Correlation-Aware Statistical Timing Analysis with Non-Gaussian Delay Distributions

Yaping Zhan, Andrzej J. Strojwas, Xin Li, Lawrence T. Pileggi,
David Newmark*, Mahesh Sharma*
Department of ECE, Carnegie Mellon University, Pittsburgh, PA,
Advanced Micro Devices Inc., Austin, TX*

ABSTRACT

Process variations have a growing impact on circuit performance for today's integrated circuit (IC) technologies. The Non-Gaussian delay distributions as well as the correlations among delays make statistical timing analysis more challenging than ever. In this paper, we present an efficient block-based statistical timing analysis approach with linear complexity with respect to the circuit size, which can accurately predict Non-Gaussian delay distributions from realistic nonlinear gate and interconnect delay models. This approach accounts for all correlations, from manufacturing process dependence, to re-convergent circuit paths to produce more accurate statistical timing predictions. With this approach, circuit designers can have increased confidence in the variation estimates, at a low additional computation cost.

Categories and Subject Descriptors

B.8.2 [Hardware]: Performance and reliability – Performance Analysis and Design Aids

General Terms: Algorithms, verification

Keywords: Statistical timing, process variation

1. INTRODUCTION

With the increasing complexity of VLSI designs and tighter timing constraints, timing verification has become a more challenging and important task. Timing information can be used for design optimization as well as yield improvement in manufacturing. For today's nanometer process technologies, circuit delays are highly dependent on manufacturing process variations, especially the intra-die variations, of both gates and interconnects. Due to the correlations among component (gates and interconnect) delays, corner case analysis using traditional Static Timing Analysis (STA) tools is very pessimistic and no longer capable of finding the circuit delay in variational environments.

As an improvement, Statistical Static Timing Analysis algorithms (SSTA) have been proposed by several researchers. Instead ofpropagating fixed delay values through gates and interconnect, SSTA propagates delay distributions characterized by delay Probability Density Functions (PDFs). Path-based SSTA algorithms were discussed in [1], [2]. Due to the large number of long paths in realistic commercial circuits, the top K longest paths must be selected before a path-based algorithm is applied. However, this is a very challenging task when both inter-die and intra-die variations are present, since variations can change the set of critical paths. So far, no effective approach has been published to solve this problem. Moreover, path-based algorithms are not incremental. When the circuit is optimized, the algorithms have to be re-run, even when only a small number of gates are re-sized.

At the same time, block-based statistical STA algorithms were proposed [3]-[7]. Unlike path-based algorithms, block-based algorithms walk through the circuit by a breadth-first search. Delay PDFs are propagated level by level from the source node to the sink node of a timing graph. Since there is no need to find the top K paths, and only two atomic operations, *sum* and *max*, are required, block-based algorithms are favored and widely accepted for their efficiency. For illustration purposes, this paper only discusses long paths problems. Short paths problems can be treated similarly by using the *min* operation in place of the *max* operation.

There are currently two categories of approaches for propagating delay PDFs in block-based SSTA algorithms. Each category is based upon its specific assumption. The first category assumes statistical independence among delays of different gates and interconnects. These algorithms are able to process any form of delay probability distributions [3], [4]. As a result, nonlinear delay models can be handled with this set of techniques. However, provided that an IC is affected by both inter-die and systematic intra-die variations, the independence assumption for different gate and interconnect delays is unrealistic, and spatial correlation must be included to get accurate delay predictions.

On the other hand, the second category assumes Gaussian distributions of all delays to take into account the delay correlations, based on the "convenient" properties of Gaussian random variables [5]-[7]. However, in order to maintain the Gaussian assumption, linear delay models are required over process parameters for all gates and interconnect delays throughout the circuit. In other words, for each gate or interconnect delay, first order Taylor Expansion has to be used to represent its delay function in terms of process parameters. Moreover, Gaussian distributions must be assumed for all signal arrival times. Therefore, this set of techniques loses the controllability over nonlinear delays, which can be caused by many sources from the nonlinear delay models due to large-scale manufacturing process variations, to the nonlinear operation *max* during block-based delay propagation. Due to these nonlinearities, the linear delay models with Gaussian distributions are often not accurate enough, so that higher order delay models must be used. The importance of using nonlinear delay models will be discussed in detail in Section 2.

DAC 2005, June 13–17, 2005, Anaheim, California, USA.
Copyright 2005 ACM 1-59593-058-2/05/0006…$5.00.

