

# Statistical Static Timing Analysis: How simple can we get?

Chirayu S. Amin<sup>†</sup>, Noel Menezes<sup>‡</sup>, Kip Killpack<sup>‡</sup>, Florentin Dartu<sup>‡</sup>,  
Umakanta Choudhury<sup>‡</sup>, Nagib Hakim<sup>‡</sup>, Yehea I. Ismail<sup>†</sup>

<sup>†</sup>ECE, Northwestern University  
Evanston, IL 60208, USA  
c-amin@northwestern.edu

<sup>‡</sup>Intel Corporation  
Hillsboro, OR 97124, USA  
noel.menezes@intel.com

## Abstract

With an increasing trend in the variation of the primary parameters affecting circuit performance, the need for statistical static timing analysis (SSTA) has been firmly established in the last few years. While it is generally accepted that a timing analysis tool should handle parameter variations, the benefits of advanced SSTA algorithms are still questioned by the designer community because of their significant impact on complexity of STA flows. In this paper, we present convincing evidence that a path-based SSTA approach implemented as a post-processing step captures the effect of parameter variations on circuit performance fairly accurately. On a microprocessor block implemented in 90nm technology, the error in estimating the standard deviation of the timing margin at the inputs of sequential elements is at most 0.066 FO4 delays, which translates in to only 0.31% of worst case path delay.

## Categories and Subject Descriptors

B.7.2 Integrated Circuits (Design Aids), *Simulation*

## General Terms

Algorithms, Performance

## Keywords

Statistical Static Timing Analysis (SSTA), Process Variations

## 1. Introduction

The loss of performance incurred by performing timing analysis at the worst-case process corner is increasingly harder to justify as parameter variations due to semiconductor manufacturing processes increase. Typical examples of process variations include the uncertainties in transistor channel length, dopant atom count, oxide thickness, inter layer dielectric thickness, etc, all of which give rise to delay variations.

While variations have always been around, they have not been large enough to warrant detailed handling in timing analysis flows. The effect of variations was typically handled by analyzing the circuits at different ‘corners’ or points of operation of the process. However, as pointed out by the authors of [1], [2] this corner-based analysis can be very conservative.

Effects of variations on the end result of timing analysis can be quite significant as shown by Figure . The figure shows the probability that a given path shows up as one of the top 50 worst case paths on a given die. The paths are ranked on the x-axis by margins (calculated deterministically) at the latching flops. As the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DAC 2005, June 13–17, 2005, Anaheim, California, USA.

Copyright 2005 ACM 1-59593-058-2/05/0006...\$5.00.

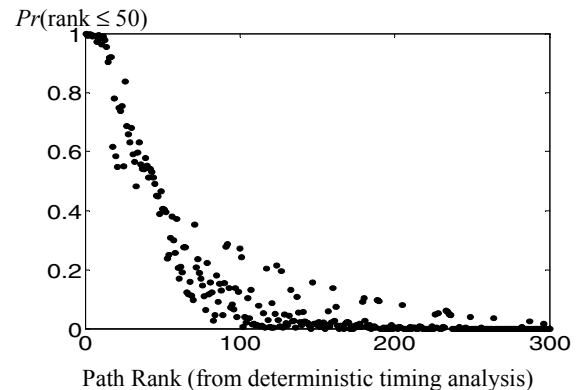


Figure 1 Probability that a path shows up in top 50 paths  
(Data from Monte Carlo simulation  
of a 90nm microprocessor block )

data shows, many paths with ranks higher than 100 will show up in the top 50 paths for the block on 10% of the dies. The result shows that deterministic timing analysis does not give an accurate path ordering.

Existing works to solve the SSTA problem include [1]–[10]. These approaches can be classified into various categories such as block-based, path-based, incremental, etc. Authors in [5]–[7] proposed techniques to bound the delay distributions instead of calculating exact distributions using path-based and block-based analysis techniques. Authors of [9] proposed an approximation approach based on a generic path analysis rather than evaluating every path statistically. Many of these researchers have advocated complicated SSTA techniques, primarily due to handling correlation and path reconvergence during the max operation fundamental to STA. In this paper, we differ from other authors and present discussion and data to show that a simple SSTA algorithm can provide very good information about variations in delays and margins at sampling flip-flops.

It should be noted that only within die variations are considered in the analysis as they primarily affect design decisions. Furthermore, our studies show that most critical paths are equally affected by die-to-die variations (Figure 7 in this paper hints at that). Hence, we do not address die-to-die variations in this paper. Moreover, since cell delay variations are larger than interconnect delay variations for our designs, we only consider variations in cell delay in this paper.

