VLSI DESIGN OF A PARALLEL ARCHITECTURE 2-D RANK ORDER FILTER

R. Roncella, R. Saletti, G. Savoia
Dipartimento di Ingegneria dell'Informazione:
Elettronica, Informatica, Telecomunicazioni,
Università di Pisa,
Via Diotisalvi 2, 56126 Pisa (Italy)
tel: +39-50-568511; fax: +39-50-568522
E-mail: roncella@iet.unipi.it

ABSTRACT

A VLSI parallel architecture implementing a new algorithm for 2-D rank order filtering, based on repeated maximum finding operations, is presented in this paper, and the design of a programmable demonstrator chip realised in standard-cell 1 µm CMOS technology is described. The chip has programmable window size and selectable rank, it can work with unitary throughput at 25 MHz, in the worst case, and its area is 7×5.5 mm^2.

1 INTRODUCTION

Rank order filtering is a non linear operation which consists in selecting the element of a specified rank among a set of N samples. In the 2-D case the samples belong to a rectangular window which is moved over an array of elements. As the window moves, the set of its elements is updated including the samples entering the window and discarding those leaving it. The rank order filter has to store the window, update it when a new column arrives, extract the element of a specified rank and use it to replace the element in the centre of the window.

Median, maximum and minimum are the most used ranks in application such as digital signal processing and image processing. The median filter is very popular for its property of removing impulse noise preserving the edges, while the filters selecting the maximum and the minimum are used in applications such as robot vision and pattern recognition for enlarging or contracting regions and lines of the images.

In this work we present an algorithm for 2-D rank order filtering, based on repeated maximum finding operations on a set of values, and a VLSI implementation of it by means of a parallel architecture; finally, we describe the design of a programmable demonstrator chip, realised in standard-cell 1 µm CMOS technology.

The desired rank and the window size of the filter are programmable to obtain a device suitable to different applications, especially for real time operations permitted by the pipeline employed in the design.

2 THE ALGORITHM

A number of algorithms for rank order filtering and their implementations can be found in the literature. They are based on different methods: insert-delete-update approach [1]; sorting networks [2]; level histogram [3]; threshold decomposition [4]; rank updating [5]; bit-serial approach [6]-[9]; max-min method [10].

The algorithm proposed here is based on repeated maximum finding operations. It allows the extraction of any ranked element among $N = RC$ samples belonging to a $R \times C$ window, where R and C are the numbers (odd) of rows and columns of the window, and it is suited to a VLSI implementation. Let us now refer to the calculation of the median.

The basic idea is to obtain the median of N samples in $(N + 1)/2$ steps, by selecting and resetting the largest element, at each step. In this way, all the elements between the maximum and the median are calculated in order. Since the sample extracted at the first step is reset, the maximum found on the modified set at the following step is the $(N - 1)$-th ranked number, which is reset in its turn. The algorithm is repeated up to step $(N + 1)/2$, when the $(N + 1)/2$-th ranked sample (i.e. the median) is obtained.

Since the maximum is calculated on N samples for $(N + 1)/2$ times, a realisation of this algorithm with an array of processors should present a complexity proportional to N^2.

The complexity of the algorithm can be reduced by keeping the columns ordered and taking advantage of the overlap between consecutive windows. The final algorithm consists of the following steps: 1) the new data column entering the window is sorted in ascending order and the matrix is updated by including the new ordered column and discarding the oldest one; 2) the maximum of the top row elements of the matrix is extracted; 3) one and only one column containing the maximum is shifted up; 4) the procedure is repeated from step 2) for $(N + 1)/2$ times; 5) the last iteration returns the element of rank $(N + 1)/2$ while the other ranked values are obtained at the preceding steps.

3 THE FILTER ARCHITECTURE

The algorithm proposed can be implemented by means of an array of processors as shown in Fig. 1, for a 3 × 3 window.

In the general case, for a window of size $R \times C$, the architecture implementing the algorithm contains a sorting network for the R elements of the new column, a shift register to store the C columns of the window, a row of processors which calculate the maximum on the C top row elements and shift up the selected column. Each processor executes a step of the algorithm: it calculates the maximum on the C samples at its input and selects the column to be shifted up in the block $\mathbf{P}_\mathbf{i}$, whereas it makes the shift executed in the proper block **S**. The modified columns then enter the next stage.

