A Modular and Scalable Architecture for the Realization of High-Speed Programmable Rank Order Filters

İ. Hatırnaz, F. K. Gürkaynak and Y. Leblebici

Worcester Polytechnic Institute
Department of Electrical and Computer Engineering
Worcester, MA 01609-2280

Abstract

We present a new scalable architecture for the realization of fully programmable rank order filters (ROF), based on Capacitive Threshold Logic (CTL) gates. Variants of ROFs, especially median filters, are widely used in digital signal and image/video processing and image enhancement. The CTL-based realization of the majority gates used in the ROF architecture allows the filter rank and the window size to be user-programmable, using a much smaller silicon area, compared to conventional realizations of digital median filters. The proposed filter architecture is completely modular and scalable, and the circuit complexity grows only linearly with maximum window size and with word length. Detailed post-layout simulations of the ROF prototype circuit indicate that the new architecture can accommodate sampling clock rates of up to 50 MHz, corresponding to an effective data processing rate of 800 Mb/s for a filter with window size 63 and word length of 16 bits.

1 Introduction

The rank order filter (ROF) is a non-linear digital filter which determines the i-th ranking element in a given window consisting of binary encoded input words (Fig 1). Special cases of rank order filters are median, minimum and maximum filters, where the outputs are the median, the minimum and the maximum values of the input words, respectively [1]. Variants of ROFs are widely used in digital signal and image/video processing because of their non-linear characteristics. Especially, median filters have found many applications in digital image enhancement, such as reducing the high frequency and impulsive noise in digital images without the extensive blurring and edge destruction[2][3]. Other successful applications of ROFs include the smoothing of noisy pitch contours in speech signals, data compression in block truncation coding schemes, speckle noise reduction in coherent imaging systems, and preprocessing data for machine vision.

Several algorithms have been proposed for rank order filters that are based on data sorting. Although these algorithms are suitable for software implementation, they result in inefficient hardware structures, since they process the input vectors at the word level. Implementations based on stack filters have an area-time complexity of $O(n^2)$, and the hardware complexity increases very rapidly with window size (m).

In recent years, some innovative bit-serial structures for rank-order-filters have been presented, which are mostly based on majority-decision algorithms [4], [9]. Yet, the majority function is typically hard to realize using conventional Boolean building blocks, since it requires a large number of gates and a large logic depth.

Consequently, such structures suffer from speed and area limitations, especially if the window size becomes larger than 10 words. Also, most of the conventional realizations result in a fixed rank and a fixed window size, which limit the flexibility of its application.

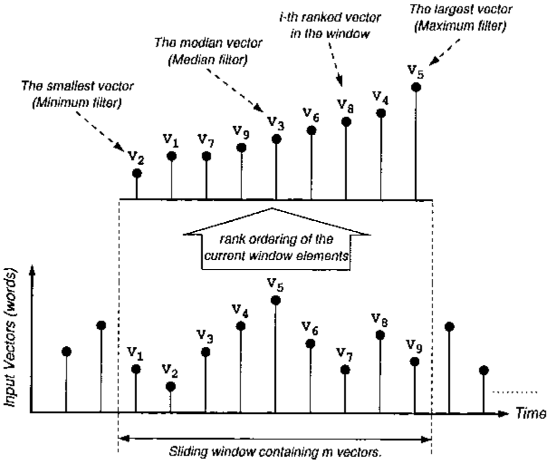

Figure 1: One-dimensional illustration of the rank-ordering process.

In this paper, we present a new architecture to realize a fully programmable ROF, based on Capacitive Threshold Logic (CTL) gates. The CTL realization of the majority gates [5] used in the ROF architecture allows the filter rank and the window size to be user-programmable, using a much smaller silicon area.

0-7803-5632-2/99/$10.00 © 1999 IEEE

The overall filter architecture is also simplified significantly, compared to conventional realizations of digital median filters [7]. The outline of a simple bit-serial algorithm for rank ordering is presented in Section 2. In Section 3, the implementation of a programmable ROF architecture is discussed. The conclusions are summarized in Section 4.

