Joining the Design and Mask Flows for Better and Cheaper Masks

P. Gupta[a], A. B. Kahng[a,b], C.-H. Park[a], P. Sharma[a], D. Sylvester[c] and J. Yang[c]

[a]UCSD ECE Dept., [b]UCSD CSE Dept., [c]U. Michigan EECS Dept.

ABSTRACT

Today's design-manufacturing interfaces have only minimal information exchange. Lack of information on either side leads to under-performance due to too much guardbanding, and increased mask cost and increased turnaround time due to over-correction. In this work we present techniques that simultaneously utilize design and manufacturing information to improve mask quality and to reduce mask cost.

1. INTRODUCTION

Optical lithography has been a key enabler of the aggressive IC technology scaling implicit in Moore's Law. Minimum feature sizes have outpaced the introduction of advanced lithography hardware solutions, so that gate-length and CD tolerances are extremely difficult to achieve. As a result, resolution enhancement techniques (RETs) such as optical proximity correction (OPC), phase shift masks (PSM), and Off-Axis Illumination (OAI) are being pushed ever closer to fundamental resolution limits.[1] RETs, which are imperative during mask-data preparation (MDP) today, increase mask cost and should be used judiciously. Existing design-manufacturing interfaces suffer from lack of communication across disciplines and/or tool sets. The result is that both design and manufacturing have limited information about each other and conservative assumptions must be made on both sides. This leads to sub-optimal performance due to too much guardbanding, and high mask costs and large turnaround time due to over-correction.

We present three techniques that link design and manufacturing for better and cheaper masks.

- Design-aware optical proximity correction (OPC). Here we attempt to pass designer's intent to OPC and reduce over-correction. OPC can increase the mask data volume by over 5X; mask cost and turnaround time are proportional to mask data volume. Our technique selectively applies levels of OPC, with higher levels of OPC being applied to devices that are considered critical to circuit performance. We show up to a 34% reduction in mask data volume.

- Placement for better Depth Of Focus (DOF). We investigate the feasibility and benefit of minor placement modifications to enhance printability. OAI improves resolution at certain pitches at the expense of others. Pitches where resolution is deteriorated due to OAI, *forbidden pitches*, are avoided using Sub-Resolution Assist Features (SRAFs). We describe a methodology that perturbs placement to reduce the occurrence of forbidden pitches and number of required SRAFs.

- Gate-length biasing. Leakage power has become one of the most critical design concerns for the system-level chip designer. While lowered supplies and aggressive clock gating can achieve dynamic power reduction, these techniques increase leakage power and therefore cause its share of total power to increase. Another problem that we address here is that of leakage power variability. We investigate the use of slightly increased gate-lengths for devices that are not timing-critical*. Unlike multi-V_{th} techniques, gate-length biasing requires no additional masks and may be performed as an MDP. Our results show leakage power reduction of up to 25% and leakage variability reduction of up to 54% with under a 4% delay penalty.

The rest of the paper is organized as follows. In Section 2, we describe our design-aware methodology for OPC effort reduction. Section 3 describes our placement alteration technique to enhance design printability. Section 4 presents our gate-length biasing methodology for leakage and leakage variability reduction. Section 5 concludes and mentions on-going work.

*It is well known that increasing gate-length reduces leakage power but increases delay.

2. DESIGN-AWARE OPC

In this work we focus on OPC, which is a major contributor to mask costs as well as design turnaround time. More than a 5X increase in data volume and several days of CPU runtime are common side effects of OPC insertion in current designs.[14] OPC affects MDP, defect inspection (and implicitly defect repair), and the mask-writing process itself. Today, variable-shaped electron beam mask writers, in combination with vector scanning*, comprise the dominant approach to high-speed mask writing. In the standard MDP flow, the input GDSII layout data is converted into the mask writer format by *fracturing* into rectangles or trapezoids of different dimensions. With OPC applied during MDP, the number of line edges increases by 4-8X over a non-OPC layout, driving up the resulting GDSII file size as well as fractured data (e.g., MEBES format) volume.[11] Mask writers are hence slowed by the software for e-beam data fracturing and transfer, as well as by the extremely large file sizes involved. Moreover, increases in the fractured layout data volume† lead to disproportionate, super-linear increases in mask writing and inspection time. Compounding these woes is the fact that the total cost to produce low-volume parts is now dominated by mask costs[10] since masks costs cannot be amortized over a large number of shipped products. There is a clear need to reduce the negative implications of OPC on total design cost while maintaining the printability improvements provided by this crucial RET step.

We observe that OPC has traditionally been treated as a purely geometric exercise wherein the OPC insertion tool tries to match every edge as best as it can. As we show in our work, and has been observed by Gupta et al.,[16] such "over-correction" leads to higher mask costs and larger runtimes. A first approach to driving RET explicitly by performance considerations was proposed at DAC-2003 by Gupta et al..[16] Their work proposes selective OPC based on an assumption of several available levels of correction. We describe a design-aware OPC methodology that is demonstrated to be highly implementable within the limitations of current industrial design flows.

2.1. Practical Methodology for Design-Aware OPC

We devise a flow to pass design constraints on to the OPC insertion tool in a form that it can understand. As previously mentioned, OPC insertion tools are driven by *edge placement error* (EPE) *tolerances*. Typical model-based OPC techniques break up edges into *edge-fragments* that are then iteratively shifted outward or inward (with respect to the feature boundary) based on simulation results, until the estimated wafer image of each edge-fragment falls within the specified EPE tolerance. EPE (and hence EPE tolerance) is typically signed, with negative EPE corresponding to a decrease in CD (i.e., moving the edge inward with respect to the feature boundary). An example of a layout fragment and its EPE is shown in Figure 1. Mask data volume is heavily dependent on the assigned EPE tolerance that the OPC insertion tool is asked to achieve. For example, Figure 2 shows the change in MEBES file size for cell with applied OPC as the EPE tolerance is varied. In this particular example, loosened EPE tolerances can reduce data volume by roughly 20% relative to tight control levels.

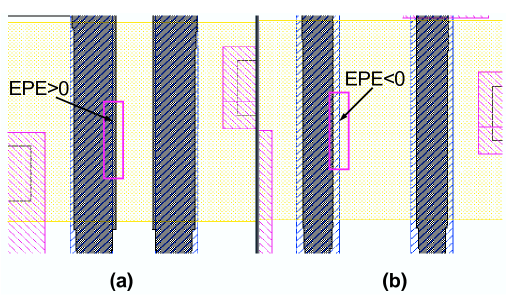

Figure 1. The signed edge placement error (EPE).

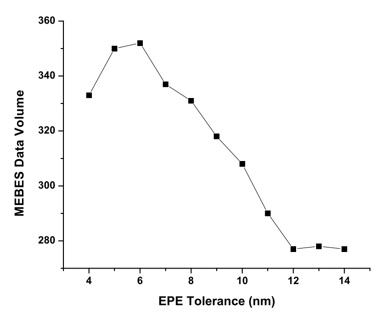

Figure 2. Mask data volume (kB) vs EPE tolerance for a NAND3X4 cell in TSMC 130nm technology.

Since model-based OPC corrects for pattern-dependent CD variation, which is systematic and predictable, we assert that OPC actually determines *nominal timing*, rather than parametric yield as assumed in the work of Gupta et al..[16] This allows us to base our OPC insertion methodology on traditional corner-case timing analysis tools instead of (currently non-existent from a commercial standpoint) statistical timing analysis tools. Our methodology adopts a slack budgeting based approach - as opposed to a sizing based approach used previously[16] -

*Compared to traditional raster scanning, vector scanning allows features to be scaled up or down in size while maintaining sharpness, but the write cost is proportional to feature complexity: the mask pattern must be decomposed into a set of disjoint "shots" or "flashes", each of which takes roughly constant (unit) time.

