Speeding up Effect-Cause Defect Diagnosis using a Small Dictionary

Wei Zou* Wu-Tung Cheng* Sudhakar M. Reddy[+] Huaxing Tang*

*Mentor Graphics Corporation
8005 SW Boeckman Road
Wilsonville, OR 97070, U.S.A

[+]ECE Dept.
University of Iowa
Iowa City, IA 52242, U.S.A

Abstract

In this paper we present a new technique to speed up the effect-cause defect diagnosis by using a dictionary of very small size. In the proposed method, a dictionary of small size is used to reduce the number of events (gate evaluations) during the simulation of failing patterns and also a procedure to select a subset of passing patterns for simulation. Although the dictionary size is smaller, experimental results show speed up of effect-cause diagnosis by up to 156X. Experimental results from industrial designs validate the effectiveness of the proposed method.

1. Introduction

As manufacturers go into volume production with 90nm and smaller feature size designs, the dominant defect type has changed from the random particle defect to the design-specific systematic defect. In order to identify the design-specific systematic defect, a large number of failing dies need to be diagnosed in a short time. Therefore, a defect diagnosis tool with high accuracy and throughput becomes very important in the initial yield ramp.

Generally speaking, defect diagnosis methods can be classified into two types: cause-effect diagnosis and effect-cause diagnosis. Cause–effect diagnosis, also called dictionary based diagnosis, pre-computes and stores the faulty responses of modeled faults in a dictionary. In the process of diagnosis, the observed failure responses are compared with the pre-computed failure responses in the dictionary. The faults whose pre-computed failure responses have the closest match with the observed failure responses will be chosen as final candidates. Since dictionary based diagnosis does not perform fault simulations during diagnosis, the speed of diagnosis is very high. However, dictionary based diagnosis needs a very large memory to store the pre-computed failure responses. Although a number of techniques [1, 2, 3] are proposed to reduce the memory size, the size of the reduced dictionary is still too large for current designs with millions of gates. Moreover, it may lose diagnosis accuracy due to information loss when dictionaries of reduced size such as pass/fail dictionaries [1] are used. Instead of simulating faults upfront, using effect-cause diagnosis [4] one only simulates the potential fault candidates obtained by back tracing from failing outputs. Compared to cause-effect diagnosis, effect-cause diagnosis [4-10] does not need a large memory to store pre-computed faulty responses and it can provide very high diagnosis accuracy. However, the run time of the effect-cause diagnosis to diagnose a failing chip is long due to a larger number of fault simulations used during diagnosis. Recently a method was proposed to reduce run time of effect-cause diagnosis procedures by reducing the numbers of faults simulated during diagnosis [11]. In this paper, we propose a new method to speed up effect-cause diagnosis by using a small sized dictionary. Compared to the traditional fault dictionaries, the dictionary used in the proposed method has a much smaller size but it helps speed up effect-cause diagnosis significantly without losing diagnosis accuracy.

The rest of the paper is organized as follows. In Section 2, we discuss the run time intensive parts of the current effect-cause defect diagnosis procedures. In Section 3 we describe the method proposed to speed up diagnosis procedures. In Section 4 we give experimental results on industrial designs. Section 5 concludes the paper.

2. Run time bottlenecks of effect-cause defect diagnosis

In this section, we discuss the run time bottlenecks of current effect-cause defect diagnosis procedures. Generally speaking, effect-cause defect diagnosis procedures have two phases which are shown in Figure 1[6]. In the first phase a set of candidate fault sites are determined from the failing patterns **G** and in the

