AN INVESTIGATION OF DEFECT DETECTION USING RANDOM DEFECT EXCITATION AND DERTERMINISTIC DEFECT OBSERVATION IN COMPLEX INTEGRATED LOGIC CIRCUITS

A Senior Thesis

By

Jennifer Dworak

1997-98 University Undergraduate Research Fellow

Texas A&M University

Group: Engineering I

An Investigation of Defect Detection using Random Defect Excitation and Deterministic Defect Observation in Complex Integrated Logic Circuits

by

Jennifer Dworak

Submitted to the
Office of Honors Programs and Academic Scholarships
Texas A&M University
in partial fulfillment of the requirements for

1997-1998 UNIVERSITY UNDERGRADUATE RESEARCH FELLOWS PROGRAM

April 16, 1998

Approved as to style and content by:

M. Ray Mercer
Dr. M. Ray Mercer (Faculty Advisor)
Department of Electrical Engineering

Susanna Finnell
Susanna Finnell, Executive Director
Honors Programs and Academic Scholarships

Fellows Group: Engineering I

Many thanks are extended to Dr. M. Ray Mercer, Mike Grimaila, and Dr. Jaehong Park for all of their valuable help, knowledge, and advice.

Table of Contents

Abstract

Whenever integrated circuits are manufactured, a certain percentage of those circuits will be defective. Defective circuits present problems for both the manufacturers who wish to maintain a good reputation with their customers and the consumers who depend upon the correct operation of the products they buy. Thus, testing must be done to detect which parts are defective so that they are not sold to unwitting consumers. Most current testing methods involve generating test patterns that will detect single stuck-at faults. Unfortunately, however, the single stuck-at fault model cannot adequately describe all of the potential defects that may occur. The requirements for exciting a fault vary depending upon the specific model (stuck-at, bridge, etc.) being used, but the observation of the fault always requires that the erroneous logic value be propagated to a primary output. The proposed new method of automatic test pattern generation involves deterministically observing all of the sites in the circuit as many times as possible while randomly exciting the defects which may occur. This research demonstrates the importance of site observation on the detection of defects and shows some of the inefficiencies and shortcomings of the current stuck-at fault ATPG.

Introduction

The Need for Testing and ATPG

In any large scale manufacturing process, a certain percentage of the final products will be defective and not meet the predefined specifications. For example, a company which makes compact disks may find that some of their CDs won't play because they are scratched. A corporation which makes clothes may find that some of the seams were not sewn correctly or that the material used for their shirts was stained. Regardless of the type of product being sold and what kinds of defects may occur during production, it is in the company's best interest to find which items do not measure up to standards so that they are not sold to unwitting consumers. Any company which does not filter out the faulty products will find their repeat business shrinking as customers choose to buy from a corporation that is more reliable. Thus, defect detection is a critical step in any manufacturing process.

The manufacture of complex integrated logic circuits is no exception. Many types of defects may occur when producing integrated circuits. Some defects are global in nature and are incredibly easy to spot [KAPU92]. For example, a large scratch may cut across the entire silicon wafer. In that case, it is obvious that the chips made of scratched silicon should not be sold to the company's customers. Unfortunately, however, many defects affect only a small physical area and cannot be discovered by a quick visual inspection. These defects are called point defects [KAPU92], and they are often caused by contamination during fabrication [OSBU88]. For example, a small piece of conducting material may erroneously connect two points which should not be connected. Part of the circuit could be permanently shorted to ground, or two points which should be connected might not be. Because these and many other potential problems are

unlikely to be found visually, other methods must be used to separate the "good" chips from the defective or "bad" ones.

One difficulty involved in testing integrated logic circuits for defects arises from the fact that the tester does not have access to all of the interior points within the circuit. The tester only has access to the primary inputs and the primary outputs. Thus, one way of discovering which circuits are defective involves applying the same logic values to the primary inputs of a circuit which is known to be good (meets the design specifications) and the circuit being tested. If the primary outputs match, nothing can be concluded except that for this particular set of primary inputs the two circuits give the same "answer." There is always a possibility that the circuit being tested could still show itself to be defective if a different set of primary inputs is applied. However, if the primary outputs of the circuits show different values, the circuit being tested is known to be defective and can be thrown away. These two situations are demonstrated in Figure 1. The primary inputs are located on the left of each circuit, and the primary outputs are located on the right.

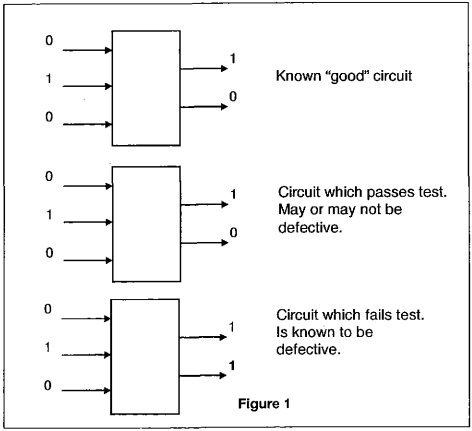


Figure 1

Because circuits which pass the test for a given set of primary inputs may still be defective for a different set of input logic values, one of the most obvious ways of ensuring that no defective parts are sold is to test each circuit for every possible input combination. Unfortunately, this is not practical for most real-life circuits because the number of possible input combinations is too big. Since each primary input must have one of two possible values (0 or 1) the number of possible input combinations is equal to 2^n where n is the number of primary inputs [Kapur92]. Thus, for a circuit with only 30 primary inputs, the number of possible input combinations is approximately one billion. Testing each circuit a billion times is too time-consuming and too expensive to be reasonable in an industrial setting, so methods must be devised to determine which input combinations are most likely to detect defective circuits.

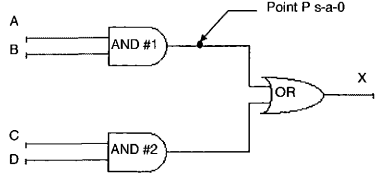
Currently, the input combinations (also called test vectors) used to test logic circuits are chosen through algorithms which perform automatic test pattern generation, or ATPG. Certain types of defects are modeled as faults, and the ATPG algorithm searches for test vectors which will detect those faults [WANG95] by creating a difference at the primary outputs between the good and the bad circuits. Current methods of ATPG chose the appropriate test vectors using deterministic excitation of the chosen faults and deterministic observation of the points where those faults may occur. Most ATPG algorithms try to detect single stuck-at faults. The stuck-at fault model was originally published by R. D. Eldred in 1959 [ELDR59].

Detection of Single Stuck-at Faults through Deterministic Observation and Excitation

Single stuck-at faults are models in which a point in the circuit is permanently "stuck" at either a logic one or a logic zero regardless of the value determined by the rest of the circuit's logic. In a real circuit, a point in the circuit could be "stuck-at zero" (s-a-0) if the point was

erroneously shorted to ground. A point might be "stuck-at-one" (s-a-1) if the point was permanently shorted to the power supply, Vcc. Thus, stuck-at faults are models of defects which might actually occur in real circuits. Current methods of ATPG use deterministic excitation and deterministic observation to find test vectors which will detect stuck-at faults.

Deterministic excitation involves ensuring that there is a difference at the point where the defect occurs between the good circuit and the bad circuit. Thus, the primary inputs must be chosen so that a *logic one* should be present at a point s-a-0 and a *logic zero* should be present at a point s-a-1. For an example of deterministic excitation, see Figure 2. If a test is to be devised which will detect point P s-a-0, the output of AND gate #1 should be a logic one in the good circuit. In order for this to occur, primary input A must be equal to one and primary input B must be equal to one.



Truth Table for AND Gate

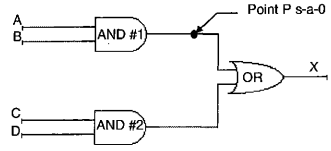
In1	In2	Out
0	0	0
0	1	0
1	0	0
1	1	1

To excite P s-a-0, choose A=1 and B=1.

