

Parameterized Timing Analysis with General Delay Models and Arbitrary Variation Sources

Khaled R. Heloue
Department of ECE
University of Toronto
Toronto, Ontario, Canada
khaled@eecg.utoronto.ca

Farid N. Najm
Department of ECE
University of Toronto
Toronto, Ontario, Canada
f.najm@utoronto.ca

ABSTRACT

In general, parameterized timing analysis must be able to handle nonlinear delay models and must account for delay variability due to both *random* process parameters (with arbitrary distributions) and *uncertain* non-random parameters (which may depend on the operating environment). Existing statistical static timing analysis (SSTA) techniques suffer from at least one of the following limitations: 1) restricting their modeling capabilities to linear models, or 2) limiting random variations to Gaussian distributions, or 3) from their inability to handle uncertain non-random parameters. In this work, we propose a general mathematical framework for resolving the *max* operation in parameterized timing analysis that can overcome all the above limitations. The framework is general, in the sense that it can be applied to a general class of nonlinear delay models. In addition, it can handle both statistical process parameters, with arbitrary distributions, as well as uncertain non-random parameters. The *max* operator is efficiently resolved using carefully chosen linear combinations that preserve the inherent nonlinearity of the delay model used. Our general technique is tested for two applications, namely multi-corner timing analysis and nonlinear non-Gaussian SSTA, where we show that its complexity is linear in both the number of process parameters and the size of the circuit. Our results show that, on average, all timing characteristics of circuit delay are predicted with less than 2% error for multi-corner analysis, and less than 1% error for SSTA.

1. INTRODUCTION

Process and environmental variations greatly impact circuit delay, and can cause parametric yield loss if not accounted for. With the traditional approaches to timing verification becoming too expensive and unable to handle local variations, new alternatives have emerged in recent years, most of which fall under the heading of parameterized timing analysis. Essentially, timing quantities are “parameterized” as a function of the underlying process and environmental parameters, allowing one to assess the effect of these parameters on circuit delay, which can be useful in determining the robustness of the design and its sensitivity to variations. It is worth mentioning that process variations¹, which are manufacturing variations or mismatches between transistors’ parameters, are *statistical* variations and can be modeled as random variables (RVs), whereas environmental variations, which are operating context variations, are non-random and must be treated as *uncertain* parameters. It is understood that both types of variations must be taken care of in the context of parameterized timing analysis.

One of the major trends in parameterized timing analysis is block-based statistical static timing analysis (SSTA) [1, 7, 2, 8, 9, 4, 6, 3], where parameters are modeled as RVs. In the past few years, several SSTA techniques have been proposed. In [1, 7], linear-time techniques were proposed using linear (first-order) delay models and Gaussian distributions for process parameters, which allows the use of tightness probability to resolve the *max* operation efficiently. It is expected, however, that nonlinearities will increase with technology scaling. On one hand, as the magnitude of process variations is increasing, first-order delay models are no longer accurate. Also, there is evidence that delay is highly nonlinear in low-voltage modes. On the other hand, process variations are not necessarily Gaussian. For example gate length has an asymmetric non-Gaussian distribution due to variation in depth of focus (DOF). To address these concerns, several attempts were made to generalize SSTA in order to handle non-

linear delay models and/or random variables with non-Gaussian distributions. In [8, 9], despite the use of quadratic models, Gaussian distributions were forced on process parameters. In [6], non-Gaussian process parameters were addressed, but first-order linear models were used. Recently, both non-Gaussian parameters and nonlinear models were addressed simultaneously, with varying degrees of success. In [4], a sampling (Monte Carlo) based regression is used to handle the *max* operation, which increases the overall runtime; [2] uses expensive multi-dimensional integration to handle the nonlinear non-Gaussian terms, which hinders the complexity of the approach; [3] proposes an efficient nonlinear non-Gaussian SSTA technique with linear complexity. This technique uses Fourier series and moment matching to approximate the *max* operation efficiently. The drawback of this approach is that it requires subdividing the region of variation of every variation source, then for every sub-region, the Fourier transform of every possible distribution used is pre-computed. Although this is done as a pre-processing step, it can introduce undesired complications, especially if these distributions are unknown. In addition, by using moment matching and being distribution dependent, all these SSTA techniques fail to handle uncertain non-random parameters, which is a requirement in general parameterized timing.

Another type of parameterized timing analysis is linear-time multi-corner static timing analysis (STA), which was introduced in [5]. The idea is to propagate *hyperplanes*, i.e., affine linear functions of the process and environmental parameters, in the timing graph. The *max* operation is resolved by raising the hyperplanes in a way to always follow the maximum corner delay. In this way, the circuit maximum corner delay is estimated accurately. Although the approach can handle both random and uncertain parameters, it has several drawbacks: while trying to be always accurate at the maximum corner delay, the approach loses accuracy at the other corners. Consequently, the sensitivity to process variations is lost and this approach can not be used for optimization because the spread of the circuit delay is not accurately estimated. In addition, the analysis is restricted to linear delay models.

In this work, we propose an efficient and general parameterized timing analysis technique that can handle nonlinear delay models and account for delay variability due to both random process parameters with arbitrary (and possibly unknown) distributions and uncertain non-random parameters (that typically may depend on the operating environment). Our technique is based on a novel and efficient method to resolve the nonlinear *max* operator by either bounding or approximating it by a linear model, while preserving the inherent nonlinearity of the delay model itself. We have tested our technique within two timing verification frameworks, namely multi-corner timing analysis and nonlinear non-Gaussian SSTA, and have shown that, at least when using quadratic delay models, the complexity of the approach is linear in both the number of process and environmental parameters and the circuit size. Our results show that the spread of the maximum circuit delay is accurately captured, whereby, on average, the maximum and minimum corner delays are predicted with less than 2% error for multi-corner analysis. As for nonlinear non-Gaussian SSTA, all timing characteristics are predicted with less than 1% average error.