2 The Rank Ordering Algorithm

2.1 Algorithm Description

A bit-serial algorithm first proposed in [6] was chosen as the basis of the programmable rank-order filter architecture implemented in this work. In this algorithm, the problem of finding a rank-order-selection for n-bit long words is reduced to finding "n" rank-order-selections for 1-bit numbers.

The algorithm starts by processing the most significant bits (MSB) of the m=(2N + 1) words in the current window, through an m-input programmable majority gate, to yield the MSB of the desired filter output. This output is then compared with the other MSBs of the window elements. The vectors whose MSB is not equal to the filter output have their MSB propagated down by one position, replacing the less significant bits of the corresponding words. This process is continued for the following bits. Thus, any bit that is not equal to the corresponding stage output is propagated down to the lesser significant positions, until the least significant bit is processed. This process ensures that at a later stage, any number which was greater (or less than) the i-th ranked number can be identified, and the i-th ranked bit sorted out.

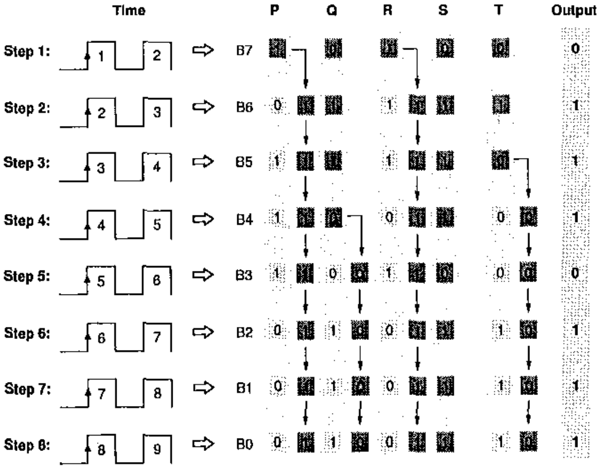

Figure 2: An illustration of the rank-ordering algorithm, for five 8-bit words.

Figure 2 shows an example where, five 8-bit words (denoted P through T with decimal values of 184, 105, 194, 117 and 75 respectively) are being rank-ordered using the algorithm described above. The window size is m=5 and the rank is r=3, indicating that the third smallest among these five numbers is being found in 8 steps. Note that the main bit-level operation at each step amounts to a majority (rank) decision among n bits of the same bit-plane. In the example, the final result after Step 8 corresponds to word S which has the decimal value of 117.

2.2 Realization of the Algorithm

The bit-serial operation flow of the algorithm described above suggests a very simple bit-level pipelined data path architecture.

The hardware implementation of the ROF algorithm consists of two main blocks:

1. The Modifier/Selector(propagator) block whose function is to store and to shift the actual data and to calculate the selector signal for the next processing block.

2. The Majority or Rank Decision block which determines the output bit as a function of n bits.

In the Modifier/Selector block, the output of the majority function is compared with the corresponding data bit, using an XNOR gate. The result of this XNOR operation is then combined (AND operation) with the select signal originating from the previous block. This provides the information if the data bit taken from the previous block is a propagating one or not. If the data bit is a propagating one, then the new select signal will be 0, indicating that this data bit will continue propagating unchanged through the following stages. Otherwise, the select signal will only depend on the result of the comparison of the filter-slice output with the current data bit. Identical 1-bit filter slices can be used in sequence (cascade configuration) in order to process input vectors of arbitrary bit-length. Thus, the filter throughput can be increased by bit-level pipelining. The modular structure of the one-bit slice described above also allows for scalable realization of the ROFs with different window sizes and word lengths.

3 Implementation of the Programmable ROF Architecture

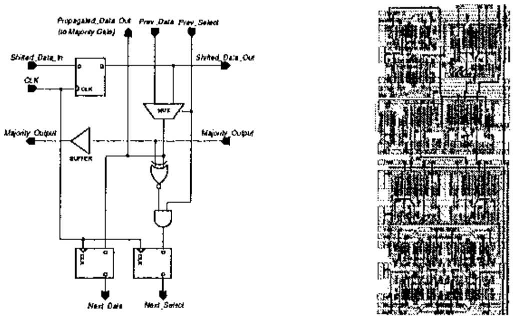

Figure 3 : Gate-level structure of a ROF cell and the corresponding layout, allowing modular expansion.

