Use of Statistical Timing Analysis on Real Designs

A. Nardi, E. Tuncer, S. Naidu, A. Antonau, S. Gradinaru, T. Lin, J. Song
Magma Design Automation - Santa Clara, California
anardi,emre,srinath,aantonau,sgradina,tao,jhsong@magma-da.com

Abstract

A vast literature has been published on Statistical Static Timing Analysis (SSTA), its motivations, its different implementations and their runtime/accuracy trade-offs. However, very limited literature exists ([1]) on the applicability and the usage models of this new technology on real designs.

This work focuses on the use of SSTA in real designs and its practical benefits and limitations over the traditional design flow. We introduce two new metrics to drive the optimization: **skew criticality** *and* **aggregate sensitivity**.

Practical benefits of SSTA are demonstrated for clock tree analysis, and correct modeling of on-chip-variations. The use of SSTA to cover the traditional corner analysis and to drive optimization is also discussed. Results are reported on three designs implemented on a 90nm technology.

1 Introduction

Figure 1 describes a generic SSTA flow. The intent here is to describe the basic components (inputs, analysis engine, and outputs) of SSTA, and provide practical considerations.

1.1 Inputs

In addition to the traditional information required for deterministic timing analysis, the following information is needed for SSTA.

Process variations are provided by the foundry and represents the process parameters to be considered as statistical rather than deterministic. Process variations affect device parameters and interconnect dimensions, such as width, thickness, interlayer dielectric thicknesses, etc. The process information is usually provided in the form of a Statistical Spice model. Process variations are usually categorized as *Global* and *local* variations. The former includes die-to-die, wafer-to-wafer and lot-to-lot variations, while the latter addresses within-die variations (gate-to-gate and transistor-to-transistor). Additionally, for local variations, different parameters have different behaviors, e.g.: oxide thickness mostly has *systematic* variations, while the number of dopants may have a completely *random* behavior.

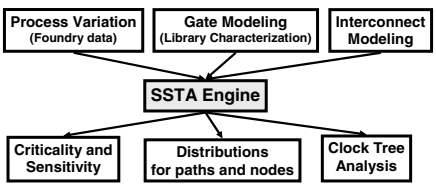

Figure 1. Generic SSTA flow

Both global and local variations need to be accounted for by SSTA.

Gate and Interconnect Models capture the sensitivity of devices and interconnects to process variations. Different models can be used: the trade-off is, as usual, between accuracy and runtime/memory.

1.2 SSTA Engine

Several approaches have been presented to implement the SSTA engine. They can be categorized into path based and block based approaches [6, 2]. A complete discussion on the pros and cons for the two approaches is beyond the scope of this paper. For all the approaches, however, choices have to be made on how to deal with correlations, Gaussian distribution assumptions, statistical min/max operations, slew/capacitance variation effects and environmental variations.

1.3 Outputs

The SSTA engine computes the delay/slack probability density function for all design components (paths, nodes) and for the design itself. This information can be used to explore the performance/yield trade-off during circuit design. Additionally, new criteria are needed to define the criticality of paths in a statistical world [6, 5]. We show how to combine path criticality and delay sensitivity to determine variability bottlenecks and drive statistical optimization (see Section 6).

1.4 Usage of SSTA in the design flow

In the traditional design flow, local variations are accounted for by adding margins for guardbanding. Not only can these margins be unrealistic (either too pessimistic or

978-3-9810801-2-4/DATE07 © 2007 EDAA

even optimistic), this approach also does not help in improving the design robustness to variations since the real distribution (and hence the sensitivity to variations) is not available. SSTA has the potential for improving the design robustness to variations, thus reducing wasted design effort due to unrealistic guardbanding while guaranteeing all violations are correctly captured.

Global variations instead are traditionally taken into account by performing timing analysis in different process and environmental (PVT) corners, which are usually described by different libraries for standard cells and different parasitic rules for interconnect. The number of corners to be considered can be quite large: the design has to meet the performance constraints in all combinations of PVT and RC corners. The number of corners in turn is often compounded to the number of design modes thus reaching hundreds of unique scenarios to consider. This can be problematic because corners have to be analyzed and optimized concurrently to avoid many iterations in the fixing flow. SSTA offers at least a partial solution to this problem.

In summary, SSTA facilitates design closure by reducing pessimism wrt guardbanding, and provides new metrics for assessing robustness of designs. Furthermore, it eliminates or reduces the need of analyzing and optimizing a large set of design corners.

