# Chapter 4
# Statistical Design of Integrated Circuits

**Sachin S. Sapatnekar**

**Abstract** The presence of process variations makes it imperative to depart from the traditional corner-based methodology and migrate to statistical design techniques. In this chapter, based on a set of variational models that capture correlated as well as uncorrelated variations, we present techniques for presilicon statistical timing and power analysis to determine the performance spread over a population of manufactured parts. In order to improve this spread, we discuss presilicon statistical optimization techniques that incorporate appropriate margins to enable improved manufacturing yield. At the post-silicon stage, we then present how a set of compact sensors may be used to predict the delay of a manufactured part, with known confidence, through a small set of measurements on the sensors: such data can then be used to drive adaptive post-silicon tuning approaches that are individualized to each manufactured part.

## 4.1 Introduction

As feature sizes have moved into tens of nanometers, it has become widely accepted that design tools must account for parameter variations during manufacturing. These considerations are important during both circuit analysis and optimization, in the presilicon as well as the post-silicon phases, and are essential to ensure circuit performance and manufacturing yield. These sources of variation can broadly be categorized into three classes:

- *Process variations* result from perturbations in the fabrication process, due to which the nominal values of parameters such as the effective channel length ($L_{\mathrm{eff}}$), the oxide thickness ($t_{\mathrm{ox}}$), the dopant concentration ($N_{\mathrm{a}}$), the transistor width

S.S. Sapatnekar (✉)
Department of Electrical and Computer Engineering, University of Minnesota,
Minneapolis, MN, USA
e-mail: sachin@umn.edu

S. Bhunia, S. Mukhopadhyay (eds.), *Low-Power Variation-Tolerant Design in Nanometer Silicon*, DOI 10.1007/978-1-4419-7418-1_4,

($w$), the interlayer dielectric (ILD) thickness ($t_{\text{ILD}}$), and the interconnect height and width ($h_{\text{int}}$ and $w_{\text{int}}$, respectively).

- *Environmental variations* arise due to changes in the operating environment of the circuit, such as the temperature or variations in the supply voltage ($V_{\text{dd}}$ and ground) levels or soft errors. There is a wide body of work on analysis techniques to determine environmental variations, both for thermal issues and voltage drop, and a reasonable volume on soft errors.
- *Aging variations* come about due to the degradation of the circuit during its operation in the field. These variations can result in changes in the threshold voltage over time, or catastrophic failures due to prolonged stress conditions.

All of these types of variations can result in changes in the timing and power characteristics of a circuit. Process variations, even random ones, are fully determined when the circuit is manufactured and do not change beyond that point. Therefore, a circuit that experiences large variations can be discarded after manufacturing test, at the cost of yield loss. An optimization process can target the presilicon maximization of yield over the entire population of die, or a post-silicon repair mechanism. On the other hand, environmental variations may appear, disappear, and reappear in various parts of the circuit during its lifetime. Since the circuit is required to work correctly at every single time point during its lifetime and over all operating conditions, these are typically worst-cased. Aging variations are deterministic phenomena that can be compensated for by adding margins at the presilicon, or by adaptation at the post-silicon phase.

For these reasons, process variations are a prime target for statistical design that attempts to optimize the circuit over a range of random variations, while environmental and aging variations are not. The move to statistical design is a significant shift in paradigm from the conventional approach of deterministic design. Unlike conventional static timing analysis (STA) which computes the delay of a circuit at a specific process corner, statistical static timing analysis (SSTA) provides a probability density function (PDF)[1] of the delay distribution of the circuit over all variations. Similarly, statistical power analysis targets the statistical distribution of the power dissipation of a circuit.

Process parameter variations can be classified into two categories: across-die (also known as inter-die) variations and within-die (or intra-die) variations. Across-die variations correspond to parameter fluctuations from one chip to another, while within-die variations are defined as the variations among different locations within a single die. Within-die variations of some parameters have been observed to be spatially correlated, i.e., the parameters of transistors or wires that are placed close to each other on a die are more likely to vary in a similar way than those of transistors or wires that are far away from each other. For example, among the process parameters for a transistor, the variations of channel length $L_{\text{eff}}$ and transistor width $W$ are seen to have such spatial correlation structure, while parameter variations such as

[1] Equivalently, its integral, the cumulative density function (CDF), may be provided.

the dopant concentration $N_A$ and the oxide thickness $T_{ox}$ are generally considered not to be spatially correlated.

If the only variations are across-die variations, as was the case in older technologies, then the approach of using process corners is very appropriate. In such a case, all variations on a die are similar, e.g., all transistor $L_{\text{eff}}$ values may be increased or decreased by a consistent amount, so that a worst-case parameter value may be applied. However, with scaling, the role of within-die variations has increased significantly. Extending the same example, such variations imply that some $L_{\text{eff}}$ values on a die may increase while others may decrease, and they may do so by inconsistent amounts. Therefore, worst-case corners are inappropriate for this scenario, and statistically based design has become important.

This chapter begins by overviewing models for process variations in Section 4.2. Next, we survey a prominent set of techniques for statistical timing and power analysis in Sections 4.3 and 4.4, respectively. Presilicon optimization methods are outlined in Section 4.5, and statistically based sensing techniques are described in Section 4.6.

## 4.2 Mathematical Models for Process Variations

### *4.2.1 Modeling Variations*

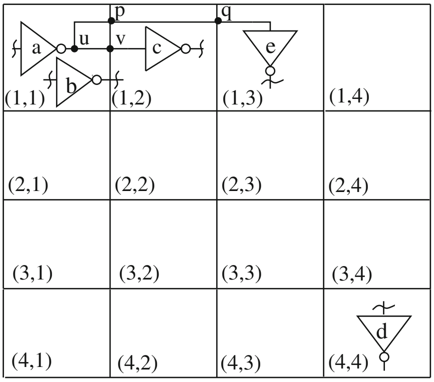
In general, the intra-chip process variation $\delta$ can be decomposed into three parts: a deterministic global component, $\delta_{\text{global}}$; a deterministic local component $\delta_{\text{local}}$; and a random component $\varepsilon$ [1]:

$$\delta = \delta_{\text{global}} + \delta_{\text{local}} + \varepsilon \tag{4.1}$$

The global component, $\delta_{\text{global}}$, is location-dependent, and several models are available in the literature to incorporate various known deterministic effects. The local component, $\delta_{\text{local}}$, is proximity-dependent and layout-specific. The random residue, $\varepsilon$, stands for the random intra-chip variation and is modeled as a random variable with a multivariate distribution $\varepsilon$ to account for the spatial correlation of the intra-chip variation. It is common to assume that the underlying distribution is Gaussian, i.e., $\varepsilon \sim N(0, \Sigma)$, where $\Sigma$ is the covariance matrix of the distribution. However, other distributions may also be used to model this variation. When the parameter variations are assumed to be uncorrelated, $\Sigma$ is a diagonal matrix; spatial correlations are captured by the off-diagonal cross-covariance terms in a general $\Sigma$ matrix. A fundamental property of covariance matrices says that $\Sigma$ must be symmetric and positive semidefinite.

To model the intra-die spatial correlations of parameters, the die region may be partitioned into $n\text{row} \times n\text{col} = n$ grids. Since devices or wires close to each other are more likely to have similar characteristics than those placed far away, it is reasonable to assume perfect correlations among the devices (wires) in the same grid,

**Fig. 4.1** Grid model for spatial correlations [2]



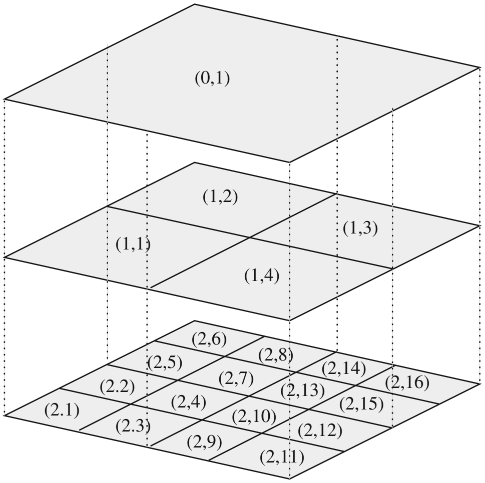
high correlations among those in close grids and low or zero correlations in faraway grids. For example, in Fig. 4.1, gates *a* and *b* (whose sizes are shown to be exaggeratedly large) are located in the same grid square, and it is assumed that their parameter variations (such as the variations of their gate length), are always identical. Gates *a* and *c* lie in neighboring grids, and their parameter variations are not identical but are highly correlated due to their spatial proximity. For example, when gate *a* has a larger than nominal gate length, it is highly probable that gate *c* will have a larger than nominal gate length, and less probable that it will have a smaller than nominal gate length. On the other hand, gates *a* and *d* are far away from each other, and their parameters are uncorrelated; for example, when gate *a* has a larger than nominal gate length, the gate length for *d* may be either larger or smaller than nominal.

Under this model, a parameter variation in a single grid at location $(x,y)$ can be modeled using a single random variable $p(x,y)$. For each type of parameter, $n$ random variables are needed, each representing the value of a parameter in one of the $n$ grids.

In addition, it is reasonable to assume that correlation exists only among the same type of parameters in different grids and there is no correlation between different types of parameters. For example, the $L_{\mathrm{g}}$ values for transistors in a grid are correlated with those in nearby grids, but are uncorrelated with other parameters such as $T_{\mathrm{ox}}$ or $W_{\mathrm{int}}$ in any grid. For each type of parameter, an $n \times n$ covariance matrix, $\Sigma$, represents the spatial correlations of such a structure.

An alternative model for spatial correlations was proposed in [3, 4]. The chip area is divided into several regions using multiple quad-tree partitioning, where at level $l$, the die area is partitioned into $2^l \times 2^l$ squares; therefore, the uppermost level has just one region, while the lowermost level for a quad-tree of depth $k$ has $4^k$ regions. A three-level tree is illustrated in Fig. 4.2. An independent random variable, $\Delta p_{i,r}$, is associated with each region $(i,r)$ to represent the variations in parameter $p$ in the region at level $r$. The total variation at the lowest level is then taken to be the sum of the variations of all squares that cover a region.

**Fig. 4.2** The quadtree model for spatially correlated variations [3]



For example, in Fig. 4.2, in region (2,1), if $p$ represents the effective gate length due to intra-die variations, $\Delta L_{\mathrm{eff}}(2,1)$, then

$$\Delta L_{\mathrm{eff}}(2,1) = \Delta L_{0,1} + \Delta L_{1,1} + \Delta L_{2,1} \tag{4.2}$$

In general, for region $(i, j)$,

$$\Delta p(i,j) = \sum_{0<l<k,(l,r)\,\mathrm{covers}\,(i,j)} \Delta p_{l,r} \tag{4.3}$$

It can be shown rather easily that this is a special case of the model of Fig. 4.1, and has the advantage of having fewer characterization parameters. On the other hand, it shows marked edge effects that result in smaller correlations between adjacent cells if they fall across the edges of early levels of the quad-tree than those that do not.

Several approaches for characterizing spatial variations have been presented in the literature. The traditional approach is based on Pelgrom's model [5], which provides a closed-form structure for the variance of process parameters, and is widely used by analog designers to model device mismatch. In [6], a technique for fitting process data was presented, with a a guarantee that the resulting covariance matrix is positive definite. In [7], the notion behind Pelgrom's model is generalized using the idea of variograms to come up with a distance-based correlation model. An alternative radially symmetric spatial correlation model, based on hexagonal cells, was presented in [8].

### 4.2.2 Gaussian Models and Principal Components

When the underlying variations are Gaussian in nature, they are completely specified by a mean vector and a covariance matrix, $\Sigma$. However, working with correlated random variables involves considerable computation, and this can be reduced if the variables are orthogonalized into a basis set of independent random variables. Principal components analysis (PCA) techniques [9] convert a set of correlated random variables into a set of orthogonal uncorrelated variables in a transformed space; the PCA step can be performed as a preprocessing step for a design. As shown in [2], by performing this orthogonalization as a preprocessing step, once for each technology, the cost of SSTA can be significantly reduced. A variation on this theme is the idea of using the Kosambi-Karhunen-Loéve expansion [10], which allows correlations to be captured using a continuous, rather than a grid-based model and is useful for more fine-grained variations; indeed, PCA is sometimes referred to as the discrete osambi-Karhunen-Loéve transform.

Given a set of correlated random variables **X** with a covariance matrix $\Sigma$, the PCA method transforms the set **X** into a set of mutually orthogonal random variables, **P**, such that each member of **P** has zero mean and unit variance. The elements of the set **P** are called principal components in PCA, and the size of **P** is no larger than the size of **X**. Any variable $x_i \in \mathbf{X}$ can then be expressed in terms of the principal components **P** as follows:

$$x_i = \mu_i + \sigma_i \sum_{j=1}^{m} \sqrt{\lambda_i} \cdot v_{ij} \cdot p_j = \mu_i + \sum_{j=1}^{m} k_{ij} p_j \tag{4.4}$$

where $p_{ij}$ is a principal component in set **P**, $\lambda_i$ is the $i$th eigenvalue of the covariance matrix $\Sigma$, $v_{ij}$ is the $i$th element of the $j$th eigenvector of $\Sigma$, and $\sigma_i$ and $\mu_i$ are, respectively, the mean and standard deviation of $x_i$. The term $k_{ij}$ aggregates the terms that multiply $p_j$.

Since all of the principal components $p_i$ that appear in Equation (4.4) are independent, the following properties ensue:

- The variance of $d$ is given by

$$\sigma_{x_i}^2 = \sum_{i=1}^{m} k_{ij}^2 \tag{4.5}$$

- The covariance between $x_i$ and any principal component $p_j$ is given by

$$\mathrm{cov}(x_i, p_j) = k_{ij} \sigma_{p_j}^2 = k_{ij} \tag{4.6}$$

- For two random variables, $x_i$ and $x_l$ are given by

$$x_i = \mu_i + \sum_{j=1}^{m} k_{ij} p_j$$

$$x_l = \mu_l + \sum_{j=1}^{m} k_{lj} p_l$$

The covariance of $x_i$ and $x_l$, $\mathrm{cov}(x_i, x_l)$ can be computed as

$$\mathrm{cov}(x_i, x_l) = \sum_{j=1}^{m} k_{ij} k_{lj} \tag{4.7}$$

> In other words, the number of multiplications is linear in the dimension of the space, since orthogonality of the principal components implies that the products of terms $k_{ir}$ and $k_{js}$ for $r \neq s$ need not be considered.

If we work with the original parameter space, the cost of computing the covariance is quadratic in the number of variables; instead, Equation (4.7) allows this to be computed in linear time. This forms the heart of the SSTA algorithm proposed in [2], and enables efficient SSTA.

### 4.2.3 Non-Gaussian Models and Independent Components

Non-Gaussian variations may be represented by a specific type of distribution in closed-form, or by a set of moments that characterize the distribution. These cases are indeed seen in practice: for example, the dopant density, $N_{\mathrm{d}}$, can be modeled using a Poisson distribution. SSTA methods that work on non-Gaussians are generally based on moment-based formulations, and therefore, a starting point is in providing the moments of the process distribution.

