MODELING THE UNKNOWN!
TOWARDS MODEL-INDEPENDENT FAULT AND ERROR DIAGNOSIS

Vamsi Boppana and *Masahiro Fujita*
Fujitsu Laboratories of America, Inc.
595 Lawrence Expressway
Sunnyvale, CA 94086
{vboppana,fujita}@fla.fujitsu.com

Abstract

In this paper, we provide techniques for fault and error diagnosis based on capturing unmodeled faulty behavior. We present a technique for capturing the effects of **all** possible faulty behaviors that can be generated from specific sets of nodes (called **X-lists** in this paper) in the circuit. Since all possible erroneous behaviors are captured, this provides a way for drawing powerful diagnostic inferences about the presence of faults *at* these sets of nodes when analyzing the observed faulty responses. We also present an efficient diagnosis algorithm that exploits the modeling of all possible behaviors and can be built in a framework of conventional test and simulation tools. Experimental results with numerous diagnosis experiments are then used to demonstrate that the techniques developed can indeed be used to achieve significant improvements in the accuracy of diagnosis.

1 Introduction

Fault diagnosis refers to the process of locating failures in integrated circuits. Accurate fault diagnosis is an integral step in any failure analysis procedure. The results of fault diagnosis are used to direct the failure analysis process to a specific part of the chip to determine the cause for failure.

Error diagnosis, on the other hand, is a similar problem, but occurs at an earlier stage in the design flow. It is typically invoked after a failed design verification step to determine possible causes for the failure. In this paper, we provide a technique for the location of design errors as well as manufacturing faults. The techniques developed are applicable to both problems because the primary contribution of the paper is a method to capture all the possible behaviors at specific nodes in the circuit. We shall develop the concepts in this paper primarily discussing the fault diagnosis application, but will use the terms fault and error interchangeably throughout the paper.

Fault diagnostic techniques can be broadly classified into three groups. The first group, called static fault diagnosis, uses precomputed information in the form of fault dictionaries for matching with the faulty responses produced by defective circuits [1–11]. In contrast, dynamic diagnosis techniques diagnose the faulty behavior of the circuit while the test set is applied [1, 12–20]. Recent trends show the increasing popularity of integrated diagnosis techniques where the focus is on using small amounts of precomputed information and coupling this with efficient dynamic algorithms to perform fault location [8, 21–23]. Orthogonal to the above classification is the body of work that has targeted at improving the accuracy of diagnosis by either strengthening the matching algorithm with a simple fault model or by the use of a more accurate fault model and a comparatively simple matching algorithm [11, 24–30]. Simple models, while being easy to construct and easy to apply to complex diagnoses may not result in the necessary diagnostic accuracy. More complex models may provide accurate diagnosis, but it is achieved only at the cost of the actual defect being of the targeted type.

It is with this background that we present the essential contrasting feature of our work. In our work, we ask and answer the question: *Can the effects of all possible faults at any given subset of nodes in the circuit be captured and effectively used in diagnosis?*
Indeed, the answer leads us to the contributions of this work. This paper presents a technique for capturing the effects of all possible faulty behaviors that can be generated from specific sets of nodes (called X-lists, and defined later in this paper) in the circuit. Since all pos-

0-7803-5092-8/98 $10.00 © 1998 IEEE

sible erroneous behaviors are captured, this provides a way for drawing powerful diagnostic inferences about the presence of faults at these sets of nodes when analyzing the responses produced by the defective chip. Earlier work that has exploited the abstractness of X-values includes the identification of subsets of undetectable faults in a circuit [31] and the generation of diagnostic test vectors to distinguish gates based on their ability to propagate an X-value [32].