2. GENERAL PARAMETERIZED TIMING

In general, the delay of gates and interconnects is a nonlinear function that can be approximated using a *delay model* that depends on the underlying process and environmental variations. As stated earlier, linear (first-order) and quadratic (second-order) delay models have been previously used in the literature. Nevertheless, for the time being, we will work under the assumption of

¹Here, we are referring to non-systematic process variations

an arbitrary delay model; we will show later that our approach can be used with a general class of nonlinear delay models. We can write the delay D of gates or interconnects as follows:

$$D = f_D(X_1, \dots, X_p) \quad (1)$$

where $f_D(\cdot)$ is an arbitrary delay model, and X_i 's represent process and environmental variations. Recall that not all variations can be modeled as random variables. While some parameters are indeed *random* such as channel length or threshold voltage, others are not statistical but rather *uncertain* parameters, such as temperature or supply voltage. These parameters typically depend on the operating environment. We will show that our analysis is indifferent to whether parameters are random or uncertain and thus the approach can be applied for both types of variations.

The benefit of parameterized timing analysis is in elevating the explicit delay dependence on process and environmental variations, i.e., the *delay model*, from the stage level to the circuit level, allowing one to assess the effect of these parameters on the total circuit delay. In block-based timing analysis, this is done by propagating timing quantities in the timing graph in topological order, using a sequence of basic operations, such as *add* operations on arrival times and arc delays, and *max* operations on the timing quantities resulting from those additions in order to get the output arrival time. If these basic operations do not distort the delay model at hand, then circuit delay can be represented in the same delay model. Unfortunately, it is known that the *max* is nonlinear. The novelty of this work is in resolving the nonlinear *max* operator using a linear model, while preserving the inherent nonlinearity of the delay model.

In the following section, we propose two methods to efficiently resolve the *max*: first by using upper and lower bounds, and second by using an approximation that minimizes the square of the error. Our methods operate irrespective of the delay model used provided it falls in a general class of nonlinear functions; they also handle random parameters with arbitrary distributions, as well as uncertain parameters. In the rest of the paper, we will refer to random and uncertain parameters simply as process/environmental parameters.

3. MATHEMATICAL FRAMEWORK

Let \mathcal{F} be a general class of (possibly) nonlinear functions that obey the following three properties,

1. \mathcal{F} is closed under linear (and/or affine) operations
2. $\forall f(\cdot) \in \mathcal{F}$, $f(\cdot)$ is bounded
3. $\forall f(\cdot) \in \mathcal{F}$, $f(\cdot)$ can be maximized and minimized efficiently

and assume that all timing quantities are represented using delay models that are in \mathcal{F} . For example, timing quantity A is modeled as follows:

$$A = f_A(X_1, \dots, X_p) = f_A(X) \in \mathcal{F} \quad (2)$$

where $X = [X_1, \dots, X_p]$ represent process/environmental parameters.

The first property essentially states that linear operations (such as addition or subtraction) will result in functions that belong to \mathcal{F} ; in other words, \mathcal{F} "survives" linear operations, that is, if $A = f_A(X) \in \mathcal{F}$ and $B = f_B(X) \in \mathcal{F}$, then $C = aA + bB + c$, where $[a, b, c] \in \mathcal{R}$, will be such that $C = f_C(X) \in \mathcal{F}$. Note that all polynomial models satisfy this property. The second and third properties imply that the delay model, and consequently any timing quantity expressed using the model, is bounded by a minimum and a maximum value over the space of parameters, i.e., $A_{\min} \leq A \leq A_{\max}$, where:

$$A_{\min} = \min_X f_A(X) \quad \text{and} \quad A_{\max} = \max_X f_A(X) \quad (3)$$

Note that the second property is trivial if we recall that all physical process and environmental parameters are bounded, which implies that delays are also bounded. The third property is only forced to guarantee that our approach, which requires maximizing and minimizing the delay model in order to operate, is computationally efficient. For instance, if maximizing/minimizing the

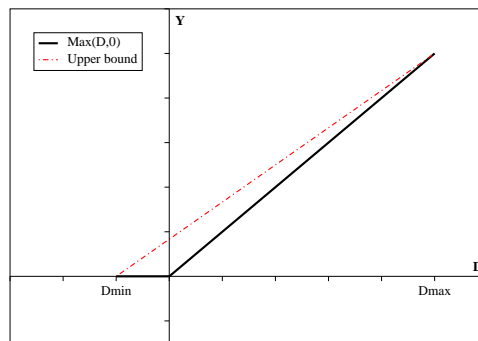


Figure 1: Upper bound on Y

delay model is linear (in the process/environmental parameters), then the overall complexity of our approach is linear in the circuit size and process/environmental parameters. By property 1, it is clear that the *add* operation, being linear, will maintain their membership in \mathcal{F} as delay models are propagated during timing analysis. However, the *max* operation, being non-linear in general, is the crux of the problem, and we now focus on it.

