1986

AN EXACT ANALYSIS FOR EFFICIENT COMPUTATION OF RANDOM-PATTERN TESTABILITY IN COMBINATIONAL CIRCUITS

Sharad C. Seth
University of Nebraska-Lincoln, seth@cse.unl.edu

Bhargab B. Bhattacharya
Indian Statistical Institute, Kolkata, India, bhargab@isical.ac.in

Vishwani Agrawal
AT&T Bell Laboratories, Murray Hill, NJ

Follow this and additional works at: http://digitalcommons.unl.edu/cseconfwork

Part of the Computer Engineering Commons, Electrical and Computer Engineering Commons, and the Other Computer Sciences Commons


This Article is brought to you for free and open access by the Computer Science and Engineering, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in CSE Conference and Workshop Papers by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.
AN EXACT ANALYSIS FOR EFFICIENT COMPUTATION
OF RANDOM-PATTERN TESTABILITY IN
COMBINATIONAL CIRCUITS

Sharad C. Seth, Bhargab B. Bhattacharya,
and Vishwani D. Agrawal

Reprinted from the 16th ANNUAL INTERNATIONAL
SYMPOSIUM ON FAULT-TOLERANT COMPUTING SYSTEMS
July 1-4, 1986 Vienna, Austria

AN EXACT ANALYSIS FOR EFFICIENT COMPUTATION OF RANDOM-PATTERN TESTABILITY IN COMBINATIONAL CIRCUITS

Sharad C. Seth
Bhargab B. Bhattacharya
University of Nebraska
Lincoln, Nebraska 68588

Vishwani D. Agrawal
AT&T Bell Laboratories
Murray Hill, New Jersey 07974

ABSTRACT - Experimental evidence shows that low testability in a typical circuit is much more likely due to poor observability than poor controllability. Thus, from theoretical and practical standpoints, it is important to develop an accurate model for observability computation. One such model, in terms of supergates, is proposed in the first part of this paper. It not only complements our earlier work. It is now possible to obtain exact random-pattern testability for each line in a circuit.

The second part of the paper analyzes the supergate structure of a circuit from a graph theoretic viewpoint. Finding a supergate is related to determining the dominator tree in a modified circuit graph which provides an exact bound on the complexity of computation. The uniqueness of the tree extends to multiple output circuits and the complexity of finding it shown to be quadratic in the number of nodes in the circuit graph. By prior determination of the maximal supergate covers, unnecessary duplication in computation of testability values can be avoided.

1. INTRODUCTION

Design for testability has assumed increasing urgency due to higher circuit densities. Consequently, the need for developing good measures of testability and for computing them efficiently with sufficient accuracy also becomes greater. Increased understanding of the limitations of the traditional deterministic measures [AGR82, GOL879] has prompted recent efforts on improved measures based on applying random patterns to the circuit [SAV84, SET85]. Random-pattern testability is shown to be effective not only in identifying areas of poor testability but also for simulation-free fault sensitivity and automatic test-pattern generation [AGR85a, AGR85b, BRG84, WUN85].

Both deterministic and probabilistic methods calculate, for each circuit node, values related to its controllability and observability. The deterministic measures, such as SCOAP [GOL89], are non-normalized and only intuitively justified. Probabilistic measures, on the other hand, are signal probabilities which always lie in the [0.0, 1.0] range. For a line in a combinational circuit, these measures are related to the boolean function realized by the line. Thus, if all input patterns are equiprobable, a line's one-controllability is identical to its syndrome [SAV80]; its observability at an output, can similarly be related to the boolean difference [SEM68].

This paper extends the supergate analysis of PREDICT [SET85] for exact computation of observabilities. Also, sections 4 and 5 analyze the supergate structure of a circuit from a graph-theoretic viewpoint. Finding a supergate is related to determining the dominator tree in a modified circuit graph thus providing an exact bound on the complexity of this computation. Determination of the maximal supergate covers avoids unnecessary duplication in the computation of testability.