25th IEEE VLSI Test Symmposium (VTS'07)
0-7695-2812-0/07 $20.00 © 2007 IEEE

second phase the candidate sites are ranked by fault simulating the candidates using passing patterns. In the first phase of effect-cause diagnosis, for each failing pattern **p**, back tracing is used to find a subset **Q** of the set of faults which can potentially explain this failing pattern. Next the faults in **Q** for failing pattern **p** are simulated to find a subset **Q'** of **Q** which can actually explain this failing pattern. A failing pattern **p** is said to be explained by a fault, say **f**, if the circuit outputs with the fault **f** injected are the same as the outputs observed on the tester when pattern **p** is applied. After all the failing patterns in **G** are analyzed, a minimum set covering algorithm is used to find a subset **S**, of minimum size, of the set of faults which can explain all the failing patterns. The faults in **S** are the final defect candidates. In the second phase of effect-cause diagnosis, the fault candidates in **S** are simulated over all the passing patterns to find the number of passing pattern mismatches for each candidate. A passing pattern mismatch or passing mismatch for short is said to occur whenever a candidate fault is detected by a passing pattern. The number of passing mismatches combined with other information such as failing pattern matches is used to calculate a score for each candidate. Next the defect candidates are ranked in descending order of their scores.

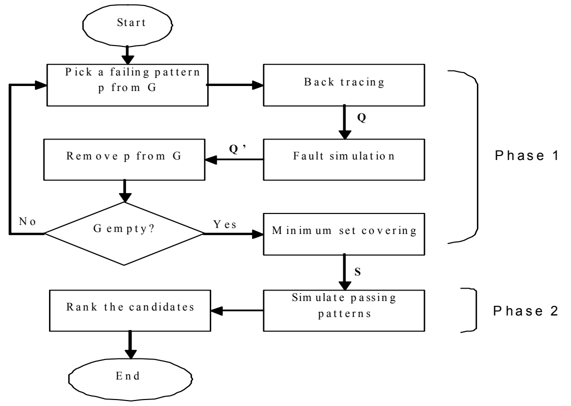

Figure 1: Effect-cause defect diagnosis flow

There are two run time intensive steps in the effect-cause diagnosis procedure described above. The first one is the time for back tracing and fault simulation time in the first phase for faults in **Q** to obtain the faults in **Q'**. This is due to the fact that back tracing procedures typically use a version of critical path tracing and we observed that the identified faults trigger a large number of events during fault simulation. An event is a gate which needs to be evaluated due to a change in its input values during fault simulation. The second bottleneck is the run time for the second phase for simulating faults in **S** using all passing patterns. Even though the number of faults in **S** is typically small the number of passing patterns is typically large.

In the method proposed in the next section we use a dictionary of small size to determine an initial list of candidates that explain a failing pattern thus avoiding back tracing. It is important to note that experimentally we determined that the faults in this list typically cause orders of magnitude fewer events than the initial set of faults **Q** obtained by back tracing. This is despite the fact that the final set of candidates obtained after fault simulation which explain a given failing pattern are identical for both the methods. An explanation for the observed reduction in events is given in Section 4 when we present experimental results. We also use information regarding clocks which is stored in the dictionary to select only a subset of the passing patterns for simulation in the second phase.

3. The proposed method

In this section we describe the proposed method to speed up effect-cause defect diagnosis. We describe the proposed diagnosis procedure in three sections. In Section 3.1 we give the structure of the proposed small dictionary. In Section 3.2 we discuss how to speed up the first phase of effect-cause diagnosis using the proposed small dictionary. In Section 3.3 we describe how to select a subset of passing patterns for simulation based on the information provided by the proposed small dictionary to speed up the second phase of effect-cause diagnosis.

3.1. Structure of the proposed small fault dictionary

A fault dictionary is a record of the errors that the modeled faults in the circuit are expected to cause. A complete dictionary is a full-response fault dictionary, which stores the values of all circuit outputs in the presence of each fault for each test pattern. The number of bits required to store a complete dictionary equals **F•V•O**, where **F** is the number of faults, **V** is the number of test patterns, and **O** is the number of outputs. For a large industrial design with several million gates, it is not practical to use a complete dictionary due to its large dictionary size. To reduce the dictionary size, a pass-fail dictionary only stores one bit of information (pass or fail) for each fault per test pattern. So the number of bits required for a pass-fail dictionary is **F•V**. Since information about the identity of the failing outputs is omitted in the pass-fail dictionary, it leads to lower diagnosis accuracy. Since a pure dictionary based fault diagnosis either requires a large memory to store the dictionary, or has lower diagnosis accuracy, we combine dictionary based diagnosis with effect-cause diagnosis. By doing this, we can speed up effect-cause fault diagnosis using a small size dictionary without losing any diagnosis accuracy.

