Binning Optimization based on SSTA for Transparently-Latched Circuits

Min Gong[1], Hai Zhou[1,2], Jun Tao[1], Xuan Zeng[1,*]
[1]State Key Lab of ASIC & System, Microelectronics Dept., Fudan University, China
[2] EECS, Northwestern University, U.S.A.

***Abstract*— With increasing process variation, binning has become an important technique to improve the values of fabricated chips, especially in high performance microprocessors where transparent latches are widely used. In this paper, we formulate and solve the binning optimization problem that decides the bin boundaries and their testing order to maximize the benefit (considering the test cost) for a transparently-latched circuit. The problem is decomposed into three sub-problems which are solved sequentially. First, to compute the clock period distribution of the transparently-latched circuit, a sample-based SSTA approach is developed which is based on the generalized stochastic collocation method (gSCM) with Sparse Grid technique. The minimal clock period on each sample point is found by solving a minimal cycle ratio problem in the constraint graph. Second, a greedy algorithm is proposed to maximize the sales profit by iteratively assigning each boundary to its optimal position. Then, an optimal algorithm of $O(n \log n)$ runtime is used to generate the optimal testing order of bin boundaries to minimize the test cost, based on alphabetic tree. Experiments on all the ISCAS'89 sequential benchmarks with 65-nm technology show 6.69% profit improvement and 14.00% cost reduction in average. The results also demonstrate that the proposed SSTA method achieves an error of 0.70% and speedup of 110X in average compared with the Monte Carlo simulation.**

Categories and Subject Descriptors:
J.6 [Computer-Aided Engineering]: Computer-Aided Design
General Terms: Design, Algorithms
Keywords: Binning Optimization, SSTA, Latched Circuits

I. Introduction

As IC technology is scaled down to nano-meter regime, process variations become more and more pronounced in fabrication. The increasing process variations lead to considerable uncertainty in circuit performance and large spread in chip speed [1]. To improve the values of fabricated chips, the concept of speed binning is developed, in which the chips are sorted based on their highest workable frequencies and then sold at different prices according to their speeds [2]. The binning process is usually employed on aggressively pipelined high performance systems such as microprocessors, where level-sensitive transparent latches are widely used because of their capability of time borrowing and low power consumption [3]. A new problem arising then is the binning optimization for transparently-latched circuits, which aims to find an optimal bin partitioning to maximize the benefit of production in the binning process.

The benefit of a design is treated as the sales profit in the existing binning research. The sales profit, which is obtained by selling all the working chips at their deserved price, is tightly dependent on the number of chips and the sales price in each bin. A model of sales profit is introduced in [4] as the binning yield-loss based on a linear penalty function, and is improved in [5] as a price-weighted design metric based on the price function in terms of the chip frequency. However, the benefit model considering the sales profit only is not enough in practice. The cost of frequency test, which becomes increasingly prominent in the total cost of today's IC fabrication, should also be taken into consideration. During the binning process, functional or structural tests are run at multiple frequencies and the chips are binned according to the highest speed test they pass [2]. To partition the fabricated chips into the right bins, all the frequencies on bin boundaries need to be tested. The test cost grows up with the increasing number of bins, which will decrease the manufacturer's earning. Therefore a practical benefit model should consider not only the sales profit but also the test cost.

The existing binning optimization methods maximize the sales profit by assigning each bin boundaries to its optimal position. So far as we know, there is only one simple approach [5] proposed for this problem. They pick one boundary at a time, change it by a small step and accept this change if there is improvement in profit. However, the optimality of this approach is not guaranteed and the choice of the step size is ad hoc.

Before the binning optimization, the statistical period, that is the distribution of workable clock period, should be estimated by performing statistical static timing analysis (SSTA) [1] on the given circuit. However most existing SSTA methods are for the circuits clocked by edge-triggered flip-flops (FFs), while the binning process is usually used in latched circuits. SSTA for latched circuits is a much more complex problem because the output time of a latch depends on its input time and all the timing constraints on feedback cycles should be checked simultaneously. Previous researches on this issue, such as [6], [7], [8], solve the clock-schedule verification problem, that is compute the yield at a given clock period, by iteratively updating the statistical arrival time at the input/output of latches and detecting negative cycles in the circuit. However, they cannot compute the statistical period which includes the yield information for all clock periods, unless their SSTA is repeatedly performed at each possible clock period, which is extremely time-consuming. Besides, these methods suffer, more or less, from the disconvergency of statistical arrival time, because the timing random variable is correlated with itself in different iterations. Furthermore, they require the assumption of Gaussian distributions on process variations, which is not always true in IC fabrication.

In this paper, we formulate the binning optimization problem for transparently-latched circuits with an objective of the benefit function considering both the sales profit and test cost. The problem aims to determine the bin boundaries and their testing order such that the benefit is maximized. We decompose the problem into three sequentially solved sub-problems as the statistical period computation, the sales profit maximization, and the test cost minimization. The main contributions of the paper are as follows.

*Corresponding author. E-mail: xzeng@fudan.edu.cn

ICCAD 2009, Nov. 2 - 5, 2009, San Jose, California, USA

1) To compute the statistical period, a sample-based SSTA approach is developed for latched circuits under process variations of arbitrary distribution, where the gSCM with Sparse Grid technique is applied. The statistical problem is translated into a set of deterministic problems on sample points, therefore the disconvergency of statistical arrival time is naturally avoided.
2) To maximize the sales profit, a greedy algorithm is proposed to find the optimal bin boundaries, where each boundary is iteratively assigned to its optimal position. We formulate a simplified problem and prove the conditional unimodality of its objective function. Then an interval-wise method is presented to find the solution to the problem.
3) To minimize the test cost, an optimal algorithm of $O(n \log n)$ runtime is proposed to generate the optimal testing order on bin boundaries. The problem of testing order determination is converted to the weighted path length minimization of alphabetic tree. The algorithm is based on Hu-Tucker coding [9].

It should be noted that the proposed approaches are not limited to latched designs. If there are FFs, each can be cut off and its input and output treated as PO and PI. It should also be mentioned that the statistical period used as the input of optimization may not only be computed by SSTA. When the chips are fabricated, some silicon testing data can be collected during the process, such as early-stage testing and post-silicon validation. These data may be used to revise the statistical period from SSTA to improve binning optimization.

The remainder of this paper is organized as follows. In Section II, the background and the motivation for binning optimization is given. In Section III, the binning optimization problem with the benefit model considering both the sales profit and the test cost is formulated. In Section IV, a greedy approach is designed to maximize the sales profit and an $O(n \log n)$ optimal algorithm is proposed to minimize the test cost. In Section V, a sample-based SSTA approach for transparently-latched circuits is developed. The efficiency of the proposed SSTA and the benefit improvement by using the proposed optimization algorithm are demonstrated by experiments in Section VI, followed by the conclusion in Section VII.

II. Background and Motivation

In this section, the existing binning optimization research with the objective of maximizing the sales profit is reviewed. Then the background of SSTA for latched circuits is given.

A. Sales Profit and Bin Boundaries

With process variations, both the minimal workable clock period T_{work} and the leakage power consumption P_{work} of a design are random variables. A chip is useless if its period is longer than an upper bound T_{targ}, or its leakage power is larger than the power limit P_{leak}. It was shown in [10] that the leakage power exhibits strong correlation with the chip speed. The main leakage failure (more than 97%) happens in the highest frequency bin. For simplicity, [5] translates leakage bound to a period lower bound T_{leak} based on the approximation of yield as

$$Pr\{P_{work} \leqslant P_{leak}\} \approx Pr\{T_{work} \geqslant T_{leak}\}.$$

The usable chips are those whose periods have $T_{leak} \leqslant T_{work} \leqslant T_{targ}$.

