A new model for computation of probabilistic testability in combinational circuits

Sharad C. Seth

Department of Computer Science and Engineering, University of Nebraska, Lincoln, NE 68588 (U.S.A.)

Vishwani D. Agrawal

AT&T Bell Laboratories, Murray Hill, NJ 07974 (U.S.A.)

Received February 1988

Abstract. Controllabilities and detectabilities for a line in a digital circuit are defined as absolute probabilities while observabilities are defined as conditional probabilities. For exact computation of these probabilities, a divide-and-conquer technique is developed in terms of subcircuits covering the original circuit. The subcircuits, called supergates, completely enclose all reconvergent fanouts. Computation of probabilities requires conditional computations within supergates with specific logical value assignments to reconvergent fanout inputs. Combining the conditional values weighted with corresponding assignment probabilities gives the total probability. Detectability computation is direct and does not require the generation of an exclusive-or function or the insertion of auxiliary gates as needed in other methods that convert the detectability problem into a controllability problem. Techniques are suggested for limiting computational effort in circuits with very large supergates where approximate computation may be desirable.

Keywords. CAD, detection probability, random-pattern testability, supergates, testability measures.

1. Introduction

In order to determine the testability of a digital circuit, we ask two questions: How difficult is it to generate tests for this circuit? and how many test vectors will be required? Both questions are related to the effort of testing and the answers must depend on the structural complexity of the circuit. The answers will be most useful if they are obtained before test generation, for, if the effort of testing were unreasonably high, testability improvements could be incorporated. Testability analysis attempts to find quantitative answers to these seemingly qualitative problems.

Previous work

Most of the work on testability analysis has been reported during the last two decades. Stephenson and Grason [31] defined testability measures. For every

Sharad C. Seth 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, India, and 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, India and at the Bell Laboratories, Murray Hill, New Jersey.

Seth serves on the Editorial Boards of the *IEEE Design & Test of Computers* magazine and the forthcoming *Journal of Electronic Testing Theory and Applications*. He has authored or coauthored over fifty 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 won a *Best Paper Award* at the 1988 International Conference on Computer Design. Seth is a senior member of the IEEE and a member of the ACM.

Vishwani D. Agrawal received the B.E. degree from the University of Roorkee, India, the M.E. degree from the Indian Institute of Science, and the Ph.D. degree from the University of Illinois, Urbana-Champaign, in 1964, 1966, and 1971, respectively.

He is a member of technical staff in the Computing Science Research Center at the AT&T Bell Laboratories, Murray Hill, New Jersey, where, earlier as supervisor of the Testing and Design Verification Group, he was responsible for the development of CAD tools and techniques for test generation, fault simulation, testability analysis, automated scan design, built-in self-test, and timing analysis of VLSI circuits. His previous employment includes a research position at the TRW Defense & Space Systems Group in Redondo Beach, California, and a teaching position at the Indian Institute of Technology in New Delhi, India. He has published over hundred papers and has coauthored a book, *Tutorial: Test Generation for VLSI Chips* (IEEE Computer Society Press, 1988). During 1985–1987, Dr. Agarwal was the editor-in-chief of the *IEEE Design & Test of Computers* magazine. From 1984 to 1986, he served as the operations vice-chairman of the IEEE Computer Society's Design Automation Standards Subcommittee. He was a co-organizer of the First and the Second International Workshops on VLSI Design held in India in December 1985 and December 1988, respectively. He is a member of the program committees of the Design Automation Conference, the International Test Conference, and the IFIP WG10.2 Working Conference on the CAD Systems using AI. He currently serves on the Board of Governors of the IEEE Computer Society.

Dr. Agarwal is a fellow of the IEEE, a fellow of the Institution of Electronics and Telecommunication Engineers (India), and a member of the ACM. He won the *Best Applications Paper Award* from the IEEE Antennas & Propagation Society in 1979, the *Best Paper Award* at the 1982 International Test Conference, a *Best Presentation Award* at the 1985 International Conference on Computer Design, the *Best Paper Award* at the 1987 AT&T Conference on Electronic Testing, and a *Best Paper Award* at the 1988 International Conference on Computer Design. Recently, he was given an *Outstanding Contribution Award* by the Computer Society of the IEEE for his work on the Design & Test magazine.

node in the circuit, they defined two quantities, *controllability* and *observability*, as indicators of controlling and observing that node. These parameters, with values between 0 and 1, were computed for all nodes in the circuit using the functional behavior of the modules. Signal correlations due to fanouts were neglected and the testability of a node was defined as the geometric mean of its controllability and observability. Attempts at interpreting the results met only with limited success. However, this work showed that testability analysis tools are possible [12].

The first popular CAD tool for testability analysis was SCOAP, developed by Goldstein [10,11]. It provided several improvements over the previous work. First, the analysis was performed at the gate level instead of the register-transfer level. This made SCOAP results easier to relate to the faults that are commonly modeled at the gate level. Second, Goldstein recognized that the controllability of a signal may be dependent on the specific logic value. For example, it is easier to control the output of an AND gate to 0 than to 1. Separate 0- and 1-controllabilities were, therefore, defined. In all, three combinational and three sequential measures are computed by SCOAP for each node in the circuit. Numerical values of these measures range between zero and infinity and are assumed to represent the *effort* required in controlling/observing the node. Higher values denote greater effort. Third, the analysis complexity was kept almost linear in the number of nodes by neglecting signal correlation. This, however, became the stumbling block for SCOAP since signal correlations caused by reconvergent fanouts are the main reason for difficulties of test generation. Analysis of SCOAP results has shown its potential in identifying circuits that are difficult to test although the results are not always reliable [1]. Production-level CAD tools have been developed for use in VLSI design [29]. Several other algorithms that fall in the same class as SCOAP have been reported in the literature [4,7,16].

There are two main problems with SCOAP-like testability measures. First, a certain arbitrariness in the definition of controllability and observability measures makes calibration difficult. SCOAP can compare the testability of two circuits but can not tell how testable a given circuit is. Second, the assumption of independence of signals produced by the same fanout limits accuracy. These problems have led to the use of probabilistic definition of testability.

Parker and McCluskey [19–21] developed procedures for computing signal probabilities in digital circuits. In their approach, the probability of a signal assuming a value 1 is computed from a Boolean expression for that signal written in terms of primary input variables. Fault detection probability is obtained as the 1-probability of the exclusive-or function of the good and the faulty outputs. Fife [9] additionally defined a *transmission probability* for each line as the probability of sensitizing a path from that line to a primary output. He computed the detection probability as the product of transmission probability and the appropriate signal probability. The computation of these probabilities, in the presence of reconvergent fanouts, required an almost exhaustive simulation. Other authors have called this transmission probability as observability. Savir showed that the derivation of detectability by multiplying controllability and observability could produce incorrect results [22]. Jain and Agrawal, in their work on statistical fault analysis [15], defined separate 0 and 1 observabilities for each line. They also defined observabilities as conditional probabilities. For example, the 0-observability of a line is the probability of observing that line at a primary output given the value on the line is 0. Thus, 0-observability can be multiplied to the 0-controllability to obtain the joint probability of setting the line to 0 and then observing it which amounts to the detection probability of stuck-at-1 fault on the line. We will further explain these definitions in Section 2.

Parker and McCluskey's work led to several applications of probabilistic analysis [2,17,18,28]. Complexity of symbolic computation for large circuits, however, restricted its applicability. This motivated the later work on numeric and approximate methods. Savir et al. [24], formulated a *cutting algorithm* in which every fanout line is cut and initialized to a controllability range [0,1]. The modified network thus becomes free from reconvergent fanouts and all controllabilities can be obtained. The price paid for this computational tractability is that for certain signals only loose upper and lower bounds on controllability can be obtained. Brglez devised a probabilistic testability algorithm, COP [5], and showed that reasonable accuracy could be obtained in some cases by neglecting signal correlation.

Recently, accurate methods for analyzing very large circuits have received attention. Bass and Grundmann [3] showed that in a combinatorial network, when just the inputs involved in producing reconvergent fanout signals are set to 0 or 1, all other signals in the network become mutually independent. Thus exact computation can be done for cases with partially enumerated inputs. In the present work, we cover the circuit by smaller blocks and use this technique in each block. Our blocks, called *supergates*, completely include reconvergent fanouts. Based on this covering, we derive algorithms to exactly compute detection probabilities. These algorithms avoid taking exclusive-or between the good and the faulty circuits [19] or inserting auxiliary AND gates as suggested by Savir et al. [24]. Only when a circuit includes very large supergates, approximation would be necessary. We present several heuristics for approximations.

