

Rapid Diagnostic Fault Simulation of Stuck-at Faults in Sequential Circuits using Compact Lists

Srikanth Venkataraman [†], Ismed Hartanto [†], W. Kent Fuchs [†]
Elizabeth M. Rudnick [‡], Sreejit Chakravarty [◇], and Janak H. Patel [†]

[†] Center for Reliable and High-Performance Computing
University of Illinois, Urbana, IL 61801

[‡] Motorola Inc., Austin, TX 78735

[◇] State University of New York, Buffalo, NY 14260

Abstract

This paper describes a diagnostic fault simulator for stuck-at faults in sequential circuits that is both time and space efficient. The simulator represents indistinguishable classes of faults as memory efficient lists. The use of lists reduces the number of output response comparisons between faults and hence speeds up the simulation process. The lists also make it easy to drop faults when they are fully distinguished from other faults. Experimental results on the ISCAS89 circuits show that the simulator runs significantly faster than an earlier work based on distinguishability matrices and is faster and more memory efficient than a recent method based on lists of indistinguishable faults. The paper provides the first reports on pessimistic and optimistic diagnostic measures for all faults of the large ISCAS circuits.

1 Introduction

The aim of *fault location* or *diagnosis* is to locate device failures. Diagnosis may be intended for identification and replacement of a faulty sub-circuit or may be performed with a view to improving the manufacturing process. Efficient generation of diagnostic test vectors requires a fast fault simulator capable of determining the diagnostic capability of a given test set. Current diagnostic fault simulators are unable to evaluate the large benchmark circuits due to time and space constraints.

During fault simulation of a circuit from an unknown state, a good or faulty sequential circuit can produce a 0, 1 or X ,

This research was supported in part by the Semiconductor Research Corporation (SRC) under grant 94-DP-109 and in part by the Joint Services Electronics Program (JSEP) under grant N00014-90-J-1270.

32nd ACM/IEEE Design Automation Conference ©

Permission to copy without fee all or part of this material is granted, provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission. © 1995 ACM 0-89791-756-1/95/0006 \$3.50

on each primary output for each test vector input, where X is an unknown value whose actual binary value depends on the initial state of the machine. If fault simulation indicates that a fault f_i produces an output of 0 and another fault f_j produces an output of 1 on the same primary output for the same input, then the faults f_i and f_j are said to be distinguished. However if a fault f_i produces an output of 0 or 1 and another fault f_j produces an output of X , then the faults f_i and f_j may possibly not be distinguished. Therefore, the pessimistic assumption is made that an output of 1 or 0 is indistinguishable from an output of X .

Camurati et al. [1] proposed two diagnostic measures. *Diagnostic Resolution (DR)* is the fraction of fault pairs distinguished by a test set. *Diagnostic Power (DP)* is the fraction of faults that are fully distinguished. A fault is fully distinguished if the test set distinguishes it from every other fault in the fault list.

A third measure [2] which gives a more complete picture is to identify sets of fault equivalence classes and report the number of *fault equivalence classes* by size. This is extended to sets of *indistinguishable classes* [3] to account for unknown values occurring at the outputs of sequential circuits during simulation. Another measure, the *Diagnostic Expectation* [4], is the average of indistinguishability class sizes over all faults. It gives the average size of the list of faults indistinguishable from any fault, given that all faults are equally likely to occur.

Diagnostic fault simulation is the process of determining these measures. Diagnostic fault simulation for sequential circuits by Rudnick et al. [3] uses a distinguishability matrix. The distinguishability matrix is an f -by- f matrix, where f is the number of faults. An entry of 1 indicates that the two faults specified at the intersection of the row and column are distinguished by some sequence of test vectors in the test set. It requires $O(f^2)$ space, and the *time complexity* is $O(v \cdot o \cdot f^2)$, where v is the number of vectors in the test set, o the number of outputs in the circuit and f the number of faults. It is obvious that the computational requirements increase considerably with the increasing number of simulated faults.

