

University of Nebraska - Lincoln

DigitalCommons@University of Nebraska - Lincoln

---

CSE Journal Articles

Computer Science and Engineering, Department  
of

---

1989

## Signal Probabilities in AND-OR Trees

Lester Lipsky

*University of Connecticut - Storrs*

Sharad C. Seth

*University of Nebraska-Lincoln, seth@cse.unl.edu*

Follow this and additional works at: <https://digitalcommons.unl.edu/csearticles>



Part of the [Computer Sciences Commons](#)

---

Lipsky, Lester and Seth, Sharad C., "Signal Probabilities in AND-OR Trees" (1989). *CSE Journal Articles*. 34.  
<https://digitalcommons.unl.edu/csearticles/34>

This Article is brought to you for free and open access by the Computer Science and Engineering, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in CSE Journal Articles by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

# Signal Probabilities in AND-OR Trees

LESTER LIPSKY AND SHARAD C. SETH, SENIOR MEMBER, IEEE

**Abstract**—In this paper, we consider a class of AND-OR tree circuits and study their response to random-pattern inputs as the depth of the tree is allowed to increase indefinitely. Each binary input of a circuit is independently chosen to be one (zero) with probability  $x$  ( $1 - x$ ). The logic of the circuit determines the probability of success (one) at the output as a monotonically increasing  $S$ -shaped function of  $x$  called the *probability transfer function*. The probability transfer function of an AND-OR tree is shown to have just one interior *fixed point* (w.r.t. changes in depth of the tree) in the  $(0, 1)$  range of  $x$ . Its value is of interest in random testing, being the input bias probability which optimizes the average length of random test for the circuit. The fixed point value is shown to be very sensitive to the fan-ins of the logic gates. As the depth of the tree becomes infinite, the probability transfer function becomes a unit step with the transition point located at the interior fixed point. We study the convergence to the unit step as a function of the circuit depth and the fan-in's of the logic gates. The results are compared to other iteratively defined circuits whose building blocks also have an  $S$ -shaped transfer function.

**Index Terms**—AND-OR trees, asymptotic behavior, probabilistic response, random testing with bias, signal probabilities.

## I. INTRODUCTION

CONSIDER a random pattern of bits generated as follows: Each bit is set according to the outcome of an independent trial with probability  $x$  of success. In many applications related to testing, it is of interest to find *signal probabilities* (that is the probabilities of a one appearing on a line) in the circuit when such a random pattern is applied to the inputs. The *random pattern detectability* of a fault is defined as the probability that a random pattern will detect a stuck-type fault on a line. It can be expressed as a product of certain signal probabilities in the circuit [1], [2].

In this paper, the signal probabilities in a special class of circuits are analyzed. Our motivation for this study came from the earlier work by Agrawal and Agrawal [3], [4] who considered fan-out-free (tree) structures composed of  $n$ -input NAND gates. They derived the signal probabilities at the output of an  $l$ -level tree as a function of  $n$ ,  $l$ , and  $x$  and showed that for large  $l$ , the output signal probabilities alternated between close to zero and one (see also [5] for a similar result.) They also analyzed the detectability of primary input faults (which are the hardest to detect by random patterns) and reported the following results. For two-input NAND's (i.e.,  $n = 2$ ) the detectability achieves its maximum for a nonuniformly random

pattern (that is,  $x \neq 0.5$ ). The value of bias probability  $x$  at which the maximum occurs is denoted as  $x_{opt}$  which is a function of the logic depth  $l$ . For large  $l$ , the detectability is very sensitive to the value of  $x$ , with a sharp peak at  $x_{opt} = 0.617$ . Thus, random patterns derived with equal bias ( $x = 0.5$ ) are not likely to be very effective in testing such tree-type structures. A practical application of these results was shown by carrying out Monte Carlo experiments on several processing-unit circuit boards of the Illiac IV computer. The logic on these boards was approximately equivalent to NAND trees with the average gate fan-in between 2 and 3. The experiments clearly demonstrated that the random-pattern detectability is very sensitive to the bias probability, with the optimum results occurring close to  $x = 0.617$ .