To account for both Non-Gaussian delay distributions and correlations, we hereby propose a novel block-based statistical timing analysis approach. In this approach, all gate/interconnect delays and signal arrival times are represented in quadratic form over a base set of variational process parameters. From experiments, we verify that nonlinear delays can be accurately approximated by quadratic models. A breadth-first search algorithm is then used to get delay expressions for all nodes of the circuit, which guarantees that the algorithm has a linear complexity. The parameter base can be derived from the Principal Component Analysis (PCA) technique [8].

The organization of the rest of the paper is as follows: In Section 2, we focus on nonlinear delay sources in block-based SSTA, which is the main motivation for us to propose this nonlinear approach. Section 3 discusses how the two atomic operations: *sum* and *max,* are performed under quadratic delay models. We show our algorithm in Section 4. Experimental results and algorithm complexity are discussed in Section 5. We give our conclusions and future work in Section 6.

2. NONLINEARITY IN SSTA

An increasing magnitude of nonlinear properties is observed in today's industrial circuits. For example, industry speed binning results, for circuits such as microprocessors, demonstrate very significant nonlinearities in the circuit delay. This is due to nonlinear gate/interconnect delay dependence on process variations. Delay functions over a certain process parameter set are often nonlinear. When the process variations are very small, first order Taylor Expansion is an accurate enough approximation. However, with the growing variations due to smaller feature sizes, first order approximation is no longer precise. Second order terms can't be ignored any more. According to the current technology trends, more than ±35% variations on the gate length are cited for 90 nanometer processes, and they are getting even larger for 65 nanometer processes [9]. Hence, the linear models and the Gaussian assumption may cause considerable errors that degrade the accuracy of those SSTA algorithms.

Moreover, models for Chemical Mechanical Polishing (CMP) and Critical Dimension (CD) variations are non-linear as well. In Figure 1, we show the delay PDF of an 800um*0.8um interconnect in 0.18u technology. In this experiment, a 30% variation is applied on the metal thickness, which can be caused by copper CMP dishing effect.

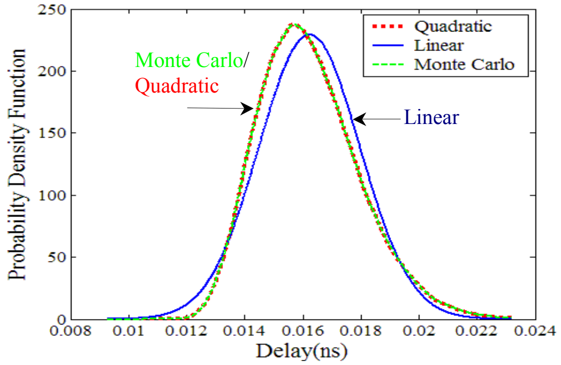

Figure 1 Interconnect Delay

A 15% modeling error is observed in the linear delay model versus Monte Carlo SPICE simulation result, while the result of quadratic delay model is overlapping the Monte Carlo result. For any Gaussian random variable, the skew (3rd order moment) is always zero. Non-zero delay skews can't be represented in linear delay models. But under nonlinear delay models, non-zero skews can be represented by the quadratic terms. In this example, the skew is 0.58, which implies the inevitable error introduced by the Gaussian approximation. The modeling error will become even larger as process variations increase or more variations sources, such as driving strength, are considered. Similarly, nonlinearity can be observed in gate delays experiments as well.

In addition to large-scale process variations, other sources can contribute to the need of use of the nonlinear delay models. The core of block-based SSTA approaches finding the probability distribution of delay D = *max*(D_1, D_2, …, D_n), where D_i is the delay of the *i*-th partial path to the current node in the statistical timing graph of a circuit. The *max* operation is performed at each node of a circuit until the sink node is reached. Because *max* is a nonlinear operation, it will generate Non-Gaussian delay distribution for D, even if all D_i's are of Gaussian distributions. Moreover, the partial path delays D_i's are often nonlinear as well, due to high order delay models caused by large-scale process variations.

Next consider the nonlinearity source that comes from the atomic operations. In propagating delay PDFs through gate/interconnect, *sum* operation is used to handle single input component, like inverters, buffers, and interconnects, while, *max* operation is used to handle multi-input gates, like NAND gates, NOR gates and etc.. *Sum* operation is a linear operation. Provided that both input signal arrival time and component delay are of Gaussian distributions, the output signal arrival time is Gaussian as well. However, the *max* operation, *max*(A_1,A_2) over two random variables is non-linear. Let's consider one hypothetical case when the two input operands, A_1 and A_2, have similar mean values, and different variances. Under this condition, *max* function produces a distinctly Non-Gaussian output (see the PDF in Figure 2). The PDF of *max*(A_1, A_2) is clearly Non-Gaussian, though A_1 and A_2 are independent Gaussian random variables conforming to N(0, 0.5) and N(1,3) respectively. Similar results can be observed when A_1 and A_2 are correlated Gaussian random variables.

