Statistical Static Timing Analysis with Conditional Linear MAX/MIN Approximation and Extended Canonical Timing Model

Lizheng Zhang, *Student Member, IEEE,* Weijen Chen, *Student Member, IEEE,*
Yuhen Hu, *Fellow, IEEE,* Charlie, Chung-ping Chen, *Member, IEEE,*

Abstract

An efficient and accurate statistical static timing analysis (SSTA) algorithm is reported in this work which features (a) a conditional linear approximation method of the MAX/MIN timing operator, (b) an extended canonical representation of correlated timing variables, and (c) a variation pruning method that facilitates intelligent trade-off between simulation time and accuracy of simulation result. A special design focus of the proposed algorithm is on the propagation of the statistical correlation among timing variables through the nonlinear circuit elements. The proposed algorithm distinguishes itself from existing block based SSTA algorithms in that it not only deals with correlations due to dependence on global variation factors, but also correlations due to signal propagation path reconvergence. Tested with ISCAS benchmark suites, the proposed algorithm has demonstrated very satisfactory performance in terms of both accuracy and running time. Compared with Monte Carlo based statistical timing simulation, the output probability distribution got from the proposed algorithm is within 1.5% estimation error while a 350 times speed-up is achieved over a circuit with 5355 gates.

I. INTRODUCTION

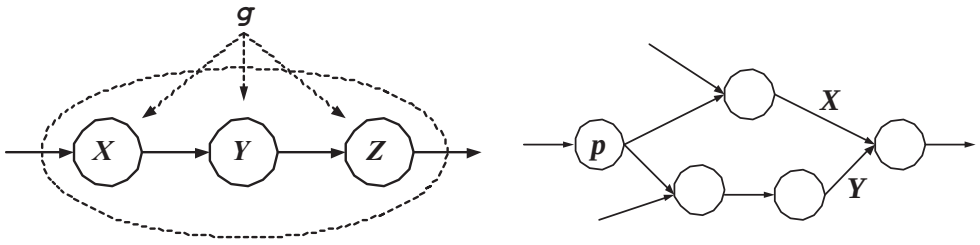
The timing performance of deep-submicron micro-architecture will be dominated by several factors. IC manufacturing process parameter variations will cause device and circuit parameters to deviate from their designed value. Low supply voltage for low-power applications will reduce noise margin, causing increased timing delay variations. Due to dense integration and non-ideal on-chip power dissipation, rising temperature of substrate may lead to hot spot, causing excessive timing variations. Classical worst case timing analysis produces timing predictions that are often too pessimistic and grossly conservative. On the other hand, statistical static timing analysis (SSTA) that characterizes timing delays as statistical random variables offers a better approach for more accurate and realistic timing prediction.

Existing SSTA methods can be categorized into two distinct approaches: **path based SSTA** [1]–[4] and **block based SSTA** [5]–[10]. The path based SSTA seeks to estimate timing statistically on selected *critical paths*. However, the task of selecting a subset of paths whose time constraints are statistically critical has a worst-case computation

complexity that grows exponentially with respect to the circuit size. Hence the path based SSTA is not easily scalable to handle realistic circuits.

The block based SSTA, on the other hand, champions the notion of *progressive computation*. Specifically, by treating every gate/wire as a timing block, the SSTA is performed block by block in the forward direction in the circuit timing graph without looking back to the path history. As such, the computation complexity of block based SSTA would grow linearly with respect to the circuit size. However, to realize the full benefit of block based SSTA, one must address a challenging issue that timing variables in a circuit could be correlated due to either *global variations*([6], [7], [10]) or *path reconvergence* ([5], [9]). As illustrated in the left hand side of Figure 1, *global correlation* refers to the statistical correlation among timing variables in the circuit due to *global variations* such as inter- or intra-die spatial correlations, same gate type correlations, temperature or supply voltage fluctuations, etc. *Path correlation*, on the other hand, is caused by the phenomenon of *path reconvergence*, that is, timing variables in the circuit can share a common subset of gate/wire blocks along their path histories. (Figure 1)

The importance of the path correlation comes from the fact that each gate/wire block in the circuit will have some *local variations* which are independent to the rest of the circuit. These local variations will propagate towards the circuit output and cause additional correlations due to the phenomenon of *path reconvergence*. Furthermore, these correlations caused by sharing local variations, cannot be correctly captured by any algorithm that deals with global variations only. So for clarity, the term *path correlation* used here and after specifically refers to the correlation caused by the local variations of the common path history.



(a) X, Y and Z depend on g (b) X and Y depend on p

Fig. 1: Global Correlations (left) and Path Correlation(right)

Several solutions have been proposed to deal with either of these two types of correlations. In [6], [7], [10], the dependence on global variations is explicitly represented using a *canonical timing model*. However, these approaches did not take into account the path correlations. In [9], a method based on common block detection is introduced to deal with the path correlations. However, this method does not address the issue of dependence on global variations. To the best of our knowledge, there is no existing method that has dealt with both types of correlations simultaneously. We present a novel block based SSTA algorithm in this paper that is designed to consider *both* global correlations and path correlations:

- We develop a novel method to conditionally approximate the MAX/MIN operator by a linear mixing operator.

Using the pre-computed skewness, we are able to determine the linearity of the MAX/MIN operator analytically. The linear approximation is then applied only when MAX/MIN behaves linear. When MAX/MIN is significantly non-linear, the MAX/MIN evaluation is postponed with a form of *Max Tuple*.

- We *extend* the commonly used canonical timing model to be able to represent all possible correlations, including the path correlations, between timing variables in the circuit. We further explore the sparse structure of the extended canonical representations of the timing variable and dynamically drop the non-significant terms so as to curtail the amount of storage and computation required for implementations.

Since $min(X,Y) = -max(-X,-Y)$, in the interests of brevity, in the rest of this paper, we will only discuss the MAX operator, with the understanding that the same results can be easily adapted to the MIN operator.

The rest of the paper is organized as following: In section II, previous block based SSTA methods are reviewed briefly; Section III discusses the non-linearity of the MAX operator and our conditional linear approximation method; Section IV describes the extended canonical timing model and the proposed SSTA algorithm with the technique to reduce computation complexity. Section V presents a real implementation of of our algorithm in C/C++ and the testing results with benchmark circuits; Section VI gives the conclusions.

II. A Brief Review of Current SSTA Algorithms

For the purpose of timing analysis, the circuit is modeled as a directed acyclic graph(DAG), called a *timing graph*, where timing blocks are used to represent the gate/wires in the circuit. Signals propagating through these blocks will add block delays into their *arrival times*. Block delays and arrival times are both called *timing variables* of the circuit. The *history* or *path history* of an arrival time is then defined as the set of block delays through which the signal ever passes.

A. Timing Variable Propagation

In statistical timing analysis, a timing variable is modeled as a *random variable* that is characterized by its distribution of *probability density function*($p.d.f.$) or equivalently, *cumulative distribution function*($c.d.f.$). The goal of statistical timing analysis is to estimate the distribution of the arrival time in the circuits given the distributions of each block delay in the circuit. To accrue the over all timing delay distribution, the timing delay random variables will be joined through two basic operators [5]:

- ADD: When an input arrival time X propagates through a block delay Y, the output arrival time will be $Z = X + Y$
- MAX: When two arrival times X and Y merge in a block, a new arrival time $Z = max(X,Y)$ will be formulated before the block delay is added.

