

DESIGN OF A HIGH-SPEED, RECONFIGURABLE DIGITAL RANK ORDER FILTER

GEORGE J. TOSCANO¹, PRAN K. SAHA², A.H.M. ZAHIRUL ALAM³

¹EEE Dept, Fac. of Engineering, American International University Bangladesh (AIUB),
Kemal Ataturk Avenue, Dhaka-1213, Bangladesh

²EEE Dept, Fac. of Electrical and Electronic Engineering, Bangladesh University of
Engineering and Technology (BUET), Dhaka-1000, Bangladesh

³ECE Dept, Fac. of Engineering, International Islamic University Malaysia (IIUM), Jalan
Gombak, 53100 Kuala Lumpur, Malaysia.

ggtoscano@gmail.com, sahapk@eee.buet.ac.bd, zahirulalam@iiu.edu.my

ABSTRACT: A new architecture to realize a modular, high-speed, reconfigurable, digital Rank Order Filter (ROF) is presented in this paper. A bit-level algorithm by Kar and Pradhan has been modified in this work to implement the proposed ROF. Using the proposed digital rank selection circuit it is possible to find the element of a certain rank in a given sequence of N elements in each window in M steps, where M is the number of bits used in binary representation for the elements of the sequence. The size of the proposed ROF increases only linearly with the number of samples in each window to be ranked. The proposed ROF is also modular in nature, which means function of each part of the ROF is well defined and so the circuit can be easily expandable for larger window size. The proposed ROF has been implemented in FPGA and post-fit simulation results are presented in this paper. HSPICE simulation of the proposed ROF is also done for 0.18 μm CMOS process. The simulation result shows that the circuit could be operated at a clock speed of 500 MHz.

KEY WORDS: *Bit update circuit (BUC), Rank order filter (ROF).*

1. INTRODUCTION

Rank Order Filter is a nonlinear filter widely used in digital signal and image processing. It involves selecting a sample with a specified rank from a one dimension or multidimensional window of samples. Special cases of Rank Order Filters are median, minimum and maximum filters, where the outputs are the median, the minimum and the maximum value.

A number of algorithms for rank order filtering and their implementations can be found in the literature [1-8]. Rank Order Filters can be implemented in VLSI using the threshold decomposition technique [1]. However, in this case hardware complexity increases exponentially with both the resolution of the numbers and the size of data window. Therefore, implementation of filters capable of handling high-resolution numbers is not

practical. Mixed signal rank order filters [2] are more susceptible to noise compared to completely digital filters. Their throughput is also very low. Rank Order Filters designed using max-min sorting network [3] or compare-and-swap circuit [4, 5] are very large in size and the complexity or size of the filter increases rapidly as the square of number of samples. Rank Order Filters based on Positive Boolean Function suffers the problem that once the device has been designed, the rank of the filter becomes fixed. The complexity of the proposed filter increases only linearly with the number of samples in each window to be ranked. The proposed Rank Order Filter is reconfigurable, which means any rank element can be obtained as output from the filter.

2. BIT-LEVEL SELECTION ALGORITHM

A modified Rank Selection algorithm based on the bit-level algorithm reported in [9] is used to design the proposed Rank Order Filter. The flowchart of the modified Rank Selection algorithm is shown in Fig. 1 and is described below.

Let W be a window of size N . W can be expressed as $W = \{x_1, x_2, \dots, x_N\}$. Let M be the size of a sample in the sequence in bits. Then each sample x_j can be represented as:

$$x_j = b_j^{M-1} b_j^{M-2} \dots b_j^0$$

where b_j^k is the $(k+1)$ -th bit of x_j with a weight of 2^k .

An element of rank r is the r th smallest element in the window of N elements. Let $Y = y^{M-1} y^{M-2} \dots y^0$ be the r th rank-order element in W . Based on [9] a modified bit level algorithm to find Y is given below.

1. Set $k=M-1, Q_1=Q_2=\dots=Q_N=0$
2. Set $B_j^k=b_j^k$ for $j=1,2,\dots,N$
3. Find sum $S^k = \sum_{j=1}^N B_j^k$
4. If $S^k \geq N-r+1$ then
Set bit $y^k=1$
Else Set bit $y^k=0$
5. For $j=1$ to N
If $(Q_j=0 \text{ AND } y^k \neq B_j^k)$ then
Reset bits $B_j^{k-1}, B_j^{k-2}, \dots, B_j^0$ to b_j^k and Q_j to 1
Elseif $(Q_j=0 \text{ AND } y^k = B_j^k)$ then
Set $B_j^{k-1} = b_j^{k-1}$
6. Set $k = k-1$
7. Repeat steps 3 to 6 until $k < 0$
8. Output $Y = y^{M-1} y^{M-2} \dots y^0$ as rank-order element.

