Death, Taxes and Failing Chips

Chandu Visweswariah
IBM Thomas J. Watson Research Center
1101 Kitchawan Road, Route 134
Yorktown Heights, NY 10598
chandu@watson.ibm.com

ABSTRACT

In the way they cope with variability, present-day methodologies are onerous, pessimistic and risky, all at the same time! Dealing with variability is an increasingly important aspect of high-performance digital integrated circuit design, and indispensable for first-time-right hardware and cutting-edge performance. This invited paper discusses the methodology, analysis, synthesis and modeling aspects of this problem. These aspects of the problem are compared and contrasted in the ASIC and custom (microprocessor) domains. This paper pays particular attention to statistical timing analysis and enumerates desirable attributes that would render such an analysis capability practical and accurate.

Categories and Subject Descriptors

B.7.2 [**Hardware**]: Integrated circuits—*Design aids*

General Terms

Algorithms, verification

Keywords

Statistical timing, parametric yield prediction, design methodology.

1. INTRODUCTION

For almost two decades, conventional static timing [1] has proved to be a reliable and efficient method for timing sign-off of digital integrated circuits. Incremental static timing [2, 3] is a key enabler of circuit optimization during logic synthesis and physical synthesis. In recent years, the static timing paradigm has been enhanced to accommodate such deep sub-micron effects as coupling noise, RC and RLC interconnect models, simultaneous switching and more accurate waveform propagation. Chip-to-chip variation has been traditionally handled by case analysis, and across-the-chip variation by *heuristic derating factors* which slow down the data relative to the clock for late-mode analysis, and vice versa. This is implemented as a linear combination of delays (as in IBM's EinsTimer) or the on-chip variation mode (or multiple operating condition mode or delay derating feature) of Synopsys's PrimeTime [4].

There are three main reasons why this paradigm is breaking down.

1. The first is that critical dimensions are scaling faster than our control of them. Thus, the variability of physical dimensions, such as the effective length of a transistor channel, is *proportionately increasing* [5].

2. In previous technologies, variability was dominated by the Front-End-of-the-Line (FEOL), or active transistors and gates. It was reasonable to assume that the dominant sources of variation were strongly correlated, and therefore case analysis with relatively few process corners provided high coverage confidence. With recent technology generations, the Back-End-of-the-Line (BEOL) or interconnect metalization has shown large variability, too. These sources of variability are relatively uncorrelated to the former, and relatively uncorrelated from one metal level to another, so the number of *significant and independent* sources of variation has dramatically increased. Concomitantly, the number of cases or corners required for confident coverage has grown tremendously.

3. *Across-the-chip Linewidth Variation* (ACLV), caused mainly by reticle and proximity effects during lithography [6] and by local density effects, is increasing with each new generation of technology. Of course, temperature and power supply gradients can also be significant across regions of the chip.

If we accept the notion that timing verification will be performed statistically, then we are presented with a unique opportunity to treat other phenomena statistically as well.

1. *Model-to-hardware correlation* and other modeling and analysis errors have been long-known problems. The inaccuracy of our models could be treated statistically in order to reduce pessimism. Just one example is that if the same identical gate is used in a data path and the corresponding clock path that latches it, any modeling inaccuracy cancels out and the design should be given "credit" instead of applying conservative models in both cases.

2. *Coupling noise* and other types of noise analysis can be integrated into a unified timing verification environment in a probabilistic manner, rather than assuming that every neighboring net of every critical path switches in the worst-case manner in the worst-case switching window in every cycle of operation.

3. Timing the chip with *aging and fatigue effects* (such as Negative Bias Temperature Instability (NBTI), hot electron effects

DAC 2003, June 2–6, 2003, Anaheim, California, USA.

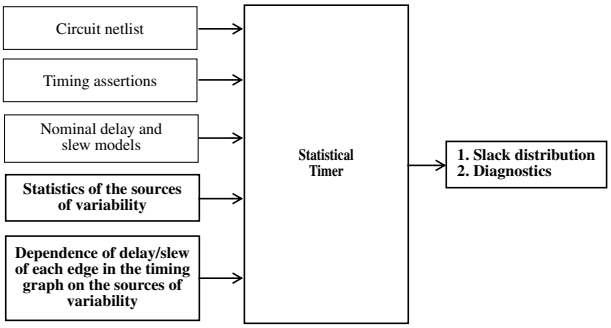

Figure 1: Block diagram of a statistical timer.

and electro-migration) as well as special timing runs to incorporate coupling noise are now mandatory for timing sign-off of integrated circuits. We are presented with an opportunity to reduce the number of sign-off timing runs by applying statistical techniques.