3.1 Max Operation

Let A and B be two timing quantities expressed using delay models in \mathcal{F} . Let $C = \max(A, B)$ be the maximum of A and B . Since the maximum operator is, in general, nonlinear, then C is not necessarily expressible using a delay model in \mathcal{F} . We are interested however in finding (1) two bounds on C , C_l and C_u , and (2) an approximation C_a , which can all be expressed using functions in \mathcal{F} , and such that:

$$C_l \leq C \leq C_u \quad (4)$$

$$C_a \approx C \quad (5)$$

where $C_l = f_{C_l}(X) \in \mathcal{F}$, $C_u = f_{C_u}(X) \in \mathcal{F}$, $C_a = f_{C_a}(X) \in \mathcal{F}$. Notice that:

$$C = \max(A, B) = B + \max([A - B], 0) \quad (6)$$

$$= B + \max(D, 0) = B + Y \quad (7)$$

where $D = A - B$ and $Y = \max(D, 0)$.

Recall from property 1 that \mathcal{F} survives linear operations (including subtraction), which means that $D = f_D(X) \in \mathcal{F}$. By properties 2 and 3, D is bounded and varies between $[D_{\min}, D_{\max}]$ where $D_{\min} \leq D_{\max}$. Depending on the signs of these extreme values of D , we can identify two cases in which the *max* operator is either linear or nonlinear. If $D_{\min} \geq 0$ then $D \geq 0 \forall X$, and $Y = D$. In this case, $C = A$ since A completely dominates B . The converse happens when $D_{\max} \leq 0$; in this case, $C = B$, since B completely dominates A . The more interesting case is when the *max* is nonlinear, which can be identified when $D_{\max} \geq 0$ and $D_{\min} \leq 0$. In this case, which we will refer to by saying that A and B are *co-dominant*, $Y = \max(D, 0)$ can not be expressed using a delay model from \mathcal{F} . We will now show how we can bound and approximate Y using functions that belong to \mathcal{F} .

3.2 Bounding the Max

3.2.1 Upper bound

Fig. 1 shows a broken solid line representing a plot of $Y = \max(D, 0)$ between D_{\min} and D_{\max} , the extreme values of D . We are interested in finding a linear function of D that is guaranteed to upper bound Y (recall, since $D \in \mathcal{F}$, then any linear function of D is also a member of \mathcal{F}). The dashed line represents an affine function of D which upper bounds Y and is exact at D_{\max} and D_{\min} . Note that the bound is closer to the exact *max* around D_{\min} and D_{\max} where either A or B dominates. The equation for Y_u , the upper bound on Y , can be expressed as follows:

$$Y_u = \frac{D_{\max}}{D_{\max} - D_{\min}}(D - D_{\min}) \quad (8)$$

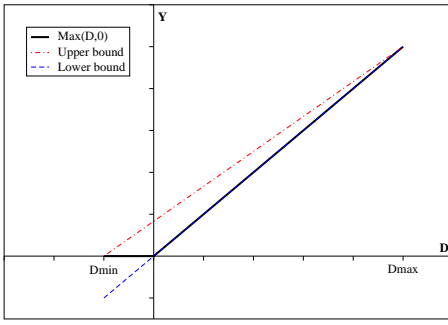


Figure 2: Lower bound $Y_l = D$

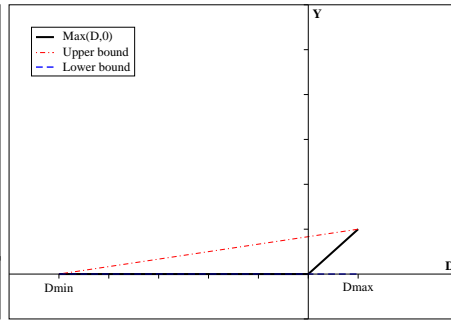


Figure 3: Lower bound $Y_l = 0$

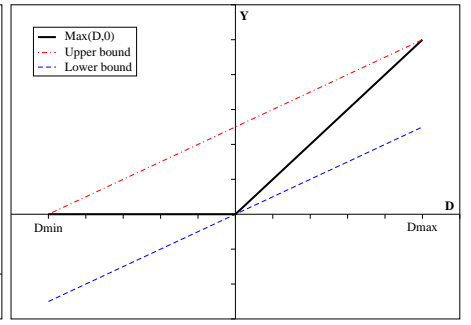


Figure 4: Lower bound on Y

By replacing Y with Y_u in (7), we get an upper bound C_u on C :

$$\begin{aligned} C_u &= B + Y_u = B + \frac{D_{\max}}{D_{\max} - D_{\min}} ([A - B] - D_{\min}) \\ &= \left(\frac{D_{\max}}{D_{\max} - D_{\min}} \right) A - \left(\frac{D_{\min}}{D_{\max} - D_{\min}} \right) B - \frac{D_{\max} \cdot D_{\min}}{D_{\max} - D_{\min}} \end{aligned} \quad (9)$$

Note that $C_u = f_{C_u}(X) \in \mathcal{F}$ since it is a linear combination of A and B (property 1). To gain a more intuitive understanding of the above relationship, let us define the following terms:

- $S = D_{\max} - D_{\min}$ to be the “spread” of D
- $S_A = D_{\max}$ to be the “strength” of A , i.e. the region where A dominates B ($D \geq 0$)
- $S_B = -D_{\min} = |D_{\min}|$ to be the “strength” of B , i.e. the region where B dominates A ($D \leq 0$)
- $\alpha = \frac{S_A}{S}$ is the fraction of space where A dominates B
- $(1 - \alpha) = 1 - \frac{S_A}{S} = \frac{S_B}{S}$ is the fraction of space where B dominates A

Then, using the above notations, we can rewrite C_u as follows:

$$C_u = \alpha A + (1 - \alpha) B + \alpha(1 - \alpha) \cdot S \quad (10)$$

where A and B are both weighted by their “extent of dominance”, so to speak, and the last term accounts for the region where both A and B are dominant, hence the product of α and $(1 - \alpha)$.

