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

Lei He[1] Andrew Kahng[2,3] 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
Blaze DFM, Inc.[3], Sunnyvale, CA 94089, USA
{lhe,ktam,jinjun}@ee.ucla.edu[1], {abk}@ucsd.edu[2]

ABSTRACT

This paper studies the impacts of Chemical Mechanical Polishing (CMP)-induced systematic variation and random channel length (L_{eff}) variation of transistors on interconnect design. We first construct a table look-up based interconnect RC parasitic model considering CMP effects with optimized fill insertion. Based on the model, we solve the simultaneous buffer insertion, wire sizing and fill insertion ($SBWF$) problem under CMP variation. We also extend the $SBWF$ problem to consider the random L_{eff} variation ($vSBWF$). We approach the resulting $vSBWF$ problem by (1) incorporating probability density function (PDF) into the $SBWF$ algorithm; and (2) developing an efficient heuristic for PDF pruning, whose practical optimality is verified by an accurate but much slower pruning. Experimental results show that the $SBWF$ design improves timing by 1.0% and reduces power by 5.7% on average with 7.4% less buffer area over the conventional buffer insertion and wire sizing design followed by fill insertion ($SBW + Fill$), and that the $vSBWF$ design reduces yield loss due to CMP and L_{eff} variations by 44.3% on average over the $SBW + Fill$ design. The runtime of $vSBWF$ is 8.3× that of $SBWF$, and $vSBWF$ for the largest example containing 3103 sinks finishes in 124 minutes.

1. INTRODUCTION

The economic engine of the semiconductor industry is based on the promise of ever more complex silicon systems delivered to the market at ever-lower prices. High performance and high yield are two keys to sustaining such a trend. However, design uncertainty in nanometer technology nodes threatens such an economic growth model. The main cause for design uncertainty is two-fold: systematic manufacturing process variation and random process variations due to small geometric dimensions [1]. For example, chemical-mechanical planarization (CMP) is an enabling manufacture process to achieve uniformity of dielectric and conductor height in back-end-of-line (BEOL) process step. However, CMP also introduces systematic design variations due to *dummy fill* insertion [2] and *dishing* and *erosion* [3]. The channel length of a transistor L_{eff} greatly affects device performance. But increasingly shrunk L_{eff} makes it difficult to print the desired geometry exactly on silicon due to the limit of existing lithographic technology. Moreover, major L_{eff} variation is attributed to random variation as pointed out by [4]. As a result of combined systematic and random variations, manufactured circuits exhibit different performance from that estimated by circuit simulation using nominal circuit parameters; therefore, high yield rate is more difficult to achieve in advanced process.

Despite its importance, there is very limited work on circuit optimization for yield improvement considering process variations. For example, statistical timing analysis [5, 6, 7] has been studied recently, but results mainly focus on analysis rather than design. A recent work [8] on buffer insertion in a routing tree considers the uncertainty in wire-length estimation but not process variations such as CMP effects and L_{eff} variation.

The first contribution of this paper develops an efficient algorithm to solve the simultaneous buffer insertion, wiring sizing and fill insertion ($SBWF$) problem. We combine the conventional dynamic programming framework for buffer insertion [9] with a table look-up based interconnect RC parasitic model that considers CMP effects (fill insertion, dishing and erosion) to produce an efficient algorithm for solving the $SBWF$ problem. The second contribution of this work extends the $SBWF$ algorithm to consider random L_{eff} variation ($vSBWF$). By incorporating the efficient piece-wise linear (PWL) model [10] for cumulative distribution function (CDF) and an effective probability density function (PDF) pruning rule into $vSBWF$, we achieve significant reduction of yield loss due to both systematic CMP-induced variation and random L_{eff} variation.

The rest of the paper is organized as follows. In Section 2, we review the CMP-related design variations and propose

*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.

ISPD '2005 San Francisco, California USA

an accurate yet efficient table look-up based CMP-aware RC parasitic models. In Section 3, we present our $SBWF$ problem formulation, algorithms and experimental results. In Section 4, we extend the $SBWF$ algorithm to consider random L_{eff} variation ($vSBWF$) and evaluate the impact of $vSBWF$ on yield optimization. We conclude the paper with discussion of our future research in Section 5.

2. MODELING OF CMP EFFECTS

2.1 CMP Induced Variations

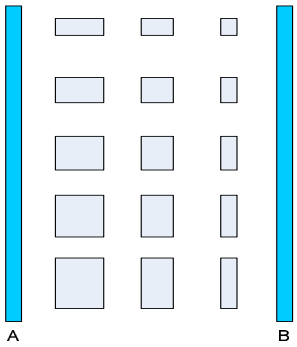

Figure 1: Fill pattern definition.

