Statistical Timing Analysis using Kernel Smoothing

Jennifer L. Wong*, Azadeh Davoodi*, Vishal Khandelwal†, Ankur Srivastava†, Miodrag Potkonjak◇
*Stony Brook University, Computer Science, jwong@cs.sunysb.edu
*University of Wisconsin, Madison, Electrical & Computer Engineering, adavoodi@wisc.edu
†Synopsys Inc., Advanced Technology Group, vishalk@synopsys.com
‡University of Maryland, College Park, Electrical & Computer Engineering, ankurs@glue.umd.edu
◇University of California, Los Angeles, Computer Science, miodrag@cs.ucla.edu

Abstract

We have developed a new statistical timing analysis approach that does not impose any assumptions on the nature of manufacturing variability and takes into account an arbitrary model of spatial correlation as well as all types of functional correlations (e.g. reconvergence-based correlations). The starting point for statistical timing analysis is small scale Monte Carlo (MC) simulation. In order to speed-up the MC simulation process we use stratified balanced sampling and postprocessing of the simulation data using non-parametric kernel estimation. The MC simulation and the statistical analysis procedure are interleaved with the calculation of the critical paths. In order to speed up simulation, we identify and simulate only gates relevant for calculation of the clock cycle time. The application of statistical techniques enable not only accurate statistical timing analysis, but also stability and scalability analysis. The approach is evaluated using MCNC benchmarks and yields more than six orders of magnitude speed improvement compared with the standard MC simulation.

1 Introduction

High design manufacturing variability, in deep submicron and nano-technologies, has a number of strong ramifications on many design and analysis steps. There is a wide consensus that traditional timing analysis procedures have to be augmented with new techniques that take into account manufacturing variability. Currently, the majority of the techniques that address this problem use probabilistic techniques which place a strong assumption on the distributions and independence between relative timings. Our goal is to demonstrate a new conceptual direction that may be better suited for eventual application to actual designs.

The starting point is one of the oldest and probably the most natural techniques - Monte Carlo simulation. Probably the most important decision during statistical timing analysis is how the distribution of the length of a particular path is calculated. For this purpose, for each scenario in terms of variability, we compute the overall delay from the primary inputs/outputs of flip-flops to the primary outputs/inputs of flip-flops. Although it may be tempting to calculate the distribution of delay incrementally by treating individual gates individually, this procedure is not sound because it does not consider the correlations between the delays of various paths introduced by manufacturing variability.

There are five major novelties in the new non-parametric statistics based timing analysis approach. The first is consistent use of non-parametric statistical techniques to analyze and smooth results, and reduce run time of the MC simulation. Statistical analysis is also used for other tasks such as the calculation of correlation between delays on the output of different gates in order to better recognize whether the simulation was long enough to obtain design insights. The second important innovation is the application of balanced resampling in Monte Carlo simulation which reduces both bias and required simulation time at a linear rate.

We believe that the single most important innovation for very large designs is interleaving the derivation of statistical models and their validation with algorithmic steps for identification of relevant gates in the design. The relevant gates are defined as gates that belong to the longest path in at least one scenario. The fourth innovation is comprehensive and consistent application of statistical validation and evaluation techniques to identify to what extent a particular conclusion can be trusted. Finally, we also use statistical techniques to analyze the robustness of our statistical timing analysis with respect to the adopted assumptions (such as the sources of manufacturing variability) and scalability analysis that statistically predict the amount of time required to predict the statistical distribution of the longest path in a targeted design.

The impact of manufacturing variability on timing characteristics of deep submicron designs has been recognized for a long time. Since the mid-90's both the impact of inter-die variability of key device parameters for transistors [3]

1-4244-1258-7/07/$25.00 ©2007 IEEE

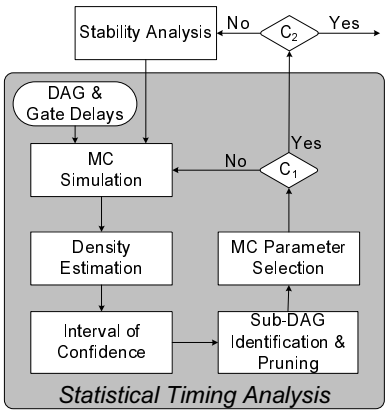


Figure 1. Global flow for non-parametric statistics-based timing analysis procedure.

and wire variability have been studied [10]. Additionally, several technological sources of variability has been identified [11, 14].

