Statistical Delay Prediction for Configuring PST Buffers using Correlation-Based Clustering

Internship Report

Submitted by: Dhiraj Pathania

An internship report submitted towards the partial fulfillment of
Master of Science in Communications Engineering

Institute for Electronic Design Automation
Technische Universität München (TUM)
Arcisstr. 21, D-80333 Munich
Germany

ABSTRACT

With shrinking feature size, process variations are becoming more and more pronounced and are affecting output yield of the manufactured chips. This makes it extremely important to design circuits considering process variations. Static timing analysis is one such approach that helps to consider these variations during design phase. However, these variations are random in nature and therefore modeling them statistically is better given their random nature. In recent years, it has become popular to compensate these variations by inserting Post-Silicon Tunable (PST) buffers during design phase. These buffers can be tuned after manufacturing to improve binning yield by compensating delay variations. However, on design side it remains a challenge to optimize design statistically rather than overdesigning. Also after manufacturing, buffer tuning requires testing of chips for target clock frequency and then tuning. Each chip needs to be tested individually and has its own settings for PST buffers. This testing is expensive because of tester time consumed for each chip. In this work, a smart approach is proposed to limit statistical uncertainty in path delays and bring them within safe limits by exploiting correlations among various parameters. In addition, our problem formulation also gives parameters that are needed to be measured to obtain optimal solution. A heuristic approach is also given.

1 INTRODUCTION

WITH advancements in technology, feature sizes are becoming smaller and smaller. This is posing many difficulties in increasing system performance while maintaining a desirable yield. Process variations and circuit aging effects require circuits to be designed with large timing margins and therefore cause overdesign. Large margins increase lower limit for clock period and make it further difficult to increase yield for high performance designs. During last decade, researchers started to model delays as random variables and this led to development of various Statistical Static Timing Analysis (SSTA) methods. These methods helped to improve yield for high performance designs as they significantly capture the variations which are random in nature.

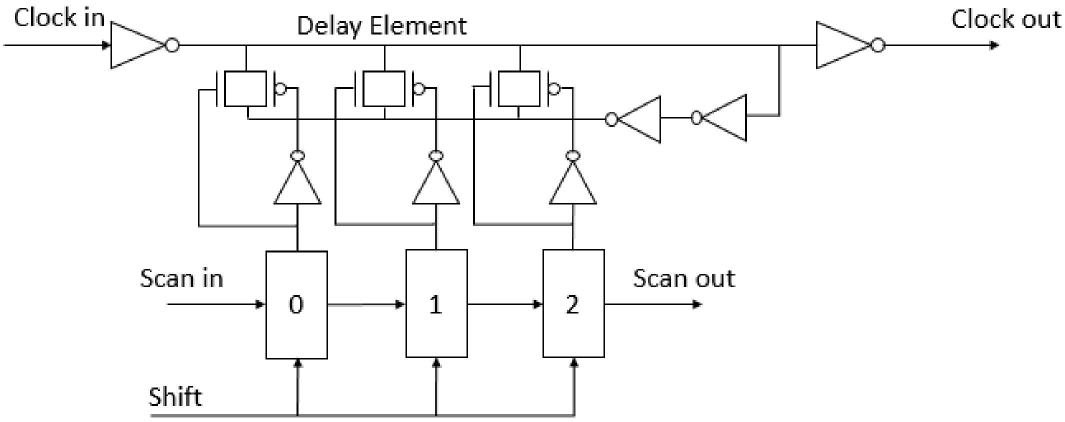


Fig. 1: Post-Silicon Tunning (PST) buffer as in [3].

Several methods have been proposed for optimizing circuits before and after manufacturing. Using delay buffers to tune clock tree skews after manufacturing is one widely adopted approach. Post-Silicon Tunable (PST) buffers are inserted into clock tree during design phase. After manufacturing, delays of these buffers can be programmed to provide some slack in critical paths. In [3], authors have proposed a delay buffer (clock vernier device) design as shown in Figure 1. This buffer can be adjusted to different delay values using configuration registers. Many researchers have worked on designing post-silicon tunable clock trees using similar delay buffers.

