Massive Statistical Process Variations

A Grand Challenge for Testing Nanoelectronic Circuits

B. Becker, University of Freiburg, Germany
S. Hellebrand, University of Paderborn, Germany
I. Polian, University of Passau, Germany
B. Straube, W. Vermeiren, Fraunhofer IIS/EAS Dresden, Germany
H.-J. Wunderlich, University of Stuttgart, Germany

***Abstract*—Increasing parameter variations, high defect densities and a growing susceptibility to external noise in nanoscale technologies have led to a paradigm shift in design. Classical design strategies based on worst-case or average assumptions have been replaced by statistical design, and new robust and variation tolerant architectures have been developed. At the same time testing has become extremely challenging, as parameter variations may lead to an unacceptable behavior or change the impact of defects. Furthermore, for robust designs a precise quality assessment is required particularly showing the remaining robustness in the presence of manufacturing defects. The paper pinpoints the key challenges for testing nanoelectronic circuits in more detail, covering the range of variation-aware fault modeling via methods for statiscal testing and their algorithmic foundations to robustness analysis and quality binning.**

I. INTRODUCTION

Nanoscale integration comes along with high defect densities, increasing parameter variations and a growing susceptibility to external noise [2][4]. This has led to a paradigm shift in design methods towards what is known as statistical design [21][24], as well as to the development of novel, robust and variation-tolerant architectures [7][16][19][22][27]. While yield improvement and online fault tolerance were considered as different tasks in the past, now a robust design has to compensate manufacturing defects and parameter variations as well as transient errors during system operation.

In this scenario testing has become particularly challenging. Classical test approaches rely on a clear distinction between the "fault free" and the "faulty" case, but parameter variations can change the impact of a defect, and also the parameter variations themselves can lead to an unacceptable behavior [1]. First approaches to address specific aspects of this problem have been published [14][29][31], and in particular it has been shown that the notion of "fault coverage" is no longer meaningful. Instead a test must be able to screen out defects for a maximum number of parameter combinations, which is reflected by new test quality metrics such as "test robustness" or "process coverage" [11][28]. Furthermore, a "go"/"no go" decision is not enough. If built-in robustness mechanisms are necessary to compensate manufacturing defects, then a more precise assessment of the residual robustness against transient errors during system operation ("quality binning") is required.

This work has been performed by within the RealTest project sponsored by the German National Science Foundation (DFG).

This special session highlights the new challenges for testing in more detail and presents first solutions for variation-aware fault modeling, statistical testing, ATPG in statistical testing as well as appropriate concepts for robustness analysis and quality binning. The following sections summarize the key ideas of the respective presentations.

II. VARIATION-AWARE FAULT MODELING

Defect-oriented testing tries to overcome the deficiencies of classical fault models by extracting the behavior of library cells in the presence of defects and using this input for automatic test pattern generation (ATPG) at gate level. In nanoscale circuits, however, parameter variations can change the behavior of both defective and defect free cells. Moreover, the interaction of a defective cell with its surrounding cells and interconnects may not allow unambiguous decisions whether a given defect is "critical" and should be targeted during test generation. To bridge the gap between low level defect information and the statistical analysis on higher levels of abstraction, the concepts of defect-oriented testing must be combined with statistical library characterization.

Primitive library characterization under these assumptions is a very challenging task because of the huge space of relevant parameters. To reduce the computational complexity Monte Carlo Simulation at the electrical level is proposed. The impact of a defect is modeled as a fault $f \in F$ to be injected into the extracted transistor netlist simultaneously with a parameter configuration $(p_1, p_2, \ldots, p_N) \in P$. The resulting circuit is then simulated at electrical level. A defect can affect the delay of a library cell or lead to a static fault, where a stuck-at fault can be viewed as an infinite delay. As a result of this Monte Carlo process for each cell a delay distribution is obtained for each defect and also for the defect-free case.

To provide a suitable interface for fault simulation and test generation tools at higher levels, the delay distributions must be represented in compact and easy to process format, such as for example histograms for a limited number of discrete delay values. Overall, the database generation is computationally expensive, but it has to be performed only once as a pre-processing step for library characterization.