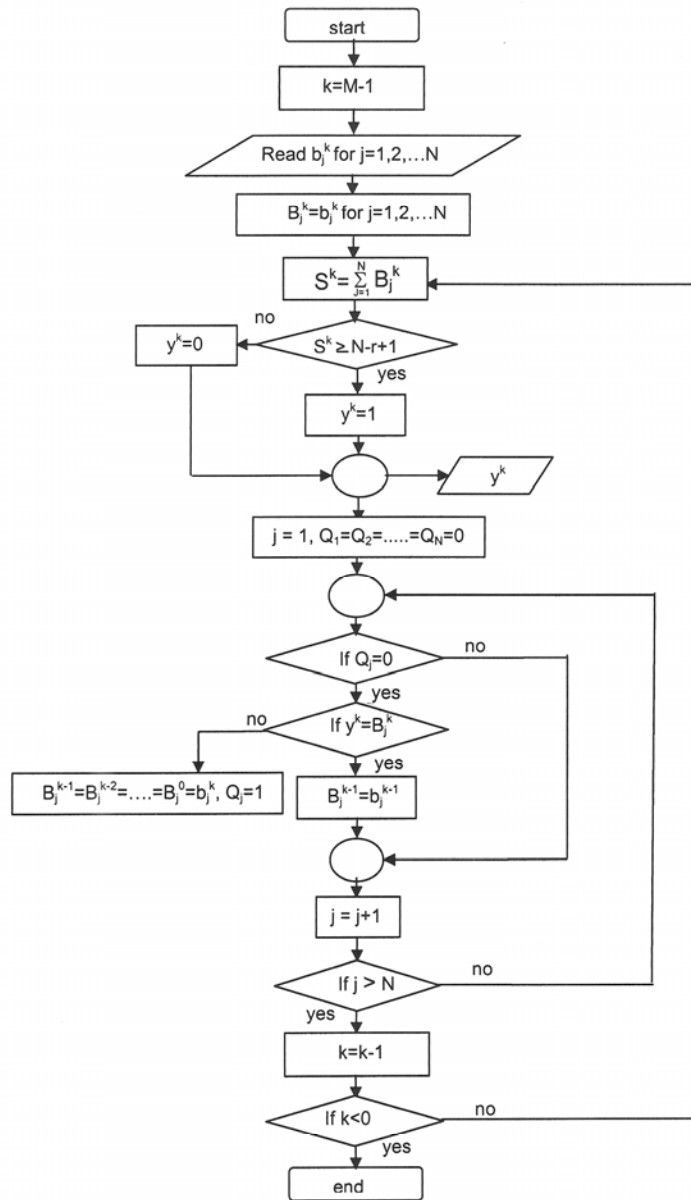


Fig. 1: Flow chart of the algorithm.

An illustration of the algorithm for $N=5, r=3, W= \{186, 105, 226, 117, 75\}$ and $M = 8$ is given in Table 1. In this case, $N-r+1$ equals '3'. During each iteration of the algorithm, the k^{th} bits of all five modified input bits are added to compute S^k . Output bit y^k is set to '1' if $S^k \geq 3$, otherwise y^k is set to '0'. Next all the bits following the k^{th} bit in each element are modified depending on the condition described in step 5. Bits affected by the condition in step 5 are shown in shaded cells in Table 1. Hence Y is found as output.

Table 1: Computation of rank-order element for $N = 5, r = 3$.

k	b_1^k (B_1^k)	b_2^k (B_2^k)	b_3^k (B_3^k)	b_4^k (B_4^k)	b_5^k (B_5^k)	$N-r+1$	$S^k = \sum_{j=1}^N B_j^k$	y^k
7(MSB)	1(1)	0(0)	1(1)	0(0)	0(0)	3	2	0
6	0(1)	1(1)	1(1)	1(1)	1(1)	3	5	1
5	1(1)	1(1)	1(1)	1(1)	0(0)	3	4	1
4	1(1)	0(0)	0(1)	1(1)	0(0)	3	3	1
3	1(1)	1(0)	0(1)	0(0)	1(0)	3	2	0
2	0(1)	0(0)	0(1)	1(1)	0(0)	3	3	1
1	1(1)	0(0)	1(1)	0(0)	1(0)	3	2	0
0(LSB)	0(1)	1(0)	0(1)	1(1)	1(0)	3	3	1
Decimal value	186	105	226	117	75			117

