Toward performance-driven reduction of the cost of RET-based lithography control*

Puneet Gupta†, Andrew B. Kahng‡, Dennis Sylvester§ and Jie Yang§

† Electrical and Computer Engineering Dept., UCSD, La Jolla, CA 92093-0114
‡CSE and ECE Departments, UCSD, La Jolla, CA 92093-0114
§ EECS Dept., University of Michigan, Ann Arbor, MI 48109

ABSTRACT

As minimum feature sizes continue to shrink, patterned features have become significantly smaller than the wavelength of light used in optical lithography. As a result, the requirement for dimensional variation control, especially in critical dimension (CD) 3σ, has become more stringent. To meet these requirements, resolution enhancement techniques (RET) such as optical proximity correction (OPC) and phase shift mask (PSM) technology are applied. These approaches result in a substantial increase in mask costs and make the cost of ownership (COO) a key parameter in the comparison of lithography technologies. No concept of function is injected into the mask flow; that is, current OPC techniques are oblivious to the design intent. The entire layout is corrected uniformly with the same effort. We propose a novel *minimum cost of correction* (*MinCorr*) methodology to determine the level of correction for each layout feature such that prescribed parametric yield is attained. We highlight potential solutions to the MinCorr problem and give a simple mapping to traditional performance optimization. We conclude with experimental results showing the RET costs that can be saved while attaining a desired level of parametric yield.

Keywords: OPC, RET, variability, correction

1. INTRODUCTION

Consistent improvements in the resolution of optical lithography techniques have been enabling to the continuation of Moore's Law. However, as minimum feature sizes continue to shrink, the wavelength of light used in modern lithography systems is no longer larger than the minimum line dimensions to be printed, e.g., today's 130nm CMOS processes use 193nm exposure tools. As a result, modern CMOS processes are operating in a subwavelength lithography regime. The International Technology Roadmap for Semiconductors (ITRS)[1] offers projections on the requirements of next generation lithography systems and states that achieving very aggressive microprocessor (MPU) gate lengths and highly controllable gate CDs are critical technology requirements (see Table 1). To meet these requirements, resolution enhancement techniques (RETs) such as optical proximity correction (OPC) and phase shift mask (PSM) technology are applied. Advanced mask manufacturing technologies such as high-precision electron beam machines (EBM), high numerical aperture exposure tools, high-resolution resists,[3] extreme ultraviolet (EUV) and perhaps electron-beam projection lithography (EPL),[5] could play roles in continued scaling of optical lithography. The consequence of all these approaches is a large increase in mask costs, giving total cost of ownership (COO) and non-recurring engineering (NRE) cost greater importance determining project starts and profitability.

1.1. Trends in mask cost

The application of RETs makes mask data preparation (MDP) a serious bottleneck for the semiconductor industry: figure counts explode as dimensions shrink and RETs are used more heavily. Figures can be viewed as features that are printed on a mask in order to realize polygonal shapes seen in the IC layout editor. With the proliferation of OPC in particular, the number of edges, bends, and added features that do not correspond to drawn layout shapes (so-called *sub-resolution assist features*) grows tremendously. Care must be taken to

*Supported in part by the Semiconductor Research Corporation under contract 2001-TJ-913 and by the MARCO Gigascale Silicon Research Center.

Year	2001	2004	2007
Technology Node	130nm	90nm	65nm
MPU gate length	90nm	37nm	25nm
Gate CD control (3 σ) (nm)	5.3	3.0	2.0

Table 1: ITRS requirements for gate dimension variation control are becoming more stringent as technology scales.

ensure that the resulting data is handled correctly. The volume of IC shapes data can easily increase by a factor of 10 or more after RET insertion, resulting in a disproportionate increase in mask-writing and inspection time. Today, the data volume for a single mask layer of a design can approach 100GB, and mask writing times are strongly superlinear in data volume.[18] Figure 2 shows the mask data volumes for various technology nodes.