3.1 System Components

There are two main blocks in the architecture, the ROF-cell and the Majority Decision gate. By using these two blocks, a programmable rank-order filter of any window size and word-length can be realized. The word-length dictates the number of the majority decision gates, whereas the window size determines the number of ROF-cells driving one of these majority gates. The programmable majority decision gates are realized using the capacitive threshold logic (CTL) circuit architecture presented earlier [5]. This allows simple implementation of programmable majority gates with up to 63 parallel inputs, using a very small silicon area ($625\mu m$ x $130\mu m$ for 63-bit majority gate). In comparison, a classical realization of the 63-bit majority gate would require an equivalent of 63 6-bit full-adder circuits, arranged in a network of a logic depth of 64 (synthesized from HDL description).

Figure 3 shows the ROF-cell block realization at gate level. At each positive clock edge, the corresponding select and data signals are fed to the next blocks. During a clock period, the majority gate output feeds all the ROF-cells in its corresponding bit-level. The signal flow between the ROF cells and the majority gates are shown in Figure 4. The modular architecture consisting of only two major blocks enables fully scalable construction of filter structures of arbitrary size.

3.2 Overall System Architecture

The top level block diagram of the programmable ROF design is shown in Figure 5. The architecture consists of three main blocks: input shift registers, ROF processing core, and output shift registers. To allow bit-level pipelined operation, the input bits are ordered using a staggered shift register array (Fig. 5).

The ROF core has (n·m) ROF cells where m=(2N+1) is the window size and n is the bit-length of the input words. The ROF cells processing the bits of same significance provide the necessary inputs to the corresponding Majority Decision block which determines the filter output bit of that level. This output bit is fed to the output shift registers and back to the ROF cells, to be used in determining the select and data signals which will be the inputs of the next stage.

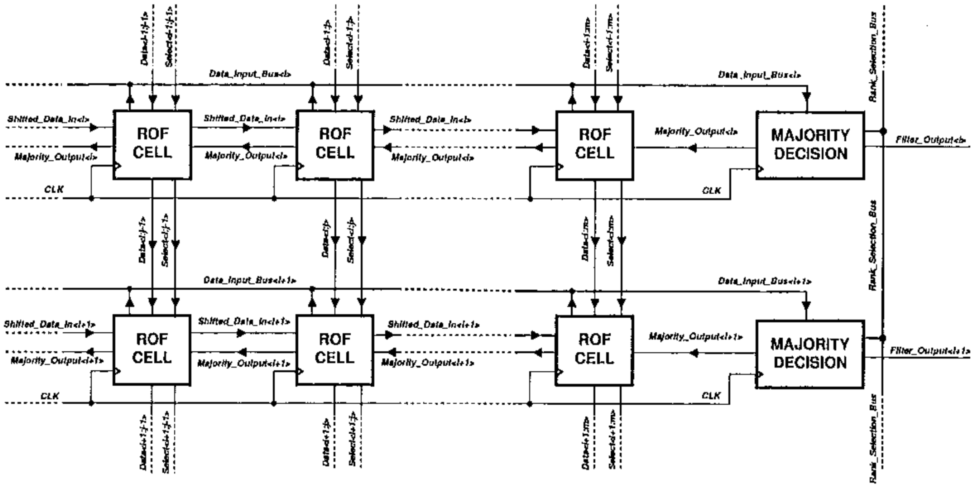

Figure 4: Detailed signal flow between modular ROF-cells and Majority Gates.