Outline of paper

To avoid confusion we state definitions of probabilistic testability in Section 2. We define controllabilities and detectabilities explicitly as probabilities. Observabilities are shown to be conditional probabilities that can be derived from controllabilities and detectabilities.

As mentioned above, we use a *divide and conquer* approach to compute these probabilities from supergates. In general, a supergate is defined as a subcircuit with possibly many inputs but just one output. In Section 3 we describe a minimum cover of the circuit in terms of supergates.

Once the circuit is covered by supergates, signal controllabilities can be computed in several ways:

(1) The Parker and McCluskey method [19] is applied to signals within a supergate while the controllabilities between the supergates are carried in the numerical form.

(2) The Parker and McCluskey method is applied to supergates with only the inputs producing reconvergent fanouts kept in the symbolic form while other inputs are given their numerical controllability values.

(3) A numeric computation is carried out for cases generated by assigning logic values to the fanout signals at the supergate input. Exact computation of controllabilities, conditional to input assignments, is straightforward [3]: when cases are appropriately combined, total controllabilities are obtained.

We adopt the third approach in Sections 4 and 5. In Section 5, we give methods for computing exact detectabilities. Detectability computation requires separate procedures for fanout stems and non-fanout signals.

The time complexity of the exact methods depends on the size of the largest supergate in the cover. In circuits where reconvergent structures include large portions of the circuit, approximations may be necessary. In the approximations discussed in Section 6, short reconvergent fanouts are treated exactly while the correlation between signals that reconverge after a large number of levels is neglected. Such approximations allow a smooth tradeoff between computational complexity and accuracy.

2. Probabilistic testability measures

Definition 2.1. The *1-controllability* (*0-controllability*) of a line is the probability of a 1 (0) on that line when a random input is applied to the circuit.

We will denote the 1- and 0-controllabilities of a line k as $C1(k)$ and $C0(k)$, respectively. In general, random inputs means that the input vectors can occur with any assigned probabilities. However, when all input vectors are assigned equal probability then 1-controllability of a line is the fraction of 1's in the truth table of the line function. It is also called the line *syndrome* [23].

Definition 2.2. The *1-detectability* (*0-detectability*) of a line is the probability of detecting the stuck-at-1 (stuck-at-0) fault on that line when a random input is applied to the circuit.

We will denote the 1- and 0-detectabilities of line k as $D1(k)$ and $D0(k)$, respectively. For equiprobable inputs, these are the fraction of 0's and 1's in the truth table of the function realized on that line from which a path is sensitized to a primary output. Detectabilities are compound measures that can be expressed in terms of controllabilities and observabilities as follows:

$$D1(k) = C0(k)B0(k) \tag{1}$$

$$D0(k) = C1(k)B1(k) \tag{2}$$

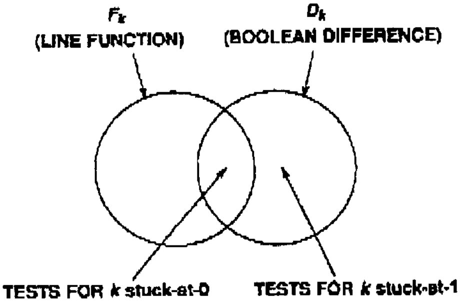


Fig. 1. Venn diagram showing relationship between Boolean difference and detectabilities.

where $B0(k)$ and $B1(k)$ are the probabilities of observing the line k given that it was set to 0 or 1, respectively. These conditional probability definitions of a line's observability differ from the traditional definitions [24] based on the Boolean difference [25]. However, for equiprobable inputs the two are closely related, as we will show using the Venn diagram in Fig. 1. The disk F_k includes the minterms of the function of the primary inputs realized on line k and the disk D_k includes the minterms of the Boolean difference of the primary output with respect to line k. We will use $|F_k|$ and $|D_k|$ to denote, respectively, the number of minterms in F_k and D_k. The *traditional* definition of line observability is simply the syndrome of D_k, i.e., $|D_k|/2^n$, where n is the number of primary inputs. It represents the fraction of input vectors for which the state of line k, whether *zero or one*, can be sensitized to the primary output. On the other hand, $D0(k)$, the probability of detecting a stuck-at-zero fault on line k, is the fraction of input vectors that set line k to one *and* sensitize it to the primary output. Thus

$$D0(k) = \frac{|F_k \cap D_k|}{2^n} = \frac{|F_k|}{2^n} \cdot \frac{|F_k \cap D_k|}{|F_k|}$$

$$= C1(k) \cdot B1(k) \text{(by definition, see Eqn. 2)}.$$

Therefore,

$$B1(k) \equiv \frac{|F_k \cap D_k|}{|F_k|}.$$

Similarly, it can be shown that

$$B0(k) = \frac{|\bar{F}_k \cap D_k|}{|\bar{F}_k|}$$

In comparison, the traditional observability, from the Venn diagram of Fig. 1, is the sum of zero and one detectabilities, $D0(k) + D1(k)$. With this definition, it is possible that for a line k, $B(k)$, $C1(k)$ and $C0(k)$ are all relatively high and yet its zero (or one) detectability is very low because of a very small overlap between the disks D_k and F_k (or $\bar{F}_k$). This phenomenon, explained naturally by our conditional definition of observabilities, appears anomalous [22] for the traditional definition.

3. Supergates and maximal supergates

The difficulty of problems in testing of combinatorial logic can be traced largely to a single culprit: the correlation of signals on reconvergent lines due to common fanout stems. For example, random-pattern testability analysis would be a rather simple single-pass process if the circuit had no reconvergent stems [23]. On the other hand, if correlated signals do reconverge at a gate, it is not always necessary to trace the whole subcircuit on which the gate output depends, in order to arrive at line signals which are mutually uncorrelated. As an example, in Fig. 2(a), the two inputs to gate 8 carry correlated signals from the fanout stem s, but we need to go back from gate 8 only upto the output of gate 5 before the line signals on input 2 and the output of gate 5 become topologically independent. This simple observation forms the basis for the definition of a supergate below. First we need a graph representation of the circuit and some basic definitions from graph theory [13].

For a combinatorial logic network N, we will consider an equivalent representation in the form of a directed graph $G(V, E)$, called the *circuit graph* whose nodes are: the primary inputs, fanout points, the primary outputs, and the gates in N; and whose edges represent the connections in N, oriented in the direction

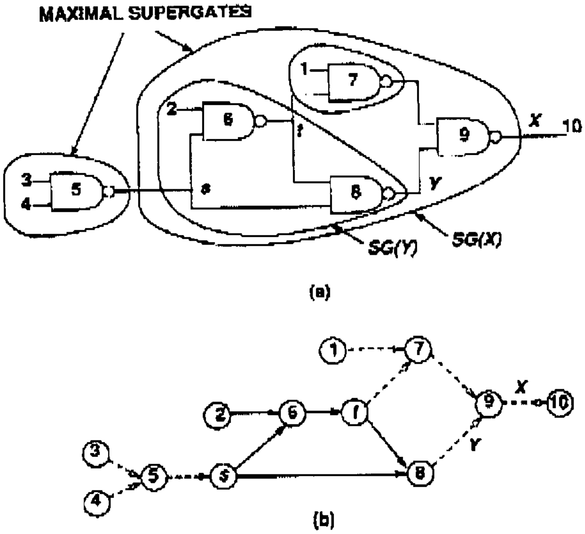


Fig. 2. An example of a circuit with recovergent fanouts and some of its supergates: (a) circuit and its supergates, (b) graph representation.

signal flow. As an example, the circuit graph of the network in Fig. 2(a) is shown in Fig. 2(b). Notice that each line (including the fanout branches) in the circuit is represented uniquely by an edge (drawn by a solid or a dashed line) in the graph. We say, that node i is an *immediate predecessor* of node j if (i, j) is an edge in the circuit graph; node i is a *predecessor* of node j if there is a directed path (of length $\geqslant 1$) from i to j. Two nodes are said to be *topologically independent* if they do not have a common predecessor.

