

Non-Gaussian Statistical Timing Analysis Using Second-Order Polynomial Fitting

Lerong Cheng
EE Dept., Univ. of California
Los Angeles, CA 90095
lerong@ee.ucla.edu

Jinjun Xiong *
IBM Research Center
Yorktown Heights, NY 10598
jinjun@us.ibm.com

Lei He
EE Dept., Univ. of California
Los Angeles, CA 90095
lhe@ee.ucla.edu †

ABSTRACT

In the nanometer manufacturing region, process variation causes significant uncertainty for circuit performance verification. Statistical static timing analysis (SSTA) is thus developed to estimate timing distribution under process variation. However, most of the existing SSTA techniques have difficulty in handling the non-Gaussian variation distribution and non-linear dependency of delay on variation sources. To solve such a problem, in this paper, we first propose a new method to approximate the max operation of two non-Gaussian random variables through second-order polynomial fitting. We then present new non-Gaussian SSTA algorithms under two types of variational delay models: quadratic model and semi-quadratic model (i.e., quadratic model without crossing terms). All atomic operations (such as max and sum) of our algorithms are performed by closed-form formulas, hence they scale well for large designs. Experimental results show that compared to the Monte-Carlo simulation, our approach predicts the mean, standard deviation, and skewness within 1%, 1%, and 5% error, respectively. Our approach is more accurate and also 20x faster than the most recent method for non-Gaussian and nonlinear SSTA.

1. INTRODUCTION

With the CMOS technology scaling down to the nanometer region, process variation becomes a major limiting factor for integrated circuit design. These variations introduce significant uncertainty for both circuit performance and leakage power. It has been shown in [1] that even for the 180nm technology, process variation can lead to 1.3X variation in frequency and 20X variation in leakage power. Such impact will become even larger for the future technology generations. Statistical static timing analysis (SSTA) is developed for full chip timing analysis under process variation. By performing SSTA, designers can obtain the timing distribution and its sensitivity to various process parameters.

There have been two types of SSTA techniques studied in the literature: path-based [2, 3] and block-based [4, 5, 6, 7, 8, 9, 10]. Because the number of paths is exponential with respect to the circuit size, the path-based SSTA is in general not scalable to large circuits. In order to solve such a problem, the block-based SSTA was proposed. The goal of block-based SSTA is to parameterize timing characteristics

of the timing graph as a function of the underlying sources of process parameters which are modeled as random variables.

The early SSTA methods [4] modeled the gate delay as linear functions of variation sources and assumed all the variation sources are mutually independent Gaussian random variables. Based on such assumption, [4] presented closed-form formulas for all atomic operations (max and sum), hence such method is efficient. However, when the amount of variation becomes larger, the linear delay model is no longer accurate [11]. In order to capture the non-linear dependence of delay on the variation sources, a higher-order delay model is thus needed [5, 6].

As more complicated or large-scale variation sources are taken into account, the assumption of Gaussian variation sources is no longer valid. For example, the via resistance has an asymmetric distribution [8], while dopant concentration is more suitably modeled as a Poisson distribution [7] than Gaussian. Some of the most recent works on SSTA [7, 8, 9, 10] took non-Gaussian variation sources into account. For example, [7] applied independent component analysis to de-correlate the non-Gaussian random variables, but it was still based on a linear delay model. [8] computed the tightness probability of the max operation through multi-dimensional numerical integration. Recently, [10] approximated the probability density function (PDF) of max results as Fourier series, but it lacks the capability to handle crossing term effects on timing.

In this paper, we introduce a time efficient non-linear SSTA for arbitrary non-Gaussian variation sources. The major contribution of this work is two-fold. (1) We propose a new method to approximate the max of two non-Gaussian random variables as a second-order polynomial function based on least-square-error curve fitting. Experimental results shows that such approximation is much more accurate than the linear approximation based on tightness probability [4, 5, 8]. (2) Based on the new approximation of the max operation, we present our SSTA technique for two different delay models: *quadratic delay model* and *semi-quadratic model* (i.e., quadratic delay model without crossing terms). In our method, only the first few moments are required for different distributions. All atomic operations in our approach are performed by closed-form formulas, so they are very time efficient. For the semi-quadratic delay model, the computational complexity of our method is linear in both the number of variation sources and circuit size. For the quadratic delay model, the computational complexity is cubic (third-order) to the number of variation sources and linear to the circuit size. Experimental results show

*Dr. Xiong's work was finished while he was with UCLA.

†This paper was partially supported by NSF CAREER 0093273/0401682 and UC MICRO program sponsored by Intel.

that compared to the Monte-Carlo simulation, our approach predicts the mean, standard deviation, and skewness within 1%, 1%, and 5% error, respectively. Our approach is more accurate and also 20x faster than the most recent method for non-Gaussian and nonlinear SSTA [10].

The rest of the paper is organized as follows: Section 2 introduces the approximation of the max operation using second-order polynomial fitting. With the approximation of max, Section 3 presents a novel SSTA algorithm for quadratic delay model with non-Gaussian variation sources. We further apply this technique to handle semi-quadratic delay model in Section 4. Experimental results are presented in Section 5, with conclusion in Section 6.

