Transactions Briefs

CMOS Threshold Gate and Networks for Order Statistic Filtering

Charng Long Lee, *Member, IEEE*, and Chein-Wei Jen, *Member, IEEE*

Abstract—The threshold gate is a very good candidate in the realization of order statistic filtering. In this brief paper, a simple procedure is developed to determine the circuit parameters for a set of output-wired CMOS inverters in order to implement threshold functions for order statistic filtering. Weighted and nonweighted order statistic filters in either binary or integer domain can be easily realized in similar architectures. An incremental scheme is also proposed to construct threshold gate networks for general or multi-stage order statistic filters.

I. INTRODUCTION

Order statistic filters are a class of nonlinear filters which include median, rank order, weighted median and weighted rank order filters. They have been widely used in many modern signal and image processing applications. The hardware realization of such filters can be done by purely binary logic gates, sort-and-select, or count-and-compare circuits. However, a very good alternative is by threshold gates.

A threshold gate[1] is defined as

$$y = f(\vec{x}) = \begin{cases} 1, & \text{if } \sum_{i=1}^{n} w_i x_i \geq T \\ 0, & \text{if } \sum_{i=1}^{n} w_i x_i < T \end{cases} \quad (1)$$

where $\vec{x} = (x_1, \cdots, x_n)$, $x_i \in \{0,1\}$ are binary inputs and w_i are their associated nonnegative weights, n is the total number of inputs, and T is a threshold. If $w_i = 1$ for $i = 1, \cdots, n$, then a threshold gate becomes a "T-out-of-n" gate, denoted as (T/n). The output is true if more than or equal to T of its n inputs are in logic "1" state.

Physical implementation of a threshold function can be either in digital or analog ways. Both count-then-compare and combinational logic are commonly used. Analog voltage division with comparator is also a good solution. However in the next section, an interesting hybrid approach realizes threshold functions by using only CMOS digital inverters. It has the advantages of simple hardware and easy design.

In the following sections, we will first describe a simple design of threshold gate involving only CMOS inverters. A procedure is developed to choose device parameters of a threshold gate design. Then, how the threshold gate can be applied for order statistic filtering is discussed in both binary and integer domains in section III. And in section IV, A systematic approach is proposed to link threshold gates as a network for general order statistic filtering.
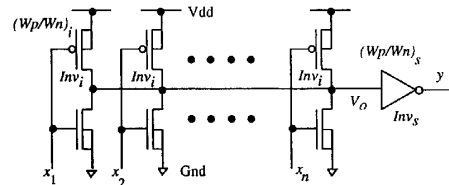
Fig. 1. A (T/n) threshold gate implemented by output-wired CMOS inverters (Inv_i) cascaded by a sensing inverter (Inv_s).

II. A CMOS THRESHOLD GATE DESIGN

A. Analysis

A (T/n) threshold gate can be implemented by wiring the outputs of n identical CMOS inverters (Inv_i). The channel length of all MOS transistors are chosen to be the minimum dimension and their channel width ratio (W_p/W_n) for P- and N-channel transistors are the same. This circuit is actually a nonlinear voltage divider and cascaded by a simple sensing amplifier (Inv_s) as shown in Fig. 1. In each input inverter, only one transistor conducts at a time because its input is either logic "1" (V_{dd}) or "0" (Gnd). Both P- and N-channel transistors act as a resistor when they conduct. Consequently, the voltage level of the wired node, V_o, depends on how many P- and N-channel transistors are conducting. As the total number of "1" among these inputs increases, more N-channel transistors will conduct. Such a change drives the node, V_o, closer to the ground level as shown in Fig. 2.

Because the nonlinear characteristics of a MOS transistor, the voltage level transition at node V_o is not uniform. A maximum voltage gap occurs at a certain threshold position. An inverter (Inv_s) is placed here to detect this maximum transition. In addition to this reason, the sensing inverter also acts as an output buffer to reshape the signal waveform and provide driving capability [2]. It can also isolate the wired node from external circuitry to reduce noise effect [3]. The location of the maximal transition gap can be changed if we properly choose a channel width ratio for these input inverters. Thus the design of a threshold gate can be controlled by tuning the device dimensions.

