2012 IEEE 21st Asian Test Symposium

Tailoring Tests for Functional Binning of Integrated Circuits

Suraj Sindia* Vishwani D. Agrawal†
Department of Electrical and Computer Engineering
Auburn University, Alabama, AL 36849, USA
*Email: szs0063@auburn.edu †Email: vagrawal@eng.auburn.edu

***Abstract*—In recent years, a number of high level applications have been reported to be tolerant to errors resulting from a sizable fraction of all single stuck-at faults in hardware. Production testing of devices targeted towards such applications calls for a test vector set that is tailored to maximize the coverage of faults that lead to functionally malignant errors while minimizing the coverage of faults that produce functionally benign errors. Given a partitioning of the fault set as benign and malignant, and a complete test vector set that covers all faults, in this paper, we formulate an integer linear programming (ILP) problem to find an optimal test vector set that ensures 100% coverage of malignant faults and minimizes coverage of benign faults. We also propose a test strategy based on selectively masking appropriate outputs of the circuit to partition the circuits at production test into three bins - malignant, benign, and fault-free. As a case study, we demonstrate the proposed ILP based test optimization and functional binning on three adder circuits: 16-bit ripple carry adder, 16-bit carry lookahead adder, and 16-bit carry select adder. We find that the proposed ILP based optimization gives a reduction of about 90% in fault coverage of benign faults while ensuring 100% coverage of malignant faults. This typically translates to an (early manufacturing) yield improvement of over 20% over what would have been the yield if both malignant and benign faults are targeted without distinction by the test vector set.**

I. Introduction

Scaling of MOS transistor dimensions (thanks to Moore's law) has led to a steady increase in functions offered by microprocessor chips. Additionally, the performance (or speed) offered by these scaled devices has also been exponentially increasing. The unprecedented growth in performance of computers, however, has come at a price of an exponential increase in power density (power per unit area). After a point, roughly starting from the later half of the last decade, manufacturers have restrained from increasing the operating frequency of microprocessor chips. This stalling in frequency has prompted microprocessor industry to shift to an alternative computing paradigm such as parallel computing, where individual computers perform at a slower rate, but manage to accomplish functional tasks concurrently to be counted as an individual computer operating at a much faster rate (being roughly equal to number of parallel processors × operating frequency of individual processor).

Another possible route to mitigate the increase in power density with successive generations of a microprocessor chip, without stalling frequency scaling, is to downscale the supply voltage. Such a scaling model is popularly referred to as constant electric field scaling [5].

Regardless of the route taken to minimize power density to keep up the performance gains, the semiconductor industry is beginning to hit a road-block attributed to increased manufacturing process related variations. Reference [1] presents an insightful discussion on the trends in frequency and voltage scaling in the face of increased process variation in advanced CMOS technology nodes. Table I (reproduced from [1]) shows scaling trends in CMOS and its impact on energy, and speed in the advanced CMOS nodes. It predicts variability in transistor characteristics, both within-die and across dice, as the single most important impediment to performance gains in highly scaled CMOS nodes. Variability in transistor characteristics within the chip has led to a few gates (also referred to in literature as "outliers"), located at spatially disjoint locations to offer delays that are significantly higher, and in many cases, these "outlier" gates lie on the critical path, or paths that would nominally (without any process variation) have had delays that are comparable to critical path delay. Presence of an "outlier" on critical path or close to critical path leads to an abrupt increase in the delay offered by these paths, consequently, reducing the maximum operable frequency at which functionality of the circuit is guaranteed to be correct.

However, if one can trade functionality for speed, that is, under the assumption that only a few paths may have these "outliers," then we should still be able to operate the

1081-7735/12 $26.00 © 2012 IEEE
DOI 10.1109/ATS.2012.78

TABLE I
TECHNOLOGY SCALING PREDICTIONS FOR THE END-OF-CMOS ERA [1]. MANUFACTURING PROCESS VARIATION IS PROJECTED AS THE SINGLE BIGGEST IMPEDIMENT FOR PERFORMANCE AND ENERGY IMPROVEMENT WITH DEVICE SCALING.

