# Probabilistic Mixed-Model Fault Diagnosis

David B. Lavo* Brian Chess* Tracy Larrabee† Ismed Hartanto‡

†Computer Engineering
University of California
Santa Cruz, CA 95064

‡Hewlett-Packard Company
Integrated Circuit Business Div.
Palo Alto, CA 94306

## Abstract

Previously-proposed strategies for VLSI fault diagnosis have suffered from a variety of self-imposed limitations. Some techniques are limited to a specific fault model, and many will fail in the face of any unmodeled behavior or unexpected data. Others apply ad-hoc or arbitrary scoring mechanisms to fault candidates, making the results difficult to interpret or to compare with the results from other algorithms. This paper outlines an approach to fault diagnosis that is robust, comprehensive, extendable, and practical. By introducing a probabilistic framework for diagnostic prediction, it is designed to incorporate disparate diagnostic algorithms, different sets of data, and a mixture of fault models into a single diagnostic result. Results from diagnosis experiments on a Hewlett-Packard ASIC and FIB'd defects are presented.

## 1 Introduction

A useful if somewhat strained analogy to the process of failure analysis is its similarity to criminal detective work: given the evidence of circuit failure, determine the cause of the failure, identifying a node or region that is the source of error. In addition to location, it is useful to identify the mechanism of failure, such as an unintentional short or open, so that remediating changes can be considered in the design or manufacturing process.

A common perception (not entirely unfounded) of failure analysis is that of a lab of hard-boiled engineers physically and aggressively interrogating the failing part, using scanning electron microscopes, particle beams, infrared sensors, liquid crystal films, and a variety of other high-tech and high-cost techniques to eventually force a confession out of the silicon scofflaw. The final result, if successful, is the identification of the actual cause of failure for the circuit, along with the requisite gory "crime scene" photograph of the defective region itself: an errant particle, missing or extra conductor, a disconnected via, and so on.

The sweaty, smoke-filled scene of the failure analysis lab is only part of the story, however, and is usually referred to as *defect identification* or *physical failure analysis*. Given the enormous number of circuit devices in modern ICs, and the number of layers in most complex circuits, physical interrogation cannot hope to succeed without first having a reasonable list of suspect locations. It is the job of the other part of failure analysis, usually called *fault diagnosis*, to do the logical detective work. Based on the data available about the failing part, the purpose of fault diagnosis is to produce an evaluation of the failing chip and a list of likely defect sites or regions. A lot is riding on this initial footwork: if the diagnosis is either inaccurate or imprecise (identifying either incorrect or excessively many fault candidates, respectively), the process of physical fault location will consume, and possibly waste, considerable amounts of time and effort.

This paper presents a diagnostic approach that acknowledges the following principles. First, that the type of defect, and the most appropriate fault model for the defect, is not known before diagnosis begins. Second, that unmodeled behavior is an unavoidable part of fault diagnosis, as is the presence of noise and uncertainty in the data. Third, that the process of diagnosis should be both practical and inclusive, with reasonable resource requirements but using every available source of information to improve the final diagnosis.

The fundamental aspects of fault diagnosis will be discussed in Section 2, including fault models, fault signatures, and diagnostic algorithms. Section 3 reviews our previously-published technique for bridging fault diagnosis. Section 4 introduces a new mixed-model probabilistic diagnosis approach, expanding on the bridging-fault technique to include multiple fault models and other sources of information. Section 5 presents experimental details and results for the technique on an industrial circuit, and Section 6 presents our conclusions about the system as implemented.

---

[*] When this work was performed, David Lavo was with the Department of Computer Engineering at UCSC. Brian Chess is a part time CE student at UCSC. David and Brian are full-time employees of Hewlett-Packard Company's Integrated Circuit Business Division.

# 2 The nature of the problem

Here is the problem of fault diagnosis in a nutshell: a circuit has failed one or more tests applied to it; from this failing information, determine what has gone wrong. The evidence usually consists of a description of the tests applied, and the pass-fail results of those tests. In addition, more detailed per-test failing information may be provided. There is a catch, however: any or all of this data might be unreliable, misleading, or downright corrupt.

Fault diagnosis would be a relatively simple problem of pattern matching (*cause-effect diagnosis*) or path tracing and intersection (*effect-cause diagnosis*) if it weren't for some messy complications. As with criminal detective work, the real world can confound even the best theoretical investigations.

## 2.1 The model crime?

Generally speaking, fault models have proved their utility for test generation. If, for example, a test is generated to detect the (abstract) situation of a circuit node stuck-at 0, there is considerable evidence to suggest that the test will, in the process, detect a wide range of related defects: the node shorted to ground, perhaps, or missing conductor to a pull-up network, or even a floating node held low from a capacitive effect. When testing a circuit for defects, the actual relation of fault model to defect is less important than whether the defect is caught or not.

But what does it mean, in the world of diagnosis, to explain the actual failures of a circuit with an abstract fault model? Try as one might, no failure analysis engineer is ever going to find a single stuck-at fault under the microscope; a stuck-at fault, strictly defined, is not a specific explanation, but is instead a useful fiction.

For diagnosis, the issue is one of resolution: the more abstract the model used, the less well the fault candidates in the final diagnosis will map to actual defects in the silicon. A stuck-at candidate, for example, may implicate a range of mechanisms or defect scenarios involving the specified stuck-at node, and the failure analysis engineer must account for this poor resolution by performing some amount of mapping to actual circuit elements. The more specific the fault model, the better the correspondence to actual defects, and the less mapping work is required: a sophisticated bridging fault candidate, with specific electrical characteristics, will usually resolve to either a single or a few defect scenarios. For this reason, other diagnosis techniques employ different and less abstract fault models; but as is often the case, these solutions simply lead to other, more troubling, problems.