In this paper we test some of these concepts of SSTA on real designs, and propose new uses.

Section 2 gives a very brief overview of the new metrics used in SSTA, mentions the challenges of implementation and categorizes possible approaches. Section 3 extends the concept of criticality to statistical skew reporting for robustness analysis of clock trees. Section 4 analyzes how SSTA can model properly on-chip-variations, while Section 5 briefly overviews the advantages and limitations of SSTA to handle multiple design corners. Section 6 introduces a new metric (aggregate sensitivity) to combine criticality and sensitivity information and drives statistical optimization.

Table 1. Statistical clock skew reporting for RISC OR1200 on 90nm technology.

Pair	Mean Skew	Skew Std. Dev.	Skew Criticality
1	51.273	15.722	34.7
2	51.334	12.554	25.6
3	24.338	16.898	7.9
4	43.479	4.358	4.7
5	43.378	8.759	4.5
6	38.684	14.081	4.3
7	34.733	15.964	3.9
8	34.666	13.523	3.1
9	25.95	13.396	2.3
10	22.834	17.619	2
11	20.525	15.726	2
12	33.202	14.888	1.3
13	28.222	11.846	1.2
14	19.439	14.455	0.7
15	18.917	14.222	0.4
16	26.535	18.316	0.4
17	20.824	15.885	0.2
18	20.706	12.131	0.2
19	21.107	15.712	0.2
20	33.246	24.379	0.1
21	20.834	15.762	0.1
22	17.328	12.665	0.1
23	22.495	14.138	0.1

2 SSTA Basics

Before reviewing the different categories of SSTA algorithms and the new metrics, we first discuss the main challenges. These include considering correlations properly and propagation of distributions under min/max operations.
Delay/Slack correlation can be due to correlation between process parameters, or to path sharing. Statistical min/max operation is not straightforward and might be runtime expensive or require approximations. The trade-offs between analytical and numerical solution have been amply discussed in [3, 4, 2].

2.1 Algorithms

SSTA implementations can be categorized into two main groups, block based ([6]) and path based ([2]). In the block based SSTA, arrival time distributions are calculated at each node as the min or max of the incident arrival times. Usually, normal distributions are assumed for complexity reasons. Both parameter based and path sharing based correlations are expensive to handle in this approach and therefore simplifications are introduces at the expense of accuracy.

In the path based approach, the analysis is performed on selected paths. The main advantage is the accuracy and the flexibility of trading accuracy for runtime/memory. The main drawback instead is that the number of paths required for accurate results can be theoretically very large. In practice, it is observed that at most 100,000 paths are need to capture circuit behavior and give accurate results.

A further classification can be done into numerical and analytical approaches. For example, Monte Carlo analysis allows accurate numerical estimation of the min/max output distribution but is more runtime expensive than a completely analytical calculation.

This paper does not further discuss the comparison between these approaches. In fact, the goal of this work is to propose examples of applications for SSTA, regardless of the choice and details of implementation.

2.2 Metrics

Traditionally, optimization targets the most critical paths/gates. The path with the most negative slack is the most critical one. When performing SSTA, the slacks are random variables and the definition of worst slack is not intuitive anymore.

Many mathematical formulations have beed published to define criticality and sensitivity ([6, 5]). Although there is certainly overlap in the concepts involved, we present our own definitions for the sole purpose of intuitive understanding of the concepts to drive the applications.

Criticality of a path represents the probability that the specific path will be the one limiting the circuit performance (that is, the one with the most negative slack). To clarify the concept, if using Monte Carlo analysis for example, the crit-

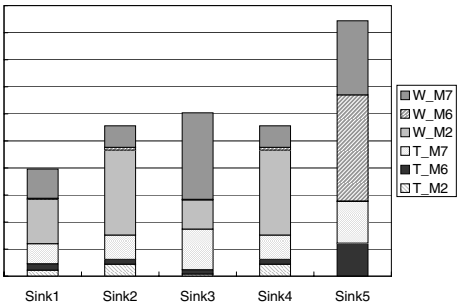

Figure 2. Standard deviation of the arrival time for five critical clock sinks in OR1200.

icality of a path can be calculated by counting the number of samples for which the specific path has the worst slack divided by the total number of samples.

Sensitivity of a path/gate performance represents the amount of performance variation due to a given amount of process variations.