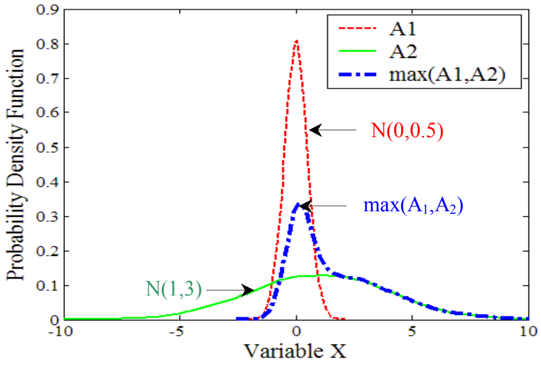

Figure 2 *max* Function

Therefore, if we only use linear approximation on *max* function as

$$A' = \max(A_1, A_2) = aA_1 + bA_2 + c \qquad (1)$$

A', the linear result in Figure 3, is still of Gaussian distribution. A considerable error is introduced by this approximation, compared with the exact *max* output. However, if we use a quadratic delay model instead of the original first-order approximation,

$$A' = a_1 A_1 + a_2 A_1^2 + b_1 A_2 + b_2 A_2^2 + c A_1 A_2 + d \qquad (2)$$

the error will drop dramatically (the quadratic result in Figure 3).

From the above discussion we conclude that linear models are not accurate enough to handle realistic manufacturing variations. Therefore, we propose to apply higher order delay models, such as the quadratic Response Surface Methodology (RSM) models in statistical timing analysis.

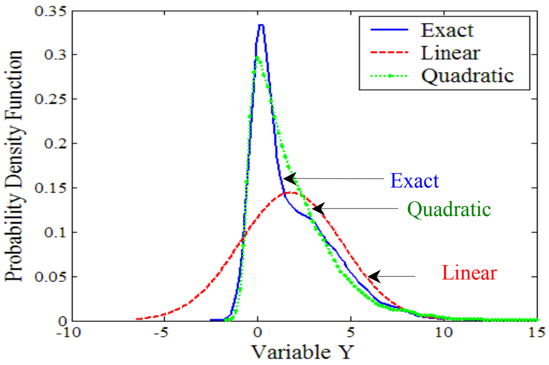

Figure 3 Linear vs. Quadratic Approximation

3. QUADRATIC DELAY MODELS

In our approach, the arrival time at each timing graph node is approximated as a quadratic function of process parameters. Unlike the existing methods [5]-[7] that only apply linear approximations, our approach can handle Non-Gaussian distributions with arbitrary correlations. In this section, we develop a novel methodology to perform the atomic operations (*sum* and *max*) under quadratic delay models.

3.1. PDFs of Quadratic Functions with Gaussian Random Variables

Since we use quadratic models, we'll first derive the PDF's of quadratic random functions. For simplicity, we only derive the single parameter random function to illustrate the basic idea. It should be noted, however, the following mathematical equations can be extended to the cases with multiple random variables by simple convolutions, as is discussed in Section 3.4.

Assume random variable x has a normalized Gaussian distribution. Its PDF f_x is:

$$f_x(x) = \frac{1}{\sqrt{2\pi}} \exp(-x^2/2) \qquad (3)$$

Random variable y is a quadratic function of x:

$$y = a(x+b)^2 \qquad a \neq 0 \qquad (4)$$

Then, according to probability theorems, if a>0, the PDF of y can be expressed as:

$$f_y(y) = \begin{cases} 0 & y<0 \\ \frac{1}{2\sqrt{2\pi ay}}[\exp(-\frac{(\sqrt{y/a}-b)^2}{2}) + \exp(-\frac{(\sqrt{y/a}+b)^2}{2})] & y \geq 0 \end{cases} \qquad (5)$$

Similarly, if a<0, the PDF is given by:

$$f_y(y) = \begin{cases} \frac{1}{2\sqrt{2\pi ay}}[\exp(-\frac{(\sqrt{y/a}-b)^2}{2}) + \exp(-\frac{(\sqrt{y/a}+b)^2}{2})] & y<0 \\ 0 & y \geq 0 \end{cases} \qquad (6)$$

If the second order coefficient in the function is zero, function y is degraded to a linear function. Thus, y is of Gaussian distribution, and its PDF can be derived easily.