The following two types of CMP effects are considered in this paper: dummy fill insertion, and dishing and erosion. Dummy fill insertion improves the uniformity of metal feature density and enhances the planarization that can be obtained by CMP. In this work, we assume rectangular, isothetic fill features aligned horizontally and vertically between two adjacent interconnects as shown in Figure 1. In the figure, conductors A and B are *active* interconnects and the metal shapes between them are dummy fills. To specify the amount of fill metal needed in the space and the resulting metal density, we need the following definitions.

Definition 1. Local metal density ρ_f – the proportion of the oxide area between two neighbouring interconnects that dummy fill metal occupies.

Definition 2. Effective metal density ρ_{Cu} – the proportion of the area in a planarization window [3] that all metal features (interconnect + dummy fill metal) occupies.

To achieve CMP planarity and yield optimization, the foundry usually requires an effective metal density ρ_{Cu} to be satisfied in a "fixed-dissection" regime [2, 11]. Fixed-dissection fill synthesis typically results in a number of tiles (i.e., square regions of layout, usually several tens of microns on a side) wherein prescribed amounts of fill features are to be inserted to meet individual tile's metal density requirement. This translates to assigning the amount dummy fill metal to the space between interconnects, and such amount is expressed in terms of local metal density ρ_f.

It has been shown in [12] that for a given local metal density ρ_f requirement between two interconnects, there exists many possible valid fill patterns that achieve the same required fill feature area and satisfy all design rules. According to [12], fill insertion significantly increases both C_c and C_s when compared to the nominal case that does not consider fill insertion; different fill patterns that are nominally "equivalent" with respect to foundry rules yield a wide range of C_c and C_s values. Moreover, it has been shown in [12] that the relative change for C_c can be more than 300%, and that even though the variation of C_s is less dramatic than C_c, variation of more than 10% from the nominal C_s can be observed. Therefore, to obtain robust designs that meet performance and yield expectation after insertion of dummy fill, the variation (i.e., increase) of both C_c and C_s must be considered in the design flow.

Figure 2 illustrates dishing and erosion phenomena due to CMP [13]. Both dishing and erosion cause loss of metal thickness and change interconnect cross-sections [3], and hence may affect interconnect parasitics. According to [12], dishing and erosion can cause wire resistance to increase by more than 30%, but have limited impact on interconnect capacitance

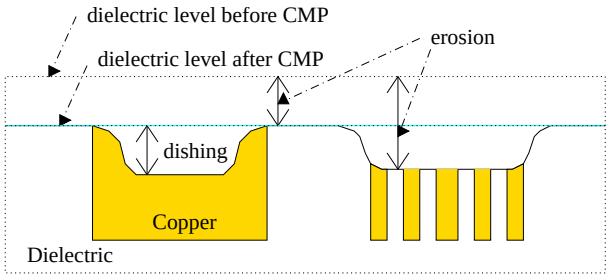

Figure 2: Dishing and Erosion in Copper CMP.

2.2 CMP-aware Table-based RC Model

Using QuickCap [14], a commercial signoff-quality tool, to extract C_c and C_s for different interconnect configurations with consideration of fill insertion, we organize the extracted capacitance in a table indexed by active interconnect width, spacing and local metal density. As different fill patterns under the same pattern density result in different capacitance values, we only save the capacitance values under the *best* fill pattern, which gives the minimum C_c among all patterns. We employ the closed-form formulae for a multi-step CMP process to calculate post-CMP interconnect geometries [13] from which we compute the resistance considering dishing and erosion. 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 are called *CMP-oblivious* RC models.

3. 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 CMP on interconnect design. We propose a new method to solve the SBW and the *fill insertion* problem simultaneously, and we denote it as $SBWF$. In contrast, current designers use a two-step approach which first solves the SBW problem, then applies either the de facto rule-based method or the more recently proposed model-based fill insertion method [15] to determine the amount of fill needed. We use this two-step approach as our baseline for comparison, which is denoted as $SBW + Fill$ in this paper.

3.1 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 *arrival time* (AT) is maximized at n_{src}, subject to (1) the slew rate constraint η at all n_s and buffers' driving points; and (2) the effective metal density requirement ρ_{Cu} for CMP planarization.

We characterize the source n_{src} by its driving resistance R_{src}; each sink n_s by its loading capacitance L_s and the required arrival time AT_s. We associate each edge $e_{i,j}$ with two center-to-edge wire width w_1 and w_2 as illustrated in Fig. 3 [1]. We express w_1 and w_2 in terms of multiples of the minimum wire width $\check{w}$. To respect the design rules, we impose $0.5 \cdot \check{w} \leq w_k \leq s_k - \check{w}$, where $k = 1, 2$ and s_k is the spacing from the center line to the edges of its two nearest neighboring wires, also in terms of the multiples of $\check{w}$. 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.

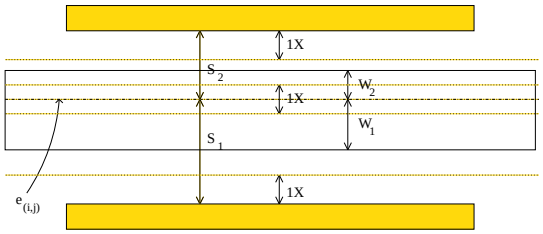

Figure 3: Illustration of asymmetric wire sizing.

