Clock Tree Synthesis with Data-Path Sensitivity Matching

Matthew R. Guthaus
Department of CE
UC Santa Cruz
Santa Cruz, CA 95064
mrg@soe.ucsc.edu

Dennis Sylvester
Department of EECS
University of Michigan
Ann Arbor, MI 48109
dennis@eecs.umich.edu

Richard B. Brown
Department of ECE
University of Utah
Salt Lake City, UT 84112
brown@coe.utah.edu

Abstract— This paper investigates methods for minimizing the impact of process variation on clock skew using buffer and wire sizing. While most papers on clock trees ignore data-path circuit variations and most papers on data-path circuit optimization disregard clock tree variation, we consider both. Using both clock and data-path variations together, we present a novel sensitivity-matching algorithm that allows clock tree skews to be intentionally correlated with data-path sensitivities to ameliorate timing violations due to variation. Our statistical tuning shows an improvement in terms of expected clock skew and clock skew variation over previously published robust algorithms.

I. Introduction

Clock distribution significantly affects the performance of all synchronous integrated circuits. A poor clock distribution network can result in limited speed due to setup timing violations, high power consumption due to excessive clock loads, or non-functional circuits due to hold timing violations. During the past several decades, many researchers have examined clock synthesis including clock routing [4, 6, 14], wire sizing [14, 17], and buffering [3, 14, 18, 20]. Several previous works have examined the impact of process variation on clock skew scheduling [13, 16] without the aid of statistical static timing analysis (SSTA). Recently, however, SSTA has become of major interest due to increasing variation in nanometer geometries and the development of several efficient analysis methods with multiple inter- and intra-die sources of variation [1, 19]. Other researchers have done studies of clock tree parameters and their impact on variation [10] and worst-case optimization under process variations [15].

In this paper, the initial clock tree topologies are selected, routed [4], and buffered [18] according to prior methods. We aim to improve them through wire and buffer sizing only. The objective is to minimize the clock skew and its variation given a power budget. This is useful in the final stages of a design when the clock loads have changed, useful skew is being leveraged, or more accurate timing information is available. The methods that are proposed are independent of the buffer and interconnect models and are general enough to consider the effects of slew rates on buffer delay and resistive shielding in interconnect delay.

We optimize the local skew rather than the global skew of the clock tree. The local skew between any two sinks i and j is

$$\text{Skew}_{i,j} = |d_i + b_{i,j} - d_j| \tag{1}$$

where d_i (d_j) is the arrival time of sink i (j) and $b_{i,j}$ is the desired skew between the two sinks. The maximum skew of a clock is defined as the maximum local skew between any logically connected pair of sink arrival times excluding the intentional useful skew, $b_{i,j}$. In general, each sink delay, d_i or d_j, is a nonlinear function of the buffer and wire sizes. Therefore, clock skew optimization is a non-linear problem.

The techniques introduced in this paper rely on the concept of statistical correlation between the clock tree and data-path to improve robustness. Many existing SSTA tools provide first-order parametric sensitivities to the process parameters. Much like a designer would manually do, we consider these sensitivities in deciding how the clock tree buffers and wires should be sized. Nominal performance is the primary objective, but the parametric sensitivities are minimized and matched as a secondary objective. As the nominal skew of the clock tree nears zero, the magnitude of the sensitivity differences is on the same order as the nominal skew and, therefore, increased emphasis is placed on improving them.

The remainder of this paper is organized as follows. In Section II, a brief review of correlation is presented. In Section III, the basis of our formulation of sensitivity matching is introduced. In Sections IV, V and VI, we further detail the method of sequential quadratic programming for data-path sensitivity matching. Finally, in Section VII, results are presented and, in Section VIII, conclusions are made.

II. Brief Review of Correlation

The correlation between two random variables describes how well the events track each other. The covariance provides the measure of the strength of the correlation between the two random variables,

$$\text{cov}(X,Y) = E[(X - E[X])(Y - E[Y])] \tag{2}$$

where $E[\cdot]$ is the expected value. The correlation coefficient between two variates, X and Y, is then defined as

$$\rho_{X,Y} = \text{cor}(X,Y) = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y}. \tag{3}$$

where σ_X and σ_Y are the standard deviations of the events. For uncorrelated variables, the covariance is zero resulting in a correlation coefficient of zero. The correlation coefficient for negatively correlated random variables is -1.0. Often, $\rho > 0.8$ or $\rho < -0.8$ are considered "strong" while correlations between these values, $-0.8 \leq \rho \leq 0.8$, are considered "weak."

