A Lower Bound Computation Method for Evaluation of Statistical Design Techniques

Vineeth Veetil
Dept of EECS
University of Michigan
Ann Arbor, MI
tvvin@umich.edu

Dennis Sylvester
Dept of EECS
University of Michigan
Ann Arbor, MI
dennis@eecs.umich.edu

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

ABSTRACT

Increase in variability in the nanometer era has contributed to pessimistic guardbands for conventional circuit design techniques that optimize at worst-case process corners. Smart deterministic approaches have been proposed that employ statistical timing analysis to reduce pessimism in the guardbands while retaining the deterministic nature of the algorithms. Other statistical optimization techniques focus on algorithms to maximize robustness of design while being aware of variability. It is not clear how much improvement can be gained using the latter set of approaches over more simple deterministic approaches. This work presents a new lower bound to evaluate these statistical optimization techniques, drawing inspiration from recent advances in sampling based SSTA. We prove that the presented lower bound gives the minimum possible area that can be achieved for a design while meeting a particular timing yield, which is the percentage of die that meeting a specified timing constraint. We then compare several statistical design optimization approaches, including one proposed in this paper called SLOP, against the computed lower bound. We show that even the simplest statistical optimization approaches produce area results which are, on average, within 9.6% of the lower bound while the best ones performed only marginally better, reaching within 3.7% of the bound. This demonstrates that the proposed bound is a close bound. In addition, it also shows that the existing optimization methods have nearly exhausted the obtainable improvement from being statistically aware and mostly provide trade-offs in runtime speed.

General Terms

Algorithms, Performance, Design.

Keywords

Statistical Optimization, Monte Carlo, Gate Sizing.

1. INTRODUCTION

Process parameter variations have taken on increasing importance in nanometer-scale CMOS due to the approaching of fundamental manufacturing limitations [1]. Rather than using simple corner models that capture worst-case behavior at the device level (and lead to large guardbands) , modern CAD tools have moved towards a more probabilistic view of circuit timing behavior. This is demonstrated by recent research on analysis and optimization of designs in the face of variability. In timing analysis, there are two primary approaches to incorporating process parameter uncertainty. The first is to perform statistical static timing analysis (SSTA) by modeling gate delays as distributions that are functions of underlying process parameters. These distribution functions are then distributed through the circuit graph to compute the overall circuit delay distribution [2-3]. We refer to these approaches as traditional SSTA. The second approach is Monte Carlo based SSTA, which involves selection of samples of the process variation space to directly obtain statistical distributions of circuit timing behavior. The application of Monte Carlo for statistical timing analysis was discussed in [4], where it was shown that MC SSTA is accurate even in scenarios with high dimensionality in the process variation space, where traditional SSTA has difficulties. The difficulty with a straightforward implementation of MC SSTA is the large sample size required to achieve good accuracy. Recent research has focused on techniques to reduce sample size for MC SSTA, drawing upon similar experiences in statistics and computational finance [5-7].

At the same time, recent research has paid significant attention to statistical design techniques, where the focus is on optimizing designs to increase their robustness under variability while minimizing any increase in cost (e.g., area or power). Optimization at worst-case corners leads to pessimism and large guardbands. Therefore, smart deterministic algorithms were introduced where the key observation is that the pessimism incurred by worst-case corner approaches is mainly due to an inability to set appropriate guardbands (i.e., timing constraints) for optimization in a deterministic setting, rather than the quality of the algorithm itself [8-11]. Therefore, it is sufficient to augment a deterministic algorithm with statistical analysis to minimize pessimism in the timing constraint used for optimization. The authors in [12] summarize these approaches and propose a technique to capture WID variation effects without explicitly modeling them in the optimization procedure. Here, a circuit-level guardband for the target timing constraint is used to capture WID (within-die) variation effects. A conventional transistor sizing algorithm similar to [13] which captures only D2D effects, is then used to minimize cost to meet this guardbanded target. The appropriate guardband is found by sweeping its value and repeating the conventional optimization followed by SSTA at each point, until timing yield is met.

Other techniques have focused on explicitly capturing statistical sensitivities at the device/gate level or introducing spatial correlation-aware margins at the device level to reduce the pessimism found in worst-case corner optimization [14-16]. The authors in [14] propose a yield optimization technique using a gradient-based non-linear optimizer. An efficient heuristic is proposed to compute yield gradient. In [15], the robust statistical optimization problem is formulated as a second order conic program (SOCP). A linear relationship between delay and parameters affecting variability is assumed. A piecewise linear gate delay model as a function of gate size is constructed to enable the problem formulation as an SOCP. In robust geometric programming (RGP) [16], a worst-case corner approach is used to incorporate process variation effects. A Geometric Programming model is used to capture the delay of gates as an analytical function of design parameters and parameters representing parasitic effects. The effect of variations is included by adding appropriate margins to the delay constraints at the output pin of each logic gate.

