

THE EFFECT OF FAULT DROPPING ON FAULT SIMULATION TIME

Rong Pan
Nur A. Touba
Edward J. McCluskey

Technical Report: CSL-TR-93-590

(CRC TR 93-5)

*This technical report has been submitted to the 12th IEEE VLSI Test Symposium.

November, 1993

This work was supported in part by the Innovative Science and Technology Office of the Strategic Defense Initiative Organization and administered through the Office of Naval Research under Contract No. N00014-92-J-1782, and in part by the National Science Foundation under Grants No. MIP-9107760.

Imprimatur: Samy Makar and LaNae Avra

Copyright © 1993 by the Center for Reliable Computing, Stanford University
All rights reserved, including the right to reproduce this report, or portions thereof, in any form.

The Effect of Fault Dropping on Fault Simulation Time

Rong Pan, Nur A. Touba, and Edward J. McCluskey

CRC Technical Report No. 93-5

(CSL TR No. 93-590)

CENTER FOR RELIABLE COMPUTING

Computer Systems Laboratory

Departments of Electrical Engineering and Computer Science

Stanford University, Stanford, California 943054055

ABSTRACT

The effect of fault dropping on fault simulation time is studied in this paper. An experiment was performed in which fault simulation times, with and without fault dropping, were measured for three different simulators. A speedup approximately between 8 and 50 for random test sets and between 1.5 and 9 for deterministic test sets was observed. The results give some indication about how much fault dropping speeds up fault simulation. These results also show the overhead of an application requiring a complete fault dictionary.

TABLE OF CONTENTS

1. INTRODUCTION	3
2. BACKGROUND	4
3. SIMULATION EXPERIMENTS	7
4. EXPERIMENT ANALYSIS	11
5. CONCLUSIONS	12
ACKNOWLEDGMENTS	13
REFERENCES	13

1. INTRODUCTION

Fault simulation plays a significant role in the testing of digital circuits. It determines which faults in the circuit are detected by a given set of test patterns. In many fault simulation applications, it is not necessary to simulate each fault for all the test patterns; a fault can be discarded after being detected. The removal of detected faults during simulation is called *fault dropping*. Fault dropping can accelerate fault simulation by drastically reducing the memory and CPU requirements of the fault simulator. However, only fault coverage is obtained by this method; other valuable data is lost.

When a circuit under test (CUT) fails, diagnosing the cause of the observed errors is desirable. *Fault diagnosis* is the first step to repair the CUT. One approach of fault diagnosis is based on a fault dictionary which is a database that stores the faults in the CUT and the responses of the CUT to the applied test set in the presence of these faults [Abramovici 90]. A fault dictionary is built during the process of fault simulation. The diagnosis involves a dictionary look-up process, in which an attempt is made to match the actual response of CUT with one of the faulty responses. If the match is successful, the fault dictionary suggests the possible faults. The degree of accuracy to which faults can be located is called *diagnosis resolution*. The detected faults are separated from one another by their distinct responses to the test set. Fault simulation with fault dropping lowers the diagnosis resolution. Faults detected by the same test pattern with the same outputs can not be distinguished. However, most of the faults can be separated by subsequent test patterns. Thus, in order to get higher diagnosis resolution, a fault dictionary needs to be generated without fault dropping.

A fault dictionary can also be used to achieve zero aliasing in signature analysis [Pomeranz 92]. A signature is the compressed circuit response to an applied test set using appropriate hardware. Zero aliasing is the case where the faulty signature is always different from the fault-free signature. In order to guarantee that the signature of each faulty circuit is different from the fault-free signature, it is necessary to build up a complete fault dictionary that keeps a record of circuit responses and corresponding test patterns.

There is a tradeoff between fault simulation with and without fault dropping. With fault dropping, fault coverage is determined in a relatively short time, but a complete fault dictionary is not generated. On the other hand, if fault dropping is not adopted in favor of a complete fault dictionary, more time is spent on fault simulation.

This paper studies the difference in simulation time between fault simulation with and without fault dropping. It is organized as follows: in Sec. 2, background and notation are introduced; in Sec. 3, experiments comparing simulation times are described; in Sec. 4, the various factors that effect the simulation results are discussed; and finally in Sec. 5, conclusions are presented.

