Early Analysis of Timing Margins and Yield [*]

Khaled R. Heloue
Department of ECE
University of Toronto
Toronto, Ontario, Canada
khaled@eecg.utoronto.ca

Farid N. Najm
Department of ECE
University of Toronto
Toronto, Ontario, Canada
f.najm@utoronto.ca

ABSTRACT

Manufacturing process variations, leading to variability in circuit delay, can cause excessive timing yield loss if not accounted for. Many techniques have been proposed for statistical timing analysis, which operate at a late stage of the design, by which time many design decisions have already been made. In this paper, we develop an **early** approach to statistical timing and yield analysis. With early access to timing yield information, one can take corrective action at a time when it is still possible to do so. The proposed technique does not propagate distributions through the circuit. Instead, it provides "yield-specific" margins on the maximum and minimum nominal circuit delays (setup and hold margins) which, if applied during standard (deterministic) timing analysis, would guarantee the desired yield. Starting from a *generic circuit* representation based on *classes* of paths with different depths, we find a *lower bound* expression on the timing yield. This lower bound is guaranteed for unknown within-die correlations, and hence the approach can be applied pre-placement. We also propose a novel method that allows "controlled" budgeting of yield loss between setup and hold violations.

1. INTRODUCTION

Process variations have an impact on circuit delay variations, and consequently can cause timing yield loss. Traditionally, process variations have been taken care of in various ways. In microprocessors, it is typical to check circuit timing with *nominal* transistor files, and to specify some *timing margin* which should be left as slack between the nominal delays and the timing constraints, in order to account for process variations. In ASICs, the practice is to typically design circuits by making sure the chip passes the timing requirements at all *process corners*, including nominal, worst, and best cases of device behavior. If these settings are too pessimistic, then designers are forced to waste time and effort optimizing a circuit using design conditions that are too stringent.

There has been considerable discussion in the literature that the traditional methods of using process corners or using a timing margin are breaking down. For one thing, for microprocessors, where nominal process files are used and a timing margin needs to be left as slack, there are no easy ways to decide what the margin should be, to account for within-die variations which have become important recently. And if a given margin is used, there is no indication as to what the resulting timing yield would be. On the other hand, for ASICs, the number of corners is increasing, making it very expensive to explore all corners. Furthermore, this traditional corner analysis approach is a pass/fail approach and cannot handle *within-die* statistical variations [1].

Statistical techniques offer an alternative approach. Due to the increased importance of within-die variations, there has been an increased interest recently in tackling the timing yield problem by employing statistical techniques as part of the circuit timing analysis step [2, 3, 4, 5, 6, 7]. The aim is to include statistical delay variations as an extension to traditional STA leading to statistical static timing analysis (SSTA).

In early work [1, 3, 8, 9], within-die variations are assumed to be totally uncorrelated. This assumption is not true in practice, however it is usually hard to express the correlations between within-die parameter variations with a model built from process data. Different attempts to model correlations have been proposed: In [10], principal component analysis (PCA) has been used to de-correlate variations on a set of independent random variables. In [11], a quad-tree partitioning is used to express a region-wise spatial correlation among within-die variations. In [5], correlation is taken care of using a canonical model, where each variation is expressed in terms of global sources of variations. All these methods depend on placement information, and rely on extensive process data, which is not always easily available. Therefore, these types of post-placement, design specific, SSTA become final sign-off tools and are unusable during early circuit design. In [7], a classification of SSTA approaches is presented:

1. early process-specific SSTA based on generic paths. This can be applied early in the design flow, to establish timing margins for generic paths in the candidate technology, even before circuit design has started, and to possibly optimize the devices or the circuit style to reduce these margins.
2. early design-specific SSTA based on a given design in a given process. This can be applied pre-placement, during the circuit design stage. This would be perhaps the most heavily used type of SSTA.
3. late post-placement design-specific SSTA, for final sign-off.

In [4, 7], the first type of SSTA is explored, i.e., a process-specific method is proposed. The analysis is based on a "generic path" concept that is representative of critical paths in a given technology. Cauchy bounds were used in [4] to get a lower bound on the timing yield. Better bounds were proposed in [7], where correlation was handled using Slepian's Inequalities. An attempt to handle both setup and hold yields was made, but the two analyses were treated separately. In practice, both setup and hold constraints should be combined in the analysis of the total timing yield. In [6], it was suggested that some current post-placement SSTA techniques unnecessarily complicate the design flow. Emphasis was made that, early and simple SSTA techniques, which keep the current timing verification methodology, should be explored.

2. SCOPE OF THIS WORK

Early prediction of statistical delay variations can lead to educated early design decisions that can save time and effort in achieving timing closure. Recent SSTA techniques have contributed greatly to predicting the distribution of delay for circuits that have already been *specified*, *placed*, and whose within-die correlations have been extracted. Hence, these techniques are final sign-off tools as they are applied at a late stage of the design.

In this paper, we propose a simple pre-placement SSTA technique that can be applied at an early stage of the design flow. Since within-die correlations are unknown at the pre-placement stage, we will use the assumptions stated in [7], to guarantee a **lower bound** on the timing yield. Namely, it was proven in [7] that if systematic within-die variations of gates are assumed to be totally correlated along a path, and if systematic within-die variations between paths are assumed uncorrelated, then this would lead to a lower bound on the timing yield.

Our analysis will result in a selection of two "yield-specific" **timing margins**: a setup margin to be used on the nominal maximum path delay, and a hold margin to be used on the minimum nominal path delay in order to guarantee a desired timing yield.

[*] This project was supported in part by Intel Corp. and by Altera Corp.

In this work, we propose a flexible generic circuit representation based on multiple classes of generic paths. Such a representation can model virtually any circuit, by simple discretization of its path delays into specified classes. Setup yield and Hold yield are both studied *and* combined together to give an expression for a lower bound on the total timing yield. Using this resulting expression, we allow *controlled budgeting* of yield loss between setup and hold violations by careful selection of setup and hold margins.