Ryan, et al. [4] mention that a more efficient way to represent indistinguished faults is by using lists of faults. Jou and

Chen [5] represent pairs of indistinguishable faults using lists. This representation is a compact implementation of the distinguishability matrix. It is equivalent to storing only those entries of the distinguishability matrix with values of 0. Here faults may appear on multiple lists. The worst case space complexity is still $O(f^2)$.

Our representation is significantly different in that it does not explicitly store the indistinguishability relationship between all pairs of faults but represents the indistinguishability relationship between classes of faults. Each fault is present in only one of the classes. This makes our representation more compact than those previously proposed [3] [5]. It also reduces the number of output response comparisons between faults and hence speeds up the simulation process. Further speedup is achieved by using fault simulation knowledge to categorize faults and avoid a number of explicit output response comparisons. This paper is the first to report results for all faults of the large ISCAS circuits. Experimental observations indicate that the memory requirements grow almost linearly with the number of faults in the benchmark circuits. The use of lists also makes it easy to drop faults when they are fully distinguished from other faults. Fault dropping reduces the total diagnostic time on an average by 6% in the benchmark circuits.

2 The Diagnostic Fault Simulator

In the following the diagnostic fault simulation procedure is described. First the definitions and concepts essential to understanding the simulation are given, next the *compact list* data structure is presented, and then the use of the *fault status* to speed up the diagnostic fault simulation process is explained.

Definition 1 Two faults f_i and f_j are indistinguishable with respect to a test set T , if for every primary output and every input vector, either one (or both) of the output values of f_i or f_j is an X , or both the output values are not X but the same.

In previous implementations of diagnostic fault simulators [3] [5], two faults were declared distinguished if and only if the value of a primary output was 1 in one faulty circuit and 0 in the other faulty circuit, and the good circuit value was known. This implies that a necessary condition for two faults to be distinguished is that one of them be detected. However, this is not a strict necessary condition. Two faults are distinguished even when they are both declared undetected if for some input vector one of them has a 0 on a primary output, and the other a 1 on the same primary output.

Definition 2 Two faults f_i and f_j are diagnostically equivalent with respect to a test set T , if for every primary output and every input vector, output values of f_i and f_j are the same. Here an X response is considered a different response from a 1 or a 0 response.

Definition 3 A diagnostic equivalence class is a set of faults such that every pair of faults in the class is diagnostically equivalent by definition 2.

Definition 2 defines an equivalence relation over the set of faults F , and partitions F into sets of diagnostic equivalence classes. Each fault $f_i \in F$ is present in only one diagnostic equivalence class. The faults within a diagnostic equivalence

Table 1: Output Responses of an Example Circuit

Tst vct	Gd. cct.	f1	f2	f3	f4	f5	f6	f7
v1	0xx	1xx	0x1	x1x	0x1	0xx	1xx	0x1
v2	x01	1xx	01x	x10	11x	x01	0xx	0x1

class cannot be distinguished under the single observation time strategy. However, these faults may not be equivalent in the boolean algebraic sense.

Definition 4 An indistinguishability class is a set of faults such that every pair of faults in the class is indistinguishable by definition 1.

Definition 1 does not define an equivalence relation. A fault $f_i \in F$ may be present in more than one indistinguishability class.

Table 1 gives the output responses of an example circuit with 3 primary outputs, 7 faults and 2 test vectors. The good and 7 faulty circuits are considered. This will serve as the explanatory example throughout the paper.

A *diagnostic tree* is a connected, directed acyclic graph. A level in a diagnostic tree corresponds to one simulated test vector, while an edge in the tree corresponds to one set of distinct primary output values after the corresponding vector is simulated. Each node represents a *diagnostic equivalence class*. Since each primary output can have a 0, 1 or X value, each level of the diagnostic tree can have a maximum of 3^P branches, where P is the number of primary outputs in the circuit. The diagnostic tree for the example circuit of Table 1 can be seen in Figure 1.