Consider a process parameter represented by a random variable $x_i$: let us denote its $k$th moment by $m_k(x_i) = E[x_i^k]$. We consider three possible cases:

*Case I*: If the closed-form of the distribution of $x_i$ is available and it is of a standard form (e.g., Poisson or uniform), then $m_k(x_i)$ $\forall$ $k$ can be derived from the standard mathematical tables of these distributions.

*Case II*: If the distribution is not in a standard form, then $m_k(x_i)$ $\forall$ $k$ may be derived from the moment generating function (MGF) if a continuous closed-form PDF of the parameter is known. If the PDF of $x_i$ is the function $f_{x_i}(x_i)$, then its moment generating function $M(t)$ is given by

$$M(t) = E[\mathrm{e}^{tx_i}] = \int_{-\infty}^{\infty} \mathrm{e}^{tx_i} f_{x_i}(x_i) dx_i \tag{4.8}$$

The $k$th moment of $x_i$ can then be calculated as the $k$th order derivative of $M(t)$ with respect to $t$, evaluated at $t = 0$. Thus, $m_k(x_i) = \frac{d^k M(t)}{dt^k}$ at $t = 0$.

*Case III*: If a continuous closed-form PDF cannot be determined for a parameter, the moments can still be evaluated from the process data files as:

$$m_k(x_i) = \sum_x x^k Pr(X_i = x) \tag{4.9}$$

where $\Pr(x_i = x)$ is the probability that the parameter $x_i$ assumes a value $x$.

For variations that are not Gaussian-distributed, it is possible to use the independent component analysis method [11, 12] to orthogonalize the variables, enabling an SSTA solution that has a reasonable computational complexity [13].

## 4.3 Statistical Timing Analysis

The problem of SSTA is easily stated: given the underlying probability distributions of the process parameters, the goal of SSTA is to determine the probability distribution of the circuit delay. Most often, this task is divided into two parts: first, translating process variations into a gate-level probabilistic delay model, and then obtaining the circuit delay distribution.

Algorithms for SSTA can be classified according to various systems of taxonomy.

- *Path-based vs. block-based methods*: Path-based methods [3, 14] attempt to find the probability distribution of the delay on a path-by-path basis, and eventually performing a "max" operation to find the delay distribution of the circuit. If the number of paths to be considered is small, these methods can be effective, but in practice, the number of paths may be exponential in the number of gates. In contrast, block-based methods avoid path enumeration by performing a topological traversal, similar to that used by the critical path method (CPM), which processes each gate once when information about all of its inputs is known. While early approaches were predominantly path-based, state-of-the-art methods tend to operate in a block-based fashion.
- *Discrete vs. continuous PDFs*: SSTA methods can also be classified by their assumptions about the underlying probability distributions. Some approaches use discrete PDFs [15–17] while others are based on continuous PDFs; the latter class of techniques tend to dominate in the literature, although the former are capable of capturing a wider diversity of distributions, and may even directly use sample points from the process.
- *Gaussian vs. non-Gaussian models*: The class of continuous PDFs can be further subdivided into approaches that assume Gaussian (or normal) parameters, and those that permit more general non-Gaussian models.
- *Linear vs. nonlinear delay models*: Under small process perturbations, it is reasonable to assume that the change in gate delays follows a linear trend. However, as these perturbations grow larger, a nonlinear model may be necessary. Depending on which of these is chosen as the underlying model, the corresponding algorithm can incur smaller or larger computational costs.

The basic Monte Carlo method is probably the simplest method for performing statistical timing analysis. Given an arbitrary delay distribution, the method generates sample points and runs a static timing analyzer at each such point, and aggregates the results to find the delay distribution. The advantages of this method lie in its ease of implementation and its generality in being able to handle the complexities of variations and a wider range of delay models. For example, spatial correlations are easily incorporated, since all that is required is the generation of a sample point on a correlated distribution. Such a method is very compatible with the data brought in from the fab line, which are essentially in the form of sample points for the simulation. Its major disadvantage can be its extremely large runtimes. Recent work on SSTA has moved towards more clever and computationally efficient implementations [18–20]. Our discussion will largely focus on the faster and more widely used block-based SSTA methods that seek closed-form expressions for the delay at the output of each gate.

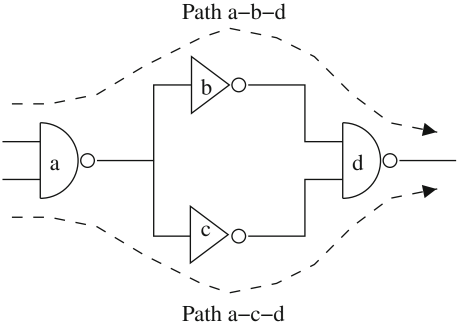


**Fig. 4.3** An example to illustrate structural correlations in a circuit

In addition to accounting for randomness, including spatial correlations, SSTA algorithms must also consider the effects of correlations between delay variables due to the structure of the circuit. Consider the reconvergent fanout structure shown in Fig. 4.3. The circuit has two paths, a-b-d and a-c-d. The circuit delay is the maximum of the delays of these two paths, and these are correlated since the delays of a and d contribute to both paths.

## *4.3.1 Modeling Gate/Interconnect Delay PDF's*

The variations in the process parameters translate into variations in the gate delays that can be represented as PDFs. Before we introduce how the distributions of gate and interconnect delays will be modeled, let us first consider an arbitrary function $d = f(\mathbf{P})$ that is assumed to be a function on a set of parameters $\mathbf{P}$, where each $p_i \in \mathbf{P}$ is a random variable with a known PDF. We can approximate $d$ using a Taylor series expansion:

$$d = d_0 + \sum_{\forall \text{parameters}\, p_i} \left[\frac{\partial f}{\partial p_i}\right]_0 \Delta p_i + \sum_{\forall \text{parameters}\, p_i} \left[\frac{\partial^2 f}{\partial p_i^2}\right]_0 \Delta p_i^2 + \cdots \tag{4.10}$$

where $d_0$ is the nominal value of $d$ calculated at the nominal values of parameters in the set $\mathbf{P}$, $\left[\frac{\partial f}{\partial p_i}\right]_0$ is computed at the nominal values of and $p_i$, and $\Delta p_i = p_i - \mu_{p_i}$ is a zero-mean random variable. This delay expression is general enough to handle the effects of input slews and output loads; for details, see [21].

If all of the parameter variations can be modeled by Gaussian distributions, i.e., $p_i \sim N(\mu_{p_i}, \sigma_{p_i})$, then clearly $\Delta p_i \sim N(0, \sigma_{p_i})$. If a first-order Taylor series approximation is used in Equation (4.10) by neglecting quadratic and higher order terms, then $d$ is a linear combination of Gaussians and is therefore Gaussian. Its mean $\mu_d$ and variance $\sigma_d^2$ are

$$\mu_d = d_0 \tag{4.11}$$

$$\sigma_d^2 = \sum_{\forall i} \left[\frac{\partial f}{\partial p_i}\right]_0^2 \sigma_{p_i}^2 + 2 \sum_{\forall i \neq j} \text{cov}(p_i, p_j) \tag{4.12}$$

where $\text{cov}(p_i, p_j)$ is the covariance of $p_i$ and $p_j$.

In cases where the variations are larger than can be accurately addressed by a linear model, then higher-order terms of the expansion should be maintained. Most such nonlinear models in the literature (e.g., [22–24]) find it sufficient to consider the linear and quadratic terms in the Taylor expansion.

## *4.3.2 Algorithms for SSTA*

### 4.3.2.1 Early Methods

Early work in this area spawned several methods that ignored the spatial correlation component, but laid the foundation for later approaches that overcame this limitation. Prominent among these was the work by Berkelaar in [25], [26], which presented a precise method for statistical static timing analysis that could successfully process large benchmarks circuits under probabilistic delay models. In the spirit of static timing analysis, this approach is purely topological and ignores the Boolean structure of the circuit. The underlying delay model assumes that each gate has a delay described by a Gaussian PDF, and observed that the essential operations in timing analysis can be distilled into two types:

*SUM*: A gate is processed when the arrival times of all inputs are known, at which time the candidate delay values at the output are computed using the "sum" operation that adds the delay at each input with the input-to-output pin delay.

*MAX*: The arrival time at the gate output is determined once these candidate delays have been found, and the "max" operation is applied to determine the maximum arrival time at the output.

The key to SSTA is to perform these two operations on operands that correspond to PDFs, rather than deterministic numbers as is the case for STA. Note that, as in STA, the SUM and MAX operators incorporate clock arrival times as well as signal arrival times.

Berkelaar's approach maintains an invariant that expresses all arrival times as Gaussians. As a consequence, since the gate delays are Gaussian, the "sum" operation is merely an addition of Gaussians, which is well known to be a Gaussian.

The computation of the "max" function, however, poses greater problems. The candidate delays are all Gaussian, so that this function must find the maximum of Gaussians. In general, the maximum of two Gaussians is *not* a Gaussian, but can be approximated as one. Intuitively, this can be justified by seeing that if $a$ and $b$ are Gaussian random variables, then

- if $a \gg b$, then $\max(a,b) = a$ is a Gaussian
- if $a = b$, then $\max(a,b) = a = b$ is a Gaussian

It was suggested in [25] that a statistical sampling approach could be used to approximate the mean and variance of the distribution; alternatively, this information could be embedded in look-up tables. In later work in [26], a precise closed-form approximation for the mean and variance, based on [27], was utilized.

#### 4.3.2.2 Incorporating Spatial Correlations

In cases where significant spatial correlations exist, it is important to take them into account. Figure 4.4 shows a comparison of the PDF yielded by an SSTA technique that is unaware of spatial correlations, as compared with a Monte Carlo simulation that incorporates these spatial correlations, and clearly shows a large difference. This motivates the need for developing methods that can handle these dependencies.

Early approaches to spatial correlation did not scale to large circuits. The work in [28] extended the idea of [25] to handle intra-gate spatial correlations, while assuming zero correlation between gates. A notable feature of this work was the use of an approximation technique from [29] that provides a closed-form formula to approximate the maximum of two correlated Gaussian random variables as a Gaussian.

Under normality assumptions, the approach in [2, 21] leverages the decomposition of correlated variations into principal components, as described in Section 4.2.2, to convert a set of correlated random variables into a set of uncorrelated variables in a transformed space. As mentioned earlier, the PCA step is to be performed once for each technology as a precharacterization. The worst-case complexity of the method in [2, 21] is $n$ times the complexity of CPM, where $n$ is the number of squares in the correlation grid (see Fig. 4.1). The overall CPU times for this method have been shown to be low, and the method yields high accuracy results.

This parameterized approach to SSTA propagates a canonical form (a term popularized in [30]) of the delay PDF, typically including the nominal value, a set

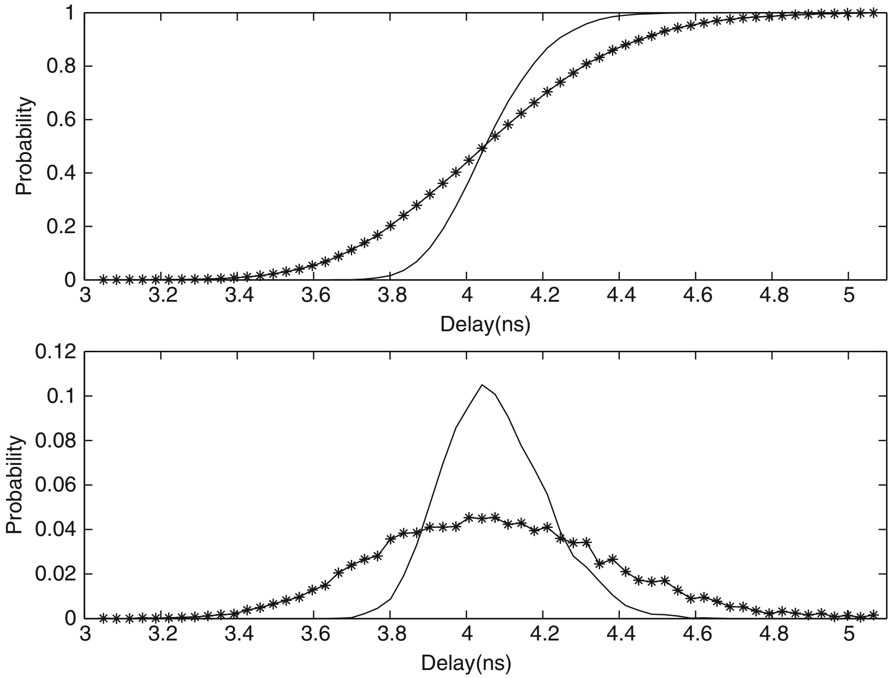


**Fig. 4.4** A comparison of the results of SSTA when the random variables are spatially correlated. The line on which points are marked with stars represents the accurate results obtained by a lengthy Monte Carlo simulation, and the the solid curve shows the results when spatial correlations are entirely ignored. The upper plot shows the CDFs, and the lower plot, the PDFs [2]

of normalized underlying independent sources of variation. For spatially correlated variations, these sources correspond to the principal components (PCs) [2], computed by applying PCA to the underlying covariance matrix of the correlated variations; uncorrelated variations are typically captured by a single independent random variable.

If the process parameters are Gaussian-distributed, then the $m$ PCs affect the statistical distribution of both the original circuit and the test structures on the same chip, and the canonical form for the delay $d$ is represented as

$$d = \mu + \sum_{i=1}^{m} a_i p_i + R = \mu + \mathbf{a}^{\mathrm{T}}\mathbf{p} + R \tag{4.13}$$

where $\mu$ is the mean of the delay distribution. The value of $\mu$ is also an approximation of its nominal value.[2] The random variable $p_i$ corresponds to the $i$th principal

[2] The nominal value of the delay of the circuit is the delay value when no parameter variations are present. This can be computed exactly by a conventional static timing analysis with all parameters

component, and is normally distributed, with zero mean and unit variance; note that $p_i$ and $p_j$ for $i \neq j$ are uncorrelated by definition, stemming from a property of PCA. The parameter $a_i$ is the first order coefficient of the delay with respect to $p_i$. Finally, $R$ corresponds to a variable that captures the effects of all the spatially uncorrelated variations. It is a placeholder to indicate the additional variations of the delay caused by the spatially uncorrelated variations, and cannot be regarded as a principal component.

Equation (4.13) is general enough to incorporate both inter-die and intra-die variations. It is well known that, for a spatially correlated parameter, the inter-die variation can be taken into account by adding a value $\sigma^2_{\text{inter}}$, the variance of inter-die parameter variation, to all entries of the covariance matrix of the intra-die variation of that parameter before performing PCA. The uncorrelated component $R$ accounts for contributions from both the inter-die and intra-die variations. Systematic variations affect only the nominal values and the PC coefficients in SSTA. Therefore, they can be accounted for by determining the shifted nominal values and sensitivities prior to SSTA, and computing the nominal values and PC coefficients in SSTA based on these shifted values.