Unlike the traditional *single-price strategy*, where all the chips with periods in $[T_{leak}, T_{targ}]$ are sold at the same price, the *binning process* sorts the chips into multiple speed bins based on their T_{work}, and sells chips in different bins with different prices. Let $F(x)$ and $f(x)$ denote the cumulative distribution function (CDF) and the probability density function (PDF) of T_{work} respectively. Suppose

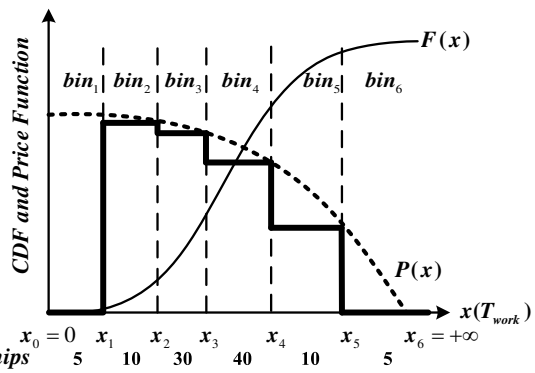


Fig. 1. The distribution function $F(x)$ versus the price function $P(x)$ with six speed bins.

the number of speed bins is n, then the sales profit $Pro(X)$ can be expressed as

$$\begin{aligned} Pro(X) &= \sum_{i=1}^{n} P(x_i) \int_{x_{i-1}}^{x_i} f(x)dx \\ &= \sum_{i=1}^{n} P(x_i)(F(x_i) - F(x_{i-1})) \end{aligned} \quad (1)$$

where $X = [x_0, x_1, x_2, ..., x_n]$ are the bin boundaries of these n bins, and $P(x)$ is a function giving the price of a chip with period x. Note that x_0 and x_n are the bounds due to physical limits, while x_1 and x_{n-1} are set to T_{leak} and T_{targ} respectively. With the four boundaries fixed in advance, $Pro(X)$ is actually a function with $n-3$ variables. Fig. 1 shows a schematic of $F(x)$ versus $P(x)$ with $n = 6$. Since all the chips in a speed bin are sold at one price, the actual sale prices are staircase. The chips in bin_1 and bin_6 have zero price because they cannot be used.

As seen from the equation (1), the sales profit *Pro* is a function of bin boundaries X, which indicates that a good choice of X can improve $Pro(X)$ for the given period distribution and price function. An approach proposed in [5] repeatedly pick one boundary at a time, change it by a small step and accept this change if there is improvement in profit. However, the optimality of the method is not guaranteed and the step size is chosen empirically. If the step is too large, their approach may miss the optimal solution, and if the step is too small, the approach may require too many iterations.

B. Statistical Period Computation and SSTA

To facilitate binning optimization for a latched circuit, $F(x)$ or $f(x)$ of the period T_{work} must be computed. They are usually estimated by SSTA on the circuit.

SSTA on a circuit with FFs can be done on the combinational part of the circuit, with well-developed techniques. However for circuits clocked by level-sensitive latches, the timing problem becomes extremely complex, because a signal can pass transparently through a latch during its enabling period and therefore the output time of a latch is dependent on its input time [11], [12]. Up till now, there is no mature technique capable of finding the statistical period for latched circuits. Recently, several model-based SSTA approaches [6], [7], [8] for latched circuits have been proposed, where the signal arrival times are treated as random variables and propagated through the circuit by repeatedly performing statistical *sum* and *max* operations. These approaches iteratively update the distribution of arrival time at the nodes in the timing graph constructed according to SMO constraint formulation [11], and compute the yield based on statistical negative cycle detection. However, all these techniques are for the clock-schedule verification, which aims to compute the timing yield

on a given clock period. To get $F(x)$ and $f(x)$ defined on the T_{work} interval $[0, \infty]$, the SSTA process should be performed at each possible value of T_{work}, which requires a great deal of time. Furthermore, since the distributions of the statistical arrival time in different iterations are tightly correlated, they may not converge in the SSTA process. Moreover, all these approaches can only deal with the process variations with Gaussian distribution, while the variations may be non-Gaussian.

While many efforts have been made to improve model-based SSTA for latched circuits, there are few approaches proposed on sample-based SSTA. The sample-based SSTA performs DSTA (deterministic STA) on a set of sample points, and fits the distribution of T_{work} by using the deterministic results at all sample points. The sample-based stochastic collocation method (SCM) for FF-clocked circuits is proposed in [13] and [14] where the variations are assumed to be Gaussian distributed. In Section V, we introduce the gSCM [15] to SSTA for latched circuits under process variations of arbitrary distribution.

III. Problem Formulation

Binning optimization problem is to find an optimal bin assignment such that the benefit of a design is maximized. However, only increasing the sales value is not enough since the test cost in the binning process must also be considered.

A frequency test aims to check whether the chips can work on a given frequency. In single-price strategy each chip only requires two tests at T_{leak} and T_{targ} respectively. While in binning process a chip requires several tests at different frequencies to identify which bin it belongs to. During the speed binning process with n bins, there are $n-1$ bin boundaries to be tested. Therefore the number of tests required for each chip ranged from 1 to $n-1$.

We have observed that the testing order of bin boundaries has a critical impact on the test cost. If a chip has passed a frequency test, then it only requires the tests at the higher frequencies, and the tests at the lower frequencies can be skipped; a chip failed a frequency test can skipped the tests at higher frequencies. For example, consider the binning process in Fig. 1, five frequencies $(x_1, x_2, x_3, x_4, x_5)$ need to be tested. Suppose the numbers of chips in the bins are 5, 10, 30, 40, 10, 5, as shown in the figure. If the testing order is $[x_1, x_2, x_3, x_4, x_5]$, all the chips are tested on x_1, and then the chips except those in interval $[x_0, x_1]$ are tested on x_2, and so on. The numbers of tests required for each chip in the five bins are 1, 2, 3, 4, 5, 5, and the total number of tests is $1 \times 5 + 2 \times 10 + 3 \times 30 + 4 \times 40 + 5 \times 10 + 5 \times 5 = 350$. If the order is changed to $[x_3, x_2, x_4, x_1, x_5]$, after the first test at x_3 the chips are partitioned into two parts. The chips in $[x_0, x_3]$ skip the test at x_4 and the chips in $[x_3, x_6]$ skip the test at x_2. The numbers of tests required for each chip are 3, 3, 2, 2, 3, 3, and the the total number of tests is only $3 \times 5 + 3 \times 10 + 2 \times 30 + 2 \times 40 + 3 \times 10 + 3 \times 5 = 230$. Assume that the cost of testing one chip at one frequency is constant, then the latter testing order is obviously better than the former. Therefore the testing order of boundaries determines the number of tests required for each chip, and thus determines the total test cost.

In this paper, the *testing rank*, which is a more general concept than the testing order, is used to describe the priority of bin boundaries in frequency test. The testing rank $R(X^-)$ represents the testing priority defined on the bin boundary set X^-, where $X^- = (x_1, x_2, ..., x_{n-1})$ denote the boundaries to be tested. The testing order of boundaries can be constructed from their ranks by the following rules.

1) The first boundary to test is the boundary with rank 0.
2) The boundary with larger rank should be tested after those with smaller ones.
3) The boundaries with the same rank can be tested in any order.

The testing orders from the same testing rank lead to the same cost. In Fig. 1, for example, if the rank $R = [2, 1, 0, 1, 2]$, the testing order can be $[x_3, x_2, x_4, x_1, x_5]$ or $[x_3, x_4, x_2, x_5, x_1]$ with the same cost.