The rest of the paper is organized as follows: The process variation model is presented in section 2 along with a pre-characterization methodology to include variation information in cell models. Section 3 describes the add and max operations along with discussions about their complexity under various circumstances. The proposed statistical timing analysis method is presented in

section 4 followed by the results in section 5. The paper is concluded in section 6.

## 2. Modeling Variations

### 2.1 Parameter variations affecting delay

The underlying causes of delay variations are the variations in the transistor channel length ( $le$ ), number of dopant atoms which changes the threshold voltage ( $vt$ ), oxide thickness ( $tox$ ), inter-layer distance, etc [11]. Out of these, the main variations affecting delay are the variation in channel length and threshold voltage. The channel length variations can be further divided into a correlated (systematic) component,  $le_s$ , and a residual random component,  $le_r$ . Hence, the channel length of a transistor can be modeled as

$$le = le_{nom} + le_s + le_r \quad (1)$$

Similarly, the threshold voltage of a transistor is modeled as

$$vt = vt_{nom} + vt_r \quad (2)$$

Values of  $\sigma_{les}^2$ ,  $\sigma_{ler}^2$ , and  $\sigma_{vtr}^2$  are typically available in the process file. We model the distributions of  $le_s$ ,  $le_r$ , and  $vt_r$  using the Gaussian [12] or normal probability distribution function. Along with the variance of  $le_s$ , the correlation between the channel lengths of two transistors is specified as a function of the distance between them. That is, for transistors  $i$  and  $j$  separated by a distance  $d_{ij}$ , the correlation coefficient  $\rho_{ij}$  is:

$$\rho_{ij} = \rho(d_{ij}) \quad (3)$$

The correlation coefficient  $\rho$  ranges between 0 and 1 and is roughly inversely proportional to the distance between transistors. Synopsys' PrimeTime also uses a similar location aware On-Chip Variation (OCV) model. The higher the distance between any two transistors on a chip, the greater the probability that their channel lengths will be different. In this case,  $\rho_{ij}$  is closer to 0 than 1. For transistors which are very close together, the probability that their channel lengths are equal is higher and hence they are highly correlated with  $\rho_{ij}$  close to 1. It will be shown later that it is the correlation between delays of various cells on a path that make variations higher and statistical timing analysis important.

With the above models for delay variation and correlation, it becomes easy to pre-characterize the delay variation of a cell as a function of the underlying variations in channel length and threshold voltage. This characterization process will be described in the next subsection.

### 2.2 Pre-characterization of cells for within-die variation

The overall variance of the delay of a cell is the sum of the variances due to each dominant component of parameter variation:

$$\sigma_{delay}^2 = \sigma_{delay,les}^2 + \sigma_{delay,ler}^2 + \sigma_{delay,vtr}^2 \quad (4)$$

The delay variation of a cell is described by equation (4), which requires mapping the variances in  $le_s$ ,  $le_r$ , and  $vt_r$  to their corresponding delay variances. Hence, we can characterize the variances due to each component in terms of the variances of the underlying parameter variation component. That is:

$$\begin{aligned} \sigma_{delay,les}^2 &= f_1(\sigma_{les}^2, \dots) \\ \sigma_{delay,ler}^2 &= f_2(\sigma_{ler}^2, \dots) \\ \sigma_{delay,vtr}^2 &= f_3(\sigma_{vtr}^2, \dots) \end{aligned} \quad (5)$$

For  $le_s$  characterization, the channel lengths of transistors within a cell are highly correlated ( $\rho_{ij} = 1$ ) since they are very close to each

other. It can be safely assumed that systematic variation in  $le$  for all transistors in a cell is same and hence each cell can be characterized by a single variation parameter,  $le$ . The channel length  $le$  can be sampled on the Gaussian distribution curve with mean equal to average  $le$  and sigma<sup>1</sup> equal to  $\sigma_{les}$ . A distribution of delays can be obtained for each input to output delay arc in the cell for the corresponding distribution of  $le$ . From the delay measurements,  $\sigma_{delay,les}$  for that arc can be calculated. The characterization is performed for a range of input slopes ( $tt$ ) and output loading ( $C_L$ ). The variance of delay is then pre-characterized as:

$$\sigma_{delay,les}^2 = f_1(\sigma_{les}^2, tt, C_L) \quad (6)$$

For  $le_r$  and  $vt_r$  characterization, there is no interaction between components from different transistors. A Monte Carlo analysis can be used to vary  $le$  and  $vt$  randomly for each of the transistors in a cell according to their respective Gaussian distributions with sigmas  $\sigma_{ler}$  and  $\sigma_{vtr}$ . Each delay arc can involve multiple transistors but their net effect on delay can be easily pre-characterized as:

$$\begin{aligned} \sigma_{delay,ler}^2 &= f_2(\sigma_{ler}^2, tt, C_L) \\ \sigma_{delay,vtr}^2 &= f_3(\sigma_{vtr}^2, tt, C_L) \end{aligned} \quad (7)$$