Due to the recognition of the increasing impact of variability of design manufacturing, a number of techniques have been proposed for statistical timing analysis. Majority of the approaches target intra-chip variations and the previously published statistical timing techniques are most often based on probability analysis mechanisms. For example, a family of continuous models use closed form probability distribution function (PDF) analysis with analytical procedures. In order to facilitate analysis, the standard assumption is that variability follows Gaussian normal distribution. Another line of attack on statistical timing analysis has been conducted using discrete methods where the discrete probabilities are propagated from the inputs to the outputs of a design [8, 1]. These types of techniques often suffer from a difficulty to address correlation and the exponential combinatorial runtime requirements. Additionally, Monte Carlo simulation has been used to analyze circuit timing on a set of selected sensitizable true paths [9].

Most of the latest efforts have been focused on addressing spatial and reconvergence-based correlation within a probabilistic analysis framework [15].The standard reference for almost all statistical techniques in this work is [7].

2 STA Methodology

The overall timing analysis flow is shown in Figure 1. The starting point for the approach is a design specified at the logic level. In our experimentation, we used the timing model generator that captures several types of correlations and randomness, as shown in the next subsection. The approach is iterative. Each iteration is based on one round of simulations. The first step of each of the rounds of simulation is pseudo-random Monte Carlo (MC) simulation. In order to properly and completely capture all correlations, a single run of simulation calculates the delay on all paths using computationally simplified Dijkstra algorithms for directed acyclic graphs (DAGs). For each of the individual simulations, we calculate the lengths of the longest paths to each input and each output of all gates. Once the initial simulation results are collected, their accuracy is enhanced using density estimation techniques. This step is based on the observation that the nature of the problem is such that small changes in the delay of each gate result only in small changes in the overall delay of the design.

Once the MC simulation, kernel smoothing, and interval of confidence steps are completed, we analyze the time delay annotated graph. Our goal is to identify and subsequently consider only the gates that impact the overall critical path. The final step, before the next round of iterations is the derivation of parameters for the impeding pseudo-random MC simulation.

The iterations are conducted until the user specified termination criteria (C_1) is invalid. Although one can envision a multitude of termination criteria, we believe that the most plausible are based on achieving a specified level of confidence on the overall timing of the design. Once the mandatory basic procedure is completed, one can conduct optional stability and scaling analysis. The stability analysis varies the input data over different distributions or statistical models in order to establish the extent to which the obtained conclusions will hold if the manufacturability process changes its characteristics. The execution of stability analysis is controlled using the user specified parameter C_2 in Figure 1. The scalability is analyzed using statistical methods.

Our statistical timing analysis framework has been integrated in SIS. We also implemented designs so that a mapped gate level description of the design is given to us along with placement information. In order to capture the spatial correlation of gate delays we used the model developed and measured parameters from the study conducted at UC Berkeley [4]. The model is implemented using a rejection method [2, 13].

We already discussed two of the most important issues related to MC simulations [5]. The first one is the generation and reuse of random numbers. The second issue is the use of techniques for variance reduction in Monte Carlo simulation. We have experimented with the three most widely used directions for variance reduction: (i) analytic reduction; (ii) auxiliary variables; and (iii) probabilistic sampling. The analytic reduction techniques usually reduce a part of the MC simulation problem using analytic techniques. In our case, we used several closed formula approximations for this purpose. However, the speed-ups were relatively limited, by factors of 2 to 4 times.

We tried two auxiliary variables techniques. The first was based on the use of control variates (essentially both positively and negatively correlated variables). The second used antithetic variates and performed significantly better results with more than an order of magnitude reduction for

a given level of accuracy. The best performing technique for a variant of probability sampling was Hall's balanced resampling [6]. The application of the Hall's balanced resampling MC approach resulted in a reduction of almost two orders of magnitude for higher accuracy.

