# A Cost-Driven Lithographic Correction Methodology Based on Off-the-Shelf Sizing Tools *

P. Gupta‡, A.B. Kahng‡, D. Sylvester* and J. Yang*
‡ECE Department, University of California at San Diego
*EECS Department, University of Michigan at Ann Arbor
(puneet,abk@ucsd.edu), (dennis,jiey@eecs.umich.edu)

## 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) $3\sigma$, 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, and 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 with minimum total RET cost. We highlight potential solutions to the MinCorr problem and give a simple mapping to traditional performance optimization. We conclude with experimental results showing that substantial RET costs may be saved while maintaining a given desired level of parametric yield.

**Categories and Subject Descriptors:** B.7.2 [Hardware]: IC; J.6 [Computer Applications]: CAD; F.2.2[Analysis of Algorithms]: Problem Complexity

**General Terms:** Algorithms, Design, Reliability, Theory

**Keywords:** VLSI Manufacturability, OPC, RET, Lithography, Yield

## 1. INTRODUCTION

Consistent improvements in the resolution of optical lithography techniques have been a key enabler for 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 several times 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 sub-wavelength 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 CD control are two critical issues (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, high numerical aperture exposure equipment, high-resolution resists [3], and extreme ultraviolet and possibly electron-beam projection lithography [5], could also play roles in continued lithography scaling. The result of each of these approaches is a large increase in mask costs, and indeed cost of ownership (COO) has become a key consideration in adoption of various lithography technologies.

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

**Table 1: ITRS requirement of gate dimension variation control is becoming more stringent as the technology scales.**

### 1.1 Trends in Mask Cost

The increasing 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. Compared with the mask set cost in 0.35um, the cost at the 0.13um generation with extensive PSM implemented is four times larger [6]. Figure counts, corresponding to polygons as seen in the IC layout editor grow tremendously due to sub-resolution assist features and other proximity corrections. Increases in the fractured layout data volume lead to disproportionate increases in mask-writing and inspection time. According to the 2001 ITRS [1], the maximum single-layer MEBES file size increases from 64GB in 130nm to 216GB in 90nm.

Another observation concerns the relationship between design type and lithography costs, namely, that the total cost to produce low-volume parts is dominated by mask costs [4]. Half of all masks produced are used on less than 570 wafers (this translates roughly to production volumes of $\leq$ 100,000 parts). At such low usages, the high added costs of RETs cannot be completely amortized and the corresponding cost per die becomes very large. Thus, designers and manufacturers are jointly faced with determining how best to apply RETs to standard cell libraries to minimize mask cost.

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

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
*DAC 2003,* June 2–6, 2003, Anaheim, California, USA.
Copyright 2003 ACM 1-58113-688-9/03/0006 ...$5.00.

## 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. Mask writers today 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 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, or the limited printability of diagonal poly lines with off-axis illumination, can be considered during library generation, custom cell layout, and routing. In this paper, we focus on reducing mask costs through process means, specifically, exploitation of multiple 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 design intent, so 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 for 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 the design's functional information will permit 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, elaborated below, is that various *levels of correction* are feasible such that functionality is not compromised but the uncertainty in $L_{eff}$ may vary. By using less aggressive OPC insertion, timing uncertainty of 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 to the 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. We 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. In this section, we describe the elements of such a flow.

## 2.1 Generic Flow

A generic approach to the *MinCorr* problem is outlined in Figure 3. We emphasize the striking similarity to conventional timing optimization flows; this mapping is a great advantage in that it enables easy adoption of our approach. The basic elements of the generic flow are as follows.

- *Statistical Static Timing Analysis (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 exponential run time complexity with circuit size, due to the dependencies created by re-converging paths in the circuit [10].
- We assume that 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 $\sigma$*-slack*. Our next step is to *decorrect* or *correct* the gates to minimize the cost while still meeting the $\sigma$-slack constraints.

## 2.2 A Linearized Approximation to Correction

To reduce the 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.[1] 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 further explain this further in Section 2.4. More specifically, we assume that

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

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

- $d_{ij}=\mu+3\sigma$ number for gate $i$ corresponding to level of correction $j$.
- $c_{ij}$ = cost of correction number for gate $i$ corresponding to level of correction $j$.