The suggestion for using a bias probability derived from the average gate fan-in in the above approach is purely intuitive; it could lead to erroneous results if the signal probabilities (and hence the line detectabilities) in a tree were very sensitive to the changes in gate fan-ins. To answer the question of sensitivity, in this paper we study a larger class of tree structures with less restricted gate fan-ins. We consider trees in which the levels are filled alternately with  $m$ -input OR's and  $n$ -input AND's, where  $m, n \geq 2$ .<sup>1</sup> We define the *probability transfer function* (or simply the *transfer function*) of a tree as the probability of one at its output, expressed as a function of  $x$ , and show that it approaches a unit step function as the number of levels in the tree is increased. The position of the step along the  $x$ -axis (called the *firing point*) corresponds to the optimum bias probability  $x_{opt}$  of [4]. It is shown to be very sensitive to the values of  $m$  and  $n$ . We analyze how the transfer function approaches the unit step as the number of levels is increased. This is done in terms of a window of uncertainty, that is, the range of  $x$  for which the transfer function changes from zero to one. It is shown that the window width diminishes exponentially with the number of levels in the tree.

The unit-step transfer function of an AND-OR tree appears equivalent to the behavior of a threshold gate with random inputs. However, it is shown that the asymptotic behavior is approached very differently in the two cases. In the last part of the paper, the transfer function of an AND-OR tree is contrasted with that of a tree of threshold functions.

## II. TWO-LEVEL AND-OR TREES

Consider two-level logic circuits of the form shown in Fig. 1(a). At the first level, there are  $m$  AND gates, each connected

<sup>1</sup> These are a generalization of the  $n$ -input NAND trees considered in [3] and [4] since, by elementary Boolean algebra, each gate can be replaced by a NAND (or a NOR). An additional level of inverters needs to be added at the inputs or the output if the NAND tree had an odd number of levels.

Manuscript received May 27, 1987; revised January 9, 1988.  
L. Lipsky is with the Department of Computer Science, University of Connecticut, Storrs, CT 06268.

S. C. Seth is with the Department of Computer Science, University of Nebraska, Lincoln, NE 68588.  
IEEE Log Number 8930815.

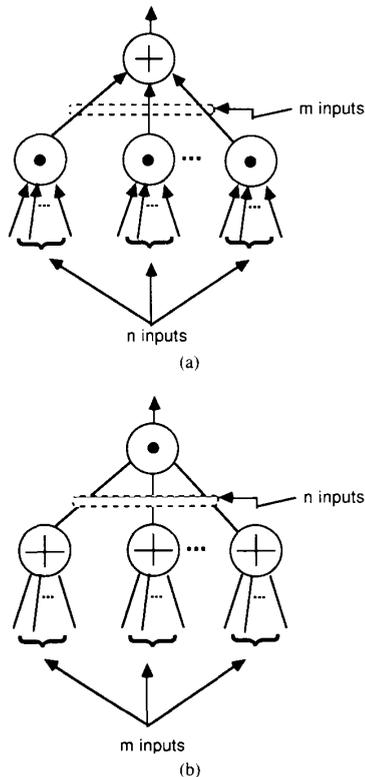


Fig. 1. Two-level AND-OR trees.

to a distinct set of  $n$  primary inputs. The outputs of the AND gates are connected to an  $m$ -input OR gate at the second level. The output of the OR gate is the primary output of the circuit. We will assume that each of the  $nm$  primary inputs is independently set to 1 with probability  $x$ , i.e., each input signal probability is  $x$ . Then the probability that the primary output is 1 is

$$g_V(x; n, m) = 1 - [1 - x^n]^m.$$

This is easily verified by noting that the quantity within the square brackets is the probability that an input to the OR gate is zero.

Similarly, for the dual case shown in Fig. 1(b), the output signal probability is

$$g_A(x; n, m) = [1 - (1 - x)^m]^n.$$

The duality of the two structures is captured in the following complementary relation:

$$g_A(1 - x; n, m) = 1 - g_V(x; m, n).$$

Functions, such as  $g_A$  and  $g_V$ , which represent the one-probability at the network output in terms of the input bias probability, will be called the (probability) *transfer function* of the network. Because of the duality, we need consider only one of the two structures shown in Fig. 1.