Mask set cost has been increasing at an accelerated rate for the past several technology nodes; increased application of RETs has been the primary driver of this trend. Compared with the mask set cost in $0.35\mu m$, the cost at the $0.13\mu m$ generation with extensive PSM implemented is four times larger.[6] Considering 500 wafer exposures per mask, the relative cost at the $0.13\mu m$ technology node is about 8.5X that of the $0.35\mu m$ mask set.[7] This brings up an important relationship between design type and lithography costs, namely, that the total cost to produce low-volume parts such as most ASIC designs is dominated by mask costs.[4] Half of all masks produced are used on less than 570 wafers. At such low usages, high RET costs cannot be completely amortized and cost per die becomes very large. Designers and manufacturers must therefore determine how best to apply RETs to standard cell libraries to minimize total cost.

1.2. Design for value

A fundamental observation with regard to the current design-manufacturing interface is that no concept of *function* is injected into the mask flow. Thus mask writers work equally hard in perfecting a dummy fill shape, a piece of the company logo, a gate in a critical path, and a gate in a non-critical path; errors in any of these shapes will trigger rejection of the mask in the inspection tool. The result is unduly low mask throughput and high mask costs. Prohibitively high mask costs motivate the need for *design for value (DFV)* methodologies[2] that attempt to achieve a requisite level of parametric yield ($ per wafer) while minimizing the total cost incurred, both at the design and process levels. For instance, we may have multiple selling points with some pre-specified *value* associated with each selling point. The total design value is then given by $\Sigma v(f) * yield(f)$, for a given value function v of performance measure f, and given parametric yield distribution $yield(f)$. Design for value seeks to find values of design parameters to maximize value, assuming normally distributed process parameters. Such probabilistic optimizations can be incorporated into modern layout design instead of the traditional nominal (or corner case) performance optimization. At the process level, selective OPC is one way of reducing mask complexity and reducing cost while ensuring a desired level of parametric yield. At the design level, issues such as the poor resolution of intermediate gate (and wire) pitches when using sub-resolution assist features and the limited printability of diagonal poly lines for off-axis illumination techniques can be considered during library generation, custom cell layout, and routing. In this paper, we focus on reducing mask costs through process means, specifically the use of various levels of OPC (e.g., aggressive, moderate) to limit mask complexity.

1.3. The cost of correction problem

As mentioned above, current OPC techniques are unaware of the design intent such that the entire layout is corrected uniformly with the same effort. Many features in the layout are not timing critical and a larger degree of process variation may be tolerable on them. At the same time, a certain minimum level of process correction is required to ensure printability of the layout (i.e., functional yield). This suggests that forward annotating OPC with design information permits less total correction to meet the parametric yield requirements. Less aggressive use of OPC directly translates to lowered costs through reduced figure counts, shorter mask write times and higher mask yields.

We define the *selling point* as the circuit delay which gives 99% parametric yield, meaning that 99% of parts would be expected to run at the target frequency or higher. Given the range of allowable corrections for each feature in the layout as well as the cost and parameter variances associated with each correction level, the *minimum cost of correction (MinCorr)* problem is to determine the level of correction for each feature such that the prescribed selling point delay is attained with minimum total correction cost.

The key idea behind our work, as elaborated more below, is that various *levels of correction* are feasible such that functional capabilities are not compromised but the uncertainty in L_{eff} may vary. By using less aggressive OPC insertion guidelines, timing uncertainty in specific gates may rise without negatively impacting parametric yield of the entire circuit. Instead of creating more complex models for model-based OPC, which is already a computationally intensive process, we show the equivalence of the MinCorr problem and traditional gate-sizing problem. *This enables the use of off-the-shelf synthesis tools to solve the MinCorr problem.*

In the remainder of this paper we describe possible solutions to the MinCorr problem. In Section 2 we describe our cost of correction methodology and propose a simple but elegant mapping of the MinCorr problem to conventional performance optimization. Section 3 outlines our experimental setup and we discuss our results in Section 4, then conclude in Section 5.

2. COST OF CORRECTION METHODOLOGY

We propose a *yield closure flow* which is very similar to traditional flows for timing closure. We describe the elements of such a flow in this section.

2.1. Generic flow

A generic approach to the *MinCorr* problem is outlined in Figure 1. We emphasize the striking similarity to conventional timing flows; this mapping is a great advantage which enables easy adoption of such an approach.