3.2.2 Lower bound

Similarly, we would like to find a lower bound on Y in order to find a lower bound $C = \max(A, B)$. Looking back at Fig. 1, it is easy to see that any function in the form $Y_l = aD$, where $0 \leq a \leq 1$, is a valid lower bound on $Y = \max(D, 0)$ and can be expressed using a function in \mathcal{F} (property 1). In practice, we have found that limiting the choice of Y_l to one of three functions depending on the values of D_{\min} and D_{\max} is sufficient. Figures 2-4 depict these three cases. In addition to showing Y in solid line and upper bound Y_u in dashed line, these figures show the lower bound Y_l to be one of the following:

$$Y_l = \begin{cases} D & \text{if } |D_{\max}| \gg |D_{\min}| \\ 0 & \text{if } |D_{\max}| \ll |D_{\min}| \\ \left(\frac{D_{\max}}{D_{\max} - D_{\min}} \right) D & \text{otherwise} \end{cases} \quad (11)$$

where the slope of Y_l in the third case is equal to that of the upper bound Y_u . Note that \gg means “much larger than”, and it was set to be at least 4 times larger in our simulations.

Replacing each case of the above in (7) gives us C_l , the lower bound on C :

$$C_l = \begin{cases} A & \text{if } |D_{\max}| \gg |D_{\min}| \\ B & \text{if } |D_{\max}| \ll |D_{\min}| \\ \alpha A + (1 - \alpha) B & \text{otherwise} \end{cases} \quad (12)$$

where α is as defined in (10). Therefore, $C_l = f_{C_l}(X) \in \mathcal{F}$ since it is a linear combination of A and B (property 1).

3.3 Least Squares Max Approximation

In the previous sections, we have shown how we can determine upper and lower bounds on $C = \max(A, B)$ using a carefully chosen linear combination of A and B . We are now interested in finding an expression C_a that approximates $C = \max(A, B)$ and can be expressed using a function that belongs to \mathcal{F} . To do that, it suffices to approximate $Y = \max(D, 0)$ in (7) by a linear approximation Y_a in the form $Y_a = aD + b$. We will choose a and b in such a way to minimize the sum of the squares of the error inside the interval $[D_{\min}, D_{\max}]$; we call this approximation the *least squares max* approximation. It is understood that this approximation is applied only when the *max* is nonlinear, i.e., in the case of co-dominance.

Let E be the total error incurred by the above approximation. Then we can express E as follows:

$$E = \int_{D_{\min}}^{D_{\max}} (aD + b - \max(0, D))^2 dD \quad (13)$$

We now determine the parameters a and b that minimize E . To do that, we first determine the partial derivatives of E , $\frac{\partial E}{\partial a}$ and $\frac{\partial E}{\partial b}$, with respect to a and b , then we set the derivatives to zero and solve for a in b . This gives us the following two equations in a and b :

$$2(D_{\max}^3 - D_{\min}^3) a + 3(D_{\max}^2 - D_{\min}^2) b = 2D_{\max}^3 \quad (14)$$

$$(D_{\max}^2 - D_{\min}^2) a + 2(D_{\max} - D_{\min}) b = D_{\max}^2 \quad (15)$$

Finally, a and b can be easily obtained by solving the system of equations (14) and (15), which result in:

$$a = \frac{D_{\max}^2(D_{\max} - 3D_{\min})}{(D_{\max} - D_{\min})^3} \quad \text{and} \quad b = \frac{2D_{\max}^2 D_{\min}^2}{(D_{\max} - D_{\min})^3} \quad (16)$$

C_a can be easily obtained by replacing Y with $Y_a = aD + b$ in (7). It is easy to see that $C_a = f_{C_a}(X) \in \mathcal{F}$ since it is a linear combination of A and B which are in \mathcal{F} (property 1). Fig. 5 shows the true *max*, $Y = \max(D, 0)$ in solid line, and the least squares *max* approximation, $Y_a = aD + b$, in dashed line.

3.4 Summary

In this section, we have presented a mathematical framework that resolves the nonlinear *max* operator in two ways; either by using bounds, or by using an approximation that minimizes the error with the true *max*. This framework is simple and general, in the sense that it can be applied to a general class of nonlinear delay models, and it uses simple linear operations, allowing to keep and propagate the same delay model to later stages. In addition, the analysis is indifferent (1) to whether the variation sources are modeled as random variables or uncertain variables, or (2) to the types of distributions used, when the variations are random. In spite of its apparent simplicity, we will now demonstrate that this approach is extremely effective and competitive in dealing with two difficult application areas: multi-corner timing analysis as well as nonlinear non-Gaussian SSTA. Crucially, this approach can also deal with the “mixed” case, which is typical of practical situations, where some variables are random while others are simply uncertain.

