# Accelerating Chip Design with Machine Learning: From Pre-Silicon to Post-Silicon

Cheng Zhuo[1], Bei Yu[2], Di Gao[1],
[1]College of ISEE, Zhejiang University, Hangzhou, China
[2]Department of CSE, The Chinese University of Hong Kong, NT, Hong Kong
[1]czhuo@zju.edu.cn

## ABSTRACT

At sub-22nm regime, chip designs have to go through hundreds to thousands of steps and tasks before shipment. Many tasks are data and simulation intensive, thereby demanding significant amount of resources and time. Unlike conventional methodologies relying on experiences to manually handle data and extract models, recent advances in machine learning techniques enable the successful applications in various complex tasks to accelerate modern chip designs, ranging from pre-silicon verification to post-silicon validation and tuning. The goals are to reduce the amount of time and efforts to process and understand data through automatic and effective learning and enhancing from examples. In this paper we review and discuss several application cases of machine learning techniques, including pre-silicon hotspot detection through classification, post-silicon variation extraction and bug localization through inference, and post-silicon timing tuning through iterative learning and optimization, so as to leverage the potentials and inspire more future innovations.

## 1. INTRODUCTION

At sub-22nm regime, due to the growing process variation, increased design complexity and reduced design margin, chip design and manufacturing has become an overwhelmingly complex procedure that involves hundreds to thousands of steps [1, 2, 3]. In order to ensure design robustness and performance target throughout entire procedure, designers and manufacturers have to conduct repeated tasks in pre-silicon verification, post-silicon validation and tuning [4, 5, 6, 7, 8]. Many of such tasks demand processing, understanding and utilizing of a significant volume of data to verify chip functionality, validate process condition, and tune chip performance. Ignorance of any of the aforementioned data-intensive tasks may inevitably degrade chip design quality causing expensive over-design or impair manufacturing yield. On the other hand, conventional approach to such data (*e.g.*, wafer process or lithography) and tasks is largely manual and ad-hoc, which is time and resource consuming and eventually delays time-to-market. Thus, it remains a critical field of research with exciting opportunities to deal with complex and data-intensive tasks for pre- and post-silicon chip design and optimization.

However, this is not a trivial work. Take post-silicon validation for example. Post-silicon validation typically begins with fabricated prototyped chips with data collected from each of them to understand the process condition, diagnose chip functionality and validate chip performance [6, 9, 10, 11]. Even though testing can be conducted with high speed, due to the significant amount of data to be collected and extremely intensive tests to be conducted on each of the chips, the initial stages of validation are already slow. Moreover, once data is fed back, designers need to understand the data, associate that with underlying chip functionalities, root-cause and fix the problems if any. Since data and measurements can be easily impacted by process and environmental variations, it remains a challenge to derive effective models between chip functionalities and data. As a result, it is highly desired to have novel and efficient methods to deal with such data-intensive complex tasks at pre- and post-silicon stages to accelerate chip designs.

With recent advances in machine learning and its unique feature to extract structure from intensive data, many researchers [3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14] proposed various promising approaches to handle complex problems ranging from pre-silicon verification to post-silicon validation and tuning. The goals are to reduce the amount of time and efforts to process and understand data through automatic and effective learning and enhancing from examples.

In the following, we will discuss the applications of various machine learning techniques in the aforementioned tasks to leverage their potential and inspire more future innovations. The discussion covers pre-silicon hotspot detection through classification, post-silicon variation extraction and bug localization through inference, and post-silicon timing tuning through iterative learning and optimization. The paper is organized as follows. Section 2 demonstrates the application of machine learning techniques in a pre-silicon verification problem of hotspot detection. Section 3 discusses the post-silicon process variation extraction and bug localization. Section 4 presents an example of post-silicon timing tuning through machine learning, followed by section 5 to conclude the paper.

## 2. PRE-SILICON HOTSPOT DETECTION

In physical design and verification stages, the hotspot detection problem is to locate hotspots on a given layout with fast turn-around-time. Conventional hotspot detection widely relies on full-chip lithography simulation, which can achieve very high accuracy but may suffer from being extremely computational expensive [15, 16].