2. BACKGROUND

A typical fault detection curve during fault simulation is shown in Fig. 1. When simulation begins, a large percentage of faults are detected in a short amount of time. However, as time goes on, the rate at which faults are detected decreases because the test patterns applied detect many faults that have already been detected. If these detected faults are not dropped, extra time is spent on resimulate these faults but the fault coverage remains the same. Thus almost all simulators drop a fault as soon as it has been detected if the goal is to just determine fault coverage. On the other hand, if the goal is to build up a fault dictionary or to predict aliasing in signature analysis, the extra time must be spent.

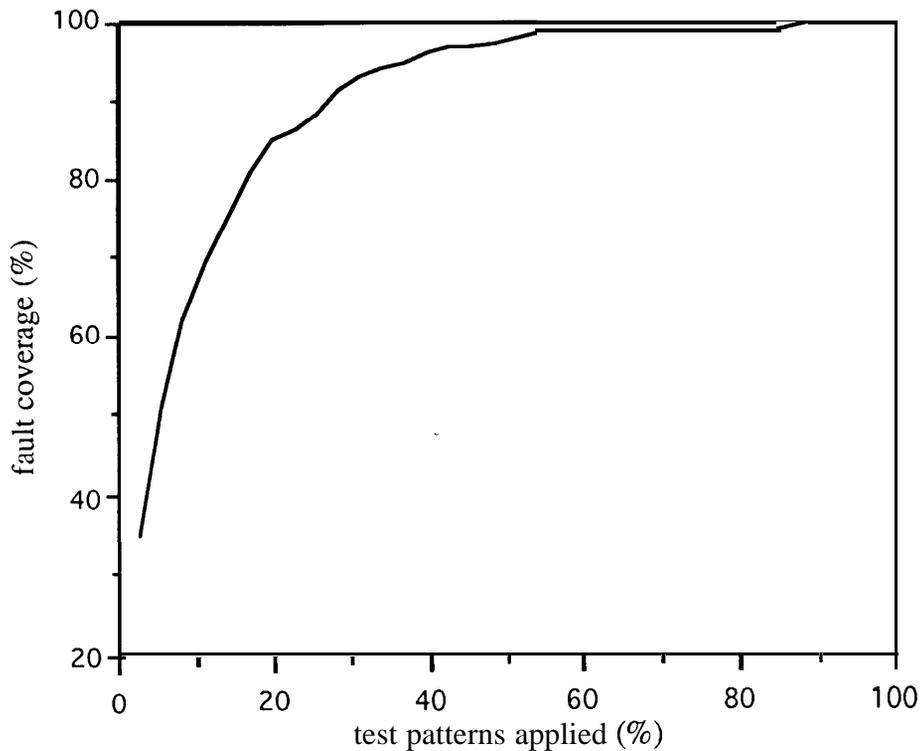


Figure 1. A typical detection curve

The following ratio, which will be referred to as the *speedup*, shows how much fault dropping affects fault simulation time:

$$\text{Speedup} = \frac{\text{fault simulation time without fault dropping}}{\text{fault simulation time with fault dropping}}$$