Let $\vec{\xi}$ denote the process variables, the problem of Binning Optimization for transparently-Latched Circuits considering process variations (BOLC for short) can be formulated as follows.

Problem BOLC : Given a latched circuit $C(\vec{\xi})$, a price function $P(x)$ and the number of bins n. Find the optimal bin boundaries X and the testing rank $R(X^-)$ of the boundaries, such that the *Benefit*, defined as follows, is maximized,

$$\textit{Benefit}(X, R) \triangleq \textit{Pro}(X) - \textit{Ct}(R)$$

where $\textit{Pro}(X)$ is the sales profit and $\textit{Ct}(R)$ is the test cost.

Since the objective function $\textit{Benefit}(X, R)$ is affected by both the boundaries X and the testing order R, the original optimization problem is very complex. A reasonable assumption can be made that the test cost is much smaller than the price of a chip. Then the problem can be simplified by dividing the objective into two parts, that is maximizing $\textit{Pro}(X)$ and minimizing $\textit{Ct}(R)$. Therefore *BOLC* can be decomposed into three sub-problems as follows.

- *Problem BOLC.0* : Given $\vec{\xi}$ and $C(\vec{\xi})$, estimate the distribution function $F(x)$ of T_{work} by SSTA.
- *Problem BOLC.1* : Given $F(x)$, $P(x)$ and n, decide the optimal bin boundaries X, such that the sales profit $\textit{Pro}(X)$ is maximized.
- *Problem BOLC.2* : Given $F(x)$ and X, find the optimal testing rank $R(X^-)$, such that the test cost $\textit{Ct}(R)$ is minimized.

Although the binning optimization can be done only after the computation of the period distribution by SSTA, we will present the binning optimization first and then discuss SSTA later in the following sections.

IV. Binning Optimization Algorithm

In this section, a greedy approach is designed to solve *BOLC.1* and then an optimal algorithm of $O(n \log n)$ runtime is proposed to solve *BOLC.2*.

A. Greedy Approach for BOLC.1

Consider a simplified problem *BOLC.1R* derived from problem *BOLC.1* as follows.

Problem BOLC.1R : Given $F(x)$, $P(x)$ defined on interval $[a, b]$, find the optimal $x \in [a, b]$, such that the profit $\textit{Pro}(x) = P(x)(F(x) - F(a)) + P(b)(F(b) - F(x))$ is maximized.

Assume that $F(x)$ and $P(x)$ are second order differentiable. $F(x)$, $P(x)$ and $f(x)$ are non-negative everywhere. A discussion on the unimodality of the objective function $\textit{Pro}(x)$ is given as Theorem 1 based on the following Lemmas.

Lemma *1:* If $f(x)$ is decreasing and $P(x)$ is strictly concave and decreasing, then $\textit{Pro}(x)$ is strictly concave on $[a, b]$.

Proof: Since $F(x)$ and $P(x)$ are second order differentiable, $\textit{Pro}(x)$ which is the combination of them is also second order differentiable.

$$\begin{aligned}\textit{Pro}''(x) =& P(x)''(F(x) - F(a)) \\ &+ f'(x)(P(x) - P(b)) + 2f(x)P'(x)\end{aligned}$$

And because

$$\begin{gathered}P(x) \text{ is concave} \Rightarrow P''(x) < 0 \\ P(x) \text{ is decreasing} \Rightarrow P'(x) < 0 \\ f(x) \text{ is decreasing} \Rightarrow f'(x) < 0 \\ a < x < b \Rightarrow F(x) > F(a),\ P(x) > P(b)\end{gathered}$$

then $\textit{Pro}''(x)$ is negative, and therefore $\textit{Pro}(x)$ is strictly concave. ■

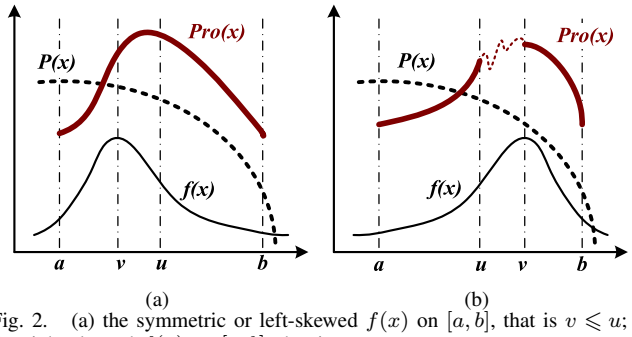


Fig. 2. (a) the symmetric or left-skewed $f(x)$ on $[a, b]$, that is $v \leqslant u$; (b) the right-skewed $f(x)$ on $[a, b]$, that is $v > u$

Lemma *2:* If $\exists m \leqslant (a+b)/2$ such that $f(x)$ is strictly increasing function on $[a, m]$, and $P(x)$ is strictly concave and decreasing, then $Pro(x)$ is strictly increasing on $[a, m]$.

Proof: $Pro(x) = (P(x)-P(b))(F(x)-F(a))+P(b)(F(b)-F(a))$. Since $f(x)$ is strictly increasing on $[a, m]$, then $F(x)$ is strictly convex on it, therefore $\forall c, d \in (a, m]$ where $c > d$,

$$\frac{F(d)-F(a)}{F(c)-F(a)} < \frac{d-a}{c-a} \tag{2}$$

Since $P(x)$ is strictly concave, then

$$\frac{P(c)-P(b)}{P(d)-P(b)} > \frac{b-c}{b-d} \tag{3}$$

Because $d < c \leqslant (a+b)/2$, therefore

$$\begin{aligned} &((a+b)/2-d)^2 > ((a+b)/2-c)^2 \\ &\Rightarrow (a+b)c-c^2 > (a+b)d-d^2 \\ &\Rightarrow (b-c)(c-a) > (b-d)(d-a) \\ &\Rightarrow \frac{b-c}{b-d} > \frac{d-a}{c-a} \end{aligned} \tag{4}$$

Combine (2), (3) and (4), we have

$$\begin{aligned} &\frac{P(c)-P(b)}{P(d)-P(b)} > \frac{F(d)-F(a)}{F(c)-F(a)} \\ &\Rightarrow Pro(c) > Pro(d) \end{aligned}$$

Therefore, $Pro(x)$ is strictly increasing on $[a, m]$. ■

Combine Lemma 1 and Lemma 2 and the fact that $\forall x \in (a, b)$, $Pro(x) > Pro(a) = Pro(b)$, we have

Theorem *1:* Let $u \triangleq (a+b)/2$ and $v \triangleq argmax(f(x))$ where $x \in [a, b]$. If $f(x)$ is unimodal and $P(x)$ is strictly concave and decreasing, the following properties holds:

1) If $v \leqslant u$, then $Pro(x)$ is unimodal on $[a, b]$;
2) If $v > u$, then $Pro(x)$ is strictly increasing on $[a, u]$ and strictly concave on $[v, b]$.

Based on the conditional unimodality of $Pro(x)$, an interval-wise method is proposed as follows to solve *BOLC.1R*.

1) If $f(x)$ is symmetric or left-skewed on $[a, b]$, the first property, illustrated in Fig. 2(a), indicates that there is only one zero of $Pro(x)$ on $[a, b]$ and it is also the maximum. Solving $Pro'(x) = 0$ on $[a, b]$ by the iterative method (for example, Newton-Raphson method) with the initial guess set at any value on $[a, b]$ gives the optimal solution.
2) If $f(x)$ is right-skewed on $[a, b]$, the second property, illustrated in Fig. 2(b), indicates that the optimal solution may appears on $[u, v]$ or $[v, b]$. On $[v, b]$, $Pro(x)$ is strictly concave and unimodal, thus its zero can be solved as in 1). On $[u, v]$ the unimodality is not clear. The interval $[u, v]$ can be divided into several small fractions and the function $Pro'(x) = 0$ is solved in each fraction. The optimal solution is chosen as the one with maximal profit among all the zeros on $[v, b]$ and $[u, v]$.

