Non-Linear Statistical Static Timing Analysis for Non-Gaussian Variation Sources

ABSTRACT

Existing statistical static timing analysis (SSTA) techniques suffer from limited modeling capability by using a linear delay model with Gaussian distribution, or have scalability problems due to expensive operations involved to handle non-Gaussian variation sources or non-linear delays. To overcome these limitations, we propose a novel SSTA technique to handle both nonlinear delay dependency and non-Gaussian variation sources simultaneously. We develop efficient algorithms to perform all statistical atomic operations (such as max and add) efficiently via either closed-form formulas or one-dimensional lookup tables. The resulting timing quantity provably preserves the correlation with variation sources to the third-order. We prove that the complexity of our algorithm is linear in both variation sources and circuit sizes, hence our algorithm scales well for large designs. Compared to Monte Carlo simulation for non-Gaussian variation sources and nonlinear delay models, our approach predicts all timing characteristics of circuit delay with less than 2% error.

1. INTRODUCTION

For the CMOS technology scaling, process variation has become a potential show-stopper if not appropriately handled. Statistical static timing analysis (SSTA), in particular, block-based parameterized SSTA [1, 2, 3, 4, 5, 6], has thus become the frontier research topic in recent years in combating such variation effects. The goal of SSTA is to parameterize timing characteristics of the timing graph as a function of the underlying sources of process parameters that are modeled as random variables. By performing SSTA, designers can obtain the timing distribution (yield) and its sensitivity to various process parameters. Such information is of tremendous value for both timing sign-off and design optimization for robustness and high profit margins.

Although many studies have been done on SSTA in recent years, the problem is far from being solved completely. For example, [1, 2] assumed that all variation sources are Gaussian and independent of one another. Based on a linear delay model, [2] proposed a linear-time algorithm for SSTA, in which all atomic operations (such as max and add) can be performed efficiently via the concept of *tightness probability*. Because all variation sources are assumed to be Gaussian, so is the delay distribution under the linear delay model.

Such a Gaussian assumption is, however, no longer tolerable as more complicated or large-scale variation sources are taken into account in the nanometer manufacturing regime. For example, via resistance is known to be non-Gaussian with asymmetric distribution [7], and dopant concentration is more suitably modeled as a Poisson distribution [6]. In addition, the linear dependency of delay on the variation sources is also not accurate, especially when variation sources become large [8]. For example, gate delay is inherently a nonlinear function of channel length and Vth [7, 3], which are two common sources of variation. Similarly, interconnect delay is also a nonlinear function of interconnect geometries [3, 4], which vary because of chemical-mechanical polishing. These combined non-Gaussian nonlinear variation effects invalidate the linear delay model with Gaussian assumption in the existing SSTA.

Recently, non-Gaussian variation sources were addressed in [6], where independent component analysis (ICA) was used to find a set of independent components (not necessary Gaussian) to approximate the correlated non-Gaussian random variables. To do this, however, a complicated moment matching algorithm has to be used to make those atomic statistical operations feasible. Moreover, it is still based on a linear delay model, which cannot capture the nonlinear dependency of delays on process parameters. To capture these nonlinear dependency effects, [3, 4] proposed to use a quadratic delay model for SSTA. But to contain the complexity, they had to assume that all variation sources must follow a Gaussian distribution, even though the delay D itself may not be Gaussian. To compute $max(D_1, D_2)$, [3] first developed closed formulas to compute the mean and variance of the quadratic form. It then treats D_1 and D_2 as a Gaussian distribution to obtain the tightness probability. There is, however, no justification on why the tightness probability formula developed for Gaussian distributions can be applied for non-Gaussian distributions. [4] tried to re-construct $max(D_1, D_2)$ through moment matching. To obtain those moments, however, many expensive numerical integration (two-dimensional) operations have to be applied.

[5] and [7] are the only only existing studies that try to handle both nonlinear and non-Gaussian effects simultaneously. However, [5] computes $max(D_1, D_2)$, by regression based on Monte Carlo simulation, which is slow; and there is no guarantee that the correlation between the regression result and variation sources could be kept. To reduce the complexity, [7] proposes to separate the variation sources into two camps: the first camp is Gaussian variation with linear timing dependency, and the second camp is non-Gaussian variation with non-linear timing dependency. This dichotomy of variation sources, however, is somehow artificial. The max operation is also through tightness probability, which is computed via expensive numerical multi-

dimensional integration. Hence its scalability to handle a large number of non-Gaussian variation sources is limited.

In this work, we propose a novel nonlinear and non-Gaussian SSTA technique (n^2SSTA). The major contributions of this work are multi-fold. (1) Both nonlinear dependency and non-Gaussian variation sources are handled simultaneously for timing analysis. (2) All statistical atomic operations are performed efficiently via either closed-form formulas or one-dimensional lookup tables. (3) The resulting parameterized timing quantities provably preserve the correlation with variation sources to the third-order. (4) The complexity of the n^2SSTA algorithm is linear in both variation sources and circuit sizes. Compared to Monte Carlo simulation for non-Gaussian variation sources and nonlinear delay models, our approach predicts all timing characteristics of circuit delay with less than 2% error.