Post-Silicon Tunable (PST) buffers are inserted into clock trees to balance clock skews and thus clock scheduling as in [4]. Algorithms are developed in [5] to insert buffers in clock trees that provide a guaranteed yield while minimizing buffers inserted or total buffer area. The yield loss due to process variations and tuning buffer cost are considered for gate sizing in [6]. In [7], a significant improvement is observed when clock tree designed using proposed buffer placement and tuning system. In [8], searching a configuration tree along with a graph pruning and buffer clustering using insertion algorithm is proposed for efficient post-silicon tuning. In [12], efficient timing analysis for such circuits is proposed.

Post-silicon testing methods are shown in [9] and [13] for circuits with tunable buffers.

Post-silicon tuning requires delays of critical paths to be measured. So far, this has been done using frequency stepping methods as proposed in [4], [8], [9], and [13]. These methods consume a lot of tester time and therefore are expensive. Liu and S. Spatnekar in [10] have proposed an approach where some test structures are placed on the die and delay variations measured from these test structures coupled with SSTA are used to narrow down timing probability density function. In [1], a efficient framework (EffiTest) is proposed to align delay measurements of critical paths and thereby enable parallel delay measurements. It also proposes exploiting path correlations which allows us to measure only some representative path delays and then determine delays of the remaining paths using measured representative delays.

2 Background And Problem Formulation

2.1 Background

Parameter variations are not completely random when considered in context with neighboring elements. There is some spatial correlation among parameter variations depending on their spacial proximity. Parameter variations can be generally categorized in two parts: intradie variations and interdie variations. There are different techniques to incorporate them when considering spatial correlation (as explained in [10]). The important aspect here is that parameter variation in one circuit/gate can reveal some information about variations in near by circuits/gates. Therefore, we do not need to measure delays of all critical paths in a neighborhood if delay of one path in vicinity is measured. As long as our delays are ensured to be within the bounds determined by clock period (T_{clk}), there is no need to measure individual delays as in [1] and [10].

In high performance circuits, Post-Silicon Tunable buffers are used to provide room for yield improvement after manufacturing. In such circuits, PST buffers are placed in clock tree during design phase and are tuned after manufacturing to adjust clock delays. This way circuits are given some slack to avoid timing violations. For example, Figure 2 shows how PST buffer at D2 can be configured to reduce clock period (T_{clk}) from 5ns to 4ns. There are various methods proposed for PST buffer tuning and [9] explains one such method.

2.2 Problem Formulation

SSTA is completed for the circuit before manufacturing. Due to practical constraints such as area, power etc., the number of flip-flops having PST buffers is limited. The maximum delay of circuit is represented by random variable x. Let vector $X = [x_1, x_2....x_n]$ is the vector of all circuit delays on a chip. Let S be the set of all representative paths that we need to measure using tester and R is the set containing all remaining paths. The objective is to use measurements from S to predict probability distribution of elements of R. In other words,