The work in [2] uses this canonical form, along with the properties of such a principal components-based representation (as described in Equations (4.5) through (4.7) to perform SSTA under the general spatial correlation model of Fig. 4.1.

The fundamental process parameters are assumed to be in the form of correlated Gaussians, so that the delay given by Equation (4.10) is a weighted sum of Gaussians, which is Gaussian.

As in the work of Berkelaar, this method maintains the invariant that all arrival times are approximated as Gaussians, although in this case the Gaussians are correlated and are represented in terms of their principal components. Since the delays are considered as correlated Gaussians, the sum and max operations that underlie this block-based CPM-like traversal must yield Gaussians in the form of principal components.

We will first consider the case where $R$ in (Equation 4.13) is zero. The computation of the distribution of the sum function, $d_{\text{sum}} = \sum_{i=1}^{n} d_i$, is simple. Since this function is a linear combination of normally distributed random variables, $d_{\text{sum}}$ is a normal distribution whose mean, $\mu_{d\text{sum}}$, and variance, $\sigma^2_{d\text{sum}}$, are given by

$$\mu_{d_{\text{sum}}} = \sum_{i=1}^{n} d_i^0 \tag{4.14}$$

$$\sigma^2_{d_{\text{sum}}} = \sum_{j=1}^{m} \sum_{i=1}^{n} k_{ij}^2 \tag{4.15}$$

at their nominal values. However, because of the approximation of the max operation in the statistical timer, the mean value computed from the topological traversal is more compatible with the rest of the canonical form.

where $d_i$ is written in terms of its normalized principal components as $d_i^0 + \sum_{j=1}^{m} k_{ij}p_j$.

Strictly speaking, the max function of $n$ normally distributed random variables, $d_{\max} = \max(d_1, \cdots, d_n)$, is not Gaussian; however, as before, it is approximated as one. The approximation here is in the form of a correlated Gaussian, and the procedure in [29] is employed. The result is characterized in terms of its principal components, so that it is enough to find the mean of the max function and the coeficients associated with the principal components.

Although the above exposition has focused on handling spatially correlated variables, it is equally easy to incorporate uncorrelated terms in this framework. Only spatially correlated variables are decomposed into principal components, and any uncorrelated variables are incorporated into the uncorrelated component, $R$, of (Equation 4.13); during the sum and max operations, the uncorrelated components of the operands are consolidated into a single uncorrelated component of the canonical form of the result. For a detailed description of the sum and max operations, the reader is referred to [21].

The utility of using principal components is twofold:

- As described earlier, it implies that covariance calculations between paths are of linear complexity in the number of variables, obviating the need for the expensive pair-wise delay computation methods used in other methods.
- In the absence of the random component, $R$, in (Equation 4.13), structural correlations due to reconvergent fanouts (see Fig. 4.3) are automatically accounted for, since all the requisite information required to model these correlations is embedded in the principal components. When $R$ is considered, the structural components associated with $R$ are lumped together and individual variational information is lost, leading to a slight degradation of accuracy. However, heuristic methods may be used to limit this degradation.

The overall flow of the algorithm is shown in Fig. 4.5. To further speed up the process, several techniques may be used:

**Input**: Process parameter variations
**Output**: Distribution of circuit delay

1. Partition the chip into $n = nrow \times ncol$ grids , each modeled by spatially correlated variables.
2. For each type of parameter, determine the $n$ jointly normally distributed random variables and the corresponding covariance matrix.
3. Perform an orthogonal transformation to represent each random variable with a set of principal components.
4. For each gate and net connection, model their delays as linear combinations of the principal components generated in step 3.
5. Using "sum" and "max" functions on Gaussian random variables, perform a CPM-like traversal on the graph to find the distribution of the statistical longest path. This distribution achieved is the circuit delay distribution.

**Fig. 4.5** Overall flow of the PCA-based statistical timing analysis method

1. Before running the statistical timing analyzer, one run of deterministic STA is performed to determine loose bounds on the best-case and worst-case delays for all paths. As in [31], any path whose worst-case delay is less than the best-case delay of the longest path will never be critical, and edges that lie only on such paths can safely be removed.
2. During the "max" operation of statistical STA, if the value of mean $+ 3 \cdot \sigma$ of one path has a lower delay than the value of mean $- 3 \cdot \sigma$ of another path, the max function can be calculated by ignoring the path with lower delay.

For the non-Gaussian case [13], the linear canonical form is similar to (Equation 4.13):

$$d = \mu + \mathbf{b}^{\mathrm{T}}\mathbf{x} + \mathbf{c}^{\mathrm{T}}\mathbf{y} + e.z \tag{4.16}$$

where $d$ is the random variable corresponding to a gate delay or an arrival time at the input port of a gate. The vector $\mathbf{x}$ corresponds to the non-Gaussian independent components, obtained from applying ICA to the non-Gaussian process parameter set, and $\mathbf{b}$ is the vector of first-order sensitivities of the delay with respect to these independent components. The Gaussian random variables are orthogonalized using PCA into the principal component vector, $\mathbf{y}$, and $\mathbf{c}$ is the corresponding linear sensitivity vector. Finally, $z$ is the uncorrelated parameter which may be a Gaussian or a non-Gaussian random variable, $e$ is the sensitivity with respect this. We assume statistical independence between the Gaussian and non-Gaussian parameters: this is a reasonable assumption as parameters with dissimilar distributions are likely to represent different types of variables and are unlikely to be correlated.

The work in [13] presents an approach that translates the moments of the process parameters to the moments of the principal and independent components in a precharacterization step that is performed once for each technology. Next, a moment-based scheme is used to propagate the moments through the circuit, using a moment-matching scheme similar to the APEX algorithm [32]. The sum and max operations are performed on the canonical form to provide a result in canonical form, with moment-matching operations being used to drive the engine that generates the canonical form.

## 4.4 Statistical Power Analysis

The power dissipation of a circuit consists of the dynamic power, the short-circuit power, and the leakage power. Of these, the leakage power is increasing drastically with technology scaling, and has already become a substantial contributor to the total chip power dissipation. Consequently, it is important to accurately estimate leakage currents so that they can be accounted for during design, and so that it is possible to effectively optimize the total power consumption of a chip.

The major components of leakage in current CMOS technologies are due to subthreshold leakage and gate tunneling leakage. For a gate oxide thickness, $T_{\mathrm{ox}}$, of over 20Å, the gate tunneling leakage current, $I_{\mathrm{gate}}$, is typically very small, while the subthreshold leakage, $I_{\mathrm{sub}}$, dominates other types of leakage in circuit. For this reason, early work on leakage focused its attention on subthreshold leakage. However, the gate tunneling leakage is exponentially dependent on gate oxide thickness, e.g., a reduction in $T_{\mathrm{ox}}$ of 2Å will result in an order of magnitude increase in $I_{\mathrm{gate}}$. While high-K dielectrics provide some relief, the long-term trends indicate that gate leakage is an important factor. Unlike dynamic and short-circuit power, which are relatively insensitive to process variations, the circuit leakage can change significantly due to changes in parameters such as the transistor effective gate length and the gate oxide thickness. Therefore, statistical power analysis essentially equates to statistical leakage analysis.

### 4.4.1 Problem Description

The total leakage power consumption of a circuit is input-pattern-dependent, i.e., the value differs as the input signal to the circuit changes, because the leakage power consumption, due to subthreshold and gate tunneling leakage, of a gate depends on the input vector state at the gate. As illustrated in [33], the dependency of leakage on process variations is more significant than on input vector states. Therefore, it is sufficient to predict the effects of process variations on total circuit leakage by studying the variation of average leakage current for all possible input patterns to the circuit. However, it is impractical to estimate the average leakage by simulating the circuit at all input patterns, and thus an input pattern-independent approach is more desirable.

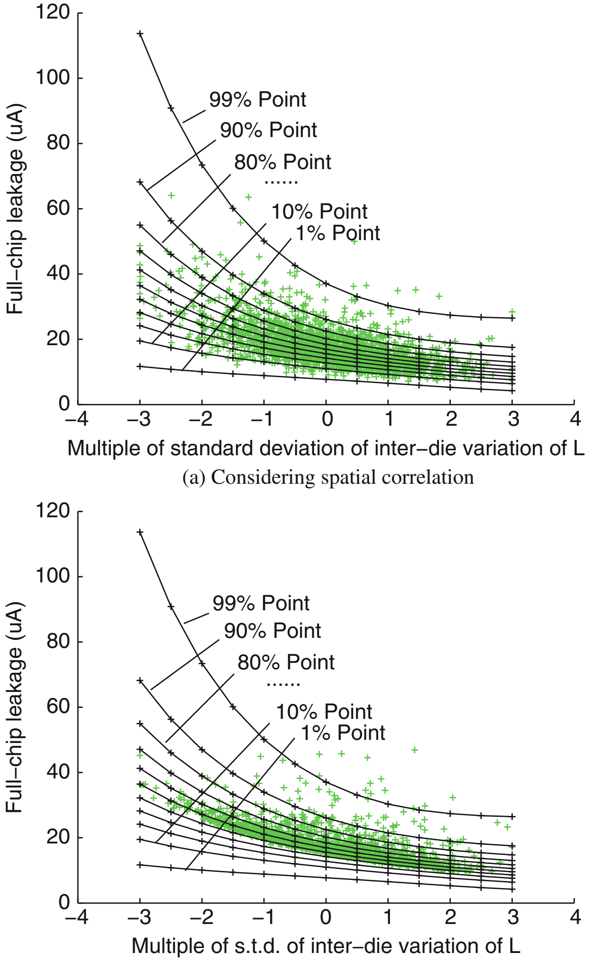
In switching power estimation, probabilistic approaches [34] have been used for this purpose. The work of [33] proposed a similar approach that computes the average leakage current of each gate and estimates the total average circuit leakage as a sum of the average leakage currents of all gates:

$$I_{\mathrm{tot}}^{\mathrm{avg}} = \sum_{k=1}^{N_{\mathrm{g}}} I_{\mathrm{leak},k}^{\mathrm{avg}} = \sum_{k=1}^{N_{\mathrm{g}}} \sum_{\forall \mathrm{vec}_{i,k}} \mathrm{Prob}(\mathrm{vec}_{i,k}) \cdot I_{\mathrm{leak},k}(\mathrm{vec}_{i,k}) \tag{4.17}$$

where $N_{\mathrm{g}}$ is the total number of gates in the circuit, $I_{\mathrm{leak},k}^{\mathrm{avg}}$ is the average leakage current of the $k$th gate, $\mathrm{vec}_{i,k}$ is the $i$th input vector at the $k$th gate, $\mathrm{Prob}(\mathrm{vec}_{i,k})$ is the probability of occurrence of $\mathrm{vec}_{i,k}$, and $I_{\mathrm{leak},k}(\mathrm{vec}_{i,k})$ is the leakage current of the $k$th gate when the gate input vector is $\mathrm{vec}_{i,k}$.

In our discussion, we consider the variations in the transistor gate length $L_{\mathrm{eff}}$ and gate oxide thickness $T_{\mathrm{ox}}$, since $I_{\mathrm{sub}}$ and $I_{\mathrm{gate}}$ are most sensitive to these parameters [35, 36]. To reflect reality, we model spatial correlations in transistor gate length, while the gate oxide thickness values for different gates are taken to be uncorrelated. Note that although only transistor gate length and gate oxide thickness are considered in this work, the framework is general enough to consider effects of any other types of process variations such as the channel dopant variation $N_{\mathrm{d}}$.

**Fig. 4.6** Comparison of scatter plots of full-chip leakage of circuit c432 considering and ignoring spatial correlation



(a) Considering spatial correlation

(b) Ignoring spatial correlation

In performing this computation, it is extremely important to consider the impact of spatial correlations. While random variations tend to cancel themselves out, spatially correlated variations magnify the extent of the variation. This difference can be visualized in Fig. 4.6, which shows the scatter plots for c432 for 2000 samples of full-chip leakage current generated by Monte Carlo simulations, with and without consideration of spatial correlations of $L_{\text{eff}}$. The x-axis marks the multiples of the standard deviation value of $\Delta L_{\text{eff}}^{\text{inter}}$, inter-die variations of effective gate length, ranging from −3 to +3, since a Gaussian distribution is assumed. The y-axis are the values of total circuit leakage current. Therefore, at each specific value of $\Delta L_{\text{eff}}^{\text{inter}}$, the scatter points list the various sampled values of total circuit leakage current due to variations in $T_{\text{ox}}$ and intra-die variation of $L_{\text{eff}}$. The plots also show a set of contour lines that correspond to, with the effect of spatial correlation taken into account, a set of percentage points of the cumulative density function (CDF) of total circuit leakage current at different values of $\Delta L_{\mathrm{eff}}^{\mathrm{inter}}$. In Fig. 4.6a, where spatial correlations are considered, nearly all points generated from Monte Carlo simulation fall between the contours of the 1 and 99% lines. However, in Fig. 4.6b, where spatial correlations are ignored, the spread is much tighter in general: the average value of 90% point of full-chip leakage, with spatial correlation considered, is 1.5 times larger than that without for $\Delta L_{\mathrm{eff}}^{\mathrm{inter}} \leq -1\sigma$; the same ratio is 1.1 times larger otherwise. Looking at the same numbers in a different way, in Fig. 4.6b, all points are contained between the 30 and 80% contours if $\Delta L_{\mathrm{eff}}^{\mathrm{inter}} \leq -1\sigma$. In this range, $I_{\mathrm{sub}}$ is greater than $I_{\mathrm{gate}}$ by one order of magnitude on average, and thus the variation of $L_{\mathrm{eff}}$ can have a large effect on the total leakage as $I_{\mathrm{sub}}$ is exponentially dependent on $L_{\mathrm{eff}}$. Consequently, ignoring spatial correlation results in a substantial underestimation of the standard deviation, and thus the worst-case full-chip leakage. For $\Delta L_{\mathrm{eff}}^{\mathrm{inter}} > -1\sigma$, $I_{\mathrm{sub}}$ decreases to a value comparable to $I_{\mathrm{gate}}$ and $L_{\mathrm{eff}}$ has a relatively weak effect on the variation of total leakage. In this range, the number of points of larger leakage values is similar to that when spatial correlation is considered. However, a large number of remaining points show smaller variations and are within the 20 and 90% contours, due to the same reasoning given above for $\Delta L_{\mathrm{eff}}^{\mathrm{inter}} \leq -1\sigma$.

