Computing Clock Skew Schedules Under Normal Process Variation

Aaron P. Hurst and Robert K. Brayton
Dept. of Electrical Engineering and Computer Sciences
University of California, Berkeley
Berkeley, CA 94720
{ahurst, brayton}@eecs.berkeley.edu

***Abstract*— Statistical timing techniques facilitate an analysis of the effects of process variation on performance and yield. Traditional worst-case linear models are necessarily overconservative, and the required margin of error is growing with device technology. Clock skew scheduling is typically formulated as a linear problem and thus suffers from the limitations of this model, resulting in significant yield suboptimality. In this paper, we present a technique for generating clock skew schedules based on a statistical model of timing. We describe a computationally tractable but accurate approximation and use conjugate gradient minimization to efficiently compute a schedule to maximize the yield for a target clock period. On a set of bechmarks, our results indicate an average 13x decrease in the failure rate for a single target clock period or an equivalent 4% improvement in the achievable clock period for a single target yield.**

I. Introduction

With greater integration and decreasing feature sizes, managing process variation is becoming an increasing challenge. The traditional solution is to include enough conservatism in timing analysis to ensure proper device functionality in the presence of the most antagonistic variations. However, the size of this required timing margin is becoming an increasing burden. It is not uncommon for manufactured designs to reliably operate at much higher frequencies than their design specifications. This results in a significant waste of design effort to meet performance requirements that were never actually a problem and a very real cost associated with this overdesign.

The effects of variation on device performance can be accounted for in a statistical timing analysis in which temporal quantities are modeled as probability distributions. Knowledge about the variation of the individual components, their correlation, and the structure of the circuit can be combined to derive statistical information about the design variables of interest, including maximum operating frequency. This result can then be used to make a more informed decision about performance requirements in the presence of process variation than is possible under the worst-case model.

Clock skew scheduling is typically formulated also using the linear worst-case timing model. The inaccuracy of this formulation under process variation results in a suboptimal clock skew schedule. A typical circuit will have several co-optimal linear solutions due to timing slack on noncritical paths, and each solution may result in a different yield. It is not clear how to choose the solution that minimizes the failure rate. For example, contrast the case where only a small subset of the paths in a design are timing critical versus the same design with every path made critical because of a particular choice of register skews. The latter is considerably more likely to result in increased timing failure, but the worst-case timing model is unable to distinguish between the two.

This concern has been recognized. [1] propose adding hardware that can be adjusted after manufacturing to compensate for process variations. While this adds the ability to correct for process variation, it would be more efficient and cost-effective to design for yield beforehand. [2] and [3] describe techniques for amending the linear timing model to reduce the chances of failure. [4] describe a method to select amongst the co-optimal schedules by balancing slacks. However, even the optimal linear schedule with the highest yield may still be suboptimal, exactly because the timing model fails to account for statistical effects. Only a weak heuristic solution is possible.

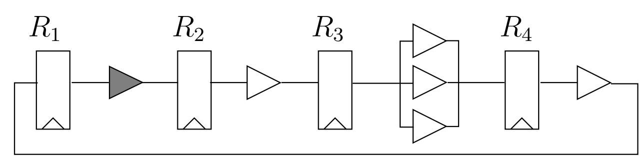


Fig. 1. A sequential circuit. Each gate has an identical worst-case delay, though the shaded gate has a smaller delay σ. (i.e. its delay is less variable under process fluctuations)

Consider the simple sequential circuit in Figure 1. If all of the gates have the same worst-case delay, then the zero-skew clock schedule is the unique linear optimum. This is no longer the case when the timing is probabilistic. Because there are three timing critical paths between R_3 and R_4, the likelihood that the setup condition of R_4 will not be met is higher than for other registers. If we were to skew register R_4 forward, the added slack on its input would improve its reliability, more than offseting the reliability loss due to the tighter timing on the single critical path from R_4 to R_1. Similarly, a yield improvement can be realized by skewing R_2 forward. Because the standard deviation of the delay path from R_1 to R_2 is small, the yield improvement that is seen by affording this path additional slack is more than is lost by borrowing it from the following path.