To provide quick feedback to circuit designers, recently many fast and coarse-grained hotspot detectors are pro-

978-1-5386-4034-0/17/$31.00 ©2017 IEEE

posed, which can be roughly classified into pattern matching based [17, 18, 19] and machine learning based (*e.g.*, [20, 21, 22, 23]).

On one hand, pattern matching works on a pre-defined pattern set, thus it is less efficient on detecting unseen hotspots. Although fuzzy-pattern matching can be applied to dynamically tune appropriate fuzzy regions around known hotspots, it is still case sensitive and lacks the generalization ability to various detection environment. Some literature work combines both the pattern matching and machine learning methods. For instance, Wen *et. al.* [19] presents an idea of fuzzy matching integrated with some learning techniques, which can dynamically tune appropriate fuzzy regions around known hotspots in multi-dimensional space.

On the other hand, machine learning technique is to construct a regression model based on the training data set. Thanks to the good generalization ability, usually machine learning method can lead to higher prediction accuracy, with slightly longer training and prediction time. In addition, this method can recognize unknown hotspots through learning relationships between layout patterns and their printability. There are several works targeting at hotspot detection accuracy. Drmanac *et. al.* [20] propose a 2-D distance transform and histogram extraction method for mask layout images, which are used to construct the SVM-based hotspot detector. Ding *et. al.* [24] present a neural network judgment based detection flow, where 2-D hotspot patterns are directly used to train an ANN model. Ding *et. al.* [22] further propose a unified meta-classifier that enables several classifiers to work together. For each layout pattern, certain hotspot features are extracted and then fed into each base classifier, which calculates the prediction decision and generates a weight based on the weighting functions. In [23, 25], density-based layout features are fed into SVM engines for high performance hotspot detection. Reference [26] considers the hotspot detection in directed self-assembly (DSA) scinario, and enables robust learning with point correspondence and segment distance features. Zhang *et. al.* [27] develop smooth boosting based hotspot detection methodology, which can be further extended to online learning scenario.

It should be noted that feature extraction is always a key component in a high performance hotspot detector. References [28, 29, 30] perform feature extraction using principle component analysis (PCA), which significantly improves the performance of SVM-based hotspot detectors. In [31], thanks to the simplified layout features, the detector can avoid over-fitting issue, and is able to detect hotspots accurately with low false alarm. Zhang *et. al.* [27, 32] further study the feature optimization.

Very recently, some work develops deep learning-based framework targeting at high performance and large scale hotspot detection [7, 12, 13]. Because of the automatic feature learning technique and highly nonlinear neural networks, deep learning has shown improvement in term of accuracy and false alarm number especially on some large scale benchmark.

# 3. POST-SILICON VARIATION EXTRACTION AND BUG LOCALIZATION

With process shrinking below 22nm, chip manufacturing is subject to increasingly significant process variability [1, 33]. Thus, process variation models are considered critical not only for pre-silicon design, but also for post-silicon applications, such as characterization, validation, diagnosis and tuning, *etc.*. Due to its criticality, it has been a common practice in modern process to deploy hundreds to thousands of tiny test structures (*e.g.*, ring oscillators and resistor arrays) within die or in the scribe line of the wafer for process monitoring and measurement [34, 35]. However, the overhead to measure all the test structures for all dies across all wafers is clearly too high [35, 36, 37]. By noting the statistical nature of process variation and the underlying process similarity from wafer to wafer, it remains an interesting yet challenging question how to reduce the testing cost by reusing measured information from earlier wafers to facilitate following wafer measurements.

To address this challenge, [9, 37, 38, 39] have proposed to simulate tens to hundreds of wafers to achieve a converged model and then use the model to reduce the number of measurements that were taken to monitor process on each wafer. However, in these approaches, the number of measurements and the choice of the measured devices were determined throughout the entire lot. In other words, they were not able to adapt to process change within the lot and from lot to lot. Instead of relying on fixed measurements, [10] proposed a Sparse Bayesian Learning based dynamic framework to monitor the process change and extract a variation model from small test structures. The extracted model considers various variation sources, including systematic variation, spatially-correlated variation and random variation. The proposed learning framework is capable to reduce the test cost by reusing the process model built from past wafer measurements and evolve the process model by taking partial measurements on the new wafer for validation and improvement.