First experimental results have been obtained using a modified version of the proprietary Fraunhofer IIS/EAS analogue fault simulator aFSIM [25] and considering the channel length L, the length reduction parameter L_{INT}, the threshold voltage V_{TH0}, the bulk effect coefficient K_1, the low-field mobility μ_0, the junction depth X_J and the oxide thickness TOX for p- and n-channel transistors, respectively, as parameters. All parameters were assumed uncorrelated and normally distributed, with mean μ and variance σ derived from the OCL library parameters [17] for typical, slow and fast process corners. The histogram database (HDB) generated on a 32-node high performance clusters contains, for each primitive cell, delay distributions for several hundred defects modeled at electrical level as well as the defect-free case.

As an example, Figure 1 shows an open fault in an embedded NAND2 cell (fault 1), modeled by replacing the parasitic resistor ($R \approx 50\ \Omega$) representing the corresponding wire by a resistor with $R = 500\ \text{k}\Omega$.

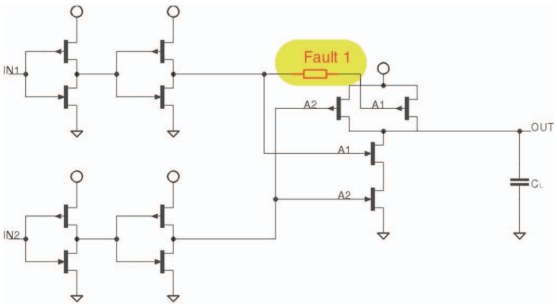

Figure 1. Schematic of the embedded NAND2 gate with injected fault 1.

The delay distributions for this defect as well as for the defect free case are shown in Figure 2. The complete histogram database for the NAND2 gate contains data for 110 resistive-open defects, 130 resistive-short defects and six different input sequences. Monte-Carlo simulations have been performed for 10,000 different parameter configurations.

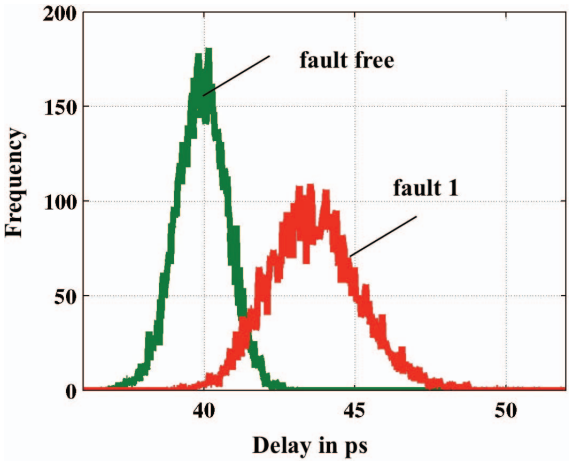

Figure 2. Histograms for the fault free case and for fault 1.

Figure 3 shows the histograms for a different fault in the same primitive cell. Some of the parameter sets result in a static behavior similar to that of stuck-at faults: the transition never completes.

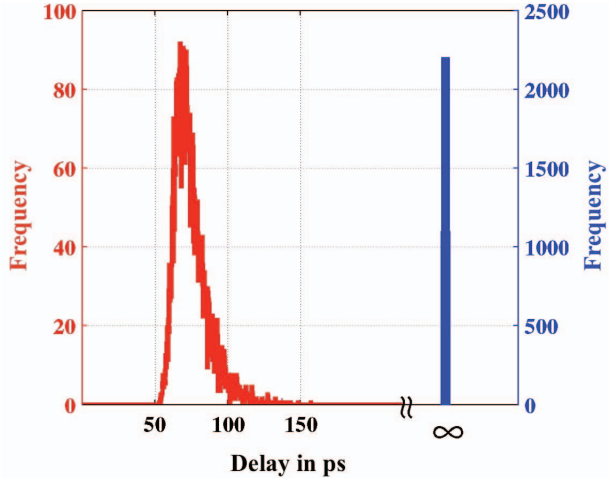

Figure 3. Histograms for fault 2 showing infinite delays.