Hence, for a desired timing yield $\mathcal{Y}$, we will work backward from the lower bound to find the required timing margins (τ_s, τ_h) to be used. Note that if these margins are used, the timing yield would be at least equal to $\mathcal{Y}$ since we're using a lower bound on the yield to find our margins and the actual yield will be higher. As a result, our approach preserves existing static timing methodology, by providing timing margins to be used during circuit design, in order to account for process induced delay variations at an early stage of the design.

3. GENERIC CIRCUIT DESCRIPTION

For early analysis, given the absence of complete details of the circuit implementation, it becomes necessary to construct high-level models that capture circuit characteristics. For the timing yield problem at hand, a model was used in [12] in which a *generic critical path* model was employed as a way to capture the dependence of circuit worst-case delays in a given technology, in the absence of a specific circuit instance. In this model, a circuit is assumed to consist of a large number of identical paths, whose delay is representative of critical path delays in the target design. Each of these generic paths consists of a specific number of identical stages (a stage is a logic gate and the interconnect at its output). This model has been successfully employed in the industry, and we also offer independent verification of the validity of the generic path model in [13], where it is shown that the statistics of a large number of such generic paths match very well the statistics of critical path delays in the actual circuit. This model was later used in [4], where the generic path model was used to allow computation of the setup margins required for a single class of paths. In this section, we will first simply *review* the construction of the generic path model, with reference to previous work, and then in section 3.4 we extend this previous work in order to allow for multiple *classes* of generic paths. We will also handle both setup and hold violations.

3.1 Generic Parameter Model

For a given circuit element or layout feature i, let $X(i)$ be a zero-mean Gaussian *random variable* (RV) that denotes the variation of a certain parameter of this element from its nominal (mean) value. Thus, for example, $X(i)$ may represent channel length variations of transistor i. It is standard practice [14] to express the correlation between these RVs by first breaking up the variations into *die-to-die* and *within-die* components, as follows:

$$X(i) = X_{dd} + X_{wd}(i) \tag{1}$$

The die-to-die component X_{dd} is an independent zero-mean Gaussian RV that takes the same value for all instances of this element on a given die, irrespective of location. The within-die component $X_{wd}(i)$ is a zero-mean Gaussian which can take different values for different instances of that element on the same die. This leads to the following relationship between the variances:

$$\sigma^2(i) = \sigma^2_{dd} + \sigma^2_{wd}(i) \tag{2}$$

Then, the within-die component is further broken down into two components, a *systematic* component and a "random" component:

$$X_{wd}(i) = X_{wds}(i) + X_{wdr}(i) \tag{3}$$

where, for each i, the random component $X_{wdr}(i)$ is an *independent* zero-mean Gaussian. A similar relationship follows for the variances:

$$\sigma^2_{wd}(i) = \sigma^2_{wds}(i) + \sigma^2_{wdr}(i) \tag{4}$$

We can write $X_{wds}(i)$ in the following way:

$$X_{wds}(i) = \sigma_{wds}(i)\, Z_{wds}(i) \tag{5}$$

where $Z_{wds}(i)$ are correlated standard normal RVs (mean 0, variance 1). Hence, our model for parameter variation $X(i)$ consists of an independent zero mean die-to-die component X_{dd} with variance σ^2_{dd}, a correlated systematic within-die component $X_{wds}(i)$ with variance $\sigma^2_{wds}(i)$, and an independent random within-die component $X_{wdr}(i)$ with variance $\sigma^2_{wds}(i)$.

Starting from process parameter variations captured with this model, we show next that gate delay variations and path delay variations can be modeled using the same generic parameter model.

3.2 Gate Delay Model

In general, there is a nonlinear relationship between gate delay and transistor parameters. Simple circuit simulations, however, reveal that this nonlinearity is not strong, especially for small transistor parameter variations. Therefore, we will simply assume that gate delay is linearly dependent on the process, and hence is Gaussian with mean equal to its nominal value.

For all the transistors within one logic gate, we assume that their channel length variations are captured with a *single* RV $L(i)$ and their threshold voltage variations are captured with a *single* RV $V(i)$, which we assume to be independent of each other. For each gate, we can extract sensitivities to the different varying process parameters using circuit simulation. Hence, if $D(i)$ is the deviation of the delay of logic gate i from it's mean (nominal) delay, we have:

$$D(i) = \alpha L(i) + \beta V(i) \tag{6}$$

where α and β are sensitivity parameters, with suitable units, that one can easily obtain from circuit simulation of a representative logic gate. Notice that, for the above process parameters, $\alpha > 0$ and $\beta > 0$. As a result, we can express the statistical variations in delay of gate i as:

$$D(i) = D_{dd} + D_{wds}(i) + D_{wdr}(i) \tag{7}$$

so that:

$$\begin{aligned} D_{dd} &= \alpha L_{dd} + \beta V_{dd} \\ D_{wds}(i) &= \alpha L_{wds}(i) + \beta V_{wds}(i) \\ D_{wdr}(i) &= \alpha L_{wdr}(i) + \beta V_{wdr}(i) \end{aligned} \tag{8}$$

and:

$$\begin{aligned} \sigma^2_{dd,D} &= \alpha^2 \sigma^2_{dd,L} + \beta^2 \sigma^2_{dd,V} \\ \sigma^2_{wds,D}(i) &= \alpha^2 \sigma^2_{wds,L}(i) + \beta^2 \sigma^2_{wds,V}(i) \\ \sigma^2_{wdr,D}(i) &= \alpha^2 \sigma^2_{wdr,L}(i) + \beta^2 \sigma^2_{wdr,V}(i) \end{aligned} \tag{9}$$

These equations provide a way in which the statistical model of gate delay (i.e., its three variances) can be computed from the underlying statistical model of transistor parameters.

