JOURNAL OF ELECTRONIC TESTING: Theory and Applications, 2, 315–323 (1991)

Bounding Fault Detection Probabilities in Combinational Circuits

GEORGE MARKOWSKY[1]
Computer Science Department, University of Maine, Orono, ME 04469

Received February 27, 1990. Revised June 7, 1991.
Editor: B. Krishnamurthy

Abstract. This paper focuses on the problem of bounding fault detection probabilities in combinational circuits. Two algorithms, the complete cutting algorithm and the gate blocking algorithm, are presented that always produce true lower bounds on the detection probability of a fault. Both algorithms can be used to identify difficult-to-test faults and to quickly construct test sets for specific faults. Both algorithms have qualitative versions which provide insight into a circuit while avoiding arithmetic calculation. Both algorithms resulted from research in trying to determine the accuracy of the safety factor heuristic of Jacob Savir.

Keywords: Bounding probabilities, detection probability, safety factor heuristic, testing.

1. Introduction

Random testing is an important technique for testing computer circuits. To construct a test set correctly it is important to know the probability that each detectable fault in a circuit can be detected at random (see Savir and Bardell [12] for details). In general, computing this probability is a #P-complete problem (see Garey and Johnson [5]) so we would not expect any fast algorithm to always produce a useful answer. Nevertheless, because of the practical importance of the problem many authors have examined a variety of approaches to this problem and the closely related problem of computing signal probabilities (see [1]–[4], [6], [7], [9]–[13]).

In Savir, Ditlow and Bardell [11] circuit cutting algorithms were introduced for bounding the signal probability of a line (see Markowsky [9] for a complete analysis). Also in [11] it was shown how "auxiliary gates" could be used to reduce the fault detection probability problem to the signal probability problem.

Unfortunately, algorithms based on auxiliary gates can return 0 as the detection probability for faults that have a nonzero detection probability. Furthermore, the auxiliary gate approach slows analysis by increasing circuit complexity.

In some proprietary notes, Jacob Savir described a heuristic, which he calls the *safety factor heuristic*, to estimate the fault detection probability of certain faults. This technique often gives very accurate estimates but has the failing that the estimates could be greater than the true probability. This paper presents algorithms that *always* return true lower bounds.

Besides Savir, other authors have studied algorithms for approximating the detection probability of a fault. Brglez [1] studies this question together with the susceptibility to random testing of large combinational networks. Seth and Agrawal [13] study this question and provide a general model for understanding the problem.

[1]This research was supported by a grant from the IBM Corporation.

1.1. Notation

B will denote the two element set $\{0, 1\}$, and for any natural number, n, $\mathbf{B}^n$ will denote the set of Boolean n-tuples. $BF(n)$ will denote all functions $f:\mathbf{B}^n \to \mathbf{B}$ and the Boolean operations AND, OR, NOT will be represented by &, | and $\sim$.

As is well known, every function in $BF(n)$ can be represented using n variables and &, | and $\sim$. *$\mathbf{B}^n$ will be considered a probability space with the uniform measure. We will use $P(X)$* to denote the probability of $X \subseteq \mathbf{B}^n$. *Given a combinational circuit, C,* the set of inputs to C may be identified with $\mathbf{B}^n$ where n is the number of primary inputs of C.

Definition 1. Let C be a combinational circuit with a single output line z. Further, let q be an arbitrary line in C and p a fault in q. Let A_q be the set of inputs to C

which cause the value 1 to appear in q, and let B_p be the set of inputs to C for which the value produced at z differs depending on whether the fault p is present in C. Thus, B_p is the set of inputs that detect the fault p.

a) The *signal probability* of q is $P(A_q)$.
b) The *fault detection probability* of p is $P(B_p)$. We will sometimes use $P_d(p)$ to represent the detection probability of p.

Generally, calculating the fault detection probability of a fault is more difficult than calculating the signal probability of the line containing that fault, since a fault can only be detected under appropriate sensitizing conditions. For example, consider the circuit shown in figure 1.

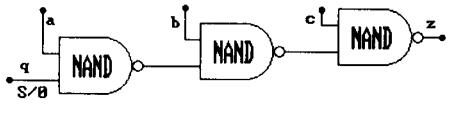

Fig. 1. A simple circuit.

To detect the stuck-at-zero fault in line q at output z requires setting lines q, a, b, c to 1. Determining the probability of this happening is at least as involved as determining the signal probability of line q. If the lines a, b, c and q are independent, the probability of detecting a stuck-at-zero fault in line q is simply the product of the signal probabilities of lines a, b, c and q. The bane of most circuit analysis is that independence cannot be assumed between arbitrary lines in circuits.