These algorithms provide reasonable solutions to the problem at hand, which is to arrive at a design that meets performance requirements with sufficient confidence given process variability.

However it is not clear the scope of improvement possible by using statistical optimization approaches rather than smart deterministic algorithms such as [12]. Given the additional runtime costs of the fully statistical approaches, it would benefit designers and future researchers to know the potential improvements available to statistical techniques.

The major contribution of this work is a lower bound computation method to compare the Quality of Results (QoR) of statistical design techniques under process variations. This method takes its inspiration from recent developments in statistical timing analysis where a sample-level view of the process variation space is taken [5-7]. We show that the lowest cost for any design to meet a specified timing yield objective is bounded by a theoretical limit. This limit is related to the exact solution obtained if different samples in the process variation space were to be optimized deterministically. Given a set of samples in the variation space, the optimal design to meet a deterministic timing T while fixing the process parameters at each sample can be obtained using an exact optimization technique. We show that for a large enough sample size, the x^{th} percentile of the cost (area, power) of the designs obtained by optimizing each individual sample to meet timing constraint T is a lower bound for any design that meets T with a specified timing yield of *x%*.

Second, in the same spirit as the lower bound computation, we propose a statistical design technique that draws upon a sample level perspective of the variation space - Sample Level Optimization in Parallel (SLOP). As the name indicates, different samples in the process variation space are optimized in parallel using two phases. In the first phase, a straightforward deterministic optimization is performed for each sample. These results are then fine tuned in the second phase by shifting the focus of the algorithm to the optimization of an intelligently selected high percentile sample taken from phase one, such that the timing yield is met. Among the solutions thus obtained at each sample, the lowest cost solution is selected. The technique is highly amenable to parallelism on multi-core machines and GPUs.

We compare the results obtained from SLOP and two other techniques proposed in literature, Burns [12], and Robust Geometric Programming or RGP [16], against the lower bound computed using the proposed technique. We show that the solutions obtained from SLOP and RGP are close to the theoretical limit for the cost. Further improvements possible to solutions computed by the Burns, SLOP and RGP methods are at most 9.6%, 7.5% and 3.7% respectively, on average for the benchmark circuits studied. The improvement in quality of solution for SLOP and RGP comes at the cost of an average of 7.4× and 41.5× increase in runtime, respectively compared to the Burns approach. Therefore, we conclude that smart deterministic approaches are sufficiently accurate while incurring low runtime cost. At the same time both Burns and SLOP also offer possibilities for massive parallelization on GPUs and multiprocessor systems.

The rest of the paper is organized as follows. Section 2 discusses previous work on exact optimization of a design at a fixed process corner. Section 3 explains the proposed technique to obtain a theoretical limit for the cost of a design to meet a given timing yield. Section 4 describes the proposed Sample Level Optimization in Parallel (SLOP) approach. Section 5 presents results and conclusions are presented in Section 6.

2. EXACT DETERMINISTIC OPTIMIZATION

Given the growing parallelism available in modern CPUs and GPUs, there is interest in using sample-based approaches to perform statistical timing and power analysis and optimization [5-7, 24]. When examining specific samples in the process variation space these approaches rely on finding the optimal solution for that given sample (die). This section discusses the existing literature on exact optimization of a design at a given point in process space. It is theoretically possible to obtain cost lower bounds at a process corner using branch and bound or simulated annealing techniques. However, in practice, these techniques exhibit very high runtimes. Other techniques to obtain the optimal design at a process corner make assumptions regarding the gate delay model. In [17], the authors propose an approach for exact transistor sizing. The method involves two phases. In the first phase, called the *D-phase*, incremental changes in delay are assigned to each node of the circuit graph. This is formulated as the dual of a min cost network flow problem and is solved exactly. In the second phase, or the W-phase, feasible transistor sizes are calculated to incorporate the node delay changes assigned in the previous phase with minimal increase in cost. This phase assumes that the transistor delay is expressible as a sum of *simple monotonic functionals*. A *simple monotonic function* applied to this case has the property that it is monotonically decreasing in one transistor parameter and montonically increasing in all other transistor parameters. Reference [18] discusses a *Geometric Programming* approach to solve circuit design optimization problems. Geometric Programming (GP) models can address a wide variety of integrated circuit design problems [18]. Also, commercial solvers are available that can handle large-scale GPs efficiently [19]. Therefore, we focus on a GP-based approach to obtain the cost lower bound for a design.