The basic elements of the generic flow are as follows.

- *Statistical Static Timing Analyzer (SSTA)* outputs the probability density function (PDF) of the arrival time at all nodes in the circuit, given deterministic arrival times at the primary inputs (PIs). Statistical timing analysis models gate delays as random variables but has traditionally suffered from runtime complexities that are exponential in circuit size, due to the dependencies created by reconvergent paths in the circuit.[10]
- We assume different levels of OPC can be independently applied to any gate in the design. Corresponding to each level of correction, there is an effective channel length L_{eff} variation and an associated cost.
- We assume that variation-aware performance library models are available for each level of correction.

A target selling point delay is assumed to be given. Given the delay mean and standard deviation at every circuit node, the SSTA tool computes the 99% yield point at each primary output. Thus, we can calculate a slack value at all primary outputs. We call this *σ-slack*. Our next step is to *decorrect* or *correct* the gates to minimize the cost while still meeting the σ-slack constraints.

2.2. A linearized approximation to correction

To reduce algorithmic complexity, we assume that the standard deviations of the gate-delays are additive, i.e., we assume a perfect positive correlation between gate-delay variations along any path.[†] If we assume that the path delay distributions remain Gaussian, then we can propagate the 99% ($\mu + 3\sigma$) yield point to the primary output. We explain this further in Section 2.4. More specifically, we assume that

$$\mu_{1+2} + 3\sigma_{1+2} = \mu_1 + 3\sigma_1 + \mu_2 + 3\sigma_2 \quad (1)$$

This also enables us to use STA instead of SSTA to verify the σ-slack correctness of the circuit. We can formulate decorrection as a mathematical programming problem, as follows.

[†]This is not so unreasonable since there is spatial correlation of process variations.[2]

Gate Sizing		MinCorr
Area	$\equiv$	Cost of Correction
Nominal Delay	$\equiv$	Delay $\mu + k\sigma$
Cycle Time	$\equiv$	Selling point delay
Die Area	$\equiv$	Total Cost of OPC

Table 2. Correspondence between the traditional gate-sizing problem and the minimum cost of correction (to achieve a prescribed selling point delay with given yield) problem.

- $d_{ij} = \mu + 3\sigma$ number for gate i corresponding to level of correction j.
- $c_{ij} =$ cost of correction number for gate i corresponding to level of correction j.
- $x_{ij} = 1$ if gate i is corrected to level j.
- $wd_i =$ worst case $\mu + 3\sigma$ at input of gate i. This is calculated using STA.
- $U =$ the $\mu + 3\sigma$ upper-bound at the POs.

$$\begin{aligned} Minimize\ \ & \Sigma_{i,j} x_{ij} c_{ij} \\ & \Sigma_j x_{ij} = 1 \\ \Sigma_j x_{ij} d_{ij} + wd_i < wd_k\ & \forall k \in fanout(i) \\ & wd_k = U\ \forall\ k \in PO \\ & x_{ij} \in 1, 2, 3 \end{aligned} \tag{2}$$

The above integer program requires running of the STA tool incrementally to update wd_i every time any x_{ij} is updated. Note that the results we obtain from solving the program are strictly pessimistic if the circuit consists of perfectly correlated paths. This is because gates would always be somewhat less than perfectly correlated, in which case the standard deviation of the sum would be less than the sum of standard deviations. However, in practice, a circuit consists of many partially correlated or independent paths. In this case, calculating the delay distribution at any primary output requires computing the maximum of delay distributions of all the paths fanning in to the PO. The resultant Max distribution may not remain Gaussian and is likely to have a larger mean and smaller variance than the parent distributions. To account for this, a generic flow would run SSTA on the decorrected circuit and compute σ-slacks at all POs again. We would then fix the negative slack at any PO by correcting the large-fanout nodes at the last few levels (close to the leaves) in the fanin cone of the PO. We distribute the positive slack among the small-fanout nodes in the first few levels of the fanin cone of the PO. We do this iteratively until we obtain σ-slacks at all POs sufficiently close to zero.