The sorting of the new column can be achieved by any method, but a solution based on repeated maximum searches allows a more regular VLSI design, since identical blocks are used.

In this case, the fundamental block of the entire structure is a processor calculating the maximum on C samples, which can be realised with a bit-serial approach. The maximum is extracted by inspecting the i-th bits of the samples and discarding when possible the elements less than it, by resetting the relevant bit of a mask register. When the maximum is extracted, the bits of the mask register corresponding to that value, will be set. Therefore, the content of this register can be used for selecting the column to be shifted.

The proposed architecture has been compared with two networks appeared in the literature [2], by evaluating the number of equivalent gates (2 input AND/OR), and the results are reported in Tab. 1. The last column of the table shows that the architecture described here is better than both the networks with which it has been compared. It is worth noting that data concerning these networks only refer to the calculation of the median, while our solution makes available all the elements with rank between $(N + 1)/2$ and N.

Moreover, all the elements with rank lesser than the median can be calculated in place of the elements with rank greater than it. In fact, it is sufficient to invert bit to bit the inputs and the outputs of the filter to obtain all the ranks between 1 and $(N + 1)/2$.

Finally, the size of the window can be changed by properly initialising the mask register of the blocks calculating the maximum. Therefore, the proposed architecture can implement a programmable rank order filter by adding a few other cells.

4 THE CIRCUIT DESIGN

The demonstrator chip realises a programmable rank

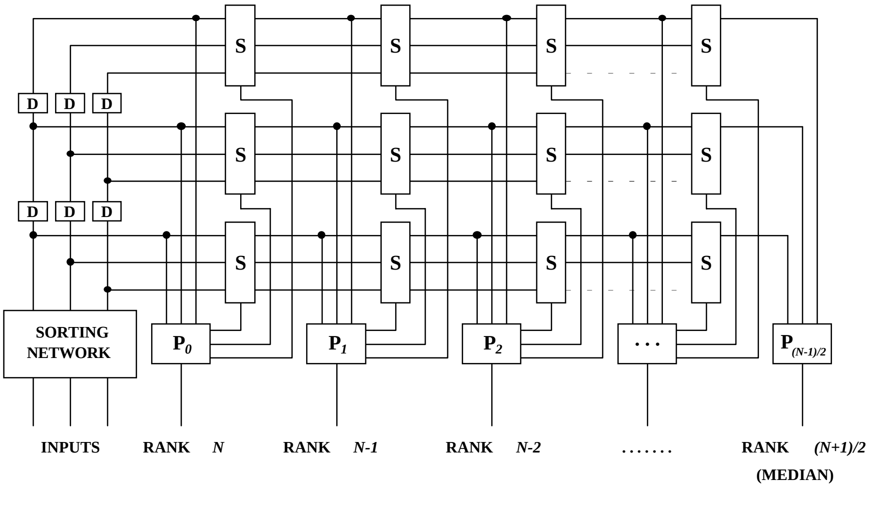

Fig. 1: Array implementing the algorithm for a 3 × 3 window. **D** are delays, **S** shift up the columns selected by $\mathbf{P}_\mathbf{i}$, the maximum finding processors.

Window size ($R \times C$)	Solution	8 bit comp		8 bit mux		Max finder		Other gates	Gate count
		Q	G	Q	G	Q	G	G	
3×3	A	22	56	30	24			464	2416
	B	16	56	22	24			352	1776
	Ours			27	24	8	88		1352
5×5	A	113	56	226	24				11752
	B								n.a.
	Ours			250	24	18	152		8736

Tab. 1: Comparison in term of estimated number of equivalent gates between our architecture and the two networks A and B reported in [2]. Q indicates the number of units; G the number of gates per unit.

order filter on 8 bit samples belonging to a window with programmable size up to 5 rows and 5 columns.

The circuit design is carried out according to the architecture described above. The only important difference is the insertion of pipeline registers between each processor to improve performance. In this way we have obtained a device with unitary throughput and worst case working speed of 25 MHz. The pipeline is inserted at the word level, but it is important to state that it could be pushed at the bit level without modifications to the chip architecture. In this case a bit level systolic array is obtained.

The chip also contains a programming unit which sets the filter parameters. The detailed structure of the filter is shown in Fig. 2.