An important benefit of our technique is that even while it retains the power of non-enumerative dynamic techniques, it can be easily incorporated into an efficient cause-effect technique. In fact, we shall demonstrate an implementation that exploits this feature of the technique. It is also worth noting that since our method has the capability of capturing all the faults at the X-lists, there is no necessity to handle any additional fault models or simulation methodologies explicitly. In other words, these are all covered by our technique. Our experiments performed on the ISCAS 85 benchmark circuits include the introduction of a large number of unmodeled errors and demonstrating that a simple application of the ideas introduced in this paper into a diagnosis algorithm is sufficient to produce significant improvements in the accuracy of the resulting diagnoses.

Important contributions of this work are listed below.

- It provides a simple, systematic method to capture the effects of all faults/errors at specific X-lists in the circuit. This leads to effective diagnosis conclusions at the X-list-nodes when analyzing observed behavior.

- An efficient implementation of a diagnosis algorithm, that exploits the ideas developed to capture unmodeled behavior at single nodes in the circuit, based on existing simulation technologies, is devised. Experiments using this algorithm and a large number of diagnoses demonstrate significant improvements in the accuracy of diagnosis.

2 Modeling the unknown

In order to motivate the need to capture the effects of different fault models, and understand how such a capture would aid diagnosis, we first consider a typical matching algorithm (similar to the ones used in [8, 18, 21, 29]). This algorithm is presented in Figure 1. The algorithm is based on stuck-at fault simulation and matching their effects with the observed responses to result in a list of candidate faults. For this algorithm, the parameters MATCHCOUNT, PARTIALMATCHCOUNT and MISMATCHPENALTY control the emphasis placed on the observation of a match, partial mismatch (because of the presence of unknowns) and mismatch with the actual observed responses. The result of this algorithm is a list of faults with a score associated with each fault. The faults may then be ranked in a decreasing order of scores to find the best candidates at the top of the sorted list.

Table 1: Primary output responses produced

Cause	POs(V1)	POs(V2)
Good ckt.	00	01
Erroneous ckt.	01	00
Gate 9's op stuck-0	01	01
Gate 9's op stuck-1	00	00
Gate 9's op **unknown**	0X	0X

2.1 Example

We shall now consider the diagnosis of a fault from an example circuit and motivate the necessity for capturing unmodeled effects. Consider the example circuit shown in Figure 2 (this is the circuit c17 from the ISCAS 85 benchmark circuits). We show two versions of this circuit in this figure. The top circuit is the good circuit and the bottom version of the circuit has an error (the discussion remains invariant of the fact that this could have been a manufacturing fault). The gate numbers are shown in square boxes on the two circuits. The error in question is caused by the fact that gate number 9 has been changed from the original NAND gate to an AND gate.

Let us assume further that the diagnosis is being performed with 2 previously generated vectors. The responses of each node in the two circuits to these two vectors are also indicated at the corresponding node. We also note the responses produced by the good circuit, the faulty circuit, the good circuit under the influence of gate 9's output stuck-at-0, the good circuit under the influence of gate 9's output stuck-at-1 and those of the good circuit under the influence of gate 9's output being *unknown (X)* as shown in Table 1.

Since there are mismatches produced by both the faults at node 9, chances are reduced for the diagnosis to contain a fault at node 9 at the top of the candidate faults. It is immediately obvious that the problem occurs because all possible behaviors at gate 9 are not handled by either of the stuck-at faults, (9-o/p-st-0) or (9-o/p-st-1). However, it is obvious that if there was a method to capture *all* the effects of all possible faults at the node 9, then the diagnosis would have been more accurate (because the fault effects can clearly be captured at node 9).

This is exactly what would be covered by the introduction of an unknown value (X) at gate 9 and observing the responses produced by the good circuit under the influence of this uncertainty. The output responses produced by such a simulation would result in the primary output combinations $0X$, $0X$ and they would *cover* the observed faulty primary output responses.

We also note that other kinds of inferences can be drawn using the X-values. Let us now consider the following statement: Can **any** error at node 6 have caused the observed behavior 01, 00? We can answer this question accurately by performing a simulation with the output of node 6 set to the value X and noticing that the expected outputs under any error or fault at node 6 would be of the form $X0$, $X1$ and hence *could not* have caused the observed behavior.