†E.g., according to the 2003 ITRS,[19] the maximum single-layer MEBES file size increases from 216GB in 90nm to 729GB in 65nm.

to determine EPE tolerance values for every feature in the design. For simplicity, our description and experiments reported here are restricted in two ways: (1) we apply selective EPE tolerances in OPC to only gate poly features, and (2) every gate feature in a given cell instance is assumed to have the same EPE tolerance (the approach may be made more fine-grained using the same techniques that we describe). Figure 3 shows our design-aware OPC flow. The quality of results generated by the flow are measured as MEBES data volume of fractured post-OPC insertion layout shapes as well as OPC insertion tool runtime, which can be prohibitive when run at the full-chip level. In the remainder of this section, we describe details of the major steps of Figure 3.

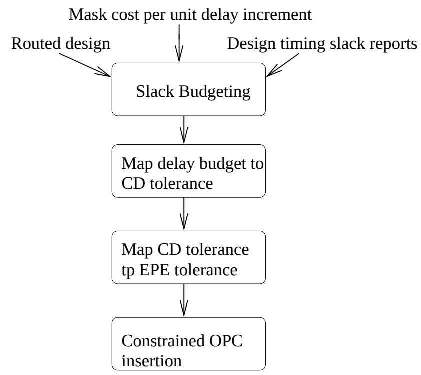

Figure 3. Design-aware OPC flow to find quantified edge placement error tolerances for layout features and drive OPC with them.

To map delay budgets found from a linear programming based formulation to CD tolerances, we require characterization of a standard-cell library with varying gate-lengths. Using such an augmented library, along with input slew and load capacitance values for every cell instance, we can map delay budgets to the corresponding gate-lengths. For example, if a particular instance with specified load and input slew rate has a delay budget of 100ps, then we can select the longest gate-length implementation of this gate type that meets this delay. This largest allowable CD will lead to a more easily manufactured gate with less RET effort. CD tolerance of each cell in the design is calculated by subtracting budgeted gate-lengths from nominal gate-lengths.

The next step in our flow maps CD tolerances to signed EPE tolerances. Again, obtaining EPE tolerances is crucial since this is the parameter which OPC insertion tools understand and can exploit. As noted above, in this work we assume positive and negative EPE tolerance to be the same. Since CD is determined by two edges, the worst-case CD tolerance is twice the EPE tolerance.

In most lithography processes, gates shrink along their entire width such that the printed gate-length is always smaller than the drawn gate-length, except at the corners of the critical gate feature. OPC typically biases the gate-length such that corrected gate-length is *larger* than the designer-drawn gate-length. Thus, model-based OPC shifts edges *outward*, i.e., in the "positive" direction, until it meets the EPE tolerance specification. If the step size of each edge move is small enough, the EPE along the gate-width will always be negative (since we are approaching the larger nominal gate-length value starting from the smaller printed gate length value). As a result, actual printed gate-length will almost always be smaller than the drawn gate-length, leading to leakier but faster devices.

To achieve a more unbiased deviation from nominal, we exploit the behavior of the OPC tool by applying simple pre-biasing of gate features in an attempt to achieve EPE tolerances that are equal to CD tolerance. Specifically, we pre-bias each gate feature by its intended EPE tolerance. An example of the average CD for a specific gate poly with and without pre-biasing is shown in Figure 4. It is clear that pre-biasing achieves its goal of attaining average CDs that are very close to the target CD (130nm in our case). Another point illustrated in Figure 4 is that the variation in CD (measured as the standard deviation of CD taken across all edge-fragments) grows as the EPE tolerance is relaxed.

2.2. Experimental Setup and Results

Now we describe our experiments and the results obtained to validate the design-aware OPC methodology.

Test Cases. We use seven combinational benchmarks drawn from ISCAS85 suite of benchmarks and Opencores.[23] These benchmark circuits are synthesized, placed and routed in a restricted TSMC 0.13 μm library containing 32 cell macros with cell types of BUF, INV, NAND2, NAND3, NAND4, NOR2, NOR3, and NOR4. The test cases are *c432* (337 cells), *c5315* (2093 cells), *c6288* (4523 cells), *c7552* (2775 cells), and *alu128* (12403 cells).

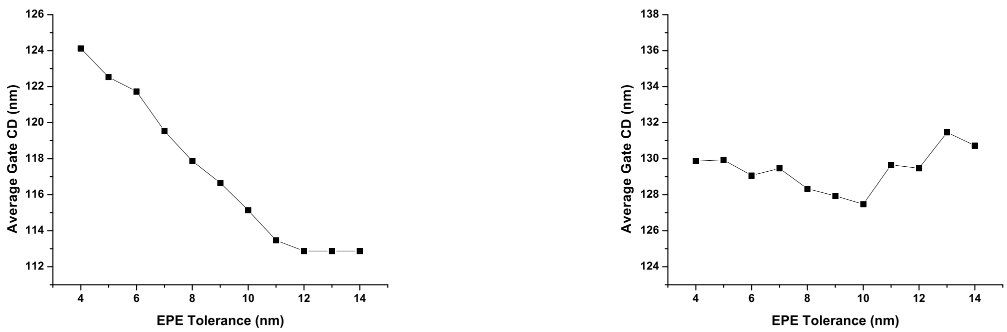

Figure 4. Comparison of average printed gate CD with and without pre-bias for the cell macro NAND3X4

Library Characterization. We assume a total of EPE tolerance levels ranging from ±4nm to ±14nm. Corresponding to each EPE tolerance, the worst case gate-length is $130nm + EPE\_Tolerance$. We map cell delays to EPE tolerance levels by creating multiple .lib files for each of the 10 worst case gate-lengths using circuit simulation. For simplicity, we neglect the dependence of delay on input slew in our analysis but this could easily be added to the framework.

Expected mask cost for each cell type is extracted as a function of EPE tolerance. We run model-based OPC using Calibre on individual cells followed by fracturing to obtain MEBES data volume numbers for each (cell, tolerance) pair. Though the exact corrections applied to a cell will depend somewhat on its placement environment, stand-alone OPC is fairly representative of data volume changes with changing EPE tolerance. Finally, we calculate the sensitivity of mask cost to delay change under the assumption that cost reduction is a linear function of delay increase. This assumption is based on linearity between gate delay and CD as well as the rough linearity shown in Figure 2 between data volume and EPE tolerance. We then build a .lib-like look-up table of correction cost sensitivities (with respect to the tightest EPE tolerance of 4nm).

Design-aware OPC with Calibre. Our OPC flow involves assist-feature insertion followed by model-based OPC. The EPE tolerance is assigned to each gate by the *tagging* command within Calibre. We first separate the entire poly layer into gate poly and field poly components. The field poly tolerance is taken to be ±14nm while gate poly tolerance ranges from ±4nm to ±14nm. We take 1nm as our step size* when applying OPC to obtain very precise correction levels. We set the iteration number to the minimum value beyond which adding mask cost and CD distribution show little sensitivity to OPCs, which is found experimentally. After model-based OPC is applied, we perform 'printimage' simulations in Calibre to obtain the expected as-printed wafer image of the layout. Average gate CD and its standard deviation are extracted from this wafer image. The corrected GDSII is fractured into MEBES using CalibreMDP. The total mask data volume is then determined based on the MEBES file sizes.

