Simultaneous Buffer Insertion and Wire Sizing Considering Systematic CMP Variation and Random Leff Variation

Lei He[1] Andrew Kahng[2] King Ho Tam[1] Jinjun Xiong[1]
EE Department, University of California at Los Angeles[1], CA 90095, USA
ECE Department, University of California at San Diego[2], CA 92093, USA
{lhe,ktam,jinjun}@ee.ucla.edu[1], {abk}@ucsd.edu[2]

***Abstract*— This paper presents extensions of the dynamic programming framework to consider buffer insertion and wire-sizing under effects of process variation. We study the effectiveness of this approach to reduce timing impact caused by Chemical Mechanical Polishing (CMP)-induced systematic variation and random L_{eff} process variation in devices. We first present a quantitative study on the impact of CMP to interconnect parasitics. We then introduce a simple extension to handle CMP effects in the simultaneous buffer insertion and wire sizing problem by simultaneously considering fill insertion ($SBWF$). We also tackle the same problem but with random L_{eff} process variation ($vSBWF$) by incorporating statistical timing into the dynamic programming framework. We develop an efficient yet accurate heuristical pruning rule to approximate the computationally expensive statisitcal problem. Experiments under conservative assumption on process variation show that $SBWF$ algorithm obtains 1.6% timing improvement over the variation-unaware solution. Moreover, our statisitical $vSBWF$ algorithm results in 43.1% yield improvement on average. We also show that our approaches have polynomial time complexity with respect to the net-size. The proposed extensions on the dynamic programming framework is orthogonal to other power/area-constrained problems under the same framework, which has been extensively studied in the literature.**

I. Introduction

Design uncertainty in nanometer technology nodes threatens cost-effectiveness of high-performance circuit manufacturing processes. Design uncertainty renders itself in the forms of systematic manufacturing process variation and random process variations due to small geometric dimensions [1]. Considered as one of the most significant source of systematic variation, chemical-mechanical planarization (CMP) is an enabling manufacturing process to achieve uniformity of dielectric and conductor height in back-end-of-line (BEOL) process step. CMP introduces systematic design variations due to *dummy fill* insertion [2] and *dishing* and *erosion* [3]. On the other hand, channel length of a transistor (L_{eff}) subjects to random variation as pointed out by [4]. Such variation has great impact on buffered interconnect timing as buffers' driving strength depends strongly on L_{eff}. As a result of combined systematic and random variations, it is unclear whether interconnects designed from variation-unaware design automation tools live up to the timing yield that we expect by means of static timing analysis. This paper studies the buffer insertion and wire-sizing problem, which is a classical physical design problem, by proposing and experimenting intuitive and efficient ways to deal with process variation.

To deal with systematic variation, it is important to understand the nature and properties of the variation source and its correlation to the design. In the case of CMP, it is understood that dummy fill insertion for CMP planarization changes interconnect capacitance, and that different dummy fill pattern brings different changes. Moreover, metal loss due to uneven polishing, which is dubbed *dishing* and *erosion* in CMP terminology, adds to variation in interconnect resistance. However, there is no extensive and quantitative study in the literature on the interconnect performance variation due to CMP. [5] assumes only one regular fill pattern array and shows that the increase of interconnect capacitance due to such a fill pattern cannot be ignored for interconnect optimization. [6] has considered the variation of total capacitance due to the Boolean-based placement of dummy fills and has shown that up to 25% variation is possible, albeit with only one fill pattern. [7] has proposed to examine the impact due to different fill patterns, however, no quantitative experiment results have been reported.

Researches start emerging on circuit optimization for yield improvement considering process variations. Statistical timing analysis [8], [9], [10] has been studied recently, but results mainly focus on analysis rather than design. Most statistical circuit optimization works focus on solving the gate-sizing problem. [11] introduces modification to the non-linear programming formulation for the gate-sizing problem through iterative delay constraint adjustment. [12] is similar except that the modification is based on scaling the objective function with a "dis-utility" function, which is an ad-hoc metric that reflects the "spread" of the overall timing distribution. More recently, [13] proposes a statistical sensitivity-based gate sizing algorithm which is based on bound computation of probability. All these works either assumes delay distributions as Gaussian or do not compute accurate CDF. Another recent work [14] presents a buffer insertion methodology in a routing tree,

Research at UCLA is partially supported by NSF CAREER award CCR-0401682, SRC grant 1100, a UC MICRO grant sponsored by Analog Devices, Fujitsu Laboratories of America, Intel and LSI Logic, and a Faculty Partner Award by IBM. Dr. Kahng is currently with Blaze DFM, Inc., Sunnyvale, CA 94089. Research at UCSD was partially supported by the MARCO Gigascale Silicon Research Center and the National Science Foundation. Address comments to Dr. He at lhe@ee.ucla.edu.

which considers the uncertainty in wire-length estimation but not process variations such as CMP effects and L_{eff} variation.

This paper first quantitatively studies interconnect parasitic variations due to CMP effects. Specifically, we study different fill patterns that are "equivalent" with respect to foundry rules, and dishing and erosion of conductors and dielectric similar to those predicted by ITRS [15] (Section II). We then present our extension of the dynamic programming framework [16] which solves the simultaneous buffer insertion and wire sizing problem [17] under CMP-induced systematic variation and random L_{eff} variation. To perform optimization under CMP effects, fill pattern design must be considered simultaneously with buffers and wire sizes, and we name the resulting problem as $SBWF$ (Section III). We then discuss the $SBWF$ problem which also considers random L_{eff} variation ($vSBWF$) by designing with statistical timing (Section IV). We propose a few techniques which accurately and efficiently handle statistical timing that avoids the exponential runtime complexity. We conclude the paper with discussion of our future research (Section V).

II. Modeling of CMP Variation

This section describes the effect of dummy fill insertion, dishing and erosion on interconnect parasitics as a result of CMP. To minimize dishing and erosion, foundaries require dummy fill insertion to even out the metal density across the die. However, dummy fill insertion may lead to increase in parasitic capacitance on interconnects. This section describes the parasitic model and presents statistics of potential impact of CMP on parasitic capacitance and resistance.

A. Fill Patterns Exploration

We explore a wide range of design-rule-check (DRC) correct fill patterns. We assume rectangular, isothetic fill features aligned horizontally and vertically between two adjacent interconnects as shown in Figure 1, which are sandwitched between an upper and lower ground planes. In the figure, conductors A and B are *active* interconnects and the metal shapes between them is dummy fill. We assume all dummy fill is implemented as floating metals in the final layout, as floating dummy fill are preferred for most ASIC designs due to the short design time and considerable area to be filled [18], [5]. Each distinct, DRC-correct *fill pattern* $P(M, N, W_i, L_j, S_{x,i}, S_{y,j})$ is specified by:

1) the number of fill rows (M) and columns (N);
2) the series of widths $\{W_i\}_{i=1,\ldots,N}$ and lengths $\{L_j\}_{j=1,\ldots,M}$ of fills, such that each is within the bound $[W_l, W_u]$;
3) the series of horizontal and vertical spacings between fills, $\{S_{x,i}\}_{i=1,\ldots,N}$ and $\{S_{y,j}\}_{j=1,\ldots,M}$, where $\{S_{x,i}\}_{i=2,\ldots,N-1}$ are at least $\overline{S_l}$ and those between metal and dummy fill $\{S_{x,i}\}_{i=1,N-1}$ are at least $\overline{S_d}$.

Foundaries require the effective metal density ρ_{Cu} to be achieved throughout the die. We express the actual amount of metal fill needed between interconnect in terms of local metal density ρ_f.

Definition 1: Effective metal density ρ_{Cu} – the proportion of the area in a planarization window [3] that all metal features (interconnect + dummy fill metal) occupies, which is usually a hard requirement from the foundry [2], [19].

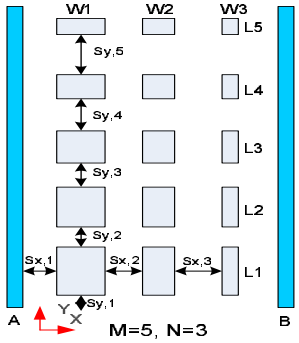


Fig. 1. Fill pattern definition.

Definition 2: Local metal density ρ_f – the proportion of the oxide area between two neighboring interconnects that dummy fill metal occupies, which is found by either rule-based method in the industry or by the recently proposed model-based method [20] to achieve ρ_{Cu}[1].

With the above definitions, we can therefore derive the width of length of the dummy metal fill features by $\rho_f \cdot S_{A,B} = \sum_i W_i \cdot \sum_j L_j = W_b \cdot L_b$, where $S_{A,B}$ is the space between interconnect A and B, and W_b and L_b are the total fill width budget and length budget, respectively. Finding a valid fill pattern is equivalent to distributing the budgets of W_b, L_b, $S_{x,b}$, and $S_{y,b}$ among their respective series $\{W_i\}$, $\{L_j\}$, $\{S_{x,i}\}$, and $\{S_{y,j}\}$, which also determines M and N.

