ELectronics EXpress

A method for the fast diagnosis of multiple defects using an efficient candidate selection algorithm

Yoseop Lim, Jaeseok Park, and Sungho Kang[a)]
Dept. of Electrical and Electronic Eng., Yonsei University
134 Sinchon-dong, Seodaemun-gu, Seoul 120-749, Korea
a) *shkang@yonsei.ac.kr*

Abstract: The demand for fault diagnosis has increased with the increasing complexity of VLSI devices. Recent analysis has found that multiple defects frequently exist in failing chips. Therefore, the diagnosis of multiple defects is very important and is needed in the industry. Here we propose a multiple-defect diagnosis method using an efficient selection algorithm that can handle various defect behaviors. The experimental results for the full-scan version of the ISCAS '89 benchmark circuits demonstrate the efficiency of the proposed methodology in diagnosing circuits that are affected by a number of different types of defects.

Keywords: failure analysis, fault diagnosis, multiple defects
Classification: Integrated circuits

References

[1] Y. Takamatsu, T. Seiyama, H. Takahashi, Y. Higami, and K. Yamazaki, "On the fault diagnosis in the presence of unknown fault models using pass/fail information," *ISCAS 2005, IEEE Int. Symp. Circuits ans Syst.*, pp. 2987–2990, 2005.

[2] N. Jha and S. Gupta, *Testing of Digital Systems*, Cambridge Univ. Press, Cambridge, 2003.

[3] X. Yu and R. D. Blanton, "Diagnosis of integrated circuits with multiple defects of arbitrary characteristics," *IEEE Trans. Comput.-Aided Des. Integr. Circuits and Syst.*, vol. 29, pp. 977–987, June 2010.

[4] S. Venkataraman and W. Fuchs, "A deductive technique for diagnosis of bridging faults," *Proc. Int. Conf. Computer-Aided Design*, pp. 562–567, 1997.

1 Introduction

The demand for fault diagnosis has increased with the increasing complexity of VLSI (very large scale integration) devices. Fault diagnosis is a process that deduces the location of the defect that caused the failures. Therefore, it

DOI: 10.1587/elex.9.834
Received March 14, 2012
Accepted April 11, 2012
Published May 07, 2012

is very important to develop an efficient fault diagnosis methodology in order to improve device quality and to reduce production costs [1].

Single-fault diagnosis is a well-studied problem with various linear-time solutions [2]. However, the single-fault model may not be adequate for diagnosing defects in modern devices, which tend to cluster and affect multiple lines in a failing chip. Therefore, the diagnosis of multiple defects is very important and is needed in industry. Multiple-fault diagnosis is a challenging problem because, in theory, the search space grows exponentially with an increasing number of faulty lines.

In [3], the defect candidates are identified through a defect site identification and elimination procedure. The effects of those candidates are then analyzed and organized in a candidate forest. However, the diagnosis time and the memory usage needed to create the candidate forest increase rapidly as the number of defects increases. To alleviate this problem, an efficient multiple-defect diagnosis method is proposed that does not require large, growing data structures. During the entire diagnosis flow in the new method, no characteristics of any failing patterns are used, and no assumptions regarding specific defect behaviors are made such that arbitrary defects can be diagnosed. First, the enhanced path-tracing procedure identifies defect candidates, and several metrics (N_{EPT}, N_{PT}) to rank the defect candidates are determined. The candidate selection procedure then ranks and analyzes the defect candidates to report accurate diagnosis results. According to comparative experiments, the proposed diagnosis method dramatically reduces the diagnosis time, and increases the diagnosability and resolution as compared to the state-of-the-art method proposed in [3].

