MODELING DEFECTIVE PART LEVEL DUE TO STATIC AND DYNAMIC DEFECTS BASED UPON SITE OBSERVATION AND EXCITATION BALANCE

A Dissertation

by

JENNIFER LYNN DWORAK

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2004

Major Subject: Electrical Engineering

MODELING DEFECTIVE PART LEVEL DUE TO STATIC AND DYNAMIC DEFECTS BASED UPON SITE OBSERVATION AND EXCITATION BALANCE

A Dissertation

by

JENNIFER LYNN DWORAK

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Approved as to style and content by:

M. Ray Mercer
(Chair of Committee)

Narasimha Reddy
(Member)

Karen Butler-Purry
(Member)

Mark Holtzapple
(Member)

Chanan Singh
(Head of Department)

May 2004

Major Subject: Electrical Engineering

ABSTRACT

Modeling Defective Part Level Due to Static and Dynamic
Defects Based upon Site Observation and Excitation Balance. (May 2004)
Jennifer Lynn Dworak, B.S., Texas A&M University;
M.S., Texas A&M University
Chair of Advisory Committee: Dr. M. Ray Mercer

Manufacture testing of digital integrated circuits is essential for high quality. However, exhaustive testing is impractical, and only a small subset of all possible test patterns (or test pattern pairs) may be applied. Thus, it is crucial to choose a subset that detects a high percentage of the defective parts and produces a low defective part level.

Historically, test pattern generation has often been seen as a deterministic endeavor. Test sets are generated to deterministically ensure that a large percentage of the targeted faults are detected. However, many real defects do not behave like these faults, and a test set that detects them all may still miss many defects. Unfortunately, modeling all possible defects as faults is impractical. Thus, it is important to *fortuitously* detect *unmodeled* defects using high quality test sets.

To maximize fortuitous detection, we do not assume a high correlation between faults and actual defects. Instead, we look at the common requirements for all defect detection. We deterministically maximize the observations of the least-observed sites while randomly exciting the defects that may be present. The resulting decrease in defective part level is estimated using the MPG-D model.

This dissertation describes the MPG-D defective part level model and shows how it can be used to predict defective part levels resulting from static defect detection.

Unlike many other predictors, its predictions are a function of site observations, not fault coverage, and thus it is generally more accurate at high fault coverages. Furthermore, its components model the physical realities of site observation and defect excitation, and thus it can be used to give insight into better test generation strategies.

Next, we investigate the effect of additional constraints on the fortuitous detection of defects—specifically, as we focus on detecting dynamic defects instead of static ones. We show that the quality of the randomness of excitation becomes increasingly important as defect complexity increases. We introduce a new metric, called excitation balance, to estimate the quality of the excitation, and we show how excitation balance relates to the constant τ in the MPG-D model.

To my favorite teachers:

Roger and Dolores Dworak

and

Ray Mercer

ACKNOWLEDGMENTS

There are many people whom I would like to thank for their support and contributions to this work. First, I would like to thank my supervising professor, Dr. Ray Mercer. He is an outstanding teacher and researcher and friend, and I will always be grateful for the chance to learn from him. Because of him, I have learned not only more about testing and logic design, but also how to enjoy giving presentations and how to convey the most important ideas to others at conferences and in the classroom. I have learned that missing a plane will not bring about the end of the world, and most importantly, I have learned how much fun it can be to do research at a university with brilliant people.

I would also like to thank the members of my committee: Dr. A. L. Narasimha Reddy, Dr. Mark Holtzapple, and Dr. Karen Butler-Purry. They were always willing to help, and I appreciate all of their valuable insights and advice.

I am also grateful to Dr. Li Wang for his guidance and suggestions. He is a truly brilliant person, and I always enjoy discussing research with him.

I would also like to thank all of the other graduate students in our group for their help, questions, knowledge, and endless supply of random quotes. I would especially like to thank Professor Michael Grimaila for helping me to get started and for teaching me C. I would like to thank Sooryong Lee for his help with Gulf and with all sorts of data collection. Jimmy Wingfield's help with sByDDer was also invaluable, along with his patience and assistance with all of my bizarre computer problems. I would also like to thank Brad Cobb for always asking insightful questions, sharing his ideas, and listening to mine. These people, along with Justin Ray, Jason VanFickell, Nathan Mickler, Bryan Baer, Jason Wicker, Amy Wang, David Dorsey, Mike Trinka, and Theresa Huth have truly helped my research by giving me a chance to discuss

my ideas and to listen to their own ideas and suggestions. My experience at Texas A&M has been much richer for the chance to interact with them all.

I would also like to thank Carolyn Warzon for being such an efficient and knowledgeable secretary as well as a good friend. She always had the answers to all my questions and helped me in countless ways. I am very grateful for the chance to know her.

I would also like to thank Tammy Carda for all of her help, knowledge, and good advice. The students in the Electrical Engineering department are very lucky to have her.

I would also like to acknowledge the Electrical Engineering department at Texas A&M, including Dr. Singh and Dr. Huang, for their help and support. I am very grateful for the chance to have studied in a department with so many excellent professors and for the financial support that helped to make it possible.

I would also like to acknowledge the National Science Foundation, the Semiconductor Research Corporation (SRC), the Texas Advanced Technology Program, and IBM for their valuable support.

I would also especially like to thank Dr. Ken Butler and Dr. Bret Stewart at Texas Instruments for all of their suggestions and for giving us the opportunity to test our ideas on industrial manufactured circuits. I am also grateful to Dr. Tom Williams for all of his help.

I would also like to thank Dr. Don Ross for being such an excellent teacher for my first class on digital logic. It was because of Dr. Ross's class that I decided that I really wanted to be an electrical engineer, and it was through his recommendation that I came to work in this area under Dr. Mercer.

Finally, I would like to thank my family, especially my parents, my sisters Ruth Ann and Sarah, and my bother David, for all of their help and support. I am especially grateful to my parents for preparing me to meet the challenges of college and life. They have taught me so many things, and I will never be able to express how much their love and confidence in me has always meant and always will.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

FIGURE Page

CHAPTER I

INTRODUCTION

Unfortunately, when digital integrated circuits are manufactured, some (hopefully small) fraction is always defective. To avoid selling these defective circuits to customers, manufacture testing is employed to identify the parts that are defective. Unfortunately, exhaustively testing integrated circuits by applying all possible combinations of logic values to the primary inputs is extremely impractical for modern designs. As a result, testing itself is imperfect, and some defective circuits will almost always escape the testing process. The goal of test pattern generation is to minimize the number of escapes. This number of escapes is quantified by a metric known as defect level, or defective part level. It specifies what fraction of all integrated circuits that pass all applied tests are actually defective. Thus, the primary goal of testing is to obtain as low a defective part level as possible while expending a reasonable amount of testing resources.

Testing has historically been thought of as a very deterministic process because particular defects are modeled in simulation and tests are generated to purposely detect those modeled defects. However, testing is, in reality, very probabilistic. Modeling all defects is impossible, and thus many defects are never modeled at all and must instead be detected *fortuitously.* Effective digital circuit testing will maximize the probability of this fortuitous detection and minimize the defective part level.

Defective part level modeling tries to capture how a particular test pattern set will interact with the circuit design and the defects to predict how many test escapes will occur. When such modeling is based upon properties of the test set and their

---

The journal model is *IEEE Transactions on Automatic Control.*

effect on circuit nodes, it can be used to not only predict the defective part level, but also to guide the creation of better automatic test pattern generation (ATPG) algorithms. Thus, there is an inherent direct relationship in which better ATPG algorithms yield lower defective part levels, and better defective part level models yield better ATPG algorithms.

In this research, we have developed a defective part level model named MPG-D, which has been shown to be able to predict defective part levels for both benchmark and industrial circuits. We will describe the underlying basis for the model and show that its components have physical meaning. Thus, it can be used to guide test pattern generation to increase the fortuitous detection of unmodeled defects. We will start with the testing of circuits for static defects and then show how fortuitous detection and test pattern generation must change as we attempt to detect more complex and difficult dynamic defects.

Site observation is a primary indicator of the effectiveness of test pattern generation for simple static defects. However, we will show that the randomness of defect excitation will become more important as defects become more complex and more constraints must be satisfied for their detection. We will introduce a metric to estimate the quality of this excitation called *excitation balance* and will show how it relates to one of the constants in the MPG-D model.

Chapter II will present background for traditional ATPG and how it has evolved to create better test pattern sets. Chapter III will show how the two requirements of fault excitation and site observation are satisfied by a test pattern generated to detect a particular fault. Chapter IV will introduce defective part level modeling. Chapter V will focus on the probability of exciting an undetected defect given site observation and will show how this probability changes as the number of observations increases. Chapter VI will incorporate this analysis into the MPG-D defective part level model. Successive variations of the model will be explained along with their advantages and disadvantages. Chapters VII and VIII will introduce the changes that must be made to adequately test more complex defects, such as timing defects. Chapter IX will present the concept of excitation balance and will discuss how it can be used to generate better test pattern sets and how it can be incorporated into the MPG-D defective part level model. Finally, Chapter X presents the major conclusions that can be drawn from this research.

CHAPTER II

TRADITIONAL ATPG AND PREVIOUS WORK

Digital integrated circuit testing consists of applying logic values to the circuit inputs and then capturing the values that appear at the circuit outputs. When the outputs match the expected values, the circuit has passed that particular test. If the output values do not match those specified by the circuit specification, then the circuit fails the test, and the part is deemed to be defective. (Note: Although many digital circuits are sequential instead of combinational, scan-based sequential designs can be viewed as combinational for testing purposes because elements of the scan chain essentially become circuit inputs and outputs in test mode. The work described in this dissertation is applicable to both combinational circuits and sequential scan designs.)

One of the most obvious ways to begin testing a combinational integrated circuit would be to apply every possible input combination as a test pattern. Unfortunately, the number of possible input combinations grows exponentially with the number of inputs. For example, a circuit with only 50 inputs has 2^{50} (over 1.1×10^{15}) possible input combinations. Thus, it is impractical to test one of today's integrated circuits by applying all possible input combinations to every chip. Only a very small subset of all possible patterns can be chosen and applied. Thus, it is important to choose a good subset that will detect a significant fraction of all defective parts. Historically, one of the ways of generating such a subset has been to target faults—specifically single stuck-at faults.

A defect is an actual imperfection in a manufactured circuit. In contrast, a fault is a model of a defect that is used to predict how that defect will affect circuit operation. Before testing begins, a set of faults is identified, and tests are generated to deterministically detect them. As the ATPG process progresses, patterns are produced by targeting each undetected fault in the fault list until all of the faults have been detected at least once or until test generation has failed for all those faults that remain. The mostly widely used fault model is the single stuck-at fault model, which was developed by Eldred in 1959 [1].

In the single stuck-at fault model, each site in the circuit (gate input or gate output) is considered to be potentially "stuck-at-zero" or "stuck-at-one" regardless of the value that should appear at that location based upon the rest of the circuit's logic. For example, a circuit site would behave this way if it were shorted to ground or shorted to power. Thus, the size of the fault list targeted during test pattern generation is at most two times the number of sites in the circuit when single stuck-at faults are used as the targets. This means that in the very worst case the maximum number of test patterns required would be equal to the number of stuck-at faults (two times the number of circuit sites). This is generally much smaller than the number of possible input combinations, but could still be very large in one of today's integrated circuits (which often have hundreds of thousands or millions of sites).

Although test pattern generation using faults is often thought to be a very *deterministic* process, fortunately, many faults are *fortuitously* detected by tests generated for other faults. For example, when we generate a test by targeting fault A, we may find that the generated pattern also detects faults B and C, even though we were not specifically trying to detect them with that test pattern. In this example, faults B and C were fortuitously detected because they were detected by a pattern for which another fault (in this case fault A) was the target. Thus, because of fortuitous detection, the number of patterns required to detect all the stuck-at faults is much lower than the number of faults. For example, we worked with a commercial chip at Texas Instruments that contained over 120,000 stuck-at faults. However, all stuck-at faults that were deemed to be detectable were detected with a test set containing only 429 patterns—a reasonable test set size in a commercial environment.

Unfortunately, the single stuck-at fault model does not accurately describe the effects of all possible defects, and as a result, test sets created to detect all stuck-at faults at least once may not detect an adequate number of the untargeted defects [2],[3],[4],[5],[6],[7]. In fact, as fault coverages approach 100%, the tests targeted at the remaining stuck-at faults are biased in favor of detecting those faults at the expense of the remaining defects [8],[9],[10]. For example, unintentional shorts between points in the circuit often can be better modeled as bridging faults. Both AND/OR and net-dominating bridging faults have been used to study these unintentional shorts [2],[11],[12],[13]. Unfortunately, the number of bridging faults grows quadratically, instead of linearly, with circuit size. Furthermore, because ATPG and fault simulation are done using gate-level netlists, stuck-at faults that occur within gates may not be detected by test set that detects all stuck-at faults on gate inputs and gate outputs at least once [14]. These defects may be best detected by two-pattern tests (such as those used to detect timing defects). Thus, even a test set with 100% stuck-at fault coverage may not really detect all possible stuck-at faults. In addition, still other defects do not manifest themselves as either stuck-at faults or bridging faults, but require another model.

As a result of the mismatch between stuck-at faults and many of the defects that actually appear in manufactured integrated circuits, different researchers have proposed various testing methods to increase the number of defective integrated circuits detected. For example, [15] used analysis of circuit layouts to find “likely” bridges in interconnects so that the “more likely” bridges could be explicitly targeted and so that the fault list would be of reasonable size. Others have suggested using multiple fault models and implementing several kinds of tests (including stuck-at patterns implemented with scan techniques, functional tests, and IDDQ tests) so that the different types of tests would catch different types of defects [3]. Furthermore, the authors of [7] suggested using extensive testing and diagnosis of test chips to monitor chip fabrication processes, to create better fault models, and thus to generate better tests. However, it is not possible to model all types of defects or to target all possible fault models. Thus, regardless of the testing strategy applied, much of the research in our group is based upon the premise that, just as faults are fortuitously detected by tests targeted for other faults during test pattern generation, defects may be fortuitously detected by tests that target faults for which there is not an exact match. The question that remains is how to maximize that fortuitous detection.

In [16], a research team investigated and reported the effect of choosing a test to detect a given stuck-at fault during test pattern generation so that either controllability or observability of circuit lines would be maximized. Controllability was used to refer to the ability to place a particular value (such as a zero for AND bridges) on a circuit line while observability was used to refer to propagating the value of a given line to an observable primary output. It was found that while the controllability criteria did not result in improved tests, using observability as a guideline did improve the detection of non-targeted defects. Similarly, [17] also discussed the importance of site observation. Although the sites where either faults or defects occur must be observed at a primary output for the fault or defect to be detected (regardless of the type of error), the excitation criteria of a defect varies depending upon the type of defect. In other words, the common requirement for detecting different types of faults and/or defects is site observations. Thus, it was suggested that increasing the observation of the least detected faults and using random excitation could improve the detection of non-targeted defects in a circuit. These ideas were integrated by our research group at Texas A&M into the DO-RE-ME (Deterministic Observation, Random Excitation, MPG defective part level optimization) method of test pattern generation [18]. This method was used to generate test patterns that detected more defects in a commercial chip than those generated by a standard commercial ATPG tool. For this method, multiple observations, especially observations of the least detected sites, are accomplished deterministically. This is done because of the fact that all defects, regardless of type, must be observed at a primary output in order to be detected, and because the number of constraints that must be satisfied for a circuit site to be observed is often large, and thus observation is often difficult to accomplish by chance.

In contrast, excitation requirements vary from defect to defect and can often be satisfied randomly. For example, bridges generally require that both ends of the bridge be set to opposite values for excitation to occur. If circuit sites do not have highly skewed probabilities of being either zero or one, once a site has been observed several times, the odds of undetected defects never having been simultaneously excited and thus remaining at that site becomes relatively low. In fact, in the experiment described in [4], no escapes (undetected defects) occurred at rated speed when a minimum of 15 detections per fault was guaranteed during testing. Thus, the DO-RE-ME method attempts to randomly excite the undetected and unknown defects that may be present in the integrated circuits while deterministically observing the sites/faults where those defects may occur as many times as possible.

However, the number of test patterns required to obtain 15 observations of every site may be larger than testing time and tester memory requirements allow. Thus, some way of generating a test pattern set with an acceptable number of patterns is necessary. One way to do this is through generating N-detect sets where every site is detected at least N times and N is chosen so that the resulting test set length remains reasonable [19]. However, the true purpose of testing is to reduce the defective part level, and it is possible for tradeoffs to exist between, for example, observing one site for the second time and observing 30 sites for the third time. Thus, a defective part level predictor which can allow test sets to be compared based upon defect levels instead of merely the number of detections of the least detected faults can be used to extract optimal (or near optimal) subsets of reasonable length from a superset of test patterns that contains too many patterns to be applied in its entirety.

CHAPTER III

TEST PATTERN GENERATION: EXCITATION, OBSERVATION, AND DEFECT DETECTION

Test patterns are generated by targeting faults. In most cases, at least when static defects are to be detected, stuck-at faults are used as the targets. Thus, test patterns are generated to specifically detect circuit sites that are either "stuck-at one" or "stuck-at zero." However, regardless of the fault or defect being detected, two conditions must be satisfied for that detection to occur: fault/defect excitation and site observation. This chapter will demonstrate in more detail what we mean by excitation and observation and will show how these requirements can be satisfied by a test pattern.

A. Fault Excitation

Consider the circuit in Fig. 1. This figure contains a modified multiplexer circuit with a single fault: point P is stuck-at a logic one. In order to excite this fault, a test pattern must create a difference in the logic value at point P between the fault-free and faulty circuits. Thus, the test pattern must set P to a logic zero in the good circuit by setting both A and B to a logic zero as shown in Fig. 2. The values in

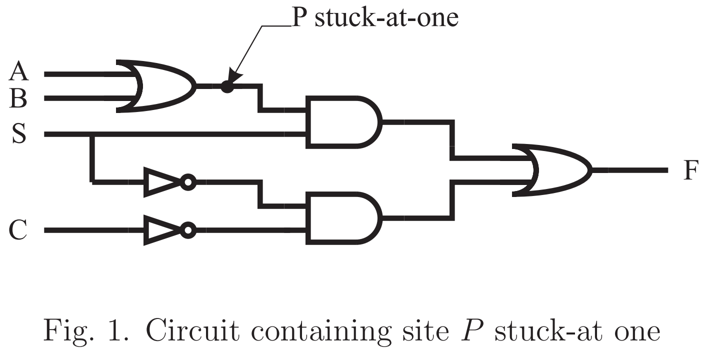


Fig. 1. Circuit containing site P stuck-at one

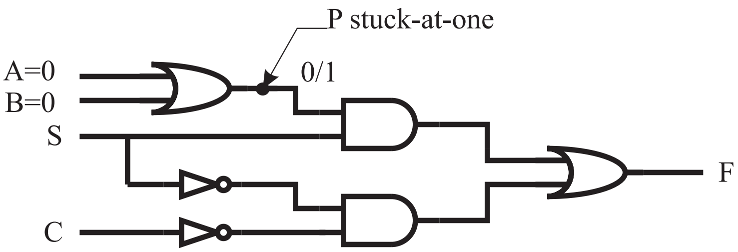


Fig. 2. Excitation for site P stuck-at one

both the good and faulty circuits are shown at point P. The first value is the value in the good circuit while the value after the slash is the value in the faulty circuit. Thus, it is apparent that these two assignments to the primary inputs A and B have successfully excited the fault.

B. Fault Observation

Unfortunately, while fault excitation is necessary for fault detection, it is not sufficient. Because a tester only has access to certain circuit points (primary outputs and elements of the scan chain), merely causing a difference in logic value at the location of the defect does not guarantee detection. We need to propagate the value at that site to a primary output (or scan chain element) so that the value at that site will determine the value at the output. This is called site observation.

To observe site P at primary output F, we need to first propagate the value at point P through the AND gate. This can be accomplished by setting S to a logic one as shown in Fig. 3. Because the S input to the AND gate is set to a non-controlling value, the other input value will determine the output value. Thus the difference in logic value is propagated through the AND gate. Furthermore, setting S equal to a logic one also automatically sets the second input to the final OR gate to a logic zero in both the good and faulty circuits. Thus, the difference in logic value propagates

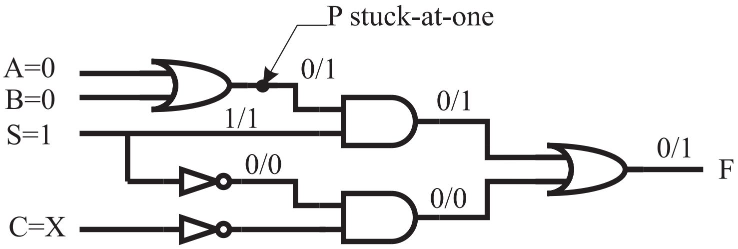


Fig. 3. Excitation and Observation for site P stuck-at one

through this OR gate as well and is visible at primary output F. Therefore, by setting S to a logic one, we were able to fulfill all of the observation requirements for this fault.

At this point, because we have both excited the fault and observed the site where the error occurs at a primary output, we have managed to detect the fault. Notice that input C did not have to be assigned and is actually a “don’t care.” Thus, what we have actually generated is a *partial pattern*. There are actually two patterns that will detect this fault equally well—one in which C is equal to zero and one in which C is equal to one. Historically, C will be given a randomly assigned value, and only one of the two patterns will actually be applied to the circuit under test.

C. Fortuitous Detection of Unmodeled Defects

Unfortunately, there is no guarantee that the defect that actually occurs in this integrated circuit will be a stuck-at fault. For example, the actual defect could very well be a bridge. However, if stuck-at faults were used as the targets during test pattern generation, the only way to detect a bridge is by fortuitously meeting its detection requirements. For example, consider the OR bridge in Fig. 4.

An OR bridge is an unintentional short between two circuit sites. If either of the sites is set equal to a logic one, the other side of the bridge will also be forced to a

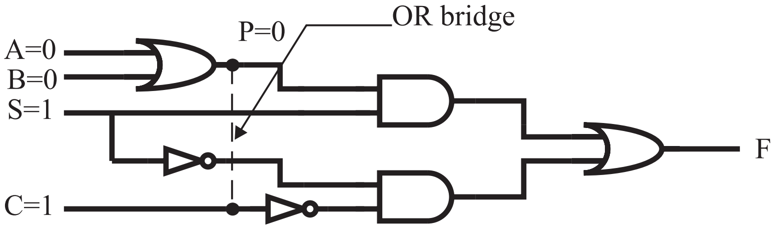


Fig. 4. Detection for OR bridge between P and C

logic one. Thus, a difference in logic value occurs when the two sides of the bridge are set to opposite values in a non-defective circuit. This will satisfy the excitation requirement. Furthermore the side of the bridge set to an incorrect logic value will need to be observed for detection to occur.

Thus, if we are lucky, we *may* detect this fault with the test generated for P stuck-at one. Site P is already being observed by the test pattern. Furthermore, site P is equal to a logic zero in the nondefective circuit. Thus, it has the potential of being forced to a logic one in the defective circuit. Whether or not this actually occurs depends on the value chosen for C. If we were lucky, and C happened to be chosen to be a logic one, then the defect will be excited and detected.

Unfortunately, there is no guarantee that whatever requirements must be satisfied to detect an unknown defect will satisfied by a test pattern targeted toward a different fault. However, the site where the error occurs must be observed for detection to occur regardless of what fault or defect is being detected. The excitation requirements are the constraints that generally differ among the defects or faults that may occur at a circuit site. However, every time a site is observed with a different pattern, the probability of a defect remaining undetected decreases. Thus, the DO-RE-ME method deterministically maximizes the observation of the least observed sites while randomly exciting whatever defects may occur there. In this way, more *unmodeled* defects can be fortuitously detected.

CHAPTER IV

EVALUATING TEST SET EFFECTIVENESS

Establishing figures of merit to quantify the effectiveness of test pattern sets is essential to the testing process. Improvements to ATPG algorithms cannot be made if there is no standard by which to judge a test pattern set. However, even more importantly, such figures of merit are needed to evaluate whether a given test pattern set is adequate and has the capability of detecting a sufficient percentage of all defective integrated circuits. Historically, two important figures of merit for evaluating test set quality have been the single stuck-at fault coverage (evaluated using fault simulation) and the defective part level, or defect level, (the exact value of which is difficult or impossible to determine).

A. Single Stuck-at Fault Coverage

Because test patterns have traditionally been generated by targeting single stuck-at faults, one of the easiest and most widely used figures of merit has historically been the single stuck-at fault coverage. The stuck-at fault coverage is equal to the percentage of the single stuck-at faults that are actually detected by a given test pattern set out of all of those faults that potentially could have been detected. Specifically, the single stuck-at fault coverage is generally calculated according to the following formula:

$$Fault\ coverage = \frac{\#\ of\ detected\ faults}{\#\ of\ detectable\ faults} \tag{4.1}$$

Although there may be a few difficulties in determining the exact value of single stuck-at fault coverage—specifically there is often a question of whether undetected faults are actually redundant (and therefore undetectable) or merely have not been detected by this particular test pattern set—in general this calculation is considered to be relatively straightforward. If an undetected fault is truly redundant, then the fact that it is not detected by a given test pattern set should not lower the fault coverage. Unfortunately, in many cases, proving that a fault is actually undetectable is computationally very expensive, and thus it is unknown whether the nondetection is truly problematic or not. However, once an assumption has been made with respect to whether a given fault is actually detectable, it is easy to compute single stuck-at fault coverage under that assumption given the results of fault simulation.

B. Defective Part Level

Unfortunately, single stuck-at fault coverage only measures the coverage of the stuck-at faults targeted during test pattern generation. A much more important figure of merit is the defect level, or defective part level. (The defect level is given as a fraction while the defective part level is given as the number of defective parts per million. Thus, the defective part level can be obtained from the defect level by multiplying by 10^6.) Unlike fault coverage—which is essentially a measure of how well an ATPG algorithm is able to detect faults—defect level is ideally a measure of how well a test set detects actual manufacturing defects. Specifically, it is a measure of how many parts escape the testing process by passing all of the tests in a test pattern set even though those parts are defective (and could have been shown defective by some test pattern not included in the applied test set.) Thus, ideally, defect level calculations will effectively predict the detection rate of unmodeled defects that are not specifically targeted in the ATPG process.

The boxes in Fig. 5 represent the ensemble of all integrated circuits under consideration. Before a test pattern set is applied, all of the integrated circuits belong to one of two categories: the yield or the undetected defective parts. The yield is

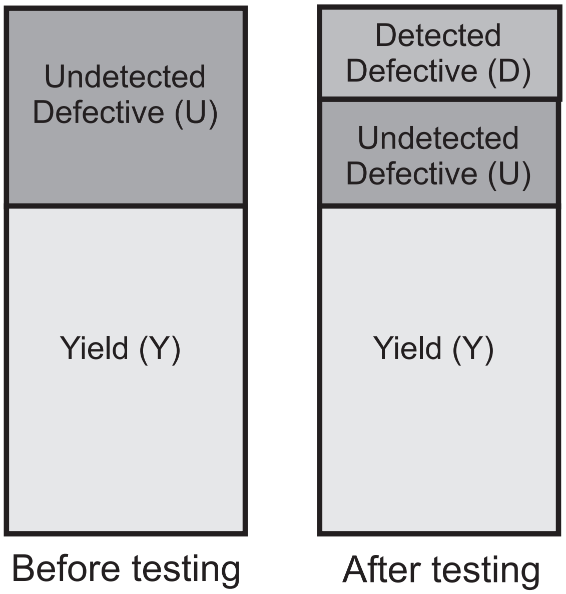


Fig. 5. Defect level changes as testing progresses

the fraction of all parts that are actually non-defective. The rest of the parts are defective. Because no test patterns have been applied yet, all of the defective parts are undetected. This situation is represented by the box on the left. However, once a set of test patterns have been applied, some of the defective parts fail at least one of the tests and show themselves to be defective. There are now three categories of parts: the yield, the undetected defective parts, and the detected defective parts. This situation is represented by the box on the right. If each of these categories is represented as the appropriate fraction of the whole ($U + D + Y = 1.0$), we can calculate the defect level according to Equation 4.2.

$$DL = \frac{U}{Y + U} \tag{4.2}$$

Thus, the defect level is merely equal to the probability that a part that passes all of the tests is still defective.

1. Previous Defect Level Models

Historically, several models have been developed to estimate the defect level. The most famous and widely used of these models is the Williams Brown model [20] shown in equation 4.3.

$$DL = 1 - Y^{1-DC} \tag{4.3}$$

This model estimates defect levels as a function of the yield (Y) and the defect coverage (DC) and assumes statistical independence among the occurrences of different defects. However, the defect coverage is always unknown, and most people have traditionally used the single stuck-at fault coverage as the defect coverage. In that case, when 100% fault coverage is reached, the Williams Brown model predicts a defect level of zero.

The Williams Brown model and several other defect level models were analyzed and compared in [21]. However, most of these models predict defect level as a function of fault coverage. One of these predictors was proposed by Vishwani D. Agrawal, Sharad C. Seth, and Prathima Agrawal. They proposed a model that predicted DL as a function of the yield, fault coverage, and the average number of defects on a chip [22]. The value of the average number of defects must be experimentally determined for this model. A subsequent version [23] by Agrawal and Seth refined this model, but still depended upon fault coverage. Finally, a different model by R. L. Wadsack calculates defect level only as a function of the fault coverage and the yield [24].