In this paper, we extend a purely statistical timing model to the problem of computing clock skew schedules. We describe a statistical formulation and then introduce an approximation that is both accurate and computationally tractable. We can then efficiently compute a clock schedule that improves the yield for a target frequency. Equivalently, we can use this technique to maximize the clock period under a particular yield constraint. Our experiments demonstrate that process-variation aware clock scheduling offers significant potential for improving performance and/or yield over traditional techniques.

II. Background

A. Clock Skew Scheduling

The latest arrival times at the latches in a single design may vary considerably. This imbalance may come as a result of timing misprediction in the design flow or because of a fundamental imbalance in the sequential partitioning of a design. Since the latches in a single clock domain must all operate at the same frequency, performance is limited by the slowest delay path, even if the others could operate at a higher speed.

Clock skew scheduling [5] offers a technique for balancing the computation by applying different non-zero delays on the clock inputs of each register. The timing of the design is then no longer limited by the single worst-case path but by the maximum average delay around any loop of register-to-register path segments. In contrast to retiming, clock skew scheduling also possesses the desirable feature of preserving circuit structure and functionality. In recent years, it has gained practical acceptance in multiple design tools, usually at the end of the flow after physical synthesis is nearly complete.

The problem of computing an optimal clock skew schedule is typically formulated as a linear program. The objective is to minimize the clock period by choosing a set of register skews, subject to the linear constraints arising from setup and hold constraints along each register-to-register timing path. Figure 2 shows a single register-to-register timing path with a maximum delay D and minimum delay d. The associated setup and hold constraints are Equations 1 and 2, respectively.

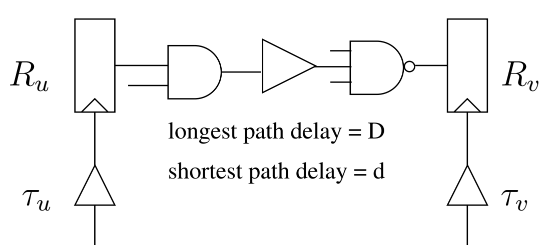


Fig. 2. A single register-to-register path with a clock skew schedule $\vec{\tau}$.

$$S_{u \to v}(T, \vec{\tau}) = D \le T - \tau_u + \tau_v \tag{1}$$
$$H_{u \to v}(\vec{\tau}) = d \ge \tau_u - \tau_v \tag{2}$$

In the linear formulation, D and d are discrete worst-case values, and the resulting linear program is Equation 3. There exist several algorithms [6], for solving this minimization problem, which are quite efficient in practice.

$$\text{minimize } T, \text{ s.t.} \quad \begin{array}{ll} \forall \text{ paths } p & \begin{array}{l} S_p(T, \vec{\tau}) \\ H_p(\vec{\tau}) \end{array} \\ \forall \tau_i \in \vec{\tau} & \tau_i \ge 0 \end{array} \tag{3}$$

B. Statistical Timing Analysis

In statistical timing analysis, timing quantities of interest are random variables. For example, in Figure 2, we can express D and d as distributions, which describe the probability that a manufactured instance has a particular shortest and longest path delay. There is then an associated probability that each of the setup and hold constraints, Equations 1 and 2, will be met.

Given information about the statistical behavior of standard cells in a technology library, several techniques have been proposed to derive global timing information about a design. A Monte Carlo static timing analysis is potentially accurate but expensive. [7], [8], and [9] describe techniques for a block-based single-pass analysis.