2.1 The Data Structure

For combinational circuits where the output response from a simulation is binary (1 or 0), two faults in different diagnostic equivalence classes are distinguished. However, we have to deal with the X value problem in sequential circuits. Two faults f_1 and f_2 in the same diagnostic equivalence class are by definition indistinguishable. However, it is also possible for faults from two different diagnostic equivalence classes to be indistinguishable when one of them has an X on some output and the other a 1 or 0 on that same output. For example, in Table 1 fault (f_3) with a response $X1X$, and faults (f_2, f_4, f_7) with a response $0X1$ on test vector $v1$, cannot be declared as distinguished as there is no primary output with a response of 1 for one class and 0 for the other. Such faults and their corresponding diagnostic equivalence classes are said to be *potentially distinguished*.

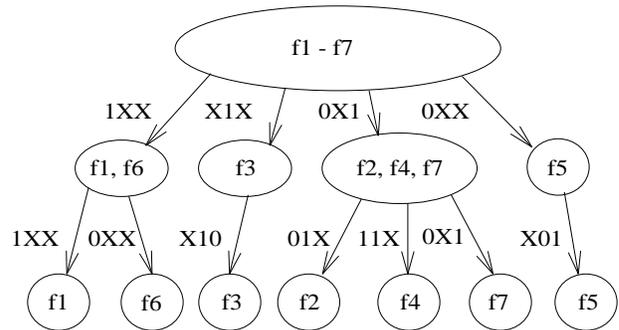


Figure 1: Diagnostic Tree for the Example Circuit

The diagnostic simulator represents the indistinguishability relationship between all faults, for the test vectors simulated until a particular point in time, in a core data structure. The data structure corresponds to a level of the diagnostic tree. It contains the lists of diagnostic equivalence classes and represents the potentially distinguished relation between these classes using *potentially distinguished pointers*. This list representation is efficient since each fault is only represented once. The data structure is modified after simulating each new test vector and obtaining the output responses. The simulation of test vectors to obtain output responses is performed by fault simulation [7].

The data structure for the example in Table 1 can be seen in Figure 2. The diagnostic equivalence classes after the simulation of test vector v1 are (f1, f6), (f3), (f2, f4, f7) and (f5). A list of potentially distinguished pointers is associated with every diagnostic equivalence class. In order to minimize the memory usage, the potentially distinguished pointers are maintained from left to right. For example, in Figure 2 after the application of the vector v1, there is a potentially distinguished pointer from class 1 to class 2 but not vice versa.

2.2 The Use of Fault Status Information

The underlying fault simulator simulates the good and faulty circuits and compares their output responses. If a faulty circuit has a response of 1 on a particular output and the good circuit has a response of 0 on the same output, or vice-versa, then the faulty circuit is declared to be *detected* by the fault simulator. If, however, the good circuit has a 0 or 1 on an output and the faulty circuit an X on the same output then the fault is said to be *potentially detected*. Since the diagnostic fault simulator is built on top of a fault simulator there is no need to repeat the comparison effort. The diagnostic fault simulator keeps track of the comparisons done by the fault simulator by looking at the *fault status*.

We define a new class of faults that is useful for diagnostic simulation.

Definition 5 A fault is potentially excludable if it is neither a detected nor a potentially detected fault and there is at least one primary output where the good circuit response is an X and the faulty circuit response is either a 1 or 0.

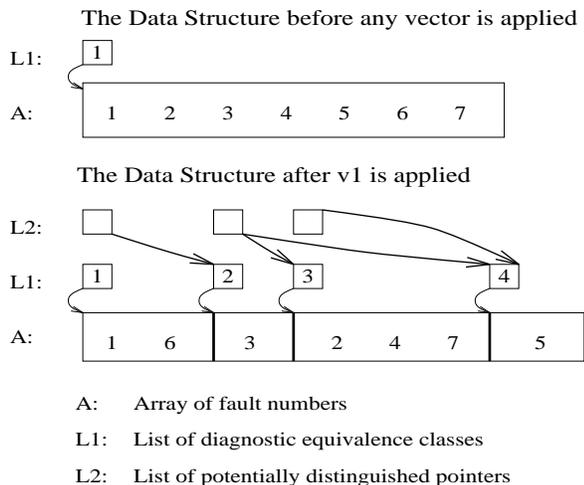


Figure 2: The Data Structure

Table 2: Diagnostic Fault Status

Test vect	f1	f2	f3	f4	f5	f6	f7
v1	D	P2	P1	P2	N	D	P2
v2	P1	D	D	D	N	P1	P1
D	Detected Faults						
P1	Potentially Detected Faults						
P2	Potentially Excludable Faults						
N	Strictly Undetected Faults						