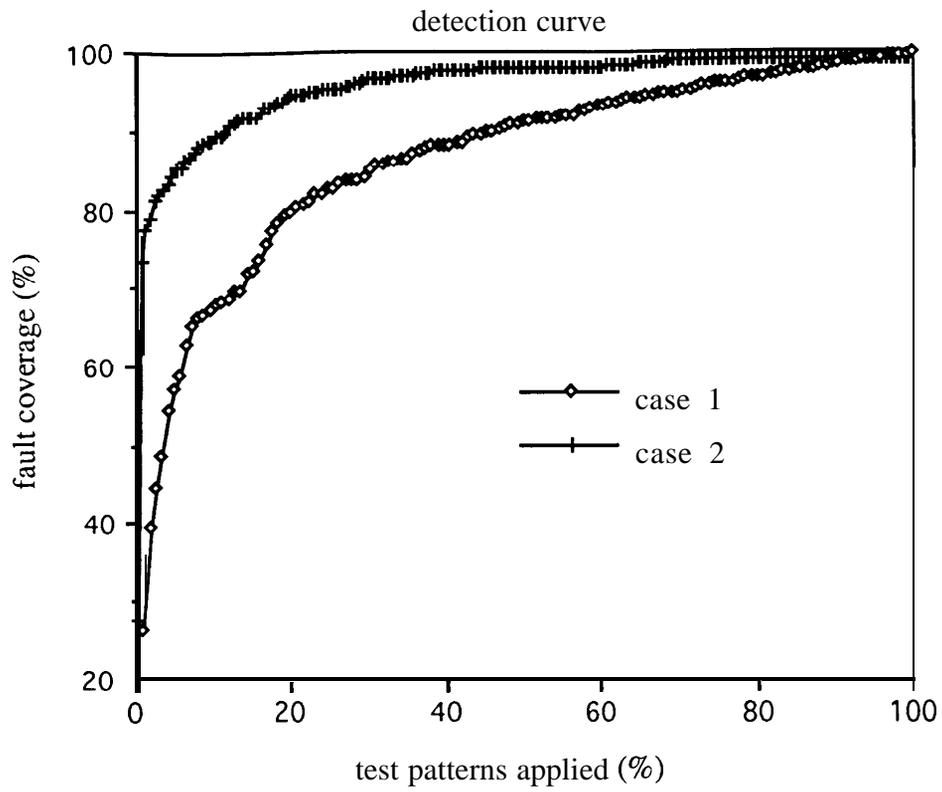
The speedup is related to the fault detection curve. The larger the area under the detection curve is, the larger the speedup. Consider the following example. For the two curves in Fig. 2(a), we see that for case 1 the first 20% of the test patterns applied yields about 80% fault coverage, and the remaining 80% of the test patterns yields about 20% coverage. While for case 2, the first 10% of the test patterns applied yields about 90% fault coverage, and the remaining 90% of the patterns only yields about 10% coverage. The percentage of faults simulated for each pattern with and without fault dropping is shown in Fig. 2(b). When fault dropping is not used in fault simulation, all of the faults are simulated for each pattern; this is shown by a straight line for both cases. When fault dropping is done, the percentage of faults that are simulated for each pattern is the percentage of faults that have not been detected yet; this is also shown in Fig. 2(b). The area under each curve represents the normalized simulation time with fault dropping. The area under the straight line represents the normalized simulation time without fault dropping.¹

For case 1: Normalized simulation time without dropping = 1,
 Normalized simulation time with dropping = 0.16,
 Speedup = 6.25;

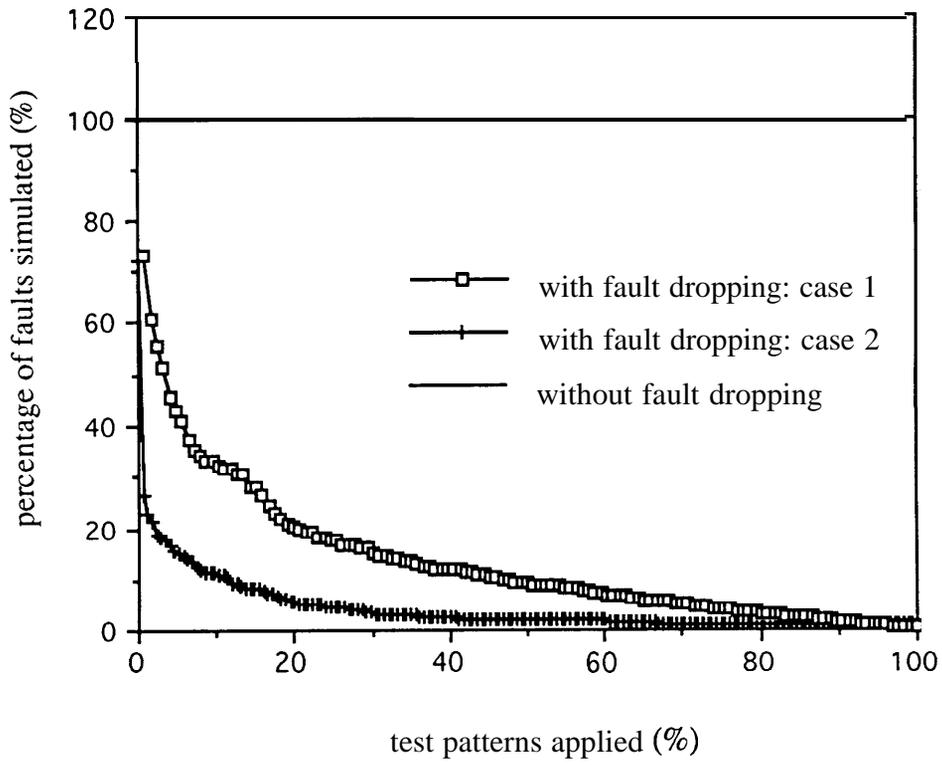
For case 2: Normalized simulation time without dropping = 1,
 Normalized simulation time with dropping = 0.06.
 Speedup = 16.67.