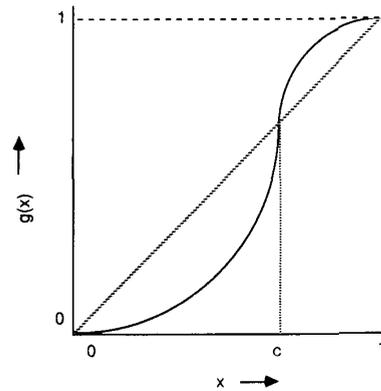


Fig. 2. The transfer function of a two-level AND-OR tree with the three fixed points.

### Basic Properties of the Transfer Functions

Let  $g(x)$  denote the transfer function  $g_V(x; n, m)$ . Also, let  $g^{(k)}(x)$  represent the  $k$ th derivative of  $g(x)$ . The following properties of  $g(x)$  are easily verified:

P1:  $g(0) = 0$ ;  $g(1) = 1$  (that is, 0 and 1 are fixed points<sup>2</sup> of  $g$ ).

P2:  $g^{(k)}(0) = 0$  for  $k = 0, 1, \dots, n - 1$ .

P3:  $g^{(k)}(1) = 0$  for  $k = 1, 2, \dots, m - 1$ .

P4:  $g(x)$  is positive and monotonically increasing in the interval  $[0, 1]$ .

P5: The slope of  $g(x)$  achieves its (unique) maximum value in the interval  $[0, 1]$  when  $x^n = (n - 1)/(nm - 1)$ , so  $g$  has one (and only one) *point of inflection* in the interval  $[0, 1]$ .

P6: From the last property, it follows immediately that  $g$  has exactly one interior *fixed point*  $c$  in the open interval  $(0, 1)$  (Fig. 2), see, for instance [6].

P7: If  $x > c$  [ $x < c$ ] then  $g(x) > x$  [ $g(x) < x$ ].

We shall refer to a function with the above properties as *S-shaped*.

### III. MULTILEVEL AND-OR TREES

There are many ways in which the two-level circuit considered above may be extended to multiple levels. Here we consider the simplest such extension. It is defined by the following restrictions:

- 1) there are two or more levels in the tree;
- 2) each level consists of either AND gates only or OR gates only;
- 3) the AND's and OR's alternate between successive levels; and
- 4) each AND gate has  $n$  inputs and each OR gate has  $m$  inputs, where  $n, m \geq 2$ .

The results reported in the literature [3]–[5] apply to the special case of constant fan-in trees ( $n = m$ ), with the binary case ( $n = m = 2$ ) analyzed most extensively.

Two simplifications allow further restriction of the class of circuits without any loss of generality:

First, because of the duality noted earlier, we need consider only trees whose root node is an OR gate. Second, the transfer

<sup>2</sup>  $x$  is a fixed point of the function  $g$  if  $g(x) = x$ .

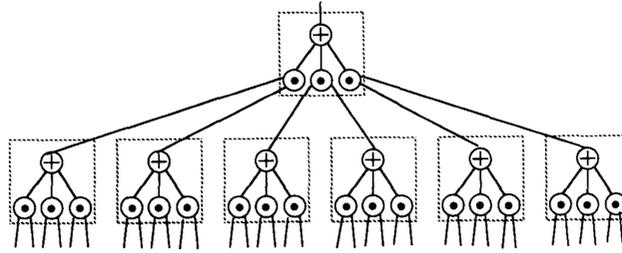


Fig. 3. A four-level AND-OR tree viewed as a two-layer tree.

function of an odd-level tree remains unchanged if we delete all the leaf-level AND gates and adjust the signal probabilities of the primary inputs (of the resulting structure) to  $x^n$ . Thus, it is enough to consider only even-level trees. Such a tree with  $2l$  levels will be viewed as an  $l$ -layer tree (Fig. 3) whose basic building block is the two-level structure shown in Fig. 1(a). Its transfer function will be denoted as  $g_l(x)$ . When  $l$  is 2, the transfer function is  $g(g(x))$ , that is,  $g_2(x) = g(g(x))$ . In general, the transfer function is obtained recursively as follows:

$$g_l(x) = g_{l-1}(g(x)).$$

It was noted earlier that  $g(x)$  is  $S$ -shaped. The following theorems show that this property holds for all  $g_l$ 's. Moreover, the transition part of the  $S$ -shaped curve becomes steeper with increasing  $l$ .