Consider the circuit in figure 2 and the three faults X_1 stuck-at-zero ($X_1 - s/0$), X_3 stuck-at-zero ($X_3 - s/0$), and a stuck-at-one ($a - s/1$). We will briefly describe the results that the safety factor heuristic obtains for the three faults just mentioned. The steps of the safety factor heuristic are:

1. Calculate signal probabilities of all lines in the tree part of the circuit, i.e., all lines which are not fed by reconvergent fanout.
2. Use the full-range cutting algorithm (see Savir, Ditlow and Bardell [11]) to estimate the signal probabilities left untouched in step 1. The full-range cutting algorithm can be run several times to obtain better bounds.
3. Combine steps 1 and 2 to get the best estimate of signal probability for each line in the circuit.
4. For each fault of interest, trace a path from the fault to a primary output. Compute an estimate for the detection probability by assuming that all relevant lines are independent.
5. Multiply the estimates in step 4 by a safety factor that is determined by the network topology and the relationship of the fault to the rest of the network.

The hope is that this final result is a true lower bound on the detection probability of the fault. We will not discuss how the safety factor is estimated since we are interested in presenting the overall strategy rather than in the particulars of the algorithm.

To motivate the subsequent discussion we briefly sketch how the safety factor heuristic is used to analyze a circuit. Figure 3 shows what the circuit of figure 2 looks like at the end of step 1. Only lines e and z do not have a signal probability since they are the only lines which are fed by reconvergent fanout.

To cut all reconvergent fanout in the circuit of figure 2 requires cutting one of lines b or b' and one of lines c or c'. Thus there are four possible sets of cuts: $Cuts_1 = \{b, c\}$; $Cuts_2 = \{b, c'\}$; $Cuts_3 = \{b', c\}$; $Cuts_4 = \{b', c'\}$.

Figures 4 to 7 show all four pairs of cuts that cut all reconvergent fanout. Following the full-range cutting algorithm, all cut lines receive a lower bound of 0 and an upper bound of 1 on their signal probability. These

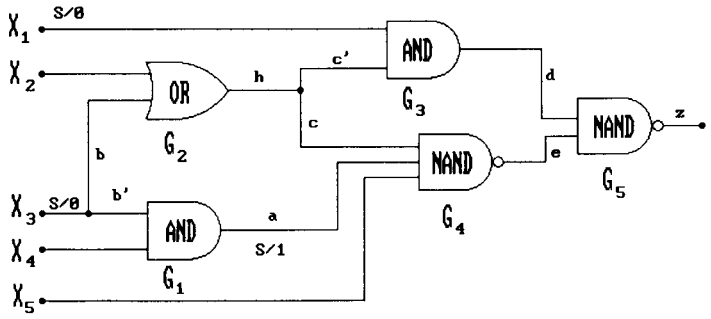

Fig. 2. A sample circuit for analysis.

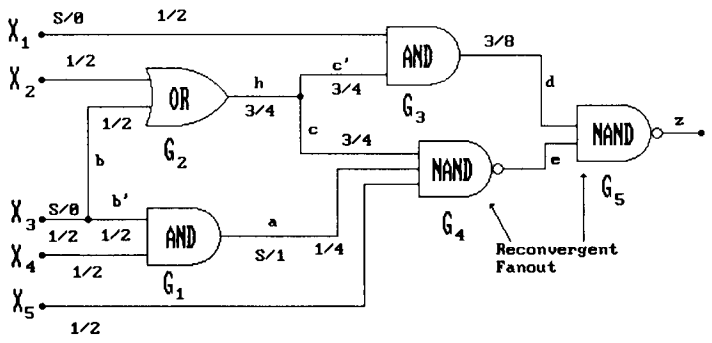

Fig. 3. After step 1 of the safety factor heuristic.

Table 1. A comparison of different techniques.

			Complete Cutting				Gate Blocking			
Fault	True Detection Probability	Safety Factor Estimate	Lower Bound $Cuts_1$	Lower Bound $Cuts_2$	Lower Bound $Cuts_3$	Lower Bound $Cuts_4$	Lower Bound $X_3 = 0$	Lower Bound $X_4 = 0$	Lower Bound $X_2 = 1$	Lower Bound $X_3 = 1$
$X_1 - s/0$	20/64	21/64	14/64	0	18/64	0	8/64	12/64	14/64	12/64
$a - s/1$	32/256	27/512	0	0	0	0	N/A	N/A	24/256	16/256
$X_3 - s/0$	16/128	7/128	0	0	12/128	0	N/A	8/128	4/128	N/A

bounds are propagated as if all lines are independent and the results yield correct bounds. Different cuts can be used to get better values. The results produced by the safety factor heuristic are summarized in table 1 along with the results produced by the algorithms discussed in this paper. Note that the estimates produced for $a - s/1$ and $X_3 - s/0$ are lower bounds for the detection probabilities for those faults while the estimate produced for $X_1 - s/0$ is not a lower bound. At this time, there are no techniques for deciding how good an estimate is produced by the safety factor heuristic.