3.2 Slew Rate Constrained SBW Algorithm

The slew rate constrained SBW algorithm largely follows the dynamic programming (DP) framework of [9], 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 at node n) must keep track of the downstream capacitance $sol_n \rightarrow C$ and the arrival time $sol_n \rightarrow AT$ associated with sol_n. The arrival time AT_n at node n is defined by

$$AT_n = \min_{n_s \forall s} (AT_n^s - d(n_s, n))$$

where $d(n_s, n)$ is the delay from the sink node n_s to node n. The pseudo-code in Table 1 summarizes the flow of the algorithm.

We start procedure DP at the source n_{src}, which recursively calls DP on all nodes in a depth first order to create the solution sets SOL_n for all $n \in V$. Upon its return at the source n_{src}, DP gives the set SOL_n^{src}, from which we pick

[1]The asymmetric wire sizing problem was first proposed in [16] without slew rate constraints, which does not consider the CMP-induced variation neither.

```
procedure DP(n)
  if (n is a sink) C_i = L_n; AT_n = required AT_n;
  else C_n = 0; AT_n = ∞;
  add (C_n, AT_n) to set SOL_n;
  for each e_{n,v} to downstream node v, do
   SOL_v = DP(v);
   for each sol_j ∈ SOL_v, do
    Propagate(e_{n,v}, sol_j, SOL_n);
  return SOL_n;
procedure Propagate(e_{n,v}, sol_j, SOL_n)
  for each sol_m in SOL_n, do
   for each wire size for e_{n,v}, do
    for each possible buffer S_buf, do
     if (S_buf = 0, i.e.  no buffer)
      newC = sol_m → C + C_{e_{i,j}} + sol_j → C;
     else
      newC = sol_m → C + S_buf → C_in;
     newAT = max(sol_m → AT, Delay(S_buf, e_{i,j}, sol_j));
     sol_new = (newC, newAT);
     if (Sbuf ≠ 0 and Slew(sol_new) > η̄)
      continue;
     if (Prune(sol_new, SOL_n) = ''survive'')
      add sol_new to set SOL_n;
```

Table 1: Pseudo-code for the $SBWF$ algorithm

.

a solution sol_{opt} that maximizes $AT_{opt} + R_{src} \cdot C_{opt}$. We obtain the actual wire sizing assignment and buffer placement by a simple backtracking algorithm using sol_{opt}.

We use the first order Elmore delay model and slew rate model [17] in our current implementation due to their high fidelity over real design metrics. Procedure $Delay$ updates the AT_n of each solution sol_n at node n by

$$\begin{aligned} AT_n &= AT_{sol}^n - r_{n,v} \cdot L_v - 0.5 \cdot r_{n,v} \cdot c_{n,v} \\ &\quad - d_{buf} - R_{eff} \cdot (L_n + c_{n,v}) \end{aligned} \tag{1}$$

where $r_{n,v}$ and $c_{n,v}$ are the resistance and capacitance of edge $e_{n,v}$, respectively; L_n is the downstream capacitance at the node n; d_{buf} and R_{eff} are buffer intrinsic delay and output resistance, respectively, and both are functions of buffer size S_{buf}. Procedure $Slew$ implements Bakoglu's slew rate metric [17] 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 $Delay$ and $Slew$ can be replaced by other more accurate delay [18] and slew [19] metrics which consider higher order moments.

The DP algorithm runs in polynomial time with respect to the tree size 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 $AT_{sol}^1 \leq AT_{sol}^2$. The procedure $Prune$ in the above pseudo-code compares the newly created solution sol_{new} against all solutions in the set SOL_n to remove inferior or dominated solutions. If sol_{new} is not dominated by any other solutions, $Prune$ returns "*survive*".

The overall time complexity of the DP is $O(|S_{buf}| \cdot r^{|V|})$ when slew rate constraint is not considered, where r is the number of available choices of wire widths, $|V|$ is the number of nodes in the interconnect tree and $|S_{buf}|$ is the number of possible sizes for buffers. Wire sizing causes exponential growth of distinct capacitance values C_{sol} as solutions propagate. When slew rate constraint is considered, there is an upper bound on the distance that a wire can run without buffering. This translates to the fact that the number of distinct C_{sol} values is quite tightly upper bounded. Since we only need to keep one solution under each distinct C_{sol}, the

number of solutions grows in the order of $O(|S_{buf}|\cdot|C_{sol}|\cdot|V|)$ instead, where $|C_{sol}|$ denotes the bounded number of distinct capacitance values. We experimentally confirm this observation in Section 4.4.3.