3 Non-Parametric Density Estimation

if one can conduct enough extensive MC simulation there is no need for additional processing and all conclusions about timing properties of a design in the presence of manufacturing variability can be correctly deduced to an arbitrary level of accuracy. However, if we take into account that modern designs can have millions of gates and in the future even billions of gates, it is easy to calculate that for industrial strength designs even on the fastest computers one can conduct only MC simulations of moderate quantity. It is well known that in order to increase the accuracy of MC simulations for an additional significant digit, one has to increase the number of simulations by a factor of 100 [5]. In order to overcome this difficulty, we use non-parametric statistical techniques to significantly (by several orders of magnitudes) reduce the quantity of necessary MC simulation trials in order to achieve high accuracy.

The main goal of density estimation, that is often called in statistical literature "kernel smoothing", is to obtain more accurate values for a PDF function. This is accomplished by using the following intuitive observation that values of the PDF function for small timing difference are relatively well representative of each other and that this similarity diminishes as the timing difference increases. It is very important to observe another implicit goal of smoothing: to find all gates that belong to the critical paths that are longest in one of the scenarios dictated by manufacturing variability. By extrapolating the PDF near its ends, we can analyze if it is possible that a particular output is on the critical path of the design for some values of the prediction variables.

We have analyzed two types of density estimation techniques for more accurate characterization of PDFs: 2-D and 2^+-D. In 2-D models we were predicting the PDF of a given gate as the function of delays at the output of that gate. 3-D and multi-dimension kernel smoothing techniques were considering PDFs as a function of not only the longest path at the output of a given gate, but also as a function of variability metrics in order to more accurately predict how the PDF depends not only on the timing delay but also on, for example, velocity saturation index α.

Conceptually, the 2-D model is attractive because of its simplicity and use of single dimension for all the available points. For model building we experimented with several kernel smoothing and local regression techniques. For kernel smoothing we used Nadaraya-Watson kernel-weighted average with Epanicechnikov, triangular, tri-cube, and Gaussian kernels [7]. The selection of smoothing parameters (e.g. scope and shape parameters of the window)

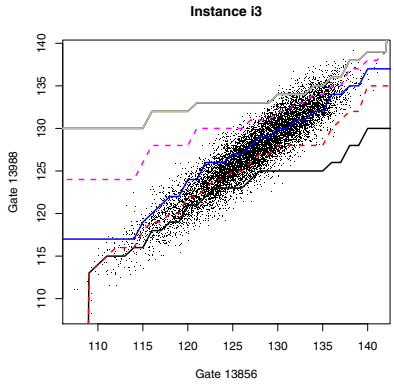


Figure 2. Design i3 mapping of gate delay between longest two outputs. Lines represent 12.5%, 25%, 50%, 75%, 87.5% PDF values.

was conducted using resubstitution after selecting 200 different groups of a randomly selected set of 60% of available measurements for the learning phase, and the rest for the test phase. There were two important observations. The first was that almost all the techniques performed approximately equally. The second is that in all situations, the best window size was 5% W, where W is the total length of nonzero bins of the initial histogram. This behavior is a direct consequence of the shape of the PDF function of timing delay: its unimodularity and non-existence of heavy tails on any of considered designs.

For 3-D and more complex models we also applied the number of Nadaraya-Watson kernel-weighted average kernels. In the case of these types of models we noticed that the window size had to be increased to 10% W. The larger kernel window size is required because of the curse of dimensionality that dictates that as the number of dimensions increases we have to have a larger number of data samples. Interestingly, 2D kernel smoothing techniques performed better. The reason is clear and simple: we insisted on very short simulation times where only tens of thousands of scenarios were examined. This number is sufficient to build very accurate 2D models, but not sufficient for higher dimensionality models. More detailed analysis of the smoothing technique indicated that in all cases the best results were obtained using Nadaraya-Watson kernel regression smoother [7] where the size of window is set automatically but adaptively in such a way to the resulting delay distribution is unimodular.

