Variation-Aware Defect Characterization at Cell Level

Najafi-Haghi, Zahra Paria; Hashemipour-Nazari, Marzieh; Wunderlich, Hans-Joachim

Proceedings of the 25th IEEE European Test Symposium (ETS'20), Tallinn, Estonia, 2020, pp. 1–6

doi: https://doi.org/10.1109/ETS48528.2020.9131600

Abstract: Small Delay Faults (SDFs) are an indicator of reliability threats even if they do not affect the behavior of a system at nominal speed. Various defects may evolve over time into a complete system failure, and defects have to be distinguished from delays due to process variations which also change the circuit timing but are benign. Based on Monte-Carlo electrical simulation at cell level, in this work it is shown that a few measurements at different operating points of voltage and frequency are sufficient to identify a defect cell even if its behavior is completely within the specification range. The developed classifier is based on statistical learning and can be annotated to each element of a cell library to support manufacturing test, diagnosis and optimizing the burn-in process or yield.

Preprint

Variation-Aware Defect Characterization at Cell Level

Zahra Paria Najafi-Haghi, Marzieh Hashemipour-Nazari, Hans-Joachim Wunderlich
Institute of Computer Architecture and Computer Engineering, University of Stuttgart, Germany
Email: {najafi-haghi, hashemsh, wu}@informatik.uni-stuttgart.de

***Abstract*—Small Delay Faults (SDFs) are an indicator of reliability threats even if they do not affect the behavior of a system at nominal speed. Various defects may evolve over time into a complete system failure, and defects have to be distinguished from delays due to process variations which also change the circuit timing but are benign.**

Based on Monte-Carlo electrical simulation at cell level, in this work it is shown that a few measurements at different operating points of voltage and frequency are sufficient to identify a defect cell even if its behavior is completely within the specification range.

The developed classifier is based on statistical learning and can be annotated to each element of a cell library to support manufacturing test, diagnosis and optimizing the burn-in process or yield.

***Index Terms*—Small delay faults, variations, reliability, defect modeling, statistical learning.**

I. Introduction

In a recent paper, significant progress has been reported to diagnose Small Delay Faults (SDFs) in combinational circuits [1]. It has been left as an open problem to decide whether the increase of the delay of a certain gate is due to a defect or due to process variations. A solution of this problem is especially urgent as today's complex circuits in FinFET technology are often able to overcome timing variations by voltage and frequency scaling which may also hide certain defects.

The failures due to defects can become catastrophic in the field, like gate oxide breakdown, hot carrier effects and electromigration [2] [3] [4] [5]. Although traditional delay test methods have primarily focused on gross delays, there is growing evidence that SDFs should be considered to detect marginal defects [4] [6] [7] [8].

Many efforts have been taken to consider process variations in the test process [9] [10] [11] [12]. Increasing random process variations can contribute to significant timing variability which is often indistinguishable from the effect of defects [13]. However, the former stays the same during the lifetime of a circuit and can be overcome by adapting voltage and frequency, but the latter would degrade further in the field and is a threat to reliability.

Fig. 1 shows the outcome of a Monte-Carlo experiment based on the SPICE [14] simulation of a NAND gate from the FinFET Open Cell Library (OCL) [15] in 15nm technology with a standard deviation of $\sigma = 0.1$ on channel width and length. The green bars show the timing distribution of the defect-free cell instances, while the red bars show the distribution after injecting a defect at the source of a nFET transistor in the cell. In this experiment, 1000 defect-free and 1000 defect instances of the NAND gate were simulated and the striped blue part denotes instances with ambiguous delays. Most of the defect instances show a timing at nominal voltage which could also be produced by a defect-free instance.