Table 1: Dictionary sizes

Design	Ckt1	Ckt2	Ckt3	Ckt4
#gates	313K	506K	2025K	5.3M
#faults	631K	1210K	4172K	8701K
#patterns	5000	6951	1000	1000
#clks	11	36	30	5
#aus	17.6	14.0	10.0	17.0
#pfd	394M	1052M	522M	1100M
#sd	45M	73M	123M	600M

Next we describe the small sized dictionary we propose. The dictionary is created in a preprocessing step and used during diagnosis of a failing device. We propose a dictionary based on signatures. For each fault and a test pattern that detects this fault, a 32 bit signature is computed using a MISR into which the identifications of the errors in the circuit outputs are fed. The identification of each output error is represented by a 32 bit integer during MISR computation. For each fault we only store unique signatures. That is, if the fault has the same signature for two different failing patterns we only store one signature. Thus the unique signatures of a fault **f** under a test pattern set **T** are the union of the signatures of fault f for test patterns of T. For example, for a fault **f** and a test pattern set **T** = {**T1**, **T2**, **T3**, **T4**}, assume that a fault **f** is detected by test patterns **T1**, **T3** and **T4** with the failures observed at 3rd and 15th observation points for **T1** and **T3** and 7th and 10th observation points for **T4**. The 32 bit identifiers in hex format for 3rd, 15th, 7th and 10th observation points are **00000003**, **0000000F**, **00000007** and **0000000A**, respectively. We feed **00000003** and **0000000F** in order into a 32 bit MISR with the primitive polynomial $1+x+x^{29}+x^{31}+x^{32}$ with initial seed of all 0s to get the signature **000000E** for **T1** and **T3**, and feed **00000007** and **0000000A** into the MISR to get the signature **00000009** for **T4**. Now the signatures of fault **f** for **T1**, **T3** and **T4** are **0000000E**, **0000000E** and **00000009,** respectively. The unique signatures of fault **f** stored in the dictionary are {**0000000E**, **00000009**}. The number of bits to store unique signatures for all the faults is **32•U•F**, where **U** is the average number of unique signatures for each fault.

Besides unique signatures, we also store, for each fault, the clocks which are associated with the scan cells where the fault effect is observed. In the proposed small dictionary, each clock is assigned one bit to indicate whether it is associated with the scan cells where the effects of a fault are observed. The number of bits for storing clock information is **F•C**, where **F** is the number of faults and **C** is the number of clocks in the circuit. The clock information is used to speed up the second phase of effect-cause diagnosis. We will describe the details of this in Section 3.2

The total number of bits needed for the proposed dictionary is **32•F•U**+**F•C**. In Table 1, we list the number of gates (#gates), the number of faults (#faults), the number of clocks (#clks), the number of patterns (#patterns), the average number of unique signatures (#aus), and the sizes of pass-fail dictionary (#pfd) and the proposed dictionary (#sd) in bytes for four industrial designs used in our experiment. From Table 1, we can see that the size of the proposed dictionary is considerably smaller than that of the corresponding pass-fail dictionary.

3.2. Speeding up the first phase of effect-cause diagnosis using the small dictionary

In the first phase of the effect-cause diagnosis procedures, a backward path tracing from failing outputs is used to determine an initial set of candidate faults **Q**. Typically the size of **Q** is large and the faults in **Q** trigger a larger number of events in the fault simulation. So it takes a long time to simulate the faults in **Q** to reduce the fault candidates to set **Q'** which contains the faults that can actually explain the observed failing response. To reduce the run time of the first phase of effect-cause diagnosis, we propose to obtain the initial list of candidate faults using the dictionary proposed in Section 3.1 and not using backward path tracing. The new procedure for the first phase of the effect-cause diagnosis is shown in Figure 2. For each failing pattern **p**, instead of backward path tracing to find the candidates **Q**, we first compress the failure data with a 32 bit MISR to get a compressed signature **R** and then search the dictionary we created to find a set **H** of candidates whose unique compressed signatures contain **R**. The faults in **H** are potential candidates explaining the failing pattern **p**. Typically the set **H** is sometimes smaller and some times larger than the set **Q** but the number of events triggered by faults in **H** is typically much smaller than that triggered by faults in **Q**. We simulate the faults in **H** to find a subset of candidates **H'** of **H** which explain the failing pattern. It can be shown that **H'** is equal to **Q'** which is the set of faults that is obtained after fault simulating the faults in **Q** derived through back tracing. After all the failing patterns are analyzed, a minimum set covering algorithm is used to find a minimum subset **K** of faults which can explain all the failing patterns. The faults in **K** are the final list of candidates. It is important to note that the final candidates in **K** are exactly the same as the candidates in **S** obtained in the standard effect-cause diagnosis method.