Let V' be a subset of the node set V. Then the *induced subgraph* $G'(V', E')$ of $G(V, E)$ is the graph whose vertex set V' is a subset of V and whose edge set E' includes all the edges of E restricted to the nodes in V'. Thus, an induced subgraph is completely determined by its vertex set. Let $X = (i, j)$ be an edge in the circuit graph and let $R(X)$ be the set which includes i and all its predecessors. We will denote $R(X)$ as the *reaching set* of X. Then the *cone of* X, denoted as $C(X)$, is the induced subgraph (of the circuit graph) whose vertex set is the reaching set $R(X)$. The subcircuit corresponding to the cone $C(X)$ completely determines the signal value on line X. Intuitively, for the supergate of line X, we are looking for the smallest subcircuit of the cone of X whose inputs are topologically independent. The same idea is captured precisely in graph theoretic terms by the following definition:

Let $X = (i, j)$ be an edge in the circuit graph $G(V, E)$. The *supergate of* X, denoted by $SG(X)$, is the induced subgraph of the cone of $C(X)$ whose vertex set V' satisfies the following conditions:

(a) i and its immediate predecessors, if any, are in V'.

(b) Let v be a node such that an immediate predecessor of v is also in V'. Then, all immediate predecessors of v are in $SG(X)$.

(c) For every pair of nodes v and w in V' neither of which have any predecessors in $SG(X)$, the sets of all predecessors of v and w in the circuit graph are mutually disjoint.

(d) V' is a minimum vertex set satisfying the above properties.

From the definition it is obvious that the supergate of every edge X is unique. The edge (or line) X is called the *output* of the supergate $SG(X)$.

Example 1. For the circuit in Fig. 2(a) assume that we want to find the supergate of the output line of gate 8. In the circuit graph (Fig. 2(b)), this line is represented by the edge $Y = (8, 9)$. Thus the supergate is a subgraph of the cone $C(Y)$. By condition (a) of the definition, nodes 8, s, and t must be included in the supergate. However, if $SG(Y)$ contained just these three nodes, neither node s nor t will contain a predecessor in $SG(Y)$ (since the induced graph will contain just two edges: $(t, 8)$ and $(s, 8)$) but will have common predecessors (nodes 3, 4, and 5) in the original circuit graph. This will violate condition (c). It can be verified that, as a minimum, node 6 must be included to satisfy condition (c). However, now by condition (b) we must also include node 2 since one of the immediate predecessor of node 6, namely node s, is already included in the supergate. At this point it is seen that the first three conditions are satisfied. Also, from the argument it is obvious that no subset of the vertices chosen thus far will satisfy the first three conditions of the definition. Thus, the supergate of line Y is the induced subgraph with the vertex set $\{8, s, t, 6, 2\}$. This is shown by solid lines in Fig. 2(b) and the corresponding subcircuit is shown as $SG(Y)$ in Fig. 2(a). We note that the superset $\{8, s, t, 6, 2, 5, 3, 4\}$ will also satisfy the first three condition but it is not the smallest such set and hence violates condition (**d**).

The procedure described in the example can be formalized in the form of an algorithm which assumes that, in a preprocessing step, each line in the circuit has been labelled with the set of primary inputs in its cone of influence [27].

From hereon, because of the simple one-to-one correspondence between the lines in the circuit and the edges in the circuit graph, we will interchangeably talk about the supergate as either a subgraph or the corresponding subcircuit.

Lemma 3.1. If Y is an internal edge in $SG(X)$ (that is, Y is neither X nor an edge connected to an input node of $SG(X)$) then $SG(Y)$ is properly contained in $SG(X)$.

Proof. Let the edge $Y-(i, j)$, where, i and j are nodes in $SG(X)$ and i is not an input node. Since the cone of Y in $SG(X)$ is a subgraph of $SG(X)$ it is easy to see that conditions (a), (**b**), and (c) in the supergate definition must be satisfied for the nodes in the cone. Thus, the cone must be at least as large as the supergate of Y. This completes the proof. □

Consider the set of supergates for all the lines in a circuit. Remove from this collection every supergate which is properly contained in another supergate. The remaining supergates will be called *maximal supergates* of the circuit. A collection of supergates is said to *cover* a circuit if each line in the circuit is either the output of a supergate or included in a supergate. A cover is said to be *minimal* if no subset of it is also a cover. It has been shown [26] that

(1) The minimum cover of single-output circuit is unique and consists of only (and all) the maximal supergates. This minimum cover is actually a partition of the circuit, that is, the supergates in the cover are vertex-disjoint (see Fig. 2(a)).

(2) The minimum cover of a multiple-output circuit is also unique and includes only maximal supergates w.r.t. individual primary outputs. However, a maximal supergate w.r.t. some primary output may be excluded from the cover if it is properly contained within another maximal supergate. Further, the maximal supergates in the unique cover need not be mutually vertex-disjoint. These differences from the single-output case are illustrated by the full adder circuit shown in Fig. 3. For the SUM output there are two maximal supergates, denoted as $SG(SUM)$ and $SG(HALF\_SUM)$ in the figure; the carry output has only one maximal supergate, $SG(CARRY)$. The unique minimum cover of the circuit is defined by the supergates $SG(SUM)$ and $SG(CARRY)$ since $SG(HALF\_SUM)$ is included in $SG(CARRY)$. Also, the two supergates in the cover are not vertex-disjoint since they share the NAND gate marked with an asterisk in the figure.

(3) The unique minimum cover of a multiple-output circuit can be found by an algorithm with the time complexity $O(g^2)$, where g is the number of gates in the circuit.

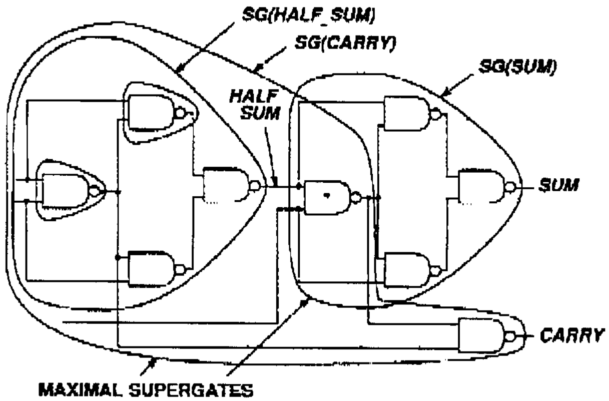


Fig. 3. A multi-output circuit with supergates shown for some lines. Maximal supergates (which may overlap) form an unique cover.

The input nodes of the supergate $SG(X)$ (that is, the nodes with zero indegree in $SG(X)$) will be denoted as $I(X)$. These will be partitioned into two classes: (a) *fanout inputs IF(X)* which have at least two directed paths in $SG(X)$ to node i, where $X=(i, j)$ (thus, at least one of the edges in each path must correspond to a fanout stem); (b) all others which will be called *non-fanout inputs* and denoted as $INF(X)$. As an example, for the supergate $SG(X)$ in Fig. 2 node s is a fanout input (being a fanout stem in $SG(X)$); node 2 is also a fanout input because stem t provides two paths to node 9. Node 1 is the only non-fanout input of $SG(X)$. This partitioning of supergate inputs will be shown to play a crucial role in determining the complexity of our computations.

4. Controllability computation

It can be easily verified [24] that for any node whose cone contains no reconvergent fanout, the controllability calculation is a straight-forward one-pass process involving only the lines in the cone. Thus we will only consider calculation of controllability for the output line X of a gate at which one or more fanouts reconverge. Let $IF(X)=\{y_1, y_2, \ldots, y_n\}$, where n is greater than 0. We will make the inductive assumption that the controllabilities of all the inputs of $SG(X)$ are known.

One way to determine supergate output controllability is to derive a symbolic expression for the output in terms of the input probabilities of that supergate [20]. Such symbolic expressions tend to be bulky even for supergates of moderate size. We, therefore, describe a numerical procedure.

Consider an assignment of binary values to the fanout inputs represented by the vector $A_i=(a_1, a_2, \ldots, a_n)$. For each such assignment we make a single pass through $SG(X)$ from inputs to outputs. First, the controllabilities of the non-fanout input lines are known by the inductive assumption; for a fanout line y_j we assume its a_j-controllability to be 1. Next, for the gates connected only to the maximal supergate primary inputs we compute output controllabilities using independence of signals at the node inputs. Formulas for this computation are straight-forward and are available in the literature [24]. The controllabilities of other nodes are similarly determined in a breadth-first fashion assuming signal independence. Eventually, we will compute the 1-controllability of X. This is the conditional controllability of X when the assignment A_i is made to the fanout inputs. We will denote this by $C1_i(X)$.