978-1-4244-1922-7/08/$25.00 ©2008 IEEE

Correlation can also be interpreted geometrically as the cosine of the angle between two vectors of random event samples,

$$\rho_{X,Y} = \cos(\theta). \tag{4}$$

As a simple example, consider two ordered sample sets each with three events: $X = (6, 1, -7)$ and $Y = (4, 2, -6)$. Please note that the method must work on *centered* data; if the data is not centered, it should be shifted by the mean. The angle between any two vectors is given by

$$\theta = \cos^{-1}\left(\frac{X \cdot Y}{|X||Y|}\right) \tag{5}$$

where $X \cdot Y$ is the dot product and $|X|$ and $|Y|$ are the vector norms. The norm of each example vector is $|X| = \sqrt{87}$ and $|Y| = \sqrt{56}$ and their dot product is $X \cdot Y = 68$. Therefore, the cosine of the angle is $\frac{68}{\sqrt{87}\sqrt{56}} = 0.9742$ which means the angle is approximately $13.0°$. Orthogonal vectors ($90°$) mean the data is independent, $\rho = 0$, while parallel vectors mean positive ($0°$, $\rho = 1.0$) or negative ($180°$, $\rho = -1.0$) correlation.

III. Sensitivity Matching

This section introduces the key contribution of this paper. Suppose that we have two multivariate quantities, such as clock sink delays, in a first-order parameterized form. The first-order, parameterized delay model is a sum of a nominal component, k correlated Gaussian sources of variation, and an independent Gaussian component of variation. The geometric interpretation of correlation in Section II is very helpful in understanding the correlation of multivariate values. Each set of sensitivities in a parameterized form can be interpreted as a vector. For example,

$$A = a_0 + \sum_{i=1}^{k} a_i X_i = a_0 + R^T X$$

where R is a vector of the sensitivities of the parameterized form in k-space when there are k sources of variation and X is a vector of the k random variables. Given the definition of the first-order form, the data is already centered about each random variable as required. The correlation of two such vector quantities, R_i and R_j, is simply the cosine of the angle between the two vectors,

$$\rho_{i,j} = \cos(\theta) = \frac{R_i \cdot R_j}{|R_i||R_j|}. \tag{6}$$

If we include an independent term, X_r, this must be treated as a completely separate dimension for each parameterized form when we interpret the vector. As an example, consider the two forms,

$$A = a_0 + (a_1, ..., a_k, a_r, 0)^T X$$
$$B = b_0 + (b_1, ..., b_k, 0, b_r)^T X,$$

which do not share the entry for a_r and b_r. It is important to note that this definition is only true for *first-order* sensitivities since the correlation coefficient is a measure of linear dependence.

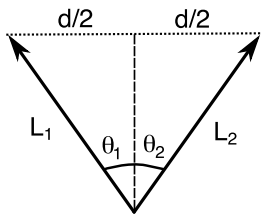

Fig. 1. Simplified trigonometry of distance and angle proportionality approximation.

Our goal is to minimize nominal skew and increase the correlation between a pair of parametric sink delays, S_i and S_j, by matching sensitivities in the parameterized delays. (6) gave the exact correlation between the two sink delays, but using only this as the objective has two challenges: it ignores the expected delay and it is non-linear. Both of these problems, however, are solved with some simple approximations. In our problem, we simply redefine the sensitivity vector to include the nominal delay. A vector S_i is then

$$S_i = (\mu_i, \sigma_{i,0}, \sigma_{i,1}, ..., \sigma_{i,k}, \sigma_{i,r})^T. \tag{7}$$

This means that the cosine of the angle between two vectors is no longer exactly the correlation, but it is observed to still be a reasonable predictor of correlation.

