Controlling DPPM through Volume Diagnosis*

Xiaochun Yu, Yen-Tzu Lin, Wing-Chiu Tam, Osei Poku, and R. D. (Shawn) Blanton
Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213, U.S.A.
{xyu1, blanton}@ece.cmu.edu

***Abstract* — We propose to achieve and maintain ultra-high quality of digital circuits on a per-design basis by (i) monitoring the type of failures that occur through volume diagnosis, and (ii) changing the test patterns to match the current failure population characteristics. Opposed to the current approach that assumes sufficient quality levels are maintained using the tests developed during the time of design, the methodology described here presupposes that fallout characteristics can change over time but with a time constant that is sufficiently slow, thereby allowing test content to be altered so as to maximize coverage of the failure types actually occurring. Even if this assumption proves to be false, the test content can be tuned to match the characteristics of the fallout population if the fallout characteristics are unchanging. Under either scenario, it should be then possible to minimize DPPM for a given constraint on test costs, or alternatively ensure that DPPM does not exceed some pre-determined threshold. Our approach does not have to cope with situations where fallout characteristics change rapidly (e.g. excursion), since there are existing methods to deal with them. Our methodology uses a diagnosis technique that can extract defect activation conditions, a new model for estimating DPPM, and an efficient test selection method for reducing DPPM based on volume diagnosis results. Circuit-level simulation involving various types of defects shows that DPPM could be reduced by 30% using our methodology. In addition, experiments on a real silicon chip failures show that DPPM can be significantly reduced, without additional test execution cost, by altering the content (but not the size) of the applied test set.**

***Key words*: test quality, DPPM, defect level, defect behavior classification, test selection, volume diagnosis.**

I. Introduction

The manufacturing test of integrated circuits (ICs) has traditionally been viewed as a filtering activity, that is, the main purpose of test has and continues to be to separate good ICs from ones that do not meet the desired operational characteristics. For several years now, test is being expanded as a value-added endeavor [1, 19, 20]. In particular, the diagnosis of test data for improving yield is a topic of great interest [2-8]. For example, the work in [3] uses diagnosis results to track failure rates for various design features. These empirically-observed failure rates are compared with expected failure rates to identify anomalies in the design and/or manufacturing process. The thinking is that identified anomalies can be remedied in order to improve yield.

Test data can also be used to continuously monitor quality. In [8], LSI describes the use of fallout (i.e., chips identified to be failing through test) from separate types of tests to determine the fraction of delay failures caused by front-end transistor defects and the portion due to back-end, in-line resistance defects. They suggest the use of adaptive testing and/or corrective action to improve test quality when a change in the relative occurrence rate of these defect types is identified. Here, we propose to use diagnosis-extracted models of chip failures along with a new model for estimating DPPM (number of defective parts per million shipped). Both are incorporated in an on-line, quality-monitoring methodology that ensures a desired level of quality by changing test content to match current fallout characteristics. This approach to quality is dynamic in nature and thus differs from the typical approach that assumes sufficient quality levels are maintained using the tests developed during the time of design or first silicon. Opposed to the aforementioned static approach, the methodology described here presupposes that fallout characteristics can change over time but with a time constant that is sufficiently slow, thereby allowing test content to be altered so as to maximize coverage of the failure types actually occurring. Even if this assumption is not true, it will then be possible to tune the test set to match the characteristics of the fallout population if the fallout characteristics are unchanging. Under either scenario, it is therefore possible to minimize DPPM for a given constraint on test costs, or alternatively, ensure that DPPM does not exceed some pre-determined threshold. Figure 1 illustrates the scenario where DPPM changes over time. At t_1, we start to predict DPPM based on a sufficient amount of data obtained during the period between t_0 and t_1. When predicted DPPM approaches a specific DPPM limit at t_2, test content is altered to lower DPPM at t_3. When DPPM is sufficiently low (e.g., t_5), we can trade-off test quality to reduce test application time, which leads to the slight increase in DPPM shown at t_6. Notice, however, at t_4, an excursion is assumed to have occurred. Our approach does not necessarily have to cope with excursions since techniques already exist to deal with them [18].

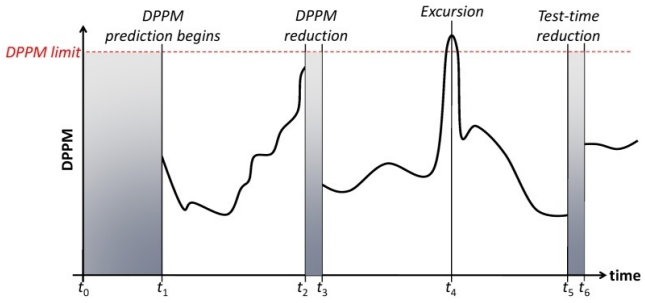

Figure 1. Illustration of how DPPM can change over time.