3.3 Generic Path Delay Model

Consider a generic path of N logic stages (a stage is a logic gate and the interconnect at its output). We will only focus on gate delays, and only on transistor L and V variations. The methodology can be easily applied when more device parameters are of interest, or when interconnect parameter variations are to be included as well. Let $D_N(j)$ denote the deviation of the delay of path j from its mean (nominal) value. For the sake of simplicity assume identical gates along a path. Hence path delay variations can be written in the following form:

$$D_N(j) = \sum_{i=1}^{N} D(i) = N D_{dd} + \sum_{i=1}^{N} D_{wds}(i) + \sum_{i=1}^{N} D_{wdr}(i) \tag{10}$$

Recall from [7] that the generic path delay variance is given by:

$$\sigma^2_{D_N}(j) = \sigma^2_{dd,D_N} + \sigma^2_{wds,D_N}(j) + \sigma^2_{wdr,D_N}(j) \tag{11}$$

where:

$$\begin{aligned}
\sigma^2_{dd,D_N} &= N^2\sigma^2_{dd,D} \\
\sigma^2_{wds,D_N}(j) &= N^2\sigma^2_{wds,D}(j) \\
\sigma^2_{wdr,D_N}(j) &= N\sigma^2_{wdr,D}(j)
\end{aligned}$$

Note that we have assumed the systematic within-die variations of gate delays to be totally correlated along the generic path. This assumption is not "simplistic"; given the absence of correlation information during pre-placement this assumption is needed to maximize path variance and hence guarantees a lower bound on the yield as was shown in [7]. Note the averaging of variations in the random within-die variance (N instead of N^2) due to random cancellations along the path. With this, we have a full statistical model of path delay, that is captured using the generic parameter model described earlier with its three components of variations.

3.4 Generic Circuit Representation

The generic path model was useful to study setup margins separately, or hold margins separately. However, in order to study the interactions among long paths and short paths, and the tradeoffs between setup and hold margins, a more detailed model is required. We now provide such a model, what we refer to as a *generic circuit* model, which involves the specification of *classes* of paths, as an extension of the generic path model.

According to this model, a generic circuit is a collection of a certain number M of classes of paths. Each class $j = 1, 2, \ldots, M$, consists of n_j identical generic paths [7] of *depth* N_j. The depth is the number of stages along a path; a stage is a logic gate or cell along with its fanout interconnect network. In a logic circuit, if a logic path is well-optimized, then every stage is loaded by a total output capacitance which is 3-4 times as big as its own input capacitance, hence the standard FO4 configuration, so that the stages on a path have approximately the same delay. Motivated by this, we use the same assumption of [7] that all stages have the same nominal delay and the same sensitivity to process variables (sensitivities are terms such as α and β in (6)). Thus all stages are identical, and all paths within the same class are identical; paths in different classes have different depth.

The classes can be viewed as the result of discretization of path delays into "bins" of paths with equal or similar delay. We further assume the existence of a "circuit depth histogram" which provides the fraction of the total number of paths that each class constitutes. We express these fractions as probabilities:

$$\mathcal{P}\{N = N_j\} = \gamma_j, \quad j = 1, 2, \ldots, M \qquad (12)$$

where N is a discrete random variable that represents path depth, N_j is the depth of paths in class j, and γ_j is the fraction of total paths that have a depth of N_j. If n is the total number of paths in a circuit, then the number of paths n_j in class j is:

$$n_j = \gamma_j \times n \quad \text{and} \quad \sum_{j=1}^{M} \gamma_j = 1 \qquad (13)$$

An example is given in Table 1, which, for a test circuit $\mathcal{A}$ with 5 classes, lists the values of N_j and γ_j for every class. Also shown is the corresponding path count n_j in every class based on an assumed total path count of $10,000$. The table also shows the nominal path delay $D^{(nom)}_{N_j}$ in every class. In the following sections, we will present the analysis leading to early timing verification of circuits represented in this generic circuit representation. Based on such a circuit model, we will provide an analysis that predicts the margins that need to be left to guarantee a desired timing yield. We will also validate our margins with Monte Carlo simulations. We consider the generic circuit to be an intuitive and useful model for logic circuits, in the same way as the generic path has been found to be useful. Again, the reader is referred to the Appendix for validation of the utility of the earlier generic path model.

Table 1: Generic Circuit with 5 Classes of Paths

j	1	2	3	4	5
N_j	4	5	6	8	9
$D^{(nom)}_{N_j}$	8 (ns)	10 (ns)	11.1 (ns)	12 (ns)	13.5 (ns)
γ_j	30%	10%	10%	20%	30%
n_j	3000	1000	1000	2000	3000

4. TIMING YIELD

Informally, timing yield is the probability that circuit delay is within the specified timing constraints. These constraints can be upper limits on the maximum circuit delay, lower limits on the minimum circuit delay, or both in what is known as interval or two-sided constraints. Yield loss is incurred if circuit timing falls outside the constraints. Note that constraints on the maximum circuit delay are also known as setup time constraints, and constraints on the minimum delay are known as hold time constraints. In the next two sections, we will separately define the setup yield and the hold yield. After finding their corresponding expressions for circuits that are generically represented using our model, we will combine the two yields and present a way to predict statistical margins that should be left as slacks to get a desired timing yield, taking both setup and hold constraints into account.

5. SETUP YIELD

Setup yield Y_S can be written in the following informal way:

$$Y_S = \mathcal{P}\{\text{All Path delays are less than max constraint}\} \qquad (14)$$

Recall that in our generic circuit representation with multiple classes, each class consists of identical generic paths whose delay variation is modeled using the generic parameter model. Let $D_{N_j}(i)$ be the path delay variation of path i in class j. Then we can express path delay variation using the parameter model in the following way:

$$\begin{aligned}
D_{N_j}(i) &= D^{(dd)}_{N_j} + D^{(wds)}_{N_j}(i) + D^{(wdr)}_{N_j}(i) &(15)\\
&= D^{(dd)}_{N_j} + D^{(wd)}_{N_j}(i) &(16)
\end{aligned}$$

where $j = 1, \ldots, M$ is the class index, $i = 1, \ldots, n_j$ is the path index in class j, and the right hand side is simply the three components of variability. Note that in the second line, we have combined the systematic and random components into one within-die component of variability whose variance is equal to the sum of the variances of the systematic and random components. Now combining the above model for path delay with (14), gives the formal expression for setup yield:

$$Y_S = \mathcal{P}\left\{ \bigcap_{j=1}^{M} \left[\bigcap_{i=1}^{n_j} \Big(D_{N_j}(i) \le \tau_j \Big) \right] \right\} \qquad (17)$$

where the $\bigcap$ operator is used to indicate that we are interested in the probability of all these *joint* events.

