Extraction of Defect Density and Size Distributions from Wafer Sort Test Results*

J. E. Nelson, T. Zanon, R. Desineni, J. G. Brown, N. Patil, W. Maly and R. D. Blanton
Center for Silicon System Implementation
Department of Electrical and Computer Engineering
Carnegie Mellon University, Pittsburgh, PA 15213
{maly, blanton}@ece.cmu.edu

Abstract

Defect density and defect size distributions (DDSDs) are key parameters used in IC yield loss predictions. Traditionally, memories and specialized test structures have been used to estimate these distributions. In this paper, we propose a strategy to accurately estimate DDSDs for shorts in metal layers using production IC test results.

1 Introduction

Yield loss estimation for nanoscale integrated circuits (ICs) is a central element of any Design for Manufacturability (DFM) activity. There are many reasons for yield loss; some of them are well understood and have sufficiently accurate models, while others are very elusive and hard to characterize [1]. Today's yield-related research addresses a large spectrum of reasons for yield loss, ranging from discovering new yield loss mechanisms to the architecture of the databases that store test results. Much attention is paid to modeling systematic yield loss. This work, however, addresses a narrow but important niche within the yield loss spectrum [2]. It proposes a new approach needed for accurate tuning of the critical area yield model—a workhorse of modern yield learning strategies [1].

Tuning yield models is very important, but is frequently an underinvested activity. Simply, the large variability of actual yield requires that yield model parameters are well characterized are frequently updated. This paper provides a blue print of a system that can automate characterization of parameters of a critical area-based model using already available wafer sort test results. The scope of this paper is limited to spot defects causing shorts. The relatively narrow focus is dictated by two reasons. First, it is important that the physics of the investigated yield loss mechanism are well understood; which is indeed the case for shorts. Second, spots of extra conducting material are still a major reason for IC malfunctions.

This paper is organized as follows. In the next section we propose a strategy for achieving the objectives of this paper. Then we formalize this strategy and explain the mathematical details of our approach. Finally, we describe an initial simulation experiment, which we have conducted to assess the validity of our methodology.

2 Proposed Approach

The objective of the research reported in this paper is to find the best approximation of key attributes of spot defects responsible for yield loss in an IC of interest. The most important characteristic of spot defects are densities of defects in all IC layers. It is intuitively obvious that defects of different sizes will have a different impact on the nature of IC functionality (*e.g.*, some small defects may be harmless altogether). Hence, we need to obtain both defect density and size distributions (DDSDs) for all layers of interest[1].

The strategy for achieving this goal is straightforward [3, 4] and is based on the simple observation that each harmful spot defect is likely to affect a specific, usually small, set of IC nodes, that are in the immediate neighborhood of the defect. Hence, a set of defects causing the same unique circuit response must have similar geometrical characteristics. Note that the probability of each unique circuit response can be determined from the tester results. Consequently, we can claim that test results can provide estimates of probabilities of occurrence of defects with specific geometrical attributes.

Note now that if we had a simulator capable of computing probabilities of test results in terms of attributes of defects of interest, we could manipulate the space of defect attributes such that the probabilities of simulated and real test responses are identical. We could then claim that the set of the defect attributes determined in this way represents the attributes of real defects in the process. Of course, the validity of the above claim is guarded by a set of conditions, which must be met. In this paper we propose a defect characterization methodology, which follows the above line of thought. That is, we develop and apply a simulation strategy that predicts probabilities of test failures given a DDSD, and then seek agreement between the test results measured by a tester and the simulated test results.

More precisely, the key challenge of the DDSD extraction

*This work is supported by the SRC under contract number 1172.001.

[1] Inspired by [2, 3]

strategy discussed in this paper is the development of a modeling technique that relates failures detected on output pins of the analyzed IC to geometries of the defect and the IC layout. In the subsequent sections, we describe two transformations that are needed to model such a relationship. First, we explain how to model the dependency between the probabilities of defects and probabilities of logic level circuit faults, and then between faults and failures of IC outputs.