We can now devise a diagnostic algorithm around the above observations that can exploit the fact that, while none of the stuck-at faults can explain the observed behavior, introduction of *X-valued nodes* (X-lists of size 1) can indeed be used to diagnose the behavior. Such algorithms will be devised in the next section.

It is also easy to observe that this analysis need not be restricted to single nodes where the X-values may be introduced. We can deal with faults involving multiple nodes and more complex fault types by increasing the cardinality of each individual X-list. Of course, a diagnosis algorithm that forms X-lists with arbitrary combinations of nodes would not scale well for large circuits. However, this problem can be circumvented by observing that typical diagnosis applications require handling only small subsets of all combinations of circuit nodes and devising X-lists to cover such sets of nodes.

3 Diagnosis with X-lists

3.1 X-lists

Definition 1 (X-list S**)** *A set of nodes whose good values would be replaced during simulation by the value X, to capture the effects of any fault completely contained in any* **dominance region** *of this set of nodes, is called an X-list.*

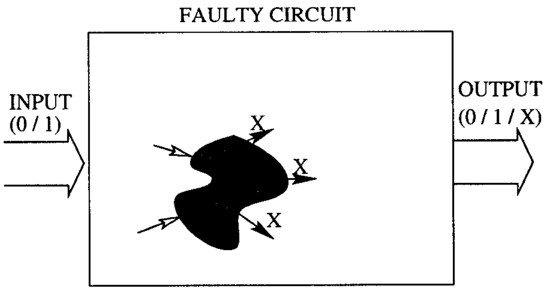

Figure 3: Model of the unknown!

Definition 2 (Dominance region(S**))** *Any set of nodes in the circuit such that every path from that set of nodes to any primary output must pass through a node in the set S can constitute a dominance region for S.*

An illustration of an X-list and its dominance region is presented in Figure 3. We now state the result that represents the diagnostic reasoning available from the use of X-lists.

Theorem 1 *If there exists a test vector V and a primary output p such that the response of the good circuit under the influence of an X-list S and the observed responses are 0(1) and 1(0) respectively, then the faulty design does not have any fault that is completely contained in any dominance region of S.*

Proof: The proof of the result follows from the fact that if there was a fault within a dominance region of S, then it should have produced a combination of signal values on the nodes in the set S that would have been completely covered by the introduction of X-values. Hence, there would be no primary output that would assume a value different from the observed output values.

The above result clearly demonstrates the power of X-lists because even without making any assumption on the nature of the fault, we are able to achieve useful diagnostic results (including multiple fault diagnosis, even with single node X-lists!). In addition, we now show the implementation of an efficient simulation-based diagnosis algorithm around the concepts developed.

3.2 Diagnosis algorithm

We shall now discuss the development of an efficient algorithm based on the concept of X-lists. The aim is to develop simulation algorithms that incorporate X-lists and efficiently use the existing suite of simulation tools.

We wish to use stuck-at fault simulation in order to provide an efficient X-list simulator. While it is not difficult to convert an existing fault simulator into an X-list simulator handling X-lists of arbitrary cardinality, we make a practical restriction (in this work) and limit our discussion to X-lists of cardinality 1 for the purpose of demonstrating the capability of X-lists.

Under this restriction, it is very efficient to implement an X-list simulation algorithm based on stuck-at fault simulation. In fact, it is easy to see that it has the the same complexity as that of standard stuck-at fault simulation. The new simulator (which we call the *s-a-x* simulator) can be used to develop a matching-oriented diagnosis algorithm exactly like the one described in Figure 1. In the new algorithm, each fault now is of the form *Node-st-x* and the routine *Response(f, CurrVec, CurrPo)* in the algorithm now utilizes the new simulator to evaluate the response of the circuit under the condition that *Node* is assigned the value X. An important difference is that since unmodeled faults can be covered by this approach, it is possible to drop X-lists as soon as a single mismatch is discovered (only if we are sure that the error is at a single node). The effect of dropping is equivalent to running the algorithm with a MISMATCHPENALTY of ∞. This feature typically leads to a significant increase in the speed of the diagnosis algorithm.