2 Enhanced path-tracing procedure

Most diagnosis methods use path-tracing [4] to reduce the defect search space. For a given failing test pattern t_k, the path-tracing procedure [4] traces back through the circuit from every failing observable point. An observable point is failing if its observed response is different from its response when the CUD (circuit under diagnosis) is defect-free. A signal line is called a *site* if the signal line has a stuck-at value $v \in \{0, 1\}$ for some test pattern t_k. Any lines encountered during the trace are implicated as potential defect sites. The logic-value of a gate input is said to be *controlling* if it determines the output value of the gate regardless of the other input values. The path-tracing processes any internal lines while tracing back through the CUD in the following manner. When a gate output is reached and all of the gate inputs have non-controlling values, the path-trace implicates all of the gate inputs and continues tracing from each input. If one or more inputs have controlling values applied, only those inputs are implicated, and the trace continues only from those controlling inputs. If a fanout branch is reached, it is implicated, and the tracing continues from the corresponding stem.

Path-tracing typically implicates many defect-free sites, so we sought to find a way to improve the accuracy of the diagnosis. In the path-tracing

DOI: 10.1587/elex.9.834
Received March 14, 2012
Accepted April 11, 2012
Published May 07, 2012

procedure, if one or more inputs have controlling values applied, those inputs are implicated, and the trace continues from those controlling inputs. In this case, all inputs that have controlling values simultaneously become defect sites to explain the fault effects. If any input that has a controlling value remains fault-free, the output is also unchanged and remains fault-free. Therefore, these controlling inputs are less likely to be real defects than the non-controlling inputs. According to this observation, if only one input has controlling values applied in the enhanced path-tracing procedure, those inputs are implicated, and the trace continues. Otherwise, only the output of the gate is implicated and the trace stops. This method simply reduces the number of implicated defect-free sites. N_{EPT} (N_{PT}) is the number of implications throughout the enhanced path-tracing (the path-tracing), and indicates how often a candidate is implicated during the enhanced path-tracing (the path-tracing) for failing patterns.

3 Candidate selection procedure

The candidates implicated through the enhanced path-tracing are ranked according to these metrics (N_{EPT}, N_{PT}), and a list of defect candidates is formed. N_{EPT} is the first priority and N_{PT} is the second priority. Candidate s_i/v_i' is said to output-explain an observable point out_j for a failing pattern t_k if the fault-free value on site s_i is v_i, and an error from s_i propagates to out_j in t_k when all of the side inputs of the on-path gates have fault-free values [3]. The candidate selection procedure selects the minimal group of candidates such that the combined candidates jointly output-explain all of the observed fail points from the defect candidate list and reports a diagnosis output.

Our approach uses a fault simulator to evaluate the error propagation of the defect candidate. The group of every failing observable point is called a *window*, and a candidate is called a *window match* if there is an erroneous observable point within a window while there is no error outside of the window. N_{WM}, the number of the window match, is how often a candidate window matches each failing pattern. The numerical difference between N_{PT} and N_{WM} can separate a real defect candidate from the others. We found that N_{WM} is identical to N_{PT} in most real defect candidates since the erroneous effect of a real defect candidate tends to be propagated in the window. On the other hand, if the erroneous effect of a certain candidate is easily propagated topologically, that effect does not tend to be propagated in only the window. The number of erroneous observable points within a window referred to as N_{EO} is also exploited in diagnosis. A candidate that has a higher N_{EO} is more likely to be a real defect. Candidates that have the same N_{EPT} and N_{PT} are distinguishable according to N_{EO}.