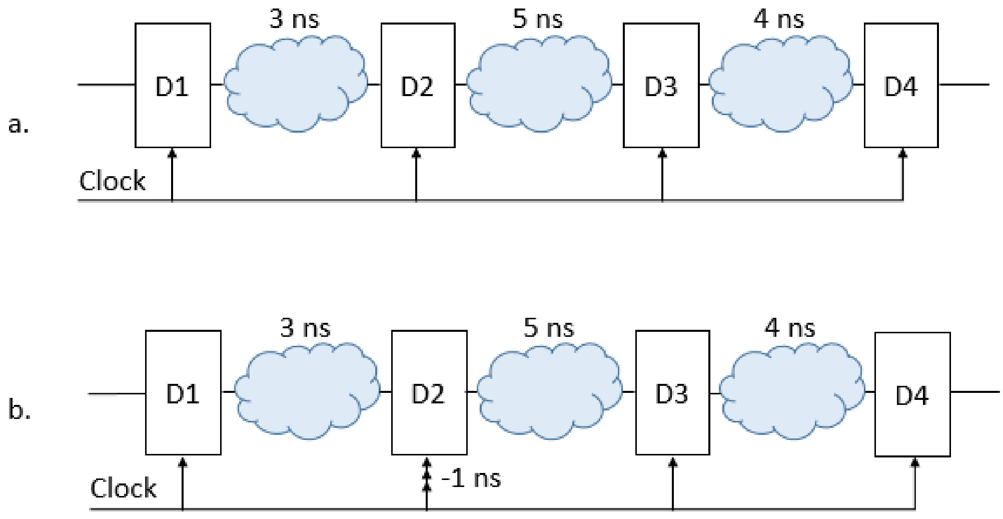


Fig. 2: (a.) Clock Period (t_{clk}) of circuit is 5ns. (b.) Clock Period (t_{clk}) of circuit reduced to 4ns using PST buffer before D2 flip-flop.

the aim is to find probability distribution of R given S($P(R|S)$) by exploiting correlation information between parameters. We have parameter covariance matrix available as input.

In general, the kind of variation should be similar in all circuits in same neighborhood. Using this technique, it is possible to predict parameter variation at different locations as long as spatial correlation is provided by SSTA. There are, however, parameters such as dopant concentration (N_A) and oxide thickness (T_{ox}), which do not show any spatial correlation. Thus, prediction of such parameters is not possible. Splitting the set X into two subsets R and S is also a major challenge. A potential approach to choose best possible S set is also provided in this work.

3 Statistical Delay Prediction

3.1 SSTA Framework

STA provides single values for a delay using corner based method. However, this approach is very pessimistic and results into overdesign. In contrast, SSTA provides probability distribution of the delay instead of a single value. Here, it is assumed that these distributions are Gaussian defined by mean (μ) and variance (σ^2) terms. However, later we will see some equations for Skewed Gaussian distributions which also requires shape parameter prediction.

Mean value (μ) is also called nominal value as it is the design value (i.e. value when process variations are not present). We define μ_k as the mean vector and σ_k^2 as the variance vector for all paths which need to be predicted (paths from set R). So we have mean and variance vectors as

$$\mu_k = \begin{bmatrix} \mu_{k1} \\ \mu_{k2} \\ \cdots \\ \mu_{kn} \end{bmatrix}$$

and

$$\sigma_k^2 = \begin{bmatrix} \sigma_{k1}^2 \\ \sigma_{k2}^2 \\ \cdots \\ \sigma_{kn}^2 \end{bmatrix}$$

Assume the number of elements of set S are N. All parameters from S are measured using frequency stepping on tester as described in [9]. If D_t is the vector containing measured delays and D_k is the delay vector to be predicted, then combined delay vector D can be written as

$$D = \begin{bmatrix} D_k \\ D_t \end{bmatrix}$$

As these delays follow Gaussian distribution D $\sim$ N(μ, Σ^2), where μ is the mean vector of D and Σ is the covariance matrix of D. Individually, $D_k \sim N(\mu_k, \sigma_k^2)$ and $D_t \sim N(\mu_t, \sigma_t^2)$. Therefore μ and Σ can be written as

$$\mu = \begin{bmatrix} \mu_k \\ \mu_t \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} \sigma_k^2 & \Sigma_{k,t} \\ \Sigma_{t,k} & \Sigma_t \end{bmatrix}$$

with $\Sigma_{k,t} = \Sigma_{t,k}^T$.