### *4.4.2 Computing the Distribution of the Full-Chip Leakage Current*

The distribution of $I_{\mathrm{tot}}^{\mathrm{avg}}$ can be calculated in two steps. First, given the probability of each input pattern vector to a gate, $\mathrm{vec}_{i,k}$, we can compute the leakage of the gate as a weighted sum over all possible vectors. Second, this quantity can be summed up over all gates to obtain the total leakage. In other words,

$$I_{\mathrm{tot}}^{\mathrm{avg}} = \sum_{k=1}^{N_g} \sum_{\forall \mathrm{vec}_{i,k}} \mathrm{Prob}(\mathrm{vec}_{i,k}) \cdot \left(I_{\mathrm{sub},k}(\mathrm{vec}_{i,k}) + I_{\mathrm{gate},k}(\mathrm{vec}_{i,k})\right) \tag{4.18}$$

where $I_{\mathrm{leak},k}$ under vector ($\mathrm{vec}_{i,k}$) is written as the sum of the subthreshold leakage, $I_{\mathrm{sub},k}(\mathrm{vec}_{i,k})$, and the gate leakage, $I_{\mathrm{gate},k}(\mathrm{vec}_{i,k})$, for gate $k$.

The commonly used model for subthreshold leakage current through a transistor expresses this current as [35]:

$$I_{\mathrm{sub}} = I_0 \mathrm{e}^{(V_{\mathrm{gs}}-V_{\mathrm{th}})/n_{\mathrm{s}} V_T}(1 - \mathrm{e}^{-V_{\mathrm{ds}}/V_T}) \tag{4.19}$$

Here, $I_0 = \mu_0 C_{\mathrm{ox}}(W_{\mathrm{eff}}/L_{\mathrm{eff}})V_T^2 \mathrm{e}^{1.8}$, where $\mu_0$ is zero bias electron mobility, $C_{\mathrm{ox}}$ is the gate oxide capacitance, $W_{\mathrm{eff}}$ and $L_{\mathrm{eff}}$ are the effective transistor width and length, respectively, $V_{\mathrm{gs}}$ and $V_{\mathrm{ds}}$ are gate-to-source voltage and drain-to-source voltage, respectively, $n_{\mathrm{s}}$ is the subthreshold slope coefficient, $V_T = kT/q$ is the thermal voltage, where $k$ is Boltzman constant, $T$ is the operating temperature in Kelvin (K), $q$ is charge on an electron, and $V_{\mathrm{th}}$ is the subthreshold voltage.

It is observed that $V_{\mathrm{th}}$ is most sensitive to gate oxide thickness $T_{\mathrm{ox}}$ and effective transistor gate length $L_{\mathrm{eff}}$ due to short-channel effects [35]. Due to the exponential dependency of $I_{\mathrm{sub}}$ on $V_{\mathrm{th}}$, a small change on $L_{\mathrm{eff}}$ or $T_{\mathrm{ox}}$ will have a substantial effect on $I_{\mathrm{sub}}$. From this intuition, we estimate the subthreshold leakage current per transistor width by developing an empirical model through curve-fitting, similarly to [36, 37]:

$$I_{\mathrm{sub}} = c \times \mathrm{e}^{a_1 + a_2 L_{\mathrm{eff}} + a_3 L_{\mathrm{eff}}^2 + a_4 T_{\mathrm{ox}}^{-1} + a_5 T_{\mathrm{ox}}} \tag{4.20}$$

where $c$ and the $a_i$ terms are the fitting coefficients. To quantify the empirical model, the values of $I_{\mathrm{sub}}$ achieved from expression (Equation 4.20) are compared with those through SPICE simulations over a ranged values of $T_{\mathrm{ox}}$ and $L_{\mathrm{eff}}$.

Under process perturbations, $I_{\mathrm{sub}}$ can be well approximated by expanding its exponent $U$ using a first-order Taylor expansion at the nominal values of the process parameters:

$$I_{\mathrm{sub}} = c \times \mathrm{e}^{U_0 + \beta_1 \cdot \Delta L_{\mathrm{eff}} + \beta_2 \cdot \Delta T_{\mathrm{ox}}} \tag{4.21}$$

where $U^0$ is the nominal value of the exponent $U$, $\beta_0$ and $\beta_1$ are the derivatives of $U$ to $L_{\mathrm{eff}}$ and $T_{\mathrm{ox}}$ evaluated at their nominal values, respectively, and $\Delta L_{\mathrm{eff}}$ and $\Delta T_{\mathrm{ox}}$ are random variables standing for the variations in the process parameters $L_{\mathrm{eff}}$ and $T_{\mathrm{ox}}$, respectively.

Expression (Equation 4.21) for $I_{\mathrm{sub}}$ can also be written[3] as $\mathrm{e}^{\ln(c) + U_0 + \beta_1 \cdot \Delta L_{\mathrm{eff}} + \beta_2 \cdot \Delta T_{\mathrm{ox}}}$. Since $\Delta L_{\mathrm{eff}}$ and $\Delta T_{\mathrm{ox}}$ are assumed to be Gaussian-distributed, $I_{\mathrm{sub}}$ is seen as an exponential function of a Gaussian random variable, with mean $\ln(c) + U_0$ and standard deviation $\sqrt{\beta_1^2 \sigma_{L_{\mathrm{eff}}}^2 + \beta_2^2 \sigma_{T_{\mathrm{ox}}}^2}$, where $\sigma_{L_{\mathrm{eff}}}$ and $\sigma_{T_{\mathrm{ox}}}$ are standard deviations of $\Delta L_{\mathrm{eff}}$ and $\Delta T_{\mathrm{ox}}$, respectively.

In general, if $x$ is a Gaussian random variable, then $z = \mathrm{e}^x$ is a *lognormal random variable*. From (Equation 4.21), it is obvious that $I_{\mathrm{sub}}$ can be approximated as a lognormally distributed random variable whose probability density function can be characterized using the values of $c$, $U_0$, and $\beta_i$'s.

Since subthreshold leakage current has a well-known input state dependency due to the stack effect [38], the PDFs of subthreshold leakage currents must be characterized for all possible input states for each type of gate in the library, for which the same approach described in this section can be applied. Once the library is characterized, a simple look-up table (LUT) can then be used to retrieve the corresponding model characterized given the gate type and input vector state at a gate.

[3] To consider the effect of varying $N_{\mathrm{d}}$ on $I_{\mathrm{sub}}$, equation (4.21) can be adapted by adding an additional term for $\Delta N_{\mathrm{d}}$ in the exponent. As in the case of $T_{\mathrm{ox}}$, the variation of $N_{\mathrm{d}}$ does not show spatial correlation, and thus $N_{\mathrm{d}}$ can be handled using a similar method as used for $T_{\mathrm{ox}}$ in the framework.

The gate oxide tunneling current density, $J_{\text{tunnel}}$, can be represented by the following analytical model [39]:

$$J_{\text{tunnel}} = \frac{4\pi m^* q}{h^3}(kT)^2 \left(1 + \frac{\gamma kT}{2\sqrt{E_{\text{B}}}}\right) \mathrm{e}^{\frac{E_{F0,\text{Si/SiO}_2}}{kT}} \mathrm{e}^{-\gamma\sqrt{E_{\text{B}}}} \tag{4.22}$$

Here $m^*$ is the transverse mass that equals $0.19m_0$ for electron tunneling and $0.55m_0$ for hole tunneling, where $m_0$ is the free electron rest mass; $h$ is Planck's constant; $\gamma$ is defined as $4\pi T_{\text{ox}}\sqrt{2m_{\text{ox}}}/h$, where $m_{\text{ox}}$ is the effective electron (hole) mass in the oxide; $E_{\text{B}}$ is the barrier height; $E_{F0,Si/SiO_2} = q\phi_S - q\phi_F - E_G/2$ is the Fermi level at the $\text{Si}/\text{SiO}_2$ interface, where $\phi_{\text{S}}$ is surface potential, $\phi_{\text{F}}$ is the Fermi energy level potential, either in the Si substrate for the gate tunneling current through the channel, or in the source/drain region for the gate tunneling current through the source/drain overlap; and $E_{\text{G}}$ is the Si band gap energy.

However, this formulation (Equation 4.22) does not lend itself easily to the analysis of the effects of parameter variations. Therefore, we again use an empirically characterized model to estimate $I_{\text{gate}}$ per transistor width through curve-fitting:

$$I_{\text{gate}} = c' \times \mathrm{e}^{b_1 + b_2 L_{\text{eff}} + b_3 L_{\text{eff}}^2 + b_4 T_{\text{ox}} + b_5 T_{\text{ox}}^2} \tag{4.23}$$

where $c'$ and the $b_i$ terms are the fitting coefficients.

As before, under the variations of $L_{\text{eff}}$ and $T_{\text{ox}}$, $I_{\text{gat}e}$ can be approximated by applying first-order Taylor expansion to the exponent $U'$ of Equation (4.23):

$$I_{\text{gate}} = c' \times \mathrm{e}^{U'{}_0 + \lambda_1 \cdot \Delta L_{\text{eff}} + \lambda_2 \cdot \Delta T_{\text{ox}}} \tag{4.24}$$

where $U'{}_0$ is the nominal value of the exponent $U'$, and $\lambda_0$ and $\lambda_1$ are the derivatives of $U'$ to $L_{\text{eff}}$ and $T_{\text{ox}}$ evaluated at their nominal values, respectively.

Under this approximation, $I_{\text{gate}}$ is also a lognormally distributed random variable, and its PDF can be characterized through the values of $c'$, $U_0'$, and $\lambda_i{}'$. Since the gate tunneling leakage current is input state dependent, the PDFs of the $I_{\text{gate}}$ variables are characterized for all possible input states for each type of gate in the library, and a simple look-up table (LUT) is used for model retrieval while evaluating a specific circuit.

#### 4.4.2.1 Distribution of the Full-Chip Leakage Current

We now present an approach for finding the distribution of $I_{\text{tot}}^{\text{avg}}$ as formulated in Equation (4.18), which is a weighted sum of the subthreshold and gate leakage values for each gate, over all input patterns to the gate. Since the probability of each $\text{vec}_{i,k}$ can be computed by specifying signal probabilities at the circuit primary inputs and propagating the probabilities into all gate pins in the circuit using routine techniques, in this section, we focus on the computation of the PDF of the weighted sum.

As each of $I_{\text{sub},k}$ $(\text{vec}_{i,k})$ or $I_{\text{gate},k}$ $(\text{vec}_{i,k})$ has a lognormal distribution, it can easily be seen that any multiplication by a constant maintains this property. Therefore, the problem of calculating the distribution of $I_{\text{tot}}^{\text{avg}}$ becomes that of computing the PDF of the sum of a set of lognormal random variables. Furthermore, the set of lognormal random variables in the summation could be correlated since:

- the leakage current random variables for any two gates may be correlated due to spatial correlations of intra-die variations of process parameters.
- within the same gate, the subthreshold and gate tunneling leakage currents are correlated, and the leakage currents under different input vectors are correlated, because they are sensitive to the same process parameters of the same gate, regardless of whether these are spatially correlated or not.

Theoretically, the sum of several lognormal distributed random variables is not known to have a closed form. However, it may be well approximated as a lognormal, as is done in Wilkinson's method [40].[4] That is, the sum of $m$ lognormals, $S = \sum_{i=1}^{m} \mathrm{e}^{Y_i}$, where each $Y_i$ is a normal random variable with mean $m_{y_i}$ and standard deviation $\sigma_{y_i}$, and the $Y_i$ variables can be correlated or uncorrelated, can be approximated as a lognormal $\mathrm{e}^Z$, where $Z$ is normally distributed, with mean $m_z$ and standard deviation $\sigma_z$. In Wilkinson's approach, the values of $m_z$ and $\sigma_z$ are obtained by matching the first two moments, $u_1$ and $u_2$, of $\mathrm{e}^Z$ and $S$ as follows:

$$u_1 = E(\mathrm{e}^Z) = E(S) = \sum_{i=1}^{m} E(\mathrm{e}^{Y_i}) \tag{4.25}$$

$$u_2 = E(\mathrm{e}^{2Z}) = E(S^2) = \mathrm{Var}(S) + E^2(S) \tag{4.26}$$

$$= \sum_{i=1}^{m} \mathrm{Var}(\mathrm{e}^{Y_i}) + 2\sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \mathrm{cov}(\mathrm{e}^{Y_i}, \mathrm{e}^{Y_j}) + E^2(S)$$

$$= \sum_{i=1}^{m} \mathrm{Var}(\mathrm{e}^{Y_i}) + 2\sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \left(E(\mathrm{e}^{Y_i}\mathrm{e}^{Y_j}) - E(\mathrm{e}^{Y_i})E(\mathrm{e}^{Y_j})\right) + E^2(S)$$

where $E(.)$ and Var(.) are the symbols for the mean and variance values of a random variable, and cov(.,.) represents the covariance between two random variables.

In general, the mean and variance of a lognormal random variable $\mathrm{e}^{X_i}$, where $X_i$ is normal distributed with mean $m_{x_i}$ and standard deviation $\sigma_{x_i}$, is computed by:

[4] An approximation of the sum of correlated lognormal random variables by Monte Carlo simulations is computationally difficult for large-sized problems. As an alternative, three analytical approaches have been overviewed and compared in [40]: Wilkinson's approach, Schwartz and Yeh's approach, and the cumulant-matching approach. Through numerical comparisons, [40] concluded that Wilkinson's method is the best in terms of computational simplicity and accuracy.

$$E(\mathrm{e}^{X_i}) = \mathrm{e}^{m_{x_i}+\sigma_{x_i}^2/2} \tag{4.27}$$

$$\mathrm{Var}(\mathrm{e}^{X_i}) = \mathrm{e}^{2m_{x_i}+2\sigma_{x_i}^2} - \mathrm{e}^{2m_{x_i}+\sigma_{x_i}^2} \tag{4.28}$$

The covariance between two lognormal random variables $\mathrm{e}^{X_i}$ and $\mathrm{e}^{X_j}$ can be computed by:

$$\mathrm{cov}(\mathrm{e}^{X_i}, \mathrm{e}^{X_j}) = E(\mathrm{e}^{X_i} \cdot \mathrm{e}^{X_j}) - E(\mathrm{e}^{X_i})E(\mathrm{e}^{X_j}) \tag{4.29}$$

Superposing Equations (4.27), (4.28), and (4.29) into Equations (4.25) and (4.26) results in:

$$u_1 = E(\mathrm{e}^Z) = \mathrm{e}^{m_z+\sigma_z^2/2} = E(S) = \sum_{i=1}^{m} (\mathrm{e}^{m_{y_i}+\sigma_{y_i}^2/2}) \tag{4.30}$$

$$u_2 = E(\mathrm{e}^{2Z}) = \mathrm{e}^{2m_z+2\sigma_z^2} = E(S^2) \tag{4.31}$$

$$= \sum_{i=1}^{m} (\mathrm{e}^{2m_{y_i}+2\sigma_{y_i}^2} - \mathrm{e}^{2m_{y_i}+\sigma_{y_i}^2}) + 2\sum_{i=1}^{m-1}\sum_{j=i+1}^{m}\Big(\mathrm{e}^{m_{y_i}+m_{y_j}+(\sigma_{y_i}^2+\sigma_{y_j}^2+2r_{ij}\sigma_{y_i}\sigma_{y_j})/2}$$
$$-\mathrm{e}^{m_{y_i}+\sigma_{y_i}^2/2}\mathrm{e}^{m_{y_j}+\sigma_{y_j}^2/2}\Big) + u_1^2$$

Where $r_{ij}$ is the correlation coefficient between $Y_i$ and $Y_j$.