If two vectors have the same length, $L = L_1 = L_2$, the angle between the two vectors, θ, will be related to the distance, d, between the two vectors heads according to simple trigonometry,

$$\theta = \theta_1 + \theta_2 = \arcsin\left(\frac{d}{2L_1}\right) + \arcsin\left(\frac{d}{2L_2}\right)$$
$$= 2\arcsin\left(\frac{d}{2L}\right),$$

as illustrated in Fig. 1. The triangle can be bisected into two right triangles only if it is isosceles. Based on (4), the correlation is then

$$\rho = \cos\left(2\arcsin\left(\frac{d}{2L}\right)\right).$$

If we minimize the distance squared, we obtain a point that also maximizes the cosine of the angle since $\arcsin(0) = 0$ and $\cos(0) = 1$. This derivation is based on the assumption that the vector lengths are *approximately* equal which, in practice, is not a bad assumption.

The distance of the two vector heads can be computed from the vector coordinates directly, but it involves a square root,

$$|S_i - S_j| = \sqrt{(s_{i,0} - s_{j,0})^2 + ... + (s_{i,k} - s_{j,k})^2 + s_{i,r}^2 + s_{j,r}^2}. \tag{8}$$

To avoid the square root, we instead minimize the squared distance between the vector heads,

$$|S_i - S_j|^2 = (s_{i,0} - s_{j,0})^2 + ... + (s_{i,k} - s_{j,k})^2 + s_{i,r}^2 + s_{j,r}^2. \tag{9}$$

It is important to note that the terms $s_{i,r}$ and $s_{j,r}$ are not subtracted since they are independent sources of variation and will not cancel out.

The local skew metric, (9), matches both the nominal and correlated delay sensitivities of a pair of clock sinks while also minimizing the independent sensitivities. The result is that the expected skew will be smaller than the case with less correlation and that the cost function is a simple quadratic rather than a non-linear one. In order to fully define the optimization framework, however, we must introduce the parametric gradient and the quadratic programming problem that are used in the optimization.

IV. Parametric Gradient

Finite differencing of SSTA is used to calculate the parametric gradient. A single SSTA run calculates the present delay of each sink in first-order, parameterized form with no size changes. The gradient procedure then iterates through each buffer and wire, perturbs the size by a small value ϵ, and approximates the gradient with

$$\frac{\partial D_i}{\partial x_j} \approx \frac{D_i(x_j + \epsilon) - D_i(x_j)}{\epsilon} \tag{10}$$

where $D_i()$ is the sink delay in parametric form as a function of the wire/buffer size. The result is a vector

$$\frac{\partial D_i}{\partial x_j} = \begin{pmatrix} \frac{\partial d_{i,0}}{\partial x_j} & \frac{\partial d_{i,1}}{\partial x_j} & \dots & \frac{\partial d_{i,k}}{\partial x_j} \end{pmatrix}^T \tag{11}$$

where i specifies the sink, k specifies the parameter, and x_j specifies the buffer or wire size. All sink sensitivities to a particular variable can be found in a single evaluation and, more importantly, the sensitivity of all parametric values can be found in the same traversal using SSTA. Incremental SSTA also reduces the complexity even more in practice. The added computational complexity over the deterministic gradient is only due to the number of parametric variables.

The finite differencing results in a $(s \times k + 1) \times n$ statistical gradient,

$$G = \begin{pmatrix} \begin{pmatrix} \frac{\partial d_{0,0}}{\partial x_0} & \frac{\partial d_{0,0}}{\partial x_1} & \dots & \frac{\partial d_{0,0}}{\partial x_n} \\ \frac{\partial d_{0,1}}{\partial x_0} & \frac{\partial d_{0,1}}{\partial x_1} & \dots & \frac{\partial d_{0,1}}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial d_{1,k}}{\partial x_0} & \frac{\partial d_{1,k}}{\partial x_1} & \dots & \frac{\partial d_{1,k}}{\partial x_n} \end{pmatrix} \\ \dots \\ \begin{pmatrix} \frac{\partial d_{s,0}}{\partial x_0} & \frac{\partial d_{s,0}}{\partial x_1} & \dots & \frac{\partial d_{s,0}}{\partial x_n} \\ \frac{\partial d_{s,1}}{\partial x_0} & \frac{\partial d_{s,1}}{\partial x_1} & \dots & \frac{\partial d_{s,1}}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial d_{s,k}}{\partial x_0} & \frac{\partial d_{s,k}}{\partial x_1} & \dots & \frac{\partial d_{s,k}}{\partial x_n} \end{pmatrix} \end{pmatrix}, \tag{12}$$

when there are s sinks, k parameters, and n total buffers/wires.