The rest of this paper is organized as follows. In section II, we describe the methodology employed for partitioning the failure population of a manufactured design into several categories that include front-end cell defects, back-end interconnect opens, and back-end bridges. The method is demonstrated using real silicon

*This work is supported by the National Science Foundation under award no. CCF-0427382.

1093-0167/09 $25.00 © 2009 IEEE
DOI 10.1109/VTS.2009.49

fail data from industrial test chips. In section III, we present a new model for estimating DPPM using diagnosis results. The method is used to estimate DPPM from the defect classification results derived in section II for various assumed values of yield. In section IV, we show how DPPM can be reduced, without additional test cost, by altering the content (but not the size) of the applied test set. In section V, we summarize our contributions, and finally, in section VI, we describe our on-going work in this area.

II. Defect Classification

In this section, we describe the methodology for classifying defects of a manufactured design into several categories that include front-end cell defects, back-end interconnect opens, back-end signal bridges, and supply bridges (i.e., permanent stuck-at defects). Specifically, conditions for various defects are analyzed in section IIA, defect classification based on various activation criteria is described in section IIB, and the calculation of expected frequencies of occurrence for various defect types is given in section IIC, and finally in section IID, real silicon data from industrial chips are used to demonstrate the defect classification.

A. Activation Conditions for Various Defects

The *neighborhood* of a site *f* is defined to be the inputs of the cell driving *f*, *f*'s physical neighbors (i.e., lines that are within some well-chosen distance of *f*), the inputs of the cells driving *f*'s physical neighbors, and the side inputs of *f*'s downstream gates [9]. If a site *f* is faulty, it is assumed that its faulty value is controlled by its neighborhood but not necessarily its entire neighborhood. In other words, depending on the nature of the defect that affects *f*, it may be only a portion of *f*'s neighborhood that controls *f*. We define the *relevant neighborhood* of a defect located at a site *f* as the set of signal lines in the neighborhood of *f* that influence the activation of the defect. The activation condition for a static defect (i.e., a defect that can be detected with a single test pattern) can be expressed as a function of its relevant neighborhood states, which are the sets of values on the signal lines in the relevant neighborhood. Next, we analyze the activation conditions for different types of defects and deduce their corresponding relevant neighborhoods.

Front-end cell: A front-end defect affecting a cell manifests as a change in the cell function. If a cell *G* driving a signal line *f* is affected by a front-end defect, *f* behaves like a temporary stuck-at-*v* fault ($v \in \{0, 1\}$) for one or more input value combinations of *G*, where the faulty implementation produces a different logic value from the defect-free implementation. Such defects can be modeled as input pattern faults [21]. Therefore, the relevant neighborhood of a front-end defect affecting a cell *G* driving a signal line *f* consists of the inputs of *G*, i.e., the inputs of the cell driving *f*.

Supply bridge: If a signal line *f* behaves like a bridge to one of the supply rails (power or ground), then *f* is permanently stuck-at-*v* ($v \in \{0, 1\}$) regardless of its neighborhood state. Therefore, a supply bridge does not have a relevant neighborhood.

Signal bridge: If there is a connection between two normally unconnected lines *f* and *g*, which is referred to as a back-end, signal-bridge defect between *f* and *g*, then *f* has a faulty value 0 (1) if the following conditions are satisfied. First, *g* is driven to a value 0 (1), while *f* is driven to the opposite value 1 (0). Second, a downstream gate of *f* interprets the voltage on *f* as an incorrect logic 0 (1). After a circuit is manufactured, the resistance and the position of the unexpected connection, the transistor sizing of the gates driving *f* and *g,* and the threshold voltage of *f*'s downstream logic are fixed (but unknown). Therefore, the interpretation of the voltage on *f* changes only when the logic values at the inputs of the gates driving *f* and *g* changes, which leads to a change in the drive strength of the cells driving *f* and *g*. Since two normally unconnected lines are only likely connected by a defect only if there is physical proximity between the two lines, the relevant neighborhood of a back-end bridge defect affecting a signal line *f* consists of (i) *f*'s physical neighbors, (ii) the inputs of the cell driving *f*, and (iii) the inputs of the cells driving *f*'s physical neighbors.