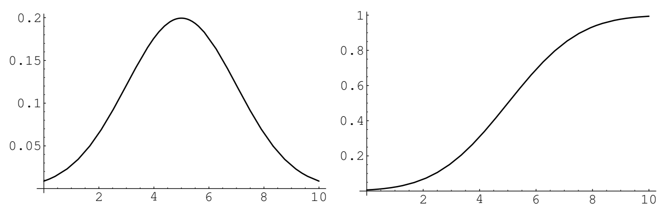
Much of the complexity in block-based statistical timing analysis is in storing and manipulating the statistical quantities and their relation with each other. Propagating arbitrary probability distributions through the circuit can lead to exponential growth in the size of the representations with the size of the circuit. Maintaining joint probabilities is also exponential, and even pair-wise simple correlation is quadratic. Simplification is clearly necessary to approach the efficiency and utility of linear static timing analysis. Le and Pileggi [10] propose approximating all statistical quantities as Gaussian (normal) distributions and conclude that this does not significantly affect the accuracy of the result. They also demonstrate a technique for controlling the number of pairwise correlations stored by discarding information when no longer needed.

III. Statistical Clock Skew Scheduling

A. Problem Formulation

To restrict the complexity of the analysis, we also approximate all statistical quantities as normal distributions. A normally distributed random variable X is defined by exactly two quantities, a mean $\mu \in \Re$ and a standard deviation $\sigma \in \Re \ge 0$. For any random variable, there is an associated probability density function (PDF), defined by Equation 4, and cumulative distribution function (CDF), defined by Equation 6. For normal random variables, the functions are of the specific forms of Equations 5 and 7. An example is illustrated in Figure 3.

$$X^{PDF}(t) = \mathcal{P}(X = t) \tag{4}$$
$$= \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(t-\mu)^2}{2\sigma^2}} \tag{5}$$
$$X^{CDF}(t) = \mathcal{P}(X \le t) \tag{6}$$
$$= \int_{-\infty}^{t} X^{PDF}(u) du$$



(a) (b)

Fig. 3. The (a) PDF and (b) CDF of a normal random variable with $\mu = 5$ and $\sigma = 2$.

$$= \frac{1}{2}\left[1 + erf\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right] \tag{7}$$

Static timing analysis requires that two classes of operations are defined for these statistical quantities: *sum*(X, Y) and *max*(X, Y). The complementary operations, *diff*(X, Y) and *min*(X, Y), can be derived similarly. To maintain that all statistical quantities are Gaussian, each operation, when restricted to a domain of normal variables, must result in a normal distribution. While the sum of two normal variables fulfills this requirement, the maximum operator does not. Clark [11] proposed a normal approximation to the maximum of a set of normal variables, which seems to be quite good in practice.

In a design, the overall yield, Equation 8, is the probability that the timing constraints (both setup and hold) are met for every one of the paths. As formulated in Equations 1 and 2, these are a function of clock period T and schedule $\vec{\tau}$. If we select a particular schedule and examine $Y(T, \vec{\tau})$ along the T axis, we know that the yield must be monotonically increasing, that is, if one slows the clock period, the yield will improve. This one-dimensional yield function is a CDF of the probability that a design will operate at a particular frequency.

$$Y(T, \vec{\tau}) = \mathcal{P}\left(\bigwedge_{\forall \text{ paths } p} S_p(T, \vec{\tau}) \wedge H_p(\tau)\right) \tag{8}$$

The statistical clock scheduling problem is that for a given period T, find some clock schedule $\vec{\tau_{opt}}(T)$ that maximizes the yield of the design. The optimal clock schedule may differ for each target period T. Also note that $\vec{\tau_{opt}}(T)$ is not unique, since all timing constraints are functions of the difference of a pair of register skews. Therefore, a single constant can be added to every skew to obtain a schedule with equivalent properties.

B. Computing Path Delays

It is first required to compute the distribution of the longest and shortest delays along each register-to-register combinational path. Given the delay distribution of each gate and the correlation between them, the distributions of the earliest and latest arrival times can be computed at each point in the circuit as a *sum* of the delays with a repeated application of the *min* and *max* operators, respectively, at each point of convergence. We follow closely the procedure described in [10].

C. Approximating the Yield

There are two obstacles to computing $\vec{\tau_{opt}}(T)$ directly. First, even though each of the components of Equation 8 is Gaussian, the conjunction is non-normal and no closed-form expression exists. Second, the yield function is non-convex in the τ-dimension space. Finding the exact optimum of the non-convex yield equation appears to be an intractable problem. To approach the statistical clock scheduling problem, introduce a simplification that lends itself to efficient approximation.