III. Statistical Testing

The goal of testing is to rule out non functional, defect chips. However, in the presence of parameter variations a clear distinction between fault free and faulty circuits is no longer possible. Instead, a circuit is called robust for a given range of parameters $P = P_1 \times P_2 \times \ldots \times P_N$, if its functional and delay specifications are fulfilled for all parameter values from that range. A defect in a primitive cell or an interconnect may lead to an increased delay, a dynamic fault or to a static fault. Fault models help to reduce the complexity of defect mechanisms, and the conventional definition of fault coverage is

$$FC = \#\,detected\ faults / \#\,modeled\ faults. \qquad (1)$$

Yet variation-aware testing must be based on new coverage metrics measuring the number of parameter combinations for which the test is effective. To obtain an appropriate testability assessment on higher levels of abstraction, it is necessary to use probability density functions for describing the behavior of the affected cells.

Gate delay faults have a certain size D and follow a probability density function $f(D)$. As for each size D we may observe an individual fault coverage $F(D)$ the total fault coverage will be

$$FC = \int_{D\geq 0} FC(D) f(D) dD. \qquad (2)$$

Furthermore, the detectability of a single fault may depend also on the delays of gates not under consideration. A test detecting a fault under arbitrary delay distributions is called a robust test [13]. As robust fault coverage is usually far below 100%, multiple test pattern pairs are required which detect a fault of size D in rather a large space of delay distributions of the circuit components. Hence under process variations, the fault detection depends on the parameter configuration $p = (p_1, p_2, \ldots, p_N) \in P$, and the fault coverage is determined by

$$FC(D) = \int_{p \in P} FC_p(D) f(p) dp, \quad (3)$$

where $f(p)$ is the probability that the parameter configuration p actually appears in a manufactured circuit instance. Equation (3) describes the percentage of the manufactured instances of the circuit in which the test set detects a given fault. There are faults which require multiple test sets depending on the different parameters, this problem is illustrated with the help of Figure 4.

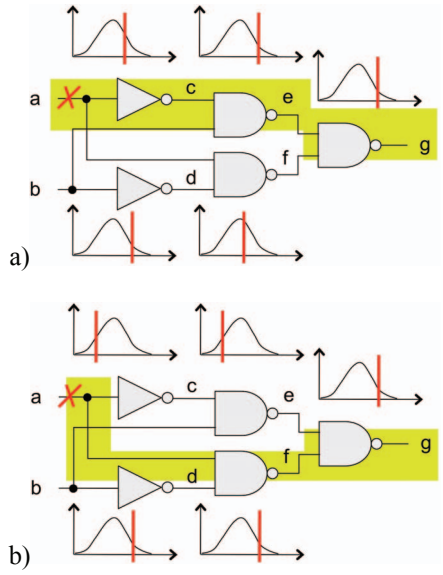

Figure 4. Fault detection under parameter variations.

The circuit in Figure 4 implements an EXOR function using NAND2 gates. To detect a delay fault on input line *a*, conventional delay test generation would try to propagate a transition along the longest path (*a*, *c*, *e*, *g*) in the circuit using the pattern sequence ((*a*,*b*), (*a*',*b*')) = ((0,1), (1,1)). For variation-aware testing the delay distributions of the cells have to be taken into account. For a circuit instance with delays as shown in Figure 4a the path (*a*, *c*, *e*, *g*) is actually the longest path, and the test is a valid test for the delay fault on input line *a*. However, if the actual delays in a circuit instance assume the values as shown in Figure 4b, then the longest path is (*a*, *f*, *g*) and the test is no longer valid. Instead, the test sequence ((0,0), (1,0)) will detect the fault. To maximize the coverage as defined in Equation 3, a test set for this circuit must include both patterns, i.e. T = {((0,1), (1,1)), ((0,0), (1,0))}.

Statistical fault simulation differs from standard fault simulation, as for each pair of patterns it has to return not only whether a fault of size D is detected, but also the range of parameters where detection is possible. Distributions for both gate delays within the specification and fault sizes have to be obtained from low level analysis and can be provided by a histogram or a density function.