In other words, the setup yield is the probability that, *over all* classes M, all path delay variations in each class are less than a timing margin τ_j (specific to each class j). This margin is the amount of "padding" or slack, that should be kept between the maximum delay constraint and the nominal path delay in class j, to account for delay variations. Using the above expression for setup yield, we will later show how we can predict a unique setup margin τ_s that should be left as slack on the maximum nominal path delay, in order to guarantee a desired setup yield for the circuit under consideration.

5.1 Yield Analysis

Starting from (17), and noting that the second intersection defines a statistical max operation, we can write the following:

$$Y_S = \mathcal{P}\left\{ \bigcap_{j=1}^{M} \left[\max_{i=1}^{n_j} \Big(D_{N_j}(i) \Big) \le \tau_j \right] \right\} \qquad (18)$$

In [7], it was proven, using *Slepian's Inequality*, that if the systematic within-die delay variations are assumed to be uncorrelated, then this would lead to a lower bound on the statistical max and min operations. We will use this result to write the following *lower bound* on the setup yield Y_S:

$$Y_S \geq \mathcal{P}\left\{ \bigcap_{j=1}^{M} \left[\max_{i=1}^{n_j} \left(D_{N_j}^{(dd)} + D_{N_j}^{(wd)}(i) \right) \leq \tau_j \right] \right\} \tag{19}$$

where $D_{N_j}^{(wd)}(i)$ are assumed to be uncorrelated for different i's (hence independent because Gaussian).

5.1.1 Asymptotic Convergence

Let Z_j be an RV equal to the maximum delay variation for class j. This is shown in the following equation:

$$\begin{aligned} Z_j &= \max_{i=1}^{n_j} \left(D_{N_j}^{(dd)} + D_{N_j}^{(wd)}(i) \right) && (20) \\ &= D_{N_j}^{(dd)} + \max_{i=1}^{n_j} \left(D_{N_j}^{(wd)}(i) \right) && (21) \\ &= D_{N_j}^{(dd)} + W_j && (22) \end{aligned}$$

where $W_j = \max_{i=1}^{n_j} \left(D_{N_j}^{(wd)}(i) \right)$. Note that because the die-to-die component is the same for all i, then we can take it outside the max operation. We will now prove that as $n_j \to \infty$, the variance of W_j will go to zero, therefore we can approximate W_j by a constant μ_j equal to its 50^{th} percentile $W_{j,\,50\%}$.

Let Y be the maximum of n independent identically distributed (iid) standard normal RVs. In studying the asymptotic properties of such an RV, it has been shown in [15], that $V = l_n(Y - l_n)$ has a unique limiting distribution that is independent of n, where l_n is a coefficient (not an RV) that is proportional to $\sqrt{2 \log n}$. Since V has a distribution that is independent of n, its variance s^2 is independent of n. Therefore:

$$s^2 = Var\left[l_n(Y - l_n)\right] = {l_n}^2 \; Var[Y] \tag{23}$$

and

$$Var[Y] = \frac{s^2}{l_n^2} \to 0 \quad \text{as} \quad n \to \infty \tag{24}$$

which is true because l_n^2 is proportional to $\log n$.

This result can be applied to W_j, which is the maximum of n_j independent zero-mean gaussian RVs $D_{N_j}^{(wd)}(i)$, after scaling each by its standard deviation to transform them to standard normal RVs. Since the variance of W_j goes to zero for large n_j, then we can approximate it by its 50^{th} percentile $W_{j,\,50\%}$ which we denote by μ_j. Now let $\hat{Z_j}$ be a random variable such that:

$$\hat{Z_j} = D_{N_j}^{(dd)} + \mu_j \tag{25}$$

Then Z_j converges to $\hat{Z_j}$ for large n_j. Note that μ_j is given by:

$$\mu_j = W_{j,\,50\%} = \Phi^{-1}\left((0.5)^{\frac{1}{n_j}} \right) \times \sigma_{N_j}^{(wd)} \tag{26}$$

where $\sigma_{N_j}^{(wd)}$ is the standard deviation of $D_{N_j}^{(wd)}$.

Fig. 1 shows plots of the distributions of Z_j (solid) and $\hat{Z_j}$ (dashed) for increasing values of $n_j = 10, 100, 1e3, 1e6, 1e7, 1e9$. The approximation is very tight for n_j greater than 100, and the two become indistinguishable for larger n_j. Since we anticipate each class of paths to contain hundreds if not thousands of paths, then the approximation is very good and $\hat{Z_j}$ will be used in lieu of Z_j in the analysis below.

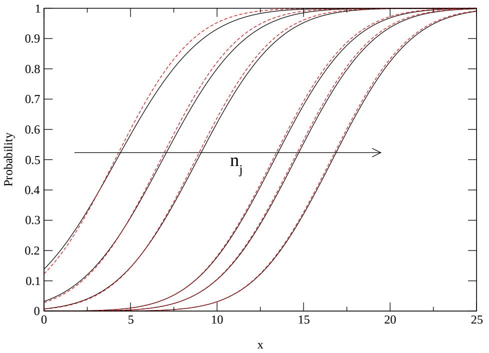

Figure 1: Approximate and true distributions

5.1.2 Setup Yield Expression

Starting from (19), and replacing the expression inside the brackets by $\hat{Z_j}$, we get the following expression for the setup yield:

$$\begin{aligned} Y_S &\geq \mathcal{P}\left\{ \bigcap_{j=1}^{M} \left(\hat{Z_j} \leq \tau_j \right) \right\} && (27) \\ &= \mathcal{P}\left\{ \bigcap_{j=1}^{M} \left(D_{N_j}^{(dd)} \leq \tau_j - \mu_j \right) \right\} && (28) \end{aligned}$$