At first glance, this gradient may seem quite large. However, in practice, it is extremely sparse. The density depends heavily on the number of buffers in the tree since these buffers shield size changes from sibling subtrees. The smallest benchmarks (s1423 and s5378) have the highest density of about 50% due to very few buffers. The larger benchmarks have densities of less than 10%.

For our experiments, we use the above analytic SSTA and finite differencing, but this is not a requirement. Any method can be used to calculate the statistical gradient including multiple perturbation, Monte Carlo of both parameter variations and buffer and wire sizes, or symbolic differentiation. The only requirement is that we obtain the statistical gradient with partial derivatives of the entire first-order form for every sink with respect to every buffer and wire size.

V. Sequential Programming

The new quadratic cost formulation fits nicely into a quadratic programming problem. Even though we are not maximizing the exact correlation, our heuristic is a reasonable approximation and includes the nominal skew as part of the objective. The squared difference is analogous to a set of springs in $k+1+s$ dimensional space that pulls both the nominal skew and each parametric sensitivity together.

The nominal delay of each sink is always bigger than the parametric sensitivities, but this does not mean that it will make the sensitivities insignificant in practice. Particularly, we are interested in minimizing the difference of the nominal delays which can be quite small. A weighting factor was investigated, but deemed unnecessary for good results in our experiments.

Our new approach uses a similar sequential programming framework that was used for deterministic optimization in our prior work [8]. The objective, however, is replaced with our new sensitivity-aware cost function,

$$\Phi(S) = \sum_{i>j} |S_i - S_j|^2, \tag{13}$$

where each local skew metric is like (9). This objective also depends on the partial derivatives of the sensitivities. Each iteration of statistical sequential quadratic programming (SSQP) is formulated with the following quadratic programming subproblem,

$$\begin{aligned} \min \quad & \Phi(S) = \sum_{i>j} |S_i - S_j|^2 \\ s.t. \quad & D + G\Delta = S \\ & P_{cur} + \beta\delta \leq P_{bnd}. \end{aligned}$$

The gradient, G, was described in Section IV. D and S are matrices of the delays of each clock sink in parameterized form with the present iteration's size changes (S) and without the proposed size changes (D). Each D_i and S_i in the matrices,

$$D = (D_0^T, D_1^T, ..., D_s^T)^T$$
$$S = (S_0^T, S_1^T, ..., S_s^T)^T,$$

is a vector that describes the parametric delay at each of the s sinks like in (7).

To exemplify how the above algorithm can improve clock sink delay correlations, the sensitivity vector plots before any sizing, after deterministic sizing, and after our new sensitivity matching approach are shown in Fig. 2 for benchmark s1423 with no data-path variation or information (i.e. the skew between all pairs is minimized). Only two dimensions of parameters are shown in addition to the nominal value due to the difficulties in representing more than three dimensions. In Fig. 2a,

