STATISTICAL PLACEMENT FOR FPGAS CONSIDERING PROCESS VARIATION*

Yan Lin — Electrical Engineering Dept., UCLA, Los Angeles, CA 90095, ylin@ee.ucla.edu

Mike Hutton — Altera Corporation, San Jose, CA 95134, mhutton@altera.com

Lei He — Electrical Engineering Dept., UCLA, Los Angeles, CA 90095, lhe@ee.ucla.edu

ABSTRACT

Process variation affecting timing and power is an important issue for modern integrated circuits in nanometer technologies. FPGAs are similar to ASICs in their susceptibility to these issues, but face unique challenges in that critical paths are unknown at test time. This paper presents the first in-depth study on applying statistical timing analysis with cross-chip and on-chip variations to speed-binning and guard-banding in FPGAs. Considering the uniqueness of re-programmability in FPGAs, we quantify the effects of timing-model with guard-banding and speed-binning on statistical performance and timing yield. We also develop a new variation aware statistical placement, which is the first statistical algorithm for FPGA layout and achieves a yield loss of 29.7% of the original yield loss with guard-banding and a yield loss of 4% of the original one with speed-binning for MCNC and QUIP designs.

1. INTRODUCTION

Modern VLSI designs see a large impact from process variation as devices scale down to nanometer technologies. As for ASIC circuits, FPGAs are subject to variations in the operation of transistors comprising the logic functionality and the switching muxes. This variation can be classified as *global*, affecting all aspects of a given chip, *spatial/regional*, affecting geographic areas of the chip, or *local*, randomly affecting a transistor. Statistical static timing analysis (SSTA) has been proposed recently in [6][10] to analyze timing considering these variations.

FPGA architects are faced with a unique problem in that the same timing model will be applied to thousands of different

*This work was carried out primarily when Yan Lin was an intern at Altera Corporation. Yan Lin and Lei He were partially supported by NSF contract CCR-0306682 and a UC MICRO grant with support from Altera.

designs, operating at unknown clock frequencies and varied conditions. Chip-test must guarantee timing operation of a device independent of the configuration. The standard practice for timing models is to add guard-band to account for process, voltage and temperature variations. In the presence of on-chip variation, any fixed timing model would be more pessimistic for register setup on designs with long critical paths, and more optimistic for designs with short critical paths. Guard-band can be arbitrarily conservative or aggressive. On the other hand, an "advantage" that FPGAs have for timing modeling under variability is that FPGAs are binned for speed-grades, which serve to isolate global variation, and can be programmed repeatedly and differently during timing chip-test.

With variation any near-critical paths may actually be statistically critical. Criticality analyzed by static timing analysis (STA) and optimized by placement is based on the single longest path and ignores near-criticality. On the other hand, the *statistical criticality* of a node or an edge is defined as the probability that this node or edge is timing critical [6]. In Fig. 1, the mean/expected arrival time for PO1 is 2.08*ns* (analyzed by SSTA) while the nominal arrival time is 2*ns*. Near-critical paths PI1->PO1 have 69% chance to be timing critical while the static critical path (PI2->PO2) with nominal delay 2.05*ns* only has a probability of 31% to be timing critical with variation. Statistical criticality has recently been studied in [6] [12] [13], and applied to statistical gate sizing in [14][15] for ASICs. However, there is no existing statistical placement for FPGAs in the literature.

For FPGA placement, one of the representative approaches is the timing-driven algorithm, *T-VPlace* [5]. The interconnect delay estimation in T-VPlace is based on 2-pin net routing without congestion for each pair of locations. The actual delay after routing may differ from the estimated delay in placement, mainly due to the impact of congestion and multi-pin nets. This introduces interconnect delay uncertainty in addition to process variation. Fig. 2 compares the probability density functions (PDFs) for post-routing delay normalized to the estimated one in placement and post-routing delay with process variation normalized to the nominal one. As shown in this figure, more than 70% of nets have an estimation error within 1%. The remaining points along the X-axis only count for a small percentage of the total nets (less than 30% in total). On the other hand, the normalized standard deviation is 6% due to process variation. Process variation leads to larger delay spread, i.e. a more significant delay variance, and needs to be considered in placement.

In this paper we study the timing and placement considering process variation for FPGAs. We first discuss the effects due to guard-banding and speed-binning. We then present our new variation aware placement algorithm, *ST-VPlace*, leveraging an SSTA engine. ST-VPlace applies the same placement across chips. In order to quantify the benefit of ST-VPlace, we use

MCNC [7] and QUIP [8] designs for evaluation purpose. ST-VPlace achieves an average yield loss of 29.7% of the original yield loss using T-VPlace. With speed-binning, ST-VPlace achieves an average yield loss of 4% of the original one. Both are measured at the same clock frequency. ST-VPlace outperforms TV-Place statistically since SSTA is more accurate than STA if variations are considered.

Due to the lack of non-proprietary information on process variation and on FPGA vendor preferences for speed-binning and yield/performance tradeoffs, we parameterize variation, guard-banding values and speed-binning in our models to analyze the issues qualitatively and then apply various assumptions on the parameter values to generate quantitative results. This is similar to, for example, the 10% variation at 3σ assumption in [9].

The rest of the paper is organized as follows. Section 2 presents the preliminaries on variation model and SSTA, guard-banding and speed-binning, and previous work. Section 3 presents our variation aware placement algorithm. Section 4 gives experimental results and we conclude in Section 5. Note that the preliminary version of this paper was published in [16].

2. PRELIMINARIES

2.1 Variation Model

Delay of a circuit element (e.g. an LUT or a routing switch) is a random variable with process variation. As in [6], delay is modeled in a canonical first-order form as

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

where a_0 is the nominal value, ΔX_i represents the variation for each global source of variation X_i, a_i represents the sensitivity to each global variation, ΔR_a is the variation of an independent random variable R_a from its mean value, and a_{n+1} is the sensitivity of R_a. By scaling the sensitivities, X_i and R_a can be assumed as standard Gaussian $N(0, 1)$. ΔX_i and R_a are assumed as a set of independent random variables with principle component analysis (PCA) [10].

Although there are numerous sources of variation, variations in lithographic effects affecting L_{eff} and dopant atoms in oxide layers affecting V_{th} are considered. To make presentation simple, we denote the variation ΔL_{eff} and ΔV_{th} as L and V, respectively. As in [9], L and V can be decomposed into local (L_l, V_l) and global (L_g, V_g) components. The canonical first-order form then becomes

$$\begin{aligned} a &= a_0 + c_1 L_g + c_2 V_g + c_1 L_l + c_2 V_l \\ &= a_0 + (c_1 \sigma_{Lg}) \frac{L_g}{\sigma_{Lg}} + (c_2 \sigma_{Vg}) \frac{V_g}{\sigma_{Vg}} + \sqrt{(c_1 \sigma_{Ll})^2 + (c_2 \sigma_{Vl})^2} \Delta R_a \end{aligned} \quad \textbf{(2)}$$

where σ_{Lg}, σ_{Ll}, σ_{Vg} and σ_{Vl} are standard deviations for L_g, L_l, V_g and V_l respectively, and ΔR_a is the sum of two independent standard Gaussians L_l/σ_{Ll} and V_l/σ_{Vl}. SPICE simulation is performed to get the sensitivity parameters c_1 and c_2 for each type of circuit element. Based on (2), the standard deviation of the circuit element delay is,

$$\sigma_a = \sqrt{(c_1 \sigma_{Lg})^2 + (c_2 \sigma_{Vg})^2 + (c_1 \sigma_{Ll})^2 + (c_2 \sigma_{Vl})^2} \quad \textbf{(3)}$$