The rest of the paper is organized as follows. Section 2 presents our nonlinear and non-Gaussian delay modeling. Section 3 discuss our n^2SSTA technique with focus on the max and add atomic operations. We present experiments in Section 4, and conclude in Section 5.

2. PRELIMINARIES AND MODELING

In general, device or interconnect delays of a design are a complicated nonlinear function of the underlying process parameters and it can be described as

$$D = F(X_1, X_2, \ldots, X_i, \ldots), \tag{1}$$

where the process parameters (such as channel length and Vth) are modeled as a random variable X_i. In reality, the exact form of function F is not known, and X_i are not necessarily Gaussian. In practice, however, we can employ Taylor expansion as an approximation to the function F.

The simplest approximation is the first- and second-order Taylor expansion as shown below

$$D \approx d_0 + \sum a_i X_i, \tag{2}$$

$$D \approx d_0 + \sum a_i X_i + \sum b_i X_i^2 + \sum\nolimits_{i \neq k} b_{i,k} X_i X_k, \tag{3}$$

where d_0 is the nominal value of D; a_i and b_i are the first- and second-order sensitivities of D to X_i, respectively; and $b_{i,k}$ are the sensitivity to the joint variation of X_i and X_k. When all X_i are assumed to be Gaussian, (2) is called the *first-order canonical form*, and is widely used for SSTA [2, 1]; whereas (3) is called the *quadratic delay model*, and has been studied in [8, 3, 4, 5]. These models based on Gaussian assumptions are limited in their modeling capability to reflect the reality. For example, not all variation sources are Gaussian, and results after max are also not Gaussian. While some may appear to be Gaussian, in reality, their variation cannot vary from $-\infty$ to $+\infty$ as a Gaussian distribution does.

Therefore, we propose a different quadratic model to represent all timing quantities in a timing graph as follows:

$$D = d_0 + \sum (a_i X_i + b_i X_i^2) + a_r X_r + b_r X_r^2, \tag{4}$$

where X_i represents global sources of variation, and X_r represents purely independent random variation. Unlike previous work, we allow X_i to follow arbitrary random distributions with bounded values[1], i.e., $-w_i \le X_i \le w_i$. We refer to the delay model (4) as *general canonical form* in this paper. Compared to existing work [5, 3, 4, 6], our model is unique in the sense that we capture the nonlinearity of timing dependence on variation sources, and handle the non-Gaussian distribution of variation sources at the same time.

For simplicity, we ignore cross terms ($X_i X_k$) in (4) and assume independence between X_i. The reasons are the timing dependency on cross terms is usually weak. When X_i and X_k are Gaussian, cross terms can be replaced by non-cross terms through orthogonalization [4]. When X_i are correlated, techniques like ICA may be used to generate a set of new independent components [6]. Without loss of generality, we assume that all variation sources are centered with zero mean values, i.e., $E[X_i] = 0$. We denote the probability density function (PDF) of X_i as $g_i(x_i)$, which can be given as either a closed formula or an empirical lookup table. Knowing the PDF of X_i, we can easily compute its t^{th}-order *raw moments*, i.e., $m_{i,t} = E(X_i^t)$. We can also compute the raw moments of D, i.e., $M_t = E(D^t)$, by using the Binomial moment evaluation technique [8]. With raw moments, *central moments* can be computed easily. For example, the first three central moments of D are

$$U_1 = M_1, \tag{5}$$

$$U_2 = M_2 - M_1^2, \tag{6}$$

$$U_3 = M_3 + 2M_1^3 - 3M_1 M_2. \tag{7}$$

Note that the first- and second-order central moments U_1 are essentially D's mean ($\mu = U_1$) and variance ($\sigma^2 = U_2$), respectively. The *skewness* of D is U_3/σ^3.

3. ATOMIC OPERATIONS FOR SSTA

To compute the arrival time and required arrival time in a block-based SSTA framework, four atomic operations are sufficient, i.e., addition, subtraction, maximum, and minimum, provided that we can represent all timing results after each operation back to the same general canonical form (4). Because of the symmetry between addition and subtraction (similarly maximum and minimum) operations, in the following, we will only discuss operations on addition and maximum. It is understood that similar discussion applies to subtraction and minimum operations, as well. That is, given D_1 and D_2 in the form of (4),

$$D_1 = d_{01} + \sum (a_{i1} X_i + b_{i1} X_i^2) + a_{r1} X_{r1} + b_{r1} X_{r1}^2, \tag{8}$$

$$D_2 = d_{02} + \sum (a_{i2} X_i + b_{i2} X_i^2) + a_{r2} X_{r2} + b_{r2} X_{r2}^2, \tag{9}$$

we want to compute $D = D_1 + D_2$ or $D = max(D_1, D_2)$ such that the resulting D can be represented as (4).

Denote $\Delta D_1 = D_1 - \mu_1$ and $\Delta D_2 = D_2 - \mu_2$ with μ_1 and μ_2 as mean values of D_1 and D_2, respectively. As both D_1 and D_2 model timing quantities in a timing graph, their values are physically lower- and upper-bounded:

$$-l \le \Delta D_1 \le l, \quad -h \le \Delta D_2 \le h. \tag{10}$$

For a practical problem, the size of the bound, l or h, can be easily determined by relating to either its minimum and maximum delays, or its sigma-sample values.

