Circuit Optimization using Statistical Static Timing Analysis

Aseem Agarwal
University of Michigan, Ann Arbor, MI
abagarwa@umich.edu

Kaviraj Chopra
University of Michigan, Ann Arbor, MI
kaviraj@umich.edu

David Blaauw
University of Michigan, Ann Arbor, MI
blaauw@umich.edu

Vladimir Zolotov
IBM T.J.Watson, Yorktown heights, NY
zolotov@us.ibm.com

Abstract

In this paper, we propose a new sensitivity based, statistical gate sizing method. Since circuit optimization effects the entire shape of the circuit delay distribution, it is difficult to capture the quality of a distribution with a single metric. Hence, we first introduce a new objective function that provides an effective measure for the quality of a delay distribution for both ASIC and high performance designs. We then propose an efficient and exact sensitivity based pruning algorithm based on a newly proposed theory of perturbation bounds. A heuristic approach for sensitivity computation which relies on efficient computation of statistical slack is then introduced. Finally, we show how the pruning and statistical slack based approaches can be combined to obtain nearly identical results compared with the brute-force approach but with an average run-time improvement of up to 89x. We also compare the optimization results against that of a deterministic optimizer and show an improvement up to 16% in the 99-percentile circuit delay and up to 31% in the standard deviation for the same circuit area.

Categories and Subject Descriptors

B.8.2 [Performance and Reliability]: Performance analysis

General Terms

Algorithms, performance, reliability, optimization

1 Introduction

In the nanometer regime, within-die variation has become a substantial portion of the overall variability and corner-based STA suffers from significant inaccuracy [1]. This has given rise to a new field of statistical timing analysis known as SSTA [2-4]. In SSTA, the circuit delay is considered a random variable and the objective of SSTA is to compute its probability distribution. From the cumulative distribution function (CDF) of the circuit delay, the user is then able to obtain the percentage of fabricated dies which meets a certain delay requirement, or conversely, the expected performance for a particular yield. In turn, gate or transistor sizing approaches should perform their optimization in a statistically aware manner.

SSTA-based optimization can significantly improve the yield of a design compared to deterministic optimization. This is due to the fact that deterministic optimization tends to create a so-called "wall" of critical and near critical paths since there is no incentive for the optimization to improve path delays that are not critical [5]. However, due to intra-die variability, some of these near critical paths can become critical causing the statistical circuit delay to deteriorate. Hence, deterministic optimization can result in circuits that are inferior from a yield perspective due to the lack of a correct statistical objective.

Some recent statistical optimization algorithms have been proposed in [6-8]. In [6] the authors propose a method to avoid the formation of a timing wall by purposely improving non-critical paths in the deterministic optimization. In [7], the statistical optimization problem has been considered as a nonlinear programming problem. However, this approach still has a sensitivity computation complexity of $O(n^2)$. In [8], a heuristic approach is proposed using the concept of statistically 'undominated' paths. However, since this approach is path-based, it cannot be applied to large circuits, such as c6288 from the ISCAS benchmark set [12].

In this paper, we therefore propose a new sensitivity based, statistical gate sizing algorithm. First, we introduce a new statistical objective function where the delay probabilities are weighted with a so-called "profit" function, expressing the merit of obtaining chips at a particular circuit delay. Since brute-force computation of the sensitivities is extremely expensive, we propose an efficient and exact pruning algorithm. Our pruning approach is based on a proposed theory of bounds on CDF perturbations due to sizing. We establish the useful property that these perturbation bounds can only diminish as the arrival time perturbations are propagated through the circuit. Based on this property, we find the most sensitive gate in a sizing iteration, without complete propagation of the perturbed arrival times for all gates. We obtain runtime improvements of approximately one order of magnitude compared to the brute-force approach.