3 Microevents and Macroevents

A spot of extra conducting material deposited in a metal layer may introduce an extra, unwanted connection between non-equipotential metal regions in this layer. In the vast majority of cases, such a connection will affect the electrical behavior of the circuit. We call any instance of such a short [5], which connects two or more non-equipotential metal islands, a *microevent*. Each microevent involves a set of circuit nodes, $S = \{n_1, n_2, \ldots, n_m\}$, that are shorted by the spot defect of a specific radius.

We define a *macroevent* to be the set of all microevents that exist for the same set of circuit nodes S. Note that there may be many microevents involving S in different layers for different defect radii. So each macroevent is described by a collection of independent microevents. Hence, the probability of a macroevent involving S is the sum of microevent probabilities involving S. It will be shown in the next section that the probability of a single, independent macroevent can be calculated using the critical area yield model [6].

3.1 Probability of a Microevent

The vulnerability of an IC to random spot defects is highly dependent on the layout. The concept of critical area was developed to provide a metric of sensitivity of a design to defects [6, 7]. Critical area is the region of the layout where if a spot defect of radius r occurs, a circuit fails.

The critical area yield model was derived from the well-known Poisson yield model. It has the form

$$Y_j = exp\left[-\int_0^\infty C_j(r)D_j(r)dr\right] \quad (1)$$

where Y_j is the probability that the defect causing the IC to fail did not occur in layer j, r is the defect radius, $C_j(r)$ is the critical area function of layer j, and $D_j(r)$ is the defect density and size distribution (DDSD) for layer j.

This model has been successfully used in the past [8, 9], and more recently was re-evaluated and shown to still be highly effective [1, 10]. In this paper we use exactly the same model but with a redefined notion of critical area. Simply, we are defining critical area as the region containing the centers of all defects causing one specific microevent. This way, the yield in the above formula is equivalent to the probability of a microevent of interest not occurring (or the yield of the microevent).

3.2 Probabilities of Macroevents

A number of techniques for critical area extraction have been proposed in the last twenty years [11, 12]. In the research discussed in this paper, we used the mask engine SiCat from PDF Solutions, which uses polygon operations to measure critical area for all microevents that can occur for a range of defect radii $[r_{min}, r_{max}]$.

Since our intermediate goal is to obtain a list of macroevents along with their associated probabilities of occurrence, we first perform critical area analysis to determine what microevents can occur, and simultaneously measure their critical area. Since a macroevent is the set of all microevents shorting the same circuit nodes, we can combine the critical area functions for each microevent to form critical area functions for each layer associated with the macroevent. Finally, using the critical area yield model, we can find the probability of each macroevent on our list. Equation (3) in Section 5.1 shows the probability of a macroevent not occurring.

4 Logic-Level Modeling

As discussed in Section 2, in the subsequent stage of our modeling we must find a mapping between the list of macroevents and test responses of the IC of interest. We define this mapping as a matrix $\boldsymbol{T}$ which is calculated by simulating all macroevents over a set of test vectors. In order to save simulation time due to the scale of the problem (even a small circuit will have hundreds of thousands of macroevents), macroevents are modeled using logic-level faults. To accurately model the macroevents at the logic-level, an accurate gate-level model of the IC design must first be derived.

4.1 From Layout to Logic

The challenge in creating an accurate gate-level model is that typical standard-cell representations abstract away the internal detail of the cell, causing important signal lines to be omitted from the logic-level netlist. Therefore, we map standard-cell layouts to the logic-level descriptions that capture the structure of static CMOS gates inside the cell using the gate primitives {NAND, NOR and NOT}. Now, rather than ignoring internal standard-cell metal routing during microevent extraction, gate outputs within a standard cell that are routed in metal layers are given global names and considered in the same way as standard cell pins. This means that any metal in the layout that is a CMOS gate output can now be mapped to a specific node in the logic-level netlist.

