Statistical Gate Sizing for Timing Yield Optimization

Debjit Sinha
ECE, Northwestern University
Evanston, IL 60208

Narendra V. Shenoy
ATG, Synopsys Inc.
Mountain View, CA 94043

Hai Zhou
ECE, Northwestern University
Evanston, IL 60208

Abstract— **Variability in the chip design process has been relatively increasing with technology scaling to smaller dimensions. Using worst case analysis for circuit optimization severely over-constrains the system and results in solutions with excessive penalties. Statistical timing analysis and optimization have consequently emerged as a refinement of the traditional static timing approach for circuit design optimization.**

In this paper, we propose a statistical gate sizing methodology for timing yield improvement. We build statistical models for gate delays from library characterizations at multiple process corners and operating conditions. Statistical timing analysis is performed, which drives gate sizing for timing yield optimization. Experimental results are reported for the ISCAS and MCNC benchmarks. In addition, we provide insight into statistical properties of gate delays for a given technology library which intuitively explains when and why statistical optimization improves over static timing optimization.

I. Introduction

Increasing significance of variability in the IC design process with shrinking feature sizes make timing verification and optimization extremely difficult. Uncertainties are attributed to the manufacturing process, environmental factors like V_{dd} and temperature, and device fatigue phenomenon. We term all factors that contribute to delay variations as parameters and refer to the range in which they can collectively vary as the parameter space. Nominally sub-critical paths may become critical in some regions due to sensitivity to the sources of variation. This can cause circuits to fail in meeting timing constraints. Consequently, it becomes imperative to improve the timing yield of a circuit, which denotes the probability that the circuit will meet its timing constraints under variations. This motivates development of robust design methodologies for circuit optimization.

Statistical timing analysis and optimization have emerged as a refinement of the existing methodology of static timing optimization. Sources of variability are considered as random variables. Gate delays are modeled as functions of these variables with a known distribution (typically a Gaussian). This allows for an analytical evaluation of the circuit delay as a probability density function (PDF). From the PDF, the probability that the circuit delay exceeds a required value is computed as the area under the PDF to the right of the required value. For sake of simplicity, we assume that circuit failure is from late arriving signals only (setup condition). The extension to include failures due to early arriving signals (hold condition) is similar. Statistical optimization provides an additional degree of freedom in the solution space where the optimizer tunes the PDFs of circuit delays instead of worst case values. This makes statistical timing optimization an attractive design approach.

Analytical approaches to statistical static timing analysis [1]–[4] and optimization [5]–[7] have emerged as an active research topic. Agarwal *et al.* present a statistical timing analysis approach which focuses on handling spatial correlations [1]. Recent literature considers gate delays as Gaussian random variables, since it facilitates fast analytical evaluation. Chang *et al.* propose a statistical timing analysis approach under this assumption which considers spatial correlations [2]. A first order incremental block based statistical timing analyzer is presented by Visweswariah *et al.* in [3]. We build a SSTA engine based on this work. In our experience, node criticalities introduced in [3] can only be estimated in closed form to be within 20% of the true values obtained from Monte Carlo simulations. We find these to be inadequate for guiding optimization. A statistical gate sizing methodology for timing yield improvement is proposed by Raj *et al.* in [5]. They use the notion of path disutilities for optimization. Another gate sizing method using a statistical delay model is proposed by Jacobs *et al.* in [6]. They ignore correlations among the gate delays and use an analytical gate delay model based on the cell speed factor. In addition, their method attains to optimize only the mean of the output arrival time PDF. Using a simple gate delay model, Agarwal *et al.* propose a coordinate descent based gate sizing algorithm for statistical timing optimization [7]. Their sensitivity metric for any gate is a function of a given percentile point of its delay distribution and size. Intra-die variability is considered, and the resultant gate delay variation is assumed to be a fixed value (10% of the nominal).

In this paper, we present a statistical gate sizing methodology for timing yield optimization. We build statistical models for gates based on industrial data from a technology library characterized at different points in the parameter space. We perform statistical timing analysis which accounts for correlations and also incorporates random gate delay components. This serves as a backbone for our gate sizing algorithm, which attains to maximize the timing yield of a circuit and not merely improve values for a single given percentile point in the circuit's delay distribution. We focus on combinational circuits in this paper. Experimental results on the ISCAS and MCNC benchmarks demonstrate significant improvements in timing yield. Static timing optimization is performed using an industrial synthesis tool and is used for benchmarking our statistical timing optimizer. We also present insight into statistical properties of gate delays from a real technology library which intuitively explains when and why statistical timing optimization gains over static timing optimization.