To obtain additional runtime improvement, we then propose a heuristic method for computing sensitivities using statistical slack. The approach requires only a single forward and backward SSTA pass and hence has a runtime that is linear with circuit size. We show that this heuristic sensitivity computation yields approximately two orders of runtime improvement over the brute-force sensitivity computation approach. Finally, we propose a combined method, where the slack-based heuristic sensitivity computation is first used to filter out the vast majority of gate sensitivities, while the exact sensitivity computation using bound-based pruning is used to select the maximum sensitivity among the remaining set. We tested the proposed methods on benchmark circuits synthesized with an industrial 0.18μm library and showed an improvement of up to 16% in the 99-percentile circuit delay and up to 31% improvement in the standard deviation for the same circuit area, compared to a deterministic optimizer.

The remainder of this paper is organized as follows. In Section 2, we present our problem formulation and the newly proposed objective optimization function. In Section 3, we present our approach for exact sensitivity computation. In Section 4, we present the heuristic sensitivity computation as well as the combined approach. In Section 5, we present our results and in Section 6 we draw our conclusions.

2 Problem Formulation

In this section we define our modeling assumption and our SSTA approach. We also formulate the statistical optimization problem and present basic definitions and the delay model. Similar to other optimization approaches [7,8] we focus on intra-die variability in this paper.

One of the difficulties in SSTA arises from reconvergent circuit structures, which results in correlations between arrival times. In this paper, we use the bounds proposed in [2] for computation of the circuit delay CDF. It is important to note that the optimization objective is defined on this bound of the circuit delay CDF and not on the exact circuit delay CDF itself, since this would lead to prohibitive runtimes. However, we show in the result section that the optimization of the bounds, as performed by our method, results in nearly equivalent improvement of the exact circuit delay, as verified using Monte-Carlo simulation.

The random component of total delay variability is increasing due to sources such as discrete doping effects [10]. Also, the spatially correlated component of variability exhibits high correlation for distances of a few hundred microns [11] and can be modeled with corner-based analysis. Hence, modeling spatial correlation is of

DAC 2005, June 13–17, 2005, Anaheim, California, USA.

more importance for full chip analysis and is of less importance for moderate circuit blocks on which sizing optimization is performed. Similar to previous optimization methods [7,8], we therefore do not model spatial correlations in this paper, although the proposed methods for a basis from which such correlations can be incorporated.

We use a simple delay model for our experiments, based on the logic effort model. For the statistical modeling of these delays we assume that the standard deviation is a fixed percentage of the nominal delay, although our method is not restricted to this model.

2.1 Optimization Objective Function

A simple statistical objective is the mean or standard deviation of the circuit delay PDF. However, it is difficult to accurately represent the profit associated with different performance levels since the shape of the PDF is not represented. Since the proposed approach uses propagation of discretized arrival time PDFs, we obtain the entire shape of the circuit delay distribution and hence can support more general objective functions. We propose an optimization objective where the yield at a particular circuit delay is weighted with a so-called *profit function*. The total merit is then computed as a weighted sum of the profit function and the corresponding probabilities of the circuit delay PDF, as shown in Figure 1.

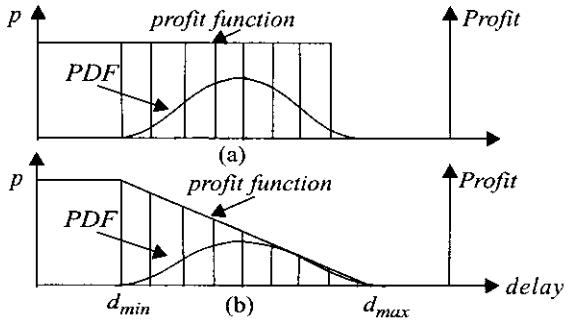

Figure 1. Optimization Objective using a Profit Function

An ASIC design typically has a strict performance constraint, where chips that fall below a specific performance level are discarded. Hence, a profit function with a step-function profile, as shown in Figure 1(a), can be used to express such a constraint. On the other hand, for high performance designs, speed binning is commonly used, and the profit associated with a part degrades gradually with the performance. Such a constraint could be expressed with the profit function having a saturated ramp profile, as shown in Figure 1(b). In this case, the profit associated with fabricated chips is maximum when circuit delay is less than d_{min} and linearly reduces to zero for chips with circuit delay greater than d_{max}.