An AND gate illustrates the problem. Its typical CMOS implementation is a NAND gate followed by a NOT gate. If the output of the NAND gate is routed in metal, then there may be microevents that involve this internal signal line. The layout-to-logic mapping utilized here allows this signal line found in the layout to be correlated to the corresponding signal line in the logic netlist, thus leading to a more accurate

fault model. Without this mapping, there is no basis for forming a logic-level fault model including the internal signal lines. Note, however, that there may still be metal structures inside some standard cells that are not mapped to the logic level, an issue we are addressing in future work.

4.2 Fault Modeling

A macroevent represents a short that may involve two or more signal lines. Two-line shorts are commonly modeled using bridging faults [13]. Since the macroevents formed from microevents extracted from the layout may involve more than two lines, causing feedback or non-feedback behavior, more complex fault models are required to represent their behavior. In this paper, the voting bridge fault model is used along with pull-up/down network drive strengths for every standard cell.

The voting works by summing the drive strengths of all lines in the macroevent driven to logic 0 and logic 1, and comparing the summed strengths. The logic value with greater drive strength is imposed on all lines, causing all lines originally driven to the opposite value to have an error. Fault tuples [14], a generalized fault representation mechanism, are used to implement the voting model described above. Despite the measures taken to accurately model the macroevents, the behavior of real spot defects is still unpredictable, and can be a possible source of error.

4.3 Fault Simulation

FATSIM [15], a concurrent fault simulator for fault tuples, is used to simulate the macroevents, which are modeled as bridge faults. To determine which macroevents are detected by which test vectors, no fault dropping is used during simulation. The resulting data is stored in the $\boldsymbol{T}$ matrix, which has the following form:

$$\boldsymbol{T} = \begin{bmatrix} t_{1,1} & t_{1,2} & \cdots & t_{1,M} \\ \vdots & & & \vdots \\ t_{V,1} & t_{V,2} & \cdots & t_{V,M} \end{bmatrix}, \quad (2)$$

where V is the number of test vectors simulated, M is the total number of macroevents, and $t_{s,i}$ is a 1 (0) indicating that macroevent i is detected (undetected) by test vector s.

Accuracy of the $\boldsymbol{T}$ matrix is crucial to the overall accuracy of the proposed DDSD extraction approach. If macroevents cannot be modeled well, and hence cannot be simulated correctly, it can lead to a poor-quality $\boldsymbol{T}$ matrix. It is possible, for example, that a more rigorous simulation (*e.g.*, transistor-level spice simulation) or actual fab data could lead to greater accuracy in the $\boldsymbol{T}$ matrix, but such a gain comes at the expense of increased simulation time. In future work, we plan to analyze the impact of the $\boldsymbol{T}$ matrix on the accuracy of the DDSD extraction.

5 DDSD Extraction Theory

This section presents the mathematical background on how the macroevent probability from critical area yield theory can be combined with the empirical yield from wafer sort testing to extract the DDSDs for the metal layers of interest.

5.1 Macroevent Probabilities

Using the critical area yield model [6, 16], the probability $\boldsymbol{p_i^M}$ of a macroevent $\boldsymbol{i}$ *not* occurring in any layer can be calculated as

$$\boldsymbol{p_i^M} = \prod_{\boldsymbol{j=1}}^{\boldsymbol{L}} \left[\boldsymbol{exp}\left(- \int_{\boldsymbol{r_{min_j}}}^{\boldsymbol{r_{max_j}}} \boldsymbol{C_{i,j}(r) D_j(r) dr} \right) \right], \quad (3)$$

where $[\boldsymbol{r_{min_j}}, \boldsymbol{r_{max_j}}]$ is the interval of defect radii in layer $\boldsymbol{j}$, $\boldsymbol{D_j(r)}$ is the DDSD and $\boldsymbol{C_{i,j}(r)}$ is the critical area function associated with macroevent $\boldsymbol{i}$ for its microevents in layer $\boldsymbol{j}$.