This example illustrates that if the area under the detection curve is larger, the speedup is higher.

¹ the data is from actual simulation times.



(a)



(b)

Figure 2. Example using two different detection curves

3. SIMULATION EXPERIMENTS

Three different fault simulators, FSIM, HILO, and HOPE, were used to do the fault simulation experiments. FSIM is a parallel fault simulator using a single propagation technique [Lee 91]. HOPE is a parallel simulator [Lee 92] for sequential circuits. HILO² system is a set of software tools providing the designer of digital circuits with design verification and test validation capability. Fault simulation with the parallel value list algorithm is one of the simulation capabilities of HILO system [Son 85].

In order to turn off fault dropping, the source code of FSIM and HOPE were modified, and special options in HILO were used. The simulations were conducted on a Sun Sparc SLC workstation. Eight of the ISCAS '85 benchmark circuits [Brglez 85] were simulated. Their characteristics are summarized in Table 1.

Table 1. Characteristics of combinational gate-level benchmark circuits

circuit	c432	c499	c880	c1355	c1908	c3540	c5315	c6288
total gates	160	202	383	546	880	1669	2307	2406
total lines	432	492	880	1355	1908	3540	5315	6288
primary inputs	36	41	60	41	33	50	178	32
primary outputs	7	32	26	32	25	22	123	32
number of faults	524	758	942	1574	1879	3428	5350	7744

Two test sets were applied to the benchmark circuits. One was a deterministic test set which was generated by Atalanta (an automatic test pattern generation system). The other was a randomly generated test set which is long enough to achieve a high fault coverage. The lengths and fault coverages of both the deterministic test set and the random test set are given in Table 2.

Table 2. Characteristics of test sets applied to benchmark circuits

circuit	c432	c499	c880	c1355	c1908	c3540	c5315	c6288
Deterministic test length	56	55	62	86	126	175	143	35
Random test length	500	1500	3000	2000	3500	10000	2000	500
Deterministic fault coverage	100	100	100	100	100	100	100	100
Random fault coverage	99.24	100	99.15	99.55	99.41	99.91	99.90	100

² HILO is a product of GenRad, Inc.

The simulation results are given in Tables 3 and 4. The definitions of the sub-headings are listed below:

- no drop: simulation time in seconds without fault dropping,
- drop: simulation time in seconds with fault dropping,
- speedup: no drop / drop.

Table 3. Results for Deterministic Test Set

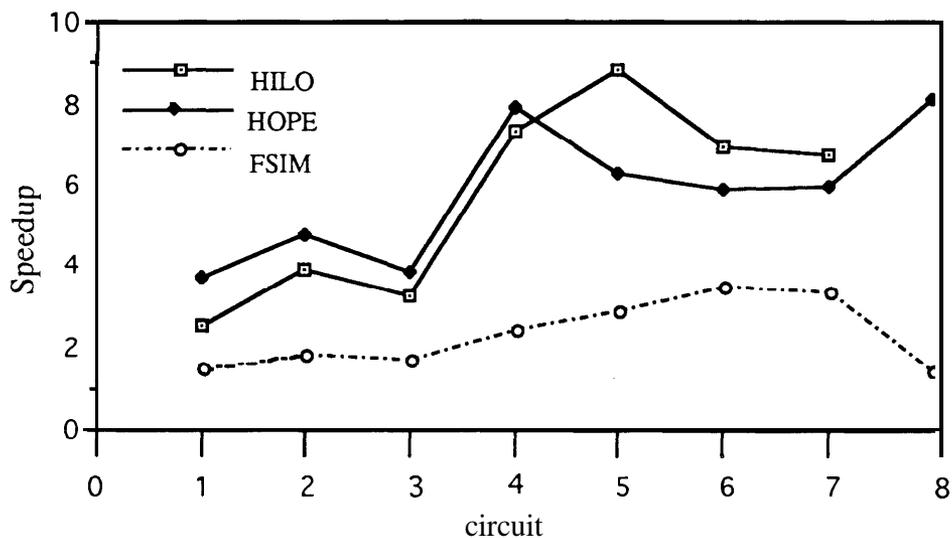
circuit	HILO ³			HOPE			FSIM		
	no drop	drop	speedup	no drop	drop	speedup	no drop	drop	speedup
c432	5.67	1.97	2.87	1.12	0.30	3.72	0.38	0.23	1.64
c499	7.33	1.87	3.92	2.15	0.45	4.78	0.33	0.17	1.99
c880	13.08	4.00	3.27	2.32	0.60	3.86	0.55	0.30	1.83
c1355	48.87	6.68	7.32	5.15	0.65	7.92	1.07	0.42	2.56
c1908	122.12	13.80	8.85	11.40	1.82	6.27	3.17	1.05	3.02
c3540	399.63	57.87	6.91	33.40	5.65	5.91	12.6	3.48	3.62
c5315	322.17	47.97	6.72	37.28	6.30	5.92	6.88	1.98	3.46
6288				22.62	2.80	8.08	15.30	9.73	1.57

