Statistical Timing Analysis using Kernel Smoothing

Jennifer L. Wong[†], Vishwal Khandelwal[‡], Ankur Srivastava[‡], Miodrag Potkonjak[†]

[†] Computer Science Department,
University of California, Los Angeles, {jwong,miodrag}@cs.ucla.edu
[‡] Electrical and Computer Engineering Department,
University of Maryland, College Park, {vishwa,ankurs}@glue.umd.edu

UCLA Computer Science Department
Technical Report #060003

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 logic synthesis benchmarks and yields more than six orders of magnitude speed improvement compared with the standard MC simulation.

1 Introduction

Inevitable high variability in design manufacturability in deep submicron and nano technologies have a number of strong ramifications on many design and analysis steps. In particular, there is a wide consensus that traditional timing analysis procedures have to be augmented with new techniques that take into account manufacturing variability. A number of landmark papers have been recently presented in this new area, typically called statistical timing analysis. Regardless of the name, the majority of the works 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 much 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, which is sometimes used in statistical timing analysis and many other tasks. However, this is where the similarities between the new and the previous approaches end. 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 one by one, this procedure is not sound because it does not consider the correlations between the delays of various paths. It has already been identified that convergence can be a source of significant correlation, but it is also important to emphasis that any two paths that share any gates are correlated and often even paths that do not share gates can be correlated because they have gates that are physically close and therefore their delays are correlated. Two simple but very influential logistic issues that greatly influence the overall runtime of statistical timing analysis is how the critical path is calculated and how the random numbers are generated. For the calculation

of the critical path it is important to derive a specialized procedure where the order in which the delay of the output of the gate is fixed in order to avoid unnecessary searches by the generic Dijkstra algorithm. Additionally, it is well know that the generation of random numbers can be very expensive (in hundreds of cycles) therefore for this purpose we generate a single long list of random numbers and store it in order to use on all our designs.

There are four 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 all results of the MC simulation. In addition, statistical analysis is 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 bias 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.

2 Related Work

The impact of manufacturing variability on timing characteristics of deep submicron designs has been recognized for a long time [36, 33]. Since the mid-90's both the impact of inter-die variability of key device parameters for transistors [23] and wire variability have been studied [24]. Additionally, several technological sources of variability has been identified [26, 5].

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 [16, 8, 4, 1, 3, 6, 2]. 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 [17]. Other notable statistical timing analysis efforts include [25, 7, 11, 30, 11, 27].

Most of the latest efforts have been focused on addressing spatial and reconvergence-based correlation within a probabilistic analysis framework [35, 13, 21, 20]. In addition, there have been several efforts to analyze and exploit the interaction between designs steps and statistical timing analysis [18, 22]. The main conceptual novelty of our approach includes the use of non-parametric modeling and validation techniques, and the interleaving of statistical and algorithmic phases. The standard reference for almost all statistical techniques in this work is [15].

3 Global Flow & Variability Model

In this section we introduce the overall flow of the non-parametric statistics based delay analysis approach. The emphasis is on capturing and presenting the key ideas at the intuitive level and on summarizing the interfaces between the individual blocks. All of the block-level procedures are described in much more detail in consequent sections.

The overall timing analysis flow is shown in Figure 1. The starting point for the approach is a design specified at the logic level. The overall 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

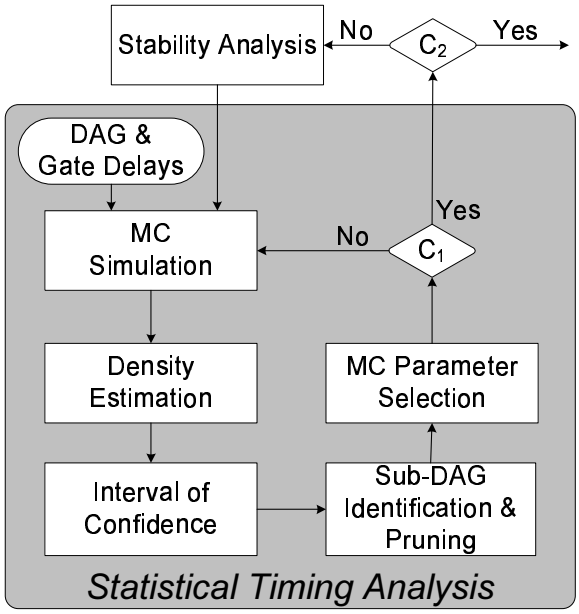


Figure 1: Global flow for non-parametric statistics-based timing analysis procedure.