We now demonstrate the improvements achievable by the simple diagnosis algorithm that exploits the concept of X-lists by performing several diagnosis experiments.

4 Experimental results and analysis

We evaluate the performance of our new diagnosis algorithm in this section. Experiments were performed on the ISCAS 85 benchmark circuits. We first describe the experimental setup. The s-a-x simulator and the diagnosis algorithms were implemented on top of the RAPSIM diagnostic fault simulator [19]. Errors were introduced in the good circuits by randomly changing the type of a gate in the circuit (example NAND to OR, AND to OR etc.). Then, the faulty responses were generated by logic simulation and used as the observed erroneous responses. The diagnosis algorithm using the s-a-x simulator was compared against the conventional diagnosis algorithm, as described in Figure 1. Diagnosis experiments were repeated for 25 different errors for each of the benchmark circuits considered. The diagnosis results were analyzed to find out the rank of the node in the sorted list of candidate faults provided by the diagnosis. We also note that the experiments were performed on a SPARCstation20 workstation with 512MB of main memory. The test vectors used in the diagnosis were adapted from the single stuck-at vectors generated by the HITEC test generator [33] available from the University of Illinois. We also note that the introduction of errors is accompanied by checking whether the error was detected. If the error was not detected, then it was immediately discarded and an alternative error was generated. This was done because test generation to help the diagnosis was not considered in this work.

It is first shown that while neither of the stuck-at faults at the exact node where the error is introduced is able to track the error, the new approach of using X-values produces remarkably accurate diagnoses. These results are summarized in Table 2. The columns represent for each of the circuits, average rank of the stuck-at-0 fault at the erroneous node, average rank of the stuck-at-1 fault at the erroneous node, average rank produced by our diagnosis algorithm and the total number of uncollapsed stuck-at faults in the circuit, respectively. The averages are computed over the 25 diagnoses performed for each of the circuits. Hence, the lesser the rank for the faulty node, the better is the accuracy of the diagnosis. Therefore, it is clear from the data in this table that our new diagnosis algorithm is indeed effective. We also make the additional interesting observation that may be useful for future work in this area. We observed that many times one of the two faults at the faulty location performs significantly better than the other. However, consistent with our original premise, none of them consistently dominates the other. This is really the reason that we have the improvements seen in this table. Another interesting result is the surprisingly large average rank produced for the circuit c1908. Upon closer analysis, it is actually revealed that there are instances of errors (3 in number) in this circuit that all the stuck-at-0, stuck-at-1 and the stuck-at-x entries carry a similar, large, rank. This can perhaps be attributed to the fact that no test vectors were generated to explicitly target the errors and some errors was not very well resolved by the given test set. This however, does not prevent us from drawing our expected conclusion that the accuracy is indeed significantly better than a conventional diagnosis approach.

In order to further analyze the ranks associated with the s-a-0, s-a-1 and the s-a-x models at each node, we provide a distribution of the ranks for each of the 25 error diagnoses for each circuit. This data is presented in Table 3. The columns in the table stand, for each type of rank, the number of diagnoses where the candidate was in the range 0-5, 6-10, 11-20 and greater than 20, respectively. Again, it is clear that the new approach provides much better diagnostic accuracy. This is clear from the larger numbers of errors that were diagnosed to be in the top few diagnosis candidates with our approach.

5 Conclusions

In conclusion, we have provided techniques for fault and error diagnosis that are aimed at capturing unmodeled behavior. We have shown that by the use of X-lists, it is indeed possible to model all unknown behaviors at any set of nodes in the circuit. We have also presented an efficient diagnosis algorithm based on restricting the cardinality of the X-lists to one and by utilizing existing, efficient, simulation tools. Experimental results demonstrated that the techniques can indeed be used to achieve remarkable improvements in the accuracy of the diagnosis. We believe that our techniques would be extremely useful to any diagnosis application where the types of defects or errors may not have been well studied or well understood.