2.3. Parallels to traditional timing optimization

Parallels can be drawn between the MinCorr problem and the well studied gate-sizing and delay-budgeting problems. The allowed "sizes" in the MinCorr problem correspond to the levels of correction. For each instance in the design there is a cost and delay σ associated with every level of correction. For instance, note the similarity between the integer program (3) and sizing[12] or budgeting.[15] The mapping between gate-sizing and MinCorr is depicted in Table 2.

This mapping is correct given the assumption of additivity as in Equation (1). We will point out in the next section how we can retain pessimism in our results without losing this desirable property. Here we *must* emphasize that Equation (1) need not be assumed if the correction (sizing) tool is driven by SSTA rather than

STA. Given Table 2, we can construct yield libraries in a similar fashion as timing libraries. This enables us to use the yield (timing) libraries with a commercial synthesis tool such as *Synopsys Design Compiler (DC)*[17] to recorrect (resize) the design to meet the yield (delay) target with the minimum cost (area). Use of a commercial tool enables us to make many interesting optimizations within practical runtimes. Examples include minimizing the cost of correction given the selling point delay, and minimizing the selling point delay given an upper bound on the cost of OPC.

2.4. Extreme order statistics and pessimism

As previously mentioned in Section 2.2, the statistical circuit delay distribution is the distribution of the maximum of *all* path delays. Such a distribution is hard to compute and may no longer remain Gaussian even if all the gate delay distributions are Gaussian. If we assume that after recorrection we can get equal σ-slacks at all the primary outputs, then we can approximate the circuit delay distribution by the maximum of all output delay distributions. Then, the mean of the circuit delay distribution is bounded by[11]

$$\mu_{circuit} \leq \mu_{output} + \sigma_{output} \frac{n-1}{\sqrt{2n-1}}. \tag{3}$$

Moreover, the variance of the circuit delay is bounded by the variance of the output delay distributions:[11]

$$\sigma_{circuit} \leq \sigma_{output}. \tag{4}$$

This gives a way to generate design and yield target specific yield libraries. For example, for a 32 output design $\mu_{circuit} + 2\sigma_{circuit}$ of the circuit delay is bounded by $\mu + 6\sigma$ of the output delay distribution. $\mu_{output} + 2\sigma_{output}$ signifies 95% parametric yield when circuit delay is Gaussian. In other cases, yield significance can be pessimistically estimated by Chebyshev's Inequality.[‡]

3. EXPERIMENTAL TESTBED

In this section we describe our experimental yield closure flow. The basic elements of the flow are as follows.

1. A yield-aware library that captures:
   (a) Delay mean and variance for each level of correction for each library master.
   (b) Relative cost of OPC at each level of correction for each master.
2. A standard off-the-shelf logic synthesis tool.

3.1. Yield-aware library characterization

Here we describe our method to generate the yield-aware libraries that the synthesis tool will use in addressing the MinCorr problem.

3.1.1. Generating timing information

We begin by reducing a standard TSMC .lib file such that it retains only basic cells such as BUF, INV, NAND (2,3,4 inputs), and NOR (2 and 3 inputs). At this point, we generate new library files corresponding to each level of OPC correction. The new .lib files reflect the different levels of variability among the whole circuit and approximate the $\mu + 3\sigma$ delay point as the cell delay. There are two ways to generate these new tables: (1) using Monte Carlo (MC) simulation or (2) a deterministic corner-based approximation. MC simulations assume that every parameter (oxide thickness (T_{ox}), channel doping (N_{ch}), channel length, etc.) varies simultaneously in a normally distributed fashion, and consequently provides the best accuracy at the cost of large runtime. Corner-based simulations use a single value for each parameter to find a single worst-case delay. Setting all

[‡]Chebyshev's Inequality states that for a random variable X, $P(|X - \mu| \geq \epsilon) \leq \frac{\sigma^2}{\epsilon^2}$.