Monte Carlo methods could provide means for statistical fault simulation, but they are computationally far too expensive, as they have to be applied for each fault site separately. Symbolic time simulation is an attractive alternative to extract conditions for fault detection. Here, it is not sufficient to look for the latest arrival times only, as this may mask error detection. For example, the glitch in Figure 5 remains undetected, as the output temporarily has the correct value at the observation time before changing again to an incorrect value.

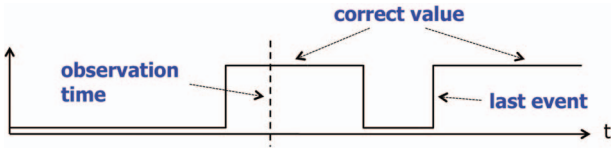

Figure 5. Error masking.

Nevertheless, there is no need to compute the complete history of a signal, only the latest section is required. The history can be described by a set of conditions, where the parameters are delay times of preceding gates. Figure 6 shows how the conditions are propagated through the circuit. Multiple occurrences of t_0 at both inputs of the AND gate indicate reconvergence which is considered in the correlation matrix used during numerical integration. Sophisticated reduction techniques are deployed during simulation to handle large circuits efficiently.

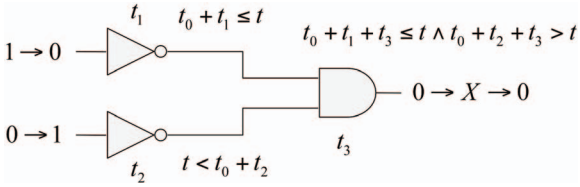

Figure 6. Waveform conditions.

Solving these conditions provides parameters where a fault is observable or not observable. This way the parameter space is defined where a test is not effective and where ATPG has to generate new patterns. Statistical ATPG must try to minimize their number and generate (compact) test sets identifying the fault in as many valid circuits as possible.

Figure 7 shows a possible iterative procedure to solve the problem of test generation in this situation. Statistical fault simulation determines the parameter range covered by the test patterns generated so far, and a variation-unaware ATPG is invoked with fixed parameter values to cover a further parameter set in the range. This is iterated until an acceptable coverage of the complete range is achieved and can be followed by the compaction of the obtained test set. The already mentioned histogram database or a density function obtained by variation-aware library characterization play a crucial role in these analysis steps (see top of Figure 7).

Based on the knowledge of process-induced variations in the individual circuit components and using high-quality variation-aware test patterns, it is possible to separate the different manufactured instances of a circuit into classes or "bins" according to the frequency or voltage they can handle ("frequency binning", "voltage binning"), thus maximizing yield. The emerging concept of "quality binning" takes into account the circuit's robustness, i.e. its expected ability to tolerate the effects of ageing or to recover from transient faults. Using the HDB data, it is possible to judge whether the circuit

is sufficiently robust, such that its deteriorations will not manifest themselves as observable defects. These system-level approaches based on the HDB data and the outcome of the variation-aware test algorithms are shown at the bottom of Figure 7.

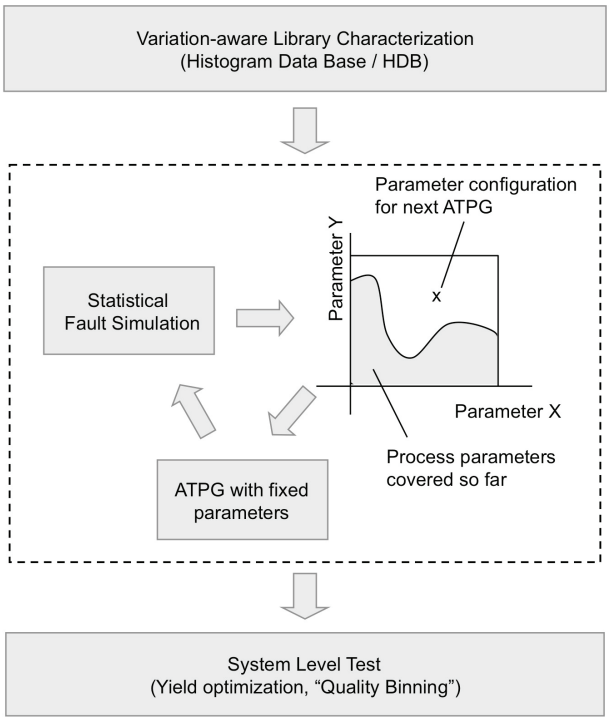

Figure 7. Overview of the statistical test flow.