D_t can be measured by using frequency stepping methods and distribution for D_k given D_t can be determined as another Gaussian distribution with mean and variance given as

$$\mu_k' = \mu_k + \Sigma_{k,t}\Sigma_t^{-1}(d_t - \mu_t) \tag{1}$$

$$\sigma_k'^2 = \sigma_k^2 - \Sigma_{k,t}\Sigma_t^{-1}\Sigma_{t,k} \tag{2}$$

3.2 Objective Function and Challenges

While various techniques have been suggested to determined conditional distributions using formulas given above, it is still challenging to split X into two parts: S and R. It is possible that while some parameters are highly correlated and others have very less correlation. Therefore, choosing which parameters should be measured and which should be predicted is challenging for large number of parameters. Since our purpose is to minimize the uncertainty associated with process variations, we need to focus on variance(σ^2) of predicted conditional distribution. Minimizing the variance means reducing effects of process variations. In general, keeping variance within a certain bound is acceptable and should result into significant yield improvement. We propose various possible approaches to deal with this problem. These are explained below along with the challenges that they pose. While first two methods are based on component selection, remaining three are clustering based techniques.

3.2.1 Parameter Selection Method

In this approach, we predict distribution in terms of selection variable $t_i's$. Assume t_i is the binary variable which represents whether a parameter x_i is measured or not (predicted). That means

$$t_i = \begin{cases} 1, & \text{if } x_i \text{ is measured} \\ 0, & \text{otherwise} \end{cases}$$

Then we represent parameter set A as follows:

$$A = \begin{bmatrix} K \\ T \end{bmatrix} = \begin{bmatrix} x_1(1-t_1) \\ x_2(1-t_2) \\ .. \\ x_n(1-t_n) \\ x_1t_1 \\ x_2t_2 \\ .. \\ x_nt_n \end{bmatrix} \tag{3}$$

where $K = \begin{bmatrix} x_1(1-t_1) \\ x_2(1-t_2) \\ .. \\ x_n(1-t_n) \end{bmatrix}$ is the matrix of predicted parameters and $T = \begin{bmatrix} x_1t_1 \\ x_2t_2 \\ .. \\ x_nt_n \end{bmatrix}$ is the matrix of measured parameters.

Conditional variance (i.e. variance of probability distribution of K given T) can be predicted using

$$\sigma_k'^2 = \sigma_k^2 - \Sigma_{k,t}\Sigma_t^{-1}\Sigma_{t,k} \tag{4}$$

Now, we need to ensure that predicted variance is within a certain bound with minimum number of parameters being measured. Mathematically,

$$minimize \sum_{i=1}^{n} t_i \quad \text{such that} \quad \sigma_k'^2 \leq c \quad \forall k \in [1, n]$$

where c is maximum variance threshold.

The challenge here is that finding matrix inverse Σ_t^{-1} as in equation (4) and therefore variance prediction have to be done in terms of $t_i's$ to formulate constraints for SAT solver. This results into very complex expressions where complexity increases exponentially as we increase number of parameters.

3.2.2 Independent Component Analysis

In computational methods, Independent Component Analysis (ICA) is used to split a multivariate signal into several independent components. In one system, various parameters have high probability of having some correlated components that influence the value of parameters measured. Such components give rise to interdependence among various parameters and result into high mutual information. However, to minimize measurement efforts, it is desirable that only independent components are measured using tester and correlation information of parameters can be used to estimate shared components. Each component has its own variation and therefore, as uncertainty in individual components reduces, variation in the parameter to be measured reduces.

Let us assume that a random variable x_1 from parameter vector X is split into multiple independent components $[d_1, d_2 ... d_n]^T$ and coefficient vector C is $[c_1, c_2 ... c_n]^T$. Therefore, parameter x can be determined from vectors C and D as shown below:

$$x_1 = C^T D = c_1 d_1 + c_2 d_2 + ... c_n d_n$$

Now similarly, other parameters such as $x_2, x_3 ...$ can also be split using ICA into a linear combination of independent components. We know that, although individual components from vector D are independent from each other (by definition), some of these components are shared among various x's as x_i's so that they become correlated. Therefore, if we can measure these shared components (d_i's), the variability of other parameters ($x_2, x_3 ...$) can be reduced.