To understand the impact of different dummy fill pattern on variation of parasitic capacitance, we explore many different fill patterns, each of which satisfies the aforementioned DRC restrictions and its metal fill target. Figure 2 shows the $x-$cross-sectional views of three different fill patterns. We plot $f(z)$ as the width of each dummy fill feature against the position of the space that we want to fill. By constraining the area under $f(z)$ to the budget width W_b, we try numerous shapes of $f(z)$, among which we discard those that violate the DRC restrictions. We apply similar enumeration to the other side of the cross-section (i.e. y-direction) to obtain a complete exploration of fill patterns.

B. Fill Pattern Induced Variation

We consider the coupling capacitance (C_c) between active interconnects and total capacitance (C_s) of an interconnect, which is the sum of C_c, area capacitance and fringe capacitance. Inserting dummy fill between signal wires effectively brings the signal wires closer together, where they couple stronger with each other through the floating metal. However, the floating metal has coupling to above and below layers which may act as alternative paths for coupling currents between these signal lines. We use QuickCap [21] to extract the *effective* C_c, which gives the capacitance that achieves the same coupling effect by replacing the dummy fill structure with a simple capacitor between the signal lines. The on-chip

[1]Although the cited reference refers to a shallow-trench isolation process, the underlying polishing process resembles that in copper CMP.

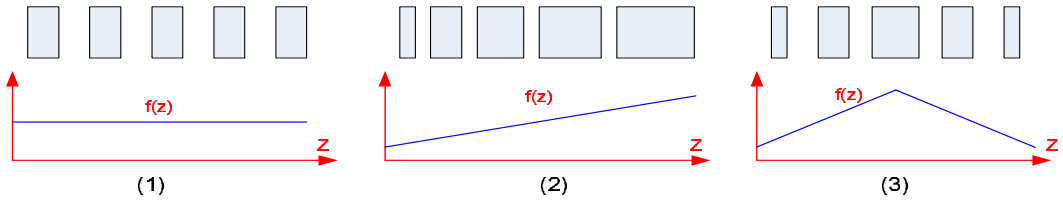


Fig. 2. Examples of cross-sectional profile function $f(z)$.

interconnect is modeled as a stripline where the interconnect layer is sandwitched between two ground planes. We study global interconnects in the ITRS $65nm$ technology node [15] with various fill pattern that we generate from Section II-A. For each layout, the interconnect width is set to the minimum width while the spacing between two active interconnects varies from $3\times$ to $10\times$ minimum spacing [2]. Interconnect length is $1000\mu m$ for all layouts. We extract and compare the nominal (i.e. wo/dummy fill) and the CMP-impacted (i.e. w/dummy fill) C_c and C_s.

Figure 3 plots the variation of coupling capacitance C_c due to dummy fill insertion. We examine the cases where $\rho_f = 0.3, 0.5$ and 0.7. We vary the spacing between interconnects from $3\times$ to $10\times$ minimum spacing. The curves with diamond symbols are the nominal C_c without fill insertion. The curves with square symbols represent the mean values of the effective C_c under dummy fill insertion. The ranges of C_c due to different dummy fill patterns are represented by their respective maximum and minimum values among all the fill patterns, which are shown by the vertical bar on each square symbol. We observe that (1) different fill patterns result in coupling capacitance variation, which can be up to 10% at $3\times$ spacing and more than double at $6\times$ spacing; that (2) fill insertion always increases the coupling capacitance when compared to the nominal case (i.e. wo/dummy fill); and that (3) the gap between the nominal C_c curve and the mean value C_c curve increases with metal fill density ρ_f.

To study the relative importance of the coupling capacitance variation versus the total capacitance variation due to fill insertion, in Figure 4 we plot the percentage of C_c over C_s with respect to different local metal densities ρ_f (0.1 to 0.7) between active interconnects with $3\times$, $5\times$ and $10\times$ minimum spacing. The gap between the maximum and minimum percentage curves shows the potential variation due to fill insertion. We see that (1) fill insertion increases the percentage of C_c/C_s ratio in all different metal densities and interconnect spacing; and that (2) variation of C_c/C_s increases with metal spacing and slightly with metal density. The exact impact of these variation on the actual delay remains to be seen, however, since large metal spacing undermines the significance of C_c and therefore its variation.

C. Dishing and Erosion Induced Variation

Figure 5 illustrates dishing and erosion phenomena due to CMP [22]. Step height is defined as the difference of height between different area on the surface of the wafer. Dishing is a special case of step height that it specifically refers to the difference between the height of the copper in the trench, which defines the metal interconnect and that of the dielectric in the space surrounding the trenches. Erosion is defined as the difference between the dielectric thickness before CMP and that after CMP. The sum of dishing and erosion is the total loss of metal thickness.

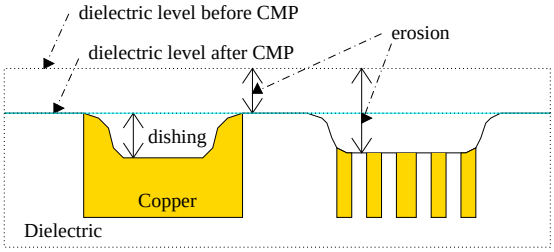


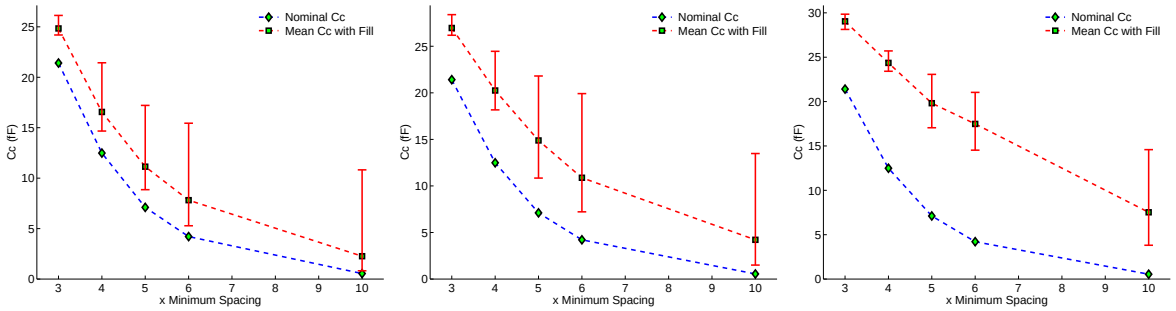
Fig. 5. Dishing and Erosion in Copper CMP.

We employ the dishing and erosion model in [22], which is a closed-form solution of a differential equation set, to calculate post-multi-step CMP interconnect geometries[3]. During interconnect formation, trenches are etched on the oxide, followed by barrier deposition on the etched surface to prevent copper diffusion into the oxide. Then a thick layer of copper are deposited on the wafer. CMP removes both the bulk copper above the trenches and the barrier on the area between the trenches. The multi-step model consists of three steps which correspond to three different polishing pads. We assume that Step 1 eliminates all the local step heights before touching the raised area, and is therefore irrelevant to the modeling of dishing and erosion. We also assume that Step 2 completely removes all the remaining copper so that there is no dishing and erosion at the moment when the polishing pad reaches the barrier on the raised area. We use the same assumption as in Gbondo-Tugbawa's model [22] that the polishing time of Step 2 after reaching the barrier layer is $20s$ and that of the entire Step 3 is $65s$.

To show the potential impact of dishing and erosion on signal wire's parasitics, we apply the model and measure the resistance and capacitance of the middle interconnect in

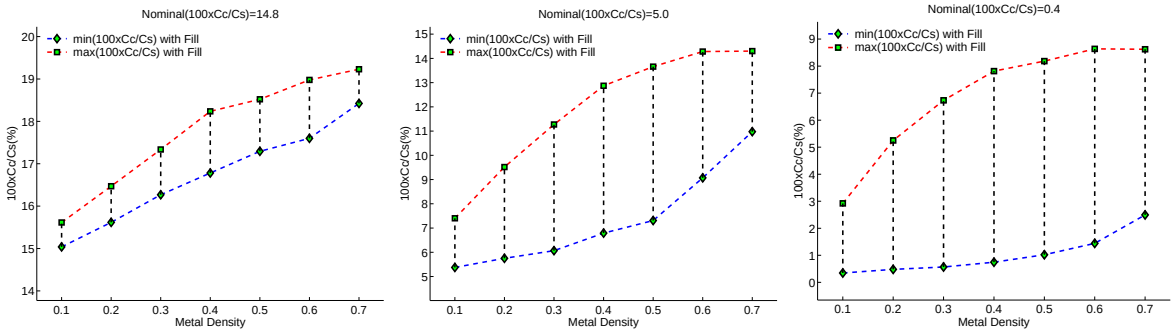
[2]To allow fill insertion between active interconnect without DRC violation, the minimum spacing between active interconnect is $3\times$ minimum spacing rule.

[3]This is the only open source of copper CMP model with parameters published in the literature. This model does not necessarily couple with the lithographic process which defines our assumed device and interconnect characteristics. This CMP model only means to provide an input source of CMP variability. Our subsequent process variation aware methodologies do not necessarily depend on this assumed CMP model.



(a) ρ_f=0.3 (b) ρ_f=0.5 (c) ρ_f=0.7

Fig. 3. Distribution of coupling capacitance C_c.