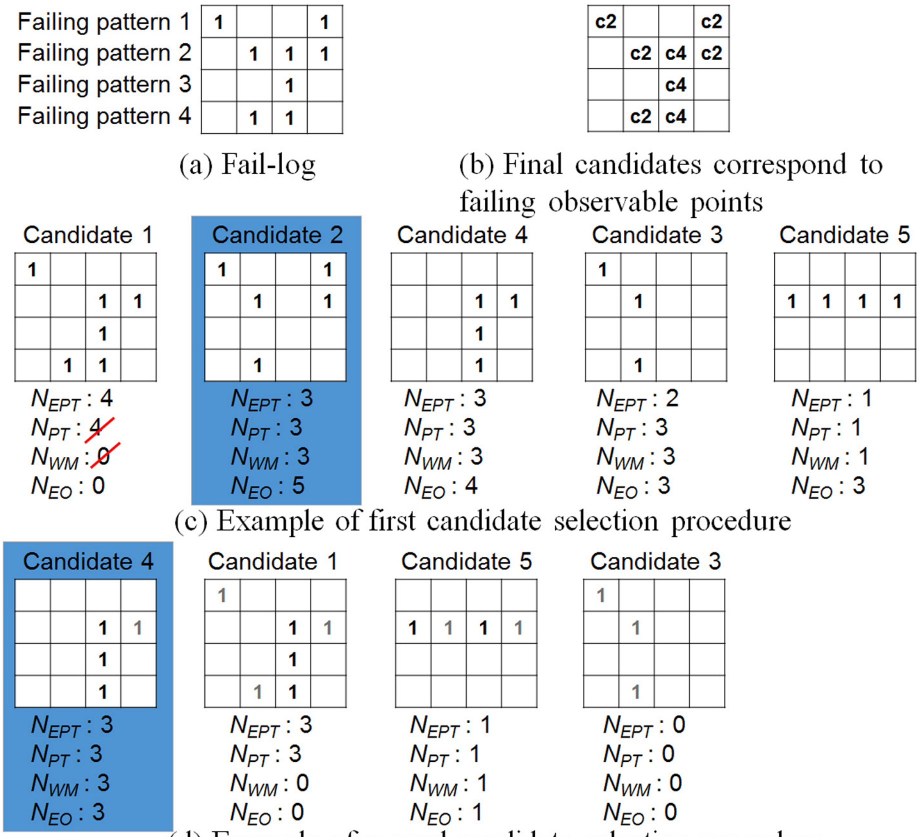
The final candidates are selected according to the procedure below.

DOI: 10.1587/elex.9.834
Received March 14, 2012
Accepted April 11, 2012
Published May 07, 2012

```
While ( the unexplained fail patterns exist ){
    /* enhanced path-tracing procedure */
    enhanced path-tracing for the unexplained fail patterns : calculate $N_{EPT}$ and $N_{PT}$
    make the defect candidate list : sort candidates according to $N_{EPT}$ and $N_{PT}$
    /* candidate selection procedure */
    While( a new finial candidate is not selected ){
        for ( the same ranked candidates from the top of the defect candidate list ){
        fault simulate candidates for the unexplained fail patterns : calculate $N_{WM}$ and $N_{EO}$
        sort candidates according to $N_{EO}$
        if ( a candidate $c_i$ == Explain Candidate && $N_{PT}(c_i)$ == $N_{WM}(c_i)${
            a candidate $c_i$ is selected as a new finial candidate }}}}
```

If a candidate output-explains a failing observable point that is not output-explained by the final candidates, the candidate is referred to as *Explain Candidate*. In order to reduce the time of candidate selection procedure, the same ranked candidates are simulated in parallel from the top of the defect candidate list.

The example shown in Fig. 1 illustrates the candidate selection procedure. Fig. 1 (a) shows the fail-log with four failing observable points for four failing patterns. In Fig. 1, a "1" indicates an erroneous value. Fig. 1 (b) shows final candidates correspond to failing observable points. An example of the initial defect candidate list is shown in Fig. 1 (c). First, Candidate 1 is checked. However, Candidate 1 is not selected since N_{PT} and N_{WM} of Candidate 1 are different. Then Candidate 2 is selected as a final candidate since it is *Explain Candidate* and N_{PT} and N_{WM} of Candidate 2 are identical. After Candidate 2 is selected, the enhanced path-tracing procedure is performed for



(a) Fail-log

(b) Final candidates correspond to failing observable points

(c) Example of first candidate selection procedure

(d) Example of second candidate selection procedure

Fig. 1. Example of the candidate selection procedure.

DOI: 10.1587/elex.9.834
Received March 14, 2012
Accepted April 11, 2012
Published May 07, 2012

the unexplained fail patterns, and N_{EPT} and N_{PT} are recalculated. Fig. 1 (d), Candidate 4 is selected and the diagnosis procedure is completed since all failing observable points are output-explained by Candidates 2 and 4.