If a potentially excludable fault has a response 1 on a primary output where the good circuit response is an X, and during diagnosis the actual binary response on that primary output is a 0 then the fault can be excluded from the list of candidate faults.

The fault status is categorized into four classes: *Detected Faults (D)*, *Potentially Detected Faults (P1)*, *Potentially Excludable Faults (P2)*, and *Strictly Undetected Faults (N)*. Strictly undetected faults are faults that have the same exact response as the good circuit on all primary outputs.

The diagnostic fault status is updated after each test vector is simulated. Table 2 shows the status of each fault at each test vector. It can be seen that faults f1 and f6 have status D for test vector v1 and status P1 for v2.

After each test vector is simulated the diagnostic fault status is used to speed up the creation of the diagnostic tree. For example, two faults, one with N status and the other with D status, are declared distinguished, without any additional comparisons of the faulty output responses. Also faults with P1 or P2 status are declared potentially distinguished from faults with N status. A complete set of rules for the usage of fault status is given in Section 2.3.

2.3 Creating the Diagnostic Tree

The diagnostic fault simulator generates each level of the diagnostic tree as a set of diagnostic equivalence classes and potentially distinguished pointers between these classes. This is then used to compute the indistinguishability classes. After the indistinguishability classes are obtained, all diagnostic measures [3] can be easily calculated. This is accomplished by simulating the test vectors, recording the primary output responses of each fault and then breaking-up the diagnostic equivalence classes into smaller classes. In the beginning there is only one diagnostic equivalence class (see Figure 2) and all faults belong to this class. After the first test vector is simulated, the output responses of each faulty machine are saved and the diagnostic fault status of each fault is recorded. Each diagnostic equivalence class is then broken up into three categories according to the diagnostic fault status of the faults in that class as shown below:

- *Group-D* : Includes detected faults (D)
- *Group-P* : Includes potentially detected faults (P1) and potentially excludable faults (P2)
- *Group-N* : Includes strictly undetected faults (N)

Next, the faults in Group-D and Group-P are sorted based on their output responses, and for each distinct output response a new class of faults is created.

The third step is the calculation of the potentially distinguished pointers. The following rules are used to reduce the comparison work done by the diagnostic fault simulator.

- By definition, there are no potentially distinguished pointers from each class in the Group-D category to the classes with Group-N category.
- Potentially distinguished pointers are created between each class in the Group-P category and every class in the Group-N category.
- The output responses of other possible combinations must be compared.
- If there is a potentially distinguished pointer between two classes then the children of these classes have to be examined with the above rules. This rule implies that if two equivalence classes are already distinguished, their output responses are never compared again.

Figure 3 shows the potentially distinguished pointers added using the fault status information and output response comparison. After vector v_1 is simulated the diagnostic equivalence class 1 (f_1, f_6) belongs to Group-D and class 4 (f_5) belongs to Group-N; therefore it is implied that there is no potentially distinguished pointer between class 1 and class 4. A potentially distinguished pointer between class 2 (f_3) and class 4 is created because they belong to Group-P and Group-N respectively. Similarly, a potentially distinguished pointer is added between class 3 and class 4. The response of diagnostic equivalence class 1 is compared with that of class 2 and class 3, and the response of class 2 is compared with that of class 3. Potentially distinguished pointers are created from class 1 to class 2, and from class 2 to class 3.