3.1 Max Operation

The max operation is the hardest operation for block-based SSTA. In this work, we propose a novel technique to efficiently compute the max of two general canonical forms,

[1] For Gaussian variables, whose lower and upper bound can be reasonably set as its k-sigma values to bound its variation in reality. For example, $w_i = 4\sigma_i$ or $5\sigma_i$ with $k = 4$ or 5.

Input: D_1 and D_2 in format of (8) and (9)
Output: $D \approx max(D_1, D_2)$ in format of (4)

1. Compute (D_1, D_2)'s JPDF $g(D_1, D_2)$ via Fourier series;
2. Compute raw moments of $max(D_1, D_2)$: $M_t = E[max(D_1, D_2)^t]$;
3. Compute $E[X_i^t max(D_1, D_2)]$ for t=1,2;
4. Compute a_i and b_i in (4) by matching $E[X_i^t max(D_1, D_2)]$ for t=1,2;
5. Compute a_r and b_r in (4) by matching $max(D_1, D_2)$'s $2^{nd}-$ and 3^{rd}-order moments;
6. Compute d_0 in (4) by matching $max(D_1, D_2)$'s 1^{st}-order moment.

Figure 1: Overall algorithm for computing $max(D_1,D_2)$.

i.e., $D = max(D_1, D_2)$, and the result D will still be in the form of (4). With respect to the overall flow in Fig. 1, we first compute the joint PDF (JPDF) of D_1 and D_2, which is achieved via an efficient algorithm based on Fourier series. Knowing JPDF of D_1 and D_2, we can compute the raw moments of $max(D_1, D_2)$ to arbitrary orders efficiently. Similarly, the joint moments (related to correlation) between $max(D_1, D_2)$ and variation sources X_i can also be computed efficiently. With the above computation ready, we re-construct the general canonical form of $D \approx max(D_1, D_2)$ by matching the joint moments between $max(D_1, D_2)$ and X_i, the first three order moments of $max(D_1, D_2)$. In the following, we discuss the details of our approach.

3.1.1 JPDF via Fourier Series

Computing JPDF is an essential step for max operation. In [2], because both D_1 and D_2 are Gaussian distribution in linear canonical form (2), their JPDF can be easily obtained by computing the covariance between D_1 and D_2. When D_1 and D_2 are non-Gaussian, however, no closed form can be easily derived to compute their JPDF. For example, [4] resorted to expensive numerical integration to obtain JPDF of two non-Gaussian distributions in quadratic form.

In the following approach, we propose a novel method to efficiently compute JPDF of D_1 and D_2 in general canonical form. Denote JPDF of ΔD_1 and ΔD_2 in (10) as $f(v_1, v_2)$, and JPDF of D_1 and D_2 as $g(v_1, v_2)$. It is easy to show that:

$$g(v_1, v_2) = f(v_1 - \mu_1, v_2 - \mu_2). \qquad (11)$$

Hence knowing $f(v_1, v_2)$ is equivalent to knowing $g(v_1, v_2)$.

To compute JPDF $f(v_1, v_2)$ in the region $[-l, l; -h, h]$, we approximate it via its first K orders of Fourier series as follows:

$$f(v_1, v_2) \approx \sum_{p,q=-K}^{K} \alpha_{pq} \cdot e^{\zeta_p v_1 + \eta_q v_2}, \qquad (12)$$

where $\zeta_p = jp\pi/l$ and $\eta_q = jq\pi/h$ with $j = \sqrt{-1}$. The Fourier coefficients α_{pq} is given by

$$\alpha_{pq} = \frac{1}{4lh} \int_{-l}^{l} \int_{-h}^{h} e^{-\zeta_p v_1 - \eta_q v_2} \cdot f(v_1, v_2) dv_1 dv_2. \qquad (13)$$

Because JPDF $f(v_1, v_2)$ is zero outside the valid region, (13) can be further simplified as

$$\begin{aligned} \alpha_{pq} &= \frac{1}{4lh} E[e^{-\zeta_p \Delta D_1 - \eta_q \Delta D_2}] \\ &= \frac{1}{4lh} e^{-Y_{c,pq}} E[e^{-Y_{r1,pq} - Y_{r2,pq} - \sum Y_{i,pq}}], \end{aligned} \qquad (14)$$

where $Y_{c,pq} = \zeta_p(d_{01} - \mu_1) + \eta_q(d_{02} - \mu_2)$; $Y_{i,pq} = (\zeta_p a_{i1} + \eta_q a_{i2})X_i + (\zeta_p b_{i1} + \eta_q b_{i2})X_i^2$; $Y_{r1,pq} = \zeta_p a_{r1} X_{r1} + \zeta_p b_{r1} X_{r1}^2$; and $Y_{r2,pq} = \eta_q a_{r2} X_{r2} + \eta_q b_{r2} X_{r2}^2$. Because all X_i's are independent, so are all $Y_{i,pq}$'s, $Y_{r1,pq}$, and $Y_{r2,pq}$. Then α_{pq} can be further simplified as:

$$\alpha_{pq} = \frac{1}{4lh} e^{-Y_{c,pq}} E[e^{-Y_{r1,pq}}] E[e^{-Y_{r2,pq}}] \prod E[e^{-Y_{i,pq}}]. \qquad (15)$$