2. The Complete Cutting Algorithm

The complete cutting algorithm is a slight modification of the safety factor heuristic algorithm and is similar to the algorithm described by Gaede, Mercer and Underwood [4]. It is more conservative than the safety factor heuristic, but always computes true lower bounds. The steps of this algorithm are:

1. Cut the least number of reconvergent fanout lines so the circuit has no remaining reconvergent fanout.
2. Let [0, 1] be the bounds on the signal probability of every cut line.
3. Propagate the bounds throughout the circuit to derive bounds on the signal probability of each line.
4. For each fault of interest, trace a path from the fault to a primary output. Compute an estimate for the detection probability by assuming that all relevant lines are independent.

The complete cutting algorithm requires that all reconvergent fanout must be cut before estimating the detection probabilities. To illustrate the complete cutting algorithm, let's apply it to the circuit of figure 2 for the four sets of cuts considered earlier.

For figure 4 we estimate the detection probabilities of the three faults using the inequalities

$$P_d(X_1 - s/0) \geq P(X_1 = 1) \times P(c' = 1) \times P(e = 1)$$
$$\geq 1/2 \times 1/2 \times 7/8 = 7/32 = 14/64$$

$$P_d(a - s/1) \geq P(a = 0) \times P(c = 1) \times P(X_5 = 1) \times P(d = 1)$$
$$\geq 3/4 \times 0 \times 1/2 \times 1/4 = 0$$

$$P_d(X_3 - s/0) \geq P(b' = 1) \times P(X_4 = 1) \times P(c = 1) \times P(X_5 = 1) \times P(d = 1)$$
$$\geq 1/2 \times 1/2 \times 0 \times 1/2 \times 1/4 = 0$$

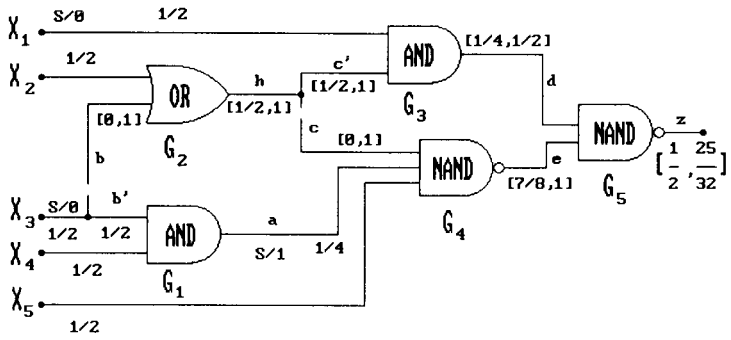

Fig. 4. The complete cutting algorithm for the first set of cuts.

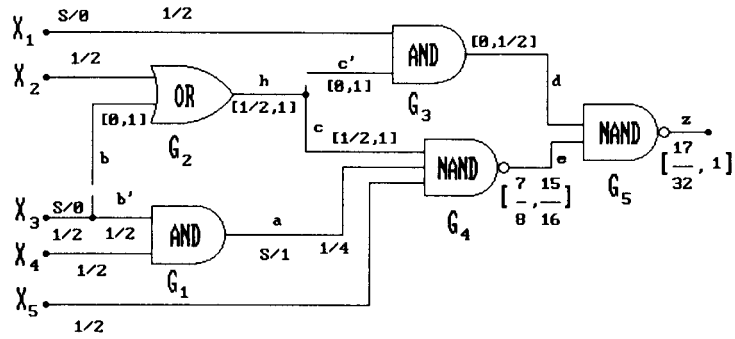

Fig. 5. The complete cutting algorithm for the second set of cuts.

where P_d denotes the detection probability and P denotes the probability of the corresponding signal being in the line. If P is not known exactly, a lower bound for it must be used. The quantities needed to bound the detection probabilities can read directly from figure 4.