Figure 2

Deterministic observation chooses the values of primary inputs which will propagate the value of the point in question to a primary output. This propagation to a primary output is necessary because the tester has no access to the interior points of the circuit. Thus, deterministic observation ensures that any difference between the good and bad circuits which occurs at the

actual location of the fault can be observed by the tester at the primary outputs. For an example of deterministic observation, see Figure 3. In order to observe the value of point P at the primary output X, point P must determine the output of the OR gate. In order for this event to occur, the other input to the OR gate must be a zero. Obviously, this means that the output of AND gate #2 must be a zero. Thus, either C must equal zero or D must equal zero to deterministically observe the fault P s-a-0.



Truth Table for OR gate

In1	In2	Out
0	0	0
0	1	1
1	0	1
1	1	1

To propagate point P to the primary output X choose C=0 or D=0

Figure 3

Thus, in order to *detect* point P s-a-0 in the circuit in Figure 3 any of the following test vectors will suffice:

A	B	C	D
1	1	0	0
1	1	0	1
1	1	1	0

Deterministic Observation and Excitation of Bridge Faults

Unfortunately, all defects cannot be modeled as single stuck-at faults. One example of this situation is a bridge defect. A bridge defect occurs when two points in the circuit are erroneously connected. Depending upon the type of connection, it is possible that either an AND bridge or an OR bridge may be formed. If an AND bridge is formed, both connected points are forced to the

lowest value dictated by the circuit's logic at either point. For example, if there is an AND bridge between point P and point Q, and the circuit's logic dictates that point P should be at a logic zero while point Q should be at a logic one, both P and Q will be at logic zero in the defective circuit. If an OR bridge is formed, both points are forced to the highest value dictated by the circuit's logic at either point. For example, if an OR bridge exists between point P and point Q, and the circuit's logic dictates that point P should be at a logic zero while point Q should be at a logic one, both P and Q will be at logic one in the defective circuit.

In order to deterministically excite a bridge fault, the primary inputs must be chosen so that the two points connected by the bridge are at different logic levels. Since the bridge will force both points to the same logic level, a difference will then exist between the good and bad circuit. For an example of a circuit containing a bridge fault, see Figure 4.

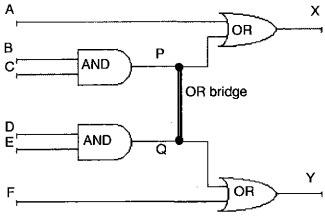


Figure 4

Assume that we want to find a test which will excite the fault which occurs when an OR bridge connects point P to point Q. This test will need to set point P to logic one and point Q to logic zero or set point P to logic zero and point Q to logic one. The circuit's logic will set point P to a logic one if both B=1 and C=1. Then, in order to set point Q equal to zero, either D or E

must equal zero. On the other hand, the circuit's logic will set point P to zero if B=0 or C=0, and point Q should be set to one when D=1 and E=1. Thus, the chart below shows the values of the primary inputs which will excite an OR bridge between point P and point Q.

A	B	C	D	E	F
X	0	0	1	1	X
X	0	1	1	1	X
X	1	0	1	1	X
X	1	1	0	0	X
X	1	1	0	1	X
X	1	1	1	0	X

Note: The X's signify "don't care" values where either a one or a zero will be acceptable.

Once the bridge fault has been excited, deterministic observation requires that the point at the incorrect logic value must be observed at the primary output. Therefore, if the circuit's logic dictated that point P should be equal to zero while point Q should be equal to one, point P must be observed since the OR bridge will force both P and Q to logic one, and P will be in error in the defective circuit. In order for this to occur, point P must determine the output of the OR gate connected to primary output X. Primary input A must be set equal to zero. However, if the fault was excited by setting point P equal to one and point Q equal to zero, point Q must be observed at a primary output to detect the error. In this case, point Q must determine the output of the OR gate connected to primary output Y, and primary input F must be set equal to zero.

The input vectors which will deterministically excite and observe an OR bridge connecting point P with point Q are listed in the table on the next page.

A	B	C	D	E	F
0	0	0	1	1	X
0	0	1	1	1	X
0	1	0	1	1	X
X	1	1	0	0	0
X	1	1	0	1	0
X	1	1	1	0	0

The Problem with Current Single Stuck-at Fault ATPG

Unfortunately, no single fault model will encompass all types of defects [WANG95], and it is impossible to know what types of defects are present in a circuit before they are found. Yet, currently most ATPG algorithms search for tests which will detect single stuck-at faults. Once a test is found for a stuck-at fault, other tests which would also detect that fault are no longer considered. There is a possibility that a test vector for a modeled stuck-at fault will fortuitously detect a non-modeled defect in the actual circuit, but there is no guarantee, and the quality of the test set depends upon how many non-targeted defects can actually be detected [BUTL90] [BUTL91a] [BUTL91b]. Unfortunately, it is almost certain that some of the discarded tests are the only tests which will detect some non-targeted defects, and this fact limits the effectiveness of the test set. However, regardless of the type of defect involved, the point where the defect occurs must be observed at the primary output for detection to take place. This fact can be seen by examining the conditions for deterministic observation of stuck-at faults and bridge faults noted earlier. In each case, the value of the point in the circuit which had been forced to an incorrect logic value had to be propagated to a primary output. Yet, the conditions necessary for excitation were considerably more varied depending upon the type of fault being targeted.

When single stuck-at fault ATPG is used, each point in the circuit is observed at least twice, once for the point stuck-at one and once for the point stuck-at zero. This research has

shown that (1) observation of points in the circuit is crucial to the detection of non-targeted defects, and the number of defects escaping detection decreases dramatically as the number of observations increases, (2) current algorithms which detect single-stuck at faults leave considerable room for improvement in the average probability of observation of points in the circuit, and (3) an algorithm which takes into account the depth of points from the primary outputs and the probability of observation of those points will be able to more efficiently allocate resources and increase the number of observations of points far from the primary outputs. As a result, a better algorithm for ATPG would involve deterministic observation of as many sites as many times as possible with the probabilistic excitation of the defects which may occur.

Methods

The data analyzed in this research was collected using a fault simulator program written by Mike Grimaila. The program reads the circuit topography from a file and the test input vectors from another file. The primary outputs of the circuit are calculated for a good circuit and then for circuits containing faults. If the primary outputs of the faulty circuit and good circuit do not match, that fault is said to be detected. Because the program checks for the detection of each point in the circuit being s-a-0 and s-a-1, whether or not each point in the circuit has been observed for a specific input vector can also be determined. Excitation of either a s-a-0 fault or a s-a-1 fault is assured because each point in the circuit must be at either a logic zero or a logic one. Thus, if either a s-a-0 or a s-a-1 fault is detected, that point in the circuit must have been observed, and if neither stuck-at fault was detected, the point was not observed for that input vector. Therefore, information about the observation of circuit nets is also listed.

The circuits used in this analysis are benchmark circuits known as C432 and C499 published by F. Brglez and H. Fujiwara [BRGL85]. These circuits have 432 and 499 sites respectively where a defect could possibly occur. C432 contains 36 inputs, 7 outputs, 40 inverters, and many other gates including 4 AND gates, 79 NAND gates, 19 NOR gates, and 18 XOR gates. C499 is a slightly larger circuit. It contains 41 inputs, 32 outputs, 40 inverters, and 56 AND gates, 2 OR gates, and 104 XOR gates.

The test vectors for the vector file where generated either randomly or through an ATPG program called Atalanta which attempts to find vectors which will detect both the s-a-0 and s-a-1 fault at each point in the circuit. Data analysis and graphing were done using programs written in C, Excel spreadsheets, and gnuplot.