Table 4. Results for Random Test Set

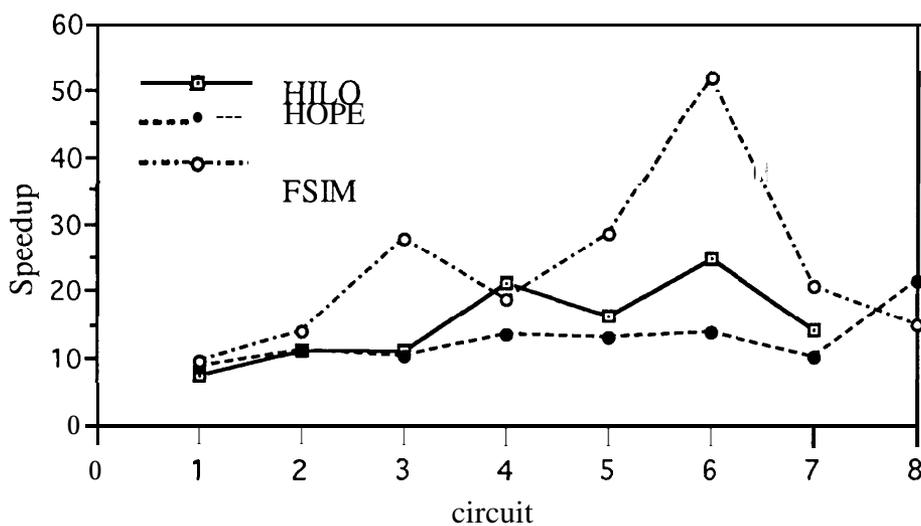
circuit	HILO			HOPE			FSIM		
	no drop	drop	speedup	no drop	drop	speedup	no drop	drop	speedup
c432	42.39	5.43	7.81	9.87	1.10	8.97	2.73	0.28	9.66
c499	203.08	18.32	11.09	61.27	5.47	11.20	4.28	0.30	14.28
c880	540.15	47.60	11.35	97.0	9.40	10.32	20.38	0.73	27.81
c1355	1143.1	53.40	21.41	122.15	8.82	13.85	17.57	0.93	18.83
c1908	2903.0	175.12	16.58	333.85	25.27	13.21	68.78	2.42	28.46
c3540	22450.	901.98	24.89	1923.1	135.68	14.17	643.63	12.35	52.12
c5315	4179.2	286.94	14.56	522.33	49.42	10.57	82.30	3.92	21.01
6288	-	-	-	274.07	12.57	21.81	159.90	10.45	15.30

³ We had problems running c6288 on the HILO simulator.

The speedup, which depends on the simulator used, the test set applied, and the benchmark circuits simulated, ranges from 1.57 to 8.85 for the deterministic test set, and from 7.81 to 52.12 for the random test set. In order to show the results more clearly, the data is plotted in Figs. 3 and 4. Figure 3 shows the speedups of different simulators when the same test set is applied. It can be seen that the speedup of FSIM is smaller than that of HOPE and HILO for the deterministic test sets, while it is the highest for the random test sets. The effect of the test sets can be clearly seen in Fig. 4. For each simulator, the speedup for random test set is greater than the speedup for the deterministic test set. More detailed discussion is given in Sec. 4.