For figure 5 we get

$$P_d(X_1 - s/0) \geq P(X_1 = 1) \times P(c' = 1) \times P(e = 1)$$
$$\geq 1/2 \times 0 \times 7/8 = 0$$
$$P_d(a - s/1) \geq P(a = 0) \times P(c = 1) \times P(X_5 = 1) \times P(d = 1)$$
$$\geq 3/4 \times 1/2 \times 1/2 \times 0 = 0$$
$$P_d(X_3 - s/0) \geq P(b' = 1) \times P(X_4 = 1) \times P(c = 1) \times P(X_5 = 1) \times P(d = 1)$$
$$\geq 1/2 \times 1/2 \times 1/2 \times 1/2 \times 0 = 0$$

and this does not supply any useful bounds.

For figure 6 we get

$$P_d(X_1 - s/0) \geq P(X_1 = 1) \times P(c' = 1) \times P(e = 1)$$
$$\geq 1/2 \times 3/4 \times 3/4 = 9/32 = 18/64$$
$$P_d(a - s/1) \geq P(a = 0) \times P(c = 1) \times P(X_5 = 1) \times P(d = 1)$$
$$\geq 1/2 \times 0 \times 1/2 \times 3/8 = 0$$
$$P_d(X_3 - s/0) \geq P(b = 1) \times P(X_2 = 0) \times P(X_1 = 1) \times P(e = 1)$$
$$\geq 1/2 \times 1/2 \times 1/2 \times 3/4 = 3/32 = 12/128$$

Finally, for figure 7 we get

$$P_d(X_1 - s/0) \geq P(X_1 = 1) \times P(c' = 1) \times P(e = 1)$$
$$\geq 1/2 \times 0 \times 13/16 = 0$$
$$P_d(a - s/1) \geq P(a = 0) \times P(c = 1) \times P(X_5 = 1) \times P(d = 1)$$
$$\geq 1/2 \times 3/4 \times 1/2 \times 0 = 0$$
$$P_d(X_3 - s/0) \geq P(b = 1) \times P(X_2 = 0) \times P(a = 1) \times P(X_5 = 1) \times P(d = 1)$$
$$\geq 1/2 \times 1/2 \times 0 \times 1/2 \times 0 = 0$$

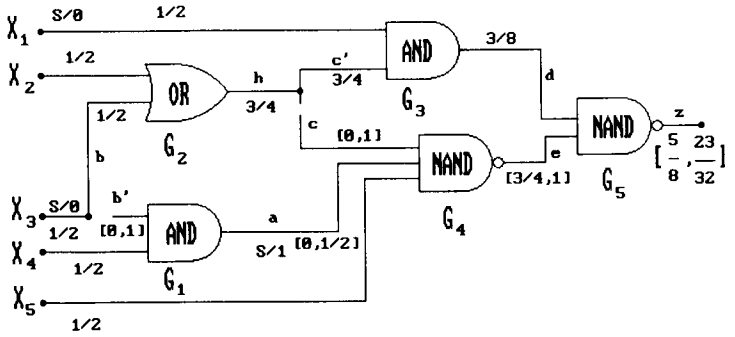

Fig. 6. The complete cutting algorithm for the third set of cuts.

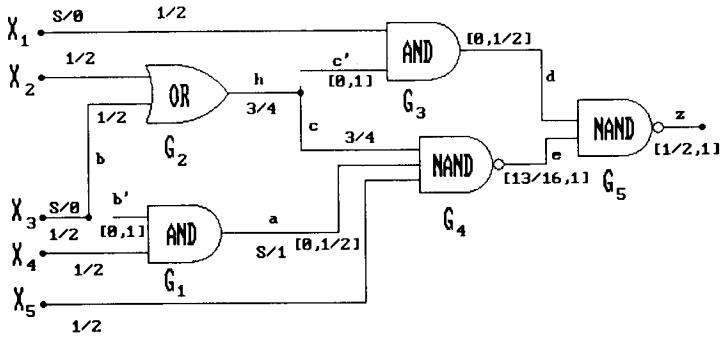

Fig. 7. The complete cutting algorithm for the fourth set of cuts.

Table 1 includes all the results obtained by the complete cutting algorithm for the circuit pictured in figure 2. Note that the complete cutting algorithm always produces lower bounds. For the $X_3 - s/0$ fault it even produces a better lower bound than the safety factor heuristic. On the other hand, the complete cutting algorithm fails to produce a nonzero lower bound for the fault $a - s/1$ whereas the safety factor heuristic produces a correct lower bound. We conclude this section by justifying the complete cutting algorithm.

Definition 2. Let f, g be in $BF(n)$. We say $f \leq g$ if and only if for all x in B^n, $f(x) = 1$ implies that $g(x) = 1$.

Lemma 1. Let f_1, f_2, g_1, g_2 be in $BF(n)$ be such that $f_1 \leq g_1$ and $f_2 \leq g_2$. Then $f_1 \,\&\, f_2 \leq g_1 \,\&\, g_2$, $f_1 \mid f_2 \leq g_1 \mid g_2$ and $\tilde{}f_1 \geq \tilde{}g_1$.