B. Design Procedure

The threshold T can be either one of the integers between 1 and n for a (T/n) threshold gate. The key issue of threshold gate design is how to adjust the transistor dimensions so that the maximum transition gap meets a specific threshold position. It is the best to find an analytical solution for the channel width ratio (W_p/W_n). However this is not possible because of some second-order effects of devices. We can only obtain the upper and lower bounds of this ratio, r_{max} and r_{min} by solving drain current equations [4]. They are

$$r_{min} = (W_p/W_n)_{min} = \frac{(\frac{T-1}{n-T+1})(\frac{\mu_n}{\mu_p})}{(1 - \frac{V_{dd}V_{dpp}-2V_{IH}V_{dnp}}{V_{IH}(V_{IH}-2V_{dn})})} \quad \text{and}$$
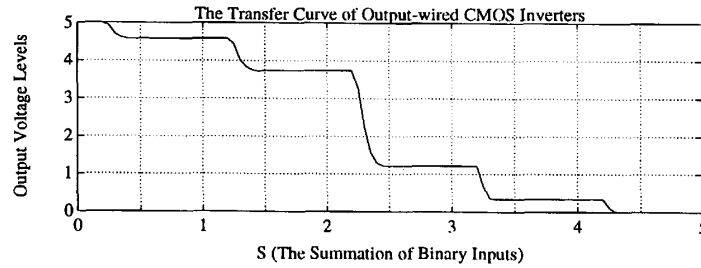
Fig. 2.   The transfer characteristics of threshold gate with $n = 5$, channel length $L = 1.2\mu m$, and channel width ratio $W_p/W_n = 5.0\mu m/1.8\mu m$.
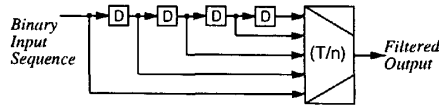

Fig. 3.   A binary order statistic filter implemented by a (T/n) threshold gates, where "D" is a delay element for one bit.
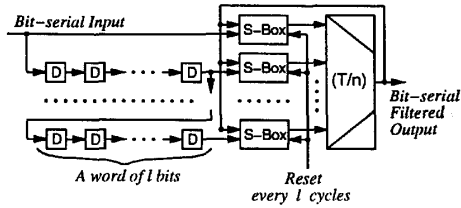

Fig. 4.   A bit-serial approach for multi-level order statistic filtering using threshold gate.
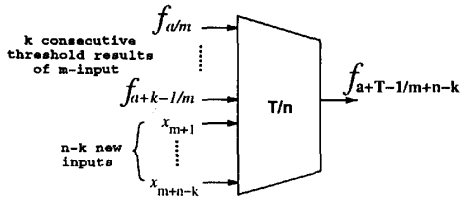

Fig. 5.   An insertion threshold gate $f_{T/n}^{a/m:k}$ merges $n - k$ new inputs to the k consecutive threshold results of m inputs.

$$r_{max} = (W_p/W_n)_{max} = \frac{(\frac{T}{n-T})(\frac{\mu_n}{\mu_p})}{(1 - \frac{V_{dd}V_{dpp}-2V_{IL}V_{dnp}}{V_{IL}(V_{IL}-2V_{dn})})},$$

where $V_{dpp} = V_{dd}-2V_{tp}$, $V_{dnp} = V_{dd}-V_{tn}-V_{tp}$, $V_{dn} = V_{dd}-V_{tn}$, V_{IH} and V_{IL} are the input high and input low voltage of an inverter, V_{dd} is the supply voltage, and V_{tp} and V_{tn} are the threshold voltage of P- and N-channel transistors. A proper channel width ratio is searched within this range $r_{max} \geq r \geq r_{min}$.

TABLE I
SELECTED RATIOS AND CHARACTERISTICS OF THRESHOLD GATES FOR $n = 5$ AND $L = 1.2\mu m$

T	1	2	3	4	5
$W_p(\mu m)$	1.8	2.0	5.0	12.0	45.0
$W_n(\mu m)$	5.0	1.8	1.8	1.8	1.8
Ave.Power(mW)	1.64	1.58	2.88	3.94	4.55
t_{dHL}(nsec)	0.64	0.87	0.68	0.63	0.66
t_{dLH}(nsec)	0.48	0.65	0.77	0.92	1.27

TABLE II
COMPARISON OF HARDWARE AND TIME COMPLEXITY OF SEVERAL BINARY ORDER STATISTIC FILTER IMPLEMENTATION METHODS

Methods	Counting	Sorting	PBF*	Threshold Gate
Hardware Complexity (A)	$O(n)$	$O(n \log n)$	C_T^n	$O(n)$
Time Complexity (T)	$O(\log n)$	$O(\log n)$	$O(1)$	$O(1)$
References	[6]	[7]	[8]	-

In summary, we can conclude a procedure for threshold gate design.