Unfortunately, there is a significant disadvantage to using fault coverage in defective part level modeling—especially if fault coverage is assumed to be equal to the defect coverage. This is often a very optimistic assumption because defect coverage and fault coverage tend to become less correlated as fault coverages become very high. This lack of correlation is due to the mismatch between the modeled faults and the actual defects on the chips. In fact, as fault coverages approach 100%, the standard deviation of the defect coverages of the test sets increases—making fault coverage an inaccurate metric for predicting defect coverage and defect level [25]. Furthermore, test pattern sets with identical fault coverages may have very different defect coverages and thus produce very different defective part levels [16],[25],[26]. Specifically, using targeted fault coverage as an estimate of the defect coverage does not take into account the fact that while the fortuitous detection of unmodeled defects is possible with tests generated for specific faults, such fortuitous detection is not guaranteed. Furthermore, even if a defect level model that uses fault coverage doesn't go to zero at 100% fault coverage, there is generally no way to analyze test sets that surpass 100% fault coverage—even though these longer test sets may detect additional defects. Thus, what is really necessary is a defect level model which does not depend exclusively upon fault coverages.

Another model was described by [21] and published in [27]. This model predicts defect level in terms of the number of patterns applied, the total number of vectors, and the number of chips failing at each vector. While this model does not depend upon fault coverage, it requires fallout data to determine the outcome of the calculations. However, most importantly, it gives no guidance with respect to how test pattern sets can be improved or optimized to detect more defects. A more complicated model that does not depend on fault coverage was published in [28]. It also predicts defect level as a function of the number of chips tested, the number that fail at each vector, the total number of vectors, the yield, and the estimated yield after n vectors have been applied. Also, like the other model, it relies on fallout data to produce its calculations, and it does not give any indication of how automatic test pattern generation techniques could be improved.

Another relatively recent model was presented by Jose T. Sousa and Vishwani D.

Agrawal in [29]. This model attempts to incorporate defect clustering into its analysis. However, one of the stated assumptions of the work is that the defect coverage equals 100% when the fault coverage equals 100%. Thus, the authors suggest that if this assumption proves inadequate, then better fault models should be used. A different recent defect level model was proposed by Li-C. Wang and his co-authors in [30].

Another recent metric was presented in [31]. In this case, test sets were evaluated based upon a predicted "Bridging Coverage Estimator." Here, defects are assumed to behave as bridges and the values at circuit sites are assumed to be uncorrelated with a 50% chance of being a one or a zero. Changes in BCE are used to estimate changes in defect level.

Ultimately, an ideal defect level model will not be based on simple fault coverage. An alternative to simple fault coverage is necessary to guide predictions of the amount of fortuitous detections of non-targeted defects and therefore the reduction in defect level, especially for test sets with very high fault coverages. Furthermore, an ideal model would be able to guide the ATPG process to generate better test pattern sets. For example, we would like to be able to choose good subsets of test patterns from supersets. Even more importantly, we would like to generate even better supersets by creating test patterns that maximize the fortuitous detection of unmodeled defects.

2. The MPG Defective Part Level Model

Because fault coverage cannot accurately predict defect coverage when fault coverages become very high, an alternative indicator for analyzing the effectiveness of a test pattern set and estimating the resulting defect level is necessary. Ideally, the components of such a model will represent some physical reality so that it can also provide direction for improving test pattern generation strategies. For this we will need an analysis of the detection requirements for not only stuck-at faults, but for the detection of the actual (often unmodeled) defects that are actually present in a given set of integrated circuits. Recall that two conditions must be satisfied for the detection of any fault or defect, regardless of the type of fault or defect that is actually occurring. First, a difference in logic value between the defective and non-defective circuits must be present at the defect location in the circuit. This is called fault/defect *excitation*. However, because this difference between the defective and non-defective circuits at an interior site cannot be detected by a tester, the difference in logic value at that site must be propagated to an observation point—either a primary output of the circuit or a flip-flop in the scan chain. This is called site *observation*. Thus, when the fault or defect is excited at the same time that the site where the difference in logic value occurs is observed, the fault or defect is detected by the test pattern. Because these two requirements must be met for fault detection, both should be included in any defect level model whose components have physical meaning in the process of defect detection. Our first model which accomplished this was named MPG in honor of its creators: M. Ray Mercer, Jaehong Park, and Michael Grimaila [18].

MPG is a significant improvement over fault coverage based defect level models because it doesn't require a close match between the modeled faults and the actual defects present in an integrated circuit. Instead, it uses the logic underlying the DO-RE-ME test pattern generation method and predicts the defect level based upon the number of times circuit sites have been observed. The equations for this model are shown below:

$$DL = 1 - \prod_{i=1}^{\#\ of\ sites} [1 - (1 - P_{excite})^{\#obs_i} P_{defect}] \tag{4.4}$$

$$P_{defect} = 1 - Yield^{1/\#\ of\ sites} \tag{4.5}$$

Like the Williams Brown model, the MPG model assumes statistical independence for the occurrence of defects at different sites. Furthermore, it assumes that every site has an equal chance of containing a defect, and thus the probability of a defect occurring is uniform across all sites. Specifically, P_{defect} is the probability of the occurrence of a defect at a given site and is calculated from the yield according to Equation 4.5. P_{excite} is the probability of exciting a defect given that the site is observed, and $\#obs_i$ is the number of times site i has been observed. Thus, as a site is observed more times, the probability that an undetected defect remains at that site ($(1 - P_{excite})^{\#obs_i} P_{defect}$) becomes dramatically smaller. This matches the reasoning underlying DO-RE-ME in which maximizing the number of observations of all sites—particularly those which would otherwise be observed the least—minimizes the number of defects that are missed during testing. However, most importantly, unlike the Williams Brown model, the defect level predicted using the MPG equation is not a direct function of fault coverage and does not go to zero when the fault coverage reaches 100%.

The MPG defect level estimator is an improvement over models based upon simple stuck-at fault coverage because it is based upon the requirements for the detection of all defects—both targeted and untargeted—and thus is capable of estimating non-target defect detection. Furthermore, each of its components ($Yield$, P_{defect}, P_{excite}, $\#obs_i$, etc.) has physical meaning, and thus this model provides an intuitive understanding how different characteristics of the circuit, the defects, and the test pattern set interact to produce the final defective part level.

3. MPG-1

One assumption made when calculating defect levels with the MPG model is that the probability of defect excitation given site observation is constant and uniform across circuit sites. However, this probability may vary dramatically from one defect to another. It is a function of the number of constraints that must be satisfied to excite the defect and the difficulty in satisfying those constraints given site observation. For example, when the probability of exciting a bridging defect is compared with the probability of exciting a corresponding stuck-at fault, the probability of exciting the bridging defect is lower. While one constraint must be satisfied to excite a stuck-at fault (the site where the stuck at fault occurs must be set to a logic one in the case of a stuck-at-zero fault or a logic zero in the case of a stuck-at-one fault) an additional constraint must be satisfied for the excitation of a bridging fault. In that case, the additional constraint involves setting the other end of the bridge to a specific logic value. (What logic value is required depends upon the type of bridge and the logic value already present at the observed end of the bridge.) Depending upon which site is involved and its relative likelihood of being set to a logic one or a logic zero, this additional constraint could be very difficult or very simple to satisfy. Thus, there is a significant range of excitation probabilities for defects.

Furthermore, there is no guarantee that the defect present at a circuit site in one chip will be identical to the defects present at that site for all chips containing a defect there. In other words, a circuit site that is defective in a particular integrated circuit could be defective due to any of a large number of potential defects—each of which have their own excitation and detection probabilities. Thus, it is important to recognize the fact that the defective part level could decrease due to the detection of a defect at a given site in a single integrated circuit, but another (possibly more difficult-to-detect defect) could still be present at that site in a different integrated circuit. Taking this type of information into account led to the development of MPG-1 [32].

MPG-1 is very similar to the MPG model, but the calculation becomes iterative and the parameters change as each test pattern is applied. Specifically, after each test pattern is applied, the probability of an *undetected* defect still being present is now lower, and the probability of exciting an *undetected* defect at a site given that the site is observed is modeled as a decaying exponential function of the number of observations:

$$P_{excite} = \alpha e^{-\frac{\#obs_i}{\tau}} \tag{4.6}$$

In this case, the constant α was chosen to be 1 for simplicity, and the constant τ was chosen to best fit the empirical data. When the right value is chosen for τ, this model provides a good match for the industrial and benchmark circuit defect level data.

CHAPTER V

THE PROBABILITY OF EXCITATION GIVEN SITE OBSERVATION

One of the characteristics of the MPG-1 model that distinguishes it from the original MPG model is its use of a varying probability of excitation as the number of site observations increases. Specifically, the MPG-1 model is concerned with the probability of exciting at least one *undetected* defect given a site observation.

A. Why Does This Probability Decrease?

Intuitively, the probability of exciting an undetected defect at a site (given that the site is observed) should decrease as the site is observed more times. Consider Fig. 6. This figure shows the change in the test spaces for defect detection at a site as more test patterns that observe that site are applied. Let each of the boxes contain the set of all test patterns that will observe the site in question. Furthermore, within each of these boxes is a set of ovals corresponding to the set of all possible tests for defects at this site. Let each of the ovals contain those test patterns which will excite the corresponding defect given that the site is observed. Thus, if we happen to choose a test pattern that falls within one of the ovals, we will excite the defect and simultaneously observe the site where it occurs—detecting that defect. Large ovals correspond to defects that are easy to excite and detect while small ovals correspond to defects that are difficult to excite.

Before any test patterns are applied, there are many undetected defects, and a majority of the test patterns that can observe this site will successfully excite and detect at least one undetected defect. This is illustrated in the first box. There is also a significant amount of overlap among the test spaces that will excite different defects—especially among those defects that are easy to excite. Thus, with the ap-

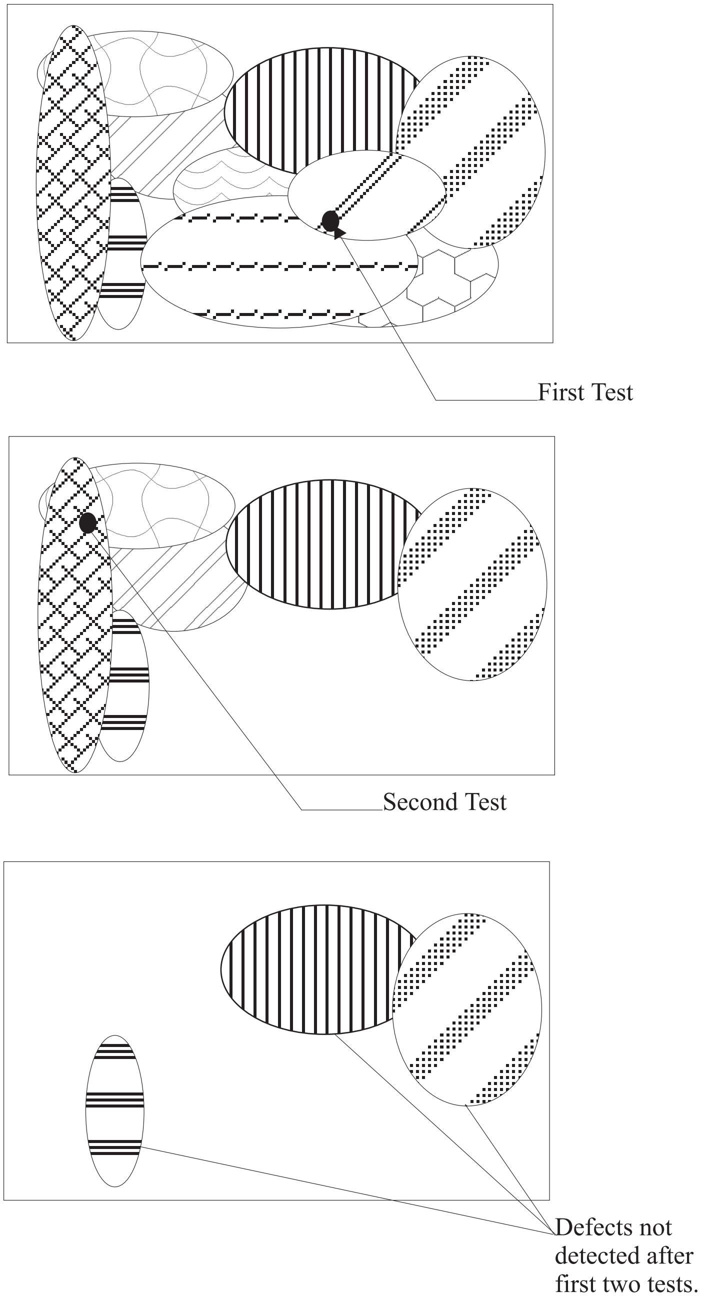


Fig. 6. The probability of excitation decreases as testing progresses

plication of the first test pattern that observes this site, we are likely to fortuitously detect many previously undetected defects.

Once a test pattern is applied, many of the easy defects will likely be fortuitously detected. The detected defects disappear from the Venn diagram in the second box because detecting those defects again will not further reduce the defective part level. There are now many more test patterns that will observe this site but will not excite any new defects. Thus, the second time the site is observed, the probability of exciting an undetected defect with that observation is lower.

When the second test pattern that observes this site is applied, any additional defects fortuitously detected by that pattern will also disappear from the Venn diagram, and the probability of exciting at least one remaining defect becomes even lower. Thus, the probability of exciting an undetected defect at a site given that the site is observed must be a monotonically decreasing function. It can never increase for the manufacturing defects present at the end of production, and as additional test patterns that observe a site are applied, this probability tends to decrease until all but the most difficult to detect defects remain undetected.

In [32] we assumed that the probability of excitation given site observation would decrease as a decaying exponential with a time constant τ. This monotonically decreasing function was chosen based upon the fact that, intuitively, it seems that the initial observations of a site should fortuitously detect many "easy" defects—causing significant reductions in this excitation probability—while subsequent observations should have a less dramatic impact because the probability of excitation would already be quite low. Using this decaying exponential equation for the probability of excitation gave good results in the MPG-1 model, but it was only later that we completed additional experiments to find the exact nature of this probability [33], [34].

B. Surrogate Simulation

To investigate the fortuitous detection of unknown defects using gate-level circuit simulation, it is important to find a way of representing the untargeted and unmodeled defects. This is done through the simulation of *surrogates*—defect models that are not targeted during test pattern generation. For example, during ATPG, test patterns are usually generated by targeting single stuck-at faults. Thus, the detection of those stuck-at faults that were targeted and simulated during the ATPG process is guaranteed and cannot be used to give us adequate estimates of the fortuitous detection of unmodeled defects. On the other hand, the detection of other fault models (such as bridges) is not guaranteed by the ATPG process because the detection requirements of bridges and stuck-at faults differ. Thus, we can use the fortuitous detection of *bridging surrogates* as an indicator of the fortuitous detection of arbitrary unmodeled defects. Therefore, the primary requirement for a choice of surrogate is that its detection requirements differ from the requirements for detecting the targeted faults so that fortuity can be adequately represented. For the following experiments, AND and OR bridges were used as surrogates.

C. Experimental Setup

Data were collected for several benchmark circuits: c432, c499, c880, c1355, c1908, and c5315 [35]. (In each case the number in the circuit name refers to the number of sites where a stuck-at fault could potentially occur. Thus, c432 contains 432 such sites and 864 possible stuck-at faults.) These benchmark circuits were analyzed using computer simulation. (None of the circuits were physically manufactured for this research.) For each circuit, the circuit schematic was read by an Automatic Test Pattern Generation (ATPG) computer program that both generates test patterns for a target fault set and contains a fault simulator. The program we used is a version of Atalanta [36]. It was modified by Sooryong Lee for this project and other related research. The program creates a fault list containing all single stuck-at faults in the circuit and generates a test set to detect those faults. (The maximum number of test patterns generated for any circuit during these experiments was 300.) The program also allows information to be obtained about the detection of the surrogates (in this case bridges) which we specify and which are used to estimate the rate of non-target defect detection when the generated test patterns are applied.

To find the probability of excitation given that a site is observed, we applied test patterns targeted for stuck-at faults to each of the seven benchmark circuits and performed surrogate simulation to obtain data on non-target defect detection for each site as it was observed. Data were collected so that a count was kept of the total number of sites that were observed once, twice, three times, etc. and how many of those sites detected a surrogate when they were observed the first time, second time, third time, fourth time, etc. Then, the probability of surrogate excitation could be calculated by dividing these two numbers. For example, if 100 sites were observed at least once and 99 of those sites had at least one surrogate detected the first time they were observed, then the probability of exciting at least one undetected defect given that the site was observed once is 0.99. These data were then graphed to show how the probability of excitation given that a site is observed changes as the number of observations increases.

Although the circuit simulation and probability of excitation analysis were performed for seven different circuits, the data from only two circuits will be shown here because the results are so similar. These circuit simulations were run with two different pattern sets (both targeted for stuck-at faults) so that the effect of different patterns could be observed. The percent difference was calculated for the probability

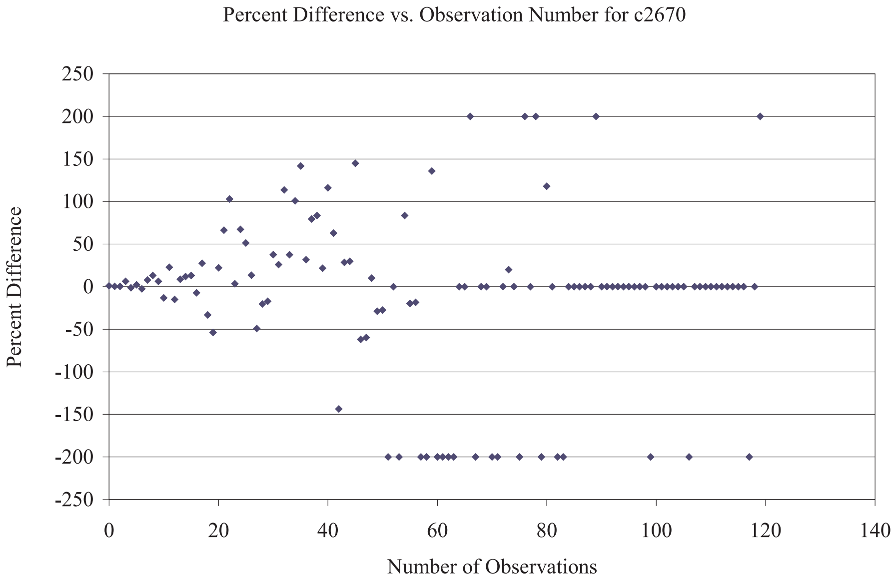


Fig. 7. Variation in probability of excitation between pattern sets for c2670

of excitation results from the two test sets at each observation number. The graphed results are shown in Figs. 7 and 8. For the first few times a point in the circuit is observed, there is considerable agreement between the two pattern sets for the average probability of excitation. However, as the number of observations increases, the behavior of the probability of excitation becomes more chaotic, and there is a large difference between the data for the different pattern sets. This result is probably a quantization effect. This quantization has two probable sources—a decrease in the number of sites included in each sample, and the fragmentation of the actual test space being measured.

As the number of observations increases, the probability of excitation becomes very low and fewer circuit sites have been observed the appropriate number of times to contribute to the sample. It is reasonable to expect that as the sample size decreases, the variance of the sample set will increase. Furthermore, once a circuit site

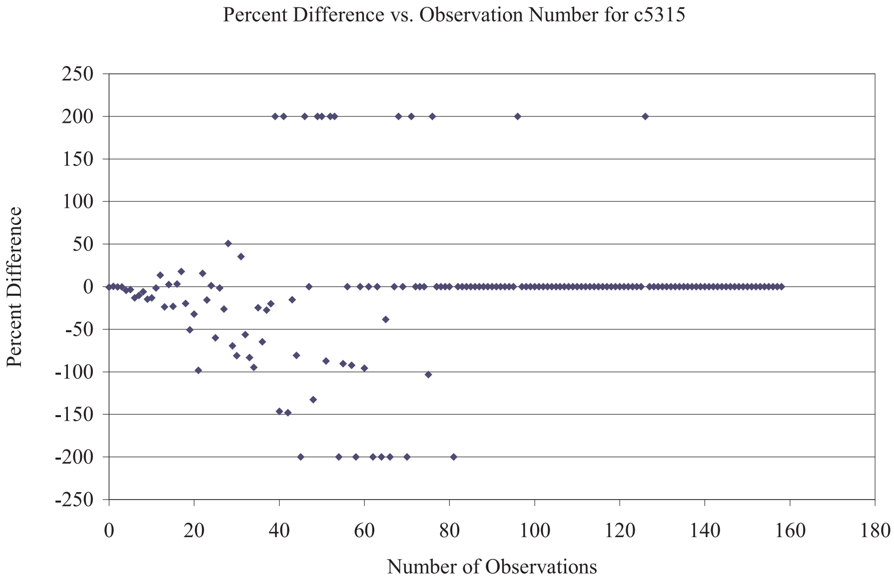


Fig. 8. Variation in probability of excitation between pattern sets for c5315

has been observed multiple times, the number of defects remaining is significantly lower, and thus the space available to excite a defect is significantly reduced and fragmented. Many test pattern sets will excite no defects for a given observation number because they do not fortuitously hit the appropriate sections of the test space for any surrogate. This result actually occurred for both test sets, and the points that show a percent difference of zero after many observations are actually due to the fact that neither test set detected an undetected defect at any site for that number of observations. However, at other times, a test pattern will be lucky and manage to hit one of the few remaining areas that allow a defect to be excited while being observed. This occurrence obviously depends upon which test pattern is being applied, which patterns have already been applied, and what defects remain. Thus, the uncertainty and unpredictability increase drastically as the probability of excitation becomes smaller when the measurement is taken in this fashion.

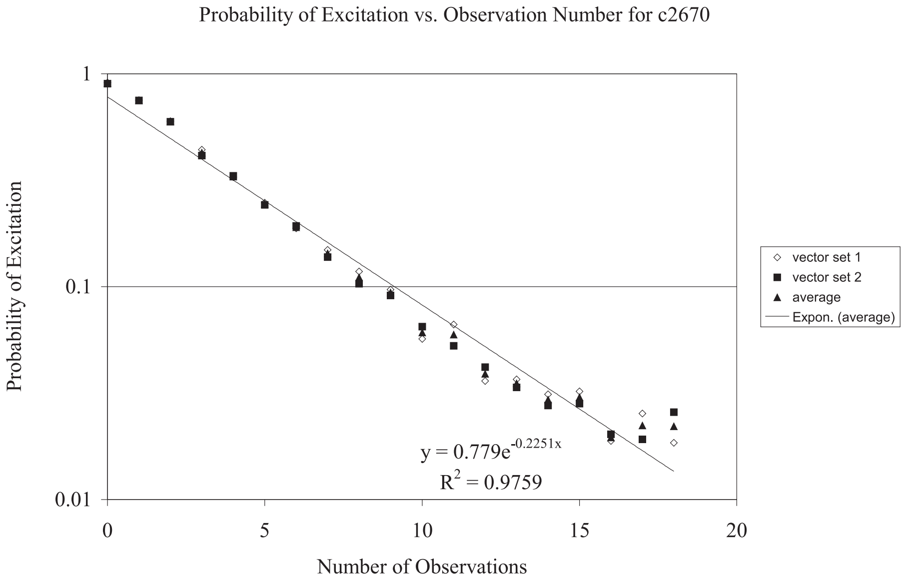


Fig. 9. Probability of excitation curve for c2670

However, it is appropriate to focus on the beginning of the curve because after a site has been observed many times, its contribution to the defective part level becomes miniscule because so few defects are left. The value of the defective part level is dominated by the sites which have been observed less frequently. Thus, when we plotted the probability of excitation curves to find the appropriate values of τ, we focused on the beginning of the curve where the percent difference was reasonable (under 100%) and used the average value of the probability of excitation to predict the best value of τ.

The probability of excitation curves for two of the seven circuits studied can be seen in Figs. 9 and 10. Both of these curves are plotted on a logarithmic scale. Thus, the fact that the data fit a straight line indicates that the shape of the curve is exponential in nature. The R^2 values are both very close to one, indicating a good fit between the data and the best-fit exponential curve. Thus, our hypothesis stated in

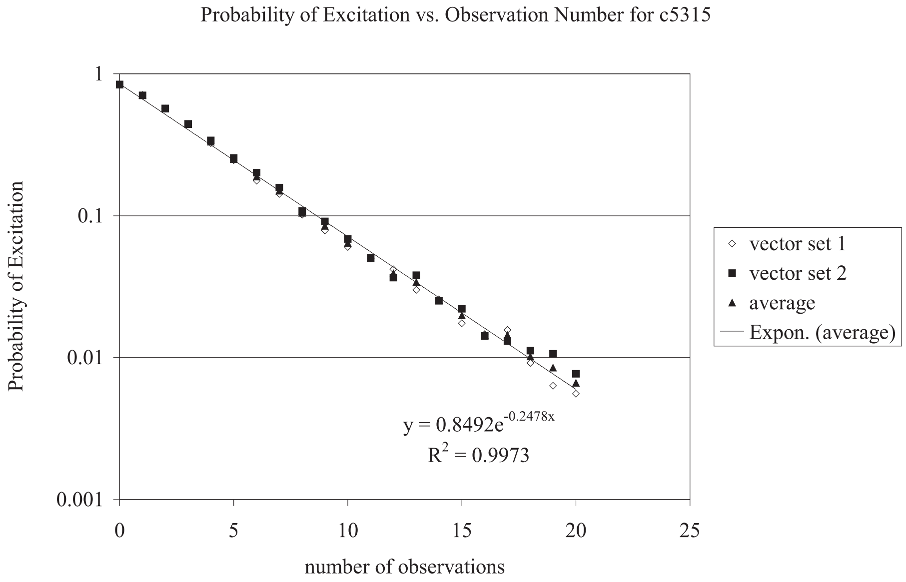


Fig. 10. Probability of excitation curve for c5315

[32] was correct, and the probability of exciting an as yet undetected defect given that the site is observed does indeed decrease exponentially as the number of observations increases. The values of τ for all seven simulated circuits can be found in Table I.

Table I. Probability of Excitation for Benchmark Circuits

circuit	τ
c432	4.275
c499	7.64
c880	3.8212
c1355	3.582
c1908	3.4554
c2670	4.4425
c5315	4.036

CHAPTER VI

THE MPG-D DEFECTIVE PART LEVEL MODEL

Like the Williams Brown model, MPG, and MPG-1 assume statistical independence among the occurrences of defects at different circuit sites. However, we have discovered that in many cases defect level predictions can be made more quickly and accurately if a disjoint probability assumption is made instead. We have used that assumption to create a new defect level model, MPG-D [33], [37].

A. The Disjoint Assumption

A disjoint assumption may be used for defect level calculations because the error introduced by using this assumption instead of an assumption of statistical independence is miniscule under most testing conditions. Ultimately, this is because the probability of a single defect occurring is low and therefore the probability of two or more defects occurring simultaneously is even lower.

Recall that the probability of a defect occurring at a circuit site can be calculated from the yield and the number of sites according to the following equation:

$$P_{defect} = 1 - Yield^{1/\#\ of\ sites} \tag{6.1}$$

For economic reasons, in most cases the yield cannot be very low. (If the yield is low, then a significant fraction of the manufactured circuits are defective and cannot be sold.) Thus, it is reasonable to expect that there is some minimum value for the yield. In this case, let the minimum yield be 0.3. Furthermore, one of today's commercial integrated circuits contains a significant number of circuit sites. In many cases, the number of circuit sites is on the order of 100,000 sites or more. Thus, if we assume that the yield is at least 0.3 and the number of circuit sites is at least 100,000,

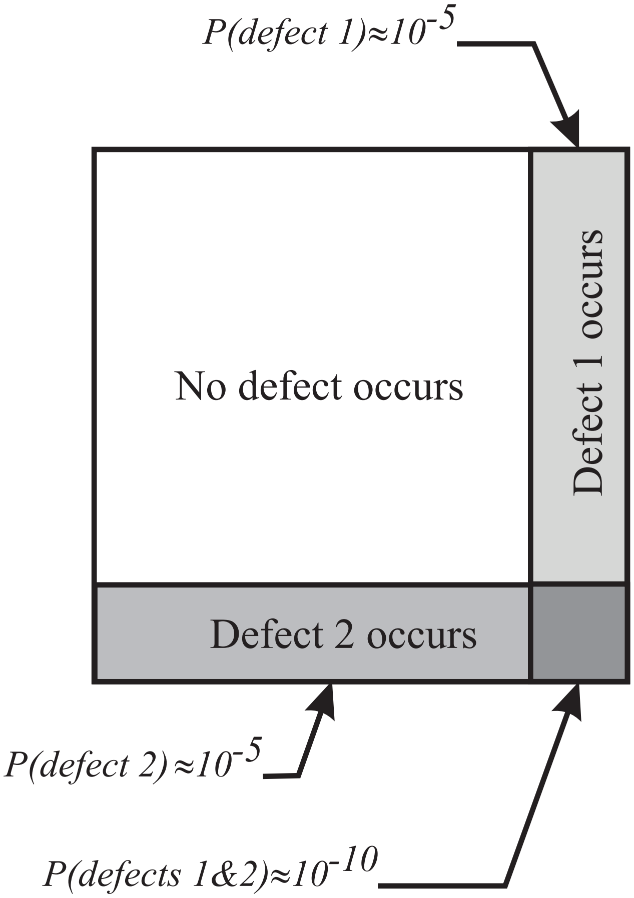


Fig. 11. Statistically independent defect probabilities

then the maximum value for the probability of a defect occurring is approximately $1.21 * 10^{-5}$. Therefore, the probability of even a single defect occurring at a given site is very low.

If the probability of defects occurring is statistically independent, and we are considering the potential occurrence of two defects, then we can represent the four possibilities with a graph like that shown in Fig. 11.