## 2.2 Who are the suspects?

Several theoretical systems have claimed great success in some variation of the following experiment: create or simulate a defect of a certain fault type, say bridging, create some bridging candidates, and run the diagnosis algorithm to choose the correct candidate out of the list. While the accuracy of these success stories is indeed laudable, the result is a little like pulling the correct culprit out of a police lineup: the job is made much easier if the choices are limited ahead of time.

It is an unfortunate fact of life, however, that what form a defect has taken, or what fault model could best represent the actual electrical phenomenon, is not known in advance. Given this fact, there are several possible ways to proceed.

Perhaps the abstractness of the stuck-at fault model can be exploited: use a stuck-at diagnosis to hopefully get close enough, regardless of defect idiosyncracies, to allow physical verification after some amount of mapping work. But this approach has some well-characterized problems in both accuracy and precision. For example, even a robust stuck-at diagnosis may identify one of two shorted nodes only 60% to 90% of the time [3]. For situations in which a 10% to 40% failure rate is unacceptable, or such partial answers (single-node explanations) are inadequate, stuck-at diagnosis alone is not the answer.

But it is known that better fault models, ones theoretically more accurate and precise, can be constructed, so a second answer might be to use as many models and algorithms as possible. Perhaps a stuck-at diagnosis, a bridging diagnosis, and a delay fault diagnosis or two could be performed, and the results from this mix of algorithms examined. But apart from the time and work required, some troubling issues remain.

First, could all the faults really be covered or considered? For example, would a separate set of bridging fault candidates and their signatures have to be constructed for each of a variety of bridge resistance values? And even if the range of behaviors of interest could be limited, a problem remains in reconciling the different results: how can one compare the top candidates from, for example, a stuck-at fault diagnosis algorithm to the top bridging candidates from a completely different algorithm? Many diagnosis techniques employ unique scoring mechanisms to rate their candidates, and even when common techniques are used, such as Hamming distance, they are often applied in different ways or to different data: a "1-bit difference" may mean something very different for a stuck-at candidate than for an $I_{DDQ}$ candidate. The challenge remains, then: either find a way to simultaneously employ multiple fault models, or pick one fault model and hope for the best.

### 2.3 Fault models will lie under pressure.

Many clever diagnosis algorithms have been proposed, using a variety of fault models, and all promise great success as long as one condition holds: nothing unexpected ever happens. These expectations come from the fault model used, the diagnostic algorithm, or both. So, if the modeled defect doesn't cause a circuit failure when expected, or if a failure occurs along an unanticipated path, the algorithm will either quit or get hopelessly off the track of the correct suspect.

If the problem is defective fault models, then maybe the solution is to work very hard to perfect the models. If the models were perfect, then diagnosis would reduce to a simple process of finding exactly the matching candidate for the observed behavior. But, once again, the cold hard world intrudes with the cold hard facts: fault model perfection is extremely difficult, and may very likely be impossible.

Perhaps best documented are the problems inherent in bridging fault modeling: many simplified bridging fault models have been proposed, and each in turn has been demonstrated to be inadequate or inaccurate in one or more important respects [1, 8, 4, 10]. Even the most complex and computationally intensive models can fail to account for the subtleties of defect characteristics and the vagaries of defective circuit behavior. And it is not only the complex models that are prone to error: even apparently simple predictions may be hard to make when real defects are involved.

The unfortunate fact is that faulty circuits have the tendency to misbehave—they are faulty, after all—and often fail in ways not predicted by the best of fault simulators or the most carefully crafted fault models. The only answer is that any diagnostic technique that hopes to be effective on real-world defective circuits has to be robust enough to tolerate at least some level of noise and uncertainty. If not, the only certain thing about the process will be the resulting frustration of a sadly misguided engineer.

## 3 Inexpensive fault diagnosis

This section gives a brief overview of our previous research on the subject of fault diagnosis. It covers the innovations developed for bridging fault diagnosis in particular, and introduces some important concepts for the proposed multiple-fault-model diagnostic framework.

Fault diagnosis using the stuck-at model has dominated in most industrial settings because the stuck-at fault model is ubiquitious in testing-related tools. Therefore, a good stuck-at fault simulator is usually available and in wide use, along with other nice details such as a fault-list, a stuck-at test set, and logic fail information from the tester. But the desire to overcome the limitations of the stuck-at model for diagnosis has motivated a great deal of research into better fault models, better algorithms, and different approaches to the problem of fault diagnosis.

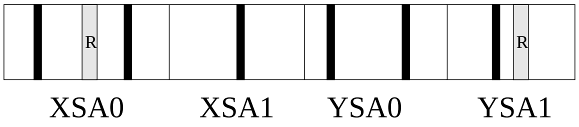

Figure 1: The composite signature of X bridged to Y with match restrictions (in black) and match requirements (labeled R)
.

The prominence of the stuck-at fault model, and the prevalence of bridging defects in CMOS, has motivated several attempts at using the stuck-at fault model to perform bridging fault diagnosis. We have previously published an improvement [6] to one such technique, by Millman, McCluskey, and Acken [9]; the improved technique demonstrated considerable success at diagnosing simulated bridging faults. Like the original technique, our approach used only stuck-at fault simulation and signatures, but improved on the original technique in three ways: considering only realistic bridges, incorporating *match restriction* (flagging some test vectors as incapable of detecting a particular bridging fault), and incorporating *match requirement* (flagging some vectors as dependably detecting a particular bridging fault).