As both $Y_{i,pq}$, $Y_{r1,pq}$ and $Y_{r2,pq}$ can be written as a general form as $Y = c_1 X_i + c_2 X_i^2$ with c_1 and c_2 being two constant values, in the following, we discuss how to compute $E[e^{-Y}]$ in its general form. By definition,

$$E[e^{-Y}] = \int_{-w_i}^{w_i} e^{-c_1 x_i - c_2 x_i^2} g_i(x_i) dx_i, \qquad (16)$$

where $g_i(x_i)$ is PDF of X_i, whose range is given by $-w_i \leq X_i \leq w_i$.

For arbitrary $g_i(x_i)$, we can also build a two-dimensional (2D) table indexed by c_1 and c_2 to speed-up computing (16). But the size of 2D-table may be very large. In the following, we present an effective solution that requires only 1D-table lookup. We divide X_i's range into M number of small sub-regions, $S_1 \ldots S_M$. Within each small sub-region, we approximate x_i^2 by its first-order Taylor expansion around the sub-region's center point x_{i0}, i.e.,

$$x_i^2 \approx x_{i0}^2 + 2x_{i0}(x_i - x_{i0}) = 2x_i x_{i0} - x_{i0}^2. \qquad (17)$$

By substituting (17) into (16), we obtain

$$\begin{aligned} E[e^{-Y}] &\approx \sum_{i=1}^{M} \int_{S_i} e^{-c_1 x_i - c_2 (2x_i x_{i0} - x_{i0}^2)} g_i(x_i) dx_i \\ &= \sum_{i=1}^{M} e^{c_2 x_{i0}^2} \mathcal{F}_i(-jc_1 - 2jc_2 x_{i0}), \end{aligned} \qquad (18)$$

where $\mathcal{F}_i(\cdot)$ is the Fourier transformation of $g_i(x_i)$ in the sub-region S_i. So we can pre-calculate all $\mathcal{F}_i(\cdot)$ for all pre-determined sub-regions for each variation source, and store these results into a 1D lookup table for SSTA. In this work, we uniformly divide the valid region of each variation source into twelve ($M = 12$) sub-regions.

	d_0	a_i	b_i	a_r	b_r
D_1	0	{2,1,3,2}	{4,3,4,4}	1	2
D_2	0	{1,2,2,1}	{3,4,3,3}	1	2

Table 1: Experiment setting to verify $max(D_1, D_2)$.

To validate our computing of JPDF of two general canonical equations, we compare our computed JPDF with Monte-Carlo simulated JPDF. One of the examples is shown in Fig. 2 with four sources of random variables (i.e., X_i for $i = 1, 2, 3, 4$) that all follow a uniform distribution in the range of $[-0.5, 0.5]$, as shown in Table 1, which will be used for the rest of this section for verification. The order of Fourier series to approximate JPDF is four ($K = 4$). Fig 2 convincingly shows that our approach is accurate in predicting the exact JPDF.

3.1.2 Raw Moments of $Max(D_1, D_2)$

In this section, we present a technique to compute raw moments $M_t = E[max(D_1, D_2)^t]$ for $Max(D_1, D_2)$. By definition, knowing (D_1,D_2)' JPDF $g(v_1, v_2)$, M_t can be computed by

$$M_t = \iint_{v_1 > v_2} v_1^t g(v_1, v_2) dv_1 dv_2 + \iint_{v_2 > v_1} v_2^t g(v_1, v_2) dv_1 dv_2. \qquad (19)$$

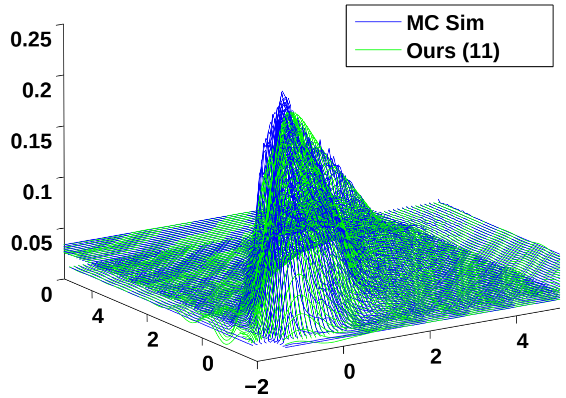

Figure 2: Joint PDF comparison.