3.3 Extension to $SBW + Fill$ and $SBWF$

The conventional design flow $SBW + Fill$ has two steps. The first step solves the slew rate constrained SBW problem using CMP-oblivious RC parameters only; the second step inserts the best fill patterns into the space between the wires of the already buffered and sized routing tree in order to satisfy the required effective metal density requirement ρ_{Cu} for CMP planarization.

In contrast, we propose an integrated approach to solve the $SBWF$ problem, and such an approach is denoted as $SBWF$ whenever there is no ambiguity. $SBWF$ uses CMP-aware table-based RC parasitic model from Section 2.2 for delay and slew rate calculation while solving the slew rate constrained SBW problem. For every edge $e_{i,j}$, we define two local dummy fill density requirements ρ_f^1 and ρ_f^2 that specify the amount of fill metal in the space between edge $e_{i,j}$ and its two neighboring wires in order to satisfy the effective metal density target ρ_{Cu}. ρ_f^1 and ρ_f^2 can be obtained from algorithms such as [15]. Note that considering different wire width necessitates the adjustment to ρ_f^1 and ρ_f^2 such that the effective metal density constraint ρ_{Cu} is still satisfied after wire sizing. Knowing the width w_i, the spacing s_i and the adjusted density ρ_f^i for $i = 1, 2$, we can lookup the CMP-aware RC tables to obtain the RC values for the corresponding best fill pattern to solve the SBW problem.

3.4 Experiment

technology	ITRS $65nm$ [20]
interconnect	global interconnect layer
delay model	Elmore delay, π-model for interconnect
slew model	Bakoglu's first order metric [17]
device	BSIM 4 [21]
R_{src}	100Ω
L_{sink} & AT_{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 um
test cases	r1∼r5: clock trees from [22] s1∼s10: random Steiner trees

Table 2: Experimental Settings

Table 2 shows the experimental settings used in this paper. We choose typical buffer sizes and wire sizes that are normally used in real designs. Because there is no physical layout information in the original test cases obtained from [22], we randomly generate the neighboring wire spacing data and the local metal density requirements for each interconnect in all test cases. All experiments are performed on an Intel Xeon 1.9Ghz Linux workstation with 2Gb of memory.

We over-constrain the maximum slew rate η in the first step of $SBW + Fill$ in order to meet the actual slew rate constraint after fill insertion. The first step of $SBW + Fill$ algorithm always under-estimates the slew rate as it does not consider CMP-induced variation on RC. The *over-constrain rate*, κ, is defined as the ratio between the over-constrained slew rate to the actual slew rate constrains. The value of κ can be obtained via a binary search, in which each iteration involves an execution of $SBW+Fill$, and is time-consuming. 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 constrains without repetition. In our current setting, we use $\kappa = 0.83$ for $SBW+Fill$, which gives maximum slew rates that satisfy the slew rate bound η in all test cases.

Table 3 compares the experiment results from $SBW+Fill$ and $SBWF$ in terms of wiring area, buffer area, maximum slew rate, required arrival time at the source n_{src} and power measured as energy per switch. We verify both $SBW + Fill$ and $SBWF$ designs under the CMP-aware parasitic model. A solution with larger AT implies smaller delay and is therefore more preferable. Comparing $SBW+Fill$ against $SBWF$ (relative change of values shown in the brackets), we see that $SBWF$ achieves larger AT for all test cases and the average increase is 1.0%. Moreover, $SBWF$ also reduces buffer area by 7.4% on average with moderate wiring area increase (on average 1.6%). 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. $SBWF$ also reduces power by 5.7% on average over $SBW + Fill$ as a result of significant reduction of buffer area. 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 determine the over-constrain rate κ as explained above and therefore costs much more time than the reported value. From all of these results, we see that designs considering CMP impacts out-perform the counterpart traditional designs in terms of delay, buffer area, power and runtime.

4. YIELD-DRIVEN SBW

4.1 L_{eff} Variation

One of the most important process uncertainty that affects circuit performance is the random variation of devices' effective channel lengths (L_{eff}) [23, 4]. The variation of L_{eff} manifests itself in changing devices' different 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 [24]. We model L_{eff} with a Gaussian distribution Δ_L with its mean value $\overline{L_{eff}}$ equal to its nominal value and the standard deviation $\widehat{L_{eff}}$ equal to 5% of the mean value.

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 with the ratio of 2:1 for symmetric rise and fall. Therefore the total input capacitance is a function of $L_{eff}^{\alpha} = L_{eff}^{n} + 2 \cdot L_{eff}^{p}$, where L_{eff}^{n} and L_{eff}^{p} are the L_{eff} of the NMOS and PMOS transistors respectively. Since L_{eff}^{n} and L_{eff}^{p} are assumed to be independent Gaussian random variables having the same Gaussian distribution Δ_L, L_{eff}^{α} is also a Gaussian random variable with