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 statistical models for all delays of interests are obtained, they are analyzed using a resubstitution procedure in order to establish the corresponding intervals of confidence.

Once the MC simulation, kernel smoothing, and interval of confidence steps are completed, we analyze the time delay annotated graph. The idea is simple, yet powerful. Our goal is to identify and subsequently consider only the gates that impact the overall critical path. In addition, our secondary objective is to identify gates and paths with high variability in order to additionally simulate only these parts of the design in order to make our knowledge about the timing behavior of the overall design more confident. 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. Finally, 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 [28]. The model is implemented using rejection method [9, 34].

4 Monte Carlo (MC) Simulation

We already explained two of the most important issues related to MC simulations [31, 10, 12]. 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 [14]. The application of the Hall's balanced resampling MC approach resulted in a reduction of almost two orders of magnitude for higher accuracy.

5 Non-Parametric Density Estimation

In principle, 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 [19, 12]. In order to overcome this difficulty, we propose the use of 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.

Even when the designs are relatively small and one can conduct extensive MC simulation there are numerous advantages for using statistical density estimation techniques for the analysis of the obtained results. For example, using statistical techniques one can gain design insights with respect to which gates are the most influential for the length of the critical path. Statistical analysis can also serve as the first potential step for theoretical analysis after we obtained closed-form approximations for delay of a particular path or set of paths. Maybe most importantly, statistical estimation techniques greatly facilitate the analysis of the interval of confidence for all of the obtained results. They can also provide an indication about the impact of different assumptions about sources of variability and eventual delays in the design.

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. Essentially 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 probability distribution function 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 [15, 32]. Since our function was very flat at minimal and maximal

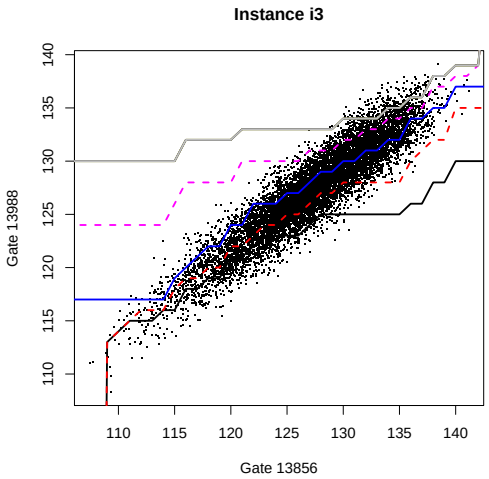


Figure 2: Design i3 mapping of gate delay between longest two outputs. Lines represent 12.5%, 25%, 50%, 75%, 87.5% PDF values.

value for the error, we did not have problems with weighted averages that are sometimes significantly biased on the boundaries of the domain. Nevertheless, we tried three local regression techniques: linear, quadratic, and penalized least squares (smoothing splines). The choosing of smoothing parameters (e.g. window scope and shape parameters of the window) 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. We believe that 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, overall 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. We expect that on large sample sizes, 2^+-D models will have superior performances.

Finally, it is important to note that we applied density estimation statistical techniques also on predicting a delay at one gate from the delay at another gate. Figure 2 shows a scatter plot and superimposed PDF for the mapping between two gates in design. Our analysis of a large number of gates in many designs indicates that for a variety of distributions for manufacturing variability factors of designs we have uni-modular PDFs.

Resubstitution is the process where a number of statistical models are built using the exact same procedure (e.g. kernel window scope and weight function) on randomly selected subsets of data. Specifically, in our simulations, we select 60% of the available data to build a model on each resubstitution run. For each resubstitution run we record the value of the PDF function at each of the k selected points. After conducting m resubstitution runs (in our experimentation m was 100), we can establish an interval of confidence for our statistical PDF model using a two stage procedure. In the first, we establish an interval of confidence for each point individually, and then by combining information from all local interval of confidences we establish a global interval of confidence. The final step of resubstitution is to build a global measure of the accuracy of the model. To build a global interval of confidence we use the following procedure. First, for each separate point in k resubstitutions, we use the highest likelihood PDF value and normalize all other values against this value. After that, we combine all data from all sampling points into one set of the size k x m. Finally, we calculate the confidence intervals of the normalized global array. Our analysis, indicates that in all results presented in the Simulation section we had intervals of confidence higher than 99.9%.