3.2. *Sum* Operation

Let's now look at the *sum* operation. For all quadratic functions, we can represent them in the following format:

$$y = x^T A x + Bx + C \qquad (7)$$

in which, $x=(x_1, x_2, \ldots x_n)'$ is the independent process parameter vector, with normalized Gaussian distributions N(0,1), derived from a PCA process. A is a symmetric $n \times n$ matrix, which contains the coefficients of second order terms of x. B is a $1 \times n$ vector, which are the coefficients of first-order terms of x. C is a scalar, which is the constant term.

Therefore, if we have two random variables $y_1 = x^T A_1 x + B_1 x + C_1$ and $y_2 = x^T A_2 x + B_2 x + C_2$, the *sum* operation is straightforward:

$$y_3 = sum(y_1, y_2) = y_1 + y_2 = x^T A_3 x + B_3 x + C_3 \qquad (8)$$

where $A_3=A_1+A_2$, $B_3=B_1+B_2$, and $C_3=C_1+C_2$.

3.3. Orthogonalization

Before we start analyzing *max* operation, we would like to discuss an orthogonalization concept. In order to simplify *max* operation, we'd like to remove the cross terms $x_i x_j$ in the quadratic expressions.

$$max(y_1, y_2) = y_1 + max(0, y_2 - y_1) \qquad (9)$$

where y_1 and y_2 are as defined in Section 3.2. A_2-A_1 is a symmetric matrix, so it can be factorized into $P^T \Sigma P$, where Σ is the diagonal matrix composed of the eigenvalues of A_2-A_1, and P is the corresponding eigenvector matrix.

Let $z = Px, \Phi = (B_2 - B_1)P^T$, we obtain

$$max(y_1, y_2) = y_1 + max(0, z^T \Sigma z + \Phi z + C_2 - C_1) \qquad (10)$$

which no longer includes cross terms in the *max* operation. Because random variables x_i's are independent and Gaussian, vector $z = (z_1, z_2, \ldots, z_n)' = Px$ is a Gaussian vector as well. Moreover, since the eigenvectors P of a symmetric matrix are orthonormal, we can prove that the z_i's are uncorrelated. According to the property of Gaussian distributions, uncorrelated Gaussian random variables are also independent. The detailed proof is given in [10]. Therefore, we can always map the original parameter base into a new base without cross terms, do *max* operation under the new base, and map results back to the original base. From now on, we'll assume that the input operands of *max* do not include cross terms.

3.4. *Max* Operation

Based on the orthogonalization presented in Section 3.3, the input operands of *max* are quadratic functions of an independent normalized base $(x_1, x_2, \ldots, x_n)'$ without cross terms. That is to say all x_i's are N(0,1) Gaussian random variables, and x_i, x_j are independent for any $1 \leq i < j \leq n$. Thus we have the following properties:

$$E(x_i)=0,\ E(x_i^2)=1,\ E(x_i^3)=0,\ E(x_i^4)=3$$

$$E(x_i \cdot x_j)=E(x_i)\cdot E(x_j)=0 \quad \forall i \neq j$$

Now assume:

$$D_1 = a_0 + \sum_{i=1}^{n}(a_{2i-1}x_i + a_{2i}x_i^2) \tag{11}$$

$$D_2 = b_0 + \sum_{i=1}^{n}(b_{2i-1}x_i + b_{2i}x_i^2) \tag{12}$$

The objective of this derivation is to calculate the coefficients C_i of

$$D = x^T C_1 x + C_2 x + C_3 = max(D_1, D_2) \tag{13}$$

We should note, however, that C_1 is not necessarily diagonal.

We now substitute both (11) and (12) into (13):

$$\begin{aligned} D = max(D_1,D_2) = D_1 + (b_{2i-1} - a_{2i-1})x_i + (b_{2i} - a_{2i})x_i^2 \\ + max[g(x_i)\ ,\ \ h(x_1,...,x_{i\text{-}1},x_{i+1},...,x_n)] \end{aligned} \tag{14}$$

where

$$g(x_i) = (a_{2i-1} - b_{2i-1})x_i + (a_{2i} - b_{2i})x_i^2 \tag{15}$$

$$\begin{aligned} &h(x_1,...,x_{i-1},x_{i+1},...,x_n) \\ &= D_2 - D_1 - (b_{2i-1} - a_{2i-1})x_i - (b_{2i} - a_{2i})x_i^2 \end{aligned} \tag{16}$$

g is a quadratic function of x_i only, and h is a quadratic function of the other (n-1) variables, but without x_i. As a result, g and h are statistically independent, since they are functions of different independent random variables. Therefore, the Joint Probability Density Function of g and h is the product of their corresponding single PDFs.