4 Experimental results

The experiments were conducted using the full-scan combinational versions of the larger ISCAS'89 benchmarks. To show the capability of our method for diagnosing faults with various fault models, we injected multiple permanent stuck-at faults, transition faults, and two-line (AND and OR-type) bridging faults in the experiment. The test patterns were generated by a commercial ATPG (automatic test pattern generation) tool and had 100% stuck-at fault test coverage. We considered 60 random fault injections for each circuit, fault type, and multiplicity. Multiple faults were injected by randomly selecting a site. The fault is injected into the circuit if a fault has not already been injected at that site and is detected by test patterns.

Diagnosability is defined as S_D/S_I, where S_D is the number of actual defect sites identified by the diagnosis, and S_I is the number of injected defect sites [3]. The resolution is defined as S_D/S_T, where S_T is the total number of sites reported by the diagnosis. For any diagnostic approach in the ideal case, the diagnosability is 1, and the resolution is also 1; that is, $S_D/S_I = S_D/S_T = 1.0$.

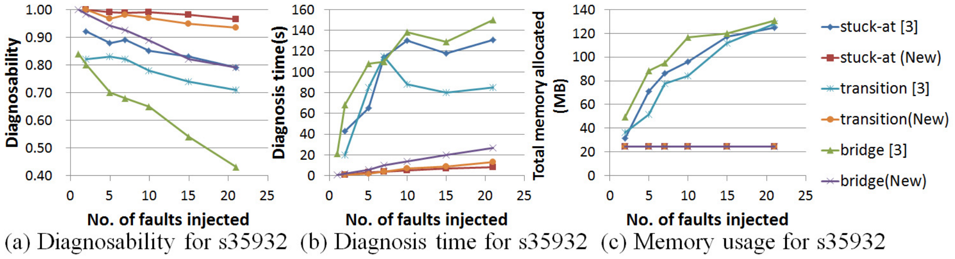
Table I shows a comparison of the diagnosability and resolution of the

Table I. The comparison of diagnosability and resolution with those of the prior work