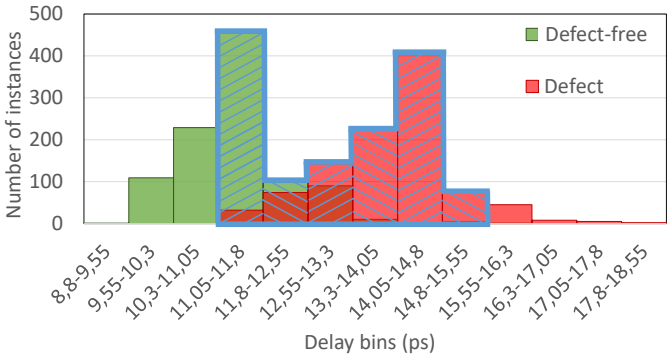

Fig. 1. Simulated delay histograms for defect-free and defect NAND cell

Depending on the process parameters values, cells can get slower or faster than their nominal timing, which are called *slow* or *fast* instances of the cell. Although the process variation in the specified range is acceptable [16], a fast cell with an injected defect (fast-defect) gets slower, and it is possible that it would be still faster than the slow defect-free cell. Fig. 2 shows the simulated delay for three slow defect-free instances of the NAND cell, as well as for a fast-defect instance of the same cell. By considering only one delay measurement at nominal voltage, it is not possible to distinguish the defect instance from defect-free ones, since the timing of a fast-defect cell is within the defect-free timing range of the cell. Even worse, for each specified voltage, there may be a defect-free instance which is slower than the defect-fast one.

Distinguishing defect and defect-free cells obviously requires more sophisticated analysis techniques, than just comparing the timing point by point.

The rest of the paper is organized as follows. The next section gives some orientation of the state of the art for defect detection under variations. Section III describes the simulation model to generate the data set for creating classifiers based on statistical learning. Section IV presents how classifiers are

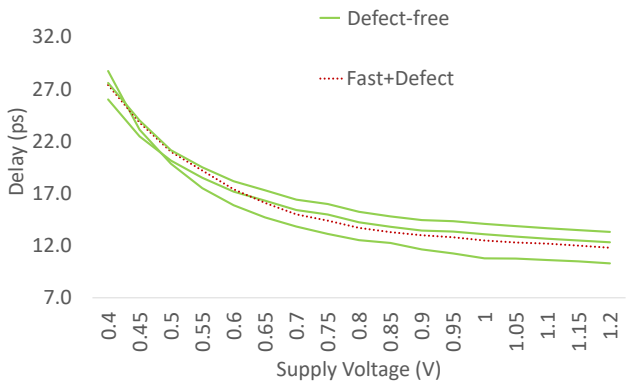

Fig. 2. Simulated delay under varying supply voltage for three defect-free instances with different process parameters and a fast-defect instance of a NAND cell

trained and validated. Section V evaluates the efficiency of the three most appropriate learning techniques. Remarks on further work and applications conclude the paper in Section VI.

II. State of the Art

Resistive bridges, resistive opens and gate-oxide pinholes are considered as defect mechanisms which may cause Early Life Failures (ELFs) and appear as delay faults [17] [18] [19] [20] [21] [22] [23] [24] [7].

The impact of operating conditions, in particular voltage, on delay faults has been investigated thoroughly. Reducing the supply voltage increases the transistor channel resistances in the circuit, thereby reducing the relative value and increasing the electrical impact of a gate drain (or gate source) resistive short defect [21].

The articles [25] and [26] showed the significant effect of operating conditions on electrical performance. [27] and [28] proposed Very-Low-Voltage testing to detect defects resulting in Early Life Failure (ELF). However, for newer technologies Very-Low-Voltage testing does not necessarily give the best fault coverage [29]. The impact of Dynamic Voltage Scaling (DVS) on the quality of manufacturing tests to detect permanent delay faults is also investigated in [19].

On the other hand, it is observed that resistive defects and non-resistive defects such as a slow transistor, behave quite differently as the supply voltage is varied. In [23] authors discussed the delay behavior associated with classic gross resistive defects and compares it with transistor variation due to lithography. In [20], [13], [21] and [17], methods to distinguish process variation from resistive defects have been provided, but the defect model in all these papers adds some fixed amount of delay to the nominal delay without considering the process variation impact also at the same defect part.

The delays introduced by process variations depend on the operation voltage [30], and the voltage impact on the joint marginal defect and process variations has to be considered. All the mentioned works are on conventional planar technology instead of the recently emerging FinFET technology [31].

In the paper at hand, the delays of defect-free and defect cells in the standard FinFET library are observed not only under nominal supply voltage, but also under other voltages in a specific range. Accordingly, varying voltage as a controllable parameter is being used actively for test purpose and provides the parametric behavior of defect-free and defect cells for training Machine Learning (ML) schemes. As far as we know, this is the first time that statistical learning algorithms are being used for marginal defect classification.

III. Cell Modeling

A. Overview

The overall flow of the cell characterization strategy is shown in Fig. 3. It consists of 2 phases for **library preprocessing** and one phase for **cell classification** based on the test outcome.