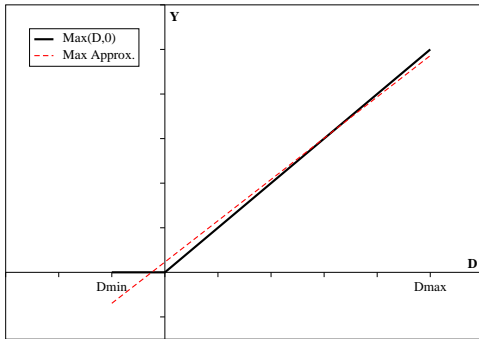


Figure 5: Least squares Max approximation

4. MULTI-CORNER STA

In corner case analysis, circuit timing must be checked at various process/environmental corners, which are typically extreme values of process/environmental parameters. This approach to timing verification is exponential in the number of varying parameters as multiple runs of STA are needed to cover all possible corners in order to determine the maximum and minimum corner delays. Instead, in practice, using parameterized timing, where timing quantities are expressed using a variational delay model that depends on process and environmental parameters, one hopes to do this task with only one traversal of the timing graph; this is what we call linear-time multi-corner STA.

In this section, we demonstrate how our simple framework can handle this otherwise complex analysis in a simple and elegant fashion. To our knowledge, and unlike [5], where the approach is restricted to linear delay models, we are the first to handle, in linear-time, multi-corner STA with linear and nonlinear delay models alike. In order for our framework to hold, all we need to show is that the three properties of section 3 are satisfied. For illustration, we will demonstrate our approach for linear and quadratic models.

4.1 Linear and Nonlinear Models

Let \mathcal{F}_1 and \mathcal{F}_2 be the sets of linear and quadratic delay models respectively. Let A be a timing quantity such that:

$$A = \begin{cases} a_o + \sum_{i=1}^p a_i X_i & \text{if } A \in \mathcal{F}_1 \\ a_o + \sum_{i=1}^p (a_i X_i + \hat{a}_i X_i^2) & \text{if } A \in \mathcal{F}_2 \end{cases} \quad (17)$$

where a_o is the nominal delay of A , and a_i and \hat{a}_i are first-order and second-order sensitivities to X_i . Note that X_i 's are arbitrary variation sources (random/uncertain variables) that are bounded, and without loss of generality, assume that $-1 \leq X_i \leq 1$.

It can be easily shown that property 1 holds, i.e., both \mathcal{F}_1 and \mathcal{F}_2 survive linear operations. In fact, if $[A, B] \in \mathcal{F}_1$, then $C = aA + bB + c$ also $\in \mathcal{F}_1$. The same applies for \mathcal{F}_2 . Property 2 also holds, since X_i 's are bounded. As for property 3, both the linear and the quadratic model can be easily maximized and minimized in $\mathcal{O}(p)$ time, where p is the number of process/environmental parameters; if $A \in \mathcal{F}_1$, then:

$$A_{\min} = a_o - \sum_{i=1}^p |a_i| \quad \text{and} \quad A_{\max} = a_o + \sum_{i=1}^p |a_i| \quad (18)$$

whereas, if $A \in \mathcal{F}_2$, then maximizing and minimizing A over the space of variations requires maximizing and minimizing the individual quadratic terms, $g(X_i) = a_i X_i + \hat{a}_i X_i^2$. In the Appendix, we show how this can be done analytically depending on whether $g(X_i)$ is monotone in $[-1, 1]$ or not. Therefore maximizing and minimization A is linear in the number of parameters, since there are p quadratic terms.

Given that all the required properties are satisfied by \mathcal{F}_1 and \mathcal{F}_2 , the methods described in sections 3.2 and 3.3 for resolving the *max* operation can be applied to demonstrate multi-corner STA with linear and nonlinear delay models. The timing graph is traversed topologically, and our bounding/approximation scheme is applied to handle the *max* operation at every stage. This results

Table 1: Our methods compared to Corner Analysis

Circuit	Lower bound		Upper bound		LS Approx	
	Min	Max	Min	Max	Min	Max
c432	0.941	0.959	1.067	1.032	1.006	0.999
c499	0.930	0.966	1.026	1.022	0.982	0.996
c880	0.935	0.961	1.103	1.059	1.018	1.012
c1355	0.928	0.946	1.153	1.097	1.041	1.030
c1908	0.945	0.960	1.122	1.077	1.031	1.016
c2670	0.929	0.951	1.037	1.013	0.983	0.982
c3540	0.932	0.951	1.070	1.037	1.001	0.993
c5315	0.936	0.949	1.115	1.057	1.022	0.999
c6288	0.912	0.932	1.183	1.120	1.051	1.031
c7552	0.942	0.953	1.153	1.082	1.042	1.012
Avg Error	-6.7%	-4.7%	10.3%	5.9%	1.8%	0.7%

in expressing the arrival time at the sink node, i.e., the maximum circuit delay, using the delay model at hand. Once this expression is obtained, it can be easily maximized and minimized as shown in this section to bound/approximate the maximum and minimum corner delays of the maximum circuit delay.

4.2 Results

We have tested our approach by first characterizing a 90nm CMOS library using HSPICE in order to determine the sensitivities of gate delay to various process parameters. We have chosen to vary channel length (L_n and L_p), and threshold voltage (V_{tn} and V_{tp}) of NMOS and PMOS transistors, and have determined the sensitivities of gate delays to these process parameters. The variation information was specified in the technology files. Our technique was then implemented using C/C++ and was tested on the ISCAS85 benchmark circuits. For each circuit, the maximum circuit delay is determined at various corners using exhaustive corner case analysis, and the true maximum and minimum corner delays are recorded. Our framework is then applied, using (1) the lower bound technique, (2) the upper bound technique, and (3) the least squares approximation, to predict the maximum and minimum corner delay for every circuit. The results are summarized in Table 1, where the values shown are normalized to the true minimum and maximum corner delays respectively, determined by corner case analysis. For example, the second column gives a lower bound on the minimum (over all corners) circuit delay, and the last column gives an approximation of the maximum (over all corners) circuit delay. The closer the values are to 1 the more accurate they are. Note that, unlike [5] where the method is only accurate in predicting the circuit delay at the maximum corner, we can accurately estimate the maximum circuit delay at both the minimum and the maximum corners, so that the spread of the maximum circuit delay is well-captured. In addition, our approach is not restricted to linear models. The last row of the table shows the average percent error for every approach; the maximum and minimum corner delays can be approximated up to 0.7% and 1.8% respectively.