Open: If a signal line *f* is affected by a back-end interconnect open defect, the voltage on *f* can depend on (i) the voltage on the signal lines in the physical neighborhood of *f*, (ii) the voltage on the side inputs of *f*'s downstream gates, and (iii) the charge on *f*. Whether the voltage on *f* is interpreted as a logic 0 or 1 depends on the threshold voltage of *f*'s downstream gates. When the chip is manufactured, the threshold voltage of *f*'s downstream gates can be safely assumed to be constant within a certain period of time and the charge on *f* is also constant if gate leakage is negligible. Thus, the interpretation of the voltage on *f* changes only when the values of *f*'s physical neighborhood change and/or signal lines that are side inputs of *f*'s downstream gates change. Therefore, the relevant neighborhood of a back-end interconnect open defect consists of (i) *f*'s physical neighbors, and (ii) side inputs of *f*'s downstream gates.

B. Defect Classification

Based on the observation that the activation of a defect is a function of its corresponding relevant neighborhood state, if the activation condition of a defect affecting some site can be deduced from diagnosis, we can predict the type of the defect affecting the site. We use the technique of [9] to perform diagnosis of scan-based failures. The output of this diagnosis technique for a given failing chip is an extracted fault model that consists of a group of suspected defective sites. For each suspect site *f*, activation conditions that describe when *f* becomes faulty are expressed as a minimized logical function of its relevant neighborhood values. Each group of suspect sites along with their corresponding activation conditions "explain" all passing and failing patterns, that is, simulation of the extracted fault produces a simulation response that exactly matches what was observed on the tester [9]. We assume that only one defect affects a suspect site but multiple defects in the circuit are allowed. For each group of suspect sites, if the activation of a defect at a signal line *g* depends on the value at a signal line *f* and the converse is true, the only possibility is that *f* and *g* are affected by the same 2-line, back-end bridge defect. This notion is easily generalized to multiple lines. Otherwise, a defect affects only one signal line.

Each diagnosed defect is placed into one of five categories that include: (1) front-end cell, (2) back-end signal bridge, (3) back-end interconnect open, (4) supply bridge, or (5) unknown. The defect classification guidelines for each category are

summarized in Table 1. For a defective site f, let RN_f be the set of signal lines that appear in the deduced activation function for the defect affecting f, PN_f be the set of physical neighbors of f, DV_f be the inputs of the cell driving f, DPN_f be the inputs of the cells driving physical neighbors of f, and SI_f be the set of side inputs of the cells driven by f.

Defect type	Site	Classification guidelines
Supply bridge	f	$RN_f = \emptyset$
Front-end cell	f	$RN_f \neq \emptyset$ and $RN_f \setminus DV_f = \emptyset$ and $RN_f \cap PN_f = \emptyset$
Back-end signal bridge	f, g	$f \in RN_g, g \in RN_f$
	f	$RN_f \cap DPN_f \setminus PN_f \neq \emptyset$ and $RN_f \cap SI_f \setminus PN_f = \emptyset$
	f	$RN_f \cap DV_f \setminus PN_f \neq \emptyset$ and $RN_f \cap PN_f \setminus DV_f \neq \emptyset$ and $RN_f \cap SI_f \setminus PN_f = \emptyset$
Back-end interconnect open	f	$RN_f \cap SI_f \setminus PN_f \neq \emptyset$ and $RN_f \cap (DV_f \cup DPN_f) \setminus PN_f = \emptyset$
Either front-end cell or back-end signal bridge	f	$RN_f \setminus DV_f = \emptyset$ and $RN_f \cap PN_f \neq \emptyset$
Either back-end signal bridge or back-end interconnect open	f	$RN_f \neq \emptyset$ and $RN_f \setminus PN_f = \emptyset$
Unknown	None of above	

Table 1. Defect classification guidelines for various defects.

C. Defect Count Prediction

Assume there are F failing chips in total and for the i^{th} failing chip C_i, the diagnosis technique from [9] reports m_i groups of potential defective sites, where m_i is some integer constant. For the j^{th} group of the potential defective sites of C_i, denoted as group G_{ij}, assume there are d_{ij} defects, which have been classified using the rules described in Section IIB. Assume that each group of defects, G_{ij}, is equally likely to be the actual group of defects affecting C_i. In other words, the probability of each group of defects, G_{ij}, being the actual group of defects affecting C_i is $\frac{1}{m_i}$. Therefore, we can derive the expected number of defects of each type ($E[X]$), where $E[X]$ is the number of defects of type X in F, and $X \in$ {supply bridge, front-end cell, back-end signal bridge, back-end interconnect open, unknown}, and the expected total number of defects in F by applying "linearity of expectation" as follows [10].