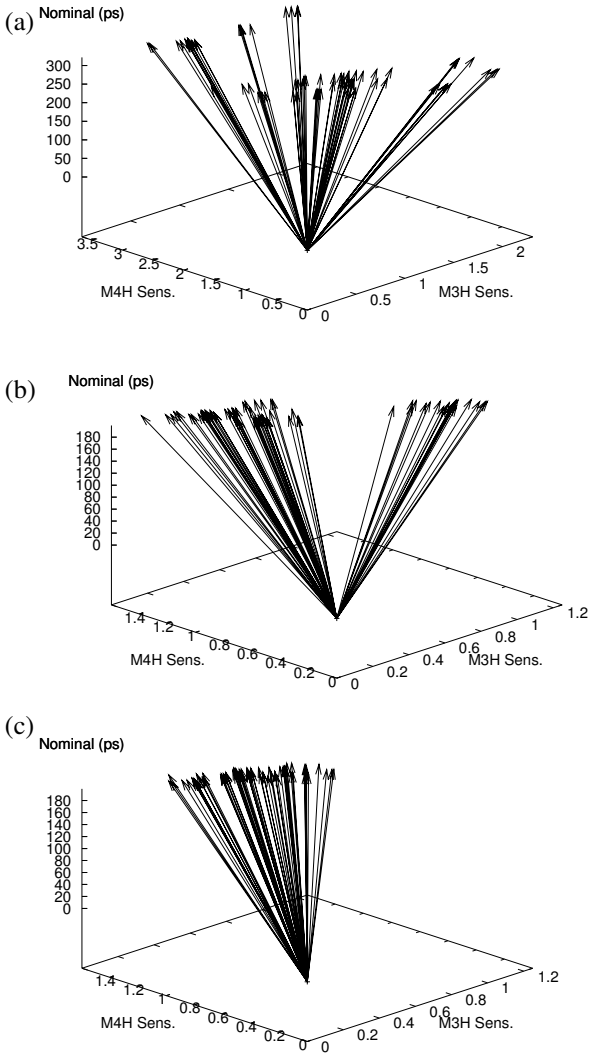

Fig. 2. Sensitivity vectors of s1423 (a) before tuning, (b) tuned for near zero deterministic skew, and (c) after sensitivity matching.

it is evident that there is considerable nominal skew before tuning. The vectors are of different lengths in the nominal dimension and are uncorrelated in terms of parametric sensitivities. Fig. 2b shows the results after deterministic tuning. It is clear that the nominal skew is balanced quite well but there are two main "clusters" of skew sensitivities. One cluster is highly dependent on M3 height variation while another cluster is more dependent on M4 height variation. The statistical sizing algorithm is able to modify these sensitivities by relying more on buffer delays to reduce the impact of the wire variation. It produces the results in Fig. 2c. There is one large cluster of vectors which means that the total of the angles between all the vectors is drastically reduced and the sink delays are therefore more correlated. If a parameter varies, rather than introducing skew, it is likely that all the other sinks are affected in the same way, thus preserving the low skew property. The statistical tuning result, therefore, is more robust and has better expected skew and skew sensitivity.

VI. Data-Path Matching

The true advantage of the sensitivity matching heuristic, however, is lost in other benchmarks when we consider the global skew between all sink pairs. In global skew, there are critical data-path connections between all pairs of sequential elements which usually results in an incompatible set of path correlations. Our algorithm then has an infeasible problem for improving correlation. Luckily, however, this is seldom seen in practice because typical designs have relatively few data-path connections compared to the total number of clock sink pairs. This same fact allows our method to make improvement when considering actual logical data-path information.

In our formulation, we wish to trade-off sensitivities of a clock sink so that the impact of variation can be mitigated. Fig. 3 helps illustrate the scenario. All timing constraints can be written as a round-trip constraint that include the latencies to each clock tree sink. The setup constraint for this example would be

$$d_{C2} + t_{dCQ} + d_{D1} + d_{D2} \leq T_{cycle} - t_s - d_{C3}$$

where t_{dCQ} is a clock-to-q flip-flop delay, t_s is a setup time, and the d values are individual gate delays. A similar constraint exists for hold violations. Decreasing the delay of clock gate C2 and increasing the delay of clock gate C3 both decrease the minimum cycle time. However, each of these skews can impact adjacent stages of logic. Prior useful skew algorithms have presented algorithms to find useful skews to achieve the optimal cycle time of a design [7], but they do not discuss the actual implementation of the clock and how process variations can negatively impact this intentional skew. In fact, useful skew budgets must still include a guard-band in practice to prevent failures due to process variations.

Our approach, on the other hand, considers the nominal useful skew target and tries to match sensitivities to produce more robust results. In Fig. 3, for example, we have two choices to improve performance: speed up the clock to FF1 or slow down the clock to FF2. Without considering process variations, there are many options that have the same effect. For example, wire n2 can be made wider so that it presents more load to gate C3; gate C3 can be made smaller so that it has larger delay; or wire n1 can be made narrower to increase its resistivity.