2. SECOND-ORDER POLYNOMIAL FITTING OF MAX OPERATION

2.1 Review and Preliminary

According to [4], given two random variables, A and B , the tightness probability is defined as the probability of A greater than B , i.e., $T_A = P\{A > B\} = P\{A - B > 0\}$. Then the max operation is approximated as:

$$\max(A, B) \approx T_A \cdot A + (1 - T_A) \cdot B + c, \quad (1)$$

where c is a term used to match the mean and variance of $\max(A, B)$. Because (1) can be further written as $\max(A, B) = \max(A - B, 0) + B$, we arrive at:

$$\max(A - B, 0) \approx T_A \cdot (A - B) + c. \quad (2)$$

According to (2), we can see that the max operation in [4] is in fact approximated by a linear function subject to certain constraints (such as matching the exact mean and variance). When both A and B are Gaussian, such a linear approximation is reasonably accurate, and the coefficients can be computed easily as both T_A and $E[\max(A, B)]$ can be obtained by closed-form formulas. However, when A and B are non-Gaussian random variables, the tightness probability T_A and $E[\max(A, B)]$ are hard to obtain. For example, T_A in [8] has to be computed via expensive multi-dimensional numerical integration, thus preventing its scalability to large designs. Moreover, because the max operation is an inherently non-linear function, linear approximation would become less and less accurate, particularly when the amount of variation increases and the number of non-Gaussian variation sources increases. To overcome these difficulties, we develop a more efficient and accurate approximation method in the next section.

2.2 New Fitting Method for Max Operation

In this section, we introduce a new fitting method to approximate the max operation. Instead of using the linear function, we propose to use a second-order polynomial function to approximate the max operation, i.e.,

$$\max(V, 0) \approx h(V, \Theta) = \theta_2 V^2 + \theta_1 V + \theta_0, \quad (3)$$

where $\Theta = (\theta_0, \theta_1, \theta_2)^T$ are three coefficients of the second-order polynomial $h(v, \Theta)$. The problem thus becomes how to obtain the fitting parameters of Θ . Different from the linear fitting method through tightness probability, we compute Θ by *matching the mean of the max operation while minimizing the square error between $h(V, P)$ and $\max(V, 0)$ within the $\pm 3\sigma$ range of V* . Mathematically, this problem can be formulated as the following optimization problem:

$$\Theta = \arg \min_{E[h(V, \Theta)] = \mu_m} \int_{\mu_v - 3\sigma_v}^{\mu_v + 3\sigma_v} (h(v, \Theta) - \max(v, 0))^2 dv \quad (4)$$

where μ_v and σ_v^2 are the mean and variance of the random variable V , respectively; while μ_m and $E[h(V, \Theta)]$ are the exact and approximated mean of $\max(V, 0)$, respectively. In other words,

$$\mu_m = E[\max(V, 0)], \quad E[h(V, \Theta)] = \theta_2(\sigma_v^2 + \mu_v^2) + \theta_1\mu_v + \theta_0. \quad (5)$$

In order to solve the problem, we first need to compute μ_m . When V is a non-Gaussian random variable, exact computation of μ_m is difficult in general. Therefore, we propose to use the following two-step procedure to approximately compute μ_m . In the first step, we approximate the non-Gaussian random variable V as a quadratic function of a standard Gaussian random variable W similar to [12], i.e.,

$$V \approx g(W) = c_2 \cdot W^2 + c_1 \cdot W + c_0. \quad (6)$$

The coefficients c_2 , c_1 , and c_0 can be obtained by matching $g(W)$ and V 's mean μ_v , variance σ_v^2 , and skewness γ_v simultaneously, i.e.,

$$\begin{aligned} E[c_2 \cdot W^2 + c_1 \cdot W + c_0] &= \mu_v \\ E[(c_2 \cdot W^2 + c_1 \cdot W + c_0 - \mu_v)^2] &= \sigma_v^2 \\ E[(c_2 \cdot W^2 + c_1 \cdot W + c_0 - \mu_v)^3] &= \gamma_v \cdot \sigma_v^3 \end{aligned} \quad (7)$$

As shown in [12], the above equations can be solved via closed-form formulas efficiently.

After obtaining the coefficients c_2 , c_1 , and c_0 by solving (7), we then approximate the exact mean of $\max(V, 0)$ by the exact mean of $\max(g(W), 0)$, i.e.,

$$\mu_m \approx E[\max(g(W), 0)]$$

Without details, we can further show that

$$E[\max(g(W), 0)] = \begin{cases} (c_2 + c_0)(1 + \Phi(t_1) - \Phi(t_2)) + \\ (c_1 + t_1)(\phi(t_2) - \phi(t_1)) & c_2 > 0 \\ (c_2 + c_0)(\Phi(t_1) - \Phi(t_2)) + \\ (c_1 + t_1)(\phi(t_2) - \phi(t_1)) & c_2 < 0 \\ c_0 \cdot \Phi(c_0/c_1) + c_1 \cdot \phi(c_0/c_1) & c_2 = 0 \wedge c_1 > 0 \\ c_0 \cdot (1 - \Phi(c_0/c_1)) - c_1 \cdot \phi(c_0/c_1) & c_2 = 0 \wedge c_1 < 0 \end{cases} \quad (8)$$

where

$$t_1 = (-c_1 - \sqrt{c_1^2 - 4c_2c_0})/2c_2, \quad t_2 = (-c_1 + \sqrt{c_1^2 - 4c_2c_0})/2c_2$$

with $\Phi(\cdot)$ and $\phi(\cdot)$ as the cumulative density function (CDF) and PDF of the standard normal distribution, respectively. According to (8), we can compute μ_m easily through analytical formulas.