The basic idea to both the original and improved technique is that of the *composite bridging fault signature*, which is the union of the four single stuck-at signatures associated with the two bridged nodes. The underlying idea of the composite bridging signature is this: if a bridging fault is detected by a test, that test will also detect one or more of the four stuck-at faults on the bridged nodes. Therefore, it is expected that the actual bridging fault signature (the set of detecting test vectors if the bridge occurs) will be a subset of the vectors found in the bridge's composite signature.

Figure 1 illustrates the composite signature of a fault candidate for node **X** bridged with node **Y**; for simplicity, the contents of each of the four component sets can be considered to be the test vectors that detect each respective fault. The figure, then, portrays the set concatenation of four stuck-at signatures. The black lines in the figure illustrate the concept of match restriction: if the same test vector (in the figure, the same vector number occupies the same relative position in each set) detects both **X** stuck-at 0 and **Y** stuck-at 0, it by definition tries to set each line to 1. When both bridged nodes are set to the same value it is highly unlikely that the bridging fault will be stimulated (no error should result), and the test vector can be marked as *restricted* or removed from the composite signature. The same holds true for any test that detects both **X** stuck-at 1 and **Y** stuck-at 1.

The lines labeled R in Figure 1 illustrate the complementary concept to match restriction, called match requirement: if the same test vector detects both **X** stuck-at 0 and **Y** stuck-at 1 (or vice-versa), that test should detect the bridging fault (since it sensitizes and propagates both simple fault conditions), and it is flagged in the composite signature as a *required* vector.

The result is a signature for the bridging fault node **X** bridged to node **Y**, but notice that only stuck-at fault signatures (and simulation) were used—no bridging fault modeling or simulation was required. This is a tremendous practical advantage, as it allows inexpensive but approximate bridging fault signatures to be created much more cheaply than with almost any bridging simulator, using tools (a stuck-at simulator or a set of stuck-at fault signatures) that are usually readily available.

As the signatures are now only approximations to actual bridging fault behaviors, the matching algorithm that selects candidates for the final diagnosis must allow and expect some mismatch between the predictions (composite signatures) and the observed behavior (actual failing test vectors). The original technique only expected that the correct candidate's composite signature would be a superset of the observed behavior. However, the elimination of the restricted vectors, and the specification of required vectors, improves the predictions, and provides the matching algorithm a means for refining its expectations and judging the goodness of each candidate compared to the observed behavior.

Our previous scoring system was lexicographic, in which each candidate was ranked on three criteria, in descending order of importance. First, as in the original technique, the observed behavior for a bridging fault is expected to be a subset of the candidate signature, so any *nonprediction* (errors seen but not predicted) is very unexpected. Second, some test vectors in each candidate are marked as required, so we can judge a candidate by how many of its required vectors actually detected the fault. Third, while some *misprediction* (errors predicted but not seen) is to be expected with composite signatures, excessive misprediction indicates a poor match with the observed behavior. The final scoring, as stated, was lexicographic, with the (smallest) amount of nonprediction having priority, followed by the number of successful required vector predictions, and finally by the (smallest) amount of misprediction.

We then ran experiments to see how well this technique could perform at diagnosing simulated bridging defects, especially in the presence of noise [7]. Various amounts of random noise were added to the simulated bridging signatures, and the technique attempted to identify the correct bridging fault in a list of 10 candidates. The results were quite successful: even in the presence of severe noise (causing the deletion of more half of the original information or the addition of half again as much spurious information or both), the scoring mechanism was able to successfully extract the correct candidate from 70% to 95% of the time.

Despite the good results, the accusation can be made that once again this is simply shooting ducks in a barrel, since it is known ahead of time that the defect is a bridge, and then bridging candidates are used to identify it. What about a more realistic scenario, where the form of the defect is unknown? Can the procedure account for another fault type, and incorporate and distinguish between varying explanations for the observed faulty behavior?

These are exactly the questions we needed to answer when we set out to transfer this technology to industrial use, performing real-world diagnoses on actual failing circuits. The next two sections of this paper detail our current work towards answering these questions, and report the experimental results obtained.

## 4 Mixed-model scoring

Given the complex nature of fault diagnosis discussed in Section 2, the most robust cause-effect diagnosis system would have the ability to include an arbitrary number of fault models, would employ all the models towards diagnosing the faulty circuit, and would report a single answer that represents the best explanation for the behavior over all candidates. Such a system would admittedly require more work as more models were added, but it could theoretically cover an arbitrary range of fault types and behaviors. Such a system is perhaps the ideal, with a model for every contingency, but in practice the number of models will probably be limited to those considered most likely or most interesting. For this research, our approach was to build towards a robust diagnosis system by starting small, with a combination stuck-at fault/bridging fault diagnostic system.

The idea behind such a two-model system is relatively modest. First, we use bridging fault candidates and (composite) signatures to diagnose actual bridging defects. Second, we use stuck-at candidates and signatures to diagnose a selected set of other defects: shorts to power or ground and "charged" opens—disconnected circuit lines that hold a high or low logic value. These defect types were chosen because they are assumed to be both commonplace and well represented by the stuck-at fault model. The diagnostic bottom line is: if the behavior looks most like a bridging candidate, score the bridge highest; if it looks most like a stuck-at candidate, score the stuck-at candidate highest; if neither happens, give some indication that the behavior is not much like any of the candidates, bridging or stuck-at.

It should be obvious that, in order for this mixed-model system to work, an improved method of scoring fault candidates is required that can be applied across fault models. This is not possible with the previously-described composite bridging fault scoring, as there is direct reference to such bridging-specific items as required and restricted vectors. Some generalization of the concept of candidate scoring needs to be defined that will work for any fault candidate, regardless of fault model.