The general procedure is as follows. Let us consider a single vector in the input space of $Y(T, \vec{\tau})$, resulting in a one-variable function $Y(t)$ where t is the distance along that vector. Because each timing constraint is a linear inequality of the single-valued variables T and $\vec{\tau}$ and one normal random variable X (either a longest or shortest path delay), it reduces to a linear constraint of the form $(X < at + b)$ along this vector. Because normal distributions remain normal under linear transformation, we can rewrite this linear constraint as the CDF of some normal variable X'.

$$\mathcal{P}(X < at + b) = \mathcal{P}(X' < t) \tag{9}$$
$$\text{where } X' = \begin{cases} \mu' = \dfrac{\mu - b}{a} \\ \sigma' = \dfrac{\sigma}{a} \end{cases}$$

Derivation:

$$\begin{aligned} \frac{t-\mu'}{\sigma'} &= \frac{t - \frac{\mu-b}{a}}{\sigma/a} \\ &= \frac{(at+b)-\mu}{\sigma} \\ \frac{1}{2}\left[1 + erf\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right] &= \frac{1}{2}\left[1 + erf\left(\frac{x-\mu'}{\sigma'\sqrt{2}}\right)\right] \\ X^{CDF}(at+b) &= X'^{CDF}(t) \\ \mathcal{P}(X < at+b) &= \mathcal{P}(X' < t) \end{aligned}$$

However, since the standard deviation, σ', of the random variable X' is defined to be a nonnegative real number, we must fix a to be nonnegative, possibly leading to a constraint of $(X' < -t)$. Whether a constraint reduces to the form $(X' < t)$ or $(X' < -t)$ depends on the original sign of t.

For a particular point in the optimization space and some t, we can apply the above procedure to find a set of equivalent normal variables $\vec{X'}$, at most one for each of the timing constraints. Since a constraint may not be a function of t, it may not have any variable X' associated with it. These variables will be grouped into two subsets: $\vec{X}'^{+}$, those that result from a constraint that is a function of a t, and $\vec{X}'^{-}$, those that result from a constraint that is a function of $-t$. There is one such partition for each direction, regardless of the choice of point.

As an example, consider the setup constraint on R_v from Figure 2. If we wish to compute $Y(\tau_u)$, the resulting X' associated with this constraint will be in the subset $\vec{X'}^{-}$ with $\mu_x = \mu_D - T - \tau_v$ and $\sigma_x = \sigma_D$.

Next, we can repeatedly apply Equation 10 and the maximum approximation to rewrite the yield in the simpler form of Equation 12. This expression has a closed form, a critical feature for allowing us to continue with the optimization.

Let Z_1 and Z_2 be two random variables.

$$\mathcal{P}(Z_1 \leq t \wedge Z_2 \leq t) = \mathcal{P}(max(Z_1, Z_2) \leq t) \tag{10}$$

$$Y(t) = \mathcal{P}\begin{pmatrix} max(X_i'^{+}, max(X_j'^{+}, ...)) \leq t \\ \wedge max(X_i'^{-}, max(X_j'^{-}, ...)) \leq -t \end{pmatrix} \tag{11}$$

Let M^+ and M^- be the maximum of the sets $\vec{X}'^{+}$ and $\vec{X}'^{-}$ respectively.

$$\begin{aligned} &= \mathcal{P}(M^+ \leq t \wedge M^- \leq -t) \\ &= \mathcal{P}(M^+ \leq t) - \mathcal{P}(M^+ \leq t \wedge -M^- \leq t) \\ &= \mathcal{P}(M^+ \leq t) - \mathcal{P}(max(M^+, -M^-) \leq t) \end{aligned} \tag{12}$$

We can use this same technique to simplify the gradient of the yield function, $(\vec{\nabla} Y)$. The above procedure is repeated to derive Y as a one-variable function of each $\vec{\tau}$. The set of derivatives of these equations is exactly $(\vec{\nabla} Y)$.