Proof. This proof is straightforward and left to the reader.

Lemma 2. Let f, g in $BF(n)$ be such that when f and g are expressed in terms of Boolean operators and n variables, they have no variables in common. Then for all a, b in B

$$P((f = a) \,\&\, (g = b)) = P(f = a) * P(g = b).$$

Proof. Suppose f uses the variables $X_1, \ldots, X_k$ and g uses the variables $X_{k+1}, \ldots, X_{k+j}$ where $k + j \leq n$. The variables can always be renumbered to make this true. Note that f may be considered a member of B^k and g a member of B^j. Let

$$W = \{w \text{ in } B^k \mid f(w) = a\}$$

$$Y = \{y \text{ in } B^j \mid g(y) = b\}.$$

Now

$$\begin{aligned} P((f = a) \,\&\, (g = b)) &= (|W \times Y| \times 2^{n-(j+k)})/2^n \\ &= |W \times Y|/2^{(j+k)} = (|W| \times 2^{n-k}/2^n) \\ &\quad \times (|Y| = 2^{n-j}/2^n) \\ &= P(f = a) \times P(g = b), \end{aligned}$$

where we have used $|X|$ to denote the cardinality of the set X.

Theorem 1. *The complete cutting algorithm is correct.*

Proof. Given a combinational circuit C, to each line assign the Boolean function that describes the signal that appears on that line. Given the line q, f_q will denote the function that normally should appear on q.

By induction we will assign two additional functions, L_q and U_q to the line q. These functions will have the following properties.

a) $L_q \leq f_q \leq U_q$
b) If two lines q and q' meet at a gate, the functions L_q, U_q, $L_{q'}$, $U_{q'}$ can be expressed in terms of Boolean variables so that functions associated with different lines have no variables in common.

In the base case of the induction we assign $L_q = f_q = U_q$ to all primary input lines q. Also, all lines that are cut receive the function that is identically 0 as L_q and the function that is identically 1 as U_q. At this point it is clear that Property (a) is satisfied.

Now proceed inductively through the gates. If q and q' meet at an AND gate which has q'' as an output line, set $L_{q''} = L_q \,\&\, L_{q'}$ and $U_{q''} = U_q \,\&\, U_{q'}$. By Lemma 1, Property (a) holds for q''. These results extend naturally to OR and NOT gates, and to gates having more than 2 input lines.

To see that Property (b) holds, simply use the representation of L_q and U_q that arises from the initial assignments and using the Boolean operations implied by the gates. Since the "cut" circuit is a tree it is impossible for two lines that meet at a gate to have any variables in common.

Now we can turn our attention to the problem of estimating detection probabilities. Given a fault and a path from that path to the output, the probability of detecting that fault can be expressed in the form $P((f_{q_1} = a_1) \,\&\, (f_{q_2} = a_2) \,\&\, \ldots \,\&\, (f_{q_k} = a_k))$ where $q_1, q_2, \ldots, q_k$ are the lines needed to sensitize the path and detect the fault, and a_i is in B for all i. Let X be the event

$$(f_{q_1} = a_1) \,\&\, (f_{q_2} = a_2) \,\&\, \ldots \,\&\, (f_{q_k} = a_k)$$

and Y the event

$$(h_{q_1} = a_1) \,\&\, (h_{q_2} = a_2) \,\&\, \ldots \,\&\, (h_{q_k} = a_k)$$

where

$$h_{q_i} = L_{q_i} \text{ if } a_i = 1 \text{ and } h_{q_i} = U_{q_i} \text{ if } a_i = 0.$$

It is easy to se that $Y \subseteq X$ so that $P(Y) \leq P(X)$.

Theorem 1 now follows by using Lemma 2 inductively.

3. The Gate Blocking Algorithm

Table 1 shows that in all cases the complete cutting algorithm only produces the trivial lower bound of 0 for the fault $a - s/1$. This is because the complete cutting algorithm requires that either c or c' be cut. Thus, one of the two lines will always supply a zero to the product used to bound the detection probability. To detect $a - s/1$ both c' and c must have the value 1.

For the fault $a - s/1$, the complete cutting algorithm fails to produce a useful lower bound because too many lines are cut. This suggests looking at algorithms that require fewer cuts. The gate blocking algorithm makes fewer cuts but still reduces a combinational circuit to one without reconvergent fanout. In essence, it simulates cuts by setting up the primary inputs appropriately and propagating the results. Similar ideas have been explored by several authors (see Chakravarty and Hunt [2] and [3], and Savir [10]).

