Performance-driven optical proximity correction for mask cost reduction

Puneet Gupta
Blaze DFM, Incorporated
1275 Orleans Drive
Sunnyvale, California 94089-1138

Andrew B. Kahng
University of California, San Diego
Department of Computer Science and Engineering
and
Department of Electrical and Computer Engineering
9300 Gilman Drive
San Diego, California 92093-0114

Dennis Sylvester
Jie Yang
University of Michigan
Department of Electrical Engineering and Computer Science
1301 Beal Avenue
Ann Arbor, Michigan 48109-2122
E-mail: maggie.jyang@gmail.com

Abstract. With continued aggressive process scaling in the subwavelength lithographic regime, resolution enhancement techniques (RETs) such as optical proximity correction (OPC) are an integral part of the design to mask flow. OPC creates complex features to the layout, resulting in mask data volume explosion and increased mask costs. Traditionally, the mask flow has suffered from a lack of design information, such that all features (whether critical or noncritical) are treated equally by RET insertion. We develop a novel minimum cost of correction (MinCorr) methodology to determine the level of correction of each layout feature, such that prescribed parametric yield is attained with minimum RET cost. This flow is implemented with model-based OPC explicitly driven by timing constraints. We apply a mathematical-programming-based slack budgeting algorithm to determine OPC level for all polysilicon gate geometries. Designs adopted with this methodology achieve up to 20% Manufacturing Electron Beam Exposure System (MEBES) data volume reduction and 39% OPC run-time improvement. *© 2007 Society of Photo-Optical Instrumentation Engineers.* [DOI: 10.1117/1.2774994]

Subject terms: resolution; integrated circuits; optical proximity correction; resolution enhancement techniques; mask cost; minimum cost of correction.

Paper 06084SSR received Nov. 3, 2006; revised manuscript received Mar. 16, 2007; accepted for publication Apr. 17, 2007; published online Sep. 17, 2007.

1 Introduction

Continued technology scaling in the subwavelength lithography regime results in printed features that are substantially smaller than the optical wavelength used to pattern them. For instance, modern 130-nm complementary metal-oxide semiconductor (CMOS) processes use 248-nm exposure tools, and the industry roadmap through the 45-nm technology node will use 193-nm (immersion) lithography. The International Technology Roadmap for Semiconductors (ITRS)[1] identifies aggressive microprocessor (MPU) gate lengths and highly controllable gate critical dimension CD control as two critical issues for the continuation of Moore's law cost and integration trajectories. To meet ITRS requirements (see Table 1), resolution enhancement techniques (RETs) such as optical proximity correction (OPC) and phase-shift masks (PSMs) are applied to an increasing number of mask layers and with increasing aggressiveness. The recent steep increase in mask costs and lithographic complexity due to these RET approaches has had a harmful impact on design starts and project risk across the semiconductor industry. Cost of ownership (COO) has become a key consideration in adoption of various lithography technologies.

1.1 *Optical Proximity Correction and 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 0.35 μm, the cost at the 0.13-μm generation with extensive PSM implemented is four times larger.[2] Figure counts, corresponding to polygons, as seen in the integrated circuits (IC) layout editor, grow tremendously due to subresolution 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 2005 ITRS,[1] the maximum single-layer MEBES file size increases from 64 GB in 130 nm to 216 GB in 90 nm. 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.[3] Half of all masks produced are used on less than 570 wafers (this translates roughly to production volumes of ≤100,000 parts). At such low usages, the high added costs of RETs cannot be completely amortized, and the corresponding cost per die be-

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

Year	2005	2007	2010	2013
Technology node	90 nm	65 nm	45 nm	32 nm
MPU gate length	32 nm	25 nm	18 nm	13 nm
MPU Gate CD 3σ	3.3 nm	2.6 nm	1.9 nm	1.3 nm

1537-1646/2007/$25.00 © 2007 SPIE

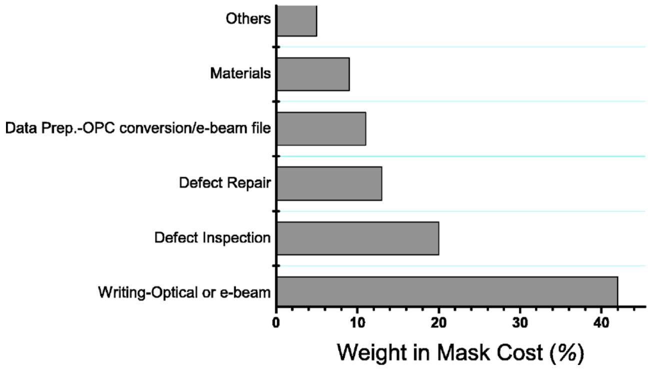

Fig. 1 Relative contributions of various components of mask cost for 130-nm design and below.[5]

comes very large. Thus, designers and manufacturers are jointly faced with determining how best to apply RETs to standard cell libraries to minimize mask cost.