Results and Data Analysis

Average Observation Probability over All Sites

The proposed new method of ATPG involving deterministic observation with probabilistic excitation requires that each point in the circuit be observed as many times as possible. As a result, one of the first tasks was to find the average observation probability over all points in the circuit. One thousand simulations of circuit C432 were done with three different kinds of input vectors. The generation of input vectors varied so that random vectors and vectors generated by ATPG both with and without fault dropping were used. (When fault dropping occurs, the ATPG algorithm does not look for input vectors to detect faults found by previous tests[WANG96]. As a result, the number of vectors generated is less. However, the ATPG algorithm was run again with a different seed value to generate the same number of vectors as those with fault dropping so

that the comparison would be valid.) The average observation probability over all points in the circuit was recorded after each simulation and used to create Graphs A1-A3 in the appendix.

It can be seen from these graphs that the curves are roughly bell-shaped and approach normal distributions. The distribution is also narrow in that there is not a great deal of variation between the largest and smallest observation probabilities. The probability of observation over all sites is approximately equal to 21% when random vectors are used, 24% when the input vectors are generated by ATPG without fault dropping, and 25% when the input vectors are generated by ATPG with fault dropping. Thus, there is a slight increase in observability over all sites when a set of ATPG generated vectors is used instead of randomly generated vectors. This increase is probably a result of the fact that the ATPG algorithm is actively trying to observe points in the circuit, and if 100% fault coverage is attained, each point in the circuit is guaranteed to be observed at least twice. In addition, the very slight increase in probability of observation which occurs when fault dropping is used (as opposed when no fault dropping is used) is probably a result of the fact that the same number of input vectors were used in both simulations. Thus, points in the circuit which were difficult to observe were targeted more often when the vectors were generated with fault dropping. It is also interesting to note that the average observation probability over all circuit sites is not very large, and there is room for improvement of observation probability if an algorithm was developed to increase that probability. In contrast, if the average observation probability had already been 85% or 90%, the potential gains from changing the ATPG algorithm would have been small.

Observation Probability by Level

The data discussed in the previous section involved the average probability of observation over all sites in the circuit. This average probability is important, but it can also be misleading. Some sites are observed much more often than others. For example, the primary outputs are always observed simply because they are primary outputs, and the tester has direct access to them. In contrast, a point in the interior of the circuit is nearly always less likely to be observed because the signal in question must propagate through several gates in order to reach the primary output. Because of this fact, the average probability of observation for circuit sites as a function of circuit depth was also investigated.

There is more than one way to assign a level to sites within a circuit. One involves counting the number of gates from the primary inputs to the point in question. This method is referred to as forward leveling. An alternative method involves counting the number of gates from the primary outputs to the point in question. This count will give the back level of the point in question. Average probabilities of observation for C432 and C499 were plotted versus both the forward and back level. The simulations for C432 were done with random vectors, with fault dropping, and without fault dropping. The simulation for C499 was done without fault dropping. Each probability graphed is the average of the observation probabilities for circuit sites at a particular level averaged over 1000 simulations. These graphs can be found in the appendix as A4 through A11.

Graphs A4 through A7 show the probability of observation versus forward level. The forward level was taken as the maximum number of gates from the primary inputs to the point in question. A number of conclusions can be drawn from the graphs for C432 (A4-A6). First, the use of random vectors as opposed to vectors generated by ATPG has little influence on the

observation probabilities. Only levels one through five show any discernible improvement by running the ATPG algorithm, and that improvement is not incredibly large. In the best case, the observation probability appears to double, and in most cases, any increase is considerably less. Thus, once again, using a current ATPG algorithm which searches for tests that detect single stuck-at faults does not dramatically increase the probability of observation above that reached by random vectors

Another conclusion which can be drawn from these graphs is that no particular pattern emerges which would readily show where in the circuit the least observable points are located. C432 has very low observation probabilities followed by very high observation probabilities and C499 has fairly constant observation probabilities except in levels nine through thirteen where it decreases dramatically. One reason for this phenomenon is that this method of assigning level does not ensure that the primary outputs (which have an observation probability of one) are at the final level. Instead, they are allowed to skew the previous level averages with their high observation probability. Thus, this version of forward leveling is not an ideal method to draw any conclusions on probability of observation vs. circuit depth.

An alternative method is back leveling. The back level is defined as the minimum number of gates from the point in question to the primary output. The minimum number of gates is taken because this is the path through which the value of the point in question is most likely to propagate and reach a primary output. Therefore, this is the path most likely to determine the observation probability.

As can be seen from the graphs for C432 (A8-A10), this method yields much cleaner results than forward leveling. First, the number of levels has decreased. Second, a definite pattern has emerged — the probability of observation decreases in a roughly exponential fashion

until the primary inputs are reached where the probability of observation increases. This trend makes perfect sense. All of the primary outputs are placed at level one and have a probability of observation of one. As the circuit depth increases (the point is farther from the primary output) the probability of observation decreases until points near the primary inputs are reached. The probability of observation increases near the primary inputs because there are more fan-out gates in that part of the circuit. These fan-out gates allow more path choices for the propagation of the value of the point in question and thus increase the probability of observation. It should also be noted that, once again, the use of an ATPG algorithm makes little difference in the probability of observation in comparison to the random vectors case.

This method of back leveling was also used to analyze the observation probabilities from C499 simulations without fault dropping. This graph is not as clean as the one for C432 because the probability of observation does not follow the same almost constant decrease as the back level increases. This phenomenon is a result of circuit topology. For example, all of the gates at level 4 have an unexpectedly high probability of observation because their outputs are connected to fan gates. Thus, the same phenomenon which occurs near the primary inputs of most circuits also occurs at level 4, leading to a jump in the probability of observation.

The average probability that a logic value was a one (ones density) was also plotted for each back level. These graphs can be found as A12 through A15 in the appendix. The graphs from C432 simulations with random vectors and with vectors generated by ATPG are A12 through A14. These graphs show that although there is some variation with respect to level, the ones density does not appear to show a true trend with respect to depth from the primary outputs. The average ones density is also greater than 50% for the most part, and this situation is probably a result of the large number of NAND gates present in C432. Two-input NAND gates have an

output of one approximately 75% of the time if the input vectors are random, and the high prevalence of NAND gates in C432 could therefore be expected to skew the logic values in favor of a one.

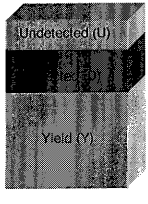
Ones density was also plotted for the C499 simulation without fault dropping (See A15). Although there appears to be slightly more variation in the average ones density compared to the results for C432, once again there doesn't appear to be a clear trend with respect to level. (Note that the scale is different for the C499 graph, and this accentuates the appearance of the variation. In reality, the variation is not significantly different from the C432 case.) However, unlike the C432 case, the average ones density is less than 50%. This phenomenon is a result of the types of gates present in the circuit. C499 contains more AND gates than C432, and several of these AND gates have 4 or 5 inputs. These several-input AND gates are especially likely to have an output of zero, and both these zeros and their propagation through the circuit serve to decrease the average ones density.

An important conclusion can be drawn from all of the graphs plotted as a function of circuit depth from the primary outputs: the probability of observation varies significantly with back level and tends to decrease as depth from the primary outputs increases. However, ones density does not vary as greatly with respect to circuit depth and does not follow a general pattern. Since ones density is related to the excitation of faults, excitation is not highly dependent on circuit depth. On the other hand, observation is very dependent upon circuit depth. Current methods of ATPG divide resources fairly evenly across circuit sites. Thus, the same amount of importance is attached to finding patterns which will observe a primary output and detect a stuck-at fault there as is attached to finding patterns which will observe points deep within the circuit where observation is much less likely to occur. This uniform targeting of circuit sites is a waste of

resources. Since points near the primary outputs are likely to be observed anyway, a better distribution of resources would involve targeting those sites in the interior of the circuit for which the probability of observation is much less. One way to determine which sites should be targeted more often would be to choose those sites on the basis of back level. For example, in C432 levels seven and eight have very low probabilities of observation, and more time and computing resources could be devoted to observing these points. Thus, an algorithm which generates tests for deterministic observation with probabilistic excitation should concentrate on finding those tests which allow the observation of the least observable points in the circuit (levels 7 and 8 for C432).