Lemma 4.1. The one-controllability of a line X is

$$C1(X) = \sum_{\text{all } A_i} C1_i(X)\ \text{Prob}(A_i)$$

where,

$$\text{Prob}(A_i) = \text{Prob}(y_1 = a_1)\ \text{Prob}(y_2 = a_2) \ldots \text{Prob}(y_n = a_n)$$

is the product of the known a_i-controllabilities of the fanout inputs, assuming independence among y's that follows from the definition of $SG(X)$.

The lemma can be proved by considering the Shannon expansion of the line function about the fanout inputs. Each term in the expansion is non-zero for precisely one assignment of binary values to the fanout inputs. The probability of a specific assignment A_i is Prob(A_i). Given the assignment, $C1_i(k)$ is the one controllability of line k, hence Prob$(A_i) \cdot C1_i(k)$ is the contribution to the one controllability of line k by all input patterns with A_i assigned to fanout inputs. The contributions due to different A_i's can simply be added because they come from mutually disjoint sets of input vectors.

Lemma 4.2. Let Y be an internal line in the supergate $SG(X)$. Assume $C1(X)$ is computed according to Lemma 4.1. Let $C1_i(Y)$ be defined as the conditional one controllability of Y when the assignment A_i is made to the fanout inputs. Then

$$C1(Y) = \sum_{\text{all } A_i} C1_i(Y)\ \text{Prob}(A_i)$$

The significance of this lemma is that with very little overhead it is possible to obtain the line controllabilities of internal lines of a supergate at the same time that its output controllabilities are being computed. An example illustrating the use of these lemmas appears later.

Computational procedure

The computational procedure has two major steps: (1) finding the minimum cover of the circuit in terms of its maximal supergates, and (2) finding the line

Table 1
Supergate data (4-bit ALU)

Supergate number	Supergate size	Number of fanout inputs
1	68 nodes	14
2	51 nodes	13
3	26 nodes	10

controllabilities of the maximal supergate using Lemmas 4.1 and 4.2. For the second step, the maximal supergates are processed in a breadth-first fashion starting from the primary inputs (this ensures that the line controllabilities of all

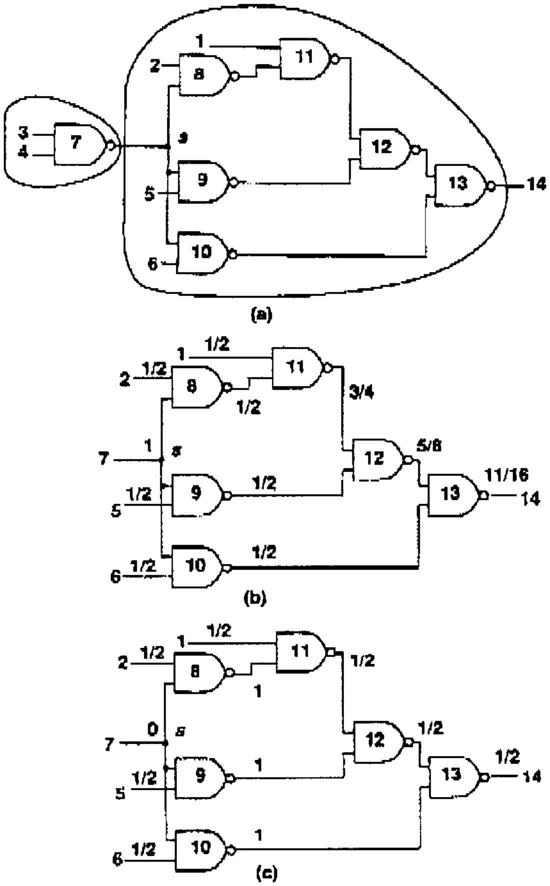


Fig. 4. Computation of conditional controllabilities: (a) example circuit with minimum cover by maximal supergates, (b) conditional 1-controllabilities in *SG* (13) when line 7 is assigned 1, (c) conditional 1-controllabilities in *SG* (13) when line 7 is assigned 0.

Table 2
Controllability calculation

	L					
	8	9	10	11	12	13
$C1(L)_{\text{stem } s=0}$	1	1	1	1/2	1/2	1/2
$C1(L)_{\text{stem } s=1}$	1/2	1/2	1/2	3/4	5/8	11/16
$C1(L)$	5/8	5/8	5/8	11/16	19/32	41/64

the inputs of supergate are known before its output controllability is determined as required by Lemma 4.1).

As already mentioned, the first step can be carried out in at most $O(g^2)$ steps, where g is the number of gates in the circuit [26]. The second step requires as many sweeps of the supergate as the number of patterns applicable to its fanout inputs which is exponential in the number of fanout inputs. Each sweep itself requires time proportional to the number of gates in the supergate. Thus large supergates with many fanout inputs would be very expensive for controllability computation. As an extreme example, Table 1 summarizes data from running the supergate procedure on the 4-bit ALU (74181) [8]. Approximations to the above procedure are suggested in Section 6 for circuits with a rich reconvergent fanout structure.

Example 2. Figure 4(a) shows a logic circuit used as an example in the literature [24]. Also shown in the figure is the minimum cover of the circuit consisting of two maximal supergates. Input probabilities are assumed to be 1/2 corresponding to equiprobable inputs. The computation complexity, in general, is independent of this assumption; we could have just as well assumed any arbitrary input probabilities. The supergate $SG(7)$ will be processed before $SG(13)$. Since $SG(7)$ has no fanout inputs, $C1(7)$, which is the same as $C1(s)$, is determined to be 3/4 directly from the gate input probabilities.

At this point we know all the input probabilities of $SG(13)$ which has only stem s as its fanout input. According to Lemma 4.1 we set stem s in turn to 0 and 1, in each case setting other input probabilities to their known values (1/2 since they are all primary inputs.) The computation for the two cases is shown respectively in Figs. 4(b) and 4(c). Thus according to Lemma 4.1

$$C1(13) = (3/4)(11/16) + (1/4)(1/2) = 41/64.$$

The one-controllability of other lines in $SG(13)$ can be determined by Lemma 4.2. The complete computation is summarized in Table 2.

5. Detectability computation

Once line controllabilities are known it is sufficient to consider computation of either line detectabilities or line observabilities since these two measures are easily related to each other by eqns. (1) and (2).

All the proposed methods of computing line observabilities (or detectabilities) from the circuit structure involve approximations and thus lead to errors [6,15]. The errors can be positive or negative and can occur for reconvergent fanout stems as well as non-stem lines [14]. They can be attributed to two distinct classes of simplifications: (1) the observability of a line through a chain of gates can be found by considering the gates independently, and (2) the observability of a reconvergent fanout stem is a fixed function of its branch observabilities. In the following section we reproduce STAFAN's rules and point out by means of a simple example how errors are introduced in the computed values of observabilities/detectabilities for nonstem lines. These errors come about due to the first simplification noted above. We show how they can be avoided for an exact computation of non-stem detectabilities. In the next section, a similar plan is followed for stem-line computations.

Computation for non-stem lines

Consider STAFAN's approximations [15] for a two-input AND gate with input lines a and b and output line c. Observing a zero value on line a requires that line b should be set to 1 and the 0 on line c should be observable. That is,

$$\begin{aligned} B0(a) &= B0(c)\ \text{Prob}(b=1 \mid a=0) \\ &= B0(c)[\text{Prob}(b=1,\ a=0)/\text{Prob}(a=0)] \\ &= B0(c)[C1(b)-C1(c)]/C0(a). \end{aligned} \tag{3}$$

Thus, by eqn. (1)

$$\begin{aligned} D1(a) &= C0(a)B0(a) \\ &= B0(c)[C1(b)-C1(c)] \\ &= D1(c)\left[\frac{C1(b)-C1(c)}{C0(c)}\right]. \end{aligned} \tag{4}$$

Similarly,

$$B1(a) = B1(c)\ \text{Prob}(b=1 \mid a=1) = B1(c)C1(c)/C1(a). \tag{5}$$

Hence by eqns. (2) and (5)

$$D0(a) = C1(a)B1(a) = B1(c)C1(c) = D0(c). \tag{6}$$

These and similar derivations for other basic gate types are summarized in Table 3. The expressions for the primary-output detectabilities follow directly from the fact that both the zero and one observabilities of the primary outputs are one by definition. Unfortunately, the expressions given in Table 3 do not always yield correct detectabilities for non-stem lines. The source of the error is to be found in eqns. (3) and (5) from which the detectabilities were derived. In eqn. (3) $B0(a)$ is obtained by multiplying two probabilities, $B0(c)$ and $\text{Prob}(b=1 \mid a=0)$. This assumes that these two probabilities correspond to independent events, not always a valid assumption. Consider the circuit shown in Fig. 5(a) and

Table 3
STAFAN rules for detectability computation

	AND $c = a \cdot b$	OR $c = a + b$	NOT $c = \bar{a}$	Primary output a
$D0(a)$	$D0(c)$	$D0(c)\left[\dfrac{C0(b)-C0(c)}{C1(c)}\right]$	$D1(c)$	$C1(a)$
$D1(a)$	$D1(c)\left[\dfrac{C1(b)-C1(c)}{C0(c)}\right]$	$D1(c)$	$D0(c)$	$C0(a)$

assume equiprobable inputs. The exact controllabilities of the lines in the circuit can be found by the method described in Section 4. These are shown in the first two rows of Table 4. The last two rows of the table show line detectabilities computed according to the rules of Table 3.