In a GP model, the objective function and the constraint functions are expressed as a general class of functions called *posynomial functions*. A posynomial function is a sum of *monomials.*

A GP can be expressed in the following form

minimize $f_0(x)$

subject to :

$$f_i(x) \le 1 \qquad i=1,\dots,m$$

$$g_i(x) = 1 \qquad i=1,\dots,p$$

Here $f_i(x)$ are posynomial functions, $g_i(x)$ are monomial functions, and x_i are optimization variables.

Generalized Geometric Programs (GGPs) [18] extend the formulation to a more general class of functions called *generalized posynomials*. A generalized posynomial is any function that is expressible using the operations of summation, multiplication, positive (fractional) powers, and maximum of posynomials. Examples of generalized posynomials are illustrated below:

$$h_1(x) = f_1(x)^{a_1} f_2(x)^{a_2} \qquad a_1, a_2 > 0$$

$$h(x) = \max(h_1(x), f_2(x))$$

where $f_1(x)$, $f_2(x)$ are posynomials and $h_1(x)$, $h(x)$ are generalized posynomials.

The formulation of an optimization problem as a GGP requires modeling the objective and constraint functions as generalized posynomials. Techniques to approximate practical functions using generalized posynomials are discussed in [18]. In this work, we employ a *Max-Monomial* fitting approach for this purpose. A max-monomial function has the form

$$h(x) = \max_{k=1,\dots,K} f_k(x) \qquad (1)$$

where $f_k(x)$, k=1,…,K are monomials. A heuristic algorithm for finding the max-monomial approximation to a data set for a fixed K is provided in [18]. The algorithm selects a subset of the data for each monomial where the monomial is the maximum compared to the other monomials. The monomial fit for this subset of data is then improved using a simple least squares linear fit after performing logarithmic operations on both sides.

With the max-monomial representation for delay, the optimization problem at a process corner can be formulated as a GGP:

$$\text{minimize} \sum_{i=1}^{N} x_i$$

s.t.:

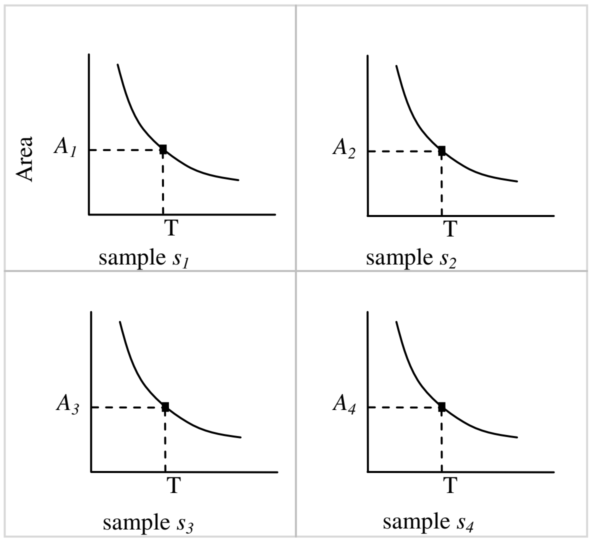
$$\begin{aligned} &\mathrm{AT_j} + \mathrm{d_{i,j}} \le \mathrm{AT_i} \\ &\mathrm{d_{i,j}} = \mathrm{h}(\mathrm{x_i}, \mathrm{s_j}, \mathrm{p}_{i,j}, \vec{r}_i) \qquad \forall j : e(j,i) \in E \\ &\mathrm{AT_i} \le \mathrm{T} \end{aligned}$$

where $h(.)$ is a max-monomial function. x_i is the size of the gate at node i, s_j is the slew at node j and $p_{i,j}$ represents the parasitic values at nodes i and j. $\vec{r}_i$ is the vector of process parameter values at the sample for the gate at node i. E is the set of edges in the circuit graph. T is the specified timing constraint, AT_i is the arrival time at node i, and $\mathrm{d_{i,j}}$ is the delay of edge $e(j,i)$.

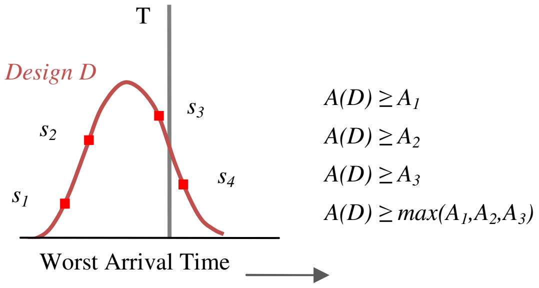
3. LOWER BOUND FOR DESIGN WITH STATISTICAL TIMING YIELD CONSTRAINT

This section describes a new technique to obtain a lower bound for the typical case cost of a design while meeting a statistical timing constraint T with yield x. The lower bound is computed using results from independent exact optimization of samples in the process variation space. The exact optimization technique used at each sample was described in Section 2. We consider the distribution of costs obtained by exact optimization of each sample in a sample set that is sufficiently large and adequately captures the distribution of process parameters. We show that the cost of any design that meets the constraint T with timing yield x is higher than the x^{th} percentile of the cost distribution.