High Volume Manufacturing	2006	2008	2010	2012	2014	2016	2018
Technology node (nm)	65	45	32	22	16	11	8
Integration capacity	4	8	16	32	64	128	256
Delay= $\frac{CV}{I}$ scaling	≈ 0.7	> 0.7	Delay scaling will slow down				
Energy/Logic Op scaling	>0.5	> 0.5	Energy scaling will slow down				
Variability	Medium		High		Very High		

circuit at its maximum speed (as if there were no process variation) albeit with a few errors. The difficulty, however, is that most test procedures in manufacturing flows today target the coverage of all possible faults–including those that do not necessarily impact the functioning of the application that runs on the digital system. Such a framework for test leads to yield loss that can be avoidable, if the test vectors used only cover faults that cause the application to violate certain specifications. The increased interest in research community on this problem of test generation for only a subset of all faults is evident from the number of papers that have appeared in the recent years [2], [3], [6], [7], [8], [9], [10], [11], [12], [13]. Research has ranged from proposing criteria that can be used to classify faults as benign (acceptable) and malignant (unacceptable) to efficient test generation algorithms. The contribution of this work is the proposal of a technique underpinned by integer linear programming (ILP) for optimizing test coverage to target faults that only cause functional errors (or violations), while minimizing the coverage of faults that do not cause functional errors. As a case study, we demonstrate error resilient testing of 16-bit ripple carry adder. We begin by generating all possible faults and then classifying them into two heaps: 1) *benign faults* - those that cause no error or an acceptable amount of error, and 2) *malignant faults* - those faults that cause a significant departure from acceptable behavior. The metric used for classifying faults into these two heaps is a simple measure – error magnitude, which is the difference between actual value (with the fault present) and the ideal value. Once the two heaps of faults are generated, we use an automatic test pattern generation (ATPG) tool to generate tests for all malignant faults in the circuit. By fault simulation we create a spreadsheet of all benign faults that are also covered by each of the generated test vectors. Finally we use an ILP to only choose those test vectors that minimize coverage of benign faults while ensuring 100% coverage of all malignant faults.

The paper is organized as follows. Section II formulates the test optimization problem for yield enhancement, constrained by 100% fault coverage of all malignant faults as an ILP. Section III describes this ILP as applied to an 16-bit ripple carry adder. In Section IV we propose a strategy for functional binning of the circuit using selective output masking. We discuss the impact on yield by dropping tests for benign faults in Section V. We conclude in Section VI.

II. PROBLEM FORMULATION

Let us consider a scenario where we have a total of N faults (including both, benign and malignant faults) that are completely covered by M test vectors. We want to find a partitioning of M test vectors that maximizes the coverage of all malignant faults while minimizing the coverage of benign faults. Note that in partitioning the test vector set, we do not try to minimize the overall test set as the objective here is only to increase the fault coverage of malignant faults with zero or little coverage of the benign faults. To achieve such a partitioning of the test vector space, we use an integer linear programming formulation of the problem. Before proceeding to formulate the objective function, we clarify our notation.

η: Objective function of the ILP f_i: denotes the i^{th} fault for all $i = 1 \cdots N$
Λ: set of all malignant faults
Γ: set of all benign faults
T_j: denotes the j^{th} test vector for all $j = 1 \cdots M$
t_j: takes a value 1 if the test vector T_j is included in the test set for all $j = 1 \cdots M$
a_{ij}: takes the value 1 if test vector T_j can detect fault f_i, and takes a value 0 if test vector T_j cannot detect fault f_i
τ_i: an indicator function that takes the value α if fault f_i is malignant, and takes the value $(-\beta)$ if fault f_i is benign
α, β: Optimization parameters whose values lie in the range

[0,1], such that $\alpha + \beta = 1$, and are selected based on what needs to be weighted more – a higher α allows greater effort to maximize coverage of malignant faults while a higher β allows greater effort to minimize coverage of benign faults.