Type of OPC	L_{eff} (nm)	3σ of L_{eff}	Figure count (relative)	(μ, σ) for NAND2X1 (ps)
Aggressive	130	5%	5X	(60.7, 7.03)
Medium	130	6.5%	4X	(60.7, 7.47)
None	130	10%	1X	(60.7, 8.79)

Table 3: Cost vs. complexity for various levels of OPC

input parameters to their worst-case value will result in highly pessimistic results since this case is unlikely to occur due to independence of the physical parameters involved. To provide a more efficient means of generating worst-case delays for a wide range of cells and capacitive loading conditions (to build the delay tables in the .lib file), we examined two cells, an inverter and a 2-input NAND in both $0.13\mu m$ and $0.18\mu m$ TSMC processes for different sizes and load capacitances. After performing MC simulations for these gates, we found the 3σ delay points corresponded nearly exactly with a deterministic corner case analysis in which three important variation sources L, T_{ox}, and V_{th0}, have their values at $\mu + 2\sigma$. The difference between the 2σ corner approach and the valid 3σ delay from full MC ranged from 0.3% to 3.5%. Based on this approach, we generated three new .lib files whose delay tables reflect the 3σ (worst-case) values at different levels of correction. For simplicity of experiments, we characterized delays for the new library files for only a single input transition time.

3.1.2. Mask cost model

The major contributors to mask cost are:[5]

1. low mask yield (due to OPC and PSM as well as stringent CD requirements);
2. increased data preparation time;
3. equipment cost; and
4. low equipment throughput.

Figure 3 shows the major drivers of mask cost.

The main drivers for increasing mask costs and turnaround time (TAT) include the increasing application of RETs and their higher write times. Variable-shaped electron beam mask writing combined with vector scanning (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) is a widely used technique for high-speed mask writing. In this method the input GDSII layout data is converted into the mask writer format through a process known as figure *fracturing*. The fractured layout data is in the form of rectangles or trapezoids of different dimensions and topologies. With OPC applied during mask data preparation, the post-processor adds enhancement features using either rule-based or model-based OPC. Rule-based OPC adds enhancement features to all rectangles in a consistent manner in order to meet a given specification; that is, a set of well-defined rules have been generated such that when the fracturing tool sees a specific geometry it will always insert the same correction shapes. Model-based OPC views each feature individually and selects enhancements to be made based on the environment of the original feature as well as its geometry. In either case, the number of line edges is increased by 4-8X over a non-OPC layout, driving data volume up.[9] Mask writers are hence slowed by the software for e-beam data fracturing and transfer along with the extremely large file sizes (write complexities) involved.

In our study, figure count is set by the current methodology for model-based OPC and is given as a multiple of the figure count found in a non-OPC layout. Based on the assumption that vector scanning is used, this should yield a reasonable prediction of the increase in write time/cost. We focus solely on critical dimensions of the polysilicon layer although OPC is applied to other levels of the design, in particular metallization. Although

the application of OPC features varies along the gate width and there will be some variation of the channel length (L_{eff}) along the width axis, we represent the device by a single L_{eff} value.

Correction cost information is included in the newly generated .lib files using the cell area attribute. Our metric for cost is given by relative figure count multiplied by the number of transistors in each cell. We use this weighted cost function to capture (1) the cost differences across the three libraries with different levels of correction applied, and (2) the relative difference in cost to correct cells of different sizes/complexities. We do not simply use the initial area as a weighting factor as we want to emphasize the correction of actual devices rather than field regions which may dominate the cell area. (Another option would have been to weight the figure count by the total transistor perimeter in a cell.) The figure count was found to be consistent across cell types. This is expected from a standard cell library as there is limited diversity in the arrangements of devices within the cell. By contrast, in full-custom circuits there may be a wider range of polysilicon gate configurations (bent gates, varied pitches, tapered stack sizes, etc.). An example of the levels of correction we consider is shown in Figure 4. An interesting note here is that existing OPC and fracturing software corrects the polysilicon layer globally within a cell and does not adequately differentiate between actual transistor gate configurations and non-critical poly sections such as intra-cell interconnections. By adding large numbers of extra vertices in field (rather than transistor active) areas, the CD variability is not strongly impacted even as figure complexity and mask cost grow. This is a good example of the lack of functional awareness of the current design-manufacturing interface. The result is that the costs associated with increasing the level of correction are not always translated to major improvements in CD controllability.