According to (11) and (12), M_t can be further written as

$$M_t = \sum_{p,q=-k}^{k} \alpha_{pq} \cdot L(t,p,q,l,h,\mu_1,\mu_2), \qquad (20)$$

where $L(t,p,q,l,h,\mu_1,\mu_2)$ is defined as follows:

$$\begin{aligned} L &= \iint_{v_1>v_2} v_1^t e^{\zeta_p(v_1-\mu_1)+\eta_q(v_2-\mu_2)} dv_1 dv_2 + \\ & \iint_{v_2>v_1} v_2^t e^{\zeta_p(v_1-\mu_1)+\eta_q(v_2-\mu_2)} dv_1 dv_2. \qquad (21) \end{aligned}$$

It is easy to see that (21) can be evaluated via closed form formulas efficiently. For example, in the case of $\mu_1 - l < \mu_2 - h$, we have $L=\frac{1}{\eta_q}e^{-\zeta_p\mu_1}\left(e^{-\eta_q\mu_2}J(t,\zeta_p+\eta_q,\mu_2-h,\mu_2+h)\text{-}(-1)^q J(t,\zeta_p,\mu_2-h,\mu_2+h)\right)+\frac{1}{\zeta_p}e^{-\eta_q\mu_2}\left(e^{-\zeta_p\mu_1}J(t,\zeta_p+\eta_q,\mu_2-h,\mu_2+h)\text{-}(-1)^p J(t,\eta_q,\mu_2-h,\mu_2+h)\right)$, where the function $J(t,\gamma,\tau_1,\tau_2)=\int_{\tau_1}^{\tau_2} x^t e^{\gamma x}dx$ and can be computed by integration by parts, i.e.,

$$J = \frac{1}{\gamma^{t+1}} \sum_{i=0}^{t} (-1)^{t-i} \frac{\gamma^i t!}{(n-i)!} (e^{\gamma\tau_2}\tau_2^i - e^{\gamma\tau_1}\tau_1^i). \qquad (22)$$

Similar equations can be derived for other cases, as well. In the interest of space, we omit the details and refer readers to our technical report to be cited upon publication.

We compare our approach to Monte Carlo simulation to validate (20) in computing the raw moments. Based on the same setting as in Table 1, Table 2 compares the first three-order raw moments of $max(D_1, D_2)$. Our computation is accurate, and the relative error is less than 5%.

Raw Moment	1^{st}-order	2^{nd}-order	3^{th}-order
This work (20)	3.62	15.31	72.68
Monte Carlo	3.65	15.61	75.33
Error	0.90%	1.92%	3.52%

Table 2: Raw moment computation.

3.1.3 Computation of $E[X_i^t \cdot Max(D_1, D_2)]$

To compute $Ec_{i,t} = E[X_i^t \cdot max(D_1,D_2)]$, we first obtain JPDF of X_i, ΔD_1, and ΔD_2 by using a technique similar to that developed in Section 3.1.1. JPDF $f(x_i,v_1,v_2)$ is approximated by the first K-order Fourier series as follows:

$$f(x_i,v_1,v_2) \approx \sum_{p,q,s=-K}^{K} \beta_{pqs}^i \cdot e^{\xi_{i,s}x_i+\zeta_p v_1+\eta_q v_2}, \qquad (23)$$

where $\xi_{i,s} = js\pi/w_i$, and coefficients β_{pqs}^i are given by

$$\beta_{pqs}^i = \frac{e^{Y_{c,pq}}}{8w_i lh} E[e^{-Y_{r1,pq}}]E[e^{-Y_{r2,pq}}]E[e^{-\hat{Y}_{i,pq}}]\prod_{k\neq i} E[e^{-Y_{k,pq}}]$$

where $\hat{Y}_{i,pq}=(\zeta_p a_{i1}+\eta_q a_{i2}-\xi_{i,s})X_i+(\zeta_p b_{i1}+\eta_q b_{i2})X_i^2$. The above expectation has the same form as (16), hence they can be easily evaluated, as well.

After obtaining JPDF $f(x_i,v_1,v_2)$ of X_i, ΔD_1, and ΔD_2, JPDF of X_i, D_1, and D_2 can be obtained as $g(x_i,v_1,v_2)=f(x_i,v_1-\mu_1,v_2-\mu_2)$. Hence $Ec_{i,t}$ can be computed by

$$\begin{aligned} Ec_{i,t} = & \iiint_{v_1>v_2} x_i^t v_1 f(x_i,v_1-\mu_1,v_2-\mu_2)dx_i dv_1 dv_2 + \\ & \iiint_{v_2>v_1} x_i^t v_2 f(x_i,v_1-\mu_1,v_2-\mu_2)dx_i dv_1 dv_2. \end{aligned}$$

As $f(x_i,v_1,v_2)$ is known from (23), we finally obtain

$$Ec_{i,t} = \sum_{p,q,s=-K}^{K} \beta_{pqs}^i J(t,\xi_{i,s},-w_i,w_i)L(1,p,q,l,h,\mu_1,\mu_2), (24)$$

using functions L and J in (21) and (22), respectively.

Table 3 compares our computed $Ec_{i,1}$ and and $Ec_{i,2}$ with Monte-Carlo simulation based on the same settings in Table 1. We see that our approach is accurate with less than 6% error compared to Monte Carlo simulation.