			$SBW + Fill$ ($\kappa = 0.83$)					$SBWF$				
test-case	wire length (m)	# sink	wire area (mm^2)	buffer area (x min)	AT (ps)	power (pJ)	run-time (s)	wire area (mm^2) (Δ%)	buffer area (x min) (Δ%)	AT (ps) (Δ%)	power (pJ) (Δ%)	run-time (s)
s1	0.03	19	0.11	1100	-510	8	1	0.11 (1.8%)	1100 (0.0%)	-510 (0.1%)	8 (0.2%)	1
s2	0.04	29	0.12	1320	-546	10	1	0.13 (4.8%)	1320 (0.0%)	-543 (0.7%)	10 (0.5%)	3
s3	0.05	49	0.16	1900	-801	14	1	0.16 (2.4%)	1700 (-10.5%)	-801 (0.1%)	12 (-8.1%)	3
s4	0.07	99	0.20	2720	-737	19	5	0.20 (-0.1%)	2460 (-9.6%)	-734 (0.5%)	18 (-7.7%)	7
s5	0.10	199	0.29	4440	-1394	31	16	0.29 (-0.0%)	3940 (-11.3%)	-1367 (1.9%)	28 (-9.1%)	20
s6	0.13	299	0.34	5700	-1298	40	52	0.34 (0.9%)	5120 (-10.2%)	-1297 (0.1%)	37 (-8.1%)	57
s7	0.16	499	0.42	7940	-2243	55	167	0.43 (0.5%)	7580 (-4.5%)	-2198 (2.0%)	53 (-3.6%)	177
s8	0.19	699	0.48	9880	-1916	69	307	0.48 (1.4%)	9300 (-5.9%)	-1889 (1.4%)	66 (-4.7%)	322
s9	0.21	799	0.51	10760	-1803	75	289	0.54 (5.1%)	10460 (-2.8%)	-1762 (2.3%)	73 (-2.0%)	413
s10	0.22	899	0.55	11220	-1692	79	467	0.56 (1.2%)	10580 (-5.7%)	-1669 (1.3%)	75 (-4.5%)	591
r1	1.32	267	3.89	36100	-2437	266	67	3.96 (1.9%)	33080 (-8.4%)	-2427 (0.4%)	250 (-6.2%)	86
r2	2.60	598	7.66	72180	-3080	531	173	7.75 (1.2%)	64080 (-11.2%)	-3044 (1.2%)	486 (-8.5%)	193
r3	3.37	862	9.55	89240	-3684	662	207	9.64 (1.0%)	80440 (-9.9%)	-3636 (1.3%)	613 (-7.4%)	257
r4	6.81	1903	19.38	183800	-5372	1358	389	19.52 (0.7%)	163180 (-11.2%)	-5319 (1.0%)	1243 (-8.5%)	459
r5	10.20	3101	28.97	273560	-6005	2025	512	29.25 (1.0%)	244720 (-10.5%)	-5960 (0.7%)	1865 (-7.9%)	727
avg								(1.6%)	(-7.4%)	(1.0%)	(-5.7%)	

Table 3: Experimental result from $SBW + Fill$ and $SBWF$ verified under CMP-perturbed RC.

mean $3 \cdot \overline{L_{eff}}$ and standard deviation $\sqrt{5} \cdot \widehat{L_{eff}}$. The 99% percentile of L^{α}_{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. We set L_{eff} of the transistors to reflect this amount in SPICE and measure the effective input capacitance. Such L^{α}_{eff} happens with a probability of 1%, and the 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}. We find the joint distribution of R_{eff} and d_{buf} due to random L_{eff} variation by Monte Carlo simulation using SPICE. Equation (3) shows the covariance matrix M of a 20× buffer, where $C_{x,y}$ is the covariance of x and y, and subscripts R and d refer to R_{eff} and d_{buf} respectively. $\sqrt{C_{R,R}}$ and $\sqrt{C_{d,d}}$ are about 15% and 6% of the their respective mean, which shows that R_{eff} and d_{buf} has significant variation due to variation in L_{eff}. The large $C_{R,d}$ also demonstrates that R_{eff} and L_{eff} are highly correlated. Therefore we characterize R_{eff} and d_{buf} accurately using a joint probability density function (JPDF) $f_{R,d}(R_{eff}, d_{buf})$.

$$M = \begin{bmatrix} C_{R,R} & C_{R,d} \\ C_{R,d} & C_{d,d} \end{bmatrix} = \begin{bmatrix} 771 & 26.5 \\ 26.5 & 14.0 \end{bmatrix} \qquad (3)$$

For a buffer with driving load L_{buf}, the delay of the loaded buffer is given by $d_{load} = L_{buf} \cdot R_{eff} + d_{buf}$. After transformation of variables [25], we obtain the probability density function (PDF) of the loaded buffer delay as

$$f_{d(L)}(d_{load}) = \int_{-\infty}^{\infty} f_{R,d}(R_{eff}, d_{load} - L_{buf} \cdot R_{eff}) dR_{eff} \quad (4)$$

Using $f_{R,d}(R_{eff}, d_{buf})$ captured from Monte Carlo simulation, we obtain the PDF $f_{d(L)}(d_{load})$ numerically.