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

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

6 Critical Sub-DAG Pruning

In this section we introduce a conceptually simple, yet exceptionally 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 specified threshold of likelihood that they can belong to the critical path during one of the pending simulations.

The main idea is remarkably simple and even more remarkably effective. After initial simulation we identify all gates that belonged to any MC trial on the critical path. In addition to considering only critical paths, we also consider all paths that are within ε of the critical path in a given trial. The consideration of ε 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 $\delta = 0.1^{*}\varepsilon$, 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. It is interesting that in none of these cases we were able to find a critical gate and path that was not detected using the presented algorithm when we started with as little as five thousand 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. Overall, only nine 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 [29] 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 columns

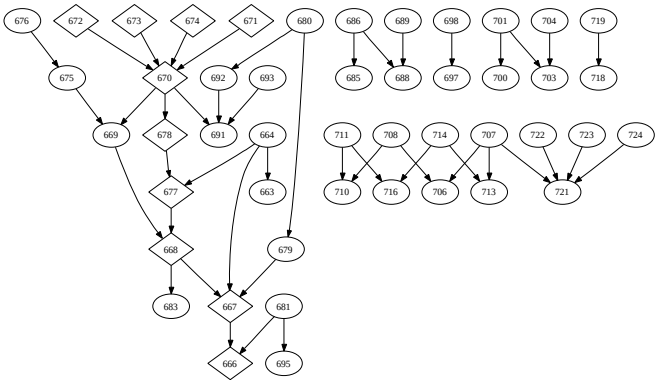
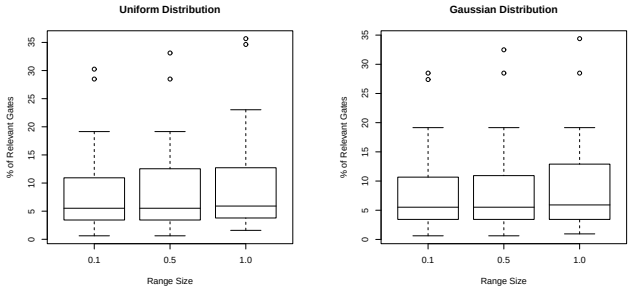


Figure 4: Design i1. Ellipses indicate non-relevant gates. Diamonds indicate relevant gates.

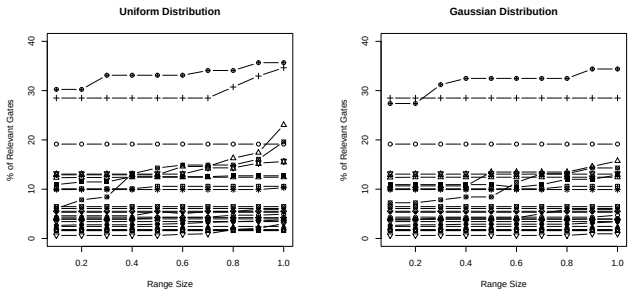


(a) Uniform (b) Gaussian

Figure 5: Boxplot that indicates the percentage of relevant gates for uniform and gaussian distribution. Boxplots capture all 25 benchmark examples for the given parameter.

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 overall. Figures 5-6 summarize graphically the same type of information for different combinations of distribution types (Uniform and Gaussian) and different levels of manufacturing variability (factor of range size of parameters around the mean value).

Figure 10(b) 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



(a) Uniform (b) Gaussian

Figure 6: Complete information about percentage of relative gates for uniform and gaussian distribution.