(a) 3× minimum spacing (b) 5× minimum spacing (c) 10× minimum spacing

Fig. 4. The percentage of C_c over C_s for different local metal density requirement ρ_f.

a strip-line structure [4]. Table I shows the RC parasitics for a 1000μm long global interconnect bus structure under the 65nm technology node. R_0 is the resistance computed from the geometry values obtained from ITRS specifications, i.e., dishing and erosion effects are not taken into account. R_f is the resistance after fill insertion which fulfills 50% metal density requirement (i.e. $\rho_{Cu} = 0.5$). Based on this, we include the metal loss due to dishing and erosion when computing R_f. From Table I, we can see that resistance variation due to dishing and erosion is significant, and that resistance is always increasing, potentially by more than 30%. As width increases, the resistance variation becomes increasingly severe. For example, when conductor width increases from 0.24μm to 2.61μm, the resistance variation increases from 29% to 31%.

All capacitance values in Table I are extracted using QuickCap [21]. $C_{c,0}$ and $C_{s,0}$ are the coupling and total capacitance without considering fill insertion or dishing and erosion effects. $C_{c,1}$ and $C_{s,1}$ are the coupling and total capacitance for the same assumed structure as in Section II-B, taking geometry variations due to dishing and erosion (but no fill insertion) into account. Finally, $C_{c,f}$ and $C_{s,f}$ are the effective coupling and total capacitance when effects due to dummy fill, dishing and erosion are all taken into consideration. The percentages in the brackets show the relative changes from values which do not consider any CMP effect (columns 3, 5 and 6). From Table I, we observe that dishing and erosion alone merely have any impact on capacitance. In light of these results, we do not consider dishing and erosion effects on capacitance in our subsequent discussion.

[4]We are interested in global wires that are $\leq 2\mu m$ wide as predicted in 65nm ITRS process [15]. Wider lines like power grids, which are out of the scope of this work, require slotting to prevent "lift-off" in CMP [23].

D. Table-based fill pattern look-up and RC Model

Based upon our study of CMP-induced RC parasitic variations, we tabulate the extracted capacitance in a table indexed by active interconnect width, spacing and local metal density under an optimized fill pattern. Note that varying metal spacing affects the local metal density requirement in the space. During interconnect optimization, each enumerated spacing option requires an appropriate adjustment to the amount of required local metal density. Therefore the fill pattern and RC of all combinations of spacing and local metal density have to be recorded in the table to accommodate any arbitrary spacing and adjusted local metal density. Moreover, as different fill patterns under the same local metal density result in different capacitance values as shown in Section II-B, each table entry only saves the fill pattern and the resulting capacitance under the *best* fill pattern, which gives the minimum C_c among all patterns. We use the briefly discussed model in Section II-C to compute the resistance under dishing and erosion effects. In the following, we denote the resulting RC models as *CMP-aware* RC parasitic models. In contrast, interconnect parasitics without consideration of fill pattern insertion, dishing or erosion effects is called *CMP-oblivious* RC model.

III. CMP-aware Buffer Insertion and Wire Sizing

In this section, we study the problem of simultaneous buffer insertion and wire sizing (SBW) to examine the impact of

TABLE I

RC PARASITIC COMPARISON FOR $65nm$ GLOBAL INTERCONNECTS.

Width	Space	wo/CMP	w/CMP	wo/CMP		Dishing/Erosion		Fill+Dishing/Erosion	
μm	μm	$R_0(\Omega)$	$R_f(\Omega)$	$C_{c,0}$	$C_{s,0}$	$C_{c,1}$ (Δ%)	$C_{s,1}$ (Δ%)	$C_{c,f}$ (Δ%)	$C_{s,f}$ (Δ%)
0.24	0.95	186	239 (28.7%)	25.16	286.06	24.48 (-2.63%)	285.12 (-0.33%)	33.48 (33.06%)	285.77 (-0.11%)
2.61	0.95	16.9	22.1 (30.6%)	26.06	966.82	25.06 (-3.78%)	964.98 (-0.19%)	32.90 (26.33%)	953.71 (-1.35%)
0.24	1.43	186	239 (28.8%)	8.35	283.75	8.57 (2.54%)	283.39 (-0.13%)	20.27 (142.71%)	289.12 (1.88%)
2.61	1.43	16.9	22.1 (30.9%)	8.68	956.84	8.32 (-4.35%)	954.04 (-0.29%)	21.02 (141.81%)	960.34 (0.36%)

CMP on interconnect design[5]. We propose an extension to the popular dynamic programming-based SBW algorithm [16], [17] to solve the SBW and the *fill insertion* problem simultaneously, and we denote it as $SBWF$. In contrast, current designers first solves the SBW problem with CMP-oblivious RC, and then hand the design off for a post-layout processing step. This second step inserts dummy fill metal into the wire space to satisfy the local metal density requirement defined in Section II-A. We use this two-step approach as our baseline for comparison, which is denoted as $SBW + Fill$.

A. Problem Formulation

Consider a routing tree $T(V, E)$, where V consists of a source node n_{src}, sink nodes $\{n_s\}$, and Steiner points $\{n_p\}$, and E is the set of directed edges (wires) that connect the nodes in V. The $SBWF$ problem is to find an assignment of buffer insertion, buffer sizing, wire sizing, and dummy fill insertion, such that the *required arrival time* (RAT) is maximized at n_{src}, subject to (1) the slew rate constraint η at all n_s and buffers' inputs; and (2) the effective metal density requirement ρ_{Cu} for CMP planarization.

We characterize the source n_{src} by a driving resistance R_{src}; each sink n_s by a loading capacitance C_n^s and a required arrival time RAT_s. We associate each edge $e_{i,j}$ with two center-to-edge wire widths w_1 and w_2 as illustrated in Fig. 6. To respect the design rules, we restrict $w_k \in \{0.5 \cdot \breve{w}, 1.5 \cdot \breve{w}, ..., s_k - \breve{w}\}$, where $k = 1, 2$, $\breve{w}$ is the minimum wire width allowed at the global metal level and s_k is the spacing from the center line to the edges of its two nearest neighboring wires. For every edge $e_{i,j}$, we define the potential buffer insertion site at the point closest to the node v_i. The buffer receives input from node v_i and drives edge $e_{i,j}$ and the downstream subtree rooted at node v_j. We express the size of buffer S_{buf} in discrete multiples of the minimum-sized buffers. All buffers are 2-stage cascaded inverters.

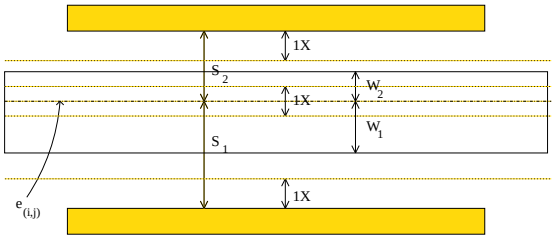


Fig. 6. Illustration of asymmetric wire sizing.

[5]The asymmetric wire sizing problem was first proposed in [17] without slew rate constraints, which does not consider the CMP-induced variation neither.

B. Slew Rate Constrained SBW Algorithm

The slew rate constrained SBW algorithm largely follows the dynamic programming (*DP*) framework of [24], where buffer insertion and asymmetric wire sizing is determined in a bottom-up (sink-to-source), recursive fashion. To obtain the optimal solution at the source in a deterministic buffer insertion regime, partial solutions sol_n at node n (i.e. partial buffer placement and wire width assignment for the subtree rooted at node n) must keep track of the downstream capacitance C_n and the arrival time RAT_n associated with sol_n. The arrival time RAT_n at node n is defined by

$$RAT_n = \min_{n_i \in \{n_s\}} (RAT_i - d(n_i, n))$$

where $d(n_i, n)$ is the delay from the node n_i to node n, RAT_i is the RAT at node n_i and $\{n_s\}$ is the set of all sink nodes. We use the first order Elmore delay model and slew rate model [25] in our current implementation due to their high fidelity over real design metrics. We update the RAT_n of each solution sol_n at node n by

$$\begin{aligned} RAT_n &= RAT_n^{old} - r_{n,v} \cdot C_n - 0.5 \cdot r_{n,v} \cdot c_{n,v} \\ &\quad - d_{buf} - R_{eff} \cdot (L_n + c_{n,v}) \end{aligned} \quad (1)$$

where $r_{n,v}$ and $c_{n,v}$ are the resistance and capacitance of edge $e_{n,v}$ respectively; d_{buf} and R_{eff} are buffer intrinsic delay and output resistance, respectively, which are both functions of buffer size S_{buf}. We use Bakoglu's slew rate metric [25] given by $\ln 9 \cdot d_T^n$, where d_T^n is the maximum delay from the output of buffer at node n to the inputs of other immediate buffers or the sinks n_s in the subtree T_n rooted at n. Note that the above can be replaced by other more accurate delay [26] and slew [27] metrics which consider higher order moments.

The overall time complexity of the $SBW+Fill$ algorithm is $O(|V|^2 \cdot c_{max} \cdot (|S_{buf}|+|S_{wire}|))$, where $|S_{wire}|$ is the number of available choices of wire widths, $|V|$ is the number of nodes in the interconnect tree, c_{max} is the maximum possible capacitance value carried by any partial solutions and $|S_{buf}|$ is the number of possible sizes for buffers [24]. The complexity depends on c_{max} if we prune *inferior* solutions in SOL_n for each node n. A solution sol_1 is said to be inferior to (or dominated by) another solution sol_2 if $C_{sol}^1 \geq C_{sol}^2$ and $RAT_{sol}^1 \leq RAT_{sol}^2$. With wire sizing, c_{max} can go exponential but is in fact upper-bounded. The slew rate bound virtually limits the distance that a wire can run without buffering, which therefore limits the maximum downstream capacitance c_{max} seen from any node.