4 Critical Sub-DAG Pruning

We introduce a conceptually simple, yet effective way to interleave MC simulation followed by statistical analysis with identification of a critical sub-DAG. A critical sub-DAG is a subgraph of the initial DAG that consists only of the nodes and edges that belong in any of the analyzed DAGs on the critical path of the DAG. In addition, a critical sub-DAG also includes nodes and edges that have a speci-

```
1.  α = M;
2.  while ( αR != empty ) {
3.     MC-simulation (DAG);
4.     Kernel Estimation(DAG);
5.     Identify set R of all the gates not on ϵ-critical path;
6.     DAG = DAG  R;
7.     α = α - δ;
8.  }
9.  MC-simulation (DAG);
10. PDF-augmentation;
```

Figure 3. Pseudo-code for critical DAG identification and pruning algorithm.

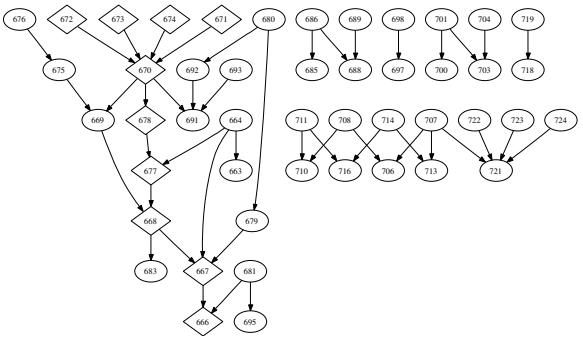


Figure 4. Design i1. Ellipses indicate non-relevant gates. Diamonds indicate relevant gates.

Design	Gates	relevant in/out	% overall relevant in/out	rel. gates	% rel. gates
i1	47	1 / 3	6.38 / 2.13	9	19.15
i3	178	4 / 20	11.24 / 2.25	41	23.03
C432	179	1 / 3	1.68 / 0.56	62	34.64
i2	181	1 / 4	2.21 / 0.55	11	6.08
i5	181	1 / 1	0.55 / 0.55	10	5.52
x1	233	1 / 1	0.43 / 0.43	8	3.43
too_large	263	2 / 2	0.76 / 0.76	17	6.46
C880	272	1 / 1	0.37 / 0.37	28	10.29
x4	307	1 / 3	0.98 / 0.33	13	4.23
C1355	314	7 / 11	3.50 / 2.23	103	32.80
C499	314	6 / 3	0.96 / 1.91	49	15.61
C1908	397	1 / 3	0.76 / 0.25	42	10.58
i6	510	33 1	0.20 / 6.47	100	19.61
x3	551	1 / 1	0.18 / 0.18	9	1.63
i9	558	15 1	0.18 / 2.69	69	12.37
rot	579	4 / 1	0.17 / 0.69	23	3.97
t481	612	1 / 1	0.16 / 0.16	15	2.45
i7	694	13 1	0.14 / 1.87	41	5.91
i8	792	17 1	0.13 / 2.15	30	3.79
C3540	865	1 / 1	0.12 / 0.12	46	5.32
dalu	994	1 / 2	0.20 / 0.10	49	4.93
C5315	1193	1 / 1	0.08 / 0.08	19	1.59
i10	1990	1 / 1	0.05 / 0.05	36	1.81
C6288	2011	1 / 1	0.05 / 0.05	250	12.43
des	2736	23 / 1	0.04 / 0.84	83	3.03

Table 1. Percentage of relevant gates assuming generalized Pelgrom model

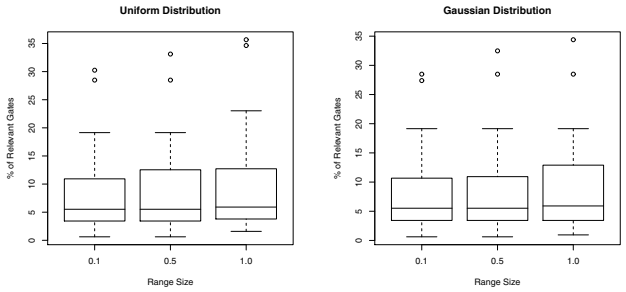
fied threshold of likelihood that they can belong to the critical path during one of the pending simulations.

The main idea is simple and effective. After initial simulation, we identify all gates that belonged in any MC trial to the critical path. In addition to considering only critical paths, we also consider all paths that are within α of the critical path in a given trial. The consideration of α critical sub-DAGs is important because although a specific path with a specific gate is not on any of the critical paths in the conducted MC trials, it maybe with relatively high probability one of the critical paths in the future trials if the difference in the path delay at that point is relatively small.