Design	# gates	# relevant inputs/outputs	% relevant inputs/outputs	% overall relevant inputs/outputs	# relevant gates	% overall relevant gates
i1	47	1 / 3	13.64 / 6.25	6.38 / 2.13	9	19.15
i3	178	4 / 20	15.15 / 66.67	11.24 / 2.25	41	23.03
C432	179	1 / 3	11.54 / 100	1.68 / 0.56	62	34.64
i2	181	1 / 4	3.88 / 100	2.21 / 0.55	11	6.08
i5	181	1 / 1	1.18 / 2.38	0.55 / 0.55	10	5.52
x1	233	1 / 1	1.39 / 3.45	0.43 / 0.43	8	3.43
too_large	263	2 / 2	4.35 / 66.67	0.76 / 0.76	17	6.46
C880	272	1 / 1	1.69 / 3.85	0.37 / 0.37	28	10.29
x4	307	1 / 3	3.80 / 1.64	0.98 / 0.33	13	4.23
C1355	314	7 / 11	27.50 / 21.88	3.50 / 2.23	103	32.80
C499	314	6 / 3	7.50 / 18.75	0.96 / 1.91	49	15.61
C1908	397	1 / 3	8.57 / 4	0.76 / 0.25	42	10.58
i6	510	33 1	0.76 / 49.25	0.20 / 6.47	100	19.61
x3	551	1 / 1	0.68 / 1.05	0.18 / 0.18	9	1.63
i9	558	15 1	1.22 / 23.81	0.18 / 2.69	69	12.37
rot	579	4 / 1	0.72 / 4.40	0.17 / 0.69	23	3.97
t481	612	1 / 1	4.55 / 100	0.16 / 0.16	15	2.45
i7	694	13 1	0.76 / 19.40	0.14 / 1.87	41	5.91
i8	792	17 1	0.88 / 20.99	0.13 / 2.15	30	3.79
C3540	865	1 / 1	1.56 / 7.69	0.12 / 0.12	46	5.32
dalu	994	1 / 2	2.60 / 6.25	0.20 / 0.10	49	4.93
C5315	1193	1 / 1	0.65 / 0.91	0.08 / 0.08	19	1.59
i10	1990	1 / 1	0.40 / 0.56	0.05 / 0.05	36	1.81
C6288	2011	1 / 1	0.37 / 3.13	0.05 / 0.05	250	12.43
des	2736	23 / 1	0.39 / 9.43	0.04 / 0.84	83	3.03

Table 1: Percentage of relevant gates assuming generalized Pelgrom model

trials the runtime growth is linear. Another study 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. Figure 8 shows the percentage of relevant gates and their trends for Uniform and Gaussian distributions of the manufacturing variability parameters.

7 Simulation Results

In this section we present simulation results used to evaluate the non-parametric statistical timing analysis approach. We evaluate the approach using conceptually two different procedures: comparison with extensive simulation and resubstitution-base statistical evaluation.

The first evaluation method is conceptually very simple, but induces extraordinary high runtime. The

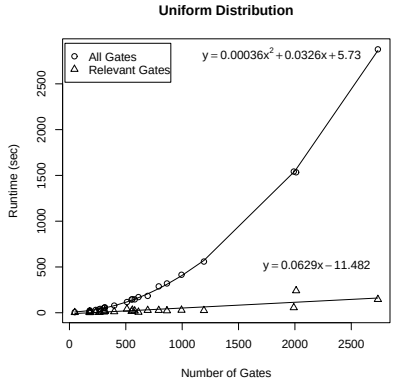


Figure 7: Scalability analysis of the non-parametric statistic and active learning-based statistical timing analysis approach.

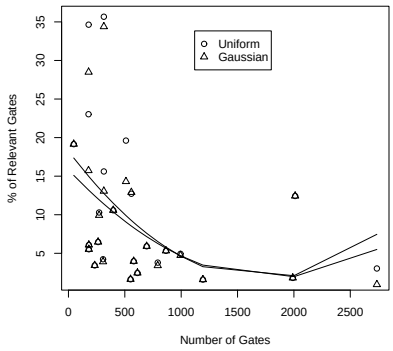
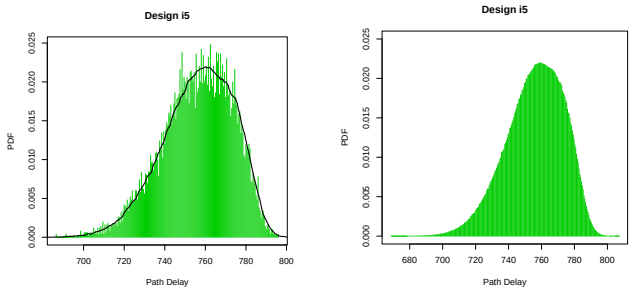


Figure 8: Percentage of relevant gates compared to the number of gates for both uniform and gaussian parameter distributions.