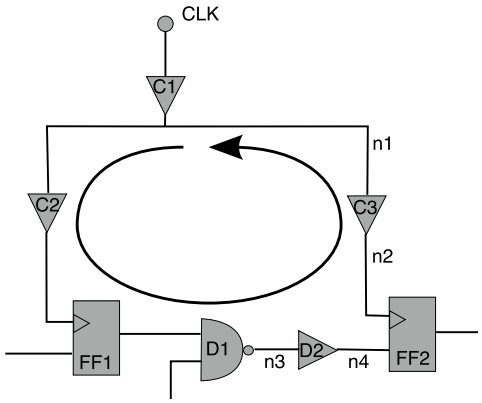

Fig. 3. Round-trip timing can leverage correlation for improved performance.

These options are just for slowing down the branch to FF2 - similar options exist to speed up the branch to FF1. Deterministic optimization would do some combination of these moves to quickly converge to a solution.

If we consider process variations, however, some of these options are less attractive than others due to the correlation between the data-path delay from FF1 to FF2 and the clock tree skew between the clock nodes of FF1 and FF2. To make the design more robust, it is best that these two delays be correlated. If they are correlated, a process parameter will affect both the data-path delay and clock skew equally and, in turn, not impact performance. For example, if the data-path is gate delay dominated, we may wish to add extra delay in the clock tree by sizing C3. If, however, the data-path is M3 dominated, we may wish to add delay in the clock tree by sizing an M3 wire to improve the correlation.

This tradeoff is performed implicitly with the sensitivity matching algorithm from Section VI. We add a budget to our cost function that includes a nominal useful skew target. The skew target can also include desired sensitivity characteristics of the most critical data-path(s) between a pair of sequential elements which is available from SSTA. The resulting objective, with the desired budget, is

$$\Phi(S) = \sum_{i>j} |S_i + B_{i,j} - S_j|^2. \quad (14)$$

VII. Experimental Results

Prior deterministic sequential linear programming (SLP) [20] and sequential quadratic programming (SQP) [8] algorithms are compared to our new statistical SQP approach. In addition, a criticality weighted SQP algorithm, similar to the criticality weighted gate sizing algorithm in [9] is also implemented for comparison.

The benchmark clock trees use a 32nm process as predicted by the ITRS [12] and the Berkeley Predictive Models for devices [22]. There were twelve sources of variation modeled including: L_{eff}, V_{th}, M3 width, M3 height, M3 ILD thickness, M3 spacing, M4 width, M4 height, M4 ILD thickness, M4 spacing, metal resistivity and dielectric constant with both inter- and intra-die components similar to [5].

Resistance and capacitance models from [11] and [21] are used, respectively. The timing analyzer implements delay and slew rate propagation in the clock tree according to buffer delay models fit to Spice and according to the S2M metric for interconnect [2]. In the statistical case, only the mean slew over interconnect is calculated according to S2M and parametric sensitivities are propagated without modification.

Data-path connectivity information was randomly generated since it is not available in the benchmark sets. Each sequential element is connected to two other sequential elements. The nominal useful skew budgets are in the range of $0 - 10ps$ and data-path sensitivity to L_{eff} variation is in the range of $0 - 20ps$. It is assumed that the data-path is routed in metal M1 and M2 while the clock tree is in layers M3 and M4. Therefore, there is no potential sharing of metal variation which makes these results pessimistic. Only the statistical algorithm considers the data-path parametric sensitivity to L_{eff}.