Results. We synthesize the benchmark circuits using *Synopsys Design Compiler*. Place and route is performed using *Cadence Silicon Ensemble*. *Synopsys Primetime* is used to output the slack report of the top 500 critical paths as well as the load capacitance for each driving pin. As noted above, STA is run with a modified 134nm (tightest EPE tolerance) library with pin capacitances corresponding to 144nm (loosest EPE tolerance) to remain conservative after slack budgeting. We use *Cplex v8.1*[15] as the mathematical programming solver to solve the budgeting linear program. Since the circuit sizes are fairly small, we use only a single iteration to solve the budgeting problem.

Table 1 compares the runtime and data volume results for design-aware OPC and traditional OPC. The budgeting approach ensures that there is no timing degradation going from the traditional to the design-aware OPC flow. Moreover, unlike sizing, budgeting does not involve iterations with timing analysis. As a result budgeting runtimes are negligibly small ranging from 1s to 11s. The important result is the amount of mask cost reductions achieved whether measures as runtime of model-based OPC or fractured MEBES data volume. Design-aware OPC flow reduces MEBES data volume by 17%-24%. Such reductions directly translate to substantial mask-write time improvements. OPC runtimes are improved by 6%-34%. These percentage numbers translate to a huge absolute turnaround time savings. For instance, the design-aware OPC flow saves 5 hours compared to the traditional OPC flow on a small 12000 gate benchmark.

*Step size is the minimum perturbation to an edge that model-based OPC can make. Smaller step sizes lead to better correction accuracy at the cost of runtime.

Test case	Traditional OPC Flow				Design-aware OPC Flow							
	CD Distribution All Gates (nm)		OPC Runtime (s)	Delay (ns)	Budgeting Runtime (s)	CD Distribution				OPC Runtime (s)	Delay (ns)	Normalized MEBES Volume
						All Gates (nm)		Critical Gates (nm)				
	mean	σ				mean	σ	mean	σ			
alu128	126.1	1.48	51516	3.28	11	131.5	4.93	130.8	2.04	33535	3.28	0.76
c7552	126.2	1.89	7149	1.59	4	132.0	4.77	130.1	1.99	5142	1.59	0.78
c6288	126.0	1.37	12830	5.21	9	131.4	4.45	129.7	1.27	9710	5.21	0.82
c5315	126.1	1.82	4539	1.94	3	131.7	4.70	129.7	1.89	4247	1.94	0.79
c432	126.8	1.57	1020	1.33	1	131.3	3.90	129.9	1.67	737	1.33	0.83

Table 1. Impact of design-aware OPC optimization on Cost and CD. All runtimes are based on a 2.4GHz Xeon machine with 2GB memory running Linux.

3. PLACEMENT FOR BETTER DOF

Combinations of RET techniques can provide certain advantages for lithography manufacturing, e.g., OAI and OPC, together with SRAF, achieve enhanced CD control and focus margin at minimum pitch. However, when OAI is used, there will always be other (non-minimum) pitches for which the angle of illumination works with the angle of diffraction to produce a bad distribution of diffraction orders in the lens. These pitches are called *forbidden pitches* because of their lower printability, and designers should avoid such pitches in the layout. However, it is very difficult to consider all possible forbidden pitches in the design stage, particularly since the forbidden pitches are dependent on optical conditions which are often tuned in manufacturing. The resulting *forbidden pitch problem* for the manufacturing-critical poly layer must be solved before detailed routing, since routing "locks in" the poly layer layout. At the same time, we wish to address the forbidden pitch problem as late as possible, to avoid extra rework upon modification of the manufacturing process recipe. In this paper, we describe a novel dynamic programming-based algorithm for *AFCorr* (Assist-Feature Correctness), which uses flexibility in detailed placement to avoid forbidden pitches and the manufacturing uncertainty caused by them.

3.1. Assist Feature Correction Methodology

Modified Design and Evaluation Flow. To account for new geometric constraints arising due to SRAF OPC in physical design, we add forbidden pitch extraction and post-placement optimization into the current ASIC design methodology. Figure 5 shows the modified design and evaluation flows in the regime of forbidden pitch restrictions. Of course, we must assume that the library cells themselves have been laid out with awareness of forbidden pitches, and indeed our experiments with commercial libraries confirm that there are no forbidden pitch violations in poly geometries within commercial standard cells. SRAF insertion rules for enhancing DOF margin are determined based on best and worst focus models.* Post-placement optimization generates a new placement which is more conducive to insertion of SRAFs, thus allowing a larger process window to be achieved. The two layouts generated by conventional and assist-correct flow undergo comprehensive SRAF OPC. The amount and impact of the applied RET is a function of the circuit layout. Thus we can evaluate how assist-correct placement impacts circuit performance and printability/manufacturability using measures of SRAF and EPE. The following subsections give more details of forbidden pitch extraction and its design implementation.

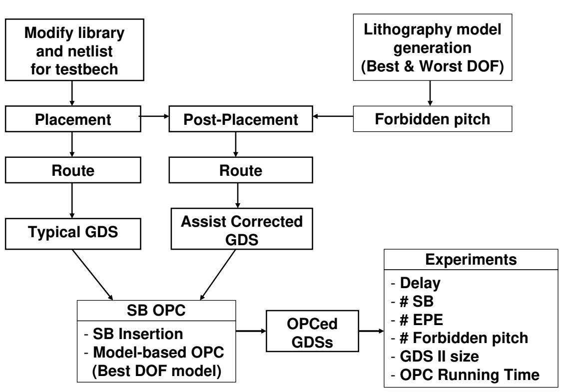

Figure 5. The modified design and evaluation flows : Note the added steps of forbidden pitch extraction and post-placement optimization to ASIC design flow

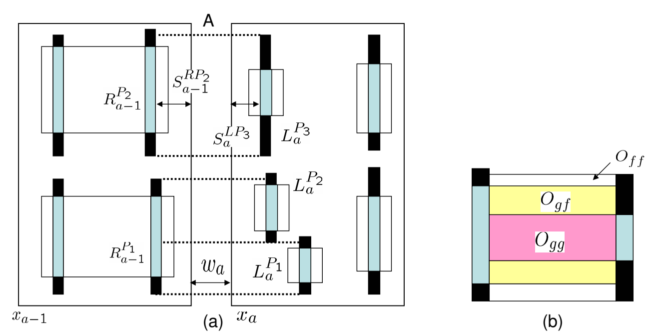

Figure 6. (a) multiple interactions of gate-to-gate, gate-to-field, and field-to-gate, and (b) overlapped area in the region A of (a).

*In general, the best focus is shifted from zero to about $0.1\mu m$ due to refraction in the resist. The worst defocus is the maximum allowable defocus corner for manufacturability in a lithography system.

SRAF and Forbidden Pitch Rules. Lack of space prohibits insertion of a sufficient number of SRAFs, and as a result patterns violate CD tolerance through defocus. Forbidden pitches are pitch values for which the tolerance of a given target CD is violated. Allowable pitches are all pitches other than forbidden pitches.

Our SRAF insertion rule is initially generated based on the theoretical background given by Shi et al..[2] Positioning of SRAFs is then adjusted based on OPC results. Large CD degradation through-pitch increases pattern bias as model-based OPC is applied, and this requires trimming of the SRAF rule to guarantee better process margin and prevent the SRAFs from printing.* After applying SRAF OPC with a best-focus model, test patterns are simulated with the worst-defocus model. This evaluation yields the forbidden pitches, considering maximum printability and manufacturability. The forbidden pitch rule is determined based on CD tolerance and worst defocus level which can be changed by requirements of device performance and yield. We report that CD tolerance is assumed to be $\pm 10\%$ of minimum line width while the worst defocus level is assumed to be $0.5\mu m$.