2.2 Statistical Static Timing Analysis

Statistical static timing analysis (SSTA) has recently been proposed to analyze timing considering variation [6][10]. The probabilistic equivalents of the "max", "min", "add" and "subtract" operations are involved in SSTA. With the delay expressed in the canonical form, addition and subtraction are performed easily [6]. The max or min of two Gaussians is not a Gaussian. We resort to the method in [11], which models the max of two Gaussians as a Gaussian by matching the first two moments of the real distribution. The max or min of two Gaussians is then modeled in the canonical form, which allows us to propagate the correlations due to global variation. With forward and backward traversals of the timing graph, the distribution of the arrival and requested arrival time for each node, and the statistical criticality for each node and edge can be calculated. The *statistical criticality* of a node or an edge is defined as the probability that this node or edge is timing critical. Given a cut-off delay T_{cut}, the *timing yield* is defined as the probability that the critical path delay is no longer than T_{cut} considering variation. Given the canonical form of the arrival time at the virtual sink, the mean T_μ and standard deviation T_σ of circuit delay can be calculated. With a cut-off delay T_{cut}, the timing yield can then be computed using cumulative density function (CDF) of standard Gaussian as $CDF((T_{cut}-T_\mu)/T_\sigma)$.

2.3 Guard-Banding

STA analyzes circuit timing based on constant delays. Without performing SSTA, a guard-band is applied for individual node to model uncertainty. The *nominal* delay of a circuit element is measured with the nominal values of V_{th} and L_{eff}. Given the sensitivity and variance of each variation source, the standard deviation σ can be obtained from (3). The individual node delay with a nominal value μ and standard deviation σ is then modeled as a constant *guard-banded* delay as $\mu+c\sigma$ (e.g., $\mu+4\sigma$ as a guard-banded longest delay and $\mu-4\sigma$ as a guard-banded minimum delay), where $c\sigma$ is the *guard-band factor*. The guard-banded minimum delay is used for register hold time analysis and is not considered in this paper. A more conservative

or aggressive guard-band would be to use, e.g., 5σ or 3σ (trading performance for timing yield). With the constant guard-banded delay for each circuit element, STA is performed to obtain the guard-banded circuit delay, T_{grd}. The *guard-band cost* is defined as (T_{grd}/T_{norm})-1, where T_{norm} is the circuit delay without guard-banding. For a given factor $c\sigma$, the cost is the percentage of guard-banding actually applied to the critical path, e.g. a nominal critical path is 10ns, but with guard-banding is evaluated as 11ns, thus giving a cost of 10%. With the delay distribution analyzed by SSTA and the guard-banded delay T_{grd} as the cut-off delay, timing yield can then be calculated using CDF of Gaussian to analyze the effect of guard-banding.

2.4 Speed-Binning

FPGAs, along with DSP processors, microprocessors and some other logic chips, have long-used speed-binning or speed-grading to handle global variation. The process of speed-binning is to test each chip's operational speed for a given timing path and thus define a chip as "fast", "medium" or "slow". While speed-binning is usually performed by post-silicon measurement, timing analysis considering speed-binning at the pre-silicon stage is equivalent to modeling global variation as a truncated Gaussian for each bin. We first model the set of global variation Gaussians ΔX_i in (1) as a single standard Gaussian ΔG_a to analyze timing considering speed-binning. Using (2) as an example, the delay of circuit element in the canonical form can be expressed as

$$\begin{aligned} a &= a_0 + \sigma_g \Delta G_a + \sigma_l \Delta R_a \\ \sigma_g &= \sqrt{(c_1\sigma_{Lg})^2 + (c_2\sigma_{Vg})^2} \quad \sigma_l = \sqrt{(c_1\sigma_{Ll})^2 + (c_2\sigma_{Vl})^2} \end{aligned} \qquad \textbf{(4)}$$

where ΔG_a is a standard Gaussian which models global variation. Speed-binning does not effectively deal with local variation. However to some extent local variation can be tested and averaged out by testing multiple similar paths across different placements on the test chip. Speed-binning based on the average speed of tested paths is equivalent to categorizing ΔG_a into different bins. Global variation ΔG_a has a *truncated Gaussian distribution* arising from binning. Fig. 3 shows an example, in which ΔG_a is categorized into three bins as [-∞, -1], [-1, 1] and [1, 3]. All chips fell into the fast bin have μ-σ_g (μ+σ_g for the medium bin and μ+$3\sigma_g$ for the slow bin) as the delay for each circuit element in the STA timing model. STA is then performed to obtain the circuit delay, T_{bin}, for each bin. T_{bin} may be relaxed by γ (*speed-bin relaxed factor*) to achieve a lower yield loss.

Speed-binning is effective to isolate global variation. However, the unique challenge in FPGAs is that the functionality is unknown because any number of different designs with different critical paths will be compiled onto FPGAs. This may result in yield loss even with binning as some designs have long critical paths which are treated optimistically by guardbanding and

short critical paths are treated less optimistically. The yield loss with speed-binning comes from two sources, the failure due to the ignored local variation and the correlation of global variation. Fig. 4(a) shows the effect of local variation. The arrival time distribution of PO is a sum of truncated global Gaussian and local Gaussian. The chips with global variable ΔG_a close to the truncated border may have a larger chance to fail affected by local variation. Fig. 4(b) shows the failure due to correlation of global variation. No local variation is assumed for simplicity. Edge *e1* and *e2* are Gaussian distributed as $1ns+N(0ns, 0.1ns)$ and $1ns+N(0ns, 0.3ns)$, respectively. ΔG_a in $[-\infty, 0]$ is categorized into the fast bin (shaded in the figure). T_{bin} of the shaded bin is $max(1ns, 1ns) = 1ns$. However, the arrival time distribution of PO given by SSTA is $max(e1, e2)$ as $1.08ns+N(0ns, 0.21ns)$. The increased mean delay is due to the fact that the mean of a set of random variables may be larger than the maximum mean of these variables. The chips operated between $1ns$ and $1.08ns$ fall into the shaded bin but fail to meet the expected timing specification T_{bin}.

To analytically calculate the timing yield with speed-binning, we first perform SSTA and obtain circuit delay distribution in canonical form as $T_\mu+\sigma_{Tg}\Delta G_a+\sigma_{Tl}\Delta R_a$, where σ_{Tg} and σ_{Tl} are the standard deviation due to global and local variation respectively. Given a specific bin k that categorizes ΔG_a into $[G^{low}(k), G^{up}(k)]$ and relaxed cut-off delay $\gamma T_{bin}(k)$, the timing yield for bin k is as

$$timing_yield(k) = \int_{G^{low}(k)}^{G^{up}(k)} pdf(\Delta G_a)\, cdf\left(\frac{\gamma T_{bin}(k) - (T_\mu + \sigma_{Tg}\Delta G_a)}{\sigma_{Tl}}\right) d\Delta G_a \quad \textbf{(5)}$$

The rationale is that given a fixed ΔG_a with probability of $pdf(\Delta G_a)$, the probability that $T_\mu+\sigma_{Tg}\Delta G_a+\sigma_{Tl}\Delta R_a$ meet the cut-off delay $\gamma T_{bin}(k)$ is $CDF(\gamma T_{bin}(k)-(T_\mu+\sigma_{Tg}\Delta G_a)/\sigma_{Tl})$. The overall timing yield for n bins are then expressed as

$$timing_yield = \sum_{k=1}^{n-1} timing_yield(k) \quad \textbf{(6)}$$

where the nth bin is assumed as the "dead" (too slow) bin and is discarded (required to avoid infinite delay).