(a) Speedup for deterministic test set



(b) Speedup for random test set

Figure 3. Speedup for each test set

Circuits : 1-c432, 2-c499, 3-c880, 4-c1355, 5-c 1908, 6-c3540, 7-c53 15, 8-c6288

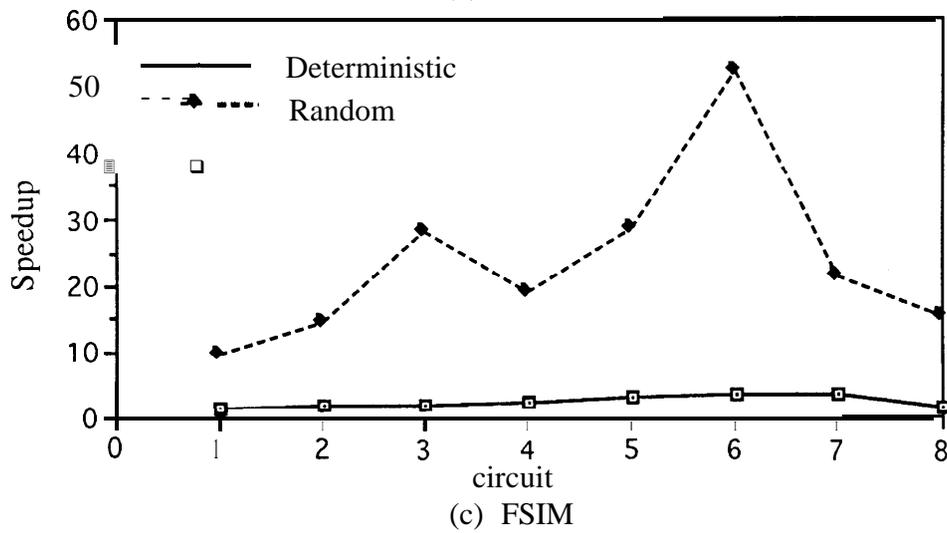
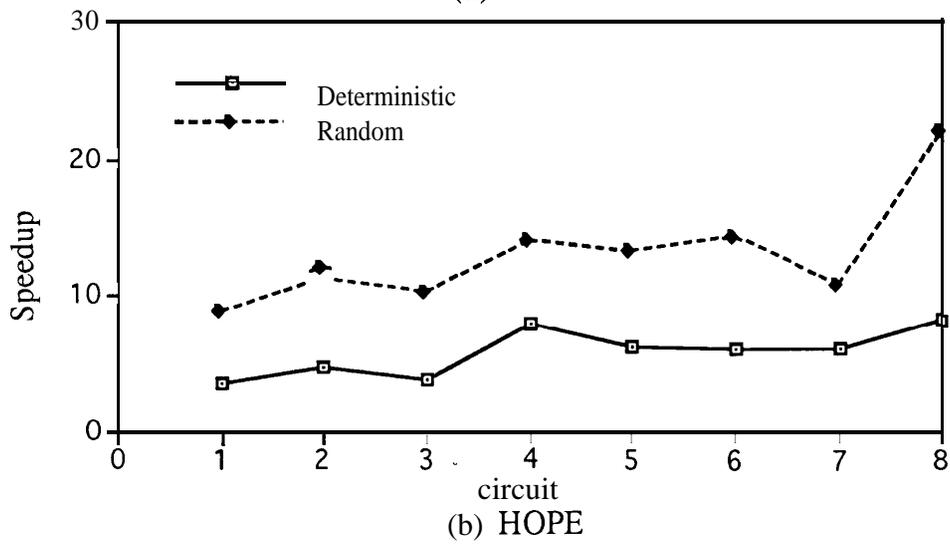
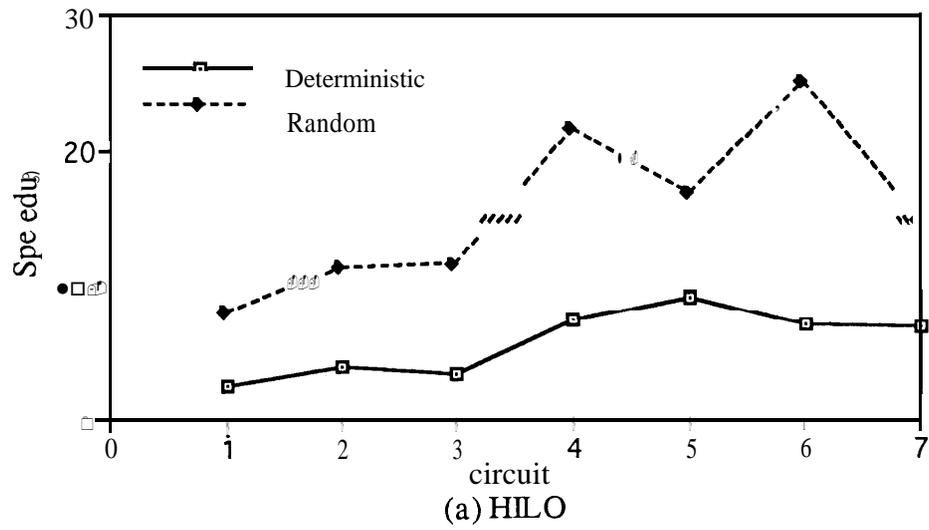


Figure 4. Speedup for each simulator

Circuits: 1-c432, 2-c499, 3-c880, 4-c1355, 5-c1908, 6-c3540, 7-c5315, 8-c6288

4. EXPERIMENT ANALYSIS

From the simulation results, it is apparent that the fault dropping speedup depends on the circuit, the test set, and the simulator used. In this section, these three factors will be discussed in turn.