C. Extension to SBW and SBWF

We extend the $SBW+Fill$ algorithm to solve the $SBWF$ problem, and such an approach is denoted as $SBWF$ when-

ever there is no ambiguity. $SBWF$ uses the CMP-aware table-based fill pattern look-up RC model from Section II-D for delay and slew rate calculation while solving the slew rate constrained SBW problem. For every edge $e_{i,j}$, we specify two local dummy fill density requirements ρ_f^1 and ρ_f^2 at minimum wire width in order to satisfy the effective metal density target ρ_{Cu}, as defined in Section II-A. The required ρ_f^1 and ρ_f^2 can be determined from algorithms such as [20]. Note that increasing wire width decreases the amount of dummy fill metal needed between wire space, which necessitates the adjustment to the required local metal densities. At each enumeration of wire spacing option, the $SBWF$ algorithm makes adjustment to ρ_f^1 and ρ_f^2, which are used with the corresponding wire widths and spacing to look up the CMP-aware fill pattern and RC table for the optimized fill pattern and the capacitance values. The algorithm collects all wire sizing and spacing options, each with timing evaluated under an optimized fill pattern. These options are then pruned against each other as in the $SBW+Fill$ algorithm to remove inferior solutions.

Note that the proposed extension is orthogonal to the baseline dynamic programming-based framework. This extension brings in the necessary book-keeping to maintain wire width, spacing and local metal density requirement, which supports calculation of dishing/erosion and the table-lookup methodology in Section II-D for optimum dummy fill patterning. This same extension can be applied to many other variants of such dynamic-programming framework, which include cost/power consideration and speed-up techniques [24], [28], [29] [6].

D. Experiment

TABLE II
EXPERIMENTAL SETTINGS

technology	ITRS $65nm$ [15]
interconnect	global interconnect layer
delay model	Elmore delay, π-model for interconnect
slew model	Bakoglu's first order metric [25]
power model	dynamic and short-circuit, from SPICE
device	BSIM 4 [30]
R_{src}	100Ω
L_{sink} & RAT_{sink}	$10fF$ & $0ps$ $\forall t_i$
slew bound $\overline{\eta}$	$100ps$ (under CMP-perturbed RC)
metal density	0∼0.8 (local fill), 0.5 (effective)
S_{buf}	20, 40, 80, 120 (x min size)
s_1, s_2	1.5∼5.5 (x min width)
w_1, w_2	0.5, 2.5, 4.5 (x min width)
segment length	500 μm
test cases	r1∼r5: clock trees from [31] s1∼s10: random Steiner trees

Table II shows the experimental settings used in this paper. We choose typical buffer sizes and wire sizes that are normally used in real designs. Since there is no physical layout information in the original test cases obtained from [31], we randomly generate the neighboring wire spacing data and the local metal density requirements for each interconnect in all test cases. We perform experiments on an Intel Xeon 1.9Ghz Linux workstation with 2Gb of memory.

In order to make a conservative review on the effect of the $SBWF$ methodology, we assume the best possible scenario that designers can account for the effect of CMP in the baseline $SBW+Fill$ approach. We first assume that designer makes the best effort to introduce the minimum over-design to the slew rate constraint η in order to meet the actual slew rate constraint under CMP-aware parasitics and inserted dummy fill. The first step of the $SBW+Fill$ algorithm always underestimates the slew rate as it does not consider CMP-induced variation on RC. The *over-constrain rate*, κ, is defined as the ratio of the over-constrained slew rate to the actual slew rate constraint. To minimize over-design, we find κ via an expensive binary search, in which each iteration involves an execution of $SBW+Fill$. In contrast, the proposed $SBWF$ algorithm uses the CMP-aware RC parasitics while solving SBW problem. Therefore, it finds an optimum solution that satisfies the slew rate constraints without repetition. In our current setting, we use $\kappa = 0.84$ for $SBW+Fill$, which gives maximum slew rates that satisfy the slew rate bound η in all test cases. Our second conservative assumption is that the post-layout processing step in $SBW+Fill$ does make an effort to choose the fill pattern that minimizes the increase in capacitance. In contrast, most works in the literature so far only considers one single pattern [5], [6], [7], which does not necessarily minimize the impact of fill insertion on parasitics.

Table III compares the experimental results from $SBW+Fill$ and $SBWF$. The objective in both $SBW+Fill$ and $SBWF$ is to optimize the required arrival time at the source. We verify both the $SBW+Fill$ design and the $SBWF$ design under the CMP-aware parasitic model. A solution with larger RAT implies smaller delay and is therefore more preferable. Comparing $SBW+Fill$ against $SBWF$ (relative change of values shown in the brackets), we see that $SBWF$ consistently achieves larger RAT for all test cases and the average increase is 1.6%. Accounting for CMP variation by $SBWF$ comes at a cost of having an average of 5.0% increase in wiring area, although buffer area drops by 4.9% on average. Over-constraining the slew rate in $SBW+Fill$ causes excessive buffer insertion in $SBW+Fill$ and leads to larger total area of buffers over $SBWF$, which does not require over-constraining the slew rate. Reduced buffer area in $SBWF$ also leads to 3.0% reduction of power on average over $SBW+Fill$. This is in stark contrast to cost-aware [32] and power-optimal wire-sizing and buffering [24], [29] where the trade-offs between timing optimality and costs (eg. buffer/wiring area, power) are much stronger. We also notice that the runtime also slightly increases from $SBW+Fill$ to $SBWF$ due to the evaluation of dishing and erosion model. However, note that the runtime reported in $SBW+Fill$ is for a single run; in practice designers have to perform multiple runs in order to minimize over-designing the slew rate constraint η as explained earlier, which may therefore cost much more run-time. From all of these results, we see that designs considering CMP impacts improve timing over nominal design, and is not prohibitively expensive in run-time as long as effects of CMP on parasitics is modeled accurately and efficiently.

[6]This work focuses on the potential impact of bringing CMP and random process variation awareness during design time. There exists a wealth of discussions in the literature about design with cost constraints and algorithmic speedup, which we consider as orthogonal issues and are therefore not discussed in length for conciseness and focus.

TABLE III
EXPERIMENTAL RESULT FROM $SBW + Fill$ AND $SBWF$ VERIFIED UNDER CMP-PERTURBED RC.

			$SBW + Fill$ ($\kappa = 0.84$)					$SBWF$				
test-case	wire length (m)	# sink	wire area (mm^2)	buffer area (x min)	RAT (ps)	power (pJ)	run-time (s)	wire area (mm^2) (Δ%)	buffer area (x min) (Δ%)	RAT (ps) (Δ%)	power (pJ) (Δ%)	run-time (s)
s1	0.03	19	0.10	2920	-1007	22	0	0.10 (0.9%)	2680 (-8.2%)	-1001 (0.6%)	21 (-6.0%)	0
s2	0.04	29	0.11	3420	-1175	26	0	0.12 (2.0%)	3140 (-8.2%)	-1133 (3.6%)	25 (-5.7%)	1
s3	0.05	49	0.14	4380	-1589	33	1	0.15 (9.5%)	4360 (-0.5%)	-1567 (1.3%)	34 (0.9%)	1
s4	0.07	99	0.18	6180	-1386	47	2	0.19 (8.0%)	6060 (-1.9%)	-1380 (0.4%)	46 (-0.5%)	2
s5	0.10	199	0.26	8820	-2436	67	4	0.27 (5.3%)	8500 (-3.6%)	-2409 (1.1%)	66 (-2.1%)	5
s6	0.13	299	0.31	11720	-2294	88	7	0.33 (5.9%)	11020 (-6.0%)	-2235 (2.6%)	84 (-3.9%)	8
s7	0.16	499	0.38	15220	-3794	113	16	0.40 (5.1%)	14520 (-4.6%)	-3787 (0.2%)	110 (-3.0%)	22
s8	0.19	699	0.43	18320	-3170	136	37	0.45 (4.7%)	17260 (-5.8%)	-3141 (0.9%)	131 (-4.0%)	47
s9	0.21	799	0.47	19700	-2967	147	34	0.49 (3.0%)	18580 (-5.7%)	-2867 (3.4%)	141 (-4.0%)	38
s10	0.22	899	0.51	21000	-2830	157	57	0.53 (3.7%)	20580 (-2.0%)	-2782 (1.7%)	155 (-1.1%)	69
r1	1.32	267	3.79	110000	-4955	838	69	3.97 (4.8%)	104180 (-5.3%)	-4844 (2.3%)	811 (-3.2%)	27
r2	2.60	598	7.32	212760	-6148	1625	0	7.74 (5.7%)	202840 (-4.7%)	-6031 (1.9%)	1582 (-2.6%)	71
r3	3.37	862	9.33	275760	-7358	2103	102	9.89 (6.1%)	261180 (-5.3%)	-7297 (0.8%)	2038 (-3.1%)	91
r4	6.81	1903	18.90	554260	-10748	4233	170	19.83 (4.9%)	522980 (-5.6%)	-10592 (1.4%)	4086 (-3.5%)	175
r5	10.20	3101	28.16	823100	-11984	6297	256	29.48 (4.7%)	777920 (-5.5%)	-11804 (1.5%)	6084 (-3.4%)	271
								(5.0%)	(-4.9%)	(1.6%)	(-3.0%)	

IV. YIELD-DRIVEN SBW

A. Leff Variation

One of the most important process uncertainty that affects circuit performance is the random variation of devices' effective channel lengths (L_{eff}) [33], [4]. The variation of L_{eff} manifests itself in changing various device characteristics, e.g., input capacitance C_{in}, effective output resistance R_{eff}, and intrinsic delay d_{buf}. To understand the effect of L_{eff} variation on the delay, we show two sets of measurements on buffers using SPICE [34]. We model L_{eff} with a Gaussian distribution Δ_L with its mean value $\overline{L_{eff}}$ equals its nominal value and the standard deviation $\widehat{L_{eff}}$ equals $5\% \cdot \overline{L_{eff}}$[7].