3 Computing Diagnostic Measures

3.1 Indistinguishability Classes

The indistinguishability relationship between faults can be represented as a graph, where the nodes correspond to diagnostic equivalence classes and the edges the indistinguishability relationship between these classes. Finding the indistinguishability classes now reduces to finding all cliques of maximal size [3].

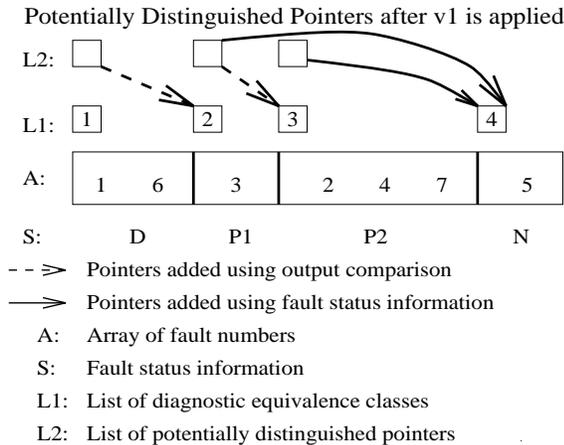


Figure 3: Potentially Distinguished Pointers

Since the number of maximal cliques in a graph can grow exponentially with the number of vertices, each maximal clique must be generated exactly once and all repetitive work must be avoided. We implemented a backtrack procedure [6] which recursively generates all maximal cliques while pruning the search tree to avoid unnecessary searching.

An example of the indistinguishability relationship graph is seen in Figure 4. There are two maximal cliques in the graph; the clique with size two forms the indistinguishability class (f_1, f_6, f_3), while the one with size three forms the indistinguishability class (f_3, f_2, f_4, f_7, f_5).

The diagnostic resolution (DR) is computed as

$$DR = 1 - \frac{(\# \text{ of indistinguishable fault pairs})}{(\text{Total } \# \text{ of fault pairs})} \quad (1)$$

The number of indistinguishable fault pairs are obtained from the diagnostic equivalence classes as

$$\sum_i (|C_i|) \cdot (|C_i| - 1) / 2 + \sum_{\forall (i,j), j > i, C_i \equiv C_j} (|C_i|) \cdot (|C_j|) \quad (2)$$

Here C_i 's refer to the diagnostic equivalence classes, $|C_i|$ refers to the number of faults in class C_i , and $C_i \equiv C_j$ implies that class C_i is potentially distinguished from class C_j . The diagnostic expectation (DE) is computed as

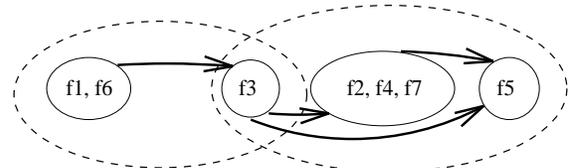
$$DE = \sum_{\forall \text{ faults } f} \frac{\text{size of } f \text{'s indistinguishability class}}{\text{total } \# \text{ of faults}} \quad (3)$$

3.2 Diagnostic Fault Dropping

The total running time of the diagnostic fault simulator can be improved by performing diagnostic fault dropping. Diagnostic fault dropping has been previously mentioned [3] [5] but not implemented. A fault can be dropped from the diagnostic fault simulation procedure after it is completely distinguished from other faults. Diagnostic fault dropping is accommodated in our list-based data structure by identifying fully distinguished faults and not injecting such faults in the fault simulation procedure.

3.3 Pessimistic versus Optimistic Measures

During simulation starting from an unknown state, if a fault f_1 has a response X and a fault f_2 has a response 1 (or 0) on the same output, then we cannot distinguish f_1 and f_2 with certainty. The pessimistic assumption has been that the two faults are indistinguishable [3]. Measures based on this assumption are *pessimistic measures*. As opposed to this, *optimistic diagnostic measures* make the assumption that the response X are considered to be distinguished from faults with a response 1 or 0. Since the response from a stuck-at fault in an



Diagnostic equivalence classes: (f_1, f_6), (f_3), (f_2, f_4, f_7), (f_5)