While the sensitivity of a path is a property of the path in isolation, criticality of a path is a property of the whole design (or set of paths): it cannot be defined for the path in isolation. Both metrics are computed by SSTA and are required to drive optimization. Examples are in Section 6.

3 Clock Tree Analysis

The construction of a well-balanced clock tree is a key step in the design of an integrated circuit. Process variations complicate the problems: results in this Section show that a clock tree well-balanced in the nominal corner is not necessarily a clock tree robust to variations. Clock skew distribution and skew criticality can be used to optimize robustness to process variations.

3.1 Skew Criticality

Traditionally, the quality of a clock tree is measured by the maximum skew. Given a pair of registers (a *skew pair*), their clock skew is the difference of the clock signal arrival time at their clock pins. The maximum clock skew among all skew pairs defines the quality of the clock tree (like the minimum slack defines the quality of the design).

Note that also for the clock tree analysis, a statistical max operation has to be performed.

In this paper, we extend the definition of critical path to the definition of critical skew pair: it is the one with the highest probability of having the largest skew.

As for the delay and the slack, in addition to the mean value, SSTA also provides the standard deviation for the specific path (or skew pair).

Table 1 reports the statistical skew analysis on the clock tree of the OR1200 RISC core mapped onto a 90nm technology considering only interconnect variations. In particular, metal width and thickness are random variables for three of the metal layers. Skew pairs are listed in decreasing order of skew criticality until the zero value is reached.

There are a few interesting observations from these data:

- Traditionally, STA lists only the worst skew pair in the design (for each clock). In fact, reporting all the skew pairs can be very expensive. However, SSTA can efficiently report all the skew pairs with non-zero skew criticality.
- Two pairs having the same nominal or mean skew, may show a significant difference in their standard deviation. For example, Pair 4 and Pair 5, show that a well balanced clock in the deterministic case may not be robust to variations which can make the clock poorly balanced.

Note that SSTA has reported a skew pair that would be neglected by STA: Pair 3 has a 7.9% probablity of being critical, although its mean value (24.338) is smaller than other skew pairs (most notably Pair 20 with mean value 33.246 but with criticality 0.1%).

3.2 Sensitivity Analysis

SSTA also calculates the contributions due to different parameters: for example device versus interconnect, or for interconnects, the sensitivity to different metal layers. In Figure 2 each bar represents the standard deviation of the insertion delay for five clock sinks among the critical ones in the OR1200 design. The sources of variations are the width (W) and thickness (T), on three metal layers: M2, M6, and M7: the standard deviations are decomposed in the contribution of the different sources of variations. For example, Sink 2 and Sink 4 track each other very well: not only their mean value is comparable but also their sensitivity to the various parameters is very similar. That is, the paths to these two clock sinks are highly correlated and therefore they are very well balanced both in the deterministic and in the statistical analysis (the mean and sigma for their skew are very small). On the contrary, although Sink 3 and Sink 4 have similar mean value, their sensitivity to the various parameters is quite different: the sink pair is well-balanced in STA but SSTA identifies a large skew distribution.

Different optimization strategies can then be used for clock tree robustness: for example, balancing the variation due to device and to interconnect parameters, or balancing routing of the clock nets among metal layers. While the exhaustive coverage and discussion of clock tree optimization strategies is beyond the scope of the paper, it is evident that SSTA is a powerful enabler for designing robust clock trees.

4 Guard-Banding vs SSTA

SSTA is the correct way of modeling effects that have been traditionally dealt with by using guard-banding.

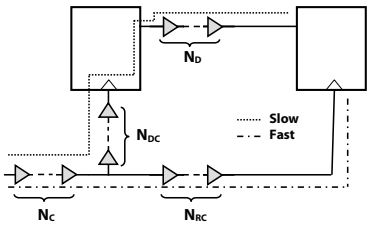

Figure 3. OCV approach for a setup check.

4.1 Guard-Banding

Before the introduction of SSTA, different approaches have been used to guard-band the circuit performance variations due to intra-die process variations.

Figure 3 illustrates a common approximation to account for intra-die variations when performing a setup analysis:

- Gates and wires along the launching clock and the data path (dotted line) are assumed to exhibit the slowest possible delay
- Gates and wires along the capturing clock (dash-dotted line) are assumed to exhibit the fastest possible delay
- To reduce pessimism, the difference between slow and fast delays is removed for the gates and wires shared by the launching and capturing clocks.
- Commonly the slowest (and the fastest) possible delay values are computed with a derating factor wrt a common operation point.