In the ADD operation, if both X and Y are Gaussian random variables, then $Z = X+Y$ will also be a Gaussian random variable whose mean and variance can be found as:

$$\mu_Z = \mu_X + \mu_Y \quad (1)$$

$$\sigma_Z^2 = \sigma_X^2 + \sigma_Y^2 + 2cov(X,Y) \quad (2)$$

where $\sigma_{XY} = cov(X,Y) = E\{(X-\mu_X)(Y-\mu_Y)\}$ is the covariance between X and Y.

Denote $Z = max(X,Y)$ to be the output of the MAX operator. Since MAX is generally a nonlinear operator, Z will not have Gaussian distribution even if both X and Y are Gaussian. However, in this situation, the mean, variance and skewness of the distribution of Z have been already derived analytically by Clark [11] in 1961 as follows:

$$\mu_Z = \mu_X \cdot Q + \mu_Y(1-Q) + \theta P \tag{3}$$

$$\sigma_Z^2 = (\mu_X^2+\sigma_X^2)Q + (\mu_Y^2+\sigma_Y^2)(1-Q) + (\mu_X+\mu_Y)\theta P - \mu_Z^2 \tag{4}$$

$$\begin{aligned}\kappa_Z^3 &= \frac{1}{\sigma_Z^3}\left\{(\mu_X^3+3\mu_X\sigma_X^2)Q + (\mu_Y^3+3\mu_Y\sigma_Y^2)(1-Q) - \mu_Z(3\sigma_Z^2+\mu_Z^2)\right. \\ &+ \left.\frac{P}{\theta}\left((\mu_X^2+\mu_X\mu_Y+\mu_Y^2)\theta^2 + 2\sigma_X^4 + \sigma_X^2\sigma_Y^2 + 2\sigma_Y^4 - 2\sigma_{XY}(\sigma_X^2+\sigma_Y^2) - \sigma_{XY}^2\right)\right\}\end{aligned} \tag{5}$$

where $\theta = \sigma_{(X-Y)}$. P and Q are the $p.d.f.$ and $c.d.f.$ of a standard normal distribution evaluated at $\lambda = \mu_{(X-Y)}/\sigma_{(X-Y)}$:

$$P(\lambda) = \frac{1}{\sqrt{2\pi}}exp(-\frac{\lambda^2}{2}) \quad and \quad Q(\lambda) = \int_{-\infty}^{\lambda} P(x)dx \tag{6}$$

The skewness of random varaible Z is defined as:

$$\kappa_Z = \frac{\sqrt[3]{E\{(Z-\mu_Z)^3\}}}{\sigma_Z} \tag{7}$$

B. Linear Approximation of MAX Operator

Although $Z = max(X,Y)$ does not have a a Gaussian $p.d.f.$ even both inputs X and Y are Gaussian-distributed, in the interests of simplicity, it is still desirable by many SSTA researchers to find a Gaussian random variable that approximates Z in some way ([6], [10]). In [6], the output of the MAX operator, Z is approximated by a Gaussian random variable $\hat{Z}$ which is a linear combination of X, Y, and an additional independent Gaussian random variable Δ:

$$Z = MAX(X,Y) \approx QX + (1-Q)Y + \Delta = \hat{Z} \tag{8}$$

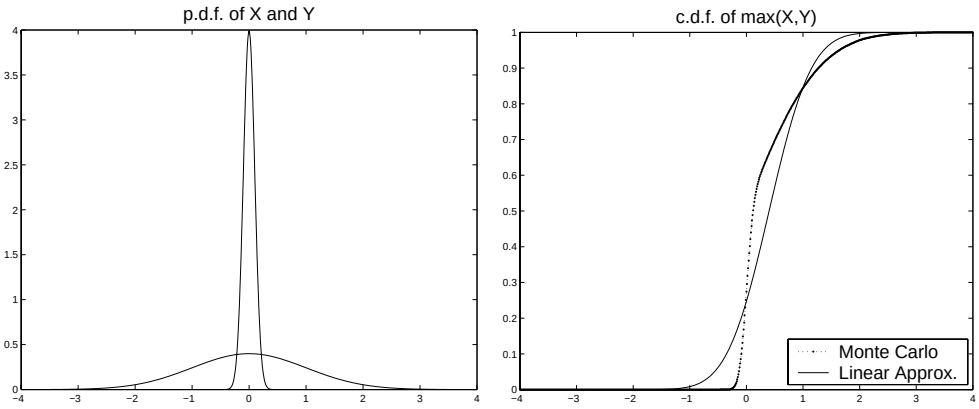
where Q is defined in Equation (6), and is called *tightness* by the authors in [6]. The purpose of the additional random variable Δ is to ensure that the mean and the variance of $\hat{Z}$ match those of Z as specified in the Clark's formula (3) and (4).

In [11], it has also been shown that if W is a Gaussian random variable, then the cross-covariance between W and $Z = MAX(X,Y)$ can be found analytically as:

$$cov(W,Z) = Q \cdot cov(W,X) + (1-Q)cov(W,Y) \tag{9}$$

Substituting equation (8), it is easy to verify that

$$cov(W,\hat{Z}) = Q \cdot cov(W,X) + (1-Q)cov(W,Y) = cov(W,Z)$$



(a) $p.d.f.$ of Independent X and Y (b) $c.d.f.$ of $max(X, Y)$

Fig. 2: Existing Linear Approximation underestimates MAX Distribution at High Probability level

Hence, a nice property of the approximator $\hat{Z}$ shown in equation (8) is that the cross-covariance between Z and other timing variable W is preserved when Z is replaced by $\hat{Z}$.

While this approximation formula is simple, it doesn't work safely when the non-linearity of the MAX operation is significant and the output of MAX operator is significantly non-Gaussian. A simple example is illustrated in figure 2 where the left panel shows the two independent input Gaussian random variables and the right panel shows the $c.d.f.$s of $max(X, Y)$ from Monte Carlo simulation and linear approximation. It can be seen from figure 2(b) that the existing linear approximation will underestimate the distribution at high probability level. This behavior is risky since decisions made upon the estimated delay may result in excessive design failure

C. Canonical Timing Model

Previously, a *canonical timing model* [6], [7], [10] has been proposed to address the delay correlations through shared global variations. In this model, the block delay is represented as a sum of three terms:

$$n_i = \mu_i + \alpha_i R_i + \sum_{j=1} \beta_{i,j} G_j \tag{10}$$

where $n_i (i = 1, 2, ...)$ is the random variable corresponding to the the i^{th} block delay in the timing graph; μ_i is the expected value of n_i; $R_i \sim N(0, 1)$, (called *local variation*), represents the localized statistical uncertainties of n_i; $G_j \sim N(0, 1)$ represents the j^{th} *global variation*; R_i and $\{G_j (j = 1, 2, ...)\}$ are additionally assumed to be mutually independent; the weight parameter α_i (named *local sensitivity*) and $\beta_{i,j}$(named *global sensitivities*) are deterministic constants, *explicitly* expressing the amount of dependence of n_i on each of the corresponding independent random variables.