Note that $\boldsymbol{C_{i,j}(r)}$ and $\boldsymbol{D_j(r)}$ are continuous functions. To reduce complexity, the integral be can approximated by splitting the range of defect sizes $[\boldsymbol{r_{min_j}}, \boldsymbol{r_{max_j}}]$ for each layer $\boldsymbol{j}$ into $\boldsymbol{N_j}$ *defect size bins* of varying width. Using bins allows the extraction of DDSDs that do not follow any predefined distribution model, such as the standard power-law distribution [7].

The probability of macroevent $\boldsymbol{i}$ *not* occurring can now be expressed as

$$\boldsymbol{p_i^M} = \boldsymbol{exp}\left[- \sum_{\boldsymbol{j=1}}^{\boldsymbol{L}} \sum_{\boldsymbol{k=1}}^{\boldsymbol{N_j}} \boldsymbol{C^a_{i,j,k} D^a_{j,k} \Delta r_{j,k}} \right], \quad (4)$$

where $\boldsymbol{\Delta r_{j,k}}$ are the widths of the $\boldsymbol{k = 1 \ldots N_j}$ defect size bins in layer $\boldsymbol{j}$, and $\boldsymbol{C^a_{i,j,k}}$ and $\boldsymbol{D^a_{j,k}}$ are the discretized, approximated versions of the continuous critical area functions and DDSDs, respectively.

5.2 Yield Per Test

To estimate the probability of a macroevent occurring, test results are used. The $\boldsymbol{T}$ matrix determines the set of macroevents that can be detected by each test vector. Combining the simulated test results, $\boldsymbol{t_{s,i}}$, from the $\boldsymbol{T}$ matrix with Equation (4) leads to the modeled probability of vector $\boldsymbol{s}$ *not failing*:

$$\boldsymbol{p_s^V} = \boldsymbol{exp}\left[- \sum_{\boldsymbol{i=1}}^{\boldsymbol{M}} \boldsymbol{t_{s,i}} \sum_{\boldsymbol{j=1}}^{\boldsymbol{L}} \sum_{\boldsymbol{k=1}}^{\boldsymbol{N_j}} \boldsymbol{C^a_{i,j,k} D^a_{j,k} \Delta r_{j,k}} \right]. \quad (5)$$

From the tester, the probability for vector $\boldsymbol{s}$ not failing, $\boldsymbol{\hat{p}_s}$, is the observed *yield per test vector*. It can be empirically measured using

$$\boldsymbol{\hat{p}_s = 1 - n_s / n} \quad (6)$$

where $\boldsymbol{n}$ is the total number of tested ICs, and $\boldsymbol{n_s}$ is the number of times vector $\boldsymbol{s}$ fails[2].

[2]Due to tester memory limitations and the high cost of test time, only test results up to the first or first few failing pattern are typically available. The

5.3 DDSD Extraction

This section presents the final steps to actually extract the DDSD. The problem is formulated using matrices to make solving such a system more tractable[3].

The key idea centers around abandoning the concept of individual DDSDs per layer. Hence, the defect densities of all defect size bins of all layers are concatenated into a single vector $\tilde{D}^a$ with $X = (\sum_{j=1}^{L} N_j)$ components. This vector contains DDSD information for all layers.

The critical area functions $C^a_{i,j,k}$, which depend on three variables i, j, and k, can be now be represented as a two-dimensional $(M \times X)$-matrix $\tilde{C}^a$, where M represents the number of macroevents and again X is the total number of DDSD components to be extracted for all layers.

After applying a logarithmic transformation to both the empirical and the modeled yields per test vector, the system consists of X variables, $\tilde{D}^a_1, \ldots, \tilde{D}^a_X$ that are used to approximate the DDSD in each layer j by minimizing the following expression:

$$\min_{\tilde{D}^a} ||ln(\hat{p}^V) + A\tilde{D}^a||^2, \quad (7)$$

where the A matrix is generated by multiplying the T matrix and the re-structured critical area matrix $\tilde{C}^a$, and $\hat{p}^V$ is a vector of the empirically measured yields per test.