Figure 1 illustrates the approach for a sample set $S = \{s_1, s_2, s_3, s_4\}$. Each of the samples in S are optimized to meet a timing constraint T with resulting (provably minimal) costs A_1, A_2, A_3, A_4, respectively. D is any design that meets the timing constraint T with respect to the sample set S with a yield of x=75% and area $A(D)$. Note that D meets the constraint T at samples s_1, s_2 and s_3. Therefore, $A(D)$ must be higher than A_1, A_2 and A_3. However, the same cannot be said of A_4. Since the 75th percentile of the distribution A_i, i = 1...4 is at least equal to max(A_1, A_2,A_3), $A(D)$ must be at least equal to the 75th percentile of distribution A_i with respect to set S. For a sample set S that captures the process parameter distribution accurately, this means that the area of any design D that meets constraint T with timing yield x is lower bounded by this value, subject to an error related to estimation of timing yield using the sample set. A more rigorous approach considering this error is presented below.



(a) Optimal area (exact) vs. timing constraint at samples in set *S*.



(b) Design D

Figure 1. Illustration of Theorem 1. A_i represents the optimal cost solution at sample s_i. Design *D* meets timing constraint T for the 75th percentile of the given sample set. The cost of this design is A(D) and exceeds A_i, *i*=1,…,3 as *D* meets the timing constraint at samples s_i, *i*=1,…,3. Therefore, A(D) exceeds the 75th percentile of the distribution A_i, *i=1,…,4* for the given sample set.

Let *S* be a set of samples in the process variation space. $A_i(T)$ denotes the lowest typical case cost for a design to meet timing constraint T at sample $s_i \in S$. Let $A_S^x(T)$ denote the x^{th} percentile of the distribution $A_i(T)$ with respect to sample set *S* for constraint T. *n(S)* denotes the number of elements in *S*. $t_S^x(D)$ is the x^{th} percentile of the worst circuit delay for design *D* w.r.t. sample set *S*.

Theorem 1. Given a design D, the following is true:

$$t_S^x(D) \leq T \Rightarrow A_S^x(T) < A(D)$$

In other words, $A_S^x(T)$ is a lower bound for the typical case cost of a design to meet T for the x^{th} percentile of the worst arrival time distribution *with respect to sample set S*.

Proof: Let *A(D)* denote the nominal cost of design *D*. If *D* meets the timing constraint T w.r.t sample $s_i \in S$, then

$$A_i(T) \leq A(D) \qquad (2)$$

Let *D* satisfy T for the x^{th} percentile of the worst arrival time distribution with respect to sample set *S* or $t_S^x(D) \leq T$. In other words , *D* meets timing constraint T for at least *x*% of the samples in *S*. It follows from (2) that

$$\exists C \subset S: \ A_i(T) \leq A(D) \ \ \forall s_i \in C$$

$$n(C) \geq n(S) * \frac{x}{100}$$

i.e., $A_i(T) \leq A(D)$ holds for at least *x*% of the samples in *S*.

This in turn imposes the following constraint on the x^{th} percentile of the distribution $A_i(T)$ with respect to sample set S.

$$A_S^x(T) \leq A(D) \qquad (3)$$

Theorem 2. Let S_n be a set of *n* samples in the process parameter space such that $\left|t_{S_n}^x(D') - t^x(D')\right| < \varepsilon$ for any design *D'* in the design space, where $t^x(D')$ is the x^{th} percentile worst circuit delay of *D'*. For a given design *D* which satisfies $t^x(D) \leq T$, $A_{S_n}^x(T+\varepsilon) \leq A(D)$

Proof:

Given $\left|t_{S_n}^x(D) - t^x(D)\right| < \varepsilon$

$$t^x(D) \leq T \Rightarrow t_{S_n}^x(D) < T + \varepsilon$$

$$\therefore A_{S_n}^x(T+\varepsilon) \leq A(D) \qquad \text{(Theorem 1)}$$

Note that the proof for Theorem 1 assumes $n(S) * \frac{x}{100}$ is an integer. This assumption can be removed easily. The proof is omitted for brevity.

Theorem 2 suggests a technique to obtain the lower bound on the cost of a design to meet a statistical timing yield constraint. This is summarized below for the case of *x*% timing yield at T.

Algorithm 1
Generate set of samples S_n in the variation space according to the process parameter distribution. Estimate error bound ε for computation of the x^{th} percentile of worst circuit delay using set S_n.
Obtain cost $A_i(T+\varepsilon)$ for the design optimized to minimize typical case cost while meeting constraint T at each corner/sample $s_i \in S_n$.
Obtain the x^{th} percentile of the cost distribution $A_{S_n}^x(T+\varepsilon)$.