The steps of the gate blocking algorithm are as follows.

1. Identify all gates that are on reconvergent fanout paths, but which are not fed by any gates that are on reconvergent fanout paths. Call such gates **minimal gates**.
2. For each minimal gate examine the consequences of **blocking** that gate, i.e., of setting one of its input lines to 0 for an AND or NAND gate or to 1 for an OR or NOR gate. The gate is said to be blocked because the values assigned to the other lines are irrelevant. Examining the consequences means tracing the ouptut of the gate through the circuit and removing irrelevant parts of the circuit.
3. Repeat Step 2 until the circuit contains no more reconvergent fanout.
4. Use the techniques of Section 2 to produce lower bounds on the detection probabilities in the reduced circuits. Multiply the lower bounds by the probability associated with the blocking conditions.

We illustrate the gate blocking algorithm on the circuit shown in figure 2. To begin with, it is clear that G_1 and G_2 are the only minimal gates in the circuit. G_1 can be blocked by setting X_3 or X_4 to 0. G_2 can be blocked by setting X_2 or X_3 to 1.

Setting X_3 to 0 causes a to go to 0 which blocks G_4. The circuit of figure 2 is effectively reduced to the circuit shown in figure 8. Figure 8 shows the signal probability of each line calculated using the standard techniques.

Of the three faults considered in table 1, only the fault $X_1 - s/0$ can appear in the circuit of figure 8. The probability of detecting the single fault $X_1 - s/0$ in the circuit of figure 8 is 1/4. To convert this to a lower bound for the circuit of figure 2 we must multiply by $P(X_3 = 0)$ which is 1/2. Thus our lower bound for the detection probability of $X_1 - s/0$ in this example is $1/8 = 8/64$.

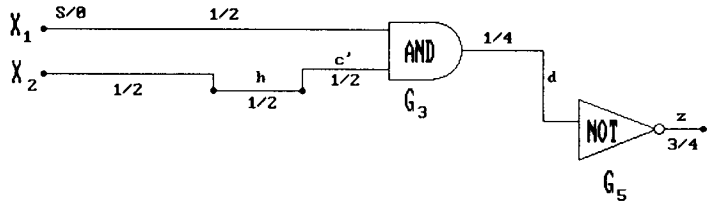

Fig. 8. The effect of blocking G_1 by setting $X_3 = 0$.

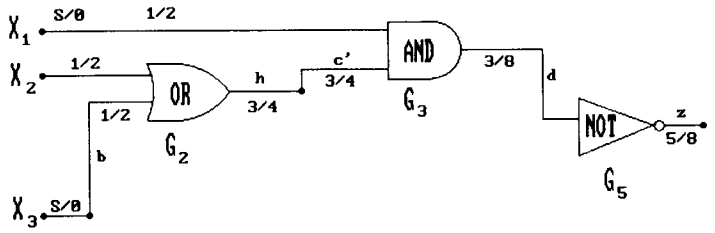

Fig. 9. The effect of blocking G_1 by setting $X_4 = 0$.

Setting X_4 to 0 yields the circuit of figure 9. This time we can produce nontrivial lower bounds for $X_1 - s/0$ and $X_3 - s/0$. In the first case we get $1/2 \times 1/2 \times 3/4 = 3/16 = 12/64$ and in the second we get $1/2 \times 1/2 \times 1/2 \times 1/2 = 1/16 = 8/128$.

Now we examine the results of blocking G_2. If we set $X_2 = 1$, the circuit in figure 10 results. All three faults considered in table 1 can be analyzed from figure 10. Thus

$$\begin{aligned} P_d(X_1 - s/0) &\geq P(X_1 = 1) \times P(e = 1) \\ &\quad \times P(X_2 = 1) \\ &\geq 1/2 \times 7/8 \times 1/2 = 7/32 = 14/64 \\ P_d(a - s/1) &\geq P(a = 0) \times P(X_5 = 1) \times P(X_1 = 1) \\ &\quad \times P(X_2 = 1) \\ &\geq 3/4 \times 1/2 \times 1/2 \times 1/2 = 3/32 \\ &\quad = 24/256 \\ P_d(X_3 - s/0) &\geq P(X_3 = 1) \times P(X_4 = 1) \\ &\quad \times P(X_5 = 1) \times P(d = 1) \\ &\quad \times P(X_2 = 1) \\ &\geq 1/2 \times 1/2 \times 1/2 \times 1/2 \times 1/2 \\ &\quad = 1/32 = 4/128 \end{aligned}$$

Note how the blocking condition, $X_2 = 1$ must be figured into each one of the detection probability lower bounds.