IV. ATPG IN STATISTICAL TESTING

Apart from statistical fault simulation (see Sec. III) the test flow depicted in Figure 7 makes use of an efficient ATPG environment. Based on the simulation results ATPG is iteratively invoked to generate delay tests for "points in the parameter space" not covered until now. This requires test generation for instances of the same circuit having different, but fixed delay parameters for the primitive cells. To do so, it is necessary to control and sensitize specific paths (or even multiple paths) in the circuit and generate a test input pattern satisfying these multiple constraints.

Recent research has demonstrated that ATPG based on Boolean satisfiability solvers has advantages over classical structural ATPG in the case of highly constrained or (even) unsatisfiable ATPG problems [5][6]. We therefore rely on TIGUAN (Thread-parallel Integrated test pattern Generator Utilizing satisfiability ANalysis), an inhouse state-of-the-art SAT-based ATPG tool [5]. TIGUAN is an effective and flexible tool to generate tests for non-standard fault models by using the conditional multiple-stuck-at fault model (CMS@). In the CMS@ a fault is modeled as an activation list – a set of constraints necessary to activate the fault – together with a victim list corresponding to a set of faulty signals in case of activation. Using a time frame expansion the CMS@ can be extended to model delay defects and control the propagation paths necessary in the context of the statistical test flow. To cope with the complexity of today's circuits and that of the fault models as well TIGUAN supports thread parallelism, thus fully utilizing the performance of multi-processor systems or multi-core processors.

In the second part of this section we shortly mention possible extensions of the concepts presented above. Besides the generation of specific tests or test sets we are interested in analysis methods to identify circuit components that are especially vulnerable to parameter variations. Having determined these components, this can be used e.g. for cost-efficient improvement of robustness.

To identify vulnerable components relevance measures for (combinatorial) components are computed by estimating the probability that a fault in a component will be visible at the circuit's outputs under the assumption of random inputs. The relevance measures for components are reduced to the computation of static and dynamic path relevance values for paths passing through the component of interest. Thereby static path relevance describes the probability that the path is sensitized for a random input and can be reduced to a so-called #SAT problem [20]. Dynamic path relevance denotes the probability that a fault in the component is propagated to the outputs through a sufficiently slow path and thus results in a fault effect. For the computation we plan to use statistical propagation procedures deduced from the method for statistical simulation.

ATPG as well as the identification of vulnerable components can be generalized to the case of systems using information redundancy to increase robustness (see also Sec. V). In this case test patterns must be generated that represent valid input code words and result in valid output code words, thus putting additional constraints into the ATPG process. Also for the computation of robustness measures of components this has to be taken into account.

V. ROBUSTNESS ANALYSIS AND QUALITY BINNING

As explained above a robust circuit design has become mandatory for nanoscale systems. Present solutions range from classical fault tolerant architectures and self-checking systems to new self-calibrating and self-adaptive solutions. To support a reliable system operation at affordable cost self-checking circuits provide an interesting solution [18][23]. They are usually designed to achieve the "totally self-checking goal" (TSC goal), i.e. to detect a fault when it produces the first erroneous output, and various design algorithms are known to guarantee this property [23].

However, synthesis tools cannot always be fully controlled in this respect, and also designers may give priority to area optimization. As a result, many self-checking designs do not guarantee to detect all internal errors. Figure 8 shows an example where logic sharing between the functional logic and the code prediction unit prevents the detection of certain faults. Consequently, design validation and verification must comprise not only the functional but also the robustness properties of the system.

Traditionally, the robustness analysis of fault tolerant designs has been accomplished by fault simulation [3][15]

[30]. To reduce the enormous computational effort first general approaches for formal robust checking based on SAT-solving have been proposed [8]. However, for specific design solutions specifically tailored analysis strategies can work more efficiently.