	Variation	X_1	X_2	X_3	X_4
$Ec_{i,1}$	Ours (24)	0.152	0.098	0.166	0.155
	MC	0.158	0.095	0.168	0.159
	Error	3.8%	2.9%	0.8%	2.4%
$Ec_{i,2}$	Ours (24)	0.355	0.362	0.356	0.366
	MC	0.338	0.345	0.338	0.347
	Error	5.0%	5.2%	5.3%	5.3%

Table 3: Computation of $Ec_{i,1}$ and $Ec_{i,2}$.

3.1.4 General Canonical Form for $D = max(D_1, D_2)$

To reconstruct $D = max(D_1, D_2)$ into the general canonical form in (4), we need to determine d_0, a_i, b_i, a_r and b_r. For computational efficiency, we rewrite D in (4) as follows:

$$\begin{aligned} D &= d_0' + \sum Z_i + Z_r, && (25) \\ Z_i &= a_i X_i + b_i(X_i^2 - m_{i,2}), && (26) \\ Z_r &= a_r X_r + b_r(X_r^2 - m_{r,2}), && (27) \\ d_0' &= d_0 + b_r m_{r,2} + \sum b_i m_{i,2}, && (28) \end{aligned}$$

where $m_{i,t}$ is the t^{th}-order moment of X_i. Because X_i's are independent with zero means, so are the Z_i's and Z_r. Therefore, according to (25), the first three-order central moments of D can be evaluated as

$$\begin{aligned} U_1 &= d_0', && (29) \\ U_2 &= \sum \mu_{zi,2} + \mu_{zr,2}, && (30) \\ U_3 &= \sum \mu_{zi,3} + \mu_{zr,3}, && (31) \end{aligned}$$

where $\mu_{zi,t}$ and $\mu_{zr,t}$ are the t^{th}-order central moment of Z_i and Z_r, respectively. According to the definition of Z_r (or

Z_i), we compute $\mu_{zr,2}$ (or $\mu_{zi,2}$) by

$$\mu_{zr,2} = (m_{r,4} - m_{r,2}^2)b_r^2 + 2m_{r,3}a_r b_r + m_{r,2}^2 a_r^2. \tag{32}$$

Similarly, $\mu_{zr,3}$ (or $\mu_{zi,3}$) is computed by

$$\mu_{zr,3} = (m_{r,6} - 3m_{r,4}m_{r,2} + m_{r,2}^3)b_r^3 + m_{r,3}a_r^3 + 3(m_{r,4} - m_{r,2}^2)a_r^2 b_r + 3(m_{r,5} - m_{r,3}m_{r,2})a_r b_r^2. \tag{33}$$

By equating (29) to (31) with (5) to (7) correspondingly, we match D in (4) with the first three-order central moments of the exact $max(D_1, D_2)$. Moreover, we also strive to match the joint moments of X_i and $max(D_1,D_2)$ to the third-order, as the latter are closely related to the correlation between X_i and $max(D_1,D_2)$. This is achieved by determining a_i and b_i as follows:

$$Ec_{i,1} = a_i m_{i,2} + b_i m_{i,3}, \tag{34}$$

$$Ec_{i,2} = \mu m_{i,2} + a_i m_{i,3} + b_i(m_{i,4} - m_{i,2}^2). \tag{35}$$

As $Ec_{i,1}$ and $Ec_{i,2}$ are known from (24) and the moments $m_{i,t}$, we solve for a_i and b_i from (34) and (35), which form a linear system of equations with two unknowns. Knowing all a_i and b_i, we determine a_r and b_r by plugging $\mu_{zr,2}$ of (32), $\mu_{zr,3}$ of (33), U_2 of (6), and U_3 of (7) into (30) and (31) and solving these system of equations. Then the only unknown left for D in (4) is d_0 can be obtained by equating (29) to (5).

To verify that our constructed D is accurate in approximating $max(D_1, D_2)$, we compare our results with Monte Carlo simulation. Based on the settings in Table 1, Fig. 3 shows that our approach matches Monte Carlo simulation accurately and it captures not only mean and variance, but also the skewness. In contrast, the Gaussian approximation that matches only mean and variance is very different from Monte Carlo simulation.

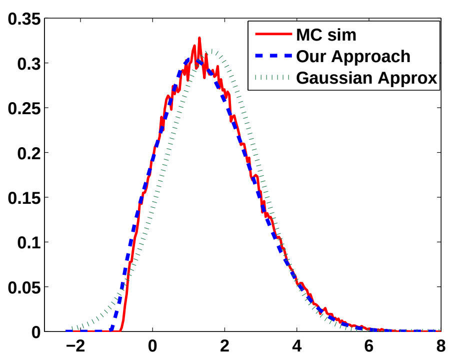

Figure 3: Comparison of PDF after max operation.

3.2 Add Operation

To compute $D = D_1 + D_2$ and put it back in (4), it can be done straight-forwardly for both the nominal value and global random variables' coefficients, as we only need to add up the corresponding terms, i.e., $d_0 = d_{01} + d_{02}$, $a_i = a_{i1} + a_{i2}$, and $b_i = b_{i1} + b_{i2}$.