Note that as the sample size n→∞, $\varepsilon \to 0$ and a closer lower bound is obtained. Also, smart sampling techniques have been proposed in literature to obtain moments and percentile values of the circuit delay distribution with a low sample size, and the error incurred in estimation of the moments and percentile values using some of these techniques have been discussed [5-7]. These techniques can be used to perform efficient computation of the lower bound with few samples.

4. SAMPLE LEVEL OPTIMIZATION IN PARALLEL (SLOP)

This section proposes a Sample Level Optimization in Parallel (SLOP) technique for statistical circuit optimization. The optimization problem addressed here is to find the minimum cost solution for a design to meet a given timing constraint at the x^{th} percentile of its worst case circuit delay distribution considering process variations. SLOP takes a sample level view of the process variation space. First, samples are generated that are essentially virtually fabricated dies. Within each virtual die, the process parameters are fixed and deterministic optimization is then performed. The optimization steps are based on a greedy strategy such as in [13] where gates in the critical path are selected and sized iteratively. The selection criterion is based on a metric that estimates the gain in circuit speed for a unit upsizing of the gate. Each virtual die can be optimized in parallel. Once the optimization steps are complete, the design with the best cost among the solutions from different virtual dies is selected.

Figure 2 illustrates the approach. SLOP consists of two phases, *S-phase* and *HPS-phase*. Note that SLOP progresses for each sample in parallel, and hence operates on a single virtual die.

- *S-phase or Sample Phase.* In this phase, a virtual die is optimized using a greedy strategy to meet the timing constraint T. The result is a set of gate sizes that will meet the performance constraint for this specific point in the process variation space (i.e., sample).
- *HPS-phase or High Percentile Sample Phase.* SSTA is performed on the design returned from the first phase and the x^{th} percentile sample is selected. It is not surprising that the design from the S-phase will not satisfy the timing constraint T for the entire process variation space. This phase seeks to zoom in on the portion of the process variation space that poses the greatest difficulties for the design from the S-phase, and re-optimize. Monte Carlo Sampling based SSTA is employed to select the x^{th} percentile sample. Analytical SSTA techniques cannot be used here as they do not provide information at the sample/virtual die level as required by SLOP. We use a smart sampling based SSTA approach proposed in [5] called SH-QMC for this purpose. SH-QMC provides good accuracy in computing a high percentile of the worst arrival time with a small numbers of samples (100-200). The selected sample is then optimized to meet the constraint T using a similar greedy approach as in the *S-phase.* The results from different optimization runs (i.e., the parallel virtual die) are compared and the design meeting T for the x^{th} percentile worst case delay with minimum cost is chosen as the final solution.