In this figure, the different parts of the rectangles represent different probabilities as a Venn diagram. The vertical rectangle on the right represents the probability that the first defect occurs. Under the assumptions stated earlier, this probability is on the order of 10^{-5}. The area to the left of this rectangle represents the probability that defect one does not occur and is therefore almost equal to one. The horizontal rectangle on the bottom represents the probability that defect two occurs and is also on the order of 10^{-5}. Similarly, the area above this rectangle represents the probability that this defect does not occur. If the occurrence of the defects is statistically independent, then the probability that both defects occur simultaneously is represented by the intersection of their detection rectangles and corresponds to the square in the lower right corner. This square is very small and the corresponding probability is on the order of 10^{-10}. If we assume disjoint probabilities, we neglect to take into account the possibility of this happening; however, because the possibility is so incredibly small, we don't introduce a significant amount of error by neglecting it.

B. Calculating Defect Level with MPG-D

By assuming disjoint probabilities instead of statistical independence, we can replace the product over all circuit sites used to calculate defect level with MPG and MPG-1 by a sum of each site's contribution to the overall defective part level. Thus, initially, before any vectors have been applied, we can distribute the total defective part level equally among all circuit sites. This will give each site i its own defect level contribution according to Equation 6.2.

$$DL\_contr_i(p) = \frac{1 - Yield}{\#\ of\ sites} \tag{6.2}$$

Of course, if layout or historical data indicate that certain sites are more likely to be defective than others, then a weighted distribution of the initial defect level may be used to assign more defect level contribution to sites that are more likely to be defective or contain more defects. However, because we usually do not have such data, we generally assign an equal initial defective part level contribution to all circuit sites.

Recall that the two requirements for defect detection are defect excitation and site observation. Every time that a site is observed, there is some probability of exciting an undetected defect at that site and further reducing the defect level. Thus, just as in MPG, and MPG-1, the final defective part level predicted through MPG-D will be a function of the number of times circuit sites are observed. Specifically, we will reduce the defective part level contribution of every site that is observed by a test pattern p according to equations 6.3, 6.4 and 6.5.

$$P_{excite_i} = e^{-\frac{\#obs_i}{\tau}} \tag{6.3}$$

$$\Delta site_i(p) = \begin{cases} DL_i(p-1) * (A * P_{excite_i}), & \text{if site } i \text{ was observed by pattern } p. \\ \\ 0, & \text{otherwise.} \end{cases} \tag{6.4}$$

$$DL\_contr_i(p) = DL_i(p-1) - \Delta site_i(p) \tag{6.5}$$

Here equation 6.3 specifies the probability of exciting an undetected defect at $site_i$ given that $site_i$ is observed and has been observed $\#obs_i$ times previously. As was shown in the previous chapter, this probability should follow a decaying exponential of the number of observations of the site and some time constant τ. However, while this equation gives us the probability of exciting an undetected defect, it does not tell us how much the defect level contribution of the site should be reduced. For this, we will use the value of $\Delta site$ as calculated in equation 6.4.

Recall that one of the underlying premises of the MPG-1 model was the fact that exciting at least one undetected defect at a site given that it was observed did not guarantee that *all* of the defects that could occur at that site (in different circuits) were also fortuitously excited. Instead, it is possible to fortuitously detect some of the defects at a circuit site while other defects still remain undetected. This idea is captured by the constant A in the MPG-D model. In equation 6.4, A represents the fraction of the remaining defective part level that will be removed given that at least one undetected defect is excited while the site is observed. Like τ, the constant A must be chosen appropriately in order to obtain an accurate defect level prediction.

Thus, if a site is observed on a given pattern, $\Delta site$ represents the average reduction in defective part level contribution that should occur at that site. If a site is not observed, $\Delta site$ is equal to zero because site observation is required for defect detection. After each pattern is applied, we can use the information from fault simulation of that test pattern to determine which sites were observed and to obtain appropriate $\Delta site$ and new defect level contribution values for every site in the circuit.

Then, the overall defect level predicted after the application of pattern p can be calculated as shown in equation 6.6.

$$DL(p) = \sum_{i=1}^{\#\_of\_sites} DL\_contr_i(p) \tag{6.6}$$

C. Simulation Experiments

Initial experiments to evaluate the effectiveness of the MPG-D model were performed on several of benchmark circuits: c432, c499, c880, c1355, c1908, and c5315 [35]. Once again, we needed a way of simulating the fortuitous detection of unmodeled

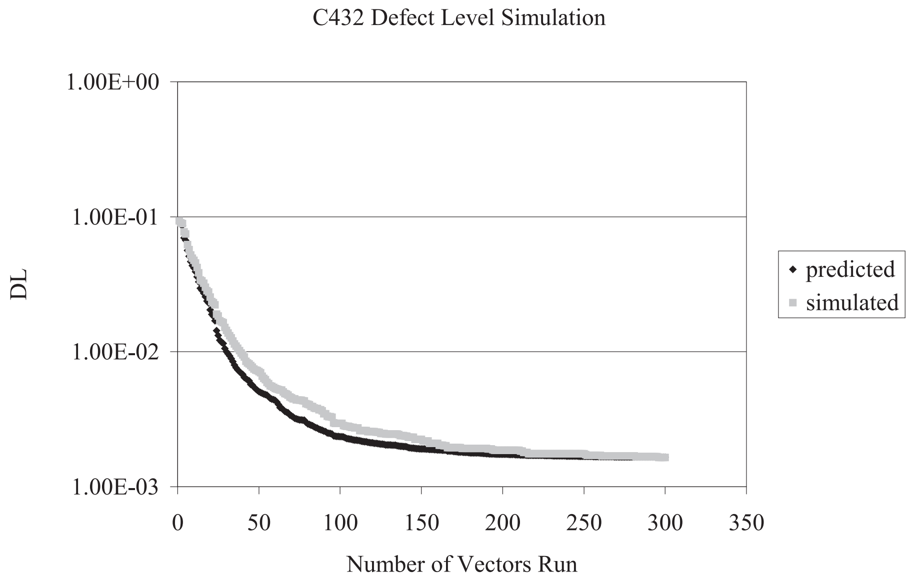


Fig. 12. C432 MPG-D prediction: $\tau = 4.275$ $A = 0.663$

and untargeted defects. Thus, we generated test pattern sets for each circuit by targeting single stuck-at faults and used the fortuitous detection of random bridging surrogates to estimate the fortuitous detection of defects that would occur in an actual manufactured integrated circuit. Specifically, all of our surrogates were either nonfeedback AND bridges or nonfeedback OR bridges. It was not determined for these experiments how many of the bridging surrogates may have been redundant. We arbitrarily assumed a yield of 0.9 and calculated defect levels as the test set was applied based upon the surrogate simulation results and compared those to the values predicted by the MPG-D defective part level model. In each of the MPG-D predictions, the value of τ was taken from Table I, and an appropriate value was found for A to best match the data. Graphs showing the simulation results along with the MPG-D predictions are shown in Figs. 12 through 18.

It can be seen from these figures that the MPG-D model gives fairly good predic-

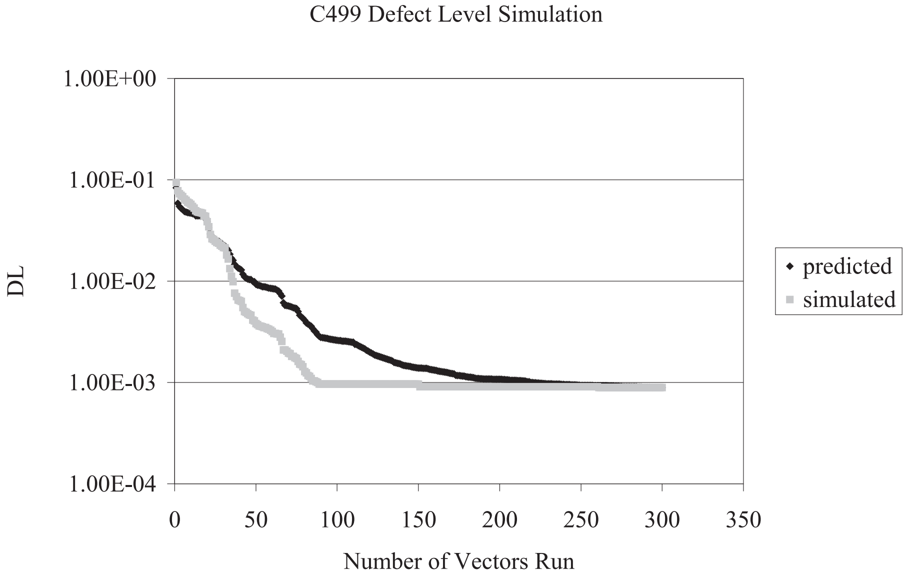


Fig. 13. C499 MPG-D prediction: $\tau = 7.64$ $A = 0.5011$

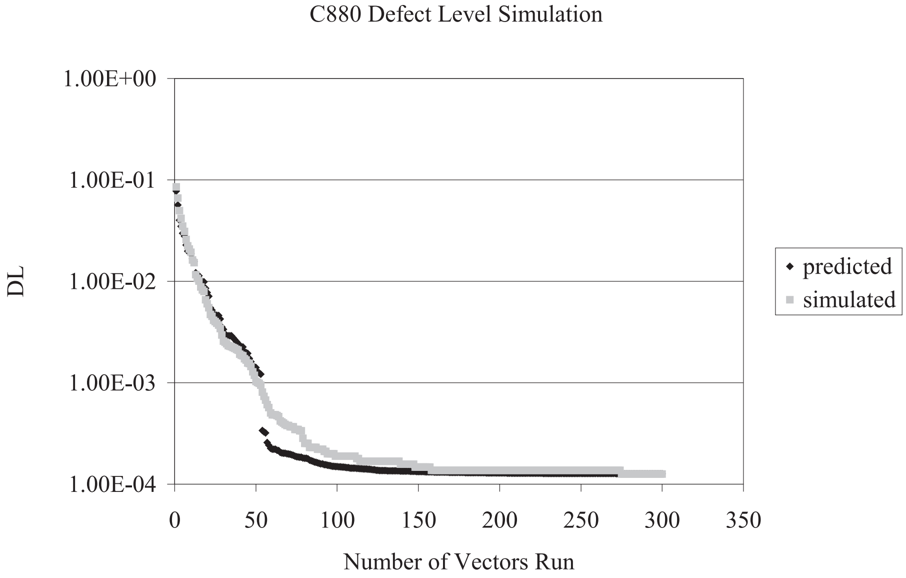


Fig. 14. C880 MPG-D prediction: $\tau = 3.8212$ $A = 0.9215$

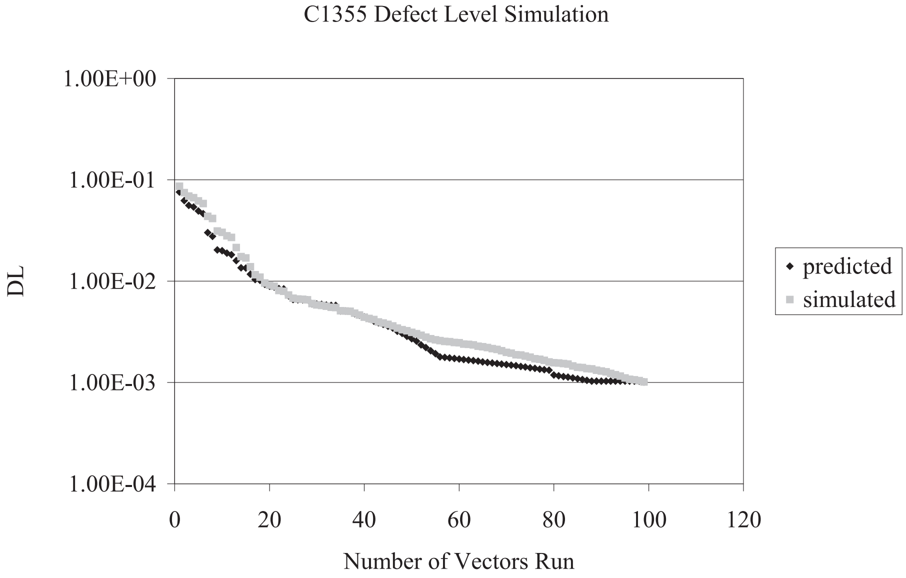


Fig. 15. C1355 MPG-D prediction: $\tau = 3.582$ $A = 0.821$

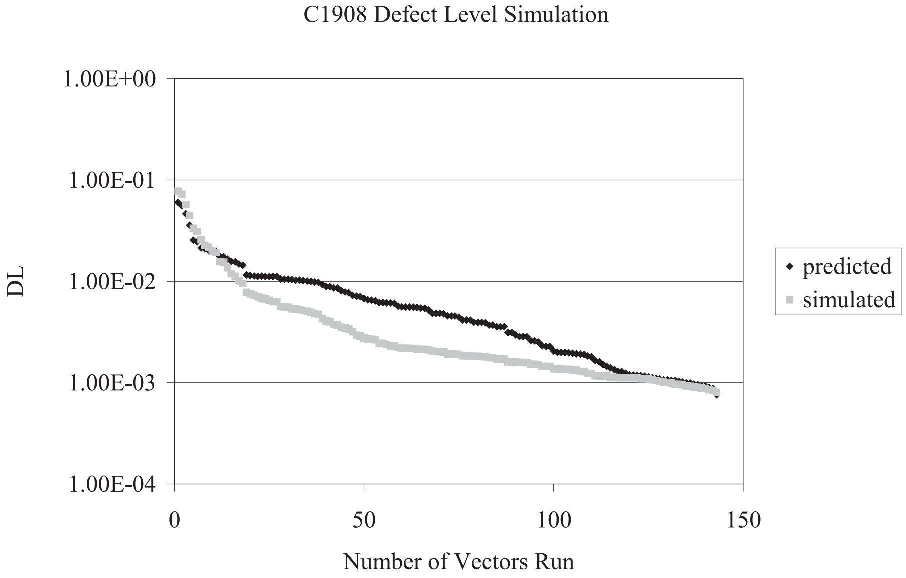


Fig. 16. C1908 MPG-D prediction: $\tau = 3.4554$ $A = 0.870$

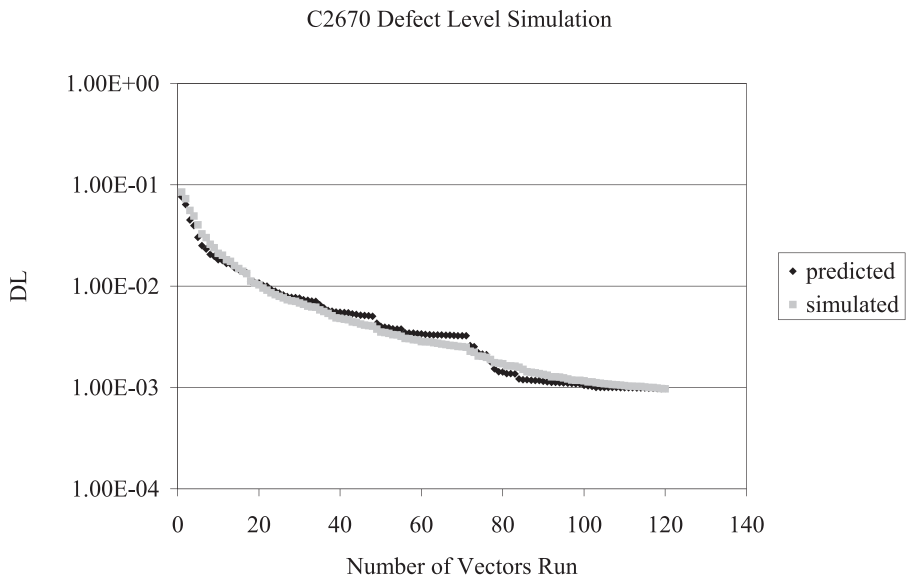


Fig. 17. C2670 MPG-D prediction: $\tau = 4.4425$ $A = 0.790$

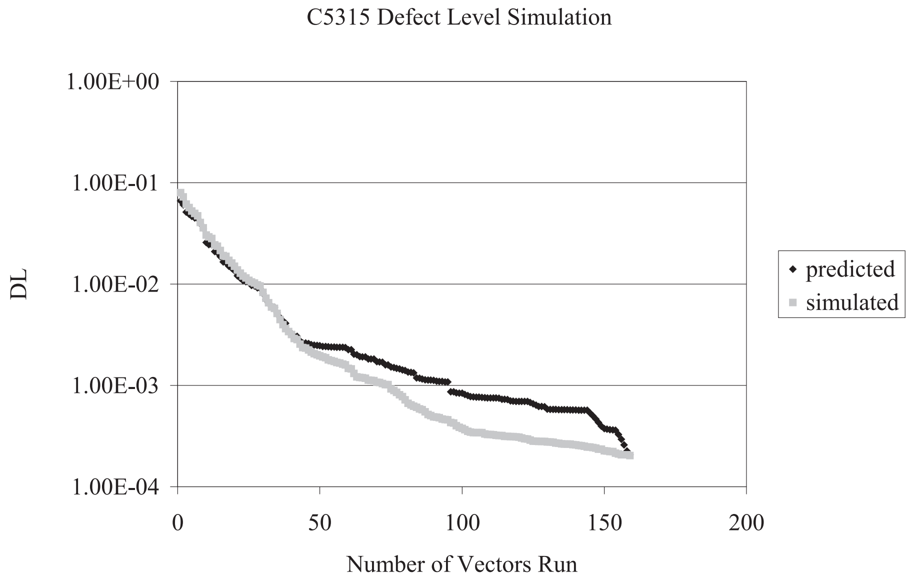


Fig. 18. C5315 MPG-D prediction: $\tau = 4.036$ $A = 0.905$

tions of defect level, especially in the case of c432 and c2670. The worst case occurs in c499. This may be because most of the sites in c499 are highly observable and so a large percentage of the easy defects get detected very quickly. However, even where the MPG-D model predictions differ from the results predicted from the surrogate simulation of the circuit, the MPG-D prediction is pessimistic. Although we would obviously like to be as accurate as possible in our predictions, if an error is going to occur, it is probably better to be somewhat pessimistic and do more testing to catch additional defects than be overly optimistic and unwittingly sell many defective parts to customers.

D. Sharing of Defects among Circuit Sites

The various components of the MPG-D model described above (the constant A, the constant τ, the probability of excitation, etc.) all correspond to the underlying physical components of the amount of fortuitous detection of unmodeled defects that occurs as a series of test patterns is applied. However, at least one aspect is not considered in the model stated above. This is the fact that a single defect may be shared between two sites and detected at either one. An excellent example of such a defect is an AND/OR bridging defect like those we used during surrogate simulation.

A bridge is excited by setting both ends of the bridge to opposite logic values. In the case of an AND bridge, the side of the bridge set to a one in the good circuit is forced to a zero in the defective circuit, and that end of the bridge must be observed for detection to occur. Similarly, in the case of an OR bridge, the side of the bridge set to a zero in the good circuit is forced to a one in the defective circuit, and that end of the bridge must be observed. However, depending upon how the bridge is excited and which side of the bridge is forced to the incorrect value, the circuit site

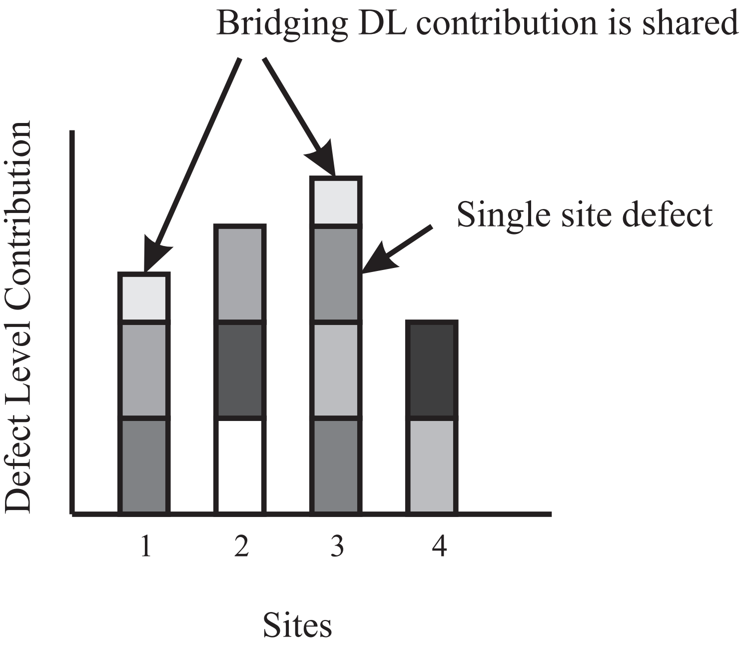


Fig. 19. Two-sided defects split DL contribution between two sites

that must be observed for detection to take place may change from one test pattern to another. Thus, the defect level contribution due to the presence of this bridge does not really belong to a single site but may be equally distributed between both of them. For example consider Fig. 19. Assume that each of the columns represents the defect level contribution remaining at each of four circuit sites after a series of test patterns have been applied. Furthermore, let each of the rectangles within the column represent the defect level contribution that exists due to the presence of a single undetected defect. Each of the tall rectangles represent the defect level contribution of a defect that can only be detected through the observation of that particular site. In contrast, the two short rectangles are half the size of one of the tall rectangles. These two short rectangles correspond to the defect level contribution of a single defect. However, because that defect can be detected at either site one or site three, the defect level contribution is equally distributed between the two sites.

If the defect is detected through an observation of site three, then the short rectangle representing the defect level contribution of this defect at site one should also be removed.

In summary, according to the MPG-D equations shown earlier, the only time a site's defect level contribution could decrease was when it was observed. However, in the case of AND/OR bridges, it is possible for a site's defect level contribution to decrease even when it was *not* observed because the *other end of the bridge* was observed and the defect was excited such that the error occurred at the observed site.

Thus, we refined the MPG-D model to take into account the sharing of defects among circuit sites. In order to do so, we had to determine the answers to two questions:

- How much of the defect level contribution removed from an observed site must also be removed from another site or sites?
- Which sites should have additional reduction in their defect level contributions?

The answer to the first question depends upon the nature of the defects. If all of the defects affect only a single site, then there is no defect sharing among sites and the only way a site's defective part level contribution could possibly decrease is when that particular site is observed. In this case, no additional reduction at another site is needed. In contrast, if all of the defects involve exactly two sites and can be detected at either involved site, then the amount of defect level contribution removed from the observed site containing an error is only *half* of the defect level reduction that was required due to the detection of those defects. Thus, in this case, an equal amount of defect level reduction must be performed at the other involved site.

The number of sites at which the defects present can be detected for the integrated circuits under test will be quantified by a new constant in MPG-D: the constant

C. The constant C will determine how much of the defective part level contribution that will be removed from an observed site should also be removed from other sites. Thus, in the case of only single-site defects, the constant C will be equal to 0. If all defects involve and can be detected at two sites, the constant C will be equal to 1. Furthermore, if the defects are assumed to be a combination of single-site and multiple-site defects, then the constant C can be adjusted according to the presumed fraction of the sites that fall into each category. However, even if we have identified how much defect level contribution must be removed from additional sites, we still need to identify from which sites that additional removal should come.

1. Equal Redistribution

The most obvious way to apportion additional defect level reductions (given that we do not have layout information of likely involved sites) is to remove an equal amount of additional defect level contribution from all other circuit sites whenever defect level contribution is removed from an observed site. Thus, given that site i is observed on pattern p and has a reduction in defect level contribution equal to $\Delta site_i(p)$, we can estimate that the amount of defective part level contribution that should be removed from every other circuit site is equal to:

$$\Delta share_i = C \frac{\Delta site_i(p)}{\#\ of\ sites\ -\ 1} \tag{6.7}$$

However, determining the amount of additional defect level removal needed from every site in this manner is somewhat time consuming because $n - 1$ calculations must be made to find the new defect level contribution of $n - 1$ sites (assuming the circuit contains n sites total) by removing the appropriate value of $\Delta share_i$ from each site's current defect level contribution value. This process will need to be repeated for every site observed by a test pattern and for every test pattern.

However, we can make these calculations somewhat simpler if we are willing to introduce a slight error into our calculations. Specifically, we can divide the additional defect level reduction among all circuit sites—even the site that was observed with the current test pattern. In that case we will apply a little less of the additional defective part level reduction from each of the $n-1$ sites and remove some additional defect level contribution from site i instead. However, because the number of sites in a circuit is generally very large, the error introduced by this change will be very slight. Specifically, $C\frac{\Delta site_i(p)}{\#\ of\ sites\ -\ 1}$ is almost equal to $C\frac{\Delta site_i(p)}{\#\ of\ sites}$ when the number of sites in the circuit is reasonably large. Also note that the total additional reduction in defect level contribution remains the same. The only difference is from which sites it is removed.

Yet, by using this approximation and introducing a small amount of error, we can simplify the calculations. Because we no longer need to worry about which site is the observed site that does not need any additional defect level removal, we can keep a running total of the total additional removal that will need to be done for a given test pattern over all sites by keeping a running total of the values of $\Delta site$ as shown in equation 6.8.

$$Total\Delta site = \sum_{i=1}^{\#\_of\_sites} \Delta site_i \tag{6.8}$$

Then we can calculate the amount that should be removed from every site according to equation 6.11

$$equal\ \Delta share = \frac{C * Total\Delta site}{\#\ of\ sites} \tag{6.9}$$

Thus, each site's new defect level contribution after the application of pattern p is:

$$DL\_contr_i(p) = DL\_contr_i(p-1) - \Delta site_i - equal\ \Delta share \tag{6.10}$$

However, there is still one problem. It is possible for $equal\ \Delta share$ to be larger than $DL\_contr_i(p-1) - \Delta site_i$. Having a negative value for a site's DL contribution in nonsensical. Thus, we need to refine our calculations to take into account the amount of DL contribution that remains at a site after that site's value of $\Delta site$ is removed to avoid having negative DL contributions.

Specifically, we can calculate a $\Delta share$ value for each site i in the following manner:

$$equal\Delta share_i = \begin{cases} \frac{C * Total\Delta site}{\#\ of\ sites}, & \text{if } \frac{C * Total\Delta site}{\#\ of\ sites} \text{ is less than} \\ & \quad DL\_contr_i(p-1) - \Delta site_i \\ DL\_contr_i(p-1) - \Delta site_i, & \text{otherwise.} \end{cases} \tag{6.11}$$

Then, instead of subtracting a common $\Delta share$ term from all sites, each site's $\Delta share$ term will be customized to ensure that no sites obtain a defect level contribution less than zero. Unfortunately, this means there may still be some additional defective part level contribution removal due to defect sharing that was not taken away from any site. In this case, another iteration of the algorithm is necessary to evenly distribute this remaining reduction among all of the sites that still contain a nonzero defect level contribution. Ultimately, several iterations may be needed before the remaining reduction that has not been applied to any site goes to zero or some other very low threshold value. However, ultimately, by using these equations we will be able to take into account some of the effect of the sharing of defects between circuit sites.

In order to investigate the effectiveness of this refined version of MPG-D which takes defect sharing into account through the "equal" distribution of additional defect level removal, we used it to predict the defect levels resulting from surrogate simulation of the test sets and benchmark circuits shown earlier in this chapter. Because

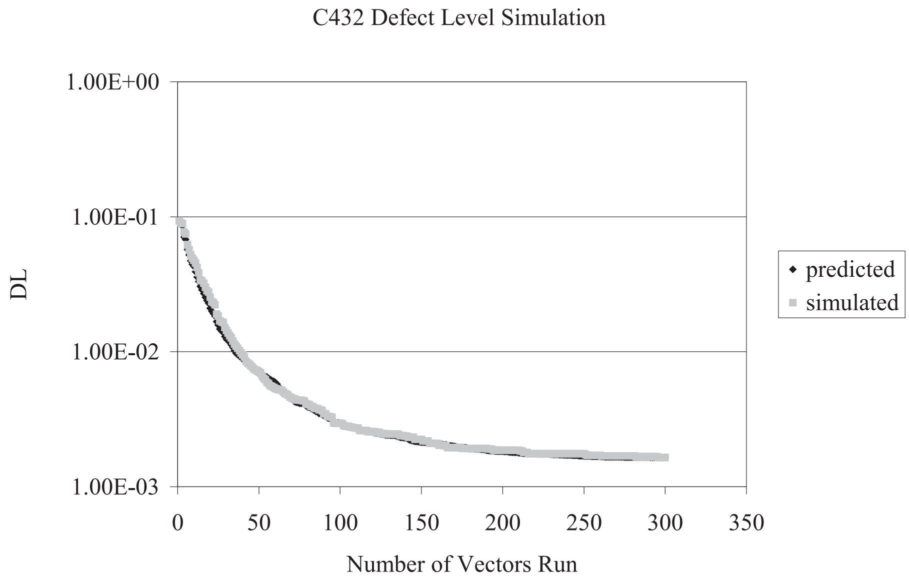


Fig. 20. C432 MPG-D prediction (equal share): $\tau = 4.28$ $A = 0.302$ $C = 1.0$

all of our surrogates are AND/OR bridges, which by definition can be detected at either end of the bridge, we chose C to be equal to one for all of our calculations. The resulting MPG-D predictions along with the results obtained previously from surrogate simulation can be found in Figs. 20 to 26.