Finally, block G_2 by setting $X_3 = 1$ yields the circuit of figure 11. Repeating the calculations for the only two available faults now yields

$$\begin{aligned} P_d(X_1 - s/0) &\geq P(X_1 = 1) \times P(e = 1) \times P(X_3 = 1) \\ &\geq 1/2 \times 3/4 \times 1/2 = 3/16 = 12/64 \\ P_d(a - s/1) &\geq P(a = 0) \times P(X_5 = 1) \times P(d = 1) \\ &\quad \times P(X_3 = 1) \\ &\geq 1/2 \times 1/2 \times 1/2 \times 1/2 = 16/16 \\ &\quad = 16/256. \end{aligned}$$

The above results are listed in table 1. Note that the complete cutting algorithm produces better bounds for some faults than the gate blocking algorithm. On the other hand, the gate blocking algorithm produces bounds for faults such as $a - s/1$ which cannot be bounded nontrivially by the complete cutting algorithm. Both algorithms produce true lower bounds. Furthermore, for each of the four cuts studied, better bounds were produced than any correct bounds that were produced by the safety factor heuristic. The arguments of Section 2 can be used to justify the gate blocking algorithm.

4. Qualitative Versions of the Algorithm

The purpose of producing lower bounds on detection probabilities is to be able to estimate the number of patterns needed to test a circuit (see Savir and Bardell [12]). In most cases all that is needed is a nonzero bound

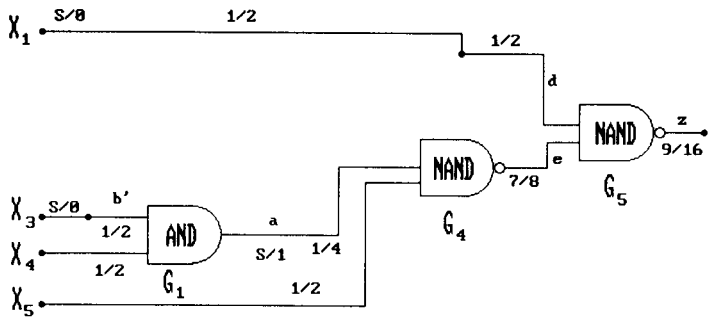

Fig. 10. The effect of blocking G_2 by setting $X_2 = 1$.

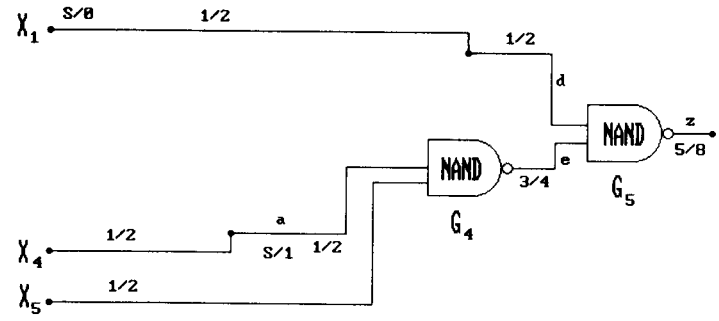

Fig. 11. The effect of blocking G_2 by setting $X_3 = 1$.

that is not too small. Since both algorithms presented here reduce a combinational circuits to fanout-free circuits it seems reasonable to apply known results from fanout-free circuits. Markowsky [8] presents a very efficient procedure that constructs a set of $n + 1$ tests that can diagnose all single faults in a fanout-free combinational circuit with n primary inputs. In particular, this means that every single fault is detectable. Actually, [8] contains further references to papers that show that any multiple fault in a fanout-free combinational circuit is detectable. This leads to the next theorem.

Theorem 2. *In a fanout-free combinational circuit with n primary inputs the detection probability of any fault is at least* $1/2^n$.

Proof. Since every fault is detectable, there must be at least one input that detects it.

Returning to figures 8–11, we see that **any** fault (single or multiple) appearing in figure 8 has a detection probability of at least 1/8 which is $1/2^2 \times 1/2$, where $1/2 = P(X_3 = 0)$. Similarly, any fault appearing in figure 9 has a detection probability of at least 1/16. For figures 10 and 11 the results are 1/32 and 1/16 respectively. While the bounds are not very accurate, they require almost no work to obtain.

Furthermore, using this approach, which could be called the **qualitative** approach since it avoids arithmetic, allows us to detect the difficult-to-test faults by seeing which lines do not appear in any of figures 8–11. In other words, from Theorem 2 it follows that any fault that appears in figures 8–11 has a detection probability bounded from below by 1/32 or better. Comparing figure 2 to figures 8–11 shows that every line except c occurs at least once in figures 8–11. Thus c is the only line that needs special attention. In fact, the fault $c - s/1$ is not detectable, so only the fault $c - s/0$ is detectable. Thus, c will never appear in any reduction of figure 2 which is fanout-free.