Here challenge of measuring independent components remains, because it is not clear how to identify components (d_i's) that are mutually shared among multiple parameters and how to separately measure them.

3.2.3 Minimum Clustering Method

In this method, covariance based clustering is done for each parameter. For each parameter, we find a group of all other parameters having certain minimum covariance with this parameter. Thus, we obtain n overlapping sets. Out of these n sets, we choose sets such that whole parameter range is covered while minimizing the number of chosen sets. If t_i is the binary variable indicating the selection of a set such that

$$t_i = \begin{cases} 1, & \text{if set } i \text{ is selected} \\ 0, & otherwise. \end{cases}$$

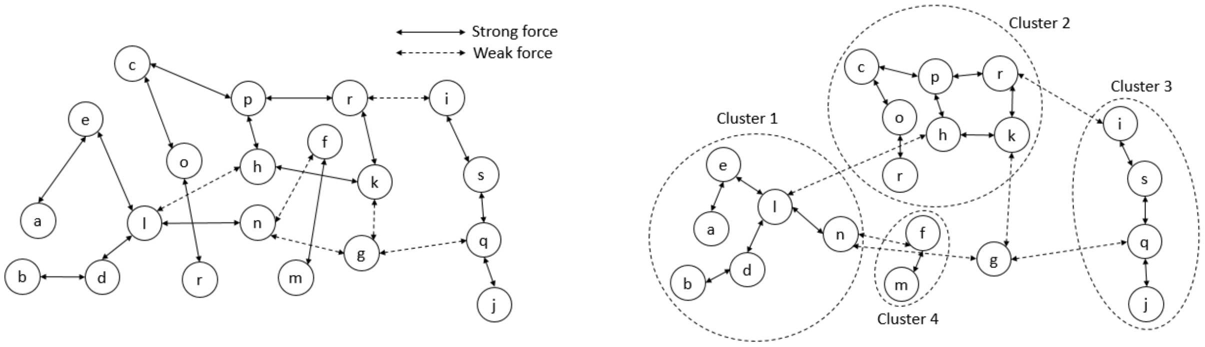
Then our objective is to minimize $\sum_{i=1}^{n} t_i$ such that union of all selected sets gives set A ($\cup_{i=1}^{n} = A$). This guarantees that we measure minimum number of variable such that maximum coverage is obtained. Furthermore, from the cluster obtained, if there are clusters with only one element, that implies that those parameters need to be measured individually. Sufficient information about these parameters is not obtained from other parameters (negligible correlation).

It is, however, possible that some variables show less correlation with other elements and are not fit for cluster formation. Such outliers can deteriorate the results obtained from above formulation. To handle such elements, we can further add constraint to maximize the length of each cluster. To avoid scenario with all clusters having an average length, penalty can be added to all clusters with less than a fixed number of elements.

3.2.4 Force Based Clustering

Force based clustering algorithm models the problem as a mechanical system where various parameters are assumed as particles and covariance between two parameters is a measure of attractive force among them. If we isolate all particles from external environment and put them in a closed system, slowly all particles will arrange themselves in a position of equilibrium where all forces will be balanced. That means total force acting on each particle will be zero under equilibrium. This will result into clustering as parameters with high covariance will stabilize closer together than ones with low covariance.

This process will result into arrangement of all parameters into multiple clusters depending upon covariance. In Figure3a, all particles are unorganized in the beginning and in Figure 3b, all particles organize themselves in clusters. Some elements might not fall within cluster thresholds and therefore need to be measured separately (e.g. node g in Figure 3b). If a parameter has equal covariance with two other parameters belonging to different clusters, they can be put in one group depending upon their vicinity with other elements of that group.