It is evident from these figures that taking into account this additional defect level reduction at sites where the defect was not observed can improve the overall defect level prediction. In many cases, the large jumps in the data have disappeared, and even in areas where the prediction is still pessimistic, the degree of pessimism is less. This is because we no longer have the full defect level contributions at a site when it is observed for the first time relatively late in the testing process. Because the bridging surrogates can be detected at either end of the bridge, a least some of that defective part level contribution was reduced due to the detection of some of those defects at the other (more observable) ends. Thus, before defect sharing

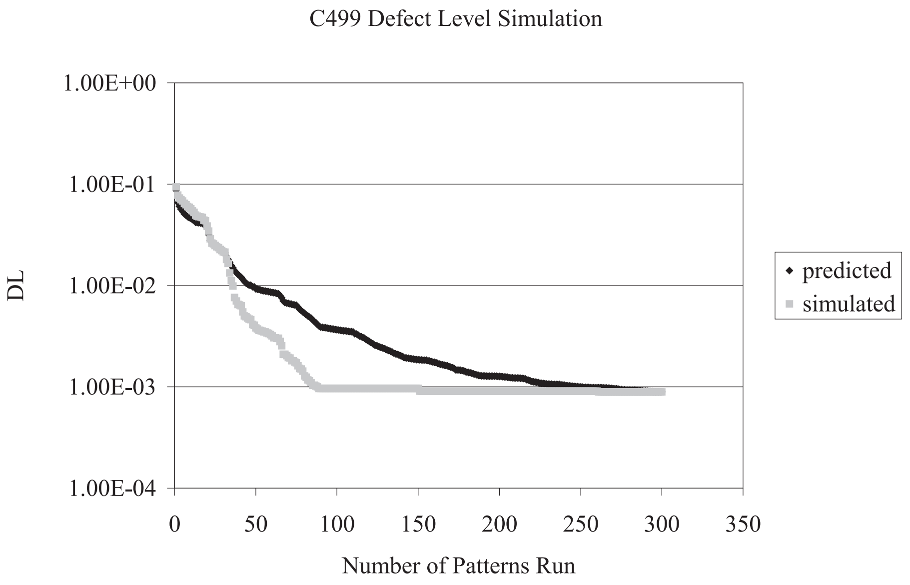


Fig. 21. C499 MPG-D prediction (equal share): $\tau = 7.64$ $A = 0.198$ $C = 1.0$

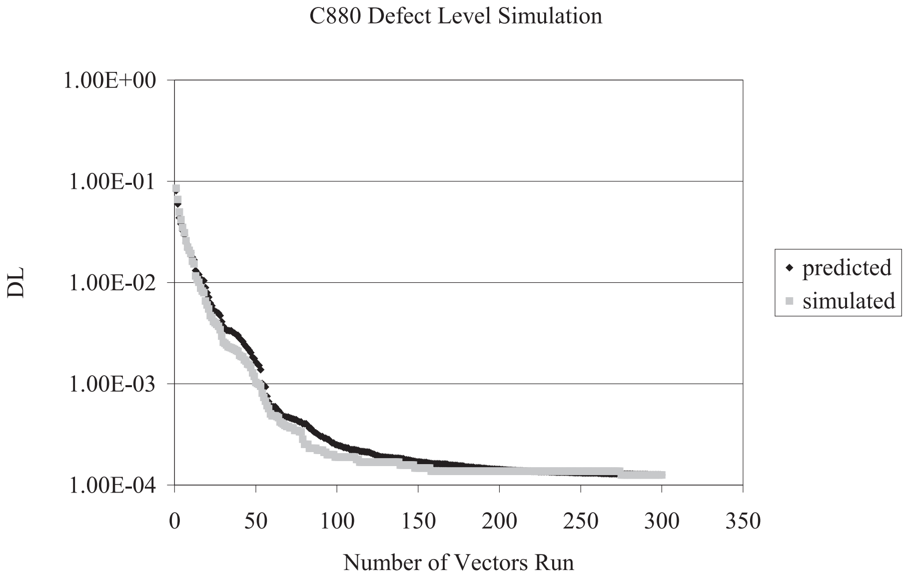


Fig. 22. C880 MPG-D prediction (equal share): $\tau = 3.82$ $A = 0.393$ $C = 1.0$

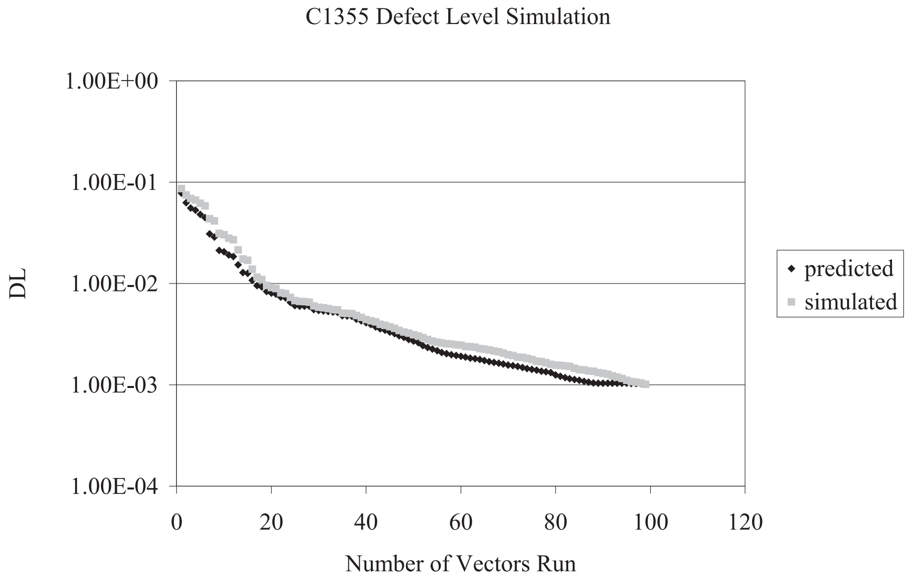


Fig. 23. C1355 MPG-D prediction (equal share): $\tau = 3.58$ $A = 0.373$ $C = 1.0$

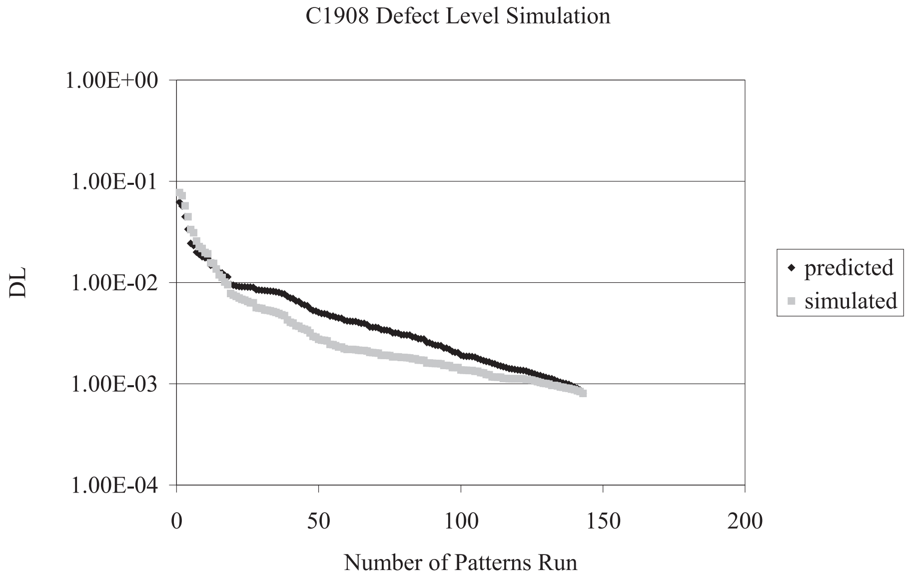


Fig. 24. C1908 MPG-D prediction (equal share): $\tau = 3.46$ $A = 0.408$ $C = 1.0$

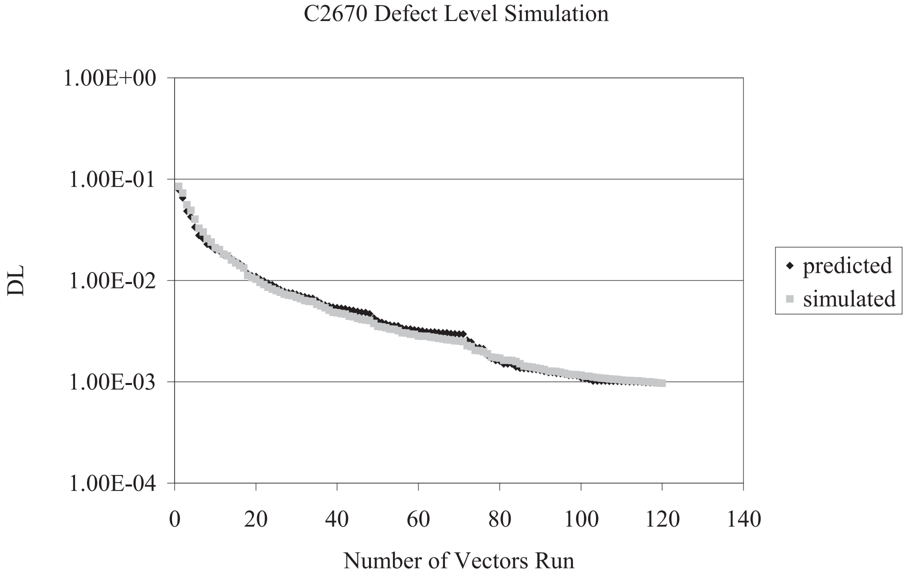


Fig. 25. C2670 MPG-D prediction (equal share): $\tau = 4.44$ $A = 0.343$ $C = 1.0$

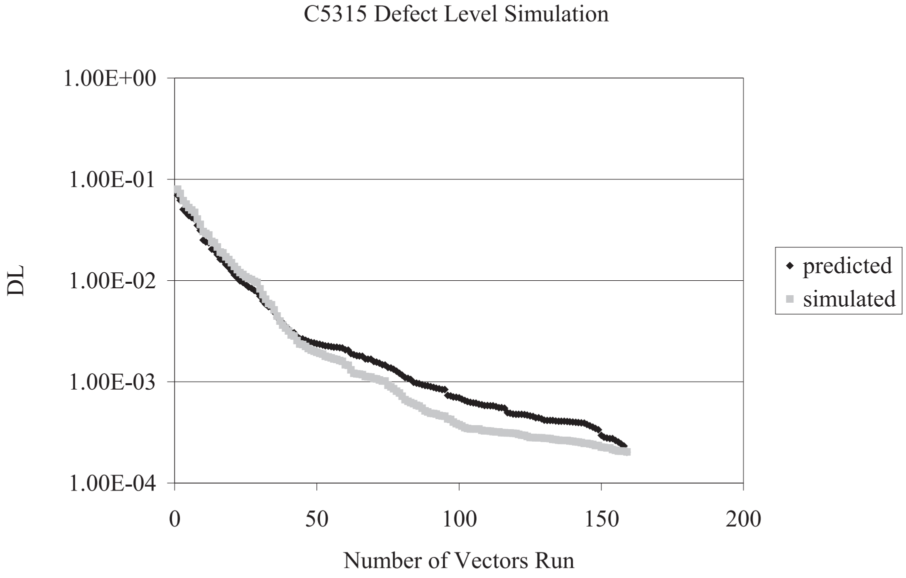


Fig. 26. C5315 MPG-D prediction (equal share): $\tau = 4.04$ $A = 0.411$ $C = 1.0$

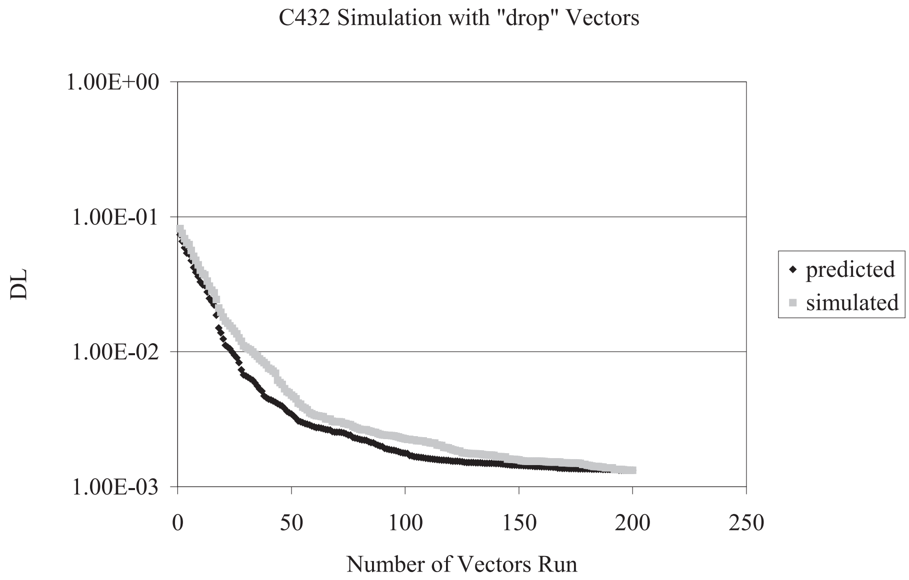


Fig. 27. “drop” MPG-D prediction (no share): $\tau = 4.28$ $A = 0.694$

was taken into account, defect level contributions at these sites remained artifically high. Furthermore, when such a site was finally observed, a significant amount of defect level contribution would suddenly be removed (depending upon the value of A), leading to the large jumps in the MPG-D prediction that were not present in the surrogate simulation data. Thus, taking defect sharing into account (when it is present) allows us to not overemphasize the importance of the least observed sites when some of their defects are detected at other more easily observed sites.

Another significant improvement of the MPG-D model, especially when defect sharing was taking into account, was the ability to use the same constants even when different test pattern generation strategies were employed. For example, consider Figs. 27 to 34.

For these experiments, four different test sets (with 200 test patterns each) were generated by Sooryong Lee for c432. If 100% fault coverage was achieved with fewer

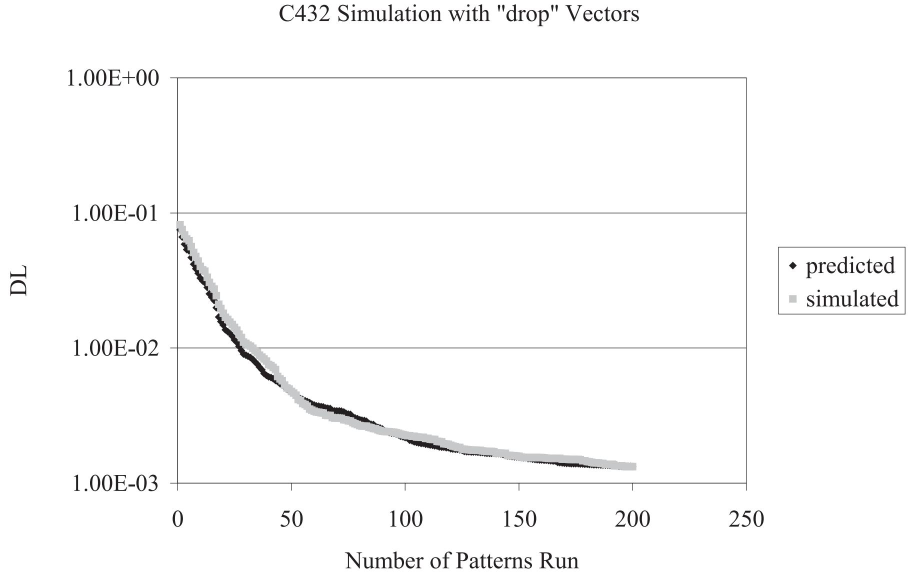


Fig. 28. “drop” MPG-D prediction (equal share): $\tau = 4.28$ $A = 0.331$ $C = 1.0$

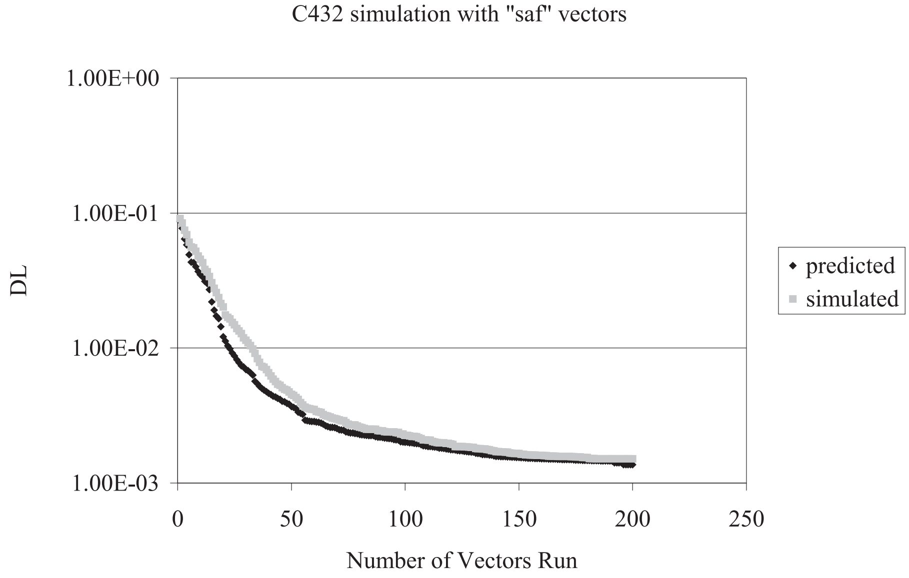


Fig. 29. “saf” MPG-D prediction (no share): $\tau = 4.28$ $A = 0.694$

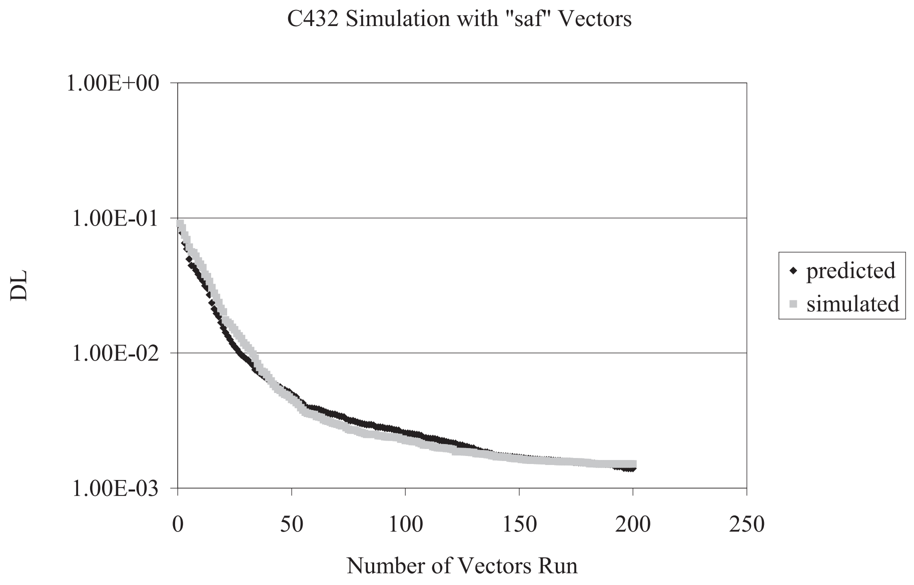


Fig. 30. “saf” MPG-D prediction (equal share): $\tau = 4.28$ $A = 0.331$ $C = 1.0$

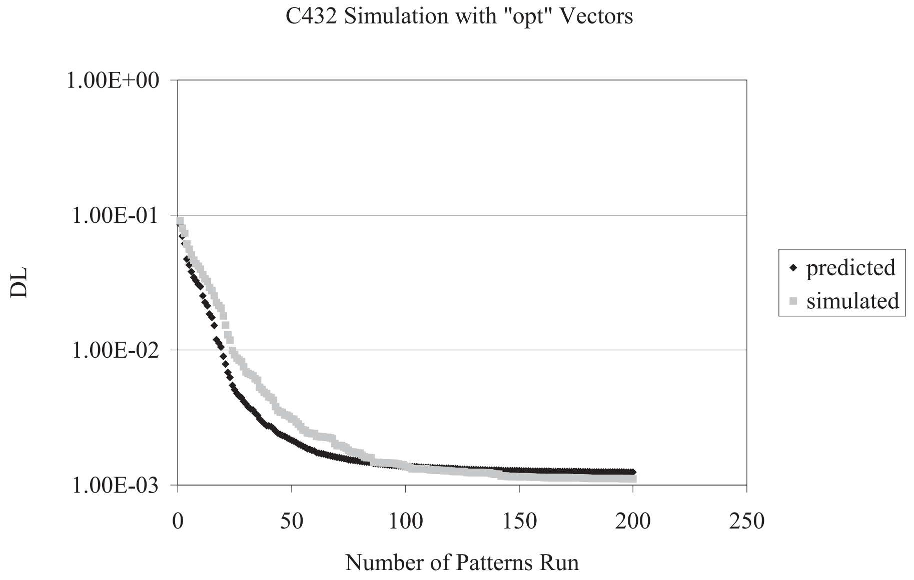


Fig. 31. “opt” MPG-D prediction (no share): $\tau = 4.28$ $A = 0.694$

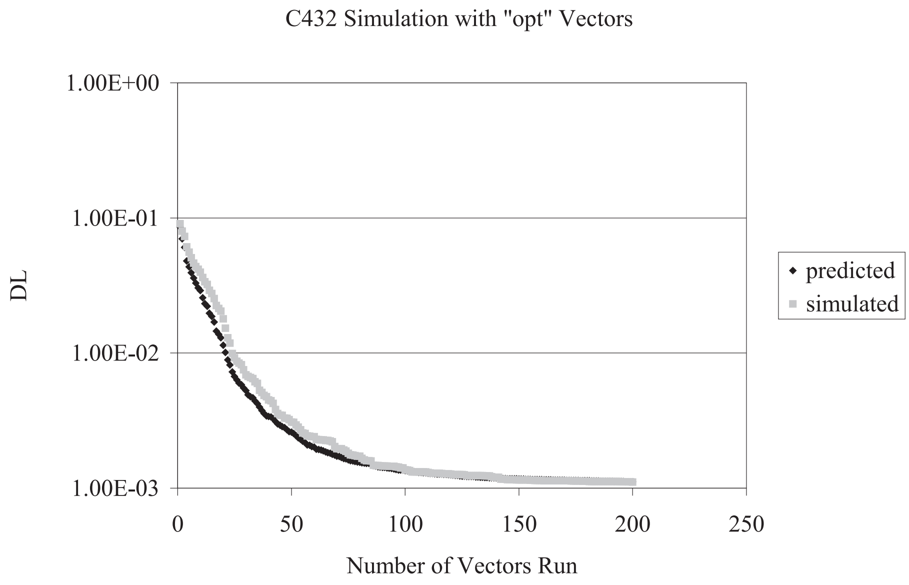


Fig. 32. “opt” MPG-D prediction (equal share): $\tau = 4.28$ $A = 0.331$ $C = 1.0$

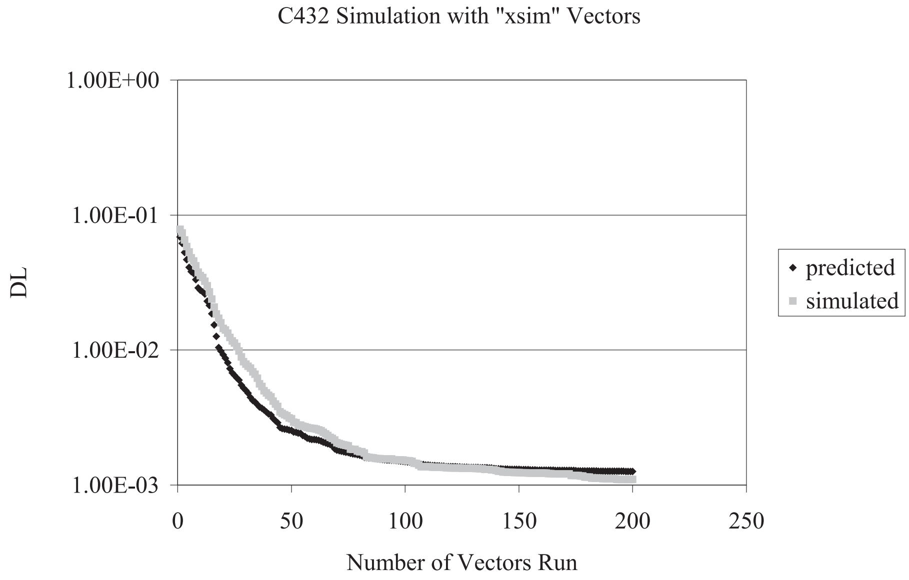


Fig. 33. “xsim” MPG-D prediction (no share): $\tau = 4.28$ $A = 0.694$

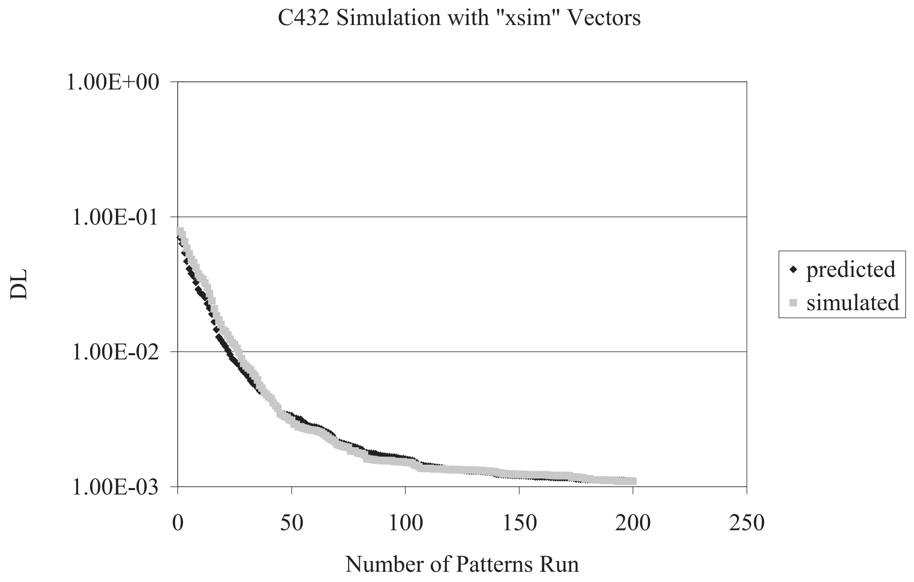


Fig. 34. “xsim” MPG-D prediction (equal share): $\tau = 4.28$ $A = 0.331$ $C = 1.0$

than 200 test patterns, then the ATPG process was repeated until 200 patterns were obtained. Each of the test sets was generated with the same underlying ATPG engine, but with different test pattern generation strategies. Specifically, the “saf” vectors were generated without fault dropping. In contrast the “drop” vectors employed fault dropping. Thus, after a test pattern was generated, all of the faults fortuitously detected by that pattern were noted and so that they would not be explicitly targeted in future patterns while other faults still remained undetected. The “opt” test pattern set was generated with the goal of maximizing site observations. Finally, the “xsim” vectors also tried to maximize site observations, but additional work was done to find good values for unassigned inputs so that even more site observations could be achieved.

It is evident that even when the same constants are used, MPG-D was able to make a good prediction for the different test pattern sets. This is a significant improvement over the MPG-1 model in which very different values of τ had to be chosen as the test set changed in order to achieve a good defective part level prediction. In that case, the appropriate value of τ was an indicator of the effectiveness of a test pattern set. While using τ as such an indicator is helpful if the appropriate value is known, for MPG-1 there was no convenient way of predicting τ accurately for two different test pattern sets a priori so that the defect level model could be used to predict their relative effectiveness. In contrast, in this experiment, MPG-D was able to effectively predict defect levels for the four different test pattern sets without changing the constants. This makes it much easier to try to compare the effectiveness of two different test pattern sets with the MPG-D model.

It should be noted that we still seem to get better predictions by using the appropriate value of C and incorporating defect sharing into our calculations. This is not terribly surprising because, by using C, we are more accurately modeling what is actually occurring in the circuit. However, we did notice that incorporating the correct value of C into our calculations can introduce additional difficulties when the defects change. Specifically, the bridges modeled in Figs. 20 to 26 were different from those modeled in Figs. 27 to 34, and the ideal value of the constant A changed from one defect set to another. In that case, we found that the value of A was less sensitive to changes in the actual defects when the simpler calculation that did not include defect sharing was used. Thus, as we incorporate more parameters into our modeling to accurately represent defect detection, it seems that our model parameters can become more sensitive to changes in the actual defects occurring.

2. Proportional Redistribution

In the previous section defect sharing was modeled by ascertaining how much additional reduction of defective part level was required at other circuit sites followed by an equal (to the largest extent possible) distribution of that reduction across all circuit sites. This was done because no information about what other sites were most likely to share defects with the site in question was available, and thus we did not want to bias our results in favor of or against any specific site unnecessarily. However, we can also consider the fact that those sites which still contain the most defect level contribution at any given time are likely to be those sites with which defects are shared. Thus, we can consider a proportional redistribution of defective part level instead of an equal distribution.

When we use proportional redistribution, we assume that those sites which currently contribute the most to the overall defect level are exactly those sites that are most likely to be the “other end” of a two (or more) sided defect. Given that we lack layout information, this is a reasonable assumption. First, a site’s defect level contribution is an indication of how likely that site is to contain defects. The higher a site’s defect level contribution, the more likely it is to contain defects, and thus the more likely it is to potentially be the other end of a two-sided defect that has just been detected. In addition, once many test patterns have been applied, a site with a high defect level contribution compared to many of the other sites will likely have been observed very few times (if at all) and is probably inherently difficult to observe. Thus, it is probable that any defects that occur at that site and are shared with other sites will ultimately be detected at a site that is easier to observe. Thus, it is likely that a significant fraction of the defective part level reduction of a difficult-to-observe site may occur through observations of other sites. Thus, we propose to remove proportionally *more* defect level contribution from those sites that have been observed the least and are most likely to contain remaining defects as evidenced by their current defect level contribution instead of equally distributing this additional defect level reduction among all circuit sites.