Now another example is given for the same data window. Let in this case the value of r to be 5. So $(N-r+1)$ equals '1'. During each iteration of the algorithm, the k^{th} bit of all five modified input bits are added to compute S^k . Output bit y^k is set to '1' if $S^k \geq 1$, otherwise y^k is set to '0'. Bits affected by the condition in step 5 are shown in shaded cells in Table 2. Hence, Y is found as output.

Another illustration of the algorithm is given in Table 3 by changing the window size, where $N=7, r=2, W=\{186, 105, 226, 117, 75, 240, 162\}$ and $M = 8$. In this case, $(N-r+1)$ equals '6'. During each iteration of the algorithm, the k^{th} bits of all seven modified input bits are added to compute S^k . Output bit y^k is set to '1' if $S^k \geq 6$, otherwise y^k is set to '0'. Bits affected by the condition in step 5 are shown in shaded cells in Table 3.

Table 2: Computation of rank-order element for $N = 5, r = 5$.

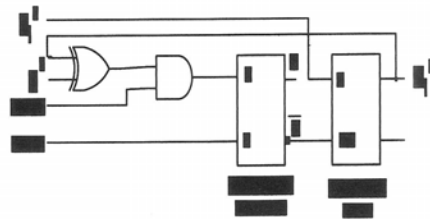
k	b_1^k (B_1^k)	b_2^k (B_2^k)	b_3^k (B_3^k)	b_4^k (B_4^k)	b_5^k (B_5^k)	N - $r+1$	$S^k = \sum_{j=1}^N B_j^k$	y^k
7(MSB)	1(1)	0(0)	1(1)	0(0)	0(0)	1	2	1
6	0(0)	1(0)	1(1)	1(0)	1(0)	1	1	1
5	1(0)	1(0)	1(1)	1(0)	0(0)	1	1	1
4	1(0)	0(0)	0(0)	1(0)	0(0)	1	0	0
3	1(0)	1(0)	0(0)	0(0)	1(0)	1	0	0
2	0(0)	0(0)	0(0)	1(0)	0(0)	1	0	0
1	1(0)	0(0)	1(1)	0(0)	1(0)	1	1	1
0(LSB)	0(0)	1(0)	0(0)	1(0)	1(0)	1	0	0
Decimal value	186	105	226	117	75			226

Table 3: Computation of rank-order element for $N = 7, r = 2$.

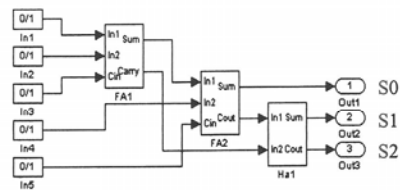
k	b_1^k (B_1^k)	b_2^k (B_2^k)	b_3^k (B_3^k)	b_4^k (B_4^k)	b_5^k (B_5^k)	B_6^k (B_6^k)	B_7^k (B_7^k)	N - $r+1$	$S^k = \sum_{j=1}^N B_j^k$	y^k
7(MSB)	1(1)	0(0)	1(1)	0(0)	0(0)	1(1)	1(1)	6	4	0
6	0(1)	1(1)	1(1)	1(1)	1(1)	1(1)	0(1)	6	7	1
5	1(1)	1(1)	1(1)	1(1)	0(0)	1(1)	1(1)	6	6	1
4	1(1)	0(0)	0(1)	1(1)	0(0)	1(1)	0(1)	6	5	0
3	1(1)	1(1)	0(1)	0(1)	1(0)	0(1)	0(1)	6	6	1
2	0(1)	0(0)	0(1)	1(1)	0(0)	0(1)	0(1)	6	5	0
1	1(1)	0(0)	1(1)	0(1)	1(0)	0(1)	1(1)	6	5	0
0(LSB)	0(1)	1(1)	0(1)	1(1)	1(0)	0(1)	0(1)	6	6	1
Decimal value	186	105	226	117	75	240	162			105