## 4.1 Scoring: Bayes decision theory

Perhaps the most intuitive method of scoring and comparing fault candidates is numeric, and specifically probabilistic. In other words, what a diagnosis should really calculate is the probability that the failures seen are due to one fault candidate or another, whether that candidate is a stuck-at fault or some other fault type. It would follow, then, that the candidate with the highest probability of having occurred is the most likely suspect.

Applying probabilistic measures to the problem of diagnosis has been recently proposed by a number of researchers. Sheppard and Simpson have developed a comprehensive approach to system-level diagnosis that they recently proposed for application to traditional fault dictionaries [11]. Thibeault [12] has developed an approach to $\mathrm{I}_{\mathrm{DDQ}}$ diagnosis that uses a form of current signatures and maximum likelihood estimation, comparing measured current levels to predictions of differential current under a given noise model. And, a method for probabilistically conducting physical failure analysis has been developed by Henderson and Soden at Sandia National Labs [5].

The probability of a fault candidate occurrence given an observed faulty behavior can be expressed literally as $p(\mathbf{c}|\mathbf{b})$, where the candididate and behavior are represented by their fault signatures $\mathbf{c}$ and $\mathbf{b}$ respectively. An obvious choice for the best candidate is the one with the maximum posterior probability of all candidates considered:

$$p(\mathbf{c}_i|\mathbf{b}) > p(\mathbf{c}_j|\mathbf{b}) \text{ for all } j \neq i$$

This is merely the simplest expression of *Bayes decision theory*, used extensively in the fields of pattern recognition and classification. The theory states that the best explanation (or classification) for a phenomenon is the explanation judged to be most likely given the phenomenon.

This is obvious, intuitive, and simple, so of course there's a catch: the probability measure $p(\mathbf{c}_i|\mathbf{b})$ is difficult to calculate directly. Fortunately, *Bayes rule* comes to the rescue:

$$p(\mathbf{c}_i|\mathbf{b}) = \frac{p(\mathbf{c}_i)p(\mathbf{b}|\mathbf{c}_i)}{\sum_{\text{all i}} p(\mathbf{c}_i)p(\mathbf{b}|\mathbf{c}_i)}$$

The value $p(\mathbf{c}_i)$ is the a-priori probability of each fault candidate: that is, the probability of a fault's occurrence over all candidates regardless of fault model. The conditional probability $p(\mathbf{b}|\mathbf{c}_i)$ is the probability that the behavior seen is the result of the candidate fault occurring. While this expression may not seem like much of an improvement, the difference now is that, unlike the probability $p(\mathbf{c}_i|\mathbf{b})$, both $p(\mathbf{c}_i)$ and $p(\mathbf{b}|\mathbf{c}_i)$ can be calculated or approximated for each candidate, as will be explained shortly.

Since the denominator in the above equation is the same for all fault candidates, calculating and comparing the numerator for each fault candidate gives a numerical ordering across all candidates, regardless of model.[1] Using the probability $p(\mathbf{c}_i)p(\mathbf{b}|\mathbf{c}_i)$ as a scoring function is the classic *Bayes decision rule*, and under some basic assumptions can be proven to give the minimum error rate of any scoring or decision method.

The a-priori probabilities $p(\mathbf{c}_i)$ can be calculated through various means. One method is inductive fault analysis, which examines the physical layout of the fabricated circuit and calculates probabilities that various defects will occur. Alternatively, defect sample statistics can be used, or other estimates based on specifics of the actual circuit. In the absence of such information, the a-priori probabilities can be approximated as equal for all candidates, implying that all faults, regardless of model, are equally likely. This is a gross approximation and will obviously affect the accuracy of the results, but it does allow a diagnosis to proceed if a good estimate of the a-priori probabilities is not available.

The conditional probability expresses the probability of the observed behavior resulting from a particular candidate fault. In other words, it is the probability that the circuit behaves in a certain way if the fault occurs. Traditional classification of physical phenomena would usually involve sampling the various candidates and describing the frequencies of their behaviors statistically. This is obviously not possible with model-based fault diagnosis; sufficient samples are simply not available for every candidate of every fault model. Instead, we will have to rely on probabilistic modeling: the conditional probability functions will be estimates based upon the information available and the inherent assumptions in each of the candidate fault models. In other words, candidate fault signatures will be treated as predictions of actual defect behavior, and the conditional probabilities will be functions of the estimated rates of prediction error.

The question is, what are the levels of confidence associated with each type of fault model used? The answer depends upon the accuracy of the models and predictions, the correlation of each model to the defects it targets, and, perhaps most importantly, the judgement of the failure

---

[1] One of the assumptions of Bayes rule is that the candidates are an exhaustive and mutually exclusive set of causes for the observed phenomenon. This will generally not be true for fault diagnosis, as unmodeled behavior may occur. Therefore, while the numerator alone still provides an ordering over the fault candidates considered, the denominator does not satisfy the rule of total probability and the complete ratio will likely be an overestimation of the posterior probability for any fault candidate.

analysis engineer.

## 4.2 The probability of error ...

The conditional probability $p(\mathbf{b}|\mathbf{c}_i)$ of a stuck-at candidate should be relatively straightforward to calculate: it is a function of the expected error rate of the stuck-at simulator that produced that candidate's signature. In other words, the likelihood that a prediction for a stuck-at fault differs from the observed behavior when that fault is realized should depend upon the accuracy of the fault simulator, and to a lesser extent upon other factors such as the reliability of the measurements and the integrity of the data.