Objective function, η (to be maximized), of the ILP can now be put down as:

$$\eta = \Sigma_{i=1}^{N} l_i \tau_i, \tag{1}$$

with the constraints:

$$\Sigma_{j=1}^{M} (a_{ij} t_j) \geq l_i; \qquad \forall f_i \in \Lambda, \tag{2}$$

$$\Sigma_{j=1}^{M} (a_{ij} t_j) \leq l_i; \qquad \forall f_i \in \Gamma, \tag{3}$$

$$l_i \ \geq \ 0, \tag{4}$$

$$0 \leq t_i, t_j \leq 1. \tag{5}$$

III. Case Study: 16-bit Ripple Carry Adder

In order to evaluate the proposed ILP formulation for test set partitioning into benign and malignant faults, we consider a 16-bit ripple carry adder. Exhaustive functional simulation is carried out with all (collapsed) single stuck-at faults introduced in the circuit. We use "error significance," originally defined in [2], as a metric for establishing whether a particular fault is to be considered benign or malignant. That is, if the result of an addition operation deviates from the fault free value by more than a pre-specified threshold κ, then we consider such a fault malignant, otherwise it is considered benign. For example, in the 16-bit adder, let us say we fix an error threshold value at $\kappa = \pm 5$; then any fault whose addition results in an error value higher than ± 5 for any pair of input vectors is considered a malignant fault. Clearly, with such a definition for "error significance," the union of malignant and benign faults encompasses the entire fault set (with any redundant faults counted as benign faults). In the subsequent sections, we describe fault simulation to find error significance of each single stuck at fault to categorize them into two groups, namely benign faults and malignant faults. We use the proposed integer linear programming formulation to optimize the test set partition for maximizing fault coverage of malignant faults and minimizing the fault coverage of benign faults.

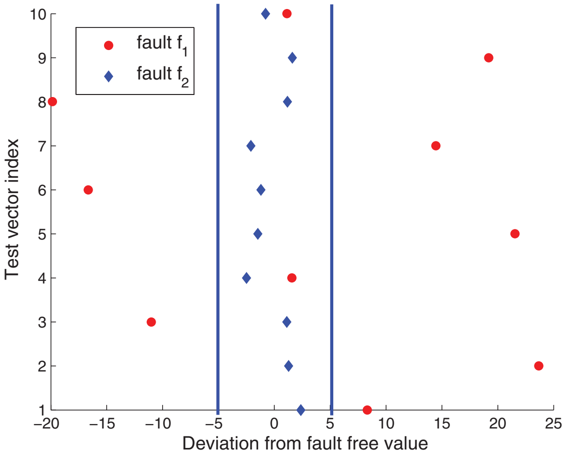

Fig. 1. Scatter plot of errors produced by the 16-bit adder in the presence of two single stuck-at faults f_1 and f_2 for 10 test vectors. Notice that fault f_1 (shown in red marker) results in errors in excess of the threshold $\kappa = \pm 5$ on either side of 0, while the benign (acceptable) fault f_2 (shown in blue marker) results in errors which are within the threshold $\kappa = \pm 5$ about 0 for all test vectors.

A. Fault Simulation for Fault Partitioning

Based on the definition of error significance, we use an error threshold $\kappa = 5$ as a qualification metric to partition faults into benign and malignant. While the threshold κ is chosen dependent on application, our choice here stems from an image processing perspective, where absolute errors of around 5 intensity levels (or less) out of a total of 256 intensity levels produce no perceptual change for the human eye [4]. In our example here we use a 16-bit adder–sum has a word length of 16–which corresponds to 65536 levels, so an absolute error of 5 is well within the limit of perception. There are a total of 432 single stuck at faults (after fault collapsing) in the 16 bit adder. Figure 1 is a scatter plot showing the difference between ideal value and output value with two different stuck at faults f_1 and f_2 for each input test vector. Faults that result in an output value that lies within the two vertical blue lines (in Fig. 1) are benign and those that lie outside this band are malignant faults.