The analysis is valid because there is a linear dependence between channel length variation and delay variation for small enough channel lengths ([16]). Moreover, it will be shown in section 3.1 that random variations on a path die out which means inaccuracies in pre-characterization of  $\sigma_{delay,ler}$  and  $\sigma_{delay,vtr}$  do not change the end result significantly.

Armed with the cell delay variation models, we can now proceed to the *add* and *max* operations required for timing analysis.

## 3. Operations for Timing Analysis

In this section, details are presented about the two most fundamental operations in static timing analysis (STA, [13]-[14]), *add* and *max*. The *min* analysis is ignored in rest of the paper because it is similar to the *max* analysis.

### 3.1 Add Operation

An *add* operation is required to calculate the arrival time at output of a cell by adding the input arrival time with cell delay. It is also used to calculate the total path delay by adding up the cell delays and wire delays on the path. For deterministic STA without any variations in arrival time and cell delays, the add operation is trivial. However for SSTA, the arrival times and cell delays are random variables and the add operation requires convolving two probability distribution functions (pdfs).

Given  $n$  independent Gaussian distributions  $P_1, P_2, \dots, P_n$  with means  $\mu_1, \mu_2, \dots, \mu_n$  respectively, and sigmas  $\sigma_1, \sigma_2, \dots, \sigma_n$ , respectively, their addition results in a distribution  $P$  [12]:

$$\begin{aligned} P &= \sum_{i=1}^n P_i(\mu_i, \sigma_i) \cong N(\mu, \sigma) \\ \mu &= \sum_{i=1}^n \mu_i \text{ and } \sigma = \sqrt{\sum_{i=1}^n \sigma_i^2} \end{aligned} \quad (8)$$

where  $N(\mu, \sigma)$  is Gaussian with mean  $\mu$  and sigma  $\sigma$ . A typical path in a microprocessor circuit has about 12-16 FO4 stages (this number is 2-3X higher for a typical ASIC) and it is projected by

<sup>1</sup> In this paper, the terms *mean* and *sigma* are used to describe average and standard deviation, respectively.

ITRS [15] that the number will hover around 12 for future technologies. This number of stages is large enough for (8) to give a reasonable estimate of the overall path delay variation where  $P_i$  is the delay distribution for each individual cell on the path.

If the distributions  $P_i$  are correlated, which is the case for a typical circuit where cell delays are correlated because of the correlation in channel length, sigma of the overall path delay distribution is given by

$$\sigma = \sqrt{\sum_{i=1}^n \sum_{j=1}^n \rho_{ij} \sigma_i \sigma_j} \quad (9)$$

where  $\rho_{ij}$  is the correlation coefficient between delays of cells  $i$  and  $j$ . Here, the covariances between cell delays also add to the variances. In the absence of correlations ( $\rho_{ij}=0$  for  $i \neq j$ ), equation (9) reduces to (8).

The correlation between cells arises from the correlated random variation in channel length of the transistors and thus  $\rho_{ij}$  is estimated by using the center-to-center distance between the cells. Here, a lumped value of  $\rho_{ij}$  is used to represent correlation between any two cells  $i$  and  $j$  because the distances between transistors belonging to different cells are roughly the same. The add operation is simple and accurate enough that no more simplification is required for statistical timing analysis. Correlation causes an insignificant runtime increase and there is no need to use a less accurate add operation.