In commercially available tools, this guard-banding solution is usually referred to as OCV (On-Chip-Variation), actually confusing the problem with the solution.

Clearly, the above approximation does not properly model neither random nor systematic intra-die variations:

- Gates along the launching path are assumed to be completely correlated (and similarly for the capturing path) thus ignoring the stochastic cancellations that might take place when modeling random variations
- Perfect negative correlation is assumed between the launching path and the the capturing path, thus overestimating the effect of spatial correlation
- The same derating factor is used for all cells/arcs/transitions/slew-load scenarios. In reality, the delay/slew sensitivity to process variations depends on all the above conditions. Using just one derating factor can be very pessimistic or even miss some real violations. Usually the derating factor is chosen to guard-band conservatively process variations and most often leads to significant pessimism.

This methodology has severe accuracy limitations and also can be very computationally expensive due to the calculation of the common point between the launching and the capturing clock. OCV use a derating factor to model delay

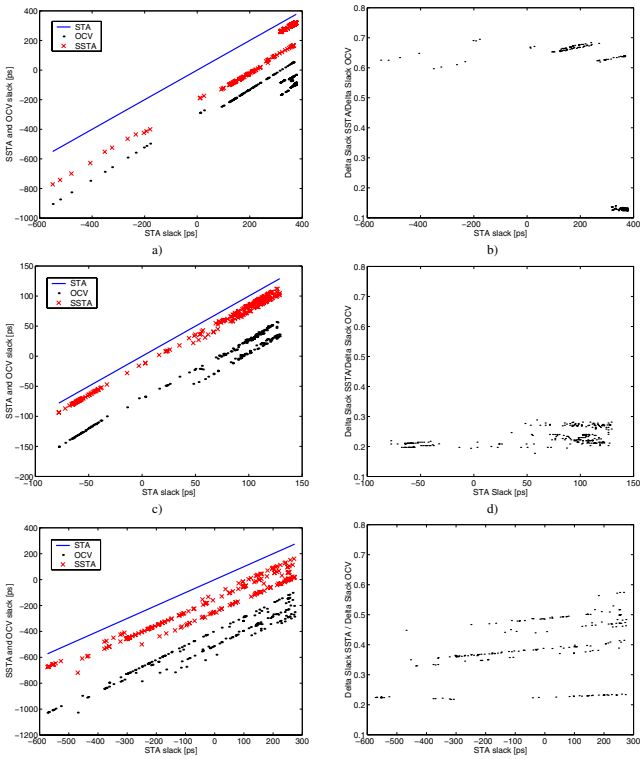

Figure 4. Guard-Banding (OCV) versus SSTA on a set of paths from the OR1200 design (a) and b)), an industrial design of approx 40k cells (c) and d)) and the ARM core (e) and f)).

sensitivity to process variations. For SSTA such sensitivity can either:

- represent a deration as well: the values are chosen such that, for example, the 3σ point of the delay distribution corresponds to the derating factor for OCV
- be derived from library characterization and depend on the gate, arc, transition and slew-load condition

Section 4.2 shows that, using the first approach, OCV is always pessimistic with respect to SSTA and derives a formula to model such phenomenon. Section 4.3 instead uses the second approach and shows how the choice of a single derating factor for OCV can be either overly pessimistic or too optimistic (and therefore miss timing violations).

4.2 Pessimism Reduction

For both SSTA and Guard-banding analysis, the net effect of intra-die variations is an additional delay ΔD to the nominal delay D_{nom}:

$$\Delta D_{SSTA} = D_{SSTA} - D_{nom} \quad (1)$$

$$\Delta D_{OCV} = D_{OCV} - D_{nom} \quad (2)$$

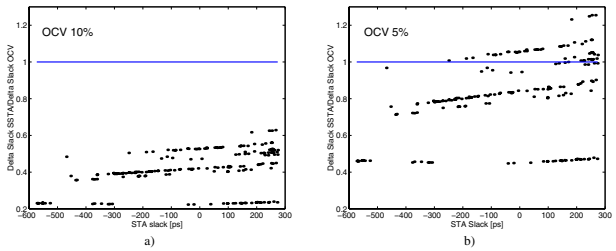

Figure 5. SSTA vs OCV using a) 10% and b) 5% deration. While 10% OCV is overly pessimistic, 5% OCV can miss some violations.

For Gaussian distributions, ΔD_{SSTA} is typically calculated as the 3σ point.