For the uncorrelated random variable, one approach is to keep the correlation between the addition result with the two input uncorrelated random variables (X_{r1} and X_{r2}). This is achieved by promoting these two variables into global random variables after addition, thus their coefficients are the same as before. The downside of this approach is that it causes the length of our general canonical form to be longer after each addition. An alternative way is to combine the two input uncorrelated random variables (X_{r1} and X_{r2}) into a new uncorrelated random variables X_r by matching both the second- and third-order central moments of the exact addition operation. This is similar to solving a_r and b_r for $max(D_1, D_2)$, hence we omit the details in the interest of space. The drawback of this approach is that the correlation between D and X_{r1} and X_{r2} is lost.

We see that the above two approaches complement each other. Following a similar idea as [9], we choose the first approach when the coefficient of X_{r1} and X_{r2} is larger than a pre-defined threshold so we do not lose correlation, and choose the second approach when the coefficient of X_{r1} and X_{r2} is small so we can keep the form compact. But either way, the result after addition will be still in the form of (4).

3.3 Complexity Analysis

For the max operation as shown in Fig. 1, the complexity is low because all computation involved is based on either closed-form formulas or one-dimensional lookup tables. The complexity of one max operation is thus $\mathcal{O}\{K^3N\}$, where K is the highest order for Fourier series, and N is the number of variation sources. In another words, our max operation is linear with respect to variation sources. In practice, both K and N are small numbers compared to circuit size, so the complexity of maximum operation is constant. Similar arguments hold for the add operation. Since both max and add can be done in constant time, our block-based SSTA can be done in linear time in circuit sizes.

4. EXPERIMENTAL RESULTS

We have implemented our n^2SSTA algorithm in C, and applied it to the ISCAS89 suite of benchmarks obtained from [10]. Because there is no variation information in the original benchmark, as a proof of concept, we randomly generate such information in this work. For each benchmark, the number of variation sources ranges from 5 to 20 depending on circuit sizes. The total variation amount ranges from 5% to 20% of its nominal value. For each variation source, it follows either a Gaussian distribution, uniform distribution, or tri-angle distribution obtained from uniform-sum distribution of degree two. For easy comparison, the final circuit delay is normalized with respect to its nominal delay, thus results reported here are unit-less. We compare the solution quality of n^2SSTA with the golden Monte Carlo simulation of 100,000 runs.

Similar to the experiment setting in [11], Table 4 compares n^2SSTA and Monte Carlo simulation in terms of the ratio between sigma and mean, the 95% yield timing, and runtime in second. In the first (or second) set of experiments, all variation sources follow a uniform (or a tri-angle) distribution. According to the six benchmarks reported, we see that our n^2SSTA algorithm can accurately predict all timing metrics with, on average, less than 2% error compared to Monte Carlo simulation, while achieving about 25× speedup. The runtime of n^2SSTA roughly grows linearly as the circuit size grows. We also show the PDF comparison result in Fig 4. We see that our n^2SSTA algorithm obtains almost the same PDF as Monte Carlo simulation. This convincingly shows the validity and accuracy of our n^2SSTA algorithm in predicting timing distribution.

We also compare n^2SSTA with our implementation of [2] (denoted as linSSTA) by assuming Gaussian variations and

Bench mark	Monte Carlo			n^2SSTA		
	σ/μ %	95% yield	run time (s)	σ/μ %	95% yield	run time (s)
Uniform Variation Sources						
s27	14.7	1.41	3.4	14.8	1.41	0.80
s386	14.9	1.41	61	14.9	1.41	2.00
s444	15.1	1.42	44	14.8	1.42	3.07
s832	15.0	1.41	91	14.5	1.41	5.24
s1494	15.4	1.41	285	15.6	1.41	7.97
s5378	15.3	1.42	855	14.9	1.42	27.1
Avg	-	-	-	1.37%	0.01%	1/22.3
Tri-angle Variation Sources						
s27	13.6	1.44	4.3	13.8	1.44	0.80
s386	13.6	1.45	61	13.7	1.45	1.88
s444	14.2	1.47	57	14.3	1.47	2.99
s832	15.0	1.48	115	15.0	1.48	6.81
s1494	14.1	1.45	284	14.3	1.45	7.60
s5378	13.9	1.45	903	14.0	1.45	25.6
Avg	-	-	-	0.73%	0.01%	1/24.4

Table 4: Experiments for non-Gaussian variations and nonlinear delay. The number in a circuit name is the number of gates in the circuit.

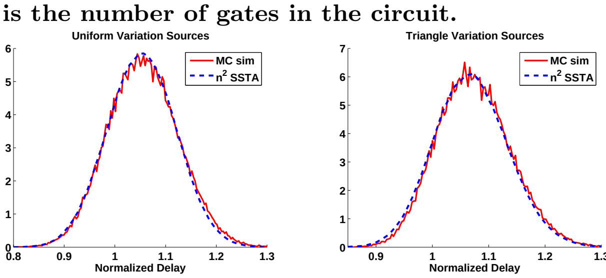

Figure 4: PDF comparison for s5378 with non-Gaussian variations and nonlinear delay

linear delay model for both. From Table 5, we see that in predicting σ/μ, n^2SSTA matches Monte Carlo simulation well with about 5.5% error, while *lin*SSTA has about 11% error. [2] This clearly shows that n^2SSTA is not only more general, but also more accurate than *lin*SSTA. Interestingly, we find that both approaches predict the 95% yield point well. This partially explains why *lin*SSTA algorithm is still useful for timing analysis, provided the variations are indeed Gaussian. The PDF comparison of the three approaches is shown in Fig. 5. We see that our n^2SSTA predicts the PDF almost the same as Monte Carlo simulation, while the PDF from *lin*SSTA deviates from that of Monte Carlo simulation.