5. NONLINEAR NON-GAUSSIAN SSTA

We now demonstrate nonlinear non-Gaussian SSTA, using a general quadratic delay model that depends on process variables modeled as RVs with arbitrary distributions, as well as uncertain non-random parameters. As has become typical in the SSTA literature [1], we assume that one can deal with within-die correlations using some simple scheme by which correlation is resolved based on the physical location of a cell on the layout surface. As a result, when it comes to those variations which are random variables (as opposed to the uncertain non-random parameters), we assume that we are only dealing with global variations and purely random variations. Before proceeding with the details, we first show how the three properties of the delay model are satisfied in order for our framework to hold.

5.1 Delay Model

Assume that gate delay D is expressed using a quadratic model,

\mathcal{F} , as follows:

$$D = d_o + \sum_{i=1}^p (d_i X_i + \hat{d}_i X_i^2) + d_r X_{D_r} \quad (19)$$

where d_o is the nominal delay, d_i and \hat{d}_i are first-order and second-order sensitivities to X_i , and d_r is the sensitivity to the purely independent random variation X_{D_r} , specific to D .

Because our framework is indifferent to whether parameters are random or uncertain, there is no restriction on X_i 's, which can be either RVs or uncertain non-random parameters alike. If it is random, X_i is modeled as an independent zero mean RV with an arbitrary distribution, such that $-1 \leq X_i \leq 1$. This can be any distribution, including common cases such as a truncated Gaussian (normal) distribution, a uniform distribution, etc. If it is uncertain, X_i is simply assumed to vary in $[-1, 1]$. Note that, in both case, X_i 's are global variation sources that are shared by all gate delays and timing quantities. As for the purely independent random variation X_{D_r} , we will assume that it can be modeled as a standard normal distribution with zero mean and unit variance. This is because X_{D_r} are typically the result of various independent random effects that add up and converge to a normal distribution by the central limit theorem.

5.1.1 Add Operation

The *add* operation and generally any linear operation can be easily performed without destroying the above quadratic model. Assume that $[A, B] \in \mathcal{F}$:

$$A = a_o + \sum_{i=1}^p (a_i X_i + \hat{a}_i X_i^2) + a_r X_{A_r} \quad (20)$$

$$B = b_o + \sum_{i=1}^p (b_i X_i + \hat{b}_i X_i^2) + b_r X_{B_r} \quad (21)$$

then $C = aA + bB + c$ can be expressed using the quadratic model as follows:

$$\begin{aligned} C &= (aa_o + bb_o + c) + \sum_{i=1}^p \left[(aa_i + bb_i) X_i + (a\hat{a}_i + b\hat{b}_i) X_i^2 \right] \\ &+ (aa_r X_{A_r} + bb_r X_{B_r}) \\ &= c_o + \sum_{i=1}^p (c_i X_i + \hat{c}_i X_i^2) + c_r X_{C_r} \end{aligned}$$

where $c_r = \sqrt{(aa_r)^2 + (bb_r)^2}$, since X_{A_r} and X_{B_r} are independent standard normal RVs, and X_{C_r} is an independent standard normal RV specific to C . Therefore, \mathcal{F} satisfies property 1.

5.1.2 Max Operation

As explained in section 3, the *max* operation $C = \max(A, B)$ is resolved by first subtracting A and B , i.e., $D = A - B$ and then bounding/approximating C using linear combinations of A and B ; recall that the coefficients of the linear combinations are functions of the minimum and maximum values of the difference D , i.e., D_{\min} and D_{\max} .

We have already shown how to perform linear operations using the quadratic model. Hence, in order to apply the results of section 3, it remains to show how the quadratic model is bounded and can be efficiently maximized and minimized over the space of variations (properties 2 and 3). Assume that we have already performed the subtraction $D = A - B$, and D is given by:

$$D = d_o + \sum_{i=1}^p (d_i X_i + \hat{d}_i X_i^2) + d_r X_{D_r} \quad (22)$$

Since X_i 's and X_{D_r} are all independent, then the maximum D_{\max} and minimum D_{\min} of D are achieved when all variations are maximized and minimized separately. In other words, we need to maximize and minimize the quadratic terms and the independent random term.

The quadratic terms $g(X_i) = d_i X_i + \hat{d}_i X_i^2$ can be analytically maximized and minimized as shown in the Appendix. As for the purely independent random variation X_{D_r} , recall that it is modeled as an RV with standard normal distribution, which suggests that X_{D_r} can take values in $-\infty$ to $+\infty$. This, however, is not realistic, and it is usually the case that standard normal distributions are truncated between $[-k, k]$, where k represents multiple standard deviations. As an example, setting $k = 3$ will cover 99.73% of the standard normal distribution. As a result, the maximum and minimum values contributed by the independent random component are kd_r and $-kd_r$ respectively. Hence, by combining the above two contributions, D_{\min} and D_{\max} can be easily determined. Having shown that the three properties of section 3 are satisfied by \mathcal{F} , we use the approaches in sections 3.2 and 3.3 to resolve the *max* operation.

