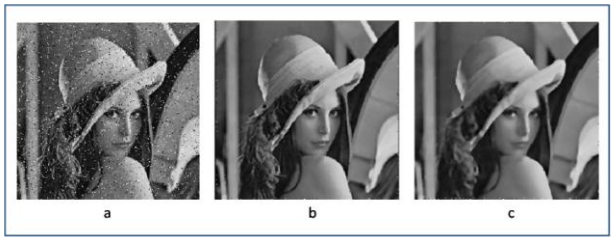

Fig. 7 Processing with Median filter

Image ‘a’ shows the corrupted image having salt and pepper noise. Image ‘b’ is the rectified image processed using MATLAB. Image ‘c’ is the rectified image that uses Pixel Sorter in the median filtering process. Image ‘c’ is very slightly blurred compared to ‘b’ which may have been due to the fixed number of bits used to represent the pixel values in the image as against the precise integer values in MATLAB. However, the number of resolution bits can always be changed according to the application after a careful analysis of number of bits v/s processing time trade-off. This ability to change system parameter according to the application is the major advantage of FPGA based approach which is referred to as reconfigurable computing.

In Fig. 7, results of the minimum filter are shown while in Fig. 8 results of maximum filter are shown.

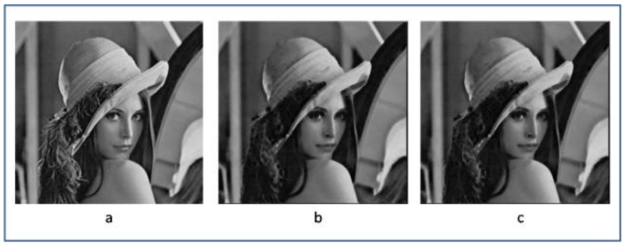

Fig. 7 Processing with Minimum filter

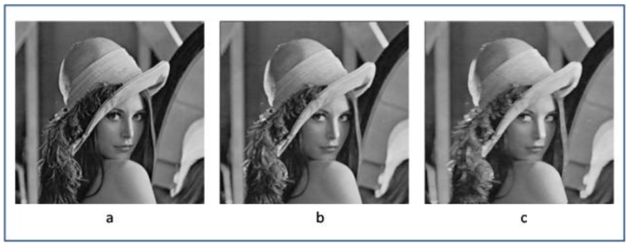

Fig. 8 Processing with Maximum filter

5. Use in Biometric Authentication System

It can be observed above that the performance of the designed hardware is quite robust and hence can be used in real-time application. We studied the possible implementation of this design in an image based biometric authentication system.

In such systems, robustness in performance is a key criterion for determining the efficiency of the biometric system. In general templates of the user are collected in database and are then used for comparison in real-time. However, for maintaining high levels of accuracy one has to use very high resolution image processing modules which escalate the cost of the biometric solution.

One possible alternative can be to use relatively modest image processing modules that have low but accepted level of resolution. These hardwares would cost less. One can significantly improve the resolution of the gathered image as pre-processing function before it is stored in database or used for real-time comparison. The design of our hardware is flexible enough to allow this modification and because of our implementation in FPGA we can do this modification in real-time as per our changing needs from time to time.

6. Conclusion

Hence the design of efficient pixel sorter for rank order filters has been achieved in this paper. We have shown the robust performance of this device through hardware software co-simulation and have discussed how it can be used for real-time applications. Future work in this direction can be done to develop a high speed pipelined comparator array that can do fast comparisons in sync with the pre-processing done by the above designed hardware. The complete biometric system can also be implemented within a single FPGA platform for better hardware integration.

References

[1] J. Astola and P. Kouosmanen, “Fundamentals of Nonlinear Digital Filtering” Boca Raton–New York: CRC Press, 1997.

[2] L. Khriji and M. Gabbouj “Vector median-rational hybrid filters for multichannel image processing”, IEEE Signal Processing Letters, July 1999, vol. 6, no. 7, pp.186-190.

[3] K. Urahama, and T. Nagao, “Direct analog rank filtering”, Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on [see also Circuits and Systems I: Regular Papers, IEEE Transactions on], 42(7):385-8, 1995

[4] M. Karaman, L. Onural, and A. Atalar, “Design and Implementation of a general purpose median filter unit in CMOS VLSI”, Solid State Circuits, IEEE Journal of, 25(2):505-13, 1990

[5] R. Mukund and A. K. Mishra, “Fast and Modular Rank Order Filter Using CMOS Technology”, IETE Journal of Research, Vol. 55, Issue 5, pp. 201-204, 2009

[6] J. Ramirez-Angulo, C. Lackey and A. Diaz-Sanchez, “Compact continuous time analog rank order filter implementation in CMOS technology”, Circuits and Systems, 2002. ISCAS 2002, IEEE International Symposium on, Vol. 5, 2002

[7] R. G. Carvajal, J. Ramirez-Angulo, G. O. Ducoudray and A. J. Lopez-Martin, “High Speed High Precision CMOS Analog

Rank Order Filter with O(n) complexity", Solid State Circuits, IEEE Journal of, 40(6):1238 – 48 2005

[8] Y. C. Hung and B. D. Liu, "A 1.2 V Rail-to-Rail Analog CMOS Rank Order Filter with k-WTA Capability", Analog Integrated Circuits and Signal Processing, 32(3):219-30, 2002

[9] A. Boudabous et. al. "Efficient Architecture and Implementation of Vector Median Filter in Co-Design Context", RADIOENGINEERING, VOL. 16, NO. 3, SEPTEMBER 2007, pp. 113-119

[10] K. Oflazer, "Design and Implementation of a Single-Chip 1-D Median Filter", IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. ASSP-31, pp. 1164–1168, Oct. 1983.

[11] Chen-Yi Lee, Po-Wen Hsieh, and Jer-Min Tsai, "High-Speed Median Filter Designs Using Shiftable Content-Adressable Memory", IEEE Transactions on Circuits and Systems for Video Technology, vol. 4, pp. 544–549, Dec. 1994.

[12] C.-T. Chen, L.-G. Chen, and J.-H. Hsiao, "VLSI Implementation of a Selective Median Filter", IEEE Transactions on Consumer Electronics, vol. 42, pp. 33–42, Feb. 1996.

[13] M. van der Horst and R. Mak, "Multi-dimensional parallel rank order filtering", Signal Processing Systems, 2007 IEEE Workshop on Oct. 2007, pp. 627–632.

[14] D. Pellerin and S. Thibault, "Practical FPGA Programming in C", Prentice Hall, April 2005.

Dr. Ajay Pratap is a distinguished professor in the Department of Computer Science & Engineering at the JCDM College of Engineering, Sirsa. He has a vast experience of research and teaching at the undergraduate and postgraduate level. He has also published numerous research papers in prestigious national and international journals. His area of research is mathematical techniques in signal processing and theoretical computer science and has published many important results.

Sanjeev Kumar is currently HOD of the Department of Electronics & Communication Engineering at Delhi College of Technology & Management affiliated to M. D. University Rohtak. His area of expertise is in Embedded System Design and has developed and designed several industrial and academic projects. He has also mentored several student embedded system projects that have won several awards in technical fests. He has authored 4 books on wireless communication, microprocessors and embedded systems.

Dibyayan Das Sharma is a 3rd year undergraduate student of the Department of Electronics & Communication Engineering at the Delhi College of Technology & Management affiliated to M.D. University Rohtak. He has published a couple of research papers in national and international journals and has presented few papers in national seminars and conferences. His research interests include hardware realization of DSP algorithms for novel and innovative applications, cryptography and image based steganography techniques.