$$E[X] = \sum_{i=1}^{F} E[X]_i = \sum_{i=1}^{F} \sum_{j=1}^{m_i} E[X]_{ij} \times \frac{1}{m_i}$$

where $E[X]_i$ is number of defects of type X in i^{th} failing chip, and $E[X]_{ij}$ is the number of defects of type X in group G_{ij}. Similarly,

$$E[Total\ No.of\ Defects\ in\ F\ failing\ chips] = \sum_{i=1}^{F} \sum_{j=1}^{m_i} d_{ij} \times \frac{1}{m_i}$$

D. Silicon Failure Classification

In our experiment, we perform diagnosis on F = 553 failing ALUs from LSI. Based on the diagnosis results, defects are classified according to the method described in Section IIB. Then, the expected number of defects of each type and the expected total number of defects are derived using the equations provided in Section IIC. A pie chart showing the defect type distribution is given in Figure 2. It shows that for the failure sample utilized in our experiment, a majority of defects are either back-end bridges involving signal lines or supply bridges. The expected number of defects for these 553 chips was calculated to be 669, indicating that some of the chips are affected by more than one defect.

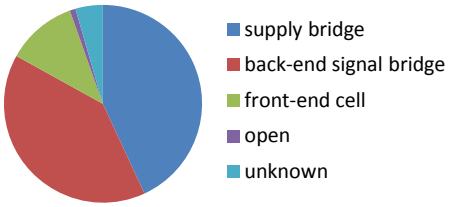

Figure 2. Defect type distribution for 553 failing chips diagnosed using DIAGNOSIX [9].

III. DPPM

In this section, we present a new model for estimating DPPM using the categorization of diagnosis-extracted faults. The derivation of the new model for estimating DPPM is given in Section IIIA, and the DPPM estimation for various assumed values of yield using the defect classification results from Section IID is provided in Section IIIB.

A. Defect Level Model

Defect level (DL) is the percentage of defective chips among the chips that are shipped to customers. Many models for predicting defect level have been proposed. For example, a model that predicts DL using multi-model fault coverage is proposed in [11]. It is based on the assumption that defects of different types are equally likely to occur, which of course may not be true in reality. In [12], a DL model that is based on the number of times each site is observed is described. The particular activation conditions of a given defect type however are not taken into account. It is quite likely that the values on the relevant neighbors of a defect that affects a site i are the same for many patterns where i is observed. In such cases, using a test set with a repeated number of observations does not necessarily lead to lower defect level especially for the static defects considered here. In addition, the model of [12] assumes that a defective chip passes the testing process and therefore is shipped if the erroneous effect of at least one actual defective site is not observed. However, in the context of multiple defects, a defective chip passes test only if no error from any of the actual defective sites are observed. So, the assumption in [12] can lead to an overestimation of defect level. We present here a new DL model that deals with the shortcomings just outlined and has some additional benefits as well.

For each site i, we define the following probabilities.

P_i(defective) = Probability of i affected by a defect.

P_i(detected) = Probability of detecting the defect that affects i.

$P_i(j)$ = Probability of a defect of type j affecting i.

$P_i(j$ detected) = Probability of detecting a defect of type j that affects site i.

$P_i(j$ activated | state k) = Probability of activating a defect of type j that affects a site i when the corresponding relevant neighborhood has state k.

To make the problem more tractable, we assume independence among the defect activation and corresponding error-propagation conditions for defects affecting different locations, and assume that a given site is affected by at most one defect. (As mentioned before, multiple defects are assumed possible but just not at the same site i.) A chip C is shipped if (i) C is defect-free, or (ii) none of the defects affecting C is detected by the testing process. In other words, a chip C is shipped if for every site i, there is no defect that affects i, or if there is one, it is not detected. Thus, the probability of C being shipped, P_{ship}, is computed as follows.

$$P_{ship} = \prod_{i \in \{all\ sites\}} (1 - P_i(defective) \times P_i(detected)) = \prod_{i \in \{all\ sites\}} (1 - \sum_{j \in \{all\ defect\ types\}} P_i(j) \times P_i(j\ detected))$$

Derivation of P_i(defective) and $P_i(j)$ are described next.