Simulation data is generated for the class of defect-free and the class of defect cells, while the latter class can be subdivided according to the defect-types injected.

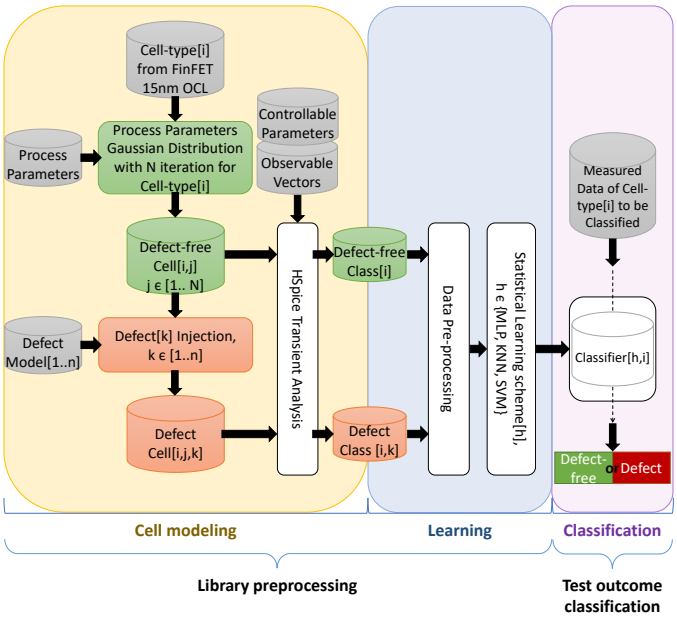

Fig. 3. Classification flow

B. Modeling defect-free cells

Cells have various performance characteristics due to process variations. Threshold voltage (V_{th}) fluctuations due to Random Dopant Fluctuations (RDF) and Line Edge Roughness (LER) are considered often as the major source of process variations [18] [13]. However, since FinFET has very lightly doped channels, no significant random dopant fluctuations may happen [32]. Therefore, in this work variations in device geometry, in particular gate length (L) and gate width (W) are considered. Each of the process parameters follows Gaussian distribution with 10% standard deviation ($\sigma = 0.1$) for V_{th}. A standard deviation of $\sigma = 0.1$ on V_{th} was also guided by experience with real processes in FinFET transistors [16]. Monte Carlo SPICE simulation is performed with many iterations to

model defect-free instances of each cell. An instance is a cell with specific process parameter values. N shows the number of Monte Carlo iterations and subsequently the number of the defect-free instances.

C. Modeling defect cells

For each cell-type in the library, various defects can be injected based on the defect mechanisms and fault locations. Since this paper reports on a prototype study we dispense with layout based defect injection as originally proposed as inductive fault analysis [33] and later commercialized as *cell-aware test* [34]. Instead, the most relevant defects are injected at electrical level as seen in the NAND example of Fig. 4

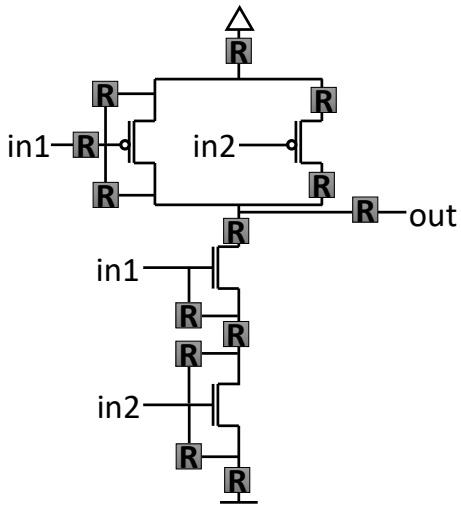

Fig. 4. 2-input NAND cell and investigated defect-types

In this example, only one defect from the sets of symmetric defects (e.g. in1 and in2) is shown. A complete characterization has to cover more locations and defect sites as it is done in cell-aware testing. In general, n denotes the number of possible *defect-types* for each cell-type. A single defect corresponding to each of the n defect-types is injected into a whole population of N instances for each cell-type, which guarantees considering process variations also in defect models. As a result, $n \cdot N$ single faults are injected. The size of the faults is set in a way to produce a small amount of extra delay which is comparable with a slow instance. This assures the challenging overlap between defect-free cells and defect ones and provides a difficult case for defects to be classified. In addition this keeps the fault size smaller than 3σ which is within the size of small delay faults [8].