First, the speedup depends on the characteristics of the CUT. Different circuits have different detection curves. If the fault simulation of one circuit detects many faults quickly under a certain test set, it will have a large speedup. Figure 5 shows the detection curves of c6288 and c432. The deterministic test sets are applied to the two circuits. It is clear that faults are detected faster in c6288 than in c432, and therefore the speedup of c6288 should be larger than that of c432. The results from the three simulators are consistent with our expectation.

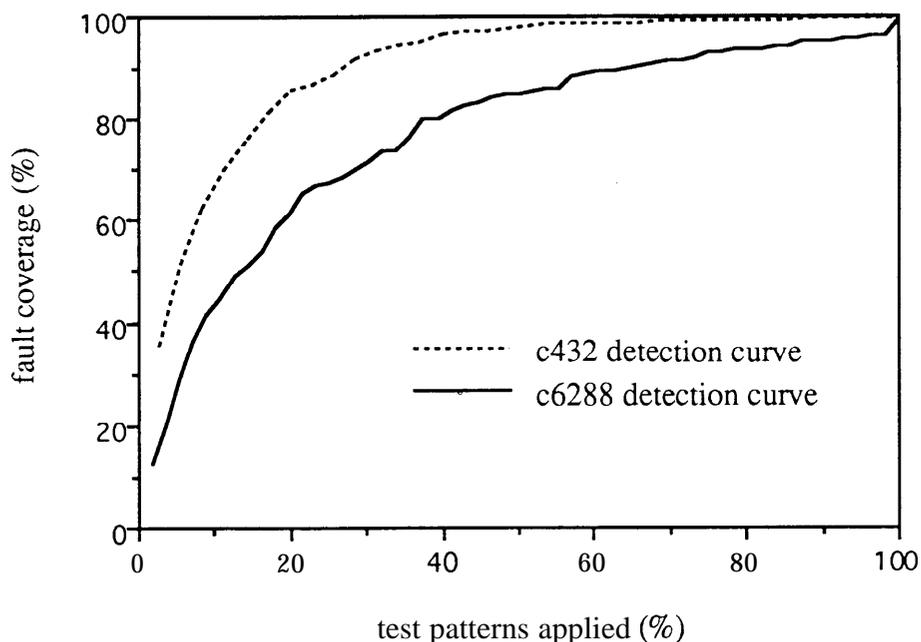


Figure 5. Detection curves for c6288 and c432

Next, the fault dropping effect is related to the test set applied. Deterministic test sets specifically target undetected faults whereas random test sets do not. In order to get a high fault coverage near 100%, many more random test patterns must be used. Generally speaking, the detection curve of a random test set increases faster than that of a deterministic test set, so the speedup of a random test set is higher. For example in Fig. 6, two different test sets are applied to c1908 benchmark circuit and two different detection curves are obtained. When the random test set

is used, notice that after 10% of test patterns are applied, the fault coverage goes up to 90%. While after the same percentage of deterministic test patterns are applied, the fault coverage goes up to only 65%. This observation explains why the **speedup** is bigger for random test sets as mentioned in Sec. 2.

Finally, the simulator used is another factor that affects the **speedup**. There is no simple relation between them. The **speedup** is influenced by the algorithm that a particular simulator is based on. Simulators have differing amounts of overhead that is unaffected by fault dropping.