3. HARDWARE IMPLEMENTATION

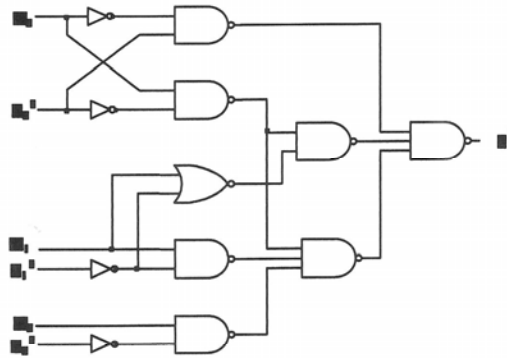
To implement the following algorithm N number of identical Bit Update Circuit, (here N is the number of samples in each window), a Parallel Counter Circuit and a Comparator Circuit is required. The function of each of the part is discussed below.



(a)



(b)



(c)

Fig. 2: Building blocks of ROF (a) Bit Update Circuit (BUC), (b) 5-bit Parallel Counter circuit, (c) 3-bit Comparator Circuit.

Bit Update Circuit (BUC):

The internal structure of the BUC is shown in Fig. 2(a). Total number of gates required to implement the BUC is only 9. N number of BUCs operate in parallel to modify the incoming bits. Most Significant Bits (MSB) from N samples enter the BUCs parallelly first. In the next clock cycle, N bits next to MSB, enter the BUCs. In M steps all the bits from all the samples from a window enter the BUCs. For each input b_j^k the BUC gives modified output B_j^k .

Parallel Counter Circuit:

The Parallel Counter circuit adds parallelly N output bits from N BUCs in each step. For $N = 5$, a 5 bit Parallel Counter Circuit is required. The Counter Circuit is shown in Fig. 2(b). In Fig. 2(b), FA means Full Adder and HA means Half Adder.

The output of the counter circuit can be expressed as:

$$S^k = \sum_{j=1}^N B_j^k \quad (1)$$

With the increase in number of input samples in each window the number of adder stages in the Parallel Counter Circuit increases. Hence, the speed of the ROF decreases as the window size increases.

Comparator Circuit:

After the Parallel Counter Circuit, a $\lceil \log_2(N+1) \rceil$ bit Comparator Circuit is needed. The Comparator compares the output of the Parallel Counter circuit S^k with an external binary input A , where the value of A is $N-r+1$. Thus controlling the value of A , any rank-order element from $r=1$ to N can be determined by the circuit. For $N=5$, a 3-bit ($\lceil \log_2 6 \rceil = 3$) Comparator circuit is required. The logic diagram of the Parallel Counter Circuit is shown in Fig. 2(c)

The output of the Comparator circuit can be written as:

$$Y^k = 1 \text{ if } S^k \geq A^k \\ 0 \text{ if } S^k < A^k \quad (2)$$

$$\text{and } A^k = \sum_{i=0}^2 A_i^k \cdot 2^i \quad (3)$$

4. OPERATION OF THE RANK SELECTION CIRCUIT

For window size $N = 5$, the total circuit diagram of the proposed Rank Order Filter is shown in Fig. 3.