The two profit functions shown in Figure 1 are only two particular examples of a wide variety of functions that can be utilized. Once, a designer has specified a particular profit function, the proposed optimization methods will maximize the total profit shifting the distribution, as well as by changing its shape. In Section 5, we show that by specifying different profit functions, the optimization generates different distribution shapes in order to maximize the specified objective.

3 Exact Sensitivity Computation

Here, we present the optimization algorithm using brute-force sensitivity computation and develop novel properties of sensitivity propagation based on which an efficient pruning algorithm is presented. For simplicity of explanation, we choose as our optimization objective the p-percentile confident point $T(p)$ of the delay distribution. However, the same analysis also applies to the more general profit functions.

3.1 Optimization approach and brute-force sensitivity computation.

In our experiments, we start from a minimum size implementation, and size up the maximum sensitivity gate in each iteration of the algorithm. However, the optimization can be easily extended to a partial steepest descent algorithm, where a partial gradient is computed using a small set of the most sensitive gates and gates are sized according to this gradient in each iteration. This necessitates a statistical timing analysis run for each gate in the circuit at every sizing step of the algorithm which has a runtime complexity of O(N*E) for every sizing iteration, where N is the number of nodes and E is the number of edges of graph G. This results in unacceptable runtimes. Therefore, we propose an approach where the gate with maximum sensitivity can be identified without explicit propagation of all arrival times.

3.2 Properties of sensitivity propagation

To allow for pruning of sensitivities, we now introduce the following useful definitions and properties of sensitivity propagation.

As shown in Figure 2, A_i is the CDF of the arrival time random variable at node i and A'_i is the corresponding perturbed CDF obtained by scaling up a gate. Their PDFs are denoted by a_i and a'_i, respectively. We define the difference in the p-percentile point of the CDFs A_i and A'_i as $\delta_i(p) = T(A_i, p) - T(A'_i, p)$. The maximum difference over all p is given by $\Delta_i = max_p \delta_i(p)$.

First, we assume that the perturbed CDF A'_i has the exact same shape as the unperturbed CDF A_i and differs from A_i only by a constant shift in time, i.e. $A_i(t) = A'_i(t - \Delta_i)$ and also $a_i(t) = a'_i(t - \Delta_i)$. This is assumed to be true for all perturbed CDFs. Under this assumption, we prove in Theorems 1 through 3 that the maximum difference Δ_i between the perturbed and unperturbed CDFs at a node can not increase as the perturbed CDFs are propagated through the circuit using convolution and statistical maximum. This property is useful in bounding the difference between the perturbed and unperturbed CDFs at the sink node, without complete propagation of the gate's perturbed CDF to the sink node.

Theorem 1. Convolution operation: If the arrival time PDF a_j and the perturbed a'_j at node j are given by $a_j = Conv(a_i, d_e)$ and $a'_j = Conv(a'_i, d_e)$, then $\Delta_i = \Delta_j$.

Proof : The proof is clear from the definition of convolution assuming independence. Proof given in [9].

Theorem 2. Max operation with multiple perturbed arrival times: If the arrival time CDF A_i and perturbed CDF A'_i at node i are given by, $A_i = max(A_{i1}, A_{i2})$ and $A'_i = max(A'_{i1}, A'_{i2})$ respectively, then $\Delta_i \leq max(\Delta_{i1}, \Delta_{i2})$.

Proof : Proof given in [9].

Note that the proof can be trivially extended for gates with more than two inputs.

Theorem 3. Max operation with single perturbed arrival time.

Proof : This is a special case of Theorem 2, where $\Delta_{i2} = 0$.