2. BACKGROUND AND NOTATION

Supergate: In PREDICT, computation of controllability for a node (a primary input or a gate output) is carried out on a circuit subgraph called the “supergate” of that node. A graph-theoretic definition of supergate appears in Section 4. To determine the supergate of a node X in the circuit, one starts tracing the circuit back from X. If X is a primary input we are done. Otherwise, the signals on the inputs of gate X are compared pairwise. If the signals are mutually independent, we are again done. Otherwise, such signals are not pairwise independent define a new “frontier” for further backtracing. One of the frontier nodes is arbitrarily chosen for backtracing and a new “frontier” is defined after tracing signals at its input. The process stops as soon as the frontier becomes null. As an example, Fig. 1 shows a NAND circuit with the supergate of primary output shown within the shaded triangle.

The inputs of a supergate are partitioned into two classes: fanout and nonfanout. The former are characterized by more than one path to the supergate output. In the example, lines c, d, and e define supergate inputs with d and e as the fanout inputs. It may be noted that a fanout input of a supergate might not fanout itself (e.g., line d) but may have multiple paths to the supergate output through an internal fanout.

Conditional Computation within a Supergate: The computation within a supergate is carried out in the context of a fixed pattern of binary inputs applied to the fanout inputs of the supergate. Assume that all possible patterns on the fanout inputs are indexed in some arbitrary way and let A_i represent the i-th pattern. Then C_i(k) is the conditional zero-controllability of line k in the supergate when A_i is applied to the fanout inputs. This is the probability of setting line k to zero when a randomly selected input to the network impresses the pattern A_i to the fanout inputs of the supergate. Conditional one-controllability is similarly defined and is denoted by C_1(k).

In [SET85] it was shown how these conditional controllability values can be propagated forward in a supergate essentially by assuming them to be independent. For example, for a two-input AND gate with lines m and n as inputs and p as the output, we have:

\[ C_0(p) = C_0(m) + C_0(n) - C_0(m)C_0(n), \]
\[ C_1(p) = C_1(m)C_1(n). \]

The zero-controllability (one-controllability) of line k is the probability of setting line k to zero (one) when a randomly selected input pattern is applied to the network. These are denoted, respectively, as C_0(k) and C_1(k) and can be computed from the conditional controllabilities [SET85].

\[ C_1(k) = \sum_{A_i} C_1(k)p(A_i) \]
where, $P(A_i)$ is the probability of $A_i$ appearing at the fanout inputs when a random pattern is applied to the network.

**Example:** For the circuit in Fig. 2(a), assume each input can be independently set to one with probability $1/2$ and we want to compute the one-controllability of line $g$. The supergate of $g$ is the whole circuit with the primary input $B$ as a fanout input. Let $A_0$ and $A_1$ represent the assignments $B=0$ and $B=1$, respectively. Then, as shown in Figs. 2(b) and 2(c), $C_1(g) = 1$ and $C_1(g) = 1/2$. Thus, the above sum yields $C_1(g) = (1 + 1/2)(1/2) = 3/4$.

The conditional zero-observability (conditional one-observability) of line $k$ is the probability of observing line $k$ at the supergate output under the condition that $A_i$ is applied to the fanout inputs and line $k$ is set to zero (one). These are denoted, respectively, as $B_0(k)$ and $B_1(k)$.

The conditional zero-detectability (conditional one-detectability) of line $k$ is the probability of simultaneously setting line $k$ to zero (one) and observing it at the supergate output when a random input is applied to the network. Denoting these by $D_0(k)$ and $D_1(k)$, respectively, we write:

$$D_0(k) = C_0(k)B_0(k)$$

$$D_1(k) = C_1(k)B_1(k)$$

$D_0(k)$ is the probability of detecting the fault "line $k$ stuck-at-one" at the supergate output when a random input to the network applies $A_i$ to the fanout inputs of the supergate. Since the observabilities, as defined above, are conditioned on setting the line being observed to a value, we avoid the anomaly, noted in [SAY83], that high values of a line's controllability and observability may not imply high testability.