The expression of $Y(t)$ in the form of Equation 12 was derived through a repeated application of the *max* operator. In general, the random variables M^+ and M^- will be non-normal because of the nature of this operation. However, if we instead use the approximated *max* operator described in [11], we can generate a normal approximation to these two quantities. If these approximation are substituted into Equation 12, let the resulting function be Y_{aprx} and the corresponding gradient be $(\vec{\nabla} Y)_{aprx}$. For each gradient direction τ_i, only the setup and hold constraints for paths adjacent to the register R_i will be functions of τ_i. The size of the resulting set X' will typically be small, requiring only a few *max* operations and introducing minimal approximation error.

D. Optimization

An important feature of using this operator as the core of the approximation is that it is correlation-aware. The probabilities associated with the setup and hold constraints are not independent, due to both global variations and shared logic. Because the procedure used to calculate the longest and shortest path delays also returns their pairwise correlation, this gives information about the correlation between the constraints.

As an optimization technique, the conjugate gradient method offers a robust and efficient procedure for the local maximization of differentiable functions. We use it to find the maximum of Y_{aprx}. While non-linear optimization only guarantees a local maximum, we seek a global maximum. Fortunately, even though Y_{aprx} is nonconvex, it has a single local maximum for each value of T. The proof follows.

E. Proof

Theorem 1: $Y_{aprx}(T, \vec{\tau})$ has a single local maximum in $\vec{\tau}$ for every value of T.

Proof: A multi-dimensional function has at most a single local maximum if every line therein has at most a single local maximum (Lemma 2). Using the previously defined technique to write Y_{aprx} as a function of a single variable, we can substitute *any* one-dimensional line into $\vec{\tau}$ and write Y_{aprx} in this form. An equation of this form has at most a single local maximum (Lemma 3). Therefore, $Y_{aprx}(T, \vec{\tau})$ can only have a single local maximum for every value of T. ■

Lemma 2: A function $f(\mathbf{x}) : \Re^n \rightarrow \Re$ has at most a single local maximum if for every line $\mathbf{L}(t) : \Re \rightarrow \Re^n = \mathbf{M} \cdot (t - \mathbf{b})$, there is at most a single local maximum in $f(\mathbf{L}(t)) : \Re \rightarrow \Re$.

Proof: Assume there are at least two local maxima, $\mathbf{x_1}$ and $\mathbf{x_2}$, in $f(\mathbf{x})$. As local maxima, for any $|\Delta x| > 0$ sufficiently small

$$\begin{aligned} f(\mathbf{x_1}) > f(\mathbf{x_1} + \mathbf{\Delta x}) \\ f(\mathbf{x_2}) > f(\mathbf{x_2} + \mathbf{\Delta x}) \end{aligned}$$

There exists a line $\mathbf{x} = \mathbf{L_{12}}(t) = \mathbf{M_{12}} \cdot (t - \mathbf{b_{12}})$ that passes through both points at t_1 and t_2, respectively. Let Δt be a small $\Re$ and $\mathbf{\Delta x} = \mathbf{M_{12}} \cdot (\Delta t - \mathbf{b_{12}})$. Consider the small movements only along this line. Substituting this into the above,

$$\begin{aligned} f(\mathbf{L}(t_1)) > f(\mathbf{L}(t_1 + \Delta t)) \\ f(\mathbf{L}(t_2)) > f(\mathbf{L}(t_2 + \Delta t)) \end{aligned}$$

$f(\mathbf{L}(t))$ must then also have local maxima at both t_1 and t_2. Therefore, if every line only has at most one local maximum, then $f(\mathbf{x})$ has at most one local maximum.

■

Equation 12 can be generated for t along any arbitrary direction in the input space of Y, not just a variable axis (as was necessary for $\vec{\nabla} Y$). Consider substituting the line vector $\mathbf{L}$(t) into $Y(T, \vec{\tau})$; each constraint will remain linear and of the form $(X < at + b)$. By applying the *max* operator one more time to the resulting M^+ and $-M^-$, we can compute a normal random variable N that is the approximated maximum, leaving only

$$\mathcal{P}(M^+ \leq t) - \mathcal{P}(max(M^+, -M^-) \leq t) \approx M^{+\,CDF}(t) - N^{CDF}(t)$$

Lemma 3: Let M^+ and N be normal random variables. $(M^{+\,CDF}(t) - N^{CDF}(t))$ has at most one local maximum.