The above three theorems were defined assuming that the perturbed CDF has the exact same shape as the unperturbed CDF. As mentioned, this may not be true in practice and hence, we define a lower bound on the perturbed CDF which has the exact same shape as the unperturbed CDF as follows.

Definition 1. The lower bound CDF B'_i of perturbed arrival time A'_i is defined as the time shifted CDF A_i by Δ_i (Figure 2).

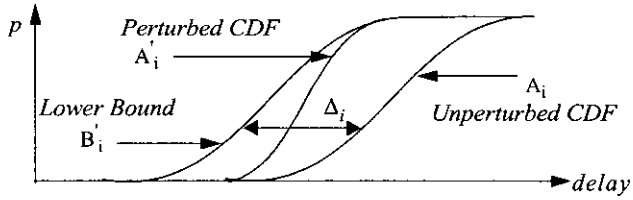

Figure 2. Arrival time CDFs at node i

Since the shape of the lower bound B'_i is the same as that of the unperturbed CDF A_i, Theorems 1 through 3 can be applied to this lower bound. Note, however, that the maximum time difference between the *lower bound* of the perturbed CDF B'_i and the unper-

turbed CDF A_i is equal to the maximum difference between the perturbed CDF A'_i *itself*, and A_i (by Definition 1). Hence, implicitly, Theorems 1 through 3 also hold for arbitrary shaped perturbations of an arrival time CDF. This allows the use of the perturbation bound Δ_i as an upper bound on the actual difference between the perturbed and unperturbed CDFs at the sink node. We can now conclude that the maximum difference between the perturbed and unperturbed CDF at the sink node is bounded by the maximum change of the perturbed and unperturbed CDFs during propagation.

3.3 Bound-based Pruning approach

The goal of an optimization iteration is to find the gate with maximum sensitivity without performing a complete SSTA run for each gate perturbation in the circuit. The idea is to propagate highly sensitive gates (i.e. gates which have a large value of S_i , which is the ratio of the change in p-percentile circuit delay per unit change in gate width) to the sink node and then use their S_i value to prune out gates which can be shown to have a lesser sensitivity using the proposed bounds. From Theorem 4 it follows that, if at any time during the propagation of arrival times of gate x, the upper bound on S_x becomes less than a previously computed sensitivity S_i of gate i, gate x can be eliminated from further consideration.

It is advantageous to identify a gate with a high sensitivity value S_i early in the analysis so that a large number of gates can be pruned. In our approach, we therefore perform level by level propagation of perturbed arrival times in an iterative manner. During every iteration the arrival times are propagated one level forward and the upper bound on S_x is recomputed. When arrival times reach the sink node, the true sensitivity S_i is computed and is used to prune other gates.

4 Heuristic Sensitivity Computation

We propose a new heuristic method to compute the sensitivities of candidate gates. We first define a so-called *impact* subgraph of a candidate gate as the subgraph of the timing graph comprised of all gates that lie on a path that passes through the candidate gate. We then break the problem into two parts: First, we compute the effect of sizing a candidate gate on the circuit delay distribution of its impact subgraph. Second, we determine the impact of the change of the impact subgraph delay on the total circuit delay. We now explain each of the two steps in more detail.