While none of this is particularly new, the number and magnitude of these effects is increasing, and worst-casing all of them is simply not practical any more. Simultaneously, time-to-market pressure is precluding exhaustive timing verification at an exploding number of combinations of cases and static timer settings. Thus we are at a situation in which not only is the timing verification effort too burdensome, it is *both pessimistic and risky* at the same time. It is pessimistic as is the nature of bounding methods that seek to deliver guarantees or bounds on the earliest and latest arrival times. Yet it is risky because it is not possible to conduct a bounding analysis at an exhaustive set of corners and cases. The solution to this problem is statistical static timing analysis, which is the main topic of this paper. *Statistical timing will simultaneously enable targeting of high-performance while providing quantitative risk management.* Implemented and applied correctly, it will reduce pessimism, improve verification turnaround time and provide means for increasing parametric yield.

It is important to note that yield loss can be classified as *catastrophic* or *parametric* (also referred to as circuit-limited yield loss). The former is due to dust particles and other contaminations during manufacturing that render the chip non-functional. The latter causes functioning chips to show a range of performance, and it is this latter type of yield loss that we are interested in predicting via statistical timing analysis.

When static timing methods evolved, they were primarily used for sign-off before fast incremental methods were developed for optimization purposes. In the case of statistical timing analysis, the same will be true, with accurate sign-off techniques being developed first and leading to subsequent incremental and fast methods, which in turn will enable statistical synthesis in the future.

2. WHAT IS A STATISTICAL TIMER?

A conventional static timing analysis program takes as input a circuit and builds a timing graph. The delay and slew (transition time) characteristics of each gate are either provided by means of delay models or computed on the fly by transistor-level time-domain simulation. The main output of the program is a final slack, from which the fastest safe clock frequency can be inferred. Additionally, the program can produce a timing report including lists of failed timing tests, arrival times, slacks, lists of critical paths, and so on.

In contrast, a statistical timer (Fig. 1) accepts additional information in terms of the sources of variation. The statistics of the sources of variation, including the type of probability distribution and variances and co-variances thereof, are produced by modeling the uncertainty in the integrated circuit manufacturing process. In addition, the static timer is provided or has means to compute the dependence of the delay and slew of each edge in the timing graph on the sources of variability. The main output of the program is the *probability distribution* of the slack. In cumulative distribution function (CDF) form, such a slack is shown in Fig. 2. The distribution in this example indicates that at a slack of -300 ps, the parametric yield of this circuit will be almost 100%, whereas at a slack of +200 ps, the yield quickly drops to almost 0%. Additionally, the statistical timer can provide probabilistic diagnostics such as slacks, critical paths and probabilities of timing tests being met at latches.

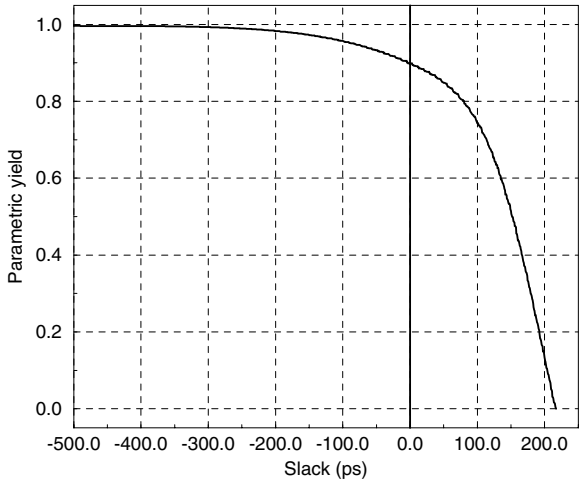

Figure 2: Sample slack distribution.

Having slack distribution information as in Fig. 2 has many benefits. In the case of at-speed-tested and binned microprocessor products, it allows the prediction of the percentage of chips that will fall in the high-speed high-profit bin. In the case of ASICs, it allows for early decision making on risk management at the chip and board levels. It also permits calculated targeting of higher clock frequencies, with at-speed testing used to separate the faster chips coming off the line, with a pre-computed guaranteed yield at the faster frequency.