With this canonical representation, the variance of a block delay n_i and its covariance with another block delay

n_k can be evaluated as:

$$\sigma_{n_i}^2 = E\{(n_i - \mu_i)^2\} = \alpha_i^2 + \sum_j \beta_{i,j}^2 \quad (11)$$

$$cov(n_i, n_k) = E\{(n_i - \mu_i)(n_k - \mu_k)\} = \sum_j \beta_{i,j}\beta_{k,j} \quad (12)$$

However, if arrival times are also expressed in this canonical model, the path correlation between them due to sharing local variations because of path reconvergence will incorrectly be ignored. For example, in Figure 1(b), both arrival times X and Y include a common path history of block p. However, the local variation of block p, R_p is no longer a part of the canonical representation of arrival times X and Y. Hence, the path correlation between X and Y due to R_p is incorrectly dropped.

III. NON-LINEARITY OF MAX OPERATOR

For Gaussian inputs, the linearity of the MAX operator will be equivalent to the Gaussianity of the output. Using Monte Carlo simulation, the Gaussianity of the output can be evaluated with a method called *QQ-Plot*. [12] Specifically, if the output is Gaussian, then the simulated output of the MAX operator will show a straight line in its QQ-Plot against a standard Gaussian distribution. And if the MAX output is non-Gaussian, such QQ-Plot will deviated from linear. The more the non-Gaussianity of the MAX output, the worse the linearity of such QQ-Plot.

Since the linearity of the QQ-Plot can be quantitatively represented by the linear correlation coefficient of the QQ-Plot, the Gaussianity of the output of the MAX operator can be statistically and quatitatively measured. However, it will be very expensive if we run extensive Monte Carlo simulation during every step of MAX operation in timing analysis. So it is desirable to establish a more convenient criteria to determine the linearity of the MAX operator.

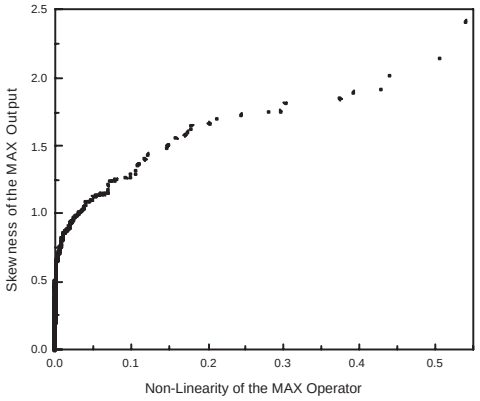


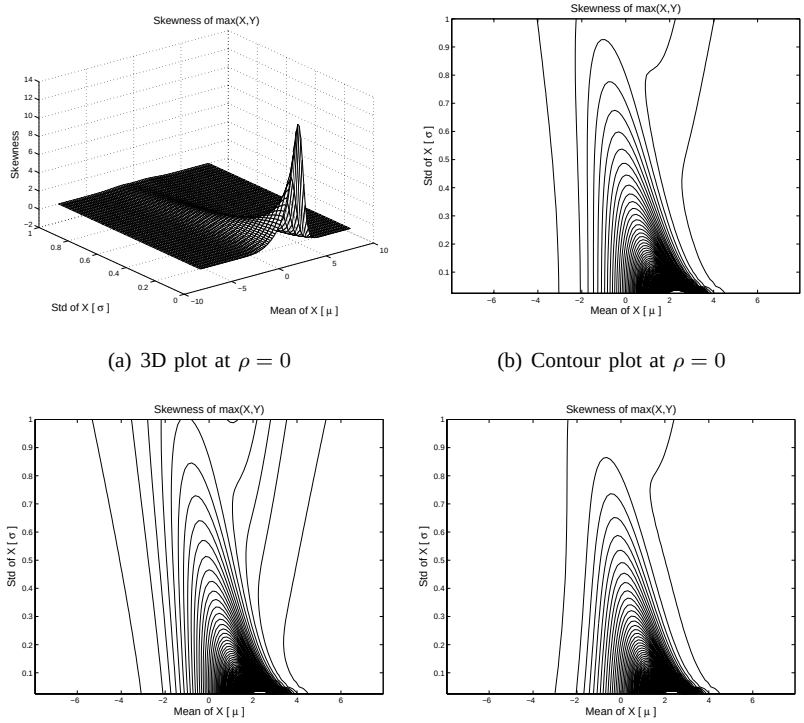
Fig. 3: Skewness of $Z = max(X, Y)$ given X and Y are Gaussian v.s. Non-Linearity of MAX Operator Determined by Monte Carlo simulation

It is well known that skewness is not a Gaussianity index for a general random variable since there are distributions which are symmetric but non-Gaussian. However, to measure the linearity of the MAX operator with Gaussian inputs,

skewness of the MAX output will be a good choice. Figure 3 shows the relationship between the non-linearity of the MAX operator and the skewness of $Z = max(X, Y)$ for Gaussian inputs X and Y. The scattering points in the figure represent 1000 random samples of the relative mean, relative variance and the correlation of Gaussian random variables X and Y. The non-linearity of the MAX operator for each set of randomly sampled mean, variance and correlation is determined by QQ-Plot method with 10,000 Monte Carlo simulations. It is very clear in the figure that the skewness of the MAX output has significant positive correlation with the non-linearity of the MAX operator. Since skewness of the MAX output given Gaussian inputs can be analytically computed by equations developed by Clark [11], it is suitable to use skewness as an accurate and efficient measurement for the non-linearity of the MAX operator.

A. Non-Linearity Condition of MAX Operator

It is clear that the linearity of the MAX operator is heavily dependent on its input parameters. Since we have a good measure of the linearity of the MAX operator, it is ready to study how the linearity changes when inputs vary.



(a) 3D plot at $\rho = 0$

(b) Contour plot at $\rho = 0$

(c) Contour plot at $\rho = -0.5$

(d) Contour plot at $\rho = 0.5$

Fig. 4: Skewness of $max(X, Y)$ when $Y \sim N(0, 1)$

Assuming the standard deviations $\sigma_Y \geq \sigma_X$ in $max(X, Y)$, then no generality will be lost if the two variances are assumed to be: $\sigma_X = \sigma \in [0, 1]$ and $\sigma_Y = 1$. This simplification is valid because of the scaling property of

the MAX operator: $max(cX, cY) = c \cdot max(X, Y)$ for any positive constant c. Aware the invariance of the MAX operator in the constant shifting as $max(X, Y) + c = max(X + c, Y + c)$, both random variables of X and Y are shifted by the mean of Y and so that the mean parameters will satisfy the range of $\mu_X = \mu$ and $\mu_Y = 0$. The last parameter that needed to specify the two input random variables involved in a MAX operation is their correlation coefficient ρ which must be in the range of $-1 \leq \rho \leq 1$. With such parameter settings, the two Gaussian random variables X and Y are fully determined. And the skewness of $Z = max(X, Y)$ are computed using equations developed by Clark [11] and are shown in Figures 4.

