CMOS Threshold Gate and Networks
for Order Statistic Filtering

Chang 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 filters. 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 domains can be easily realized in similar architectures. An incremental scheme is also proposed to construct threshold gate networks for general or multistage 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

\[ y = \begin{cases} 1, & \text{if } \sum_{i=1}^{n} w_i x_i \geq T \\ 0, & \text{if } \sum_{i=1}^{n} w_i x_i < T \end{cases} \]

where \( x = (x_1, \ldots, x_n), \) \( x_i \in \{0, 1\} \) 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 \( i = 1, \ldots, n \), then a threshold gate becomes a \( \left( T/n \right) \)-out-of-\( n \) gate, denoted as \( \left( T/n \right) \)-OS. 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.

Manuscript received April 5, 1993; revised January 27, 1994. This work was supported in part by the National Science Council, Taiwan, Republic of China under Grant NSC-80-0008-O9-01-13.

C. L. Lee is with the Video Signal Processing Department, Taiwan, Republic of China.

C.-W. Jen is with the Institute of Electronics and Department of Electrical Engineering, National Chiao Tung University, Hsinchu, 300, Taiwan, Republic of China.

IEEE Log Number 9401426.

II. A CMOS THRESHOLD GATE DESIGN

A. Analysis

A \( \left( T/n \right) \)-OS threshold gate can be implemented by wiring the outputs of \( n \) identical CMOS inverters (\( I_{now} \)). The channel length of all MOS transistors is 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 (\( I_{now} \)) as shown in Fig. 1. In each input inverter, only one transistor conducts at a time because its input is either logic 1 or 0 (\( V_{dd} \) or \( Gnd \)). Both \( P \)- and \( N \)-channel transistors act as a resistor when they conduct. Consequently, the voltage level of the wired node, \( V_{x_i} \), 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, \( V_{x} \), closer to the ground level as shown in Fig. 2.

Because the nonlinear characteristics of a MOS transistor, the voltage level transition at node \( V_{x} \) is not uniform. A maximum voltage gap occurs at a certain threshold position. An inverter (\( I_{now} \)) 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 maximum 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 \( \left( T/n \right) \)-OS 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, \( W_{max} \) and \( W_{min} \), by solving drain current equations [4]. They are:

\[
W_{max} = \left( \frac{W_p}{W_n} \right)_{max} = \left( \frac{-e_{max}}{e_{0} V_{dd}} \right) \quad \text{and} \quad W_{min} = \left( \frac{W_p}{W_n} \right)_{min} = \left( \frac{-e_{min}}{e_{0} V_{dd}} \right)
\]

where \( e_{max} \) is the maximum channel thickness of the \( P \)-channel transistor and \( e_{min} \) is the minimum channel thickness of the \( N \)-channel transistor.
The Transfer Curve of Output-wired CMOS Inverters

Fig. 2. The transfer characteristics of threshold gate with \( n = 5 \), channel length \( L = 1.2\mu m \), and channel width ratio \( W_p/W_n = 5.0\mu m/1.8\mu m \).

<table>
<thead>
<tr>
<th>( T/n )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>( W_p(\mu m) )</td>
<td>1.8</td>
<td>2.0</td>
<td>5.0</td>
<td>12.0</td>
<td>45.0</td>
</tr>
<tr>
<td>( W_n(\mu m) )</td>
<td>5.0</td>
<td>1.8</td>
<td>1.8</td>
<td>1.8</td>
<td>1.8</td>
</tr>
<tr>
<td>Avg.Power(nW)</td>
<td>1.64</td>
<td>1.58</td>
<td>2.88</td>
<td>3.94</td>
<td>4.55</td>
</tr>
<tr>
<td>( t_{ON}(\text{ms}) )</td>
<td>0.64</td>
<td>0.87</td>
<td>0.68</td>
<td>0.63</td>
<td>0.66</td>
</tr>
<tr>
<td>( t_{OFF}(\text{ms}) )</td>
<td>0.48</td>
<td>0.60</td>
<td>0.62</td>
<td>0.66</td>
<td>0.70</td>
</tr>
</tbody>
</table>