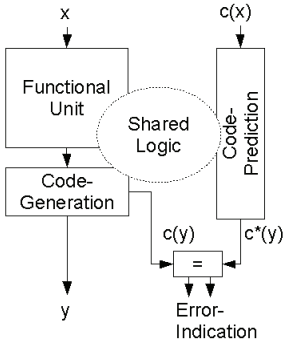

Figure 8. Self-checking circuit using separable codes.

Considering for example strongly fault-secure circuits, which can be seen as the widest class of circuits achieving the totally self-checking (TSC) goal, an ATPG-based approach provides both an efficient solution for robustness analysis and for test preparation [9][10]. To deal with fault accumulation, here the effects of all possible fault sequences have to be taken into consideration. A considerable speedup for the complex multiple fault analysis is achieved by deriving detectability or redundancy information for multiple faults from the respective information for single faults.

A robust design can also compensate manufacturing defects and help to increase yield. Thus, the higher design effort has a twofold benefit. However, if redundancy is used to improve yield, then classical yield models estimating the probability of a fault free device are no longer sufficient. Instead the probability of fault free operation even in the presence of manufacturing defects must be determined as shown in Equation 4 [26]. Here $P(i)$ denotes the probability that i defects occur in the systems and $R(i)$ represents the probability that i defects can be compensated.

$$Y = \sum_{i=0}^{\infty} R(i)P(i) \tag{4}$$

However, equation (4) does not reflect the fault tolerance capabilities of the manufactured system. To both estimate the probability of a correct system functionality and assess the fault tolerance capabilities, the conditional probabilities $R(i + k \mid i)$ that up to k additional faults can be tolerated after compensating i manufacturing defects must be analyzed. Adding them as weight coefficients in Equation (4) provides the "fault tolerant" yield $Y_{FT}(k)$ in Equation (5).

$$Y_{FT}(k) = \sum_{i=0}^{\infty} R(i+k \mid i)R(i)P(i) \tag{5}$$

Exact yield estimation requires a complex multiple fault analysis, so that in general upper and lower bounds must be used. Figure 9 shows preliminary results for a TMR system based on the benchmark circuit b13 compared to the yield of a single module [12]. For growing defect density the yield decreases, but the yield for the TMR system is slightly higher for this example. The fault tolerant yield for the same system is shown in Figure 10.

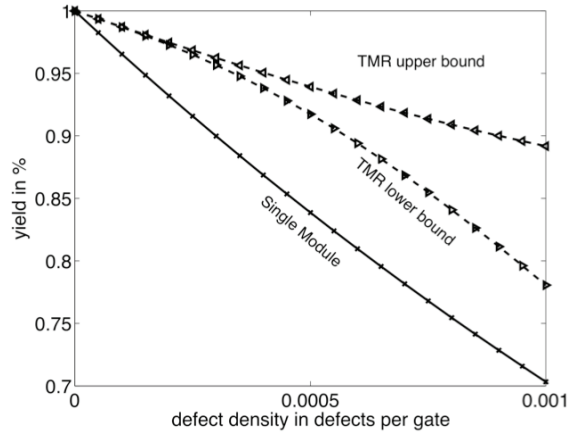

Figure 9. Yield for TMR system based on b13.

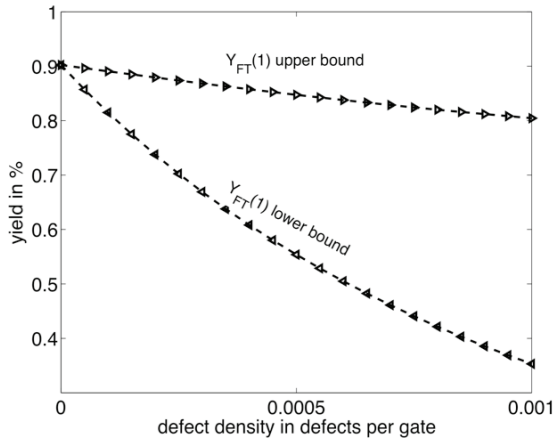

Figure 10. Fault tolerant yield for TMR system based on b13.