Solving (Equation 4.30) and (Equation 4.31) for $m_z$ and $\sigma_z$ yields:

$$m_z = 2\ln u_1 - \frac{1}{2}\ln u_2 \tag{4.32}$$

$$\sigma_z^2 = \ln u_2 - 2\ln u_1 \tag{4.33}$$

The computational complexity of Wilkinson's approximation can be analyzed through the cost of computing $m_z$ and $\sigma_z$. The computational complexities of $m_z$ and $\sigma_z$ are determined by those of $u_1$ and $u_2$, whose values can be obtained using the formulas in (Equation 4.30) and (Equation 4.31). It is clear that the computational complexity of $u_1$ is dominated by that of $u_2$, since the complexity of calculating $u_1$ is $O(m)$, while that of $u_2$ is $O(m \cdot N_{\mathrm{corr}})$, where $N_{\mathrm{corr}}$ is the number of correlated pairs among all pairs of $Y_i$ variables. The cost of computing $u_2$ can also be verified by examining the earlier expression of $u_2$ in (Equation 4.26), in which the second term in the summation, in fact, corresponds to the covariance of $Y_i$ and $Y_j$, which becomes zero when $Y_i$ and $Y_j$ are uncorrelated. Therefore, if $r_{ij} \neq 0$ for all pairs of $Y_i$ and $Y_j$, the complexity of calculating $u_2$ is $O(m^2)$; if $r_{ij} = 0$ for all pairs of $i$ and $j$, the complexity is $O(m)$.

As explained earlier, for full-chip leakage analysis, the number of correlated lognormal distributed leakage components in the summation could be extremely large, which could lead to a prohibitive amount of computation. If Wilkinson's method is applied directly, when the total number of gates in the circuit is $N_{\mathrm{g}}$, the complexity for computing the sum will be $O(N_{\mathrm{g}}^2)$, which is impractical for large circuits. In the remainder of this section, we will propose to compute the summation in a more efficient way.

#### 4.4.2.2 Reducing the Cost of Wilkinson's Method

Since Wilkinson's method has a quadratic complexity with respect to the number of correlated lognormals to be summed, we now introduce mechanisms to reduce the number of correlated lognormals in the summation to improve the computational efficiency.

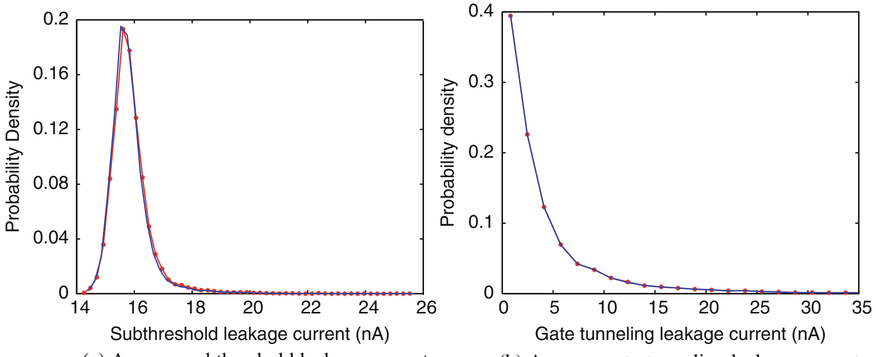
The work of [41] proposes a PCA-based method to compute the full-chip leakage considering the effect of spatial correlations of $L_{\mathrm{eff}}$. The leakage current of each gate is rewritten in terms of its principal components by expanding the variable $\Delta L_{\mathrm{eff}}$ as a linear function of principal components, i.e.,

$$I_{\mathrm{sub}}^{i} = \mathrm{e}^{U_{0,i}+\sum_{t=1}^{N_{\mathrm{p}}} \beta_{1,i}k_t^i \cdot p_t+\beta_{2,i}\cdot \Delta T_{\mathrm{ox},i}} \tag{4.34}$$

where $N_{\mathrm{p}}$ is the number of principal components. The sum of such lognormal terms can be approximated as a lognormal using Wilkinson's formula. The benefit of using a PCA form is that the mean and variance of a lognormal random variable can be computed in $O(N_{\mathrm{p}})$, as can the covariance of two lognormal random variables in PCA form. Therefore, the computation of all values and coefficients in $I_{\mathrm{sub}}^{h}$, and thus the sum of two lognormals in PCA form, can be computed in $O(N_{\mathrm{p}})$. As mentioned in the description of Wilkinson's method, the computation of full-chip leakage current distribution requires a summation of $N_{\mathrm{g}}$ correlated lognormals. Thus, the PCA-based method has an overall computational complexity of $O(N_{\mathrm{p}} \cdot N_{\mathrm{g}})$.

A second approach, presented in [42], which we refer to as the "grouping method," uses two strategies for reducing the computations in applying Wilkinson's formula. First, the number of terms to be summed is reduced by identifying dominant states [38, 43] for the subthreshold and gate tunneling leakage currents for each type of gate in the circuit. As shown in Fig. 4.7a, the leakage PDF curves for simulations using dominant states only, and using the full set of states, for the average subthreshold leakage current of a three-input NAND gate are virtually identical. Similar results are seen for other gate types.

Second, instead of directly computing the sum of random variables of all leakage current terms, by grouping leakage current terms by model and grid location, and calculating the sum in each group separately first, the computational complexity in the computation of full-chip leakage reduces to quadratic in the number of groups. The key idea here is to characterize the leakage current per unit width for each stack type (called a model – these are $N_{\mathrm{models}}$ in number). The summation can be grouped by combining similar models in the same grid. Each group summation can be computed in linear time with respect to the number of leakage terms in the group. The results of the sums in all groups are then approximated as correlated lognormal



(a) Average subthreshold leakage current (b) Average gate-tunneling leakage current

**Fig. 4.7** Comparison of PDFs of average leakage currents using dominant states with that of full input vector states for a 3-input NAND gate, by Monte Carlo simulation with $3\sigma$ variations of $L_{\mathrm{eff}}$ and $T_{\mathrm{ox}}$ 20%. The solid curve shows the result when only dominant states are used, and the starred curve corresponds to simulation with all input vector states

random variables that can then be computed directly using Wilkinson's method, so that we must perform the summation over $N_{\mathrm{groups}} = N_{\mathrm{models}} N_{\mathrm{g}}$ terms. Since the number of groups is relatively small, a calculation that is quadratic in the number of groups is practically very economical.

Specifically, the computational complexity for estimating the distribution of full-chip leakage current is reduced from $O(N_{\mathrm{g}}^2)$ for a naïve application of Wilkinson's formula to a substantially smaller number $O(N_{\mathrm{models}}^2 \cdot n^2)$, where $n$ is the number of correlation grid squares.

A third approach [44], called the "hybrid method," combines the PCA-based and grouping methods, which attack the problem in orthogonal ways. As in the second approach, the leakage of each group is computed in terms of the original random variables. During the summation over all groups, the PCA approach is used to reduce the overall cost. The results in this paper show that the second approach outperforms the first, and that the third (hybrid) method outperforms the second as the number of grid squares, $n$, becomes larger.

The results of full-chip leakage estimation are presented in Fig. 4.8, which show the distribution of total circuit leakage current achieved using a statistical approach (the accuracy of the three methods is essentially indistinguishable) and using Monte Carlo simulation for circuit c7552: it is easy to see that the curve achieved by the basic method matches well with the Monte Carlo simulation result. For all test cases, the run-time of these methods is in seconds or less, while the Monte Carlo simulation takes considerably longer: for the largest test case, c7552, this simulation takes 3 h.

In terms of accuracy, the three methods are essentially very similar. However, they differ in terms of run-time efficiencies. In Tables 4.1 and 4.2, we show the run-times for different methods for ISCAS85 and ISCAS89 benchmark sets,

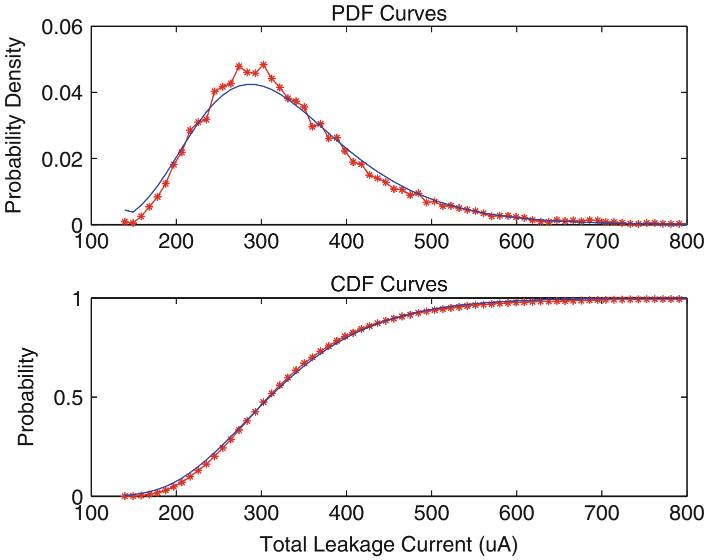


**Fig. 4.8** Distributions of the total leakage against Monte Carlo simulation method for circuit c7552. The solid line illustrates the result of the proposed grouping method, while the starred line shows the Monte Carlo simulation results

**Table 4.1** Run-time comparison of the PCA-based, grouping and hybrid methods for the ISCAS85 benchmarks

| Benchmark | c432 | c880 | c1908 | c2670 | c3540 | c6288 | c5315 | c7552 |
|---|---|---|---|---|---|---|---|---|
| Number of grids | 4 | 4 | 16 | 16 | 16 | 16 | 64 | 64 |
| PCA-based method (s) | 0.03 | 0.06 | 0.18 | 0.27 | 0.40 | 0.57 | 1.43 | 1.82 |
| Grouping method (s) | 0.01 | 0.02 | 0.04 | 0.06 | 0.09 | 0.10 | 0.24 | 0.29 |
| Hybrid method (s) | 0.01 | 0.03 | 0.06 | 0.09 | 0.12 | 0.14 | 0.19 | 0.25 |

**Table 4.2** Run-time comparison of the proposed PCA-based, grouping, and hybrid methods for the ISCAS89 benchmarks

| Benchmark | s5378 | s9234 | s13207 | s15850 | s35932 | s38584 |
|---|---|---|---|---|---|---|
| Number of grids | 64 | 64 | 256 | 256 | 256 | 256 |
| PCA-based method (s) | 0.93 | 1.62 | 7.58 | 8.97 | 17.38 | 24.28 |
| Grouping method (s) | 0.22 | 0.32 | 5.89 | 5.91 | 4.97 | 10.04 |
| Hybrid method (s) | 0.16 | 0.30 | 0.47 | 0.56 | 1.03 | 1.34 |

respectively. In general, the grouping method is about 3–4 times faster than the PCA-based method. As expected, the hybrid approach does not show any run-time advantage over the grouping method for smaller grid sizes. However, run-time of both the grouping and the PCA-based methods grows much faster with the grid size than the hybrid method. In Tables 4.1 and 4.2, when the number of grids grows to greater than 64, the hybird approach is about 100 times faster than the other approaches. Therefore, the run-time can be significantly improved by hybridizing the PCA-based with the grouping approach.

Follow-up work in [45] presents alternative ideas for speeding up the summation of these lognormals, introducing the idea of a virtual-cell approximation, which sums the leakage currents by approximating them as the leakage of a single virtual cell.

## 4.5 Statistical Optimization

Process variations can significantly degrade the yield of a circuit, and optimization techniques can be used to improve the timing yield. An obvious way to increase the timing yield of the circuit is to pad the specifications to make the circuit robust to variations, i.e., to choose a delay specification of the circuit that is tighter than the required delay. This new specification must be appropriately selected to avoid large area or power overheads due to excessively conservative padding.

The idea of statistical optimization is presented in Fig. 4.9, in a space where two design parameters, $p_1$ and $p_2$, may be varied. The upper picture shows the constant value contours of the objective function, and the feasible region where all constraints are met. The optimal value for the deterministic optimization problem is the point at which the lowest value contour intersects the feasible set, as shown. However, if there is a variation about this point that affects the objective function, then after

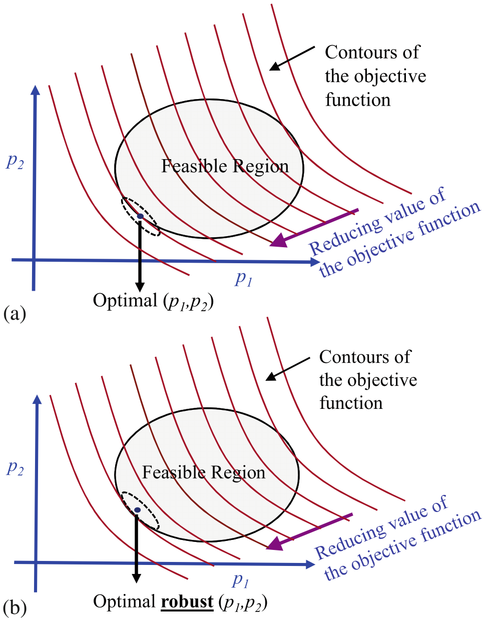

Contours of the objective function
$p_2$
Feasible Region
Reducing value of the objective function
$p_1$
Optimal ($p_1$,$p_2$)
(a)
Contours of the objective function
$p_2$
Feasible Region
Reducing value of the objective function
$p_1$
Optimal **robust** ($p_1$,$p_2$)
(b)

**Fig. 4.9** A conceptual picture of robust optimization

manufacturing, the parameters may shift from the optimal design point. The figure shows an ellipsoidal variational region (corresponding to, say, the 99% probability contours of a Gaussian distribution) around an optimal design point: the manufactured solution may lie within this with a very high probability. It can be seen that a majority of points in this elliptical variational region lie outside the feasible set, implying a high likelihood that the manufactured circuit will fail the specifications. On the other hand, the robust optimum, shown in the lower picture, will ensure that the entire variational region will lie within the feasible set.

Therefore, statistical optimization is essentially the problem of determining the right amount by which the specifications should be "padded" in order to guarantee a certain yield, within the limitations of the process models. Too little padding can result in low yield, while too much padding can result in high resource overheads. More precisely, real designs are bounded from both ends. If the delays are too large, then the timing yield goes down, and if the delays are too small, this may be because of factors such as low threshold voltages in the manufactured part: in such a case, the leakage power becomes high enough that the part will fail its power specifications.

In the remainder of this section, we will first introduce techniques for finding statistical sensitivities – a key ingredient of any optimization method – and then overview some techniques for statistical optimization.

### *4.5.1 Statistical Sensitivity Calculation*

A key problem in circuit optimization is the determination of statistical timing sensitivities and path criticality. Efficient computational engines for sensitivity analysis play an important role in guiding a range of statistical optimizations.