### 3. EXACT CALCULATION OF DETECTABILITY

We consider the problem of determining the detectability of a line in a supergate at its output. All the proposed solutions to this problem (based on the circuit structure) involve approximations and thus are inexact [JAI85, SET85, BRG84]. The errors arising in such computations can be attributed to two distinct classes of simplifications: (1) the observability of line through a chain of gates can be found by considering the gates one at a time, and (2) the observability of a reconvergent fanout stem is a fixed function of its branch observabilities. The first simplification is, indeed, valid in the context of a pattern applied to fanout inputs. This leads to an efficient procedure for computing detectabilities for all lines which are not reconvergent.

**Observabilities of Non-reconvergent Lines:** Consider 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,

$$B_0(a) = B_0(c)\frac{\text{Prob}(b=1 | a=0)}{\text{Prob}(a=0)}$$

$$B_1(a) = B_1(c)\frac{\text{Prob}(b=1 | a=1)}{\text{Prob}(a=0)}$$

Expressions for observabilities for other types of gates are derived in a similar manner [JAI85]. Notice that in the above $B_i(a)$ is obtained by multiplying two probabilities. $B_i(c)$ and $\text{Prob}(b=1 | a=0)$. This assumes that these two probabilities correspond to independent events.

To examine the validity of this assumption, consider again the circuit shown in Fig. 2(a). Line $g$ is a primary output with observabilities set to unity. If the inputs $A$ and $B$ have equal probabilities of 0 and 1, then the expressions of the above type give the following result:

<table>
<thead>
<tr>
<th>$a$</th>
<th>$b$</th>
<th>$e$</th>
<th>$f$</th>
<th>$g$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1/2</td>
<td>1/2</td>
<td>3/4</td>
<td>1/2</td>
</tr>
<tr>
<td>1</td>
<td>1/2</td>
<td>1/2</td>
<td>1/4</td>
<td>1/2</td>
</tr>
<tr>
<td>0</td>
<td>1/6</td>
<td>1/3</td>
<td>1/2</td>
<td>1/3</td>
</tr>
<tr>
<td>1</td>
<td>1/2</td>
<td>1/1</td>
<td>1/1</td>
<td>1/1</td>
</tr>
</tbody>
</table>

Line $b$, a reconvergent stem, will be considered in the next section. Line $a$ is observable at $g$ whenever $b$ is 1. Therefore, $B_0(a) = B_1(a) = 1/2$. However, our computation, in the above table, incorrectly gives $B_0(a) = 1/6$. This is because the observability of $e$ depends on $f$, which is derived from $b$. Such situations are common in the presence of reconvergent fanouts. The Stefan formulas take account of the correlation between the lines at the input of the same gate, e.g., $e$ and $f$ are correlated due to the common influence of $b$ on them; their observabilities are correctly computed. However, correlations extending beyond the inputs of a single gate are ignored by Stefan.

Consider a two-input gate, with inputs $a$ and $b$, and output $c$, embedded in a supergate. Assume a pattern $A_i$ on the fanout inputs of the supergate. Under these conditions, the signals on the two inputs of the AND gate indeed become independent, therefore, Stefan's
assumption are true for conditional computation. In particular, to
detect a 0 on line a (when \( A_i \) is applied to fanout inputs of the
supergate), a 1 must be applied to line b and the resulting 0 on line c
must be observed. Similar arguments will apply to detection of 1 on
line a. Thus,

\[
C_1 = \begin{cases}
D_0(b) - C_1(b) & (1a) \\
D_1(a) - C_1(b)D_1(c) + C_0(c) & (1b)
\end{cases}
\]

Similarly, for a two-input OR gate:

\[
C_2 = \begin{cases}
D_0(a) - C_2(b)D_0(c) & (2a) \\
D_1(a) - C_2(b)D_1(c) + C_0(c) & (2b)
\end{cases}
\]

These and similar equations for other elementary gates may be used to
obtain conditional detectabilities of all lines up to (but not including)
fanout stems.