The add operation has a significant impact on the path delay variation caused by  $le_r$  and  $vt_r$ . The random (uncorrelated) variations in  $le$  and/or  $vt$  cause some cells on a path to speed up and other cells on the same path to slow down. Moreover, the speed up or slow down of one cell is independent of the speed up or slow down of other cells on the path. Hence, given enough cells on a path, the overall variation in path delay due to  $le_r$  and  $vt_r$  is very small. In other words, the variance of the path delay is a much smaller percentage of the mean path delay as compared the variance of cell delay as a percentage of the mean cell delay. This physical phenomenon is also captured by equation (8). Assuming that all the  $n$  cells on a path are similar (same mean delays and variances),

$$\left(\frac{\sigma}{\mu}\right)_{path} = \frac{\sqrt{\sum_{i=1}^n \sigma_i^2}}{\sum_{i=1}^n \mu_i} = \frac{\sqrt{n \cdot \sigma_{cell}^2}}{n \cdot \mu_{cell}} = \frac{1}{\sqrt{n}} \cdot \left(\frac{\sigma}{\mu}\right)_{cell}, \quad (10)$$

which means that the path delay variation due to purely random variations ( $le_r$  and  $vt_r$ ) dies out as  $1/\sqrt{n}$ .

Unlike  $le_r$  and  $vt_r$ , delay variations because of  $le_s$  are correlated and do not die out on a path. If the cells on a path span a small distance, they have highly correlated channel lengths ( $\rho_{ij} \approx 1$ ). Hence, by equation (9) we have:

$$\begin{aligned} \sigma &= \sqrt{\sum_{i=1}^n \sum_{j=1}^n (1) \sigma_i \sigma_j} = n \cdot \sigma_{cell} \\ \Rightarrow \left(\frac{\sigma}{\mu}\right)_{path} &= \frac{n \cdot \sigma_{cell}}{n \cdot \mu_{cell}} = \left(\frac{\sigma}{\mu}\right)_{cell} \end{aligned} \quad (11)$$

Equation (11) shows that the correlated variation due to  $le_s$  does not die out. Physically,  $\rho_{ij}$  is not exactly 1 and equation (11) gives a rough upper bound on the sigma of path delay as a percentage of mean path delay.

The above discussion shows that the correlated systematic variations in channel length are more important than the

uncorrelated random variations in channel length and threshold voltage. It is precisely why modeling inaccuracies in  $\sigma_{delay,ler}$  and  $\sigma_{delay,vtr}$  do not cause significant errors in overall analysis.

### 3.2 Max Operation

In regular static timing analysis (STA), the *max* operation is required for multi-input cells where the worst case output arrival time is the *maximum* of all the output arrival times, each corresponding to a delay arc from input to output.

Unlike the add operation which produces a Gaussian result, the maximum of two or more Gaussian distributions may or may not be Gaussian. This fundamental problem with the max operation is the reason why many researchers [1]-[8] have devised advanced techniques to calculate max or the bounds on max accurately and efficiently. With algorithmic work in the SSTA field reaching a certain level of maturity, it is important to step back and gauge the benefits of these techniques versus a less complicated solution.

Assume that we want to obtain the maximum of two independent Gaussian distributions  $P_1$  and  $P_2$  with means  $\mu_1$  and  $\mu_2$  respectively, and sigmas  $\sigma_1$  and  $\sigma_2$ , respectively. If the distributions do not overlap significantly as shown in Figure , then the distribution with higher mean is clearly the maximum of  $P_1$  and  $P_2$  [1]. This is an accurate estimate of the max operation since even if  $P_2$  takes on the smallest value, it is still higher than the largest possibility of  $P_1$ .

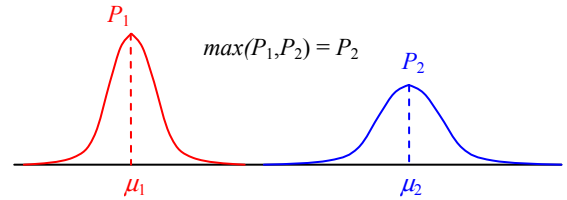


Figure 2 The *maximum* of two non-overlapping distributions is the distribution with higher mean

Mathematically,

$$P = \max(P_1, P_2) = \begin{cases} P_2, & \text{if } \mu_2 - 3\sigma_2 > \mu_1 + 3\sigma_1 \\ P_1, & \text{if } \mu_2 + 3\sigma_2 \leq \mu_1 - 3\sigma_1 \end{cases} \quad (12)$$

where the  $3\sigma$  bound around the mean value is used as a reasonable boundary of the distributions.

Now suppose  $P_1$  and  $P_2$  are highly and positively correlated ( $\rho \approx 1$ ). For this case, the statistical max operation is also trivial if the sigmas of  $P_1$  and  $P_2$  are comparable. The situation is illustrated in Figure .