II. Statistical Modeling and Timing Analysis

A. Statistical Modeling

Statistical gate modeling involves expressing a gate delay as a function of the parameters of variation, which are modeled as random variables. Based on the work in [3], we assume that gate delays are approximated by a linear function of the parameters. In addition, we assume that these parameters are independent, since a dependent set of parameters are transformed into an equivalent set of independent parameters using PCA [2].

Each gate from a technology library is characterized at different process corners which span the entire parameter space. A least-squares fit is then employed to express the gate delay as a linear function of these parameters. The parameters are finally modeled as random variables having a Gaussian distribution. This makes gate

delays a weighted sum of Gaussians, and are expressed as

$$a_0 + \sum_{i=1}^{n} a_i \Delta X_i + a_{n+1} \Delta R_a \qquad (1)$$

where, a_0 is the mean or nominal value of the delay, ΔX_i ($\forall i = 1, 2, \ldots, n$) represents the variation of n global parameters (X_i, $\forall i = 1, 2, \ldots, n$) from their nominal values, and a_i ($\forall i = 1, 2, \ldots, n$) are the sensitivities to their corresponding sources of variation. ΔR_a denotes the variation from the nominal of an independent random variable R_a associated with each gate. a_{n+1} represents the sensitivity of the gate delay to R_a. Statistical gate modeling is performed for all gates in the circuit before statistical timing analysis and can be done incrementally.

B. Statistical Timing Analysis

Statistical timing analysis involves propagating delay distributions through the circuit. Since we express gate delays as a first order sum of random variables, the *sum* operation is performed easily. The *max* operation, on the other hand, is an intricate operation. Under the assumption that gate delays are linear functions of Gaussian random variables, we use the method by Clark [8] to obtain the moments of the *max* of two Gaussian distributions. The distribution of the *max* of two Gaussians is then approximated to a Gaussian distribution by matching the first and second order moments as in [2], [3], [8]. We use (a_0, σ_A) and (b_0, σ_B) to represent the (mean, standard deviation) of two timing quantities A and B respectively. a_i and b_i for $i = \{1, 2, \ldots n + 1\}$ represent the sensitivity of these timing quantities to the given parameters of variation as described earlier. ρ is used to represent their correlation coefficient. We define

$$\phi(x) \quad \triangleq \quad \frac{1}{\sqrt{2\pi}} exp(-\frac{x^2}{2}) \qquad (2)$$

$$\Phi(y) \quad \triangleq \quad \int_{-\infty}^{y} \phi(x) dx \qquad (3)$$

$$\theta \quad \triangleq \quad (\sigma_A^2 + \sigma_B^2 - 2\rho\sigma_A\sigma_B)^{1/2} \qquad (4)$$

$$\alpha \quad \triangleq \quad \frac{a_0 - b_0}{\theta} \qquad (5)$$

The mean μ and variance var of $max(A, B)$ can be expressed analytically (from [8]) as

$$\mu \quad = \quad a_0\Phi(\alpha) + b_0\Phi(-\alpha) + \theta\phi(\alpha) \qquad (6)$$

$$var \quad = \quad (\sigma_A^2 + a_0^2)\Phi(\alpha) + (\sigma_B^2 + b_0^2)\Phi(-\alpha) + \qquad (7)$$
$$(a_0 + b_0)\theta\phi(\alpha) - \mu^2$$