**Theorem 1:** Let \( k \) be a non-reconvergent line in a supergate
and let \( D_0(k) \) and \( D_1(k) \) represent, respectively, its zero and one
detectabilities. Then

(a) \( D_0(k) = \sum_{A_i} D_0(s)P(A_i) \)

(b) \( D_1(k) = \sum_{A_i} D_1(s)P(A_i) \)

where, \( P(A_i) \) is the probability of applying pattern \( A_i \) to the fanout
inputs of the supergate.

Returning to our example of Fig. 2, the conditional detectabilities can be
computed from Figs. 2(b) and 2(c). Assuming again that line g is a primary output, using Eqns. (1) and (2), the results are tabulated below.

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1&lt;sub&gt;b-0&lt;/sub&gt;</td>
<td>1/2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>C1&lt;sub&gt;b-1&lt;/sub&gt;</td>
<td>1/2</td>
<td>1/2</td>
<td>0</td>
<td>1/2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D0&lt;sub&gt;b-1&lt;/sub&gt;</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D0&lt;sub&gt;b-0&lt;/sub&gt;</td>
<td>1/2</td>
<td>1/2</td>
<td>1/4</td>
<td>1/2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1&lt;sub&gt;b-1&lt;/sub&gt;</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1&lt;sub&gt;b-0&lt;/sub&gt;</td>
<td>1/2</td>
<td>1/2</td>
<td>1/4</td>
<td>1/2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

From Theorem 1, therefore, \( D_0(a) = D_1(a) = (1/2 + 0)(1/2) = 1/4 \),
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 \( D_0(a) = D_0(s)/D_0(s) = 1/2 \). Thus non-stem line observabilities
can also be correctly computed.

**Detectabilities of Reconvergent Stems:** It is attractive to think of
extending the above method to determine the observabilities of
reconvergent stems. Then, all line detectabilities could be found in a
single backward trace of the circuit. Unfortunately, a stem's
detectability is not just a function of its branch detectabilities but also
depends on the inversion parity of the branches along reconvergent
paths. The solution proposed in this paper gives the goal of
determining all detectabilities in a single backward trace. Instead,
reconvergent stem detectabilities are found in a forward trace
combined with the computation of line controllabilities.

Let \( s \) be a reconvergent fanout stem in a supergate and let \( A_i \) be applied to the fanout inputs of the supergate. Note that, from the definition of
fanout inputs to supergate, it follows that the value on stem \( s \) is
uniquely determined by \( A_i \). 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 \( s \),
symbolically written as \( S(s, k_0) \) and \( S(s, k_1) \). 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.

The conditional sensitizations of line \( k \) from stem \( s \) can be stated
in functional terms. Let the vector \( X = (x_1, x_2, ..., x_n) \) be the inputs
to the supergate and let \( k(X) \) be the function realized on line \( k \) (we will
abbreviate \( k(X) \) as \( k \) whenever there is no ambiguity). Denote:
(a) the restriction of \( k(X) \) when \( A_i \) is assigned to the fanout inputs as
the function \( k(X)_{A_i} \);
(b) the syndrome \([SAV80]\) of \( k(X) \) (defined as the number of one's in
the truth table of \( k(X) \) divided by 2) as \([k(X)]\);
(c) \( k(X) \) considered as a function of the stem \( s \) and supergate inputs \( x \)
as \( k(s, X) \), and
(d) the boolean difference \([SEL68]\) of \( k \) with respect to \( s \) as \( k' \).

**Lemma 1:** Under the assumption that all input patterns to a supergate
are equiprobable,

(a) \( S(s, k_0) = [k'k'] \)

(b) \( S(s, k_1) = [kk'] \)

**Proof:** The boolean difference \( k' \) represents the condition of sensitizing
\( s \) to line \( k \). Thus, the boolean function \([k'k']\), 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 \( S(s, k_1) \) under the
equiprobable assumption as stated above. A similar proof can be
given for (b).