(a) Smooth (b) Brute-force

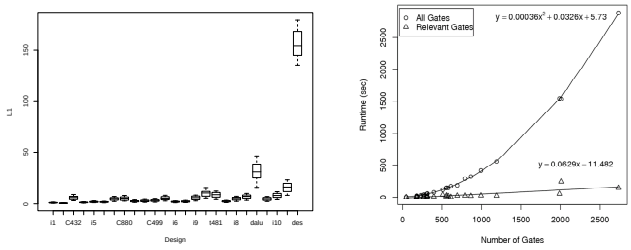
Figure 9: Evaluation of new statistical timing analysis by comparison to exhaustive simulation: (a) PDF before (shaded area) and after statistical modeling (curve) and (b) PDF after 100,000,000 MC trials.

idea is very simple, 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. Although we used 15 state-of-the-art PCs and ran our experiments for more than three weeks we were able to complete only the five smallest designs in Table 1 in the second mode. Figure 9(a) shows PDF of critical path obtained using the new approach and Figure 9(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 using the following procedure. We sort 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} sub-sample in the second comprehensive simulation.

We evaluated all designs using the following statistical procedure that consist of three steps. In the first

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-paramteric statstics-based approach.



(a) (b)

Figure 10: (a) Boxplot of the L_1 norm for 25 benchmarks obtained using resubstitution. (b) The runtime of MC simulation-based procedure for statistical timing analysis (STA) as a function of the total number gates and the number of gates that relevant for STA as identified by our STA procedure.

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 prediction abilities of the critical path PDF of each design. In the third step we do 200 cases of 10 thousand trials. Note that in all these 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. Table 3 shows the average values for six standard error measures for predicting each design from focused MC simulations and statistically processed PDFs. We see that the application of the statistical techniques reduces the error by almost a factor of 2. Finally, Figure 10(a) shows boxplots for all 200 trials of each design.

Design	L_1	L_2	L_∞	RL_1	RL_2	RL_∞
i1	1.088	1.358	5.24	0.003	0.074	0.015
i3	0.554	0.690	2.47	0.004	0.060	0.018
C432	6.004	7.519	29.10	0.002	0.120	0.007
i2	1.322	1.656	6.66	0.004	0.092	0.020
i5	1.971	2.465	9.19	0.003	0.089	0.012
x1	1.633	2.042	8.36	0.003	0.087	0.015
too_large	4.638	5.78	21.931	0.005	0.184	0.021
C880	5.350	6.705	24.23	0.002	0.142	0.011
x4	2.633	3.299	11.78	0.002	0.096	0.010
C1355	2.946	3.696	13.97	0.003	0.114	0.013
C499	3.186	3.988	14.28	0.003	0.123	0.013
C1908	5.453	6.851	26.68	0.003	0.156	0.014
i6	1.901	2.385	8.80	0.001	0.045	0.003
x3	2.253	2.825	10.71	0.002	0.078	0.008
i9	5.532	6.908	25.53	0.001	0.092	0.004
rot	10.256	12.779	48.76	0.005	0.280	0.023
t481	8.329	10.404	36.65	0.005	0.256	0.022
i7	2.294	2.884	10.35	0.001	0.050	0.003
i8	4.709	5.900	22.33	0.001	0.091	0.005
C3540	6.733	8.439	30.23	0.002	0.156	0.010
dalu	30.855	38.632	134.31	0.007	0.577	0.029
C5315	4.597	5.751	20.56	0.002	0.127	0.010
i10	7.903	9.877	36.72	0.002	0.137	0.007
C6288	15.588	19.447	69.29	0.003	0.253	0.012
des	179.749	223.486	798.19	0.009	1.588	0.037

Table 3: Analysis of the new statistical timing analysis approach using resubstituion.

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. With respect to stability, we state that while different distribution significantly impact overall timing characteristics of the benchmarks, they had only nominal impact on the runtimes of the approach. Figure 10(b) 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.

Finally, the analysis indicates that Hall's balanced sampling [14] reduces runtimes by almost two order

of magnitude, non-parametric models reduce the runtime for a given level of accuracy by four orders of magnitude at average, and that DAG pruning reduces runtimes by more than an order of magnitude. All improvements grow as the size of design increases.

Another study 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. Figure 10(b) 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.

8 Future Work and Conclusion

Two of the most important pending next steps for enabling industrial application of non-parametric statistics-based statistical timing analysis are: (i) collection of actual data and application and validation of the proposed approach using collected data from the industrial-strength designs; and (ii) the development and incorporation of statistical techniques for identification and addressing of false paths. A major phase in the first step will be the development of techniques that will be able to recover from the measurements of delays at the inputs and outputs, the actual parameters of the manufacturing variability for a given design. We plan to address this step again using non-parametric statistical techniques.