For the simple case in Figure 3, it can be shown that the reduction of pessimism expected when using SSTA to model random variations with respect to using OCV is:

$$\frac{\Delta D_{SSTA}}{\Delta D_{OCV}} = \frac{1}{\sqrt{N}} \tag{3}$$

where $N = N_{DC} + N_D + N_{RC}$ is the number of all gates not belonging to the common path. The meaning of N_{DC}, N_D, N_{RC} is illustrated in Figure 3.

The formula has been derived in a very simple case, where all delays are identical and process variations are a percentage of the nominal delay. However, it gives an indication of the expected behavior also in real testcases.

Results from three testcases mapped on a 90nm technology are reported in Figures 4.

Figures 4.a, 4.c and 4.e report the slack calculated with SSTA (crosses) and with OCV (dots) versus the nominal slack (x-axis and solid line for reference): the slack gets smaller due to process variations, and OCV calculation is much more pessimistic than SSTA values.

Figures 4.b , 4.d and 4.f represent $\frac{\Delta D_{SSTA}}{\Delta D_{OCV}}$: the smaller the ratio the larger the advantage of using SSTA vs OCV. The design in the first row of Figure 4 clearly shows the behavior predicted by Equation (3): the paths with ratio approx. 0.12 have a very large number of gates not in the common path (approx. 60) and a predicted ratio of 0.129. Those correspond to the points in Figure 4.a for which the difference between ΔD_{SSTA} and ΔD_{OCV} is more evident. Conversely, the paths for with ratio approx. 0.65 have a very small number of gates not in the common path (approx. 3) and a predicted ratio of 0.577.

Different heuristics have been devised to reduce the pessimism implied by OCV. However, these heuristics just struggle to get closer to what is very naturally captured by SSTA. Even when accuracy is improved by using smarter variants, the computational complexity (or the memory) remains a bottleneck for OCV. Although the discussion focused on the random local variations, similar observations apply to modeling systematic variations.

4.3 Guard-Banding Inaccuracy

Using library characterization to model the slew and delay sensitivity to process variations it becomes evident that this sensitivity is not just a derating factor. Not only it varies according to the gate type, but it also depends on the timing arc, on which transition (rising or falling) and which input slew and output load are applied. Moreover, usually more than one statistical parameter is defined to model process variations and different gates, arcs, transitions, etc have different sensitivity to each parameter, therefore, the single derating factor chosen for OCV analysis could be too pessimistic for some gates and too optimistic for others.

Results have been collected from an ARM core mapped onto a 90nm technology. Process variations are represented by four parameters and library characterization has been run to collect the sensitivity data. OCV for comparison uses a derating factor of 10% (Figure 5.a)) and 5% (Figure 5.b)). The graphs show the ratio between the slack calculated with SSTA and the slack calculated with OCV versus the nominal slack. For this design, OCV at 10% is always overly pessimistic wrt SSTA. However, if 5% is used as derating factor, OCV becomes less pessimistic for some paths, but for other paths is too optimistic and does not report a violation that is instead caught by SSTA.

In summary, within-die variations can be elegantly modeled using SSTA and this technology will replace the current guard-banding approaches: both runtime/memory and accuracy are greatly improved.

5 Corner vs SSTA

Standard cell libraries are traditionally characterized at different process and environmental conditions (PVT corners) to capture the effect of global variations and different operating conditions. Similary, wire variations are modeled by preparing rules for parasitics extraction in different process corners (RC corners).

The design has to be anlyzed and optimized in all combinations of PVT and RC corners. If the optimization is performed sequentially, fixing one corner poses the risk of creating violations in another corner. This can be a very lengthy process and solutions are being pursued to perform the optimization concurrently in all corners.

Furthermore, although the design corners are supposed to represent the worst conditions for the design, they might not provide exhaustive coverage of the variation space.

SSTA inherently builds a parametric model of process variations and thus guarantees to cover exhaustively the variation space. All the process corners can be analyzed at once, while the environmental corners are still analyzed separately: the number of corners to be considered has been greatly reduced.

The SSTA framework can be extended to also support

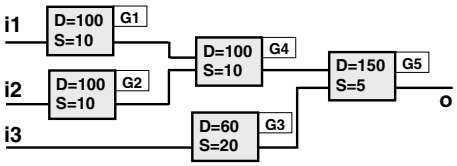

Figure 6. Circuit to illustrate the metrics for optimization.

environmental variations: the topic is not discussed here in the interest of space.