1) Calculate the channel width ratio boundaries r_{max} and r_{min} for a specific (T/n) threshold gate.
2) Search for a proper channel width ratio r within $[r_{max}, r_{min}]$ and justified by circuit simulations.
3) Select an inverter for the sensing buffer whose threshold voltage, V_{inv}, is the middle point of the desired transition gap.

The major advantage of this CMOS threshold gate design is it's simplicity. It's delay time is only around two inverter delays. However, this class of CMOS threshold gates consume static power because the current flows through the nonlinear voltage divider most of the time.

III. ORDER STATISTIC FILTERING BY CMOS THRESHOLD GATES

A. Binary Domain OSFs

An order statistic filter (OSF) is defined to select the Tth largest element among n inputs [5]. In integer domain, it is known as a rank order chooser which is usually implemented by a sorting or selection network. In the binary domain, the Tth largest element among n binary inputs can be determined by a (T/n) threshold gate. As a result, the application of threshold gate to binary order statistic filtering is straightforward. We can design a (T/n) threshold gate according to the method and procedure presented in the last Section.
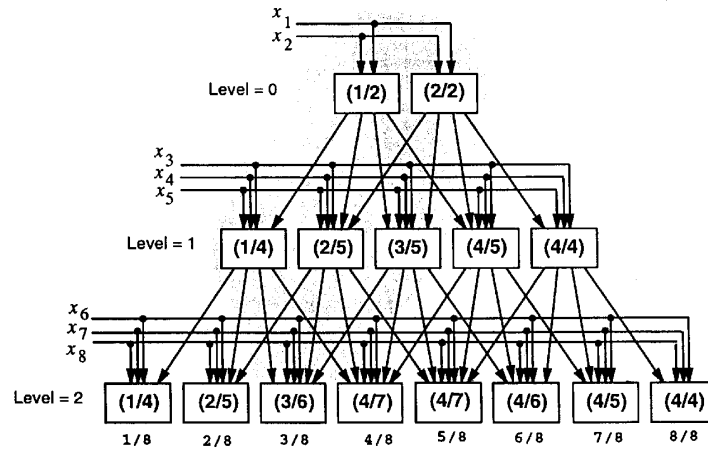
Fig. 6. A general threshold gate network for $m = 2, k' = 3$.

It takes $n+1$ CMOS inverters and n of them are identical. A complete set of threshold gate design for $n = 5$ is listed in Table I for example.

The threshold gate designs can be applied for binary rank order filtering as shown in Fig. 3. A binary input sequence is tapped out at n positions with $n - 1$ delay elements indicated by "D." At each time instance, the (T/n) threshold gate gives an order statistic filtered result as the output. Thus, a complete filtered out sequence can be obtained after the original sequence passing through this filtering structure. A brief comparison of hardware and time complexity in different binary order statistic filter implementation methods is given in Table II. It appears that the implementation by threshold gate is superior in the measure of area-time product.

B. Weighted OSFs in Binary Domain

A weighted order statistic filter (WOSF) is a general order statistic filter with nonequal weighting on each input. The ith input element x_i will be duplicated for w_i times if its weighting is w_i. Thus, the equivalent number of inputs becomes $W = \sum_{i=1}^{n} w_i$. A weighted order statistic filter choose the Tth largest element among these W elements. This function can be realized by a (T/W) threshold gate in binary domain. The choosing of optimal channel width ratio for threshold T should be based on W, rather than n inputs. Suppose that the inputs of the (T/W) threshold gate are $y_1, y_2, \cdots, y_W$ and the inputs of the weighted order statistic filter are $x_1, x_2, \cdots, x_n$, then each x_i should be connected to y_j for $j = W_{i-1} + 1, W_{i-1} + 2, \cdots, W_i$, where $W_i = \sum_{k=1}^{i} w_k$, for $i = 1$ to n. That is to assign w_i consecutive inputs positions for the x_i variable. A simplified scheme is to merge the w_i identical inverters into one inverter whose channel width ratio is $w_i(W_p/W_n)$. This design reduce the number of inverter from W to n without affecting its functionality. The advantage is that both threshold position and weighting of each input are programmed in the device dimensions.

C. Integer Domain OSFs

In the integer domain of multi-valued signals, the threshold gate can not be applied to order statistic filtering directly as in binary domain. A threshold decomposition method can solve this problem

by decomposing the l-bit binary-coded signal sequence into 2^l levels of binary sequences [5] so that each level of binary signal sequence can be processed as in binary domain. The results of each level are then accumulated together and represented again in l-bit as for final output.