Minimizing Equation (7) determines the value for each $\tilde{D}^a_x$ that leads to the least error between the empirical and the theoretically predicted yield per test vector using the critical are yield model. To uniquely minimize Equation (7), the number of equations (V) must be greater than the number of unknowns (X).

Hence, the minimization problem can be stated as a linear regression analysis where the number of test vectors is the sample size of the regression, the entries in the columns of the A matrix are the independent variables (regressors), the yields per test ($\hat{p}^V$) are the dependent variables (response), and the unknown defect densities ($\tilde{D}^a$) are the regression coefficients to be determined by the regression analysis. Regression analysis of this form can be performed with many available tools.

6 Simulation Experiment

In this section, a simulation-based experiment to demonstrate the viability of the proposed methodology is presented. Specifically, defects based on a presumed DDSD are inserted into a chosen demonstration circuit, and the resulting yield per test vector is measured via tester emulation. The DDSD for each layer is then extracted using the proposed methodology, and compared against the inserted DDSD.

6.1 Demonstration Circuit

The demonstration circuit for this experiment is based on the ISCAS'85 benchmark circuit c3540. The circuit is first logically optimized and then technology-mapped for a 0.18μm standard-cell library. The final design utilizes five metal layers and covers an area of approximately 100 μm$\times$100 μm.

Considering the defect densities exhibited by modern IC manufacturing, a single c3540 design is very small and is likely free of any random spot defects. To obtain a reasonable demonstration circuit for simulation, 10,000 instances of c3540 are "combined" to generate a total die area of approximately 1 cm^2. We assume, however, that each instance of c3540 retains its controllability and observability, allowing the entire die to still be tested with a test set for a single instance of c3540. Although this approach increases the amount of total critical area for the demonstration circuit, it lacks the design diversity found in circuits of similar sizes. Nevertheless, the impact of design diversity on the DDSD extraction methodology is not the prime focus of this initial experiment and is therefore left to future work.

Given this demonstration circuit, the next step in the simulation experiment involves macroevent extraction, modeling, and simulation to generate the T matrix. The extraction stage finds critical area functions, $C^a_{i,j,k}$, for defect sizes ranging from 0.2-2μm for metal layers one through four, and 0.35-2μm for metal layer five. For each metal layer, the critical area function was sampled over the defect size range at 50 nm intervals, which results in a total of 182 critical area points. The lower bounds of the defect sizes utilized in the extraction is determined by the minimum line spacing, and the upper bound is meant to capture a significant portion of the tail of the inserted DDSD. The total discretized critical area function (sum of critical area functions of all the microevents involving the layer) for each of the five metal layers, $C^a_{j,k}$ for one instance of c3540 is shown as the 182 white symbols in Figure 1. The 19 larger symbols are explained in Section 6.3.

A 100% SSL fault coverage test set consisting of 155 test patterns is used as the production test set. Finally, the macroevents are fault simulated against the test set using FATSIM [15] to generate the T matrix.

6.2 Tester Emulation

The proposed DDSD extraction methodology utilizes structural test results in the form of yield per test vector. In the absence of tester data from real ICs for this initial experiment, we use an emulation approach to generate the yields per test. Defects are first generated according to a stochastic Poisson process (independent events) and follow the well-known power-law defect size distributions with the defect densities[4] shown in Table 1.

Based on the assumption of statistical independence of the

proposed strategy can be easily adapted to account for this as shown in [4].

[3] A more detailed description is provided in [4].

[4] The injected defect densities are assumed higher than the levels typically found in real manufacturing lines. This was done to reduce the simulation time required for tester emulation.