A straightforward approach in [46] involves perturbing gate delays to compute their effect on the circuit output delay. The complexity of the computation is reduced using the notion of a cutset belonging to a node in the timing graph: it is shown that the statistical maximum of the sum of arrival and required times across all the edges of a cutset gives the circuit delay distribution. If all sensitivities are to be computed, the complexity of this approach is potentially quadratic in the size of the timing graph.

For comprehensive sensitivity computation, one of the earliest attempts to compute edge criticalities was proposed in [30], which performs a reverse traversal of the timing graph, multiplying criticality probabilities of nodes with local criticalities of edges. However, this assumes that edge criticalities are independent, which is not a valid in practice. Follow-up work by the same group in [47] extends the cutset-based idea in [46] to compute the criticality of edges by linearly traversing the timing graph. The criticality of an edge in a cutset is computed using a balanced binary partition tree. Edges recurring in multiple cutsets are recorded in an array-based structure while traversing the timing graph.

Another effort in [48] approaches the problem by defining the statistical sensitivity matrix of edges in the timing graph with respect to the circuit output delay, and uses the chain rule to compute these values through a reverse traversal of the timing graph. Due to the matrix multiplications involved, albeit typically on sparse matrices, the complexity of the approach could be large, especially if the principal components are not sparse.

Like [46, 47], the work in [49] proposes an algorithm to compute the criticality probability of edges (nodes) in a timing graph using the notion of cutsets. Edges crossing multiple cutsets are dealt with using a zone-based approach, similar to [50], in which old computations are reused to the greatest possible extent. This work shows that without appropriate reordering, the errors propagated during criticality computations that use to Clark's *MAX* operation can be large; this is an effect that was ignored by previous approaches. Further, the work proposes a clustering-based pruning algorithm to control this error, eliminating a large number of non-competing edges in cutsets with several thousand edges. An extension in [51] investigates the effect of independent random variations on criticality computation and devises a simple scheme to keep track of structural correlations due to such variations.

### *4.5.2 Performance Optimization*

Gate sizing is a valuable tool for improving the timing behavior of a circuit. In its most common form, it attempts to minimize an objective function, such as the area or the power dissipation, subject to timing constraints. In the literature, it is perhaps the most widely used target for statistical approaches, primarily because it is a transform that is applied at the right level, where design uncertainty does not overwhelm process uncertainty.

Early approaches to variation-tolerant gate sizing, which incorporate statistical timing models, include early work in [26], which formulates a statistical objective and timing constraints and solves the resulting nonlinear optimization formulation. However, this is computationally difficult and does not scale to large circuits. Other approaches for robust gate sizing that lie in the same family include [46, 52–54]: in these, the central idea is to capture the delay distributions by performing a statistical static timing analysis (SSTA), as opposed to the traditional STA, and then use either a general nonlinear programming technique or statistical sensitivity-based heuristic procedures to size the gates. In [55], the mean and variances of the node delays in the circuit graph are minimized in the selected paths, subject to constraints on delay and area penalty.

More formal optimization approaches have also been used. Approaches for optimizing the statistical power of the circuit, subject to timing yield constraints, can be presented as a convex formulation, as a second-order conic program [56]. For the binning model, a yield optimization problem is formulated [57], providing a binning yield loss function that has a linear penalty for delay of the circuit exceeding the target delay; the formulation is shown to be convex.

A gate sizing technique based on robust optimization theory has also been proposed [58, 59]: robust constraints are added to the original constraints set by modeling the intra-chip random process parameter variations as Gaussian variables, contained in a constant probability density uncertainty ellipsoid, centered at the nominal values.

Several techniques in the literature go beyond the gate sizing transform. For example, algorithms for statistically aware dual threshold voltage and sizing are presented in [60, 61]. Methods for optimal statistical pipeline design are present in [62], which explores the tradeoff between the logic depth of a pipeline and the yield, as well as gate sizing. The work argues that delay-unbalanced pipelines may provide better yields than delay-balanced pipelines.

## 4.6 Sensors for Post-Silicon Diagnosis

With the aid of SSTA tools, designers can optimize a circuit before it is fabricated, in the expectation that it will meet the delay and power requirements after being manufactured. In other words, SSTA is a presilicon analysis technique used to determine the range of performance (delay or power) variations over a large population of dies. A complementary role, after the chip is manufactured, is played by post-silicon diagnosis, which is typically directed toward determining the performance of an individual fabricated chip based on measurements on that specific chip. This procedure provides particular information that can be used to perform post-silicon optimizations to make a fabricated part meet its specifications. Because presilicon analysis has to be generally applicable to the entire population of manufactured chips, the statistical analysis that it provides shows a relatively large standard deviation for the delay. On the other hand, post-silicon procedures, which are tailored to individual chips, can be expected to provide more specific information. Since tester time is generally prohibitively expensive, it is necessary to derive the maximum possible information through the fewest post-silicon measurements.

In the past, the interaction between presilicon analysis and post-silicon measurements has been addressed in several ways. In [63], post-silicon measurements are used to learn a more accurate spatial correlation model, which is fed back to the analysis stage to refine the statistical timing analysis framework. In [64], a path-based methodology is used for correlating post-silicon test data to presilicon timing analysis. In [57], a statistical gate sizing approach is studied to optimize the binning yield. Post-silicon debug methods and their interaction with circuit design are discussed in [65].

In this section, we will discuss two approaches to diagnosing the impact of process variations on the timing behavior of a manufactured part. In each case, given the original circuit whose delay is to be estimated, the primary idea is to determine information from specific on-chip test structures to narrow the range of the performance distribution substantially. In the first case, we use a set of ring oscillators, and in the second, we synthesize a representative critical path whose behavior tracks the worst-case delay of the circuit. In each case, we show how the results of a limited measurement can be used to diagnose the performance of the manufactured part. The role of this step is seated between presilicon SSTA and post-silicon full chip testing. The approaches used here combine the results of presilicon SSTA for the circuit with the result of a small number of post-silicon measurements on an individual manufactured die to estimate the delay of that particular die.

An example use case scenario for this analysis in the realm of post-silicon tuning. Adaptive Body Bias (ABB) [66–68] is a post-silicon method that determines the appropriate level of body bias to be applied to a die to influence its performance characteristics. ABB is typically a coarse-grained optimization, both in terms of the granularity at which it can be applied (typically on a per-well basis) as well as in terms of the granularity of the voltage levels that may be applied (typically, the separation between ABB levels is 50–100 mV). Current ABB techniques use a replica of a critical path to predict the delay of the fabricated chip, and use this to feed a phase detector and a counter, whose output is then used to generate the requisite body bias value. Such an approach assumes that one critical path on a chip is an adequate reflection of on-chip variations. In general, there will be multiple potential critical paths even within a single combinational block, and there will be a large number of combinational blocks in a within-die region. Choosing a single critical path as representative of all of these variations is impractical and inaccurate. In contrast, an approach based on these test structures implicitly considers the effects of all paths in a circuit (without enumerating them, of course), and provides a PDF that concretely takes spatially correlated and uncorrelated parameters into account to narrow the variance of the sample, and has no preconceived notions, prior to fabrication, as to which path will be critical. The $3\sigma$ or $6\sigma$ point of this PDF may be used to determine the correct body bias value that compensates for process variations.

A notable approach [69, 70] addresses the related problem of critical path identification under multiple supply voltages. Since the critical paths may change as the supply voltage is altered, this method uses a voltage sensitivity-based procedure to identify a set of critical paths that can be tested to characterize the operating frequency of a circuit. An extension allows for sensitive paths to be dynamically configured as ring oscillators. While the method does not explicitly address process variations, the general scheme could be extended for the purpose. Overall, this method falls under the category of more time-intensive test-based approaches, as against the faster sensor-based approach described in the rest of this section, and plays a complementary role to the sensor-based method in post-silicon test.

### *4.6.1 Using Ring Oscillator Test Structures*

In this approach, we gather information from a small set of test structures such as ring oscillators (ROs), distributed over the area of the chip, to capture the variations of spatially correlated parameters over the die. The physical sizes of the test structures are small enough that it is safe to assume that they can be incorporated into the circuit using reserved space that may be left for buffer insertion, decap insertion, etc. without significantly perturbing the layout.

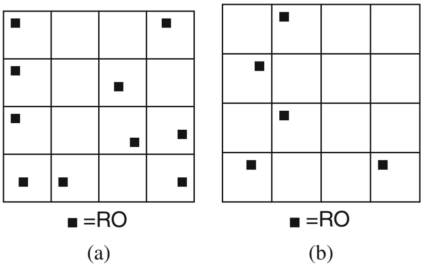


**Fig. 4.10** Two different placements of test structures under the grid spatial correlation model

To illustrate the idea, we show a die in Fig. 4.10, whose area is gridded into spatial correlation regions. For simplicity, we will assume in this example that the spatial correlation regions for all parameters are the same, although the idea is valid, albeit with an uglier picture, if this is not the case. Fig. 4.10a,b show two cases where test structures are inserted on the die: the two differ only in the number and the locations of these test structures. The data gathered from the test structures in Fig. 4.10a,b are used in this paper to determine a new PDF for the delay of the original circuit, conditioned on this data. This PDF has a significantly smaller variance than that obtained from SSTA, as is illustrated in Fig. 4.11.

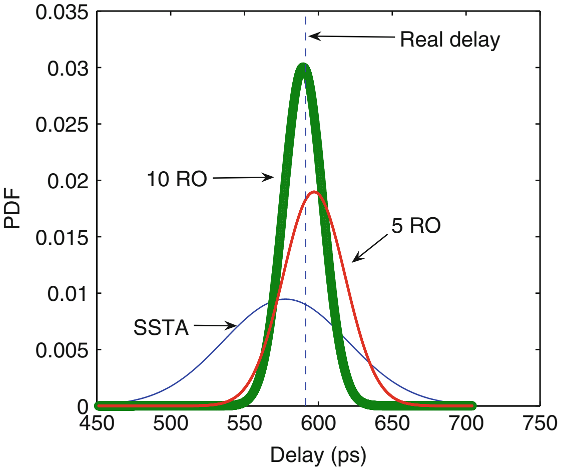


**Fig. 4.11** Reduced-variance PDFs, obtained from statistical delay prediction, using data gathered from the test structures in Fig. 4.10

The plots in Fig. 4.11 may be interpreted as follows. When no test structures are used and no post-silicon measurements are performed, the PDF of the original circuit is the same as that computed by SSTA. When five ROs are used, a tighter spread is seen for the PDF, and the mean shifts toward the actual frequency for the die. This spread becomes tighter still when 10 ROs are used. In other words, as the number of test structures is increased, more information can be derived about variations on the die, and its delay PDF can be predicted with greater confidence: the standard deviation of the PDF from SSTA is always an upper bound on the standard deviation of this new delay PDF. In other words, by using more or fewer test structures, the approach is scalable in terms of statistical confidence.

If we represent the delay of the original circuit as $d$, then the objective is to find the conditional PDF of $d$, given the vector of delay values, $\mathbf{d}_r$, corresponding to the delays of the test structures, measured from the manufactured part. Note the $\mathbf{d}_r$ corresponds to one sample of the probabilistic delay vector, $\mathbf{d}_t$, of test structure delays. The corresponding means and variances of $d$ are unsubscripted, and those of the test structures have the subscript "t."

We appeal to a well-known result to solve this problem: given a vector of jointly Gaussian distributions, we can determine the conditional distribution of one element of the vector, given the others. Specifically, consider a Gaussian-distributed vector $\begin{bmatrix}\mathbf{X}_1\\ \mathbf{X}_2\end{bmatrix}$ with mean $\boldsymbol{\mu}$ and a nonsingular covariance matrix $\boldsymbol{\Sigma}$. Let us define $\mathbf{X}_1 \sim N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_{11}), \mathbf{X}_2 \sim N(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_{22})$. If $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are partitioned as follows,

$$\boldsymbol{\mu} = \begin{bmatrix}\boldsymbol{\mu}_1\\ \boldsymbol{\mu}_2\end{bmatrix} \text{and} \quad \boldsymbol{\Sigma} = \begin{bmatrix}\boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12}\\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22}\end{bmatrix}, \tag{4.35}$$

then the distribution of $\mathbf{X}_1$ conditional on $\mathbf{X}_2 = \mathbf{x}$ is multivariate normal, and its mean and covariance matrix are given by

$$\mathbf{X}_1|(\mathbf{X}_2 = \mathbf{x}) \sim N(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\Sigma}}) \tag{4.36a}$$

$$\bar{\boldsymbol{\mu}} = \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\mathbf{x} - \boldsymbol{\mu}_2) \tag{4.36b}$$

$$\bar{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}. \tag{4.36c}$$

We define $\mathbf{X}_1$ as the *original subspace*, and $\mathbf{X}_2$ as the *test subspace*. By stacking $d$ and $\mathbf{d}_t$ together, a new vector $\mathbf{d}_{\text{all}} = \left[d\ \mathbf{d}_t^{\mathrm{T}}\right]^{\mathrm{T}}$ is formed, with the original subspace containing only one variable $d$ and the test subspace containing the vector $\mathbf{d}_t$. The random vector $\mathbf{d}_{\text{all}}$ is multivariate Gaussian-distributed, with its mean and covariance matrix given by:

$$\boldsymbol{\mu}_{\text{all}} = \begin{bmatrix}\mu\\ \boldsymbol{\mu}_t\end{bmatrix} \text{and} \quad \boldsymbol{\Sigma}_{\text{all}} = \begin{bmatrix}\sigma^2 & \mathbf{a}^{\mathrm{T}}\mathbf{A}_t\\ \mathbf{A}_t^{\mathrm{T}}\mathbf{a} & \boldsymbol{\Sigma}_t\end{bmatrix}. \tag{4.37}$$

We may then apply the above result to obtain the conditional PDF of $d$, given the delay information from the test structures. We know that the conditional distribution of $d$ is Gaussian, and its mean and variance can be obtained as:

$$\text{PDF}(d_{\text{cond}}) = \text{PDF}\left(d|(\mathbf{d}_t = \mathbf{d}_r)\right) \sim N(\bar{\mu}, \bar{\sigma}^2) \tag{4.38a}$$

$$\bar{\mu} = \mu + \mathbf{a}^{\mathrm{T}}\mathbf{A}_t\boldsymbol{\Sigma}_t^{-1}(\mathbf{d}_r - \boldsymbol{\mu}_t) \tag{4.38b}$$

$$\bar{\sigma}^2 = \sigma^2 - \mathbf{a}^{\mathrm{T}}\mathbf{A}_t\boldsymbol{\Sigma}_t^{-1}\mathbf{A}_t^{\mathrm{T}}\mathbf{a}. \tag{4.38c}$$

From Equations Equation (4.38b) and Equation (4.38c), we conclude that while the conditional mean of the original circuit is adjusted making use of the result vector, $\mathbf{d}_r$, the conditional variance is *independent* of the measured delay values, $\mathbf{d}_r$.