Recall that the die-to-die component of variation affects all the circuit in the same way, i.e. the variation is governed by a single random variable.

$$D_{N_j}^{(dd)} = \sigma_{N_j}^{(dd)} Z_0 \tag{29}$$

where $\sigma_{N_j}^{(dd)}$ is the standard deviation of the die-to-die component and Z_0 is a standard normal RV. Note that $\sigma_{N_j}^{(dd)}$ depends on j and is different for different classes. Combining (28) and (29) gives the following expression for setup yield:

$$\begin{aligned} Y_S &\geq \mathcal{P}\left\{ \bigcap_{j=1}^{M} \left(Z_0 \leq \frac{\tau_j - \mu_j}{\sigma_{N_j}^{(dd)}} \right) \right\} && (30) \\ &= \mathcal{P}\left\{ Z_0 \leq \min_{j=1}^{M} \left(\frac{\tau_j - \mu_j}{\sigma_{N_j}^{(dd)}} \right) \right\} && (31) \\ &= \Phi\left(\min_{j=1}^{M} \left(\frac{\tau_j - \mu_j}{\sigma_{N_j}^{(dd)}} \right) \right) = Y_o && (32) \end{aligned}$$

where Y_o is the desired lower bound on the setup yield Y_S.

5.1.3 Setup Margin τ_s

The above equation for setup yield Y_S is very important because it will allow us to budget each margin τ_j and to finally assign a unique setup margin τ_s for the circuit which guarantees a desired setup yield, as follows. Let a_{min} be such that:

$$a_{min} = \min_{j=1}^{M} \left(\frac{\tau_j - \mu_j}{\sigma_{N_j}^{(dd)}} \right) \tag{33}$$

Recall that μ_j and $\sigma_{N_j}^{(dd)}$ are determined and fixed for each class, but τ_j is to be determined. If a setup yield of, say $Y_S = \mathcal{Y}$, is required, then we can use the lower bound Y_o to get a value of a_{min} using (32), by setting $Y_o = \mathcal{Y}$, then:

$$a_{min} = \Phi^{-1}(\mathcal{Y}) \tag{34}$$

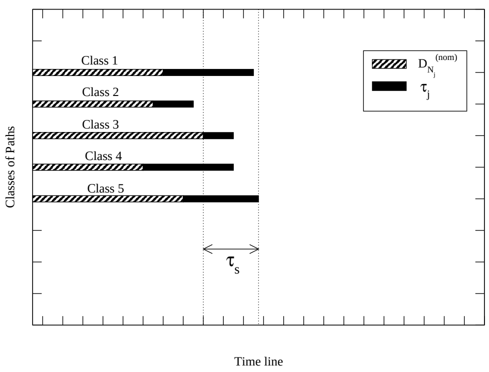

Figure 2: Setup Margin Computation

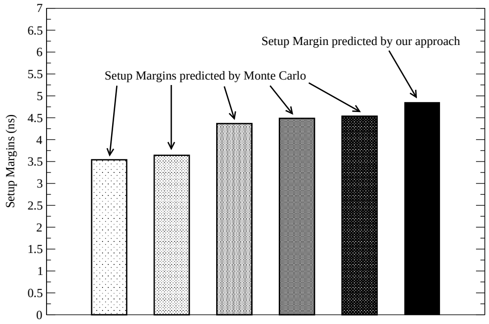

Figure 3: Monte Carlo Setup Margins for 99.73% Setup Yield

We can then use this value of $a_{min}$to budget the yield between the different classes. In order to reduce the design effort for class j, one would like the margin τ_j to be small, and so we require:

$$\frac{\tau_j - \mu_j}{\sigma_{N_j}^{(dd)}} = a_{min}, \quad j = 1, \ldots, M \tag{35}$$

where effectively we require the different class contributions to the yield to be equal. In this way, we have "relaxed" the yield requirements for each class, while ensuring that the desired yield overall is met. As mentioned, this is important, since it has the benefit of reducing the design effort required to meet the timing and yield constraints.

This gives us the following expression for each class margin:

$$\tau_j = \sigma_{N_j}^{(dd)} \times a_{min} + \mu_j, \quad j = 1, \ldots, M \tag{36}$$

Recall that the margin τ_j represents how much designers should leave as slack between the nominal path delay of class j and the maximum delay constraint (setup constraint). Let $D_{N_j}^{(nom)}$ be the nominal path delay of the generic paths of class j. Then we can define a unique setup margin τ_s to be padded on top of the maximum nominal path delay for all classes, in order to guarantee that the desired setup yield is attained. This is expressed in the following way:

$$\tau_s = \max_{j=1}^{M} \left(D_{N_j}^{(nom)} + \tau_j \right) - \max_{j=1}^{M} \left(D_{N_j}^{(nom)} \right) \tag{37}$$

which may be explained with the help of Fig. 2. For each class with nominal path delay of $D_{N_j}^{(nom)}$ (hashed bars), we get the class margins τ_j (solid bars) using (36). We then apply (37), subtracting the maximum nominal delay from the maximum delay with added margin τ_j to get the setup margin τ_s. Therefore, we are providing a unique margin to be allowed on the nominal maximum circuit delay, in order to guarantee the required yield. Note that no other path delays exceed the point defined by the maximum nominal delay with the added setup margin τ_s.

5.2 Monte Carlo Validation

In this section, we will validate our approach with Monte Carlo analysis. For this matter, we will use circuit $\mathcal{A}$ that was defined in Table 1. This circuit has five classes of paths with different depths, nominal delay values, and numbers of paths. We also assume that two process parameters are varying (channel length and threshold voltage), and that we have gate delay sensitivities to these process parameters for each class of paths. In our experiment, we varied the parameters in such a way to get $20\% - 27\%$ 3σ gate delay variation. We have also assumed a breakdown of (50%, 25%, 25%) of total parameter variance into die-to-die, systematic within-die, and random within-die variances respectively. Applying the analysis described in section 3, we can get the path delay model for each class of paths.