Proof: This equation is the difference of two CDFs, and the roots of its first derivative are

$$\frac{\partial}{\partial t} M^{+\ CDF}(t) - N^{CDF}(t) = 0$$

$$M^{+\ PDF}(t) - N^{PDF}(t) = 0$$

$$\left(\frac{1}{\sigma_m\sqrt{2\pi}} e^{-\frac{(t-\mu_m)^2}{2\sigma_m^2}}\right) - \left(\frac{1}{\sigma_n\sqrt{2\pi}} e^{-\frac{(t-\mu_n)^2}{2\sigma_n^2}}\right) = 0$$

$$\ln\left[\frac{1}{\sigma_m\sqrt{2\pi}} e^{-\frac{(t-\mu_m)^2}{2\sigma_m^2}}\right] = \ln\left[\frac{1}{\sigma_n\sqrt{2\pi}} e^{-\frac{(t-\mu_n)^2}{2\sigma_n^2}}\right]$$

$$\ln(\sigma_m) + \frac{(t-\mu_m)^2}{2\sigma_m^2} = \ln(\sigma_n) + \frac{(t-\mu_n)^2}{2\sigma_n^2}$$

$$\begin{bmatrix} \sigma_m^2 - \sigma_n^2 \\ 2\sigma_n^2\mu_m - 2\sigma_m^2\mu_n \\ 2\sigma_n^2\sigma_m^2 \ln\frac{\sigma_n}{\sigma_m} + \sigma_m^2\mu_n^2 + \sigma_n^2\mu_m^2 \end{bmatrix}^T \begin{bmatrix} t^2 \\ t \\ 1 \end{bmatrix} = 0$$

This system has up to two roots, only one of which may be a maximum. ■

IV. Experimental Results

A. Accuracy

The use of the normal approximation to the *max* operator has been experimentally demonstrated to contribute minimal error in combinational statistical timing analysis [10]. It appears to be equally applicable to the sequential timing domain with a similarly small error penalty. Figure 4 plots the functions $Y(T)$ (computed for each point using a Monte Carlo simulation of 10,000 iterations) and $Y(T)_{aprx}$ with three fixed schedules. The approximation error is visibly small.

Nonetheless, we present our results using the non-approximated yields. While the approximation is necessary in the optimization procedure, the yield improvements from using the statistical-timing aware scheduling are just as tangible when measured against a nonapproximated simulation of the yield function. All of the experimental data presented here is the result of Monte Carlo simulations with at least an order of magnitude more sample points than is statistically likely to demonstrate a failure under the particular yield constraint.

B. Comparison

We ran the following experiments on a subset of the ISCAS benchmark circuits, mapped to a standard technology library. Because the library lacked information about the statistical timing behavior of each gate, we arbitrarily assumed that the worst-case gate delays were 3σ values, i.e. 99.87% of manufactured instances of each gate meet the worst-case timing. With this assumption, a normal delay distribution can be derived from the typical and worst-case values. We also assumed that half of the variation of each gate delay was globally correlated, and half was local.

Our statistical timing tool was written in C++ and run on a 3.0 GHz Intel machine. A standard implementation of conjugate gradient minimization based on [12] was used.

We applied our clock schedule optimization algorithm to compute a yield-aware clock schedule for each of these benchmarks, and then compared it against two alternatives: no clock schedule, and an optimal schedule under the linear formulation. To solve the linear clock scheduling problem, we used an implementation of Howard's algorithm [13]. This has been empirically verified [6] to be the fastest maximum mean cycle solver for VLSI applications. Along with the maximum mean cycle time, the algorithm returns a schedule that meets this optimal time, and we use this result as the point of comparison.