The variation and cost corresponding to each level of correction are listed in Table 3.[16] The channel length variations are given relative to the drawn channel length (130nm in our process) and are based on simulation of the polysilicon layer for various types of cells. The figure count is based on the fracturing of the polygons in a cell using industry-standard photomask manufacturing data preparation software. We use the variation number as 3σ in our analysis. The variation numbers reflect intra-die variability sources only and as such are somewhat smaller than the total expected fluctuations in channel length which may be in the range of $10-15\%$ of nominal.

3.2. Synthesis tool

The most elegant aspect of our flow is that we enable the use of off-the-shelf logic synthesis tools to solve the MinCorr problem. We use *Synopsys DC* as our synthesis tool. We input the yield library in which identical cells in the original timing library show up as three "sized" versions with same cell function but different "areas" and "timing". We then use DC to perform gate-resizing on the synthesized netlist with a selling point delay constraint given as the maximum circuit delay constraint. This has the advantage of being able to use well-tested sizing methods built into the tool. The use of a synthesis tool also enables us to try out interesting variants of the MinCorr problems such as cost-constrained selling point delay minimization.

4. RESULTS AND DISCUSSION

As a proof of concept, we test our techniques on four small combinational designs.

- *alu128* is an industry testcase which synthesizes to 9410 gates.
- *c7552* is the largest of the ISCAS85 testcases and synthesizes to 2075 gates.
- *c6288* is 2537-gate ISCAS85 benchmark.
- *c5315* is 1601-gate ISCAS85 benchmark.

The results for a sweep of selling point delay on the four designs are shown in Table 4. The results show that there is little (about 4%) variation ini either overall cycle time or the selling point from max-corrected to min-corrected versions of the design. This is due to the small degree of delay change for these levels of correction as exemplified in Table 3 for a unit-sized two-input NAND gate. In addition, we do not consider the impact of

Testcase	Normalized Cost	Normalized delay
alu128	**5.0000 (Aggressive OPC)**	**0.9644**
	4.0000 (Medium OPC)	**0.9739**
	1.0000 (No OPC)	**1.0000**
	1.0657	0.9644
	1.0267	0.9739
	1.0180	0.9857
	1.0119	0.9976
c7552	**5.0000 (Aggressive OPC)**	**0.9606**
	4.0000 (Medium OPC)	**0.9724**
	1.0000 (No OPC)	**1.0000**
	1.3428	0.9606
	1.2833	0.9606
	1.2549	0.9606
	1.1951	0.9606
	1.1517	0.9606
	1.4558	0.9646
	1.1256	0.9843
c6288	**5.0000 (Aggressive OPC)**	**0.9660**
	4.0000 (Medium OPC)	**0.9755**
	1.0000 (No OPC)	**1.0000**
	3.2673	0.9660
	2.5065	0.9660
	2.0390	0.9660
	1.7813	0.9717
	1.4305	0.9811
	1.0996	0.9906
c5315	**5.0000 (Aggressive OPC)**	**0.9657**
	4.0000 (Medium OPC)	**0.9755**
	1.0000 (No OPC)	**1.0000**
	1.2253	0.9657
	1.1699	0.9657
	1.0907	0.9755
	1.0627	0.9804
	1.0294	0.9902

Table 4: Cost of correction vs. selling point delay.

input capacitance changes with L_{eff}; this will emphasize the change in delay as applying maximum correction to a gate will simultaneously improve the delay of that stage as well as the delay of the prior stage. So, while the reported change in selling point delay is not large, we expect such changes to be larger when second-order effects are considered.

5. CONCLUSIONS AND ONGOING WORK

In this work we have shown:

- It is possible to reduce the total cost of OPC while still meeting yield and cycle time targets by making OPC design-aware.
- Conventional gate-sizing methods can be easily modified to solve the MinCorr cost of correction problem. We have given a recipe to use an industry standard synthesis tool to perform the job.