We will assume that the desired setup yield is $\mathcal{Y} = 99.73\%$. Applying our analysis on circuit $\mathcal{A}$, we get a setup margin $\tau_s = 4.85$ ns that corresponds to 36% of the maximum nominal path delay of 13.5 ns found in Table 1. Recall that this margin is valid for any circuit placement, and hence any within-die correlation since it was derived from the lower bound. It is therefore a conservative timing margin; this means that for a given correlations setting, the true setup margin is expected to be less than the margin that we have predicted. We will verify this statement through Monte Carlo simulations for different within-die correlations. To this end, we have considered a $k \times k$ grid on which we place all the $10,000$ paths of circuit $\mathcal{A}$. We also use a number (p) of RVs to model the within-die correlations between paths. If p is large, within-die correlation between paths is weak. If p is small, correlation is strong. We have run Monte Carlo for $10,000$ samples and generated the distribution of maximum path delay. From this distribution, we determined the 99.73% delay percentile and subtracted from it the maximum nominal path delay to get the setup margin. We have run this experiment for different values of p to emulate situations ranging from high correlation, to almost independence. Monte Carlo analysis predicted setup margins ranging from 3.54 ns (for strong correlations) to 4.54 ns (for weak correlations) as opposed to our 4.85 ns predicted margin. Fig. 3 shows the predicted margins for the experiments that we performed. Note that the left-most Monte Carlo margin corresponds to the case of strong correlations, and the right most-Monte Carlo margin corresponds to the case of weak correlations.

6. HOLD YIELD

In the previous section, we presented in detail the analysis that led to the final expression for setup yield Y_S, and the corresponding setup margin τ_s. In this section, we will briefly repeat the same analysis for the hold yield. Recall that the hold yield can be informally defined in the following way:

$$Y_S = \mathcal{P}\{\text{All Path delays are greater than min constraint}\} \tag{38}$$

Using the same analysis as before, we can rewrite the hold yield to be:

$$Y_H = \mathcal{P}\left\{ \bigcap_{j=1}^{M} \left[\bigcap_{i=1}^{n_j} \left(D_{N_j}(i) \geq -\tau_j^{'} \right) \right] \right\} \tag{39}$$

where $\tau_j^{'}$ is the margin (positive) that should be left between the minimum constraint and the nominal path delay for class j. The above equation can be transformed in the same way as previously done:

$$\begin{aligned} Y_H &= \mathcal{P}\left\{ \bigcap_{j=1}^{M} \left[\min_{i=1}^{n_j} \left(D_{N_j}(i) \right) \geq -\tau_j^{'} \right] \right\} && (40) \\ &= \mathcal{P}\left\{ \bigcap_{j=1}^{M} \left[D_{N_j}^{(dd)} + \min_{i=1}^{n_j} \left(D_{N_j}^{(wd)}(i) \right) \geq -\tau_j^{'} \right] \right\} && (41) \end{aligned}$$

Again using Slepian's Inequality [7], the assumption that $D_{N_j}^{(wd)}(i)$ are independent will give us a lower bound on the hold yield. This allows us again to approximate the minimum of n_j independent gaussian RVs by a constant $\mu_j^{'}$ that is equal to $U_{j,\,50\%}$, the 50^{th} percentile of the distribution of $U_j = \min_{i=1}^{n_j}\left(D_{N_j}^{(wd)}(i)\right)$ [15]:

$$\mu_j^{'} = U_{j,\,50\%} = \Phi^{-1}\left(1-(0.5)^{\frac{1}{n_j}}\right) \times \sigma_{N_j}^{(wd)} \tag{42}$$

The final expression for the hold yield is:

$$Y_H \;\geq\; 1-\mathcal{P}\left\{ Z_0 \leq \max_{j=1}^{M}\left(\frac{-\tau_j^{'}-\mu_j^{'}}{\sigma_{N_j}^{(dd)}}\right)\right\} \tag{43}$$

$$= \; 1-\Phi\left(\max_{j=1}^{M}\left(\frac{-\tau_j^{'}-\mu_j^{'}}{\sigma_{N_j}^{(dd)}}\right)\right) \tag{44}$$

In the same way as before, we will choose a b_{max} to guarantee a desired hold yield of say $Y_H \geq Y_o = \mathcal{Y}$:

$$b_{max} \;=\; \max_{j=1}^{M}\left(\frac{-\tau_j^{'}-\mu_j^{'}}{\sigma_{N_j}^{(dd)}}\right) = \Phi^{-1}\left(1-\mathcal{Y}\right) \tag{45}$$

Then we can budget $\tau_j^{'}$ in such a way that:

$$\frac{-\tau_j^{'}-\mu_j^{'}}{\sigma_{N_j}^{(dd)}} = b_{max}, \quad j=1,\ldots,M \tag{46}$$

This gives us the margins that should be left between the minimum delay constraint (hold constraint) and the nominal path delay of every class of paths:

$$\tau_j^{'} = -\sigma_{N_j}^{(dd)} \times b_{max} - \mu_j^{'}, \quad j=1,\ldots,M \tag{47}$$

Note here that $\tau_j^{'}$ is positive because b_{max} and $\mu_j^{'}$ are negative.

Finally, we combine the above margins with the values of nominal path delay $D_{N_j}^{(nom)}$ to get a unique hold margin τ_h for the circuit. This margin should be left as slack between the minimum constraint and the minimum nominal path delay over all classes:

$$\tau_h = \min_{j=1}^{M}\left(D_{N_j}^{(nom)}\right) - \min_{j=1}^{M}\left(D_{N_j}^{(nom)} - \tau_j^{'}\right) \tag{48}$$

6.1 Monte Carlo Validation

We have repeated the same Monte Carlo experiment described previously and determined the distribution of the minimum path delay for different ranges of within-die path to path correlation, going from strong to weak correlation. From this distribution, we have recorded the 99.73% delay percentile and subtracted it from the nominal minimum path delay in order to determine the hold margin. Monte Carlo simulations predicted margins ranging from 1.56 ns (for high correlation) to 2.05 ns (for weak correlation), while our approach predicted a margin of 2.22 ns corresponding to 28% of the nominal minimum delay of 8 ns in Table 1. Fig. 4 shows Monte Carlo predicted margins for different within-die correlations, going from strong (left) to weak (right). It also shows that the predicted hold margin is valid as a conservative choice under unknown within-die correlations.