After obtaining μ_m , we need to find Θ in (3) by solving the constrained optimization problem of (4). In the following, we show that (4) can be solved analytically as well. We first write the constraint in (4) as follows:

$$\theta_0 = \mu_m - \theta_2(\mu_v^2 + \sigma_v^2) - \theta_1\mu_v. \quad (9)$$

By replacing θ_0 in (4) by (9), the square error in (4) can be written as:

$$\begin{aligned} &\int_{l_1}^0 (\theta_2(v^2 - \mu_v^2 - \sigma_v^2) + \theta_1(v - \mu_v) + \mu_m)^2 dv + \\ &\int_0^{l_2} (\theta_2(v^2 - \mu_v^2 - \sigma_v^2) + \theta_1(v - \mu_v) + \mu_m - v)^2 dv, \end{aligned} \quad (10)$$

where $l_1 = \mu_v - 3\sigma_v$ and $l_2 = \mu_v + 3\sigma_v$. By expanding the square and integral, we can transform the constrained optimization of (4) to the following unconstrained optimization problem, which is a quadratic form of $\Theta' = (\theta_1, \theta_2)^T$:

$$\Theta' = \arg \min \Theta'^T S \Theta' + Q \Theta' + t, \quad (11)$$

where $S = (s_{ij})$ is a 2×2 matrix, $Q = (q_i)$ is a 1×2 vector, and t is a constant. The parameters of S , Q , and t can be

computed as:

$$\begin{aligned}
s_{11} &= (l_2^3 - l_1^3)/3 + (l_2 - l_1)\mu_v^2 - (l_2^2 - l_1^2)\mu_v \\
s_{22} &= (l_2^5 - l_1^5)/5 + (l_2^2 - l_1^2)(\mu_v^2 + \sigma_v^2)^2 - 2(l_2^3 - l_1^3)(\mu_v^2 + \sigma_v^2)/3 \\
s_{12} = s_{21} &= (l_2^4 - l_1^4)/4 + (l_2 - l_1)\mu_v(\mu_v^2 + \sigma_v^2) + \\
&\quad (l_2^2 - l_1^2)\mu_v/3 - (l_2^2 - l_1^2)(\mu_v^2 + \sigma_v^2)/2 \\
q_1 &= l_2^2\mu_v + (l_2^2 - l_1^2)\mu_m - 2l_2^3/3 - 2(l_2 - l_1)\mu_v\mu_m \\
q_2 &= l_2^2(\mu_v^2 + \sigma_v^2) + 2(l_2^3 - l_1^3)\mu_m/3 - 2l_2^3/3 - 2(l_2 - l_1)(\mu_v^2 + \sigma_v^2)\mu_v \\
t &= l_2^3/3 + (l_2 - l_1)\mu_m^2 - l_2^2\mu_m
\end{aligned} \tag{12}$$

Because the square error is always positive no matter what value the Θ' is, S is a positive definite matrix. Therefore, (11) is to minimize a second-order convex function of Θ' without constraints. Then the optimum of Θ' can be obtained by setting the derivation of (11) to zero, resulting a 2×2 system of linear equations. Such a system of linear equations can be solved efficiently to obtain Θ' . With $\Theta' = (\theta_1, \theta_2)^T$, we can compute θ_0 from (9).

From the above discussion, we see that for a random variable V with any distribution, if we know its mean μ_v , variance σ_v^2 , and skewness γ_v , we can obtain the fitting parameters Θ for $\max(V, 0)$ through closed-form formulas.

To show the accuracy of our second-order fitting approach to the max approximation, we compare the results obtain from our approach, the linear fitting approach, and the exact (or Monte Carlo) approach. For example, when $V \sim N(0.7, 1)$, results from these three approaches in computing $\max(V, 0)$ are shown in Fig. 1(a), where the x-axis is V , and y-axis is the results of $\max(V, 0)$. The corresponding PDFs of the three approaches are shown in Fig. 1(b). From the figures, we see that our proposed second-order fitting method is more accurate than the linear fitting method. In particular, our second-order fitting method predicts the impulse of the exact PDF well as shown in Fig. 1(b). In contrast, the linear fitting method can only give a smooth PDF, which is very different from the exact max result.

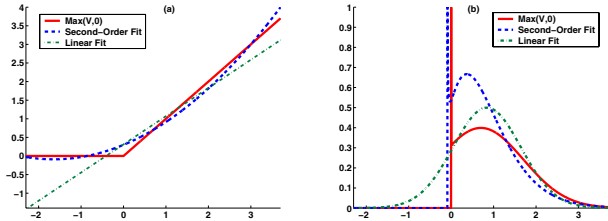


Figure 1: (a) Comparison of exact computation, linear fitting, and second-order fitting for $\max(V, 0)$. (b) PDF comparison

3. QUADRATIC SSTA

3.1 Quadratic Delay Model

In Section 2, we introduce the second-order fitting of max operation. Here we will apply such fitting in SSTA. In this section, we assume that the delay is a quadratic function of variation sources:

$$D = d_0 + AX + X^T BX + rR \tag{13}$$

where d_0 is the nominal delay, $X = (X_1, X_2, \dots, X_n)^T$ are n variation sources, $A = (a_1, a_2, \dots, a_n)$ are the linear delay sensitivity coefficients of the variation sources, $B = (b_{ij})$ are the second-order sensitivity coefficients, which is an $n \times n$

matrix, R is the random variation, and r is the delay sensitivity coefficient of the random variation. Without loss of generality, it is assumed that all X_i and R are mutually independent with mean=0 and variance=1. We allow the variation X_i s to be any arbitrary distribution, and the random variation R is modeled as a Gaussian random variable. In the rest of this paper, we use $m_i(k)$ and $m_r(k)$ to represent the k th moment for X_i and the k th moment for R , respectively.