If we re-write function $g(x_i)$ in the format of (4), we can immediately get its PDF $f_g(\xi)$ analytically from (5) or (6), because x_i is of N(0,1) distribution. Similarly, if we re-write function $h(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$ as:

$$\omega = h(x_1,...,x_{i-1},x_{i+1},...,x_n) = u + \sum_{j \neq i} y_j = u + \sum_{j \neq i} r_j (x_j + t_j)^2 \tag{17}$$

we'll have analytical PDFs for each of the square terms $y_{j=} r_j (x_j + t_j)^2$ according to Section 3.1. Since y_j's are statistically independent, the PDF $f_\omega(\eta)$ of random variable ω is the convolution of all the PDF's of y_j's. Once we have the PDF of ω, its CDF $F_\omega(\eta)$ can be derived by numerical integration easily.

With all this information in hand, we can now start calculating the first order and second order moments of x_i's on D of (14).

The first order moment of x_i is

$$\begin{aligned} \underset{(x_1,...x_n)}{E}(x_i D) &= b_{2i-1} + \underset{(x_1,...,x_n)}{E}(x_i \cdot max(g,h)) \\ &= b_{2i-1} + \underset{h \le g}{E}(x_i g(x_i)) + \underset{h \ge g}{E}(x_i h(x_1,...,x_{i-1},x_{i+1},...,x_n)) \\ &= b_{2i-1} + \int_{\xi} \int_{\eta \le g(\xi)} \xi \cdot g(\xi) \cdot f_{x_i}(\xi) \cdot f_\omega(\eta) \cdot d\eta d\xi \\ &\quad + \int_{\xi} \int_{\eta \ge g(\xi)} \xi \cdot \eta \cdot f_{x_i}(\xi) \cdot f_\omega(\eta) \cdot d\eta d\xi \end{aligned} \tag{18}$$

$$\begin{aligned} \underset{(x_1,...x_n)}{E}(x_i D) = b_{2i-1} + \int_{\xi} \xi \cdot g(\xi) \cdot f_{x_i}(\xi) \cdot F_\omega[g(\xi)] \cdot d\xi \\ + \int_{\eta} \eta \cdot f_\omega(\eta) \underset{g(x_i) \le \eta}{E}(x_i) \cdot d\eta \end{aligned} \tag{19}$$

For illustration purposes, we re-write equation (14) as

$$D = \max(D_1, D_2) = \sum_{i,j} \alpha_{ij} x_i x_j + \sum_i \beta_i x_i + \gamma \tag{20}$$

From (20), we can also get the first moment of x_i,

$$\underset{(x_1,...x_n)}{E}(x_i D) = \beta_i E(x_i^2) = \beta_i \qquad 1 \le i \le n \tag{21}$$

because x_i is of N(0,1) distribution, and x_i is independent to all x_j's when $j \neq i$. Equating (19) and (21), we've got all the first-order term coefficients in the quadratic form by first-order moment matching.

Now, let's look at the second order moments in a similar manner.

$$\begin{aligned} &\underset{(x_1,...x_n)}{E}(x_i^2 D) = 3b_{2i} + \sum_{j \neq i} a_{2j} + \underset{(x_1,...,x_n)}{E}(max(g,h)) \\ &= 3b_{2i} + \sum_{j \neq i} a_{2j} + \underset{h \le g}{E}(x_i^2 \cdot g(x_i)) + \underset{h \ge g}{E}(x_i^2 \cdot h(x_1,..., x_{i-1}, x_{i+1},..., x_n)) \\ &= 3b_{2i} + \sum_{j \neq i} a_{2j} + \int_{\xi} \xi^2 \cdot g(\xi) \cdot f_{x_i}(\xi) \cdot F_\omega[g(\xi)] \cdot d\xi \\ &\qquad + \int_{\eta} \eta \cdot f_\omega(\eta) \cdot \underset{g(x_i) \le \eta}{E}(x_i^2) \cdot d\eta \end{aligned} \tag{22}$$

Again, from expression (20), we have

$$\begin{aligned} E(x_i^2 D) &= \alpha_{ii} E(x_i^4) + \sum_{j \neq i} \alpha_{jj} E(x_j^2) E(x_i^2) + \gamma E(x_i^2) \\ &= 3\alpha_{ii} + \sum_{j \neq i} \alpha_{jj} + \gamma \end{aligned} \tag{23}$$

plus matching the mean value of D, which is:

$$E(D) = \int_{\eta} \eta \cdot f_D(\eta) d\eta = \sum_{i=1}^{n} \alpha_{ii} + \gamma \tag{24}$$