Although the bounds are rather loose for larger defect densities, the curves show that it is possible to tolerate additional faults. However, even a single manufacturing defect in a TMR system implies that no longer all possible single faults can be tolerated. For high reliability requirements a hybrid strategy, for example combining TMR with fault detection and checkpointing, can provide a solution.

Finally, for quality binning, the actual residual fault tolerance of manufactured systems must be determined. For this, appropriate design for testability measures have to support the detection of structural, but functionally redundant, faults as well as efficient diagnostic procedures to assess the impact of these faults on the system robustness.

VI. CONCLUSIONS

A multi-step process for dealing with massive statistical process variations during test has been presented. The process starts with variation-aware library characterization providing a database of delay distributions for both defective and defect-

free cells. To provide a suitable interface between the low level library information and test generation as well as simulation tools at higher abstraction levels the delay distributions are stored in a histogram database. Combining statistical fault simulation with appropriate test generation algorithms provides an iterative procedure for variation-aware test generation. Finally, the low-level information in the histogram database and the test generation procedures enable robustness analysis and quality binning at the system level.

ACKNOWLEDGEMENT

The authors would like to thank Infineon Technologies, NXP Semiconductors Hamburg and Mentor Graphics Hamburg for their continuous support.

REFERENCES

[1] R. C. Aitken, "Defect or Variation? Characterizing Standard Cell Behavior at 90 nm and Below," IEEE Trans. on Semiconductor Manufacturing, Vol. 21, No. 1, February 2008, pp. 46-54.

[2] R. Baumann, "Soft errors in advanced computer systems," IEEE Design and Test, Vol. 22, No. 3, 2005, pp. 258-266

[3] C. Bolchini, et al., "The design of reliable devices for mission-critical applications," IEEE Trans. on Instrumentation and Measurement, Vol. 52, Dec. 2003, pp. 1703-1712

[4] S. Borkar, "Designing reliable systems from unreliable components: the challenges of transistor variability and degradation," IEEE Micro, Vol. 25, Nov.-Dec. 2005, pp. 10-16

[5] A. Czutro, I. Polian, M. Lewis, P. Engelke, S.M. Reddy and B. Becker, "TIGUAN: Thread-parallel Integrated test pattern Generator Utilizing satisfiability Analysis, Accepted for publication in Int'l Jour. of Parallel Programming, 2010

[6] R. Drechsler, S. Eggersglüß,, G. Fey,, A. Glowatz, F. Hapke, J. Schlöffel,, D. Tille,: "On acceleration of SAT-based ATPG for industrial designs. IEEE Trans. CAD 27(7), 2008, pp. 1329–1333

[7] D. Ernst, S. Das, S. Lee, D. Blaauw, T. Austin, T. Mudge, N. S. Kim, and K. Flautner, "Razor: Circuit-Level Correction of Timing Errors for Low-Power Operation," IEEE Micro, Vol. 24, No. 6, November-December 2004, pp. 10-20

[8] G. Fey and R. Drechsler, "A Basis for Formal Robustness Checking," Proc. 9th Int. Symp. Quality Electronic Design (ISQED'09), March 2008, pp. 784-789

[9] M. Hunger and S. Hellebrand, "Verification and analysis of self-checking properties through ATPG," Proc. 14th IEEE Int. On-Line Testing Symp. (IOLTS'08), July 2008, pp. 25-30

[10] M. Hunger, S. Hellebrand, A. Czutro, I. Polian, and B. Becker, "ATPG-Based Grading of Strong Fault-Secureness," Proc. 15th IEEE Int. On-Line Testing Symp. (IOLTS'09), July 2009, pp. 269-274

[11] U. Ingelsson, B.M. Al-Hashimi, S. Khursheed, S.M. Reddy, and P. Harrod. "Process variation-aware test for resistive bridges," IEEE Trans. on CAD, Vol. 28, No. 8, Aug. 2009, pp. 1269-1274

[12] ITC'99 Benchmarks: Benchmark information and circuits available at http://www.cerc.utexas.edu/ itc99-benchmarks/bench.html