References

[1] M. Abramovici, M. A. Breuer, and A. D. Friedman, *Digital System Testing and Testable Design*, New York, NY: Computer Science Press, 1990.

[2] R. E. Tulloss, "Size optimization of fault dictionaries", in *Proc. Intl. Test Conf.*, Nov. 1978, pp. 264–265.

[3] R. E. Tulloss, "Fault dictionary compression: Recognizing when a fault may be unambiguously represented by a single failure detection", in *Proc. Intl. Test Conf.*, Nov. 1980, pp. 368–370.

[4] J. Richman and K. R. Bowden, "The modern fault dictionary", in *Proc. Intl. Test Conf.*, Sept. 1985, pp. 696–702.

[5] V. Ratford and P. Keating, "Integrating guided probe and fault dictionary: An enhanced diagnostic approach", in *Proc. Intl. Test Conf.*, Sept. 1986, pp. 304–311.

[6] I. Pomeranz and S. M. Reddy, "On the generation of small dictionaries for fault location", in *Proc. Intl. Conf. Computer-Aided Design*, Nov. 1992, pp. 272–279.

[7] P. G. Ryan, W. K. Fuchs, and I. Pomeranz, "Fault dictionary compression and equivalence class computation for sequential circuits", in *Proc. Intl. Conf. Computer-Aided Design*, Nov. 1993, pp. 508–511.

[8] P. G. Ryan, "Compressed and dynamic fault dictionaries for fault isolation", Tech. Rep. UILU-ENG-94-2234, Center for Reliable and High-Performance Computing, University of Illinois at Urbana-Champaign, Sept. 1994.

[9] V. Boppana and W. K. Fuchs, "Fault dictionary compaction by output sequence removal", in *Proc. Intl. Conf. Computer-Aided Design*, Nov. 1994, pp. 576–579.

[10] V. Boppana, I. Hartanto, and W. K. Fuchs, "Full fault dictionary storage based on labeled tree encoding", in *Proc. VLSI Test Symp.*, Apr. 1996, pp. 174–179.

[11] D. B. Lavo, T. Larrabee, and B. Chess, "Beyond the byzantine generals: Unexpected behavior and bridging fault diagnosis", in *Proc. Intl. Test Conf.*, Oct. 1996, pp. 611–619.

[12] Miron Abramovici and M. A. Breuer, "Multiple Fault Diagnosis in Combinational Circuits Based on Effect-Cause Analysis", *IEEE Trans. Computers*, vol. C-29, no. 6, pp. 451–460, June 1980.

[13] Miron Abramovici and M. A. Breuer, "Fault Diagnosis in Synchronous Sequential Circuits Based on Effect-Cause Analysis", *IEEE Trans. Computers*, vol. C-31, no. 12, pp. 1165–1172, Dec. 1982.

[14] H. Cox and J. Rajski, "A method of fault analysis for test generation and fault diagnosis", *IEEE Trans. Computer-Aided Design*, vol. 7, no. 7, pp. 813–833, July 1988.

[15] J. A. Waicukauski, E. B. Eichelberger, D. O. Forlenza, E. Lindbloom, and T. McCarthy, "Fault simulation for structured VLSI", *VLSI Systems Design*, vol. VI, no. 12, pp. 20–32, Dec. 1985.

[16] J. A. Waicukauski and E. Lindbloom, "Failure diagnosis of structured VLSI", *IEEE Design & Test of Computers*, pp. 49–60, Aug. 1989.

[17] M. Abramovici and M. A. Breuer, "Fault diagnosis based on effect-cause analysis", in *Proc. Design Automation Conf.*, June 1987, pp. 69–76.