The Effect of Observation on Defect Level

Although the previous two sections have demonstrated that current methods of ATPG do not significantly increase the probability of observation over all points in the circuit and do not concentrate resources where they are needed the most, these two conclusions would be unimportant if observation was not an essential part of detecting defects. The true test of any ATPG algorithm is how many defective parts will be erroneously labeled as "good" after all the input vectors generated have been tried [BUTL91b]. Thus, a better set of test vectors will allow fewer defective parts to escape the testing process. This figure of merit which shows the false pass ratio is known as the defect level.



For the calculations of defect level, the defects were assumed to be disjoint. As the testing process progresses, each part is either good, known to be defective, or an undetected defective part. These situations can be seen in Figure 5.

$$Y + D + U = 1.0$$

Figure 5

The block represents all of the parts which have been produced in a manufacturing process. After the process has been completed, some portion of the parts will meet specifications and not be defective. This fraction of the parts which is not defective is called the yield. The rest of the parts are defective. As testing progresses, some of the defective parts are detected while the rest remain undetected. The parts which are known to be defective are thrown out, and the problem arises from the undetected defective parts—the parts which erroneously passed the test. The defect level is defined as

$$DL = \frac{U}{Y + U}$$

where Y+D+U=1.0. Thus, the defect level is defined as the fraction of the parts which remain after testing that are still defective.

Because defect level is the most important parameter for evaluating the effectiveness of testing processes, analysis was done regarding how the defect level for C432 decreases as more tests are run and more points are observed. For these simulations, bridges were inserted into C432 as surrogates and tests devised to detect single-stuck at faults were run. The bridges were the model for non-targeted defects in a circuit, and the detection of the bridges was used as an indication of how the defect level would decrease in an actual testing process.

For the calculations of defect level, a yield of 0.9 was assumed, and the number of bridges which had been detected versus the total number of bridges which had been inserted into the circuit was counted. Graph A16 in the appendix shows how the defect level decreases after each test vector is run. The pass number increases by one with each test vector. The value for DL plotted is the average value over several simulations, and the error bars show plus or minus two

standard deviations above and below that average. As should be expected, the defect level is equal to 0.1 when no tests have been run (pass number=0) because no defects have yet been detected. As the pass number increases, the defect level decreases. However, the rate of decrease also decreases as the number of vectors run increases suggesting that, at some point, very little improvement returned for the effort. This decrease in the rate is a result of the fact that the remaining surrogates are more difficult to find.

The defect level was also plotted versus the total number of observations over all circuit sites for the C432 simulations. This plot can be found as A17 in the appendix. The graph clearly shows that the defect level is a function of the number of observations and strongly suggests that increasing the number of observations of points in the circuit will dramatically decrease the defect level. Thus, it clearly supports the proposal of deterministic observation with probabilistic excitation.

Conclusion

The new proposed method of ATPG shows great promise because of the high correlation between observation of points in the circuit and defect detection. Current ATPG algorithms did not significantly improve the observation of sites when compared to vectors generated randomly. This result indicates that there is considerable room for improvement in the observation of circuit sites which may be exploited by an ATPG algorithm designed to observe as many points in the circuit as many times as possible.

Current ATPG algorithms also poorly allocate resources. Equivalent importance is attached to finding tests which will deterministically excite and observe stuck-at faults near the primary outputs and tests which will detect those faults deep within the circuit. The circuit sites

near the primary outputs are generally much more likely to be observed, and there are often more tests available which will detect those faults. A better allocation of resources would involve spending more effort trying to deterministically observe the points in the interior of the circuit which do not have a high probability of observation. The points near the primary output will often be observed simultaneously, and the overall probability of observation should increase.

One way of determining which circuit sites should be targeted for deterministic observation more often would be to assign priorities on the basis of circuit depth from the primary outputs. An alternative would be to do fault simulation and actually target the least observed sites specifically. The advantage of the level method is that a running tally of observations from circuit simulation would not have to be made. The disadvantage is that circuit topology (the kinds of gates, number of inputs to those gates, and presence of fan-outs) could mean that sites with a high probability of observation may be located at a level whose sites are assumed to be rarely observed, and sites with a low probability of observation may be found at a level with a generally high observation probability. Thus, resources could once again be misallocated.

Finally, this research has shown that the defect level is highly linked to the total observations of sites in the circuit. This indicates that increasing the number of observations will decrease defect level and allow fewer defective parts to reach consumers. Thus, deterministically observing as many sites in the circuit as many times as possible should decrease the defect level. Since defect level is the ultimate indicator of the success of the testing process, deterministic observation with random excitation should be an improvement over current methods of single stuck-at fault ATPG.

Future research will require devising tests specifically for deterministic observation with random excitation. These vectors can then be used to see at what point the random excitation of

defects is no longer sufficient to obtain a significant decrease in defect level. For example, it may be necessary to ensure that the ones density which results from the chosen test vectors is between a maximum and minimum level in order to ensure better excitation of defects. However, the research already completed has shown that the new proposed method of ATPG holds great potential.

Bibliography

[BRGL85] F. Brglez and H. Fujiwara, "A Neutral Netlist of 10 Combinational Benchmark Circuits and a Target Translator in FORTRAN," *Proc. IEEE Int. Symp. on Circ. Syst. (ISCAS)*, June 1985, pp. 695-698.

[BUTL90] K. M. Butler and M. R. Mercer, "The Influences of Fault Type and Topology on Fault Model Performance and the Implications to Test and Testable Design", *Proc. 27th ACM/IEEE DAC*, Orlando, FL, June 24-28, 1990, pp. 673-678.

[BUTL91a] K. M. Butler and M. R. Mercer, *Assessing Fault Model and Test Quality*, Kluwer Academic Publishers, 1991.

[BUTL91b] K. M. Butler and M. R. Mercer, "Quantifying Non-Target Defect Detection by Target Fault Test Sets," *Proc. of The European Test Conference*, Munich, Germany, April 10-12, 1991, pp. 91-100.

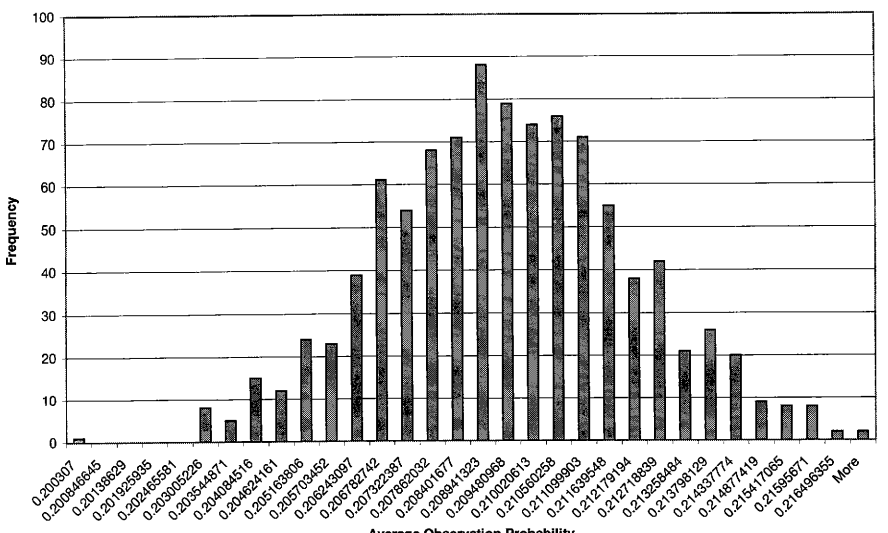
[ELDR59] R. D. Eldred, "Test Routines Based on Symbolic Logic Statements," *Journal ACM*, Vol. 6, no. 1, 1959, pp. 33-36.