If X is the optimal bin boundaries in the original problem *BOLC.1*, then each boundary x_i $(i = 2, ..., n-2)$ must be the optimal solution to the simplified problem *BOLC.1R* for interval $[x_{i-1}, x_{i+1}]$. We design a greedy algorithm to solve *BOLC.1*, where each bin boundary is iteratively assigned to its optimal position by solving the corresponding *BOLC.1R*. Algorithm 1 gives the pseudo-code.

Algorithm 1 OPT_LOCATION(X_{init}, $F(x)$, $P(x)$, n)

```
X := X_init, HasGain := true;
while HasGain do
    HasGain := false;
    for i = 2 to n − 2 do
        a := x_{i−1}, b := x_{i+1};
        x_new := the solution to BOLC.1R on [a, b];
        if x_i ≠ x_new then
            x_i := x_new;
            HasGain := true
        end if
    end for
end while
```

In Algorithm 1, X_{init} denotes the initial location of bins. The algorithm will find local optimal near X_{init}. Furthermore, it is found in our experimental results that the algorithm always converges to the same solution from different X_{init}, indicating that there may be only one local optimal solution to *BOLC.1*.

B. Optimal Algorithm for BOLC.2

For the binning process with n bins, there are $n-1$ bin boundaries to be tested, which are $X^- = (x_1, x_2, ..., x_{n-1})$. Let $B_{i,j}$ denotes the bins between the boundary x_i and x_j and $N_{i,j}$ denotes the percentage of chips in $B_{i,j}$. Then for the i-th bin,

$$N_{i-1,i} = F(x_i) - F(x_{i-1})$$

Without loss of generality, the test cost per frequency per chip is set to 1. Then the test cost *Ct* in the problem *BOLC.2* equals to the total number of tests as follows,

$$Ct = \sum_{i=1}^{n} h_{i-1,i} N_{i-1,i} \tag{5}$$

where $h_{i-1,i}$ is the number of tests required for each chip in $B_{i-1,i}$.

By constructing an alphabetic binary tree [9] according to the testing rank, the objective function (5) is actually *the weighted path length* of the tree. An example of the alphabetic tree is shown in Fig. 3 with $n = 6$ and $R(X^-) = [1, 3, 2, 0, 1]$. The tree is composed of two kinds of nodes, the leaf nodes (squares) and the internal nodes (circles). The i-th leaf node represents $B_{i-1,i}$ with weight $N_{i-1,i}$. All the leaf nodes are restricted to alphabetic order as $(B_{0,1}, B_{1,2}, ..., B_{n-1,n})$. The internal node $B_{i,j}$ with two children nodes represents a *super-bin* consisting of all bins between x_i and x_j. The node $B_{0,n}$ is called the *root*. Each internal node also represents the bin boundary separating its children. For example, $B_{0,6}$ represents boundary x_4. In a binning process with n bins, the tree has n leaf nodes and $n-1$ internal nodes.

There is a unique path from the root to every node, and the path length (number of arcs) of a node is called the *rank*. The rank of the root is zero. A chip in $B_{i,j}$ takes the frequency test at every boundary in the path from the root to $B_{i,j}$. Therefore, the rank of $B_{i-1,i}$ gives the number of tests required for each chip in the i-th bin, that is $h_{i-1,i}$ in (5). The rank of an internal node represents the testing priority of the corresponding bin boundary.

Since *Ct* in (5) is equivalent to the weighted path length of the binary tree, *BOLC.2* aims to construct an optimal tree with minimal weighted path length. This problem can be solved by an optimal

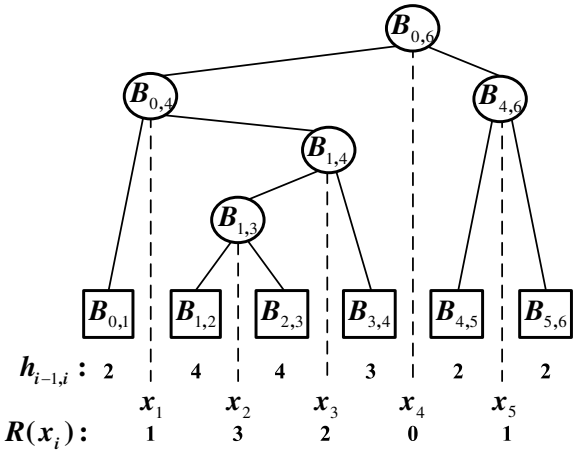


Fig. 3. The alphabetic tree with $n = 6$ and $R(X^-) = [1, 3, 2, 0, 1]$

algorithm of $O(nlogn)$ runtime based on Hu-Tucker coding [9]. The algorithm is composed of the following three phases.

- *Phase 1, Bin Combination.* Given an initial sequence of leaf nodes as $(B_{0,1}, B_{1,2}, ..., B_{n-1,n})$, repeatedly choose two successive nodes $B_{i,j}$ and $B_{k,l}$ such that the sum of their weights $N_{i,j}+N_{k,l}$ is the minimal among all the possible pairs, combine them into a super-bin $B_{i,l}$ with weight $N_{i,l} = N_{i,j}+N_{k,l}$, delete $B_{k,l}$ and replace $B_{i,j}$ by $B_{i,l}$. Two nodes are called successive in the sequence if there is no nodes or there are only internal nodes between them. Phase 1 ends when all the leaf nodes are combined into a single root node.
- *Phase 2, Bin Rank Assignment.* Mark the rank of the root as zero, and then break internal nodes in the reverse order of Phase 1, assign the rank to all the nodes in the tree. If a node has rank r, then the two nodes as its sons has rank $r + 1$.
- *Phase 3, Boundary Rank Computation.* Delete all the internal nodes and their ranks in Phase 1 and 2. Create new super-bins by repeatedly combining $B_{i,j}$ and $B_{j,k}$ into $B_{i,k}$ if 1) there is no node between $B_{i,j}$ and $B_{j,k}$; 2) the ranks of $B_{i,j}$ and $B_{j,k}$ are identical and are the lowest ranks among the remaining ranks; 3) i and k is the minimal among all possible pairs that satisfy 1) and 2). The rank of $B_{i,k}$ is the rank of $B_{i,j}$ minus one.

When the three phases ends, an optimal tree with the minimal weighted path length is constructed. The ranks of all the internal nodes updated in Phase 3 are the optimal solution $R(X^-)$ to *BOLC.2*.

V. Sample-Based Statistical Period Computation for Latched Circuits

In this section, a sample-based SSTA approach is proposed for statistical period computation in latched circuits. The gSCM [15], [16] based on generalized Polynomial Chaos (gPC) and generalized Sparse Grid quadrature is applied to deal with the process variations of arbitrary distribution. The period computation problem on each sample point is converted to a minimal cycle ratio problem in the constraint graph.

A. The gSCM for SSTA

Let $\vec{\xi}$ denotes a set of independent random variables of arbitrary distribution which can be obtained after the PCA (Principal Component Analysis) [17] or ICA (Independent Component Analysis) [18] process for the correlated process parameters. Both gate/interconnect delays and arrival times can be approximated by gPC [15], [16] given as,

$$D(\vec{\xi}) \approx \hat{D}(\vec{\xi}) \triangleq \sum_{i_1+...+i_N=0}^{M} d_{i_1,...i_N} H_N^{i_1,...,i_N}(\vec{\xi}) \quad (6)$$

where D is the exact value and $\hat{D}$ is the approximated value. N is the number of random process variables, M is the highest order of polynomial, $H_N^{i_1,...,i_N}(\vec{\xi})$ denotes the N-dimensional gPC and $(i_1 + ... + i_N)$ denotes the order of gPC. The coefficients $d_{i_1,...i_N}$ are estimated by equating delays D and the corresponding polynomial chaos (6) at a set of collocation points in the parameter space.