**Theorem 1:** Let  $g(x)$  denote the function  $g_V(x; n, m)$  with the fixed point  $c$ . Consider any point  $x$  in the half open interval  $[b, c)$  where  $b > 0$ . Then

$$g_l(x) < b \quad \text{for some finite } l,$$

that is,

$$\lim_{l \rightarrow \infty} g_l(x) = 0 \quad \text{for } 0 \leq x < c.$$

*Proof:* If  $g(x) < b$  then the theorem is true for  $l = 1$ . Otherwise, consider the sequence,  $g_1(x) > g_2(x) > \dots > g_l(x) \dots$ . From the property P7, the sequence is monotonically decreasing. Furthermore, it is bounded from below (by zero) and thus must have a greatest lower bound (say,  $\alpha$ ) which must be the limit of the sequence. Then  $\alpha$  must be a fixed point of  $g$  since  $\alpha = \lim_{l \rightarrow \infty} g_l(x) = \lim_{l \rightarrow \infty} g_{l+1}(x)$ . That is,  $\alpha = g(\alpha)$ . But by the properties of an  $S$ -shaped function, there can be only one interior fixed point of  $g$ ; hence,  $\alpha$  must be zero. Clearly then,  $g_l(x) < b$  for some  $l$ . ■

The following dual of Theorem 1 can be proved in a similar fashion.

**Theorem 2:** Let  $g(x)$  denote the function  $g_V(x; n, m)$  with the fixed point  $c$ . Consider any point  $x$  in the half open interval  $(c, d]$  where  $d < 1$ . Then

$$g_l(x) > d \quad \text{for some finite } l.$$

**Asymptotic Behavior**

We next study the asymptotic behavior of the transfer function  $g_l(x)$  as  $l$  approaches infinity. Let  $c$  be the interior fixed point of  $g(x)$ , i.e.,  $g(c) = c$ .

TABLE I  
FIRING POINT AS A FUNCTION OF GATE FAN-INS

		n						
		2	3	4	5	6	7	8
m	2	.618	.848	.921	.951	.967	.977	.982
	3	.389	.682	.805	.867	.902	.925	.940
	4	.282	.580	.724	.803	.850	.881	.903
	5	.220	.511	.666	.755	.810	.847	.873
	6	.181	.461	.622	.717	.778	.820	.849
	7	.153	.423	.587	.687	.752	.800	.829
	8	.133	.392	.558	.661	.730	.777	.812

From Theorem 1 we have

$$\lim_{l \rightarrow \infty} g_l(x) = 0 \quad \text{for } 0 \leq x < c.$$

Similarly, from Theorem 2, we can show that

$$\lim_{l \rightarrow \infty} g_l(x) = 1 \quad \text{for } c < x \leq 1$$

therefore,

$$\lim_{l \rightarrow \infty} g_l(x) = U(x; c) \quad \text{for } x \neq c$$

where  $U(x; c)$  is the unit step function at  $c$  (notice that at  $x = c$ ,  $g(c) = c$  while  $U(c; c) = 1$ ). We will use the term *firing point* for the input probability  $c$  at which the jump occurs in the output probability. The firing point has been computed for all  $2 \leq m, n \leq 8$ , and displayed in Table I. It may be noted that its value for  $m = n = 2$  is the same as the optimum bias probability obtained in [4] by Monte Carlo simulation. The values shown in the table, in contrast, were obtained by solving (polynomial) fixed point equations and could be computed to any desired degree of precision. In general, the firing point for any combination of  $m$  and  $n$  corresponds to the optimum bias probability for random testing.

Clearly, the firing point is very sensitive to the fan-in parameters  $m$  and  $n$ .