From the figures, it is clear that in most of the cases, the skewness is zero which means $Z = max(X, Y)$ is normally distributed and MAX operator is linear. As a thumb rule, the non-linearity of MAX operator is significant when the following *Non-Linear Condition* satisfies:

> *Given X and Y are Gaussian, max(X,Y) will be significantly non-Gaussian if X and Y have very similar mean but very different variance or if X and Y have similar mean and variance but very negative correlation*

B. Conditional Linear MAX Approximation

Given two Gaussian random variables X and Y, $Z = max(X, Y)$ could be significantly skewed if the non-linear condition is satisfied. If the MAX operator is significantly non-linear, significant error will occur if a linear operator is forced to approximate the MAX operator. But for the purpose of timing analysis, it is not necessary to explicitly compute the MAX output at every step.

1) Max Tuple: During timing analysis, arrival time propagates from block to block with two elemental operations: ADD and MAX. If during a propagation step of MAX, $max(X, Y)$, the output arrival time is not Gaussian, no actual computation will be done and the output will be simply recorded as a *max tuple*: $Mt\{X, Y\}$. With such max tuples, the arrival time propagation will have the following computations:

- ADD: a gate/wire delay, D, is added into a max tuple $Mt\{X, Y\}$ as:

$$Mt\{X, Y\} + D = Mt\{X + D, Y + D\}$$

- aMAX: an arrival time, A, is MAXed with a max tuple $Mt\{X, Y\}$ as:

$$max(A, Mt\{X, Y\}) = Mt\{A, X, Y\}$$

- tMAX: two max tuples are MAXed together:

$$max(Mt\{X, Y\}, Mt\{U, V\}) = Mt\{X, Y, U, V\}$$

2) Tuple Size: To practically implement such tuple-based MAX evaluation, the number of arrival times in the max tuple, i.e. the tuple size, has to be maintained as small as possible. This is realized by the obvious combinational rule of max tuple as:

$$Mt\{A, X, Y\} = Mt\{max(A, X), Y\} = Mt\{A, max(X, Y)\} = Mt\{X, max(A, Y)\}$$

so if any two Gaussian random variables in the max tuple doesn't satisfy the non-linear condition, then they can be replaced by a new Gaussian random variable by approximating the MAX with a linear operator and so that the size of the max tuple is reduced. This reduction process will be done iteratively to minimize the tuple size.

Such kind of tuple size reduction method is realized by associating each max tuple with a skewness matrix which stores the output skewness if pairs of random variables in the max tuple are actually MAXed out. And also a threshold of skewness κ_{th} is set before-hand to decide if the MAX result is Gaussian or non-Gaussian. Also, to prevent the explosion of the tuple size, a safe-guard maximum allowed size for max tuple is also set and if any of the tuple size exceeds the maximum size, the skewness threshold will be increased to tolerate more tuple size reduction.

Finally, in the primary output of the circuit, if the circuit delay is reported as max tuple, the output distribution can be easily evaluated by Monte Carlo simulation. For limited size of max tuple, such evaluation is efficient and accurate.

IV. Extended Canonical Timing Model

The canonical timing model [6], [7], [10] is a powerful tool to represent the numerous timing variables for a given circuit. However, as pointed out in the previous section, in its original format, it can only handle timing correlations caused by global variations. In this work, we propose an *extended canonical timing (ECT)* model that is capable of capture *all* correlations between any pair of timing variables in the circuit be it a block delay or an arrival time.

A. Extended Canonical Timing Model

Assume that there are N gate/wire blocks and M global variations in the timing graph, if every block delay is modeled by the canonical format shown in equation (10), and MAX is approximated by a linear combination operator, then every time variable, including all block delays and arrival times will then have the *extended canonical timing(ECT)* expression as:

$$X \quad = \quad \mu_X + \sum_{i=1}^{N} \alpha_{X,i} R_i + \sum_{j=1}^{M} \beta_{X,j} G_j \tag{13}$$

where $R_i \sim N(0,1)$ is the local and independent variation only related with block i, $G_j \sim N(0,1)$ is the j^{th} global variation, $\alpha_{X,i}$ and $\beta_{X,j}$ are the corresponding sensitivity factors. To differ our approach from the existing canonical timing model, the word "*extended*" is used to indicate that the local variations are additionally included to the timing model. With such "extended" timing model, both global and path correlations can be handled elegantly. More specifically, global variations are represented by the set of global sensitivity terms $\{\beta_{X,j}\}$, and dependence on path history are represented by non-zero local sensitivity terms $\alpha_{X,k}$.

Equation (13) can be rewritten in a compacted vector format as

$$X \sim L(\mu_X, \boldsymbol{\alpha}_X, \boldsymbol{\beta}_X) = \mu_X + \boldsymbol{\alpha}_X^* \boldsymbol{r} + \boldsymbol{\beta}_X^* \boldsymbol{g} \tag{14}$$

where "*" means transpose and $\boldsymbol{r} \equiv [R_1, \cdots, R_N]^* \sim N(\mathbf{0}, \boldsymbol{I})$ and $\boldsymbol{g} \equiv [G_1, \cdots, G_M]^* \sim N(\mathbf{0}, \boldsymbol{I})$ are mutually independent *local variation vector* and *global variation vector* respectively. $\mathbf{0}$ is a zero vector and $\boldsymbol{I}$ is the unit matrix. $\boldsymbol{\alpha}_X = [\alpha_{X,1}, \alpha_{X,2}, ..., \alpha_{X,N}]^*$ and $\boldsymbol{\beta}_X = [\beta_{X,1}, \beta_{X,2}, ..., \beta_{X,M}]^*$ are deterministic *local* and *global sensitivity vectors*.

Authors in [10] proves the correlation evaluation formula between timing variables represented by the canonical timing model of equation (10). We here prove a similar formula for correlation evaluation between time variables expressed with the ECT model as equation (13) or (14).

Theorem 1: *Given timing variables $X \sim L(\mu_X, \boldsymbol{\alpha}_X, \boldsymbol{\beta}_X)$ and $Y \sim L(\mu_Y, \boldsymbol{\alpha}_Y, \boldsymbol{\beta}_Y)$, the correlation between them can be evaluated as:*

$$cov(X, Y) = \boldsymbol{\alpha}_X^* \boldsymbol{\alpha}_Y + \boldsymbol{\beta}_X^* \boldsymbol{\beta}_Y \tag{15}$$