B. Test Vector Partitioning for Optimum Coverage of Malignant Faults

Given that we have a test set that covers all the malignant and benign faults, the ILP formulated earlier, conjures a new test set that maximizes coverage of malignant faults while minimizing coverage of benign faults. Figures 2, 3 and 4 show the pre-optimized and post-optimized fault coverage based on

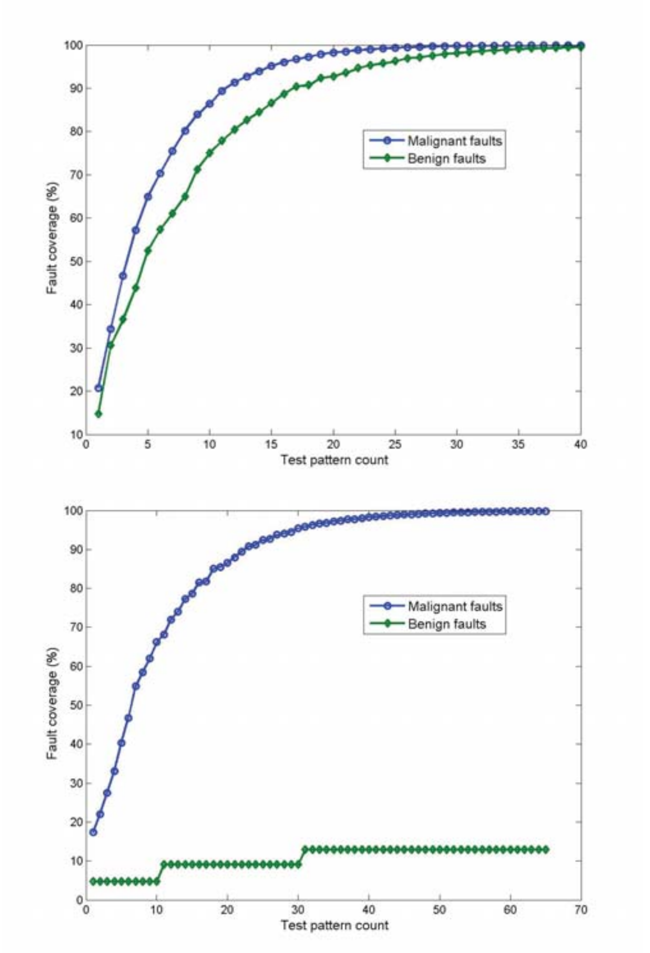

Fig. 2. Fault coverage as a function of test pattern count, before optimization (top) and after optimization (bottom), for 16-bit ripple carry adder.

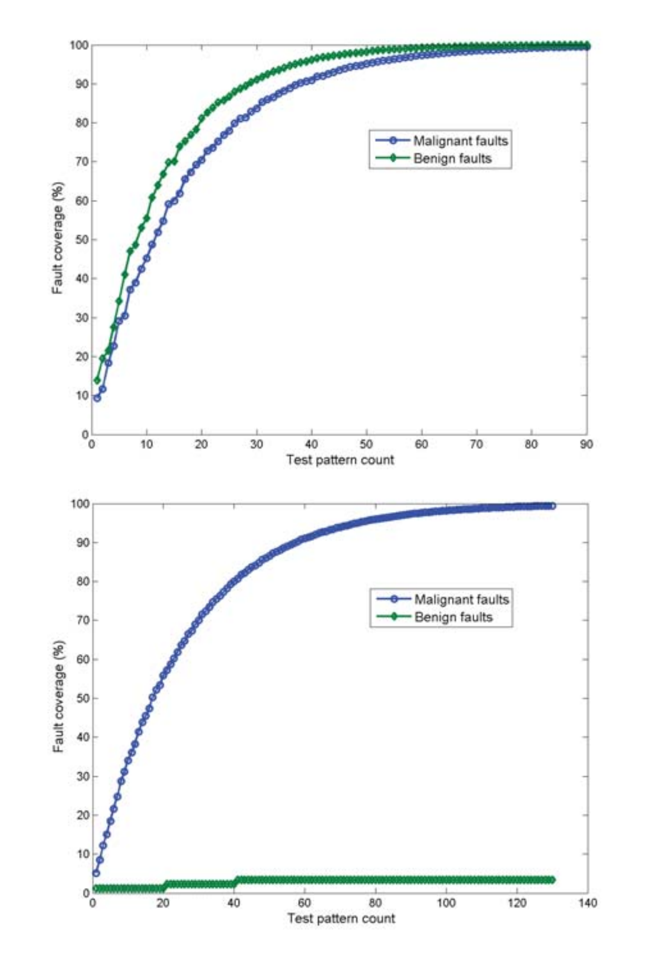

Fig. 3. Fault coverage as a function of test pattern count, before optimization (top) and after optimization (bottom), for 16-bit carry lookahead adder.