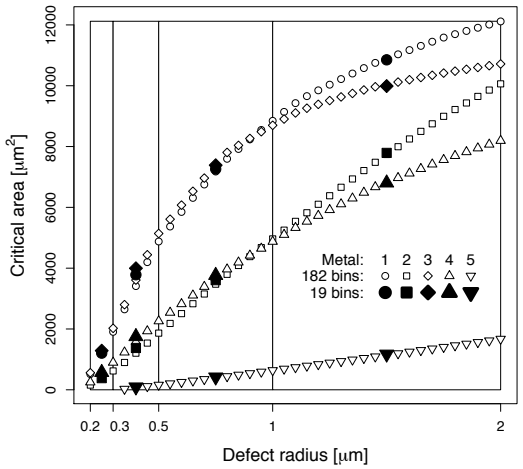

Figure 1: Critical area functions (white symbols) extracted from all metal layers of a single instance of c3540. The black symbols represent the critical area functions after combining a range of defect size bins.

	m1	**m2**	**m3**	**m4**	**m5**
$\boldsymbol{D_0}$ [cm^{-2}]	1	2	2	1	3
$\boldsymbol{p}$	3	4	3	2	3

Table 1: Injected DDSDs for this experiment follow the power-law distribution with the power parameter $\boldsymbol{p}$ and peak-probability parameter $\boldsymbol{X_0 = 0.05\mu m}$ for each metal layer. $\boldsymbol{D_0}$ [cm^{-2}] represents the defect density.

defects, the occurrence of macroevents can also be considered as an independent Poisson process. Consequently, the frequency of occurrence of each macroevent for each die is governed by a Poisson process with a rate determined by the critical area function associated with the macroevent and the assumed DDSDs. The rate of occurrence for the number of macroevents per die for the sample size analyzed in this experiment (50,000) is shown in Table 2. Two main observations can be made from Table 2: 1) 50,000 dice is likely an adequate sample size, since the rate of occurrence of the number of macroevents per die closely follows the rates expected from Poisson theory, and 2) (only) a small percentage of the simulated dice are affected by multiple macroevents (defects).

Macroevents per die	0	1	2	3
Number of dice	94.17%	5.67%	0.15%	0.01%

Table 2: The rate of occurrence for the number of macroevents per die for a sample size of 50,000 dice.

At this stage, it is known which macroevent(s), if any, occurs on each die. The yield per test vector is subsequently obtained by simply inspecting the $\boldsymbol{T}$ matrix. The yield per test for each test varies slightly around an average of 98%.

For dice with multiple macroevents, it is assumed that a test that detects any of the individual macroevents will cause the test to fail for the die neglecting any potential masking effects among macroevents. Nevertheless, this simplification does not have a significant impact on the yields per test vector, since the number of dice with multiple macroevents is very small ($\approx$0.15%).

6.3 Extraction of DDSD

In Section 5.3, the DDSD extraction process was formulated as a minimization problem to be solved using linear regression analysis. In this section, the details of the regression analysis procedure used for extraction of the DDSD for the demonstration circuit are described.

As described in Section 6.1, the total number of defect size bins for all layers is 182. The individual defect densities in these 182 bins make up the DDSD vector $\tilde{\boldsymbol{D}}^{\boldsymbol{a}}$ to be solved for in Equation (7). However, given that there are only 155 test vectors, only 155 yields per test ($\hat{\boldsymbol{p}}^{\boldsymbol{V}}$) can be obtained. Consequently, there are more unknowns than equations, which makes the minimization an under-determined problem with an infinite number of solutions.

To reduce the number of unknowns from 182, sample points for defect sizes are lumped together into *fewer, wider* bins. The bins are shown in Figure 1 separated by vertical lines. The number of bins for all five metal layers totals to 19. The critical area functions are recalculated for the new bins and are shown as the larger, black symbols in Figure 1. These new critical area functions are reflected in a new $\boldsymbol{A}$ matrix. Note that the $\boldsymbol{T}$ matrix is not at all affected by binning and therefore does not have to be regenerated. Using the defect size bin configuration of Figure 1, the unknown DDSDs, $\tilde{\boldsymbol{D}}^{\boldsymbol{a}}$, are found using principal component regression (PCR). 95% confidence intervals for the extracted DDSDs are obtained using standard bootstrapping techniques.