The input stage is the sorting network and a barrier of XORs which invert the input, when the control bit *S* is set for low ranks evaluation.

The 5 samples of the entering column are sorted and stored in the shift register which contains the 5 columns of the actual window.

The chain of 13 max finders, together with the SHIFT&REG blocks, processes these columns according to the algorithm.

Each SHIFT&REG block shifts up the input elements when necessary and stores them in a pipeline register. The outputs of the processors feed three-state gates short circuited together to reduce the output pad number. The element with the desired rank is selected by the *CTRL* word.

The output of the filter passes through another barrier of XORs which completes the function of that in the input stage.

CTRL and *S* are produced by the programming unit according to the desired values of the rank and window dimensions. This unit also provides the proper initialisation of the mask register as well. The first 4 lines of the input bus are used for filter programming, according to the value of an external programming pin.

The entire structure, from the sorting network to the processors, is based on a block calculating the maximum of 5 elements with a bit-serial approach. The *i*-th bit of the maximum y_i is evaluated in the elemental network shown in Fig. 3, which also updates the mask register *m*, by inspecting the *i*-th bits b_{ij} of the input samples and using the previous content of the mask register *r*.

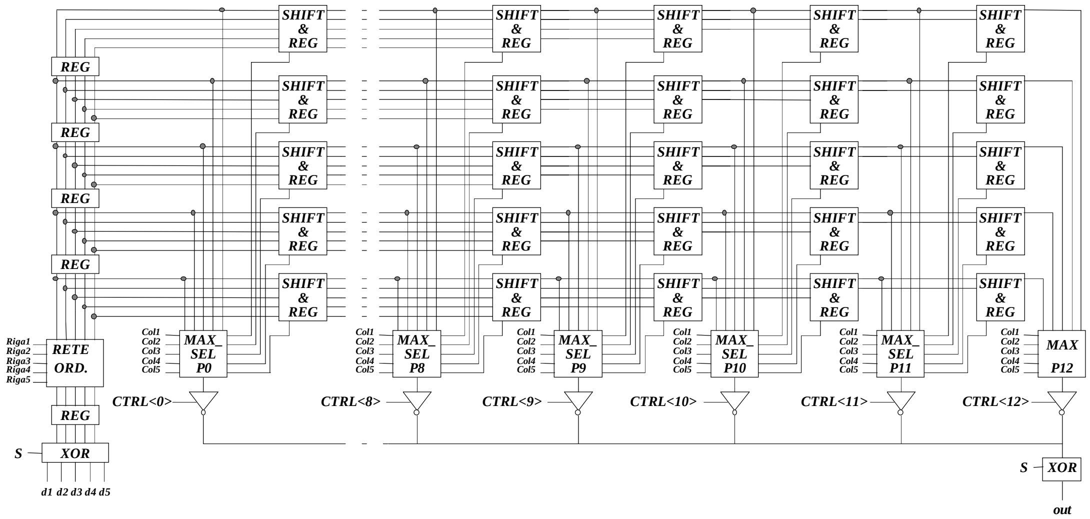

Fig. 2: Block diagram of the chip architecture.

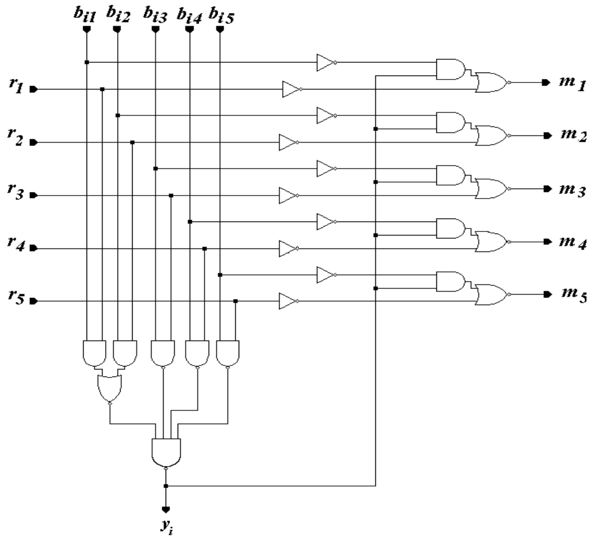

Fig. 3: Elemental block for the calculation of the *i*-th bit of the maximum and the updating of the mask register.