the objective function given in Eqn. 1 on a ripple carry adder, carry look ahead adder, and carry save adder respectively. We see an average reduction in the benign fault coverage by about 90%, while the malignant fault coverage is still at 100% with a penalty of test pattern count increase by about 25%.

IV. Selective Output Masking for Functional Binning

The previous section dealt with optimizing the test set for targeting malignant faults while minimizing the test coverage for benign faults. However, for applications where circuits are to be binned, for instance into three categories, namely - malignant, benign, and fault-free, one can can propagate faults to specific outputs and selectively observe them, while ignoring the remaining outputs of the circuit. This can be explained as follows. For an adder, a circuit having a malignant fault, by definition, has at least one input vector whose output deviates from the fault free value by more than some tolerance threshold (κ). So, beginning the test flow by testing the circuit for all malignant faults by using vectors that propagate the fault only to the more significant bit positions (MSB) (precisely bit positions that have value $\geq \kappa$) will weed out any circuits that have malignant faults, and can be binned into malignant category. Note that since all tests are propagate faults only to the more significant bits, the lower significant bits (LSB) (i.e., bit positions that have value $< \kappa$) can be masked or need not be observed during this portion of the test flow. Circuits that pass all the malignant fault tests can then be tested for all the benign faults. Now only the lower significant bits need to be observed. This is because benign faults, by definition, have no input vector that results in an output whose value deviates from the fault free value by more than tolerance threshold (κ). Circuits that fail any of these benign fault tests are binned into benign category. Circuits

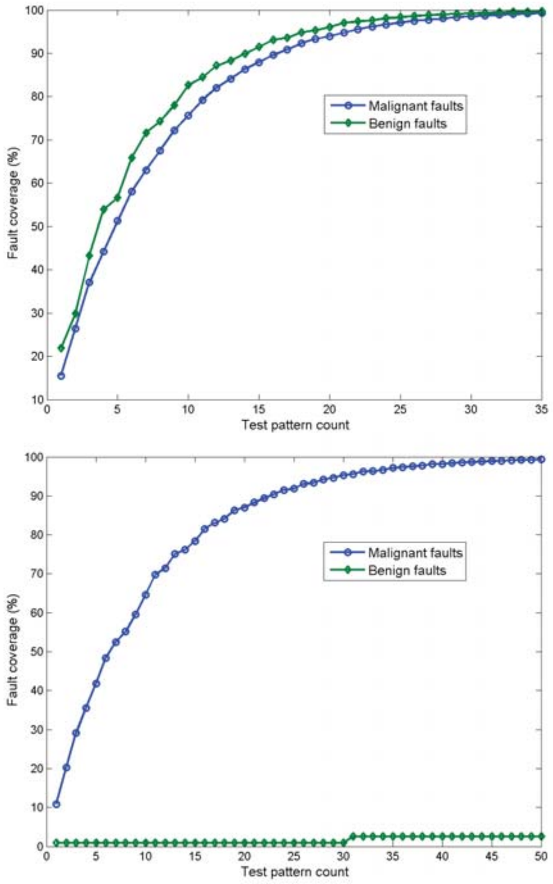

Fig. 4. Fault coverage as a function of test pattern count, before optimization (top) and after optimization (bottom), for 16-bit carry save adder.

that pass all the benign fault tests are binned into the fault-free category. Figure 5 illustrates the test flow for functional binning into three categories. Note that this procedure can be extended to more binning levels by first partitioning the entire fault set into desired number of benignity levels, then selectively propagating faults in each of these benignity levels to specific outputs and masking the rest. The test flow always begins by testing malignant faults (by propagating them to the MSB) and then proceeds towards benign faults (that are propagated to LSB).