A defect of type j affecting a site i is detected if there is a test pattern t where the defect is activated and i is sensitized (despite the possible existence of other defects) to some observable point. Therefore, $P_i(j\ detected)$

= 1 – P(j is not activated for all patterns where i is sensitized)

$$= 1 - \prod_{k \in RNS_{ij}} (1 - P_i(j\ activated | state\ k))$$

where RNS_{ij} consists of all the states for the relevant neighborhood for a defect of type j at site i for the test patterns where site i is sensitized. This information is easily obtained from analyzing the applied test set.

Assume the probability of a defect of type j affecting a site i is the same for all sites. Thus, the fraction of a failure population affected by a defect of type j, which can be derived using the defect categorization method described in Section IIA, is equal to $\frac{P_i(j)}{\sum_{j \in \{all\ defect\ types\}} P_i(j)}$. Moreover, yield, which is the percentage of the manufactured chips that are defect-free, can be expressed as $Yield = \prod_{i \in \{all\ sites\}} (1 - \sum_{j \in \{all\ defect\ types\}} P_i(j))$. We therefore have $\sum_{j \in \{all\ defect\ types\}} P_i(j) = 1 - \sqrt[\sum_i 1]{Yield}$. Therefore, $P_i(j)$ can be computed for all types of defects at every site i for a certain yield value.

Using the equations for $P_i(j\ detected)$ and $P_i(j)$, we can derive P_{ship} for a given test set. Defect level can therefore be expressed as $DL = \frac{P_{ship} - Yield}{P_{ship}}$.

B. Experiment Results

Given that the level of yield is proprietary, we perform experiments for several assumed values of yield. For each yield value, P_{ship} and DL are computed according to the method described in Section IIIA using the defect categorization results derived in Section IID.

For simplicity, we assume for a defect of type j that affects a site i that for any state $k \in RNS_{ij}$, $P_i(j\ activated | state\ k) = \frac{1}{2}$. In other words, we assume each state for the relevant neighborhood for a defect of type j at a site i is equally likely to activate the defect at 50%. However, this probability can be made more precise for back-end defects by using defect density distribution and critical area information in the real application. Similar approaches can be used for front-end defects as well. The production test set is simulated to derive $RNS_{ij}\ \forall i, j$, that is, the set of relevant neighborhood states for each defect-type, site pair (i, j). Next, $P_i(j\ detected)$ is computed for every site i and every defect type j using the equation for $P_i(j\ detected)$ in Section IIIA. The probability of each type of defect affecting a site is derived using the method described in Section IIIA by plugging in the assumed yield value and the defect type distribution numbers from Figure 2, Section IID. Then, P_{ship} is computed using $P_i(j\ detected)$ and $P_i(j)$, and DL is derived using the equation given in Section IIIA. DL, which is reported as DPPM, is listed in Table 2 for various values of yield.

Yield	P_{ship}	DPPM
0.95	0.9514	1500
0.90	0.9028	3100
0.85	0.8541	4800
0.80	0.8053	6600
0.70	0.7074	10500

Table 2. DPPM for various yield values.

IV. Custom Test for Quality

In this section, we explore customizing the content (but not increasing the number of tests) of the applied test set to match the fallout characteristics for reducing DPPM. We describe the use of a test selection method [13, 14] for reducing DPPM in Section IVA, and predict the improvement in DL in Section IVB.

A. Test Selection for Reducing DPPM

According to the defect level model described in Section IIIA, DL can be reduced if for every site i, the defect possibly affecting i is activated using additional relevant neighborhood states for test patterns that sensitized i. In other words, we can reduce DL by applying a new test that sensitizes the site i and activates some defect affecting i using an uncovered relevant neighborhood state, i.e., a relevant neighborhood state which has not been applied by any test where i has been sensitized. Since the probability of a defect of a particular type affecting a given site is different, which is demonstrated in section IIB, a test that activates s uncovered relevant neighborhood states for one type of defect can be more preferable in reducing DPPM than a test that activates s uncovered relevant neighborhood states for another defect type. This observation is analyzed next.

For a given yield, since $DL = \frac{P_{ship} - Yield}{P_{ship}}$, defect level decreases as P_{ship} decreases. From Section IIIA, recall the expression for P_{ship}.