In Figure 3, a candidate gate x is shown with its impact subgraph

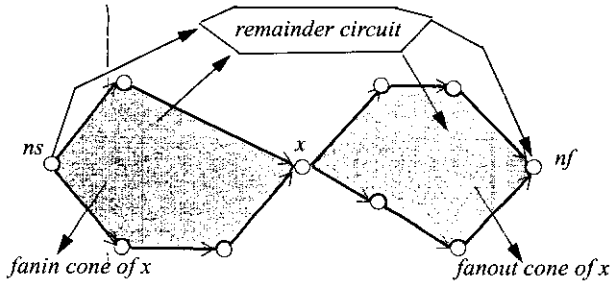

Figure 3. Impact Subgraph and Heuristic Sensitivity Computation

shown shaded. As can be seen, a delay change of gate x will only impact the arrival times in its fanout cone. Hence, we first perform two statistical timing runs on the circuit, one in the forward direction and the other in the backward. At every gate x in the circuit we then obtain a forward arrival time PDF A_{xf} and a backward arrival time PDF A_{xb}. We then compute the delay distribution A_{xc} of the impact subgraph of x by convolving its corresponding A_{xf} and A_{xb}. This represents a partial circuit delay PDF, excluding the edges which are not present in both the fanin cone and fanout cone of x. To compute the change in this subgraph delay, we then size up gate x, and recompute the perturbed forward arrival time PDF A'_{xf}, including the loading effect on its fanin gates. We then convolve this new arrival time A'_{xf} with the original A_{xb} and obtained the perturbed partial circuit delay PDF A'_{xc} of the impact subgraph.

However, in addition to the change in the delay of the impact subgraph, it is necessary to determine whether this difference will propagate to the sink node. It is difficult to determine the exact impact of the delay change on the total circuit delay and hence we propose the following effective heuristic. We first compute a statistical maximum of the impact subgraph delay PDF A_{xc} with the exact circuit delay of the total circuit PDF A_{nf}. We also compute the same maximum with the perturbed impact subgraph delay, A'_{xc}, and then obtain the difference in the p-percentile point of the two maxed PDFs.

While this heuristic is clearly not exact, it was found to work well because it adheres to the following two properties: 1) if the impact subgraph delay is small compared to the delay of the total circuit, the total circuit delay will dominate the statistical maximum, and the perturbation will have a significantly diminished impact. 2) if the difference between the perturbed and unperturbed impact subgraph delay is small, the impact on the total circuit delay will also be small.

Finally, the sensitivity is computed again as the ratio of the change in the p-percentile point to Δw_x . Since the approach requires only two statistical timing analyses to compute the forward and backward circuit delays A_{xf} and A_{xb} and since the computation of the impact subgraph delays A_{xc} and A'_{xc} is independent of the circuit size, the approach has linear runtime complexity with circuit size, and is extremely fast.

Combined approach for circuit optimization.

The slack based heuristic provides significant speedup as compared to our pruning approach, however, since the computed sensitivities are heuristic, it suffers from inaccuracies and a reduced optimization quality. Hence, we propose a combined approach, where we obtain the top k sensitivities using the slack based heuristic, and then apply our pruning algorithm to obtain the highest sensitivity gate from that set. The value k can be tuned based on the accuracy required at different points in the optimization trajectory. However, we found that with a value of k = 30, the optimization obtained nearly the same result quality as optimization using the exact bound-based pruning while providing significant speedup over that approach.

5 Results

The proposed statistical optimization methods were implemented and tested on a synthesized version of ISCAS'85 [12] benchmark circuits using a 180nm commercial cell library. Intra-die process variation was modeled using a truncated Gaussian gate delay distribution. The standard deviation was 10% of the nominal delay and the distribution was truncated at the 3 sigma point. The deterministic optimization that we use for comparison is based on coordinate descent method. However, we have also performed deterministic optimization using MINOS, which is a non-linear optimizer and verified the accuracy of our coordinate descent method.

Table 1 shows a comparison between the bound-based pruning, statistical slack-based approach, combined approach and deterministic optimization for the 99-percentile circuit delay point, after performing 800 sizing iterations, in *columns* 2-8 (Delay units are in ns). *Column* 2 and 3 show the 99-percentile delay obtained from deterministic and statistical slack-based approach, respectively. The % improvement obtained from slack-based optimization over deterministic optimization is shown in *column* 4. *column* 5 and 7 show the 99-percentile delay obtained using bound-based and our combined approach, respectively and *column* 6 and 8 show its % improvement over deterministic optimization. The average improvement is 7.6% over all benchmarks with a maximum of 16.5%. *Column* 9 shows the % improvement in the sigma of the circuit delay PDF, with maximum of 31.4%.