Proof: By definition:

$$\begin{aligned} & cov(X, Y) = E\{(X - \mu_X)(Y - \mu_Y)\} \\ = \; & cov(\boldsymbol{\alpha}_X^* \boldsymbol{r}, \boldsymbol{\alpha}_Y^* \boldsymbol{r}) + cov(\boldsymbol{\alpha}_X^* \boldsymbol{r}, \boldsymbol{\beta}_Y^* \boldsymbol{g}) + cov(\boldsymbol{\alpha}_Y^* \boldsymbol{r}, \boldsymbol{\beta}_X^* \boldsymbol{g}) + cov(\boldsymbol{\beta}_X^* \boldsymbol{g}, \boldsymbol{\beta}_Y^* \boldsymbol{g}) \\ = \; & E\{\boldsymbol{\alpha}_X^* \boldsymbol{r} \boldsymbol{r}^* \boldsymbol{\alpha}_X\} + E\{\boldsymbol{\beta}_X^* \boldsymbol{g} \boldsymbol{g}^* \boldsymbol{\beta}_X\} = \boldsymbol{\alpha}_X^* \boldsymbol{\alpha}_Y + \boldsymbol{\beta}_X^* \boldsymbol{\beta}_Y \end{aligned}$$

where the independence of $\boldsymbol{r}$ and $\boldsymbol{g}$ is applied. ■

to get the variance of a time variable, it is easy to prove the following corollary:

Corollary 1: *Given timing variable $X \sim L(\mu_X, \boldsymbol{\alpha}_X, \boldsymbol{\beta}_X)$, its variance is:*

$$\sigma_X^2 = \boldsymbol{\alpha}_X^* \boldsymbol{\alpha}_X + \boldsymbol{\beta}_X^* \boldsymbol{\beta}_X \tag{16}$$

which is actually the special case when $X = Y$ of theorem 1.

B. SSTA Algorithm

Before timing analysis, the delay sensitivities of each individual gate/wire are extracted from its Spice model and a gate/wire delay library is then formed. This library, together with the circuit being analyzed, serves as the input of the SSTA algorithm. A SSTA algorithm will then calculate the distributions for all arrival times in the entire circuit by carrying out ADD and MAX operation at each gate/wire block. The overall data flow of the algorithm is summarized in figure 5 where the timing graph in the SSTA is represented by a file with *standard delay variance correlation format(sdvcf)* where both gate/wire delays and connections among gate/wires are specified.

Assuming $X \sim L(\mu_X, \boldsymbol{\alpha}_X, \boldsymbol{\beta}_X)$ and $Y \sim L(\mu_Y, \boldsymbol{\alpha}_Y, \boldsymbol{\beta}_Y)$, the output distribution of an ADD operation $Z = (X + Y) \sim L(\mu_Z, \boldsymbol{\alpha}_Z, \boldsymbol{\beta}_Z)$ can be easily computed as:

$$\mu_Z = \mu_X + \mu_Y; \quad \boldsymbol{\alpha}_Z = \boldsymbol{\alpha}_X + \boldsymbol{\alpha}_Y; \quad \boldsymbol{\beta}_Z = \boldsymbol{\beta}_X + \boldsymbol{\beta}_Y; \tag{17}$$

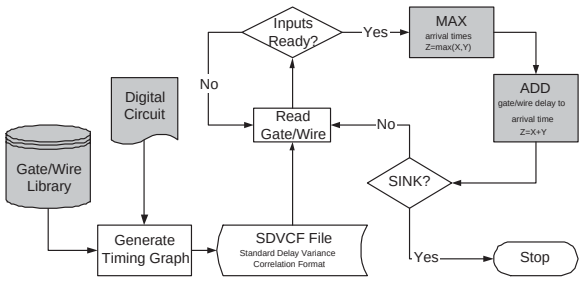


Fig. 5: Block-Based SSTA Algorithm

According to the linear MAX approximation equation (8), the output distribution of MAX operator $Z = max(X, Y)$ will be:

$$\mu_Z = Q\mu_X + (1-Q)\mu_Y + \theta P; \quad \boldsymbol{\alpha}_Z = Q\boldsymbol{\alpha}_X + (1-Q)\boldsymbol{\alpha}_Y; \quad \boldsymbol{\beta}_Z = Q\boldsymbol{\beta}_X + (1-Q)\boldsymbol{\beta}_Y; \tag{18}$$

Clearly the complexity of a single iteration of the SSTA algorithm comes from the sensitivity vector computation and the correlation evaluation involved in the MAX operation. Assuming there are totally M global variations and N gate/wire blocks in the circuit, The overall SSTA complexity will then be $O[(N+M)N]$.

C. Exploration of Sparsity

While working with benchmark circuits, we noticed that many components in the variation vectors have very small sensitivity values, indicating that their contributions to the overall correlation is insignificant. By setting these small coefficient to zero, the sensitivity vector will become a sparse vector that contains many zero components. Motivated by this observation, we apply a *drop-and-lump* method to exploit the sparsity of the sensitivity vector and to further decrease the average complexity of the SSTA algorithm.

For this purpose, a *drop threshold* is selected such that if $\alpha_{X,i}$ or $\beta_{X,j}$ is smaller than this threshold, it is deemed to have small value and will be dropped from the sensitivity vector. However, dropping $\alpha_{X,i}$ or $\beta_{X,j}$ with small magnitude directly is the same as applying truncation to the sensitivity vector. In subsequent computations, the quantization error may accumulate, causing non-negligible error. This is a problem that can not be overlooked for large circuits as demonstrated in the appendix I. Our solution to this problem is to lump those dropped components into a single correction term

$$x_{lump} \quad = \quad \sqrt{\sum x^2_{\textit{dropped Components}}} \tag{19}$$

Using this drop and lump method, the average number of non-zero terms in global sensitivity vector $\boldsymbol{\beta}$ will be M_C and the complexity of the proposed SSTA algorithm will be of $O[(M_C + \Gamma)N]$ where the average number of non-zero terms in local sensitivity vector $\boldsymbol{\alpha}$ is Γ. So what is really dropped in the local sensitivity vector $\boldsymbol{\alpha}$ during computation is then the path correlation and the length of the local sensitivity vector actually gives a good indication to the extent of path correlation in the circuit. So Γ is given a special name of *path correlation length* for a given drop threshold.

Theoretically speaking, if a block is not in any statistically critical paths, its variation will be automatically dropped. On the other hand, if the block is in the critical path but is not statistically important, it will be dropped too. Furthermore, the importance of the variation will decrease after it propagates through a long path. In real circuits, usually only a few blocks in the circuit will survive the arrival time propagation and so that $\Gamma << N$ and $M_C << N$. The computation complexity of the proposed method will be practically $O[(\Gamma + M_C) \cdot N] \approx O[N]$ and a significant reduction of complexity is achieved with the above drop and pool mechanism although it is important know that the actual complexity reduction is highly dependent on the topology of the circuit being analyzed.

V. Simulations and Discussions

Our SSTA algorithm, named as *CLECT*, has already been implemented in C/C++ and tested by benchmark circuits. Before testing, however, all benchmark circuits are re-mapped into a library which has gates of *not, nand2, nand3, nor2, nor3 and xor/xnor*. All library gates are implemented in $0.18\mu m$ technology and their delays are characterized by Monte Carlo Spice simulation with Cadence tools assuming all variation sources follow Gaussian distribution.

For illustration purpose, only three parameter variation are considered global: channel length(L), supply voltage(Vdd) and temperature(T). All other variation sources, specified in the $0.18\mu m$ technology file, are assumed to be localized in the considered gate only. Furthermore we don't address the spatial dependency of the gate delays just for demonstration purpose. In real life, gate delay parameters are position dependent and our method is still applicable.