V. Impact of Test Set Optimization on Yield

The most notable gain in optimizing test set to target only functionally malignant faults is yield improvement. We now demonstrate with a simple example, the gain in yield by tailoring test set to cover only malignant faults. Let Y, be the yield of a manufactured die in production. The die consists of a circuit that has a total of N possible single stuck-at-faults.

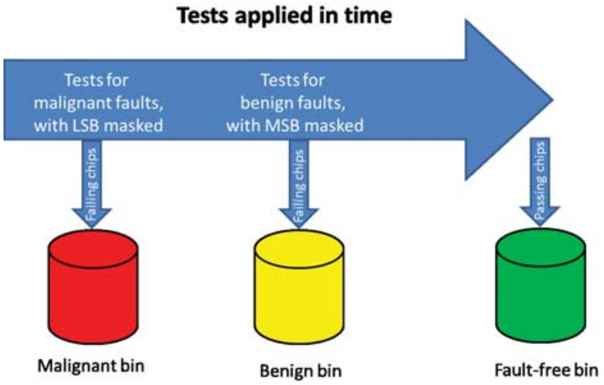

Fig. 5. Test flow for functional binning into three categories, namely malignant, benign, and fault-free by selective masking.

For simplicity, let us assume that each of these N faults can occur independent of others and with the same probability of occurrence p. Then we have the yield Y, which is simply the probability of none of these faults occurring, given by:

$$Y = (1-p)^{N} . \tag{6}$$

Now suppose there are b faults out of N that are functionally benign. If we have a test set that targets only the remaining $(N-b)$ malignant faults, then the modified yield expression becomes:

$$Y_{mod} = (1-p)^{N-b} . \tag{7}$$

To get a numerical feel for this result, let us plug in some hypothetical numbers. Though hypothetical, these numbers still preserve flavor of the problem and its solution. Let the probability of fault occurrence be $p = 0.01$ when the integrated circuit (IC) manufacturing begins. Let $N = 100$ be the total number of single-stuck-at-faults, out of which $b = 25$ are functionally benign. Yield obtained without the test set being cognizant of functionally benign faults is calculated from 6 as $Y = 3.66\%$. If the test set is cognizant of the 25 faults as being functionally benign, then the modified yield $Y_{mod} = 4.7\%$. This results in a yield improvement of $\frac{4.7-3.66}{3.66} \times 100 = 22.12\%$. Now suppose, as the manufacturing flow matures, the probability of fault occurrence drops to $p = 0.005$. Then the traditional yield $Y = 60.58\%$ and modified yield becomes $Y_{mod} = 68.66\%$. Despite the smaller percentage yield improvement only 13.34%, this occurs during the maturity period of the manufacturing process, resulting in greater absolute number of chips being classified as good. This implies that if the number of targeted faults is reduced by about 25%, then the nominal yield improvement can be anywhere

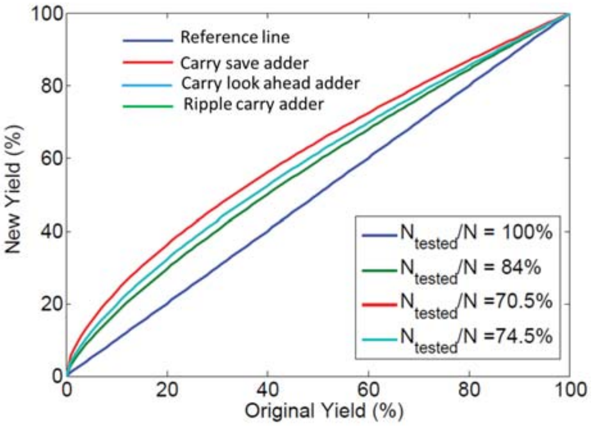

Fig. 6. New yield, obtained by not testing benign faults, is plotted as a function of the original yield. New yield numbers are calculated assuming uniform probability of occurrence of all faults in a circuit.

between 10% and 20%. For the examples we studied earlier, namely, ripple carry adder, carry lookahead adder and carry save adder, as plotted in Fig. 6, the improvements in yield can be as much as 15% – 30% depending on the individual fault probabilities.