We wish to express the $max(A, B)$ as a Gaussian distribution in the canonical form

$$C \quad = \quad c_0 + \sum_{i=1}^{n} c_i \Delta X_i + c_{n+1} \Delta R_c, \quad \text{w}here \qquad (8)$$

$$c_0 \quad = \quad \mu \qquad (9)$$

$$c_i \quad = \quad a_i\Phi(\alpha) + b_i\Phi(-\alpha) \quad \forall i \in 1, 2, \ldots, n \qquad (10)$$

$$c_{n+1} \quad = \quad (var - \sum_{i=1}^{n} c_i^2)^{1/2} \qquad (11)$$

We prove in Appendix I that $(var - \sum_{i=1}^{n} c_i^2)$ is always non-negative. The required time estimation during timing analysis is performed by a backward propagation and involves the *subtract* and *min* operations. When a gate has more than two fanins (fanouts), the *max* (*min*) operation for the arrival (required) time PDF calculation is done one pair a time. Care is taken to determine an ordering on these operations

to minimize the loss in accuracy due to approximations. For example, multiple identical Gaussians in a set on which a *max* (*min*) operation is to be performed, are always reduced to a single Gaussian. Statistical timing analysis can be performed in an incremental manner like a standard incremental static timer. It can incorporate separate PDFs for the rise and fall delays. Slack estimation during timing analysis involves *subtract* operations which can be performed on the canonical forms of the timing distributions. A *min* operation on the slack distributions at the primary outputs gives the circuit slack, the PDF of which is evaluated to obtain the probability of the circuit meeting its timing requirements.

III. STATISTICAL GATE SIZING

Gate sizing has been an active and a proven timing optimization methodology over the decades. We consider extending static gate sizing that focuses on improving the slack of a circuit to statistical gate sizing that considers improving the probability that the circuit slack is non-negative. Given the circuit slack (after statistical timing analysis) in the form of a Gaussian distribution with mean μ_S and standard deviation σ_S, the timing yield of the circuit denotes the probability that the slack is non-negative. Formally, timing yield is defined as the following.

$$P \quad \triangleq \quad \frac{1}{\sqrt{2\pi}\sigma_S} \int_{0}^{\infty} exp[-\frac{(x - \mu_S)^2}{2\sigma_S^2}]dx \qquad (12)$$

We attain to maximize P, which we next show is equivalent to maximizing $\frac{\mu_S}{\sigma_S}$.

Theorem 1:

$$max \; \frac{1}{\sqrt{2\pi}\sigma_S} \int_{0}^{\infty} exp[-\frac{(x - \mu_S)^2}{2\sigma_S^2}]dx \quad \equiv \quad max \; \frac{\mu_S}{\sigma_S}$$

Proof: We define $y \triangleq (\mu_S - x)/\sigma_S$. Under variable transformation,

$$\frac{1}{\sqrt{2\pi}\sigma_S} \int_{0}^{\infty} exp[-\frac{(x - \mu_S)^2}{2\sigma_S^2}]dx$$
$$= \quad \frac{1}{\sqrt{2\pi}} - \int_{\mu_S/\sigma_S}^{-\infty} exp[-\frac{y^2}{2}]dy = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\mu_S/\sigma_S} exp[-\frac{y^2}{2}]dy,$$

which is strictly increasing with $\frac{\mu_S}{\sigma_S}$. This proves our claim. ∎

Our statistical gate sizing approach thus attains to maximize $\frac{\mu_S}{\sigma_S}$. An alternative objective function that intuitively optimizes timing yield is expressed in the form of maximizing $\mu_S - 3\sigma_S$. This objective function focuses on pushing the slack distribution to the right, that is, attains to improve the worst case slack.

We present a Statistical Global Gate Sizing (SGGS) algorithm for timing yield optimization. The global sizing algorithm is a mix of a multi-dimensional descent based optimization, a perturbed propagation based heuristic that avoids gradient re-computations, and a global perturbation technique to get out of local minimums. Our choice of the global sizing algorithm is motivated by results obtained by Coudert *et al.* in [9], which show that this algorithm is the best for sizing optimizations in terms of performance and power/delay curves.

The proposed algorithm considers the circuit as a network N and a global cost function $Cost$ that it attains to maximize. The pseudo code of the SGGS algorithm is given in Fig.1. A set $update$ maintains a list of nodes whose costs (gradients) are to be computed. This set is initialized with all nodes in N. Another set $moves$ maintains a list of nodes that can potentially be resized, that is, can be mapped to a different gate from the given technology library. This set is kept empty initially. For every node n in $update$, a local network