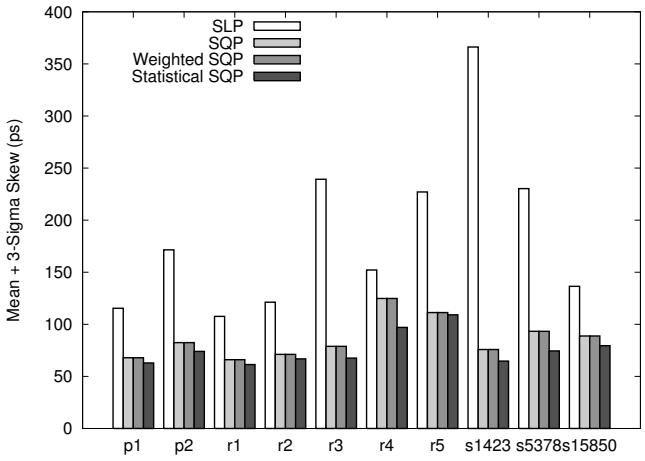

Fig. 4. 99.8% quantile skew for deterministic and variation-aware tuning methods.

Fig. 4 demonstrates the $\mu + 3\sigma$ or 99.8% quantile skew for the different optimization methods. In this technology, the deterministic SLP approach is woefully inadequate in every benchmark, but deterministic SQP does fairly well. The criticality-weighted SQP approach makes absolutely no improvement over the deterministic SQP approach. The Statistical SQP algorithm, however, does better in every case and significantly better in some cases. The largest improvement is on r4 and is about 27ps. The least improvement is on r5 and is slightly less than 3ps.

Table I shows the results for all the benchmarks with SQP and Statistical SQP. The skew improvements typically consume a small amount of additional power despite being given the same power constraint. The average power overhead is approximately 2%. It is interesting to note, however, that sometimes there is no overhead as in the case of s1423 and sometimes there is a decrease as in benchmark r5. r5, however, also showed the least improvement in skew.

The one disadvantage of the SSQP approach is run-time. The average benchmark uses approximately $50\times$ the CPU time of deterministic SQP, but it is still significantly less than general purpose non-linear optimization. The large gradient, though sparse, has more constraints than the deterministic algorithms. In fact, the number of constraints are multiplied by the number of sources of variation. Since 12 variation parameters were implemented, this accounts for at least an order of magnitude more constraints. Since we assumed that the data-path used M1/M2 while the clocks were on M3/M4, we did not need to model all of the parameters. Reducing the number of variation parameters to only significant ones would significantly reduce the run-time.

VIII. Conclusions

This paper is the first demonstration of a heuristic to intentionally correlate clock and data-path parametric sensitivities to make a design more robust. Previous works have considered clock trees and data-paths in isolation, but this work optimizes clock trees so that they are aware of data-path sensitiv-

TABLE I
OPTIMIZATION RESULTS FOR SQP AND STATISTICAL SQP.

	SQP					Statistical SQP				
Benchmark	Det. Skew	μ (ps)	σ (ps)	$\mu + 3\sigma$ (ps)	Cap (pF)	Det. Skew	μ (ps)	σ (ps)	$\mu + 3\sigma$ (ps)	Cap (pF)
p1	13.2	35.4	10.8	67.8	6.6	13.2	30.4	10.8	62.8	7.0
p2	19.8	47.6	11.6	82.4	15.9	15.2	40.7	11.0	74.0	16.3
r1	14.8	34.0	10.6	66.0	6.8	12.9	29.6	10.5	61.4	7.1
r2	12.9	37.8	11.0	71.1	14.3	12.3	34.1	10.8	66.8	14.8
r3	30.9	45.9	10.9	78.8	18.2	18.9	35.5	10.6	67.5	18.7
r4	61.1	79.4	15.1	124.8	35.0	38.2	60.3	12.2	97.0	35.9
r5	40.0	67.7	14.5	111.2	53.8	27.0	68.4	13.5	109.1	52.0
s1423	11.0	38.6	12.3	75.8	5.4	11.1	32.6	10.6	64.6	5.4
s15850	42.6	57.7	11.8	93.2	23.0	14.2	40.3	11.3	74.4	23.7
s5378	39.6	53.5	11.7	88.7	8.6	17.1	43.6	11.9	79.4	8.8
Average Improvement (%)						30.2	16.3	5.9	11.9	-2.1

ities, which results in improved robustness. A quadratic programming heuristic was introduced that is able to make this improvement in correlation without sacrificing deterministic skew by performing sensitivity matching. The resulting trees have an average of 16.3% improvement in expected skew and 11.9% improvement in the $\mu + 3\sigma$ skew over a previous deterministic SQP algorithm from [8]. The improvements are made with only about 2% power overhead. The main disadvantage of the method is added run-time. However, by isolating the most significant sources of variation and implementing fully sparse constraints, we can potentially reduce the number of constraints and hence the overall run-time.