The gate blocking algorithm can be used to produce a test set for the faults that appear in the reduced circuit: combine the conditions needed for blocking gates with the test sets needed to test the resulting fanout-free circuit. Markowsky [8] provides a simple procedure for constructing test sets that diagnose all single faults and provides references for constructing test sets that detect all multiple faults in the resulting fanout-free circuit.

Goldberg and Lieberherr [6] provide an algorithm that produces a test vector for any specific fault for which has a nonzero lower bound on its detection probability. Because it is easy to construct the test vectors

directly for all the faults in any fanout-free circuit produced by the gate blocking algorithm, one does not need to use their algorithm to construct the test vectors.

5. Conclusion

The two algorithms and their qualitative versions presented here always produce correct lower bounds for detection probabilities of faults. The short cut can even efficiently produce test vectors for any fault for which it returns has a nonzero lower bound. This approach might be worth pursuing further as an alternative and/or a supplement to random testing.

As we saw in the analysis of the circuit of figure 2 there are faults for which zero lower bounds are returned even though the detection probability is nonzero. Because of the *#P*-complete nature of the detection probability problem one expects this to happen with any algorithm. The usefulness of the techniques described here need to be established by testing on real circuits. The author hopes that researchers with the facilities to conduct these tests will find the ideas presented here sufficiently interesting to warrant testing them. Some of the issues involved in empirical verification of results are discussed in Gaede, Mercer and Underwood [4], and Krishnamurthy and Tollis [7].

Acknowledgment

I would like to acknowledge many interesting conversations on the safety factor heuristic and other aspects of random testing with Jacob Savir and Paul Bardell. I would also like to thank the IBM Corporation for its support. I would also like to thank the referees and editors for suggestions and additional references which improved the quality of the paper.

References

1. F. Brglez, "On testability analysis of combinational circuits," *Proc. ISCAS*, pp. 221–225, 1984; (reproduced in *Test Generation of VLSI Chips*, V.D. Agrawal and S.C. Seth, IEEE CS Press, Washington, pp. 293–297, 1988).
2. S. Chakravarty and H.B. Hunt III, "On the computation of detection probability for multiple faults," *Proc. ITC*, pp. 252–262, 1986.
3. S. Chakravarty and H.B. Hunt III, "On the generalized probability problem with application to testing and reliability analysis," SUNY Buffalo, Dept. of Computer Science, Tech. Report 87-06, May 1987.
4. R.K. Gaede, M.R. Mercer, and B. Underwood, "Calculation of Greatest Lower Bounds Obtainable by the Cutting Algorithm," *Proc. 1986 Intl. Test Conf.*, pp. 498–505.
5. M.R. Garey and D.S. Johnson, *Computers and Intractability*, W.H. Freeman, San Francisco, 1979.
6. A.V. Goldberg and K.J. Leiberherr, "Efficient test generation algorithms," *Proc. 1985 Intl. Test Conf.*, pp. 508–515.
7. B. Krishnamurthy and I.G. Tollis, "Improved techniques for estimating signal probabilities," *IEEE Trans. Computers*, 38(7): 1041–1045, July 1989.
8. G. Markowsky, "Diagnosing single faults in fanout-free combinational circuits," *IEEE Trans. on Computers*, C-28(11): 863–864, November 1979.
9. G. Markowsky, "Bounding signal probabilities in combinational circuits," *IEEE Trans. on Computers*, C-36(10): 1247–1251, October 1987.
10. J. Savir, "Syndrome-testing of 'syndrome-untestable' combinational circuits," *IEEE Trans. Computers*, C-30(8): 607–608, August 1981.
11. J. Savir, G.S. Ditlow and P.H. Bardell, "Random Pattern Testability," *IEEE Trans. on Computers*, C-33(1): 79–90, January 1984.
12. J. Savir and P.H. Bardell, "On random pattern test length," *IEEE Trans. on Computers*, C-33(6): 467–474, June 1984.
13. S.C. Seth and V.D. Agrawal, "A new model for computation of probabilistic testability in combinational circuits," *Integration, the VLSI Journal*, 7: 49–75, 1989.

George Markowsky received his Ph.D. in Mathematics from Harvard University in 1973. He worked at IBM Thomas J. Watson Research Center from 1973 to 1984. Since that time he has been a member of the Computer Science Department at the University of Maine. He has written papers on a variety of topics in computer science and mathematics.