5.2 Complexity Analysis

We have shown that both *add* and *max* operations are resolved using simple linear operations involving the delay model; in addition, the coefficients of the linear combination needed for the *max* operation involve a subtraction and a maximization/minimization performed on the model. If p is the total number of variation sources, then performing an addition or a subtraction is $\mathcal{O}(p)$; also, performing a maximization/minimization of the model is $\mathcal{O}(p)$. This means that both *add* and *max* operations are linear in the number of variation sources. Therefore, the overall complexity of a block-based SSTA technique using the above operations is $\mathcal{O}(pn)$, where n is the circuit size.

5.3 Results

We have implemented our nonlinear non-Gaussian block-based SSTA technique using C/C++ and have tested it on the ISCAS85 benchmark suite. Similarly to [3], and as a proof of concept, we have generated the variation information randomly. We have chosen the coefficients of the quadratic delay model in such a way that every variation source causes 10% to 20% deviation in the nominal delay. In addition to the purely random variation which follows a truncated Gaussian distribution, 4 global variation sources X_i 's were used, each following either a truncated Gaussian, a uniform, or a triangular distribution as shown in Fig. 6. The accuracy of our technique is compared to Monte Carlo analysis with 10,000 runs. All delays reported are normalized to the nominal circuit delay. Fig. 7 shows a plot of the cumulative distribution functions (CDF) for benchmark c1355 for the case of variations with a uniform distribution, generated using our SSTA technique and compared to Monte Carlo simulation. Note that the bounds are valid and accurate, and the least squares (LS) approximation is very close to the true distribution. Table 2 compares the following timing metrics: the 95 and the 99 delay percentiles, and the ratio of standard deviation to the mean σ/μ , generated using the least squares approximation (LS-SSTA) to Monte Carlo analysis, for Gaussian, uniform, and triangular distributions. The results show that our approach is consistently accurate for all metrics and different distributions, with an average error less than 1%.

5.4 Summary

We have shown how our simple framework can be applied in the context of SSTA. The resulting approach is, to our knowledge, the first simple and accurate linear-time technique that is general enough to handle nonlinear delay models, random variations with arbitrary distributions, and uncertain non-random parameters. In the mixed case, where one is dealing with both random variables and uncertain variables, the overall (random) circuit delay and the interesting statistical metrics (like mean, variance, percentiles) are effectively functions of the uncertain variables. If this functional dependence is complex, then the desirable worst-case values (of the interesting statistical metrics) must be found by a process of search or optimization. This type of follow-up analysis is beyond the scope of this paper, and is the topic of our continued work under this project. For now, the above results were obtained at a specific setting (the nominal) of the uncertain variables. This does not diminish the value of the results, because one strength of our existing approach is that it

Table 2: Least Squares SSTA vs Monte Carlo analysis for Gaussian, Uniform, and Triangular distributions

Circuit	Gaussian Distributions						Uniform Distributions						Triangular Distributions					
	LS-SSTA			Monte Carlo			LS-SSTA			Monte Carlo			LS-SSTA			Monte Carlo		
	95% -tile	99% -tile	σ/μ %	95% -tile	99% -tile	σ/μ %	95% -tile	99% -tile	σ/μ %	95% -tile	99% -tile	σ/μ %	95% -tile	99% -tile	σ/μ %	95% -tile	99% -tile	σ/μ %
c432	1.20	1.30	10.7	1.20	1.30	10.6	1.37	1.45	18.4	1.37	1.45	18.4	1.26	1.36	13.2	1.26	1.36	13.3
c499	1.22	1.31	9.35	1.19	1.28	9.18	1.38	1.48	15.6	1.36	1.46	15.9	1.27	1.37	11.4	1.25	1.34	11.4
c880	1.20	1.28	8.98	1.19	1.27	9.07	1.35	1.46	15.1	1.34	1.44	15.1	1.25	1.34	11.0	1.23	1.32	11.0
c1355	1.17	1.24	7.86	1.16	1.23	7.76	1.30	1.39	13.4	1.30	1.38	13.6	1.21	1.29	9.67	1.21	1.28	9.78
c1908	1.19	1.28	9.95	1.19	1.27	9.75	1.35	1.46	16.8	1.35	1.46	16.7	1.24	1.34	12.2	1.25	1.34	12.1
c2670	1.22	1.32	11.7	1.21	1.32	11.5	1.39	1.49	19.8	1.40	1.48	19.8	1.28	1.39	14.3	1.28	1.39	14.2
c3540	1.22	1.32	9.52	1.20	1.29	9.64	1.37	1.45	16.2	1.36	1.43	16.4	1.28	1.37	11.7	1.25	1.35	11.7
c5315	1.22	1.33	11.6	1.22	1.32	11.6	1.39	1.48	19.8	1.40	1.48	19.9	1.28	1.39	14.3	1.27	1.38	14.1
c6288	1.21	1.30	9.56	1.19	1.28	9.50	1.37	1.46	16.3	1.36	1.45	16.5	1.26	1.35	11.8	1.25	1.35	12.0
c7552	1.22	1.33	11.7	1.22	1.33	11.7	1.40	1.48	19.8	1.39	1.48	19.7	1.28	1.38	14.3	1.27	1.38	14.2
AvgErr%	-0.80	-0.80	-0.51	-	-	-	-0.37	-0.54	0.55	-	-	-	-0.68	-0.59	0.04	-	-	-