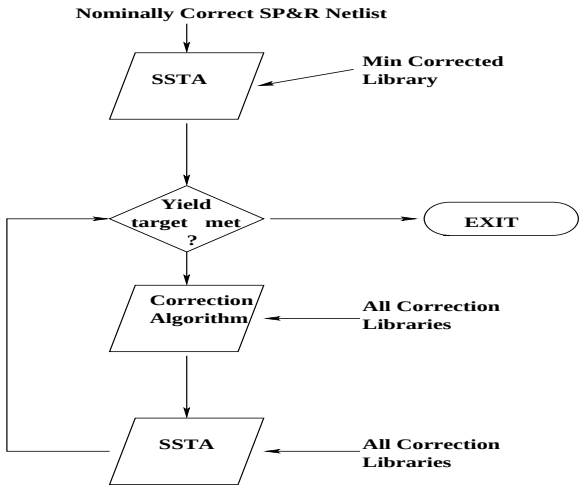

Figure 1: The design flow for yield closure.

From our results, we see up to a 5X cost improvement at just a 4% selling point delay penalty going from no OPC to aggressive OPC. This small difference suggests that OPC might be more of a manufacturability issue rather than a performance or yield issue. With sizing-based optimizations and selective OPC, we can save up to 79% RET cost compared to aggressive OPC, without increasing the selling point delay. Our results indicate that design performance oblivious RET techniques suffer from large cost overheads.

Our ongoing work is in the following directions.

1. *Statistical Static Timing Analysis (SSTA) based correction.* Using SSTA[10] in the core correction flow may not be feasible due to runtime and scalability issues. Since our linear approximation of correction may not remain pessimistic in all cases, we intend to use SSTA to validate the sizing results. We can then iterate over the correction flow as in Figure 1. Another option is to heuristically "fix" the sizing solution. Generally speaking, good candidates for correction are the gates that fanout to a large number of critical paths. Good candidates for decorrection are the gates that fanout to a small number of critical paths. Various existing approaches[12,13] may be applicable here.

2. *Alternative approaches to correction.* We are exploring other potential solutions to the MinCorr problem such as the following.

   (a) *Transistor sizing* instead of gate sizing can offer a finer granularity of MinCorr optimization at the cost of runtime. Correcting different transistors to different levels can either be incorporated by generating a more accurate yield library (i.e., all pin-to-pin delays need to be correctly estimated) for gate-sizing or by constructing complex delay models and doing explicit transistor sizing as in TILOS.[13]

   (b) *Cost based delay budgeting* methods[15] are also applicable. Though simple and fast delay/slack budgeting methods such as ZSA[14] may be applied, they suffer from lack of cost awareness.

3. *More accurate correction.* Input slew awareness in the yield libraries and including interconnect in the analysis are immediate goals of our ongoing work. We also intend to consider dependence of gate input capacitance on L_{eff} variation in the yield libraries.

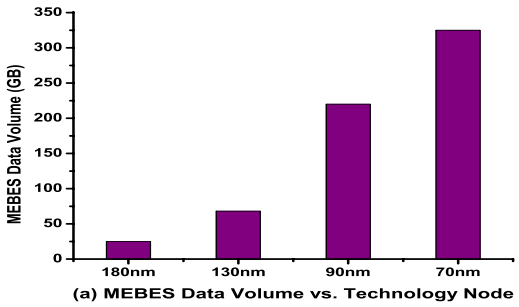

Figure 2: Mask data volume as the technology scales.

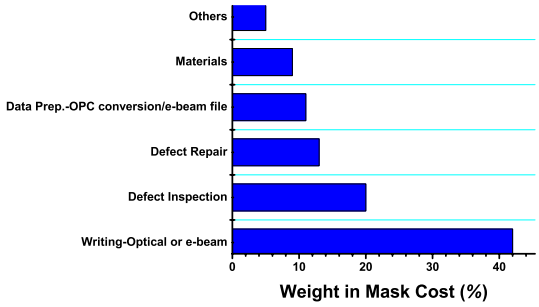

Figure 3: Relative contributions of various components of mask cost.