Statistical timers can operate on a *block basis* or a *path basis*. Traditional propagation of arrival time and required arrival time through a timing graph is said to be on a block basis, since all the information required for the propagation is local to a particular gate. As we will see later in this paper, block-based algorithms are inherently linear in complexity (like traditional static timing), but constitute a poor platform for capturing correlations such as between a clock path and a data path. Path-based algorithms, while being in a much better position to capture such correlations, are inherently exponential and must appeal to several artifices to be tractable. Regardless of a block or path basis, there are two main numerical methods employed by statistical timers.

2.1 Performance-space methods

These methods work in the space of performance variables such as delays, arrival times, required arrival times and slacks [7, 8, 9]. In its simplest incarnation, the method can be used to find the probability distribution of the longest path delay of a block of combi-

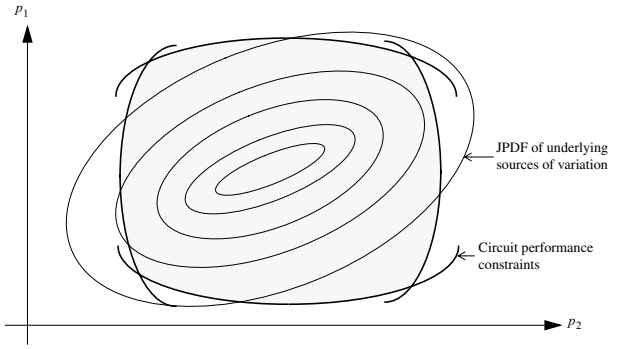

Figure 3: Integration of the JPDF of two sources of variation in the feasible region.

Table 1: Key ASIC and microprocessor methodology differences

Attribute	ASIC	Microprocessor
Test	Often no at-speed test	Sorted/binned
Methodology	Flat	Hierarchical
Circuits	Library-based	Custom + library cells
Timing	Focus on worst-case	Focus on nominal

national logic. In this case, if there are N paths, the problem is to find the probability of the longest of these paths having a delay value η, for all values of η. Depending on the problem at hand, we may be required only to compute the probability that all paths simultaneously satisfy a single clock cycle requirement, or we may be required to compute the entire curve as in Fig. 2. On a path basis, this computation can be thought of as the integration of an N-dimensional joint probability density function (JPDF) of path delays in an N-dimensional hypercube of side η.

As a practical matter, these methods do not integrate functions in a dimensionality equal to the number of paths, since that would be computationally prohibitive. Instead, they often treat timing quantities such as arrival time, delay and slack as probabilistic quantities and (approximately) propagate them through the network in a manner that is similar to traditional static timing analysis.

2.2 Parameter-space methods

Parameter-space methods work in the space of the sources of variation. Each circuit requirement such as delay is represented as a constraint in the parameter space, on one side of which the circuit is feasible and on the other side of which it fails. The intersection of all these constraints defines a *feasible region* in which the JPDF of the underlying sources of variation is integrated to come up with a yield prediction (see, for example, [10, 11, 12]). This method is illustrated in Fig. 3.

Mathematical techniques such as principal-component analysis or SVD-based reduction [13] can be applied to reduce the dimensionality of the sources of variation, with only a small loss in accuracy.

3. METHODOLOGY

Applying statistical methods is a four-pronged effort consisting of modeling, analysis, methodology and synthesis. Some methodology aspects will be briefly discussed in this section, and sections devoted to analysis, synthesis and modeling will follow.

Table 1 outlines some of the methodology differences between ASICs and microprocessors.

ASIC methodology is quite different from microprocessor methodology in many ways. Often, ASICs undergo IDDQ and functional test, but not at-speed test. In this case, one way to deal with parametric testing is to take measurements on strategically placed Performance-Sensitive Ring Oscillators (PSROs), and declare the chip to be useful if the ring oscillators satisfy certain pre-determined criteria. Alternatively, PSRO screening criteria and analysis methods must be simultaneously evolved to obtain the highest possible performance while managing risk. In this situation, the ultimate goal of a statistical timing methodology is to compute the *conditional probability* that the chip meets its performance goals given that the PSROs are within specifications. Thus ASIC sign-off is often based on worst-case timing, and in the absence of at-speed test, risk management is deferred to the board or system-level. The library-based timing models make it a little easier to produce statistical delay models.