- Algorithm: **SGGS**($N, Cost$)
- **Output:** Optimized circuit with improved timing yield
- begin
  1) $update = N$;
  2) $moves = \emptyset$;
  3) do
  4) $old\_cost = Cost(N)$;
  5)   foreach $n \in update$
  6)   Extract local network N' around n;
  7)   Find best-gate g for n wrt $Cost(N')$;
  8)   if $g \neq gate(n)$
  9) $n.move = g$;
  10) $moves = moves \cup \{n\}$;
  11) $moved = MultiMove(N, Cost, moves)$;
  12) $update = PerturbedNodes(moved)$;
  13) until $(Converge(old\_cost, Cost(N), moved))$
- end

Fig. 1.   Statistical global gate sizing algorithm

Circuit	AT μ (ns)		AT σ (ns)		Run Times (s)	
	SSTA	MC	SSTA	MC	SSTA	MC
C432	2.194	2.197	0.165	0.172	0.1	6.5
C499	1.316	1.311	0.095	0.095	1.2	14.6
C880	1.973	1.968	0.143	0.143	0.4	14.0
C1355	1.829	1.831	0.141	0.139	0.8	20.4
C1908	2.208	2.214	0.161	0.160	0.7	14.3
C2670	1.950	1.957	0.177	0.174	1.5	24.7
C3540	3.242	3.234	0.261	0.260	0.8	37.7
C5315	3.029	3.024	0.246	0.242	7.3	63.0
C6288	9.996	9.968	0.779	0.789	0.7	85.1
C7552	3.313	3.305	0.261	0.254	5.1	71.9
cm85a	0.425	0.423	0.029	0.029	0.0	1.6
sct	0.485	0.484	0.030	0.030	0.1	2.8
alu2	2.584	2.590	0.211	0.213	0.2	12.9
too_large	1.048	1.047	0.071	0.073	0.1	13.6
frg2	1.486	1.490	0.097	0.098	1.3	29.3

around it is extracted and statistical timing analysis is performed on the extracted network. For each alternative gate in the given library that can be mapped to the same node, statistical delay modeling is carried out. A run of statistical timing analysis determines if the new gate improves the local timing yield, that is, the timing yield at the output of the local network. The steps of incremental statistical modeling and timing analysis are performed during the evaluation of the $Cost$ function. If alternate gates are found that improve the timing yield, the node-gate pairs are stored in the $moves$ set. Next the *MultiMove* routine picks a sub-set of gates from the set $moves$ that provides the maximum cumulative improvement in the original circuit's timing yield and updates those nodes with the new gates. This routine determines the sub-set for the move based on the descent direction or by a conjugation of directions of the node costs [9]. The new set of nodes whose costs need to be recomputed are now derived based on the nodes that are resized. This process is repeated till convergence, wherein future iterations do not improve the global timing yield further.

IV. IMPLEMENTATION AND EXPERIMENTAL RESULTS

The statistical modeling, timing analysis and optimization methodologies presented in the previous sections are implemented in an industrial tools environment. Experiments are performed on combinational ISCAS and MCNC benchmarks mapped to a 0.13μ technology library from a foundry.

Statistical modeling involves obtaining delay equations for gates in the circuit as a function of parameters under consideration. For our experiments, we choose V_{dd} and temperature as parameters. This is because our industrial library characterizations are available only for various corners in the parameter space of V_{dd} and temperature. However, our approach is not limited to the use of any particular parameters of variation and can handle variations from other parameters like L_e and V_T similarly. Gates are characterized at different temperature and V_{dd} values. The sources of variation are then normalized by subtracting their corresponding typical values and dividing by their standard-deviations respectively. For example, if gates are characterized for V_{dd} values ranging from $1.08V$ to $1.32V$, with typical value set to $1.2V$, the standard deviation is set as

$$3\sigma_V = 1.32V - 1.20V$$

Similarly gates are characterized for temperature variations from 0^oC to 125^oC. For any characterization point X the delay equation is set up as

$$D_X = D_0 + D_1\frac{T_X - T_0}{\sigma_T} + D_2\frac{V_X - V_0}{\sigma_V}$$

where T_0, V_0 represent typical or mean values of the temperature and V_{dd} respectively. D_0 represents the typical delay obtained by characterizing the gate at T_0 and V_0. This formulation is scalable to any number of parameters. A least squares fit procedure is employed to obtain the coefficients D_is. The accuracy of this approach is dependent on the number of characterization points. Our modeling is constrained to the available pre-characterizations of different gates in the given technology library. This procedure is followed to model all rise and fall gate delay arcs in the circuit.