2.5 Previous Work

Placement algorithms have been extensively studied for ASICs including min-cut partitioning [18], simulated annealing [19], and analytical methods [20]. However, there is no exiting work for statistical placement for ASICs considering process variation in the literature partially due to the fact that the interconnect delay prediction is inaccurate. As a result, the delay uncertainty during placement phase due to process variation is dominated by the effect of interconnect delay prediction. Similar to ASICs, various placement algorithms have been proposed for FPGAs. A simulated annealing based timing-driven

placement algorithm is presented in [5] considering both timing and wiring costs. A partition-based timing-driven placement with adaptive delay computation is presented in [17] for hierarchical PLD architectures. Simultaneous placement and clustering for wire length and delay reduction is proposed in [21]. Different from ASICs, the interconnect structure is more regular in FPGAs. As a result, the interconnect delay estimation is quite accurate as shown in Fig. 2 using the placement algorithm, T-VPlace [5]. The delay spread due to process variation is much wider than that due to interconnect uncertainty. This accurate interconnect estimation enables consideration of process variation in placement. Variation aware optimization has been studied in the literature. Statistical gate sizing leveraging statistical criticality as a guidance is proposed in [14][15]. An efficient statistical buffer insertion is studied in [23]. Statistical V_{dd}-level assignment is proposed in [22] to minimize FPGA interconnect power under timing yield constraint. With process variation, any near-critical path may become statistically timing critical. Statistical placement is needed to tackle this problem for FPGAs. However, no previous work has been published for statistical FPGA placement.

3. VARIATION AWARE PLACEMENT

For FPGAs, the dominant placement is simulated annealing as in the timing-driven algorithm *T-VPlace* [5] in VPR [4]. Here we present a new variation aware statistical timing-driven algorithm, *ST-VPlace*, to optimize timing statistically and maximize timing yield.

3.1 Timing-Driven Placement T-VPlace

Simulated annealing is a heuristic and iterative algorithm in which moves (swaps of logic cells) are accepted or rejected based on a cost function and annealing temperature. T-VPlace considers both wiring and timing costs. Wiring cost is as

$$Wiring_Cost = \sum_{i=1}^{N_{nets}} q(i)[bb_x(i) + bb_y(i)] \quad \textbf{(7)}$$

where N_{nets} is the number of nets in the circuit. The cost of net i is determined by its horizontal and vertical spans, $bb_x(i)$ and $bb_y(i)$. Scaling factor $q(i)$ compensates for multi-terminal nets.

Timing cannot be optimized explicitly since it is too expensive to perform a timing analysis after each move. The timing cost is a heuristic and based on *static criticality* of each edge (i, j), the delay of each edge $d(i, j)$ and criticality exponent β. The timing cost of edge (i, j) and for the placement solution are as

$$\begin{aligned} Timing_Cost(i, j) &= d(i, j) \bullet criticality(i, j)^{\beta} \\ criticality(i, j) &= 1 - slack(i, j) / D_{max} \end{aligned} \quad \textbf{(8)}$$

$$Timing_Cost = \sum_{i,j} Timing_Cost(i, j) \quad \textbf{(9)}$$

where *d(i, j)* is obtained from the delay lookup matrix and the current placement, D_{max} is the critical path delay, and slack is the amount of delay that can be added to routing edge *(i ,j)* without increasing the critical path delay. Both D_{max} and slack are calculated by STA, which is performed once at every annealing temperature. The criticality exponent β is used to control the relative importance of connections with different criticalities.

The overall cost function is then shown in (10), where λ is a trade-off variable between timing and wiring cost. Previous timing and previous wiring cost are updated once every temperature. The temperature and ΔC are used to decide whether a move is to be accepted or rejected. It was shown in [5] that $\beta=8$ and $\lambda=0.5$ give the best timing and wiring trade-off.

$$\Delta C = \lambda \frac{\Delta Timing_Cost}{Previous_Timing_Cost} + (1-\lambda) \frac{\Delta Wiring_Cost}{Previous_Wiring_Cost} \quad \textbf{(10)}$$

3.2 Variation Aware Placement ST-VPlace

Timing yield depends on both of the mean and variance of circuit delay. Under presence of variation, any near-critical path may actually be statistically critical. However, the cost function of T-VPlace may not optimize the timing yield, and further cannot see the effect of near-critical paths as per the discussion of Fig. 1. To make placement variation aware, we introduce the concept of *statistical criticality* and develop a new algorithm, *ST-VPlace* (see Fig. 5), to optimize timing yield considering variation.

There are three main differences between ST-VPlace and T-VPlace. Firstly, in addition to the delay matrix, we calculate a delay variance matrix for each pair of locations for clusters and input/output pads. First-order canonical form is pre-characterized for all circuit elements using SPICE. The canonical form for delay of a routing path is then calculated by performing statistical addition for the interconnect switches in that path. Secondly, given the delay and variance for each edge, SSTA instead of STA is performed at each temperature to obtain the statistical criticality for each edge. For simplicity, we implement the block-based SSTA from [6] with statistical criticality calculation for each edge. Spatial variation is not considered but can however been easily modeled with PCA [10]. Finally, instead of using the static timing cost function in (8), we define the statistical timing cost function for each routing edge *(i, j)* and a placement solution as

$$\begin{aligned} STiming_Cost(i, j) &= d_{\mu}(i, j) \bullet SCriticality(i, j)^{\theta} \\ STiming_Cost &= \sum_{i,j} STiming_Cost(i, j) \end{aligned} \quad \textbf{(11)}$$

where $d_\mu(i, j)$ is the nominal delay for each edge *(i, j)* and *SCriticality(i, j)* is the statistical criticality, i.e., the probability that edge *(i, j)* is in critical path. *SCriticality(i, j)* is updated at each new annealing temperature using SSTA. Statistical criticality exponent, θ, is a constant parameter. We experimentally tune θ to be 0.3 to obtain the minimum mean and standard deviation of circuit delay. We use the same wiring cost in (7) and λ of 0.5 for the same timing and wiring trade-off in ST-VPlace. The same annealing scheme in T-VPlace is also adopted in ST-VPlace. The goal of ST-VPlace is to perform placement considering variations and to optimize for the maximum probabilistic timing yield leveraging the back-end SSTA.

4. EXPERIMENTAL RESULTS

In this section, we conduct the experiments on the largest MCNC [7] and QUIP [8] designs. We use Berkeley predictive device model [2] at ITRS [3] 65*nm* technology node. Suggested in [9] for higher yield, we use the min-ED (energy-delay product) device setting (V_{dd} = 0.9v and V_{th} = 0.3v). An island style FPGA architecture resembling Altera's Stratix® device [1] with 10 4-LUT clusters and 60% length-4 and 40% length-8 wires is used. T-VPlace [5] serves as the baseline. 1.2X of minimum routing channel width obtained by T-VPlace is used for each design in both placers. The same timing-driven router is performed for two placers and delay is analyzed after routing. We also assume a variation in each of L_{eff} and V_{th} of 10% at 3σ (i.e. a 99.73% chance that variation is within +/- 10% deviated from the nominal value) for both global and local variation unless specified otherwise.