$P_{ship} = \prod_{i \in \{all\ sites\}} (1 - \sum_{j \in \{all\ defect\ types\}} P_i(j) \times P_i(j\ detected))$

Taking the log of both sides of the equation yields the expression:

$\log(P_{ship}) = \sum_{i \in \{all\ sites\}} \log(1 - \sum_{j \in \{all\ defect\ types\}} P_i(j) \times P_i(j\ detected))$

Therefore, activating a defect of type j at a sensitized site i in t_k using an uncovered relevant neighborhood state S_{ijk} can decrease $\log(P_{ship})$ by $abs(\log\left(\frac{1 - \sum_{j \in \{all\ defect\ types\}} P_i(j) \times P_i(j\ detected)'}{1 - \sum_{j \in \{all\ defect\ types\}} P_i(j) \times P_i(j\ detected)}\right))$, where $P_i(j\ detected) = 1 - \prod_{k \in RNS_{ij}} (1 - P_i(j\ activated | state\ k))$ and $P_i(j\ detected)' = 1 - \prod_{k \in RNS_{ij} + \{S_{ijk}\}} (1 - P_i(j\ activated | state\ k))$.

Again, RNS_{ij} is the set of all of relevant neighborhood states for a defect type j in the already selected test patterns when i is sensitized.

The total decrease in log (P_{ship}) due to applying a new test t_k is the summation of the decrease in log (P_{ship}) due to the activation of every uncovered relevant neighborhood state of each defect at every sensitized site i in t_k. The test that leads to the largest decrease in log (P_{ship}) is most preferable. Thus, our test generation approach selects tests, one at a time, from a large test set in a way that greedily decreases log (P_{ship}). The pseudo-code for test selection is given in Procedure 1.

Procedure 1 Test selection for reducing defect level.

```
Initialize the selected test set, T_sel, to be empty
Generate an N-detect test set, T_N-det
T_pool = T_N-det ⋃ original test set
while the size of T_sel is less than the original test set size do
  for each test t_k ∈ T_pool do
    Weight(t_k) = 0
    if t_k ∉ T_sel then
      for each uncovered relevant neighborhood state S_ijk established
      for a defect of type j at a sensitized site i in t_k do
        Weight(t_k) = Weight(t_k) + decrease of log(P_ship) due to S_ijk
      end for
    end if
  end for
  Add the test t_k with largest Weight(t_k) to T_sel
end while
```

B. *Improved Test Quality*

In this experiment, we use the methodology described in section IVA to select a test set of the same size as the original test set. A test set consisting of the union of a 20-detect test set generated by Encounter Test [15] and the original test set is used as the test pool T_{pool} for selection. The decrease in log (P_{ship}) due to activating a defect of type j at a sensitized site i using an uncovered relevant neighborhood state S_{ijk} is derived using the equation given in section IVA by plugging in the following information. First, the probability of a defect of type j affecting a site i is computed using the method described in section IIIB. Second, the likelihood of a relevant neighborhood state S_{ijk} to activate a defect of type j at site i is assumed to be 0.5 as was done in section IIIB. Finally, the relevant neighborhood states for the defect of type j in the already selected test patterns when i is sensitized is obtained from fault simulation.

The resulting DPPMs achieved by both the original and the selected test set for various assumed values of yield are given in Table 3. From Table 3, one can observe that DPPM can be reduced about 6% for the various levels of yield. Moreover, it is achieved without increasing the number of tests. Conversely, test selection can be continued until a desired DPPM is achieved.

In order to compare the quality of the new selected test set with traditional N-detect test sets, we generate several N-detect test sets using Encounter Test [15]. The selected test set achieves a lower DPPM than traditional N-detect test sets for any N ≤ 40. A 50-detect test set consisting of 908 patterns is able to achieve the same DPPM for various assumed yield values as the selected test set, which consists of 234 test patterns. Figure 3 shows the DPPM comparison between the selected test set and the traditional N-detect test set for a yield value of 85%. It shows that our selected test set improves test quality (i.e. reduces DPPM) with a smaller test set size than an N-detect test that is not neighborhood aware [22], nor focused on the characteristics of the failure population.

Yield	DPPM	
	Original test set	Selected test set
0.95	1500	1400
0.90	3100	2900
0.85	4800	4500
0.80	6600	6200
0.70	10500	9900

Table 3. DPPM comparison between the original test set and a new test set selected by our methodology for various assumed values of yield.

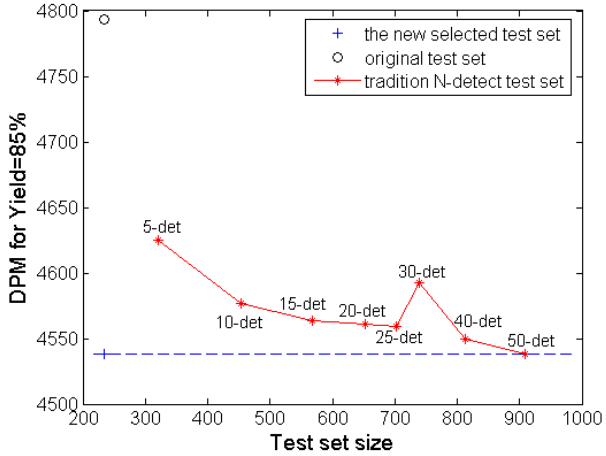

Figure 3. DPPM comparison between the original, selected, and N-detect test sets for an assumed yield = 85%.