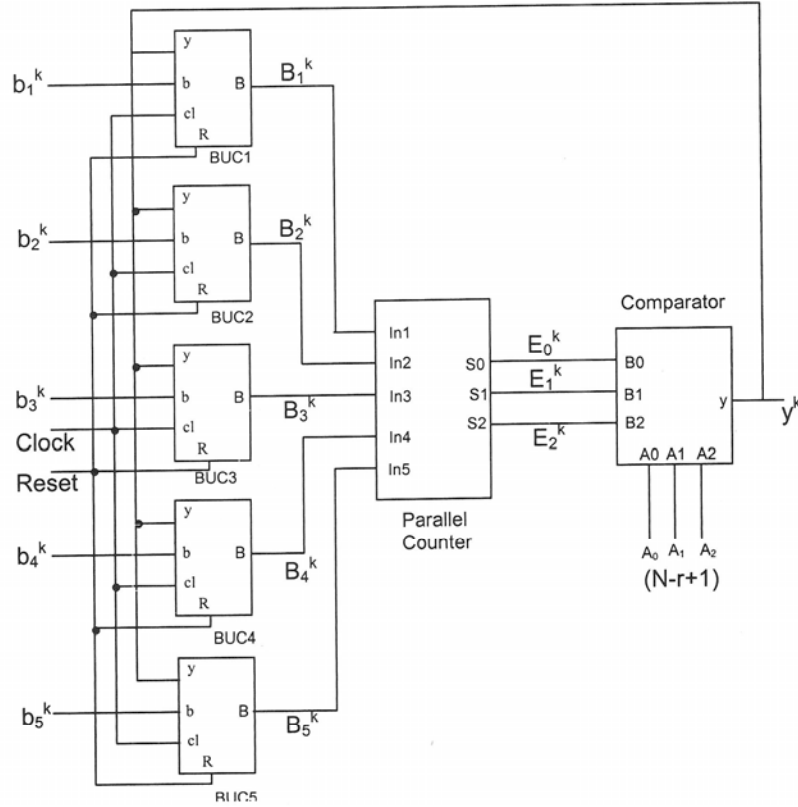


Fig. 3: Architecture for Rank Selection Circuit.

As mentioned earlier the total number of BUC required to implement the ROF is equal to the window size (N) of the ROF. Thus, five BUCs are needed to implement a 5-input ROF. Now with each clock pulse one bit from each sample starting from MSB will enter each BUC. If the number of bit in each sample is $M = 8$, then with the first clock pulse the first entering bits to the five BUCs are $b_1^7, b_2^7, b_3^7, b_4^7, b_5^7$. With the next clock pulse the entering bits are $b_1^6, b_2^6, b_3^6, b_4^6, b_5^6$. In this way bits from the samples enter the BUC. With the eighth clock pulse the entering bits are $b_1^0, b_2^0, b_3^0, b_4^0, b_5^0$. Initially the Reset input of all the BUC's are momentarily kept High. So the inverted output of the SR latch will be High, which is input to the Enabler pin of the gated D latch. As a result during the first clock cycle the output bits from the five BUCs ($B_1^7, B_2^7, B_3^7, B_4^7, B_5^7$) are same as the MSBs of the input samples ($b_1^7, b_2^7, b_3^7, b_4^7, b_5^7$). These five bits are added by the Parallel Counter circuit. Thus, the number of 1's output from the BUCs are obtained. To determine the 3rd smallest ($r = 3$) element using the ROF the external input to the Comparator circuit should be $A = N - r + 1 = 5 - 3 + 1 = (3)_{10} = (011)_2$. The Comparator compares the output of the Parallel Counter circuit S^k with A . If $S^k \geq A$ then output of the

Comparator circuit will be '1', otherwise the output will be '0'. This will be the MSB of the desired rank order element. Now if the output of the Comparator circuit is different from the incoming bit of the samples to the BUC then output of the XOR gate will be High. When the output of the XOR gate and clock both are High, the active High SR latch becomes Set and the inverted output of the SR latch will become Low. This will disable the gated D latch and the output of the D latch will remain the same and will not be affected by new incoming sample bits.

The output of the BUC can be expressed as:

$$B_j^k = b_j^k \bar{Q}_j + B_j^{k+1} Q_j \tag{4}$$

$$\text{where, } Q_j = Q_j + (b_j^k \oplus y^k) \tag{5}$$

5. IMPLEMENTATION IN FPGA

As the proposed Rank Order Filter is fully digital, so the architecture of the filter can be coded in VHDL language. The code is then simulated in RTL level using Xilinx 5.1i ISE Webpack, to verify the correctness of the design. The logic synthesis is carried out to optimize the designs, and the gate-level simulation is then performed. The target device is Spartan2E-xc2S50e. The output of post-fit VHDL model is shown in Fig. 4.

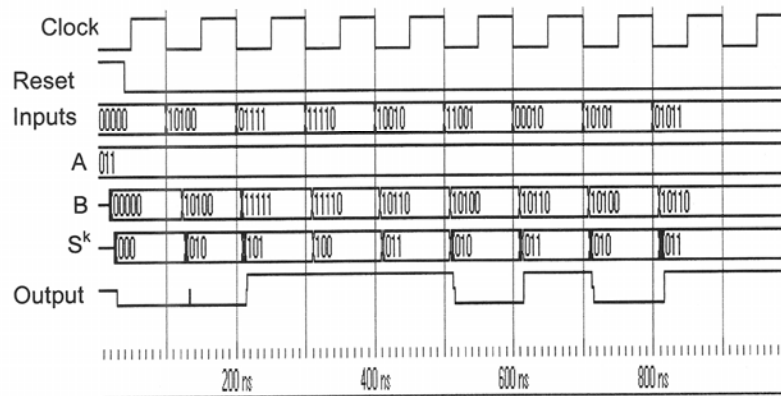


Fig. 4: Post-fit simulation output for the inputs as shown in Table 1.