Suppose we have an AND-OR tree with large enough number of layers so as to approximate its asymptotic behavior. The addition of one more layer of logic to the tree is not going to affect the observed behavior of the circuit appreciably. But what if the building blocks of the added layer did not implement the same  $S$ -shaped function as those in the original circuit? The following analysis shows that we must distinguish the two cases shown in Fig. 4. These correspond to adding the

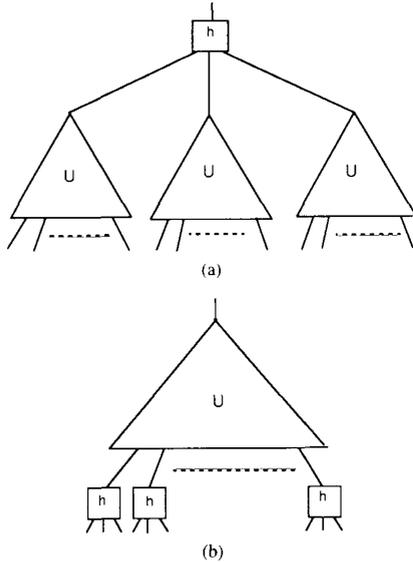


Fig. 4. Addition of an extra layer at the root (a) or at the leaves (b) of a large tree.  $U$  is the transfer function of the tree and  $h$  is the transfer function of the basic component in the added layer.

new layer, respectively, at the root or the leaves of the original tree. In the first case, Fig. 4(a), the augmented tree will have the transfer function  $h(U(x; c))$ , where  $h(x)$  is the transfer function of the new building blocks. For  $x < c$ ,  $U(x; c)$  is zero, therefore,  $h(U(x; c)) = h(0) = 0$ . Similarly, for  $x > c$ ,  $U(x; c)$  is one and so also is  $h(U(x; c))$ . In other words, *the effect of the root function  $h(x)$  is negligible on the asymptotic behavior*. In the second case, Fig. 4(b), the augmented tree will have the transfer function  $U(h(x); c)$  which fires when  $h(x) = c$ , or equivalently, when  $x = h^{-1}(c)$ . Thus, *the effect of the leaf function  $h(x)$  is to shift the firing point from  $c$  to  $h^{-1}(c)$* . It is easy to prove that the new firing point always lies in between  $c$  and  $c'$ , where  $c'$  is the interior fixed point of  $h(x)$  (see Fig. 5).

**Rate of Convergence**

An interesting question presents itself as to *how far from  $x = c$  can we expect nondeterministic behavior?* That is, if  $x$  is close to 0 we can always expect the output of a multilayer tree to be zero, while if  $x$  is close to 1, we can expect it to be always one. If  $x = c$ , then we know that, independently of  $l$ , the output will be one with probability  $c$ . There is some window about  $c$  for which the probability of one at the output is not *almost surely* 1 or 0. The window may be defined by the range of  $x$  in which  $g_l(x)$  rises from a low value  $y_0$  greater than zero (say, 0.1) to a high value  $y_1$  less than 1 (say, 0.9). Assume the corresponding range of  $x$  is  $[x_0, x_1]$ . Then the window width will be  $(x_1 - x_0)$ . Such a definition of the window, however, is not very convenient for studying the rate of convergence of  $g_l$  to a step function. We propose the following alternative to approximate the same concept (see Fig. 6).

*Definition:* The *window width* of an  $l$ -layer AND-OR tree is the inverse of the slope of  $g_l$  function at its interior fixed

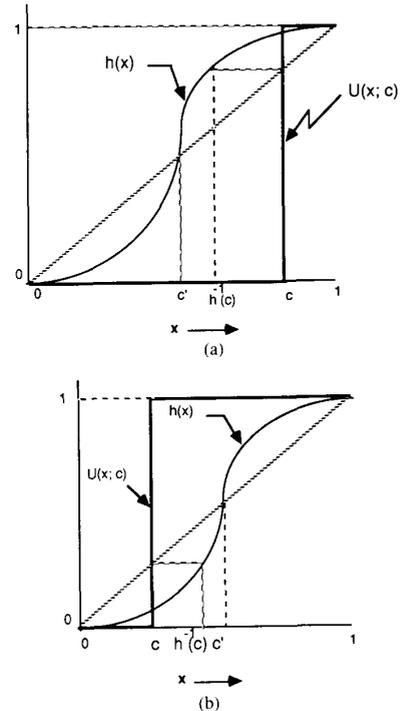


Fig. 5. The effect of a leaf function with the internal fixed point  $c'$  is to shift the firing point from  $c$  to  $h^{-1}(c)$  which lies between  $c$  and  $c'$ .