5 CONCLUSIONS

An algorithm for 2-D rank order filtering based on repeated maximum finding operations has been described in this work and a possible architecture for its implementation has been defined. The architecture mainly consists of a row of maximum finding processors and allows a very regular and structured VLSI design. Pipeline registers are easily inserted to improve the performance and the results obtained on a demonstrator chip, designed with a standard-cell 1 µm CMOS technology for a programmable size up to a 5 × 5 window, are an area of 7 × 5.5 mm^2 and a worst case operation frequency of 25 MHz. The layout of the demonstrator chip is shown in Fig. 4.

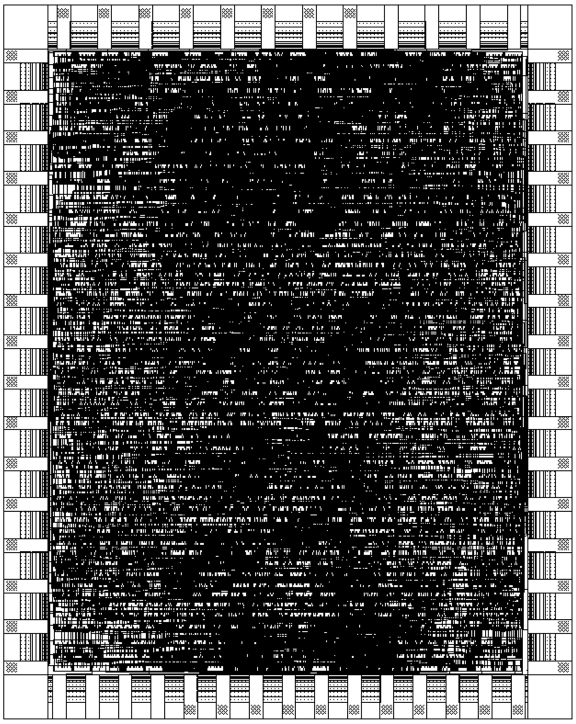

Fig. 4: Layout of the demonstrator chip.

Another interesting feature of the proposed architecture is the possibility of pushing the pipeline to the bit level without any significant change. In this case a bit level systolic architecture is obtained and a simple and regular full custom design can lead to an improved working speed and computational capability.

REFERENCES

[1] R.Roncella, R.Saletti, P.Terreni, “70-MHz 2-µm CMOS Bit-level Systolic Array Median Filter,” *JSSC*, vol. 28, pp. 530-536, May 1993.

[2] C.Chakrabarti, “Sorting Network Based Architectures for Median Filters,” *IEEE Trans. Circuits and Systems-II: Analog Digital Signal Proc.*, vol. 40, pp. 723-727, Nov. 1993.

[3] T.S.Huang, G.J.Yang, G.Y.Tang, “A Fast Two-Dimensional Median Filtering Algorithm,” *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-27, pp. 13-18, Feb. 1979.

[4] L.W.Chang, S.S.Yu, “A New Implementation of Generalized Order Statistic Filter by Threshold Decomposition,” *IEEE Trans. Signal Proc.*, vol. 40, pp. 3062-3065, Dec. 1992.

[5] C.Chakrabarti, “High Sample Rate Array Architectures for Median Filters,” *IEEE Trans. Signal Proc.*, vol. 42, pp. 707-712, Mar. 1994.

[6] B.K.Kar, D.K.Pradhan, “A New Algorithm for Order Statistic and Sorting”, *IEEE Trans. Signal Proc.*, vol. 41, pp. 2688-2694, Aug. 1993.

[7] K.Chen, “Realizations of a Class of Non-Linear Filters Using a Bit-Serial Approach,” ISCAS '88, pp. 1749-1752.

[8] L.W.Chang, J.H.Lin, “A Bit-Level Systolic Array for Median Filter”, *IEEE Trans. Signal Proc.*, vol. 40, pp. 2079-2083, Aug. 1992.

[9] C.L.Lee, C.W.Jen, “Binary Partition Algorithms and VLSI Architectures for Median and Rank Order Filtering,” *IEEE Trans. Signal Proc.*, vol. 41, pp. 2937-2942, Sep. 1993.

[10] J.P.Fitch, “Software and VLSI Algorithms for Generalized Ranked Order Filtering,” *IEEE Trans. Circuits Syst.*, vol. CAS-34, pp. 553-559, May 1987.