5. RESULTS

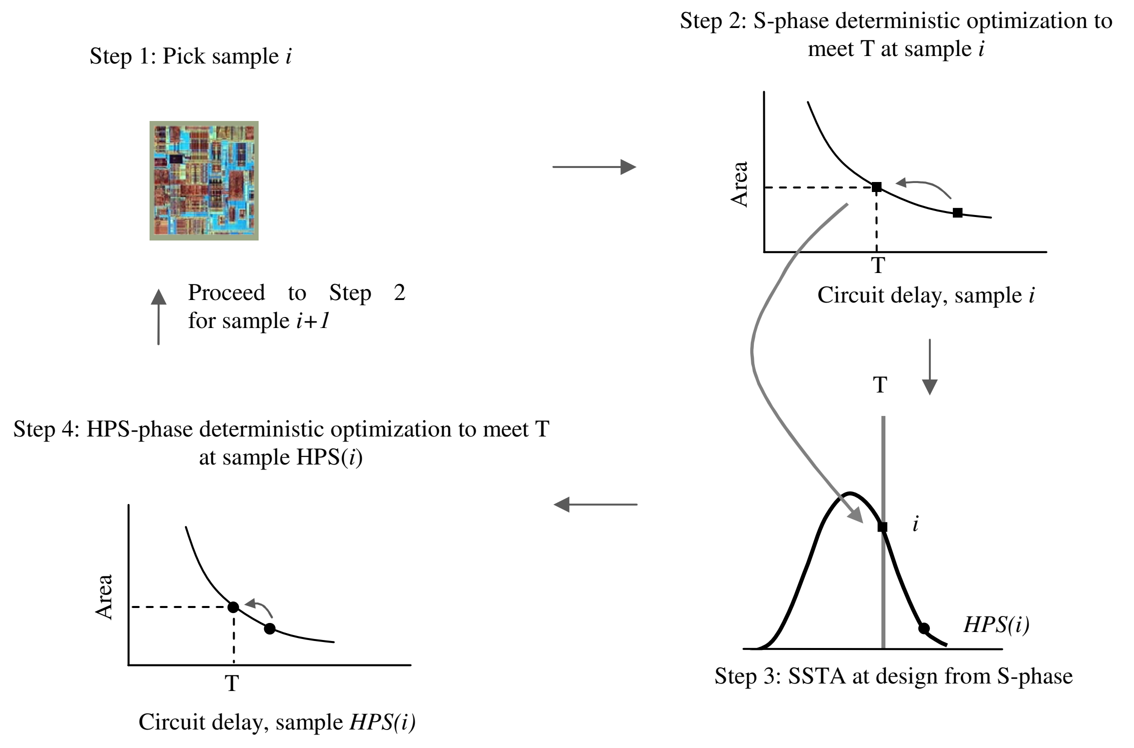

Step 1: Pick sample *i*
Step 2: S-phase deterministic optimization to meet T at sample *i*
Area
T
Circuit delay, sample *i*
Proceed to Step 2 for sample *i+1*
T
i
HPS(i)
Step 3: SSTA at design from S-phase
Step 4: HPS-phase deterministic optimization to meet T at sample HPS(*i*)
Area
T
Circuit delay, sample *HPS(i)*

Figure 2. SLOP overview: Each virtual die *i* is optimized in parallel. The optimization consists of an S-Phase and an HPS-Phase. In the S-Phase, the virtual die is optimized to meet the timing constraint T using a greedy approach. In the subsequent HPS-Phase, the xth percentile sample of the design *i* optimized in S Phase, called HPS(*i*), is selected. If T is already met at HPS(*i*), optimization terminates. Else, the design is optimized further, this time with sample HPS(*i*) constrained to meet a timing T. The best among parallel solutions is selected.

Simulation results in this paper are based on a 65nm industrial technology library. The implementation considers channel length, oxide thickness, and threshold voltage variations as process parameters. Inter-die, spatially correlated intra-die and uncorrelated random components of variation are considered for each parameter. The relative amounts of process parameter variation among die-to-die, spatially correlated, and random sources have been reported in the literature [20-22]. An increase in systematic die-to-die component of variation accompanied by an increase in random WID variation has been reported in [20] for a 45nm technology compared to 90nm. Systematic component of frequency variation is estimated at 52% in [21] for a 65nm technology node. In our implementation, the standard deviation in channel length variation is 5%. The standard deviation for oxide thickness is 1.3%. The contribution of D2D components of channel length and oxide thickness are fixed at 50% while dividing the random and spatially correlated WID components equally. The threshold voltage variation is modeled based on [22], where a Pelgrom model is used to compute the random component of threshold voltage variation. The number of grids in the spatial correlation model for individual circuits is varied linearly with post-placement area starting from 2 × 2 for the smallest circuit to 16 × 16 for the largest circuit. This corresponds to a grid area of approximately 40µm × 40µm for all the circuits.

Simulations are performed on ISCAS85 benchmark circuits [23] and three additional large benchmark circuits: Viterbi Decoder 1 (VD1), Viterbi Decoder 2 (VD2), and USB 2.0 Core (USB) with gate counts ranging from approximately 15,000 to 35,000. We perform synthesis and APR on all the benchmarks using commercial tools. Simulations were performed on a 16-core 2.0GHz AMD machine with 32GB RAM.

Table 1 shows the lower bound in area for benchmark circuits to meet a specified timing constraint with a timing yield of 99%. As noted in Section 2 the lower bound computation is constrained by the ability to obtain a perfect model of gate delay as a function of gate size, parasitics, and process variation effects. Hence, we model gate delays in the standard cell library with the approach described in Section 2 that uses the *max-monomial* fitting algorithm. We use 11 monomial terms in our implementation. Further increases in the number of monomials are limited by the physical memory constraints on the server system for the largest benchmark circuits studied. For the purpose of lower bound computation, we use this gate delay model in the results for the various approaches considered. This enables us to make conclusions about the robustness of each of these techniques in comparison with the exact lower bound. Timing constraints were set such that the hardware intensity of each benchmark circuit is 1.0. The hardware intensity of a design is defined as the magnitude of the ratio of percentage change in cost to percentage change in timing constraint. The solutions obtained from three different approaches – Burns [12], Robust Geometric Programming (RGP) and the proposed SLOP technique are

Table 1. Comparison of Burns, RGP and SLOP approaches against the lower bound for area at benchmark circuits. RGP is implemented on a CPU. Burns and SLOP are implemented on a CPU with a GPU co-processor, to utilize the parallelism available in the algorithms.