Microprocessors, on the other hand, are sorted and binned, with the chips in the highest frequency bin being sold for the highest profit. The sorting is on the basis of at-speed testing, which is a difficult engineering task since we are simultaneously trying to minimize time on the tester, maximize the frequency assigned to each chip and minimize the probability that a chip is mis-binned. Clearly, custom transistor-level circuitry must be accommodated in the analysis of such integrated circuits. These chips are designed hierarchically to manage complexity and to enable large development teams to work in parallel. What this means is that parametric yield predictions must also be conducted in a hierarchical manner. Thus statistical timing of individual macros will produce *statistical abstractions* including covariance information for use in the global chip-level yield predictions.

In either case, it is tempting to treat environmental variables (like temperature or power supply voltage) statistically. But such sources of variation are inherently different in nature. This is because a chip should be considered functional at a certain frequency only if it meets that frequency target *anywhere* in the specified environmental window. Thus environmental variables are ideally treated in a semi-infinite or bounding fashion.

In either ASIC or custom digital design, the number of timing runs required for sign-off is quickly growing prohibitive, which leaves an opportunity for statistical methods to get a foot in the door.

4. DESIRABLE ATTRIBUTES OF A STATISTICAL TIMER

This section describes the required attributes of a statistical timer in order for such a timer to be practical.

4.1 Correlations, correlation, correlations

Taking correlations into account is a crucial capability of a statistical timer [12]. A simple example will demonstrate this point. Suppose the chip has 50,000 latches, each with a setup and hold test. Assume that each of the 100,000 tests has a 99.99% probability of passing. If all the tests are perfectly correlated, the parametric yield of the circuit is 99.99%. If the tests are independent, the yield is 0.005%! The truth, while somewhere in-between, is thankfully closer to the former than the latter.

There are many kinds of correlations that must be considered. The first is correlations that arise due to data paths sharing some gates along the way, otherwise known as the *reconvergent fanout*

problem. The approach in [14, 15] suggests a method of taking such correlations into account on a block basis.

The second type of correlation arises due to commonality between data and clock. A launching and capturing path, for example, may go through the same gates. In this case, it would be pessimistic to assume a faster gate for the clock than the data in late mode, and vice versa in early mode. The most obvious such forms of pessimism are currently removed by post-process limited path tracing algorithms (called "Common Path Pessimism Removal" [16] in IBM's EinsTimer and "Clock Reconvergence Pessimism Removal" in Synopsys's PrimeTime). But there are other types of commonality as well: being in the same geographical region of the chip, sharing the same gate types, sharing the same metal levels, sharing the same voltage island, etc. All of these correlations, when taken into account correctly, will reduce the pessimism of the analysis.

The third and most important source of correlation is dependence on global parameters. The delay and slew of pretty much every edge in the timing graph is correlated with every other edge's delay and slew. This is because the chip-to-chip, wafer-to-wafer and lot-to-lot components of variability are not seen *across* a single chip. Thus, there is a component of variability that says that if NFETs are fast, they are likely to be fast all over the chip. Or if metal level 3 is thicker than nominal, all interconnect on metal level 3 everywhere on the chip will likely be affected in a similar manner.

The fourth type of correlation is related to proximity. This includes temperature gradients, power supply gradients and ACLV. It is simply impossible for two gates right next to each other to have the best and worst-case characteristics with respect to these sources of variation, respectively. Any timer that allows such situations in its quest to produce guaranteed timing bounds is being needlessly pessimistic.

4.2 Bounded vs. statistical analysis

A good statistical timer should be able to treat each source of variation in either a bounded or statistical manner in a unified framework. This type of flexibility results in many advantages. For example, environmental variables may be better handled in a bounding manner. Parameters that show a small amount of variation can be handled in a bounding manner to reduce the dimensionality of the full-blown statistical analysis, while incurring relatively small amounts of pessimism. Further, a bounded analysis can be employed as a pre-filter to pose a smaller and more refined problem to the statistical engine.

4.3 Slew and load dependence

We know that the delay of each edge in the timing graph is a function of input slew and output load. Unfortunately, input slew and load also vary with process, environment and ACLV. While these effects may seem to be of a second-order nature, they must be taken into account for accurate sign-off purposes.

Slew and output load dependencies make the underlying problem harder to decouple and therefore more complicated. For example, delay may be modeled as a separable function of the sources of variation in a convenient analytic form. However, it is also a nonlinear function of input slew and output load, which in turn are (perhaps simple and separable) functions of the sources of variation. The nonlinear dependence, however, destroys the separable or simple nature of the delay models.

In addition, the downstream propagation of slews, already complicated in the deterministic case [17], is now even more difficult since we must propagate a (probabilistic) slew downstream at each timing point, composed from the upstream arrival time and slew probability distributions!