Based on the delay model in (6), the statistical minimal clock period can also be approximated as a gPC expansion of $\vec{\xi}$ as (7).

$$T_{work}(\vec{\xi}) \approx \hat{T}_{work}(\vec{\xi}) \triangleq \sum_{i_1+...+i_N=0}^{M} t_{i_1,...i_N} H_N^{i_1,...,i_N}(\vec{\xi}) \quad (7)$$

Given a circuit topology and the statistical delay $D(\vec{\xi})$ of each device in the circuit, the proposed SSTA approach finds the unknown coefficients $t_{i_1,...i_N}$ in (7) by the following three steps, and then the statistical period can be obtained.

Step 1: Generate a set of collocation points for $\vec{\xi}$ as $\{\vec{\xi}_k | k = 1, 2, ..., P\}$, where P is the number of collocation points.

Step 2: Compute the minimal clock period $T_{work}(\vec{\xi}_k)$ at each collocation point $\vec{\xi}_k$, which is a DSTA problem.

Step 3: Calculate the unknown coefficients $t_{i_1,...i_N}$ in (7). The gSCM seeks for the optimal solution to minimize the error between $T_{work}(\vec{\xi})$ and $\hat{T}_{work}(\vec{\xi})$ by Galerkin Approach as setting,

$$\langle T_{work}(\vec{\xi}) - \hat{T}_{work}(\vec{\xi}), H_N^{i_1,...,i_N}(\vec{\xi})\rangle = 0$$

for all $i_1 + ... + i_N = 0, 1, ..., M$. In virtue of the orthogonality of gPC, the unknown coefficients can be computed as,

$$t_{i_1,...i_N} = \langle T_{work}(\vec{\xi}), H_N^{i_1,...,i_N}(\vec{\xi})\rangle \quad (8)$$

Equation (8) is a multi-dimensional integral which can be solved with numerical quadrature using the value of integrand taken on a set of collocation points as,

$$t_{i_1,...i_N} = \sum_{k=1}^{P} w_k T_{work}(\vec{\xi}_k) H_N^{i_1,...,i_N}(\vec{\xi}_k)$$

where $\vec{\xi}_k$ is the k-th collocation point, w_k is the corresponding weight and $T_{work}(\vec{\xi}_k)$ is the accurate value of T_{work} at the collocation point $\vec{\xi}_k$, which is solved in *Step 2*.

The CDF and PDF of T_{work} can be computed after the three steps. The proposed SSTA for latched circuits computes the yield at all periods at one time, while the traditional approaches fails because they only compute the yield at a given period.

B. Sparse Grid Technique

Sparse Grid is a remarkable sampling technique developed for gSCM. Compared with the direct tensor product scheme [19], Sparse Grid can significantly reduce the number of collocation points for multi-dimensional integration in (8) [15].

Let $\Theta_1^{i_j}$ and $W_1^{i_j}$ denote the set of collocation points and the weights for jth-dimensional i_j-level accuracy Gaussian quadrature rule using the roots of $H_1^{i_j}(\xi_j)$. The set of collocation points generated by Sparse Grid for a d-dimensional quadrature of k-level accuracy is a linear combination of the tensor product of $\Theta_1^{i_j}$, as given in (9),

$$\Theta_d^k = \bigcup_{k+1 \le |\vec{i}| \le d+k} (\Theta_1^{i_1} \times ... \times \Theta_1^{i_d}) \quad (9)$$

where $|\vec{i}| = i_1 + ... + i_d$. The weight corresponding to the collocation point $(\vec{\xi}^{i_1}_{j_{i_1}}, ..., \vec{\xi}^{i_d}_{j_{i_d}}) \in \Theta_d^k$ is expressed as (10)

$$w^{i_1,...i_d}_{j_{i_1},...,j_{i_d}} = (-1)^{d+k-|\vec{i}|} \binom{d-1}{d+k-|\vec{i}|} (w^{i_1}_{j_{i_1}} \cdots w^{i_d}_{j_{i_d}}) \quad (10)$$

It has been proved in [20] that Sparse Grid is exact for all d-variables polynomials of order at most $(2k+1)$. And the amount of collocation points for Sparse Grid is given as,

$$N_{sg} = dim(\Theta_d^k) \sim \frac{2^k}{k!} d^k \sim 2^k dim(\pi_d^k), \qquad d \gg 1$$

where π_d^k denotes the space of all d-dimensional gPC of order at most k. Compared with the direct tensor product scheme where the amount of sample points is $(k+1)^d$, Sparse Grid technique avoids the exponential growth of computation cost with respect to the dimensionality [20].

C. Minimal Period Computation on Each Sample Point

On each sample point $\vec{\xi}_k$, a DSTA problem needs to compute $T_{work}(\vec{\xi}_k)$. The SMO [11] formulation of constraints is widely used in DSTA for latched circuit, where the circuit is abstracted as a directed graph $G = (V, E)$ called *Timing Graph*. The node set V represent the primary inputs/outputs of the circuit and the output pins of all elements (gates and latches). The edge set E represent all the timing arcs between pins in gates and latches.

Let A_i (a_i) represents the latest (earliest) signal arrival time at pin i, and Δ_{ji} (δ_{ji}) the maximal (minimal) delay of the timing arc (j, i). For simplicity, it is assumed that all latches are controlled by the same clock phase with period Tc and active interval $Tp = 0.5Tc$. Note that the proposed algorithm can be extended to any clock model with multi-phase and non-50% duty cycle. The setup time and the hold time of the latches are S and H respectively. The SMO constraints formulation can be described as follows,
If (j, i) is a gate,

$$A_i = \max_{(j,i)\in E}(A_j + \Delta_{ji}) \tag{11}$$

$$a_i = \min_{(j,i)\in E}(a_j + \delta_{ji}) \tag{12}$$

If (j, i) is a latch,

$$A_i = \max(A_j + \Delta_{ji}, Tc - Tp) - E_{ji} \tag{13}$$

$$a_i = Tc - Tp - E_{ji} \tag{14}$$

$$A_j \leqslant Tc - S \tag{15}$$

$$a_j \geqslant H \tag{16}$$

where $E_{ji} = Tc$ because of the assumption of single clock phase. The earliest constraint of a latch in (14) follows a conservative formulation instead of the aggressive one as $a_i = \max(a_j + \delta_{ji}, Tc - Tp) - E_{ji}$. The reason is that the aggressive formulation might be incorrect in the situations such as a latch driven by a qualified clock signal or the circuits where the clock is permitted to be stopped to save power [21].

The formulation consists of two sets of constraints, the setup time constraints (11)(13)(15) and the hold time constraints (12)(14)(16). For each set of constraints, a constraint graph can be constructed, and the minimal period problem can be converted to the minimal cycle ratio problem in the corresponding graph.