Figure 3 shows the pseudo-code for the critical sub-DAG identification and pruning algorithm. In the first line, we set ε to a large constant, M. In our experimentation for M we used the length of the longest critical path in the first MC. The lines 2-8 form the main body of the algorithm. In line 3 we conduct k MC simulation trials. The number of trials is subject to user discretion. Our suggestion is that the number of simulations is at least $\frac{n}{10}$ where n is the number of gates in the design. In line 4, we apply kernel smoothing in order to better predict the likelihood of any output to be on the critical path. In line 5, we identify the set of all ε critical gates by tracing paths from the critical outputs. These gates and their incoming and outgoing edges are eliminated from the DAG in step 6. Once the step is completed, we reduce our ε by a δ decrement. Again the δ decrement is under the discretion of the user. However, in our experimentation we used δ = 0.1*ε, which is the value we recommend as a sensible and conservative choice, at least for design of the same structure as our benchmark set. This value for δ is experimentally and statistically derived from our experiments. This process is repeated as long as during one run of the while loop no additional gates are identified that are not on the ε critical path. Finally, in the last three steps we complete the MC simulation of the pruned DAG followed by kernel smoothing of all obtained delays. The final step in line 11 adds additional PDF values to delays in order to compensate for potentially missed critical paths due to pruning. We simulated the majority of our designs for 10 million MC trials, and for a few smaller 100 million MC trials.

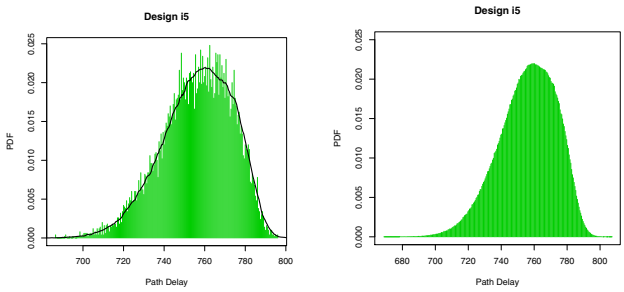
For example, Figure 4 shows the DAG of the i1 design from MCNC benchmark set. The design has a total of 47 gates. Each non-critical path gate is drawn in an ellipse, while gates which were on any critical path found during MC simulation are shown using diamonds. A directed edge indicates the flow of the signals between two gates. The figure shows four critical inputs at the top of the figure and one critical output at the bottom. Only 9 of the 47 gates ($\sim$19%) were on any critical paths found in all MC simulation.

Study of all our designs found that on average 10.03% of the gates in each instance are relative gates. In Table 1 we present the results of this study performed on the instances with generalized Pelgrom model [12] of the parameters on the full range of parameter values. The first column states the name of the benchmark, while the second column states the number of gates in the benchmark. The next three



(a) Uniform (b) Gaussian

Figure 5. Percentage of relevant gates for uniform and gaussian distributions for all 25 benchmark examples for the given parameter.



(a) Smooth (b) Brute-force

Figure 6. PDF (a) before (shaded area) and after statistical modeling (curve) and (b) after 100,000,000 MC trials.

Design	L_1	L_2	L_∞	RL_1	RL_2	RL_∞
i1	0.68	0.98	2.22	0.0017	0.044	0.094
i3	0.41	0.62	0.99	0.0034	0.036	0.102
C432	3.25	7.89	22.53	0.0009	0.727	0.042
i2	1.02	1.56	3.89	0.0026	0.622	0.0098
i5	2.42	3.20	19.80	0.0045	1.120	0.0815

Table 2. Six different error norms for discrepancy of PDFs obtained using pseudo-exhaustive simulation and the new non-parametric statistics-based approach.

columns represent information concerning the number of relevant inputs and outputs for each design (number of relevant inputs/outputs, percentage of relevant inputs/outputs, percentage of relevant inputs/outputs over the entire benchmark). The final two columns state the number of relevant gates in the benchmark and the percentage of relative gates.