In this work we focus on OPC, which is a major contributor to mask costs as well as design turnaround time (TAT). More than a 5× increase in data volume and several days of CPU run time are common side effects of OPC insertion in current designs.[4] With respect to the cost breakdown shown in Fig. 1, OPC affects mask data preparation (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 where run time is roughly proportional to feature complexity, comprise the dominant approach to high-speed mask writing. In the standard mask data preparation flow, the input Graphic Design System II (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 increases by 4 to 8× over a non-OPC layout, driving up the resulting GDSII file size as well as fractured data (e.g., MEBES format) volume.[5] 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 (e.g., according to the 2005 ITRS,[1] the maximum single-layer MEBES file size increases from 216 GB in 90 nm to 729 GB in 65 nm) 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,[3] since mask 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.

1.2 *Design Function in the Design-Manufacturing Interface*

A primary failing of current approaches to the design-manufacturing interface is in lack of communication across disciplines and/or tool sets. For example, it is well documented that mask writers do not differentiate among shapes being patterned. Given this, gates in critical paths are given the same priority as pieces of a company logo, and errors in either of these shapes will cause mask inspection tools to reject a mask. In this light, 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, such "overcorrection" leads to higher mask costs and larger run times.

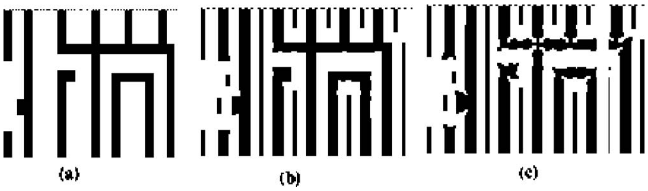

Fig. 2 An example of three levels of OPC.[5] (a) no OPC, (b) medium OPC, and (c) aggressive OPC.

1.3 *Performance-Driven Optical Proximity Correction Methodology*

In this work, we propose a performance-driven OPC methodology that is demonstrated to be highly implementable within the limitations of current industrial design flows. Contributions of our work include the following.

- *Quantified CD error tolerance.* We propose a mathematical programming based budgeting algorithm that outputs edge placement error tolerances (in nanometers) for layout features.
- *Integration within a commercial MDP flow.* We describe a practical flow implementable with commercial tools and validate the minimum cost of correction methodology.
- *Reduction of OPC overhead.* We measure OPC overhead in terms of additional MEBES features as well as run time of the OPC insertion tool, and show substantial improvements in both.

2 General Cost of Correction Flow (Minimum Cost of Correction) Based on Sizing

We describe a generic yield closure flow that is very similar to traditional flows for timing closure. In this section, we describe the elements of such a flow.

In this generic *sizing-based minimum cost of correction (MinCorr)* flow, we emphasize the striking similarity to conventional timing optimization flows. The key analogy—and assumption—is that there are discrete allowed "sizes" in the MinCorr problem that correspond to allowed levels of OPC aggressiveness (see Fig. 2).[6] Furthermore, for each instance in the design there is a cost and delay penalty associated with every level of correction. The mapping between traditional gate sizing and the MinCorr problems is reproduced in Table 2. This flow involves construction of cost/yield aware libraries for each level of correction, and a commercial STA tool together with a selling point yield bonding algorithm, which applies timing driven cost optimization. We acknowledge the following facts during the flow development process,

- We assume that different levels of OPC can be independently applied to any gate in the design. Corre-

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.

Gate sizing		MinCorr
Area	≡	Cost of correction
Nominal delay	≡	Delay $\mu + k\sigma$
Cycle time	≡	Selling point delay
Die area	≡	Total cost of OPC

sponding to each level of correction, there is an effective channel length L_{eff} variation and an associated cost.

- Differentiate field-poly from gate-poly features. Field-poly features do not impact performance and hence any delay-constrained MinCorr approach should not change the correction of field poly. Moreover, quality metrics of field poly are different from those of gate poly (e.g., contact coverage). By recognizing these two types of poly features, we may avoid overestimating cost savings achieved with this approach.
- The mask writing time that dominates mask cost[7] is a linear function of figure count numbers.[8] These numbers, as proxies for mask cost implications, are extracted for the cells from post-OPC layout with a commercial OPC insertion tool.
- OPC corrects the layout for pattern-dependent through-pitch CD variation. Such variations are predictable, for example, by lithography simulations.