Line b, a reconvergent stem, will be considered in the next section. Line a is observable at g whenever b is 1. Therefore, $B0(a) = B1(a) = 1/2$ and therefore $D1(a) = D0(a) = 1/4$ by eqns. (1) and (2). However, our computation, in the table, incorrectly gives $D1(a) = \frac{1}{12}$. Indeed, since the circuit is so simple, the entries in Table 4 can be readily checked by exhaustive means. It will be found that the only other error occurs in $D1(c)$ which should be 0 (i.e. the fault c stuck-at-1 is not detectable).

In analyzing the cause of error in computing $D1(a)$, we note that eqn. (3) assumes independence of $B0(e)$ and Prob($b = 1 \mid a = 0$), but the observability of line e depends on the signal value on line f which is the complement of the signal value on line b. The STAFAN rules correctly take account of signal correlations

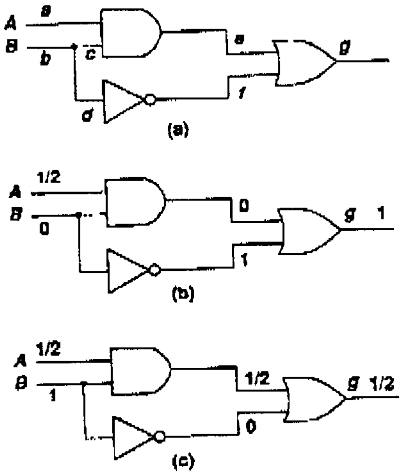


Fig. 5. Exact detectability computation for non-recovergent lines: (a) an example circuit, (b) 1-controllabilities when $B = 0$, (c) 1-controllabilities when $B = 1$.

Table 4
Non-stem detectabilities for circuit in Fig. 5(a) using STAFAN formulas

	a	b	c	d	e	f	g
$C0$	1/2	1/2	1/2	1/2	3/4	1/2	1/4
$C1$	1/2	1/2	1/2	1/2	1/4	1/2	3/4
$D0$	1/4		1/4	1/4	1/4	1/2	3/4
$D1$	1/12 [a]		1/12 [a]	1/2	1/4	1/4	1/4

[a] denotes erroneous values.

between the lines at the input of the same gate, e.g., signals on lines e and f are correlated due to the common influence of b; their detectabilities are correctly computed. However, correlations extending beyond the inputs of a single gate, e.g., while computing $D1(a)$ or $D1(c)$, are ignored by STAFAN.

In summary, what prevents the use of simple rules for detectability computation in a single backward sweep of the circuit is no different from what prevents the use of simple forward-chaining rules for controllability computation; the culprit in both cases is signal correlation in the circuit. In Section 4, we presented an exact technique for controllability computation which avoided signal correlations by assigning fixed binary values to fanout inputs of a supergate. We will use the same idea in proposing an exact method for detectability computation for non-stems.

Consider a two-input AND gate, with inputs a and b, and output c, embedded in a supergate. An input pattern to the supergate is obtained by a combination of deterministic and probabilistic procedure: a binary pattern A_i is applied to fanout inputs and each non-fanout line is set to one with a probability equal to its one-controllability. For an arbitrary line k in the supergate we define its *conditional 1-detectability (conditional 0-detectability)* as the probability of detecting the fault "k stuck-at-1" ("k stuck-at-0") at the output of the supergate when such an input pattern is applied to the supergate inputs. Notationally, we will use $D1_i(k)$ and $D0_i(k)$ to represent these conditional detectabilities. Further, conditional observabilities, from eqns. (1) and (2), are given by:

$$B0_i(k) = \frac{D1_i(k)}{C0_i(k)},$$

$$B1_i(k) = \frac{D0_i(k)}{C1_i(k)}.$$

Once the fanout inputs of the supergate are fixed, the two input lines of the AND gate depend only on mutually disjoint sets of (non-fanout) supergate inputs. Thus the signals on the two inputs to the AND gate are independent and the STAFAN's detectability rules given in Table 3 will yield correct answers for *conditional* computation. Thus, for the AND gate,

$$D0_i(a) = D0_i(c)$$

$$D1_i(a) = D1_i(c)\frac{C1_i(b) - C1_i(c)}{C0_i(c)}$$

Table 5
Exact detectabilities for the circuit in fig. 5(a)

	a	b	c	d	e	f	g
$C1_{b=0}$	1/2	0	0	0	0	1	1
$C1_{b=1}$	1/2	1	1	1	1/2	0	1/2
$D1_{b=0}$	0		0	1	0	0	0
$D1_{b=1}$	1/2		0	0	1/2	1/2	1/2
$D0_{b=0}$	0		0	0	0	1	1
$D0_{b=1}$	1/2		1/2	1/2	1/2	0	1/2
$D0$	1/4		1/4	1/4	1/4	1/2	3/4
$D1$	1/4		0	1/2	1/4	1/4	1/4

Similarly, the rules for OR, NOT, and primary output given in Table 3 can be easily adapted for conditional computations.

Theorem 5.1. Let k be a non-reconvergent line in a supergate and let $D0(k)$ and $D1(k)$ represent its zero and one detectabilities, respectively, at the supergate output. Then

$$D0(k) = \sum_{\text{all } A_i} D0_i(k) P(A_i)$$

$$D1(k) = \sum_{\text{all } A_i} D1_i(k) P(A_i)$$

where, $P(A_i)$ is the probability of applying pattern A_i to the fanout inputs of the supergate. This theorem can be proved using the Shannon expansion in the same way as Lemma 4.1.

Returning to our example of Fig. 5, the conditional detectabilities can be computed from Figs. 5(b) and 5(c). Assuming line g to be a primary output the results are tabulated in Table 5.

From Theorem 5.1, for example, $D0(a) = D1(a) = (1/2 + 0)(1/2) = 1/4$, which is verified by noting that only one of the four input patterns detects each of the faults: "a stuck-at-0" and "a stuck-at-1". Note that $B0(a) = D1(a)/C0(a) = 1/2$. Thus non-stem line observabilities can also be correctly computed.

Computation for reconvergent stems

It is attractive to think of extending the above method to determine the detectabilities of reconvergent stems in a backward trace of the circuit. To carry out this plan we should be able to define the detectabilities of a stem as a function of its branch detectabilities. Unfortunately, it is not possible to ignore the detailed circuit topology and express even good bounds on a stem's detectability in terms of its branch detectabilities alone. For example, it can be shown that the stem detectability may exceed the maximum of its branch detectabilities or that it may be strictly less than either the minimum or the union of the branch detectabilities [14]. The solution proposed in this paper is to calculate the stem detectabilities in a forward trace of the maximal supergates, overlapping the computation with that for line controllabilities. The procedure may be thought of as a probabilistic analog of single-fault simulation.

Let s be a reconvergent fanout stem in a supergate and let a binary pattern A_r be applied to the fanout inputs of the supergate. From the definition of fanout inputs in Section 3, it follows that the value on stem s is uniquely determined by A_r. For an arbitrary line k in the supergate, we consider the probabilities of disjoint events of sensitizing k to 0 and 1 values from s. These will be called, respectively, *conditional zero-sensitization* and *conditional one-sensitization* of k from s, or symbolically, $S_i(s, k_0)$ and $S_i(s, k_1)$. If k has only even-parity or only odd-parity inversion paths from s, one of these probabilities will be zero but, in general, both probabilities must be considered.