To compute the arrival time in a block-based SSTA framework, two atomic operations, max and sum, are needed. That is, given D_1 and D_2

$$\begin{aligned}
D_1 &= d_{01} + A_1 X + X^T B_1 X + r_1 R_1, \\
D_2 &= d_{02} + A_2 X + X^T B_2 X + r_2 R_2,
\end{aligned} \tag{14}$$

we want to compute

$$\begin{aligned}
D_m &= \max(D_1, D_2) = d_{m0} + A_m X + X^T B_m X + r_m R_m, \\
D_s &= D_1 + D_2 = d_{s0} + A_s X + X^T B_s X + r_s R_s.
\end{aligned} \tag{15}$$

The sum operation is straightforward, as the coefficients of D_s can be computed by adding the correspondent coefficients of D_1 and D_2 , i.e.,

$$d_{s0} = d_{01} + d_{02} \quad A_s = A_1 + A_2 \quad B_s = B_1 + B_2 \quad r_s = \sqrt{r_1^2 + r_2^2}. \tag{16}$$

3.2 Max Operation

The max operation is the most difficult operation for block-based SSTA. Based on the second-order polynomial fitting method as discussed in Section 2.2, we propose a novel technique to compute the max of two random variables. The overall flow of the max operation is illustrated in Figure 2. Considering

$$D_m = \max(D_1, D_2) = \max(D_1 - D_2, 0) + D_2. \tag{17}$$

Denote $D_p = D_1 - D_2$, and without loss of generality, we assume $E[D_p] > 0$. We first compute the mean and variance of D_p . Because in (4) we try to minimize the mean square error within the ± 3 sigma range, when $\mu_{D_p} > 3\sigma_{D_p}$, we have $D_m = D_1$. Otherwise, we compute the joint moments between D_p and X_i s and the skewness of D_p . Knowing the mean, variance, and skewness of D_p , we apply the method as shown in Section 2.2 to find the fitting coefficients $\Theta = (\theta_0, \theta_1, \theta_2)$ for $\max(D_p, 0)$. Finally, we use the moment matching method to reconstruct the quadratic form of D_m .

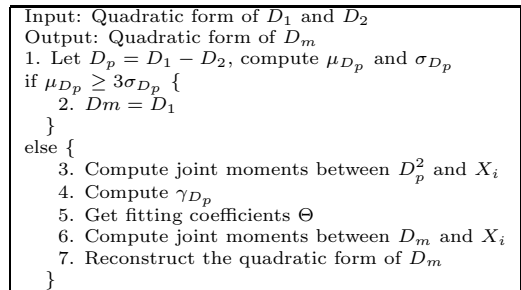


Figure 2: Algorithm for computing $\max(D_1, D_2)$.

3.2.1 Mean and Variance

In order to compute the mean and variance, we first obtain the quadratic form of D_p as follows:

$$\begin{aligned}
D_p &= D_1 - D_2 = d_{p0} + A_p \cdot X + X^T B_p X + r_1 \cdot R_1 - r_2 \cdot R_2 \\
d_{p0} &= d_{01} - d_{02} \quad A_p = A_1 - A_2 \quad B_p = B_1 - B_2
\end{aligned} \tag{18}$$

Because X_i s, R_1 , and R_2 are mutually independent random variables with mean 0 and variance 1, the mean and variance of D_p can be computed as:

$$\begin{aligned}\mu_{D_p} &= E[D_p] = d_{p0} + \sum_{i=1}^n b_{pii} \\ \sigma_{D_p}^2 &= E[(D_p - \mu_{D_p})^2] \\ &= r_1^2 + r_2^2 + \sum_{i=1}^n (a_{pi}^2 + b_{pii}m_{i,4} + 2a_{pi}b_{pii}m_{i,3}) + \\ &\quad \sum_{i < j} 2(b_{pij}^2 + b_{pii}b_{pjj}) - (\sum_{i=1}^n b_{pii})^2\end{aligned}\quad (19)$$

3.2.2 Joint Moments and Skewness

From the definition of D_p as shown in (18), the joint moments between D_p^2 and X_i , X_i^2 , and R can be computed as:

$$\begin{aligned}E[X_i D_p^2] &= 2d_{p0}a_{pi} + (a_{pi}^2 + 2d_{p0}b_{pii})m_{i,3} + 2a_{pi}b_{pii}m_{i,4} + \\ &\quad b_{pii}^2 m_{i,5} + 2 \cdot \sum_{j \neq i} (a_{pi}b_{pjj} + 2a_{pj}b_{pij} + \\ &\quad (b_{pii}b_{pjj} + 2b_{pij}^2)m_{i,3} + 2b_{pij}b_{pjj}m_{j,3})\end{aligned}\quad (20)$$

$$E[R_1 D_p^2] = r_1^2 m_{r,3} + 2r_1 \mu_{D_p} \quad E[R_2 D_p^2] = r_2^2 m_{r,3} - 2r_2 \mu_{D_p} \quad (21)$$