With these facts, the *MinCorr* problem is summarized as follows. Given a range of allowable corrections for each feature in the layout as well as the mask data volume and CD deviation associated with each level of correction, find the level of correction for each feature such that prescribed circuit performance is attained with minimum total correction cost. Commercial OPC tools are driven by *edge placement errors* (EPEs), rather than critical dimensions (CDs). Thus, we specify a practical *MinCorr* with a practical implementation—*EPEMinCorr*. We can summarize the key contribution of EPEMinCorr as: *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* (e.g., Fig. 2 shows OPC layers driven by different EPE requirements). 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 Fig. 3. Mask data volume is heavily dependent on the assigned EPE tolerance that the OPC insertion tool is asked to achieve. For example, Fig. 4 shows the change in MEBES file size for cells 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.

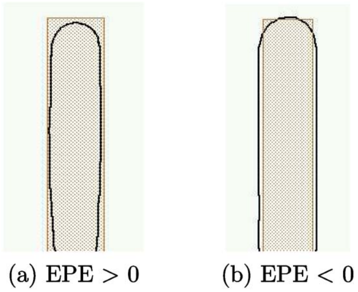
(a) EPE > 0 (b) EPE < 0

Fig. 3 The signed edge placement error (EPE): (a) EPE>0 and (b) EPE<0.

Since model-based OPC corrects for pattern-dependent CD variation, which is systematic and predictable, we assert that OPC actually determines *nominal timing*. This allows us to base our OPC insertion methodology on traditional corner-case timing analysis tools instead of (currently nonexistent from a commercial standpoint) statistical timing analysis tools. Our methodology adopts a slack budgeting based approach, as opposed to the sizing-based approach as mentioned earlier, 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 5 shows our EPEMinCorr 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 run time, which can be prohibitive when run at the

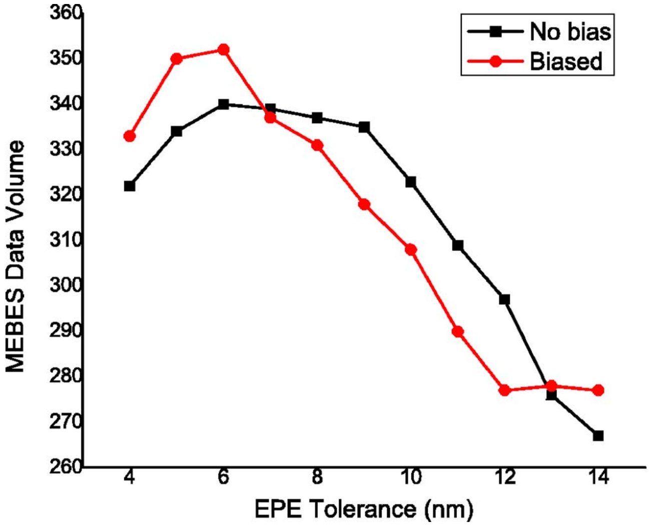

Fig. 4 Mask data volume (kB) versus EPE tolerance for a NAND3X4 cell in TSMC 130-nm technology.

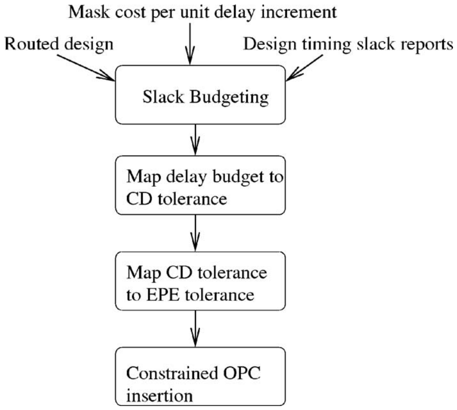
Mask cost per unit delay increment
Routed design
Design timing slack reports
Slack Budgeting
Map delay budget to CD tolerance
Map CD tolerance to EPE tolerance
Constrained OPC insertion

Fig. 5 The EPEMinCorr flow to find quantified edge placement error tolerances for layout features and drive OPC with them.

full-chip level. In the remainder of this section, we describe details of the major steps of the Fig. 5 EPEMinCorr flow.

2.1 *Slack Budgeting*

The slack budgeting problem seeks to distribute slack at the primary inputs of combinational logic (i.e., sequential cell outputs) to various nodes in the design. One of the earliest and simplest approaches, the zero-slack algorithm (ZSA),[9] iteratively finds the minimum-slack timing path and distributes its slack equally among the nodes in the path. The maximal-independent-set based algorithm (MISA) for slack budgeting proposed in Ref. 10 distributes slack iteratively to an independent set of nodes. As with ZSA, the objective is to maximize the total added incremental delay budget on timing arcs. A weighted version of MISA is also proposed in Ref. 10.

We observe the following.