Take the setup time constraints for example, (11)(13)(15) can be translated into the following set of inequalities. Let $\tilde{A}_i$ denote $-A_i$ for all elements in circuit,
$\forall$ (j, i) is a gate

$$\tilde{A}_i \leqslant \tilde{A}_j + (-\Delta_{ji}) + 0 \cdot Tc$$

$\forall$ (j, i) is a latch

$$\tilde{A}_i \leqslant \tilde{A}_j + (-\Delta_{ji}) + 1 \cdot Tc$$
$$\tilde{A}_i \leqslant 0 + 0 + 0.5 \cdot Tc$$
$$0 \leqslant \tilde{A}_j + (-S) + 1 \cdot Tc$$

The new graph $G_s = (V, E_s, w_s, r_s)$ called *Setup Constraint Graph* can be established based on the above inequalities, where the node set V are the same as that of G, and the edge set E_s are constructed by the following steps. Let w_s and r_s denote the cost function and the gain function defined on edges E_s.

1) Add a source node with arrival time $\tilde{A}_0 = 0$.
2) For each inequality $\tilde{A}_q \leqslant \tilde{A}_p + w + r \cdot Tc$, add an edge from node p to q. The 0 arrival time corresponds to the source node.
3) For the new edge, attach its cost $w_s(p, q)$ with w and its gain $r_s(p, q)$ with r.

For a cycle C in this graph, the cycle ratio $Ro(C)$ is defined as the ratio of the sum of edge cost to the sum of edge gain, as described in (17).

$$Ro(C) = \frac{w(C)}{r(C)} = \frac{\sum_{(p,q)\in C} w(p, q)}{\sum_{(p,q)\in C} r(p, q)} \tag{17}$$

On the other hand, sum up all the inequalities on the cycle C, it can be derived that,

$$\begin{aligned} 0 &\leqslant \textstyle\sum_{(p,q)\in C} w(p, q) + Tc \cdot \sum_{(p,q)\in C} r(p, q) \\ &\Rightarrow Tc \geqslant -\frac{\sum_{(p,q)\in C} w(p, q)}{\sum_{(p,q)\in C} r(p, q)} = -Ro(C) \end{aligned} \tag{18}$$

The circuit pass the timing constraints if the equation (18) is satisfied for all the cycles, as described in (19).

$$\begin{aligned} Tc &\geqslant max(-Ro(C)) = -min(Ro(C)), \qquad C \in G_s \\ &\Rightarrow T_{setup} = min(Tc) = -min(Ro(C)) \end{aligned} \tag{19}$$

In this way, the minimal period problem due to set up constraints is converted to the minimal cycle ratio problem of G_s, which can be solved by various algorithms based on graph theory [22]. Howard's algorithm [23] is applied in this paper because of its practical efficiency.

Similarly, the *Hold Constraint Graph*, constructed from the inequalities translated from (12)(14)(16), gives an upper bound of clock period as $Tc \leqslant T_{hold}$. If $T_{setup} \leqslant T_{hold}$, the minimal period of the circuit is T_{setup}, while if $T_{setup} > T_{hold}$, the design is considered as failed because no period is feasible under both the setup and hold constraints.

VI. Experimental Results

The proposed binning optimization algorithm and the sample-based SSTA approach for transparently-latched circuits have been implemented and validated on all the 30 ISCAS'89 sequential benchmarks with 65nm process technology. The flip-flops in the circuits are replaced by level-sensitive latches, which are clocked by a single-phase clock with 50% duty cycle. All experiments are run on a 3.0GHz Linux serve.

Table. I gives the relative error of the statistical period computed by the proposed SSTA approach compared with 10000 Monte Carlo (MC) simulations. The speedup of the proposed approach is also shown in this table. The quadratic delay model used is based on six independent random variables derived from PCA or ICA process (that is, N=6 and M=2 in (6)). To verify the accuracy of the proposed SSTA approach under process variations of arbitrary distribution, three kinds of distributions, Gaussian, Uniform, and Rayleigh [24], with variances set to 10% of their means, are used on the six random variables in each circuits. Since the non-linear delay models are used and the input distribution may not be Gaussian, $F(x)$ may not be Gaussian. Therefore, in addition to the mean and variance, the relative error *ferr* of $F(x)$, defined as (20), is also shown in Table. I,

$$ferr = \frac{\sqrt{\sum_{i=1}^{S}(F(T_i) - F^{MC}(T_i))^2}}{\sqrt{\sum_{i=1}^{S} F^{MC}(T_i)^2}} \tag{20}$$

where S is the number of compared points and T_i is the period of the i-th point. $F(\cdot)$ denotes the CDF function computed by our approach and $F^{MC}(\cdot)$ denotes that by MC simulations. Due to space limit, Table. I shows 10 of 30 circuits, including the case with worst and best accuracy, and the average result of all the 30 circuits. It can be seen from the table that our approach fits very well in all distribution types with an average *ferr* of 0.70% and an average speedup of 110X, compared with the MC simulations.

Table. II shows the percentages of sales profit improvement and test cost reduction obtained by performing the proposed binning optimization. The statistical periods used in the optimization are computed in SSTA, with the random variables assuming Gaussian distributions. The experiments have tested three bin numbers as 5, 6 and 7, and three different price functions as linear, quadratic, and cubic functions. As stated earlier, all the three functions are decreasing and concave, which are defined as follows.

$$\begin{aligned} \textit{Linear}: \quad & P(x) = a_1 x + a_2 \\ \textit{Quadratic}: \quad & P(x) = a_3 (x - a_4)^2 + a_5 \\ \textit{Cubic}: \quad & P(x) = a_6 x^3 + a_7 \end{aligned}$$

where the coefficients $a_1, ..., a_7$ are calculated in such a way as in [5] to make constant the ratio of the prices at the highest and lowest permissible periods, that is, $\textit{Ratio} = P(T_{leak})/P(T_{targ}) = 5$. Fig. 4(a) plots the three normalized price functions versus T_{work} for the benchmark s13207. The two fixed boundaries are set as $T_{leak} = \mu - 3\sigma$ and $T_{targ} = \mu + 3\sigma$, where μ and σ are the mean and variance of T_{work}, respectively. The *Pro+* in Table. II denotes the improvement of profit by adjusting the bin boundaries to X_{init}, while the initial boundaries X_{init} are set to make each period bin have the same yield. The improvement percentage is defined as,

$$\textit{Pro+} = (\textit{Pro}(X_{opt}) - \textit{Pro}(X_{init}))/\textit{Pro}(X_{init})$$

The *Ct−* denotes the test cost reduction by assigning the testing rank of boundaries as the optimal rank R_{opt} compared with a binary-search-like rank R_{bny}, which starts from the middle boundary and then hierarchically moves towards the low and fast directions. The R_{bny} used here is defined as,

$$\begin{aligned} n = 5: \quad & R_{bny} = [2, 1, 0, 1] \\ n = 6: \quad & R_{bny} = [2, 1, 0, 2, 1] \\ n = 7: \quad & R_{bny} = [2, 1, 2, 0, 2, 1] \end{aligned}$$

The reduction percentage is defined as,

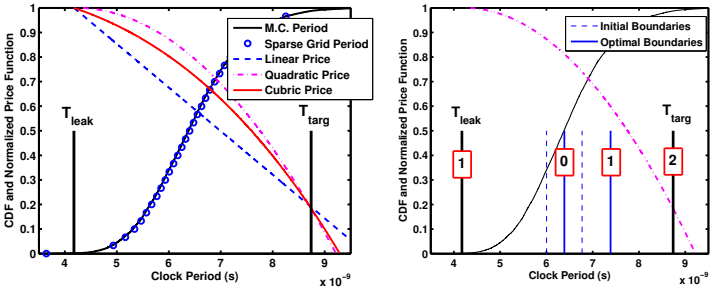
$$\textit{Ct−} = (\textit{Ct}(R_{bny}) - \textit{Ct}(R_{opt})/\textit{Ct}(R_{bny})$$

In average 6.69% profit improvement and 14.00% cost reduction are shown in Table. II, which is a considerable benefit in chip sales and frequency test. It can be seen that more improvement on sales profit can be obtained with quadratic and cubic price function than linear price function. It is because the nonlinear price functions widen the price gap of chips with different T_{work} and raise the importance of bin location assignment. When the number of bins changes from 5 to 6 and 7, the profit improvement is slightly decreasing. The reason is that the increasing number of bins leads to dense binning, where the boundaries cannot change greatly and may have less impact on the profit.