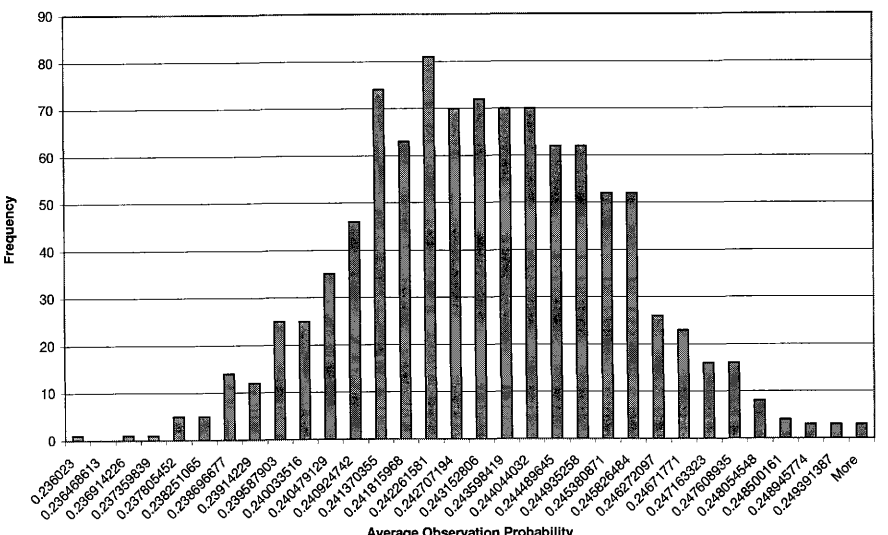
[KAPU92] R. Kapur, J. Park, and M. R. Mercer, "All Tests for a Fault are Not Equally Valuable for Defect Detection," *Proc. 1992 International Test Conference*, Baltimore, MD, September 20-24, 1992, pp. 762-769.

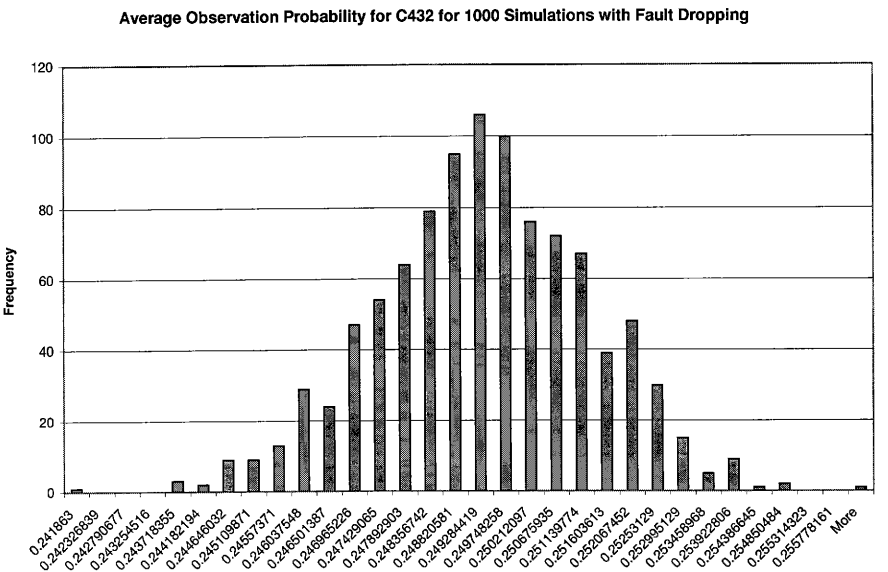
[OSBU88] C. M. Osburn, H. Berger, R.P. Donovan, and G.W. Jones, "The Effects of Contamination on Semiconductor Manufacturing Yield," *Journal of Environmental Sciences*, March-April 1998, pp. 45-57.

[WANG95] Li-C. Wang, M. Ray Mercer, and T. W. Williams, "On Efficiently and Reliably Achieving Low Defective Part Levels," *Proc. International Test Conference*, Washington D. C. 1995, pp. 616-625.

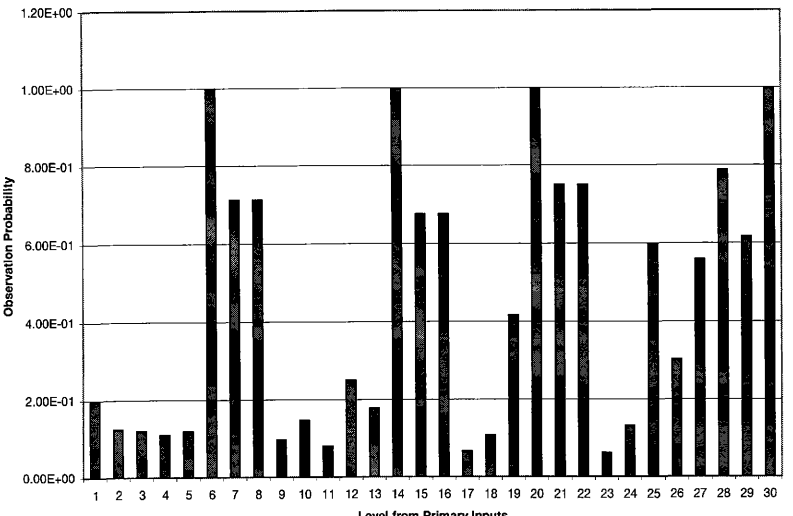
[WANG96] Li-C. Wang, M. Ray Mercer, and T. W. Williams, "A Better ATPG Algorithm and Its Design Principles," *Proc. International Conference on Computer Design,* Austin, Texas 1996, pp. 248-253.


Average Observation Probability for C432 for 1000 Simulations using Random Input Vectors
Frequency
100
90
80
70
60
50
40
30
20
10
0
0.200307
0.200846645
0.20138629
0.201925935
0.202465581
0.203005226
0.203544871
0.204084516
0.204624161
0.205163806
0.205703452
0.206243097
0.206782742
0.207322387
0.207862032
0.208401677
0.208941323
0.209480968
0.210020613
0.210560258
0.211099903
0.211639548
0.212179194
0.212718839
0.213258484
0.213798129
0.214337774
0.214877419
0.215417065
0.21595671
0.216496355
More
Average Observation Probability

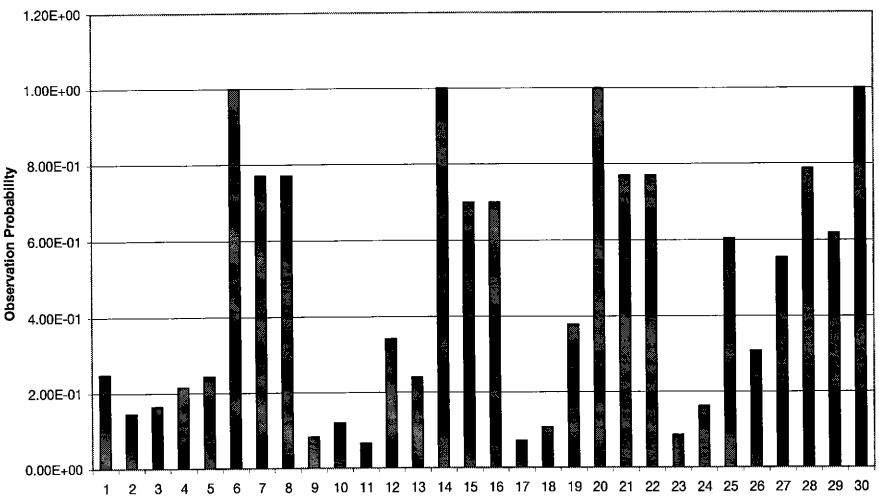

Average Observation Probabilities for C432 for 1000 Simulations without Fault Dropping
Frequency
90
80
70
60
50
40
30
20
10
0
0.236023
0.236468613
0.236914226
0.237359839
0.237805452
0.238251065
0.238696677
0.23914229
0.239587903
0.240033516
0.240479129
0.240924742
0.241370355
0.241815968
0.242261581
0.242707194
0.243152806
0.243598419
0.244044032
0.244489645
0.244935258
0.245380871
0.245826484
0.246272097
0.24671771
0.247163323
0.247608935
0.248054548
0.248500161
0.248945774
0.249391387
More
Average Observation Probability