D. Data generation

Electrical level SPICE simulation is performed on the defect-free cells and defect ones to observe the delay of each instance under varying supply voltage. The supply voltage range has ω different values and is given to the SPICE transient analysis. Simulation results represent the parametric behavior for each instance. Populations of defect-free and defect parametric behaviors make the defect-free class and defect one, where the defect-free class has N members and defect class has $n \cdot N$ ones. Each member of these classes has ω simulated delay values, where t_i, $i = [1..\omega]$ are the simulated delays at the supply voltage V_i.

IV. Classification by Machine Learning

A. Learning schemes

For each defect or defect-free cell instance, an array of ω elements is generated which contains the cell delay for each operation voltage, as the result of the electrical simulation. Based on this data, supervised learning is performed with three different statistical learning schemes: Multi-Layer Perceptron (MLP) [35], k-Nearest Neighbors (kNN) [36] and Support Vector Machines (SVM) [37]. These schemes have been selected to minimize the requirements for memory and computation time, since the resulting classifier has to be mapped to the cell library elements.

MLP: The Multi-Layer Perceptron classifier is a feed-forward artificial neural network [38]. The MLP classifier used here consists of three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses an activation function which maps the weighted inputs to the output of each neuron. Based on the comparison of the actual output to the expected one, connection weights after each data process are changed and learning occurs in the perceptron. The *adam* [39] and *relu* [40] are used as solver function for weight optimization and activation function for the hidden layer respectively.

kNN: The k-Nearest Neighbors algorithm assigns a data point to the class to which the majority of the k-nearest data points belong.

The distance between two instances is computed by the Euclidean metric, where the t_i is as discribed above.

$$Distance = \sqrt{\sum_{i=1..\omega} (|t_i^{test} - t_i^{train}|)^2}$$

SVM: Support Vector Machines construct a hyper-plane to separate data points with the largest amount of margin, which means the largest distance to the nearest training data points. The Support Vector Classifier (SVC) [41] with *rbf* [42] kernel is used in this approach.

For each cell, the data set comprises $(n+1) \cdot N$ arrays of length ω, one for the defect-free case and n for the defects times the sample size N. The goal of classification is defect detection but not defect diagnosis. Hence, it is sufficient to output just the defect detect information. However, internally, n classifiers Cl_i, $i = 1, ..n$, are trained, one for each defect-type. If just one of the Cl_is classifies the corresponding defect, a defect cell is announced.

B. Evaluation of the learning schemes

There are $(n+1) \cdot N$ instances available for each cell-type, to be used either for training or for test. These instances are partitioned randomly in 10 sets for cross-validation [43].

One of the sets is used as the test set while the union of the nine other sets provide the training data. This forms 10 experiments, and the quality metrics *Recall*, *6*Precision and *F1-score* are computed based on the confusion matrix for each set and reported also as the average from all experiments [44].

Let $TruePositive(TP)$, $TrueNegative(TN)$, $FalsePositive(FP)$, and $FalseNegative(FN)$ be the numbers of instances "correctly classified as defect", "correctly classified as defect-free", "incorrectly classified as defect", and "incorrectly classified as defect-free" respectively.

The quality metrics for the **defect** cells are defined in Eq. 1 and Eq. 2.

$$Recall := \frac{TP}{TP + FN} \tag{1}$$

$$Precision := \frac{TP}{TP + FP} \tag{2}$$

and for the **defect-free** cell we have in an analog way:

$$Recall := \frac{TN}{TN + FP} \tag{3}$$

$$Precision := \frac{TN}{TN + FN} \tag{4}$$

In both cases the $F1 - score$ is defined as:

$$F1 - score := \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}} \tag{5}$$

"Recall" denotes the ratio of the instances in the related database which are classified correctly. For instance, for defect cells as presented by Eq. 1, recall shows how much of the defect instances in the database are detected. In consequence, it shows the rate of defect escape. Secondly, "precision" represents the proportion of the classified instances which actually belong to that class. For example, for defect cells as demonstrated by Eq. 2, precision represents how much of the detected instances are actually defect. In contrast, it indicates the rate of overdetection. These numbers will be reported in the next section for the complete data sets. However, this validation has two major drawbacks. It does not consider how difficult the classification is. If the two histograms in Fig. 1 have only a small overlap area, any ML approach will have high quality. The same is true if the histograms of Fig. 1 are separated over some range. For this reason, a more challenging validation has been performed as well.