In Fig. 4,

- 'A' indicates the external input of the comparator ($N-r+1$). For $N = 5$ and $r = 3$, $A = (3)_{10} = (011)_2$.
- S^k indicates the output of the Counter circuit and
- 'B' indicates the output of BUC.

The 3rd (r^{th}) smallest element of the window is obtained as output from the filter.

The gate level description of the filter is downloaded in FPGA board and the circuit is found to operate at a clock speed of 10 MHz.

For the same data window, another example is given in Fig. 5, where the value of 'r' is 5. So $A = N-r+1 = (001)_2$. In this example, the sample with highest magnitude is obtained as output.

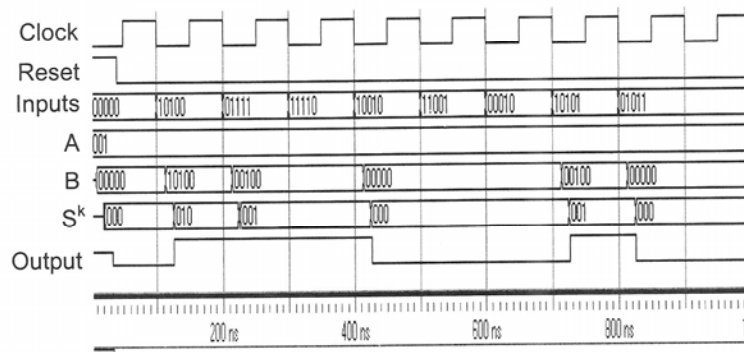


Fig. 5: Post-fit simulation output for the inputs as shown in Table 2.

For another data window with window size $N = 7$, $r=2$, $W=\{186,226,117,75,240,162\}$ an example is given in Fig. 6. Here external input to the Comparator Circuit is $A = N-r+1 = (110)_2$. In this example, the sample with 2nd lowest magnitude is obtained as output.

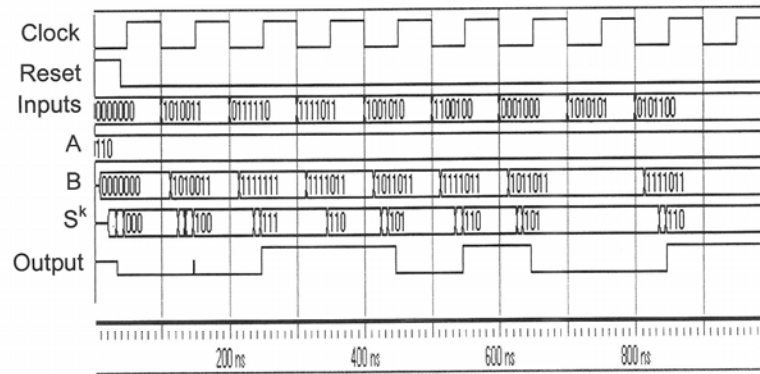


Fig. 6: Post-fit simulation output for the inputs as shown in Table 3.

6. HSPICE SIMULATION FOR 5-INPUT ROF

The transistor level HSPICE simulation output of the Rank Order Filter is shown in Fig. 7. The simulation is performed in $0.18 \mu\text{m}$ CMOS process for the inputs as given in Table 1. The value of r is selected to be three for the simulation. The 3rd smallest element or the median of the five input samples should be the output of the ROF. The simulation gives the median as output without any error at a clock speed of 500MHz. The average delay in the ROF filter circuit is found to be from 600 ps to 950 ps. The total number of transistors required to implement the 5-input ROF is only 320.

In Fig. 7, (a)-(e) represent the inputs to the 5-input ROF and (f) represents the output of the ROF.