7. TOTAL TIMING YIELD

In the previous two sections, we presented a way to verify that a desired setup yield *or* hold yield is met. The method is simple, in that it provides a setup margin τ_s, and a hold margin τ_h, that should be allowed on the nominal maximum and minimum path delays respectively, to guarantee desired yields. Both setup and hold analyses were presented as separate entities, which means that when we computed the setup yield expression, we did not take into consideration hold yield.

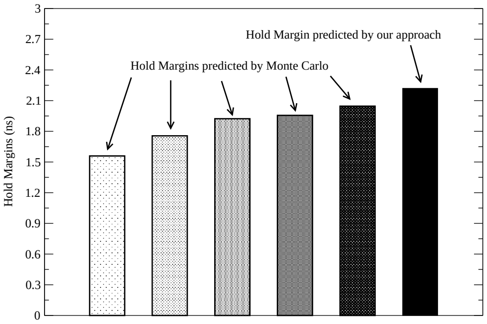

Figure 4: Monte Carlo Hold Margins for 99.73% Hold Yield

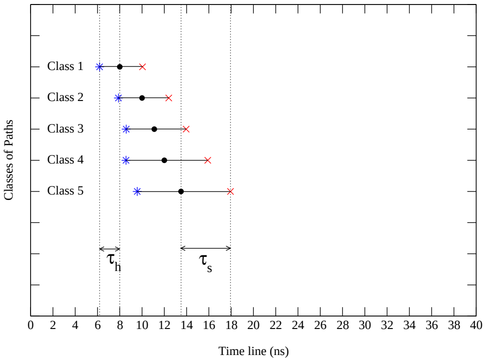

Figure 5: Setup and Hold Margins Selection, total yield $Y_T = 95\%$, $\xi = 0.8$

We now combine the two analyses together, and give an expression for the total yield Y_T, which captures both setup and hold yield loss. This provides control over the amount of yield loss from either setup or hold violations. This means that if hold yield is dear, one can pick the margins in such a way that *all* timing yield loss is due to setup yield loss, and vice versa.

We now show how this can be done. The total yield is:

$$Y_T = \mathcal{P}\{ \bigcap_{j=1}^{M} \left[\, (D_{N_j}^{(dd)} + \min_{i=1}^{n_j}(D_{N_j}^{(wd)}(i)) \geq -\tau_j^{'}) \bigcap \right. \\ \left. (D_{N_j}^{(dd)} + \max_{i=1}^{n_j}(D_{N_j}^{(wd)}(i)) \leq \tau_j) \,\right] \} \tag{49}$$

Using Slepian's inequality for two-sided constraints [16] that guarantees a lower bound when the within-die variations are assumed uncorrelated, and then applying the approximations that we presented earlier, mainly replacing the minimum and maximum operations by their constant estimators, gives the final expression for the total yield:

$$Y_T \;\geq\; \mathcal{P}\left\{\bigcap_{j=1}^{M}\left[-\tau_j^{'}-\mu_j^{'} \leq D_{N_j}^{(dd)} \leq \tau_j - \mu_j\right]\right\} \tag{50}$$

$$= \; \mathcal{P}\left\{b_{max} \leq Z_0 \leq a_{min}\right\} \tag{51}$$

$$= \; \Phi\left(a_{min}\right) - \Phi\left(b_{max}\right) \tag{52}$$

where μ_j, $\mu_j^{'}$, Z_0, b_{max}, and a_{min} are as defined before. Note that we need to assign values for a_{min} and b_{max} in order to guarantee a desired total yield Y_T. Once we have chosen their values, the same previous analysis applies, i.e. find τ_j and $\tau_j^{'}$ from a_{min} and b_{max} using (36) and (47) respectively, then determine the setup margin τ_s and the hold margin τ_h using (37) and (48) respectively.

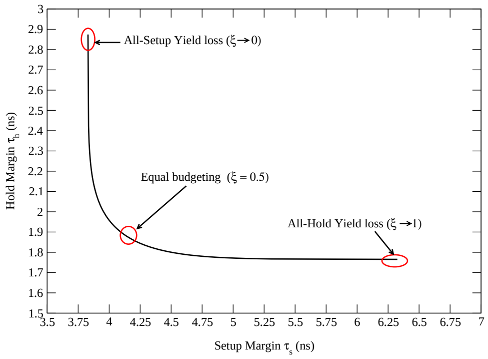

Figure 6: Yield Loss Budgeting: sweep over $\xi \; \epsilon \; (0\ ,\ 1)$, $Y_T = 95\%$

7.1 Yield Loss Budgeting

Looking at (52), it is easy to see that if a total yield of say $Y_T \geq 95\%$ is desired, then different solutions (a_{min}, b_{max}) exist. For each solution, the amount of yield loss from setup and hold violations differ, but the total yield loss is the same. Let L_T, L_S, and L_H be the total, setup, and hold yield losses respectively. Then we can write:

$$\begin{aligned} L_T &= 1 - Y_T = 1 - \Phi\left(a_{min}\right) + \Phi\left(b_{max}\right) && (53) \\ L_S &= 1 - Y_S = 1 - \Phi\left(a_{min}\right) && (54) \\ L_H &= 1 - Y_H = \Phi\left(b_{max}\right) && (55) \end{aligned}$$

Note that the total yield loss equals to the sum of yield loss from setup and hold. We now define the "yield loss budget" ξ in $[0, 1]$, to be the fraction of total yield loss induced by hold violations. This means that:

$$\begin{aligned} L_H &= \xi \; L_T && (56) \\ L_S &= (1 - \xi) \; L_T && (57) \end{aligned}$$

Combining the above four equations enables us to get a unique (a_{min}, b_{max}), and hence a unique (τ_h, τ_s) solution for a specific yield loss budget ξ:

$$\begin{aligned} b_{max} &= \Phi^{-1}\left(\xi \; L_T\right) && (58) \\ a_{min} &= \Phi^{-1}\left(1 - (1 - \xi) \; L_T\right) && (59) \end{aligned}$$

Hence, for a total yield of 95%, and a yield loss budget of $\xi = 0.8$, we can use the above equations with $L_T = 5\%$ to get specific values for a_{min} and b_{max}. Then, using the previous analysis for setup and hold yield, we get a setup and hold margin (τ_s, τ_h) that should be used for the circuit to meet timing constraints with probability $Y_T = 95\%$.