$$\begin{aligned}E[X_i^2 D_p^2] &= d_{p0}^2 + r_1^2 + r_2^2 + 2d_{p0}a_{pi}m_{i,3} + 2a_{pi}b_{pii}m_{i,5} + \\ &\quad (a_{pi}^2 + 2d_{p0}b_{pii}) \cdot m_{i,4} + b_{pii}^2 m_{i,6} + \\ &\quad \sum_{j \neq i} (a_{pj}^2 + 2d_{p0}b_{pjj} + 2(a_{pi}b_{pjj} + 2a_{pj}b_{pij})m_{i,3} + \\ &\quad 2a_{pj}b_{pjj}m_{j,3} + 2(b_{pii}b_{pjj} + 4b_{pij}^2) \cdot m_{i,4} + \\ &\quad b_{pjj}^2 m_{j,4} + 4b_{pjj}b_{pij}m_{i,3}m_{j,3}) + \\ &\quad 2 \cdot \sum_{j, k \neq i, j < k} (b_{pjj}b_{pkk} + 2b_{pj}^2)\end{aligned}\quad (22)$$

$$\begin{aligned}E[X_i X_j D_p^2] &= 2a_{pi}a_{pj} + 4d_{p0}b_{pij} + 2 \cdot (2a_{pi}b_{pij} + a_{pj}b_{pii})m_{i,3} + \\ &\quad 4b_{pii}b_{pij}m_{i,4} + 2 \cdot (2a_{pj}b_{pij} + a_{pi}b_{pjj})m_{j,3} + \\ &\quad 4b_{pjj}b_{pij}m_{j,4} + 2 \cdot (2b_{pij}^2 + b_{pii}b_{pjj})m_{i,3}m_{j,3} + \\ &\quad 4 \cdot \sum_{k \neq i, j} b_{pij}b_{pkk}\end{aligned}\quad (23)$$

With the joint moments computed above, the raw moments, central moments, and skewness of D_p is computed as:

$$\begin{aligned}E[D_p^2] &= \mu_{D_p}^2 + \sigma_{D_p}^2 \\ E[D_p^3] &= E[D_p \cdot D_p^2] \\ &= E[(d_{p0} + A_p X + X^T B_p X + r_1 R_1 - r_2 R_2) D_p^2] \\ &= d_{p0} \cdot E[D_p^2] + r_1 \cdot E[R_1 D_p^2] - r_2 \cdot E[R_2 D_p^2] + \\ &\quad \sum_{i=1}^n (a_{pi} \cdot E[X_i D_p^2] + b_{pii} \cdot E[X_i^2 D_p^2]) + \\ &\quad \sum_{0 \leq i < j \leq n} b_{pij} E[X_i X_j \cdot D_p^2] \\ m_{D_p}(3) &= E[(D - \mu_D)^3] = E[D_p^3] - 3\mu_{D_p} \cdot E[D_p^2] + 2\mu_{D_p}^3 \\ \gamma_{D_p} &= m_{D_p}(3) / \sigma_{D_p}^3\end{aligned}\quad (24)$$

3.2.3 Reconstruction of Quadratic Form

Knowing μ_{D_p} , σ_{D_p} , and γ_{D_p} , we apply the second-order fitting method to compute the fitting parameters $\Theta = (\theta_0, \theta_1, \theta_2)$ for $\max(D_p, 0)$. Then D_m can be represented as:

$$\begin{aligned}D_m &= \max(D_1, D_2) = \max(D_p, 0) + D_2 \\ &\approx \theta_2 \cdot D_p^2 + \theta_1 \cdot D_p + \theta_0 + D_2 \\ &= \theta_2 \cdot D_p^2 + D_q + \theta_0 \\ D_q &= \theta_1 \cdot D_p + D_2\end{aligned}\quad (25)$$

The above equation gives the closed-form formula of D_m . However, because D_p and D_q are in quadratic form as (13), the representation of D_m in (25) is a 4th order polynomial of X_i s. In order to reconstruct the quadratic form for D_m , we first compute the the mean of D_m , and joint moments between D_m and variation sources:

$$\begin{aligned}E[D_m] &= p_2 \cdot E[D_p^2] + E[D_q] + p_0 \\ E[X_i D_m] &= p_2 \cdot E[X_i D_p^2] + E[X_i D_q] \\ E[X_i^2 D_m] &= p_2 \cdot E[X_i^2 D_p^2] + E[X_i^2 D_q] + p_0 \\ E[X_i X_j D_m] &= p_2 \cdot E[X_i X_j D_p^2] + E[X_i X_j D_q] \\ E[R_1 D_m] &= p_2 \cdot E[R_1 D_p^2] + E[R_1 D_q] \\ E[R_2 D_m] &= p_2 \cdot E[R_2 D_p^2] + E[R_2 D_q]\end{aligned}\quad (26)$$

The joint moments between D_p^2 and X_i s are computed in (20), (21), (22), and (23). The joint moments between D_q and variation sources are:

$$\begin{aligned}E[D_q] &= d_{q0} + \sum_{i=1}^n b_{qii} \\ E[X_i D_q] &= a_{qi}m_{i,2} + b_{qii}m_{i,3} \\ E[X_i^2 D_q] &= d_{q0} + a_{qi}m_{i,3} + b_{qii}m_{i,4} + \sum_{j \neq i} b_{qjj} \\ E[X_i X_j D_q] &= 2b_{qij} \\ E[R_1 D_q] &= p_1 \cdot r_1 \\ E[R_2 D_q] &= (1 - p_1) \cdot r_2\end{aligned}\quad (27)$$

Because we want to reconstruct D_m in the quadratic form, as shown in (15), by applying the moment matching technique similar to [6], we have:

$$\begin{aligned}E[D_m] &= \sum_{i=1}^n b_{mii} + d_{m0} \\ E[X_i D_m] &= a_{mi} + b_{mii} \cdot m_{i,3} \\ E[X_i^2 D_m] &= a_{mi} \cdot m_{i,3} + b_{mii} \cdot m_{i,4} + \sum_{j \neq i} b_{mjj} \\ E[X_i X_j D_m] &= 2b_{mij}\end{aligned}\quad (28)$$

With $E[D_m]$, $E[X_i D_m]$, $E[X_i^2 D_m]$, and $E[X_i X_j D_m]$ computed in (27), the sensitivity coefficients $A_m = (a_{mi})$ and $B_m = (b_{mij})$ can be obtained by solving the linear equations above.