Fig. 4(b) illustrates an example of binning optimization for the benchmark s13207 with 5 bins and the quadratic price function. The initial and optimal boundaries are shown as the dashed lines and the solid line respectively. The optimal testing rank of boundaries is labeled on the boundaries. With the optimal rank $[1, 0, 1, 2]$, the number of tests per chip in each bin is $h = [2, 2, 2, 3, 3]$ and the average number of tests for all chips is $Ct = \sum_{i=1}^{5}(F(X_i) - F(X_{i-1}))L(i) = 2.12$ while another rank $[2, 1, 0, 1]$ leads to $h = [3, 3, 2, 2, 2]$ and $Ct = 2.57$. The test cost of the optimal testing order is reduced by 17.34%.

TABLE III
COMPARISON OF OUR GREEDY ALGORITHM AND [5] ON PROFIT IMPROVEMENT PERCENTAGE AND RUNTIME COST

Circuit	Step=$(x_{n-1} - x_1)/10$				Step=$(x_{n-1} - x_1)/100$			
	n=6		n=7		n=6		n=7	
	Δ*Pro+*(%)	*SpUp*	Δ*Pro+*(%)	*SpUp*	Δ*Pro+*(%)	*SpUp*	Δ*Pro+*(%)	*SpUp*
s298	1.07	3.63X	1.21	3.05X	0.02	14.26X	0.02	14.26X
s386	0.71	3.29X	0.87	4.74X	0.08	15.63X	0.08	15.63X
s510	1.11	5.68X	1.14	6.27X	0.09	24.71X	0.09	24.71X
s953	1.40	3.42X	1.42	4.96X	0.09	12.59X	0.09	12.59X
s1196	1.24	4.25X	1.35	3.85X	0.02	20.27X	0.02	20.27X
s1238	1.25	3.04X	1.10	4.90X	0.06	11.32X	0.06	11.32X
s5378	1.40	3.88X	1.25	3.22X	0.08	22.05X	0.08	22.05X
s9234	1.13	3.51X	1.12	4.13X	0.03	15.15X	0.03	15.15X
s13207	1.12	4.25X	1.11	4.19X	0.11	20.00X	0.11	20.00X
s35932	1.16	5.06X	1.07	5.19X	0.03	21.24X	0.03	21.24X
AVERAGE	1.14	3.76X	1.15	4.53X	0.06	15.81X	0.06	15.81X



(a) (b)

Fig. 4. (a) Three normalized price functions versus CDF function of the benchmark s13207; (b) Binning optimization of s13207

Table. III compares the sales profit improvement and runtime cost of our Algorithm 1 and the algorithm in [5]. In the table, $\Delta\textit{Pro+} = \textit{Pro+}(\textit{our}) - \textit{Pro+}([5])$ and *SpUp* is the speedup of our algorithm compared with [5]. The step size in [5] is set as $1/10$ and $1/100$ of the interval size $(x_{n-1} - x_1)$. The step size is chosen totally empirically in their algorithm. Too large step $(1/10)$ may miss the optimal solution, while too small step $(1/100)$ will greatly slow down the algorithm. The statistics show that no matter how long the step is set, our algorithm can always achieve a better profit improvement than [5] while the runtime cost is much less than it.

VII. CONCLUSIONS

Binning process is widely used by manufacturers to improve benefit of fabricated chips, especially for transparently-latched high-performance designs. This paper formulated binning optimization as a problem to maximize the benefit from the chips, considering both sale prices and the binning test cost. The problem has been decomposed into three sub-problems and solved sequentially. First, a sample-based SSTA approach based on gSCM with Sparse Grid technique is developed to compute the clock period distribution for transparently latched circuits. On each sample point, the minimal period problem is converted to the corresponding minimal cycle ratio problem in the constraint graph. Second, in order to maximize the sales profit, a greedy algorithm is proposed to adjust each bin boundaries to its optimal position. Then, the optimal testing order of the boundaries is generated by an alphabetic-tree based algorithm of $O(n \log n)$ runtime to minimize the test cost.

Tested by all the sequential circuits in ISCAS'89 benchmark with 65-nm technology, the proposed optimization algorithm achieves 6.69% profit improvement and 14.00% cost reduction on average.

TABLE I

COMPARISON OF ACCURACY AND EFFICIENCY WITH THE MC SIMULATIONS

Circuit	Gaussian				Uniform				Rayleigh			
	Related Error (%)			Speed	Related Error (%)			Speed	Related Error (%)			Speed
	mean	var	*ferr*	Up	mean	var	*ferr*	Up	mean	var	*ferr*	Up
s298	0.38	1.00	1.23	110 X	0.08	1.05	0.31	94 X	0.16	0.98	0.64	86 X
s386 (worst)	1.28	7.22	5.57	88 X	1.05	5.43	4.76	96 X	1.35	6.06	6.29	93 X
s510	0.38	0.33	1.21	104 X	0.40	0.91	1.16	99 X	0.16	0.02	0.52	95 X
s953	0.02	0.15	0.05	107 X	0.03	0.09	0.07	106 X	0.00	0.01	0.01	102 X
s1196	0.02	0.23	0.09	100 X	0.01	0.09	0.07	101 X	0.02	0.19	0.10	97 X
s1238	0.53	2.10	3.00	102 X	0.17	1.35	1.14	101 X	0.04	1.44	0.49	103 X
s5378 (best)	0.01	0.16	0.02	127 X	0.01	0.08	0.01	124 X	0.01	0.10	0.02	129 X
s9234	0.03	0.28	0.08	135 X	0.03	0.26	0.09	117 X	0.02	0.17	0.08	122 X
s13207	0.19	0.97	0.93	177 X	0.06	0.11	0.29	145 X	0.20	0.82	0.98	138 X
s35932	0.02	0.10	0.06	238 X	0.04	0.14	0.13	162 X	0.00	0.01	0.01	237 X
AVERAGE (30 circuits)	0.22	0.81	0.85	113 X	0.18	0.88	0.63	108 X	0.15	0.58	0.62	111 X