We first tune the cost function for ST-VPlace and compare the yield loss between ST-VPlace and T-VPlace using the same cut-off delay as 3σ guard-banded delay in T-VPlace. We then compare the two algorithms with guard-banding and speed-binning.

4.1 Yield Loss Comparison

To tune the statistical criticality exponent θ in (11), we perform ST-VPlace and SSTA to obtain the geometric mean of mean delay and standard deviation of circuit delay after routing over all designs. Various values for θ from 0.1 to 2 are evaluated as shown in Fig. 6. It is clear that θ of 0.3 leads to the minimum mean and standard deviation of delay, which result in the maximum timing yield. In the rest of the paper, we set θ as 0.3 for experiments.

We compare ST-VPlace and T-VPlace in Table 1. The geometric mean is shown for the aggregate 20 MCNC and 20 QUIP designs along with 6 representative individual designs. Column 2 presents the number of clusters for each design. Columns 3-5 present the results obtained by T-VPlace, where "T_{norm}" is the nominal delay given by STA, "T_{grd}" is the 3σ guard-banded delay, and "YL_{grd}" is the yield loss using "T_{grd}" as the cut-off delay analyzed by SSTA. The *yield loss* is

defined as the number of parts that fail to meet the timing requirement out of 10,000 parts, in short, parts per 10K (pp10K). On average, when evaluating with 3σ guard-banded delay, the yield loss for MCNC, QUIP and overall designs are 7.24, 1.42 and 3.20 pp10K respectively. Columns 6-10 present the results achieved by ST-VPlace. "T_{mean}" and "T_{sigma}" are the mean and standard deviation of circuit delay obtained by SSTA. "T_{st-v}" represents the delay when holding the same yield loss with T-VPlace as in column 5 ("YL_{grd}"). On average, ST-VPlace reduces the delay by 4% (up to 12%), 2% (up to 5.4%) and 3% for MCNC, QUIP and overall designs. "YL_{st-v}" represents the yield loss with ST-VPlace when using the same cut-off delay of T-VPlace as in column 4 ("T_{grd}"). On average, ST-VPlace achieves a yield loss of 19.7% (up to 0.31%), 44.8% (up to 11.9%) and 29.7% of the original yield loss for MCNC, QUIP and overall design set, respectively. Though not shown in the table, the wire length overhead of ST-VPlace is negligible (less than 1%) compared to T-VPlace.

1	2	3	4	5	6	7	8	9	10
circuit	# of clusters	T-VPlace			ST-VPlace				
		T_{norm}	T_{grd}	YL_{grd}	T_{mean} (ns)	T_{sigma} (ns)	T_{st-v} (ns)	YL_{st-v} (pp10K)	runtime
apex2	213	22.19	32.04	10.84	19.76	2.73	28.13 (-12.22%)	0.03 (0.31%)	1.37X
clma	1358	38.05	56.05	6.09	35.26	4.90	51.10 (-8.83%)	0.11 (1.80%)	1.26X
s38584	704	20.14	32.04	2.56	19.50	3.16	30.49 (-4.84%)	0.37 (14.4%)	1.27X
pdc	568	25.34	36.49	7.78	25.14	3.14	35.08 (-3.87%)	1.50 (19.3%)	1.11X
s38417	847	24.32	38.35	4.62	24.78	3.81	37.42 (-2.42%)	1.88 (40.7%)	1.27X
seq	198	16.83	24.34	8.89	17.08	2.32	24.33 (-0.04%)	8.76 (98.0%)	1.51X
…	…	…	…	…	…	…	…	…	…
Geo. (MCNC)	**296**	**21.65**	**32.35**	**7.24**	**21.32**	**3.04**	**31.06 (-4.00%)**	**1.42 (19.7%)**	**1.29X**
oc_des_des3area	115	32.40	51.39	3.23	31.10	5.14	48.61 (-5.41%)	0.39 (11.9%)	1.14X
oc_wb_dma	577	20.60	32.31	2.11	19.85	3.12	30.85 (-4.52%)	0.32 (15.4%)	1.42X
oc_des_perf_opt	534	10.70	17.01	3.10	11.13	1.62	16.67 (-2.01%)	1.39 (45.1%)	1.58X
oc_cordic_p2r	282	22.44	39.03	0.37	22.51	4.00	38.33 (-1.81%)	0.18 (47.2%)	1.45X
oc_mem_ctrl	446	22.40	38.37	1.31	23.18	4.01	37.82 (-1.45%)	0.76 (57.8%)	1.52X
idea_parallel	534	69.36	117.85	0.98	71.20	12.40	117.40 (-0.38%)	0.85 (86.2%)	1.44X
…	…	…	…	…	…	…	…	…	…
Geo. (QUIP)	**150**	**21.05**	**34.87**	**1.42**	**21.27**	**3.55**	**34.16 (-2.03%)**	**0.64 (44.8%)**	**1.51X**
Geo. (ALL)	**189**	**21.35**	**33.59**	**3.20**	**21.29**	**3.28**	**32.57 (-3.02%)**	**0.95 (29.7%)**	**1.40X**

Table 1. Comparison between ST-VPlace and T-VPlace

Column 10 presents the runtime of ST-VPlace normalized to T-VPlace. The block-based SSTA is linear time complexity of $O(kn)$ where k is number of global variation sources. Since SSTA is only performed once at every annealing temperature, the average complexity of *SSTA()* (see Fig. 5) is $O(1)$. ST-VPlace has the same average complexity of $O(n^{4/3})$ as T-VPlace. ST-VPlace consumes an average of 1.29X, 1.51X and 1.40X runtime for MCNC, QUIP and overall designs, respectively.

ST-VPlace reduces both the mean value and the variance compared to T-VPlace. Fig. 7 presents the mean and standard deviation of circuit delay obtained by ST-VPlace for the overall 40 designs. Each is normalized to its counterpart in T-VPlace. As shown in this figure, ST-VPlace consistently reduces the mean delay for all designs and standard deviation of delay for most designs, which in turn provides a higher yield than T-VPlace does. Note that ST-VPlace achieves slightly

larger standard deviation for some designs. It is due to the heuristic cost function in ST-VPlace. On average, ST-VPlace reduces the mean delay by 4.2%, 2.5% and 3.3%, and reduces the standard deviation of delay by 3.5%, 1.4% and 2.5% for MCNC, QUIP and overall design suite. This tightened and shifted delay distribution with smaller mean and standard deviation of circuit delay contributes to the significant yield loss reduction of ST-VPlace. Note that ST-VPlace is based on the same simulated annealing framework as T-VPlace. More sophisticated statistical placement algorithm can be developed for more significant improvement.

With variation, the static criticality analyzed by STA may not reflect how critical the near-critical path is. Fig. 8 compares the static criticality defined in (8) and the statistical criticality analyzed by SSTA for design *apex2* in T-VPlace. The statistical criticality does not increase monotonically with static criticality (see Fig. 8(a)), i.e., a larger static criticality does not necessarily lead to a larger probability that this edge is timing critical with variation. As shown in Fig. 8(b), the statistical criticality may vary significantly with a same static criticality (up to 227X). Since ST-VPlace considers statistical criticality explicitly, it optimizes the near-critical paths with variation.