The first set studies the sensitivity of the effective input capacitance of buffers to L_{eff} variation. We set the total L_{eff} of the transistors at the input of an inverter to an unlikely large value and show that the increase in the input capacitance as a consequence is small. We size the PMOS and the NMOS of the buffers to the ratio of 2:1 for symmetric rise and fall. Therefore the total input capacitance is a function of $L^{tot}_{eff} = L^{n}_{eff} + 2 \cdot L^{p}_{eff}$, where L^{n}_{eff} and L^{p}_{eff} are the L_{eff} of the NMOS and PMOS transistors respectively. Since L^{n}_{eff} and L^{p}_{eff} are assumed to be independent Gaussian random variables having the same Gaussian distribution Δ_L, L^{tot}_{eff} is also a Gaussian random variable with mean $3 \cdot \overline{L_{eff}}$ and standard deviation $\sqrt{5} \cdot \widehat{L_{eff}}$. The 99% percentile of L^{tot}_{eff} is given by

$$L^{\alpha}_{eff} = \sqrt{5} \cdot CDF^{-1}_{gaussian}(0.99) \cdot \widehat{L_{eff}} + 3 \cdot \overline{L_{eff}} \qquad (2)$$

where $CDF^{-1}_{gaussian}(x)$ is the inverse Gaussian cumulative distribution function. Such L^{α}_{eff} happens with a probability of 1%. We first employ the simplified model from [35] that the transistor gate capacitance C_g operated in saturation region is given by

$$C_g = C_{ox} \cdot W_d \cdot \left(\frac{2}{3} \cdot L_{eff} + 2 \cdot L_{int} \right) \qquad (3)$$

[7]ITRS [15] allows a budget of 10% from the nominal value for 3× standard deviations of *random* variation (excluding all systematic variation like across-chip line-width variations). Other related works in the literature [11], [13] assumes this budget to be 15–30%.

where C_{ox} is the gate oxide capacitance per unit area, W_d is the drawn transistor width and L_{int} is the length of lateral diffusion. According to the default values in the BSIM 4 65nm device model [30], we set $\overline{L_{eff}} = 33 \cdot 3 = 99nm$ and $L_{int} = 16 \cdot 3 = 48nm$. We apply Equation 2 to obtain $L^{\alpha}_{eff} = (99 + 8.58)nm$. Using Equation (3), we find that the capacitance increases by only 3.5% when L^{tot}_{eff} increases from $3 \cdot \overline{L_{eff}}$ to L^{α}_{eff}. To verify this, we increases the L^{tot}_{eff} of the transistors to from $3 \cdot \overline{L_{eff}}$ to L^{α}_{eff} in SPICE, from which we find that the measured effective input capacitance only increases by less than 3% for all sizes of buffers in our experiment. This is equivalent to a negligibly small $4.1fF$ increase in the input capacitance for our largest (120×) buffer. Therefore, we conclude that the effective input capacitance is rather insensitive to random L_{eff} variation and we treat it as constant in our work without much loss of accuracy.

The second set of measurement shows that L_{eff} variation has a much larger contribution to the variation of the effective output resistance R_{eff} and the intrinsic delay d_{buf}. To account for the dependency of R_{eff} and d_{buf} on the common variation source of L_{eff}, we model the variation in R_{eff} and d_{buf} using a joint distribution, which can be obtained from Monte Carlo simulation using SPICE in the inner loop. We collect the covariance matrix as a statistical metric to observe the variability of R_{eff} and d_{buf} under L_{eff} variation, which is given by

$$M = \begin{bmatrix} \zeta_{R,R} & \zeta_{R,d} \\ \zeta_{R,d} & \zeta_{d,d} \end{bmatrix} = \begin{bmatrix} 771 & 26.5 \\ 26.5 & 14.0 \end{bmatrix} \qquad (4)$$

Equation (4) shows the covariance matrix M of a 20× buffer, where $\zeta_{x,y}$ is the covariance of $x, y \in \{R, d\}$, and subscripts R and d refer to R_{eff} and d_{buf} respectively. The standard deviations of R_{eff} ($\sqrt{\zeta_{R,R}}$) and d_{buf} ($\sqrt{\zeta_{d,d}}$) are about 15% and 6% of their mean values respectively. This shows that R_{eff} and d_{buf} can deviate significantly from their respective nominal values due to L_{eff} variation. Moreover, the large covariance between R_{eff} and d_{buf} (i.e. $\zeta_{R,d}$) also demonstrates that R_{eff} and d_{buf} are positively correlated, which means that an occurrence of positive (negative) variation in R_{eff} from the nominal value is likely to be accompanied

by a positive (negative) variation in d_{buf} [36]. Therefore, we characterize R_{eff} and d_{buf} using a joint probability density function (JPDF) $f_{R,d}(R_{eff}, d_{buf})$, which accurately models the occurrence probability of the (R_{eff}, d_{buf}) pair, and can be computed by Monte Carlo simulation. Let us consider the delay of a buffer driving a capacitance C_L, which is given by

$$d = C_L \cdot R_{eff} + d_{buf} \tag{5}$$

in the deterministic case. We express d_{buf} in terms of d and R_{eff} using Equation (5), substitute this into $f_{R,d}(R_{eff}, d_{buf})$ and then integrate $f_{R,d}$ over R_{eff} to obtain the probability density function (PDF) of the loaded buffer delay, which is given by

$$f_d(C_L, d) = \int_{-\infty}^{\infty} f_{R,d}(R_{eff}, d - C_L \cdot R_{eff}) dR_{eff} \tag{6}$$

B. vSBWF Problem Formulation

We call the $SBWF$ problem considering L_{eff} random variation as $vSBWF$. Owing to the statistical nature of $vSBWF$, we treat the RAT at each node as a random variable in $vSBWF$. The objective of $vSBWF$ becomes maximizing a routing tree's statistical *timing yield*. The timing yield is defined as

$$\Upsilon = P(RAT_s \geq \Gamma_\Upsilon) \tag{7}$$

where Γ_Υ is the *yield cut-off point* at $\Upsilon \cdot 100\%$. This equation essentially says that the probability of RAT_s at the source n_{src} being at least Γ_Υ is Υ.

As an important step towards runtime control in any contemporary buffer insertion algorithms [16], [24], [37], pruning using any nominal value such as mean or worst-case values is deficient when timing is subject to random variations. To illustrate, suppose that we are evaluating the merging node n_m where two identical subtrees join. Two buffering solutions from each subtree are propagated to n_m, which have discrete delay distributions of $sol_A = (50\% \cdot 200ps, 50\% \cdot 300ps)$ and $sol_B = (80\% \cdot 242.5, 20\% \cdot 305ps)$ respectively. Let us assume for now that these two solutions do not differ in other metrics (for example, downstream capacitance). We are interested in finding out how each of the pruning metrics (mean, worst-case, statistical) picks their solution among all produced at n_m. By enumeration, we obtain the following solutions at n_m:

1) $sol_\alpha = max(sol_A, sol_A) = (25\% \cdot 200ps, 75\% \cdot 300ps)$: mean=$250ps$, worst-case=$300ps$;
2) $sol_\beta = max(sol_A, sol_B) = (40\% \cdot 242.5ps, 40\% \cdot 300ps, 20\% \cdot 305)$: mean=$255ps$, worst-case=$305ps$; and
3) $sol_\gamma = max(sol_B, sol_B) = (64\% \cdot 242.5ps, 36\% \cdot 305ps)$: mean=$255ps$, worst-case=$305ps$.

Among these solutions, it is clear that all pruning strategies based on nominal values (mean, worst-case) prefer sol_α as it has the smallest delay in both metrics. However, we notice that both solutions sol_α and sol_β have a much higher proportion of $\geq 300ps$ instances than sol_γ. Therefore the merged solution sol_γ is considered a statistically superior solution, which pruning using statistical timing shall be able to identify. It is also possible that statistical pruning may help improve the mean timing of the optimized statistical distribution. For example, the statistical timing distribution of sol_γ has a mean of $265ps$, while those of sol_α and sol_β have means of $275ps$ and $278ps$ respectively.