By putting the n second order moments and mean matching together, we get n+1 linear equations:

$$\underbrace{\begin{pmatrix} 3 & 1 & \ldots & 1 & 1 \\ 1 & 3 & \cdots & 1 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & \cdots & 3 & 1 \\ 1 & 1 & \cdots & 1 & 1 \end{pmatrix}}_{n+1} \bullet \begin{pmatrix} \alpha_{11} \\ \alpha_{22} \\ \vdots \\ \alpha_{nn} \\ \gamma \end{pmatrix} = \begin{pmatrix} E(x_1^2 \cdot D) \\ E(x_2^2 \cdot D) \\ \vdots \\ E(x_n^2 \cdot D) \\ E(D) \end{pmatrix} \tag{25}$$

By solving (25), we get all the coefficients of the square terms and the constant term as well.

So far, we have all the coefficients except those of the cross terms, i.e. $\alpha_{ij}, i \neq j$. To calculate them, we need to modify (14) a little bit.

We re-write (14) as (26):

$$\begin{aligned} &D = max(D_1, D_2) \\ &= D_1 + (b_{2i-1} - a_{2i-1})x_i + (b_{2i} - a_{2i})x_i^2 + (b_{2j-1} - a_{2j-1})x_j + (b_{2j} - a_{2j})x_j^2 \\ &+ max[g(x_i, x_j)\ ,\ h(x_1,...,x_{i-1}, x_{i+1},..., x_{j-1}, x_{j+1},..., x_n)] \end{aligned} \tag{26}$$

where

$$\begin{aligned} g(x_i, x_j) = (a_{2i-1} - b_{2i-1})x_i + (a_{2i} - b_{2i})x_i^2 \\ + (a_{2j-1} - b_{2j-1})x_j + (a_{2j} - b_{2j})x_j^2 \end{aligned} \tag{27}$$

$$\begin{aligned} h = D_2 - D_1 - (b_{2i-1} - a_{2i-1})x_i - (b_{2i} - a_{2i})x_i^2 \\ - (b_{2j-1} - a_{2j-1})x_j - (b_{2j} - a_{2j})x_j^2 \end{aligned} \tag{28}$$

g is a quadratic function of parameters x_i, x_j only, and h is a quadratic function of the other (n-2) variables, but without x_i and x_j.

Applying the same calculations as in (18)(19), we get

$$\underset{(x_1,\ldots x_n)}{E}(x_i x_j D) = \iint\limits_{\xi,\eta} \xi \cdot \eta \cdot g(\xi,\eta) \cdot f_{x_i}(\xi) f_{x_j}(\eta) \cdot F_\omega[g(\xi,\eta)] \cdot d\xi \cdot d\eta$$
$$+ \int\limits_{\zeta,\eta}\int \zeta \cdot \eta \cdot f_{x_i}(\eta) \cdot f_\omega(\zeta) \cdot \underset{\zeta \geq g(x_i,\eta)}{E}(x_i) \cdot d\eta \cdot d\zeta \quad (29)$$

From (20), again we have

$$E(x_i x_j D) = \alpha_{ij} E(x_i^2) E(x_j^2) = \alpha_{ij} \quad (30)$$

Therefore, by equating (29) and (30), all cross term coefficients α_{ij}'s are obtained.

Putting together all the results from (19), (25) and (29), we have derived all the coefficients that we need for the quadratic expression in (20). The integrations in (19) (22), (24) and (29) can be numerically computed, e.g. using the piece-wise linear approximation algorithm proposed in [3]. By substituting the fitted quadratic result (20) into (13), and mapping the orthogonal base back to the original base, we manage to fit the nonlinear *max* operation as a quadratic form of the normalized independent Gaussian base, which is derived from a PCA process.

Returning to Figure 3 in Section 2, the quadratic curve is the result fitted from the approach shown above. There is a huge improvement from the linear model approach in terms of accuracy. Using the same analysis, similar quality fit for *min* operation can be derived as well. To summarize, we have demonstrated a novel fitting approach, which is the core of doing nonlinear block-based Statistical Timing Analysis under quadratic delay models.

4. ALGORITHM

The main algorithm of our block-based algorithm is shown in Figure 4.

```
block_based_algorithm {
        initialize();
        breadth_first_search {
                if (current==single_input_component)
                        sum operation;
                else{ /*current==multi_input_component*/
                        max operation pair-wisely;
                        second_order_term_prune();
                }
        }
}
```

Figure 4 Block_based SSTA Algorithm

In the algorithm, initialize() first applies a PCA process to get a normalized independent Gaussian base. Then, component delays with quadratic models are extracted. Next, a breadth-first search is launched to propagate delay distributions from source to sink, level by level.