This sounds relatively simple, but of course there are details and complications. First, the comparison of signatures must be defined, and the concept of "difference" established. The most common approach is to compare the per-test results between prediction and behavior; a prediction error occurs, for example, when the chip fails a test that the fault candidate is predicted to pass. (For this and the next section, the discussion of predictions and behaviors will be limited to pass-fail results only.)The probability of this is $p$(chip fails | candidate predicts pass); in the standard notation of diagnosis, a 0 in a fault signature indicates a passing response and a 1 indicates a failing response, so the above expression reduces to $p(b=1|c=0)$, or more simply, $p(1|0)$.

Now, to continue calculating the required probabilities, we make a simplifying assumption: namely, that the outcomes (success or failure) of candidate predictions are independent. While dependencies will almost surely exist for most candidate predictions, accurate calculation of these dependencies is likely to be impractical or unnecessary given the inherent imprecision of fault simulation. (The limits of precision are especially obvious in the case of composite bridging fault signatures.) With this assumption of independence, the full conditional probability for a candidate can be expressed as

$$p(\mathbf{b}|\mathbf{c}) = \prod_{k=1}^{n} p(b_k|c_k)$$

where $k$ is the index over all $n$ test vectors, $b_k$ is the $k$th bit of the behavior signature, and $c_k$ is the $k$th bit of the candidate signature.

The value, then, of $p(\mathbf{b}|\mathbf{c}_i)$ for stuck-at candidates should be relatively easy to calculate: assuming an unbiased simulator with

$$p(0|1) = p(1|0) = (1.0 - p(1|1)) = (1.0 - p(0|0)) = x,$$

if a value or estimate can be assigned to $x$ (the *probability of prediction error* or *prediction error rate*) the score of each candidate can be expressed simply as the product of per-test probabilities. But this of course begs the question of what a good value for $x$ is, or what the expected rate of prediction error is for a particular stuck-at simulator.

It is possible that this value can, in some cases, be obtained statistically: perhaps sufficient failure analysis has been performed on a significant number of stuck-at defects to determine this probability with a high degree of confidence. Lacking this information, however, an estimate will have to suffice.

## 4.3 ... vs. acceptance criteria

It is widely accepted that the stuck-at fault has no single direct analog in the realm of silicon defects. Its closest manifestation would probably be a circuit node shorted to power or ground. If such shorts are the only defects targeted diagnostically with the stuck-at model, then the error rates for stuck-at predictions should be quite low, as a good correlation of defect behavior to simulation should occur. But if the stuck-at candidates are meant to target a wider range of defects, with less direct correlation to classical stuck-at faults, then higher error rates will have to be expected.

Regardless of the value chosen, the role of operator choice points out that the process of scoring fault candidates is largely an arbitrary one, in which the assignment of probabilities is really a matter of establishing *acceptance criteria* for the various fault models used. If a low stuck-at prediction error rate is used, then a stuck-at candidate with large error will be assigned a low conditional probability; compared with a candidate of another model with the same number of errors but a higher error rate, the stuck-at candidate will be scored lower.

Assigning an error rate for fault models has been done implicitly by almost every traditional fault diagnosis algorithm, and by failure analysis engineers who must reconcile the results from different diagnosis tools. Some algorithms do not tolerate any prediction error; they implicitly assign a zero probability of error and reject any imperfect candidate. Others expect much more error in one direction than another (such as in our previous bridging-fault scoring) and express this as lexicographic ratings. And, if an engineer uses multiple fault diagnosis algorithms and the top candidate reported by a stuck-at diagnosis tool, say, misses the same number of fault predictions as the top candidate from a bridging diagnosis program, then it is human judgement that decides how much error to tolerate in each model and therefore which candidate to prefer.

By adopting Bayes decision theory for fault diagnosis, then, we are arguing for making these implicit judgements explicit: the assignment of error rates to each fault model and its predictions is equivalent to stating acceptance criteria for each type of candidate employed. In the case of our proposed two-model diagnosis system, we will obviously have to accept or tolerate more prediction error with composite bridging signatures than with stuck-at candidates. In the spirit of full disclosure, specifying these usually-implicit values is intended to codify the assumptions and knowledge about the various fault models into a single diagnosis tool—where they can be examined and updated as necessary.

## 4.4 Stuck-at scoring

For this research no statistical information was available about the behavior of stuck-at defects in actual manufactured circuits. Therefore, the approach taken for candidate scoring necessarily involved an arbitrary assignment of per-vector prediction error for stuck-at candidates.

In general, fault diagnosis will be more effective and accurate if it targets more specific fault types and ties the models more directly to the defects targeted. It will be more effective because the increased precision greatly facilitates the subsequent work of physical failure analysis, and it will be more accurate because the fault predictions will be more accurate and therefore easier to match to the associated defects. This point was suggested in an *HP Journal* article by Maxwell and Aitken [3]; although their article recommended accurate bridging fault models (replaced here by cheaper composite signatures), the main idea still holds: use a fault model to diagnose only the defects that it best represents. The implication of this idea is that the expected error rate for stuck-at candidates should be set relatively low.

This philosophy argues for a relatively tight link between the stuck-at predictions used and the defects targeted for diagnosis. To this end, an expected prediction error rate of 1% was arbitrarily chosen for stuck-at fault candidates in the presented diagnosis system. This number was chosen simply as a guesse based on previous but limited experience with power or ground shorts and opens, the two defect types explicitly targeted with the stuck-at candidates.

## 4.5 0th-order bridging fault scoring

Since the assignment of an error rate for stuck-at candidates is largely arbitrary, the value of the error rate for bridging candidates is similarly arbitrary. It is the value of error rate *relative* to the stuck-at error rate that will determine the selection of bridging or stuck-at candidates for any particular diagnosis.