From all the generated instances, only those are selected for the test set which overlap at least for some voltages. This means the defect cell is faster than some defect-free cells.

This method is called *overlap-based validation* in this work. Fig. 5 describes how training data and test data are selected from the defect-free and defect instances in this method.

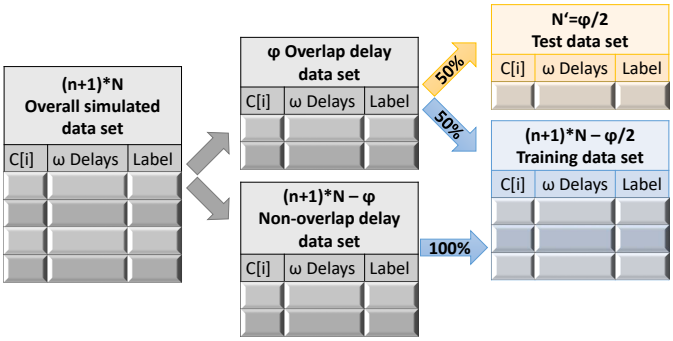

Fig. 5. Selecting training data and test data for the overlap-based validation method

The overlapping instances can be chosen so that defect and defect-free cases are balanced, which has the additional advantage that a single number metric can be applied as well.

$$Accuracy := \frac{TP + TN}{N'} \tag{6}$$

Here N' is the size of the overlap test data. "Accuracy" denotes how much of the test data are generally classified correctly.

V. Simulation Results

In the following, the NanGate 15nm OCL [15] is investigated for modeling, characterization and simulation. Experiments on the 2-input NAND gate of Fig. 4 are demonstrated as a case study.

A. Cell modeling

Monte Carlo simulation is performed with $N = 1000$ iterations to simulate the Gaussian distribution of gate length (L) and gate width (W). It models 1000 defect-free instances. Fig. 4 shows $n = 13$ defect-types which are investigated to model the defect cells. One single defect is injected in each fault simulation and 13000 defect instances are modeled after all fault simulations.

The defect-free class and defect class are built by applying SPICE transient analysis on 1000 defect-free and 13000 defect instances. Supply voltage as controllable parameter gets the values between 0.3V and 1.2V with the step of 0.05V ($\omega = 19$). The simulated delay is observed and stored. Overall, the data set is represented by a 14000 x 19 matrix.

B. Classification by Machine Learning

The generated data is inserted into 3 different ML algorithms: MPL, kNN and SVM. Each entry shows the parametric behavior of the instances of the defect-free or defect classes. Each of these classifiers is evaluated based on the 10-set cross validation as well as on three iterations for the more challenging overlap-based validation. TP, TN, FN, and FP are extracted for each set and classifier using Python [45]. The minimum, average and maximum of the *Recall*, *Precision*, and

TABLE I
SIMULATION RESULTS FOR QUALITY METRICS USING 10-SET CROSS VALIDATION

	kNN						SVM						MLP					
	Defect			Defect-free			Defect			Defect-free			Defect			Defect-free		
	min	avg	max	min	avg	max	min	avg	max	min	avg	max	min	avg	max	min	avg	max
Precision	1.0	1.0	1.0	0.98	0.99	0.99	0.89	0.91	0.92	0.95	0.91	0.91	0.92	0.92	0.92	0.97	0.97	0.98
Recall	0.98	0.99	0.99	1.0	1.0	1.0	0.97	0.97	0.98	0.89	0.90	0.92	0.97	0.98	0.98	0.92	0.93	0.95
	min			avg			max			min			avg			max		
F1-score	0.99			0.99			1.0			0.93			0.94			0.95		
	min			avg			max			min			avg			max		
	0.95			0.95			0.97			0.95			0.95			0.97		