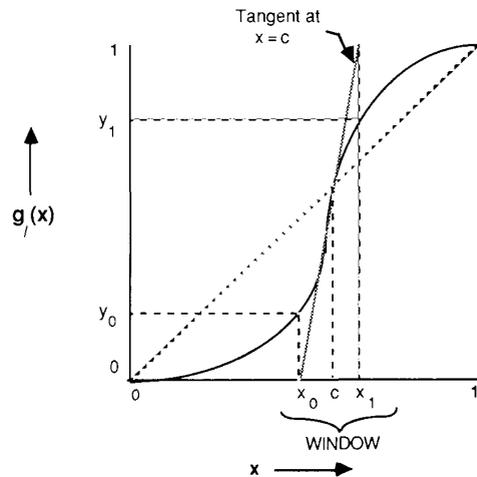


Fig. 6. The window of uncertainty may be defined as the inverse of the slope at the fixed point  $c$ .

point  $c$ , i.e.,

$$w_l = 1/g'_l(c)$$

with

$$x_0 = c - \frac{c}{g'_l(c)} \quad \text{and} \quad x_1 = c + \frac{(1-c)}{g'_l(c)}$$

A computer program was written to test the validity of this definition for different AND-OR trees of varying depths. In most cases,  $g_l(x_0) \approx 0.1$  and  $g_l(x_1) \approx 0.9$  and both were relatively

TABLE II  
WINDOW WIDTH VERSUS TREE DEPTH FOR BINARY TREES

No. of Layers $l$	Window width $w_l$	$z_0$	$z_1$	$y_0$	$y_1$
1	.655	.214	.868	.089	.939
2	.428	.353	.782	.107	.922
3	.280	.445	.725	.110	.914
4	.184	.505	.688	.110	.910
5	.120	.544	.664	.108	.908
6	.079	.569	.648	.107	.907
7	.051	.586	.638	.105	.906
8	.034	.597	.631	.105	.905
9	.022	.604	.626	.104	.905
10	.014	.609	.624	.104	.905

independent of  $l$ . Sample calculations are shown in the Table II for binary trees, i.e.,  $m = n = 2$ .

We notice that  $w_l$  indeed approximates an ideal window quite well. A geometrical narrowing of the window with the number of levels is also apparent. This is a consequence of the following theorem.

**Theorem 3:** The window width of an AND-OR tree decreases exponentially with its number of layers  $l$ . Specifically,

$$w_l = (w_1)^l$$

where  $w_1$  is the window width of a single-layer tree given by  $1/g'(c)$ .

**Proof:** By definition,  $w_l = 1/g'(c)$ . Now, for  $l = 2$ , we have

$$g'_2(x) = [g(g(x))]' = g'(g(x))g'(x).$$

Therefore,

$$g'_2(c) = [g'(c)]^2.$$

In general, it can be shown by induction that

$$g'_l(c) = [g'(c)]^l$$

from which the theorem follows immediately. ■

As a corollary to the theorem, for each increment of  $l$ , the window width will decrease by the factor  $w_1$ . For the binary tree example considered above,  $w_1 = 0.655$ ; hence, each window width in Table II is roughly two thirds of the immediately preceding value.

#### AND-OR Trees Versus One-Level Threshold Functions

We saw from Theorem 3 and the empirical data in Table II that the *width of uncertainty* narrows dramatically with the depth of the tree. Intuitively, it may be argued that the major reason for this phenomenon is really the exponential rise in the number of inputs to the tree with the depth. To be specific, as the number of inputs  $(nm)^l$  grows, the probability that the fraction of 1's will deviate appreciably from  $x$  decreases as  $1/\sqrt{(nm)^l}$ . But if the asymptotic behavior of an AND-OR tree could be explained entirely, or even to a large extent, by the probability of having a certain number of 1-inputs, then there would be no way of differentiating the tree circuits from a single-level threshold function [7] with an appropriately chosen threshold. The argument below will show that such is not the case.

If the firing point of a tree with  $N$  inputs is  $c$ , then it is clear that the chosen threshold for the single-level threshold function must be  $cN$ . Therefore, we define the corresponding transfer function as

$$G(x; k) = \sum_{k \geq cN} \binom{N}{k} x^k (1-x)^{N-k}$$

where  $G(0; k) = 0$  and  $G(1; k) = 1$ .