**Theorem 2:** Let \( S(s, k_b) \) 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 \( p \):

\[
S(s, p) = S(s, m_b)C_1(m) - S(s, n_b)
\]

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

\[
S(s, p) = S(s, n_b)C_1(m) - S(s, m_b)
\]

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

\[
S(s, p) = S(s, m_b)
\]

**Proof:** (c) is trivial. The proofs of (a) and (b) are very similar, hence
we will only prove (a). Consider the case when \( b = 1 \). From Lemma 1
\( S(s, p) = [[pp']] \)

where, \( p' \) is the boolean difference of \( p \) with respect to \( s \). Now
\( p - m.n \) for an AND gate, therefore \([BRE76, Section 2.11]\),

\[
p' = (mn)' = mn' \oplus mn' \oplus m'n'
\]

and

\[
p' = mn' \oplus mn' \oplus m'n'
\]

Therefore, from Lemma 1

\[
S(s, p) = \left[ [[pp']] \right] \left[ [[mmn'] \oplus mm'n' \oplus mm'n'] \right]
\]

Since all the three terms are mutually disjoint on the right hand side, the
syndrome is the sum of syndromes of each term, hence
\[ S_i(s, \alpha_j) = |(m_m \cap m_n) + |(m_m \cap m_n)| + |m_m \cap m_n| \]
\[ = |m_m| |m_n| + |m_m| + |m_n| \]
\[ + |m_m| |m_n| \]
Since \(|m_m|\) and \(|m_n|\) do not involve any variables in common.
\[ C_i(m) = |m| - |(m \cap m')| - |m| + |m_m| \]
\[ = S_i(s, m) + |m_m| \]

Therefore,
\[ |m_m| = C_i(m) - S_i(s, m) \]

When this substitution is made on the right hand side of (3) we get the desired result. A similar proof can be given for the case when \(b = 0\).

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 2. The stem's observabilities at \(x\) are obtained by appropriately combining these conditional sensitizations as indicated in the following theorem:

**Theorem 3:** \([\text{SET6}]\) Let \(s\) be a reconvergent stem in a supergate and let \(Z_s(N_j)\) be the subset of patterns on the fanout inputs of the supergate which cause a zero (one) to appear on \(s\). Then

(a) \(D(s) = \sum \left( S_i(s, x) + S_i(s, x) \right) P(A) \)

(b) \(D(s) = \sum \left( S_i(s, x) + S_i(s, x) \right) P(A) \)

Example: Consider the carry-logic circuit of Fig. 3(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 \(S_i(s, k)\) and \(S_i(s, k)\) as an ordered pair within square brackets. Also shown for line \(k\) is \(C_i(k)\). Stem \(s\) is 1 for three combinations of values on \(A\) and \(B\): 00, 01, and 10. Figures 3(a)-3(c) show the conditional sensitizations (obtained from Theorem 2) and one-controllabilities (as derived in PREDICT) for the three cases. Each of these combinations occurs with probability, \(P(A) = 1/4\), therefore

\[ D(s) = [(+1) + (1/2+0) + (1/2+0)] P(A) = 1/2 \]

**4. SUPERTUBE STRUCTURE AND ACCELERATION OF TESTABILITY ANALYSIS**

As we saw in Section 2, informally, the supergates of a node \(X\) in the circuit is the smallest predecessor subnetwork feeding \(X\) whose inputs are logically independent, i.e., have no signal correlation. We now introduce a formal definition.

For simplicity, consider first a single-output combinational network \(N\). The structure of \(N\) has an equivalent representation in the form of a directed graph \(G(V, E)\), called the circuit graph, whose nodes are the primary inputs, the primary outputs, and the gates in \(N\), and whose edges represent the connections in \(N\), oriented in the direction of signal flow (see Figs. 4(a) and 4(b)). A fanout in \(N\) is represented by a node with outdegree greater than one in \(G\); a primary input (output) becomes a node whose indegree (outdegree) is zero. Two distinct directed paths \(P_1\) and \(P_2\) in \(G\) are said to be reconvergent if they emanate from a common vertex (say \(A\)) and terminate at another common vertex (say \(B\)). Node \(B\) is called a reconvergent node. Obviously, \(B\) is a fanout node.

Let \(R(X)\) denote the set of all nodes in \(G\), from which node \(X\) is reachable by a directed path. Then, the cone of influence \(C(X)\) of node \(X\) is a proper subgraph \(G'(V', E')\) of \(G\), such that \(V' = R(X)\).