C. *Simulation Validation*

In order to show that the number of test escapes can be reduce by adjusting the test content according to information deduced from diagnosing the current failure population, we perform a simulation experiment using the full-scan version of the ISCAS89 benchmark s713 [16]. We generated a pool of defective circuits by randomly injecting front-end, gross interconnect open, missing via, and two-line bridge defects, one at a time, into the layout of s713. Specifically, there are 419 front-end cell defects, 1,829 gross interconnect opens, 806 missing vias, and 512 two-line bridges. (Note that these defect-type counts do not adhere to our assumptions and is thus a good case to examine.) For each layout with an injected defect, we extracted the corresponding SPICE [17] netlist and perform circuit-level simulation using a test set that achieves 100% SSL fault efficiency (i.e., a 1-detect test set). We use a clock cycle that is much slower than the rated clock in order to mimic scan test. During simulation, we determine which defects are detected and those that escape detection.

To mimic application in a production environment, we do the following. For each defect type, we randomly select 25% of the defective circuits affected by that type. The selected defective circuits, referred to as $D_{current}$, constitute the fallout population that we analyze using our methodology. The remaining 75%, which are referred to as D_{future}, are assumed to be defective chips that are fabricated in the future. The test set applied to $D_{current}$ is a 1-detect test set. The test pool (T_{pool}) used for test selection is the union of a 50–detect test set generated by Encounter Test [15] and

the original 1-detect test set. Defect classification is performed on circuits in $D_{current}$ that fail the 1-detect test set to generate the defect type distribution using the method described in section II. Then, assuming yield is 85%, we select a test set T_{sel} of the same size as the original 1-detect test set using the test-selection method described in section IV-A and the derived defect-type distribution. Finally, for the defective circuits in D_{future}, we again use SPICE simulation to identify the test escapes, that is, the defective circuits in D_{future} that are not detected by our new selected test set. The test-set sizes and the number of defective circuits in D_{future} that escape each test set are listed in Table 4.

The simulation results reveal that the test set selected by our method is able to reduce the number of test escapes in D_{future} by 30% without increasing the number of tests. Moreover, the DPPM of our selected test set is only 1.7% higher than $T_{pool,}$, even though the size of $T_{pool,}$ is more than three times the size of our selected test set.

Test set	**Test set size**	**No. of test escapes**			
		Front-end	Missing via	Interconnect open	Bridge
1-det	48	27	39	66	37
T_{sel}	48	25	20	37	37
T_{pool}	155	24	20	37	36

Table 4. Comparison of test escapes corresponding to the original 1-detect test set, the new test set T_{sel} selected by our methodology, and $T_{pool,}$ for various defect types.

V. SUMMARY

We proposed an on-line, quality-monitoring methodology that ensures a desired level of quality on a per-design basis by changing test content to match diagnosed-derived fallout characteristics. The methodology starts from partitioning the failure population of a manufactured design into several categories using diagnosis results. Then, DPPM is estimated using a new defect-level model based on diagnosis results, and finally the content (but not the size) of the applied test set is altered using a test-selection methodology to reduce DPPM. The methodology was demonstrated using real silicon failures, resulting in a 6% reduction in DPPM without incurring any increase in test set size. The time taken to select the new test set using this methodology is around two hours using python [23]. The run time is expected to decrease significantly by exploring parallelism in the various steps in the methodology and using C instead of python for implementation. We also demonstrated the efficacy of the approach using detailed SPICE simulations. Specifically, defects of various types were injected into the layout of a benchmark circuit, extracted into a circuit-level netlist, and then simulated to produce virtual test responses. Applying our methodology to 25% of the virtual fail population showed that escapes from the remaining 75% could be reduced by 30%. This methodology currently accommodates failing chips affected by one or more static defects, but it can be extended to cope with sequential/delay dependent defects. Any detected defect, even if its behavior is not modeled by existing fault models, contributes to the reduction of DPPM. Although not demonstrated here, the methodology can also be easily used to trade-off DPPM and test execution cost in order to meet and maintain desired DPPM objectives.