It is known (see, for example, [8, p. 174 ff]) that for large  $N$  and fixed  $x$ , the largest term in the above sum is

$$\binom{N}{k_m} x^{k_m} (1-x)^{N-k_m} \quad \text{with } k_m \approx xN$$

and that only those terms in the range  $k_m \pm 2x\sqrt{N}$  are comparable in size. It then follows that  $G(x; k_m)$  is very small if  $x < c - 2c/\sqrt{N}$  and it must be very close to 1 if  $x > c + 2c/\sqrt{N}$ . Thus, the window width of  $G(x; k_m)$  is  $\approx 4c/\sqrt{N}$ . To compare this to the AND-OR trees discussed earlier, we must choose the number of inputs  $N$  as  $(mn)^l$ . That is, the window width for the one-level threshold narrows as  $(1/\sqrt{mn})^l$  as compared to  $(1/g')^l$  for the AND-OR tree. Then, as  $l$  increases, the threshold function has a narrower width than the comparable AND-OR tree. For example, for binary trees,  $n = m = 2$ , and  $g'(c) \approx 1.527$  for  $c = 0.618$ . Then for  $l > 5$ ,  $4c(1/2)^l < (1/1.527)^l$ . Thus, the threshold function has a sharper approach to the unit step than the AND-OR tree.

#### Trees of Threshold Functions

It may also be interesting to compare the AND-OR tree behavior to a *tree* of threshold functions, instead of just a one-level tree. For example, the threshold function *majority of 3 bits* has the transfer function  $G(x; 2) = 3x^2 - 2x^3$ , with fixed points (i.e.,  $G(x; 2) = x$ ) at  $x = 0, 1$ , and  $1/2$  as expected. However, for the threshold function corresponding to the majority of an even number of bits, the firing point can be quite different than one would expect. For instance, for a threshold gate implementing the *majority of 4 bits*, the firing point is 0.232. This implies that in a large tree of such threshold gates, it takes less than 25 percent of the inputs randomly set to 1 to make the output almost surely a 1. In general, for the gate fan-in  $n$  and threshold  $\tau$ , the firing point  $F(n, \tau)$  satisfies the following relationships:

$$F(n, \tau) < \frac{\tau - 1}{n - 1} \quad \text{for } \tau \leq n/2$$

$$F(n, \tau) \geq \frac{\tau - 1}{n - 1} \quad \text{for } \tau > n/2$$

$$F(2\tau - 1, \tau) = \frac{1}{2}$$

$$G(x; n, \tau) = x \Leftrightarrow G(1-x; n, n-\tau+1) = (1-x).$$

Table III shows the firing points of threshold trees for a few examples.

TABLE III  
FIRING POINT OF A TREE OF THRESHOLD GATES

Threshold $\tau$	Fan-in $n$			
	3	4	5	6
2	0.500	0.232	0.131	0.084
3	1.0	0.767	0.500	0.347
4		1.0	0.869	0.652
5			1.0	0.916

#### IV. CONCLUSION

The AND-OR trees in this paper generalize the class of the fixed fan-in NAND trees discussed in the literature and have been shown to have applications in random-pattern testing. The input vectors are composed of bits independently chosen to be one with a fixed probability  $x$ . We have used the probability transfer function of a two-level tree as the basis for analysis since it plays a key role in defining the random-pattern testability of the network. This S-shaped function finds the output signal probability as a function of the input bias probability  $x$  and has exactly one fixed point other than zero or one. It is shown that viewing each layer of an iterated tree as a fixed point function brings out many properties of the tree. In particular, it is shown that the probability transfer function of every iterated tree approaches the step function, where the step is at  $c = g(c)$ . While our analysis is in terms of regular AND-OR trees, we conjecture that the behavior of any large AND-OR tree would resemble a step function for sufficiently large depths. However, the position of the step will depend strongly on the specific details of the trees, being more dependent on the nodes near the leaves than the root.

The limiting behavior of the AND-OR trees (as its size is increased) resembles that of a single-level threshold gate but the approach to the limiting behavior is shown to be quite different in the two cases.