Assist Feature Correction. Given a cell C_a, let LP^a and RP^a be the sets of valid poly geometries in the cell which are located closest to left and right outlines of the cell respectively. Only the geometries with length larger than minimum allowable length of SRAF features are considered. Define $s_a^{LP_i}$ to be the space between the left outline of the cell and the i^{th} left border poly geometry. Also assume a set $AF = AF_1, \ldots, AF_m$ of spacings which are "assist-correct". I.e., if the spacing between two gate poly shapes belongs to the set AF, then required number of assist features can be inserted between the two poly geometries. AF_j denotes the j^{th} member of the set of assist-feature correct spacings AF when AF is assumed to be sorted in increasing order. Note that the set AF may contain a number of spacings which correspond to varying SRAF widths. Let w_a denote the width of cell C_a and x_a denote its placement coordinate (leftmost) in the given standard cell row indexed from left to right. Then the assist-correct placement perturbation problems is as follows.

$$\begin{aligned} &\text{Minimize} \sum \mid \delta_i \mid \\ \delta_{a+1} + x_{a+1} - x_a - \delta_a - w_a + s_{a+1}^{LP_k} + s_a^{RP_g} &\in AS \\ &\text{s.t. } LP_k \text{ and } RP_g \text{ overlap} \end{aligned}$$

The objective can be made aware of cells in critical paths by a weighting function. Since the available number of allowable spacings is very small, obtaining a completely assist-correct solution is usually not possible in a fixed cell row width context. Therefore, a more tractable objective is to minimize the expected CD error at a predetermined defocus level. We solve this "continuous" version of the above problem by a dynamic programming approach. The recurrence relation is given below.

$$\begin{aligned} Cost(1,b) &= \mid x_1 - b \mid \\ Cost(a,b) &= \lambda(a) \mid (x_a - b) \mid + \\ Min_{i=x_{a-1}-SRCH}^{x_{a-1}+SRCH} &\{Cost(a-1,i) + HCost(a,b,a-1,i)\} \end{aligned}$$

$Cost(a,b)$ is the cost of placing cell a at placement site number b. The cells and the placement sites are indexed from left to right in the standard cell row. We restrict the perturbation of any cell to $\pm SRCH$ placement grid points. This is done to contain the delay and runtime overheads of AFCorr placement post-processing. λ is a factor which decides the relative importance of preserving the initial placement and the final AFCorr benefit achieved. It is a function of the cell instance. In the current implementation it is directly proportional to the number of critical paths that pass through the given cell instance. $HCost$ corresponds to the printability deterioration in defocus conditions for the vertically oriented poly geometries closest to the cell boundary. It depends on the difference between the current nearest-neighbor spacing of the polys and the closest assist-feature correct spacing. The method of computing $HCost$ is shown in Figure 7.

O_{gg}, O_{ff} and O_{gf} correspond to length of overlapped area in the cases of gate-to-gate and field-to-field, gate-to-field poly as shown in Figure 6. Also, c_{gg}, c_{ff}, and c_{gf} are proportionality factors which decided relative importance of printability for gate and field poly. Typically, gate poly geometries need to be better controlled through process as they impact performance directly. Therefore, a typical order is $c_{gg} \geq c_{fg} \geq c_{ff}$. $slope(j)$ is defined as delta CD difference over delta pitch between AF_j and AF_{j+1}. Thus perturbation cost is a function of *slope*, length and weight of overlapped polys, and space for SRAF insertion. The algorithm takes a legal placement as an input and outputs a legal placement with better depth of focus properties. The runtime of the algorithm is $O(n \times SRCH^2)$.

*More complicated approaches to SRAF rule generation may involve co-optimization of model-based OPC and SRAF insertion. We do not address such involved optimizations of OPC, since the focus of our work is OPC-aware design and not OPC itself.

HCost(a,b,a-1,i) of Cell C_a

Input:
User-defined weight for overlapping field polys : c_{ff}
User-defined weight for overlapping gate polys : c_{gg}
User-defined weight for overlapping gate and field polys : c_{gf}
Origin x (left) coordinate and length of cell C_a = b
Origin x (left) coordinate and length of cell C_{a-1} = i
Width of cell $C_a = w_a$
Width of cell $C_{a-1} = w_{a-1}$

Output:
Value of $HCost$

Algorithm:
01.**Case** $a = 1$: $HCost(1, b) = 0$
02.**Case** $a > 1$ **Do**
03. N:= cardinality of the set RP_{a-1}
04. M:= cardinality of the set LP_a
05. **For** $(k = 1 ; k = N ; k = k + 1)\{$
06. **For** $(g = 1 ; g = M ; g = g + 1)$ {
Let $Hspace(k, g)$ denote the horizontal spacing between RP^k_{a-1} and LP^g_a. $O_{ff}(k, g)$, $O_{fg}(k, g)$ and $O_{gg}(k, g)$ denote the field-to-field, field-to-gate and gate-to-gate overlap lengths between RP^k_{a-1} and LP^g_a.
$slope(j)$ is the degradation of CD with respect to pitch when spacing between two poly geometries is between AF_j and AF_{j+1}.
/* Calculate overlap weight between RP^k_a and LP^g_{a-1} */
07. $weight(g, k) = slope(j) \times (Hspace(k, g) - AF_j)$
$\times(c_{ff}O_{ff}(k, g) + c_{gf}O_{gf}(k, g) + c_{gg}O_{gg}(k, g))$
s.t. $AF_{j+1} > Hspace(k, g) \geq AF_j$,
08. $Hcost(a, b, a-1, i) \mathrel{+}= weight(g, k)$
}
}

Figure 7. $HCost$ calculation.

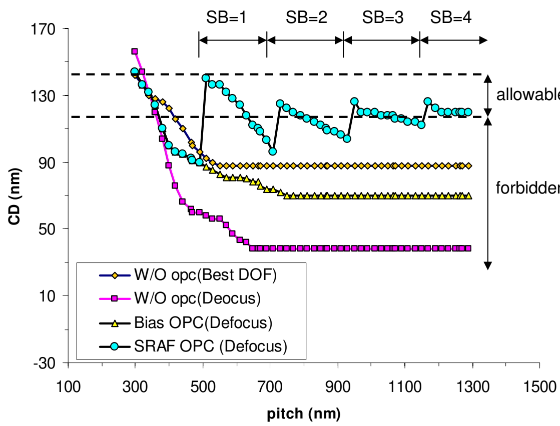

Figure 8. Evaluation of proximity plots in through-pitch: Best focus without OPC, worst defocus without OPC, worst defocus with BIAS OPC, worst defocus with SRAF OPC.

3.2. Experiments and Discussion

Experimental Setup. We synthesize *alu*128 benchmark design from *Opencores* in *Artisan TSMC 0.13µm* and *Artisan TSMC 0.09µm* libraries using *Synopsys Design Compiler v2003.06-SP1*. *alu*128 synthesizes to 13279 cells and 8722 cells in 130nm and 90nm technologies respectively. The synthesized netlists are placed with row utilization ranging from 50% to 90% using *Cadence First Encounter v3.3*. All designs are trial routed before running timing analysis. On the lithography side, we use *KLA-Tencor Prolith* to generate models for OPC. *Mentor Graphics Calibre* is used for model-based OPC, SRAF OPC and optical rule checking (ORC). Simulation is performed with wavelength λ =248nm and numerical aperture NA = 0.6 for 130nm and λ =193nm and NA = 0.75 for 90nm. An annular aperture with σ =0.85/0.65 is used for both processes.