4.4 Within-die variation

A statistical timer must be able to handle within-die variation. The across-chip variation due to power supply voltage, for example, may be random in the early stages of the design and become deterministic during final sign-off when the power grid characteristics and locations of gates are known. For algorithms which depend on having a relatively small number of independent sources of variability, [18] provides a method for modeling ACLV with strong local variations but looser correlation with increasing distance.

4.5 The tail matters!

Depending on the application at hand, the tail of the predicted slack distribution has critical importance. For example, the sign-off criterion on an ASIC may be the -3σ clock frequency. Thus the variance prediction of any algorithm that is used must be very good, and the accuracy of the modeling and correlation must be precise in order to have confidence in such a prediction. If Monte Carlo analysis is used, various methods such as intelligent sampling and importance sampling [19, 20] can be employed to improve the confidence of the variance prediction without necessarily increasing the number of samples required.

4.6 Flexibility and methodology interaction

As was discussed earlier, there is clearly a need for accurate sign-off timing, as well as faster and less accurate timing in the inner loop of optimization in, for example, a physical synthesis flow. In the latter case, what matters is that the probabilistic information communicated by the timer to the optimizer merely point the optimizer in a "good direction" to improve the parametric performance of the circuit. In both types of timing, the relevance of the diagnostics provided by the timer will determine the usefulness of the tool. For example, a detailed timer may be able to provide information to the manufacturing line about which parameters require tighter control in order to achieve the most improvement of parametric yield. In the context of an incremental timer, diagnostics such as the identity of the (probabilistically) most critical paths is crucial in guiding the optimizer.

5. SYNTHESIS

While statistical synthesis is a topic of mere speculation at this point, it is important to begin considering the means by which synthesis can be tailored to consider parametric yield. One obvious method is for synthesis to invoke the services of an efficient and incremental statistical timer in its inner loop to determine the performance impact of transforms and changes. Synthesis programs can be urged to maximize sharing (of gate types, metal levels, voltage islands and geographic regions) between launching and capturing paths so that the effects of variability are minimized. Various methods such as error-correcting circuitry or dynamic adjustment of reverse-bias will be used to make the best possible use of otherwise faulty hardware. Synthesis will be encouraged to employ regular layouts where the concomitant performance loss is acceptable. Synthesis methods will be required not only to put such circuitry in place, but to estimate the impact of the risk reduction afforded by such techniques. Finally, physical synthesis methods will be in a position to make tradeoffs between catastrophic yield improvement (by reducing congestion and spacing out wires, for example) and parametric yield improvements (by reducing the distance between launching and capturing paths).

6. MODELING & CHARACTERIZATION

All the methods discussed in this paper assume the existence of a statistical model of the underlying transistors, gates and wires. It is already a complicated and tedious task to produce adequate models for a case-based timing sign-off methodology. Producing accurate and self-consistent statistical models is even more difficult, and it is a challenge to create models that are not needlessly complicated. Often, the assumption is made that delays depend linearly on the sources of variation. While such models are often reasonable and useful, if the underlying source of variation is L_{eff} or V_{dd}, a linear model is known to be inaccurate. Of course, if the variations are small, then a linear model is valid in a small excursion around nominal.

While the details and difficulties of modeling and characterization are not intended to be subjects covered in any detail in this paper, it is clear that these are difficult but necessary tasks that represent a significant divergence from current practice.

7. CONCLUSIONS

Dealing with variability in a quantitative manner is essential for timing verification of digital integrated circuits. Statistical timing has the potential to improve turnaround time, reduce pessimism, help manage risk and guarantee first-time-right hardware. However, it will have a profound impact on the modeling, analysis, verification, synthesis and methodology of high-performance integrated circuits. This paper focused on the requisite attributes of statistical timing algorithms and the ways in which they would impact a successful design flow.

8. ACKNOWLEDGMENTS

The author gratefully acknowledges collaborations and discussions with S. R. Naidu, K. Kalafala, J. A. G. Jess, V. B. Rao, D. J. Hathaway, P. Feldmann, R. H. J. M. Otten, A. Suess and J. P. Soreff.

9. REFERENCES

[1] R. B. Hitchcock, Sr., G. L. Smith, and D. D. Cheng, "Timing analysis of computer hardware," *IBM Journal of Research and Development*, pp. 100–105, January 1982.