[18] E. M. Rudnick, W. K. Fuchs, and J. H. Patel, "Diagnostic fault simulation of sequential circuits", in *Proc. Intl. Test Conf.*, Oct. 1992, pp. 178–186.

[19] S. Venkataraman, I. Hartanto, W. K. Fuchs, E. M. Rudnick, S. Chakravarty, and J. H. Patel, "Rapid diagnostic fault simulation of stuck-at faults in sequential circuits using compact lists", in *Proc. Design Automation Conf.*, June 1995, pp. 133–138.

[20] S. Venkataraman, I. Hartanto, and W. K. Fuchs, "Dynamic diagnosis of sequential circuits based on stuck-at faults", in *Proc. VLSI Test Symp.*, Apr. 1996, pp. 198–203.

[21] P. Ryan, S. Rawat, and W. K. Fuchs, "Two-stage fault location", in *Proc. Intl. Test Conf.*, Oct. 1991, pp. 963–968.

[22] V. Boppana and W. K. Fuchs, "Integrated fault diagnosis targeting reduced simulation", in *Proc. Intl. Conf. Computer-Aided Design*, Nov. 1996, pp. 267–271.

[23] V. Boppana, "State information-based solutions for sequential circuit diagnosis and testing", Tech. Rep. CRHC-97-20, Ph.D. thesis, Center for Reliable and High-Performance Computing, University of Illinois at Urbana-Champaign, July 1997.

[24] Steven D. Millman, Edward J. McCluskey, and John M. Acken, "Diagnosing CMOS Bridging Faults with Stuck-at Fault Dictionaries", in *Proc. Intl. Test Conf.*, Oct. 1990, pp. 860–870.

[25] Sreejit Chakravarty and Minsheng Liu, "Algorithms for Current Monitoring Based Diagnosis of Bridging and Leakage Faults", in *Proc. Design Automation Conf.*, June 1992, pp. 353–356.

[26] Sreejit Chakravarty and Yiming Gong, "An Algorithm for Diagnosing Two-Line Bridging Faults in CMOS Combinational Circuits", in *Proc. Design Automation Conf.*, June 1993, pp. 520–524.

[27] Brian Chess, David B. Lavo, F. J. Ferguson, and Tracy Larrabee, "Diagnosing of Realistic Bridging Faults with Stuck-at Information", in *Proc. Intl. Conf. Computer-Aided Design*, Nov. 1995, pp. 185–192.

[28] Robert C. Aitken, "Finding Defects with Fault Models", in *Proc. Intl. Test Conf.*, Oct. 1995, pp. 498–505.

[29] K. De and Arun Gunda, "Failure Analysis for Full-Scan Circuits", in *Proc. Intl. Test Conf.*, Oct. 1995, pp. 636–645.

[30] R. C. Aitken and P. C. Maxwell, "Better models or better algorithms? Techniques to improve fault diagnosis", *Hewlett-Packard Journal*, pp. 110–116, Feb. 1995.

[31] S. B. Akers, B. Krishnamurthy, S. Park, and A. Swaminathan, "Why is less information from logic simulation more useful in fault simulation?", in *Proc. Intl. Test Conf.*, Sept. 1990, pp. 786–800.

[32] S.-Y. Kuo, "Locating logic design errors via test generation and don't-care propagation", in *Proc. European Design Automation Conf.*, 1992, pp. 466–471.

[33] T. Niermann and J. H. Patel, "HITEC: A test generation package for sequential circuits", in *Proc. European Design Automation Conf.*, Feb. 1991, pp. 214–218.

```
// Each f is assumed here to be of the form
// Node-st-0 or Node-st-1
// MATCHCOUNT, PARTIALMATCHCOUNT and MISMATCHPENALTY are input parameters
while ((CurrVec = GetNextVector) != NULL)
  while ((CurrPo = NextPo) != NULL)
    for each fault f do
      if (Match (Response(f, CurrVec, CurrPo), FaultyResponse(CurrVec, CurrPo) == MATCH))
        Score[f] += MATCHCOUNT;
      elseif (Match (Response(f, CurrVec, CurrPo), FaultyResponse(CurrVec, CurrPo) == PARTIALMATCH))
        Score[f] += PARTIALMATCHCOUNT;
      else // MISMATCH
        Score[f] -= MISMATCHPENALTY;
      endif
    endfor
  endwhile
endwhile
end
```