REFERENCES

[1] A. Agarwal, D. Blaauw, and V. Zolotov. Statistical clock skew analysis considering intra-die process variations. In *ICCAD*, pages 914–920, 2003.

[2] K. Agarwal, D. Sylvester, and D. Blaauw. A simple metric for slew rate of RC circuits based on two circuit moments. *TCAD*, 23(9):1346–1354, September 2004.

[3] C. Albrecht, A. B. Kahng, B. Liu, I. Mandoiu, and A. Zelikovsky. On the skew-bounded minimum buffer routing tree problem. In *SASIMI*, pages 250–256, 2001.

[4] K. Boese and A. Kahng. Zero-skew clock routing trees with minimum wirelength. In *ASIC Conf.*, pages 1.1.1–1.1.5, 1992.

[5] Y. Cao, P. Gupta, A. B. Kahng, D. Sylvester, and J. Yang. Design sensitivities to variability: Extrapolations and assessments in nanometer VLSI. In *ASIC/SOC*, pages 411–415, September 2002.

[6] T.-H. Chao, Y.-C. Hsu, and J. Ho. Zero skew clock net routing. In *DAC*, pages 518–523, 1992.

[7] J. P. Fishburn. Clock skew optimization. *IEEE Transactions on Computers*, 39(7):945–951, July 1990.

[8] M. R. Guthaus, D. Sylvester, and R. B. Brown. Clock buffer and wire sizing using sequential quadratic programming. In *DAC*, pages 1041–1046, July 2006.

[9] M. R. Guthaus, N. Venkateswaran, C. Visweswariah, and V. Zolotov. Gate sizing using incremental parameterized statistical timing analysis. In *ICCAD*, pages 1029–1036, November 2005.

[10] M. Hashimoto, T. Yamamoto, and H. Onodera. Statistical analysis of clock skew variation in H-tree structure. In *ISQED*, 2005.

[11] S. Im, N. Srivastava, K. Banerjee, and K. E. Goodson. Scaling analysis of multilevel interconnect temperatures for high performance ICs. *IEEE Transactions on Electron Devices*, 52(12):2710–2719, 2005.

[12] International technology roadmap on semiconductors. http://www.itrs.net/Common/2005ITRS/Home2005.htm, 2005.

[13] I. S. Kourtev and E. G. Friedman. A quadratic programming approach to clock skew scheduling for reduced sensitivity to process parameter variations. In *ASIC/SOC*, pages 210–215, 1999.

[14] I.-M. Liu, T.-L. Chou, A. Aziz, and D. F. Wong. Zero-skew clock tree construction by simultaneous routing, wire sizing and buffer insertion. In *ISPD*, pages 33–38, 2000.

[15] B. Lu, J. Hu, G. Ellis, and H. Su. Process variation aware clock tree routing. In *ISPD*, pages 174–181, 2003.

[16] J. L. Neves and E. G. Friedman. Optimal clock skew scheduling tolerant to process variations. In *DAC*, pages 623–629, 1996.

[17] S. Pullela, N. Menezes, and L. T. Pillage. Reliable non-zero skew clock trees using wire width optimization. In *DAC*, pages 165–170, 1993.

[18] G. E. Tellez and M. Sarrafzadeh. Minimal buffer insertion in clock trees with skew and slew rate constraints. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 16(4):333–342, 1997.

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

[20] K. Wang and M. Marek-Sadowska. Buffer sizing for clock power minimization subject to general skew constraints. In *DAC*, pages 159–164, 2004.

[21] S.-C. Wong, G.-Y. Lee, , and D.-J. Ma. Modeling of interconnect capacitance, delay, and crosstalk in VLSI. *IEEE Transactions on Semiconductor Manufacturing*, 13(1):108–112, February 2000.

[22] W. Zhao and Y. Cao. New generation of predictive technology model for sub-45nm design exploration. In *ISQED*, pages 585–590, 2006.