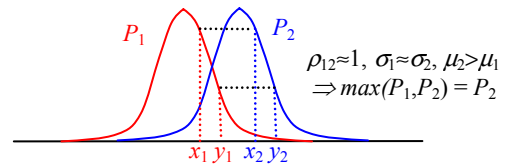
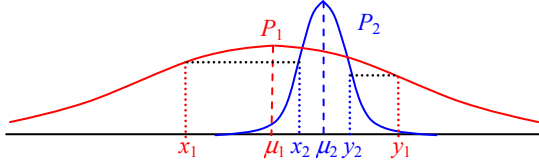


Figure 3 The *maximum* of two highly correlated distributions is the distribution with higher mean

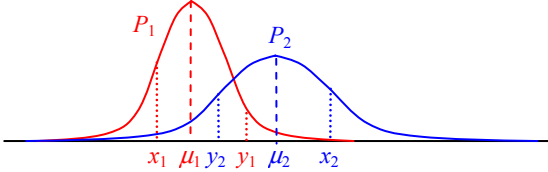
Intuitively, the values of random variables  $P_1$  and  $P_2$  move together in the sample space. If  $P_2$  has a higher mean, then all joint samples of  $P_1$  and  $P_2$  will have a value of  $P_2$  greater than the value of  $P_1$ . For example, if a sample has a value of  $P_1$  equal to  $x_1$ , then the same sample will have the value of  $P_2$  equal to  $x_2$  such that  $x_2 > x_1$ .

Complications in the max operation arise in the following two cases. The first scenario occurs if the distributions overlap, are highly correlated, and their sigmas are not comparable. The situation is shown in Figure 4. Here, some samples take on values  $(P_1, P_2) = (x_1, x_2)$  with  $x_2 > x_1$  and others take on values  $(P_1, P_2) = (y_1, y_2)$  with  $y_1 > y_2$ . High correlation still cause  $P_1$  and  $P_2$  to move together but because the sigmas of these two distributions are not comparable, the maximum of the two can be  $P_1$  in some cases and  $P_2$  in others.



**Figure 4 The max operation is not trivial if the sigmas of overlapping distributions are not comparable**

The second scenario occurs if the distributions are independent and they overlap. The situation is illustrated in Figure 5. Here, there is no relation between  $P_1$  and  $P_2$  and they are free to take on values independently. Hence, the max operation can not be computed trivially.



**Figure 5 The max operation is not trivial for two overlapping independent distributions**

The max operation becomes fairly complicated and sophisticated techniques have to be applied in order to obtain a resultant distribution. For uncorrelated random variations, the main problem is that for two independent random variables  $t_1$  and  $t_2$ , the exact max operation is given by:

$$\Pr(\max(t_1, t_2) \leq t) = \Pr(t_1 \leq t, t_2 \leq t) = C_1 \cdot C_2 \quad (13)$$

where  $C_1$  and  $C_2$  are the cumulative distribution functions (cdfs) of  $t_1$  and  $t_2$ , respectively. In this scenario, the resultant distribution can become non-Gaussian and, possibly, bi-modal (two peaks). The mean and variance values, in this case, do not necessarily characterize the complete distribution meaningfully. This would indicate that propagating the max for uncorrelated random should get special attention in SSTA.

The above discussion covered all possible *scenarios* for the max operation. *The scenarios where a non-trivial max operation is required hardly occur for high performance circuits.*

Uncorrelated delay distributions only arise because of the uncorrelated variations in  $le$  and  $vt$ . As we have shown previously in Section 3.1, uncorrelated variations in  $le$  and  $vt$  die out quickly on multi-stage critical paths typically seen on most designs. This claim is also supported by results for a microprocessor block from 90nm product (See Section 5).

With respect to systematic variations, the cell delays for cells close together are highly correlated because of high correlation in  $le$  variations for cells along typical critical paths. When a max operation is required at a multi-input cell, the arrival time distributions are highly correlated because of the close proximity of the fan-in cells to the cell at which the max operation is being

performed. In other words, the paths are short enough that  $\rho$  between any two cells on the path is very high.

In addition to these highly correlated arrival time distributions, the means and sigmas of these distributions also match up quite well because of balanced design. In this case, the output of the max operation is the distribution with larger mean.

As will be shown in the section 5, a complicated statistical max operation is not required at all. The situations illustrated in Figure 4 and Figure 5 do not occur. The distributions of arrival times at multi-input cells are highly correlated with comparable sigmas and the random variations die out quickly on a path. These physical effects do not require a complex max operation and the results in Section 5 agree with the above discussion.