The *max* operation basically follows what we demonstrated in Section 3.4. It first performs the orthogonalization of the base to get rid of the cross terms. After that, it convolves the analytical PDFs of the square terms in (17). Since we keep all the convolution results in a table, we can reuse them for different moments. For example, both E(x_1D) and E(x_2D) need the convolution results of the square terms from x_3 to x_n. Then we use numerical integration to do the moment matches of (19), (22), (24) and (29). After that, the linear equations are solved through LU factorization. It is worth pointing out that, once the number of parameters, i.e. the parameter base size, is determined, the linear equations (25) are fixed for all *max* operations except for the right hand side moments. Therefore, we pre-calculate the inverse matrix of the coefficient matrix on the left hand side and save it for computational efficiency. Finally, the solutions are mapped back to the original base.

The second_order_term_prune() function simplifies the delay expression by removing insignificant second order terms with small coefficients. As you may notice, this algorithm can be easily extended for higher order delay models. For example, if we want to use third-order delay models, we just need to include more moments like $\underset{(x_1,\ldots,x_n)}{E}(x_i^3 \cdot max(D_1, D_2))$ and then fit the coefficients. The tradeoff is the computation complexity due to more coefficients to be calculated.

5. EXPERIMENTS

5.1 Implementation and Results

We've implemented the proposed algorithm in C++, and compared the following in two experimental set-ups: 1) 10,000 Monte Carlo simulations; 2) first-order SSTA with linear delay models; and 3) the proposed quadratic approach with nonlinear delay models. In the first case, we tested this approach on the ISCAS85 benchmark circuits implemented in TSMC 0.18µm technology. Both inter-die variations and spatial correlated intra-die variations were included in the experiments. Experiments were performed on a 900MHz Sunfire workstation. The second experiment was performed on a portion of an industrial chip M1 with 1K gates, in a 90nm technology.

Table 1 shows the results for all ISCAS85 benchmark circuits. The row #Grids provides the spatial correlation information. For example, we divided circuit C432 into four groups. In each group, same intra-die variations (perfect correlations) are assumed because of the close spatial location. In between those groups that are closer to each other, higher correlations are considered in the intra-die variations. All groups share the same inter-die variations. Table 1 then lists the circuit delay error on the 99% yield point for first-order methodology (EOL) and second-order methodology (EOQ) compared to the 10,000 Monte Carlo results. First-order methodology has an average of 7.74% error, while the proposed methodology has a perfect result with almost zero error. The error is measured as the difference in delay value over 6σ of the delay distribution (Δdelay/6σ). The CPU time of the proposed methodology (TOQ) is also compared with both first-order methodology (TOL) and the Monte Carlo approach (TOM). The runtime of our methodology is in the order about 1/100 of the 10,000 Monte Carlo analysis with comparable accuracy. With the increase of circuit size for ISCAS85 benchmark circuits, the run time of the proposed methodology increases almost linearly.

For the industrial chip example in the second experiment, the circuit delay PDFs for the three approaches are shown in Figure 5. Our quadratic approach result practically overlaps the 10,000 Monte Carlo simulation results. There are some errors between the first-order SSTA (the linear curve) and the Monte Carlo results at both tails of the PDF. As we noted above, this is due to the non-zero skew introduced by nonlinear delay models. The 99% yield point error is calculated for both first-order and quadratic approaches. The first-order methodology has an 8.09% error while the proposed approach only has a 0.68% error. The CPU runtime for the proposed quadratic approach is 11.01s compared to 4521s for 10,000 Monte Carlo simulations.

It should be noted out that in both experimental set-ups, all cross terms are ignored in our quadratic approach, which seems to have very little impact on the overall accuracy.