TABLE II
SIMULATION RESULTS FOR QUALITY METRICS USING OVERLAP-BASED VALIDATION

	kNN						SVM						MLP					
	Defect			Defect-free			Defect			Defect-free			Defect			Defect-free		
	it.#1	it.#2	it.#3	it.#1	it.#2	it.#3	it.#1	it.#2	it.#3	it.#1	it.#2	it.#3	it.#1	it.#2	it.#3	it.#1	it.#2	it.#3
Precision	1.0	1.0	0.99	0.91	0.91	0.91	0.85	0.87	0.83	0.68	0.67	0.71	0.88	0.87	0.78	0.75	0.81	0.84
Recall	0.91	0.91	0.90	1.0	1.0	0.99	0.6	0.59	0.66	0.89	0.91	0.86	0.71	0.82	0.89	0.9	0.87	0.87
	it.#1			it.#2			it.#3			it.#1			it.#2			it.#3		
F1-score	0.95			0.95			0.95			0.74			0.74			0.76		
Accuracy	0.95			0.95			0.95			0.74			0.74			0.76		
	it.#1			it.#2			it.#3			it.#1			it.#2			it.#3		
	0.80			0.84			0.78			0.80			0.84			0.80		

F1-score metrics from Eq. 1 to Eq. 5 are presented in Table. I for 10-set cross validation.

Cross validation on the entire data set shows for all the three classifiers high values for precision, recall and the F1-score. However, the kNN classification uniformly outperforms the two other techniques by values close to 1.0. Only the average recall of 0.99 for defect cells indicates that a very small number of defect instances is overlooked. The precision value of 1.0 for the defect case shows there will be no yield loss of the kNN classification.

The results of the more challenging overlap-based validation is shown in Table. II. Here, only defect instances are considered as test data, which have a delay similar to defect-free cells under variations. This pre-selection of hard test cases prevents the use of 10-set cross validation, instead three random iterations are used.

Also for these more challenging experiments, kNN outperforms both SVM and MLP, but recall for defect cells and precision for defect-free cells are now just above 0.9. Still we have only 10% of defect escape, which match the timing of certain defect-free cells for some voltage values and no yield loss. All together, in this classification an accuracy of 0.95 is reached, which shows only 5% of the whole challenging test set including defect and defect-free instances are classified incorrectly.

The superiority of kNN is also underlined by the implementation cost of the classifier in terms of memory. For the NAND cell, 24985.6 Bytes have to be stored, while SVM need 228966.4 Bytes and MLP 1093222.4 Bytes. Memory usage is relevant since these data will be stored and added to the cell library. In addition, for all schemes the computation time is in the order of seconds.

VI. CONCLUSION AND FURTHER WORK

Cells with marginal defects may behave like cells under process variations, but they can form a reliability risk. A method for statistical learning is developed based on electrical simulation data which can classifiy defect and defect-free cells with high accuracy. The method uses kNN, and can classify with very moderate time and memory requirements.

The results obtained so far are extremely encouraging for work towards automizing the classification. Further steps are on the extraction of defects causing SDFs from the layout like in cell-aware test, and the minimization of observation points (delay, voltage) to allow the classification based on a low number of frequencies and measurements. For the analysis of cells in an entire circuit, the propagation along paths and the reduction of the slack by Faster-than-At-Speed Test (FAST) techniques [3] will be investigated further.

ACKNOWLEDGEMENTS

This work is part of the project grant WU 245/19-1 funded by the German Research Foundation (DFG).

REFERENCES

[1] S. Holst, E. Schneider, M. A. Kochte, X. Wen, and H. Wunderlich, "Variation Aware Small Delay Fault Diagnosis on Compressed Test Responses," in *Proc. IEEE Int'l Test Conf. (ITC)*, Oct. 2019.

[2] Y. M. Kim, Y. Kameda, H. Kim, M. Mizuno, and S. Mitra, "Low-Cost Gate-Oxide Early-Life Failure Detection in Robust Systems," in *Proc. IEEE VLSI Circuits Symp. (VLSIC)*, Jun. 2010, pp. 125–126.

[3] M. Kampmann, M. A. Kochte, C. Liu, E. Schneider, S. Hellebrand, and H. Wunderlich, "Built-In Test for Hidden Delay Faults," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Syst. (TCAD)*, vol. 38, no. 10, pp. 1956–1968, Oct 2019.

[4] P. Nigh, "The Increasing Importance of On-line Testing to Ensure High-Reliability Products," in *Proc. IEEE Int'l Test Conf. (ITC)*, Sep. 2003, pp. 1281–1281.

[5] M. H. Woods, "MOS VLSI Reliability and Yield Trends," *Proceedings of the IEEE*, vol. 74, no. 12, pp. 1715–1729, Dec. 1986.

[6] S. Hellebrand, T. Indlekofer, M. Kampmann, M. A. Kochte, C. Liu, and H. Wunderlich, "FAST-BIST: Faster-than-at-Speed BIST Targeting Hidden Delay Defects," in *Proc. IEEE Int'l Test Conf. (ITC)*, Oct. 2014, pp. 1–8.