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

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

The major advantage of this CMOS threshold gate design is its simplicity. Its 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 \( T \)th 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 \( T \)th 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.
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 output 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 of each input. The \(i\)th 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 \(T\)th 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 \(y_1, y_2, \ldots, y_W\) and the inputs of the weighted order statistic filter are \(x_1, x_2, \ldots, x_n\), then each \(x_i\) should be connected to \(y_j\) for \(j = W/n + 1, W/n + 2, \ldots, W\), where \(W_i = \sum_{i=1}^{n} w_i\) 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 \(w_i(W/W_i)\). 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 result.

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 throughout 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 \(T_{(n+k)}(m+k)\), is a threshold gate where \(k\) of its \(n\) inputs are connected to the output of \(k\) consecutive threshold functions of \(f_{x_v}(z_{v+1})\) to \(f_{x_n-1}(z_{v+k})\) with \(z_{v+k} = \{x_{v+1}, \ldots, x_{n+k}\}\). The rest \(n-k\) signals are fresh inputs for \(f_{x_{n+1}}(z_{n+k})\). It is illustrated in Fig. 5.
Theorem I: The output of an insertion threshold gate \(f^{(m+n+1)}_{r/m+k} \) is equivalent to the result of threshold function \(f_{m+1}^{(r-1)/m}(f_{n}^{m}) \), where \(f_{m+1}^{(r-1)/m}(f_{n}^{m}) \) and \(m = n - k \), if and only if the valid range of threshold \(T \) satisfies either one of the following conditions: (1) If \( n = 1 \) then \( 1 \leq T \leq k \); (2) If \( n > 1 \) then \( 1 \leq T \leq k \) or \( n \leq k \). If \( n < 1 \), then \( m - k + 1 \leq T \leq k \).

Proof: By some simple substitutions, we can obtain that in order to generate the output of an IGG as \( f_{m+1}^{(r-1)/m} \), without uncertainty, the threshold position of \( T \) should be fallen in the intersection of the following integer intervals.

\[
\{1 \leq T \leq k\} \cap \{2 \leq T \leq k + 1\} \cap \ldots \cap \{n - k + 1 \leq T \leq n\}
\]

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 \( n = 1 \) and \( f_{m+1}^{(r-1)/m} \) exists, the valid range of \( T \) is \( 1 \leq T \leq k \). The other case is when \( n = k - 1 \) and \( f_{m+1}^{(r-1)/m} \) exists, the valid range of \( T \) is \( n - k + 1 \leq T \leq n \).

Corollary I: For an insertion threshold gate \( f^{(m+n+1)}_{r/m+k} \), if \( 1 < n < m - k + 1 \), then \( k \geq \frac{5}{7}m \).

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

1) In the initial stage, level 0, we employ full range of \( m \) input threshold gates to obtain the threshold results of \( f_{m+1}^{(r-1)/m}(f_{n}^{m}) \) for the inputs \( f_{n}^{m} \), simultaneously.

2) Then we insert \( k' \) new binary variables in the next level with the current \( n \) threshold results, where \( k' = n - k \) as defined in Definition I. 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 I.

4) In level \( s \), we will have \( k' \) low-threshold gates of \( f_{m+1}^{r/(m+k')} \) in the middle, and \( k' \) high-threshold gates of \( f_{m+1}^{r/(m+k')} \) on the right. An exception is when \( m < k \), there will be no majority gates of \( f_{m+1}^{r/(m+k')} \) in level 1. It will contain \( m \) low-threshold gates of \( f_{m+1}^{r/(m+k')} \) on the left, \( k' - m \) middle-threshold gates of \( f_{m+1}^{r/(m+k')} \) in the middle, and \( m \) high-threshold gates of \( f_{m+1}^{r/(m+k')} \) on the right.

5) This network can generate threshold function from \( f_{m+1}^{r/(m+k')} \) to \( f_{m+n+k'/m+k} \) 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 \( f_{m}^{2} \) 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 \( f_{n}^{m} \) 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 I. The threshold gate network is correct because all of the outputs are unambiguously defined.