The bit serial approach presented in [9] modified the idea of threshold decomposition to operate on the l-bit representation. This method reduce the hardware cost greatly in comparison to threshold decomposition. The penalty is to spend l cycles to get a complete results.

A simplified block diagram is shown in Fig. 4. The binary order statistic filtering is performed by a threshold gate. The switch box is to ensure the stacking property of the succeeding bits so that the same order statistic function is valid through out the whole l steps of filtering processing [9]. As described above, any integer domain order statistic filters can be realized by threshold gate easily with such a bit serial approach. Thus order statistic filtering in both binary and integer domains can be easily realized by threshold gate in similar architectures.

IV. CONSTRUCTION OF THRESHOLD GATE NETWORKS

Although the CMOS threshold gate design discussed above is a good approach in implementing order statistic filters, it has some inherent limitations. For examples, its operating condition may be shifted due to process variations and its total number of inputs can not be too large. As the input number increases, the maximum voltage gap may become too small to be detected by a simple inverter while the threshold position may be shifted by process variations. The suggested number of inputs is 11 and below [2]. Because of such limitations, a threshold gate network is required for order statistic filtering of large window. The basic idea is to insert $n - k$ new signals into the old k consecutive threshold results by threshold gates.

Definition 1: An insertion threshold gate (ITG), denoted as $f_{T/n}^{(a/m):k}$ is a threshold gate where k of its n inputs are connected to the output of k consecutive threshold functions of $f_{a/m}(\vec{x}_m)$ to $f_{a+k-1/m}(\vec{x}_m)$ with $\vec{x}_m = (x_1, \cdots, x_m)$. The rest $n - k$ signals are fresh inputs for $(x_{m+1}, \cdots, x_{m+n-k})$. It is illustrated in Fig. 5.

Theorem 1: The output of an insertion threshold gate $f_{T/n}^{(a/m):k}$ is equivalent to the result of threshold function $f_{a+T-1/m'}(\vec{x}_{m'})$, where $\vec{x}_{m'} = (x_1, \cdots, x_m, x_{m+1}, \cdots, x_{m+n-k})$ and $m' = m + n - k$, if and only if the valid range of threshold T satisfies either one of the following conditions: (1) If $a = 1$ then $1 \leq T \leq k$; (2) If $a = m - k + 1$, then $n - k + 1 \leq T \leq n$; (3) If $1 < a < m - k + 1$, then $n - k + 1 \leq T \leq k$.

Proof: By some simple substitutions, we can obtain that in order to generate the output of an ITG as $f_{a+T-1/m'}$ without uncertainty, the threshold position of T should be fallen in the intersection of the following integer intervals.

$$\{1 \leq T \leq k\} \cap \{2 \leq T \leq k+1\} \cap \cdots \cdots$$
$$\cap \{n - k + 1 \leq T \leq n\}$$

where $\cap$ is the intersection operator. Thus, the valid range of threshold T is $n - k + 1 \leq T \leq k$. There are two exceptions. One case is when $a = 1$ and $f_{1/m}(\vec{x}_m)$ exists, the valid range of T is $1 \leq T \leq k$. The other case is when $a + k - 1 = m$ and $f_{m/m}(\vec{x}_m)$ exists, the valid range of T is $n - k + 1 \leq T \leq n$. □

Corollary 1: For an insertion threshold gate $\tilde{f}_{T/n}^{(a/m):k}$, if $1 < a < m - k + 1$, then $k \geq \frac{n+1}{2}$.

This corollary is a direct consequence of condition (3) of Theorem 1. It tells that more than half of the inputs of a threshold gate should be devoted to the consecutive threshold results, otherwise the output result will be ambiguous. According to this observation, we can built a general threshold decision network by insertion threshold gates in an incremental fashion. The procedure is as follow:

1) In the initial stage, level 0, we employ full range of m-input threshold gates to obtain the threshold results of $f_{1/m}, \cdots, f_{m/m}$ for the inputs $(x_1, \cdots, x_m)$, simultaneously.
2) Then we insert k' new binary variables in the next level with the current m threshold results, where $k' = n - k$ as in Definition 1. Thus each new level will have k' more insertion threshold gates than the previous level. In level s, we will have $(m + s \cdot k')$ insertion threshold gates.
3) The output of each insertion threshold gate should connect to $(k' + 1)$ neighboring cells in the next level, according to Corollary 1.
4) In level s, we will have k' low-threshold gates of $f_{1/k'+1}$, $\cdots$, $f_{k'/2k'}$ on the left, $(m + s \cdot k' - 2k')$ majority gates of $f_{k'+1/2k'+1}$ in the middle, and k' high-threshold gates of $f_{k'+1/2k'}$, $\cdots$, $f_{k'+1/k'+1}$ on the right. An exception is when $m \leq k'$, there will be no majority gates of $f_{k'+1/2k'+1}$ in level 1. It will contain m low-threshold gates of $f_{1/k'+1}$, $\cdots$, $f_{m/k'+m}$ on the left, $(k' - m)$ middle-threshold gates of $f_{m+1/k'+m}$, $\cdots$, $f_{k'+1/k'+m}$ in the middle, and m high-threshold gates of $f_{k'+1/k'+m-1}$, $\cdots$, $f_{k'+1/k'+1}$ on the right.
5) This network can generate threshold function from $f_{1/m+s \cdot k'}$ to $f_{m+s \cdot k'/m+s \cdot k'}$ at level s for any integer $m \geq 2$ and $k' \geq 1$.