Table 2 shows a comparison of runtimes between brute force statistical optimization and our accelerated approaches. Our bound-based pruning approach provides an average runtime improvement of up to 20x for large circuits. In *column* 2 and 3, we report the average runtime per iteration (computed over 800 iterations) using the brute force and our bound-based pruning approach, respectively. *column* 4 shows the runtime improvement factor. In *columns* 5 and 7, we show the average runtime per iteration for our combined approach and slack-based approach, respectively and *columns* 6 and 8 show their improvement factors over brute-force. Our combined

Table 1. Results for the 99-percentile delay pt.

name	det.	slack-based		bound-based		combined approach		
	delay	delay	%impr	delay	%impr	delay	%imp	%sigma imp.
c432	3.45	3.40	1.4	3.25	5.8	3.25	5.8	12.2
c499	4.05	3.48	14.0	3.38	16.5	3.38	16.5	31.4
c880	4.18	4.04	3.3	3.94	5.74	3.94	5.74	13.8
c1355	4.70	4.25	9.5	4.10	12.7	4.10	12.7	30.7
c1908	6.20	6.02	2.9	5.82	6.1	5.82	6.1	10.8
c2670	3.61	3.55	1.7	3.50	3.0	3.50	3.0	3.0
c3540	5.98	5.80	5.2	5.70	6.9	5.70	6.9	13.5
c5315	5.90	5.70	3.4	5.40	8.47	5.40	8.47	14.8
c6288	15.8	15.5	1.9	15.05	4.75	15.05	4.75	23.0
c7552	8.10	7.80	3.8	7.60	6.17	7.60	6.17	13.4

approach matches almost exactly with brute-force and shows runtime improvement of up to 89x.

Table 2. Results for the runtime improvement

Circuit name	Average time per iteration (sec)						
	brute-f	b.b. prune	imp. f	comb.	imp. f	slack	imp. f
c432	5	1.21	4.13	0.78	6.4	0.49	10.2
c499	90	19.9	4.52	3.8	23.7	1.75	51.5
c880	15	3.37	4.45	1.07	14.0	0.85	17.6
c1355	95	19.8	4.79	3.8	25.0	1.6	59.4
c1908	102	22	4.63	5.97	17.0	2.1	48.6
c2670	43	4.47	9.62	1.36	31.6	1.2	35.8
c3540	194	23	8.43	5.8	33.4	3.4	57.0
c5315	403	33	12.2	6.8	59.3	4.2	96.0
c6288	3600	180	20.0	50.3	71.6	35.0	103
c7552	1190	87	13.68	13.4	89.0	8.4	142

Figure 4 shows the area-delay curve using our combined approach and deterministic optimization for c3540. The 99-percentile points of the circuit delay CDF are plotted on the x-axis and the corresponding total gate size value on the y-axis, for every sizing iteration. We have also plotted the 99-percentile points of the circuit delay using Monte Carlo simulations. As shown, there is a very small difference between the bounds and Monte Carlo results.

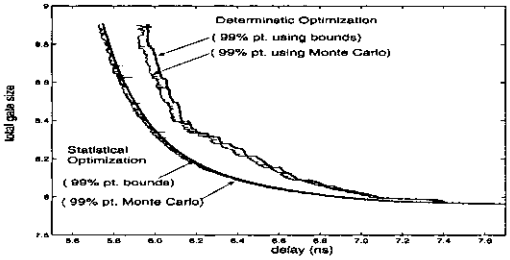

Figure 4. Area- delay curve for c3540 (compared w/ M.C.)

In Figure 5, we show the circuit delay PDF as obtained by the deterministic and statistical optimization methods, after the total gate size has been increased by 33% for c880. We can see that statistical optimization shifts the entire distribution to the left, along with reducing the variability of circuit delay. The deterministically optimized PDF is obtained using an optimal solution of the nonlinear optimization package MINOS.