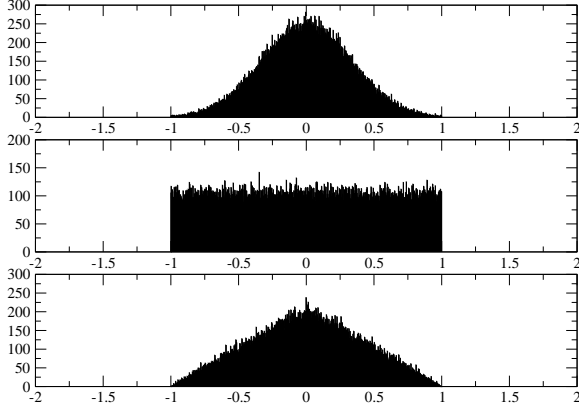


Figure 6: Truncated Gaussian, uniform, & triangular distributions

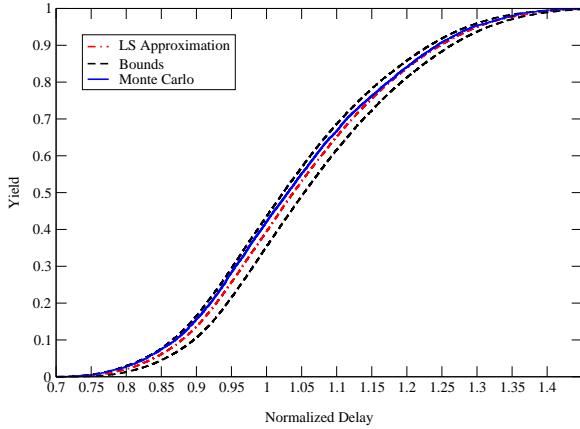


Figure 7: CDF comparison for c1355 with Uniform distributions

provides an *explicit* dependence of the total circuit delay on the uncertain parameters, so that one does not simply have to repeat the overall SSTA for different settings.

6. CONCLUSION

In this work, we have proposed a general parameterized timing analysis technique that can handle nonlinear delay models and account for delay variability due to both *random* process parameters with arbitrary distributions and *uncertain* non-random parameters which depend on the operating environment. Central to this technique is a novel and efficient method to resolve the *max* operator by bounding it and approximating it using linear models, while preserving the inherent nonlinearity of the delay model itself. We have tested our technique within two timing verification frameworks, namely multi-corner timing analysis and nonlinear non-Gaussian SSTA, and have shown that the complex-

ity of the approach is linear in both the number of process and environmental parameters and the size of the circuit. Our results show that, on average, all timing characteristics of circuit delay are predicted with less than 2% error for multi-corner analysis, and less than 1% error for SSTA.

7. REFERENCES

- [1] H. Chang and S. S. Sapatnekar. Statistical timing analysis considering spatial correlations using a single PERT-like traversal. In *ICCAD*, pages 621–625, San Jose, CA, November 9-13 2003.
- [2] H. Chang, V. Zolotov, S. Narayan, and C. Visweswariah. Parameterized block-based statistical timing analysis with non-gaussian parameters, nonlinear delay models. In *DAC*, pages 71–76, Anaheim, CA, June 13-17 2004.
- [3] L. Cheng, J. Xiong, and L. He. Non-linear statistical static timing analysis for non-Gaussian variation sources. *Proc. DAC*, pages 250–255, 2007.
- [4] V. Khandelwal and A. Srivastava. A general framework for accurate statistical timing analysis considering correlations. *DAC*, pages 89–94, 2005.
- [5] S. Onaissi. and F. N. Najm. A linear-time approach for static timing analysis covering all process corners. *ICCAD*, pages 217–224, 2006.
- [6] J. Singh and S. Sapatnekar. Statistical timing analysis with correlated non-gaussian parameters using independent component analysis. *DAC*, pages 155–160, 2006.
- [7] C. Visweswariah, K. Ravindran, K. Kalafala, S. Walker, and S. Narayan. First-order incremental block-based statistical timing analysis. In *DAC*, pages 331–336, San Diego, CA, June 7-11 2004.
- [8] Y. Zhan, A. J. Strojwas, X. Li, and L. T. Pileggi. Correlation-aware statistical timing analysis with non-gaussian delay distributions. In *DAC*, pages 77–82, Anaheim, CA, June 13-17 2005.
- [9] L. Zhang, W. Chen, Y. Hu, J. Gubner, and C. Chen. Correlation-preserved non-gaussian statistical timing analysis with quadratic timing model. *DAC*, pages 83–88, 2005.

APPENDIX

For completeness, we show the (simple) process by which one can maximize or minimize a quadratic function of the form, $g(x) = ax + bx^2$, where $-1 \leq x \leq 1$, and $[a, b] \in \mathcal{R}$. Recall that the minimum/maximum is at $x_v = -a/(2b)$. If x_v falls outside $[-1, 1]$ then $g(x)$ is monotone in $[-1, 1]$, so that its maximum and minimum occur at the domain boundaries:

$$\begin{aligned} \max_x g(x) &= \max(g(-1), g(1)) = |a| + b \\ \min_x g(x) &= \min(g(-1), g(1)) = -|a| + b \end{aligned}$$

On the other hand, if x_v falls in $[-1, 1]$, then $g(x)$ is not monotone in $[-1, 1]$, i.e., the maximum and minimum can be either at the vertex or at the boundaries of the domain:

$$\begin{aligned} \max_x g(x) &= \max(g(-1), g(1), g(x_v)) = \max(|a| + b, -a^2/4b) \\ \min_x g(x) &= \min(g(-1), g(1), g(x_v)) = \min(-|a| + b, -a^2/4b) \end{aligned}$$

These operations can all be done efficiently in constant time.