As detailed previously, the composite bridging fault signatures used in this system are only approximations to the expected behaviors, and a significant amount of prediction error is anticipated. Accordingly, the error rate assigned for bridging fault candidates should be significantly higher than that assigned to stuck-at candidates. For our purposes a significant difference will be at least an order of magnitude, so a 0th-order estimate for the bridging candidate error rate, given the stuck-at rate specified above, would be 10%. While this is admittedly a gross estimate, it is not far from the value seen in our previous experience with composite signatures vis-a-vis simulated bridging fault behavior [6].

## 4.6 1st-order bridging fault scoring

| Probability | Description |
|---|---|
| $p(\mathsf{sv})$ | Probability that a test puts the same logic value on both bridged nodes. |
| $p(\mathsf{hr})$ | Probability that high resistance of the short prevents (per-vector) any fault effect, regardless of gate type or topology. |
| $p(\mathsf{wf})$ | Probability that one node wins a drive fight and asserts a definite (faulty) logic value on the other node, but the corresponding stuck-at fault is not detected, causing no fault effect to result from the bridge (non-required vector only). |
| $p(\mathsf{bg})$ | Probability that a Byzantine Generals situation and reconvergent fanout downstream from the bridge invalidate a pass/fail prediction. |
| $p(\mathsf{fb})$ | Probability that fault-induced feedback invalidates a pass/fail prediction. |
| Assumptions: | The events $\{\mathsf{sv}, \mathsf{hr}\}$ are considered independent.<br>The events $\{\mathsf{sv}, \mathsf{wf}, \mathsf{bg}\}$ are considered mutually exclusive, as are $\{\mathsf{hr}, \mathsf{wf}, \mathsf{bg}\}$.<br>The events $\{\mathsf{fb}, \mathsf{hr}, \mathsf{wf}\}$ are considered mutually exclusive.<br>The events $\{\mathsf{fb}, \mathsf{bg}\}$ are approximated as independent; $p(\mathsf{fb})$ is dependent on $\mathsf{sv}$: $p(\mathsf{fb}) = p(\mathsf{fb}\|\mathsf{sv}) + p(\mathsf{fb}\|\neg\mathsf{sv})$. |

Table 1: Set of likely effects that can invalidate composite bridging fault predictions.

A better estimate for the bridging fault candidate error rate can be obtained by looking at the components of the composite signature, described earlier. Doing so points out that different per-vector predictions in a composite signature have very different expected errors. As the name implies, a required vector prediction is expected to fail very infrequently; similarly, a restricted vector should produce a passing result nearly all of the time. Also, misprediction is significantly more probable than nonprediction. Given these factors, we might reasonably assign individual error rates to the various types of composite predictions, again relative to the stuck-at error rate previously assigned: 10% for non-required vectors, 1% for nonprediction and and required vectors (since they rely on stuck-at assumptions), and 0.1% for restricted vectors. These values are consistent with those we have seen over thousands of simulated bridging-fault diagnoses, and provide a somewhat more accurate basis for discrimination than the simplistic 0th-order estimate given above.

### 4.7 2nd-order bridging fault scoring

It is possible to refine the estimates of prediction error further by examining the possible causes for the actual behavior to diverge from prediction. While this approaches the complicated topic of bridging fault modeling (the avoidance of which was the basis of the composite signature idea), the salient factors affecting composite bridging signatures can be identified relatively easily. They are summarized in Table 1.

With a little bit of thought, the relevant conditional error probabilities can be expressed as:

$$\begin{aligned}
p(0|1) &= p(\mathsf{sv}) + p(\mathsf{hr}) - p(\mathsf{sv})p(\mathsf{hr}) + p(\mathsf{bg}) \\
&\quad +(1 - p(\mathsf{bg}))(p(\mathsf{fb}|\mathsf{sv}) + p(\mathsf{fb}|\neg\mathsf{sv})) + p(\mathsf{wf}) \\
p(1|0) &= p(\mathsf{bg}) + (1 - p(\mathsf{bg}))(p(\mathsf{fb}|\mathsf{sv}) + p(\mathsf{fb}|\neg\mathsf{sv})) \\
p(1|0*) &= p(\mathsf{fb}|\mathsf{sv}) \\
p(0|1*) &= p(\mathsf{hr}) + p(\mathsf{bg}) + (1 - p(\mathsf{bg}))p(\mathsf{fb}|\neg\mathsf{sv}))
\end{aligned}$$

where $p(1|0*)$ refers to the restricted vector error rate and $p(0|1*)$ refers to the required vector error rate. While this degree of decomposition requires more calculation, it does offer certain benefits over the simpler 1st-order approximations. First, some of the probabilities are easy to estimate: $p(\mathsf{sv})$ can be approximated as 0.5, and $p(\mathsf{wf})$ as 0.25. Second, simulator and netlist information can provide accurate values for $p(\mathsf{sv})$, $p(\mathsf{wf})$, and $p(\mathsf{fb})$ on a per-candidate basis. But, values for such probabilities as $p(\mathsf{hr})$ and $p(\mathsf{bg})$ would require either extensive bridging fault characterization, or the assignment of estimates as described earlier (most likely relative to the stuck-at error rates). Given the philosophy of an inexpensive diagnosis system based on stuck-at simulation only, we have decided that the most practical and consistent approach is to use order-of-magnitude estimates for these values.

## 5 Experimental results

In order to evaluate the diagnosis approach just described, we implemented the technique and performed several diagnosis experiments on a production industrial circuit. The experiments were performed at Hewlett-Packard, with their support and equipment; the circuit used was a Hewlett-Packard ASIC. Defects were inserted into the circuits using a focused ion beam (FIB). (Knowing the exact form and location of a defect is obviously very useful for validation [2]; experiments on failing production chips will be the next step.) The circuit was built with a 0.5 micron process, and its ATPG model had approximately 150,000 gates.