The final extracted results of the analysis for three of the five metal layers are shown in Figure 2. The white triangles show the 19 extracted DDSD components that make up $\tilde{\boldsymbol{D}}^{\boldsymbol{a}}$, while the small circles indicate the DDSD components that were inserted for the experiment. While not perfect, the inserted DDSD and the extracted DDSD correlate well—a positive and promising result.

Also shown in Figure 2 are the 95% confidence intervals for each DDSD component. Some of the confidence intervals are quite large. The source of this problem can be traced to the properties of the $\boldsymbol{A}$ matrix, where the columns are highly correlated. This multi-colinearity problem is an open research problem that suggests an opportunity for ATPG to generate test sets that have desirable properties for the proposed DDSD extraction methodology and therefore is a focus of our future work.

7 Conclusions

A novel methodology has been described to extract the defect density and size distribution (DDSD) for all layers of a process using the product being manufactured. Microevent extraction from the layout to form macroevents, high-fidelity

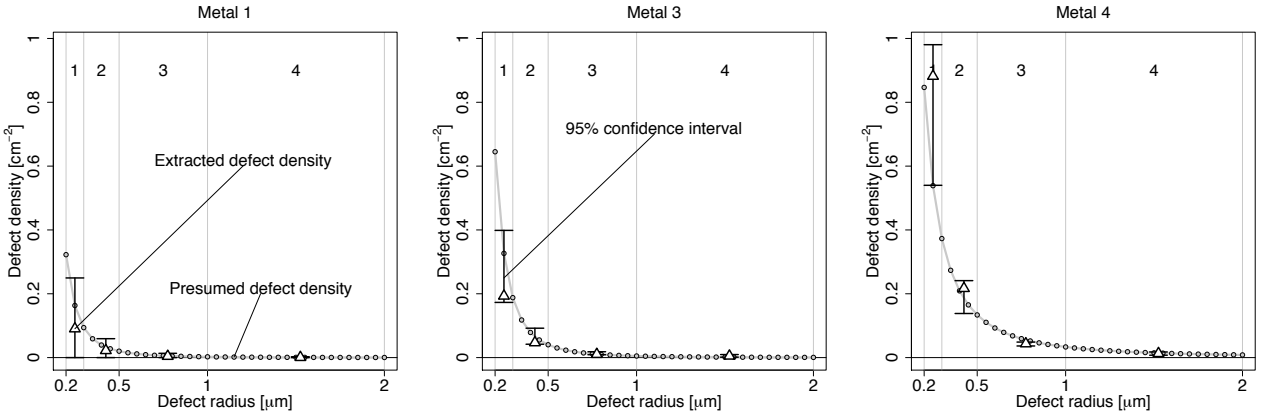

Figure 2: Assumed and extracted DDSDs for three of the five metal layers and the corresponding 95% confidence intervals.

fault modeling and simulation, and empirical tester results were combined to form a system of equations that are solved using linear regression to derive the DDSD. A simulation experiment seems to indicate that the approach will work, and certainly merits further investigation.

Significantly, the proposed methodology does not require the extra effort of designing, fabricating, and testing special test structures, but instead uses the existing product being manufactured and available test results. In other words, we use an ordinary digital circuit as a **virtual test structure** to extract DDSDs for short type defects, which thus far has been accomplished using test structures and memory only.

Our strategy offers a unique opportunity for **fabless** companies to gain **insight** into the fabrication of their chips. Now, such companies can independently compute defect characteristics about their product, and improve their design yield by tuning their design for a given fabline.