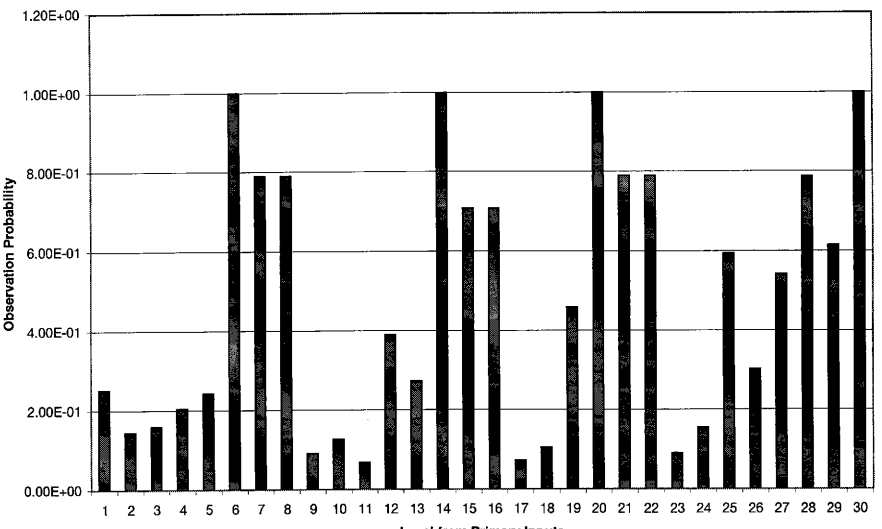

Average Observation Probability for C432 for 1000 Simulations with Fault Dropping
Frequency
120
100
80
60
40
20
0
0.241863
0.242326839
0.242790677
0.243254516
0.243718355
0.244182194
0.244646032
0.245109871
0.24557371
0.246037548
0.246501387
0.246965226
0.247429065
0.247892903
0.248356742
0.248820581
0.249284419
0.249748258
0.250212097
0.250675935
0.251139774
0.251603613
0.252067452
0.252531129
0.252995129
0.253459968
0.253922806
0.254386645
0.254850484
0.255314323
0.255778161
More
Average Observation Probability

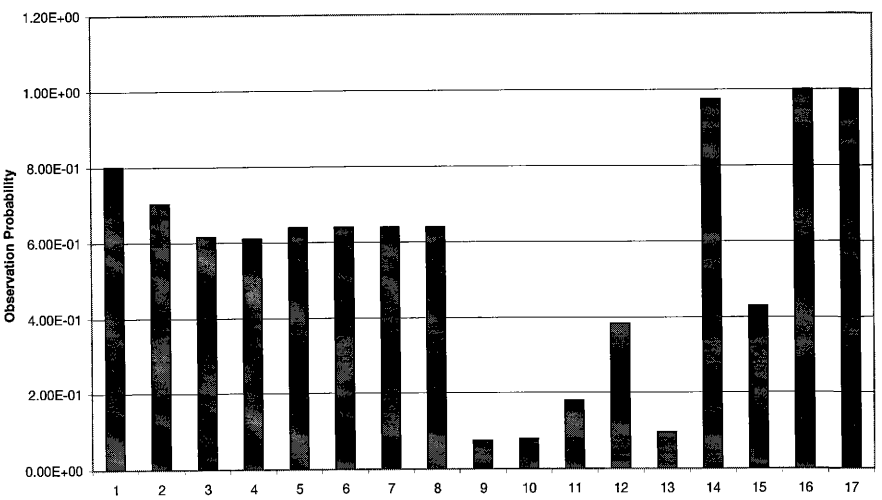
Average Observation Probability by Forward Level for C432 Simulation with Randomized Input Vectors

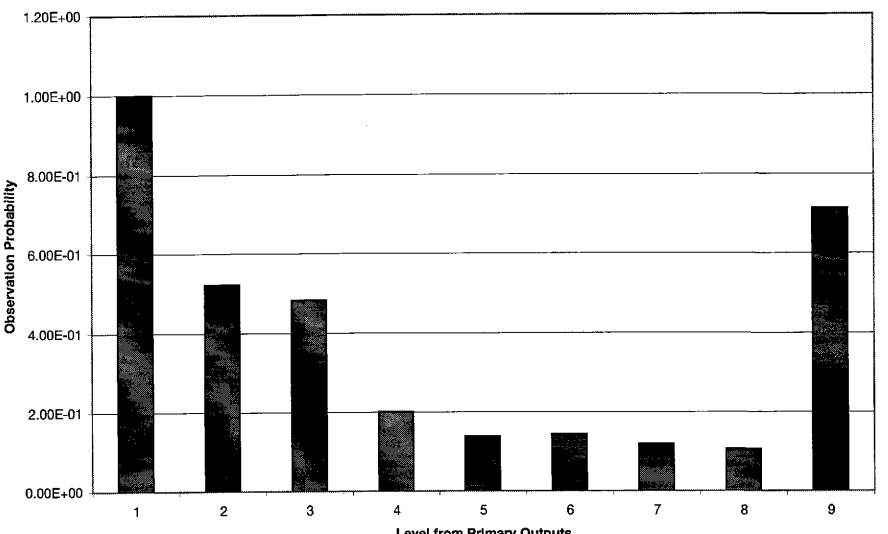


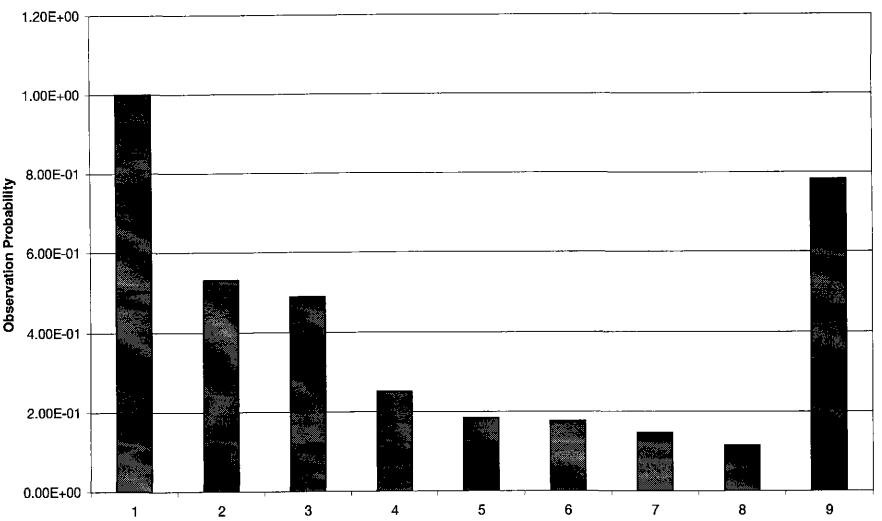
Average Observation Probability by Forward Level for C432 Simulation Without Fault Dropping