The reasons for reduced run time of phase 1 using the proposed method are:

(1) Backward path tracing is not used to determine the initial set of candidates that could potentially explain failing patterns, and

(2) The numbers of events triggered by the faults in the initial set of candidates **H** is typically much smaller than that in the initial set of candidates **Q** obtained in the standard diagnosis procedures

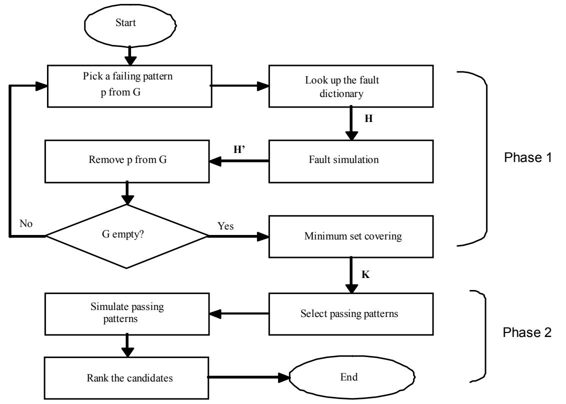

Figure 2: The proposed diagnosis flow

3.3. Speeding up the second phase of effect-cause diagnosis

In effect-cause diagnosis, the purpose of simulating passing patterns is to distinguish the defect candidates derived from the failing patterns in phase 1. For example, assume that we obtain two fault candidates **f1** and **f2** from phase 1 of the effect-cause diagnosis and there are four passing patterns **p1**, **p2**, **p3** and **p4**. Assume that when we simulate **f1** and **f2** over the passing patterns from **p1** to **p4**, we find that **f1** causes **p1** to fail and **f2** causes **p1** and **p3** to fail. So the number of passing pattern mismatches of **f1** and **f2** are 1 and 2, respectively. The number of passing pattern mismatches can be used to rank the fault candidates in a decreasing order or combine other information to give a score for each defect candidate. However, for a passing pattern say **p**, if all the fault candidates pass passing pattern **p** during fault simulation, then **p** contributes nothing to distinguish fault candidates. So the simulation of such a passing pattern **p** is not necessary. In the example, fault candidates **f1** and **f2** pass patterns **p2** and **p4** during fault simulation. So **p2** and **p4** do not help to distinguish **f1** and **f2**. Therefore simulation of **p2** and **p4** is not necessary. If we only simulate a subset of the passing patterns which contribute to distinguish the fault candidates derived from phase 1 of the effect-cause diagnosis, the run time of the effect-cause diagnosis can be reduced.

We propose a heuristic method to select the passing patterns based on information regarding the clocks that were active when the fault candidates failed a test. We record this information separately for each fault. The proposed method is motivated by the fact that hardware designs these days typically make use of several clocks. This is partly to save power by clock gating and also to allow the design to communicate with parts of the external environment running asynchronously.

Consider the following example. Let a fault **f** be detected by the test patterns **p1**, **p2** and **p3** of a given test pattern set, and let under test patterns **p1**, **p2** and **p3**, the effect of the fault **f** is observed at scan cells **SC1**, **SC2** and **SC3**, respectively. Let the clocks associated with the scan cell **SC1**, **SC2** and **SC3** be **CLK1**, **CLK2** and **CLK1**, respectively. Thus, we know that using the test patterns in the test pattern set used the fault **f** can only be detected by the test patterns where either **CLK1** or **CLK2** is pulsed. However, there may be some test patterns where either **CLK1** or **CLK2** or both are pulsed but the fault **f** is not detected. Thus the test patterns where either **CLK1** or **CLK2** is pulsed is a superset of the test patterns from the test pattern set used that detect the fault **f**. Based on the above analysis, we propose the following heuristic to select the passing test patterns for fault simulation.