There are two challenges in solving the $vSBWF$ problem, which are (1) how to efficiently represent and compute RAT that is not a deterministic value but a random variable; and (2) how to define pruning rules that remove statistically inferior solutions while keeping the algorithm tractable. We address these challenges in the following sections.

C. Representing and Computing RAT

To solve $vSBWF$ via the same DP framework as shown in Section III-B, we have to replace the deterministic RAT computation with its statistical counterpart. Since a random variable can be completely characterized by its cumulative distribution function (CDF), we choose to base all statistical computation in terms of RAT^i_{sol}'s CDF in any solution sol_i[8]. We represent CDF in the form of *piecewise-linear curve* (PWL) as in [38]. Representing CDF in the form of PWL has the advantage that operations on a complicated function become a series of operations on ramp functions, which often have closed-form solutions. For example, using PWL reduces statistical addition and maximum operations to convolution of steps and ramps and multiplication of ramps respectively, both of which have closed-form quadratic solutions. [38] has depicted operations for Elmore delay calculation and have provided closed-form quadratic formulae. After all operations on these ramp and step functions, adding the resulting quadratic curves forms a "piece-wise quadratic curve". This curve is then "sampled" at the pre-defined percentile to produce the final CDF in the PWL.

The application of PWL is not limited to the first order delay and slew models used in this work. Our immediate observation is that the PWL model can also apply to at least second-order models. For example, delay and slew rate metrics in [26] and [27] require the computation of the second moment. The second moment computation involves multiplication of two independent random variables and squaring of random variables, both of which can be expressed analytically. By modeling CDFs with PWL curves, we can find the analytical solution for each PWL component and proceed with the same methodology to compute CDFs.

D. Efficient Pruning in vSBWF

A useful pruning rule must (1) not discard any partial solution that may lead to the optimal solution sol_{opt} at the source n_{src}; and (2) keep the growth of number of solutions polynomial with respect to the tree size. We propose an efficient *Yield Cut-off Dominance*-pruning heuristic. This heuristic provably keep the solution growth at a linear rate. Although we cannot prove analytically that such heuristic preseves the optimal solution, we experimentally shows that the optimality of the solution's timing is comparable to the

[8]In our implementation, we consider the negative of RAT_{sol}, i.e. $-RAT_{sol}$, for the sake of simpler mathematical manipulation. This converts all "min" operations at branch merging points into "max" operations, which are equivalent to simple multiplications of CDFs in the statistical domain.

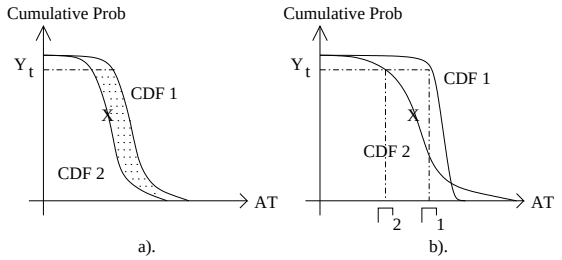


Fig. 7. CDF of RATs to illustrate the definition of timing yield, yield cut-off point and and pruning rules

CDF Dominance-pruning rule, which is provably optimal but leads to exponential runtime.

1) CDF Dominance: Figure 7(a) shows the *CDF Dominance* relationship. Area CDF 1 is completely on the right-hand-side of CDF 2. As a result CDF 2 is said to be *dominated* and is discarded under this relationship. To see why pruning under this relationship preserves optimality, we show mathematically that $\widetilde{CDF}_1(x)$ and $\widetilde{CDF}_2(x)$ computed from $CDF_1(x)$ and $CDF_2(x)$ in delay and slew rate computations has the same relative superiority as $CDF_1(x)$ and $CDF_2(x)$. Suppose that $CDF_1(x) \geq CDF_2(x)\ \forall x$. Statistical maximum corresponds to CDF multiplication, which is obtained by

$$\begin{aligned} \widetilde{CDF}_1(x) &= CDF_1(x) \cdot CDF(x) \\ &\geq CDF_2(x) \cdot CDF(x) = \widetilde{CDF}_2(x) \end{aligned} \quad (8)$$

since $CDF(x)$ is non-negative. Statistical addition corresponds to the convolution of CDF and PDF, which is

$$\widetilde{CDF}_i(x) = \int_{-\infty}^{\infty} CDF_i(\tau) \cdot PDF(x-\tau)d\tau \quad (9)$$

where $i = 1, 2$ and $PDF(x) = \frac{d}{dx}CDF(x)$. Since $CDF_1(x) - CDF_2(x) \geq 0$ and $PDF(x) \geq 0 \quad \forall x$, we have

$$\begin{aligned} &\int_{-\infty}^{\infty} (CDF_1(\tau) - CDF_2(\tau)) \cdot PDF(x-\tau)d\tau \\ &= \widetilde{CDF}_1(x) - \widetilde{CDF}_2(x) \geq 0 \end{aligned} \quad (10)$$

and therefore we have $\widetilde{CDF}_1(x) \geq \widetilde{CDF}_2(x)$ again. However, this dominance relationship does not establish a total order among all RAT_{sol} because one curve does not dominate another if they cross in the shaded area of Figure 7(a). Therefore the pruning effect is weak.

2) Yield Cut-off Dominance: It is clear from figure 7(b) that we only use the yield cut-off Γ_Υ for comparing the CDFs of the RATs. Since $\Gamma_1 > \Gamma_2$, CDF 1 is said to dominate CDF 2. Under this rule, the relative dominance between all pair of curves is definitely defined, therefore all options are totally ordered. This preserves the property that for each distinct value of load, we only need to retain one solution (which has the largest Γ_Υ). Following from the complexity analysis in Section III-B, the number of distinct capacitance values are tightly upper bounded and hence the number of non-dominating solutions is bounded by $O(|S_{buf}| \cdot c_{max} \cdot |V|)$, where $|S_{buf}|$, c_{max} and $|V|$ are the number of possible buffer sizes, the maximum capacitance value and the number of tree nodes respectively. We conceive this pruning rule from the observation that we pick the optimum solution sol_{opt} at the source n_{src} by finding the largest Γ_Υ among all solutions at n_{src}. Therefore it is reasonable to prune solutions at the same yield point Υ at all nodes without considering the part of CDF larger than Υ, which is irrelevant to obtaining the optimal solution.

Notice that even though pruning under *Yield Cut-off Dominance* only compares one point, it is different from corner case designs since we obtain Γ_Υ from accurate RAT distributions, which are derived from statistical calculation. In the corner case design, we get the worst case RAT from extreme interconnect and buffer parameters. Using such worst case RAT leads to severe over-design.

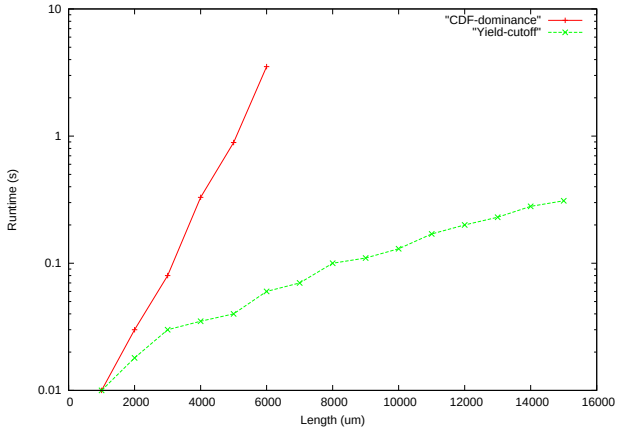


Fig. 8. Runtime in log-scale with different pruning rules

3) Evaluating the Pruning Rules: Figure 8 shows the log-plot of the runtime trends when straight wires of different lengths undergo the $vSBWF$ algorithm with the two pruning rules. The number of nodes grows linearly with the length of the wire. The figure shows that the runtime from *CDF Dominance*-pruning grows exponentially with respect to the wire length. In contrast, the curve for *Yield Cut-off Dominance*-pruning plateaus, which shows that the runtime is polynomial with respect to the line length. The algorithm using *CDF Dominance*-pruning is able to finish in a reasonable time only for some small test cases but takes over 24 hours for any of the test benches in Section IV-E.

TABLE IV
COMPARISON BETWEEN PRUNING USING *CDF Dominance* AND *Yield Cut-off Dominance*

	CDF		Yield Cut-off	
Test-bench	Mean (ps)	SD (ps) (ps)	Mean (ps) (Δ%)	SD (ps) (Δ%)
line	-6569	338	-6569 (0%)	338 (0%)
5-sink	-11543	505	-11545 (0%)	511 (1.2%)
6-sink	-9189	437	-9192 (0.03%)	438 (0.002%)

Table IV shows the statistics of solutions produced using the two pruning rules respectively. We hand-craft these test cases so that $vSBWF$ with *CDF Dominance*-pruning can finish in hours. It is quite obvious that the *Yield Cut-off Dominance*-pruning loses almost no optimality when used in place of the theoretically delay-optimal *CDF Dominance*-pruning. With this observation and the runtime concern, *we shall use the Yield Cut-off Dominance-pruning in practice and in our subsequent*

discussion in the experiment section.