The supergate of a node \(X\) in a circuit graph \(G\), denoted by \(SG(X)\), is a proper subgraph \((V', E')\) of the cone of influence \(C(X)\), such that the following conditions hold:

(i) \(X\) and all its predecessor nodes (i.e., those with edges directed towards \(X\)) are in \(SG(X)\).
(ii) \(SG(X)\) is a proper subgraph of \(G\).
(iii) For every pair of nodes \(v, v_j\) in \(C(X)\), each with indegree zero, \(R(v)\) is disjoint from \(R(v_j)\).
(iv) \(V'\) is a minimum vertex set satisfying the above properties.

From the definition it is obvious that the supergate of every node is unique. As an example, the vertex set of the supergate of node 16 in Fig. 4(b) is given by \([16, 13, 12, 11]\).

The set of input nodes \(I(X)\) of a supergate \(SG(X)\) is the set of nodes in \(SG(X)\) with in degree zero. These are further partitioned into two groups:

(i) \(INF(X)\), the non-fanout inputs, are those with a unique directed path to \(X\), and
(ii) \(INF(X)\), the fanout inputs, are those inputs each of which has at least two distinct direct paths to \(X\) (this does not imply that these nodes have an outdegree greater than one, since two distinct paths may have a common initial segment).

The output node of a supergate \(SG(X)\) is the node \(X\) itself.

**Remark 1:** For every node \(Y\) in \(SG(X)\) which is not an input node, \(SG(Y)\) is a proper subgraph of \(SG(X)\), i.e., subgraphs of interior nodes in \(SG(X)\) are contained within \(SG(X)\).

A supergate is said to be maximal if it is not properly contained in a larger supergate. As an example, for the network of Fig. 4(a), the circuit graph and the maximal supergates (shown within dotted lines) are given in Fig. 4(b).

**Remark 2:** From the previous remark and the definition of maximal supergates, it follows immediately that the partitioning of the circuit graph affected by maximal supergates is unique.

For a single-output combinational circuit, there exists an interesting topological relationship among the maximal supergates. We define a reduced circuit graph (RCG) as a directed graph \((V, E_v)\), where \(V\) denotes the set of supergates and a directed edge \((i, j)\) exists if the output node of \(SG(i)\) is an input node of \(SG(j)\).
such that RG) and R(j) both include at least one common (fanout) input of this linear ordering, the dominance relation induces a partial order and that the set of nodes which dominate a given vertex is linearly ordered [AH074]. Because of this linear ordering, the immediate dominator of a node n can be uniquely defined as the dominator that is closest to n on any path from the source s to n. This allows depiction of dominance relationship amongst vertices using a tree called the dominator tree [TAR74] in which the source vertex is the root and the predecessor of every other node is its immediate dominator.

Algorithm: To find all maximal supergates in a circuit graph of a single-output combinational network, proceed as follows:

Step 1: Given the circuit graph G, construct a directed flow graph FG as follows: (i) Delete the primary-output node and its incident edge. (ii) Reverse the directions on the edges which are incident on the node corresponding to the output gate and make all other edges bidirectional.

Step 2: Construct the dominator tree of FG.

Step 3: Start from the root node of the tree and collect all the nodes (including the root) which are children of the root. Include also all single successors of these children, if any. (This set gives the maximal supergate corresponding to the output node.) Remove all the edges so far covered and iterate the process until the tree is empty.

Example: Consider the circuit graph in Fig. 4(b). The dominator tree corresponding to its flow graph FG is shown in Fig. 5. The maximal supergates are SG(19), SG(11), SG(12), SG(7), and SG(8), the vertex sets of which are given by:

- \(
  V[SG(19)] = \{19, 18, 17, 16, 15, 14, 13, 11, 12\}
\)

- \(
  V[SG(11)] = \{11, 7, 8\}
\)

- \(
  V[SG(12)] = \{12, 9, 10, 6, 5\}
\)

- \(
  V[SG(7)] = \{7, 1, 2\}
\)

- \(
  V[SG(8)] = \{8, 3, 4\}
\)

Remark 4: Clearly, the complexity of the above algorithm is dominated by that of Step 2, which can be implemented with a space complexity of \(O(|V| + |E|)\) and with time complexity \(O(|V|\log|V| + |E|)\), by using an efficient depth-first search technique [TAR74], where \(V\) and \(E\) denote the vertex and edge sets of the circuit graph.