Indistinguishability classes: (f_1, f_6, f_3), (f_3, f_2, f_4, f_7, f_5)

Figure 4: Computing the Indistinguishability Classes

Table 3: Diagnostic Results for STG3 Test Vectors

Ckt.	Test vec.	Flts	Flt. cov.	Exec. Time (s)		DSIM Mem (KB)
				PROOFS Time †	DSIM Time	
s298	162	308	85.7	3.7	0.9	92
s344	91	342	96.2	2.5	1.1	100
s400	1282	426	82.9	46.2	9.5	104
s420	173	455	5.1	4.8	0.4	92
s526	754	555	75.3	40.7	5.3	108
s641	133	467	86.3	4.5	0.7	100
s713	107	581	80.9	4.8	0.8	136
s820	411	850	81.9	26.1	3.4	160
s832	377	870	81.4	24.6	3.1	160
s953	16	1079	7.8	1.4	0.2	88
s1238	349	1355	94.7	27.4	2.0	200
s1423	36	1515	24.4	4.3	0.5	528
s1488	590	1486	92.6	130.5	17.0	276
s1494	469	1506	91.1	102.6	17.0	292
s5378	408	4603	74.0	232.2	39.5	852
s35932	86	39094	88.0	1926.3	1505.9	16464

† w/o fault dropping

Table 4: Diagnostic Results for HITEC Test Vectors

Ckt.	Test vec.	Flts	Flt. cov.	Exec. Time (s)		DSIM Mem (KB)
				PROOFS Time †	DSIM Time	
s298	259	308	86.0	5.5	1.5	92
s344	108	342	96.2	3.1	1.1	96
s400	2069	426	82.6	74.4	13.4	96
s420	166	455	5.3	4.8	0.5	100
s526	192	555	19.8	9.2	0.8	104
s641	211	467	86.5	7.6	0.8	108
s713	175	581	81.9	7.2	0.9	120
s820	968	850	95.6	56.5	5.3	152
s832	967	870	93.8	56.7	5.7	152
s953	14	1079	8.2	1.3	0.2	88
s1238	478	1355	94.7	36.6	2.5	200
s1423	88	1515	36.6	10.5	0.7	168
s1488	1192	1486	97.2	247.0	22.2	244
s1494	1285	1506	96.5	269.7	23.8	248
s5378	900	4603	68.4	473.5	71.0	764
s35932	383	39094	89.2	5412.3	2346.7	13400

† w/o fault dropping

actual circuit is binary, i.e., a 1 or 0, the average diagnostic capability of a test set in actual circuits will be between the pessimistic and optimistic diagnostic measures.

The actual diagnostic measure will probabilistically be closer to the optimistic measure than the pessimistic measure in almost all cases since distinguishing two potentially distinguished faults only requires different binary values on one of the primary outputs on any test vector. Suppose two potentially distinguished faults f_1 and f_2 have responses of $XXXXX$ and 11111 . Assuming that 1 and 0 responses have an equal probability, the probability of an actual failing circuit with f having a 11111 response is $1/2^5$, which is far less than the probability of it having at least one 0 in its response.

4 Experimental Results

The diagnostic fault simulator consists of the diagnostic fault simulation procedure (DSIM) implemented on top of the PROOFS [7] sequential circuit fault simulator. After a test vector is simulated by PROOFS, and the output responses for the faulty circuits are obtained, the DSIM procedure updates the compact list data structure by comparing the primary output responses, creating new diagnostic equivalence classes, and updating the potentially distinguished pointers. After all test vectors are simulated, DSIM computes the indistinguishability classes. The diagnostic fault simulator was run on a SUN SPARCstation 2 with 64MB of memory for the IS-CAS89 benchmark circuits [8]. The diagnostic capabilities of two different test sets were evaluated. The first was generated by STG3 [9] and the second by HITEC [10].