circuit	F.T.	2 faults				5 faults				15 faults				21 faults			
		Diag.		Res.		Diag.		Res.		Diag.		Res.		Diag.		Res.	
		[3]	New	[3]	New	[3]	New	[3]	New	[3]	New	[3]	New	[3]	New	[3]	New
s13207	S.	0.92	**1.00**	0.44	**1.00**	0.86	**0.99**	0.59	**0.98**	0.78	**0.97**	0.63	**0.95**	0.76	**0.96**	0.61	**0.95**
	T.	0.80	**0.99**	0.34	**0.98**	0.74	**0.96**	0.54	**0.91**	0.66	**0.93**	0.56	**0.89**	0.69	**0.91**	0.59	**0.89**
	B.	0.79	**0.97**	0.62	**0.93**	0.74	**0.93**	0.62	**0.91**	0.65	**0.85**	0.59	**0.75**	0.64	**0.84**	0.61	**0.76**
s15850	S.	0.94	**1.00**	0.41	**1.00**	0.88	**0.99**	0.55	**1.00**	0.78	**0.93**	0.62	**0.84**	0.69	**0.91**	0.56	**0.82**
	T.	0.87	**1.00**	0.45	**0.99**	0.85	**0.98**	0.61	**0.94**	0.71	**0.92**	0.58	**0.87**	0.72	**0.89**	0.60	**0.84**
	B.	0.85	**0.99**	0.51	**0.93**	0.75	**0.93**	0.65	**0.80**	0.65	**0.84**	0.60	**0.65**	0.60	**0.82**	0.57	**0.63**
s35932	S.	0.92	**1.00**	0.40	**1.00**	0.88	**0.99**	0.65	**1.00**	0.83	**0.98**	0.49	**0.97**	0.79	**0.97**	0.44	**0.92**
	T.	0.82	**1.00**	0.35	**0.99**	0.83	**0.97**	0.52	**0.92**	0.74	**0.95**	0.46	**0.89**	0.71	**0.93**	0.45	**0.86**
	B.	0.80	**0.98**	0.46	**0.96**	0.70	**0.94**	0.49	**0.90**	0.54	**0.82**	0.40	**0.72**	0.43	**0.79**	0.38	**0.67**
s38417	S.	0.97	**0.98**	0.68	**0.99**	0.98	**0.99**	0.84	**0.99**	0.93	**0.99**	0.74	**0.98**	0.90	**0.98**	0.71	**0.96**
	T.	0.90	**0.98**	0.32	**0.99**	0.88	**0.98**	0.69	**0.99**	0.86	0.84	0.71	**0.95**	0.85	0.83	0.71	**0.94**
	B.	0.90	**0.98**	0.77	**1.00**	0.87	**0.88**	0.79	**0.95**	0.81	0.76	0.70	**0.84**	0.75	**0.75**	0.63	**0.81**
s38584	S.	0.97	**1.00**	0.55	**0.99**	0.92	**1.00**	0.67	**1.00**	0.93	**0.97**	0.82	**0.97**	0.94	**0.96**	0.85	**0.93**
	T.	0.90	**1.00**	0.51	**0.98**	0.86	**0.98**	0.77	**0.96**	0.85	**0.95**	0.82	**0.90**	0.87	**0.94**	0.82	**0.90**
	B.	0.74	**0.99**	0.64	**0.98**	0.78	**0.96**	0.81	**0.92**	0.77	**0.89**	0.80	**0.80**	0.74	**0.87**	0.78	0.77
Average	S.	0.94	**1.00**	0.50	**1.00**	0.90	**0.99**	0.66	**0.99**	0.85	**0.97**	0.66	**0.94**	0.82	**0.95**	0.63	**0.92**
	T.	0.86	**1.00**	0.40	**0.99**	0.83	**0.97**	0.63	**0.94**	0.76	**0.92**	0.62	**0.90**	0.77	**0.90**	0.63	**0.88**
	B.	0.82	**0.98**	0.60	**0.96**	0.77	**0.93**	0.67	**0.90**	0.68	**0.83**	0.62	**0.75**	0.63	**0.81**	0.59	**0.73**

F.T.: Fault Types, S.: Stuck-at faults, T.: Transition faults, B.: Bridge faults

© IEICE 2012
DOI: 10.1587/elex.9.834
Received March 14, 2012
Accepted April 11, 2012
Published May 07, 2012



(a) Diagnosability for s35932 (b) Diagnosis time for s35932 (c) Memory usage for s35932

Fig. 2. The comparative experimental result for s35932.

proposed method with those of the state-of-the-art method [3]. In Table I, the first column is the circuit name, and the second column is the type of injected faults. The rest of the columns are the average diagnosabilities and resolutions of different numbers of injected fault. The cases in which the results of the proposed diagnosis method are better than the results reported in [3] are in bold. In most cases, the diagnosability and resolution of the proposed method are better than those of the method developed in [3].

Fig. 2 shows the diagnosability, the diagnosis time and memory usage of s35932 for a varying number of injected faults. The diagnosability of the proposed method is better than that of the method in [3] in every case, while the proposed method is significantly faster than that in [3]. The diagnosis time of the proposed method increased very slowly and the memory usage remains unchanged as the number of injected faults increased.

5 Conclusion

In this paper, we proposed a multiple defect diagnosis method that can cope with a large number of defects, various defect behaviors, and arbitrary failing pattern characteristics. The diagnosability and resolution of the proposed method are high, while the proposed method is significantly fast. Therefore, the proposed diagnosis method is very practical for large industrial designs and has the flexibility to be applied in various environments.

Acknowledgments

This work was supported by the IT R&D program of the MKE/KEIT, Rep. of Korea [2009-10034834, A development of ASIC chip for next generation high speed ATE].

DOI: 10.1587/elex.9.834
Received March 14, 2012
Accepted April 11, 2012
Published May 07, 2012