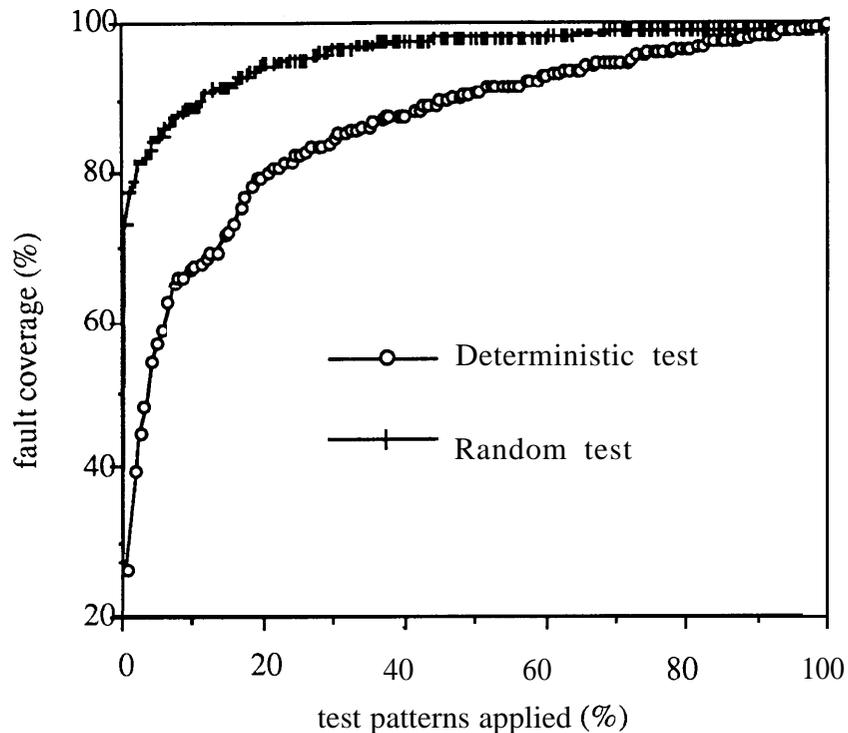


Figure 6. Detection curves of c1908 for each test set

5. CONCLUSIONS

Fault simulation times with and without fault dropping were presented for three simulators using both a deterministic and a random test set. The results show that fault dropping speeds up fault simulation making it from a few times up to tens of times faster. This gives a measure of the extra simulation time required to generate a complete fault dictionary, thereby giving some indication of the practicality of applications which require a complete fault dictionary. The results also show that the **speedup** due to fault dropping is much greater for random test sets than for deterministic test sets.

ACKNOWLEDGMENTS

The authors would like to thank Samy Makar, LaNae Avra, Piero Franco, and Prof. Xingning Xu for their valuable comments. This work was supported in part by the Innovative Science and Technology Office of the Strategic Defense Initiative Organization and administered through the Office of Naval Research under Contract No. N00014-92-J-1782, and by the National Science Foundation under Grant No. MIP-9 107760.

REFERENCES

- [Abramovici 90] Abramocivi, M., M.A. Breuer, and A.D. Friedman, " Digital System Testing and Testable Design," **Computer Science Press**, 1990, New York, pp. 541-548.
- [Brglez 85] Brglez, F., P. Pownall, and R. Hum, "Accelerated ATPG and Fault Grading Via Testability Analysis," **Proc. of ISCAS 85**, 1985, pp. 695-698.
- [Gai 9 13 Gai, S., and P.L. Montessoro, "The Fault Dropping Problem in Concurrent Event-Driven Simulation," **IEEE Transaction on Computer-aided Design**, Vol. 10, No. 8, Aug. 1991, pp. 968-97 1.
- [Lee 91] Lee, H.K. and D.S. Ha, "An Efficient, Forward Fault Simulation Algorithm Based on The Parallel Pattern Single Fault Propagation," **Proc. of International Test Conference, 1991**, pp. **946-954**.
- [Lee 92] Lee, H.K., and D.S. Ha, "HOPE: An Efficient Parallel Fault Simulator for Synchronous Sequential Circuits," **Design Automation Conference, 1992**, pp. 336-340.
- [Pomeranz 92] Pomeranz, I., S.M. Reddy, and R. Tangirala, "On Achieving Zero Aliasing for Modeled Faults," **Proc. of European Conference on Design Automation, 1992**, pp. 291-299.
- [Son 85] Son, IS., "Fault Simulation with The Parallel Value List Algorithm," **VLSI System Design**, Dec. 1985, pp. 36-43.
- [Song 92] Song, O., and P.R. Menon, "Acceleration of Fault Propagation Trace-Based Fault Simulation of Combinational Circuits," **IEEE International Symposium on Circuit and System , 1992**, pp. 1117-1 120.
- [Takahashi 91] Takahashi, N., N. Ishiura, and S. Yajima, "Fault Simulation For Multiple Faults Using Shared BDD Representation of Fault Sets," **IEEE International Conference on Computer-Aided Design, 1991**, pp. 550-553.