After statistical modeling is performed, arrival times and required times are set at the primary inputs and primary outputs of the circuit respectively. Statistical timing analysis is performed to obtain the arrival and required time PDFs for all nodes in the circuit. A *min* operation is performed on the obtained slack PDFs at the primary outputs to determine the global circuit slack PDF S, with mean μ_S and variance σ_S^2. Timing yield of the circuit is obtained from (12).

Table I presents statistical timing analysis results (arrival time mean and sigma values) for the ISCAS and MCNC benchmarks with comparisons to those obtained from Monte Carlo simulations. Figures reported are for 10000 random samples of Monte Carlo simulations, which give a standard error of less than 1%. Benchmark sizes ranged from about 100 to 2000 gates. The results demonstrate the accuracy and run-time efficiency of SSTA.

The proposed statistical global gate sizing algorithm is implemented for timing yield optimization. The local network extraction around a node involves creation of a network that includes the node and its fanin nodes as internal nodes. Fanins and fanouts of these internal nodes are set as primary inputs and primary outputs of the local network respectively. The arrival and required time PDFs on the primary inputs and outputs are set to the values as in the original circuit. Alternative gates from the same class are mapped on the node and the timing yield of the local network is used as the metric for obtaining the best gate within the given constraints. Once the alternate node and best-gate pairs are evaluated, the timing yield improvement on slack S of the original circuit is used as a metric to determine the nodes that are resized. We attain to maximize the metric μ_S/σ_S in our approach since it is equivalent to maximizing the timing yield. The process is repeated until convergence. The complexity of this
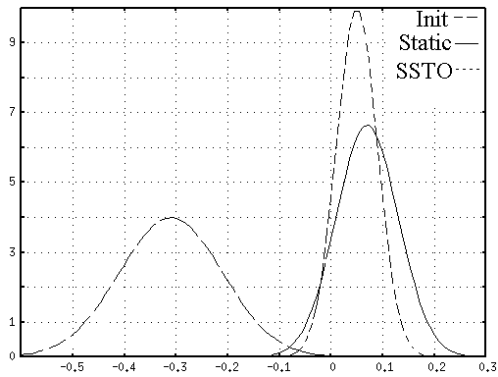
Fig. 2. Pre and post optimization slack PDFs for a test circuit

TABLE II
STATISTICAL TIMING YIELD OPTIMIZATION RESULTS

Circuit	Statistical Timing Yield Improvement	
	Optimizing $(\mu_S - 3\sigma_S)$	Optimizing (μ_S/σ_S)
C432	0.0105	0.0122
C499	0.3032	0.3158
C880	0.0037	0.0037
C1355	0.4939	0.4963
C1908	0.1319	0.1584
C2670	0.1730	0.2153
C3540	0.4949	0.4977
C5315	0.4923	0.4925
C7552	0.4998	0.4995
cm85a	0.0569	0.2037
sct	0.1821	0.4342
alu2	-0.0570	0.1240
too_large	0.4269	0.4580
frg2	0.4516	0.3774

algorithm using best-fit polynomial is $N^{1.2}$, where N denotes the number of internal nodes [9]. Experiments are also performed with $\mu_S - 3\sigma_S$ as the metric for optimization. Results for both metrics are presented. Fig.2 shows the circuit slack PDFs for a test circuit. The three PDFs denote the slack distributions for the unoptimized circuit *(Init)*, static timing optimized circuit *(Static)* and statistical timing optimized circuit *(SSTO)*. The reduced variance of the *SSTO* slack PDF improves the timing yield (from 0.87 to 0.89) even though it has a smaller mean as compared to *Static* slack PDF.