4.2 Guard-Banding

Without SSTA, a guard-band is applied for individual nodes to model uncertainty. A larger *guard-band factor* leads to a larger *guard-band cost* (defined in Section 2.3) but a smaller yield loss. For instance, the nominal delay from LUT-A input to output is 551*ps* with a standard deviation of 110*ps* under 10%/10% global/local variation. The standard deviation becomes 25*ps* and 221*ps* under 0%/5% and 20%/20% variation respectively. Fig. 9 presents the cost with different guard-band factors under various variation assumptions[1]. The cost and yield loss in pp10K are the arithmetic and geometric means for all designs, respectively. The cost ranges from 20% to 100% with guard-band factor of 3σ under different variation assumption. When there is no global variation, the yield loss of T-VPlace drops from more than 5000 pp10K with no guard-banding to 1 pp10K with guard-band factor of 1σ. It is due to the fact that long switching paths are present in FPGA and dampen the local variation. When both global and local variations are present, around 10 pp10K may still fail even with a guard-band factor of 3σ. It is due to the fact that global variation affects all circuit elements on chip and the variation is aggregated in a long path. Note that in either case (with and without global variation), the yield loss does not depend on variation significantly but only on guard-band factor since a larger variation leads to a larger guard-banded delay. This correlation results in a relatively stable yield loss with a constant guard-band factor.

Compared to T-VPlace, ST-VPlace always obtains a smaller yield loss with different guard-band factors and various variation assumptions. When only local variation is present, ST-VPlace reduces yield loss by around one magnitude (e.g. from 1 to 0.1 pp10K with guard-band factor of σ). With both global and local variations, the gain of ST-VPlace increases with larger guard-band factor under the same variation assumption (e.g. from 1000 to 500 pp10K with guard-band factor of 1σ but from 10 to 0.5 pp10K with guard-band factor of 3σ under 5%/5% variation) and decreases with the same guard-band factor but under larger variation (e.g. with guard-band factor of 3σ, from 10 to 0.5 pp10K under 5%/5% variation but from 5 to 1 pp10K under 20%/20% variation).

4.3 Speed-Binning

In practice, speed-binning is used to handle global variation in FPGAs. We arbitrarily generate the bins such that the "fast", "medium" and "slow" bins contain 40%, 30% and 29.999% chips, respectively. The slowest 0.001% (or 1 part per 100K parts which is 0.1pp10K) chips are discarded.

Fig. 10 compares the overall yield loss in pp10K given by ST-VPlace and T-VPlace with different *speed-bin relaxed factors* (Section 2.4). 10% at 3σ is assumed for both global and local variation. Timing yield is analytically calculated from (6). When the cut-off delay of each bin is not relaxed, on average 1565 chips fail to meet timing using T-VPlace. The yield loss is due to the ignored local variation and the correlation of global variation, and can be reduced by relaxing the cut-off delay as discussed in Section 2.4. When cut-off delays are relaxed by 5% (γ=1.05), the yield loss reduces to 45 pp10K. With the same cut-off delay, the yield loss of ST-VPlace is reduced to 1.8 pp10K (or 4% of original yield loss). ST-VPlace always achieves a lower yield loss with different relaxed factors. With sufficiently large relaxed factor, the yield loss of ST-VPlace and T-VPlace tends to be 0.1 pp10K since we always discard the slowest 0.1 pp10K. Overall, ST-VPlace achieves a yield loss of 10% of the original one in the presence of speed-binning, effectively reducing the need for relaxation factors.

5. CONCLUSIONS AND DISCUSSIONS

In this paper, we have quantified the effect of timing-models with guard-banding and speed-binning on statistical performance and timing yield of FPGAs for future process generations with large on-chip variation. We have developed an analytical timing yield model for the truncated Gaussian distribution for global variation arising from speed-binning. We have also offered a new variation aware placement algorithm, ST-VPlace. Using the the same cut-off delay as 3σ guard-

[1] The average yield loss of T-VPlace (3.2 pp10K) and ST-VPlace (0.95 pp10K) in Table 1 match the 3 sigma points in the middle right figure in Fig. 9.

banded delay in T-VPlace, ST-VPlace achieves an average yield loss of 29.7% of the original one by T-VPlace. With speed-binning, ST-VPlace achieves an average yield loss or 4% or the original one at the same performance. Though not done here, ST-VPlace can be applied with spatial correlated variation model.

In the future, we will study the FPGA timing with non-Gaussian variation and spatial correlated variation. To make the work more directly applicable, it would be useful to perform device-specific transistor parameterization (e.g. on L) as would be done on more sensitive transistors of commercial FPGAs to minimize variation (rather than the general single variation model).

6. REFERENCES

[1] D. Lewis *et.al.*, "The Stratix routing and logic architecture", *Proc. Int'l Symp. FPGAs*, Monterey, Feb 2003. pp 14-20

[2] Berkeley Device Group, "Predictive technology model", in http://www.device.eecs.berkeley.edu/ptm/modfet.html, 2002.

[3] International Technology Roadmap for Semiconductors, http://public.itrs.net, 2003

[4] V. Betz, J. Rose, and A. Marquardt, *Architecture and CAD for Deep-Submicron FPGAs*, Kluwer, 1999

[5] A. Marquardt, V. Betz and J. Rose, "Timing-Driven Placement for FPGAs", *Proc. Int'l Symp. FPGAs*, Monterey, Feb 2000

[6] C. Visweswariah et al, "First-Order Incremental Block-based Statistical Timing Analysis", *Proceeding. of Design Automation Conference*, San Diego, June 2004, pp. 331-336

[7] S. Yang, "Logic synthesis and optimization benchmarks, version 3.0", Microelectronics Center of North Carolina, Tech. Rep., 1991.

[8] Altera Corp, "QUIP for Quartus II V5.0", available at http://www.altera.com/education/univ.

[9] H. Wong, L. Cheng, Y. Lin and L. He, "FPGA device and architecture evaluation considering process variations", *Proceeding of International Conference on Computer-Aided Design,* San Jose, Nov. 2005, pp. 19-24

[10] H. Chang and S. Sapatnekar, "Statistical timing analysis considering spatial correlations using a single pert-like traversal", *Proceeding of International Conference on Computer-Aided Design,* San Jose, Nov. 2003, pp. 621-625

[11] C. E. Clark, "The greatest of a finite set of random variables", in *Operation Research, Vol. 9, No.2,* 1961

[12] J. Xiong and et al, "Criticality computation in parameterized statistical timing", in *Proceeding. of Design Automation Conference*, San Francisco, July 2006, pp. 63-68

[13] X. Li and et al, “Defining statistical sensitivity for timing optimization of logic circuits with large-scale process and environmental variations”, *Proceeding of International Conference on Computer-Aided Design,* San Jose, Nov. 2005, pp. 844-851

[14] M. R. Guthaus and et al, “Gate sizing using incremental parameterized statistical timing analysis”, *Proceeding of International Conference on Computer-Aided Design,* San Jose, Nov. 2005, pp. 1029-1036

[15] D. Sinha, N. Shenoy and H. Zhou, “Statistical gate sizing for timing yield optimization”, *Proceeding of International Conference on Computer-Aided Design,* San Jose, Nov. 2005, pp. 1037-1041