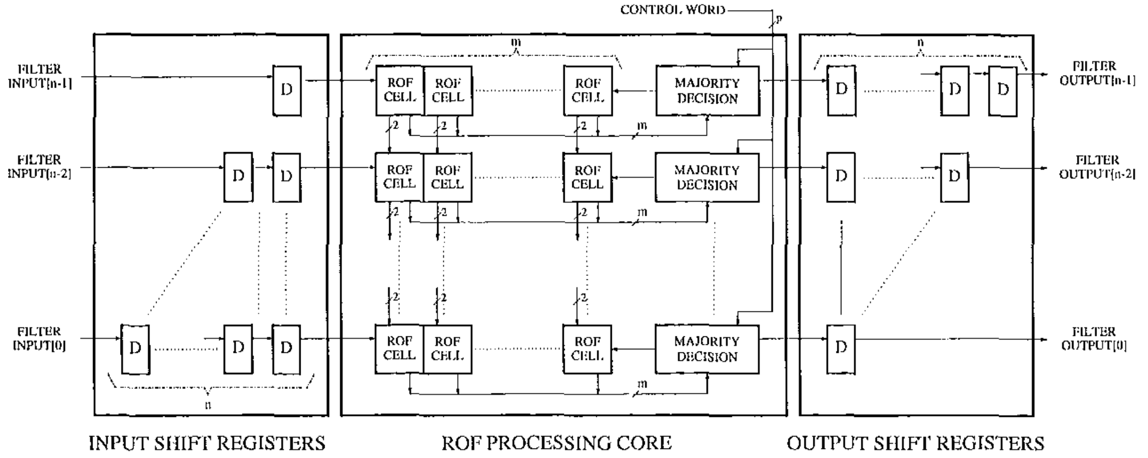

Figure 5 : The top-level architecture of a (n · m) programmable CTL based ROF. Here n denotes the bit-length of the input vectors, and m denotes the maximum window size.

The top-level layout of a programmable ROF core with a maximum window size of 63 samples and a word-length of 16 bits is shown in Fig. 6. The circuit occupies a silicon area of approximately (5 *mm* x 5 *mm*), and operate with a latency of 16 clock cycles at the clock frequency of 50 MHz, which results in an effective data rate of 800 Mbits/s. This compares very favorably with any of the existing filter architectures proposed so far [7], [8].

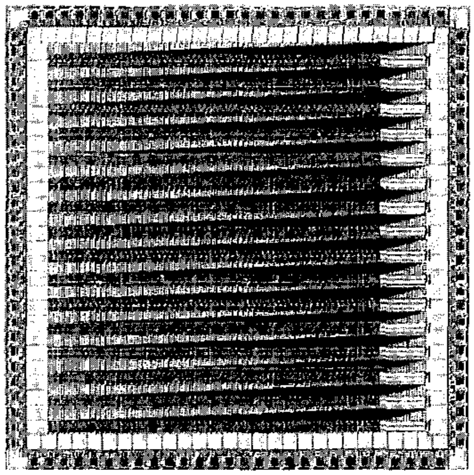

Figure 6 : Top-level layout of the ROF circuit with a window size of 63 and word-length of 16 bits. The silicon area is approximately (5 *mm* x 5 *mm*).

3.3 Advantages of the Proposed Architecture

The realization of fully programmable rank order filters has traditionally been a very challenging design problem, mainly due to the fact that the rank selection function (programmable majority function) is extremely hardware-intensive using conventional design approaches. As a result, most of the design efforts so far have either been constrained to median-only filters without any rank selection capability, and/or to relatively small window sizes [7], [8].

The CTL-based ROF architecture presented here is superior to other ROF implementations, with its following capabilities:

1. The CTL realization of the majority gates used in the ROF architecture allows the filter rank and the window size to be fully programmable, using a much smaller silicon area.
2. The rank-ordering algorithm implemented with this architecture does not require the elements of the input window to be pre-ordered, as opposed to other, stack-based ordering algorithms [7].
3. The proposed ROF has a modular architecture which enables easy expandability of the window size and bit-length of the input words without a dramatic change in performance.
4. The overall circuit complexity increases *linearly* with maximum window size (m) and with word length (n).

3.4 Simulation Results and Experimental Validation

A prototype ROF circuit has been designed and fabricated using a 0.8 micron double-poly CMOS process, to validate the main operation principles of the architecture. The prototype blocks consist of four bit-level pipeline stages, each of which contains a 63-bit programmable majority gate to handle the rank selection. To limit overall circuit complexity and to enable easier testing, each stage is designed to process a maximum window size of four samples.