Extensive Monte Carlo simulations with 10,000 repetitions are used as "Golden Value" for each benchmark circuit. Each repetition is a process of static timing analysis by fixing global and block variation into a set of randomly sampled values. The global variations are sampled once for each repetition while block variation for each gate is sampled every time when the gate delay is computed.

A. Accuracy Improvement with Max Tuple

One simple circuit is given in figure 6 where the overall delay and all internal MAX operators are significantly non-linear as revealed by Monte Carlo simulation shown in figure 7. The skewness of the output distribution is $\kappa = 2.2$ which is significantly larger than zero.

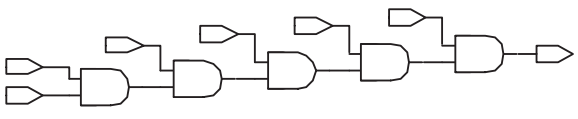


Fig. 6: Circuit whose timing variables are NOT Gaussian

As shown in figure 7, it is very clear that for circuit where MAX operators are significantly non-linear, the existing linear approximation cannot correctly capture the $c.d.f.$ behavior. Especially, the existing method significantly

underestimate the distribution at the high probability level and so that it is risky to use such distribution to predict the circuit performance.

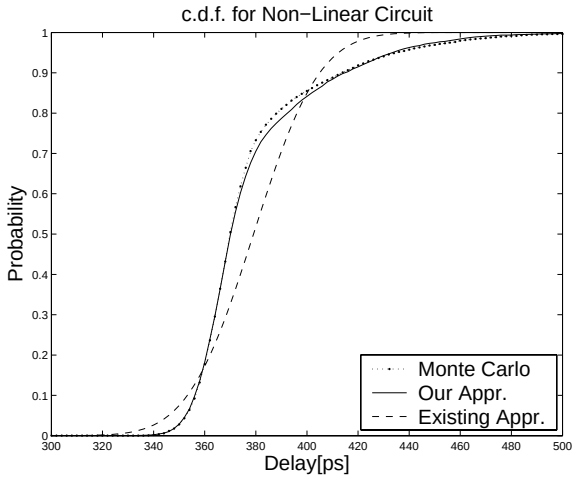


Fig. 7: Comparison of $c.d.f.$ for Non-Linear Circuit between (1)existing linear approximation (2)conditional linear approximation with skewness threshold 0.5 and final tuple size of 3.

Our conditional linear MAX approximation method, on the other hand, matches the exact distribution much better than the existing method. Especially, at the high probability level , the computed distribution is almost exact the same as the one got from Mont Carlo simulation. Such significant accuracy improvement over the existing method makes our method more suitable to predict the performance for non-linear circuits.

B. Accuracy Improvement by Including Path Correlation

Our SSTA method is also tested in the ISCAS benchmark suite. From Monte Carlo simulation, all ISCAS combinational circuits are Gaussian circuits whose MAX operators can be well approximated by linear operators. Both our conditional linear MAX approximation method and existing linear approximation will be good MAX approximation method for these circuits. This nice property comes from the fact that for arrival times in these circuits, bigger mean usually means bigger variance and arrival times are usually positively correlated. So the non-linear condition will not be satisfied for them.

Table I summarizes the error of the arrival time distribution parameters computed at the primary output for each testing circuit from three methods: (1) our method of *CLECT*; (2) *NoPath* where the existing *canonical timing model* is used and no path correlation is considered; (3)*NoCorr* where neither global correlation nor path correlation is considered. μ and σ are mean and standard variation of the distribution. $\tau_{97} = \mu + 2\sigma$ is the 97% delay quantile estimated assuming output delay distribution is Gaussian.

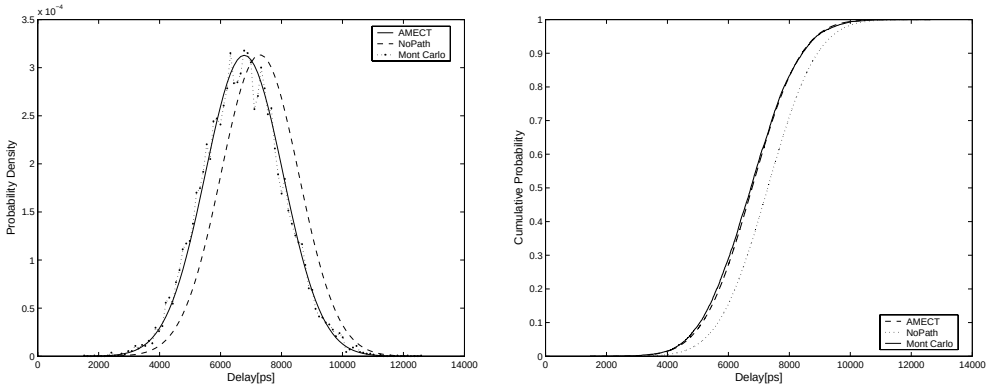
From Table I, it is very clear that method *NoCorr* fails to give reasonable variance estimation because no correlation is considered. This is a good example demonstrating the importance of correlations in SSTA. Table I

Circuit	Mean Error($\delta\mu$)			Variance Error($\delta\sigma$)		
	CLECT	NoPath	NoCorr	CLECT	NoPath	NoCorr
c432	0.79%	4.64%	8.05%	0.50%	1.50%	89.8%
c499	1.04%	4.82%	6.99%	0.89%	0.34%	88.5%
c880	0.15%	1.22%	1.81%	0.53%	0.79%	93.8%
c1355	1.07%	5.81%	6.32%	0.28%	0.95%	95.0%
c1908	0.75%	2.93%	3.66%	0.27%	0.24%	91.8%
c2670	0.35%	3.09%	4.58%	0.00%	0.84%	94.3%
c3540	0.19%	3.75%	4.10%	0.66%	0.36%	95.3%
c5315	0.23%	3.12%	6.06%	0.11%	0.09%	92.8%
c6288	0.53%	8.17%	8.68%	0.65%	1.06%	98.8%
c7552	0.25%	3.27%	6.05%	1.46%	1.09%	92.5%

TABLE I: Distribution Error Respecting to Monte Carlo Results: (1)CLECT:Our Method with Extended Canonical Model;(2)NoPath: Existing Canonical Model where no path correlation is consideered;(3)NoCorr: neither global correlation nor path correlation is considered

also shows that method *NoPath* has significantly larger error in mean estimation than *CLECT* although it shows similar accuracy in variance estimation. As a consequence, method *NoPath* has significantly larger error in 97% delay quantile estimation. This consistently larger error in all simulated circuits shows the importance to use the *extended canonical timing model* and consider the path correlations.

To further elaborate the accuracy improvement of *CLECT* over $NoPath$, Figure 8 shows the $p.d.f.$ and $c.d.f.$ for circuit c6288 from three methods: Mont Carlo, *CLECT* and *NoPath*. Apparently enough, *CLECT* shows excellent accuracy since it considers path correlation. And *NoPath* has significant distribution shift because it uses canonical timing model and path correlation is dropped.