VI. FUTURE WORK

Our future work is focused on expanding this methodology to address the various issues that stems from application within a production test environment. For example, we envision this methodology being applied in an on-going basis, to every design in production. This is necessary since each design has a unique susceptibility to various defect types. This means that the fallout of an in-production design is constantly being diagnosed to monitor the defect types occurring. During this monitoring, we envision that DPPM will fluctuate but the challenge will be to determine when intervention should occur in order to customize test to match the fallout in order to meet and maintain DPPM objectives.

REFERENCES

[1] Semiconductor Industry Association, *The International Technology Roadmap for Semiconductors*, Austin, TX: Int. SEMATECH, 2005.

[2] L. M. Huisman, M. Kassab, and L. Pastel, "Data Mining Integrated Circuit Fails with Fail Commonalities," *International Test Conf*., 2004.

[3] M. Keim, N. Tamarapalli, H. Tang, M. Sharma, J. Rajski, C. Schuermyer, and B. Benware, "A Rapid Yield Learning Flow Based on Production Integrated Layout-Aware Diagnosis," *International Test Conf*., 2006.

[4] H. Tang, S. Manish, J. Rajski, M. Keim, B. Benware, "Analyzing Volume Diagnosis Results with Statistical Learning for Yield Improvement," *European Test Symposium*, 2007.

[5] B. Seshadri, I. Pomeranz, S. Venkataraman, M. E. Amyeen, S. M. Reddy, "Dominance Based Analysis for Large Volume Production Fail Diagnosis," *VLSI Test Symp.*, 2006.

[6] J. Rajski, "Logic Diagnosis and Yield Learning," *Design and Diagnostics of Electronic Circuits and Systems,* 2007.

[7] B. Kruseman, A. Majhi, C. Hora, S. Eichenberger, J. Meirlevede, "Systematic Defects in Deep Sub-micron Technologies," *International Test Conference*, 2004.

[8] C. Schuermyer et al., "Behavioral Diagnosis of TDF/IRF Defects for Test Learning," *International Silicon Debug and Diagnosis Workshop*, 2006.

[9] R. Desineni, O. Poku, and R. D. Blanton, "A Logic Diagnosis Methodology for Improved Localization and Extraction of Accurate Defect Behavior," *International Test Conf.*, 2006.

[10] L. Wasserman, *All of Statistics: a Concise Course in Statistical Inference*, Chap. 3, Springer, 2004.

[11] S.-K. Lu, T.-Y. Lee, and C.-W. Wu, "Defect Level Prediction Using Multi-model Fault Coverage," *Asian Test Symposium*, 1999.

[12] M. R. Grimaila et al., "REDO-random Excitation and Deterministic Observation-first Commercial Experiment," *VLSI Test Symposium*, 1999.

[13] S. Seshu and D. N. Freeman, "The Diagnosis of Asynchronous Sequential Switching Systems," *IRE Trans. on Electronic Computers*, Vol. EC-11, Aug. 1962.

[14] Y.-T. Lin, O. Poku, N. K. Bhatti, and R. D. Blanton, "Physically-Aware N-Detect Test Pattern Selection", *Design and Test in Europe*, 2008.

[15] www.cadence.com

[16] F. Brglez et al. "Combinational Profiles of Sequential Benchmark Circuits," *International Symp. on Circuits and Systems*, 1989.

[17] Quarles, Thomas L., "Analysis of Performance and Convergence Issues for Circuit Simulation," *Memorandum No. UCB/ERL M89/42*, University of California, Berkeley, April 1989.

[18] W. Shindo et al., "Effective Excursion Detection by Defect Type Grouping in In-line Inspection and Classification," *IEEE Trans. on Semiconductor Manufacturing,* Vol. 12, pp. 3-10 Feb. 1999.

[19] S. Eichenberger et al., "Towards a World Without Test Escapes: The Use of Volume Diagnosis to Improve Test Quality," *International Test Conference*, 2008.

[20] K. M. Butler, J. M. Carulli, and J. Saxena, "Modeling Test Escape Rate as a Function of Multiple Coverages," *International Test Conference*, 2008.

[21] R. D. Blanton, and J. P. Hayes, "Properties of the Input Pattern Fault Model," *International Conference on Computer Design*, 1997.

[22] Y.-T. Lin, O. Poku, R. D. Blanton, P. Nigh, P. Lloyd, and V. Iyengar, "Evaluating the Effectiveness of Physically-Aware N-Detect Test Using Real Silicon," *International Test Conference*, 2008.

[23] M. Lutz, D. Ascber, Learning Python, *O'Reilly & Associates*, 2008.