Fig. 5 shows how to compute τ_s and τ_h, for a yield of 95% and a yield loss budget $\xi = 0.8$. The filled dots represent nominal path delays for each class as listed in Table 1. The cross and star represent the "padding" or class margins τ_j and $\tau_j^{'}$ that need to be kept between the nominal delays and the constraints. The resulting margins are $\tau_s = 4.44$ ns (33% of the maximum nominal delay of 13.5 ns) and $\tau_h = 1.81$ ns (23% of the minimum nominal delay of 8 ns). The above results are obtained for the same 3σ variation of $20\% - 27\%$ in nominal gate delay that was used for the previous two experiments.

Fig. 6 shows a plot of τ_h versus τ_s, for the same experiment for a desired yield of 95%, by sweeping the yield loss budget ξ between 0^+ and 1^-. As expected, when the setup margin is increased causing setup yield loss to decrease, we can reduce the hold margin and incur more hold yield loss while maintaining the same total yield. This gives designers more flexibility in the choice of their margins.

8. CONCLUSION

An early analysis technique was proposed, which allows one to *determine* and *budget* the setup and hold margins required for a proposed chip design. The technique works irrespective of (and for any possible setting of) the within-die correlations, which allows one to handle situations of unknown correlations as well as pre-placement analysis. The process requires up-front specification of the target process technology and the circuit style, through a *generic circuit description* in terms of any number of *classes of generic paths*. As a result, the proposed technique provides setup and hold timing margins which, if observed for the longest and shortest nominal path delays during subsequent circuit design, would allow this chip to meet the timing yield target. These margins are not unique, but may be traded-off against each other, so that one may allow more setup yield loss than hold yield loss, or vice-versa, by simply changing a single parameter.

9. REFERENCES

[1] A. Gattiker, S. Nassif, R. Dinakar, and C. Long. Timing yield estimation from static timing analysis. In *IEEE Int'l Symp. on Quality Electronic Design*, pages 437–442, San Jose, CA, March 26-28 2001.

[2] J. A. G. Jess, K. Kalafala, W. R. Naidu, R. H. J. M. Otten, and C. Visweswariah. Statistical timing for parametric yield prediction of digital integrated circuits. In *Design Automation Conf.*, pages 932–937, June 2-6 2003.

[3] A. Agarwal, D. Blaauw, V. Zolotov, and S. Vrudhula. Computation and refinement of statistical bounds on circuit delay. In *Design Automation Conf.*, pages 348–353, Anaheim, CA, June 2-6 2003.

[4] F. N. Najm and N. Menezes. Statistical timing analysis based on a timing yield model. In *Design Automation Conference*, pages 460–465, San Diego, CA, June 7-11 2004.

[5] C. Visweswariah, K. Ravindran, K. Kalafala, S. Walker, and S. Narayan. First-order incremental block-based statistical timing analysis. In *Design Automation Conference*, pages 331–336, San Diego, CA, June 7-11 2004.

[6] C. Amin, N. Menezes, K. Killpack, F. Dartu, U. Choudhury, N. Hakim, and Y. Ismail. Statistical static timing analysis: how simple can we get? In *Design Automation Conference*, pages 652–657, June 13-17 2005.

[7] K. R. Heloue and F. N. Najm. Statistical timing analysis with two-sided constraints. In *IEEE/ACM International Conference on Computer Aided Design*, pages 829–836, San Jose, CA, November 6-10 2005.

[8] A. Devgan and C. Kashyap. Block-based static timing analysis with uncertainty. In *IEEE/ACM In'l Conf. on Computer-Aided Design*, pages 607–614, San Jose, CA, November 9-13 2003.

[9] S. Bhardwaj, S. B.K. Vrudhula, and D. Blaauw. tau: Timing analysis under uncertainty. In *IEEE/ACM Int'l Conf. on Computer-Aided Design*, pages 615–620, San Jose, CA, November 9-13 2003.

[10] H. Chang and S. S. Sapatnekar. Statistical timing analysis considering spatial correlations using a single PERT-like traversal. In *IEEE/ACM Int'l Conf. on Computer-Aided Design*, pages 621–625, San Jose, CA, November 9-13 2003.

[11] A. Agarwal, D. Blaauw, and V. Zolotov. Statistical timing analysis for intra-die process variations with spatial correlations. In *IEEE/ACM Int'l Conference on Computer-Aided Design*, pages 900–907, Nov. 9-13 2003.

[12] K. A. Bowman, S. G. Duvall, and J. D. Meindl. Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration. *IEEE Journal of Solid-State Circuits*, 37(2):183–190, February 2002.

[13] F. N. Najm, N. Menezes, and I. A. Ferzli. A yield model for integrated circuits and its application to statistical timing analysis. *IEEE Transactions on Computer-Aided Design*, 26(2), 2007.

[14] S. G. Duvall. Statistical circuit modeling and optimization. In *Int'l Workshop on Statistical Metrology*, pages 56–63, June 2000.

[15] H. A. David and H. N. Nagaraja. *Order Statistics.* Wiley-Interscience, New York, NY, 3rd edition, 2003.

[16] Y. L. Tong. *The multivariate normal distribution.* Springer-Verlag, New York, NY, 1990.