Using a proportional distribution of defect level removal requires that we alter our equations for defect sharing. First, we must find the fraction of the overall defect level that is contributed by each site. This is done for pattern p according to equation 6.12:

$$site\_proportion_i(p) = \frac{DL\_contr_i(p-1) - \Delta site_i(p)}{DL(p-1) - Total\Delta site(p)} \tag{6.12}$$

Then each site is given its own value of $\Delta share$ based upon its relative contribution to the defect level as shown in equation 6.13:

$$proportional\_\Delta share_i(p) = C * Total\Delta site * site\_proportion_i(p) \tag{6.13}$$

Finally, each site's new defect level contribution after the application of pattern p is found by removing defect level contribution due to that site's observation status on pattern p ($\Delta site$) and the proportion defect level reduction due to sharing of defects between sites as shown in equation 6.14.

$$DL\_contr_i(p) = DL\_contr_i(p-1) - \Delta site_i(p) - proportional\_\Delta share_i(p) \tag{6.14}$$

One advantage of using proportional redistribution and calculating it according to these equations is purely computational. Specifically, it is unnecessary to go through multiple iterations of redistribution for a single pattern because we will never try to take away more defect level contribution from a site than is already present. This is because a site's proportional contribution to defect level in equation 6.12 is calculated according to the proportion of the defect level that will remain at that site after all of the defect level reductions due to $\Delta site$ terms are included. Specifically, as long as reasonable values are chosen for C and A, then the total amount of DL reduction due to sharing will be less than (or equal to) the total DL that remains when the $\Delta site$ reductions are included. (Otherwise, we would end up with a negative DL, which is impossible.) Thus,

$$DL(p-1) - Total\Delta site(p) > C * Total\Delta site \tag{6.15}$$

So,

$$\frac{C * Total\Delta site}{DL(p-1) - Total\Delta site(p)} < 1 \tag{6.16}$$

and because $proportional\_\Delta share_i(p)$ is equal to:

$$(DL\_contr_i(p-1) - \Delta site_i(p)) * \frac{C * Total\Delta site}{DL(p-1) - Total\Delta site(p)} \tag{6.17}$$

we can say that

$$proportional\_\Delta share_i(p) < DL\_contr_i(p-1) - \Delta site_i(p) \tag{6.18}$$

Thus,

$$DL\_contr_i(p-1) - \Delta site_i(p) - proportional\_\Delta share_i(p) > 0 \tag{6.19}$$

and no site will have a negative defect level contribution. This is a significant saving of computational time and power over the equal distribution method in which many iterations may have been used before the remaining defect level reduction was sufficiently distributed.

However, even saving a significant amount of time will be useless if the refined analysis cannot model the defect level. To investigate this, we were fortunate to be able to use data collected from two lots of integrated circuits at Texas Instruments. Two test pattern sets of 3000 patterns each were both applied to every circuit. The first test pattern set was named the “Commercial” set because it was generated with a commercial ATPG tool according to traditional industrial practice and had a single stuck-at fault coverage of 97%. The second test pattern set was designated the "Research" test pattern set. In this case, the underlying patterns were generated with the same ATPG engine, but they were optimized according to the DO-RE-ME method by maximizing the number of times the "difficult-to-observe" sites were actually observed. The single stuck-at fault coverage of the "Research" set was 96.7%. These two test sets were generated for an industrial design that contained more than 75,000 two-input NAND equivalent logic gates, and two lots of these circuits were tested—one in October 1998 and one in February 1999. Thus, it was possible to compare the effectiveness of the two test pattern sets and estimate changes in defective part levels based upon the actual failure rate of these integrated circuits during testing.

The first lot of production wafers was tested in October 1998 and contained 6,986 die that passed all parametric tests. Of these, 220 failed the "Commercial" test pattern set while 229 failed the "Research" test pattern set. The total number of defective die that passed both test pattern sets is unknown, and we arbitrarily assume that a total of 12 die pass both test sets even though they are defective. (Note: This is an arbitrary assumption and is not meant to imply anything about the actual yield or defective part levels of TI integrated circuits. Furthermore, these experiments do not take into account the detection of any additional defects with other testing techniques, such as functional patterns and IDDQ.)

The second lot of production wafers was tested in February 1999. In this case, 20,591 die passed all parametric tests and were tested with the two test pattern sets. Of these, 246 die were declared defective using the DO-RE-ME test pattern set, while 245 die failed at least one of the patterns in the traditional commercial test pattern set. When calculating defect level we once again assumed that 12 defective die escaped both test pattern sets. The corresponding defect levels for the October and February experiments are plotted in Figs. 35 and 36.

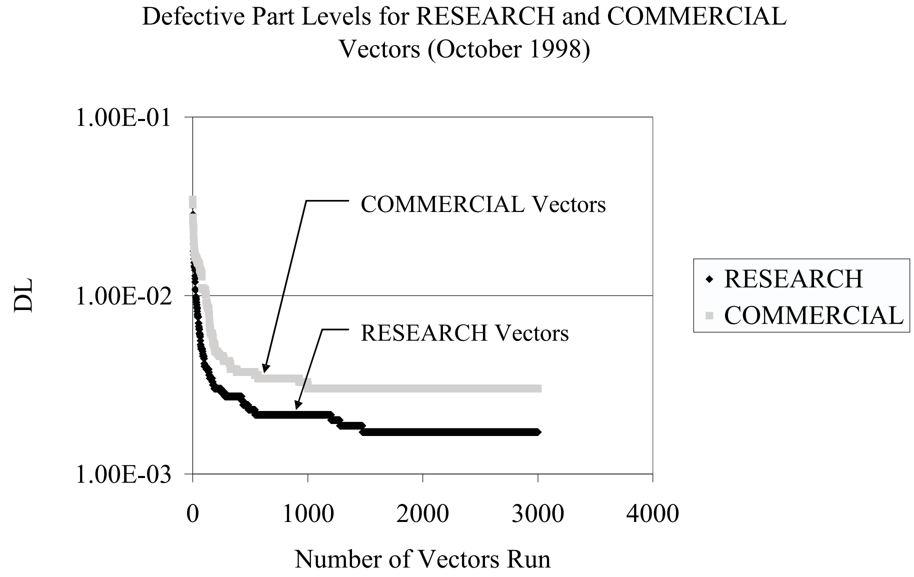


Fig. 35. Defect levels estimated for October production circuits

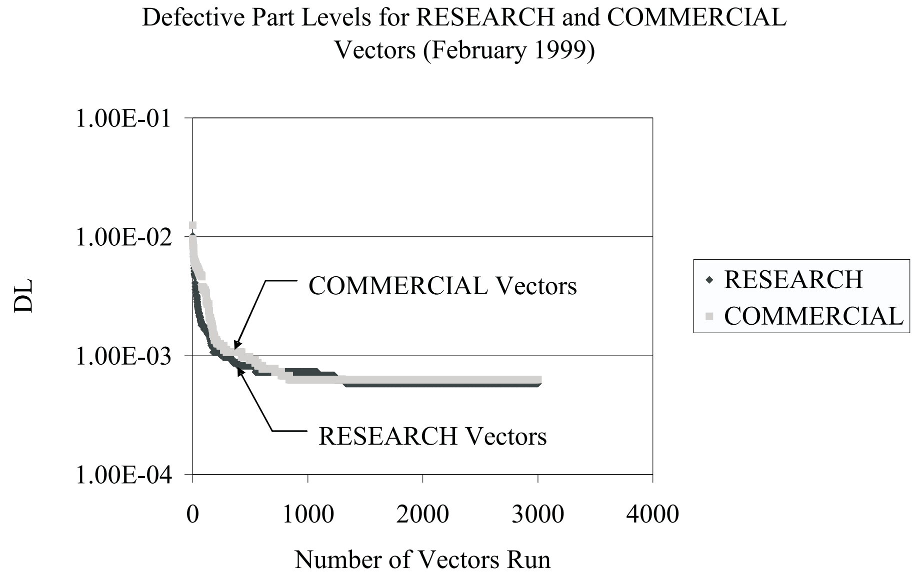


Fig. 36. Defect levels estimated for February production circuits

We wanted to find how well MPG-D could predict defect levels for commercial integrated circuits tested and analyzed under these conditions. Because this design was considerably larger than the benchmark circuits we had simulated, the commercial ATPG tool used at Texas Instruments was used to provide the fault dictionaries for defective part level analysis. However, the fault simulator used a collapsed fault list. (Thus, for example, only one representative of a set of equivalent faults should have appeared in the fault list. Faults are equivalent if their detection requirements are identical. For example, a stuck-at zero fault on one input of a two-input AND gate is equivalent to a stuck-at zero fault on the other input and to a stuck-at zero fault on the gate output. The only way to detect these faults is to set both inputs to the AND gate to logic ones.) Thus, all of our detection information on a pattern-by-pattern basis corresponded only to those faults in the list. Furthermore, we did not have access at Texas A&M to gate-level design information that would have allowed us to match a stuck-at zero fault at a circuit site with the corresponding stuck-at one fault at that site. Thus, we were not able to determine how many times each circuit *site* was observed. Instead, we were able to determine how many times each *fault* was observed. Thus, in each of the MPG-D calculations, we replaced sites with faults in the fault list.

The end effect is very similar. The detection/observation of a fault requires the observation of the site where that fault occurs. The primary difference is whether the value at the site in a good circuit is important. Stuck-at fault detection/observation requirements are more stringent than simple site observation because they require the site to be set to a particular value. Thus, there may be some advantage in using fault observations over site observations because it helps emphasize the importance of observing the site under different conditions—specifically observing the site when it is set to different logic values in the good circuit. The only disadvantage comes

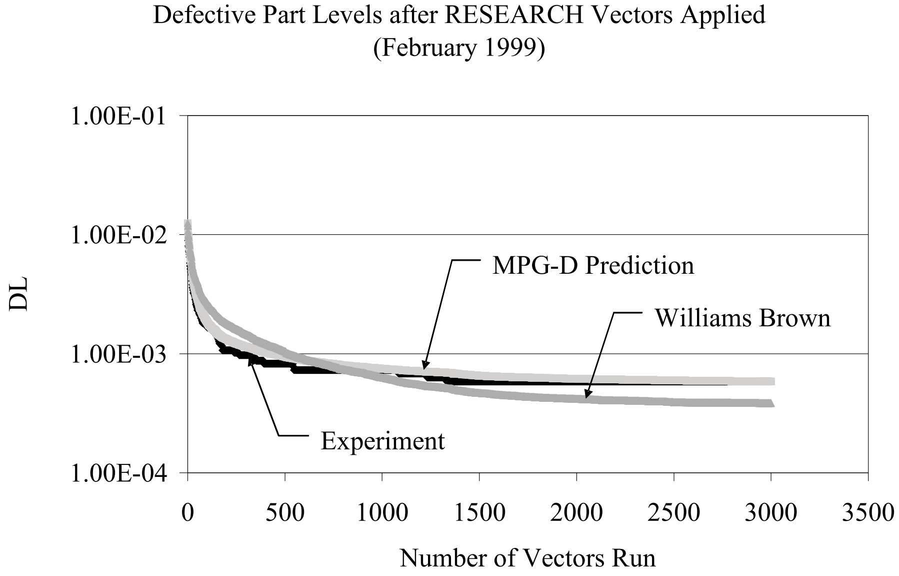


Fig. 37. Defect level predictions for February 1999 research set

when using the collapsed fault list. It is possible that a single fault may correspond to multiple sites, and we cannot take this account during MPG-D calculations unless we know how many sites are involved.

However, under the assumptions and conditions stated above, we were able to use the MPG-D and Williams Brown models to predict defect levels for both test pattern sets and both production lots. The results can be found in Figs. 37 to 40. We chose the constants to best match the experimental data from the February 1999 application of the “Research” test vector set and used the same constants to try to predict defect levels for the other three experiments. Proportional distribution of additional defect level removal was employed.

As can be seen from the graphs, MPG-D does a good job of predicting final defect levels for all of the cases except the October 1998 application of the “Commercial” test set. Furthermore, it does a better job of predicting defect levels than the Williams-

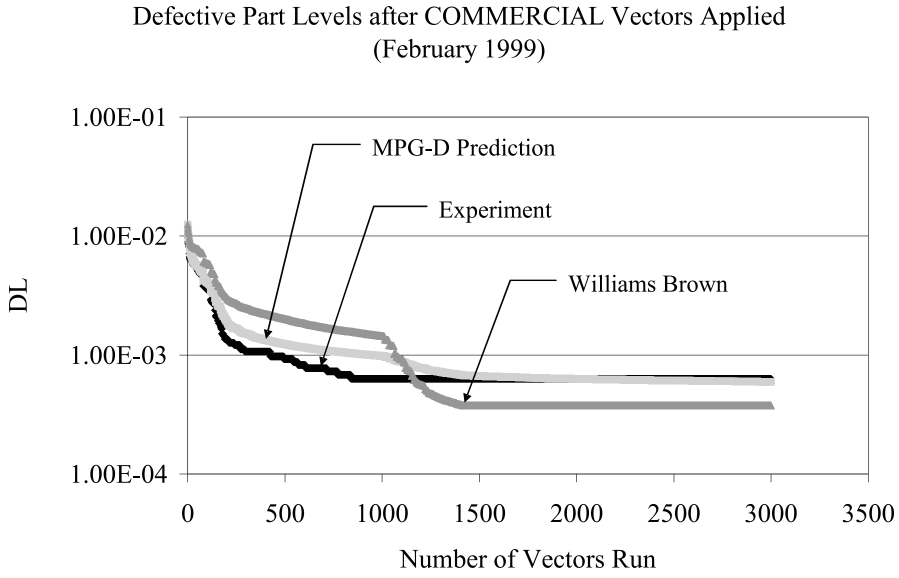


Fig. 38. Defect level predictions for February 1999 commercial set

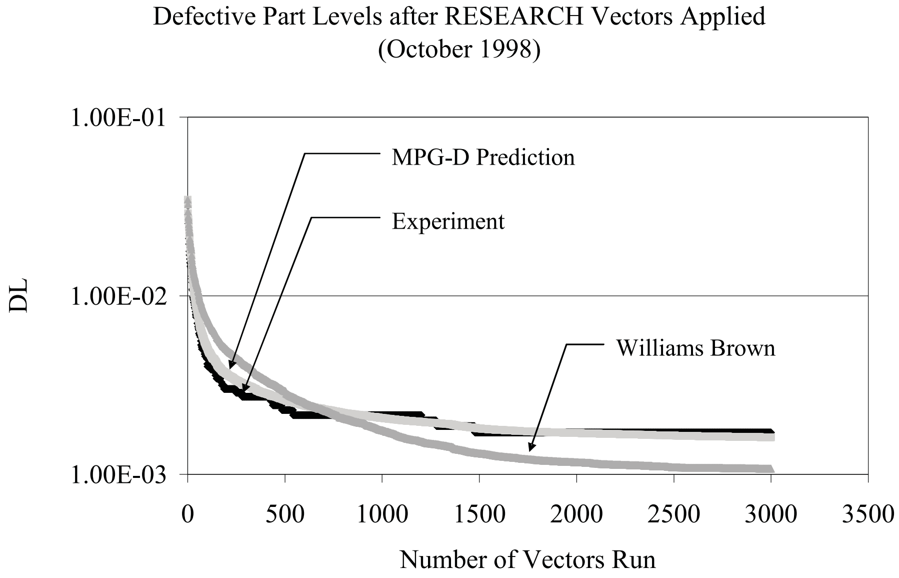


Fig. 39. Defect level predictions for October 1998 research set

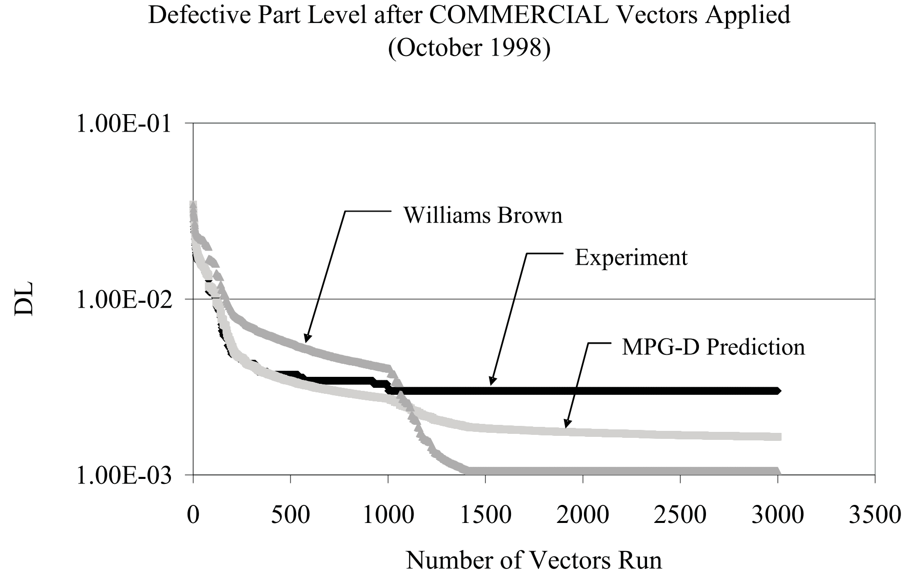


Fig. 40. Defect level predictions for October 1998 commercial set

Brown model in all cases. Most importantly, in contrast to the MPG-D model, the Williams-Brown model predicts that the “Commercial” test set should detect more defects because it has a slightly higher fault coverage. However, in reality, the “Research” test set is actually more effective for both lots of integrated circuits. Thus, MPG-D’s ability to incorporate more precise information about how many times sites were detected and how important each of those detections are for reducing defect level is a significant improvement over the simple stuck-at fault coverage generally used in the Williams-Brown model.

It was also encouraging to see that MPG-D was able to make fairly accurate predictions for three of the test pattern experiments with a single set of constants. Predictions continued to be good when the test pattern set alone changed (“Research” vs. “Commercial” Patterns) and when the defects changed (February vs. October). (There also appeared to be a significant difference in yield between October and

February—probably due to the maturing of the process—and this may have altered the nature of the defects present in the chips.) The only test set for which we did not get an excellent prediction was the "Commercial" test set applied in October. In this case, *both* the defects (and accompanying yield) and the test patterns changed, and this may have lead to the difficulty of using the same constants.

E. Benefits of the MPG-D model

Thus, we have shown that the MPG-D defective part level model has definite advantages in predicting defect levels of integrated circuits over traditional defect level models. One of the most important aspects is the ability to utilize observation data collected through fault simulation in order to make a more accurate and precise analysis of changes in defective part level than can be achieved through traditional single stuck-at coverage data alone. This is because, when single-stuck at fault coverage is used as a figure of merit for test pattern sets and to predict defective part level, it merely specifies whether or not each fault was detected at some point during testing or not. This is essentially a single bit of information, and while valuable, it certainly does not encompass all of the ways that test pattern sets interact with the circuit design and the defects present to determine the defect levels. Alternatively, some have also suggested specifying a minimum number of times each fault should be detected and evaluating a test pattern set based upon the number of faults that were detected at least that minimum number of times. This is an improvement over the simple single-stuck at fault coverage because it promotes additional fault detections and the accompanying site observations, but it is still essentially only a single bit of information. All that has changed is the threshold required to change that bit from a zero to a one to denote "detection."

In contrast, the MPG-D model essentially provides multiple bits of information that correspond directly to defect level (as opposed to fault coverage) because the model specifies how the observations of a circuit site will affect the defect level contribution that remains at that site. As described earlier, this amount changes based upon how many times the site has been observed previously (and how that affects the probability of exciting additional defects), and potentially also changes based upon sharing of defects with other circuit sites where those defects were actually observed and detected. Thus, not only are we able to make a more detailed analysis of defect level changes, but we also relate those changes to the actual physical probabilities and processes that affect defect detection. Thus, our model can be used to give intuition and guidelines for creating better test pattern generation strategies that detect more defects.

However, one of the most important aspects of MPG-D is that it allows different portions of the circuit to be analyzed with respect to their contribution to the overall defect level. It can be used to analyze which sites are most in need of additional observations and more importantly to specify the amount of defect level reduction that will likely occur with those additional observations. (This is especially true when defect sharing is not significant.) Thus, it can be used to guide the generation of a single test set by allowing tradeoffs to be made between different combinations of site observations. For example, it is possible to analyze the tradeoff between observing a few sites for the second time with one test pattern or observing many more sites for the third time with a different test pattern. Thus, the MPG-D model can be used to optimize test pattern sets and to choose a near optimal subset from a superset of test patterns.

MPG-D is also a significant improvement over MPG-1 from a calculation standpoint, particularly if the constant C is set equal to zero. When C equals zero, there is no defect sharing among circuit sites, and the only indicator of a site's remaining defect level contribution is the number of times that that site has been observed, independent of the number of observations of other circuit sites. Thus, it is possible to use the MPG-D equations stated earlier to calculate, at the beginning, the defect level contribution of a site with x observations. This can be stored for later reference in an array indexed by the number of observations. Then, as test patterns are applied and sites are observed, the values in this array can be retrieved based upon the current site's observation statistics and summed with the values at other sites—giving a defect level prediction for the application of all patterns up to that point. This is a significant improvement over the iterative calculation for MPG-1 and allows a near optimal subset from a superset of test patterns to be found in a reasonable amount of time—even for large circuits.

Finally, the success of the DO-RE-ME test pattern sets and the ability of MPG-D to in many cases accurately predict defect levels based upon site observations has confirmed the importance of increasing the number of observations of the least observed sites in order to generate better test pattern sets that detect more defects.

CHAPTER VII

INCREASING DEFECT COMPLEXITY

So far, we have focused on the fortuitous detection of static defects. In many cases these defects can be detected by satisfying relatively few constraints, and thus their fortuitous detection is not difficult. However, as circuits become faster and smaller there is concern that more complex defects (such as timing defects) will become increasingly important and that this could have a significant impact on ideal test pattern generation strategies.

A. The Meaning of Complexity

There are multiple ways that one could characterize defect complexity, but one of the most useful is probably as a function of the number of constraints that must be satisfied for detection of that defect to occur. For example, a test pattern that detects a single stuck-at fault must satisfy constraints related to the observation of the site where the fault occurs in addition to constraints related to the excitation of that fault. At a minimum, the excitation of a stuck-at fault will require that the site where the fault occurs be set to a particular logic value. In addition, meeting this constraint may also imply that other circuit sites upstream must be set to particular logic values. The observation constraints are a little more difficult to specify because there are often several paths through which a defect may be propagated. However, even in this case, there is still a minimum path length and an average path length along which the error caused by the fault must be propagated. For whatever path is ultimately chosen, there will be a series of constraints along that path which must be met for propagation of the error value to occur. Specifically, side inputs to the path (other inputs to the gates on the path through which the incorrect values do not propagate) should be set to non-controlling values (one in the case of an AND/NAND gate and zero in the case of an OR/NOR gate). Furthermore, setting the appropriate values for these side inputs may themselves imply that other sites must be set to specific values in their own incoming cones of influence. Finally, it is also possible that some sites in the circuit must be set to specific values so that excitation and observation can occur simultaneously. If we use this approach, defect/fault complexity is inherently linked to the difficulty of defect/fault detection.

However, we are not only concerned with the differences in defect detectability *within* a particular class. We would also like to compare the complexity *across* different defect classes.

B. Comparison of Static and Dynamic Defects

Up to this point, we have concentrated on the detection of static defects. However, as clock speeds increase, dynamic, or timing defects, are becoming more important. Unfortunately, dynamic defects are generally more complex than static defects, and this may have important implications for test pattern generation strategies and for our ability to achieve reasonable defect levels.

Static defects transform a circuit in such a way that it no longer realizes the correct function. In the steady-state (after all of the inputs have achieved their final values), there will be some input combination for which the defective circuit does not produce the appropriate values at the output. Since logic functions are defined as a mapping between input combinations and output values, this change in the output values for at least one input combination indicates that the circuit now realizes a different (erroneous) function.

In contrast, a dynamic defect does not change the steady-state behavior of the circuit or its logic function. If we wait long enough, the appropriate values will appear on the circuit outputs for every possible set of assignments to the primary inputs. However, we cannot wait indefinitely for this to happen. The correct values must appear at the outputs (and flip-flops) before the end of the current clock period. Thus, in most cases, we are concerned about dynamic defects that introduce additional delay into the circuit's operation. For at least one *pair* of input combinations applied in succession, there will be some transition of logic value that does not appear at an output quickly enough.

Because dynamic defects involve circuit timing instead of merely its logical behavior, obvious changes must be employed to successfully test for these defects. First, because a timing defect affects circuit operation when a circuit output (or flip-flop) does not change its value quickly enough, in order to detect such a defect, we must create *transitions* at circuit sites. We then must observe whether or not transitions occur quickly enough and are propagated to an output in the required time period. Thus, instead of applying a single pattern at a time to detect a defect, we must apply a test pattern *pair*. We must also capture the resulting values shortly after the application of the second pattern of the pair (in a time compatible with the clock period). Obviously, these considerations are not required for detection of a stuck-at fault; thus, a different fault model is needed if we are going to target timing faults during ATPG. The simplest fault model used for generating dynamic test pattern pairs is the *transition fault model*.

C. ATPG with the Transition Fault Model

A transition fault is defined as a circuit site that is either “slow-to-rise” or “slow-to-fall.” Thus, just as in the case of stuck-at faults, there are two possible transition

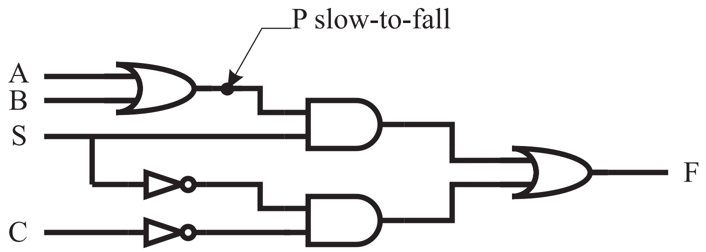


Fig. 41. Circuit containing transition fault

faults at every site, and the number of cases we must consider (the number of faults in the fault list) grows linearly with circuit size. As with all other faults and defects, a transition fault must be both excited and observed to be detected. The observation requirement is satisfied if the logic value at the site where the fault occurs is propagated to an observation point by the second pattern. In contrast, excitation requires that the appropriate logic value transition be created at the site in question by the application of two successive test patterns.

For example, consider Fig. 41. This is a modified multiplexer circuit, and we would like to detect a slow-to-fall transition fault at site P. Generating a test pattern pair for this fault will require exciting it by causing a falling transition at the site in a non-faulty circuit. (If the value at this circuit site does not fall, we cannot tell if it will fall quickly enough or not.) Thus, for the first pattern of the pair, we must set P to a logic one. We can do this by setting A to a logic one or B to a logic one. (Because the first pattern of a pair generates the circuit conditions to allow the circuit to start in the appropriate state for the transition required, we often refer to this first pattern as the “preconditioning pattern.”) A normal ATPG program will choose one of these two options and leave the other input as an unassigned “don’t care.” Let us assume that our ATPG program randomly chooses A to be set to a logic one and leaves B unassigned as shown in Fig. 42. At this point we have generated a partial pattern

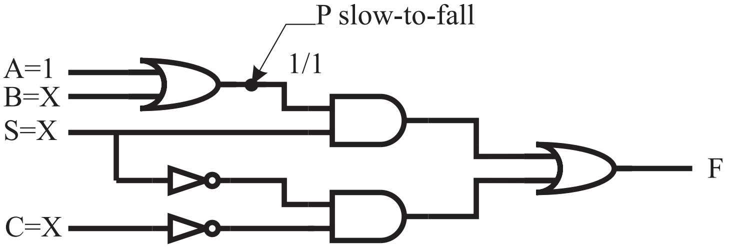


Fig. 42. ATPG preconditioning pattern for transition fault

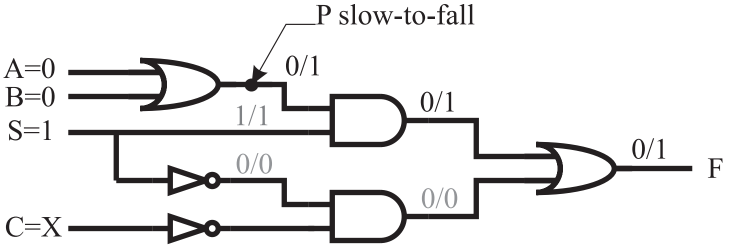


Fig. 43. ATPG observation pattern for transition fault

that fulfills all of our requirements for the necessary preconditioning and contains one assigned input and three unassigned inputs. Thus, when we randomly determine the unassigned inputs, we will be choosing one of eight possible patterns.

The second pattern in the pair must make sure that the appropriate transition is completed for fault excitation. Thus, point P must be set to a logic zero by this pattern. An ATPG algorithm will thus set both A and B to logic zeroes. As a result, once the second pattern is applied, the value at point P will "fall" in the good circuit. However, in the faulty circuit, point the value at P will not fall quickly enough, and it will erroneously remain at a high logic value. Thus it is after the application of the second pattern that a *difference* in logic value occurs between the good and faulty circuits. This is demonstrated in Fig. 43. (Recall that the value to the left of the slash is the value in the good circuit and the value to the right is the value in the faulty circuit.) It is this difference that must be propagated to an output and observed. Thus, with the second pattern, we must not only complete the requirements for excitation, but we must also observe the fault site. Thus, the second pattern in a pair is often called the "observation pattern." To accomplish this observation, the ATPG tool will set input S to a logic one, propagating the value at point P to an output. These three assignments are all that is needed for meeting all of the requirements for the second pattern. Thus, when a random assignment is made to the unassigned input, we will be choosing from one of two possible patterns.

It is also important to notice that the second pattern of the pair is identical to a test for point P stuck-at one. The additional constraint that must be satisfied consists of setting point P to a logic one in the first time period with the preconditioning pattern. Thus, by our characterization of complexity, transition faults are inherently more complex than stuck-at faults because an additional constraint must be satisfied by the preconditioning pattern. Intuitively, satisfying this additional constraint will make detecting a transition fault harder than detecting the associated stuck-at fault.