Figure 1: Matching algorithm

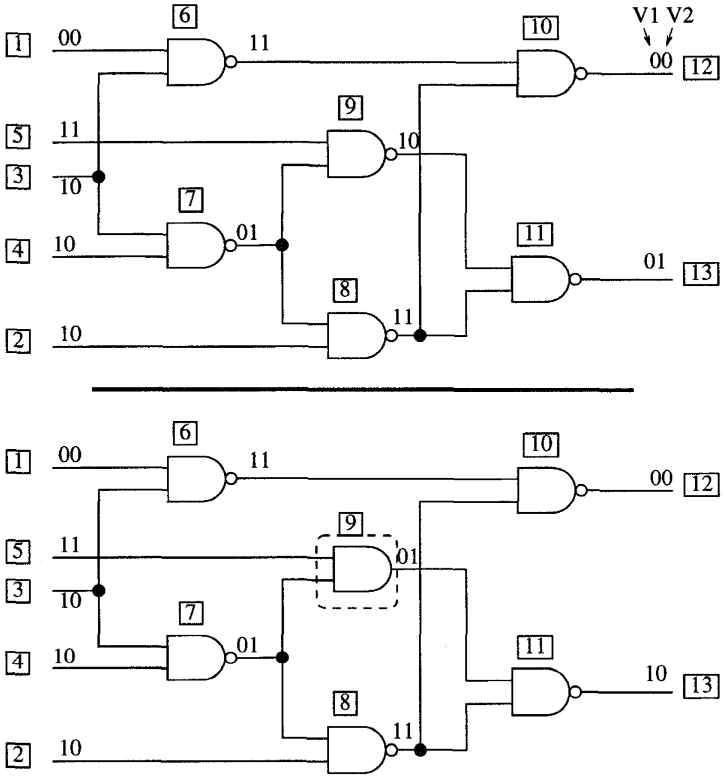

Figure 2: Example circuit (c17) - good and faulty versions along with values taken at each gate on the application of two vectors

Table 2: Diagnosis results - Rank of the faulty node

Ckt.	s-a-0Rank	s-a-1Rank	s-a-xRank	Faults
c432	245.20	366.44	2.20	1452
c499	137.72	368.08	7.72	1430
c880	222.16	344.12	3.44	2448
c1355	833.52	1005.92	53.00	3430
c1908	1661.04	854.56	287.28	4920
c2670	1436.88	973.36	12.72	7108
c3540	1304.76	869.40	4.16	9398
c5315	1573.76	2526.52	7.16	14234
c6288	2206.16	2100.04	2.00	14624
c7552	1314.20	3490.52	3.88	20158

Table 3: Diagnosis rank distribution

Circuit	s-a-0Rank				s-a-1Rank				s-a-xRank			
	< 5	< 10	< 20	> 20	< 5	< 10	< 20	> 20	< 5	< 10	< 20	> 20
c432	11	3	4	7	7	0	2	16	25	0	0	0
c499	6	0	4	15	12	0	0	13	7	12	6	0
c880	7	1	1	16	5	2	3	15	21	3	1	0
c1355	12	4	0	9	8	1	1	15	12	9	2	2
c1908	6	2	0	17	6	2	1	16	6	8	8	3
c2670	6	1	1	17	5	3	1	16	11	8	4	2
c3540	5	1	2	17	9	6	3	7	19	4	2	0
c5315	7	0	7	11	8	6	1	10	18	3	2	2
c6288	5	0	2	18	16	1	1	7	25	0	0	0
c7552	8	5	0	12	6	1	5	13	21	2	1	1