Circuit	# gates	Area	Area/Runtime(s)			Area Lower Bound	Δ Area w.r.t. Lower Bound (%)			
		Worst Corner	Burns	RGP	SLOP		W.C.	Burns	RGP	SLOP
C432	256	687.6	647.6/0.3	614/2.2	635.1/0.4	588.1	16.9	10.1	4.4	8.0
C499	544	1234.7	1168.2/0.4	1126/5.8	1163.8/1.2	1065.3	15.9	9.7	5.7	9.2
C880	500	1648.2	1558.3/0.7	1506/10.3	1541.4/1.5	1448.8	13.8	7.6	3.9	6.4
C1908	603	1654.3	1555.9/0.5	1492/6.6	1533.8/1.4	1454.1	13.8	7.0	2.6	5.5
C2670	780	2217.6	2095.7/0.5	1952/9.2	2084.5/2.1	1891.0	17.3	9.3	3.2	10.2
C3540	1163	3653.2	3477.1/1.4	3269/19	3422.1/6.5	3169.7	15.3	9.7	3.1	8.0
C5315	1692	5595.6	5236.1/1.7	4980/31	5206.5/9.7	4871.6	14.9	7.5	2.2	6.9
C6288	3834	8923	8447.1/6.1	7847/75	8024/32.1	7639.0	16.8	10.6	2.7	5.0
C7552	2152	6128.9	5833.7/3.8	5513/38	5785.9/23.6	5329.1	15.0	9.5	3.5	8.6
VD1	14503	45780.6	37302/16.9	35593/310	36933/129.7	34598.5	32.3	7.8	2.9	6.7
VD2	34082	97959	81308/108	79932/3846	80265/794.7	78305.7	25.1	3.8	2.1	2.5
USB	32898	107131	90866/83.1	79856/4950	84291/658.4	74275.0	44.2	22.3	7.5	13.5

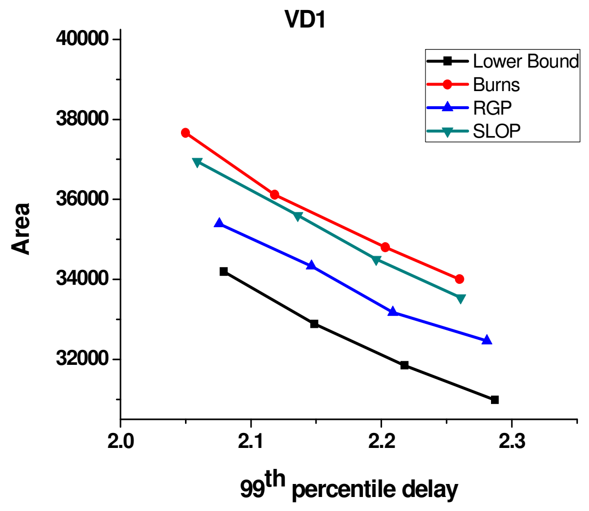


Figure 3. Comparison of sizing curves for Burns, RGP and SLOP against the lower bound for area computed at varying timing constraints for the 99% timing yield

compared to the lower bound. The runtime values reported for Burns and SLOP are for implementation on a CPU using a GPU as a co-processor to exploit the parallelism available in these algorithms. The GPU is an Nvidia Tesla S1080 system with 4 cards. On average 12.8× and 6.4× improvements are obtained through such parallelism compared to a purely CPU-based implementation for Burns and SLOP, respectively. RGP is implemented on a CPU since no straightforward source of parallelism is available in the algorithm.

Table 1 indicates that the area of solutions obtained by Burns, RGP and SLOP are on average 9.6%, 3.7% and 7.5% higher, respectively, than the lower bound. This shows that the sub-optimality in the results obtained from these methods are low compared to the absolute best solution possible. Figure 3 shows sizing curves for the different optimization approaches as well as the lower bound computed using the proposed technique. The figure illustrates that the room for further improvement of results beyond smart deterministic approaches is low.

6. CONCLUSION

This paper proposes a lower bound computation method to evaluate statistical design optimization techniques. Using this bound, we evaluated several current design optimization methods. We found that worst-corner based deterministic approaches result in a pessimistic design with an average area 20% greater than the theoretical lower bound clearly motivating the need for statistically informed methods. Among these, we compared a smart deterministic approach (Burns) and a robust statistical optimization technique (RGP), as well as a method proposed in this paper call SLOP that uses sample level view of the process variation space. Results show that all statistically aware techniques have areas within 10% of the lower bound, on average. More statistically aware techniques (SLOP, RGP) do achieve lower areas; however the additional improvement is only 5.9% on average for RGP and are within 3.7% of the lower bound, with additional runtime cost of 41.5X compared to Burns. SLOP has higher area compared to RGP by 3.8% on average, however is faster than RGP by 5.6X. Overall, the lower bound shows that all statistical methods produce results that are provably close to the theoretical minimum and trade-off additional run time for approaching this minimum to within a couple percent.