Examining Equation (4.38c) more closely, we see that for a given circuit, the variance of its delay before measuring the test structures, $\sigma^2$, and the coefficient vector, **a**, are fixed and can be obtained from SSTA. The only variable that is affected by the test mechanism is the coefficient matrix of the test structures, $\mathbf{A}_t$, which also impacts $\boldsymbol{\Sigma}_t$. Therefore, the value of the conditional variance can be modified by adjusting the matrix $\mathbf{A}_t$. We know that $\mathbf{A}_t$ is the coefficient matrix formed by the sensitivities with respect to the principal components of the test structures. The size of $\mathbf{A}_t$ is determined by the number of test structures on the chip, and the entry values of $\mathbf{A}_t$ is related to the type of the test structures and their locations on the chip. Therefore if we use the same type of test structures on the circuit, then by varying their number and locations, we can modify the matrix $\mathbf{A}_t$, hence adjust the value of the conditional variance. Intuitively, this implies that the value of the conditional variance depends on how many test structures we have, and how well the test structures are distributed, in the sense of capturing spatial correlations between variables.

If the number of test structures equals the number of PCA components, the test structures collectively cover all principal components, and all variations are spatially correlated, then it is easy to show [71] that the test structures can exactly recover the principal components, and the delay of the manufactured part can be exactly predicted (within the limitations of statistical modeling). When we consider uncorrelated variations, by definition, it is impossible to predict these using any test structure that is disjoint from the circuit. However, we can drown these out by increasing the number of stages in the ring oscillator. This is shown in Fig. 4.12, which shows the effects of increasing the number of ring oscillator stages on predicting the delays of circuits s13207 and s5378. It is easily observed that the curves

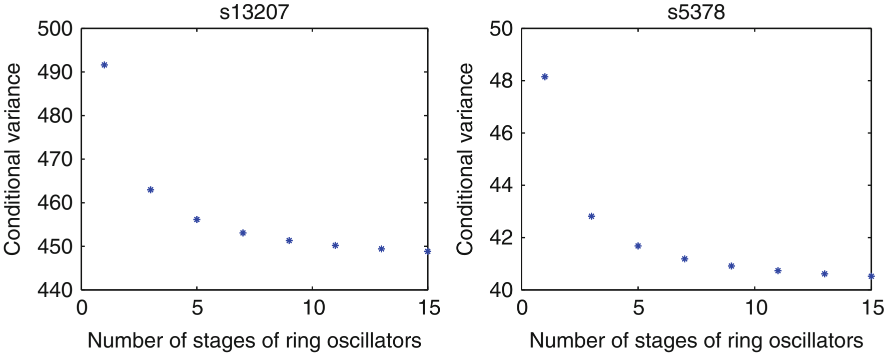


**Fig. 4.12** Conditional variance of the delay of the original circuit with respect to the number of stages of ROs

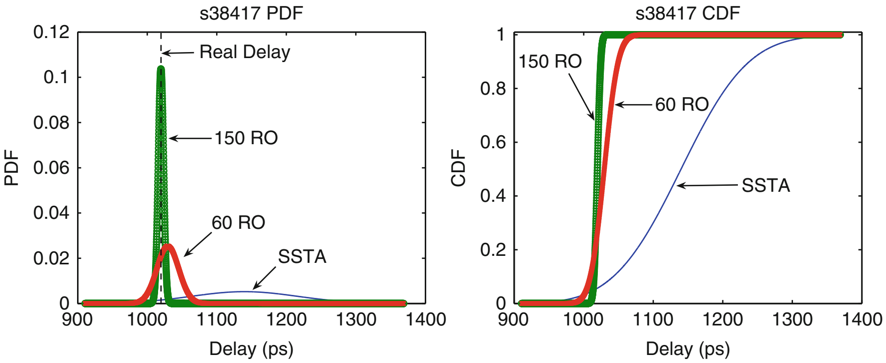


**Fig. 4.13** PDF and CDF with insufficient number of test structures for circuit s38417 (considering *L*)

are monotonically decreasing. The results are similar for all other circuits in the benchmark set.

Finally, as was illustrated in Fig. 4.11, if a smaller number of test structures are used, then the variance of the conditional distribution increases. Figure 4.13 shows the predicted delay distribution for a typical sample of the circuit s38417, the largest circuit in the ISCAS89 benchmark suite. Each curve in the circuit corresponds to a different number of test structures, and it is clearly seen that even when the number of test structures is less than *G*, a sharp PDF of the original circuit delay can still be obtained using our method, with a variance much smaller than provided by SSTA. The tradeoff between the number of test structures and the reduction in the standard deviation can also be observed clearly. For this particular die, while SSTA can only assert that it can meet a 1400 ps delay requirement, using 150 test structures we can say with more than 99.7% confidence that the fabricated chip meets a 1040 ps delay requirement, and using 60 test structures we can say with such confidence that it can meet a 1080 ps delay requirement.

### *4.6.2 Using a Representative Critical Path*

Another approach to post-silicon diagnosis involves the replication of a critical path of a circuit. As mentioned earlier, such techniques have been used in [66–68] in connection with adaptive body bias (ABB) or adaptive supply voltage (ASV) optimizations, where a replica of the critical path at nominal parameter values (we call this the *critical path replica* (CPR)) is used; its delay is measured to determine the optimal adaptation. However, such an approach has obvious problems: first, it is likely that a large circuit will have more than a single critical path, and second, a nominal critical path may have different sensitivities to the parameters than other near-critical paths, and thus may not be representative. An alternative approach in [71] uses a number of on-chip ring oscillators to capture the parameter variations of the original circuit. However, this approach requires measurements for hundreds of ring oscillators for a circuit with reasonable size and does not address issues related to how these should be placed or how the data can be interpreted online.

In this section, we describe how we may build an on-chip test structure that captures the effects of parameter variations on all critical paths, so that a measurement on this test structure provides us a reliable prediction of the actual delay of the circuit, with minimal error, for all manufactured die. The key idea is to synthesize the test structure such that its delay can reliably predict the maximum delay of the circuit, under across-die as well as within-die variations. In doing so, we take advantage of the property of spatial correlation between parameter variations to build this structure and determine the physical locations of its elements.

This structure, which we refer to as the *representative critical path* (RCP), is typically different from the critical path at nominal values of the process parameters. In particular, a measurement on the RCP provides the worst-case delay of the whole circuit, while the nominal critical path is only valid under no parameter variations, or very small variations. Since the RCP is an on-chip test structure, it can easily be used within existing post-silicon tuning schemes, e.g., by replacing the nominal critical path in the schemes in [66–68]. While our method accurately captures any correlated variations, it suffers from one limitation that is common to any on-chip test structure: it cannot capture the effects of spatially uncorrelated variations, because by definition, there is no relationship between those parameter variations of a test structure and those in the rest of the circuit. To the best of our knowledge, this work is the first effort that synthesizes a critical path in the statistical sense. The physical size of the RCP is small enough that it is safe to assume that it can be incorporated into the circuit (using reserved space that may be left for buffer insertion, decap insertion, etc.) without significantly perturbing the layout.

An obvious way to build an RCP is to use the nominal critical path for this prediction: this is essentially the critical path replica method [66–68]. However, the delay sensitivities of this nominal path may not be very representative. For instance, under a specific variation in the value of a process parameter, the nominal critical path delay may not be affected significantly, but the delay of a different path may be affected enough that it becomes critical. Therefore, we introduce the notion of building an RCP, and demonstrate that the use of this structure yields better results than the use of the nominal critical path.

The overall approach is summarized as follows. For the circuit under consideration, let the maximum delay be represented as a random variable, $d_c$. We build an RCP in such a way that its delay is closely related to that of the original circuit, and varies in a similar manner. The delay of this path can be symbolically represented by another random variable, $d_p$. Clearly, the ordered pair $(d_c, d_p)$ takes on a distinct value in each manufactured part, and we refer to this value as $(d_{cr}, d_{pr})$. In other words, $(d_{cr}, d_{pr})$ corresponds to one sample of $(d_c, d_p)$, corresponding to a particular set of parameter values in the manufactured part. Since the RCP is a single path, measuring $d_{\mathrm{pr}}$ involves considerably less overhead than measuring the delay of each potentially critical path. From the measured value of $d_{\mathrm{pr}}$, we will infer the value, $d_{\mathrm{cr}}$, of $d_{\mathrm{c}}$ for this sample, i.e., corresponding to this particular set of parameter values.

It can be shown mathematically [72] that in order to predict the circuit delay well, the correlation coefficient, $\rho$, between the RCP delay and the circuit delay must be high, i.e., close to 1. This is also in line with an intuitive understanding of the correlation coefficient. However, what is not entirely obvious is that this implies that the means of these delays can be very different, as long as $\rho$ is high. In other words, we should try to match $\rho$ rather than the mean delay, as is done when we choose the nominal critical path.

Assume that the circuit delay is listed in the canonical form in (Equation 4.13), and that the RCP delay $d_{\mathrm{c}}$ is also in canonical form as:

$$d_{\mathrm{c}} = \mu_c + \sum_{i=1}^{m} a_i p_i = \mu_c + \mathbf{a}^{\mathrm{T}}\mathbf{p} + R_c \tag{4.39}$$

where all terms inherit their meanings from Equation (4.13).

The correlation coefficient is then given by

$$\rho = \frac{\mathbf{a}^{\mathrm{T}}\mathbf{b}}{\sigma_{\mathrm{c}}\sigma_{\mathrm{p}}} \tag{4.40}$$

where $\sigma_{\mathrm{c}} = \sqrt{\mathbf{a}^{\mathrm{T}}\mathbf{a} + \sigma_{R_c}^2}$ and $\sigma_{\mathrm{p}} = \sqrt{\mathbf{b}^{\mathrm{T}}\mathbf{b} + \sigma_{R_p}^2}$. An important point to note is that $\rho$ depends only on the coefficients of the PCs for both the circuit and the critical path and their independent terms, and not on their means.

Although the problem of maximizing $\rho$ can be formulated as a nonlinear programming problem, it admits no obvious easy solutions. Therefore, the work in [72] presents three heuristic approaches for finding the RCP. The first begins with the nominal critical path with all gates at minimum size, and then uses a greedy TILOS-like [73] heuristic to size up the transistors with the aim of maximizing $\rho$. The second builds the critical path from scratch, adding one stage at a time, starting from the output stage, each time greedily maximizing $\rho$ as the new stage is added. The third combines these methods: it first builds the RCP using the second method, sets all transistors in it to minimum size, and then upsizes the transistors using a TILOS-like heuristic to maximize $\rho$ greedily at each step.

The first method is cognizant of the structure of the circuit, and works well when the circuit is dominated by a single path, or by a few paths of similar sensitivity. When the number of critical paths is very large, choosing a single nominal path as a starting point could be misleading, and the second method may achieve greater benefits.

The results of the three methods are generally within similar ranges of accuracy. As expected, Method I performs better with circuits with a small number of critical paths, and Method II on circuits with more critical paths. Method III performs better

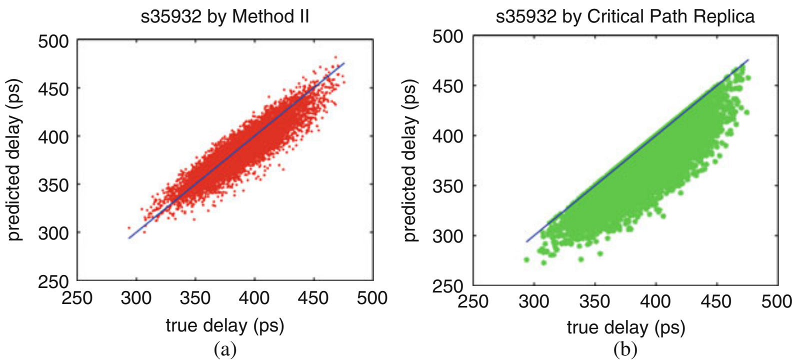


**Fig. 4.14** The scatter plot: (**a**) true circuit delay vs. predicted delay by Method II and (**b**) true circuit delay vs. predicted delay using the CPR method

than Method II. With its more limited search space, Method II is the fastest of the three.

As an example result, we show scatter plots for both Method II and CPR for the circuit s35932 in Fig. 4.14a, b, respectively. The horizontal axis of both figures is the delay of the original circuit for a sample of the Monte Carlo simulation. The vertical axis of Fig. 4.14a is the delay predicted by our method, while the vertical axis of Fig. 4.14b is the delay of the nominal critical path, used by the CPR method. The ideal result is represented by the $(x = y)$ axis, shown using a solid line. It is easily seen that for the CPR method, the delay of the CPR is either equal to the true delay (when it is indeed the critical path of the manufactured circuit) or smaller (when another path becomes more critical, under manufacturing variations). On the other hand, for Method II, all points cluster closer to the $(x = y)$ line, an indicator that the method produces accurate results. The delay predicted by our approach can be larger or smaller than the circuit delay, but the errors are small. Note that neither Method II nor the CPR Method is guaranteed to be pessimistic, but such a consideration can be enforced by the addition of a guard band that corresponds to the largest error. The RCP approach has a clear advantage of a significantly smaller guard band in these experiments.

## 4.7 Conclusion

This chapter has presented an overview of issues related to the statistical analysis of digital circuits. Our focus has been on modeling statistical variations and carrying these into statistical timing and power analyses, which in turn are used to drive statistical optimization at the presilicon stage. Finally, we overview initial forays into the realm of using fast post-silicon measurements from special sensors to determine circuit delay characteristics.

**Acknowledgments** The author would like to thank Hongliang Chang, Qunzeng Liu, and Jaskirat Singh, whose work has contributed significantly to the content of this chapter.