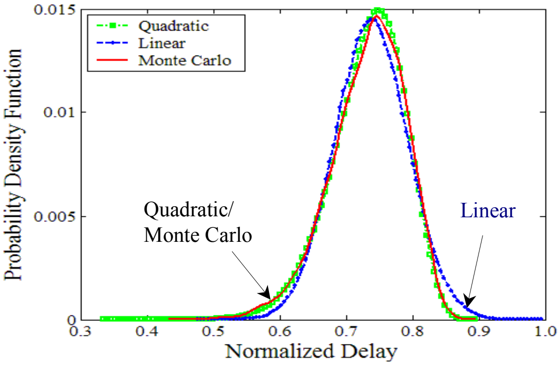

Figure 5 Circuit Delay PDF of M1

5.2 Complexity

As shown in Section 4.1, we only need to walk through the circuit in a breadth-first manner once. That determines the complexity of the algorithm which is linear with respect to the circuit size. As for the number of parameters, it is quadratic if we want to fit the full quadratic terms. This is because if we have *n* parameters, we will have 1 constant term, *n* first-order terms, and n^2 second-order terms. However, from our experiments, we find that most cross terms are negligible. As shown in Section 5.1, almost zero errors are observed between the proposed approach and Monte-Carlo results even without cross terms. Therefore, it is sufficient to keep only several important cross terms and ignore the majority of them. This makes the complexity of this algorithm improve from O(n^2) to O(n) in terms of the parameter base size. Another cost in this algorithm lies in the numerical convolutions and integrations. Since fast convolutions algorithm with FFT are used, it is of *klog(k)* complexity, where *k* is the number of points being used. Therefore, the algorithm is an efficient and practical even for large industrial circuits.

6 CONCLUSIONS

The main contributions of this work can be viewed in the following two aspects. First of all, we have proposed and developed a novel block-based Statistical Static Timing Analysis algorithm with nonlinear delay modeling. The quadratic approximation of the nonlinear *max* operation is performed via moment matching, which considerably increases the robustness of the block-based SSTA algorithm. Therefore, our algorithm is capable of propagating quadratic delay models with high accuracy. Our prototype was implemented and tested on ICSAS85 benchmark circuits as well as industrial circuits. The results show both accuracy and efficiency under nonlinear delay models. Secondly, we have linked SSTA approaches to more realistic process variational models via accurate characterization. This makes the proposed nonlinear modeling approach practical for realistic industrial circuits. Future work includes circuit optimization based on SSTA results and more general statistical delay modeling.

7. ACKNOWLEDGEMENT

This work was supported by the MARCO Focus Center for Circuit & System Solutions (C2S2, www.c2s2.org) under contract 2003-CT-888.

8. REFERENCES

[1] M. Orshansky and K. Keutzer, "A General Probabilistic Framework for Worst Case Timing Analysis", Proc. DAC, pp 556-561, June 2002

[2] J. A. G. Jess and K. Kalafala et al, "Statistical timing for parametric yield prediction of digital integrated circuits", Proc. DAC, pp. 932-937, June 2003

[3] A. Devgan and C. Kashyap, "Block-based Static Timing Analysis with Uncertainty", IEEE ICCAD, pp. 607-614, November 2003

[4] A. Agarwal, D. Blaauw, V. Zolotov And S. B. K. Vrudhula, "Statistical Timing Analysis with Uncertainty", DATE, pp. 62 -67, 2003

[5] H. Chang, S. S. Sapatnekar, "Statistical timing analysis considering spatial correlations using a single PERT-like traversal", IEEE ICCAD, pp. 621-625 November 2003

[6] C. Visweswariah, K. Ravindran, K. Kalafala, S. G. Walker, S. Narayan, "First-Order Incremental Block-Based Statistical Timing Analysis", Proc. 2004 DAC, pp. 331-336, June 2004

[7] J. Le, X. Li, L. T. Pileggi, "STAC: Statistical Timing Analysis with Correlation", Proc. DAC, pp. 343-348, June 2004

[8] D. F. Morrison, "Multivariate Statistical Methods", New York: McGraw-Hill, 1976

[9] S. R. Nassif, "Modeling and Analysis of Manufacturing Variations", IEEE CICC, pp. 223-228, 2001

[10] X. Li, J. Le, P. Gopalakrishnan, and L. T. Pileggi, "Asymptotic Probability Extraction for Non-Normal Distributions of Circuit Performance", IEEE ICCAD, pp. 2-9, November 2004.

Table 1 Results of ISCAS85 Benchmark Circuits

Circuit	C17	C432	C499	C880	C1355	C1908	C2670	C3540	C5315	C6288	C7552
#Grids	1	4	4	4	16	16	16	16	64	64	64
EOL(%)	8.23	7.70	9.78	7.45	8.21	7.09	7.88	7.59	7.44	6.62	7.13
EOQ(%)	0.06	0.12	0.40	0.11	0.03	0.20	0.19	0.18	0.07	0.10	0.23
TOM(s)	3.02	85.41	74.52	138.84	200.62	317.24	433.75	609.74	847.95	904.82	1283.14
TOL(s)	0.01	0.04	0.08	0.14	0.26	0.54	1.06	1.46	2.03	2.92	3.48
TOQ(s)	0.02	0.17	0.42	0.78	3.75	3.55	3.29	4.9	5.31	7.86	12.59