We have developed a new approach for statistical timing analysis. 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. Another important direction is the study of interaction between design steps and static timing analysis. For example, an attractive idea would be to analyze partly programable or reconfigurable designs and figure out which one from a target set of applications should be implemented on which instance of chip in such a way that the overall throughput of all designs is maximized.

References

[1] A. Agarwal, et al. Statistical timing analysis using bounds and selective enumeration. In *TAU*, pages 16–21, 2002.

[2] A. Agarwal, et al. Computation and refinement of statistical bounds on circuit delay. In *DAC*, pages 348–353, 2003.

[3] A. Agarwal, et al. Statistical timing analysis using bounds. In *DATE*, pages 10062–10068, 2003.

[4] A. B. Agarwal, et al. Path-based statistical timing analysis considering inter- and intra-die correlations. In *TAU*, pages 16–21, 2002.

[5] A. K. Sultania, et al. Tradeoffs between gate oxide leakage and delay for dual t_{ox} circuits. In *DAC*, pages 761–766, 2004.

[6] A. Agarwal, D. Blaauw, and V. Zolotov. Statistical timing analysis for intra-die process variations with spatial correlations. In *ICCAD*, pages 900–907, 2003.

[7] M. Bolt, M. Rocchi, and J. Engel. Realistical statistical worst-case simulations of VLSI circuits. *IEEE Trans. Semiconductor Manufacturing*, Aug. 1991.

[8] A. Devgan and C. Kashyap. Block-based static timing analysis with uncertainty. In *ICCAD*, pages 607–614, 2003.

[9] 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.

[10] G. S. Fishman. *Monte Carlo Concepts, Algorithms and Applications.* Springer Verlag, 1996.

[11] A. Gattiker, S. Nassif, R. Dinakar, and C. Long. Timing yield estimation from static timing analysis. In *ISQED*, pages 437–442, 2001.

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

[13] H. Chang, et al. Parameterized block-based statistical timing analysis with non-gaussian parameters and nonlinear delay functions. In *DAC*, 2005.

[14] 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.

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

[16] J. Liou, et al. Fast statistical timing analysis by probabilistic event propagation. In *DAC*, pages 661–666, 2001.

[17] J. Liou, et al. False-path-aware statistical timing analysis and efficient path selection for delay testing and timing analysis. In *DAC*, pages 566–569, 2002.

[18] J.L. Tsai, et al. A yield improvement methodology using pre- and post-silicon statistical clock scheduling. In *ICCAD*, pages 611–618, 2004.

[19] D. Kahaner, C. Moler, and S. Nash. *Numerical methods and software*. Prentice-Hall, Inc., 1989.

[20] V. Khandelwal and A. Srivastava. A general framework for accurate statistical timing analysis considering correlations. In *DAC*, 2005.

[21] L. Zhang, et al. Correlation-preserved non-gaussian statistical timing analysis with quadratic timing model. In *DAC*, 2005.

[22] L. Zhang, et al. Statistical timing analysis with extended pseudo-canonical timing model. In *DATE*, pages 952–957, 2005.

[23] M. Eisele, et al. 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.

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

[25] S. R. Nassif. Within-chip variability analysis. In *Tech. Digest*, pages 283–286, 1998.

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

[27] M. Orshansky and A. Bandyopadhyay. Fast statistical timing analysis handling arbitrary delay correlations. In *DAC*, pages 337–342, 2004.

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

[29] M. J. Pelgrom, A. C. J. Duinmaijer, and A. R. G. Welbers. Matching properties of MOS transistors. *IEEE J. of Solid-State Circuits*, 24(5):1433–1440, Oct. 1989.

[30] R. Rao, et al. Statistical estimation of leakage current considering inter- and intra-die process variation. In *ISLPED*, pages 88–89, 2003.

[31] R. Y. Rubinstein. *Simulation and the Monte Carlo Method*. Wiley, 1981.

[32] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986.

[33] R. Spence and R. Soin. *Tolerance Design of Electronic Circuits*. Addison-Wesley, 1988.

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

[35] Y. Zhan, et al. Correlation-aware statistical timing analysis with non-gaussian delay distributions. In *DAC*, 2005.

[36] J. C. Zhang and M. A. Styblinski. *Yield and Variability Optimization of Integrated Circuits*. Kluwer Academic Publishers, 1995.