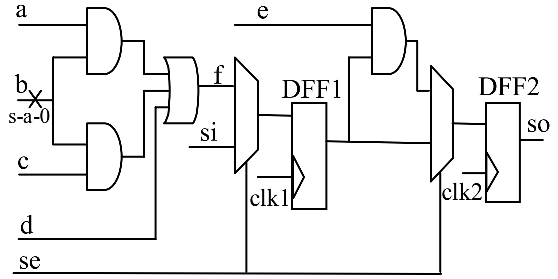

Figure 3: An example of passing pattern selection

In the proposed dictionary, we store the clocks which are associated with the scan cells where the fails are observed for each fault. In the second phase of the effect-cause diagnosis, we search the dictionary to find all the clocks which are associated with the fault candidates we obtained in phase 1. Then only the passing patterns which pulse at least one of these clocks are chosen for fault simulation during phase 2. It can be shown that the selected subset of passing patterns is a superset of patterns that can cause passing mismatch which is the goal of phase 2. Thus no loss of accuracy is introduced by the proposed method.

Next we use the circuit in Figure 3 to illustrate the proposed heuristic. Considering the circuit in Figure 3, there are two clocks, **clk1** and **clk2**, driving the flip-flops **DFF1** and **DFF2**, respectively. **DFF1** and **DFF2** are connected serially to form a scan chain. Suppose that there is a stuck-at 0 defect at input pin **b**. After applying a test pattern set **T** shown in Table 2 to the above circuit, the failures are only observed at **DFF1** from test patterns **p1** and **p2**. From the failing patterns **p1** and **p2**, the effect-cause defect diagnosis first identifies fault **b/0** (**b** stuck-at **0**) and **f/0** as the initial defect candidates, and then simulates **b/0** and **f/0** over all the passing patterns **p3**, **p4** and **p5**. In this case, **f/0** will cause passing pattern **p3** to fail but **b/0** will pass all the passing patterns. So the effect-cause defect diagnosis will rank the candidate **b/0** higher than **f/0** since **b/0** has less passing pattern mismatches. In the proposed method, since **b/0** and **f/0** can only propagate the faulty effect to **DFF1**, **b/0** and **f/0** can only be detected by the test patterns which pulse clock **clk1**. So we only need to simulate passing pattern **p3** instead of simulating **p3**, **p4** and **p5**. Therefore the number of passing pattern needing to be simulated is reduced from 3 to 1.

Table 2: Test pattern set

test pattern	a	b	c	d	e	si	se	clk1	clk2	DFF1	DFF2
p1	1	1	0	0	X	X	0	pulse	0	X	X
p2	0	1	1	0	X	X	0	pulse	0	X	X
p3	1	0	1	1	X	X	0	pulse	0	X	X
p4	X	X	X	X	1	X	0	0	pulse	1	X
p5	X	X	X	X	0	X	0	0	pulse	1	X