Proximity plots with fixed line width of 0.13μm are illustrated in Figure 8. Exposure dose focuses on the pattern in the minimum pitch of 0.13μm. CD degradation increases through-pitch as the defocus level increases. Patterns in the pitches of over 0.4μm before OPC are outside the allowable tolerance range at the worst defocus of 0.5μm. After BIAS OPC, pitches up to 0.38μm are allowable for CD tolerance while all pitches larger than than 0.38μm should be forbidden. After evaluating SRAF OPC patterns with the worst defocus model, a set of forbidden pitches is obtained as follows: [0.37, 0.509], [0.635, 0.729], [0.82, 0.949], and [1.09, 1.169]. Forbidden pitches still remain after SRAF OPC even though OPC considerably reduces forbidden pitches in comparison to BIAS OPC. SRAF rules are generated based on the criteria mentioned above, with results as shown in Table 2. SRAF width is 60nm for 130nm and 40nm for 90nm technology.

	0.13μm Litho.		0.09μm Litho.	
	Pitch($X : \mu m$)	Slope	Pitch($X : \mu m$)	Slope
#SRAF = 0	$0 \leq X < 0.51$	0.28	$0 \leq X < 0.41$	0.162
#SRAF = 1	$0.51 \leq X < 0.73$	0.22	$0.41 \leq X < 0.57$	0.075
#SRAF = 2	$0.73 \leq X < 0.95$	0.105	$0.57 \leq X < 0.73$	0.062
#SRAF = 3	$0.95 \leq X < 1.17$	0.07	$0.73 \leq X < 0.89$	0.050
#SRAF = 4	$1.17 \leq X$	0.02	$0.89 \leq X$	0.012

Table 2. SRAF rule table in 0.13μm and 0.09μm lithography.

Experimental Results. The post-placement optimization is performed based on forbidden pitches and slopes of CD error within them. After AFCorr placement perturbation, we obtain a new placement wherein the coordinates of cells have been adjusted to avoid the forbidden pitches. We use three printability quality metrics. *Forbidden Pitch Count* is the number of border poly geometries estimated as having greater than 10% CD error through-focus. *EPE Count* is the number of edge fragments on border poly geometries having greater than 10% edge placement error at the worst defocus level. This is estimated by ORC. *SB Count* is the total number of scattering bars or SRAFs inserted in the design. A higher number of SRAFs indicates less through-focus variation and hence is desirable. We use $c_{fg} = c_{gg} = c_{ff} = 0.33$, $lambda(a) =$

	Utilization (%)	90		80		70		60		50	
	Flow	Typical	AFCorr	Typical	AFCorr	Typical	AFCorr	Typical	AFCorr	Typical	AFCorr
130nm	# Forbidden	8315	1305	6883	389	4224	121	2347	38	185	3
	# SB	158987	169158	173673	183172	185493	191874	195741	198948	212079	212365
	# EPE	6572	2462	5098	1312	4198	1210	2760	742	216	50
	Runtime (s)	6721	6732	6839	6899	6878	6923	6943	6944	7032	7039
	GDS (MB)	42.9	41.9	41.8	42.3	42.2	42.2	44.9	44.9	45.2	45.4
	Delay (ns)	4.21	4.49	4.547	4.444	4.501	4.372	5.142	4.976	5.051	4.942
90nm	# Forbidden	5171	965	3484	510	1801	323	1130	291	53	10
	# SB	115652	124262	139182	142167	153904	155120	164264	165397	182572	182579
	# EPE	9118	2505	6229	1292	2468	635	2134	600	349	33
	Runtime (s)	4835	5011	5451	5535	5529	5632	5685	5698	5943	5944
	GDS (MB)	41.1	42.3	41.2	43.2	42.2	42.3	42.9	42.8	43.6	43.6
	Delay (ns)	2.478	2.305	2.458	2.602	2.522	2.47	2.867	3.176	3.113	3.046

Table 3. Summary of AFCorr results. Runtime denotes the runtime of SRAF insertion and model-based OPC. The AFCorr perturbation runtime ranges from 2 to 3 minutes for all test cases. GDS size is the post SRAF OPC data volume.

$\frac{sitewidth}{10}$ × number of top 200 critical paths passing through cell a and $SRCH = 5$. All the results have been tabulated in Table 3. Reductions of EPE and forbidden pitch are investigated in each utilization. The increase in total number of SRAFs inserted is shown in Table 3. Forbidden Pitch Count improves 84%-98% in 130nm and 72%-90% in 90nm. EPE Count enhances 62%-76% in 130nm and 74%-85% in 90nm. In addition, SB Count has the range of improvement 0.1%-6.4% for 130nm and 0%-7.4% for 90nm. Note that these numbers are small as they correspond to the entire layout rather than just the border poly geometries.

The number of total SRAFs increases as the utilization* decreases due to increased white space between cells. AFCorr benefit decreases for lower utilization due to the increased availability of whitespace for SRAF insertion. Due to additional number of SRAFs inserted there is a small increase in SRAF OPC runtime ($< 3.6\%$) and final data volume ($< 3\%$). The change in estimated post-trial route circuit delay ranges from -7% to +11%.

4. GATE-LENGTH BIASING

Dynamic power reduction techniques like lowered supply voltages and aggressive clock gating increase leakage power and therefore cause its share of total power to increase. The only mainstream approach to reduce leakage power during *active*, or *runtime*, mode is the multi-V_{th} manufacturing process.[3–5] In this approach, cells in non-critical paths are assigned a high V_{th} while cells in critical paths are assigned a low V_{th}. The major drawback to this technique has traditionally been the rise in process costs due to additional steps and masks. The increased costs have been outweighed by the substantial leakage reductions they provide, and multi-V_{th} processes are now standard.

Manufacturers face the additional challenge of *leakage variability*: recent data from Intel indicates that leakage of microprocessor chips from a single 180nm wafer can vary by as much as 20x. Leakage variability is caused by variability in V_{th}, which occurs in part due to random doping fluctuations, as well as worsened DIBL (Drain Induced Barrier Lowering) and short-channel effects (SCE) in devices with lower channel doping. High leakage variability forces several chips to be discarded due to their unacceptably high leakage power.

The problem of rising leakage and its variability requires new, *manufacturable* techniques. Here, we describe a novel methodology based on increasing the device gate-lengths slightly (under 10%) to reduce leakage and its variability. It is well known that leakage power decreases exponentially, and delay increases linearly, with increasing gate-length. Thus, it is possible to increase gate-length only marginally to take advantage of the exponential leakage reduction, while impairing performance only linearly. Further, cells in which device gate-lengths have only been marginally increased are layout interchangeable with their nominal versions. Thus, the technique can be applied as a post-layout or post-reticle enhancement step. To have minimal circuit delay penalty, we apply the technique to those devices that do not appear on critical paths. To highlight the real value of the technique, we first apply the multi-V_{th} technique, and then use gate-length biasing to show further reduction in leakage.

4.1. L_{Gate} Biasing Methodology

In this section we describe the proposed gate-length biasing methodology. We characterize and then augment a standard-cell library, such that each master also has a *biased* L_{Gate} variant. A sizing tool is then used to incorporate slower but low-leakage cells into non-critical paths, while retaining faster, high-leakage cells in critical paths. Reflecting the experiments below, our discussion focuses on the introduction of a single biased variant for each cell in the library, and on an industry 130nm process technology. Of course, the approach also extends to multiple biased variants.