Bench mark	Monte Carlo		n^2SSTA		*lin*SSTA	
	σ/μ %	95%	σ/μ %	95%	σ/μ %	95%
s27	15.9	1.50	14.9	1.48	13.9	1.47
s386	15.7	1.50	14.9	1.48	14.1	1.46
s444	15.7	1.49	14.9	1.47	14.2	1.46
s832	15.7	1.49	14.8	1.46	14.1	1.45
s1494	16.1	1.50	15.5	1.47	14.4	1.46
s5378	15.8	1.48	14.6	1.46	14.0	1.46
Avg Error	-	-	5.5%	1.61%	10.9%	1.88%

Table 5: Results for Gaussian variation sources.

5. CONCLUSIONS

A novel SSTA technique n^2SSTA has been presented to handle both nonlinear delay dependency and non-Gaussian variation sources simultaneously. We have shown that all statistical atomic operations (such as max and add) can be performed efficiently via either closed-form formulas or one-dimensional lookup table. It has been proved that the complexity of n^2SSTA is linear in both variation sources and circuit sizes. Compared to Monte Carlo simulation for non-Gaussian variations and nonlinear delay models, our approach predicts all timing characteristics of circuit delay with less than 2% error. In the future, we will extend our work to consider more general delay models, such as non-polynomial delays and/or dependency on variations' cross terms.

[2] Note that n^2SSTA has a larger error for Gaussian variation sources in Table 5 than for uniform or triangle variation sources in Table 4. This is because n^2SSTA needs to use bigger bounds defined in (10) for Gaussian variations than for uniform or triangle variations.

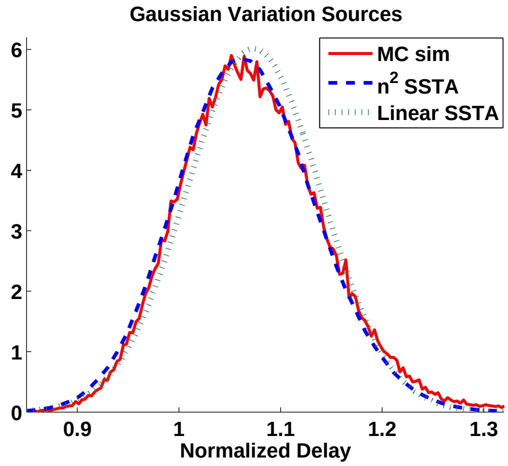

Figure 5: PDF comparison for s5378 with Gaussian variations and linear delay

6. REFERENCES

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

[2] C. Visweswariah, K. Ravindran, K. Kalafala, S. Walker, and S. Narayan, "First-order incremental block-based statistical timing analysis," in *DAC 04*, June 2004.

[3] L. Zhang, W. Chen, Y. Hu, J. A. Gubner, and C. C.-P. Chen, "Correlation-preserved non-gaussian statistical timing analysis with quadratic timing model," in *DAC*, pp. 83 – 88, June 2005.

[4] Y. Zhan, A. J. Strojwas, X. Li, and L. T. Pileggi, "Correlation-aware statistical timing analysis with non-gaussian delay distribution," in *DAC*, pp. 77–82, June 2005. Anaheim, CA.

[5] V. Khandelwal and A. Srivastava, "A general framework for accurate statistical timing analysis considering correlations," in *DAC*, pp. 89 – 94, June 2005.

[6] J. Singh and S. Sapatnekar, "Statistical timing analysis with correlated non-gaussian parameters using independent componenet analysis," in *ACM/IEEE International Workshop on Timing Issues*, pp. 143–148, Feb. 2006.

[7] H. Chang, V. Zolotov, C. Visweswariah, and S. Narayan, "Parameterized block-based statistical timing analysis with non-Gaussian and nonlinear parameters," in *DAC*, pp. 71–76, June 2005. Anaheim, CA.

[8] X. Li, J. Le, and P. Pileggi, "Asymptotic probability extraction for non-normal distributions of circuit performance," in *ICCAD*, Nov 2004.

[9] L. Zhang, W. Chen, Y. Hu, J. A. Gubner, and C. C.-P. Chen, "Statistical timing analysis with extended pseudo-canonical timing model," in *Proc. Design Automation and Test in Europe*, pp. 952 – 957, March 2005.

[10] C. Lin and H. Zhou, "Optimal wire retiming without binary search," *TCAD*, vol. 25, pp. 1577–1588, Sept. 2006.

[11] D. Sinha, N. V. Shenoy, and H. Zhou, "Statistical timing yield optimization by gate sizing," *TCAD*, vol. 25, pp. 1140–1146, Oct. 2006.