4.2 $vSBWF$ Problem Formulation

We call the $SBWF$ problem considering L_{eff} random variation as $vSBWF$. Owing to the statistical nature of $vSBWF$, we treat the AT 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

$$\Upsilon = P(AT_s \geq \Gamma_{\Upsilon}) \qquad (5)$$

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

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

4.3 Representing and Computing AT

To solve $vSBWF$ via the same DP framework as shown in Section 3.2, we have to replace the deterministic AT 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 AT^i_{sol}'s CDF in any solution sol_i.

In our implementation, we consider the negative of AT_{sol}, i.e. $-AT_{sol}$, for the sake of simpler mathematical manipulation. For example, to obtain the new AT^z_{sol} at node n_z, we take the minimum of AT^p_{sol} and AT^q_{sol} propagated from child nodes n_p and n_q. When negative AT is considered, we take the maximum of $-AT^p_{sol}$ and $-AT^q_{sol}$ instead. The CDF CDF_z of AT^z_{sol} is simply given by the closed-form formula $CDF_z = CDF_p \cdot CDF_q$, where CDF_p and CDF_q are the CDFs of AT^p_{sol} and AT^q_{sol} respectively.

We represent CDF in the the form of *piecewise-linear curve* (PWL) as in [10]. 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 to convolutions of steps and ramps and multiplication of ramps, both of which have closed-form quadratic solution. [10] has

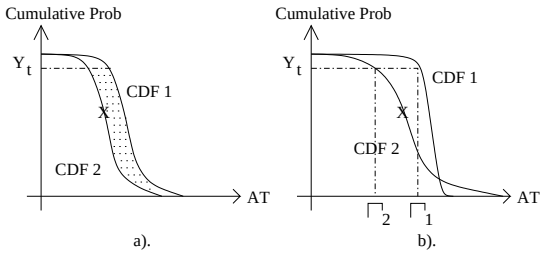

Figure 4: CDF of ATs to illustrate the definition of timing yield, yield cut-off point and and pruning rules

depicted operations for Elmore delay calculation and have provided closed-form quadratic formula. 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 form.

Even though the first order Elmore delay and slew rate model are used in this work, the application of PWL is not limited to these first order models. In fact, it can be applied to other higher order models. For example, delay and slew rate metrics in [18] and [19] 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 have analytical (though not closed-form for general non-linear CDF) solutions. By modeling CDFs with PWL curves, we can apply the analytical formula for each ramp component and proceed with the same methodology to compute CDFs in the PWL form.

4.4 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 rule, and the optimality of which is experimentally supported by an alternative slow but theoretically sound *CDF Dominance*-pruning rule.

4.4.1 CDF Dominance

Figure 4(a) shows the *CDF Dominance* relationship. In the shaded area CDF 1 is 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 $CDF_1'(x)$ and $CDF_2'(x)$ 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 $CDF_1(x) \leq CDF_2(x)\ \forall x$. Statistical maximum corresponds to CDF multiplication, which is obtained by

$$\begin{aligned} CDF_1'(x) &= CDF_1(x) \cdot CDF(x) \\ &\geq CDF_2(x) \cdot CDF(x) = CDF_2'(x) \end{aligned} \quad (6)$$

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

$$CDF_i'(x) = \int_{-\infty}^{\infty} CDF_i(\tau) \cdot PDF(x-\tau) d\tau \quad (7)$$

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

$$\begin{aligned} &\int_{-\infty}^{\infty} (CDF_2(\tau) - CDF_1(\tau)) \cdot PDF(x-\tau) d\tau \\ = \ & CDF_2'(x) - CDF_1'(x) \geq 0 \end{aligned} \quad (8)$$

and therefore we have $CDF_1'(x) \leq CDF_2'(x)$ again. However, this dominance relationship does not establish a total order among AT_{sol} for solutions $sol \in SOL$ because one curve does not dominate another if they cross in the shaded area of Figure 4(a). Therefore the pruning effect is weak.

4.4.2 Yield Cut-off Dominance

It is clear from figure 4(b) that we only use the yield cut-off Γ_Υ for comparing the CDFs of the ATs. Since $\Gamma_1 > \Gamma_2$, CDF 1 is said dominate CDF 2. All options are totally ordered under this rule, which preserves the property that for each distinct value of load, we retain only the largest Γ_Υ. Following from the complexity analysis in Section 3.2, the number of distinct capacitance values are tightly upper bounded and hence the number of non-dominating solutions is bounded by $O(|S_{buf}| \cdot |C_{sol}| \cdot |V|)$, where $|S_{buf}|$, $|C_{sol}|$ and $|V|$ are the number of possible buffer sizes, distinct capacitance values and 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 such point from accurate AT distributions, which are derived from statistical calculation. In corner case design, we get the worst case AT from extreme interconnect and buffer parameters. Using such worst case AT leads to sub-optimal designs.