However, in the face of a real, industrial design, challenges arise that were not addressed in this paper. Noise in the test results caused by defect types other than shorts or even multiple shorts, limited ATE resources for storing pass/fail results, and more accurate defect modeling and simulation are among those challenges and are currently being investigated.

8 Acknowledgments

The authors would like to thank Dennis Ciplickas, Mariusz Niewczas, and many others from PDF Solutions for useful discussions that helped define the objectives of this work, and for providing access to SiCat. We would also like to acknowledge the financial support of the Semiconductor Research Corporation under contract number 1172.001.

References

[1] J. Kibarian, "The nature of yield ramping: Keeping ahead of evolution," *International Test Conference, Keynote Address*, 2005.

[2] Y. J. Kwon and D. M. H. Walker, "Yield Learning via Functional Test Data," *International Test Conference*, pp. 626–635, Oct. 1995.

[3] W. Maly, "Spot Defect Size Measurements Using Results of Functional Test for Yield Loss Modeling of VLSI IC," *White Paper, CMU*, Sep. 2004.

[4] J. E. Nelson, T. Zanon, R. Desineni, J.G. Brown, N. Patil, W. Maly, and R. D. Blanton, "Extraction of Defect Densities and Size Distributions from Wafer Probe Test Results," *CSSI Technology Report #05-02, Carnegie Mellon University*, Feb 2005.

[5] J. Khare, D. Feltham and W. Maly, "Accurate Estimation of Defect-Related Yield Loss in Reconfigurable VLSI Circuits," *IEEE Journal of Solid State Circuits*, vol. 8, no. 2, pp. 146–156, Feb. 1993.

[6] W. Maly and J. Deszczka, "Yield Estimation Model for VLSI Artwork Evaluation," *Electronic Letters*, vol. 19, no. 6, pp. 226–227, March 1983.

[7] C. H. Stapper, "Modeling of Integrated Circuit Defect Sensitivities," *IBM Journal of Research and Development*, vol. 27, no. 6, pp. 549–557, Nov. 1983.

[8] D. Schmitt-Landsiedel, D. Keitel-Schulz, J. Khare, S. Griep and W. Maly, "Critical Area Analysis for Design Based Yield Improvements of VLSI Circuits," *Quality and Reliability Engineering International*, vol. 11, pp. 225–232, 1995.

[9] D. J. Ciplickas, X. Li and A. J. Strojwas, "Predictive Yield Modeling of VLSICs," *Proc. of International Workshop on Statistical Metrology*, pp. 28–37, 2000.

[10] P. Simon, "Yield Modeling for Deep Sub-Micron IC Design," *Ph.D. Thesis, Technical University of Eindhoven*, 2001.

[11] G. A. Allan and A. J. Walton, "Efficient Critical Area Algorithms and Their Application to Yield Improvement and Test Strategies," in *Proc. of IEEE International Workshop on Defect and Fault Tolerance in VLSI Systems*, pp. 88–96, Oct. 1994.

[12] C. Ouyang and W. Maly, "Efficient Extraction of Critical Area in Large VLSI ICs," *Proc. of IEEE International Workshop on Defect and Fault Tolerance of VLSI Systems*, pp. 301–304, 1996.

[13] K. C. Y. Mei, "Bridging and Stuck-at Faults," *IEEE Trans. on Computers*, vol. 23, no. 7, pp. 720–727, July 1974.

[14] R. D. Blanton, "Methods for Characterizing, Generating Test Sequences for, and Simulating Integrated Circuit Faults Using Fault Tuples and Related Systems and Computer Program Products," Dec. 2004, U.S. Patent No. 6,836,856.

[15] K. N. Dwarakanath, "Fault Tuples: A Paradigm for Universal Test Analysis," *Technology Report. CMU-CAD 01-21, Carnegie Mellon University*, Nov. 2001.

[16] P. K. Nag and W. Maly, "Yield Estimation of VLSI Circuits," *Proc. of TECHCON 90*, pp. 267–270, Oct. 1990.