[2] R. P. Abato, A. D. Drumm, D. J. Hathaway, and L. P. P. P. van Ginneken, "Incremental timing analysis," *U. S. Patent 5,508,937*, April 1993.

[3] L. Stok, D. S. Kung, D. Brand, A. D. Drumm, A. J. Sullivan, L. N. Reddy, N. Hieter, D. J. Geiger, H. H. Chao, and P. J. Osler, "Booledozer: Logic synthesis for ASICs," *IBM Journal of Research and Development*, pp. 407–430, July 1996.

[4] M. Weber, "My head hurts, my timing stinks, and I don't love on-chip variation," *Proc. Synopsys User Group Meeting*, 2002. Boston, MA.

[5] "International technology roadmap for semiconductors 2001 edition," tech. rep., Semiconductor Industry Association, 2001. Available at `http://public.itrs.net/Files/2001ITRS/Home.htm`.

[6] M. Orshansky, L. Milor, P. Chen, K. Keutzer, and C. Hu, "Impact of systematic spatial intra-chip gate length variability on performance of high-speed digital circuits," *IEEE International Conference on Computer-Aided Design*, pp. 62–67, November 2000. San Jose, CA.

[7] J.-J. Liou, K.-T. Cheng, S. Kundu, and A. Krstic, "Fast statistical timing analysis by probabilistic event propagation," *Proc. 2001 Design Automation Conference*, pp. 661–666, June 2001. Las Vegas, NV.

[8] A. Gattiker, S. Nassif, R. Dinakar, and C. Long, "Timing yield estimation from static timing analysis," *Proc. IEEE International Symposium on Quality Electronic Design*, pp. 437–442, 2001.

[9] M. Orshansky and K. Keutzer, "A general probabilistic framework for worst case timing analysis," *Proc. 2002 Design Automation Conference*, pp. 556–561, June 2002. New Orleans, LA.

[10] P. Feldmann and S. W. Director, "Integrated circuit quality optimization using surface integrals," *IEEE Transactions on Computer-Aided Design of ICs and Systems*, vol. 12, pp. 1868–1879, December 1993.

[11] J. M. Wojciechowski and J. Vlach, "Ellipsoidal method for design centering and yield estimation," *IEEE Transactions on Computer-Aided Design of ICs and Systems*, vol. 12, pp. 1570–1579, October 1993.

[12] J. A. G. Jess, K. Kalafala, S. R. Naidu, R. H. J. M. Otten, and C. Visweswariah, "Statistical timing for parametric yield prediction of digital integrated circuits," *Proc. 2003 Design Automation Conference*, June 2003. Anaheim, CA.

[13] Z. Li, X. Lu, and W. Shi, "An algorithm for process variation reduction based on SVD," *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2003. Bangkok, Thailand, accepted for publication.

[14] A. B. Agarwal, D. Blaauw, V. Zolotov, and S. Vrudhula, "Statistical timing analysis using bounds and selective enumeration," *Proc. 2002 TAU (ACM/IEEE workshop on timing issues in the specification and synthesis of digital systems)*, pp. 29–36, December 2002. Monterey, CA.

[15] A. B. Agarwal, D. Blaauw, V. Zolotov, and S. Vrudhula, "Computation and refinement of statistical bounds on circuit delay," *Proc. 2003 Design Automation Conference*, June 2003. Anaheim, CA.

[16] D. J. Hathaway, J. P. Alvarez, and K. P. Belkhale, "Network timing analysis method which eliminates timing variations between signals traversing a common circuit path," *U. S. Patent 5,636,372*, June 1997.

[17] D. Blaauw, V. Zolotov, S. Sundareswaran, C. Oh, and R. Panda, "Slope propagation in static timing analysis," *IEEE International Conference on Computer-Aided Design*, pp. 338–343, November 2000. San Jose, CA.

[18] A. B. Agarwal, D. Blaauw, V. Zolotov, S. Sundareswaran, M. Zhao, K. Gala, and R. Panda, "Path-based statistical timing analysis considering inter- and intra-die correlations," *Proc. 2002 TAU (ACM/IEEE workshop on timing issues in the specification and synthesis of digital systems)*, pp. 16–21, December 2002. Monterey, CA.

[19] R. Y. Rubinstein, *Simulation and the Monte Carlo method*. John Wiley and Sons, 1981.

[20] R. Spence and R. S. Soin, *Tolerance design of electronic circuits*. Addison-Wesley Publishing Company, 1988.