Figure 7 illustrates the scalability analysis for the non-parametric statistic and active learning-based statistical timing analysis approach. The figure shows the overall MC simulation runtime (in seconds) for 10,000 trials for each of the 25 designs with respect the size of the original design for two sets of data: all gates and relative gates. As the number of gates increases in the original design the MC simulation runtime grows quadratically. However, when only the identified relevant gates are considered in the MC simulation trials the runtime growth is linear. Further analysis showed that the percentage of relevant gates with respect to the size of the original design decreases as the design size grows. This is the main reason for the linear runtime growth in MC simulations on relevant gates only.

5 Simulation Results

We evaluate the approach using extensive simulation. We compare the PDF of critical path delay of the design obtained using the new approach and obtained using extensive simulation. Specifically, we limit out approach to use at most 10 thousand scenarios in the MC simulation. On the other hand we used brute-force 100 million MC trials. Figure 6(a) shows PDF of critical path obtained using the new approach and Figure 6(b) shows the PDF of the same i5 design obtained using comprehensive brute-force MC simulation. Table 2 shows six standard error norms (L_1, L_2, L_∞, relative L_1 (RL_1), relative L_2 (RL_2), relative L_∞ (RL_∞)) for the five smallest designs compared using extensive simulation of the fast procedure. For design i5, we see that the difference in all cases is less than 0.2%. For example, we compute L_1 error norm first sorting all values for the length of the critical path in both networks. After we take each 10 thousandth sample from the second simulation. Now the L_1 error norm is calculated using formula $\frac{\sum |(s_1(i) - s_2(i))|}{n}$ where n is the number of samples, $s_1(i)$ is the i^{th} smallest sample in the fast simulation, and $s_2(i)$ is the i^{th} subsample in the second comprehensive simulation.

We evaluated all designs using the following statistical procedure that consist of three steps. In the first steps we find the error model of each design using 10 thousand MC simulation trials. In the second step we use kernel smoothing techniques to enhance the statistical accuracy of the critical path PDF of each design. In the third step we do 200 cases of 10 thousand trials. We spend the majority of the MC simulation trials conducting on only a relatively small subset of gates of the initial design as shown in Table 1.

We also evaluated the stability and scalability of the new statistic timing analysis approach. Due to space limitations we show only a part of scalability analysis. Figure 7 shows the compound runtimes for 10 executions of the overall approach on a 900 MHz laptop. We see that if we simulate all gates the best fit is quadratic, while if we use DAG pruning, the runtime has linear scalability.

Another study showed that the percentage of relevant gates with respect to the size of the original design decreases as the design size grows. Figure 7 graphically presents this information and strongly suggests that the new procedure for statistical timing analysis using non-parametric kernel smoothing and pruning of designs is highly scalable.

One of the essential questions in statistical timing analy-

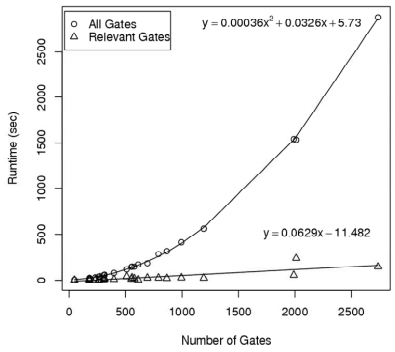


Figure 7. The runtime of STA as a function of the total number gates and relevant gates for STA.

Design	10^2	10^3	10^4	10^5
i1	14.4/5.6	8.0/2.2	5.2/1.3	4.4/0.7
i3	16.7/5.9	8.8/1.9	5.4/1.2	5.0/0.7
C432	17.2/6.6	8.9/2.0	6.2/1.5	5.2/1.0
i2	17.9/5.7	7.4/2.2	6.3/1.5	5.1/0.9
i5	16.6/6.8	9.7/2.4	6.6/1.7	5.2/0.8
C880	19.7/5.6	8.9/2.2	6.0/1.5	5.2/0.8
C2540	20.0/7.0	9.8/2.5	7.4/1.8	5.7/0.9

Table 3. The average CDF discrepancy for MC simulation of a given number of trials.

sis is the accuracy of obtained results. In the case of Monte Carlo simulation-based approach, this question translates into how many simulation trials are required for a certain level of STA accuracy. We define as the measure of accuracy, the maximal discrepancy between cumulative density functions (CDF) of the critical path for a given circuit with respect to one obtained using extensive Monte Carlo simulation. Specifically, we used as the yardstick the results obtained using 100,000,000 MC trials. Tables 3 and 4 shows relative average and maximal discrepancy respectively over a subset of our benchmark set for different lengths of Monte Carlo simulation. The discrepancy is calculated starting from 200 different runs of MC simulation and kernel smoothing of particular length.