6 Optimization

In this Section we use a very simple example to show how to combine criticality and sensitivity (see Section 2.2) into a new concept of *aggregate sensitivity* to drive the statistical optimization, whose goal is to improve the probability that the circuit slack is positive.

For a given cell and process parameter, we define the Aggregate Sensitivity $AS = \sum_{arcs} s \cdot c$ where s is the sensitivity for the arc and c is the criticality for the path to which the arc belongs. If the arc belongs to more than one path, the criticality is properly added for all the paths.

Intuitively, the idea is that a path/gate with a large spread (that is, a large sensitivity) does not need to be optimized for robustness if it is not a critical path/gate. Conversely, if a gate has a small sensitivity but it belongs to a large number of critical paths, then it should be optimized.

Table 2.a shows the SSTA report for the circuit in Figure 6. The values inside each gate are the delay D and the sensitivity S. Note that the path from $i3$ to o has the largest standard deviation (and sensitivity), but its criticality measure is zero.

Table 2. SSTA a) and aggregate sensitivity b) report for circuit in Figure 6.

Start End	Mean	Std. Dev	Crit.
i2 - o / FF	-160	15.0	51.3
i1 - o / FF	-160	15.0	48.7
i2 - o / RR	-160	15.0	0.0
i1 - o / RR	-160	15.0	0.0
i3 - o / FF	-20	20.6	0.0
i3 - o / RR	-20	20.6	0.0

a)

Cell	Agg. Sens.	Slack	Max Sens.
G4	10.0	-160	10.0
G2	5.1	-160	10.0
G5	5.0	-160	5.0
G1	4.9	-160	10.0
G3	0.0	-20	20.0

b)

Table 2.b reports the aggregate sensitivity metric for the circuit in Figure 6. Traditionally, gates would be listed for optimization according to their worst negative slack: SSTA can instead pass the gates to the optimizer according to their aggregate sensitivity value. Cells with a high aggregate sensitivity should have a higher priority for optimization. For the simple example in Figure 6, the cell to be optimised first would be $G4$, since all of the critical paths in the design pass through that cell, and $G4$ has a non-negligible sensitivity.

7 Conclusions

The main focus of this work is to present some of the possible uses of Statistical Static Timing Analysis (independently of its implementation) in a design flow, using data from real designs when possible. Two new metrics are also introduced: skew criticality and aggregate sensitivity. The former is used during the statistical clock tree analysis. Our case study shows that a clock-tree well balanced in the nominal or mean case is not necessarily robust to process variations. Interestingly, the statistical clock tree analysis has reported a critical sink pair that would be neglected by the STA. Aggregate sensitivity encompasses the information of both criticality and sensitivity to drive the statistical optimization. Furthermore, for the three designs, it has been shown that SSTA can be significantly more accurate than approaches based on a single derating OCV factor (guardbanding). In fact, a single derating OCV factor can be either very pessimistic on some paths, or too optimistic on others, thus missing timing violations that are reported by SSTA. A simple model to estimate the pessimism reduction has also been reported and compared to the case study data. In summary, this work presents some of the benefits and limitations of SSTA when applyed to real designs, and demonstrates new uses for the SSTA engines. We will report on new findings as we keep applying this new technology to more designs.

References

[1] A. Agarwal, V. Zolotov, and D. Blaauw. Statistical clock skew analysis considering intradie-process variations. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 23(8):1231–1242, August 2004.

[2] C. S. Amin, N. Menezes, K. Killpack, F. Dartu, U. Choudhury, N. Hakim, and Y. I. Ismail. Statistical static timing analysis: how simple can we get? In *Design Automation Conference Proceedings*, pages 652–657, June 2005.

[3] C. E. Clark. The greatest of a finite set of random variables. *Operations Research*, 9:85–91, 1961.

[4] J. Le, X. Li, and L. T. Pileggi. Stac: Statistical timing analysis with correlation. In *Design Automation Conference Proceedings*, pages 343–348, June 2004.

[5] X. Li, J. Le, M. Celik, and L. T. Pileggi. Defining statistical sensitivity for timing optimization of logic circuits with large-scale process and environmental variations. In *Proceedings of the International Conference on Computer-Aided-Design*, pages 843–851, November 2005.

[6] C. Visweswariah, K. Ravindran, K. Kalafal, S. G. Walker, and S. Narayan. First-order incremental block-based statistical timing analysis. In *Design Automation Conference Proceedings*, pages 331–336, June 2004.