VI. Conclusion and Future Work

The paper proposed an ILP formulation for optimizing the test set to increase the fault coverage of malignant faults while minimizing the coverage of functionally benign faults. We used three 16-bit adders, namely, ripple carry, carry lookahead, and carry save adders as examples to demonstrate the proposed optimization scheme. We found that in the presented examples, an average reduction in fault coverage of benign faults was about 90%, with malignant fault coverage still at 100%. The average increase in test pattern count to achieve this is about 30%. We also discussed the yield improvement obtained by using an optimized test set that targets only a subset of faults that are functionally malignant. We found that improvement in yield can be as much as 15% – 30% depending on individual fault probabilities. The area of error resilient testing is still at its infancy and there is sufficient scope for future work along the lines of establishing appropriate error metrics for fault classification for different functional blocks or applications, and test vector generation that is cognizant of benign faults.

References

[1] S. Borkar, "Design Perspectives on 22nm CMOS and Beyond," in *Proc. 46th ACM/IEEE Design Automation Conference*, July 2009, pp. 93–94.

[2] M. A. Breuer, S. K. Gupta, and T. M. Mak, "Defect and Error Tolerance in the Presence of Massive Numbers of Defects," *IEEE Design & Test of Computers*, vol. 21, no. 3, pp. 216–227, May 2004.

[3] M. A. Breuer and H. Zhu, "An Illustrated Methodology for Analysis of Error Tolerance," *IEEE Design & Test of Computers*, vol. 25, no. 2, pp. 168–177, Mar. 2008.

[4] C. H. Chou and Y. C. Li, "A Perceptually Tuned Subband Image Coder Based on the Measure of Just-Noticeable-Distortion Profile," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, no. 6, pp. 467–476, Dec. 1995.

[5] B. Davari, R. H. Dennard, and G. G. Shahidi, "CMOS Scaling for High Performance and Low Power-the Next Ten Years," *Proceedings IEEE*, vol. 83, no. 4, pp. 595–606, Apr. 1995.

[6] T.-Y. Hsieh, K.-J. Lee, and M. A. Breuer, "An Error-Tolerance-Based Test Methodology to Support Product Grading for Yield Enhancement," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 6, pp. 930–934, June 2011.

[7] H. Ichihara, K. Sutoh, Y. Yoshikawa, and T. Inoue, "A Practical Approach to Threshold Test Generation for Error Tolerant Circuits," in *IEEE Asian Test Symposium*, Nov. 2009, pp. 171–176.

[8] T. Inoue, N. Izumi, Y. Yoshikawa, and H. Ichihara, "A Fast Threshold Test Generation Algorithm Based on 5-Valued Logic," in *IEEE International Symposium on Electronic Design, Test and Application*, Jan. 2010, pp. 345–349.

[9] Z. Jiang and S. K. Gupta, "Threshold Testing: Improving Yield for Nanoscale VLSI," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 12, pp. 1883–1895, Dec. 2009.

[10] K. J. Lee, T. Y. Hsieh, and M. A. Breuer, "A Novel Test Methodology Based on Error-Rate to Support Error-Tolerance," in *Proc. International Test Conference*, Nov. 2005, pp. 1–9. Paper 44.1.

[11] K.-J. Lee, T.-Y. Hsieh, and M. A. Breuer, "Efficient Overdetection Elimination of Acceptable Faults for Yield Improvement," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, no. 5, pp. 754–764, May 2012.

[12] Z. Pan and M. A. Breuer, "Estimating Error Rate in Defective Logic Using Signature Analysis," *IEEE Transactions on Computers*, vol. 56, no. 5, pp. 650–661, May 2007.

[13] S. Shahidi and S. K. Gupta, "ERTG: A Test Generator for Error-Rate Testing," in *Proc. International Test Conference*, Oct. 2007, pp. 1–10. Paper 27.1.