#### ACKNOWLEDGMENT

The authors would like to thank Prof. G. Nagy for his suggestion to look at trees of threshold functions as an alternative to AND-OR trees.

#### REFERENCES

- [1] J. Savir, G. S. Ditlow, and P. H. Bardell, "Random pattern testability," *IEEE Trans. Comput.*, vol. C-33, pp. 79-90, Jan. 1984.
- [2] S. C. Seth, B. B. Bhattacharya, and V. D. Agrawal, "An exact analysis for efficient computation of random pattern testability in combinational circuits," in *16th Int. Symp. Fault Tolerant Computing (FTCS-16) Dig. Papers*, Vienna, Austria, 1986, pp. 318-323.
- [3] P. Agrawal and V. D. Agrawal, "Probabilistic analysis of random test generation method for irredundant combinational logic networks," *IEEE Trans. Comput.*, vol. C-24, pp. 691-695, July 1975.
- [4] —, "On Monte Carlo testing of logic tree networks," *IEEE Trans. Comput.*, vol. C-25, pp. 664-667, June 1976.

- [5] B. Krishnamurthy, "An interesting property of and-or trees," *Tektronix, Comput. Res. Lab. Rep. CR-85-15*, Apr. 1985.
- [6] S. D. Conte and C. de Boor, *Elementary Numerical Analysis—An Algorithmic Approach*, 3rd. ed. New York: McGraw-Hill, 1980.
- [7] M. Dertouzos, *Threshold Logic: A Synthesis Approach*. Cambridge, MA: MIT Press, 1965.
- [8] W. Feller, *An Introduction to Probability Theory and Its Applications*. New York: Wiley, 1968.



**Lester Lipsky** was born in The Bronx, New York, and received a bachelor's degree in mechanical engineering from C.C.N.Y. in 1956. He attended Brandeis University from 1956-1958, and then worked for Pratt and Whitney Aircraft until 1961. He received the Ph.D. degree in theoretical atomic physics from The University of Connecticut, Storrs, in 1965.

He then spent a year as a Postdoctoral Fellow at the National Bureau of Standards in Washington, DC, followed by two years as a Visiting Research Fellow at Royal Holloway College of the University of London, London, England. In 1968, he joined the Department of Computer Science at the University of Nebraska, Lincoln, as Assistant, then Associate (1971), and then Full Professor (1976). In 1976-1977, he was a Senior Research Fellow at the University of Nijmegen in The Netherlands. Since 1986, he has been a Professor of Computer Science and Engineering at the University of Connecticut. His major research interests are in the analytical modeling of the performance of computer systems and networks, and related mathematical and numerical problems. In recent years, he has become increasingly interested in the theoretical aspects of queueing theory, particularly nonsteady-state phenomena, using the matrix-algebraic formulation. He has authored or co-authored over 40 research articles.

Dr. Lipsky is a member of the Association for Computing Machinery, the American Physical Society, Sigma Xi, Upsilon Pi Epsilon, and Sigma Pi Sigma.



**Sharad C. Seth** (S'66-M'70-SM'82) received the B.Eng. degree in 1964 from University of Jabalpur, India, the M.Tech. degree in 1966 from the Indian Institute of Technology, Kanpur, and the Ph.D. (EE) degree in 1970 from the University of Illinois, Urbana-Champaign.

Since 1970 he has been with the Department of Computer Science at the University of Nebraska, Lincoln, where he is currently a Professor teaching courses in the areas of VLSI/design/testing and computer architecture. He has held visiting positions at the Indian Institute of Technology, Kanpur, during the years 1974-1975 and 1982-1983. He has worked at the Bell Laboratories, Murray Hill, NJ, as a member of Technical Staff and a consultant since 1980.

Dr. Seth was a member of the program committees of the following IEEE conferences: International Test Conference (1982-1986) and the International Symposium on Fault Tolerant Computing (1988-1989). He is the Editor of Short Papers for the *IEEE Design & Test of Computers* magazine. He has authored or coauthored over forty technical papers in the areas of testing, switching theory, and analysis of optically scanned documents and has coauthored the tutorial text *Test Generation for VLSI Chips*. He is a member of the IEEE Computer Society and the Association for Computing Machinery.