Theorem 5.2. Let $S_i(s, k_b)$ denote the conditional b-sensitization of a line k in the supergate from the stem s, where b is either 0 or 1.

(a) For a two-input AND gate with inputs m and n and output p:

$$\begin{aligned} S_i(s, p_b) = {} & S_i(s, m_b)[C1_i(n) - S_i(s, n_1) \\ & + S_i(s, n_b)[C1_i(m) - S_i(s, m_1)] \\ & + S_i(s, m_b)S_i(s, n_b) \end{aligned}$$

(b) For a two-input OR gate with inputs m and n and output p:

$$\begin{aligned} S_i(s, p_b) = {} & S_i(s, m_b)[C0_i(n) - S_i(s, n_0)] \\ & + S_i(s, n_b)[C0_i(m) - S_i(s, m_0)] \\ & + S_i(s, m_b)S_i(s, n_b) \end{aligned}$$

(c) For a NOT gate with input m and output p:

$$S_i(s, p_b) = S_i(s, m_{\bar{b}})$$

A proof of the theorem appears in the Appendix. Here we provide an intuitive justification. Part (c) is trivial while (b) is the dual of (a) hence it is enough to consider only (a). How can a stem value be sensitized to the output of an AND gate from its inputs? We can distinguish three mutually disjoint cases, each representing one term of the expression in Part (a): the stem value is sensitized through input m alone, through input n alone, or simultaneously through m and n. In the first case, input n should be a non-sensitized 1, an event with the probability $C1_i(n) - S_i(s, n_1)$. It is clear that this event is independent of sensitizing the stem value to the input m (since the circuit is acyclic), hence the two probabilities can be multiplied to accurately define the probability for the first case. The second case is symmetrical. Finally, it is easy to see that if the sensitized values on inputs m and n are opposite, simultaneous sensitization to the output is not possible. On the other hand, if the sensitized input values are

Table 6
Computation of stem sensitization

		x						
		a	b	c	d	e	f	g
$b=1$	$C1_i(x)$	1/2	1	1	1	1/2	0	1/2
	$C0_i(x)$	1/2	0	0	0	1/2	1	1/2
	$S_i(b, x_0)$	–	0	0	0	0	1	1/2
	$S_i(b, x_1)$	–	1	1	1	1/2	0	0
$b=0$	$C1_i(x)$	1/2	0	0	0	0	1	1
	$C0_i(x)$	1/2	1	1	1	1	0	0
	$S_i(b, x_0)$	–	1	1	1	1/2	0	0
	$S_i(b, x_1)$	–	0	0	0	0	1	1/2

alike then the output will be sensitized with the same value. This explains the last term in the expression.

Next, suppose the output line of a supergate is x. For a stem s, it is possible to determine its conditional sensitizations to line x by Theorem 5.2. The stem's observabilities at x are obtained by appropriately combining these conditional sensitizations as indicated in the following theorem:

Theorem 5.3. Let s be a reconvergent stem in a supergate and let Z_s (N_s) be the subset of patterns on the fanout inputs of the supergate which cause a zero (one) to appear on s. Then

(a) $D0(s) = \sum_{A_i \in Z_s} [S_i(s, x_0) + S_i(s, x_1)] P(A_i)$

(b) $D1(s) = \sum_{A_i \in N_s} [S_i(s, x_0) + S_i(s, x_1)] P(A_i)$

A proof of theorem can again be given by considering the Shannon expansion of the supergate output about the fanout inputs. The argument is similar to that used in the proof of Lemma 4.1.

Example 3. We will apply Theorems 5.2 and 5.3 to compute the detectabilities of the stem b in Fig. 5(a). The steps of the computation are summarized in Table 6. The two halves of the table represent conditional computation as indicated by Theorem 5.2, for the two cases $b = 1$ and $b = 0$, respectively; these will be used to compute $D0(b)$ and $D1(b)$ at the primary output g according to Theorem 5.3. Thus, from the top half of Table 6,

$$D0(b) = [S_i(b, g_0) + S_i(b, g_1)] \text{ Prob}(b=1)$$
$$= [1/2 + 0]1/2 = 1/4$$

Similarly, from the lower half of Table 6,

$$D1(b) = [S_i(b, g_0) + S_i(b, g_1)] \text{ Prob}(b=0)$$
$$= [0 + 1/2]1/2 = 1/4$$

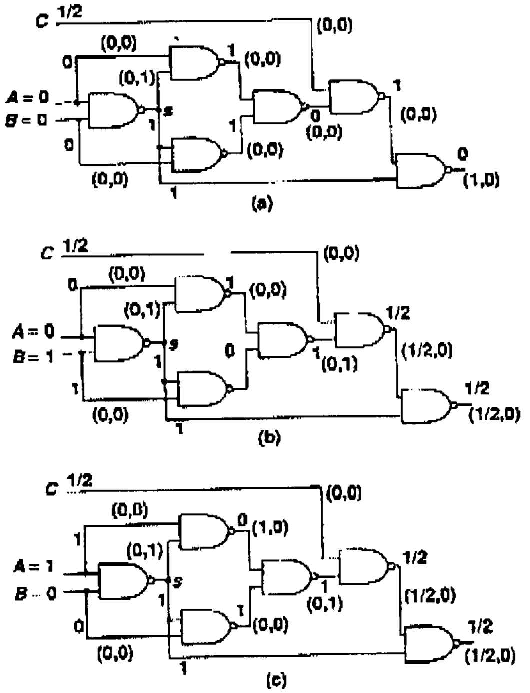


Fig. 6. Exact 1-detactability computation for recovergent stem s. The three cases, (a), (b) and (c), correspond to the possible input assignments producing 1 at s. The numbers inside parentheses are 0 and 1 sensitizations from s. The numbers outside parentheses are 1-controllabilities.

Example 4. Consider the carry-logic circuit of Fig. 6(a). It is easily verified that the supergate of the carry output is the whole circuit with A and B as the fanout inputs. For each line k in the network, we show the sensitizations $S_i(s, k_0)$ and $S_i(s, k_1)$ as an ordered pair within parentheses. Also shown for line k is $C1_i(k)$. Stem s is 1 for three combinations of values on A and B: 00, 01, and 10. Figures 6(a)–6(c) show the conditional sensitizations (obtained from Theorem 5.2) and 1-controllabilities for the three cases. Each of these combinations occurs with probability, $P(A_i) = 1/4$, therefore

$$D1(s) = [(1+0) + (1/2+0) + (1/2+0)](1/4) = 1/2.$$

6. Approximate computation

In the worst case, the supergate for a node may include all the primary inputs as fanout inputs, e.g., the first supergate for the 4-bit ALU mentioned in Section 4. In such cases, the exact controllability procedure outlined above becomes equivalent to an exhaustive true-value simulation of the supergate, that is, the computational time is exponential in the number of primary inputs and linear in the size of the supergate. In this section we illustrate two ways of reducing the time complexity to manageable proportions while sacrificing some accuracy in the computed results. The first is based on a heuristic and the second involves sampling. For simplicity, only controllability computations are illustrated.

Heuristic approximation

For two nodes in a circuit graph we define the *distance* as the number of edges in the shortest path between the nodes; single input nodes (inverters or buffers) may be ignored in computing the distance. Let T be an integer used as a parameter of the heuristic. We restrict the supergate computation to the subgraph in which no node is farther than a distance T from the supergate output.

Intuitively, we will be considering the effect on the supergate output of all reconvergent fanouts within a distance *threshold* T from it but ignore others which are more "global" in scope. At one extreme, when the distance threshold is unity, each node will be assumed to carry independent signal, as in Goldstein's analysis [10]. At the other extreme, when the threshold value exceeds the maximum supergate depth, all calculations will be exact.

Example 5. Consider again the circuit shown in Fig. 4(a). Its circuit graph is shown in Fig. 7. Figure 8(a) shows the supergates $SG(12)$ and $SG(13)$ determined for this circuit when the threshold value is either 1 or 2. For either threshold value the reconvergence of the stem s at 12 or 13 is ignored so the supergate in each case includes just the node and all its inputs. The resulting 1-controllability of each node is as shown. However, when the threshold is increased to 3 (Fig. 8(b)) the reconvergence is picked up for node 12 but only partially for node 13. The controllability values for node 12 will therefore be computed exactly and for node 13 approximately. The values in brackets in Fig. 8(b) are conditional controllabilities. Finally, for the threshold value of 4 all computations become exact. Figure 8(c) summarizes these results for nodes 12 and 13 and shows the error in controllability values in each case.