Monte Carlo simulations are performed to determine the accuracy of the developed statistical timer. It is observed that the loss in accuracy due to approximations in the *max* and *min* operations do not affect the output PDF's mean and variance significantly. Additional experiments are performed on inverter chains, results from which match identically to those from Monte Carlo simulations. Results from static gate sizing using an industrial synthesis tool are used as references to evaluate improvements due to statistical gate sizing on the timing yield. Circuits are initially optimized with the static timing optimizer. Statistical timing analysis is then performed. We set the required time of the circuit to be the mean of the arrival time. The circuit slack is therefore a distribution with 0 mean. The timing yield of the circuit after static timing optimization is consequently 0.5. Statistical timing optimization is then performed on the circuit under area constraints and a new slack PDF is obtained post optimization. We now obtain the new timing yield after statistical optimization, and the value by which it exceeds the timing yield after static optimization is called the timing yield improvement. Table II shows timing yield improvement values for both $\mu_S - 3\sigma_S$

and μ_S/σ_S as optimization objective functions. We observe that the proposed μ_S/σ_S metric guides better optimization. This is because the other objective function attempts to maximize the worst case slack, instead of the timing yield. The proposed gate sizing algorithm took between 1 to < 480 mins for all benchmarks on a 400 Mhz Sun Ultra 4 machine with 4 GB RAM. Results demonstrate timing yield improvements by up to 0.49 and on an average by 0.3.

V. ANALYSIS OF STATISTICAL PROPERTIES OF GATE DELAYS

Static gate sizing picks faster gates to map on nodes to meet the timing constraints of a circuit. It does not consider the statistical properties of gate delays. Consider the output arrival time PDF of a given gate in a circuit. If a smaller mean of the arrival time implied smaller variance in the arrival time distribution monotonically, static gate sizing would theoretically leave no room for statistical optimization. For any gate, the arrival time at its output can be considered to be a function of its size. It is known that the arrival time decreases with increase in size initially as the larger gate has better driving capability. However, due to increasing effective loading capacitance as seen by its driving gates, the arrival time starts increasing with further increase in size. It is also known that larger gates have lesser variability. A larger gate might yield a greater mean arrival time than one the static timing optimizer picks, but lesser variability could eventually help improve the overall timing yield.

We perform analysis of statistical properties of gate delays on some gate classes from a 0.13μ technology library. We present sigma versus mean plots of the arrival time PDFs for various gates in a class. Fig.3 presents plots for a class of inverters. Dots on the plot represent gates which are sorted on their mean output arrival times when mapped to a given node and not in any order of their sizes. A node can be mapped by any of these gates and the optimizer selects the best gate for mapping. Though most gates of a class make the plots monotonic, there exist exceptions. While the static timing optimizer chooses a gate with a smaller mean arrival time ignoring the fact that it may have larger variability, the statistical timing optimizer is found to make a smarter choice by picking gates which help increase the overall timing yield. These gates give statistical timing optimization an edge over static timing optimization. Similar plots are observed for other gate-classes in the library.
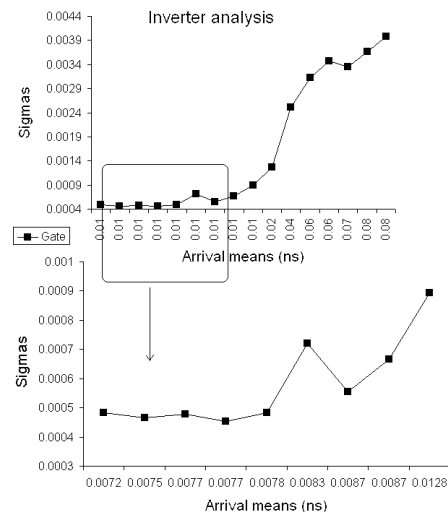

Fig. 3. Arrival means and sigmas for a class of inverters

VI. CONCLUSIONS

We present a statistical gate sizing approach for timing yield optimization in this paper. Experimental results show improvements in timing yield by up to 0.49 and on an average by 0.3. In addition, we provide insight into statistical properties of gate delays from a given technology library which intuitively explains when and why statistical optimization improves over traditional timing optimization.

ACKNOWLEDGMENTS

APPENDIX I

We prove that the variance matching method in (11) never involves computing the root of a negative quantity. Using the notations defined earlier, we show ($\xi \triangleq var - \sum_{i=1}^{n} c_i^2) \geq 0$.