The results of these three scheduling techniques are compared using two different target metrics: reliability (Table I) and performance (Table II). For a set of target clock periods (arbitrarily selected to be the $T_{3\sigma}$ points of the statistical schedules), Table I demonstrates that the statistical scheduling results in a (logarithmic) average 13x decrease in the failure rate, compared to the yield of the same design with optimal linear scheduling. Similar results are seen for other target clock periods.

Equivalently, statistical scheduling can be used to improve performance under a yield constraint. In II, two types of performance results are presented: T_{wc} and $T_{3\sigma}$. T_{wc} is the worst-case delay along the longest path; this is the overconservative result of a traditional static timing analysis. $T_{3\sigma}$ is the clock period that will result in failures for manufactured instances that fall outside of 3 standard deviations worse than the mean performance. Although this information results from a statistical timing analysis and is not part of a design flow built on the worst-case model, we wish to demonstrate that our optimization clock schedule is better under comparable analyses. The yield-aware schedule realizes a better yield that translates into a 4% gain in performance. If a purely non-statistical design flow is used, our result is 19% better than the worst-case prediction of a linearly scheduled design. Since the problems with the linear model are even worse for complex designs, we expect the improvements to be greater for larger examples.

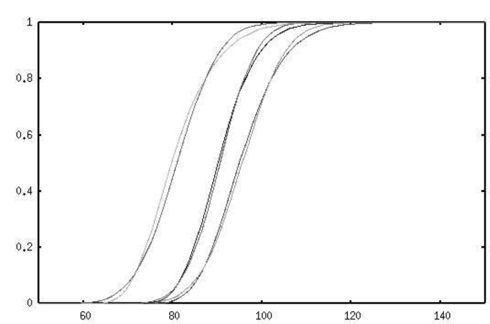


Fig. 4. The yield function $Y(T)$ of design s1196 with no clock scheduling, an optimal linear clock schedule, a statistical clock schedule for an arbitrary target clock period $0.8 * T_{WC}$ (right to left). Each is computed using 10,000 iterations of Monte Carlo simulation per point. The function $Y(T)_{aprx}$ is also drawn.

V. Conclusion

As technology progresses, non-statistical timing analysis requires an increasing amount of conservatism to guarantee worst-case behavior, leading to designs whose manufactured behavior differs significantly from the predictions of design

Design	Size, # Cells	# Setup Constraints	T_{target}	No Sched. Yield	Opt. Linear Sched. Yield	Opt. Linear Sched. Runtime	Statistical Sched. Yield	Improv. f_{fail} (v. Lin.) Raw	Improv. f_{fail} (v. Lin.) Log_{10}	Runtime
s208	81	53	41.6	0.000	0.815	<0.001s	0.999	142x	2.2	0.010s
s298	97	76	38.2	0.417	0.998	0.001s	0.999	2x	0.2	0.008s
s349	149	109	51.8	0.001	0.992	<0.001s	0.999	6x	0.8	0.036s
s382	180	133	40.8	0.000	0.408	<0.001s	0.999	455x	2.7	0.034s
s400	162	133	42.0	0.000	0.925	<0.001s	0.999	58x	1.8	0.038s
s510	248	45	65.4	0.165	0.977	<0.001s	0.999	18x	1.3	0.056s
s641	227	96	65.0	0.956	0.996	<0.001s	0.999	3x	0.5	0.037s
s713	226	96	65.0	0.941	0.996	<0.001s	0.999	3x	0.5	0.036s
s820	360	36	86.0	0.880	0.972	<0.001s	0.999	22x	1.3	0.134s
s1196	646	57	87.7	0.497	0.998	<0.001s	0.999	2x	0.2	0.373s