Figure 1 provides an overview of the flow chart for the proposed framework. As can be seen from the figure, the framework starts with initial measurements conducted on the first (or first few) wafer(s) to train an initial process model that includes deterministic pattern (at wafer- and reticle levels) and non-deterministic variations. This model is achieved with high measurement density (*i.e.*, measuring every possible test structure for this particular wafer). This measurement policy will later be changed with more wafers coming and model maturing. In other words, the number of measurements on each wafer is gradually decreased over time when the model fidelity increases. After training, for each wafer, a few reticles are selected to decide whether the process model is still valid or the process has gone through some unexpected change and needs adjustion. After cross-validation, a small amount of tests is taken from untested reticles to estimate process variation parameters together with the trained models and further adapt the process model if necessary. Experimental results show that such measurement policy is able to significantly reduce the test cost while allowing the model to adapt and improve with the process changes.

Similar as process variation extraction, bug localization in diagnosis, especially for analog and mixed signal circuits, is also confronted with a large number of possibilities, and almost impossible to execute all the paths and tests [6, 40, 41, 42]. Thus, it is not surprising the task is often manually handled by experienced designers with months of efforts [6, 41, 42].

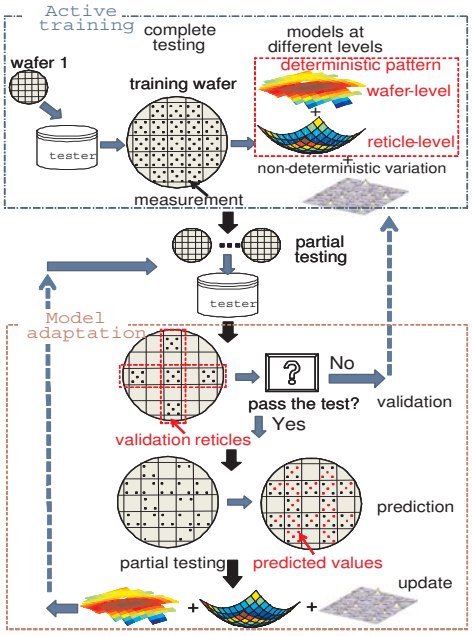


Figure 1: Flow chart of the Sparse Bayesian Learning based framework in [10].

Several machine learning and data mining techniques [6, 14, 43] have been proposed to extract potential invariants and variants in the model from the data. However, this still relies on a large amount of test data and suffers from the sensitivity to subtle process variations. Reference [11] proposes an approach to model the circuit using probabilistic graphical models and Bayesian inference with partially observed circuit responses. The approach is capable of modeling both circuit and environmental uncertainties and incorporating a small set of known data to the probabilistic graphical model to localize and rank the potential bugs. Although its efficiency may depend on the partial observations and induce false warnings, this still remains an effective approach to prune significant portion of the solution space, thereby making localization more tractable.

## 4. POST-SILICON TIMING TUNING

The growing process variations at sub-22nm have caused unaffordable timing margins during the design phase. These variations can be modeled as random variables, leading to statistical static timing analysis (SSTA) to analyze circuit yield [44]. In addition, new techniques ranging from circuit structure to special devices and mechanisms have been proposed to improve circuit performance or yield in view of these statistical characteristics. For example, the Razor method [45] increases the clock frequency until timing errors occur during operation.

Another way to alleviate the effect of process variations is to deploy post-silicon tuning devices into the circuit. After manufacturing, chips are adapted according to the final effect of process variations. Since this adaptation happens to each chip individually, chips with timing failures due to process variations have an additional chance to be rescued by a customized tuning scheme.