Finally, we consider the random term of D_m . Because the random term in D_m comes from the random terms in D_1 and D_2 , we assume $r_m R_m = r_{m1} R_1 + r_{m2} R_2$. Because the random variation sources R_m , R_1 , and R_2 are Gaussian random variables, by applying the moment matching technique similar to (28), we have:

$$r_{m1} = E[R_1 \cdot D_m] \quad r_{m2} = E[R_2 \cdot D_m] \quad r_m = \sqrt{r_{m1}^2 + r_{m2}^2} \quad (29)$$

where $E[R_1 \cdot D_m]$ and $E[R_2 \cdot D_m]$ are computed in (26).

3.3 Complexity of Quadratic SSTA

For each max operation in SSTA based on a quadratic delay model, we need to calculate n^2 joint moments between D_p and $X_i X_j$ s; and for each joint moment, we need to compute the sum of n numbers; hence the computational complexity is $\mathcal{O}\{n^3\}$, where n is the number of variation sources. For the sum operation, we need to compute the sum of two $n \times n$ metric, hence the computational complexity is $\mathcal{O}\{n^2\}$. Because the total number of max and sum operations is linear with respect to circuit sizes N , the total complexity is $\mathcal{O}\{n^3 N\}$.

4. SEMI-QUADRATIC SSTA

4.1 Semi-Quadratic Delay Model

It has been argued that the impact of the crossing terms may be ignored under certain circumstances [6, 9]. In such

cases, we may further speed up our SSTA algorithm without affecting the accuracy too much. In other words, the delay model without crossing terms becomes

$$D = d_0 + AX + BX^2 + rR, \quad (30)$$

where d_0 , A , r , and R are defined in the similar way as the quadratic delay model in (13); $X^2 = (X_1^2, X_2^2, \dots, X_n^2)^T$ are the square of variation sources; and $B = (b_1, b_2, \dots, b_n)$ are the second order sensitivity coefficients for the square terms. We defined such quadratic model without crossing terms as *Semi-Quadratic Delay Model*.

A straightforward way of performing SSTA under the semi-quadratic delay model is to use the same algorithm as discussed for the quadratic delay model by setting the coefficients of all crossing terms as zero. However, by exploiting the special property of this new delay model, we can achieve a more efficient algorithm as shown in the following. As the sum operation is still the same, we will only present the max operation for the semi-quadratic delay model in the rest of this section.

4.2 Max Operation

The overall flow of the max operation of semi-quadratic delay model is similar to that of quadratic delay model as illustrated in Figure 2. The only difference is that we do not need to compute the joint moments between the crossing terms and D_m .

4.2.1 Moments

We defined $D_p = D_1 - D_2$ in a similar way as (18). In order to compute the central moments of D_p , we first rewrite D_p as the following form:

$$D_p = d'_{p0} + \sum_{i=1}^n Y_{pi} + r_1 R_1 - r_2 R_2 \\ d'_{p0} = d_{p0} + \sum_{i=1}^n b_{pi}, \quad Y_{pi} = a_{pi} X_i + b_{pi} X_i^2 - b_{pi} \quad (31)$$

Because X_i s are mutually independent random variables with mean 0 and variance 1, Y_{pi} s are independent random variables with zero mean. Therefore, the first three central moments of D_p are:

$$\mu_{D_p} = d'_{p0} \quad \sigma_{D_p}^2 = r_1^2 + r_2^2 + \sum_{i=1}^n E[Y_{pi}^2] \\ m_{D_p}(3) = E[(D_p - \mu_{D_p})^3] = r_1^3 + r_2^3 + \sum_{i=1}^n E[Y_{pi}^3] \quad (32)$$

where

$$E[Y_{pi}^2] = b_{pi}^2(m_i(4) - 1) + 2a_{pi}b_{pi}m_i(3) + a_{pi}^2 \\ E[Y_{pi}^3] = 3a_{pi}b_{pi}^2(m_i(5) - 2m_i(3)) + 3a_{pi}^2b_{pi}(m_i(4) - 1) + \\ a_{pi}^3m_i(3) + b_{pi}^3(m_i(6) - 3m_i(4) + 2) \quad (33)$$

With the central moments of D_p , the raw moments and skewness can be computed easily.