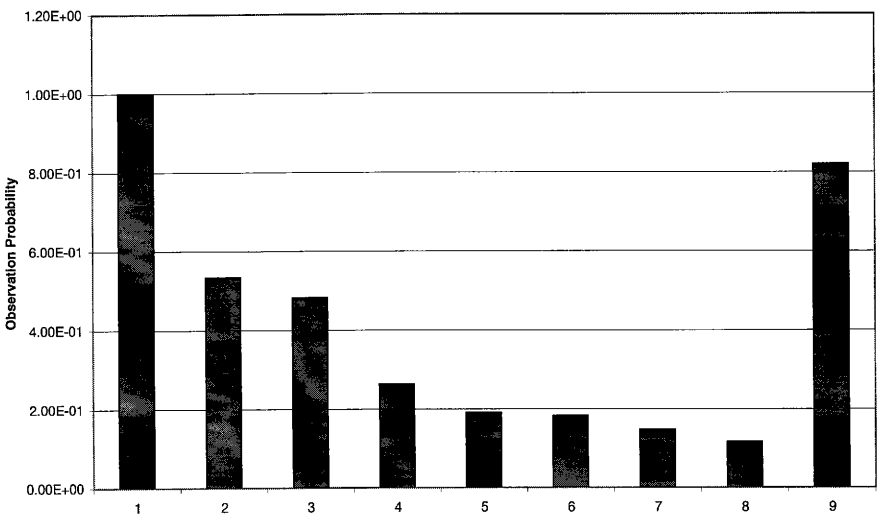


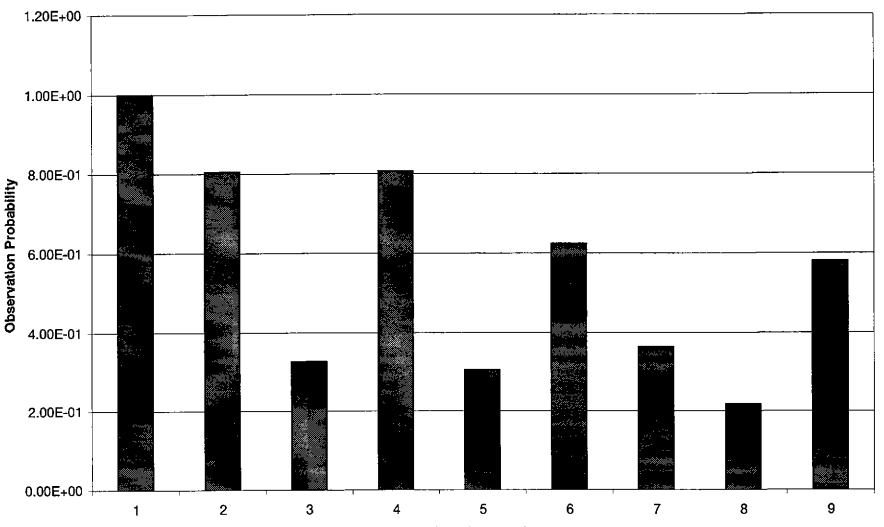

Average Probability of Observation by Forward Level for C432 Simulation with Fault Dropping
1.20E+00
1.00E+00
8.00E-01
6.00E-01
4.00E-01
2.00E-01
0.00E+00
Observation Probability
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
Level from Primary Inputs

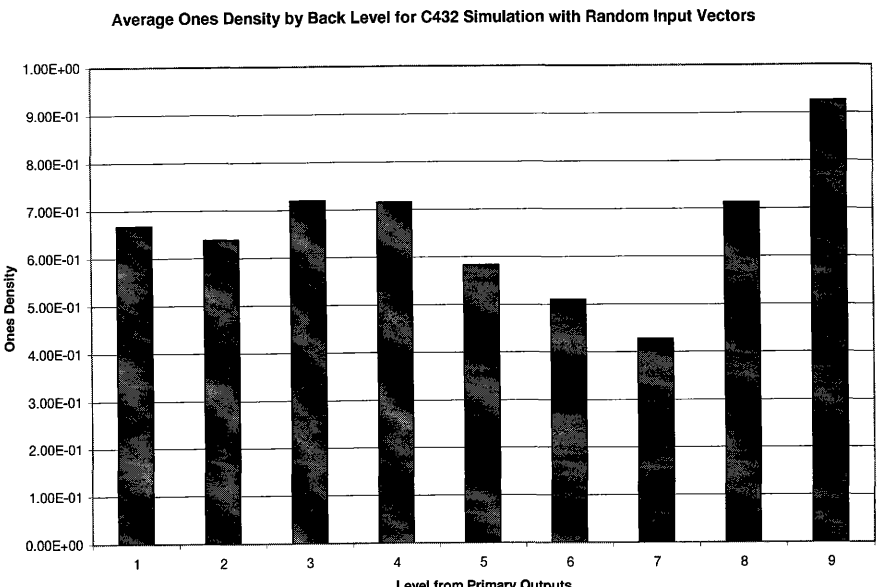

Average Probability of Observation by Forward Level for C499 Simulation without Fault Dropping
1.20E+00
1.00E+00
8.00E-01
6.00E-01
4.00E-01
2.00E-01
0.00E+00
Observation Probability
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
Level from Primary Inputs

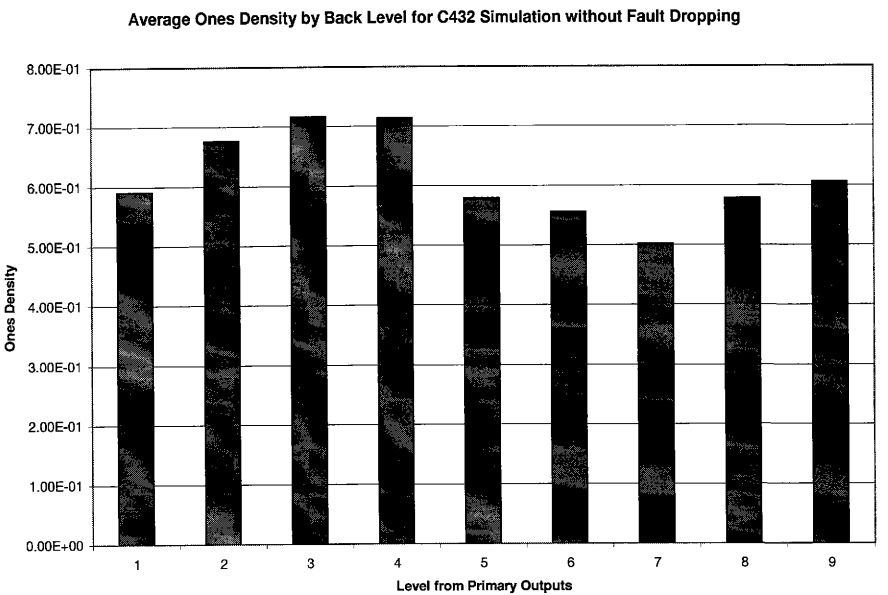

Average Probability of Observation by Back Level for C432 Simulation with Random Vectors
1.20E+00
1.00E+00
8.00E-01
6.00E-01
4.00E-01
2.00E-01
0.00E+00
Observation Probability
1
2
3
4
5
6
7
8
9
Level from Primary Outputs