4. Experimental results

To evaluate the run time speed up using the proposed diagnosis method, we conducted experiments on four industrial designs described in Table 1. We used a commercial effect-cause diagnosis tool and a modified version of the tool using the proposed dictionary. The gate count, the number of test patterns, and the number of faults are listed in Table 1. For designs **Ckt1** and **Ckt2**, we created 1000 test cases for each design by injecting defects. For design **Ckt4**, we only created 89 test cases due to its long simulation times. In each test case for these three circuits, we injected one single stuck-at fault into the circuit and simulate the faulty circuit to get the failure responses. The failure responses are used in the diagnosis. For **Ckt3**, we used real failure data collected from the tester for diagnosis. The number of test cases for **Ckt3** is 1000. The experimental results for designs from **Ckt1** to **Ckt4** are reported in Table 3 and Figures 4 through 7. Recall that **Q** and **H** are the sets of candidate faults that are simulated in the standard effect-cause procedure and the proposed procedure using a small dictionary, respectively. In Table 3, **AVG Q** is the average number of faults in **Q** over all the test cases. **AVG H** is the average number of faults in **H** over all the test cases. **EVE Q** is the average number of events triggered by the faults in **Q** during the fault simulation for each test case. **EVE H** is the average number of events triggered by the faults in **H** during the fault simulation for each test case. **AVE SF** is the average speed up of phase 1 of effect-cause diagnosis and **AVE SP** is the average speed up of phase 2. **AVE** and **Median** are the average and median speed up factor of the complete diagnostic procedure. From Table 3 we can see that the proposed method can reduce the number of events in fault simulation dramatically though the average number of faults in **H** may be larger than **Q**. Experiments show that this is due to the fact that while the faults in set H propagate events only in the input cones of the outputs that produced faulty values during the application of a failing pattern the faults in **Q** tend to propagate events to many circuit outputs. We also notice that phase 2 is not speeded up by the proposed method for **Ckt3** and **Ckt4**. The reason is that the test patterns for **Ckt3** and **Ckt4** are clock compacted. In each pattern of **Ckt3** and **Ckt4**, all clocks are pulsed and the information on clocks we use can not help select a smaller subset of passing pattern for simulation. However, the complete diagnostic procedure is speeded up for all circuits as seen from Table 3.

In Figures 4 to 7 the X-axis is the index of a test case and the Y-axis gives the corresponding speed up factor. Each dot in the figure corresponds to individual test case and the thick horizontal line is the average speed up for all test cases. From Figures 4 to 7, we can see that the proposed method can speed up the effect-cause diagnosis by up to 156X. The average (median) speed up factors for the designs **Ckt1** to **Ckt4** are 8.0 (6.6), 7.8 (4.4), 9.6 (7.9), and 48 (31), respectively. That is, the over all diagnosis times are reduced on average by these factors. Thus the proposed method is seen to be quite effective in speeding up effect-cause diagnosis.

We also observed that the run time of the proposed method does not vary widely for different defect diagnosis instances as the standard effect-diagnosis procedures do. To illustrate this, in Figures 8 and 9 we plot normalized run time of the diagnosis procedures applied to **Ckt4** with and without the proposed dictionary, respectively, by dividing the run time by the average of the run times over all test cases. It can be seen that the run times of the standard diagnosis procedures vary widely whereas the run times of the procedure using the dictionary does not vary as widely.

5. Conclusions

In this paper, we proposed a technique based on a small dictionary to speed up the effect-cause fault diagnosis. The proposed method uses a very small sized dictionary and it can speed up the effect-cause diagnosis by up to 156X for the designs used in the experiment without losing any diagnosis accuracy.

6. References

[1] I. Pomeranz and S. M. Reddy, "On the Generation of Small Dictionaries for Fault Location", in Proc. ICCAD, 1992, pp. 272-279.

[2] V. Boppana, I. Hartantet and W. K. Fuchs, "Full Fault Dictionary Storage Based on Labeled Tree Encoding", in Proc. VTS, 1996, pp. 174-179.

[3] B. Chess and T. Larrabee, "Creating Small Fault Dictionaries", IEEE TCAD, Vol. 18, No.3, 1999, 346 – 356.

[4] M. Abramovici and M. A Breuer, "Fault Diagnosis Based on Effect-Cause Analysis: An Introduction", in Proc. DAC, 1980, pp. 69-76.

[5] J. A. Waicukauski and E. Lindbloom, "Failure Diagnosis of Structured Circuits", IEEE Design and Test of Computers, Vol. 6, No. 4, 1989, pp. 49-60.

[6] W. Zou, W. -T. Cheng, S. M. Reddy, H. X. Tang, "On Methods to Improve Location Based Logic Diagnosis", in Proc. VLSI Design, 2006, pp. 181-187.

[7] T. Bartenstein, D. Heaberlin, L. Huisman and D. Sliwinski, "Diagnosing Combinational Logic Designs Using the Single Location At-a-Time (SLAT) Paradigm", in Proc. ITC, 2001, pp. 287-296.