Post-silicon tuning can be applied in different domains. For example, a tunable delay buffer can be used to adjust the arrival time of a clock signal to the corresponding flip-flop. The structure of the delay buffer from [46] is shown in Figure 2, where the configuration bits in the three registers can be set through the scan chain to change the delay of the buffer. In high-performance designs, such buffers are inserted during the design phase. After manufacturing, more timing slack can be assigned to critical paths, which might be different in different chips, by tuning the delays of the buffers to shift clock edges toward the stages with smaller delays, so that the yield of the circuit can be improved [47, 48].

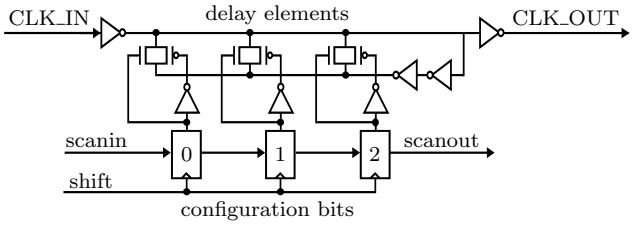


Figure 2: Post-silicon tuning buffer in [46].

Since the size of a tuning buffer is quite large, it is not profitable to insert a tuning buffer for each flip-flop in the circuit. Instead, only the flip-flops related to the paths affecting the circuit performance/yield should be considered as candidates for buffer insertion. The straightforward technique to insert clock tuning buffers to critical paths, however, does not produce a good result, because tuning the clock edge at a flip-flop affects all the paths incoming to or outgoing from it. Considering the fact that all the gate/path delays should be considered as statistical when determining the buffer locations during the design phase, this problem can be formulated as a learning problem illustrated in Figure 3.

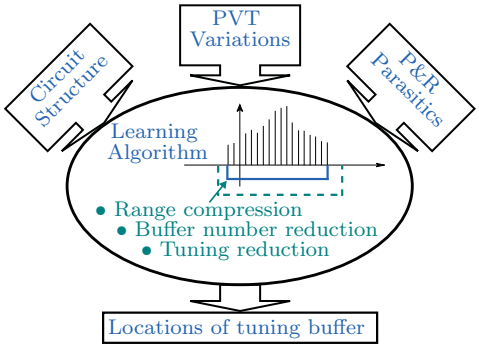


Figure 3: Learning scenario for post-silicon circuit tuning.

To determine the optimal locations of tuning buffers, the learning algorithm takes the circuit structure, the variation information from manufacturing process, supply voltage and operating temperature (PVT), as well as the results of physical design and parasitic extraction as input. From these data, the learning algorithm can construct the statistical characteristics of path delays in the circuit. Afterwards, these statistical delays are discretized to represent different manufacturing cases. For each sample, the learning algorithm evaluates potential buffer locations. Considering the probabilities of these samples, the learning algorithm gradually approaches the optimal combination of buffer locations while reducing buffer area and potential test cost.

In the research field of electronic design automation, machine learning methods have been applied extensively, *e.g.*, for test path selection under process variations [49], for placement of sensors to manage noise dynamically [50], and for yield calculation of analog-/mixed-signal integrated circuits [51]. Specifically for allocating post-silicon tuning buffers considering process variations, the method [52] first samples process variations. For each sample, a limited number of buffer locations are identified. The buffer locations that appear most often in the results are selected as the result. This method is extended in [8] to process multiple samples simultaneously, so that the relation between discretized samples is also incorporated. The execution efficiency of this learning process is improved by using a low-discrepancy sample sequences (Sobol sequence). After manufacturing, the inserted buffers can also be used to reduce testing time for clock tuning, as explained in [53]. These methods are, however, still evolved from the traditional Monte Carlo method, and a direct combination of machine learning algorithms with the sampling-learning concept can potentially improve the existing results significantly further.