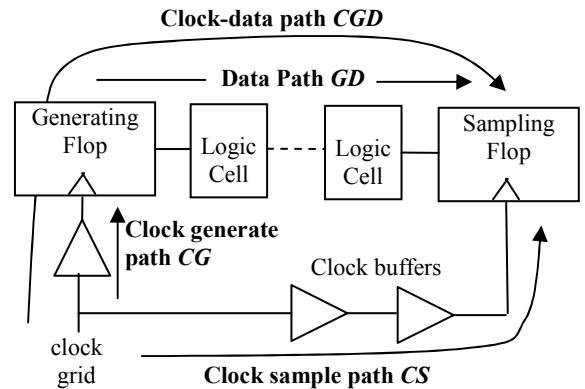
The next section describes the algorithm we use for SSTA.

## 4. Path-based Statistical Static Timing Analysis

Static timing analysis tools typically implement a path-based or block-based traversal of the cells from inputs to outputs. These methods are classified as depth-first-search and breadth-first-search, respectively. Industrial tools use a combination of these methods to perform static timing analysis. While block-based methods are fast and reasonably accurate, designers are more comfortable with path-based methods because they identify the specific paths in a design that need to be fixed. However, since path-based techniques cannot efficiently identify all the paths likely to be critical, block-based methods are often applied as a preprocessing step to extract information that is later used to identify critical paths.

We apply a path-based algorithm in our SSTA flow. The algorithm is a post-processor which performs statistical analysis for the top  $N$  paths after regular deterministic timing analysis, with  $N$  chosen to cover a sufficient spread of paths. We chose a path-based approach because it is easier to integrate in a regular flow as a post-processor. We later experimentally verified that our path-based SSTA technique provided results close to the golden Monte Carlo simulation reference.

Figure 6 shows a data path and the associated clock paths. Typical timing analysis performs the setup and hold checks at the sampling flop.



**Figure 6 Typical path-based analysis**

The typical setup check is given by:

$$setup: t_{GD, \max} + t_{setup} + skew_{\max} \leq T \quad (14)$$

where  $t_{GD,max}$  is the maximum possible delay of the path  $GD$ ,  $t_{setup}$  the setup time of the receiving flop,  $T$  the desired cycle time, and  $skew_{max}$  is the estimated variation in skew for the slow process corner.

Similarly, the hold check is given by:

$$hold : t_{GD,min} \geq t_{hold} + skew_{min} \quad (15)$$

where  $t_{GD,min}$  is the minimum possible delay of the path  $GD$ ,  $t_{hold}$  is the hold time of the receiving flop, and  $skew_{min}$  is the skew for the fast process corner.

The skew values,  $skew_{max}$  and  $skew_{min}$ , in the equations above are usually computed at slow and fast corners of the process, respectively. Most timing analysis flows account for process variations in calculating these skews by applying a process variation penalty in addition to the nominal clock skew. This penalty can be derived from approximate first-order formulas based on the clock path delays or from skew tables which store pre-computed values of these skews. No production timing analysis methodology performs an explicit computation of this process variation induced skew on-the-fly during timing analysis.

Instead of treating the skew as an abstraction that is used for setup and hold time checks in equations (14) and (15), we actually compute the margin for the check according to statistical theory. The basic failure mechanism for a setup check is that the time it takes for a signal to reach the receiving flop via path  $CGD$  should be greater than that of the sampling  $CS$  augmented by a cycle delay. This difference called the max margin is a quantity that should be analyzed statistically. The reason for statistical analysis of margin is that differences are treated differently for statistical quantities than for deterministic quantities. The deterministic margin at the receiving flop is given by:

$$margin = t_{CS} + T - t_{CGD} \quad (16)$$

Equation (16) is a valid equation for computing the mean of the margin. The variance of the margin according to statistical theory is given by:

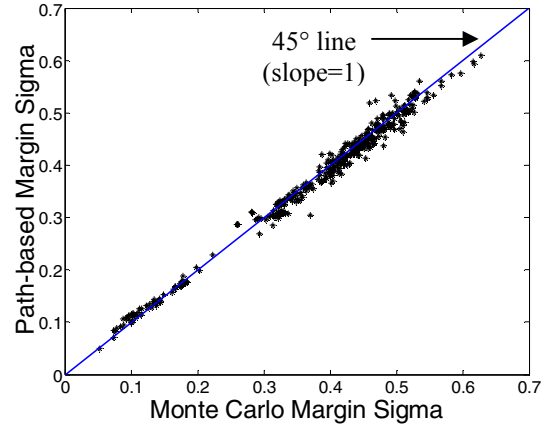
$$\sigma_{margin}^2 = \sigma_{delay,CS}^2 + \sigma_{delay,CGD}^2 - 2 \text{cov}(t_{CS}, t_{CGD}) \quad (17)$$

where  $\text{cov}(t_{CS}, t_{CGD})$  represents the covariance due to process variations in the respective path delays. It can be seen from (17) that the variation of data path delay adds to the overall variation which is in contrast to the subtraction of mean delay of path  $CGD$  in equation (16). Moreover, a component of the statistical variation represented by the common variations of both paths -- the covariance term in (17) -- can be recovered from the margin. This recovery of margin because of the correlation in the systematic component of clock and data path delay variation allows for a less pessimistic (and more accurate) estimate of setup and hold margins thereby expanding the design window.

Above analysis requires the calculation of the variances of the clock sample path and generate-data path delays. The calculation of these variances can be handled easily with the pre-characterized models presented in section 2 and the add operation presented in Section 3.1.

The covariance between paths  $CS$  and  $CGD$  is given by the covariance of the correlated components of delay variation ( $le_s$ ) for each pair of cells:

$$\begin{aligned} \text{cov}(t_{CS}, t_{CGD}) &= \sum_{i \in CS} \sum_{j \in CGD} \text{cov}(cell_i, cell_j) \\ &= \sum_{i \in CS} \sum_{j \in CGD} \rho_{ij} \sigma_{delay,les,i} \sigma_{delay,les,j} \end{aligned} \quad (18)$$



**Figure 7 Margin sigma comparison between path-based SSTA and Monte Carlo (statistical) analysis for top 492 end nodes given by deterministic analysis (Margins normalized by FO4 delay)**

For purely random variations due to  $le_r$  and  $vt_r$ , Equation (17) still applies. However, the covariance term is zero.

## 5. Results

We applied the path-based SSTA algorithm presented in the previous section to a large microprocessor block (> 100K cells) implemented in a 90nm process. The entire library was characterized using the methods outlined in Section 2.

First, block-based deterministic analysis was applied to identify the top  $N$  critical end-nodes (flop inputs, primary outputs, etc). The top worst case path ending up at each of the  $N$  nodes was identified by back-tracking and path-based analysis was performed on it. Then path-based SSTA was applied as a post-processing algorithm for these top  $N$  worst case paths. Moreover, a block-based Monte Carlo analysis was also performed to create a set of golden results for evaluating the accuracy of path-based SSTA. For the Monte Carlo simulation, 600 sample dies were created each representing a profile of the system channel length variation according to the spatial correlation formula for the process specified in Equation (3). During the course of our work, we empirically determined that 600 die samples were enough to ensure that each cell in the block went through a  $\pm 3\sigma$  variation. The number of required samples is a function of the block size. The 600 samples produced results very similar to as many 8000 samples, including accurate mean, standard deviation, and the overall distribution shape.

Figure 7 shows the comparison of path-based SSTA results with Monte Carlo analysis for the 492 most-critical end nodes/paths. We see that path-based SSTA produces results almost identical to Monte Carlo analysis.

The maximum observed difference in predicting the standard deviation of the timing margin variation between path-based SSTA and Monte Carlo simulation is 0.066 FO4 delays, occurring for a 21 stage path. The error in standard deviation of the margin translates into 0.31% of the worst case path delay, making the error insignificant. On average, the error in computing the standard deviation of the margin was 0.19% of the path delay.

Based on these results, we further investigated the cause of the efficacy of the path-based SSTA technique: During the Monte Carlo analysis, at each critical end-node where the margin was

calculated, we also stored the contributing worst-case path. Analysis of these top paths for each of the  $N$  nodes showed that on more than 80% of the 600 samples, only one or two different paths were showing up as the worst case paths. Moreover, the path given by worst case deterministic analysis showed up as the worst case path on most of the samples. This analysis revealed why path-based SSTA was able to achieve results close to Monte Carlo simulation without applying the max operation. Most of the time, there are one or two clear worst case paths that end up at a node. Ordering of paths ending up at a node does not change dramatically with statistical analysis. This is in line with expectations because paths ending up at a node are highly correlated and if a path has higher delay on one of the samples, then chances are that all the neighboring path delays will also scale similarly. These results also demonstrate that random uncorrelated variations in  $l_e$  and  $vt$  (relative to their mean values) die out and are not large enough to cause reordering of critical paths converging on an end-node.