5. SUPERGATE STRUCTURE OF MULTIOUTPUTS

For a multioutput circuit all maximal supergates can be easily found by using the earlier dominance algorithm for each output individually. Obviously, the computational effort required to evaluate the detectability of the lines in the circuit depends strongly on the complexity of maximal supergates, determined by their size and the number of fanout inputs. For a multioutput circuit, a maximal supergate of a primary output might partially or fully overlap a maximal supergate of another primary output. To avoid computation of testability parameters for overlapping nodes/edges more than once we require a minimum covering of all the circuit nodes with maximal supergates. We will show that because of certain structural relations amongst the supergates, such a minimum cover is unique and can be found in polynomial time.

Example: Fig. 6 shows a NAND realization of a full-adder circuit and its circuit graph. The vertex sets of the maximal supergates corresponding to the sum output are:

- \(
  V[SG(11)] = \{11, 10, 9, 8, 7, 1\}
\)

- \(
  V[SG(7)] = \{7, 6, 5, 4, 3, 2\}
\)

and for the carry output

- \(
  V[SG(13)] = \{13, 8, 1, 7, 5, 6, 4, 3, 2\}
\)

Note that SG(13) properly includes SG(7). Clearly, the computation of controllability and detectability for the nodes in SG(13) obviates the need to recompute these parameters for the nodes in SG(7). In a multioutput circuit, such a redundant computation can be eliminated by finding a minimum set of maximal supergates which cover all the nodes in the circuit graph.

Lemma 2: [SET86] In the graph of a multioutput circuit, every maximal supergate SG(X) corresponding to a primary output is either properly contained within another maximal supergate SG(Y) (corresponding to another primary output), or has at least one node which is not covered by any other maximal supergate.

Corollary: In a multioutput circuit graph, no maximal supergate can be completely covered by the union of all other supergates, unless it is totally covered by one maximal supergate alone.

Theorem 4: The minimum cover of all nodes in a multioutput circuit graph with maximal supergates is unique.

Example: In Fig. 6, the minimum cover consists of [SG(11), SG(13)]. [Nodes 12 and 14 can be ignored without loss of information.]

Remark 5: From the uniqueness of the cover, it follows that once maximal supergates for individual primary outputs are determined by the algorithm in the last section, finding the minimum cover in a circuit graph having \(|V|\) nodes is at most \(O(|V|^2)\). Since this dominates the complexity of finding maximal supergates (Remark 4) for
Fig. 5 Dominator tree and maximal supergate partitioning.

Fig. 6 The full-adder circuit and its circuit graph with the maximal supergates.

individual primary outputs, the unique cover can be identified in at most $O(|V|^2)$.

If so identified in a preprocessing step, the minimum cover of a multioutput circuit graph will help accelerate computation.

6. CONCLUSION

We have introduced a model for exact computation of line observabilities but, in order to do so it is necessary to handle non-reconvergent lines in a supergate differently from reconvergent stems. The observabilities of the latter are computed in a forward trace through the circuit along with all the line controllabilities. Then, the primary output observabilities are initialized and a backward trace determines the observabilities of non-reconvergent lines. Whenever a reconvergent stem is encountered, the backward trace restarts with its observability value computed earlier. PREDICT-like approximations are easily extended to include this more accurate method for observability computation. For multiple output circuits, the order in which the supergates are considered, affects the efficiency of computation. 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. Further, finding such cover has the worst-case time complexity that is quadratic in the number of nodes in the circuit graph.

Acknowledgment — Authors thank F. Brglez for technical discussions and M. Pashan for comments on manuscript.

REFERENCES