(a) $p.d.f.$ from Three Methods (b) $c.d.f.$ from Three Methods

Fig. 8: $p.d.f.$ and $c.d.f.$ comparison for c6288 from three methods

C. Performance and Path Correlation Length

It has been mentioned in Section IV-C that path correlation length (Γ) is an interesting macro property of the simulated circuit and gives a good indication of the extent of the path correlation existing in that circuit. For the above ISCAS circuits, the path correlation length(Γ) at drop threshold of 1% is summarized in Table II. where the run time improvement over Monte Carlo simulation is also shown.

Name	c432	c499	c880	c1335	c1908
Gate Counts	280	373	641	717	1188
Γ	22.0	11.1	14.2	19.3	27.0
CPU Improve	217x	273x	297x	268x	239x
Name	c2670	c3540	c5315	c6288	c7552
Gate Counts	2004	2485	3865	2704	5355
Γ	15.4	21.2	14.4	80.9	16.0
CPU Improve	399x	350x	220x	22x	355x

TABLE II: Path Correlation Length(Γ) and CPU time Improvement over Monte Carlo Simulation

From Table II, we can firstly conclude that the correlation length Γ is much smaller than the circuit size and basically independent on the circuit size since it remains about $10 - 20$ when circuit size changes dramatically. This observation helps the conclusion we made before about the complexity reduction of our method by using the technique of flexible vector format.

Secondly, the only exceptional high path correlation length among the tested circuits happens with the circuit c6288 which is known as a 16-bit array multiplier. Since there are large amount of equal delay paths in the circuit, large path correlation length is natural: Fewer local sensitivities can be dropped due to the equal importance.

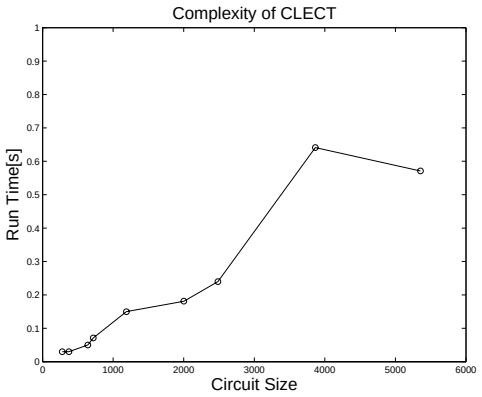


Fig. 9: Run time Complexity of Our SSTA Algorithm

Shown in the figure 9 is the run time complexity of the proposed timing algorithm where the run time and circuit size of all circuits except c6288 are shown. From the figure, it is clear that the run time is almost linear with

respecting to the circuit size even when the circuit size changes dramatically. This result clearly demonstrates our complexity discussion in section IV-C.

To study the relationship between path correlation length and the accuracy of the SSTA method, an experiment is conducted for circuit c6288 and results are shown in figure 10 where the error in τ_{97} and path correlation length are both plotted against the drop threshold. It is clear that the path correlation length drops sharply when the drop threshold changes slightly from zero and maintain almost constant after that. But the error changes steadily when drop threshold changes. This phenomenon proves the efficiency of the drop mechanism introduced in this work since it means we can sacrifice very little accuracy to gain very significant reduction in the path correlation length and so as to save significant amount of CPU time.

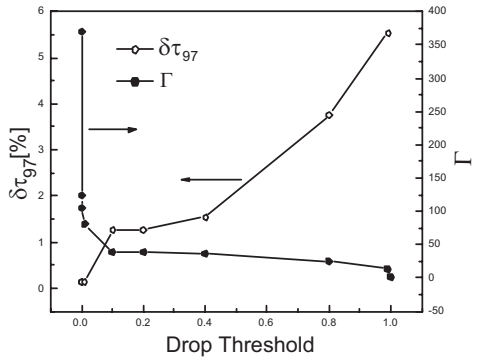


Fig. 10: Path correlation length(Γ) and Error in 97% delay ($\delta\tau_{97}$) when drop threshold changes

VI. CONCLUSIONS

This paper presents a novel method for block-based statistical static timing analysis. We firstly disclose a new method to approximate the MAX operation with a linear operator assisted by the skewness-based linearity evaluation. Secondly we extend the commonly used canonical timing model into an "extended" version to represent the possible occurred path correlation. With these theoretical progress, we are able to evaluate and propagate *both* global and path correlation in the circuit timing graph. We also design a novel algorithm, **CLECT** which treat both global and path correlation simultaneously and systematically. This algorithm, with the help with a drop-and-lump method achieves high accuracy and high performance at the same time as tested by ISCAS circuits and compared with Monte Carlo "golden values".

APPENDIX I
IMPORTANCE OF LUMPING DROPPED VARIATIONS

Using circuit c499 as the example, the variation lumping method introduced in section IV-C is compared with a simple dropping method at drop level of 100%. From table III, the advantage of using the lumping mechanism is clear: the estimation error for 97% delay quantile(τ_{97}) is improve from the 6.1% of the simple dropping mechanism to the 3.4% when the lumping mechanism is used.

Method	Simple Dropping	Lumping	MonteCarlo
τ_{97}	1343ps	1482ps	1431ps

TABLE III: Error Comparison with Approach of [6]

REFERENCES

[1] J.-J. Liou, A. Krstic, L.-C. Wang, and K.-T. Cheng, "False-path-aware statistical timing analysis and efficient path selection for delay testing and timing validation," *Design Automation Conference, 2002. Proceedings. 39th*, pp. 566 – 569, June 2002.

[2] M. Orshansky, "Fast computation of circuit delay probability distribution for timing graphs with arbitary node correlation," *TAU'04*, Feb 2004.

[3] M. Orshansky and K. Keutzer, "A general probabilistic framework for worst case timing analysis," *Design Automation Conference, 2002. Proceedings. 39th*, pp. 556 – 561, June 2002.

[4] A. Agarwal, D. Blaauw, V. Zolotov, S. Sundareswaran, M. Zhao, K. Gala, and R. Panda, "Statistical delay computation considering spatial correlations," *Design Automation Conference, 2003. Proceedings of the ASP-DAC 2003. Asia and South Pacific*, pp. 271 – 276, Jan 2003.

[5] A. Agarwal, V. Zolotov, and D. Blaauw, "Statistical timing analysis using bounds and selective enumeration," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 9, pp. 1243 –1260, Sept 2003.

[6] C. Visweswariah, K. Ravindran, and K. Kalafala, "First-order parameterized block-based statistical timing analysis," *TAU'04*, Feb 2004.

[7] A. Agarwal, D. Blaauw, and V. Zolotov, "Statistical timing analysis for intra-die process variations with spatial correlations," *Computer Aided Design, 2003 International Conference on. ICCAD-2003*, pp. 900 – 907, Nov 2003.

[8] S. Bhardwaj, S. B. Vrudhula, and D. Blaauw, "τau: Timing analysis under uncertainty," *ICCAD'03*, pp. 615–620, Nov 2003.

[9] A. Devgan and C. Kashyap, "Block-based static timing analysis with uncertainty," *ICCAD'03*, pp. 607–614, Nov 2003.