There were three rounds of experimentation. In the first, we connected arbitrary signal lines to either power or ground in order to mimic stuck-at behavior. In the second round, we joined neighboring signal lines in order to represent bridging faults; in the third round we broke signal lines in order to simulate open defects.

The diagnosis experiments were performed despite several practical challenges. First, only pass-fail signatures were readily available, so no information about failing outputs was used. Second, we did not have access to a realistic bridging fault candidate list, so the diagnosis program had to consider all bridges to be possible. Third, no gate descriptions or simulator information was available for refinement of the $p(\mathsf{wf})$, $p(\mathsf{sv})$, or $p(\mathsf{fb})$ estimates used for composite bridging scoring. Fourth, no statistical analysis of fault or defect frequencies (such as IFA) was performed, so a uniform prior was used for the Bayesian scoring of candidate faults. It is assumed that the addition of any or all of these missing elements would improve the accuracy and resolution of the resulting diagnoses.

It is also important to reiterate that no information or tool was used for diagnosis other than a stuck-at faultlist, a pass-fail dictionary (from a stuck-at fault simulator), and a list of failing vectors for each faulty circuit.

The diagnosis program requires estimates of prediction error, or sources of error, for bridging and stuck-at fault candidates. The initial assignment for bridging faults was $p(\mathsf{sv}) = 0.5$, $p(\mathsf{wf}) = 0.25$, $p(\mathsf{hr}) = p(\mathsf{bg}) = p(\mathsf{fb}) = 0.01$, $p(\mathsf{fb}|\neg\mathsf{sv}) = 100 * p(\mathsf{fb}|\mathsf{sv})$. The resulting probabilities of error were $p(0|1) = .78$, $p(1|0) = 0.02$, $p(1|0*) = 0.0001$, $p(0|1*) = 0.03$. For stuck-at faults, $p(0|1) = p(1|0) = 0.01$. In most cases, the estimates were probably pessimistic.

The experiments were designed to see if the proposed algorithm could 1) distinguish between stuck-at and bridging defects, and 2) correctly identify the nodes involved in the defect. Also, we wanted to know how open defects would be diagnosed in this system, and whether suspicions about their similarity to stuck-at behaviors are justified. The results are given in Tables 2, 3, and 4.

Each of the three tables presents results from a round of experiments, for stuck-at, bridging, and open defects respectively. Each row in a table is an individual diagnosis experiment on a single defective circuit. The first column of each row gives the defect number. The second column (*Top Candidate*) indicates which candidate was scored highest by the diagnosis algorithm. More than one candidate can get the same top score, so the third column (*Num. Tied for First*) reports the number of top-scoring candidates. The fourth column (*Classification*) classifies each diagnosis, and the last column (*Notes*) gives a short qualitative description or details on each diagnosis.

In the tables, candidates are described by their model and quality of match to the actual inserted defect. The

| Defect | Top Candidate | Num. Tied for First | Classification | Notes |
|---|---|---|---|---|
| 1.1 | bf-partial | 2 | Partial success | Significantly (17%) non-stuckat behavior;<br>8 of top 10 candidates are bf-partial |
| 1.2 | sa-exact | 1 | Success | Next 100 candidates are bf-partial |
| 1.3 | sa-exact | $> 300$ | Ambiguous | |
| 1.4 | sa-exact | 9 | Success | |
| 1.5 | sa-exact | 3 | Success | Other 2 top cand. are near |
| 1.6 | sa-exact | $> 100$ | Ambiguous | |
| 1.7 | sa-exact | 1 | Success | Next 100 candidates all bf-partial, all same score |
| 1.8 | sa-exact | 1 | Success | Next 100 candidates all bf-partial, all same score |
| 1.9 | sa-exact | 2 | Success | Other top candidate is near; next 100 candidates are bf-partial, all same score |
| 1.10 | sa-exact | 3 | Success | Other top candidates are near; next 100 candidates are bf-partial, all same score |
| 1.11 | sa-exact | 1 | Success | Next 100 candidates all bf-partial, all same score |
| 1.12 | sa-exact | 2 | Success | Other top candidate is near; next 100 candidates are bf-partial, all same score |

Table 2: Diagnosis results for round 1 of the experiments: twelve stuck-at faults.

| Defect | Top Candidate | Num. Tied for First | Classification | Notes |
|---|---|---|---|---|
| 2.1 | bf-exact | 1 | Success | Defect is a feedback bridging fault |
| 2.2 | bf-exact | 1 | Success | Next 9 candidates are bf-partial |
| 2.3 | bf-exact | 1 | Success | Second candidate is sa-partial |
| 2.4 | bf-partial | 1 | Success | Second candidate is bf-exact;<br>other node in top candidate is near |
| 2.5 | bf-exact | 1 | Success | Next three candidates are bf-partial |
| 2.6 | bf-partial | 1 | Success | Second candidate is bf-partial,<br>third candidate is bf-exact |
| 2.7 | sa-partial | 1 | Partial success | Dominated node: see text |
| 2.8 | sa-partial | 1 | Partial success | Dominated node: see text |
| 2.9 | sa-partial | 1 | Partial success | Dominated node: see text |

Table 3: Diagnosis results for round 2 of the experiments: nine bridging faults.

two candidate models are *bf* for bridging fault and *sa* for stuck-at fault. Three grades of match between candidate and the actual defect are specified. An *exact* match exactly identifies the single node or pair of nodes involved in the defect. A *partial* match either identifies one out of two bridged nodes (for a stuck-at candidate), or pairs a stuck-at or open node with another unrelated node (for a bridging candidate). A *misleading* match does not correctly identify any faulted nodes, although the table indicates if an apparently unrelated node is logically near (within two simple logic gates up or downstream from) the fault site.