D. Fortuitous Detection for Static and Dynamic Defects

The extra constraint that must be met in the preconditioning vector makes dynamic defects harder to detect than related static defects. As a result, we expect that the fortuitous detection of these dynamic defects should be less than the fortuitous detection of static defects. To see why this is the case, consider Fig. 44. The box on the left is a Venn diagram containing all possible input combinations (or test patterns). Inside this box is an oval that contains all input combinations that will detect a particular static defect. In this case, given that we choose a random input combination, we appear to have approximately a 20% chance of fortuitously detecting this static defect with that test pattern. Furthermore, because this is a static defect,

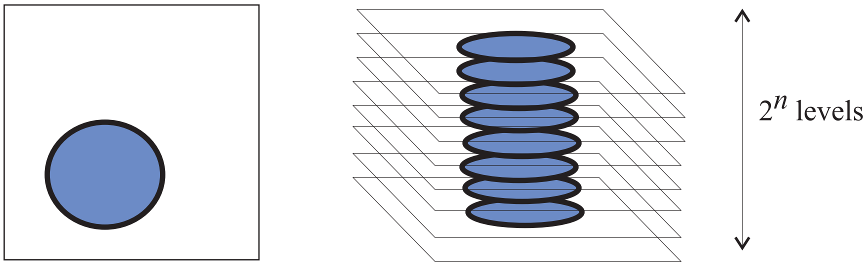


Fig. 44. Test space for static defect detection–single patterns or pairs

only a single test pattern is needed to meet both the excitation and observation requirements of this defect (provided that the right test pattern is chosen.)

However, let's assume that (for some reason) we are applying test pattern pairs instead of single test patterns. In that case, we have the situation presented on the right. Here we have a stack of Venn diagrams. These Venn diagrams are exact duplicates of the one on the left. In other words, they each contain all possible input combinations. However, now we have a copy of this Venn diagram for every possible preconditioning pattern. Thus, the stack of Venn diagrams is actually 2^n levels high for n circuit inputs. Because the static defect detection will occur (or not occur) regardless of what pattern is applied for the preconditioning vector, the oval appears in each level unchanged. Thus, for a random choice of all possible test pattern *pairs*, the probability of detecting this static defect is unchanged—it is still approximately 20%.

Now consider the case where we are interested in fortuitously detecting an untargeted defect given that we are already detecting a targeted defect. This situation is depicted in Fig. 45. Once again, the box on the left is a Venn diagram containing all possible input combinations, and within this box are ovals representing those test patterns that will detect a particular defect. Let's say that one of these two defects is modeled as a fault and deterministically targeted during ATPG. The second defect is

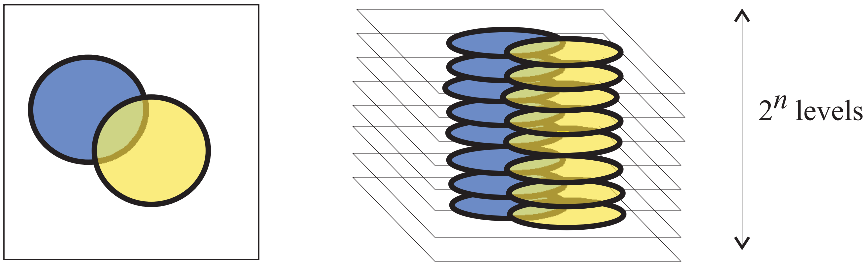


Fig. 45. Fortuitous detection of other static defects

not targeted. Then, the probability of fortuitously detecting the second defect given that the first defect is detected is merely equal to the size of the overlap of the two test spaces divided by the size of the test space of the first defect.

On the right side of Fig. 45, we once again have a stack of Venn diagrams where each element in the stack corresponds to a different preconditioning input pattern (assuming that we are applying test pattern pairs). Once again, because both our defects are static defects, their detection is independent of what preconditioning pattern was applied. Thus, the probability of fortuitously detecting the second defect given that the first defect was detected does not change when we consider the set of all test pattern pairs instead of the set of all test patterns. However, (not surprisingly) this will no longer be the case when we consider dynamic defects.

In the dynamic defect case, we need a preconditioning pattern for detection to occur because we need to ensure that the right transitions happen for defect excitation. Thus, detection is no longer independent of the preconditioning vector, and we now have a situation similar to the one in Fig. 46. Here we once again have a stack of Venn diagrams, but now the Venn diagrams are no longer identical. Specifically, some of the Venn diagrams in the stack contain no test patterns for this defect because the preconditioning pattern corresponding to that level does not allow for its detection. If we assume that this defect corresponds to a transition fault, then each of the levels

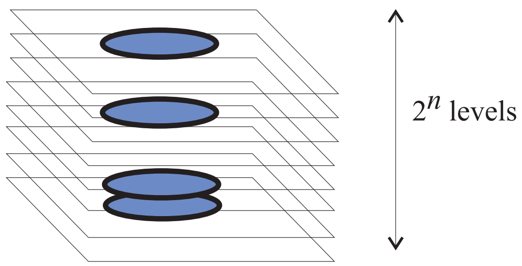


Fig. 46. Test space for dynamic defect detection

that contain an oval indicating detection will contain that oval that corresponds to the detection of the stuck-at fault associated with the transition fault. (For example, it will contain the test space for the associated stuck-at-zero fault if the transition fault is slow-to-rise.) Furthermore, only a single site will need to be set to a particular logic value by the preconditioning pattern for the transition fault excitation requirements in the first pattern to be met. Thus, if we assume that sites in the circuit are equal to a logic one for approximately half of all possible input combinations and are equal to a logic zero for the other half, then we are likely to find that we have met the preconditioning requirement in approximately half of the levels in the stack as shown in the figure. Therefore, if we consider the set of all possible test pattern pairs, a dynamic defect (in this case a transition fault) is less likely to be fortuitously detected than an associated static defect. Furthermore, in this particular example, we are about half as likely to fortuitously detect the transition fault as we are to detect the corresponding static stuck-at fault.

Now consider how the fortuitous detection of other dynamic defects is affected by the additional constraint in the preconditioning vector. This situation is illustrated in Fig. 47. Once again, we see that because these defects are dynamic and require that particular constraints be satisfied with the preconditioning pattern, the Venn diagrams at each level of the stack are not identical. Thus, the fortuitous detection

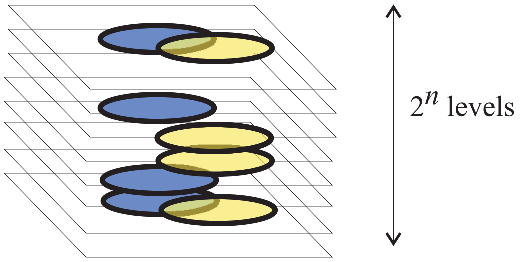


Fig. 47. Fortuitous detection of other dynamic defects

of the second defect given the detection of the first defect will depend not only on compatibility during the application of the observation pattern, but on compatibility during the application of the preconditioning pattern.

Let's think of the particular dynamic defects shown here as being analogous to transition faults and consisting of a static defect (such as a stuck-at fault) and one additional site value constraint that must be satisfied by the preconditioning pattern. Then, we will find that, for a given defect, approximately half of the levels will likely correspond to preconditioning patterns that allow for the potential detection of this defect. Unfortunately, there is no guarantee that the same preconditioning patterns will always be appropriate for both defects. In fact, if we assume that the values at the two circuit sites associated with the two defects are statistically independent, then we can say that approximately one-fourth of the preconditioning patterns will allow for the detection of neither defect, approximately one half will allow for the potential detection of only one of the defects, and approximately one fourth of the of the preconditioning patterns will allow for the potential detection of both defects. Fortuitous detection of the second defect given that the first defect is detected will obviously only be possible in the last case, and then it will depend upon the overlap in the Venn diagram for the corresponding observation pattern. Thus, the probability of fortuitously detecting the second defect given that the first defect was detected is equal to the total number of patterns in all of the overlaps divided by the total number of patterns in all of the levels that allow the first defect to be detected. Because the two defects can be detected in approximately one-fourth of the levels, and the first defect can be detected in approximately half of the levels, the probability of fortuitously detecting the second defect given the detection of the first defect in the dynamic case is about half of the probability of fortuitously detecting the second defect in the static case.

Thus, we expect fortuitous detection of another dynamic defect to be less than the fortuitous detection of another static defect—and in a case such as this where one additional circuit site must be set to the appropriate value with the preconditioning pattern in the dynamic case, we expect the amount of fortuitous detection to decrease by half. Obviously, the example displayed in Figs. 46 and 47 is only one representation of many possibilities. For example, we could find that the values at the different circuit sites are actually dependent—skewing our probabilities. We could also find that in some cases some of the other preconditioning patterns (that now allow for no detection of one of the defects) would allow detection if a different set of observation patterns (a different oval disjoint from the oval in the other levels) were chosen. However, in general the end result would be the same—the additional constraints required by dynamic defect detection would decrease the fortuitous detection.

To quantify the amount of the decrease in fortuitous detection for faults in a circuit when defects are dynamic instead of static, we analyzed the test pattern sets for both stuck-at and transition faults in c432 using Ordered Binary Decision Diagrams (OBDDs) [38]. OBDDs are very powerful tools for representing logic functions. They are directed graphs composed of nodes corresponding to the function variables where each node has two children—a zero child and a one child. Every path through the OBDD ends at one of two terminating vertices, indicating the value of the logic

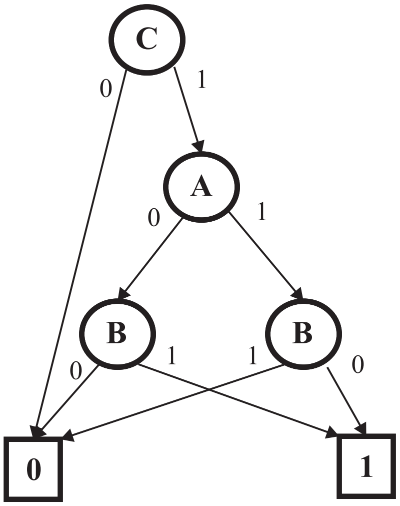


Fig. 48. OBDD for the function $F = C \bullet (A \oplus B)$

function corresponding to the assignments to the logic variables along that path.

For example, consider Fig. 48. This figure contains an OBDD for the function $F = C \bullet (A \oplus B)$. The input logic variables are A, B, and C, and each of the nodes corresponds to one of those logic variables as indicated. To evaluate an input combination, we merely choose to follow the one-path or the zero-path, depending upon the logic value assigned to that variable. Thus, for example, if we want to evaluate the function when $A = 1$, $B = 0$, and $C = 0$ we start at the root and begin traversing the graph. Since the first node corresponds to variable C, and C is equal to zero, we take the path to the left. Because this function is always equal to zero if C is equal to zero, we immediately reach the logic zero terminal vertex, showing that the function is equal to zero for this input combination.

In contrast, if we wanted to evaluate the function when $A = 1$, $B = 0$, and $C = 1$, we would begin by following the right path from node C. We would then reach a node labeled A. Because we have set A equal to one, we follow its one-path to the right and reach a node labeled B. Because B is equal to zero we follow its zero-path

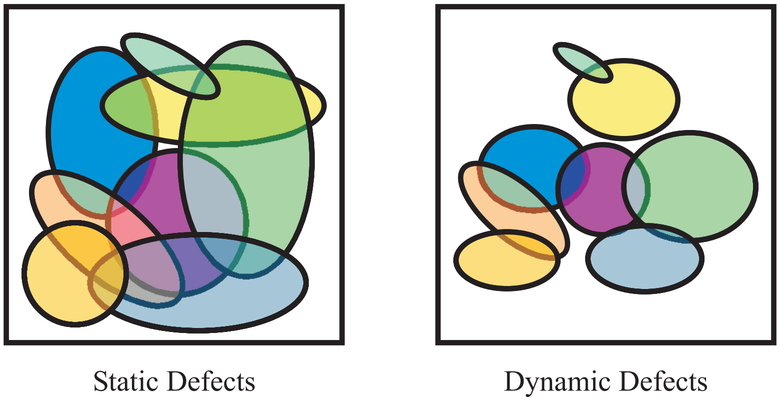


Fig. 49. Venn diagrams for static and dynamic defects

and reach the logic one terminal vertex, indicating that the function is equal to one for this input combination.

Ordered binary decision diagrams are very powerful because they can be manipulated to give important data. As just shown, we can use them to evaluate a given input combination. We can also do logic operations (such as AND, OR, XOR, or INVERT) on OBDDs and obtain an OBDD that is the result of those operations. They may also be used to find the number of minterms (input combinations that evaluate to a logic one for a given function). James Wingfield, a member of our research group, developed an OBDD tool to analyze the probability of fault excitation, observation, and detection, and to create test pattern sets [39]. It is also possible to use this tool to find the expected number of faults that will be fortuitously detected given the detection of a particular fault.

Consider Fig. 49 Let the box on the left be a Venn diagram containing all possible test patterns, and let each of the ovals contain those patterns which will detect a particular stuck-at fault. Because stuck-at faults are relatively easy to detect, there is a lot of overlap, and thus we expect to detect stuck-at faults given that particular fault was targeted and detected. In contrast, let the box on the right represent the set of all possible test pattern pairs, and let each of the ovals on the right represent those test pattern pairs that will detect a particular transition fault. The relative size of the ovals on the right is smaller than those on the left, and the amount of overlap is less, indicating we will fortuitously detect fewer transition faults than stuck-at faults. We would like to use OBDDs to quantify how much smaller the overlaps are and how much less the fortuitous detection is in the case of c432.

Thus, using the tool described in [39], we can create a *detection* OBDD for every stuck-at fault in the circuit. This detection OBDD will represent a function that is equal to one if that input combination detects the fault in question and zero otherwise. Thus, this OBDD is analogous to one of the ovals in Fig. 49. To find the overlap between two ovals, we merely need to AND the detection OBDDs together. The size of the overlap can be found by identifying the number of minterms in the resulting OBDD. We can do something similar in the case of the transition faults by also taking into account the preconditioning patterns. This data can then be used to find the expected number of fortuitously detected faults given that another fault f has been detected:

$$EFD(f) = \sum_{i=1}^{\#\ of\ faults-1} \frac{|\ detection\ set\ f \cap detection\ set\ i\ |}{|\ detection\ set\ f\ |} \tag{7.1}$$

This equation finds the size of the intersection of the detection set for fault f with the detection set for every other fault i in the fault list and divides by the size of the detection set for fault f. It essentially gives an estimate of the average number of faults that will be fortuitously detected.

We calculated EFD for every stuck-at and transition fault in c432, and the results appear in Fig. 50. In this figure, the stuck-at faults and transition faults have been sorted in increasing EFD values for the stuck-at faults. Thus, the transition fault that corresponds to a particular stuck-at fault is paired with that fault and

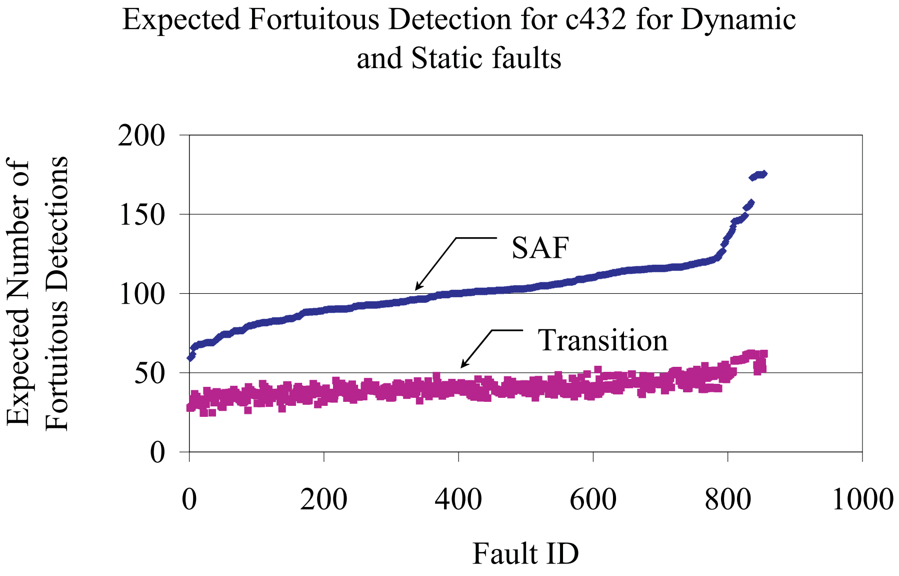


Fig. 50. Expected fortuitous detection for c432

appears at the same location on the x-axis (although a different location on the y-axis). It is apparent from this figure that, just as we expected, the expected number of fortuitous detections of transition faults given another transition fault detection is lower than the expected number of fortuitous detections of stuck-at faults given a stuck-at fault detection. Furthermore, if we take the ratio of the EFD values for a stuck-at fault and its associated transition fault, that ratio varies between 1.8 and 3.4. This matches well with our earlier analysis of the effect that the extra condition required for transition fault detection has on fortuitous detection. In general, we are fortuitously detecting approximately half as many transition faults as stuck-at faults.

The primary effect of this change in the amount of fortuitous detection is an increase in test set length, especially if aggressive compaction techniques are not employed. For example, in [40], we generated test sets by choosing test patterns from the targeted fault's detection OBDD. We showed that the average number of test pattern pairs needed to obtain a minimum number of detections of every fault was roughly twice (1.45 to 2.48) the number of patterns needed to obtain that same number of minimal detections of every stuck-at fault. Thus, as faults become more complex and additional constraints are required for fault detection, the number of resources devoted to testing will increase. These resources will either include the additional time and tester memory needed to apply longer test pattern sets or additional ATPG effort to find and exploit any remaining areas of overlap among the fault detection sets.

CHAPTER VIII

ANALYSIS OF COMPLEX DEFECT DETECTION

While faults and defects are not identical, there is an obvious relation between them. Specifically, while defects are physical imperfections that occur in integrated circuits and adversely affect their behavior, faults are *models* of defects used for test generation. Thus, just as we have shown that the fortuitous detection of faults decreases when faults become more complex, the fortuitous detection of defects also decreases when those defects become more complex. This has the potential to significantly affect defective part levels.

A. Defect Levels for Static and Dynamic Defects

To investigate the effect that additional constraints have on the detection of untargeted dynamic defects and the resulting defective part level, we once again used surrogate simulation to analyze detection rates. Recall that *surrogates* are models of defects that are not targeted during ATPG, and thus, their detection or non-detection by a test pattern set serves as an indication of how unmodeled defects may be detected in general by that set.

For all of the following experiments, we used logical crosstalk surrogates as our surrogates for dynamic defects. They were essentially transition faults with two additional site constraints. Specifically, the aggressor line needed to be transitioning in the opposite direction as the victim line. Thus, for example, assume that site A is the victim, site B is the aggressor, and that we are trying to detect that site A is slow-to-rise because of crosstalk. In that case, we need to detect the transition fault “A slow-to-rise” while site B has a falling transition. Thus, site B must be set to a logic one by the preconditioning pattern and a logic zero by the observation pattern.

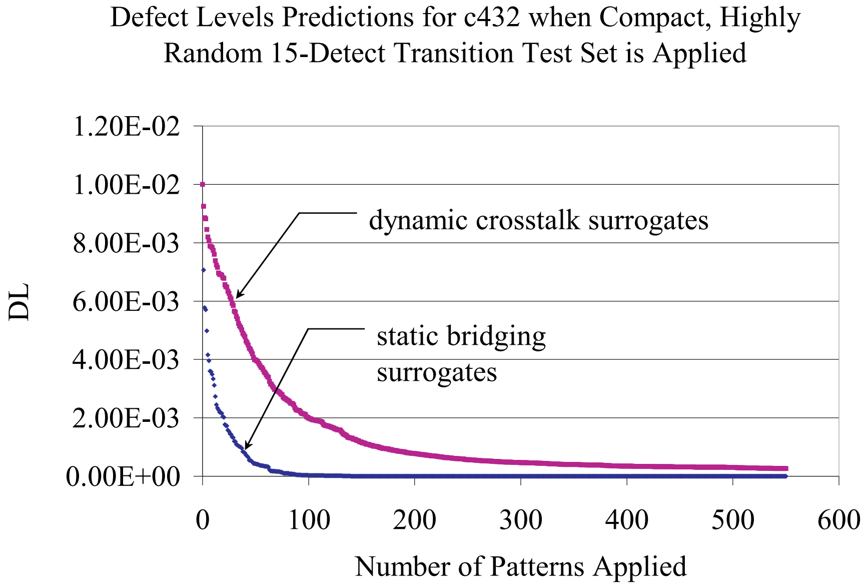


Fig. 51. Defect levels from static and dynamic surrogate simulation

To compare the detection of unmodeled static and dynamic defects, we ran surrogate simulation with both dynamic and static surrogates. The dynamic surrogates were the crosstalk surrogates described above while the static surrogates were the AND/OR bridges described in previous chapters. We generated a 15-detect transition test pattern set to apply to both sets of surrogate defects. (Recall that the second test pattern of a test pattern pair is actually a stuck-at pattern. Thus it is possible to catch static defects (such as those modeled by stuck-at faults) with the observation patterns of a transition test set.) We used bridging surrogate simulation to determine which AND/OR bridges were detected with each observation pattern of a transition test pattern set, and we used crosstalk surrogate simulation to determine which crosstalk surrogates were detected by each pattern pair. Defective part levels were calculated in both cases, and the results can be seen in Fig. 51 where the scale for the y-axis is linear and in Fig. 52 where the scale for the y-axis is a log scale. It

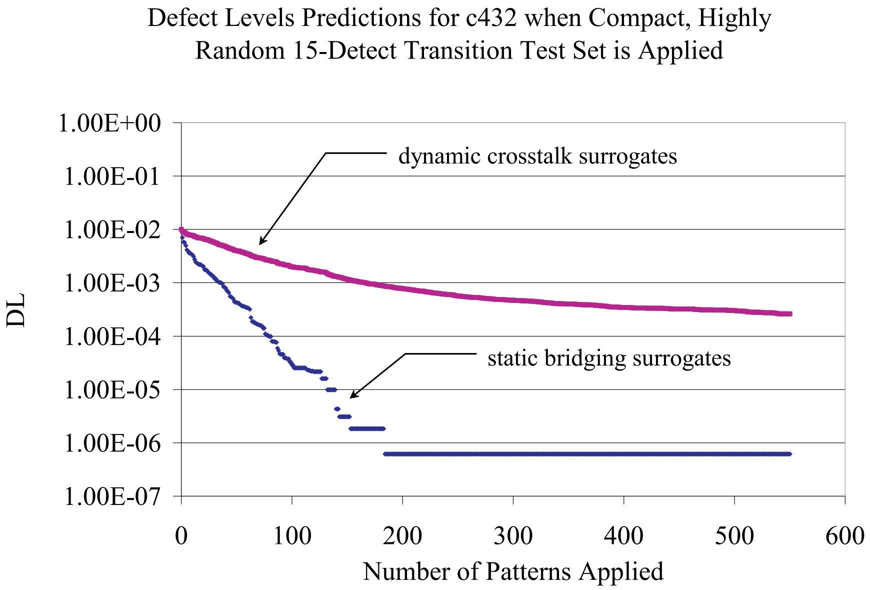


Fig. 52. Defect levels from static and dynamic surrogate simulation—log scale

is obvious from this figure that the static surrogates were much easier to detect than the dynamic surrogates. The defective part level falls off much more quickly for the static bridges than for the dynamic crosstalk surrogates. Even after every transition fault has been observed a minimum of 15 times, there are still many surrogates that remain undetected. In contrast, almost all of the static surrogates are detected both when a minimum of 15 detections of every stuck-at fault has been achieved (with approximately 92% of the observation patterns) and when a minimum of 15 detections of all transition faults has been achieved and all of the observation patterns have been applied. In fact, after all of the tests have been applied, the defective part levels due to static and dynamic defects actually differ by several orders of magnitude.

This seems to indicate that if we are able to expend enough resources to do a good job of testing for dynamic defects, that we will likely automatically also do an excellent job of detecting static defects with those same test patterns. This is especially true if the patterns in a pair are independent, and every possible pattern is a potential candidate for the observation pattern of the pair. Thus, a prudent distribution of testing resources would likely involve generating test pattern pairs with the intention of maximizing dynamic defect detection and allotting significantly fewer resources to the explicit detection of static defects.

Another important conclusion that can be drawn from this data is that even though a minimum of 15 detections of every stuck-at fault may be sufficient for excellent static defect detection, a minimum of 15 detections of every transition fault may be insufficient for correspondingly good dynamic defect detection. Thus, increased complexity of dynamic faults and defects will naturally lead to the expenditure of additional testing resources. However, there is a subtle difference between the mechanism of increase that results from more complex faults and the type of increase that results from more complex defects.

In the case of more complex *faults*, test set *lengths* for a given minimum number of fault detections will likely increase. Alternatively, significant computational resources may be employed to minimize the increase in test set length at the expense of massive amounts of ATPG time and memory resources. (For example, in [41] we showed that using OBDDs to find the complete detection sets for all faults and explicitly finding their intersections can be used to achieve the same number of minimal fault detections with very few additional patterns pairs. However, generating and manipulating these OBDDs is very expensive in terms of computational time and computer memory—limiting the capability of using these full OBDDs to generate test sets for industrial circuits.) In contrast, in the case of more complex *defects*, the *minimum number of fault detections* or the *minimum number of site observations* required to achieve sufficient fortuitous detection of the unmodeled defects will likely increase. Thus, while more complex defects are likely to lead to longer test set lengths as well, the reason for those increased test lengths is different, and unfortunately, an increase in the needed number of site observations is unlikely to be as amenable to compaction techniques.

B. The Observation Criteria

Because the additional constraints that must be satisfied for the detection of more complex dynamic defects can make them so much harder to detect, it is important to generate test pattern pairs carefully to maximize their fortuitous detection. In the case of the less complex static defects that we studied earlier, the way to increase the chance of fortuitously detecting more defects was by focusing upon deterministically observing the sites in the circuit that would normally be observed the fewest number of times more often. As a result, the MPG-D predictions of test set effectiveness calculate defective part levels as a function of the number of times different circuit sites were observed. To a large extent, it seems as though this focus on maximizing site observation count should carry over to the case of more complex defects with more detection requirements—especially when the specifics of those additional detection requirements are unknown. The primary expected difference would be an increase in the minimum number of site observations, as stated above.

To investigate the importance of site observation to the detection of more complex dynamic defects such as our crosstalk surrogates, we performed surrogate simulation on the benchmark circuit c432 when different test pattern sets were applied. Two different ATPG tools were used to generate the test sets. The sByDDer tool described in [39] used OBDDs to find all possible test pattern pairs for detecting a particular targeted transition fault. Of these, a single pair was randomly chosen, and the other faults fortuitously detected by this pair were identified before the next “least detected” transition fault was targeted.

The second ATPG tool was a “greedy” ATPG tool described in [42]. This tool uses a greedy algorithm to produce compact test sets that detect faults multiple times. Unlike the OBDD tool, which operates in the functional realm, this tool simulates 32 patterns in parallel at the logic gate level. Inputs are chosen in groups of five, and all thirty two possible input assignments for those five inputs are simulated. Guaranteed and potential fault detections are identified, and evaluated according to a “greed” metric which takes into account whether a fault detection is guaranteed or only possible in addition to how many times that fault had been detected in the past. The best assignment to the five inputs is then chosen based upon the values of the “greed” metric for the 32 potential assignments.

Because 15-detect test sets are generally considered close to ideal for the detection of static defects, initially both tools were used to generate test sets that detected every transition fault a minimum of fifteen times. Thus, the minimum number of times a site was observed was thirty. Surprisingly, the data showed a significant difference in the amount of crosstalk surrogates detected the two test pattern sets. Specifically, the test set generated with OBDDs detected many fewer surrogates than the greedy test set. Although some difference was to be expected, we did not expect it to be very large because the minimum number of site observations was identical in both cases. However, the exact number of observations of a particular site differed depending upon which test pattern set we were considering. Thus, it was thought that possibly the greedy test set was simply doing a better job of observing sites overall or just happened to better observe those sites where crosstalk surrogates were occuring. To investigate this possibility, a new test pattern set was generated by sByDDer that detected every transition fault a minimum of thirty times. Thus, every site in the circuit was observed a minimum of sixty times. It was expected that this would force

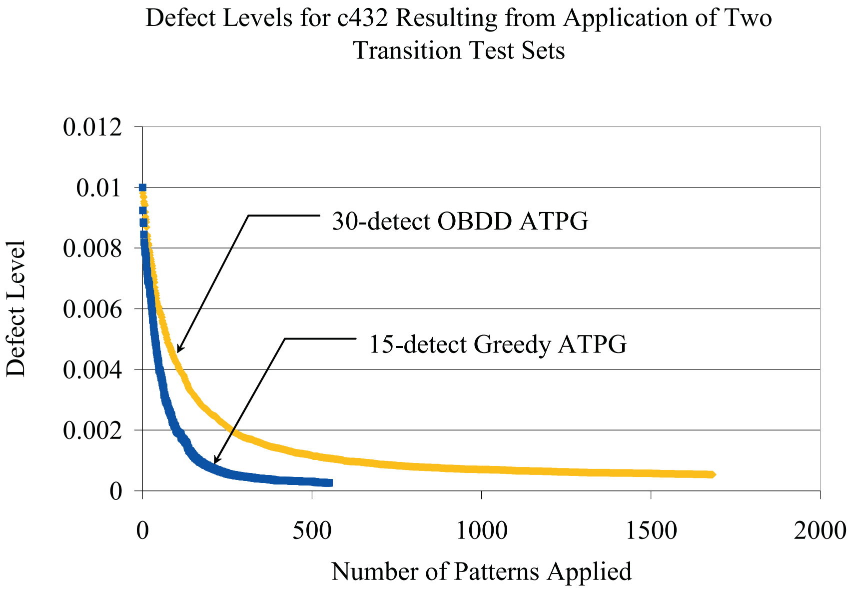


Fig. 53. Defect levels from OBDD and Greedy ATPG algorithms

the OBDD test set to “win” and detect more defects. The resulting defect levels are shown in Fig. 53.