[10] H. Chang and S. S. Sapatnekar, "Statistical timing analysis considering spatial correlations using a single pert-like traversal," *ICCAD'03*, pp. 621–625, Nov 2003.

[11] C. Clark, "The greatest of a finite set of random variables," *Operations Research*, pp. 145–162, March 1961.

[12] P. Lewis and E. Orav, *Simulation Methodology for Statisticians, Operations Analysts, and Engineers*. Chapman and Hall/CRC, 1988.

APPENDIX II
ANSWER REVIEWING QUESTIONS

The authors would like to thank anonymous reviewers for their very insightful and constructive comments. We realize that the MAX approximation method in the last draft gets the most questions, so in this revised manuscript, we have included our most recent research advances in addition to the suggested changes by the reviewers. We demonstrate the drawback of the existing linear approximation method in section II-B. We propose in this draft a new MAX approximation method to make it flexible and accurate for both linear and non-linear cases.(section III) By using skewness, we are able to decide the linearity of the MAX operator analytically. So the linear approximation is *conditionally* applied when MAX is linear. While MAX is non-linear, we delay the evaluation with a form of *Max Tuple*. We have also clearly demonstrated the advantages of such conditional linear approximation over the existing linear approach by using an example circuit in section V-A.

Below are our answers to specific comments.

A. Reviewer Number 1

Q. *...Therefore, it is desirable to explain the test case setup having so good linear behavior.*

A. We have included the explanation in section V-B the reason why the ISCAS circuits behave so linear based on the discussion of the non-linear condition in section III. We also include a non-linear example in section V-A to show the advantage of our conditional linear approximation method.

Q. *The paper says "... rising temperature of substrate may lead to hot spot causing excessive timing variations." However this timing variations are mainly deterministic and repeatable from chip to chip. Therefore statistical timing analysis is not a proper technique to solve this problem.*

A. We believe that there will be some random components in the circuit temperature fluctuation due to the randomness of the power dissipation during run time although we agree that the temperature may still have some deterministic distribution pattern which is predictable in design time. Our approach is useful for all random variations, not specifically restricted in the thermal noise which is only one possible source and may be even not important.

Q. *...It is not clear the meaning of the expression "block by block". What kind of "blocks" are assumed in the timing graph*

A. The meaning of "block" is clarified in this draft to be the gate/wire associated with time delay in the circuit.

Q. *...The only difference is considering independent random variables correspondent to each gate. Therefore it is more correct to say about including all gates delay variations into the canonical form. The previous publications avoided doing it because for real industrial circuits that kind of analysis is infeasible due to too many gates and the corresponding independent variables.*

A. We use the word "extended" just to indicate this is an extension of the existing and generally used canonical model. This extension, together with the intelligent variation pruning method, provide a good way to balance the need for high accuracy and less run time. As illustrated in section V-B, the inclusion of the path correlation

increase the accuracy significantly while the number of extra terms remains very small. Such beneficial property is more detailed explained in section V-C figure 10: small amount of extra terms will have big increase in accuracy.

Q. *...If it is a format it is good to explain it by a figure. If it means the technique to drop small terms, or merge them together into one term then it is not a format. Anyway dropping smaller terms is not anything novel. All approximations are based on that.*

A. We name it as a format to indicate the way we implement it. It is really a technique to drop small items. But it is different than normal dropping since it pool the dropped terms together to reduce the cumulative error that the simple dropping will result in.

Q. *....It is not clear the goal of using non-traditional definition of timing graph and non-traditional terminology which is inconvenient for readers getting used to traditional terminology of timing analysis*

A. We use such non-traditional definition since it reflects the actual way we implement our algorithm. To avoid the confusion, we remove the terminologies of “nodes” and “edges”. Instead, we directly base our discussion on “gate/wire delay” and “arrival time”.

Q. *...Real industrial circuits are usually optimized and therefore they have lots of equally or almost equally critical paths. Otherwise there would be no necessity in block-based timing analysis. If a circuit has only few critical paths then path based timing analysis is the best tool. It is even possible to apply accurate simulation for few critical paths.*

A. We agree with that the extension of the complexity reduction will depend on the topology of the circuit. But we argue that in the case of a lot of equal-length paths, the importance of a gate/wire delay will decrease exponentially when signal propagates through logic levels since they will be multiplied by a factor of 0.5 at each logic level. (section IV-C)

B. Reviewer Number 2

Q. *...So, throwing it away may not be proper. Some deeper understanding of these issues and clarification by the authors is needed before the claim that the non-linearity of max means that we don’t need to match the covariance can be accepted.*

A. We have changed the way to approximate the MAX operator. With the current method, the dependency of of the MAX output on its input, reflecting by the covariance, is preserved. And the MAX is explicitly evaluated only when it is linear and linear approximation is good.(section III)

C. Reviewer Number 3

Q. *...In Subsection A, the authors are incorrect in stating that there is no benefit in preserving the correlation structure during the max() approximation. It is important to note that one of the purposes of timing is to provide insights/directions during the design/synthesis process and hence this contention is incorrect because an accurate picture of the correlation to the global sources of variations will give us better guidance during*

the design process, For Ex., undue sensitivity to a global PFET parameter (ex. mobility) in a particular path can be avoided. In fact, by always picking the minimum of the sensitivities during the max(), this method drops this information.

...In Subsection B, the linear approximation error analysis is provided and Fig. 3 shows the error approximation of the two methods as a function of the difference between the mean of two independent random variables. In fact, the method in [6] and [10] can be shown to minimize the error function in equation (16). For a simple demonstration, X = a z + b, Y = c z + d, can be used as a 1-D example and the integration performed in the z space. Further, Fig. 3 provides an incomplete picture of the approximation error in the different methods, since the error depends not just on the mean difference between X and Y, but also the correlation between them, which is not included here.

A. We changed the way to compute the linear approximation of MAX operator. We will use the existing linear approach only when MAX operator behaves linear. If it is really non-linear, the evaluation is delayed with a form of *max tuple*.

Q. *Subsection C, shows one example circuit with results from the two methods and Table I, provides only the 97% point compared to the "golden" MC metric. This table should be expanded to show other statistical quantities too, similar to Table III, because despite the non-Guassian nature of the actual distribution, it is still of significant interest to the designer to know if the distribution is shallow or not as well as the skew in it (namely the statistical quantities: mean, variance, skew, kurtosis)....*

A. Our new method matches the $c.d.f.$ got from Monte Carlo method much better than the existing method shown in figure 6.

Q. *...Since the primary justification for the drop-threshold technique is memory/CPU usage, it should be tabulated in Table V., to allow the reader to judge its utility and cost*

A. It is done in this draft.

Q. *A better explanation of "NoPath" and "NoCorr" in Table III is needed....*

A. It is done in the text and the cation of the table as well.

Q. *Is there a type on Page 16, first line giving the* $tau_9 7 = mu + 2sigma$ *Did the authors mean "+ 3 sigma" here ?*

A. It is not a typo. For single side range, 97% percentile is equal to mu + 2 sigma.