It is worth noting that while more nodes assume their exact controllability values as the threshold is increased, the convergence to exact value for an individual node may not be monotonic. This is evident in the increased error magnitude for $C1(13)$ in Fig. 8(c).

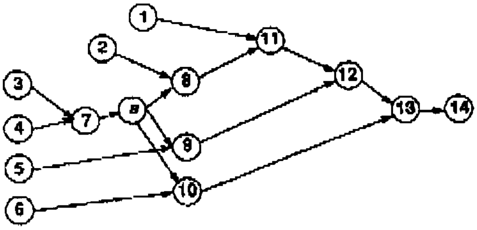
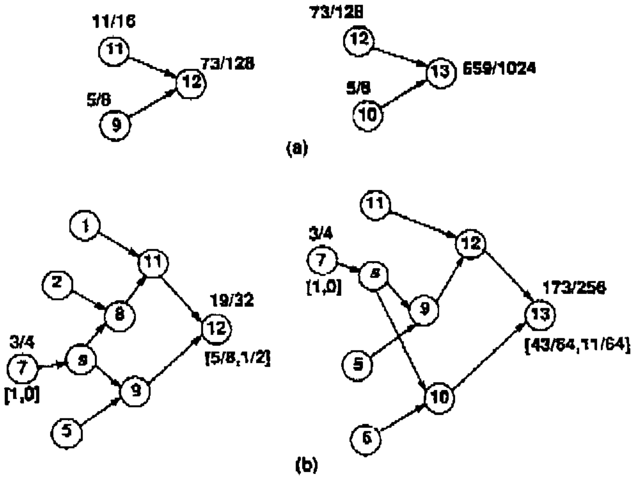


Fig. 7. Graph representation of circuit in Fig. 4.



(a)

(b)

Threshold	Controllability		Error	
	C1(12)	C1(13)	C1(12)	C1(13)
1	73/128	659/1024	-3/128	3/1024
2	73/128	659/1024	-3/128	3/1024
3	19/32	173/256	0	9/256
4	19/32	41/64	0	0

(c)

Fig. 8. Supergates for node 12 and 13 of the circuit in Fig. 4 with restricted distance threshold heuristic: (a) threshold = 1 or 2, (b) threshold = 3, (c) computed controllabilities.

A Pascal program was written implementing the distance threshold heuristic. As the distance threshold was increased, the accuracy of the computed probabilities improved. The root-mean-square error (averaged over all circuit nodes) was found to decrease monotonically with inceasing threshold. For a large threshold, the program computed exact probabilities. It is interesting to observe the influence of supergate size on computation time. The results for an 880-line circuit, C880 [30], appear in Table 7. The much smaller average size of a supergate compared to the 4-bit ALU (Table 1) is worth noting, as also the shallowness of supergates represented by relatively small average depth.The circuit C880 has 60 primary inputs and 26 primary outputs. The maximal supergate cover of the circuit was found to contain 99 supergates. All but 9 of these had fewer than three fanout inputs and depth smaller than four. The average error settles down to a stable value after a distance threshold of 3. On the other hand, the computation cost rises exponentially with the distance threshold.

The basic idea of the above heuristic is to define a distance measure between a pair of signals in the circuit. Other definitions of the distance function may also

Table 7
Distance threshold heuristic (circuit C880)

Distance threshold	Supergate Statistics			Controllability computation time (normalized)
	Number of supergates	Average depth	Average no. of nodes	
1	383	1.00	2.90	1.00 [a]
2	264	1.26	3.64	2.50
3	163	1.32	5.51	7.38
4	141	1.45	6.88	23.47
5	134	1.63	8.19	45.67
6	127	1.63	8.99	78.54

[a] 1.00 = 5.76 CPU s on VAX/11-780.

Table 8
Computational error vs. sample size (4-bit ALU)

	Sample size			
	10	50	100	1000
r.m.s. error	0.103	0.052	0.046	0.011
max. error	0.300	0.140	0.145	0.036
std. dev.	0.073	0.035	0.030	0.007

be used. For example, a distance function may be based on the set of primary inputs in the cone of influence of a line [27].

A sampling technique

Another alternative to exact computation involves *sampling* of the space defined by the fanout inputs to a supergate. We assume that the primary input controllabilities are known so it is possible to generate a sample of fanout inputs. Only a small sample of assignments indicated in Lemma 4.1 is tried. The weighted sum in Lemma 4.1 must, however, be normalized by dividing the sample probability into the result. Sampling may be restricted only to cases where the number of fanout inputs of a supergate exceeds a certain threshold value.

In an experiment the sampling technique was used to compute the controllability values. The effect of sample size on the accuracy of computation in the 4-bit ALU circuit can be seen in Table 8. We note that a sample of as small as 50 patterns on the fanout inputs can provide quite accurate results.

7. Conclusion

We have defined controllabilities, detectabilities, and observabilities as probabilistic measures of circuit testability. Supergate cover allows exact computation of these quantities with reduced effort. In case of detectabilities, it is necessary to treat non-reconvergent lines differently from reconvergent stems. The detectabilities of the latter are computed in a forward trace through the circuit along with all the line controllabilities. Then, the primary output detectabilities are initialized and a backward trace determines the detectabilities of non-reconvergent lines.

The concept of supergates introduced in this paper is crucial to the analysis. We have analyzed the supergate structure of single and multiple output circuits and shown that the covering of the circuit in terms of maximal supergates is unique in both cases. Finding such cover has the worst-case time complexity that is quadratic in the number of nodes in the circuit graph. A distinct feature of our approach is that both the exact and approximate analyses are possible within a common framework, with time complexities ranging from exponential to linear in circuit size. Experimental results show that reasonable accuracy may be achievable even by linear algorithms.

Acknowledgment

Authors thank B. Bhattacharya and F. Brglez for technical discussions and M. Abramovici and M. Pashan for useful comments on the manuscript. L. Pan provided invaluable help with experimental results.

Appendix

Proof of Theorem 5.2.

As explained in the intuitive argument following the statement of the theorem, we need to consider only Part (a) with $b = 1$. The following notation will be used:

$X = (x_1, x_2, \ldots, x_n)$: Input vector to supergate.

$k(X)$: function of supergate inputs realized on line k (abbreviated to k whenever there is no ambiguity.)

s: stem line of supergate whose detectability is desired.

$\kappa(s, X)$: function of s and supergate inputs realized on line k (abbreviated to κ; to determine κ, consider the modified supergate in which the subcircuit feeding s has been removed and s is made an additional input to the supergate.)

$[k(X)]_i$: Restriction of k when binary pattern A_i is assigned to the fanout inputs; the resultant function depends only on the non-fanout inputs.

$[\kappa(s, X)]_i$: Restriction of κ when A_i is assigned as above *and* the (binary) value on s in the original circuit resulting from the assignment is appled to input s; the resulting function depends only on the non-fanout inputs.

$\kappa' = (d\kappa(s, X)/ds)$: The Boolean difference of κ w.r.t. s.

$|f| = $ (*number of true minterms of* $f(X))/2^n$: The syndrome of Boolean function $f(X)$.

Lemma A.1. $[k(X)]_i = [\kappa(s, X)]_i$

The proof is immediate from the definitions.

Lemma A.2. Under the assumption that all input patterns to a supergate are equiprobable,

$$(a)\ S_i(s, k_1) = |[k\kappa']_i|,$$

$$(b)\ S_i(s, k_0) = |[\bar{k}\kappa']_i|.$$

Proof of Lemma A2. The Boolean difference κ' represents the condition of sensitizing s to line k. Thus, the Boolean function $[k\kappa']_i$ represents the condition of sensitizing a 1 to line k from s when A_i is applied to fanout inputs. The syndrome of this function is then equal to $S_i(s, k_1)$ under the equiprobable assumption as stated above. A similar proof can be given for (b).