# 5. CONCLUSIONS

The paper discusses a few novel applications of machine learning techniques in chip design and manufacturing. With a growing number of data intensive tasks, it is highly desired to fully unlock the potential of machine learning in classification, inference and optimization. Its successful applications may mitigate designers' workload to extract the key information from massive data set, eventually accelerating chip design procedure. However, there remain many challenges and future works in this field. For example, in most prior research, it is often not well discussed how to train and validate the model, which is apparently crucial in practice. Moreover, it remains an open question to apply machine learning to tasks with both data and simulation intensity. One example of such tasks is power delivery simulation and optimization, which has a huge design space and demands many rounds of expensive simulations from early design stage to chip sign off [50, 54, 55]. The simulation itself is typically data dependent and sensitive to any error introduced in the solution space. It is interesting and crucial to develop data-independent machine learning algorithms by taking advantage of circuit structure to accelerate the simulation and optimization procedure. The impact of such a solution may significantly change the traditional back-end design methodologies and tools.

# 6. REFERENCES


[1] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De, "Parameter variations and impact on circuits and microarchitecture," *Proc. DAC*, pp. 338–342, 2003.

[2] L. Capodieci, "Data analytics and machine learning for design-process-yield optimization in electronic design automation and IC semiconductor manufacturing," *Proc. CSTIC*, pp. 1–3, 2017.

[3] D. Kim, P. Kang, S. Cho, H. Lee, and S. Doh, "Machine learning-based novelty detection for faulty wafer detection in semiconductor manufacturing," *Elsevier ESA*, vol. 39, no. 4, pp. 4075–4083, 2012.

[4] Y. Meidan, B. Lerner, G. Rabinowitz, and M. Hassoun, "Cycle-time key factor identification and prediction in semiconductor manufacturing using machine learning and data mining," *IEEE Trans. SM*, vol. 24, no. 2, pp. 237–248, 2011.

[5] S. Wang and W. Wei, "Machine learning-based volume diagnosis," *Proc. DATE*, pp. 902–905, 2009.

[6] S. Mitra, S. A. Seshia, and N. Nicolici, "Post-silicon validation opportunities, challenges and recent advances," *Proc. DAC*, pp. 12–17, 2010.

[7] H. Yang, J. Su, Y. Zou, B. Yu, and E. F. Y. Young, "Imbalance aware lithography hotspot detection: A deep learning approach," *Proc. DAC*, pp. 62:1–62:6, 2017.

[8] G. L. Zhang, B. Li, J. Liu, Y. Shi, and U. Schlichtmann, "Design-phase buffer allocation for post-silicon clock binning by iterative learning," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 36, 2017.

[9] W. Zhang, X. Li, and R. Rutenbar, "Bayesian virtual probe: minimizing variation characterization cost for nanoscale IC technologies via bayesian inference," *Proc. DAC*, pp. 262–267, 2010.

[10] C. Zhuo, K. Agarwal, D. Blaauw, and D. Sylvester, "Active learning framework for post-silicon variation extraction and test cost reduction," *Proc. ICCAD*, pp. 508–515, 2010.

[11] S. Youn, C. Gu, and J. Kim, "Probabilistic bug localization via statistical inference based on partially observed data," *Proc. DAC*, pp. 1–6, 2014.

[12] M. Shin and J.-H. Lee, "Hotspot prevention and detection method using an image-recognition technique based on higher-order local autocorrelation," *JM3*, vol. 15, no. 4, pp. 043507-1–043507-12, 2016.

[13] H. Yang, L. Luo, J. Su, C. Lin, and B. Yu, "Imbalance aware lithography hotspot detection: A deep learning approach," *Proc. SPIE Advanced Lithography*, 2017.

[14] A. DeOrio, Q. Li, M. Burgess, and V. Bertacco, "Machine learning-based anomaly detection for post-silicon bug diagnosis," *Proc. DATE*, pp. 491–496, 2013.

[15] J. Kim and M. Fan, "Hotspot detection on Post-OPC layout using full chip simulation based verification tool: A case study with aerial image simulation," *Proc. SPIE*, vol. 5256, 2003.

[16] E. Roseboom, M. Rossman, F.-C. Chang, and P. Hurat, "Automated full-chip hotspot detection and removal flow for interconnect layers of cell-based designs," *Proc. SPIE*, vol. 6521, 2007.

[17] J. Xu, S. Sinha, and C. C. Chiang, "Accurate detection for process-hotspots with vias and incomplete specification," *Proc. ICCAD*, pp. 839–846, 2007.