TABLE II

PERCENTAGE OF PROFIT IMPROVEMENT AND COST REDUCTION (%) OBTAINED BY BINNING OPTIMIZATION

Circuit	T_{leak}	T_{targ}	n=5						n=6						n=7					
	(ns)	(ns)	linear		quadratic		cubic		linear		quadratic		cubic		linear		quadratic		cubic	
			Pro+(%)	*Ct-*(%)	*Pro+*(%)	*Ct-*(%)	*Pro+*(%)	*Ct-*(%)	*Pro+*(%)	*Ct-*(%)	*Pro+*(%)	*Ct-*(%)	*Pro+*(%)	*Ct-*(%)	*Pro+*(%)	*Ct-*(%)	*Pro+*(%)	*Ct-*(%)	*Pro+*(%)	*Ct-*(%)
s298	0.92	2.35	5.80	12.38	10.52	16.82	9.99	16.70	3.89	9.16	7.46	13.17	7.06	12.59	3.17	13.88	5.79	14.42	5.48	14.94
s386	0.93	1.86	6.39	16.39	9.22	18.71	8.75	18.40	4.69	8.59	6.57	12.48	6.25	11.37	3.56	14.12	4.99	14.73	4.78	14.73
s510	1.17	2.92	5.77	14.27	10.62	17.35	10.09	17.35	4.24	11.10	7.73	11.56	7.36	11.56	3.25	13.44	5.86	14.92	5.48	15.68
s953	1.48	4.28	6.72	13.54	13.78	18.94	13.13	18.94	4.79	9.51	9.57	14.26	9.10	12.39	3.60	15.19	7.23	14.45	6.92	14.45
s1196	1.55	4.06	5.60	13.00	10.82	17.50	10.27	17.15	4.22	9.04	7.73	10.83	7.37	10.83	3.17	14.51	6.07	14.63	5.78	14.94
s1238	1.58	4.04	5.81	13.16	10.56	17.82	9.93	15.78	4.20	9.47	7.64	12.40	7.25	12.39	3.15	13.98	5.77	15.01	5.49	15.03
s5378	2.38	6.57	6.29	12.66	12.54	18.60	11.92	17.41	4.51	9.91	8.68	12.63	8.23	12.04	3.58	14.12	6.57	14.30	6.32	15.31
s9234	3.84	10.06	5.66	14.30	10.77	16.55	10.24	16.55	4.14	9.15	7.65	12.47	7.19	11.13	3.24	14.41	5.92	15.32	5.62	15.32
s13207	3.69	9.18	5.70	12.80	9.92	17.34	9.40	16.06	3.99	8.94	7.16	14.14	6.70	11.54	3.03	15.14	5.50	15.18	5.22	15.18
s35932	1.88	4.56	5.16	13.17	9.44	18.02	8.93	18.02	3.78	9.11	6.79	11.64	6.45	11.64	3.00	13.88	5.20	15.99	4.95	16.07
AVERAGE (30 circuits)			5.80	13.77	10.69	17.55	10.16	17.28	4.17	9.18	7.60	12.35	7.21	11.76	3.20	14.42	5.83	14.81	5.53	14.92

The results of SSTA show an error of 0.70% and 110X speedup on average in comparison with the Monte Carlo simulation.

VIII. ACKNOWLEDGEMENT

This research is supported partly by NSFC research project 60976034, 60676018 and 60806013, National Basic Research Program of China under the grant 2005CB321701, National major Science and Technology special project 2008ZX01035-001-06 and 2009ZX02023-4-3 of China during the 11-th five-year plan period, the doctoral program foundation of Ministry of Education of China 200802460068, the International Science and Technology Cooperation program foundation of Shanghai 08510700100, the program for Outstanding Academic Leader of Shanghai, and NSF under CCF-0238484 and CCF-0811270.

REFERENCES

[1] S. Borkar, T. Kamik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De, "Parameter variations and impact on circuits and microarchitecture," in *IEEE Proc. DAC*, June 2003, pp. 338–342.

[2] B. D. Cory, R. Kapur, and B. Underwood, "Speed binning with path delay test in 150-nm technology," *IEEE Design Test Comput.*, vol. 20, no. 5, pp. 41–45, Oct. 2003.

[3] C. Ebeling and B. Lockyear, "On the performance of level-clocked circuits," in *Proc. Advanced Research VLSI*, Mar. 1995, pp. 342–356.

[4] A. Davoodi and A. Srivastava, "Variability driven gate sizing for binning yield optimization," *IEEE Trans. on VLSI*, vol. 16, no. 6, pp. 683–692, June 2008.

[5] A. Datta, S. Bhunia, J. H. Choi, S. Mukhopadhyay, and K. Roy, "Profit aware circuit design under process variations considering speed binning," *IEEE Trans. on VLSI*, vol. 16, no. 7, pp. 806–815, July 2008.

[6] R. Chen and H. Zhou, "Statistical timing verification of transparently latched circuits," *IEEE Trans. on CAD*, vol. 25, no. 9, pp. 1847–1855, Sep. 2006.

[7] M. C.-T. Chao, L.-C. Wang, K.-T. Cheng, and S. Kundu, "Static statistical timing analysis for latch-based pipeline designs," in *IEEE Proc. ICCAD*, Nov. 2004, pp. 468–472.

[8] L. Zhang, J. Tsai, W. Chen, Y. Hu, and C. C.-P. Chen, "Convergence-provable statistical timing analysis with level-sensitive latches and feedback loops," in *IEEE Proc. ASPDAC*, Jan. 2006, pp. 941–946.

[9] T. C. Hu and A. C. Tucker, "Optimal computer search trees and variable-length alphabetical codes," *SIAM Journal on Applied Mathematics*, vol. 21, no. 4, pp. 514–532, Dec. 1971.

[10] R. Rao, A. Devgan, D. Blaauw, and D. Sylvester, "Parametric yield estimation considering leakage variability," in *IEEE Proc. DAC*, June 2004, pp. 442–447.

[11] K. A. Sakallah, T. N. Mudge, and O. a. Olukotun, "Analysis and design of latch-controlled synchronous digital circuits," *IEEE Trans. on CAD*, vol. 11, no. 3, pp. 322–333, Mar. 1992.

[12] T. G. Szymanski and N. Shenoy, "Verifying clock schedules," in *IEEE Proc. ICCAD*, Nov. 1992, pp. 124–131.

[13] S. Bhardwaj, P. Ghanta, and S. Vrudhulat, "A framework for statistical timing analysis using non-linear delay and slew models," in *IEEE Proc. ICCAD*, Nov. 2006, pp. 225–230.

[14] Y. Wang, X. Zeng, J. Tao, H. Zhu, X. Luo, C. Yan, and W. Cai, "Adaptive stochastic collocation method (ascm) for parameterized statistical timing analysis with quadratic delay model," in *IEEE Proc. ISQED*, Mar. 2008, pp. 62–67.

[15] H. Zhu, X. Zeng, X. Luo, and W. Cai, "Generalized stochastic collocation method for variation-aware capacitance extraction of interconnects considering arbitrary random probability," *IEICE Trans. Electron*, vol. E92-C, no. 4, pp. 508–516, Apr. 2009.

[16] X. Wan and G. E. Karniadakis, "Multi-element generalized polynomial chaos for arbitrary probability measures," *SIAM J. Sci. Comput.*, vol. 28, no. 3, pp. 901–938, 2006.

[17] Y. Zhan, A. J. Strojwas, X. Li, L. T. Pileggi, D. Newmark, and M. Sharma, "Correlation-aware statistical timing analysis with non-gaussian delay distributions," in *IEEE Proc. DAC*, June 2005, pp. 77–82.

[18] J. Singh and S. Sapatnekar, "Statistical timing analysis with correlated non-gaussian parameters using independent component analysis," in *IEEE Proc. DAC*, July 2006, pp. 155–160.

[19] D. B. Xiu and J. S. Hesthaven, "High order collocation method for differential equations with random inputs," *SIAM Journal of Sci. Comput*, vol. 27, no. 3, pp. 1118–1139, 2005.

[20] E. Novak and K. Ritter, "Simple cubature formulas with high polynomial exactness," *Constructive Approximation*, vol. 15, no. 4, pp. 449–522, Dec. 1999.

[21] T. G. Szymanski, "Computing optimal clock schedules," in *IEEE Proc. DAC*, June 1992, pp. 399–404.

[22] A. Dasdan, "Experimental analysis of the fastest optimum cycle ratio and mean algorithms," *ACM Trans. on Design Automation of Electronic Systems*, vol. 9, no. 4, pp. 385–418, Oct. 2004.

[23] R. A. Howard, *Dynamic Programming and Markov Process*. The M.I.T. Press, Cambridge, 1960.

[24] P. Athanasios, *Probability, Random Variables, and Stochastic Processes, 2nd ed*. New York: McGraw-Hill, 1984.