## References

1. Liu Y, Nassif SR, Pileggi LT, Strojwas AJ (Jun 2000) Impact of interconnect variations on the clock skew of a gigahertz microprocessor. In: Proceedings of the ACM/IEEE design automation conference, Los Angeles, CA, pp 168–171
2. Chang H, Sapatnekar SS (Nov 2003) Statistical timing analysis considering spatial correlations using a single PERT-like traversal. In: Proceedings of the IEEE/ACM international conference on computer-aided design, San Jose, CA, pp 621–625
3. Agarwal A, Blaauw D, Zolotov V, Sundareswaran S, Zhao M, Gala K, Panda R (Jan 2003) Statistical timing analysis considering spatial correlations. In: Proceedings of the Asia/South Pacific Design Automation Conference, Kitakyushu, pp 271–276
4. Agarwal A, Blaauw D, Zolotov V, Sundareswaran S, Zhao M, Gala K, Panda R (Dec 2002) Path-based statistical timing analysis considering inter- and intra-die correlations. In: Workshop notes, ACM/IEEE international workshop on timing issues in the specification and synthesis of digital systems (TAU), Monterey, CA, pp 16–21
5. Pelgrom MJM, Duinmaijer ACJ, Welbers APG (Oct 1989) Matching properties of MOS transistors. IEEE J Solid-State Circuits 24(5):1433–1440
6. Xiong J, Zolotov V, He L (Apr 2006) Robust extraction of spatial correlation. In: Proceedings of the ACM international symposium on physical design, San Jose, CA, pp 2–9
7. Liu F (Jun 2007) A general framework for spatial correlation modeling in VLSI design. In: Proceedings of the ACM/IEEE design automation conference, San Diego, CA, pp 817–822
8. Chen R, Zhang L, Zolotov V, Visweswariah C, Xiong J (Jan 2008) Static timing: back to our roots. In: Proceedings of the Asia/South Pacific design automation conference, Seoul, pp 310–315
9. Morrison DF (1976) Multivariate statistical methods. McGraw-Hill, New York, NY
10. Bhardwaj S, Vrudhula S, Ghanta P, Cao Y (Jul 2006) Modeling of intra-die process variations for accurate analysis and optimization of nano-scale circuits. In: Proceedings of the ACM/IEEE design automation conference, San Francisco, CA, pp 791–796
11. Bell T. An ICA page – papers, code, demos, links. Available at http://www.cnl.salk.edu/tony/ica.html (Accessed Sep 22, 2010)
12. Hyvärinen A, Oja E (2000) Independent component analysis: algorithms and applications. Neural Networks, 13(4–5):411–430
13. Singh J, Sapatnekar SS (2006) Statistical timing analysis with correlated non-gaussian parameters using independent component analysis. In: Proceedings of the ACM/IEEE design automation conference, Anaheim, CA, pp 155–160
14. Orshansky M, Keutzer K (Jun 2002) A general probabilistic framework for worst case timing analysis. In: Proceedings of the ACM/IEEE design automation conference, New Orleans, LA, pp 556–561
15. Deguchi Y, Ishiura N, Yajima S (Jun 1991) Probabilistic CTSS: analysis of timing error probability in asynchronous logic circuits. In: Proceedings of the ACM/IEEE design automation conference, San Francisco, CA, pp 650–655
16. Liou JJ, Cheng KT, Kundu S, Krstic A (Jun 2001) Fast statistical timing analysis by probabilistic event propagation. In: Proceedings of the ACM/IEEE design automation conference, Las Vegas, NV, pp 661–666
17. Devgan A, Kashyap CV (Nov 2003) Block-based statistical timing analysis with uncertainty. In: Proceedings of the IEEE/ACM international conference on computer-aided design, San Jose, CA, pp 607–614
18. Ramalingam A, Nam G-J, Singh AK, Orshansky M, Nassif SR, Pan DZ (Nov 2006) An accurate sparse matrix based framework for statistical static timing analysis. In: Proceedings of the IEEE/ACM international conference on computer-aided design, San Jose, CA, pp 231–236

19. Singhee A, Singhal S, Rutenbar RA (Nov 2008) Practical, fast Monte Carlo statistical static timing analysis: why and how. In: Proceedings of the IEEE/ACM international conference on computer-aided design, San Jose, CA, pp 190–195
20. Kanoria Y, Mitra S, Montanari A (Apr 2010) Statistical static timing analysis using Markov chain Monte Carlo. In: Proceedings of design, automation, and test in Europe, Dresden, pp 813–818
21. Chang H, Sapatnekar SS (Sep 2005) Statistical timing analysis under spatial correlations. IEEE Trans Comput Aided Des Integr Circ Syst 24(9):1467–1482
22. Chang H, Zolotov V, Narayan S, Visweswariah C (Jun 2005) Parameterized block-based statistical timing analysis with non-Gaussian parameters, nonlinear delay functions. In: Proceedings of the ACM/IEEE design automation conference, Anaheim, CA, pp 71–76
23. Zhan Y, Strojwas A, Li X, Pileggi L (Jun 2005) Correlation-aware statistical timing analysis with non-Gaussian delay distributions. In: Proceedings of the ACM/IEEE design automation conference, Anaheim, CA, pp 77–82
24. Feng Z, Li P, Zhan Y (Jun 2007) Fast second-order statistical static timing analysis using parameter dimension reduction. In: Proceedings of the ACM/IEEE design automation conference, San Diego, CA, pp 244–249
25. Berkelaar M (Dec 1997) Statistical delay calculation, a linear time method. In: Workshop notes, ACM/IEEE international workshop on timing issues in the specification and synthesis of digital systems, Austin, TX, pp 15–24
26. Jacobs E, Berkelaar MRCM (Mar 2000) Gate sizing using a statistical delay model. In: Proceedings of design, automation, and test in Europe, Paris, pp 283–290
27. Papoulis A (1991) Probability, random variables, and stochastic processes, 3rd edn. McGraw-Hill, New York, NY
28. Tsukiyama S, Tanaka M, Fukui M (Jan 2001) A statistical static timing analysis considering correlations between delays. In: Proceedings of the Asia/South Pacific design automation conference, Yokohama, pp 353–358
29. Clark CE (1961) The greatest of a finite set of random variables. Oper Res 9:85–91
30. Visweswariah C, Ravindran K, Kalafala K, Walker SG, Narayan S (Jun 2004) First-order incremental block-based statistical timing analysis. In: Proceedings of the ACM/IEEE design automation conference, San Diego, CA, pp 331–336
31. Jyu HF, Malik S, Devadas S, Keutzer KW (Jun 1993) Statistical timing analysis of combinational logic circuits. IEEE Trans VLSI Syst 1(2):126–137
32. Li X, Le J, Gopalakrishnan P, Pileggi LT (Nov 2004) Asymptotic probability extraction for non- normal distributions of circuit performance. In: Proceedings of the IEEE/ACM international conference on computer-aided design, San Jose, CA, pp 2–9
33. Acar E, Devgan A, Rao R, Liu Y, Su H, Nassif S, Burns J (Aug 2003) Leakage and leakage sensitivity computation for combinational circuits. In: Proceedings of the international symposium of low power electronic devices, Seoul, pp 96–99
34. Najm FN (Dec 1994) A survey of power estimation techniques in VLSI circuits. IEEE Trans VLSI Syst 2(4):446–455
35. Taur Y, Ning TH (1998) Fundamentals of modern VLSI devices. Cambridge University Press, New York, NY
36. Mukhopadhyay S, Roy K (Aug 2003) Modeling and estimation of total leakage current in nanoscaled CMOS devices considering the effect of parameter variation. In: Proceedings of the international symposium of low power electronic devices, Seoul, pp 172–175
37. Rao R, Srivastava A, Blaauw D, Sylvester D (Aug 2003) Statistical estimation of leakage current considering inter- and intra-die process variation. In: Proceedings of the international symposium of low power electronic devices, Seoul, pp 84–89
38. Sirichotiyakul S, Edwards T, Oh C, Zuo J, Dharchoudhury A, Panda R, Blaauw D (Jun 1999) Stand-by power minimization through simultaneous threshold voltage selection and circuit sizing. In: Proceedings of the ACM/IEEE design automation conference, New Orleans, LA, pp 436–441

39. Bowman KA, Wang L, Tang X, Meindl JD (Aug 2001) A circuit level perspective of the optimum gate oxide thickness. IEEE Trans Electron Devices 48(8):1800–1810
40. Abu-Dayya AA, Beaulieu NC (Jun 1994) Comparison of methods of computing correlated lognormal sum distributions and outages for digital wireless applications. In: IEEE 44th vehicular technology conference, vol 1, Stockholm, pp 175–179
41. Srivastava A, Shah S, Agarwal K, Sylvester D, Blaauw D, Director SW (Jun 2005) Accurate and efficient gate-level parametric yield estimation considering correlated variations in leakage power and performance. In: Proceedings of the ACM/IEEE design automation conference, Anaheim, CA, pp 535–540
42. Chang H, Sapatnekar SS (Jun 2005) Full-chip analysis of leakage power under process variations, including spatial correlations. In: Proceedings of the ACM/IEEE design automation conference, Anaheim, CA, pp 523–528
43. Lee D, Kwong W, Blaauw D, Sylvester D (Jun 2003) Analysis and minimization techniques for total leakage considering gate oxide leakage. In: Proceedings of the ACM/IEEE design automation conference, Anaheim, CA, pp 175–180
44. Chang H, Sapatnekar SS (Apr 2007) Prediction of leakage power under process uncertainties. ACM Trans Des Autom Electron Syst 12(2); Article 12:27 pp
45. Kim W, Do KT, Kim YH (Apr 2010) Statistical leakage estimation based on sequential addition of cell leakage currents. IEEE Trans VLSI Syst 18(4):602–615
46. Chopra K, Shah S, Srivastava A, Blaauw D, Sylvester D (Nov 2005) Parametric yield maximization using gate sizing based on efficient statistical power and delay gradient computation. In: Proceedings of the IEEE/ACM international conference on computer-aided design, San Jose, CA, pp 1023–1028
47. Xiong J, Zolotov V, Venkateswaran N, Visweswariah C (Jul 2006) Criticality computation in parameterized statistical timing. In: Proceedings of the ACM/IEEE design automation conference, San Francisco, CA, pp 63–68
48. Li X, Le J, Celik M, Pileggi LT (Nov 2005) Defining statistical sensitivity for timing optimization of logic circuits with large-scale process and environmental variations. In: Proceedings of the IEEE/ACM international conference on computer-aided design, San Jose, CA, pp 844–851
49. Mogal H, Qian H, Sapatnekar SS, Bazargan K (Nov 2007) Clustering based pruning for statistical criticality computation under process variations. In: Proceedings of the IEEE/ACM international conference on computer-aided design, San Jose, CA, pp 340–343
50. Yoshimura T, Kuh ES (Jan 1982) Efficient algorithms for channel routing. IEEE Trans Comput Aided Des Integr Circ Syst 1(1):25–35
51. Mogal H, Qian H, Sapatnekar SS, Bazargan K (Mar 2009) Fast and accurate statistical criticality computation under process variations. IEEE Trans Comput Aided Des Integr Circ Syst 28(3):350–363
52. Choi SH, Paul BC, Roy K (Jun 2004) Novel sizing algorithm for yield improvement under process variation in nanometer technology. In: Proceedings of the ACM/IEEE design automation conference, San Diego, CA, pp 454–459
53. Sinha D, Shenoy NV, Zhou H (Nov 2005) Statistical gate sizing for timing yield optimization. In: Proceedings of the IEEE/ACM international conference on computer-aided design, San Jose, CA, pp 1037–1042
54. Agarwal A, Chopra K, Blaauw D, Zolotov V (Jun 2005) Circuit optimization using statistical static timing analysis. In: Proceedings of the ACM/IEEE design automation conference, Anaheim, CA, pp 338–342
55. Raj S, Vrudhala SBK, Wang J (Jun 2004) A methodology to improve timing yield in the presence of process variations. In: Proceedings of the ACM/IEEE design automation conference, San Diego, CA, pp 448–453
56. Mani M, Devgan A, Orshansky M (Jun 2005) An efficient algorithm for statistical power under timing yield constraints. In: Proceedings of the ACM/IEEE design automation conference, Anaheim, CA, pp 309–314

57. Davoodi A, Srivastava A (Jul 2006) Variability driven gate sizing for binning yield optimization. In: Proceedings of the ACM/IEEE design automation conference, San Francisco, CA, pp 956–964
58. Singh J, Sapatnekar SS (Jan 2008) A scalable statistical static timing analyzer incorporating correlated non-Gaussian and Gaussian parameter variations. IEEE Trans Comput Aided Des Integr Circ Syst 27(1):160–173
59. Singh J, Nookala V, Luo T, Sapatnekar S (Jun 2005) Robust gate sizing by geometric programming. In: Proceedings of the ACM/IEEE design automation conference, Anaheim, CA, pp 315–320
60. Srivastava A, Sylvester D, Blaauw D (Jun 2004) Statistical optimization of leakage power consider process variations using dual-Vth and sizing. In: Proceedings of the ACM/IEEE design automation conference, San Diego, CA, pp 773–778
61. Srivastava A, Sylvester D, Blaauw D (Jun 2004) Power minimization using simultaneous gate sizing, dual Vdd and dual Vth assignment. In: Proceedings of the ACM/IEEE design automation conference, San Diego, CA, pp 783–787
62. Datta A, Bhunia S, Mukhopadhyay S, Banerjee N, Roy K (Mar 2005) Statistical modeling of pipeline delay and design of pipeline under process variation to enhance yield in sub-100 nm technologies. In: Proceedings of design, automation, and test in Europe, Munich, pp 926–931
63. Lee B, Wang L, Abadir MS (Jul 2006) Refined statistical static timing analysis through learning spatial delay correlations. In: Proceedings of the ACM/IEEE design automation conference, San Francisco, CA, pp 149–154
64. Wang L, Bastani P, Abadir MS (Jun 2007) Design-silicon timing correlation–a data mining perspective. In: Proceedings of the ACM/IEEE design automation conference, San Diego, CA, pp 385–389
65. Abranmovici M, Bradley P, Dwarakanath K, Levin P, Memmi G, Miller D (Jul 2006) A reconfigurable design-for-debug infrastructure for SoCs. In: Proceedings of the ACM/IEEE design automation conference, San Francisco, CA, pp 7–12
66. Tschanz JW, Kao JT, Narendra SG, Nair R, Antoniadis DA, Chandrakasan AP, De V (Nov 2002) Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage. IEEE J Solid-State Circuits 37:1396–1402
67. Tschanz JW, Narendra S, Nair R, De V (May 2003) Effectiveness of adaptive supply voltage and body bias for reducing the impact of parameter variations in low power and high performance microprocessors. IEEE J Solid-State Circuits 38:826–829
68. Tschanz JW, Narendra S, Keshavarzi A, De V (May 2005) Adaptive circuit techniques to minimize variation impacts on microprocessor performance and power. In: Proceedings of the IEEE international symposium on circuits and systems, Kobe, pp 23–26
69. Paul S, Krishnamurthy S, Mahmoodi H, Bhunia S (Nov 2007) Low-overhead design technique for calibration of maximum frequency at multiple operating points. In: Proceedings of the IEEE/ACM international conference on computer-aided design, San Jose, CA, pp 401–404
70. Paul S, Mahmoodi H, Bhunia S (Feb 2010) Low-overhead *Fmax* calibration at multiple operating points using delay-sensitivity-based path selection. ACM Trans Des Autom Electron Syst 15(2), Article 19:34 pp
71. Liu Q, Sapatnekar SS (Jun 2007) Confidence scalable post-silicon statistical delay prediction under process variations. In: Proceedings of the ACM/IEEE design automation conference, San Diego, CA, pp 497–502
72. Liu Q, Sapatnekar SS (Apr 2009) Synthesizing a representative critical path for post-silicon delay prediction. In: Proceedings of the International symposium on physical design, San Diego, CA, pp 183–190
73. Fishburn JP, Dunlop AE (Nov 1985) TILOS: a posynomial programming approach to transistor sizing. In: Proceedings of the IEEE international conference on computer-aided design, Santa Clara, CA, pp 326–328