[16] Y. Lin, M. Hutton and L. He, “Placement and Timing for FPGAs Considering Variations”, *Proceeding of International Conference on Field Programmable Logic and Applications*, Spain, August 2006

[17] M. Hutton, K. Adibsamii and A. Leaver, “Timing-Driven Placement for Hierarchical Programmable Logic Devices”, *Proceedings of the international symposium on Field programmable gate arrays*, Monterey, Feb. 2001, pp. 3-11

[18] A. Caldwell, A. B. Kahng and I. Markove, “Can recursive Bisection Alone Produce Routable Placements?”, in *Proceeding. of Design Automation Conference*, Los Angeles, June 2000, pp. 477-482

[19] C. Chang, J. Cong and X. Yuan, “Multi-level Placement for Large-scale Mixed-size IC Designs”, in *Proc. Asia South Pacific Design Automation Conference*, Jan. 2003, pp. 325-330

[20] J. Kleinhans, G. Sigl, F. Johannes and K. Antreich, “Analytical placement: a linear or a quadratic objective function?”, in *Proceeding. of Design Automation Conference*, June 1991, pp. 427-432

[21] G. Chen and J. Cong, “Simultaneous timing driven clustering and placement for FPGAs”, in *Proc. Conference on Field Programmable Logic and its Applications*, August 2004, pp. 158-167

[22] Y. Lin and L. He, “Statistical Dual-Vdd Assignment for FPGA Interconnect Power Reduction”, *Design Automation and Test in Europe*, France, April 2007

[23] J. Xiong and L. He, “Fast Buffer Insertion Considering Process Variations”, *International Symposium on Physical Design*, San Jose, CA, April 2006, pp. 128-135,

List of Figures

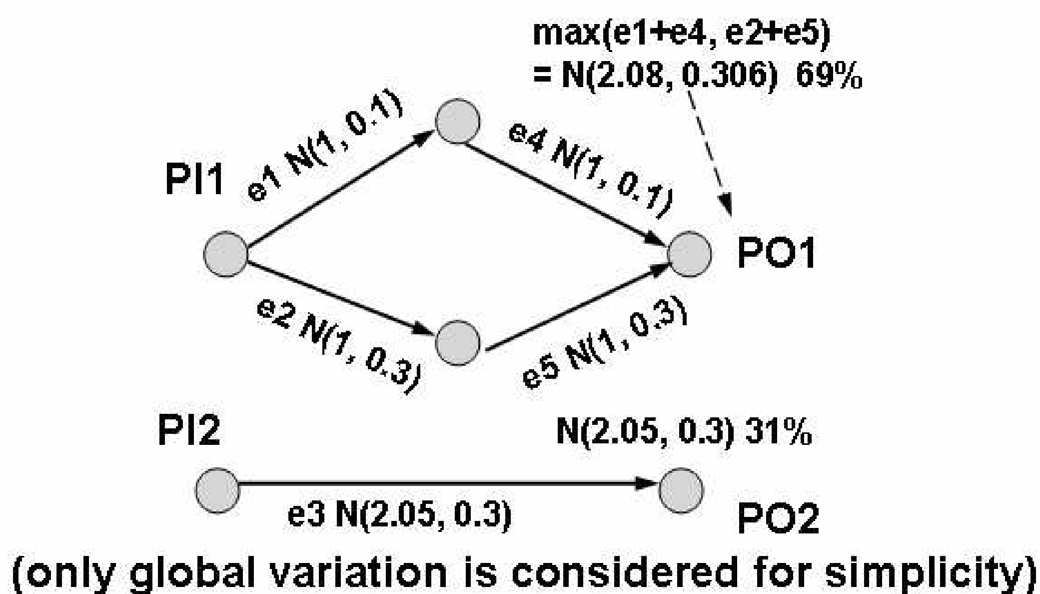

Fig. 1. Near-critical path analysis with variation and normal distribution is assumed

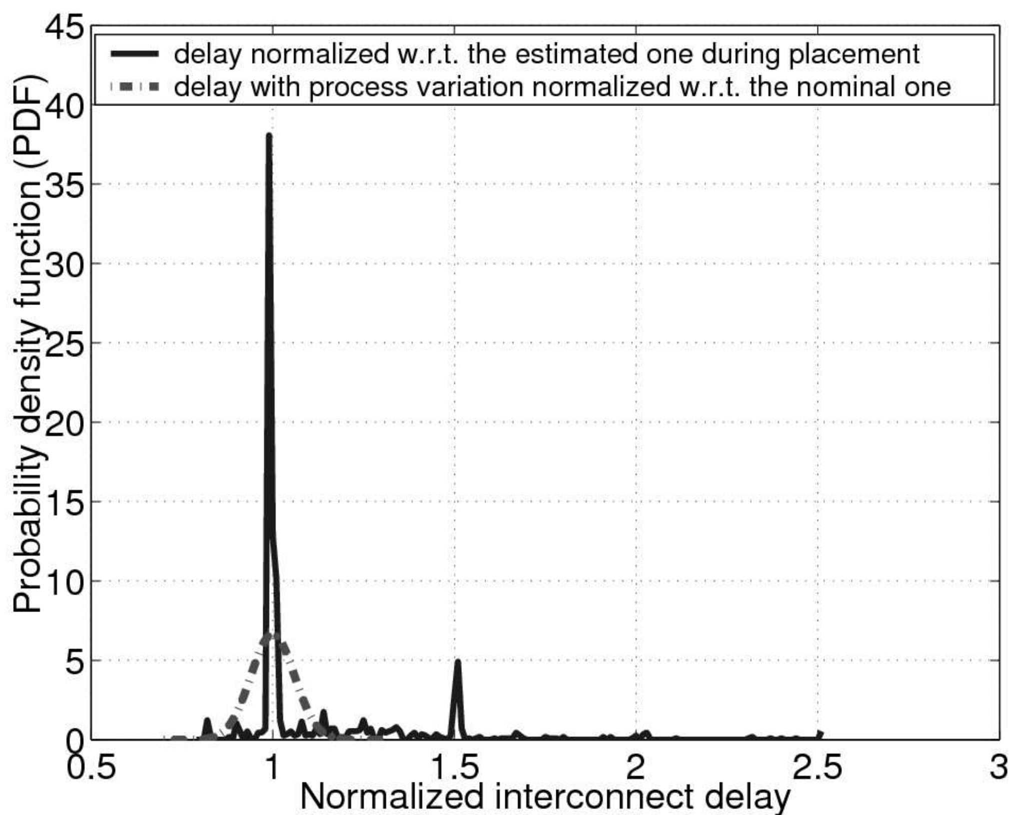

Fig. 2. Comparison between uncertainty due to interconnect estimation and process variation