For the two-input AND gate, we will use m, n, and p to denote the line functions of supergate inputs X; the corresponding line functions in the modified circuit (after deleting the subcircuit feeding s and making s an input) will be denoted by μ, ν, and π respectively. Note that $p = mn$ and $\pi = \mu\nu$.
From Lemma A2

$$S_i(s, p_1) = |[p\pi']_i|.$$

Now from the theory of Boolean differences [25]

$$\pi' = (\mu\nu)' = \mu\nu' \oplus \mu'\nu \oplus \mu'\nu'.$$

Therefore,

$$[p\pi]_i = [mn\mu\nu']_i \oplus [mn\mu'\nu]_i \oplus [mn\mu'\nu']_i.$$

Consider the first term

$$[mn\mu\nu']_i = [m]_i[n]_i[\mu]_i[\nu']_i.$$

From Lemma A1, $[m]_i = [\mu]_i$, hence

$$[mn\mu\nu']_i = [m]_i[n]_i[\nu']_i = [mn\nu']_i.$$

Similarly, $[mn\mu'\nu]_i = [mn\mu']_i$, hence

$$[p\pi']_i = [mn\nu']_i \oplus [mn\mu']_i \oplus [mn\mu'\nu']_i$$
$$= [mn(\nu' \oplus \mu')]_i \oplus [mn\mu'\nu']_i.$$

Now, $\nu' \oplus \mu' = \overline{\nu'} \oplus \overline{\mu'}$, therefore

$$[p\pi']_i = [mn\overline{\nu'}]_i \oplus [mn\overline{\mu'}]_i \oplus [mn\mu'\nu']_i.$$

The three terms on the right hand side are disjoint, hence,

$$S_i(s, p_1) = |[p\pi']_i| = |[mn\overline{\nu'}]_i| + |[mn\overline{\mu'}]_i| + |[mn\mu'\nu']_i|.$$

Since, under the assignment of pattern A_i the signals on lines m and n become independent, we have

$$S_i(s,\ p_1) = |[m]_i|\,|[n\overline{\nu'}]_i| + |[m\overline{\mu'}]_i|\,|[n]_i| + |[m\mu']_i|\,|[n\nu']_i| \qquad \text{(A.1)}$$

Note that

$$C1_i(n) = |[n]_i| = |[n(\nu' \oplus \overline{\nu'})]_i| = |[n\nu']_i| + |[n\overline{\nu'}]_i|$$
$$= S_i(s,\ n_1) + |[n\overline{\nu'}]_i|.$$

Therefore,

$$|[n\overline{\nu'}]_i| = C1_i(n) - S_i(s,\ n_1).$$

Similarly, $|[m\overline{\mu'}]_i| = C1_i(m) - S_i(s,\ m_1)$.

When these substitutions are made on the right hand side of Eqn. (A.1), we get the desired result.

References

[1] Agrawal, V.D. and M.R. Mercer, Testability measures—What do they tell us?, Proc. Int. Test Conf., Philadelphia, PA, November 1982, pp. 391–396.

[2] Bass, S.C., The application of probabilistic modeling to the design and fault analysis of digital networks, Proc. Int. Symp. on Circ. Syst. (ISCAS), Houston, TX, April 1980, pp. 753–756.

[3] Bass, S.C. and J.W. Grundmann, Expected value analysis of combinatorial logic networks, *IEEE Trans. Circuits Syst.* **CAS-28** (1981) 367–382.

[4] Bennetts, R.G., C.M. Maunder and G.D. Robinson, CAMELOT: A computer-aided measure for logic testability, Proc. Int. Conf. Circ. Comp. (ICCC-80), Port Chester, NY, October 1980, pp. 1162–1165; also *IEE Proc.*, **128** (Pt. E) (1981) 177–189.

[5] Brglez, F., On testability analysis of combinatorial networks, Proc. Int. Symp. on Circ. Syst. (ISCAS), Montreal, Canada, May 1984, pp. 221–225.

[6] Brglez, F., P. Pownall and R. Hum, Applications of testability analysis: From ATPG to critical path tracing, Proc. Int. Test Conf., Philadelphia, PA, November 1984, pp. 705–712.

[7] Chang, H.Y. and G.W. Heimbigner, Controllability, observability, and maintenance engineering technique (COMET), *Bell Syst. Tech. J.* **53** (1974) 1505–1534.

[8] Eng. Staff of Texas Instruments, Inc., The TTL Data Book for Engineers, 1981.

[9] Fife, J.L., Predicting fault detectability in combinatorial circuits—A new design tool?, Proc. 12th Des. Auto. Conf., 1975, pp. 290–295.

[10] Goldstein, L.H., Controllability/observability analysis of digital circuits, *IEEE Trans. Circuits Syst.* **CAS-26** (1979) 685–693.

[11] Goldstein, L.H. and E.L. Thigpen, SCOAP: Sandia controllability/observability analysis program, Proc. 17th Des. Auto. Conf., Minneapolis, MN, June 1980, pp. 190–196.

[12] Grason, J., TMEAS - A testability measurement program, Proc. 16th Des. Auto. Conf., San Diego, CA, June 1979, pp. 156–161.

[13] Harary, F., *Graph Theory* (Addison-Wesley, Reading, MA, 1969).

[14] Huisman, L.M. and V. Iyengar, Clarifying statistical fault analysis, *IEEE Design & Test Comput.* **2** (1985) 6.

[15] Jain, S.K. and V.D. Agrawal, Statistical fault analysis, *IEEE Design & Test Comput.* **2** (1985) 38–44.

[16] Kovijanic, P.G., Testability analysis, Int. Test Conf. Digest of Papers, Cherry Hill, NJ, October 1979, pp. 310–316.

[17] Magnhagen, B., Practical experiments from signal probability simulation of digital designs, Proc. 14th Des. Auto. Conf., 1977, pp. 216–219.

[18] Ogus, R.C., The probability of a correct output from a combinatorial circuit, *IEEE Trans. Comput.* **C24** (1975) 534–544.

[19] Parker, K.P. and E.J. McCluskey, Analysis of logic circuits with faults using input signal probabilities, *IEEE Trans. Comput.* **C24** (1975) 573–578.

[20] Parker, K.P. and E.J. McCluskey, Probabilistic treatment of general combinatorial circuits, *IEEE Trans. Comput.* **C-24** (1975) 668–670.

[21] Parker, K.P. and E.J. McCluskey, Sequential circuit output probabilities from regular expressions, *IEEE Trans. Comput.* **C-27** (1978) 222–231.

[22] Savir, J., Good controllability and good observability do not guarantee good testability, *IEEE Trans. Comput.* **C-32** (1983) 1198–1200.

[23] Savir, J., Syndrome testable design of combinatorial circuits, *IEEE Trans Comput.* **C-29** (1980) 442–451.

[24] Savir, J., G.S. Ditlow and P.H. Bardell, Random pattern testability, *IEEE Trans. Comput.* **C-33** (1984) 79–90.

[25] Sellers, E.F., M.Y. Hsiao and L.W. Bearnson, Analyzing errors with the Boolean difference, *IEEE Trans. Comput.* **C-17** (1968) 676–683.

[26] Seth, S.C., B.B. Bhattacharya and V.D. Agrawal, An exact analysis for efficient computation of random-pattern testability in combinatorial circuits, Fault-Tolerant Computing Symposium (FTCS-16) Digest of Papers, Vienna, Austria, July 1986, pp. 318–323.

[27] Seth, S.C., L. Pan, and V.D. Agrawal, PREDICT - Probabilistic Estimation of Digital Circuit Testability, Fault-Tolerant Computing Symposium (FTCS-15) Digest of Papers, Ann Arbor, MI, June 1985, pp. 220–225.

[28] Shedletsky, J.J. and E.J. McCluskey, The error latency of a fault in a sequential circuit, *IEEE Trans. Comput.* **C-25** (1976) 655–659.

[29] Singer, D.M., Testability Analysis of MOS VLSI circuits, Proc. Int. Test Conf., Philadelphia, PA, October 1984, pp. 690–696.

[30] Special Session: Recent Algorithms for Gate-Level ATPG with Fault Simulation and Their Performance Assessment. Proc. Int. Symp. Circuits and Systems (ISCAS), Kyoto, Japan, June 1985, pp. 663–698. Session Organizers: F. Brglez and H. Fujiwara.

[31] Stephenson, J.E. and J. Grason, A testability measure for register transfer level digital circuits, Fault-Tolerant Comp. Symp. (FTCS-6) Digest of Papers, Pittsburgh, PA, June 1976, pp. 101–107.