[13] C. Lin and S. Reddy, "On delay fault testing in logic circuits," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol.5, No. 6, June 1987, pp. 694-703

[14] J.-J. Liou, A. Krstic, Y.-M. Jiang, and K.-T. Cheng, "Modeling, testing, and analysis for delay defects and noise effects in deep submicron devices," IEEE Trans. on CAD, Vol. 22, No. 6, June 2003, pp. 756-769

[15] J.-C. Lo and E. Fujiwara, "Probability to achieve TSC goal," IEEE Trans. on Computers, Vol. 45, No. 4, April 1996 pp. 450-460

[16] S. Mitra, "Globally Optimized Robust Systems to Overcome Scaled CMOS Reliability Challenges," Proceedings Design Automation and Test in Europe, Munich, Germany, March 2008, pp. 941-946

[17] Nangate 45nm Open Cell Library, http://www.nangate.com

[18] M. Nicolaidis and Y. Zorian, "On-line testing for VLSI – a compendium of approaches," J. Electronic Testing, Vol. 12, No. 1-2, 1998, pp. 7-20.

[19] M. Nicolaidis, "GRAAL: A New Fault-Tolerant Design Paradigm for Mitigating the Flaws of Deep Nanometric Technologies," Proceedings IEEE International Test Conference, San Jose, CA, USA, 2007, pp. 1-10

[20] T. Sang, P. Beame and H. Kautz, "Heuristics for Fast Exact Model Counting", Proceedings 8th International Conference on Theory and Applications of Satisfiability Testing, 2005, pp. 226-240

[21] U. Schlichtmann, M. Schmidt, H. Kinzelbach, M. Pronath, V. Glöckel, M. Dietrich, U. Eichler, J. Haase, "Digital Design at a Crossroads – How to Make Statistical Design Industrially Relevant" Proc. Design, Automation and Test in Europe, DATE 2009, Nice, France, April 20-24, 2009, pp. 1542-1547.

[22] M. Simone, M. Lajolo, D. Bertozzi, "Variation tolerant NoC design by means of self-calibrating links," Proceedings Design Automation and Test in Europe, Munich, Germany, March 2008, pp. 1402-1407

[23] J. Smith and G. Metze, "Strongly fault secure logic networks," IEEE Trans. on Computers, Vol. C-27, June 1978, pp. 491-499

[24] A. Srivastava, D. Sylvester, and D. Blaauw, "Statistical Analysis and Optimization for VLSI: Timing and Power," Springer, New York, NY, USA, 2005

[25] B. Straube, B. Müller, W. Vermeiren, C. Hoffmann, S. Sattler, "Analogue fault simulation by aFSIM," Design, Automation and Test in Europe Conference and Exhibition, DATE 2000 – User Forum, Paris, March 27-30, 2000, pp. 205-210.

[26] C. Stroud, "Yield modeling for majority voting based defect-tolerant VLSI circuits," Proceedings of the IEEE SoutheastCON, Lexington, Kentucky, USA, 1999, pp. 229-236

[27] D. Sylvester, D. Blaauw, and E. Karl, "ElastIC: An Adaptive Self-Healing Architecture for Unpredictable Silicon," IEEE Design and Test, Vol. 23, No. 6, November-December 2006, pp. 484-490

[28] J. Xiong, Y. Shi, V. Zolotov, and C. Visweswariah. "Statistical multilayer process space coverage for at-speed test," Proc. Design Automation Conf. 2009, pp. 340-345.

[29] M. Yilmaz, K. Chakrabarty, and M. Tehranipoor. "Interconnect-aware and layout-oriented test pattern selection for small-delay defects," Proc. Int'l Test Conf. 2008, Paper 28.3

[30] S. Zhang and J. C. Muzio, "Evaluating the safety of self-checking circuits," J. Electronic Testing, Vol. 6, No. 2, 1995, pp. 243-253

[31] V. Zolotov, C. Visweswariah, and J. Xiong. "Voltage binning under process variation," Proc. Design Automation Conf. 2009, pp. 425-432