[1] This is not unreasonable for die-to-die variation [2].

- $x_{ij} = 1$ if gate $i$ is corrected to level $j$.
- $wd_i$ = worst case $\mu + 3\sigma$ delay at input of gate $i$, calculated using STA.
- $U = \mu + 3\sigma$ delay upper bound at the POs.

$$\begin{aligned} Minimize\ \Sigma_{i,j} x_{ij} c_{ij} \quad (2) \\ \Sigma_j x_{ij} = 1 \\ \Sigma_j x_{ij} d_{ij} + wd_i < wd_k\ \forall k \in fanout(i) \\ wd_k = U\ \forall\ k \in PO \\ x_{ij} \in \{0,1\} \end{aligned}$$

The above integer program requires running 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 contains many partially correlated or independent paths. In this case, calculating the delay distribution at any primary output requires computing the maximum of the delay distributions of all the paths fanning out to the PO. The resultant Max distribution may not remain Gaussian and is likely to have larger mean and smaller variance than the parent distributions. To account for this, our generic flow again runs SSTA on the decorrected circuit and compute $\sigma$-slacks at all POs. We 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 $\sigma$-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 key analogy is that allowed "sizes" in the MinCorr problem correspond to the allowed levels of correction. For each instance in the design, there is a cost and delay $\sigma$ associated with every level of correction. This correspondence immediately highlights a strong similarity between the integer program (3) and the sizing approach outlined in [12] or the budgeting method given in [15]. Our mapping between gate sizing and MinCorr is depicted in Table 2, and is correct to the extent of assuming 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 in 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 obtain equal $\sigma$-slacks at all the primary outputs, then we can approximate the circuit delay distribution by the maximum of all output delay distributions. The mean of the circuit delay distribution is then bounded by [11]

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

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

$$\mu_{circuit} \leq \mu_{output} + \sigma_{output} \frac{n-1}{\sqrt{2n-1}} \quad (3)$$

Moreover, the variance of the circuit delay is bounded by the variance of the output delay distributions [11], i.e.,

$$\sigma_{circuit} \leq \sigma_{output} \quad (4)$$

This gives a way to generate yield libraries that are specific to a given design and yield target. 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 the Chebyshev's Inequality.[2]

## 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, and
   (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

We begin by pruning a standard TSMC .lib file so that it retains only basic cells such as BUF, INV, NAND (2,3,4 inputs), and NOR (2 and 3 inputs). We generate two sets of new library files: (1) 3 files corresponding to the worst case ($\mu + 3\sigma$ delay point) of each level of OPC, and (2) 500 .lib files for statistical static timing analysis (SSTA), assuming delay has a Gaussian distribution with $\sigma$ values corresponding to each RET level. There are two ways to generate new worst case timing model: using Monte-Carlo (MC) simulation, or using 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 provide 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 input parameters to their worst-case value will result in highly pessimistic

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

| Type of OPC | $L_{eff}$ (nm) | $3\sigma$ of $L_{eff}$ | Figure count (relative) | $(\mu, 3\sigma)$ for NAND2X2 (ps) |
|---|---|---|---|---|
| Aggressive | 130 | 5% | 5X | (64.82, 2.14) |
| Medium | 130 | 6.5% | 4X | (64.82, 2.80) |
| None | 130 | 10% | 1X | (64.82, 4.33) |

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

results since this case is unlikely to occur due to independence of the physical parameters involved. After verifying that error between the MC simulation and the deterministic corner case analysis is within 3.5%, we choose the corner based analysis, in which three important variation sources L, $T_{ox}$, and $V_{th0}$ have their values set at $\mu + 2\sigma$, to generate the worst case library files. Input capacitance variation is considered in both (1) and (2) under the assumption that they vary randomly within each level of RET but have perfect correlation with delays, given the shared dependence on $L_{eff}$.[3] For simplicity of experiments, we characterized delays for the new library files for only a single input transition time.