As is apparent from the figure, even greatly increasing the number of site observations (and transition detections) did not allow the 30-detect OBDD test set to match the surrogate detection of the greedy 15-detect set! This was incredibly counterintuitive. In the past, significant increases in the number of site observations always led to additional reduction in defect level. However, in this case, the greedy test pattern set not only reaches a low defective part level more quickly than the OBDD test set, but it achieves a lower final defect level with fewer patterns and significantly *fewer* observations. For some reason, the additional site observations achieved by the 30-detect test set were not effective.

To try to understand this anomaly, we went back to the principles underlying the DO-RE-ME method. Recall that DO-RE-ME stands for Deterministic Observation, Random Excitation, and MPG-D Defective Part Level Estimation. Although multiple observations of circuit sites will obviously be required to achieve high levels of complex dynamic defect detections, in this case, deterministic observation obviously cannot explain the results shown in Fig. 53. Thus, we turned to the "randomness" of excitation. (Here the randomness of excitation refers to the excitation of the unknown defects—not the targeted faults.) We hypothesized that in the case of complex dynamic defects, truly random excitation might become more important to ensure that additional observations of a circuit site would actually be effective in reducing defective part levels. The questions that remained were how to define and quantify this "randomness" and whether or not this would explain the results in Fig. 53.

CHAPTER IX

RANDOMNESS AND EXCITATION BALANCE

The second component of the DO-RE-ME method is the random excitation of the untargeted and unknown defects whenever circuit sites are observed. This randomness involves ensuring that different patterns and thus different circuit conditions are used whenever circuit sites are observed. For example, observing a circuit site multiple times with an identical test pattern obviously will not detect any additional defects (unless some other circuit conditions such as voltage or temperature have changed.) In the static case, we seemed to obtain good defect coverage without paying a great deal of attention to the quality of the random excitation. Simply ensuring that random decisions were made during the ATPG process and that unassigned inputs were randomly filled after the generation of a partial pattern seemed to be sufficient. However, it seems that as more circuit constraints are added and defects become more complex, the quality of the randomness of the excitation may indeed become more important.

A. Excitation Requirements

The difficulty inherent in exciting unknown defects is related to the uncertainty regarding what conditions must be satisfied for that excitation to occur. These excitation requirements differ in a broad sense from one defect type to another and, on an individual basis, within a particular defect class. For example, in the static defect case, a bridging defect requires that both ends of the bridge be set to opposite values for defect excitation to occur. This is in contrast to a static defect that can be modeled as a stuck-at fault where only a single site must be set to a specific value. Furthermore, even if we somehow knew that our defect was guaranteed to be a bridge,

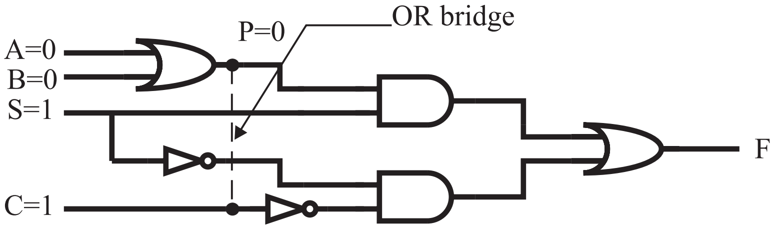


Fig. 54. To excite OR bridge for observation at P, set P to zero and C to one

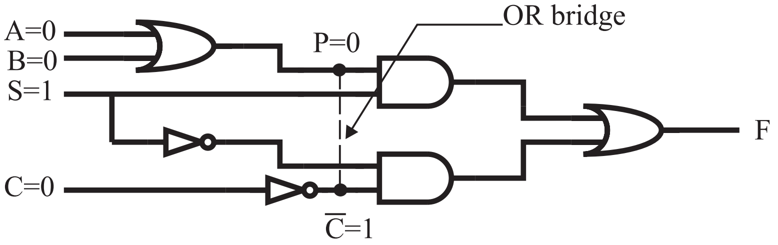


Fig. 55. To excite OR bridge for observation at P, set P to zero and $\overline{C}$ to one

we still wouldn't know *which* bridge was occurring or how to excite it. For example, in order to excite the OR bridge in Fig. 54, we must set both ends of the bridge to opposite values. This can be achieved with the pattern shown because P is set to a logic zero while C is set to a logic one. In contrast, to excite the OR bridge in Fig. 55, P and $\overline{C}$ must be set to different logic values. One way to achieve this is by setting $\overline{C}$ to a logic one by setting C to a logic zero. Thus, without knowing which bridge is present, we don't know if the value at site P must differ from the value at C or $\overline{C}$.

Furthermore, the exact requirements for excitation of a defect (so that it can be detected by a given test pattern) may change depending upon which site is observed and how that observation is achieved. For example, in both Fig. 54 and Fig. 55, site P is observed at output F because S is set equal to 1. In addition, setting S to a logic 1 makes it impossible to observe site C or site $\overline{C}$ with this pattern. Thus, in order to detect this defect, we must not only excite it, we must also excite it in such a way as to allow the error to appear at point P. This is done in both figures by ensuring that P is set equal to zero in the good circuit. Then, in the defective circuit, it is site P that will be forced to a logic one by the other end of the OR bridge in a defective circuit. However, if the bridge involved had been an AND bridge instead of an OR bridge, then we would have had to set P to a logic one in the good circuit for the error to appear at point P when the defect was excited.

Thus, it is incredibly difficult, and often impossible, to predict what any particular site should be set to in order to excite an unknown defect. Not surprisingly, this difficulty does not decrease as defect complexity increases and we try to detect dynamic defects instead of static ones. For example, Will Moore and his colleagues investigated delays due to crosstalk, opens, resistive bridges, and power rail coupling [43]. They found that the mechanism for maximizing the chance of detecting delays due to dynamic defects varied depending upon the type of defect. For example, in the case of crosstalk, the aggressor must transition in a direction opposite to the victim. However, in the case of a resistive bridge, the best chance of detecting the bridge may result when the bridged aggressor is kept constant instead of transitioning. Furthermore, they suggest that other aggressors that are not involved in the bridge should be transitioned. Overall, they found that the detection of particular defects is highly susceptible to both the values at other circuit nodes and the presence of noise, making it difficult to specifically model the detection requirements for deep sub-micron timing defects. Thus, they suggested that focusing on using site observation as we had suggested for static tests might be the best solution. In addition, experiments at Stanford that involved applying different dynamic test pattern sets generated with different fault models to manufactured circuits showed that none of them (transition, stuck-open, path delay, etc.) detected as many defects as an n-detect test pattern set of similar length applied at-speed [44]. Thus, it seems to be very difficult to adequately model dynamic defects to reliably predict what their detection requirements may be.

Because it is so difficult to determine what the excitation and detection requirements for unknown defects may be, it is likely impossible to find a way of deterministically detecting them. The best we are likely to do is to maximize our chances of *fortuitous* detection by ensuring that our excitation is truly random when we observe sites multiple times with different test patterns.

In the past, other researchers have proposed different ways of promoting diversity among test patterns in a given test pattern set. For example, in previous work on increasing the diversity among test patterns in an multi-detect set, some have suggested ensuring that different sets of stuck-at faults be detected simultaneously [45] or that two different detections should be counted only if a vector composed of the common primary inputs (where primary inputs that disagree are set to X) no longer detects the fault [46]. While both of these methods would help ensure different patterns, both are focused on faults instead of defects. Other related work has considered excitation diversity in the context of the circuit's physical design (such as the physical region around a signal line) [47]. Furthermore, [31] suggests that faults should be propagated to multiple observation points to increase diversity and discusses the corresponding Fault Observation Coverage (FOC). However, these methods do not truly provide a convenient metric for quantifying the quality of the excitation for random defects and the amount of diversity.

B. Excitation Balance

To create a good metric for evaluating the quality and "randomness" of defect excitation, we must consider how to best excite unknown defects. Unfortunately, because the actual excitation requirements are so unpredictable, it is difficult to purposely bias the excitation in favor of meeting the requirements of those defects. Most likely, the best we can do is ensure that our test patterns are not biased *against* meeting those requirements. We propose to do this by evaluating *excitation balance.*

To ensure that we are not biased against meeting the excitation requirements for unknown defects, we would like each of the circuit sites to be as likely to be a logic one as a logic zero. Thus, ideally the logic probabilities at every site would be perfectly *balanced.* Unfortunately, achieving such a balance is not necessarily easy—especially because we want site values to be balanced not only in the context of the entire test pattern set, but also in the context of those patterns that observe a particular site.

Every time a site is observed (or a fault is detected) with a given test pattern or pattern pair, some circuit sites must be set to particular values for that observation or detection to occur. Of course, it is possible to observe a site through different paths or to excite a targeted fault in different ways. Given that we do not know if one of these ways is "better" for the detection of an unknown defect, we would ideally like to encourage using as many of these options as possible instead of always observing the site along the same path or exciting a fault with the same logic value assignments. However, unfortunately, some set of value assignments will often be required whenever a site is observed. Thus, as opposed to the ideal situation, there is generally a distribution of probabilities among circuit site values as shown in Fig. 56. This figure depicts an example of how some sites may be more "balanced" than others for those patterns that observe a particular site or detect a particular fault. The sites

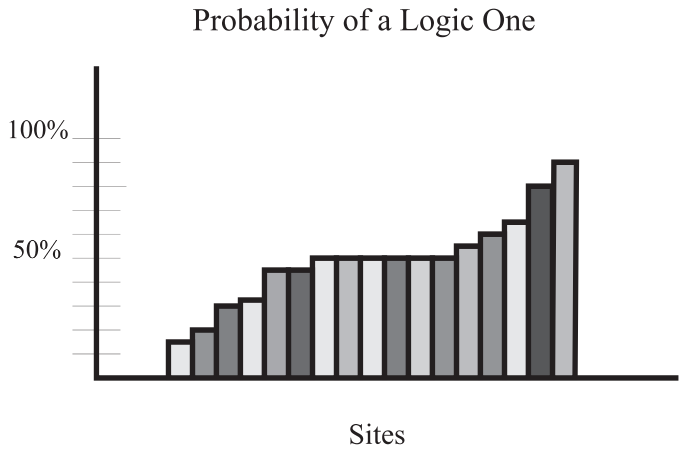


Fig. 56. Example distribution of ones probabilities when a site is observed

in the middle of the figure are equally likely to be a logic one as to be a logic zero. If any of these sites needed to be set to a particular value to allow for the excitation of an unmodeled defect, we are equally likely to meet that constraint as not meet it with a single observation of the site in question. In contrast, sites to the left of the figure are much more likely to be a logic zero than a logic one. This is obviously not problematic if the required site value to detect a particular defect is a logic zero; however, we have no guarantee that that will be the case, and even if it is true for some defects, it is very unlikely to be true for all of them. Similarly, the sites to the right of the graph are much more likely to be one.

Thus, given that we have a variety of unknown defects with unknown detection requirements, an ideal distribution of site probabilities can be found in Fig. 57. Of course, this is highly unlikely because, when meeting the requirements for site observation, certain sites will naturally be more likely to equal one (or zero) to allow that observation to occur. However, we would like to get as close to this ideal as possible. Thus, we propose to estimate the quality of the excitation balance for those test patterns that observe a site (or detect a fault) by taking into account the difference

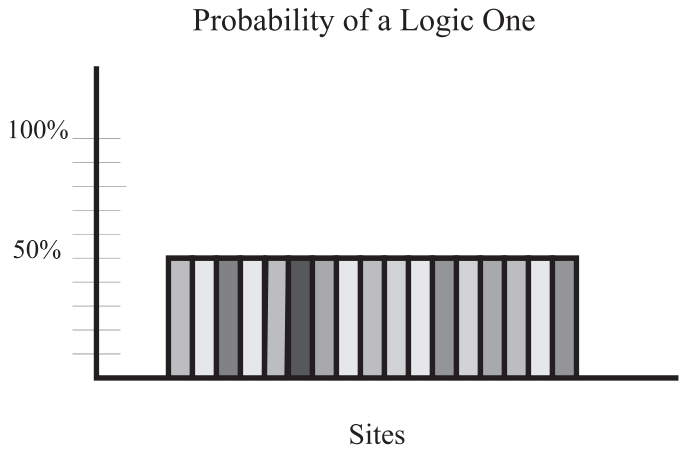


Fig. 57. Ideal distribution of ones probabilities when a site is observed

between each site's ones probability and the ideal—in other words, we are interested in how far a site's one probability is from 50%.

Unfortunately, problems that result from a site not being equally likely to be a logic one as a logic zero increase as the distance from 50% increases. For example, consider Fig. 58. This figure shows the probability of satisfying a single additional site constraint given that some other site in the circuit is being observed. The number of observations required before we can have high confidence of having met this constraint depends upon the probability of the required site being a logic one or a logic zero.

Let's assume that we need this site to be set to a logic one in order to excite a defect and cause an error at the observed site. Then, consider what happens as the probability of getting a logic one at this site changes. If there is a 50% chance that the site will be equal to one when the other site in question is observed, then we rapidly gain great confidence that we have met this requirement. By the time the site has been observed 6 or 7 times, the probability of having met this constraint is greater than 98%.

In contrast, consider what happens when the chance of the site being equal to

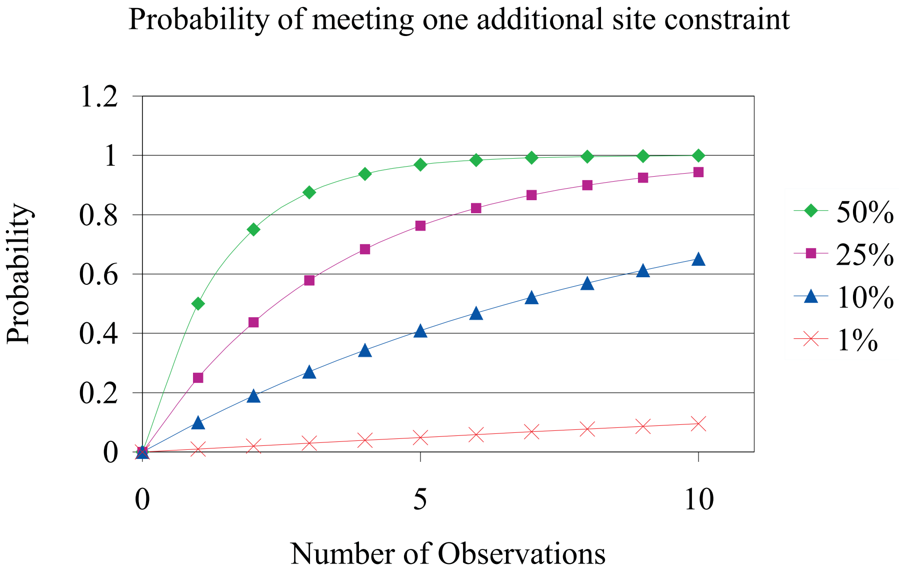


Fig. 58. Comparison of the difficulty in meeting constraints with less balance

one is lower. If the ones probability is equal to 25%, then after 10 observations we only have a 94% chance of having met this requirement. This isn't terrible. Many sites in the circuit will be observed greater than 10 times during the application of even a short test pattern set. However, we obviously have less chance of detecting the additional defect in this instance. If the ones probability of the site is only equal to 10%, then the curve rises toward one even more slowly. Although it can't be seen on the graph, 27 observations will be required before we have a 94% chance of having met this additional constraint and 38 observations will be required before we have a 98% chance of meeting this constraint. Finally, if we have a ones probability of only 1%, then we will need an incredible number of observations before we have a decent chance of having satisfied this condition. In fact, 280 observations will be required before we have a 94% chance of having satisfied it, and 390 observations will be required before we have a 98% chance of having had the site equal to a logic one while the other site in question was observed. Thus, the more skewed a site's value is, the more troublesome meeting a contrary site value requirement will be.

All of this was taken into account when we designed a metric to estimate the quality of excitation balance. The result can be found in Equation 9.1 below.

$$Excitation\ balance_j = 0.25 - \frac{\sum_{i=1}^{\#\_of\_sites}(0.5 - ones\_prob_{i|j\ observed})^2}{\#\_of\_sites} \tag{9.1}$$

This equation calculates the excitation balance for a given site j. The ones probability is calculated for every site i in the circuit given that site j is observed. The difference between the ones probability of site i and the ideal value of 50% is calculated by subtracting from 0.5. To emphasize the effect of those sites which are highly skewed, this difference is then squared. The resulting number is that site's contribution to the "unbalance" in the circuit. These values are then averaged for all circuit sites. Finally, because we want to calculate excitation balance and would like high numbers to signify better balance, we subtract the average from its maximum value of 0.25. The resulting number tells us how balanced the values at circuit sites are whenever site j is observed.

We used this equation to estimate the quality of the excitation for the OBDD and Greedy test pattern sets described earlier. In this case, we calculated the excitation balance numbers given that each transition fault was detected (instead of given that every site was observed). We calculated excitation balance for both the preconditioning and observation pattern of each test pattern pair. Little difference was seen between the excitation balance numbers for the two test sets with respect to the preconditioning patterns, but a dramatic difference was apparent with respect to the observation patterns. The results are shown in Fig. 59.

In this figure, the excitation balance numbers for the observation patterns of

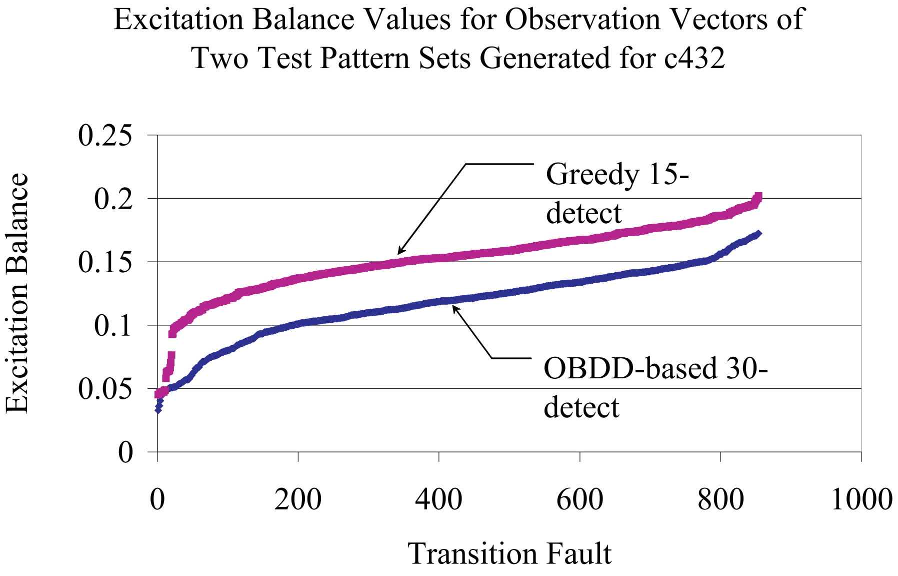


Fig. 59. Excitation balance for observation patterns of Greedy and sByDDer test sets

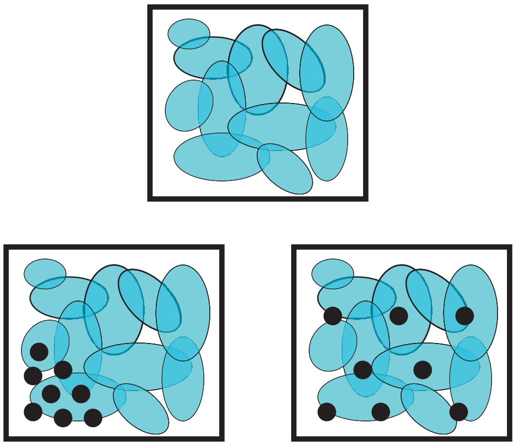
both test pattern sets are shown. For each set, the faults were sorted in terms of increasing excitation balance numbers. (Thus, fault 100 for the Greedy set may not be identical to fault 100 for the OBDD test set.) It is apparent that (aside from a few exceptions where both test sets have low excitation balance numbers) the Greedy test set has considerably better excitation balance statistics than the OBDD test pattern set. Thus, this seems to support our hypothesis that the difference between these two test pattern sets (that led to the very puzzling defect level results) was the difference in the quality of excitation.

Thus, while the OBDD test pattern set has significantly more observations, the values at its circuit sites seem to be more skewed for those patterns that detect each transition fault. This has the effect of making each of those subsequent observations less effective because many circuit conditions remain the same from one fault detection/site observation to another. In contrast, the Greedy test pattern set seems to detect faults with more diversity in circuit conditions, and thus the probabilities at circuit sites are less skewed. This increases the effectiveness of each of the site observations for the Greedy test pattern set and apparently lead to its ability to achieve a lower defective part level.

C. Excitation Balance and τ

In the MPG-D model, the constant τ determines how quickly the probability of exciting at least one additional defect decreases as a circuit site is observed more times. When τ is relatively high, later site observations are more likely to detect at least some defects. In contrast, when τ is low, the probability of excitation falls dramatically with additional site observations, and those additional observations are likely to have a much smaller impact on defect level reduction. Many factors likely contribute to the best value of τ for a given test pattern set and set of integrated circuits. For example, the difficulty of the defects present may significantly impact the appropriate value of τ. If all of the defects are easy to detect with large areas of overlap, then the value of τ may be relatively low because all of the defects are detected with the first few observations and there are simply no defects left to be detected by subsequent site observations. (In this case, the constant A would likely be high.) Furthermore, we have historically found that the best value of τ varies from circuit to circuit—implying that some circuit designs may correspond to higher values than others.

Obviously, a low value of the constant τ is not problematic if the reason for the low value is that all of the defects are detected within the first few site observations. However, this is not the only reason the value of τ may be low. Specifically, it seems logical that τ may be low when there is not enough diversity among test patterns.



Low τ,
Low excitation
balance

High τ,
High excitation
balance

Fig. 60. τ depends on excitation balance

For example, in the extreme case, if a site is always observed with an identical test pattern, then no additional defect detections are possible with all of the subsequent observations. In this case, the value of τ would approach zero. In a less extreme case, it may simply be true that many of the sites in the circuit are always in the same configuration when the site in question is observed. In that case, we would also expect τ to be relatively low.

For example, consider Fig. 60. Each of the boxes in this figure contain all possible circuit configurations (as defined by the logic values at circuit sites) that are compatible with the observation of a particular circuit site. (Note that if we consider inputs to be circuit sites, each of the circuit configurations in this box actually corresponds to a unique test pattern.) Multiple defects occur at this site in different circuits. Each of these defects is represented by one of the ovals (as shown in the box on top). Each of the ovals contain the circuit configurations/test patterns that are compatible with their detection. Unfortunately, while fortuitous detection of defects is definitely possible, the amount of overlap is not large enough to allow for all of the defects to be detected by the first few observations. Thus, τ is unlikely to be low entirely because of the nature of the defects. However, the value of τ may still be low because of the actual test pattern set we choose.

Consider the two boxes on the bottom. Each of the black circles within these boxes represents a test pattern in a test set that observes this particular circuit site. In the box on the left, many sites are skewed to particular logic values and there is not much change in the circuit site values from one pattern to another. As a result, the excitation balance is low. Furthermore, many of these site observations are essentially useless with respect to detecting additional defects at this site because they are only compatible with those defects already detected by another pattern. Thus, we expect the value of τ for this test set to be low as well. In contrast, the distribution of the test patterns in the box on the right is much more random. There is considerably less skew with respect to site probabilities, and the excitation balance is much better. Furthermore, many of these observations will detect at least one new defect. Thus, the value of τ will be higher.

As a result it makes sense to investigate more formally the relationship between τ and excitation balance. Accordingly, we performed simulations on four different benchmark circuits: c432, c499, c1908, and c1355. For each of these circuits we generated 15-detect transition test sets. Each set was generated with one of five different ATPG tools. These tools included:

- the Greedy ATPG tool
- the random OBDD-based tool
- a commercial ATPG tool

- an OBDD-based tool in which the weighted choice of path was altered
- a pattern-based dynamic compaction OBDD tool as described in [41]

We performed crosstalk surrogate simulation for each test pattern set and found the appropriate value of τ for the MPG-D model to predict the simulation results. We also calculated excitation balance numbers for every test pattern set. However, unlike the data in Fig. 59, we wanted a single number to capture the quality of excitation for each circuit and test pattern set. Thus, we averaged the excitation balance numbers for the observation patterns for the 5% least detected faults. These particular faults were chosen because it is the least detected faults that generally make the most significant contributions to the overall defective part level. As can be seen from Fig. 58, the fewer site observations that occur during testing, the more critical it is that the other circuit sites not be highly skewed when that site is observed if site value constraints are to have a high probability of being met. Furthermore, because these faults *are* hard to detect, they are most likely to have mandatory site assignments, and thus may naturally have lower excitation balance numbers—especially if the ATPG tool is not sufficiently random.

For each of these circuits, we then plotted the excitation balance number calculated for each test set against the chosen value of τ for that set. The results can be seen in Figs. 61 through 64. Although there is not perfect correlation between the value of the constant τ and the excitation balance of each test set, there is obviously a relationship, especially for the first three circuits analyzed. Furthermore, at least for these particular test sets and circuits, the relationship appears to be linear. The R^2 value varies from 0.8929 for c499 to 0.9821 for c1908.