The operation of the ROF architecture is demonstrated with detailed post-layout simulation in Figure 7(a). Here, the window size is 4, and the rank is selected to be 2 - meaning that the second largest input word in the sample window will have to be selected. The input words for this sample window are "0101", "0110", "0111" and "1000". The correct output, "0111" appears after 4 clock cycles. Measured vector sequences from the prototype circuit (Fig. 7(b)) confirm the operation of the circuit, with the same input pattern as in Fig. 7(a).

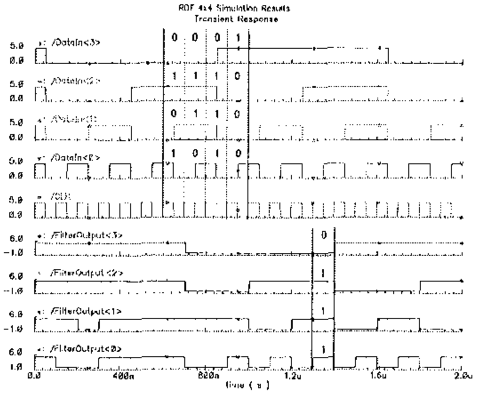

Figure 7(a): Post-layout simulation results of the ROF prototype.

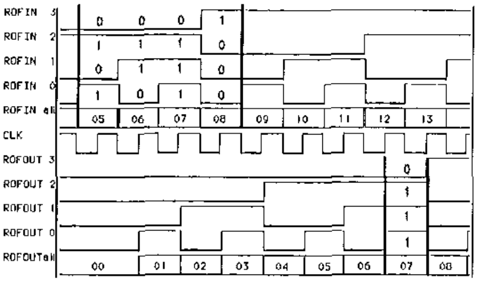

Figure 7(b): Measured output sequences of the prototype circuit.

4 Conclusion

In this paper, we have presented a new architecture for realizing a fully programmable ROF, based on the Kar-Pradhan rank ordering algorithm and Capacitive Threshold Logic (CTL) majority gates. The bit-serial realization of the rank ordering algorithm offers a simple pipelined filter architecture which is highly modular and easily expandable. The CTL realization of the majority gates used in the ROF architecture allows the filter rank and the window size to be user-programmable, resulting in a much smaller silicon area. In addition, the CTL based majority gates enable a much simpler overall filter architecture compared to conventional digital median filter realizations.

References

[1] D.S. Richards, "VLSI median filters", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp.145-152, January, 1990.

[2] J.P. Fitch, E.L. Coyle and N.C. Callagher, "Median filtering by threshold decomposition", *IEEE Trans. Acous. Speech, Signal Proc.*, vol 32, pp.1183-1188, 1984.

[3] T.S. Huang and G.J. Yang, "Median filters and their applications to image processing", *School of Elec. Eng.*, Purdue Univ., West Lafayette, IN, TR-EE 80-1, Jan. 1980.

[4] A. Gasteratos, I. Andreadis, Ph. Tsalides, "Realization of rank order filters based on majority gate", *Pattern Recognition*, vol.30, no. 9, pp 1571-1576, 1997.

[5] Y. Leblebici, F.K. Gurkaynak, D. Mlynek, "A compact 31-input programmable majority gate based on capacitive threshold logic", *in Proc. IEEE Int. ASIC Conference 1998*, pp. 281-285.

[6] B.K. Kar, D.K. Pradhan, "A new algorithm for order statistic and sorting", *IEEE Trans. on Signal Processing*, vol. 41, pp.2688-2694, August 1993.

[7] C.C. Lin, C.J. Kuo, "Fast response 2-D rank order algorithm by using max-min sorting network", *International Conference on Image Processing 1996*, Vol. 1, pp. 403-406.

[8] C. Chen, L. Chen, T. Chiueh, J. Hsiao, "An efficient pipelined VLSI implementation of rank order filter", *ISSIPNN 1994*, Vol. 2, pp. 630-633.

[9] C.L. Lee and C.W. Jen, "Bit-sliced median filter design based on majority gate", *in Proc. Ins. Elec. Eng.-G*, vol 139, pp.63-71, 1992.