[7] F. Hapke, J. Schloeffel, W. Redemund, A. Glowatz, J. Rajski, M. Reese, J. Rearick, and J. Rivers, "Cell-Aware Analysis for Small-Delay Effects and Production Test Results from Different Fault Models," in *Proc. IEEE Int'l Test Conf. (ITC)*, Sep. 2011, pp. 1–8.

[8] C. Liu, E. Schneider, M. Kampmann, S. Hellebrand, and H. Wunderlich, "Extending Aging Monitors for Early Life and Wear-Out Failure Prevention," in *Proc. IEEE Asian Test Symp. (ATS)*, Oct. 2018, pp. 92–97.

[9] F. Hopsch, B. Becker, S. Hellebrand, I. Polian, B. Straube, W. Vermeiren, and H. Wunderlich, "Variation-Aware Fault Modeling," in *Proc. IEEE Asian Test Symp. (ATS)*, Dec. 2010, pp. 87–93.

[10] A. Czutro, M. E. Imhof, J. Jiang, A. Mumtaz, M. Sauer, B. Becker, I. Polian, and H. Wunderlich, "Variation-Aware Fault Grading," in *Proc. IEEE Asian Test Symp. (ATS)*, Nov. 2012, pp. 344–349.

[11] I. Polian, B. Becker, S. Hellebrand, H. Wunderlich, and P. Maxwell, "Towards Variation-Aware Test Methods," in *Proc. IEEE European Test Symp. (ETS)*, May 2011, pp. 219–225.

[12] T. M. Mak and S. Nassif, "Guest Editors' Introduction: Process Variation and Stochastic Design and Test," *IEEE Design and Test*, vol. 23, no. 6, pp. 436–437, Jun. 2006.

[13] X. Qian and A. D. Singh, "Distinguishing Resistive Small Delay Defects from Random Parameter Variations," in *Proc. IEEE Asian Test Symp. (ATS)*, Dec. 2010, pp. 325–330.

[14] L. W. Nagel and D. O. Pederson, "SPICE (Simulation Program with Integrated Circuit Emphasis)," EECS Department, University of California, Berkeley, Tech. Rep. UCB/ERL M382, Apr. 1973.

[15] NanGate Inc. (2017) NanGate15nm Open Cell Library. [Online]. Available: http://www.nangate.com/

[16] V. P. Yanambaka, S. P. Mohanty, E. Kougianos, D. Ghai, and G. Ghai, "Process Variation Analysis and Optimization of a FinFET-Based VCO," *IEEE Trans. on Semiconductor Manufacturing (TSM)*, vol. 30, no. 2, pp. 126–134, May 2017.

[17] A. Karel, F. Azais, M. Comte, J. Galliere, M. Renovell, and K. Singh, "Detection of Resistive Open and Short Defects in FDSOI under Delay-Based Test: Optimal VDD and Body Biasing Conditions," in *Proc. IEEE European Test Symp. (ETS)*, May 2017, pp. 1–2.

[18] F. Forero, J. Galliere, M. Renovell, and V. Champac, "Analysis of Short Defects in FinFET Based Logic Cells," in *Proc. IEEE Latin American Test Symp. (LATS)*, Mar. 2017, pp. 1–6.

[19] N. B. Zain Ali, M. Zwolinski, B. M. Al-Hashimi, and P. Harrod, "Dynamic Voltage Scaling Aware Delay Fault Testing," in *Proc. IEEE European Test Symp. (ETS)*, May 2006, pp. 15–20.

[20] Y. Haihua and A. D. Singh, "A Delay Test to Differentiate Resistive Interconnect Faults from Weak Transistor Defects," in *Proc. Int'l Conf. of VLSI Design (VLSID)*, Jan. 2005, pp. 47–52.

[21] X. Qian, Chao Han, and A. D. Singh, "Detection of Gate-Oxide Defects with Timing Tests at Reduced Power Supply," in *Proc. IEEE VLSI Test Symp. (VTS)*, Apr. 2012, pp. 120–126.

[22] Y. Liu and Q. Xu, "On Modeling Faults in FinFET Logic Circuits," in *Proc. IEEE Int'l Test Conf. (ITC)*, Nov 2012, pp. 1–9.