7. REFERENCES

[1] C.Visweswariah, "*Death, taxes and failing chips.*", Design Automation Conference, 2003. pp. 343-347.

[2] H.Chang, S.S.Sapatnekar, "*Statistical Timing Analysis considering spatial correlation using a single PERT-like traversal.*", International Conference on Computer Aided Design 2003. pp. 621-625.

[3] C.Visweswariah et al, "*First-order incremental blcok-based statistical timing analysis.*", Design Automation Conference 2004, pp. 331-336.

[4] L.Scheffer., "*The Count of Monte Carlo.",* ACM/IEEE TAU Workshop, 2004.

[5] V.Veetil, D.Sylvester,D.Blaauw, "*Efficient Monte Carlo based incremental statistical timing analysis",* Design Automation Conference 2008, pp. 676-681.

[6] A.Singhee, R.A.Rutenbar*, "From Finance to Flip-Flops: A Study of Fast Quasi Monte Carlo Methods From Computational Finance Applied to Statistical Circuit Analysis.",* ISQED 2007, pp. 685-692.

[7] J.Jaffari, M.Anis*, "On efficient Monte Carlo-based statistical static timing analysis of digital circuits.",* International Conference on Computer Aided Design 2008, pp. 196-203.

[8] Y.Abulatia, A.Kornfeld*, "Estimation of FMAX and ISB in microprocessors.",* IEEE Transacations on VLSI Systems, Oct 2005, pp. 1205-1209.

[9] A.Agarwal, K.Chopra, D.Blaauw, V.Zolotov *, "Circuit optimization using statistical static timing analysis.*", Design Automation Conference 2005, pp. 321-324.

[10] S.Borkar, T.Karnik, S.Narendar, V.De *, "Parameter variations and impact on circuits and microarchitecture.",* Design Automation Conference 2003, pp. 338-342.

[11] K.A.Bowman, S.G.Duvall, J.D.Meindl*, "Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration.",* IEEE J. Solid-State Circuits, Nov 2002, pp. 183-190.

[12] S.M.Burns, M.Ketkar, N.Menezes, K.A.Bowman, J.W.Tschanz, V.De *, "Comparative Analysis of Conventional and Statistical Design Techniques.",* Design Automation Conference , 2007. pp. 238-243.

[13] J.P.Fishburn, A.E.Dunlop *, "TILOS: A posynomial programming approach to transistor sizing",* International Conference on Computer Aided Design, 1985.

[14] K.Chopra, S.Shah, A.Srivastava, D.Blaauw, D.Sylvester *, "Parametric yield maximization using gate sizing based on efficient statistical power and delay gradient computation.",* International Conference on Computer Aided Design, 2005. pp. 1023-1028.

[15] M.Mani, A.Devgan, M.Orshansky, "*An efficient algorithm for statistical minimization of total power under timing yield constraints.",* Design Automation Conference, 2005. pp. 309-314.

[16] J.Singh, L.Zhi-Quan, S.S.Sapatnekar *, "Robust Gate Sizing by Geometric Programming.",* Design Automation Conference, 2005. pp. 315-320.

[17] V. Sundararajan, S.Sapatnekar, K.K.Parhi *, "Fast and Exact Transistor Sizing Based on Iterative Relaxation.",* IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2002, Vol. 21, No. 5.

[18] S.Boyd, S.J.Kim, L.Vandenberghe, A.Hassibi *, "A tutorial on geometric programming.",* Springer Netherlands, 2007, Vol. 8, No. 1.

[19] http://www.mosek.com. [Online]

[20] L.Pang, B.Nikolic*, "Measurement and analysis of variability in 45nm strained-Si CMOS technology.",* Custom Integrated Circuits Conference, 2008. pp. 129-132.

[21] S.Reda, S.R.Nassif*, "Analyzing the impact of process variations on parametric measurements: Novel models and applications.*", Design, Automation and Test in Europe, 2009. pp. 375-380.

[22] M.Kanno et. al, "*Empirical Characteristics and Extraction of Overall Variation for 65-nm MOSFETs and Beyond",* IEEE Symposium on VLSI Technology, 2007. pp. 88-89.

[23] F.Brglez, H.Fujiwara *, "Neutral netlist of ten combinational benchmark circuits and a target translator in FORTRAN.",* Proc. ISCAS, 1985. pp. 695-698.

[24] S.H.Kulkarni, D.Sylvester, D.Blaauw, "*A Statistical Framework for Post-Silicon Tuning through Body Bias Clustering", ICCAD 2006.*