To maximize the timing yield Υ, the best solution to pick at the source n_{src} is the one which has the largest yield cut-off point Γ_Υ. The timing yield Υ can be chosen by designers to fulfill their yield objective.

E. Experiment

We carry out the experiment on the same test cases in Section III-D. We use $SBW+Fill$, which reflects the current design methodology, as our baseline case. To show whether any "partial solutions" is adequate to address the $vSBWF$ problem, we also compare $vSBWF$ against $SBWF$ from Section III, which considers CMP but not L_{eff} variation, and $vSBW+Fill$ which considers L_{eff} variation without CMP. The assumptions on L_{eff} follows from Section IV-A. The $vSBWF$ problem requires a different slew rate constraint due to its random nature, therefore all $SBW+Fill$, $SBWF$ and $vSBW+Fill$ require different over-constrain rates from the one used in Section III-D. We again rely on the binary search using $SBW+Fill$, $SBWF$ and $vSBW+Fill$ to find this new over-constrain rate. We choose the new slew rate constraint to be $P(slew \leq \eta) \geq 99\%$ at all inputs of buffers and sinks t_i, where η = 100ps. This means that the slew rate at all buffer inputs and sinks t_i must have 99% chance meeting the bound η. Under this new requirement, we have found that the over-constrain rate κ for $SBW+Fill$, $SBWF$ and $vSBW+Fill$ are 0.75, 0.78 and 0.85 respectively. In contrast, the $vSBWF$ algorithm considers the random variation during optimization and therefore directly produces optimum solution sol_{opt} that meet such slew rate constraint. The yield Υ we optimize for is set to 0.9. We use the same computing platform as in Section III-D to perform these experiments. To verify the solutions, we perform statistical, CMP-aware timing analysis on the solutions from $SBW+Fill$, $SBWF$, $vSBW+Fill$ and $vSBWF$ through Monte Carlo simulation, which is set to achieve 0.1% error in mean values with 99% confidence.

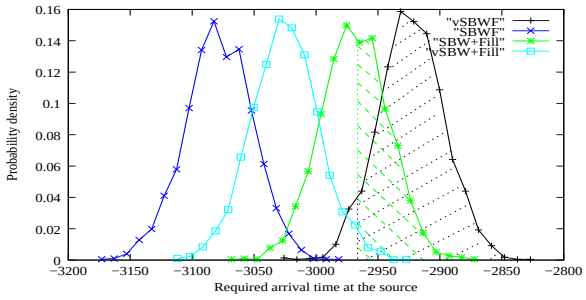


Fig. 9. Probability density distribution of net "s10".

To compare the solutions produced by $SBW+Fill$, $SBWF$, $vSBW+Fill$ and $vSBWF$ in the random L_{eff} regime, we use the concept of timing yield. To illustrate, Figure 9 shows the PDFs of the RATs from the optimized solutions on a large net "s10". We use the *90% yield cut-off point*, $\Gamma_{90\%}$, of the $vSBWF$'s RAT solution, which is 2962ps, as the threshold for timing tests. We regard the proportion of the PDF that has RAT better than $\Gamma_{90\%}$=2962ps as yield. In other words, the PDF of $vSBWF$ has a yield rate of 90% shown in the shaded area under its curve. Under this comparison, the yield from the PDF of $SBW+Fill$ is 37.7%, which is shown in the shaded area under the curve for $SBW+Fill$, while those of $SBWF$ and $vSBW+Fill$ are almost 0%.

Table IV-E shows the comparison between $SBW+Fill$, $SBWF$, $vSBW+Fill$ and $vSBWF$ under both CMP and random L_{eff} variation. We report the yield of $SBW+Fill$ designs in the fifth column of Table IV-E. $SBW+Fill$ results in a significant 43.1% yield loss on average compared to the $vSBWF$ designs. We notice that the $vSBWF$ design reduces buffer area in most cases, but increases wiring area compared to $SBW+Fill$. In general, we observe that considering CMP tends to decrease buffer area due to over-constraining slew rate as explained in Section III-D, while considering random L_{eff} variation tends to increase buffer area for extra design margin. On the other hand, considering either CMP or random variation alone, as in the case of $SBWF$ or $vSBW+Fill$, does not produce the desired optimal buffering and wire sizing solutions, which are shown by the poor yields in the seventh and the ninth columns. The runtime of $vSBWF$ is roughly 25$\times$ of $SBWF$ [9].

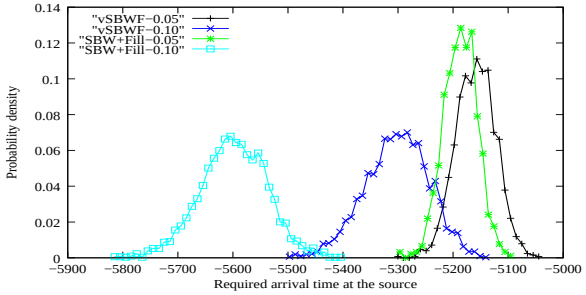


Fig. 10. Probability density distribution of net "r1" assuming $\widehat{L_{eff}}$ = 5% and 10% of $\overline{L_{eff}}$.

We also look into the effectiveness of statistical design on the possible increased random variation in the future process technologies. Figure 10 shows the probability distributions of net "r1" optimized using $SBW+Fill$ and $vSBWF$ under the assumption of standard deviation $\widehat{L_{eff}}$ = 5% (curves' label suffixed with "0.05") and 10% (curves' label suffixed with "0.10") of the mean $\overline{L_{eff}}$ respectively. The curves are much flatter when $\widehat{L_{eff}}$ increased to $10\% \cdot \overline{L_{eff}}$, with the distribution of timing now spans more than 5% of the mean delay. Moreover, $vSBWF$ is now capable of achieving bigger improvement in timing. The yield improvement of "r1" using $vSBWF$ over $SBW+Fill$ is reported to be 12% from Table IV-E under the 5% $\widehat{L_{eff}}$ assumption, while that under the 10% assumption is almost 90%. The nominal delay improvement by $vSBWF$ over $SBW+Fill$ increases from less than 1% under the 5% $\widehat{L_{eff}}$ assumption to more than 5% under the 10% assumption. Experiments on other testcases show similar trend. This shows that statistical design methodologies like $vSBWF$ will become more important for timing closure as process variation increases in future technologies.

[9]Runtime of s1–s5 are not compared since overhead of PWL calculation dominates the runtime of these small test cases

TABLE V

EXPERIMENTAL RESULT OF $SBW + Fill$, $SBWF$, $vSBW + Fill$ AND $vSBWF$ VERIFIED UNDER RANDOM L_{eff} VARIATION AND CMP EFFECTS.

	$SBW + Fill$ ($\kappa = 0.75$)				$SBWF$ ($\kappa = 0.78$)		$vSBW + Fill$ ($\kappa = 0.85$)		$vSBWF$			
test-case	wire area (mm^2)	buffer area ($10^3\times$)	nominal RAT (ps)	yield (%)	nominal RAT (ps) ($\Delta\%$)	yield (%)	nominal RAT (ps) ($\Delta\%$)	yield (%)	wire area (mm^2) ($\Delta\%$)	buffer area ($10^3\times$) ($\Delta\%$)	nominal RAT (ps) ($\Delta\%$)	run-time (s)
s1	0.10	3.3	-1105	12%	-1105 (0%)	6%	-1107 (-0%)	5%	0.11 (8%)	3.2 (-1%)	-1059 (4%)	23
s2	0.11	3.5	-1176	97%	-1232 (-5%)	7%	-1177 (-0%)	93%	0.12 (7%)	3.3 (-6%)	-1176 (0%)	28
s3	0.14	4.9	-1677	90%	-1728 (-3%)	18%	-1678 (-0%)	95%	0.15 (8%)	4.8 (-1%)	-1676 (0%)	33
s4	0.18	6.7	-1460	10%	-1533 (-5%)	0%	-1441 (1%)	49%	0.19 (7%)	6.5 (-3%)	-1412 (3%)	77
s5	0.26	9.7	-2579	93%	-2724 (-6%)	0%	-2587 (-0%)	86%	0.29 (11%)	9.6 (-1%)	-2579 (0%)	174
s6	0.31	12.7	-2400	90%	-2516 (-5%)	0%	-2454 (-2%)	17%	0.35 (12%)	12.7 (-0%)	-2399 (0%)	265
s7	0.38	15.8	-4024	35%	-4225 (-5%)	0%	-4083 (-1%)	1%	0.43 (12%)	16.1 (2%)	-3967 (1%)	558
s8	0.43	19.8	-3337	35%	-3464 (-4%)	0%	-3338 (-0%)	31%	0.49 (13%)	19.3 (-3%)	-3284 (2%)	1022
s9	0.47	21.6	-3092	11%	-3174 (-3%)	0%	-3095 (-0%)	10%	0.52 (12%)	21.3 (-1%)	-3024 (2%)	1080
s10	0.50	22.0	-2967	38%	-3078 (-4%)	0%	-3023 (-2%)	1%	0.56 (12%)	22.8 (3%)	-2922 (2%)	1610
r1	3.74	116.6	-5177	78%	-5604 (-8%)	0%	-5312 (-3%)	0%	4.09 (10%)	115.9 (-1%)	-5160 (0%)	690
r2	7.31	229.4	-6511	30%	-7029 (-8%)	0%	-6715 (-3%)	0%	7.97 (9%)	226.4 (-1%)	-6458 (1%)	1663
r3	9.32	299.0	-7716	60%	-8280 (-7%)	0%	-7989 (-4%)	0%	10.17 (9%)	295.9 (-1%)	-7669 (1%)	2189
r4	18.74	596.6	-11439	24%	-12369 (-8%)	0%	-11735 (-3%)	0%	20.54 (10%)	595.8 (-0%)	-11344 (1%)	3682
r5	28.07	895.1	-12796	0%	-13830 (-8%)	0%	-13119 (-3%)	0%	30.57 (9%)	885.2 (-1%)	-12502 (2%)	5480
				47%	(-5%)	2%	(-1%)	26%	(10%)	(-1%)	(1%)	