The PROOFS execution time (without fault dropping), DSIM execution time, and memory requirements for the DSIM procedure are shown in Tables 3 and 4. This paper provides the first report on diagnostic fault simulation results for the complete fault list of s35932; previous papers reported results for 1/5 of the s35932 faults [3] [5]. It is seen that the execution time for the diagnostic procedure is much smaller than that of

the fault simulation time (without fault dropping) for most circuits.

The diagnostic measures are reported in Table 5. Both pessimistic and optimistic measures are reported. In some cases optimistic diagnostic expectations are as small as a third of the pessimistic ones. Table 6 gives the execution times with fault dropping for fully distinguished faults. The total diagnostic fault simulation time (PROOFS plus DSIM) is reduced when the diagnostic power is greater than 25%. In s1238 about 80% are fully distinguished and the total diagnostic fault simulation time is reduced by half. The memory requirements versus the number of simulated faults for the benchmark circuits are shown in Figure 5.

Table 7 compares our diagnostic simulation procedure (DSIM) with previous approaches [3] [5]. For most cases our diagnostic fault simulator is faster than the two previous approaches. We reran the program used by Rudnick, et al. [3] on a SUN SPARCstation 2 machine, and report the time for updating the distinguishability matrix and calculating the indistinguishability classes in the column labeled Prev1.

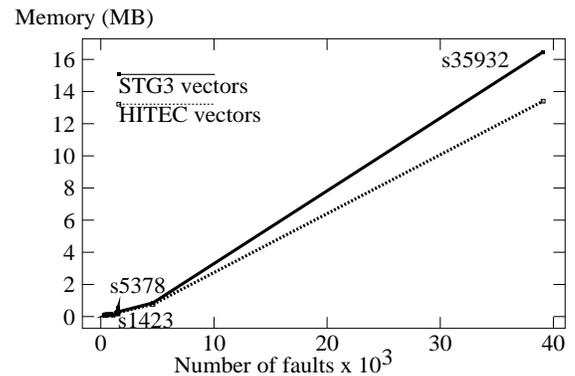


Figure 5: Memory Usage

Table 5: Diagnostic Measures for STG3 Test Vectors

Ckt.	Pessimistic			Optimistic		
	Diag. Pow.	Diag. Res.	Diag. Expt.	Diag. Pow.	Diag. Res.	Diag. Expt.
s298	0.0	94.2	18.8	45.5	98.4	6.0
s344	0.0	96.7	12.1	73.4	99.8	1.7
s400	0.0	92.4	33.4	42.5	96.6	15.3
s420	0.0	9.6	411.3	6.2	34.6	298.1
s526	0.0	89.6	58.7	29.4	93.6	36.2
s641	39.0	97.7	11.7	61.2	98.4	8.4
s713	31.0	95.9	24.9	51.8	97.2	17.4
s820	8.7	96.4	31.9	63.5	96.8	28.4
s832	8.5	96.3	32.9	63.4	96.5	31.1
s953	1.4	14.9	918.1	4.9	27.8	779.7
s1238	81.3	99.7	4.9	81.8	99.7	4.9
s1423	1.8	42.5	870.9	7.0	52.2	724.3
s1488	4.0	99.1	14.1	81.4	99.5	8.8
s1494	3.8	98.7	20.9	79.9	99.3	12.1
s5378	25.7	93.2	316.1	46.3	94.9	234.3
s35932	0.0	98.5	569.0	28.5	98.6	560.8

5 Summary

A diagnostic fault simulator has been described that represents indistinguishable classes of stuck-at faults as memory efficient lists. This representation is different from previous work, in that it does not explicitly store indistinguishability relationships between pairs of faults as a matrix [3] or as lists [5]. Experimental results show that the method is faster than previous proposals [3] [5]. The paper provides the first reports on pessimistic and optimistic diagnostic measures for all faults of the large ISCAS circuits, results on diagnostic fault dropping and diagnosis for faults declared to be undetected.