Unfortunately, the data doesn't look nearly as good for c1355. In this case, there is one test set (the test set generated by the OBDD tool using pattern based dynamic

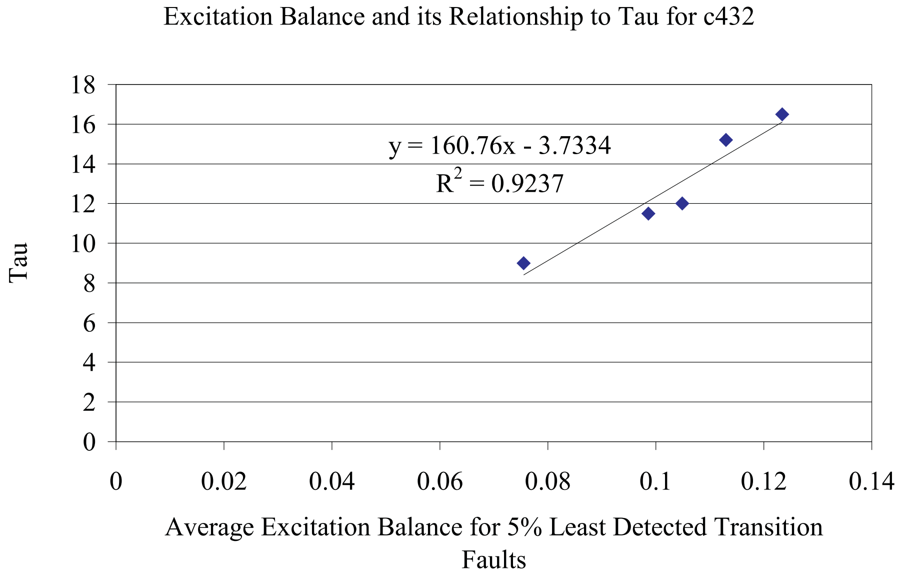


Fig. 61. Excitation balance and τ for c432

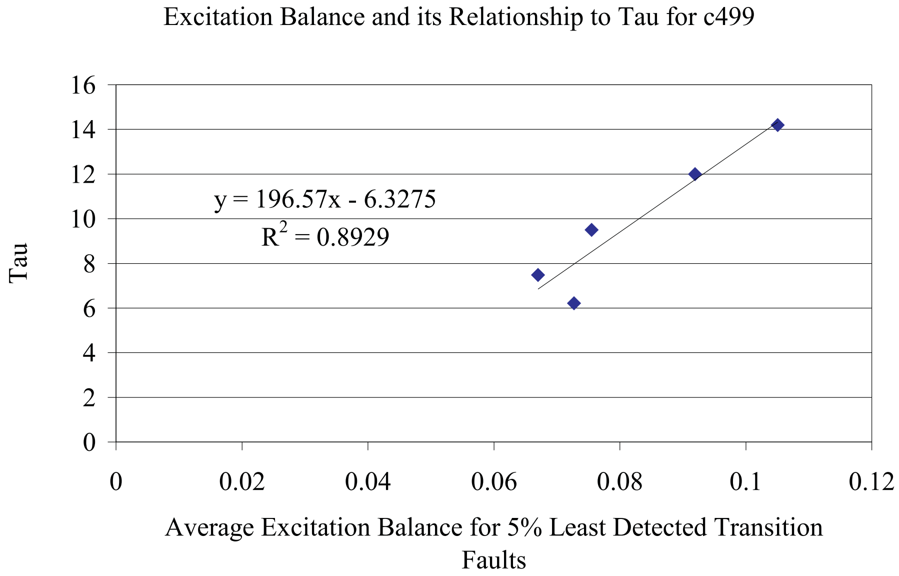


Fig. 62. Excitation balance and τ for c499

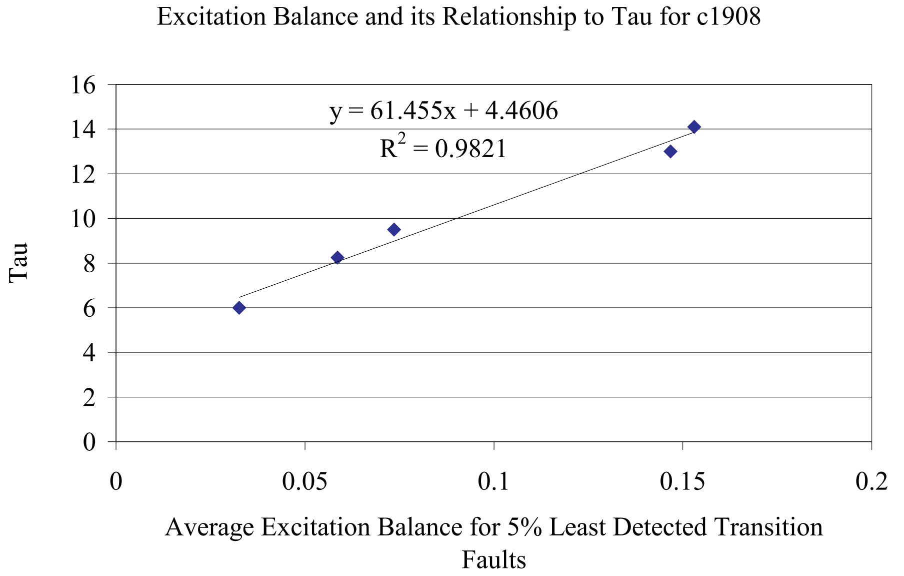


Fig. 63. Excitation balance and τ for c1908

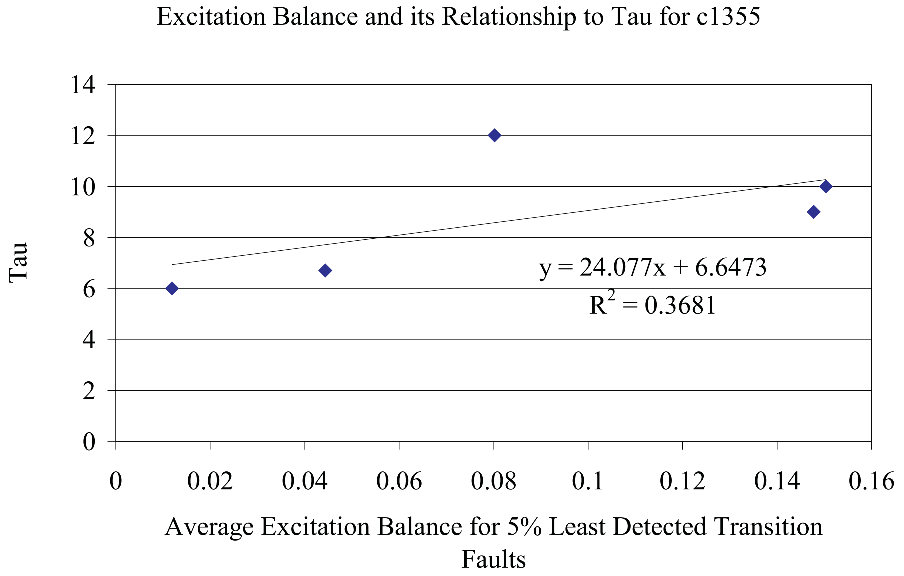


Fig. 64. Excitation balance and τ for c1355

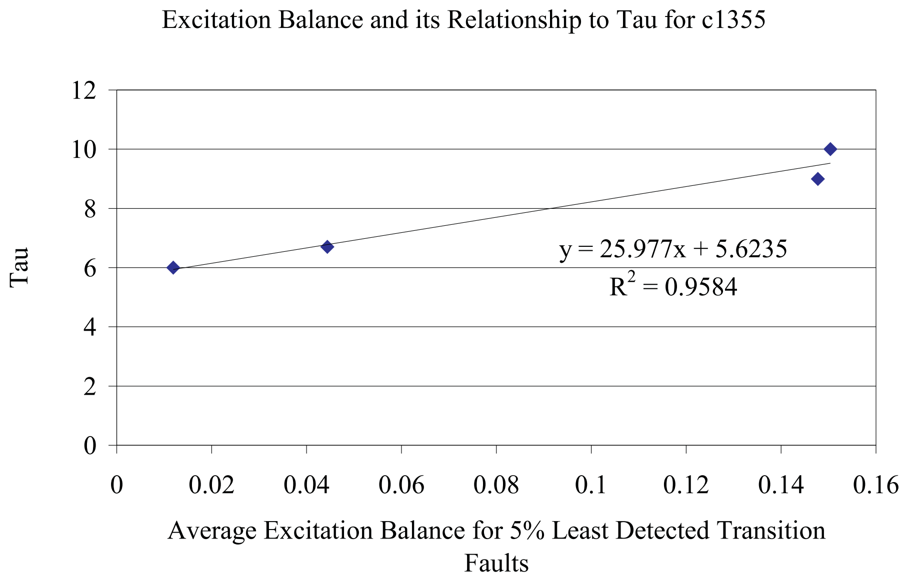


Fig. 65. Excitation balance and τ for c1355, 4 test sets

compaction) that does not match the rest of the data. The exact reason for this anomaly is unknown; however, it seems to indicate that other factors also contribute to the value of τ as stated earlier. In this particular case, the value for the constant τ is much higher than we would expect from the excitation balance numbers. However, the value for A for this test set is also lower. Thus, for some reason fewer defects may be being detected by each observation, and thus it is possible that, even with somewhat low excitation balance numbers, there are still enough defects that remain undetected in that portion of the circuit configuration test space to allow τ to remain high. However, whether this is the best explanation for this particular instance and what the underlying cause might be is still a mystery.

However, if we consider this particular test set to be an outlier, we obtain the results in Fig. 65. It is obvious that without the outlier, a definite linear relationship does exist for c1355 as well. This is very encouraging. It seems to indicate that both

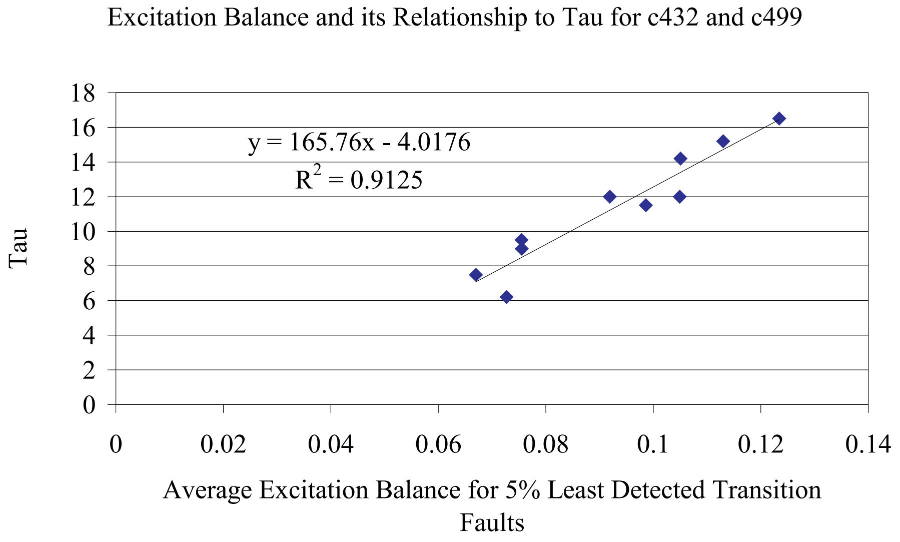


Fig. 66. Excitation balance predicts τ for two circuits

the constant τ and the excitation balance of a test pattern set have true physical meaning and help to describe the effectiveness of each additional site observation of a test pattern set. Furthermore, it seems that excitation balance is actually a very important component for determining precisely what that value of τ will be and thus ultimately influencing the final defective part level.

We also investigated plotting data for multiple circuits on the same axis. We started with plotting c432 and c499 on a single axis. The results can be found in Fig. 66 The R^2 value is very high, and the data seem to indicate that, in this case, we can use a single equation to accurately predict the appropriate value of τ from our excitation balance measurements. This is especially noteworthy because c432 and c499 are very different circuits. For example, c499 is a very "permeable" circuit. Because it contains a lot of exclusive OR gates, many of its sites are highly observable. In contrast, while some sites in c432 are also highly observable, there is

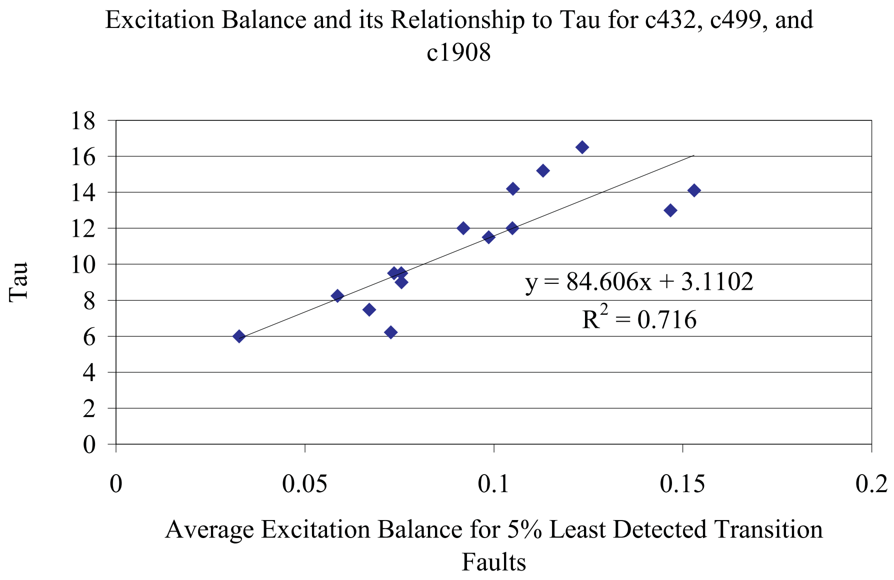


Fig. 67. Excitation balance predicts τ for three circuits

a higher percentage of sites with low observation probabilities in c432.

Unfortunately, when we added c1908, we found that the amount of correlation decreased as shown in Fig. 67. Unfortunately, the slope of the line for c1908 is not the same as for c432 and c499. One possibility is that the difference in circuit size is affecting the way excitation balance contributes to the constant τ. For example, it is possible that there are more sites in c1908 that have nothing to do with detecting a particular fault or observing a particular site. Since the more highly skewed sites are, the more harmful their lack of excitation balance may be, it is possible that a lot of sites with good balance numbers are actually diluting the effect of the more important highly skewed sites. Thus, perhaps, we want to only look at the skew of a subset of all circuit sites—specifically those that are most highly skewed.

Furthermore, if we add the data from the four test sets from c1355, we find that our correlation again becomes worse. The R^2 value goes down to 0.5062. Once again, this difference could be due to the differences in circuit size. However, further study is needed to see if it is possible to unite correlations between τ and excitation balance as a general rule across different circuits, and if so, how to accomplish that.

CHAPTER X

CONCLUSIONS

High quality test sets are necessary to detect an adequate fraction of all manufacturing defects during the test process. Historically, the prevailing strategy for generating such test sets has often been very deterministic. Research and standard industrial practice have focused on ways of modeling defects as faults and then deterministically targeting those faults while generating test patterns. The goal has been to create a short test pattern set that detects as many of the faults as possible at least once. There was a presumption that if all (or at least most) of the faults in the fault list were detected by a test set, then that test set would have also detected all (or most) of the actual defects.

Unfortunately, the detection requirements for the targeted faults and the actual defects are often different. Thus, if these unmodeled defects are detected by a test set generated by targeting mismatched faults, the detection of those unmodeled defects is *fortuitous*—not deterministic. Such detection is not guaranteed. It only occurs when test patterns generated to meet the requirements of the targeted fault or faults also happen to meet the detection requirements of the unknown defects. Thus, it is crucial to not only generate test pattern sets that do a good job of detecting modeled faults, but also to maximize the fortuitous detection of unmodeled defects.

Thus, we have approached test pattern generation from the perspective of probabilistically detecting defects instead of deterministically detecting faults. The common requirement for detecting defects of different types is often the observation of the circuit sites where those defects occur. In contrast, the excitation conditions that are needed to create those errors generally vary from defect to defect. Accordingly, our method for generating test sets to maximize fortuitous defect detection has involved deterministically maximizing the number of times difficult sites are observed with different test patterns while randomly and probabilistically meeting the excitation requirements of the unknown defects. The effectiveness of the resulting test set and the number of test escapes is estimated using the MPG-D defective part level model. Thus, this ATPG method has been named DO-RE-ME for Deterministic Observation, Random Excitation, and MPG-D defective part level Estimation.

Defective part level models such as MPG-D are very valuable tools. They allow the quality of a test set to be evaluated so that decisions can be made about whether the amount of testing being done is adequate and so that different test pattern sets can be compared. Historically, many defective part level models have relied on fault coverage to make their predictions. However, such models generally predict a defective part level of zero when 100% fault coverage is reached, and as stated earlier, this is often an overly optimistic assumption. Furthermore, we and others have performed experiments in which test sets with high fault coverages actually detected fewer real defects than test sets with slightly lower fault coverages.

Thus, in accordance with our probabilistic philosophy of testing, the MPG-D defective part level model does not focus on fault coverage. Instead, it attempts to estimate the probability of fortuitously meeting the detection requirements for unknown defects and therefore detecting them. Thus, the MPG-D estimate is based on the number of times that circuit sites are observed and the probability of exciting undetected defects at those sites with each of those observations. We have shown that this probability of excitation given site observation can be modeled as a decaying exponential function of the number of times a circuit site has been observed. This decaying exponential model is especially accurate for lower observation counts when sites have been observed few times and multiple defects potentially remain to be detected at those sites. A time constant τ determines how quickly this excitation probability decreases and for how long additional observations of a circuit site are likely to allow additional defect detection.

We have also incorporated into the MPG-D model the fact that in some cases defects may introduce errors at multiple circuit sites. The number of involved sites is encapsulated in the constant C. The value of this constant determines how much defect level contribution needs to be removed from other sites given that some site is observed and defects are detected. In the past, we have distributed this additional defective part level as evenly across all circuit sites as possible. However, we have shown that, in some cases, faster and more accurate calculations can be made with a proportional distribution of this additional reduction.

Although we initially started our analysis of fortuitous detection in the realm of relatively simple static defects, we also investigated what happens as defects became more complex and the number of circuit conditions that must be satisfied for their detection increased. Accordingly, we expanded our investigations to the realm of more complex dynamic defects. Because these defects affect the timing behavior of the circuit instead of merely its steady-state performance, a pair of test patterns (instead of a single pattern) must be applied for excitation to occur. Thus, at a minimum, at least one additional site constraint must be satisfied. For example, in the case of a transition fault, the site where the fault occurs must be set to a particular value by the preconditioning pattern so that the appropriate transition occurs when the observation pattern of the pair is applied.

These additional constraints reduce the probability of fortuitously detecting these more complex defects. In fact, when we compare the expected number of fortuitous stuck-at fault detections given the detection of a particular stuck-at fault to the expected number of fortuitous transition fault detections given the detection of the related transition fault, we find the ratio is generally approximately equal to two. The effect of this decrease in the fortuitous detection of faults as conditions are added is that longer test set lengths are needed to achieve the same number of minimum fault detections. Furthermore, we find that the fortuitous detection of unmodeled defects also decreases as their complexity and detection difficulty increase. As a result, in order to achieve the same reduction in defective part level, we will need to increase the minimum number of observations of circuit sites.

However, one of the most important effects of the increasing number of circuit conditions that must be satisfied for dynamic defect detection is the increasing importance of truly random excitation. Because we do not know the excitation requirements for unmodeled defects, it seems logical that we can maximize our chances of meeting those requirements by ensuring the circuit values are not skewed. Thus, we would like the sites in the circuit to be as likely to be equal to a logic one as a logic zero every time a given site is observed (or every time a given fault is detected). In other words, we would like our sites to be *balanced* with respect to their likely logic values for all those patterns that observe a particular site. Accordingly, we have introduced a metric called *excitation balance* to estimate the quality of the excitation by measuring the degree to which the values at all circuit sites are skewed from equal probabilities given that a particular site is observed. Furthermore, sites that are highly skewed impact the overall excitation balance calculation more than sites that are only slightly skewed.

This metric of excitation balance has been shown to explain otherwise anomalous experimental results and to correlate well with the time constant τ in the MPG-D model. Specifically, we have shown that when a 30-detect test set detected fewer dynamic surrogates than a 15-detect test set, the results could be explained by the fact that the excitation balance metric favored the 15-detect set. Thus, as defects become more complex and difficult to detect, it is likely that additional site observations will be less effective if the degree of excitation balance in inadequate. Furthermore, there appears to be a significant linear relationship between the constant τ in MPG-D and excitation balance. Although other factors may also influence the optimal value of τ for a given circuit and test pattern set, excitation balance seems to be one of the most important.

Although defective part level modeling is valuable from the perspective of estimating final defective part levels, when its components incorporate physical realities, it can also be used to guide the generation of better test pattern sets. For example, the MPG-D model is based upon the two requirements for defect detection: site observation and defect excitation. With this model, we can estimate how additional site observations or more balanced defect detection could affect the final defective part level. Then we can tailor our ATPG algorithms to improve whichever component is lacking. For example, when excitation balance is poor, the test pattern set is often biased toward certain circuit configurations and unlikely to detect those defects that require a different configuration. In addition, when choosing a subset from a superset, we can make sophisticated tradeoffs among different numbers of site observations and their impact on the overall defective part level instead of merely focusing on the minimum number of fault detections.

Unfortunately, we have found that the lack of significant defect detection with subsequent test patterns does not always indicate that most or all of the defects have been detected. However, by considering the behavior of the different components of the MPG-D model, we can optimize the effectiveness of each of those components. This should improve the quality of individual test patterns (or test pattern pairs) and increase the fortuitous detection of static and dynamic defects by the entire test set. Ultimately, this should reduce the likelihood that a significant number of defective parts will ever reach the consumer.

REFERENCES

[1] R. D. Eldred, "Test routines based on symbolic logical statements," *Journal of the ACM*, vol. 6, no. 1, pp. 33–36, 1959.

[2] S. D. Millman and E. J. McClusky, "Detecting bridging faults with stuck-at test sets," in *Proc. International Test Conference*, 1988, pp. 773–783.

[3] P. C. Maxwell, R. C. Aitken, V. Johansen, and I. Chiang, "The effectiveness of I_{DDQ}, functional and scan tests: how many fault coverages do we need?," in *Proc. International Test Conference*, 1992, pp. 168–177.

[4] S. C. Ma, P. Franco, and E. J. McClusky, "An experimental chip to evaluate test techniques: experiment results," in *Proc. International Test Conference*, 1995, pp. 663–672.

[5] P. Franco, W. D. Farwell, R. L. Stokes, and E. J. McClusky, "An experimental chip to evaluate test techniques: chip and experimental design," in *Proc. International Test Conference*, 1995, pp. 653–662.

[6] K.M. Butler and M. Ray Mercer, "Quantifying non-target defect detection by target fault test sets," in *Proc. European Test Conference*, Munich, Germany, 1991, pp. 91–100.

[7] A. Pancholy, J. Rajski, and L. J. McNaughton, "Empirical failure analysis and validation of fault models in CMOS VLSI circuits," *IEEE Design and Test of Computers*, vol. 9, no. 1, pp. 72–83, March 1992.

[8] Li-C. Wang, M. R. Mercer, and T. W. Williams, "On efficiently and reliably achieving low defective part levels," in *Proc. International Test Conference*, 1995, pp. 616–625.

[9] Li-C. Wang, M. R. Mercer, T. W. Williams, and S. W. Kao, “On the decline of testing efficiency as fault coverage approaches 100%,” in *Proc. VLSI Test Symposium*, 1995, pp. 74–83.

[10] Li-C. Wang, M. R. Mercer, and T. Williams, “A better ATPG algorithm and its design principles,” in *Proc. International Conference on Computer Design*, 1996, pp. 248–253.

[11] K. C. Mei, “Bridging and stuck-at faults,” *IEEE Transactions on Computers*, vol. C-23, no. 7, pp. 720–727, July 1974.

[12] J. Saxena, K.M. Butler, H. Balachandran, D.B. Lavo, B. Chess, T. Larrabee, and F.J. Ferguson, “On applying non-classical defect models to automated diagnosis,” in *Proc. International Test Conference*, 1998, pp. 748–757.

[13] I. Shaik S. Ma and R. S. Fetherston, “A comparison of bridging fault simulation methods,” in *Proc. International Test Conference*, 1999, pp. 587–595.

[14] J. Rearick and P. Maxwell, “Deception by design: Fooling ourselves with gate-level models,” in *Proc. International Test Conference*, 2000, pp. 921–929.

[15] F. J. Ferguson and T. Larrabee, “Test pattern generation for realistic bridge faults in CMOS ICs,” in *Proc. International Test Conference*, 1991, pp. 492–499.

[16] R. Kapur, J. Park, and M. R. Mercer, “All tests for a fault are not equally valuable for defect detection,” in *Proc. International Test Conference*, 1992, pp. 762–769.

[17] Li-C. Wang, M. R. Mercer, and T. W. Williams, “Using target faults to detect non-target defects,” in *Proc. International Test Conference*, 1996, pp. 629–638.

[18] M. R. Grimaila, S. Lee, J. Dworak, K. M. Butler, B. Stewart, H. Balachandran, B. Houchins, V. Mathur, J. Park, Li-C Wang, and M. R. Mercer, "REDO—random excitation and deterministic observation—first commercial experiment," in *Proc. VLSI Test Symposium*, 1999, pp. 268–274.

[19] S.M. Reddy, I. Pomeranz, and S. Kajihara, "On the effects of test compaction on defect coverage," in *Proc. VLSI Test Symposium*, 1996, pp. 430–435.

[20] T. W. Williams and N. C. Brown, "Defect level as a function of fault coverage," *IEEE Transactions on Computers*, vol. C-30, no. 12, pp. 987–988, December 1981.

[21] D. V. Das, S. C. Seth, P. T. Wagner, J. C. Anderson, and V. D. Agrawal, "An experimental study on reject ratio prediction for VLSI circuits: Kokomo revisited," in *Proc. International Test Conference*, 1990, pp. 712–720.

[22] V. D. Agrawal, S. C. Seth, and P. Agrawal, "Fault coverage requirements in production testing of LSI circuits," *IEEE Journal of Solid State Circuits*, vol. SC-17, no. 1, pp. 57–61, February 1982.

[23] S. C. Seth and V. D. Agrawal, "Characterizing the LSI yield equation from wafer test data," *IEEE Trans. Computer-Aided Design of Int. Circuits and Systems*, vol. CAD-3, pp. 123–126, April 1984.

[24] R. L. Radsack, "Fault coverage in digital integrated circuits," *Bell System Technical Journal*, vol. 57, pp. 1475–1488, May-June 1978.

[25] J. Park, M. Naivar, R. Kapur, M. R. Mercer, and T. W. Williams, "Limitations in predicting defect level based on stuck-at fault coverage," in *Proc. VLSI Test Symposium*, 1994, pp. 186–191.

[26] P. C. Maxwell, R. C. Aitken, V. Johansen, and I. Chiang, “The effect of different test sets on quality level prediction: when is 80% better than 90%?,” in *Proc. International Test Conference*, 1991, pp. 358–364.

[27] S. C. Seth and V. D. Agrawal, “On the probability of fault occurrence,” in *Defect and Fault Tolerance in VLSI Systems*, 1989, pp. 47–52.

[28] D. Das, Sharad C. Seth, and V. D. Agrawal, “Accurate computation of field reject ratio based on fault latency,” *IEEE Trans. on VLSI Systems*, vol. 1, no. 4, pp. 537–545, December 1993.

[29] J. T. Sousa and V. D. Agrawal, “Reducing the complexity of defect level modeling using clustering effect,” in *Proc. Design, Automation, and Test in Europe Conference and Exhibition, DATE 2000*, 2000, pp. 640–644.

[30] Li-C. Wang, A. Krstic, L. Lee, K.-T. Cheng, M. R. Mercer, T. W. Williams, and M. S. Abadir, “Using logic models to predict the detection behavior of statistical timing defects,” in *Proc. International Test Conference*, 2003, pp. 1041–1050.

[31] B. Benware, C. Schuermyer, S. Ranganathan, R. Madge, P. Krishnamurthy, N. Tamarapalli, K.-H. Tsai, and J. Rajski, “Impact of multiple-detect test patterns on product quality,” in *Proc. International Test Conference*, 2003, pp. 1031–1040.

[32] J. Dworak, M. R. Grimaila, S. Lee, Li-C. Wang, and M. R. Mercer, “Modeling the probability of defect excitation for a commercial IC with implications for stuck-at fault-based ATPG strategies,” in *Proc. International Test Conference*, 1999, pp. 1031–1037.

[33] J. Dworak, M. R. Grimaila, S. Lee, Li-C. Wang, and M. R. Mercer, “Enhanced

DO-RE-ME based defect level prediction using defect site aggregation—MPG-D," in *Proc. International Test Conference*, 2000, pp. 930–939.

[34] Jennifer L. Dworak, "Modeling the probability of excitation and the defective part level as testing progresses," M.S. thesis, Texas A&M University, College Station, Texas, May 2000.

[35] F. Brglez and H. Fujiwara, "A neutral netlist of 10 combinational benchmark circuits and a target translator in FORTRAN," in *Proc. IEEE Symp. on Circuits and Systems; Special Session on ATPG and Fault Simulation*, June 1985, pp. 695–698.

[36] H. K. Lee and D. S. Ha, "On the generation of test patterns for combinational circuits," Technical Report No. 12, Department of Electical Engineering, Virginia Polytechnic Institute and State University, Blacksburg, 1993.

[37] J. Dworak, J. D. Wicker, S. Lee, M. R. Grimaila, K. M. Butler, B. Stewart, Li-C. Wang, and M. R. Mercer, "Defect oriented testing and defective-part-level prediction," *IEEE Design and Test of Computers*, vol. 18, no. 1, pp. 31–41, January–February 2001.

[38] R. E. Bryant, "Graph-based algorithms for boolean function manipulation," *IEEE Transactions on Computers*, vol. C-35, no. 8, pp. 677–691, August 1986.

[39] James Wingfield, "Approaches to test set generation using binary decision diagrams," M.S. thesis, Texas A&M University, College Station, Texas, December 2003.

[40] J. Dworak, J. Wingfield, B. Cobb, S. Lee, Li.-C. Wang, and M. R. Mercer, "Fortuitous detection and its impact on test set sizes using stuck-at and transition

faults," in *Proc. of the 17th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, Vancouver, Canada, 2002, pp. 177–185.

[41] J. Wingfield, J. Dworak, and M. R. Mercer, "Function-based dynamic compaction and its impact on test set sizes," in *Proc. of the 18th International Symposium on Defect and Fault Tolerance in VLSI Systems*, Cambridge, Massachusetts, 2003, pp. 167–174.

[42] S. Lee, B. Cobb, J. Dworak, M. R. Grimaila, and M. R. Mercer, "A new ATPG algorithm to limit test set size and achieve multiple detections of all faults," in *Proc. Design, Automation, and Test in Europe Conference and Exhibition, DATE 2002*, Paris, France, 2002, pp. 94–99.

[43] W. Moore, G. Gronthoud, K. Baker, and M. Lousberg, "Delay-fault testing and defects in deep sub-micron ICs—does critical resistance really mean anything?," in *Proc. International Test Conference*, Atlantic City, New Jersey, 2000, pp. 95–104.

[44] E. J. McCluskey and C-W. Tseng, "Stuck fault tests vs. actual defects," in *Proc. International Test Conference*, Atlantic City, New Jersey, 2000, pp. 336–342.

[45] I. Pomeranz and S.M. Reddy, "Stuck-at tuple detection: a fault model based on stuck-at faults for improved defect coverage," in *Proc. VLSI Test Symposium*, Monterey, California, 1998, pp. 289–294.

[46] I. Pomeranz and S. Reddy, "Definitions of the numbers of detections of target faults and their effectiveness in guiding test generation for high defect coverage," in *Proc. Design, Automation, and Test in Europe Conference and Exhibition, DATE 2001*, Munich, Germany, 2001, pp. 504–508.

[47] R. D. Blanton, K. N. Dwarakanath, and A. B. Shah, "Analyzing the effectiveness of multiple-detect sets," in *Proc. International Test Conference*, Charlotte, North Carolina, 2003, pp. 876–885.

VITA

Contact information:

Jennifer Lynn Dworak
4622 Colony Hills Drive
Sugar Land, Texas 77479
Phone: (281) 265-1856

Education:

Ph.D., Electrical Engineering, Texas A&M University, May 2004
M.S., Electrical Engineering, Texas A&M University, May 2000
B.S., Electrical Engineering, Texas A&M University, December 1998

Research Interests:

Digital Circuit Testing and ATPG, Defect Level Modeling, Field Failure Rates due to Manufacturing Defects, & Multilevel Logic Minimization

Professional Experience:

Summer Engineering Intern at Schlumberger Anadrill, 1995-1998

Teaching Experience:

Graduate Assistant Lecturer for ELEN 248: *Introduction to Digital Systems Design,* Spring 2001, Fall 2001, Spring 2002

Honors and Awards:

National Science Foundation Graduate Research Fellowship, 2000 - 2003
Best Paper Award, VLSI Test Symposium, Dana Point, CA, 1999
Best Student Presentation Award, International Test Synthesis Workshop, March 2002
Ethel Ashworth-Tsutsui Memorial Award for Research, 2002 (generally one or two female Texas A&M graduate students chosen per year)
The Association of Former Students & the Office of Graduate Studies Distinguished Graduate Student Award, Texas A&M University, 2000
Best Presentation in Section, University Undergraduate Fellow, 1997-1998

The typist for this thesis was Jennifer Lynn Dworak.