4.4.3 Evaluating the Pruning Rules

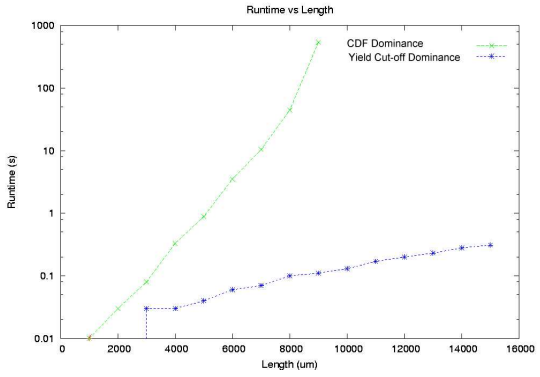

Figure 5: Runtime in log-scale with different pruning rules

Figure 5 shows the log-plot of the runtime trends when straight wires of different lengths undergo $vSBWF$ algo-

rithm with the two pruning rules. The number of nodes grows linearly with the length of the wire. The figure shows that the runtime for *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 4.5.

	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 4: Comparison between pruning using *CDF Dominance* and *Yield Cut-off Dominance*

Table 4 shows the statistics of *solutions* produced by using the two-pruning rules. We hand-craft these test cases so that $vSBWF$ with *CDF Dominance*-pruning finishes in hours. It is quite obvious that the heuristic *Yield Cut-off Dominance*-pruning loses almost no optimality when used in place of the theoretically plausible *CDF Dominance*-pruning. With this observation and the runtime concern, *we shall use 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 requirement objective.

4.5 Experiment

We carry out the experiment on the same test cases in Section 3.4. Section 4.1 has already explained the assumptions on L_{eff}. The $vSBWF$ problem requires a different slew rate constraint due to its random nature, therefore the $SBW + Fill$ algorithm requires a different over-constrain rate from the one used in Section 3.4 to satisfy the new constraint. We again rely on the timing consuming binary search using the $SBW + Fill$ algorithm 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 $\eta = 100ps$. 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 is 0.75. In contrast, $vSBWF$ 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 3.4 to perform these experiments.

To compare the solutions produced by $SBW + Fill$ and $vSBWF$ in the random L_{eff} regime, we use the concept of timing yield. Figure 6 shows the PDFs of the ATs from the optimum solutions produced by $SBW + Fill$ and $vSBWF$ algorithms respectively on a large net. We use the *90% yield cut-off point* $\Gamma_{90\%}$ of $vSBWF$ the optimum AT, which is $7227ps$, as the threshold for timing tests. We regard the proportion of the PDF that has AT better than $\Gamma_{90\%}$=$7227ps$ as yield. Under this comparison, the yield from the PDF of $SBW + Fill$ is 25.1%, which is shown in the shaded area

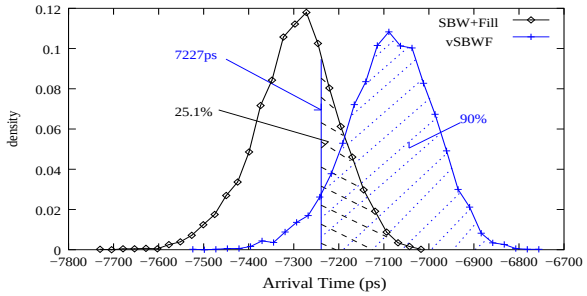

Figure 6: The definition of yield using CDFs.

under the curve for $SBW + Fill$. The PDF of $vSBWF$ has a yield rate of 90% shown in the shaded area under its curve.

Table 5 shows the comparison between $SBW + Fill$ and $vSBWF$. We report the yield of $SBW + Fill$ designs in the fourth column of Table 5. $SBW + Fill$ results in 45.7% yield loss on average compared to the $vSBWF$ designs. This suggests that our variability-driven design is also a yield-driven design and that the resulting yield improvement is significant. It is interesting to notice that the $vSBWF$ design also reduces buffer area in most cases, but increases wiring area compared to $SBW + Fill$. In general, we observe that considering CMP tends to decreases buffer area due to over-constraining slew rate as explained in Section 3.4, while considering random L_{eff} variation tends to increase buffer area for extra design margin. Wire sizes tend to increase as a result of both CMP and random variation. Increased wire size (1) compensates for the increased resistance caused by dishing and erosion; and (2) reduces the effect of the large R_{eff} variation on delay. The runtime of $vSBWF$ is roughly 8.3× of $SBWF$ [2], which again shows that $vSBWF$ is a polynomial time algorithm rather than exponential with respect to the tree size.