4.2.2 Reconstruction of Semi-quadratic Form

Similar to the quadratic SSTA, by knowing the mean, variance and skewness of D_p , the fitting coefficients Θ can be obtained. Then the joint moments between X_i s and D_m can be computed in the similar ways as (26). Here the joint moments between X_i s and D_p^2 , D_q are:

$$E[X_i D_p^2] = E[X_i Y_{pi}^2] + 2d'_0 E[X_i Y_{pi}] \\ E[X_i^2 D_p^2] = E[X_i^2 Y_{pi}^2] + 2d'_0 E[X_i^2 Y_{pi}] + E[X_i^2] E[(D_p - Y_{pi})^2] \\ E[X_i D_q] = E[X_i Y_{qi}] \quad E[X_i^2 D_q] = E[X_i^2 Y_{qi}] \quad (34)$$

where $E[Y_{pi}^2]$ are computed in (32), $E[(D_p - Y_{pi})^2] = \sigma_{D_p}^2 - E[Y_{pi}^2]$, and Y_{qi} s are defined in the similar way as Y_{pi} s. The

joint moments between X_i s and Y_{pi} s are:

$$E[X_i^2 Y_{pi}^2] = (a_{pi}^2 - 2b_{pi}^2)m_i(4) + b_{pi}^2(m_i(6) - 1) + \\ 2a_{pi}b_{pi}(m_i(5) - m_i(3)) \\ E[X_i^2 Y_{pi}] = a_{pi}m_i(3) + b_{pi}(m_i(4) - 1) \\ E[X_i Y_{pi}^2] = b_{pi}^2m_i(5) + 2a_{pi}b_{pi}(m_i(4) - 1) + (a_{pi}^2 - 2b_{pi}^2)m_i(3) \\ E[X_i Y_{pi}] = a_{pi} + b_{pi}m_i(3) \quad (35)$$

The joint moments between X_i s and Y_{qi} s can be computed in the same way.

Knowing the joint moments between X_i s and D_m , by applying the moment matching technique, we will have similar equations as (28). And then A_m and B_m can be obtained by solving such equations. Finally, we can compute the random term of D_m in the same way as the quadratic model, as shown in (29).

From the discussion above, it is easy to see that for the semi-quadratic SSTA, the computational complexity for both max and sum operation is $\mathcal{O}\{n\}$, while the total complexity of the algorithm is $\mathcal{O}\{nN\}$, where n is the number of variation sources and N is the circuit size. Compared to the straightforward application of the algorithm under quadratic-delay model, which has total complexity of $\mathcal{O}\{n^3N\}$, the new approach has much lower complexity, and this will be further confirmed in our experiments.

5. EXPERIMENTAL RESULT

We have implemented our SSTA algorithm in C for both the quadratic delay model (*Quad SSTA*) and semi-quadratic delay model (*Semi-Quad SSTA*). We also define three comparison cases: (1) our implementation of the linear SSTA for Gaussian variation sources in [4], which we refer to as *Lin Gau*; (2) our implementation of the non-Gaussian SSTA with Fourier Series in [10], which we refer to as *Fourier SSTA*; (3) 100,000-sample Monte-Carlo simulation (*MC*). We apply all the above methods to the ISCAS89 suite of benchmarks in TSMC 90nm technology.

In our experiment, we consider two types of variation sources L_{eff} and V_{th} . For each type of variation source, inter-die, intra-die spatial, and intra-die random variation are considered. We use the grid-based model in [13] to model the spatial variation. The number of grids (the number of spatial variation sources) is determined by the circuit size and larger circuits have more variation sources. We also assume that the 3σ value of the inter-die, intra-die spatial, and intra-die random variation are 10%, 10%, and 5% of the nominal value, respectively. In our experiment, we assume that L_{eff} has a normal distribution and V_{th} has a Poisson distribution.

Fig. 3 illustrates the PDF comparison for circuit s15850. From the figure, we find that, compared to the Monte-Carlo simulation, the *Quad SSTA* is the most accurate, the *Semi-Quad SSTA* and *Fourier SSTA* have similar accuracy, and *Lin Gau* is the least accurate. Such result is expected, because the *Quad SSTA* captures all the second-order effects; the *Semi-Quad SSTA* and *Fourier SSTA* both use semi-quadratic delay model and capture only partial second order effects; while the *Lin Gau* captures only linear effects and ignores all non-linear effects.

Table 1 compares the run time in second (T), and the error percentage of mean (μ), standard deviation (σ), and skewness (γ). In the table, the error percentage of mean is computed as $100 \times (\mu_{MC} - \mu_{SSTA}) / \sigma_{MC}$, the error per-

bench name	G	N	Quad SSTA				Semi-Quad SSTA				Fourier SSTA				Lin Gau				MC
			μ	σ	γ	T	μ	σ	γ	T	μ	σ	γ	T	μ	σ	γ	T	
s444	119	25	-0.605	-0.809	3.269	0.16	-0.522	-0.955	-22.4	0.13	-0.671	-1.865	-23.3	3.4	-0.565	-1.284	-65.1	0.07	41.76
s832	262	33	-0.362	-0.263	-3.256	0.3	-0.37	-0.344	-23.5	0.14	-0.546	-1.295	-25.2	3.9	-0.546	-1.295	-66.93	0.07	102.28
s1494	588	68	-0.601	-0.798	1.715	1.3	-0.534	-1.374	-16.85	0.15	-0.773	-2.276	-17.2	4.5	-0.266	-1.611	-58.35	0.1	320.35
s9234	2027	99	-0.135	-0.944	2.529	10.9	0.325	-1.873	-12.46	0.31	0.009	-3.112	-13.2	7.5	1.853	-2.518	-68.95	0.19	2803
s15850	3448	135	0.831	-0.552	0.415	39.9	1.66	-0.875	-12.15	0.54	1.398	-2.02	-15.5	12.5	3.573	-1.93	-66.45	0.35	6742
s38417	8709	176	0.775	-0.215	4.598	204	0.928	-0.435	-6.091	1.11	0.691	-1.571	-9.5	22.1	1.568	-1.3	-63.88	0.65	19129
s38584	11448	176	-0.766	-0.587	-3.414	219	-0.384	-1.224	-14.71	1.2	-0.684	-2.383	-18.8	23.7	0.645	-2.282	-65.32	0.7	19423
Ave	-	-	0.467	0.553	3.469	1/107	0.56	0.963	15.37	1/12179	0.548	2.017	64.37	1/568	1.062	1.624	64.47	1/20512	-

Table 1: Percentage of relative error comparison for mean, variance, and skewness, as well as runtime comparison.