An example of $m = 2, k' = 3$ is shown in Fig. 6. The threshold results of $f_{1/8}, \cdots f_{8/8}$ are available at the outputs of level 2. We must notice that not all of these cells are required if only a single threshold result is desired. It only takes the insertion threshold gates in the shaded area to reach the output of $f_{3/8}$ as shown in Fig. 6.

Theorem 2: A network built of insertion threshold gates according to the above scheme provides a realization of general threshold functions.

The proof of this theorem is simple. One can check the input variables, output results, and threshold T of each threshold gate in the network to find out if they all satisfy the condition of Theorem 1. The threshold gate network is correct because all of the outputs are unambiguously defined.

V. CONCLUSION

We have proposed a design of threshold gate by output-wired CMOS inverters for order statistic filtering. The threshold position of such a threshold gate is programmed on the channel width ratio (W_p/W_n) of each input inverter. Its design simplicity and small hardware makes it attractive for many applications. Order statistic filters in both binary and integer domain can be easily implemented in simple architectures by the threshold gates.

In applications to small window size, an order statistic filter can be accomplished by a single threshold gate. However, many detail-preserving order statistic based filters are multi-stage filters [10], such as max/median filters, center weighted median filters, and morphological filters [11]. They can be realized in a multi-stage threshold gate networks.

A network scheme is also proposed for general threshold decision of large window size. The use of threshold gates provides an efficient realization of order statistic, rank order, and median related filters.

REFERENCES

[1] S. Muroga, *Threshold Logic and Its Applications.* New York: John Wiley & Sons, 1971.
[2] C. L. Lee and C. W. Jen, "Bit-sliced median filter design based on majority gate," in *IEE Proc. Part G,* vol. 139, no. 1, pp. 63–71, Feb. 1992.
[3] K. J. Schultz, R. J. Francis, and K. C. Smith, "Ganged CMOS: Trading standby power for speed," *IEEE J. Solid-State Circ.,* vol. 25, no. 3, pp. 870–873, June 1990.
[4] J. P. Uyemura, *Fundamentals of MOS Digital Circuits.* Reading, MA: Addison-Wesley, 1988.
[5] O. Yli-Harja, J. Astola, and Y. Neuvo, "Analysis of the properties of median and weighted median filters using threshold logic and stack filter representation," *IEEE Trans. Signal Processing,* vol. 39, no. 2, pp. 395–410, Feb. 1991.
[6] C. D. Thompson and H. Yasuura, "On the area-time optimal design of l-selectors," in *1985-86 Asilomar Conf. Cir. Sys. Comput.,* pp. 365–368, 1986.
[7] M. Ajtai, J. Komlos, and E. Szemredi, "An $O(n \log n)$ sorting network," in *15th Annual. ACM Symp. Theory Comput.,* pp. 1–9, 1983.
[8] J. P. Fitch, "Software and VLSI algorithms for generalized ranked order filtering," *IEEE Trans. Circ. and Syst.,* vol. CAS-34, no. 5, pp. 553–559, May 1987.
[9] K. Chen, "Bit-serial realizations of a class of nonlinear filters based on positive boolean functions," *IEEE Trans. Circ. and Syst.,* vol. 36, no. 6, pp. 785–794, June 1989.
[10] G. R. Arce and R. E. Foster, "Detail-preserving ranked-order based filters for image processing," *IEEE Trans. Acoustic, Speech, Signal Processing,* vol. 37, no. 1, pp. 83–98, Jan. 1989.
[11] R. L. Stevenson and G. R. Arce, "Morphological filters: Statistics and further syntactic properties," *IEEE Trans. Circ. and Syst.,* vol. CAS-34, no. 11, pp. 1292–1305, Nov. 1987.