### *3.1.1 Mask Cost Model*

According to [5], the major contributors to mask cost are:

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 1 shows the major drivers of mask cost and turnaround time (TAT), which include the increasing application of RETs and their higher write times. Variable-shaped electron beam mask writing combined with vector scanning[4] is a widely used technique for high-speed mask writing. In the standard mask data preparation 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 mask data preparation, 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, as well as by the extremely large file sizes involved.

In our study, figure count is set by the current methodology for model-based OPC[5] and is given as a multiple of the figure count found in a non-OPC layout.[6] 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 of correcting cells with 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 is to weight the figure count by the total transistor perimeter in a cell. We found figure count to be consistent across cell types, as would be expected from a standard-cell library that has limited diversity in the arrangements of devices within the cell. This is in contrast to full-custom circuits in which 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 [19].

The variation and cost corresponding to each level of correction is listed in Table 3 [16]. The channel length variations[7] 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.

## 3.2 Synthesis Tool

The most elegant part of our flow is that we enable the use of off-the-shelf synthesis tools to solve the MinCorr problem. We use *Synopsys DC* as our synthesis tool. We input a 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 8064 gates.
- *c7552* is the largest of the ISCAS85 testcases and synthesizes to 2081 gates.
- *c6288* is a 2769 gate ISCAS85 benchmark.

[3]Note that this yield library characterization approach models die-to-die variation only and ignores within-die variation.

[4]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.

[5]The RET insertion 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 invariably insert the required feature. 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.

[6]An interesting note here is that existing fracturing software, which is used to add OPC features to a basic layout, corrects the polysilicon layer globally within a cell and does not 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 impacted while the cost grows. 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.

[7]The variation listed in Table 3 is calculated *within-die* but we expect die-to-die and within-die variation to be nearly equal in magnitude [20].

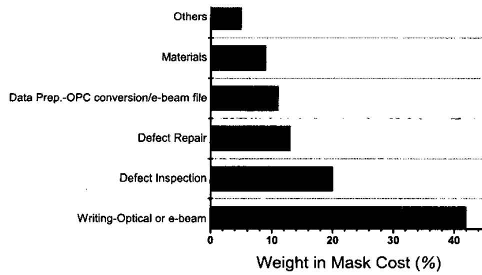


**Figure 1: Relative contributions of various components of mask cost.**

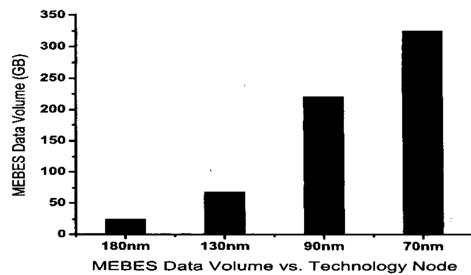


**Figure 2: Mask data volume as the technology scales.**

| Testcase | Normalized Cost | Normalized delay |
|---|---|---|
| alu128 | **5.0000 (Aggressive OPC)** | **0.9479** |
| | **4.0000 (Medium OPC)** | **0.9623** |
| | **1.0000 (No OPC)** | **1.0000** |
| | 1.6038 | 0.9479 |
| | 1.5172 | 0.9515 |
| | 1.4781 | 0.9605 |
| | 1.4448 | 0.9694 |
| | 1.3751 | 0.9857 |
| c7552 | **5.0000 (Aggressive OPC)** | **0.9432** |
| | **4.0000 (Medium OPC)** | **0.9621** |
| | **1.0000 (No OPC)** | **1.0000** |
| | 1.5627 | 0.9432 |
| | 1.4639 | 0.9432 |
| | 1.3747 | 0.9659 |
| | 1.2079 | 0.9848 |
| c6288 | **5.0000 (Aggressive OPC)** | **0.9480** |
| | **4.0000 (Medium OPC)** | **0.9642** |
| | **1.0000 (No OPC)** | **1.0000** |
| | 4.1530 | 0.9480 |
| | 3.3704 | 0.9588 |
| | 2.1789 | 0.9767 |
| | 1.8020 | 0.9856 |
| c5315 | **5.0000 (Aggressive OPC)** | **0.9471** |
| | **4.0000 (Medium OPC)** | **0.9615** |
| | **1.0000 (No OPC)** | **1.0000** |
| | 1.3127 | 0.9480 |
| | 1.3015 | 0.9567 |
| | 1.0713 | 0.9808 |

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

- *c5315* is a 1601 gate ISCAS85 benchmark.