$$\mu = a_0 \Phi(\alpha) + b_0 \Phi(-\alpha) + \theta\phi(\alpha)$$

$$var = (\sigma_A^2 + a_0^2)\Phi(\alpha) + (\sigma_B^2 + b_0^2)\Phi(-\alpha) + (a_0 + b_0)\theta\phi(\alpha) - \mu^2$$

$$\sum_{i=1}^{n} c_i^2 = \sum_{i=1}^{n} \left[a_i \Phi(\alpha) + b_i \Phi(-\alpha) \right]^2$$

$$\begin{aligned}
\xi &= var - \sum_{i=1}^{n} c_i^2 \\
&= (\sigma_A^2 + a_0^2)\Phi(\alpha) + (\sigma_B^2 + b_0^2)\Phi(-\alpha) + (a_0 + b_0)\theta\phi(\alpha) \\
&\quad - a_0^2 \Phi(\alpha)^2 - b_0^2 \Phi(-\alpha)^2 - \theta^2 \phi(\alpha)^2 - 2a_0 \Phi(\alpha)\theta\phi(\alpha) \\
&\quad - 2b_0 \Phi(-\alpha)\theta\phi(\alpha) - 2a_0 b_0 \Phi(\alpha)\Phi(-\alpha) \\
&\quad - \Phi(\alpha)^2 \sum_{i=1}^{n} a_i^2 - \Phi(-\alpha)^2 \sum_{i=1}^{n} b_i^2 - 2\Phi(\alpha)\Phi(-\alpha) \sum_{i=1}^{n} a_i b_i \\
&= \sigma_A^2 \Phi(\alpha) + a_0^2 \Phi(\alpha) + \sigma_B^2 \Phi(-\alpha) + b_0^2 \Phi(-\alpha) \\
&\quad + (a_0 + b_0)\theta\phi(\alpha) - a_0^2 \Phi(\alpha)^2 - b_0^2 \Phi(-\alpha)^2 \\
&\quad - 2a_0 \Phi(\alpha)\theta\phi(\alpha) - 2b_0 \Phi(-\alpha)\theta\phi(\alpha) - \theta^2 \phi(\alpha)^2 \\
&\quad - 2a_0 b_0 \Phi(\alpha)\Phi(-\alpha) - \sigma_A^2 \Phi(\alpha)^2 + a_{n+1}^2 \Phi(\alpha)^2 \\
&\quad - \sigma_B^2 \Phi(-\alpha)^2 + b_{n+1}^2 \Phi(-\alpha)^2 \\
&\quad - 2\Phi(\alpha)\Phi(-\alpha) \sum_{i=1}^{n} a_i b_i \\
&= (\sigma_A^2 + \sigma_B^2 + a_0^2 + b_0^2)\Phi(\alpha)\Phi(-\alpha) + (a_0 + b_0)\theta\phi(\alpha) \\
&\quad - \theta^2 \phi(\alpha)^2 - 2a_0 \Phi(\alpha)\theta\phi(\alpha) - 2b_0 \Phi(-\alpha)\theta\phi(\alpha) \\
&\quad - 2a_0 b_0 \Phi(\alpha)\Phi(-\alpha) + a_{n+1}^2 \Phi(\alpha)^2 + b_{n+1}^2 \Phi(-\alpha)^2 \\
&\quad - 2\Phi(\alpha)\Phi(-\alpha) \sum_{i=1}^{n} a_i b_i \\
&= \Phi(\alpha)\Phi(-\alpha)\left[(\sigma_A^2 + \sigma_B^2 - 2\sum_{i=1}^{n} a_i b_i) + (a_0 - b_0)^2 \right] \\
&\quad + \theta\phi(\alpha)\left[a_0(1 - 2\Phi(\alpha)) + b_0(1 - 2 + 2\Phi(\alpha)) \right] \\
&\quad - \theta^2 \phi(\alpha)^2 + a_{n+1}^2 \Phi(\alpha)^2 + b_{n+1}^2 \Phi(-\alpha)^2
\end{aligned}$$

To show $\xi \geq 0$, it is sufficient to show that

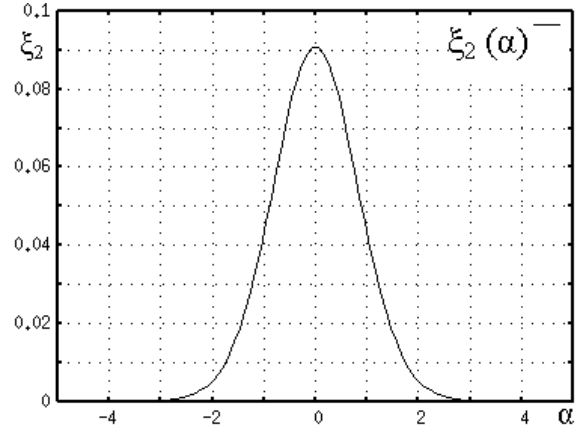$$\xi_1 \triangleq \xi - a_{n+1}^2 \Phi(\alpha)^2 - b_{n+1}^2 \Phi(-\alpha)^2 \geq 0$$