[8] S. Venkatarman and S. Drummonds, "POIROT: A Logic Fault Diagnosis Tool and Its Applications", in Proc. ITC, 2000, pp. 253-262.

[9] D. B. Lavo I. Hartanto, and T. Larrabee, " Multiplets, Models, and the Search for Meaning: Improving Per-Test Fault Diagnosis", in Proc. ITC, 2002, pp. 250-259.

[10] Z. Wang, M. M. Sadowska, et al., "An Efficient and Effective Methodology on the Multiple Fault Diagnosis", in Proc. ITC, 2003, pp. 329-338.

[11] B. Seshadri, I. Pomeranz, S. Venkataraman and S.M. Reddy, "Dominance Based analysis for Large Volume Production Fail Diagnosis", in Proc. VTS 2006, pp. 392-399.

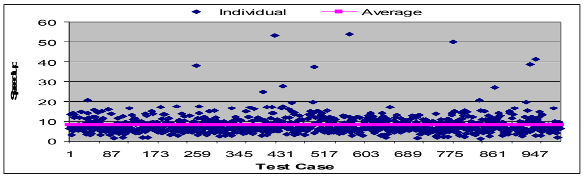

Figure 4: Speed up for Ckt1

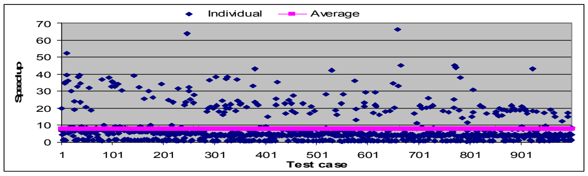

Figure 5: Speed up for Ckt2

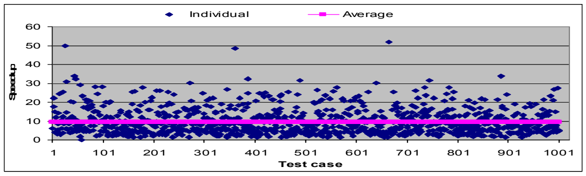

Figure 6: Speed up for Ckt3

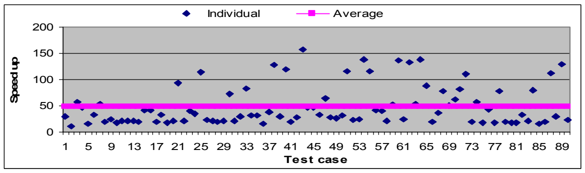

Figure 7: Speed up for Ckt4

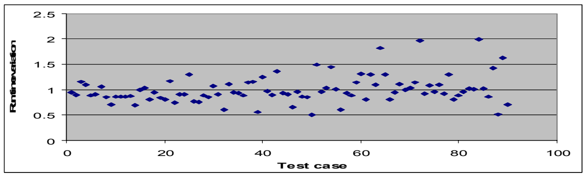

Figure 8: Run time variation of effect-cause diagnosis with using a dictionary for Ckt4

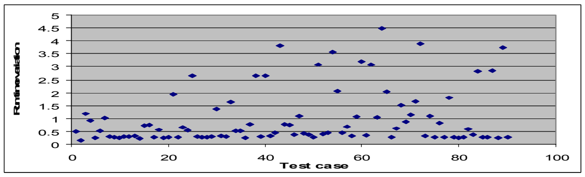

Figure 9: Run time variation of effect-cause diagnosis without using a dictionary for Ckt4

Table 3 Experimental Results for Four Industrial Circuits

Circuit	Ckt1	Ckt 2	Ckt 3	Ckt 4
AVG Q	134	51	210	794
AVG H	197	141	154	219
AVG Q/AVG H	0.68	0.36	1.36	3.6
EVE Q	2046.4K	380K	72.4M	2680M
EVE H	45.6k	48K	143K	1.1M
EVE Q/EVE H	45	7.9	506	2436
AVE SF	10.5	2.6	24	211
AVE SP	6.3	8.9	1	1
AVE	8	7.8	9.6	48
Median	6.6	4.4	7.9	31