The results for a sweep of selling point delay on these designs are shown in Table 4. We observed little (about 5%) variation in overall cycle time and the selling point from max-corrected to min-corrrected versions of the design. This is due to the small degree of delay change across these levels of correction, as exemplified in Table 3 for a two-input NAND gate.

To verify the linearity assumptions inherent in our approach, we compare the yield calculated from our approach with the result of 500 random runs of STA. To emulate a SSTA tool, we generate 500 random versions of .lib timing libraries wherein the cell delays are drawn from a Gaussian distribution as in Table 3. This models die-to-die variation only. We then compute the circuit delay using *Synopsys PrimeTime* (500 times). Table 5 compares $\mu + 3\sigma$ values for the circuit delay distributions calculated using Monte-Carlo PrimeTime and our approach. It is clear that our approach retains pessimism and fidelity.

| Testcase | OPC Level | MC PrimeTime SSTA(ps) | Our Approach (ps) |
|---|---|---|---|
| alu128 | Aggressive | 5.083 | 5.28 |
| | Medium | 5.116 | 5.36 |
| | None | 5.181 | 5.57 |
| c7552 | Aggressive | 2.414 | 2.49 |
| | Medium | 2.436 | 2.54 |
| | None | 2.477 | 2.64 |
| c6288 | Aggressive | 5.113 | 5.29 |
| | Medium | 5.150 | 5.38 |
| | None | 5.221 | 5.58 |
| c5315 | Aggressive | 1.907 | 1.97 |
| | Medium | 1.923 | 2.00 |
| | None | 1.947 | 2.08 |

**Table 5: $\mu + 3\sigma$ values of circuit delay distribution.**

# 5. CONCLUSIONS AND ONGOING WORK

In this work, we have shown the following.

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

From our results, we see up to a 5X cost improvement at just a 5% selling point delay penalty, going from no OPC to aggressive OPC. This small difference suggests that OPC might be more of a manufacturability issue than a performance or yield issue. With sizing-based optimizations and selective OPC, we can save up to 69% of the 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 based correction*: Using SSTA such as [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 3. The other 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 approaches such as [13, 12] appear useful here.

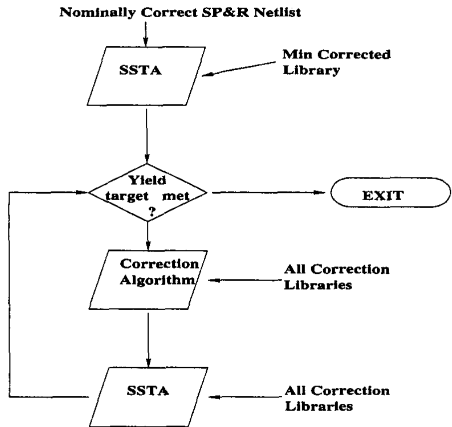


**Figure 3: The design flow for yield closure.**

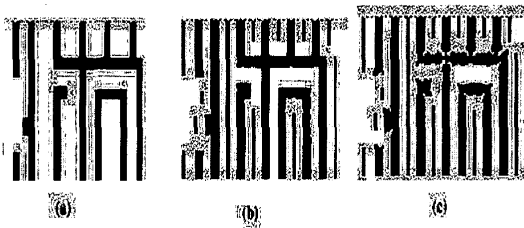


**Figure 4: An example of the three levels of OPC: (a) No OPC; (b) Medium OPC; (c) Aggressive OPC.**

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 such as [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 inclusion of interconnect and interconnect variability in the analysis, are immediate goals of our ongoing work.

4. *Accounting for within-die variation*: We currently ignore within-die variation in our analysis. Systematic within-die variation does not cause variation in the circuit delay distribution, while die-to-die and random within-die variations impact the yield. Correct modeling of all variations is part of our ongoing work.

# 6. 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-$\mu$m 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.

[20] K. Bowman, Intel Corp., personal communication, April 2003.