*Cell utilization is the percentage of floorplan area used for actual cell placements. Lower utilization implies larger whitespace in the design

L_{Gate} Biasing Granularity. Gate-length biasing can be performed at several levels of granularity, namely, technology-level, cell-level and device-level. Finer granularity leads to more difficult implementation but more flexibility in optimization and potentially larger leakage benefits. We have considered the following three levels of biasing granularity.

1. *Technology-Level.* All gates in the library have the same biased L_{Gate}. As a result there are *exactly* two distinct gate-lengths (in a dual-L_{Gate} approach) in the technology library.
2. *Cell-Level.* Every library cell master has its own specific biased gate-length. All devices within a given cell share this characteristic L_{Gate}, but different cell masters are allowed to have different biased gate-lengths.
3. *Device-Level.* Ideally, a device-level gate-length biasing approach will allow independent biasing of every gate in the library. However, as this is computationally impractical within our characterization and search framework* we restrict ourselves to independent biasing of PMOS and NMOS devices within a cell. In other words, all PMOS devices within a given cell master have the same gate-length bias which is independent of the bias of NMOS devices in the cell as well as the PMOS devices in other cells. This simplification permits us to exhaustively search for the "optimal" biased L_{Gate} for devices. The rationale for this biasing approach is that in complementary MOS technologies, the NMOS devices in a cell typically have identical topology (e.g. series connected for NAND gates) and PMOS devices have identical topology (e.g. parallel connected for NAND gates). Leakage has a strong dependence on topology, with stacked devices leaking much less than unstacked ones.[6]

Biased-L_{Gate} Selection. The key question in our methodology is the value of L_{Gate} for each transistor in the cells. We consider less than 10% biasing of the gate-length. The reasons for such a small bias are as follows.

- An increase in drawn dimension that is less than the layout grid resolution (typically 10nm for 130nm technology) ensures pin-compatibility with the unsized version of the cell. This is very important to ensure that multi-L_{Gate} optimizations can be done post-placement or even after detailed routing without ECOs. In this way, we retain the layout transparency that has made multi-V_{th} optimization so adoptable within chip implementation flows. Biases smaller than the layout grid-pitch also ensure design-rule correctness for the biased cell layout, as long as the unbiased version is correct.
- The nominal gate-length of the technology is usually very close to or beyond the "knee" of the leakage vs. L_{Gate} curve. For large bias, the advantage of super-linear dependence of leakage on gate-length is lost.
- From a manufacturability point of view, having two prevalent pitches (which are not close enough) in the design can harm printability properties (i.e., size of process window). Note that we retain the same poly-pitch as the unbiased version of the cell. There is a small decrease in spacing between gate poly geometries but it is still well within minimum spacing required by the process.

Impact of L_{Gate} bias on delay and leakage is computed by detailed (HSPICE) circuit simulation. To determine the appropriate gate-length biasing of devices in a cell, we restrict the cell delay penalty a prescribed $delay_{penalty}$ and bias the devices to minimize cell leakage power. Our bias selection uses $delay_{penalty} = 10\%$†.

Library Generation. An important component of the methodology is layout and characterization of the dual-L_{Gate} library. Since we investigate very small biases to the gate-length, the layout of the biased library cell does not need to change except for simple automatic scaling of dimensions. Moreover, since the bias is smaller than the minimum layout grid pitch, design rule violations are highly unlikely. Of course, after the slight modifications to layout, the biased versions of the cell are put through the standard extraction and power/timing characterization process.

4.2. Experiments and Results

We now describe our test flow for validation of the L_{Gate} biasing methodology, and present experimental results. We consider up to two gate-lengths and two threshold voltages. We perform experiments for the following scenarios – Single-V_{th}, single-L_{Gate} (SVT-SGL); Dual-V_{th}, single L_{Gate} (DVT-SGL); Single-V_{th}, dual-L_{Gate} (SVT-DGL); Dual-V_{th}, dual L_{Gate} (DVT-DGL). The dual-V_{th} flow uses nominal and low values of V_{th} while the single-V_{th} flow uses only the low value of V_{th}. The basic elements of our flow are a dual L_{Gate} library that captures the effects of L_{Gate} biasing on leakage, delay and input capacitance; and a tool to perform leakage-aware sizing.

*We use exhaustive search to find the best biased gate-length values. When every device L_{Gate} is allowed to vary independently, the search space becomes too large; effective identification of biased variants in this ideal framework is an open direction.

†The number 10% is determined empirically. Larger bias can lead to larger per-cell leakage saving at a higher performance cost. However, in a resizing setup (described below) with a delay constraint, the leakage benefit over the whole design can decrease as the number of instances which can be replaced by their biased version (slower but less leaky) is reduced.

Test Case	SVT-SGL			SVT-DGL-tech			SVT-DGL-device		
	Delay	Leakage	Dynamic	Delay	Leakage	Dynamic	Delay	Leakage	Dynamic
c5315	1	1	1	1.017	0.779	1.034	1.017	0.789	1.032
c6288	1	1	1	1.038	0.857	1.023	1.022	0.876	1.020
c7552	1	1	1	1.018	0.743	1.044	1.009	0.752	1.018
alu128	1	1	1	1.040	0.741	1.044	1.03	0.753	1.042
Test Case	DVT-SGL			DVT-DGL-tech			DVT-DGL-device		
	Delay	Leakage	Dynamic	Delay	Leakage	Dynamic	Delay	Leakage	Dynamic
c5315	1.034	0.325	0.974	1.017	0.296	1.004	1.034	0.299	1.002
c6288	1.027	0.557	0.984	1.033	0.534	0.993	1.027	0.538	0.991
c7552	1.009	0.202	0.968	1.028	0.171	1.010	1.02	0.171	1.010
alu128	1.020	0.248	0.971	1.040	0.218	1.004	1.03	0.221	1.001

Table 4. Normalized critical-path delay, leakage power, and dynamic power results for various V_{th} and gate-length scenarios. The second gate-length is determined by technology-level or device-level L_{Gate} selection.

For library characterization we prune the TSMC 130nm library to contain only eight commonly used cells: INVX4, NANDX4, BUFX4, ANDX6, NORX4, ORX6, AO22X4 and OA22X4. To get the delay and leakage number, HSPICE[20] simulations are run using TSMC 130nm netlists and STMicroelectronics 130nm spice models*. Dual L_{Gate} optimization is done using a sizer similar to *Duet* proposed in.[3] All cells are sorted in decreasing order of $\Delta leakage \times slack$ where $\Delta leakage$ is the improvement in leakage after a cell is replaced with its less leaky variant, and *slack* is its timing slack after the replacement has been made. We use *Design Compiler v2003.06-SP1 (DC)*[22] for final validation of all timing and power results as well as computation of dynamic power†. Our test cases are simple combinational circuits drawn from the ISCAS85 benchmark suite and Opencores.[23] The four test cases synthesize to 2069 (c5315), 4070 (c6288), 2360 (c7552) and 13279 (alu128) gates. In our results, we report leakage, dynamic power and circuit delay. We do not assume any wire-load models, as a result of which the dynamic power and delay are underestimated.