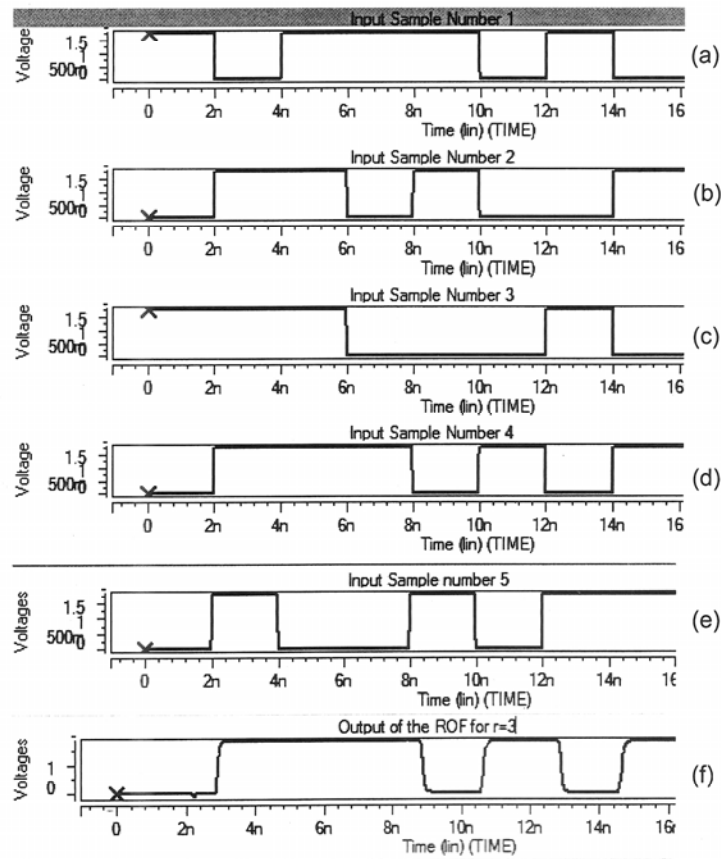


Fig. 7: Simulation result of ROF for $N = 5$ and $r = 3$

7. CONCLUSION AND RECOMMENDATION

The proposed Rank Order Filter is fully digital. The number of logic gates required to implement the Bit Update Circuit in this design is only 9. But the other designs employing BUC/ bit processing circuit / Polarization subsystem [2, 8] requires 22 and 20 logic gates respectively. So the proposed filter occupies less Silicon area and consumes less power. The size of some of the ROFs [3, 5] increases as the square of the size of the ROF. But the size of the proposed ROF increases only linearly with the window size. The proposed Rank Order Filter is very modular and it can be easily designed for any window size. From the transistor level HSPICE simulation, the operating speed of the proposed ROF is found to be 500 MHz which is also very high. In the proposed system the throughput (measured in output sample values over time) of the circuit is found to be the clock frequency divided by M (M is the number of bits in each sample). But the throughput of the circuit can be increased by M times by employing parallel processing technique such as data pipelining and block processing.

REFERENCES

- [1] J. P. Fitch, E. J. Coyle and N. C. Gallagher, Jr., "Threshold decomposition of multidimensional ranked order operations." IEEE Transaction on Circuits and Systems, Vol. 32, pp.445-450, 1985.
- [2] Dulal C. Kar and Pran K. Saha, "A Parallel, Operational Amplifier Based Design of a Selection Circuit for Rank-Order Filtering," Proceedings of second International Conference on Electrical and Computer Engineering, Dhaka, Bangladesh, pp. 26-28, 2002.
- [3] Ching C. Lin and Chung J. Kuo, "Two-Dimensional Rank-Order Filter by Using Max-Min Sorting Network" IEEE Transaction on Circuits and Systems for Video Technology, Vol. 8, No. 8, Dec 1998.
- [4] L. E. Lucke, K. K. Parhi, "Parallel processing architectures for rank order and stack filters," IEEE Transaction on Signal Processing Vol 42, pp 1178-1189, May 1994.
- [5] Ahmed Hiasat, Omar Hasan, "Bit-serial architecture for rank order and stack filters," INTEGRATION, the VLSI journal 36, pp. 3-12, 2003.
- [6] K. Chen, "Bit-serial realizations of a class of nonlinear filters based on positive Boolean functions", IEEE Transaction on Circuits and Systems, Vol 36, pp 785-794, 1989.
- [7] A. Gasteratos, I. Andreadis and Ph. Tsalidis, "Realization of Rank Order Filters based on Majority Gate," Journal of Pattern Recognition, Vol. 30, No. 9, pp. 1571-1576, 1997.
- [8] L. Brevegliery, V. Piuri, "Digital Median Filters," Journal of VLSI signal Processing Systems for Signal, Image and Video Technology Vol 31, No. 3 pp 191-206, July 2002
- [9] B. K. Kar, D.K. Pradhan, "A new algorithm for order statistics and sorting," IEEE Transaction on Signal Processing Vol. 41, No. 8, pp. 2688-2694, August 1993.