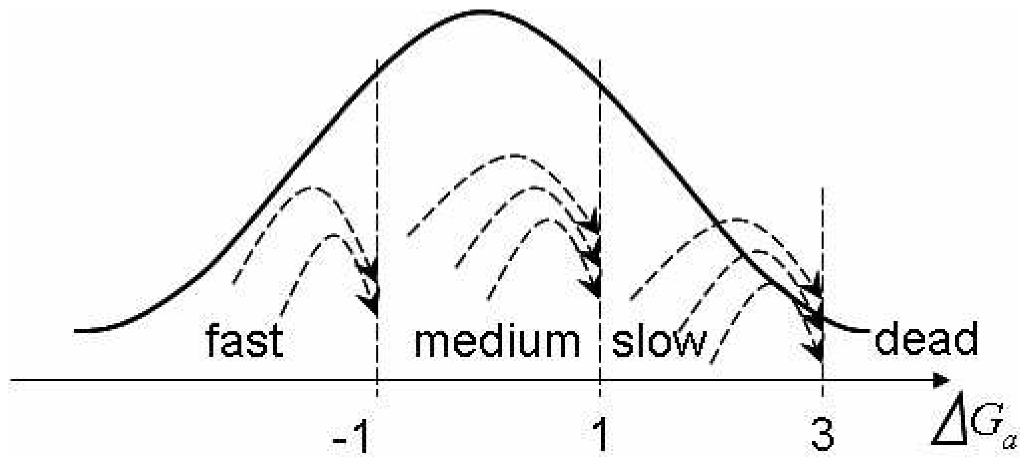

Fig. 3. Example of speed-binning for global variation

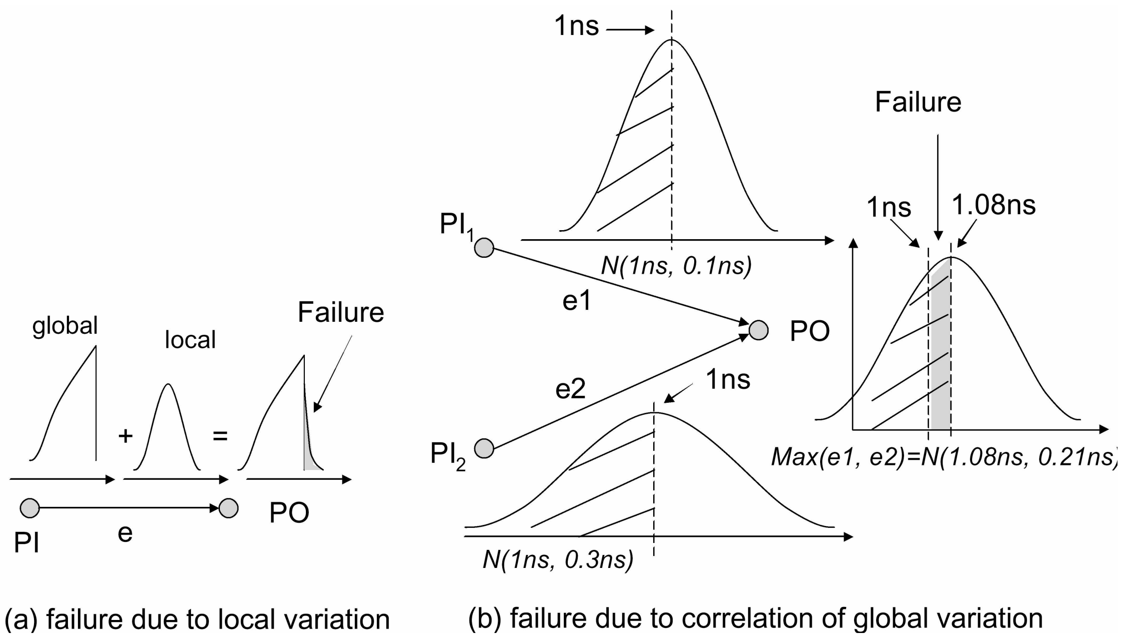

Fig. 4. Illustration of yield loss with speed-binning

```
InitPlacementTemperatureRlimit(&S, &T, &Rlimit);
ComputeDelayVarianceMatrix();
while (ExitCriterion () == False) { /* "Outer loop" */
    SSTA();
    Previous_Wiring_Cost =Wiring_Cost(S);
    Previous_STiming_Cost = STiming_Cost(S);
    while (InnerLoopCriterion () == False) { /* "Inner loop" */
        Snew = GenerateViaMove (S, Rlimit);
        ΔSTiming_Cost=STiming_Cost(Snew)–STiming_Cost(S);
        ΔWiring_Cost=Wiring_Cost(Snew) -Wiring_Cost(S);
        ΔC =λ·(ΔSTiming_Cost/Prev_STiming_Cost) +
        (1-λ)·(ΔWiring_Cost/Previous_Wiring_Cost);
        if (ΔC < 0) {
            S = Snew /* Move is good, accept */
        }
        else {
            r = random (0,1);
            if (r < e^(-ΔC/T))
                S = Snew; /* Move is bad, accept anyway */
        }
    } /* End "inner loop" */
    UpdateTempRlimit(&T, &Rlimit);
} /* End "outer loop" */
```

Fig. 5. Overall algorithm of ST-VPlace

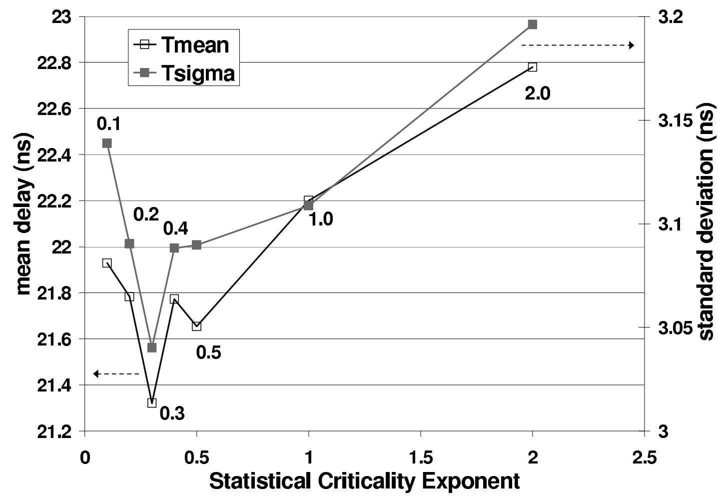

Fig. 6 Cost function tuning for ST-Vplace

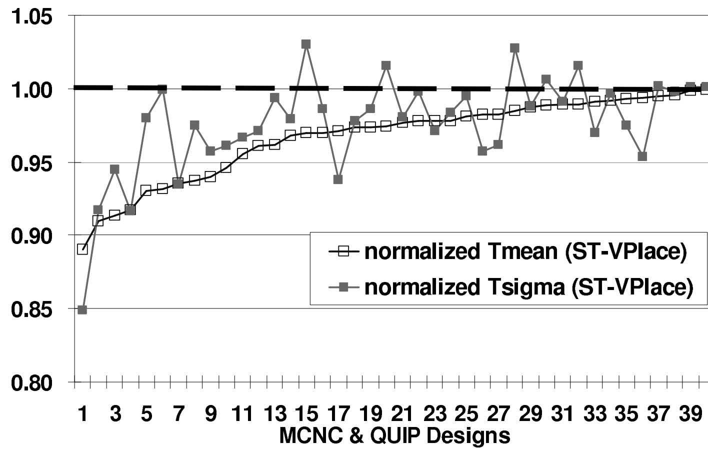

Fig. 7 Normalized mean and standard deviation of circuit delay obtained by ST-VPlace