As described in Subsection 4.1, we choose the L_{Gate} bias at the technology-level and at the device-level. We do not present results for cell-level gate-length biasing as this offers no advantage over device-level biasing in terms of quality, nor over technology-level biasing in terms of ease of implementation. The nominal gate-length for the technology is 130nm. The technology-level biased L_{Gate} is calculated to be 136nm based on an allowable 10% delay penalty. For device-level biasing our methodology biases devices to 136-139nm based on an allowable 10% delay penalty, and there is a 9%-36% leakage power benefit for the eight cells in our library. This strongly supports our hypothesis that small biases in L_{Gate}, intelligently applied, can afford significant leakage savings with virtually no performance impact.

The timing constraint we give to the synthesis tool is very close to the minimum achievable by any combination of threshold voltages and gate-length. Synthesis is performed using low-V_{th}, nominal-L_{Gate} library. For introduction of a V_{th} or L_{Gate} we relax the timing constraint by 2% to give sizing more room to recover power. Results for this delay-constrained sizing for leakage recovery are shown in Table 4. Adding a gate-length to single V_{th} designs can save 14.3% to 25.9 % leakage power with less than 4% delay penalty. For dual V_{th} implementations the leakage benefit is less than 12%. The dynamic power penalty is less than 3.3% in all cases.

Lithography: Manufacturability. We now investigate certain manufacturability implications of our L_{Gate} biasing approach. Since our method relies on biasing of drawn gate-length, it is important to correlate it with actual printed gate-length on the wafer. This is even more important as the bias we introduce in gate-length is of the same order as typical critical dimension (CD) tolerance in manufacturing processes. To validate our multiple gate-length approach in a post-manufacturing setup, we follow a reticle enhancement technology (RET) and process simulation flow for an example cell master. We use the layout of the AND2X6 from TSMC 0.13μm and perform model-based optical proximity correction (OPC) on it using *Calibre v9.3_2.5*.[21] ‡ The printed image of the cell is then calculated using *printimage* simulation in Calibre. We measure the gate-length for every device in the cell, for both biased and unbiased versions. The results for the printed gate dimensions are shown in Table 5. As expected, biased and unbiased gate-lengths track each other well. There are some outliers which may be due to simplicity of the OPC model being used. High correlation between *printed* dimensions of biased and unbiased versions of the cells shows that benefits of biasing estimated using *drawn* dimensions will not be lost during the RET and manufacturing flows.

Another potentially valuable benefit of even slightly larger gate-lengths is possible improved printability. Poly spacing is much larger than poly gate-length, so that the process window (which is constrained by the minimum resolvable dimension) tends to be larger as gate-length increases. For example, the depth of focus for various

*The library contains nominal (NMOS: 0.187V, PMOS: -0.16825V) and low (NMOS: 0.107V, PMOS: -0.08825V) V_{th} devices. The nominal gate-length is 130nm.

†There is a small mismatch between static timing engines of *DC* and *Duet*. We report results from *DC* only.

‡Model-based OPC is performed using annular optical illumination with $\lambda = 248$nm and $NA = 0.7$.

values of exposure latitude with the same illumination system as above for 130nm and 136nm lines is shown in Table 6.*

Device Number	Gate Length (nm)					
	PMOS			NMOS		
	Unbiased	Biased	Diff.	Unbiased	Biased	Diff.
1	125	132	+7	126	132	+6
2	124	126	+2	126	129	+3
3	124	126	+2	126	129	+3
4	121	127	+6	124	130	+6
5	121	127	+6	122	128	+6
6	122	128	+6	122	128	+6
7	125	131	+6	124	131	+7

Table 5. Comparison of printed dimensions of unbiased and biased versions of AND2X6. The unbiased nominal gate-length is 130nm while the biased nominal is 136nm. Note the high correlation between unbiased and biased versions.

DOF (μm)	ELAT for 130nm (%)	ELAT for 136nm (%)
0.09	7.66	7.71
0.33	6.97	7.04
0.5	5.98	6.23
0.67	4.67	5.02
1	2.06	2.71

Table 6. Process window improvement with gate-length biasing. The CD tolerance is kept at 13nm. ELAT=Exposure latitude; DOF=Depth of Focus.

Process Variability. A number of sources of variation can cause fluctuations in gate-length, and hence in performance and leakage. This has been a subject of much discussion in the recent literature (e.g.,[8, 9]). Up to 20X variation in leakage has been reported in practice.[7] For leakage, the reduction in variation post-biasing is likely to be substantial as the larger gate-length is closer to the "flatter" region of the leakage vs. L_{Gate} curve. To validate these intuitions, we study the impact of gate-length variation on leakage and performance both pre- and post-biasing using a simple worst-case approach. We assume the CD variation budget to be $\pm$10nm. The performance and leakage of the test case circuits is measured at the worst-case, nominal and best-case process corners which consider just gate-length variation. This is done for the technology-level L_{Gate} biasing approach as an example. The results are shown in Table 7. For the four test cases, we see a 39% to 54% reduction in leakage power uncertainty caused by line-width variation. Such huge reductions in uncertainty can potentially outweigh benefits of alternative leakage control techniques. We note that the corner case analysis just models the inter-die component of variation, which typically constitutes half of the total CD variation. To assess the impact of both within-die (WID) and die-to-die (DTD) components of variation, we run 2000 Monte-Carlo simulations with $\sigma_{WID} = \sigma_{DTD} = 3.33$nm. The variations are assumed to follow a Gaussian distribution with no correlations. We compare the results for three single V_{th} scenarios: unbiased, technology-level biasing and uniform biasing of the entire design by 6nm. Leakage distributions for the test case *alu*128 are shown in Figure 9.

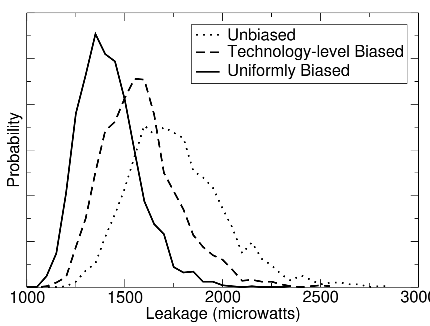

Figure 9. Leakage distributions for unbiased, uniform-biased and technology-level selectively-biased alu128. Note the "left-shift" of the distribution with the introduction of biasing.

Test Case	Circuit Delay (ns)						
	Unbiased			Uniform Bias			% Spread
	BC	WC	NOM	BC	WC	NOM	Reduction
c5315	0.58	0.76	0.66	0.63	0.81	0.72	0
c6288	1.80	2.35	2.05	1.95	2.52	2.26	-3.6
c7552	1.02	1.35	1.18	1.11	1.46	1.29	-5.7
alu128	0.95	1.25	1.10	1.04	1.35	1.20	-3.3
	Leakage (mW)						
	Unbiased			Uniform Bias			% Spread
Test Case	BC	WC	NOM	BC	WC	NOM	Reduction
c5315	0.289	0.137	0.181	0.214	0.122	0.151	+39.4
c6288	0.579	0.276	0.364	0.430	0.247	0.305	+53.5
c7552	0.322	0.156	0.200	0.240	0.140	0.171	+39.7
alu128	1.936	0.930	1.230	1.440	0.833	1.023	+39.6

Table 7. Reduction in performance and leakage power uncertainty with biased gate-length in presence of inter-die variations. The uncertainty spread is specified as a percentage of nominal. The results are given for nominal V_{th}. Uniform bias is 6nm.