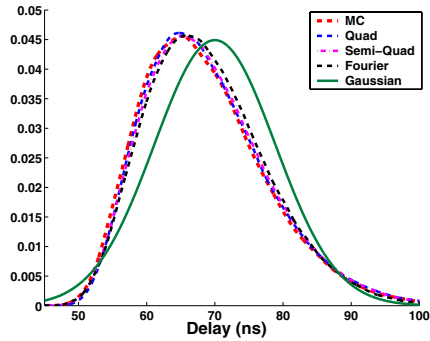


Figure 3: PDF comparison for circuit s15850

centage of standard deviation is computed as $100 \times (\sigma_{MC} - \sigma_{SSTA}) / \sigma_{MC}$, and the error percentage of skewness is computed as $100 \times (\gamma_{MC} - \gamma_{SSTA}) / \gamma_{MC}$. Moreover, the average error in the table is average of the absolute value; and the average runtime is the average runtime ratio between SSTA and Monte-Carlo simulation. For each benchmark, G refers to the number of gates; and N refers to the total number of variation sources. From the table, we see that for the *Quad SSTA* the error of mean and standard deviation is within 1%, and the error of skewness is within 5%. *Semi-Quad SSTA* and *Fourier SSTA* result similar error, but run time of *Semi-Quad SSTA* is 20X faster than *Fourier SSTA*. And all the non-linear SSTA give better results than *Lin Gau*, especially for skewness. This is because the *Lin Gau* ignores all non-linear effects which significantly affect the skewness. Moreover, we also find that *Semi-Quad SSTA* has similar run time as *Lin Gau*, but the run time of *Quad SSTA* is longer especially when the number of variation sources is large. This is because the computational complexity of *Semi-Quad SSTA* and *Lin Gau* is the same, but *Quad SSTA* has higher complexity than the other SSTA methods. But the run time of all the SSTA methods is significantly shorter than the Monte-Carlo simulation (more than $100\times$ speed-up).

6. CONCLUSION AND DISCUSSION

In this paper, we have proposed a new method to approximate the max operation of two non-Gaussian random variables using second-order polynomial fitting. It has been shown that such approximation is more accurate than the approximation using linear fitting through tightness probability. By applying such approximation, we present a new SSTA algorithm for two different delay models, i.e., quadratic and semi-quadratic models. All atomic operations of this algorithm are performed by closed-form formulas, hence they

are very time efficient. The computational complexity of the semi-quadratic delay model is linear to the number of variation sources and that of the quadratic delay model is cubic (third-order) to the number of variation sources. Moreover, the computational complexity is linear to the circuit size for both delay models. Compared to Monte-Carlo simulation for non-Gaussian variation sources and quadratic delay models, the error of mean, standard deviation, and skewness of our approach is within 1%, 1%, and 5%, respectively. Our approach is more accurate and also 20x faster than the most recent method for non-Gaussian and nonlinear SSTA.

7. REFERENCES

- [1] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De, "Parameter variations and impacts on circuits and microarchitecture," in *DAC 03*, Jun 2003.
- [2] M. Orshansky and A. Bandyopadhyay, "Fast statistical timing analysis handling arbitrary delay correlations," in *DAC 04*, June 2004.
- [3] A. Ramalingam, A. K. Singh, S. R. Nassif, G.-J. Nam, M. Orshansky, and D. Z. Pan, "An accurate sparse matrix based framework for statistical static timing analysis," in *ICCAD*, Nov 2006.
- [4] C. Visweswariah, K. Ravindran, K. Kalafala, S. Walker, and S. Narayan, "First-order incremental block-based statistical timing analysis," in *DAC 04*, June 2004.
- [5] 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.
- [6] 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.
- [7] J. Singh and S. Sapatnekar, "Statistical timing analysis with correlated non-gaussian parameters using independent component analysis," in *ACM/IEEE International Workshop on Timing Issues*, pp. 143–148, Feb. 2006.
- [8] 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.
- [9] S. Bhardwaj, P. Ghanta, and S. Vrudhula, "A framework for statistical timing analysis using non-linear delay and slew models," in *ICCAD*, Nov 2006.
- [10] L. Cheng, J. Xiong, and L. He, "Non-linear statistical static timing analysis for non-gaussian variation sources," in *DAC*, Jun 2007.
- [11] X. Li, J. Le, and P. Pileggi, "Asymptotic probability extraction for non-normal distributions of circuit performance," in *ICCAD*, Nov 2004.
- [12] L. Zhang, J. Shao, and C. C.-P. Cheng, "Non-gaussian statistical parameter modeling for ssta with confidence interval analysis," in *Proc. Int. Symp. on Physical Design*, Apr 2006.
- [13] J. Xiong, V. Zolotov, and L. He, "Robust extraction of spatial correlation," *TCAD*, 2007.