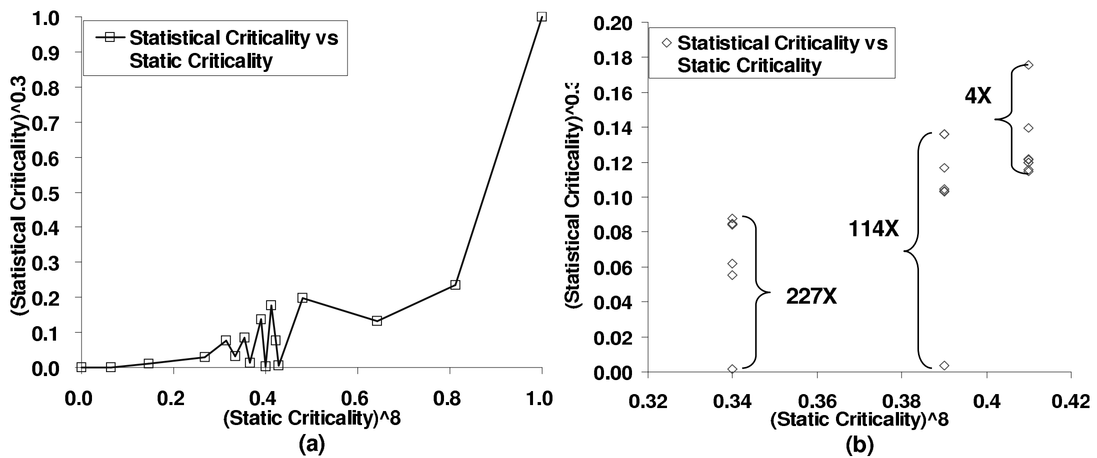

Fig. 8 Statistical criticality versus static criticality

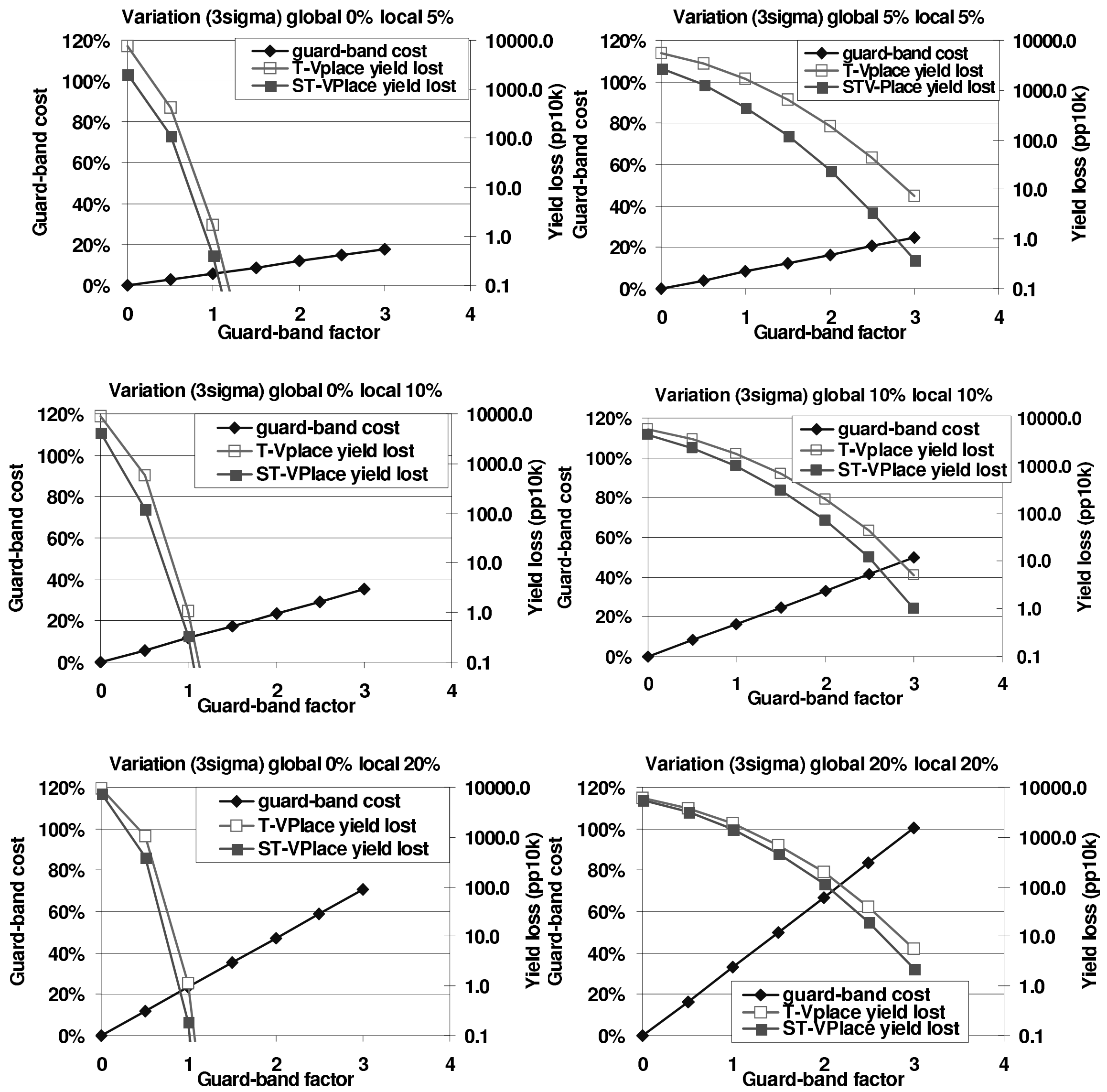

Fig. 9 Guard-band cost and yield loss comparison under different variation assumptions

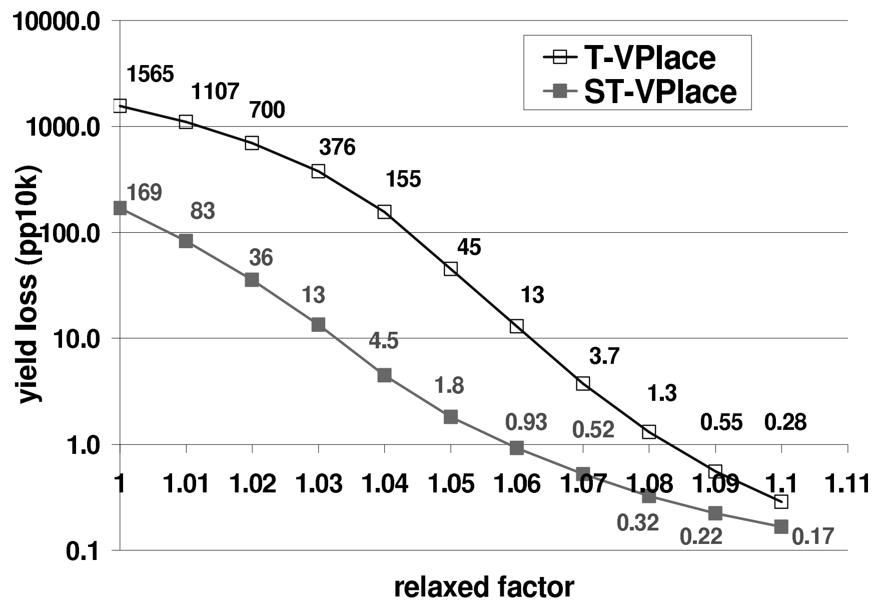

Fig. 10 Comparison of yield loss between ST-VPlace and T-VPlace considering speed-binning effect