V. CONCLUSION

In this paper, we have presented extensions to the dynamic programming algorithm for simultaneous wire sizing and buffer insertion (SBW) to account for the impacts of Chemical Mechanical Polishing (CMP)-induced and random channel length (L_{eff}) process variation on parasitics and timing performance. We have first quantitatively studied the potential impact of CMP-variation on interconnect parasitics, based on which we have developed an accurate, table look-up-based RC model considering systematic CMP variation with pre-calculated, optimized fill-patterns that minimize coupling capacitance. Equipped with such a model, we have studied the simultaneous buffer insertion, wire-sizing and fill insertion problem ($SBWF$). Experiment under conservative assumptions on process variation have shown that the proposed $SBWF$ designs consistently achieve 1.6% delay reduction on average over nominal design ($SBW+Fill$). We also approach the SBW problem considering both systematic CMP variation and random L_{eff} variation ($vSBWF$) by incorporating efficient statistical timing analysis into the $SBWF$ algorithm. We have developed an efficient heuristic for PDF pruning, whose practical optimality is comparable to a provably optimal yet expensive pruning rule. Experimental results show that ($vSBWF$) increases timing yield by 43.1% on average, compared to $SBW + Fill$ which considers nominal L_{eff} value. All these extensions do not change the fundamental dynamic programming framework, therefore they are compatible with other extensions that consider power/area-constrained SBW optimization, which have been intensively studied recently.

In this work, we assume a fixed routing topology with buffer insertion and wire sizing as a post-routing optimization. In the future, we plan to study simultaneous routing topology generation with buffer insertion and wire sizing considering both interconnect and device variations.

REFERENCES

[1] C. Visweswariah, "Death, taxes and failing chips," in *DAC 03*, Jun 2003.

[2] Y. Chen, P. Gupta, and A. B. Kahng, "Performance-impact limited area fill synthesis," in *DAC*, Jun 2003.

[3] T. Tugbawa, T. Park, D. Boning, T. Pan, P. Li, S. Hymes, T. Brown, and L. Camilletti, "A mathematical model of pattern dependencies in cu cmp processes," in *CMP Symposium, Electrochemical Society Meeting*, Oct 1999.

[4] P. Gupta and F. Heng, "Towards a systematic-variation aware timing methodology," in *DAC 04*, Jun 2004.

[5] B. Stine, D. Boning, J. Chung, L. Camilletti, F. Kruppa, E. Equi, W. Loh, S. Prasad, M. Muthukrishnan, D. Towery, M. Berman, and K. A., "The physical and electrical effects of metal-fill patterning practices for oxide chemical-mechanical polishing processes," vol. 45, pp. 665 – 679, March 1998.

[6] K.-H. Lee, J.-K. Park, Y.-N. Yoon, D.-H. Jung, J.-P. Shin, Y.-K. Park, and J.-T. Kong, "Analyzing the effects of floating dummy-fills: from feature scale analysis to full-chip rc extraction," in *Electron Devices Meeting, 2001. IEDM Technical Digest. International*, pp. 31.3.1 – 31.3.4, Dec. 2001.

[7] W. Grobman, M. Thompson, R. Wang, C. Yuan, R. Tian, and E. Demircan, "Reticle enhancement technology: implications and challenges for physical design," in *Proc. Design Automation Conf*, pp. 73 – 78, 2001.

[8] J. Jess, K. Kalafala, S. Naidu, R. Otten, and C. Visweswariah, "Statistical timing for parametric yield prediction of digital integrated circuits," in *DAC 03*, Jun 2003.

[9] A. Agarwal, D. Blaauw, V. Zolotov, and S. Vrudhula, "Computation and refinement of statistical bounds on circuit delay," in *DAC 03*, Jun 2003.

[10] C. Visweswariah, K. Ravindran, K. Kalafala, S. Walker, and S. Narayan, "First-order incremental block-based statistical timing analysis," in *DAC 04*, Jun 2004.

[11] S. Choi, B. Paul, and K. Roy, "Novel sizing algorithm for yield improvement under process variation in nanometer technology," in *DAC 04*, Jun 2004.

[12] S. Raj, S. Vrudhula, and J. Wang, "A methodology to improve timing yield in the presence of process variation," in *DAC*, Jun 2004.

[13] A. Agarwal, K. Chopra, and D. Blaauw, "Statistical timing based optimization using gate sizing," in *DATE*, Mar 2005.

[14] V. Khandelwal, A. Davoodi, A. Nanavati, and A. Srivastava, "A probabilistic approach to buffer insertion," in *ICCAD 03*, Nov 2003.

[15] Semiconductor Industry Association, *International Technology Roadmap for Semiconductors*, 2003.

[16] L. P. P. P. van Ginneken, "Buffer placement in distributed RC-tree networks for minimal Elmore delay," in *Proc. IEEE Int. Symp. on Circuits and Systems*, pp. 865–868, 1990.

[17] J. Cong, L. He, C. Koh, and Z. Pan, "Interconnect sizing and spacing with considering of coupling capacitance," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 20, no. 9, pp. 1164–1169, 2001.

[18] J.-K. Park, K.-H. Lee, J.-H. Lee, Y.-K. Park, and J.-T. Kong, "An exhaustive method for characterizing the interconnect capacitance considering the floating dummy-fills by employing an efficient field solving algorithm," in *Simulation of Semiconductor Processes and Devices, 2000. SISPAD 2000. 2000 International Conference on*, pp. 98 – 101, Sept. 2000.

[19] P. Gupta and A. B. Kahng, "Manufacturing-aware physical design," in *ICCAD*, Oct 2003.
[20] R. Tian, X. Tang, and D. Wong, "Dummy-feature placement for chemical-mechanical polishing uniformity in a shallow trench isolation process," vol. 21, pp. 63 – 71, Jan 2002.
[21] "Quickcap user manual," in *http://www.magma-da.com/*.
[22] T. E. Gbondo-Tugbawa, *Chip-Scale Modeling of Pattern Dependencies in Copper Chemical Mechanical Polishing Process*. PhD thesis, Massachusetts Institute of Technology, 2002.
[23] A. Kahng, G. Robins, A. Singh, H. Wang, and A. Zelikovsky, "Filling and slotting: Analysis and algorithms," in *Proc. Design Automation and Test in Europe*, 1998.
[24] J. Lillis, C. K. Cheng, and T. T. Y. Lin, "Optimal wire sizing and buffer insertion for low power and a generalized delay model," in *Proc. Int. Conf. on Computer Aided Design*, pp. 138–143, Nov. 1995.
[25] H. Bakoglu, *Circuits, Interconnects and Packaging for VLSI*. Addison-Wesley, 1990.
[26] C. Alpert, D. Devgan, and C. Kashyap, "Rc delay metrics for performance optimization," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 20, no. 5, pp. 571–582, 2001.
[27] K. Agarwal, D. Sylvester, and D. Blauuw, "A simple metric for rc circuit based on two circuit moments," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, no. 9, pp. 1346–1354, 2004.
[28] Z. Li, C. Sze, C. Alpert, J. Hu, and W. Shi, "Making fast buffer insertion even faster via approximation techniques," in *ASP-DAC*, Jan 2005.
[29] K. Tam and L. He, "Power optimal dual-vdd buffered tree considering buffer stations and blockages," in *Proc. Design Automation Conference*, Jun 2005.
[30] "Berkeley predictive technology model," in *http://www-device.eecs.berkeley.edu/ ptm*.
[31] R.-S. Tsay, "An exact zero-skew clock routing algorithm," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. CAD-12, pp. 242–249, Feb. 1993.
[32] W. Shi, Z. Li, and C. Alpert, "Complexity analysis and speedup techniques for optimal buffer insertion with minimum cost," in *ASP-DAC*, Jan 2005.
[33] Y. Cao, P. Gupta, A. Kahng, D. Sylvester, and J. Yang, "Design sensitivities to variability: Extrapolations and assessments in nanometer vlsi," in *ASIC/SOC Conference*, Sept 2002.
[34] N. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*. Addison-Wesley, 2004.
[35] J. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits - A Design Perspective*. Prentice Hall Inc., 2003.
[36] A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering*. Addison-Wesley, 1994.
[37] W. Shi and Z. Li, "An o(nlogn) time algorithm for optimal buffer insertion," in *DAC*, Jun 2003.
[38] A. Devgan and C. Kashyap, "Block-based static timing analysis with uncertainty," in *ICCAD 03*, Nov 2003.