6 Conclusion

We have developed a new statistical timing analysis approach. The consistent application of non-parametric statistical techniques enables accurate consideration of all types of correlations and rapid and easy to characterize statistical timing analysis. Interleaving of the algorithmic and statistical analysis steps enables additional speed-up of the statistical timing analysis that combined with balanced resampling and kernel smoothing modeling yields overall more than six orders of magnitude improvement over direct Monte Carlo simulations even on small designs for the same level of accuracy.

Design	10^2	10^3	10^4	10^5
i1	88/15.7	61/11.7	37/8.3	22/4.0
i3	82/17.9	56/10.8	41/5.0	25/3.7
C432	83/18.8	70/12.7	56/6.0	19/2.9
i2	56/19.2	71/19.6	41/5.9	18/4.8
i5	81/20.9	68/17.2	48/11.2	24/4.9
C880	95/22.7	72/16.2	46/10.2	24/5.2
C2540	102/20.5	73/16.6	39/8.3	25/3.9

Table 4. The average CDF discrepancy for MC simulation of a given number of trials. Each entry indicates the relative discrepancy before and after smoothing.

References

[1] A. Agarwal, D. Blaauw, V. Zolotov, and S. Vrudhula. Computation and refinement of statistical bounds on circuit delay. In *DAC*, pages 348–353, 2003.

[2] L. Devroye and R. Neininger. Density approximation and exact simulation of random variables that are solutions of fixed-point equations. *Advances in Applied Probability*, 2002.

[3] M. Eisele, J. Berthold, D. Schmitt-Landsiedel, and R. Mahnkopf. The impact of intra-die device parameter variations on path delays and on the design for yield of low voltae digital circuits. *IEEE TVLSI*, pages 360–368, 1997.

[4] P. Friedberg et al. Modeling within-die spatial correlation effects for process-design co-optimization. In *ISQED*, pages 516–521, 2005.

[5] J. E. Gentle. *Elements of Computational Statistics*. Springer, 2002.

[6] P. Hall. Performance of balanced bootstrap resampling in distribution function and quantile problems. In *Probability Theory and Related Fields*, volume 85, pages 239–360, 1990.

[7] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer-Verlag, 2001.

[8] J.-J. Liou, K.-T. Cheng, S. Kundu, and A. Krstic. Fast statistical timing analysis by probabilistic event propagation. In *DAC*, pages 661–666, 2001.

[9] J.-J. Liou, A. Krstic, L.-C. Wang, and K.-T. Cheng. False-path-aware statistical timing analysis and efficient path selection for delay testing and timing analysis. In *DAC*, pages 566–569, 2002.

[10] V. Mehrotra, S. Nassif, D. Boning, and J. Chung. Modeling the effects of manufacturing variation on high-speed microprocessor interconnect performance. In *IEDM*, 1998.

[11] S. R. Nassif. Design for variability in dsm technologies. In *ISQED*, pages 451–455, 2000.

[12] M. J. Pelgrom, A. C. J. Duinmaijer, and A. R. G. Welbers. Dual-threshold voltage assignment with transistor sizing for low poer CMOS circuits. *IEEE J. of Solid- State Circuits*, 24(5):1433–1439, Oct. 1989.

[13] W. H. Press et al. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, 2nd edition, 2002.

[14] A. K. Sultania, D. Sylvester, and S. Sapatnekar. Tradeoffs between gate oxide leakage and delay for dual t_{ox} circuits. In *DAC*, pages 761–766, 2004.

[15] L. Zhang, W. Chen, Y. Hu, J. A. Gubner, and C. Chen. Correlation-preserved non-gaussian statistical timing analysis with quadratic timing model. In *DAC*, pages 83–88, 2005.