(a)

(b)

Fig. 3: Force Based Clustering showing (a.) Initial placement of all parameters in system irrespective of their covariance with other parameters. (b.) Final equilibrium position of parameters classifying them in different clusters.

3.2.5 Heuristics Based Method

In a heuristics based method, for each parameter, a list of all parameters with certain minimum covariance is maintained. This gives n overlapping sets. First we select the cluster with maximum coverage (largest set). Now remove this set and all the sets corresponding to its elements from potential clusters. Repeat this process until full coverage is achieved, i.e., all selected clusters when combined give set A.

$$\sum_{i=1}^{n} t_i \leq c$$

$$\cup_{i=1}^{k} = A$$

Although this method does not guarantee optimal clustering, it provides a fairly small number of clusters with very good coverage. This algorithm is fast and can handle large number of parameters.

From all cluster based methods, we can find the parameters with maximum neighbors and measure those parameters to predict all other parameters directly linked to them. Once we have found the suitable parameters to be measured, other parameters can be predicted using these parameters. For parameters with Gaussian distribution, mean $(\mu'_k)$and variance (σ'^2_k) can be predicted using following equations:

$$\mu'_k = \mu_k + \Sigma_{k,t}\Sigma_t^{-1}(d_t - \mu_t) \tag{5}$$

$$\sigma'^2_k = \sigma_k^2 - \Sigma_{k,t}\Sigma_t^{-1}\Sigma_{t,k} \tag{6}$$

For Skewed Gaussian Distribution, measurement of mean $(\mu'_k)$and variance (σ'^2_k) remains same, but we also need to find the skew parameter. However, for simplicity, we limit our discussion to Gaussian Distributions only. If needed, prediction of skew parameter can be done using formulas given in [11] where in addition to mean (μ) and variance (σ) information, skew of measured parameters also needs to be determined beforehand. Refer to appendix for more information.

4 Experiments

We have implemented heuristic method using C++. Minimum clustering method can also be easily implemented with the help of a SAT solver. Heuristic method implemented using C++ is computational inexpensive and gives fairly good results. For some sample data, Table 1 shows the number of clusters obtained.

TABLE 1: Number of clusters using Heuristic Clustering

Correlation Group	Number of Parameters	Number of Clusters
ac97_ctrl_syn	464	7
mem_ctrl_syn	2706	8
pcibridge32_syn	2508	12
s9234	64	1
s13207	384	1
s15850	327	2
s38584	314	1
usbfunct_syn	450	9

It must be noted that for each cluster we need to measure only one parameter. Therefore, number of clusters obtained from a given set of parameters is also the number of measurements we need to do using tester. From Table 1, we can see that we are left with far less parameters to be measured than total number of parameters. Furthermore, it is also identified that it is the first parameter of each cluster that exhibits a covariance greater than our threshold value with all the other parameters of the same cluster. Therefore, we have to measure only first parameter of each cluster using tester and all the other parameters of the same cluster can be predicted.

5 Conclusion

There are various techniques suggested here in this report. However, actual implementation of these techniques poses certain challenges. Without solving these challenges, it is not possible to implement these approaches. Due to time limitations, only work on clustering has been furnished. From the Table 1, it can be seen that number of parameters that need to be measured according to our proposal is much lower than total number of parameters in the chip. In future, these techniques can be appropriately modified to achieve the desired objective. Also as far as parameter measurement using tester is concerned, delay alignment can be used to reduce the number of measurements further.

Appendix A
Skewed Gaussian Conditional Distributions

This has been discussed in detail in [11]. Section 5.2 in [11] discusses conditional distributions with due consideration to shape parameter that appears in Skewed Gaussian Distribution.

Suppose Y is a Skewed Gaussian distributed function with Y_1 and Y_2 components having corresponding μ, σ and α partitions.

Now if Y_1 is measured, then Y_2 can be predicted if there is a correlation between them.