Fig. 4. Plot of ξ_2 against α

$$\begin{aligned}
\xi_1 &= \Phi(\alpha)\Phi(-\alpha)\left[(\sigma_A^2 + \sigma_B^2 - 2\sum_{i=1}^{n} a_i b_i) + (a_0 - b_0)^2 \right] \\
&\quad + \theta\phi(\alpha)\left[a_0(1 - 2\Phi(\alpha)) + b_0(-1 + 2\Phi(\alpha)) \right] - \theta^2 \phi(\alpha)^2 \\
&= \Phi(\alpha)\Phi(-\alpha)\left[\theta^2 + (a_0 - b_0)^2 \right] - \theta^2 \phi(\alpha)^2 \\
&\quad + \theta(a_0 - b_0)(1 - 2\Phi(\alpha))\phi(\alpha) \\
&= \theta^2 \left[\Phi(\alpha)\Phi(-\alpha) - \phi(\alpha)^2 \right] + \theta(a_0 - b_0)(1 - 2\Phi(\alpha))\phi(\alpha) \\
&\quad + \Phi(\alpha)\Phi(-\alpha)(a_0 - b_0)^2
\end{aligned}$$

If $\theta = 0$, $\xi_1 = \Phi(\alpha)\Phi(-\alpha)(a_0 - b_0)^2 \geq 0$. For positive θ (since $\theta \geq 0$), it is sufficient to show

$$\xi_2 \triangleq \xi_1/\theta^2 \geq 0$$

$$\begin{aligned}
\xi_2 &= \Phi(\alpha)\Phi(-\alpha) - \phi(\alpha)^2 + \frac{a_0 - b_0}{\theta}(1 - 2\Phi(\alpha))\phi(\alpha) \\
&\quad + \Phi(\alpha)\Phi(-\alpha)\frac{(a_0 - b_0)^2}{\theta^2} \\
&= \Phi(\alpha)\Phi(-\alpha) - \phi(\alpha)^2 + \alpha(1 - 2\Phi(\alpha))\phi(\alpha) + \Phi(\alpha)\Phi(-\alpha)\alpha^2
\end{aligned}$$

$\xi_2(\alpha)$ is symmetric and is found to be non-negative for all real values of α. For values of $|\alpha| \geq 3$, ξ_2 approaches 0 with both $\phi(\alpha)$ and $\Phi(\alpha)$ tending to 0. Fig. 4 shows the plot of ξ_2 as a function of α.

REFERENCES

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

[2] H. Chang and S. S. Sapatnekar, "Statistical timing analysis considering spatial correlations using a single PERT-like traversal," in *ICCAD*, 2003, pp. 621–625.

[3] C. Visweswariah, K. Ravindran, K. Kalafala, S. G. Walker, and S. Narayan, "First-order incremental block-based statistical timing analysis," in *DAC*, 2004, pp. 331–336.

[4] D. Sinha and H. Zhou, "A unified framework for statistical timing analysis with coupling and multiple input switching," in *ICCAD*, 2005.

[5] S. Raj, S. B. K. Vrudhula, and J. Wang, "A methodology to improve timing yield in the presence of process variations," in *DAC*, 2004, pp. 448–453.

[6] E. T. A. F. Jacobs and M. R. C. M. Berkelaar, "Gate sizing using a statistical delay model," in *DATE*, 2000.

[7] A. Agarwal, K. Chopra, D. Blaauw, and V. Zolotov, "Statistical timing based optimization using gate sizing," in *ACM Intl. Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, 2005.

[8] C. E. Clark, "The greatest of a finite set of random variables," in *Operations Research, Vol. 9, No. 2 (Mar - Apr)*, 1961, pp. 145–162.

[9] O. Coudert, R. Haddad, and S. Manne, "New algorithms for gate sizing: A comparative study," in *DAC*, 1996, pp. 734–739.