5. CONCLUSION

In this paper, we have studied the impacts of Chemical Mechanical Polishing (CMP)-induced systematic variation and random channel length (L_{eff}) variation of transistors on interconnect design. We have constructed an accurate, table look-up based RC model considering systematic CMP variation effects with pre-calculated optimum fill insertion. Using the model, we have studied the simultaneous buffer insertion, wire-sizing and fill insertion problem ($SBWF$). Experimental results show that the proposed $SBWF$ designs achieve 1.0% delay reduction, 5.7% power reduction and 7.4% buffer area reduction on average when compared to the designs produced from the conventional design flow which performs fill insertion after buffer insertion and wire sizing ($SBW + Fill$). We also approach the SBW problem considering both systematic CMP variation and random L_{eff} variation ($vSBWF$) by (1) incorporating probability density function (PDF) into the $SBWF$ algorithm; and (2) developing an efficient heuristic for PDF pruning, whose practical optimality is verified by an accurate but much slower pruning. Experimental results show that $vSBWF$ increases timing yield by 44.3% on average, compared to $SBW + Fill$ which considers nominal L_{eff} value.

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

	$SBW + Fill$ ($\kappa = 0.75$)				$vSBWF$				
test-case	wire area (mm^2)	buffer area (x min)	yield (%) (s)	run-time (mm^2)	wire area (Δ%)	buffer area (x min)	$\Gamma_{90\%}$ (ps)	yield (%)	runtime (s)
s1	0.11	1140	50.0%	1	0.11 (6.1%)	1140 (0.0%)	-559	90.0%	28
s2	0.12	1460	97.3%	2	0.13 (7.5%)	1440 (-1.4%)	-613	90.0%	56
s3	0.15	1860	92.8%	3	0.16 (8.6%)	1680 (-9.7%)	-843	90.0%	47
s4	0.20	3040	87.4%	8	0.21 (6.1%)	2780 (-8.6%)	-820	90.0%	119
s5	0.28	4540	54.8%	20	0.30 (7.0%)	4340 (-4.4%)	-1452	90.0%	205
s6	0.34	6340	3.2%	44	0.36 (6.6%)	5740 (-9.5%)	-1344	90.0%	307
s7	0.42	7900	66.8%	145	0.45 (6.5%)	7840 (-0.8%)	-2339	90.0%	786
s8	0.48	10820	44.8%	261	0.52 (9.2%)	10480 (-3.1%)	-1983	90.0%	1690
s9	0.53	11720	69.5%	320	0.56 (6.0%)	11160 (-4.8%)	-1853	90.0%	1823
s10	0.57	12380	95.4%	507	0.61 (7.4%)	12020 (-2.9%)	-1801	90.0%	2402
r1	3.98	40940	0.1%	101	4.14 (3.9%)	37780 (-7.7%)	-2542	90.0%	1054
r2	7.71	80140	6.7%	213	8.12 (5.4%)	74420 (-7.1%)	-3228	90.0%	2126
r3	9.73	100240	5.9%	277	10.26 (5.4%)	94440 (-5.8%)	-3798	90.0%	2140
r4	19.57	201720	9.0%	607	20.66 (5.6%)	191060 (-5.3%)	-5599	90.0%	4429
r5	29.17	297780	1.5%	972	30.83 (5.7%)	281500 (-5.5%)	-6210	90.0%	7440
			45.7%		6.5%	-5.1%		90.0%	8.3x

Table 5: Experimental result of $SBW+Fill$ and $vSBWF$ verified under random L_{eff} variation and CMP effects on RC parasitics.

In this work, we assume a fixed routing topology with buffer insertion and wire sizing as a post layout synthesis process. In the future, we plan to study simultaneous routing topology generation with buffer insertion and wire sizing considering systematic and random variations due to both CMP and device effects.

6. 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] 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.

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

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

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

[9] 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.

[10] A. Devgan and C. Kashyap, "Block-based static timing analysis with uncertainty," in *ICCAD 03*, Nov 2003.

[11] P. Gupta and A. B. Kahng, "Manufacturing-aware physical design," in *ICCAD*, Oct 2003.

[12] L. He, A. B. Kahng, K. Tam, and J. Xiong, "Design of integrated-circuit interconnects with accurate modeling of chemical-mechanical planarization," in *Proc. SPIE Microlithography*, Mar 2005.

[13] T. E. Gbondo-Tugbawa, *Chip-Scale Modeling of Pattern Dependencies in Copper Chemical Mechanical Polishing Process*. PhD thesis, Massachusetts Institute of Technology, 2002.

[14] "Quickcap user manual," in *http://www.magma-da.com/*.

[15] R. Tian, D. Wong, and R. Boone, "Model-based dummy feature placement for oxide chemical-mechanical polishing manufacturability," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 20, no. 7, pp. 902–910, 2001.

[16] 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.

[17] H. Bakoglu, *Circuits, Interconnects and Packaging for VLSI*. Addison-Wesley, 1990.

[18] 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.

[19] 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.

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

[21] "Berkeley predictive technology model," in *http://www-device.eecs.berkeley.edu/ ptm*.

[22] 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.

[23] 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.

[24] N. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*. Addison-Wesley, 2004.

[25] A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering*. Addison-Wesley, 1994.