TABLE I

EXPERIMENTAL RESULTS, YIELD IMPROVEMENT

Design	No Scheduling T_{WC}	No Scheduling $T_{3\sigma}$	Optimal Linear Schedule T_{WC}	Optimal Linear Schedule $T_{3\sigma}$	Statistical Schedule $T_{3\sigma}$	Statistical Schedule % Improv. (v. Lin. $T_{3\sigma}$)
s208	67.8	52.8	53.2	43.8	41.6	4.9%
s298	54.8	45.4	47.0	39.2	38.2	2.5%
s349	88.4	72.7	62.5	53.1	51.8	2.5%
s382	71.9	59.4	51.6	44.6	40.8	8.4%
s400	72.1	59.2	49.9	44.1	42.0	4.9%
s510	98.8	81.3	76.6	68.0	65.4	3.8%
s641	87.0	68.0	83.8	65.7	65.0	1.0%
s713	87.6	68.0	83.8	65.8	65.0	1.2%
s820	117.5	91.8	106.4	91.0	86.0	5.4%
s1196	97.87	125.3	110.7	88.2	87.7	0.7%

TABLE II

EXPERIMENTAL RESULTS, PERFORMANCE IMPROVEMENT

tools. Statistical timing analysis offers a solution by accurately measuring the relationship between performance and yield. The tightest possible timing requirements to meet a yield target can then be computed at design-time.

Traditional clock skew scheduling utilizes a linear model of worst-case timing. We have shown that this leads to suboptimal results when the scheduled circuit is analyzed in the statistical domain. With a statistical formulation of the clock skew scheduling problem, we are able to achieve an increase in yield and/or performance.

ACKNOWLEDGMENT

This work was funded in part by the MARCO Focus Center for Circuit & System Solutions (C2S2, `www.c2s2.org`), under contract 2003-CT-888.

REFERENCES

[1] J.-L. Tsai, D. Baik, C. C.-P. Chen, and K. K. Saluja, "A yield improvement methodology using pre- and post-silicon statistical clock scheduling," in *Proceedings of the International Conference on Computer-Aided Design*, pp. 611–618, 2004.

[2] J. L. Neves and E. G. Friedman, "Optimal clock skew scheduling tolerant to process variations," in *Proceedings of the 33rd Annual Conference on Design Automation*, pp. 623–628, ACM Press, 1996.

[3] I. S. Kourtev and E. G. Friedman, "Clock skew scheduling for improved reliability via quadratic programming," in *Proceedings of the International Conference on Computer-Aided Design*, pp. 239–243, IEEE Press, 1999.

[4] C. Albrecht, B. Korte, J. Schietke, and J. Vygen, "Cycle time and slack optimization for vlsi-chips," in *ICCAD '99: Proceedings of the 1999 IEEE/ACM international conference on Computer-aided design*, pp. 232–238, IEEE Press, 1999.

[5] J. P. Fishburn, "Clock skew optimization," *IEEE Trans. on Computing*, vol. 39, pp. 945–951, July 1990.

[6] A. Dasdan, S. S. Irani, and R. K. Gupta, "Efficient algorithms for optimum cycle mean and optimum cost to time ratio problems," in *Proceedings of the 36th ACM/IEEE conference on Design automation*, pp. 37–42, ACM Press, 1999.

[7] J.-J. Liou, K.-T. Cheng, S. Kundu, and A. Krstic, "Fast statistical timing analysis by probabilistic event propagation," in *Proc. 38th Design Automation Conference*, June 2001.

[8] A. Devgan and C. Kashyap, "Block-based static timing analysis with uncertainty," in *Proceedings of the ICCAD*, pp. 607–614, 2003.

[9] H. Chang and S. S. Sapatnekar, "Statistical timing analysis considering spatial correlations using a single pert-like traversal," in *Proceedings of the ICCAD*, pp. 621–626, 2003.

[10] J. Le, X. Li, and L. T. Pileggi, "Stac: Statistical timing analysis with correlation," in *Proceedings of the 41st Annual Conference on Design Automation*, pp. 343–348, ACM Press, 2004.

[11] C. Clark, "The greatest of a set of random variables," *Operations Research*, 1961.

[12] W. H. Press, W. T. Vetterling, S. A. Teukolsky, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 2 ed., 1992.

[13] J. Cochet-Terrasson, G. Cohen, S. Gaubert, M. McGettrick, and J.-P. Quadrat, "Numerical computation of spectral elements in max-plus algebra," in *Proceedings of the IFAC Conference on System Structure and Control*, July 1998.