[18] Y.-T. Yu, Y.-C. Chan, S. Sinha, I. H.-R. Jiang, and C. Chiang, "Accurate process-hotspot detection using critical design rule extraction," *Proc. DAC*, pp. 1167–1172, 2012.

[19] W.-Y. Wen, J.-C. Li, S.-Y. Lin, J.-Y. Chen, and S.-C. Chang, "A fuzzy-matching model with grid reduction for lithography hotspot detection," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 33, no. 11, pp. 1671–1680, 2014.

[20] D. G. Drmanac, F. Liu, and L.-C. Wang, "Predicting variability in nanoscale lithography processes," *Proc. DAC*, pp. 545–550, 2009.

[21] D. Ding, J. A. Torres, and D. Z. Pan, "High performance lithography hotspot detection with successively refined pattern identifications and machine learning," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 30, no. 11, pp. 1621–1634, 2011.

[22] D. Ding, B. Yu, J. Ghosh, and D. Z. Pan, "EPIC: Efficient prediction of IC manufacturing hotspots with a unified meta-classification formulation," *Proc. ASPDAC*, pp. 263–270, 2012.

[23] Y.-T. Yu, G.-H. Lin, I. H.-R. Jiang, and C. Chiang, "Machine-learning-based hotspot detection using topological classification and critical feature extraction," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 3, pp. 460–470, 2015.

[24] D. Ding, X. Wu, J. Ghosh, and D. Z. Pan, "Machine learning based lithographic hotspot detection with critical-feature extraction and classification," *Proc. ICICDT*, pp. 219–222, 2009.

[25] J.-Y. Wu, F. G. Pikus, A. Torres, and M. Marek-Sadowska, "Rapid layout pattern classification," *Proc. ASPDAC*, pp. 781–786, 2011.

[26] Z. Xiao, Y. Du, H. Tian, M. D. F. Wong, H. Yi, H.-S. P. Wong, and H. Zhang, "Directed self-assembly (DSA) template pattern verification," *Proc. DAC*, pp. 55:1–55:6, 2014.

[27] H. Zhang, B. Yu, and E. F. Y. Young, "Enabling online learning in lithography hotspot detection with information-theoretic feature optimization," *Proc. ICCAD*, pp. 47:1–47:8, 2016.

[28] J.-R. Gao, B. Yu, and D. Z. Pan, "Accurate lithography hotspot detection based on PCA-SVM classifier with hierarchical data clustering," *Proc. SPIE*, vol. 9053, 2014.

[29] B. Yu, J.-R. Gao, D. Ding, X. Zeng, and D. Z. Pan, "Accurate lithography hotspot detection based on principal component analysissupport vector machine classifier with hierarchical data clustering," *JM3*, vol. 14, no. 1, pp. 011003-1–011003-12, 2015.

[30] H. Nosato, H. Sakanashi, E. Takahashi, M. Murakawa, T. Matsunawa, S. Maeda, S. Tanaka, and S. Mimotogi, "Hotspot prevention and detection method using an image-recognition technique based on higher-order local autocorrelation," *JM3*, vol. 13, no. 1, pp. 011007-1–011007-12, 2014.

[31] T. Matsunawa, J.-R. Gao, B. Yu, and D. Z. Pan, "A new lithography hotspot detection framework based on AdaBoost classifier and simplified feature extraction," *Proc. SPIE*, vol. 9427, 2015.

[32] H. Zhang, F. Zhu, H. Li, E. F. Y. Young, and B. Yu, "Bilinear lithography hotspot detection," *Proc. ISPD*, pp. 7–14, 2017.

[33] B.E. Stine, D.S. Boning, and J.E. Chung "Analysis and decomposition of spatial variation in integrated circuit processes and devices," *IEEE Trans. SM*, vol. 10, no. 1, pp. 24–41, 1997.

[34] M. Ketchen and M. Bhushan. "Product-representative at-speed test structures for cmos characterization," *IBM JRD*, vol.50 no.4/5:451–468, 2006.

[35] A. Gattiker, "Using test data to improve IC quality and yield," *Proc. ICCAD*, pp. 771–777, 2008.