[23] R. C. Aitken, "Defect or Variation? Characterizing Standard Cell Behavior at 90 nm and Below," *IEEE Trans. on Semiconductor Manufacturing (TSM)*, vol. 21, no. 1, pp. 46–54, Feb. 2008.

[24] S. Khursheed, S. Zhong, R. Aitken, B. M. Al-Hashimi, and S. Kundu, "Modeling the Impact of Process Variation on Resistive Bridge Defects," in *Proc. IEEE Int'l Test Conf. (ITC)*, Nov 2010, pp. 1–10.

[25] K. D. Wagner and E. J. McCluskey, "Effect of Supply Voltage on Circuit Propagation Delay and Test Applications," in *Proc. IEEE Int'l Conf. on Computer-Aided Design (ICCAD)*, Nov. 1985, pp. 42–44.

[26] S. Nassif, "Delay Variability: Sources, Impacts and Trends," in *Proc. IEEE Int'l Solid-State Circuits Conf. (ISSCC)*, Feb. 2000, pp. 368–369.

[27] H. Hao and E. J. McCluskey, "Very-Low-Voltage Testing for Weak CMOS Logic ICs," in *Proc. IEEE Int'l Test Conf. (ITC)*, Oct. 1993, pp. 275–284.

[28] J. T. Chang and E. J. McCluskey, "Detecting Delay Flaws by Very-Low-Voltage Testing," in *Proc. IEEE Int'l Test Conf. (ITC)*, Oct. 1996, pp. 367–376.

[29] P. Engelke, I. Polian, M. Renovell, B. Seshadri, and B. Becker, "The Pros and Cons of Very-Low-Voltage Testing: An Analysis Based on Resistive Bridging Faults," in *Proc. IEEE VLSI Test Symp. (VTS)*, Apr. 2004, pp. 171–178.

[30] M. Alioto, G. Palumbo, and M. Pennisi, "Understanding the Effect of Process Variations on the Delay of Static and Domino Logic," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems (TVLSI)*, vol. 18, no. 5, pp. 697–710, May 2010.

[31] ITRS Inc. (2013) Int'l Tech. Roadmap for Semiconductors. [Online]. Available: http://www.itrs.net/reports

[32] D. D. Lu, M. V. Dunga, C. Lin, A. M. Niknejad, and C. Hu, "A Multi-Gate MOSFET Compact Model Featuring Independent-Gate Operation," in *Proc. IEEE Int'l Electron Devices Meeting (EDM)*, Dec 2007, pp. 565–568.

[33] J. P. Shen, W. Maly, and F. J. Ferguson, "Inductive Fault Analysis of MOS Integrated Circuits," *IEEE Design and Test*, vol. 2, no. 6, pp. 13–26, Dec. 1985.

[34] P. Maxwell, F. Hapke, and H. Tang, "Cell-Aware Diagnosis: Defective Inmates Exposed in their Cells," in *Proc. IEEE Int'l European Test Symp. (ETS)*, May 2016, pp. 1–6.

[35] S. K. Pal and S. Mitra, "Multilayer Perceptron, Fuzzy Sets, and Classification," *IEEE Trans. on Neural Networks*, vol. 3, no. 5, pp. 683–697, Sep. 1992.

[36] T. Cover and P. Hart, "Nearest Neighbor Pattern Classification," *IEEE Trans. on Information Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967.

[37] C. Cortes and V. Vapnik, "Support-Vector Networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[38] H. K. Kwan, "Systolic architectures for Hopfield network, BAM and Multi-layer Feed-forward Network," in *Proc. IEEE Int'l Symp. on Circuits and Systems (ISCAS)*, May 1989, pp. 790–793.

[39] D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proc. Int'l Conf. on Learning Representations (ICLR)*, Dec. 2014.

[40] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," in *Proc. Int'l Conf. on Machine Learning (ICML)*, Jun. 2010, pp. 807–814.

[41] M. Schmidt and H. Gish, "Speaker identification via support vector classifiers," in *IEEE Int'l Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, May 1996, pp. 105–108.

[42] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2001.

[43] R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," in *Proc. Int'l Joint Conf. on Artificial Intelligence (IJCAI)*, Aug. 1995, pp. 1137–1143.

[44] S. Marsland, *Machine Learning: An Algorithmic Perspective*, 2nd ed. Chapman and Hall/CRC, 2014.

[45] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Machine Learning Research*, vol. 12, pp. 2825–2830, Nov. 2011.