Therefore, distribution of Y_2 conditioned on $Y_1 = y_1$ is given as:

$$\mu'_k = \mu_k + \Sigma_{k,t}\Sigma_t^{-1}(d_t - \mu_t)$$

$$\sigma_k'^2 = \sigma_k^2 - \Sigma_{k,t}\Sigma_t^{-1}\Sigma_{t,k}$$

$$\alpha'_k = (\alpha_1 + \omega_1\Sigma_t^{-1}\Sigma_{t,k}\omega_2^{-1}\alpha_2)/(1 + \alpha_2^T\bar{\Sigma}_{k.1}\alpha_2)^{1/2}$$

where

$$\omega_1 = \sqrt{\Sigma_t}$$

$$\omega_2 = \sqrt{\Sigma_k}$$

and

$$\bar{\Sigma}_{k.1} = \omega_2^{-1}\Sigma_{k.1}\omega_2^{-1}$$

Here μ'_k and $\sigma_k'^2$ are determined by normal conditional formulas as shown various sections. However, α'_k is shape parameter needed for finding shape of marginal distribution of Y_1.

Refer to [11] for more detailed discussion and parameter handling.

ACKNOWLEDGMENTS

I would like to thank my supervisor Dr. Bing Li for the continuous support and guidance. It would not have been possible to work on such challenging topic without his supervision. He has provided crucial guidance and helped understand a lot of concepts which were completely new to me.

References

[1] B. Li, N. Chen and U. Schlichtmann, *Efficient Delay Test and Statistical Prediction for Configuring Post-Silicon Tunable buffers*, Design Automation Conference (DAC), 2016.

[2] Q. Liu and S. Sapatnekar, *A framework for scalable postsilicon statistical delay prediction under process variations*, Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, vol. 28, no. 8, pp. 1201-1212, August 2009.

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

[4] J. Tsai, D. Baik, C. C.-P. Chen, and K. K. Saluja, *A yield improvement methodology using pre- and post-silicon statistical clock scheduling*, In Proc. Int. Conf. Comput.-Aided Des., pages 611-618, 2004.

[5] J. Tsai, L. Zhang and C.C.-P. Chen, *Statistical timing analysis driven post-silicon-tunable clock-tree synthesis*, Proc. Int. Conf. Comput.-Aided Des., pp. 575-581, 2005.

[6] V. Khandelwal and A. Srivastava, *Variability-driven formulation for simultaneous gate sizing and post-silicon tunability allocation*, In Proc. Int. Symp. Phys. Des., pages 11-18, 2007.

[7] K. Nagaraj and S. Kundu, *A study on placement of post silicon clock tuning buffers for mitigating impact of process variation*, Proc. Design Autom. and Test Europe Conf., pp. 292-295, 2009.

[8] Z. Lak and N. Nicolici, *A novel algorithmic approach to aid post-silicon delay measurement and clock tuning*, IEEE Trans. Comput., vol. 63, no. 5, pp. 1074-1084, May 2014.

[9] K. Nagaraj and S. Kundu, *An automatic post silicon clock tuning system for improving system performance based on tester measurements*, Proc. Int. Test. Conf., pp. 1-8, 2008.

[10] Q. Liu and S. Sapatnekar, *Capturing post-silicon variations using a representative critical path*, IEEE Trans. Comput.-Aided Des., vol. 29, no. 2, pp. 211-222, 2010.

[11] A. Azzalini and A. Capitanio, *Statistical applications of the multivariate skew-normal distributions*, J. R. Statist. Soc. B 61(1999), 579-602, February 1998.

[12] B. Li, N. Chen, and U. Schlichtmann, *Fast statistical timing analysis for circuits with post-silicon tunable clock buffers* In Proc. Int. Conf. Comput.-Aided Des., pages 111-117, 2011.

[13] D. Tadesse, J. Grodstein, and R. I. Bahar, *AutoRex: An automated post-silicon clock tuning tool* In Proc. Int. Test Conf., pages 1-10, 2009.