[36] K. Qian and C. J. Spanos, "A comprehensive model of process variability for statistical timing optimization," *Proc. SPIE*, vol. 69251, 2008.

[37] S. Reda and S. Nassif, "Analyzing the impact of process variations on parametric measurements: novel models and applications," *Proc. DATE*, pp. 375-380, 2009.

[38] X. Li, R. Rutenbar, and R. Blanton, "Virtual probe: a statistically optimal framework for minimum-cost silicon characterization of nanoscale integrated circuits," *Proc. ICCAD*, pp. 433–440, 2009.

[39] L. Cheng, P. Gupta, C. J. Spanos, K. Qian, and L. He, "Physically justifiable die-level modeling of spatial variation in view of systematic across wafer variability," *Proc. DAC*, pp. 104-109, 2009.

[40] E. M. Clarke, O. Grumberg, and D. Peled, "Model checking," MIT press, 1999.

[41] B. R. Liblit., "Cooperative bug isolation," PhD thesis, University of California at Berkeley, Berkeley, CA, USA, 2004.

[42] L. S. Milor, "A tutorial introduction to research on analog and mixed-signal circuit testing," *IEEE Trans. CAS-II*, vol. 45, no. 10, pp. 1389–1407, 1998.

[43] W. Li, A. Forin, and S. A. Seshia, "Scalable specification mining for verification and diagnosis," *Proc. DAC*, pp. 755–760, 2010.

[44] D. Blaauw, K. Chopra, A. Srivastava, and L. Scheffer, "Statistical timing analysis: from basic principles to state of the art," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 4, pp. 589–607, 2008.

[45] D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner, and T. Mudge, "Razor: A low-power pipeline based on circuit-level timing speculation," *Proc. MICRO*, pp. 7–18, 2003.

[46] S. Naffziger, B. Stackhouse, T. Grutkowski, D. Josephson, J. Desai, E. Alon, and M. Horowitz, "The implementation of a 2-core, multi-threaded Itanium family processor," *IEEE J. Solid-State Circuits*, vol. 41, no. 1, pp. 197–209, 2006.

[47] B. Li, N. Chen, and U. Schlichtmann, "Fast statistical timing analysis for circuits with post-silicon tunable clock buffers," *Proc. ICCAD*, pp. 111–117, 2011.

[48] B. Li and U. Schlichtmann, "Statistical timing analysis and criticality computation for circuits with post-silicon clock tuning elements," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 11, pp. 1784–1797, 2015.

[49] J. Xiong, Y. Shi, V. Zolotov, and C. Visweswariah, "Statistical multilayer process space coverage for at-speed test," *Proc. DAC*, pp. 340–345, 2009.

[50] T. Wang, C. Zhang, J. Xiong, and Y. Shi, "Eagle-eye: a near-optimal statistical framework for noise sensor placement," *Proc. ICCAD*, pp. 437–443, 2013.

[51] F. Gong, H. Yu, Y. Shi, and L. He, "Variability-aware parametric yield estimation for analog/mixed-signal circuits: Concepts, algorithms, and challenges," *IEEE Design & Test*, vol. 31, no. 4, pp. 6–15, 2014.

[52] G. L. Zhang, B. Li, and U. Schlichtmann,

"Sampling-based buffer insertion for post-silicon yield improvement under process variability," *Proc. DATE*, pp. 1457–1460, 2016.

[53] G. L. Zhang, B. Li, and U. Schlichtmann, "EffiTest: Efficient delay test and statistical prediction for configuring post-silicon tunable buffers," *Proc. DAC*, pp. 60:1–60:6, 2016.

[54] C. Zhuo, G. Wilke, R. Chakraborty, A. Aydiner, S. Chakravarty, and W. Shih, "A silicon-validated methodology for power delivery modeling and simulation," *Proc. ICCAD*, pp. 255–262, 2012.

[55] C. Zhuo, H. Gan, and W. Shih, "Early-stage power grid design: extraction, modeling and optimization," *Proc. DAC*, pp. 79:1–79:6, 2014.