5. CONCLUSIONS

We presented three techniques that consider the design and manufacturing information in conjunction to improve mask quality and to make masks cheaper. The first technique, design-aware OPC, proposes a practical means of reducing masks costs and the computational complexity of OPC insertion through formalized performance-driven OPC assignment. In particular we focus on the use of edge placement errors to drive OPC insertion tools and leverage EPEs as the mechanism to direct these tools to correct only to the levels required to meet timing specifications. Our results on several benchmarks ranging from 300 to 12000 cells show up to 24% reductions in MEBES data volume which is frequently used a metric for RET complexity. Furthermore, the runtime of the OPC insertion tool is reduced by up to 34% - this is critical since running OPC tools at the full-chip level is an extremely time-consuming step during the physical verification stage of IC design.

In the second technique, we have presented a novel placement-perturbation technique, called AFCorr, as a feasible and effective approach to achieve assist feature compatibility in physical layouts. AFcorr leads to

*The process simulation was performed using *Prolith v8.0*.[24]

reduced CD variation and enhanced DOF margin. Our results indicate the following. (1) AFCorr placement perturbation can achieve up to 98% reduction in number of cell border poly geometries having forbidden pitch violations. The corresponding reduction in edge placement error is up to 85%. (2) We achieve up to 7.4% increase in number of inserted scattering bars in the benchmark design. (3) The increases of data size, OPC running time and maximum delay overheads of AFCorr are within 3%, 4% and 11% respectively. (4) The runtime of AFCorr placement perturbation is negligible ($\sim$ 3 minutes) compared to the running time of OPC ($\sim$ 2 hours).

The third technique, gate-length biasing, presents a novel methodology that uses selective, *small* L_{Gate} biases to achieve an *easily manufacturable* approach to runtime leakage reduction. For our test cases we have observed the following. (1) The gate-length bias we propose is always less than the pitch of the layout grid. This avoids design rule violations. Moreover, it implies that the biased and unbiased cell layouts are completely pin-compatible and hence layout interchangeable. This allows biasing-based leakage optimization to be possible at any point in design flow unlike sizing-based methods. (2) With simple uniform technology-level biasing applied to the entire design 12%-28% leakage improvement can be achieved at the cost of 8%-12% delay penalty and 3%-6% dynamic power penalty. (3) Using simple sizing techniques, we are able to achieve up to 25% leakage savings with just 4% timing and 5% dynamic power overhead. With dual-L_{Gate} libraries constructed with a smaller $delay_{penalty}$ and multiple versions of frequently used cells, the improvements can be much better. (4) The devices with biased gate-length are *more* manufacturable and have a larger process margin than the nominal devices. Biasing does not require any extra process steps unlike multiple-threshold based leakage optimization methods. (5) L_{Gate} biasing leads to more process-insensitive designs with respect to leakage current. Biased designs have up to 54% less leakage worst-case variability in presence of inter-die variations as compared to nominal gate-length designs. (6) In presence of both inter- and intra-die CD variations, selective L_{Gate} biasing can yield designs less sensitive to variations.

Our on-going and future work involves improving the described techniques in terms of quality, optimization time and acceptability. We will extend the design-aware OPC framework to consider sequential circuits and leakage power constraints. We plan to extend AFCorr to account for interactions between adjacent cell rows and to bias it towards devices and cells that can not tolerate process variations. To increase the benefits of gate-length biasing, we plan to investigate the use of more than two gate-lengths for frequently used and leaky cells. We are also developing novel techniques that utilize design and manufacturing information for better and cheaper masks.

REFERENCES

1. P. Gupta and A. B. Kahng, “Manufacturing-Aware Physical Design”, in Proc. IEEE/ACM ICCAD, November 2003, pp. 681-687.
2. X. Shi, S. Hsu, F. Chen, M. Hsu, R. Socha, and Micea Dusa, “Understanding the Forbidden Pitch Phenomenon and Assist Feature Placement”, Proc. SPIE, Vol. 4689, pp. 985-996, 2002.
3. S. Sirichotiyakul, T. Edwards, C. Oh, R. Panda and D. Blaauw., “Duet: An Accurate Leakage Estimation and Optimization Tool for Dual-V_{th} Circuits”, in *TVLSI*, Vol. 10, No. 2, April 2002, pp. 79-90.
4. L. Wei, Z Chen, M. Johnson, K. Roy and V. De, “Design and Optimization of Low Voltage High Performance Dual Threshold CMOS Circuits”, in Proc. IEEE/ACM DAC, 1998, pp. 489-494.
5. M. Ketkar and S. Saptnekar, “Standby Power Optimization via Transistor Sizing and Dual Threshold Voltage Assignment”, in Proc. IEEE/ACM ICCAD, 2002, pp. 375-378.
6. S. Mukhopadhyay, C. Neau, R.T. Cakici, A. Agarwal, C.H. Kim and K. Roy, “Gate Leakage Reduction for Scaled Devices Using Transistor Stacking”, *TVLSI*, 11(4), 2003, pp. 716-730.
7. S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi and V. De, “Parameter Variations and Impact on Circuits and Microarchitecture”, in Proc. IEEE/ACM DAC, 2003, pp. 338-342.
8. Y. Cao, P. Gupta, A.B. Kahng, D. Sylvester and J. Yang, “Design Sensitivities to Variability: Extrapolations and Assessments in Nanometer VLSI”, *Proc. ASIC/SOC* , 2002, pp. 411-415.
9. R. Rao, A. Srivastava, D. Blaauw and D. Sylvester, “Statistical Estimation of Leakage Current Considering Inter- and Intra-Die Process Variation”, *Proc. ISLPED*, 2003, pp. 84-89.
10. M.L. Rieger, J.P. Mayhew and S. Panchapakesan, “Layout Design Methodologies for Sub-Wavelength Manufacturing”, in Proc. IEEE/ACM DAC, 2001, pp. 85-92.
11. S. Murphy, Dupont Photomask, *SEMATECH: Mask Supply Workshop*, 2001.
12. P. Gupta, A.B. Kahng, P. Sharma and D. Sylvester, ”Selective GateLength Biasing for Cost-Effective Runtime Leakage Control”, in Proc. IEEE/ACM DAC, June 2004, pp. 327-330.
13. R. Nair, C.L. Berman, P.S. Hauge and E.J. Yoffa, “Generation of Performance Constraints for Layout”, in *TCAD*, 8(8), 1989, pp. 860-874.
14. P. Gupta, F.-L. Heng and M. Lavin, “Merits of Cellwise Model-Based OPC”, *Proc. SPIE International Symposium on Microlithography*, 2004, to appear.
15. http://www.ilog.com
16. P. Gupta, A.B. Kahng, D. Sylvester and J. Yang, “A Cost-Driven Lithographic Correction Methodology Based on Off-the-Shelf Sizing Tools”, in Proc. IEEE/ACM DAC, June 2003, pp. 16-21.
17. E. Bozorgzadeh, S. Ghiasi, A. Takahashi and M. Sarrafzadeh, “Optimal Integer Delay Budgeting on Directed Acyclic Graphs”, in Proc. IEEE/ACM DAC, 2003.
18. C. Chen, E. Bozorgzadeh, A. Srivastava and M. Sarrafzadeh, “Budget Management with Applications”, *Algorithmica*, vol 34, No. 3, July 2002, pp. 261-275.
19. International Technology Roadmap for Semiconductors, 2003, http://public.itrs.net
20. http://www.synopsys.com/products/mixedsignal/hspice/hspice.html
21. http://mentor.com/calibre/datasheets/opc/html/
22. http://www.synopsys.com/products/logic/design_compiler.html
23. http://www.opencores.org/projects/
24. http://www.kla-tencor.com