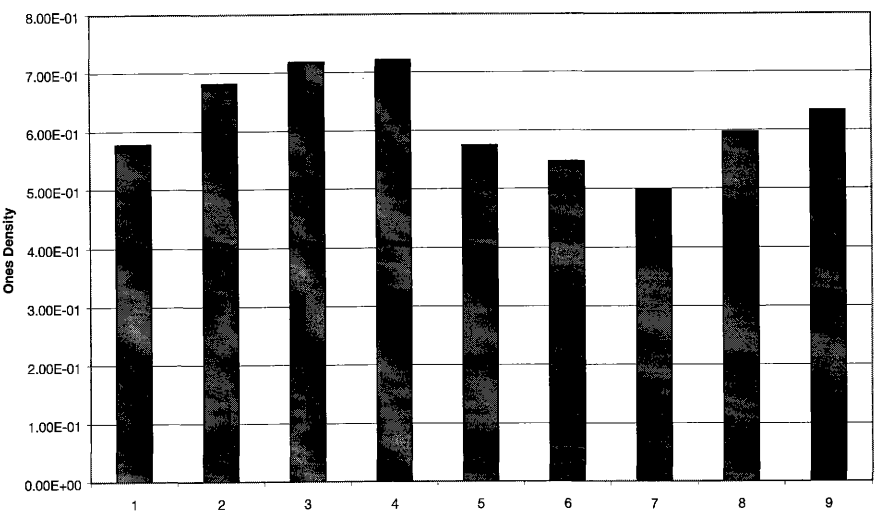

Average Probability of Observation by Back Level for C432 Simulation without Fault Dropping
1.20E+00
1.00E+00
8.00E-01
6.00E-01
4.00E-01
2.00E-01
0.00E+00
Observation Probability
1
2
3
4
5
6
7
8
9
Level from Primary Outputs

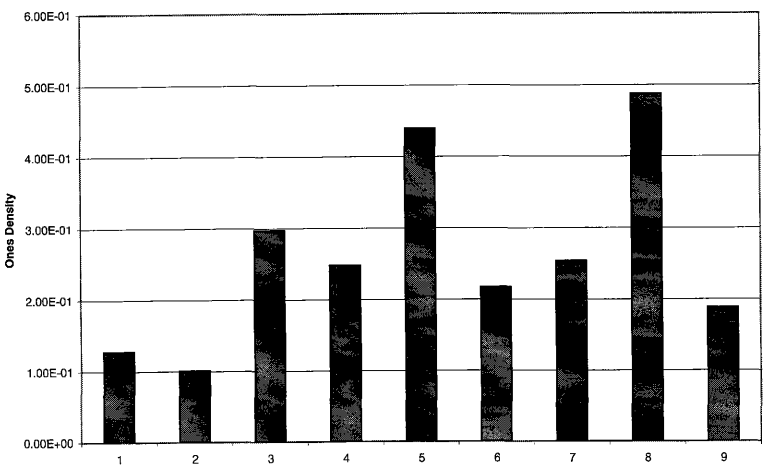

Average Probability of Observation by Back Level for C432 Simulation with Fault Dropping
1.20E+00
1.00E+00
8.00E-01
6.00E-01
4.00E-01
2.00E-01
0.00E+00
Observation Probability
1
2
3
4
5
6
7
8
9
Level from Primary Outputs

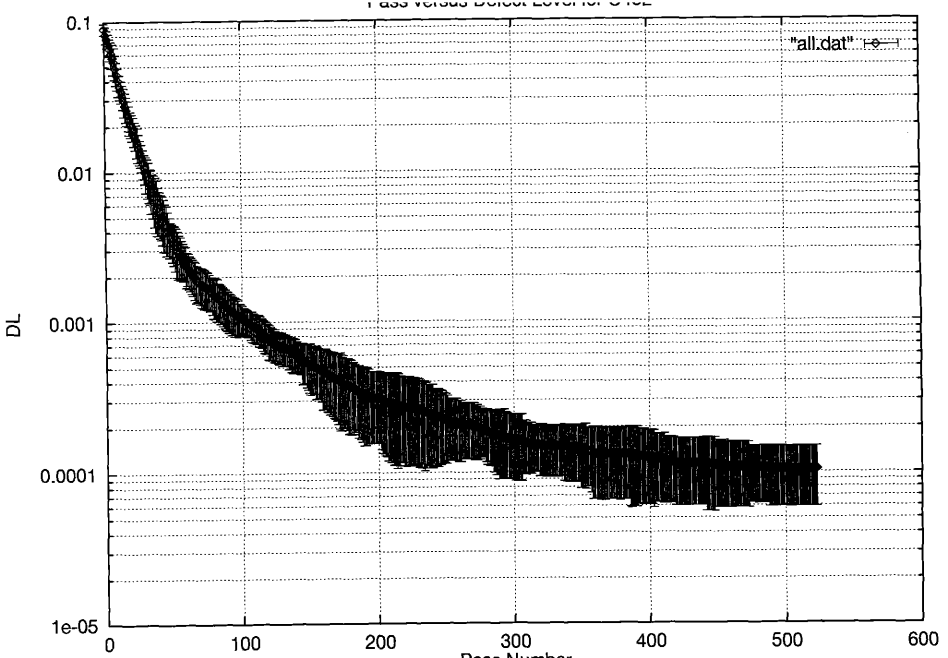

Average Probability of Observation by Back Level for C499 Simulation without Fault Dropping
1.20E+00
1.00E+00
8.00E-01
6.00E-01
4.00E-01
2.00E-01
0.00E+00
Observation Probability
1
2
3
4
5
6
7
8
9
Level from Primary Outputs

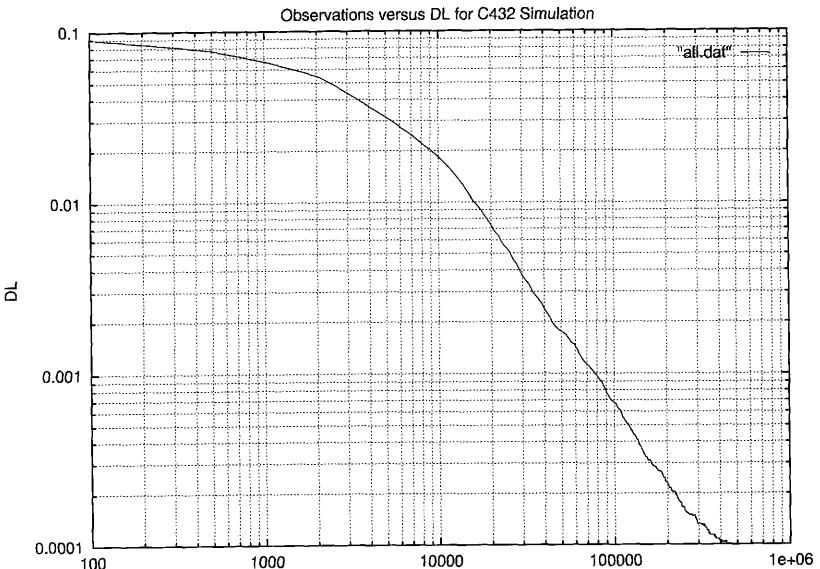

Average Ones Density by Back Level for C432 Simulation with Random Input Vectors
1.00E+00
9.00E-01
8.00E-01
7.00E-01
6.00E-01
5.00E-01
4.00E-01
3.00E-01
2.00E-01
1.00E-01
0.00E+00
Ones Density
1
2
3
4
5
6
7
8
9
Level from Primary Outputs


Average Ones Density by Back Level for C432 Simulation without Fault Dropping
8.00E-01
7.00E-01
6.00E-01
5.00E-01
4.00E-01
3.00E-01
2.00E-01
1.00E-01
0.00E+00
Ones Density
1
2
3
4
5
6
7
8
9
Level from Primary Outputs


Average Ones Density by Back Level for C432 Simulation with Fault Dropping
8.00E-01
7.00E-01
6.00E-01
5.00E-01
4.00E-01
3.00E-01
2.00E-01
1.00E-01
0.00E+00
Ones Density
1
2
3
4
5
6
7
8
9
Level from Primary Outputs


Average Ones Density by Back Level for C499 Simulation without Fault Dropping
6.00E-01
5.00E-01
4.00E-01
3.00E-01
2.00E-01
1.00E-01
0.00E+00
Ones Density
1
2
3
4
5
6
7
8
9
Level from Primary Outputs


"all.dat"
0.1
0.01
0.001
0.0001
1e-05
DL
0
100
200
300
400
500
600
Pass Number


Observations versus DL for C432 Simulation
"all.dat"
DL
0.1
0.01
0.001
0.0001
100
1000
10000
100000
1e+06
Obs.