To illustrate, a stuck-at candidate that identifies one of a pair of shorted nodes would be considered *sa-partial*. A bridging candidate that pairs the correct stuck-at node with another would be *bf-partial*, as would a bridging candidate that only correctly identifies one of a pair of shorted nodes. For open defects, either stuck-at fault on the open nodes is considered *sa-exact*.

There are four diagnosis classifications for each experiment. A *success* indicates that an exact match is found in the top 10 candidates. A *partial success* indicates that at least a partial match, but no exact match, is contained in the top 10. A diagnosis is a *failure* if no exact or partial matches rank in the top 10. In any event a diagnosis is considered *ambiguous* if the top 10 matches, or more, all receive the same score. An ambiguous diagnosis indicates that more information (such as failing outputs, for example) is needed to distinguish between highly-ranked candidates.

A point of detail: The last three bridging defects, defects 2.7 to 2.9, all followed the same scenario, and are considered only partial successes. In all three cases, the FIB bridged the outputs of two inverters, one having a much stronger drive strength than the other. The result in such cases is that fault effects only initiate from the weaker node, the stronger node never being overdriven. All three of the diagnoses reflect this situation: in all three cases, the top candidate is the weaker of the two inverter outputs stuck-at either 1 or 0. Without the dominating node ever being the source of error effects, it is doubtful whether another algorithm could do better looking only at the logic failures of the circuit; perhaps $I_{DDQ}$ diagnosis

| Defect | Top Candidate | Num. Tied for First | Classification | Notes |
|---|---|---|---|---|
| 3.1 | sa-exact | 3 | Success | Behavior identical to node stuck-at 1 |
| 3.2 | sa-exact | 2 | Success | Behavior identical to node stuck-at 1 |
| 3.3 | sa-exact | 2 | Success | Behavior identical to node stuck-at 1 |
| 3.4 | sa-misleading | 2 | Failure | Significantly (21%) non-stuckat behavior; 14th candidate is bf-partial |

Table 4: Diagnosis results for round 3 of the experiments: four open faults.

has a chance of identifying this type of defect.

The results indicate that the approach works quite well at accurately diagnosing and distinguishing a mixture of fault types. The one failed diagnosis occurred on the last open defect, when the behavior was significantly (21%) different than the signature for the node stuck-at 1. Whether this behavior is typical or not is an area of further research; answering this question may lead to a refinement of the acceptance criteria for stuck-at candidates, or possibly the addition of another fault model specifically for open defects.

# 6 Conclusion

This paper describes an approach to fault diagnosis built around a probabilistic evaluation of a set of fault candidates given all the available data about a failing circuit. The introduction of probability as a common measure of diagnostic inference allows different algorithms to process different sets of data, using different sets and types of candidates, to produce a single diagnostic result.

Preliminary experiments on actual industrial circuits indicate that the proposed probabilistic multiple-fault-model approach to diagnosis has merit, at least for a limited implementation of two fault models. Developing and extending this approach into a comprehensive framework for failure analysis remains a goal of continuing research.

# References

[1] J. M. Acken and S. D. Millman. Fault model evolution for diagnosis: Accuracy vs precision. *Proceedings of the Custom Integrated Circuits Confernce*, 1992.

[2] R. Aitken. Finding defects with fault models. In *Proceedings of International Test Conference*, pages 498–505. IEEE, 1995.

[3] R. Aitken and P. Maxwell. Better models or better algorithms? On techniques to improve fault diagnosis. *Hewlett-Packard Journal*, February 1995.

[4] G. Greenstein and J. Patel. EPROOFS: a CMOS bridging fault simulator. In *Proceedings of International Conference on Computer-Aided Design*, pages 268–271. IEEE, 1992.

[5] C. Henderson and J. Soden. Signature analysis for IC diagnosis and failure analysis. In *Proceedings of International Test Conference*, pages 310–318. IEEE, 1997.

[6] D. B. Lavo, B. Chess, T. Larrabee, and F. J. Ferguson. Diagnosing realistic bridging faults with single stuck-at information. *IEEE Transactions on Computer-Aided Design*, pages 255–268, March 1998.

[7] D. B. Lavo, T. Larrabee, and B. Chess. Beyond Byzantine Generals: Unexpected behavior and bridging-fault diagnosis. In *Proceedings of International Test Conference*, pages 611–619. IEEE, 1996.

[8] P. Maxwell and R. Aitken. Biased voting: a method for simulating CMOS bridging faults in the presence of variable gate logic thresholds. In *Proceedings of International Test Conference*, pages 63–72. IEEE, 1993.

[9] S.D. Millman, E.J. McCluskey, and J.M. Acken. Diagnosising CMOS bridging faults with stuck-at fault dictionaries. In *Proceedings of International Test Conference*, pages 860–870. IEEE, 1990.

[10] R. Rodriguez-Montanes, E. M. J. G. Bruls, and J. Figueras. Bridging defects resistance measurements in a CMOS process. In *Proceedings of International Test Conference*, pages 892–899. IEEE, 1992.

[11] J. W. Sheppard and W. R. Simpson. Improving the accuracy of diagnostics provided by fault dictionaries. In *Proceedings of the 14th VLSI Test Symposium*, pages 180–185. IEEE, 1996.

[12] C. Thibeault. A novel probabilistic approach for IC diagnosis based on differential quiescent current signatures. In *Proceedings of the 1997 VLSI Test Symposium*, pages 80–85. IEEE, 1997.