In Figure 6, we show circuit delay PDFs obtained by using our proposed linear cost function and the 99-percentile delay objective. We can observe that the PDF shape changes according to the objective function. The circuit delay PDF obtained by applying the cost function is better optimized for the cost than the 99-percentile delay objective.

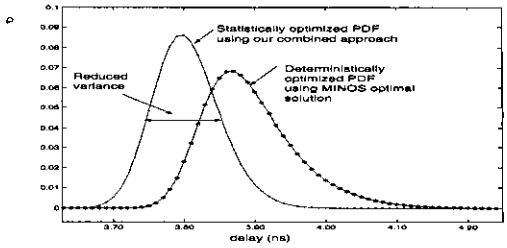

Figure 5. Output PDFs obtained for the same area

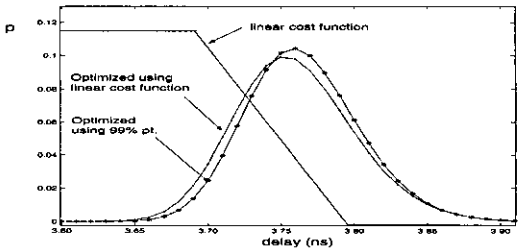

Figure 6. circuit delay PDF for c880 using different cost functions

6 Conclusions

We have demonstrated the need for a fast statistical optimization algorithm and shown that there is a clear advantage in using statistical optimization compared to a deterministic one. We proposed a fast pruning-based sensitivity computation which is exact in comparison with a brute force sensitivity computation but provides significant speedup. We also proposed a statistical slack-based heuristic which is extremely fast, and a combined approach which provides a significant speedup with negligible loss in accuracy. Our results show an average runtime improvement up to a factor of 89 using the combined approach over a number of test cases. Future work includes extending this framework to include spatial correlations.

7 Acknowledgements

This research was supported by SRC contract 2001-HJ-959 and NSF grant CCR-0205227.

8 References

[1] S. Nassif, "Delay Variability: Sources, Impacts and Trends," Proceedings of ISSCC, 2000.

[2] A. Agarwal, D. Blaauw, V. Zolotov, S. Vrudhula, "Computation and Refinement of Statistical Bounds on Circuit Delay," DAC 2003.

[3] A. Devgan, C. Kashyap, "Block-based Static Timing Analysis with Uncertainty," ICCAD 2003, pp.607-614.

[4] H. Chang, S. Sapatnekar, "Statistical Timing Analysis Considering Spatial Correlations using a Single Pert-like Traversal," ICCAD'03.

[5] H. Hashimoto, H. Onodera. "Increase in delay uncertainty by performance optimization", ISCAS 2001, pp. 379-382.

[6] X. Bai, C. Visweswariah, P. N. Strenski, and D. J. Hathaway. "Uncertainty-aware circuit optimization", *DAC 2002*, pp. 58-63

[7] E.T.A.F. Jacobs, M.R.C.M Berkelaar. "Gate sizing using a statistical delay model", DATE 2000, pp. 283-289.

[8] S. Raj, S. Vrudhula, J. Wang. "A methodology to improve timing yield in the presence of process variations", DAC 2004.

[9] A. Agarwal, K. Chopra, D. Blaauw, "Statistical Timing Based Optimization using Gate Sizing", DATE 2005.

[10] K. Bowman et. al., "Impact of die-to-die and within-die Parameter Fluctuations on the Maximum Clock Frequency Distribution for Gigascale Integration," IEEE J. Solid-State Circuit, Feb 2002.

[11] Personal communication, Kerry Bernstein, IBM Corp, Burlington, VT.

[12] F. Brglez, H.Fujiwara, "A Neutral Netlist of 10 Combinatorial Benchmark Circuits", Proc. ISCAS, 1985, pp.695-698.