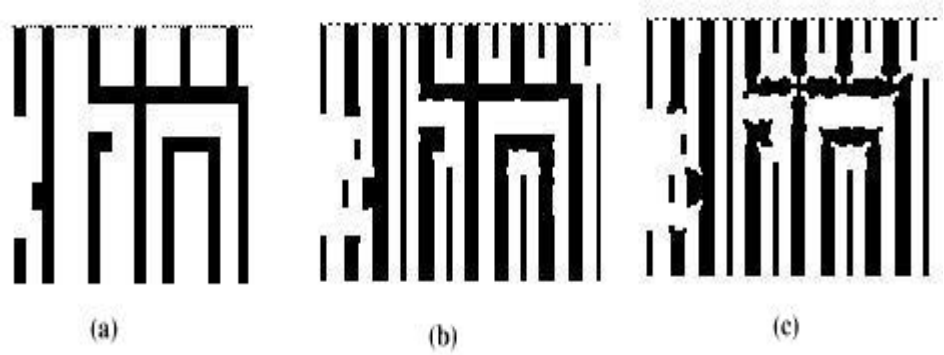

Figure 4: An example of three levels of OPC.[19] (a) No OPC, (b) Medium OPC, (c) Aggressive OPC.

REFERENCES

1. International Technology Roadmap for Semiconductors, December 2001 http://public.itrs.net/
2. Y. Cao, P. Gupta, A.B. Kahng, D. Sylvester and J. Yang, "Design Sensitivities to Variability: Extrapolations and Assessments in Nanometer VLSI", *IEEE International ASIC/SOC Conference*, 2002, pp. 411-415.
3. "The Outlook for Semiconductor Processes and Manufacturing Technologies in the 0.1-μm Age", http://www.cyberfab.net/events/013mmts /links013.html.
4. M.L. Rieger, J.P. Mayhew, and S. Panchapakesan, "Layout Design Methodologies for Sub-Wavelength Manufacturing", *Proceedings of Design Automation Conference*, 2001, pp. 85-92.
5. SEMATECH: Mask Supply Workshop, 2001.
6. Chiang Yang, "Challenges of Mask Cost and Cycle Time", *SEMATECH: Mask Supply Workshop, Intel*, 2001.
7. W. Carpenter, "International SEMATECH: A Focus on the Photomask Industry", http://www.kla-tencor.com/company_info/magazine/autumn00/ Inter_SEMATECH_photomaskindustry_AutumnMag00-3.pdf.
8. B. Bruggeman, et al, "Microlithography Cost Analysis", *Interface Symposium*, 1999.
9. S. Murphy, Dupont Photomask, *SEMATECH: Mask Supply Workshop*, 2001.
10. A. Agarwal, D. Blaauw, V. Zolotov, and S. Vrudhula, "Statistical Timing Analysis using Bounds and Selective Enumeration", *ACM/IEEE International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, 2002, pp. 29-36.
11. Robert R. Kinnison, *Applied Extreme Value Statistics*, Battelle Press, 1985.
12. W. Chuang, S.S. Sapatnekar and I.N. Hajj, "Delay and Area Optimization for Discrete Gate Sizes under Double-Sided Timing Constraints", *Proc. IEEE Custom Integrated Circuits Conference*, 1993, pp. 9.4.1-9.4.4.
13. A.E. Dunlop, J.P. Fishburn, D.D. Hill and D.D. Shugard, "Experiments using Automatic Physical Design Techniques for Optimizing Circuit Performance", *Proc. IEEE International Symposium on Circuits and Systems*, (2), 1990, pp. 847-851.
14. R. Nair, C.L. Berman, P.S. Hauge and E.J. Yoffa, "Generation of Performance Constraints for Layout", *IEEE Transactions on Computer Aided Design*, 8(8), 1989, pp. 860-874.
15. M. Sarrafzadeh, D.A. Knol and G.E. Tellez, "A Delay Budgeting Algorithm Ensuring Maximum Flexibility in Placement", *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, 16(11), 1997, pp. 1332-1341.
16. D. Pramanik, Numerical Technologies Inc., personal communication, November 2002.
17. Synopsys Design Compiler, http://www.synopsys.com/products/logic/logic.html
18. P. Buck, *ISMT Mask-EDA Workshop*, Dupont Photomasks, 2001.
19. K. Wampler, ASML MaskTools, personal communication, March 2003.