What changes in statistical timing analysis is the relative importance of nodes with respect to each other. Our Monte Carlo analysis showed that paths that are physically far away from each other in the mega-block showed a tendency to switch their relative ordering than paths that are close to each other. This is also in line with intuition. As the distance between cells increases, the correlation between cell delays decreases and the delay distributions move independently of each other to cause a change in relative ordering. This result is highlighted in Figure in the introduction.

Statistical path ordering is a different topic and is not touched in this paper. However, path-based SSTA does provide enough information for a path ordering method. Path-based SSTA provides variances for each path as well as the covariance between any two paths. This information can be used to order paths statistically.

## 6. Conclusions

In this work, we have demonstrated that statistical timing analysis is important for current and future technologies. Furthermore, we have shown that a path-based SSTA algorithm implemented as a post-processor is able to capture the effects of process variations on timing which obviates the need for advanced SSTA algorithms. Our path-based SSTA technique is justified based on the characteristics of the underlying design – several stages along a path – and the spatial variation profiles of the underlying process technology. (We have reason to believe that this variation trend will be sustained for future process technologies [17].) In addition, a method for characterizing a library for delay variation has also been presented. The error in estimating the standard deviation of timing margin was at most 0.066 FO4 delays, which translates in to only 0.31% of worst case path delay. On average, the error in computing the standard deviation of the margin was 0.19% of the path delay.

## 7. References

- [1] H. Chang and S. S. Sapatnekar, "Statistical timing analysis considering spatial correlations using a single Pert-like traversal," *ICCAD '03*, pp. 621-625.

- [2] C. Visweswariah, K. Ravindran, K. Kalafala, S. G. Walker, and S. Narayan, "First-order incremental block-based statistical timing analysis," *DAC '04*, pp. 331-336.
- [3] J. A. G. Jess et. al., "Statistical timing for parametric yield prediction of digital integrated circuits," *DAC '03*, pp. 932-937.
- [4] A. Devgan and C. Kashyap, "Block-based static timing analysis with uncertainty," *ICCAD '03*, pp. 607-614.
- [5] M. Orshansky and A. Bandyopadhyay, "Fast statistical timing analysis handling arbitrary delay correlations," *DAC '04*, pp. 337-342.
- [6] M. Orshansky and K. Keutzer, "A general probabilistic framework for worst case timing analysis," *DAC '02*, pp. 556-561.
- [7] A. Agarwal, V. Zolotov, and D. Blaauw, "Statistical timing analysis using bounds and selective enumeration," *IEEE Trans. on CAD*, vol. 22, no. 9, pp. 1243-1260, September 2003.
- [8] A. Agarwal, D. Blaauw, and V. Zolotov, "Statistical timing analysis for intra-die process variations with spatial correlations," *ICCAD '03*, pp. 900-907.
- [9] F. N. Najm and N. Menezes, "Statistical timing analysis based on a timing yield model," *DAC '04*, pp. 460-465.
- [10] A. Gattiker, S. Nassif, R. Dinakar, and C. Long, "Timing yield estimation from static timing analysis," *ISQED '01*, pp. 437-442.
- [11] S. Nassif, "Delay variability: sources, impact and trends," *ISSCC '00*, pp. 368-369.
- [12] A. Papoulis and S. U. Pillai, *Probability, Random Variables, and Stochastic Processes*, fourth edition, McGraw Hill, 2002.
- [13] J. K. Ousterhout, "A switch-level timing verifier for Digital MOS VLSI," *IEEE Trans. on CAD*, vol. 4, no. 3, pp. 336-349, July 1985.
- [14] N. P. Jouppi, "Timing analysis and performance improvement of MOS VLSI Designs," *IEEE Trans. on CAD*, vol. CAD-6, no. 4, pp. 650-665, July 1987.
- [15] *International Technology Roadmap for Semiconductors (ITRS)*, 2003 edition, Semiconductor Industry Association (<http://public.itrs.net/>).
- [16] T. Sakurai and A. Newton, "Alpha-power law MOSFET model and its application to CMOS inverter delay and other formulas," *IEEE JSSC*, vol. 25, no. 2, pp. 584-594, April 1990.
- [17] S. Samaan, "The Impact of Device Parameter Variations on the Frequency and Performance of VLSI Chips," *ICCAD '04*. (Also published in *ISSCC '04*.)