References

- [1] P. Camurati et al. , “A Diagnostic Test Pattern Generation Algorithm,” in *Proc. IEEE Intl. Test Conf.*, pp. 52–58, Sep. 1990.
- [2] K. Kubiak, S. Parkes, W. K. Fuchs, and R. Saleh, “Exact Evaluation of Diagnostic Test Resolution,” in *Proceedings of the Design Automation Conference*, pp. 347–352, June 1992.
- [3] E. M. Rudnick, W. K. Fuchs, and J. H. Patel, “Diagnostic Fault Simulation of Sequential Circuits,” in *Proc. IEEE Intl. Test Conf.*, pp. 178–186, Oct. 1992.
- [4] P. G. Ryan, W. K. Fuchs, and I. Pomeranz, “Fault Dictionary Compression and Equivalence Class Computation for Sequential Circuits,” in *Proc. IEEE/ACM Intl. Conf. on CAD*, pp. 508–511, Nov. 1993.
- [5] J. M. Jou and S.-C. Chen, “A Fast and Memory-Efficient Diagnostic Fault Simulation for Sequential Circuits,” in *Proc. IEEE/ACM Intl. Conf. on CAD*, pp. 723–726, Nov. 1994.
- [6] E. M. Reingold, J. Nievergelt, and N. Deo, *Combinatorial Algorithms*. Prentice-Hall, Inc., pp. 353–359, 1977.
- [7] T. M. Niermann, W. T. Cheng, and J. H. Patel, “PROOFS : A Fast, Memory-Efficient Sequential Circuit Fault Simulator,” in *IEEE Trans. on CAD*, pp. 198–207, Feb. 1992.

Table 6: Diagnostic Fault Dropping (STG3 Vectors)

Ckt.	Total Execution time (sec.)		
	PROOFS w/ fault dropping	DSIM † plus PROOFS	DSIM ‡ plus PROOFS
s298	0.8	4.7	4.5
s344	0.6	3.6	3.6
s400	9.8	55.7	56.7
s420	4.7	5.3	5.2
s526	8.8	46.0	48.9
s641	1.0	5.2	3.9
s713	1.2	5.6	4.6
s820	4.0	29.5	29.6
s832	3.9	27.7	28.1
s953	1.7	1.6	1.7
s1238	3.6	29.4	14.6
s1423	3.2	4.7	5.0
s1488	9.8	147.5	148.2
s1494	8.2	119.6	118.5
s5378	43.8	271.7	218.8
s35932	163.5	3432.3	3451.4

† w/o fault dropping

‡ w/ diagnostic fault dropping

- [8] F. Brglez, D. Bryan, and K. Kozminski, “Combinational Profiles of Sequential Benchmark Circuits,” *IEEE Intl. Symp. on Circuits & Systems*, pp. 1929–1934, May 1989.
- [9] W. T. Cheng and S. Davidson, “Sequential Circuit Test Generator STG Benchmark Results,” *IEEE Intl. Symp. on Circuits & Systems*, pp. 1938–1941, May 1989.
- [10] T. Niermann and J. H. Patel, “HITEC : A Test Generation Package for Sequential Circuits,” in *Proc. Euro. Design Automation Conference*, pp. 214–218, Feb. 1991.

Table 7: Diagnostic Execution Time Comparison

Ckt	Test vec.	Execution time(seconds)		
		Prev1 [3]	Prev2 [5]	DSIM ‡
s298	162	2.1	1.6	0.9
s344	91	2.4	1.5	1.1
s400	1282	25.4	24.5	9.5
s420	173	0.3	0.3	0.4
s526	754	24.4	21.2	5.3
s641	133	12.2	0.6	0.7
s713	107	14.3	0.9	0.8
s820	411	91.8	4.0	3.4
s832	377	83.7	3.6	3.1
s953	16	1.3	0.2	0.2
s1238	349	99.9	1.9	2.0
s1423	36	6.1	1.7	0.5
s1488	590	392.2	19.5	17.0
s1494	469	332.2	24.7	17.0
s5378	408	3575.4	40.5	39.5
s35932	86	– †	– †	1505.9

† results reported for 1/5 of the faults

‡ w/o diagnostic fault dropping