- Neither MISA variant is guaranteed to provide optimal solutions.
- ZSA is much faster than MISA, and a weighted version of ZSA can also be formulated.
- While Ref. 11 formulates the budgeting problem as a convex programming problem, full-chip MISA or mathematical programming is, as far as we can determine, too CPU intensive for inclusion in a practical flow.

We propose to approximate full-chip mathematical programming by iteratively solving a sequence of linear programs (LPs). In each iteration, slack is budgeted among the top k available paths. Once a budget is obtained for a node, this budget is retained as an upper bound for subsequent iterations. The process is repeated until all nodes have been assigned a slack budget, or path slack is sufficiently large. The basic LP has the following form:

$$\text{Maximize} \sum_{i=1}^{n} C_i s_i$$

$$\sum_{j \in P_k} s_j \leq S_k \,\forall\, k \in \text{ current path list}$$

$$s_j \leq s_j^f \,\forall\, j \in F, \tag{1}$$

where C_i denotes the correction cost decrease per unit delay increase for cell i, and s_i is the slack allocated to cell i. The notation P_k is used to denote the k'th most critical path, and S_k is the slack of this path. Finally, F denotes the set of nodes with slacks fixed from previous iterations. An example sequence of LPs might be obtained by allowing k to take on the range from 1 to 100 in the first iteration, 101 to 200 in the second iteration, and so on.

We observe that when a budgeting formulation is adopted in place of a sizing formulation, the method of accounting for changes in next-stage input pin capacitance becomes an open question. To be conservative, we generate timing reports with pin input capacitances that correspond to the loosest tolerance (i.e., largest pin capacitance) but gate delays corresponding to the tightest achievable tolerance. C_i is obtained via a prebuilt look-up table (similar to .lib format) containing the increase in data volume, mapped against delay change.

Our budgeting procedure yields positive delay budgets leading to positive EPE tolerances. Since EPE tolerance is a signed quantity (e.g., in Mentor Calibre, a common OPC insertion tool), negative EPE tolerances (corresponding to reduced gate length and faster delay) can also be obtained in a similar way based on hold time or leakage power constraints. However, in this work we assume equal positive and negative EPE tolerances, since we deal with purely combinational benchmarks and focus on timing rather than power.

2.2 *Calculation of Critical Dimension Tolerances*

To map delay budgets found from the previous 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 100 ps, 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. Subtracting these budgeted gate lengths from nominal gate lengths yields the CD tolerance for every cell in the design.

2.3 *Calculation of Edge Placement Error Tolerances*

The next step in our flow maps CD tolerances to signed EPE tolerances. Again, obtaining EPE tolerances is crucial, since this is the parameter that OPC insertion tools understand and can exploit. As noted before, 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 the 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 prebiasing of gate features in an attempt to achieve EPE tolerances that are equal to CD tolerance. Specifically, we prebias each gate feature by its intended EPE tolerance. For instance, for a drawn gate length of 130 nm and EPE tolerance of 10 nm, the printed CD would typically lie between 110 and 130 nm (each edge shifts by 10 nm inward). If the gate length is biased by 10 nm so that the OPC tool views 140 nm as the target CD, the printed CD would lie between 120 and 140 nm, which amounts to a ±10-nm CD tolerance. In this way, prebiasing achieves CD tolerances equal to the EPE tolerance. An example of the average CD for a specific gate poly with and without prebiasing is shown in Fig. 6. It is clear that prebiasing achieves its goal of attaining average CDs that are very close to the target CD (130 nm in our case). Another point illustrated in Fig. 6 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. This is shown more clearly in Sec. 3.4.

2.4 *Constrained Optical Proximity Correction*

We enforce the obtained EPE tolerances within a commercial OPC insertion flow. We use *Calibre*[12] (Mentor Graphics Corp., Wilsonville, OR) as the OPC insertion tool; details of constraining the tool are described in the next section.

3 Experimental Setup and Results

In this section we describe our experiments and the results obtained to validate the EPEMinCorr methodology.

3.1 *Test Cases*

We use several combinational benchmarks drawn from the ISCAS85 suite of benchmarks and Opencores.[13] These benchmark circuits are synthesized, placed, and routed in a restricted TSMC 0.13-μm library containing a total of 32 cell macros with cell types of BUF, INV, NAND2, NAND3, NAND4, NOR2, NOR3, and NOR4. The test case characteristics are given in Table 3.

3.2 *Library Characterization*

We assume a total of EPE tolerance levels ranging from ±4 to ±14 nm. Corresponding to each EPE tolerance, the worst-case gate length is 130 nm+*EPE_tolerance*. We map cell delays to EPE tolerance levels by creating multiple .lib files for each of the ten 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

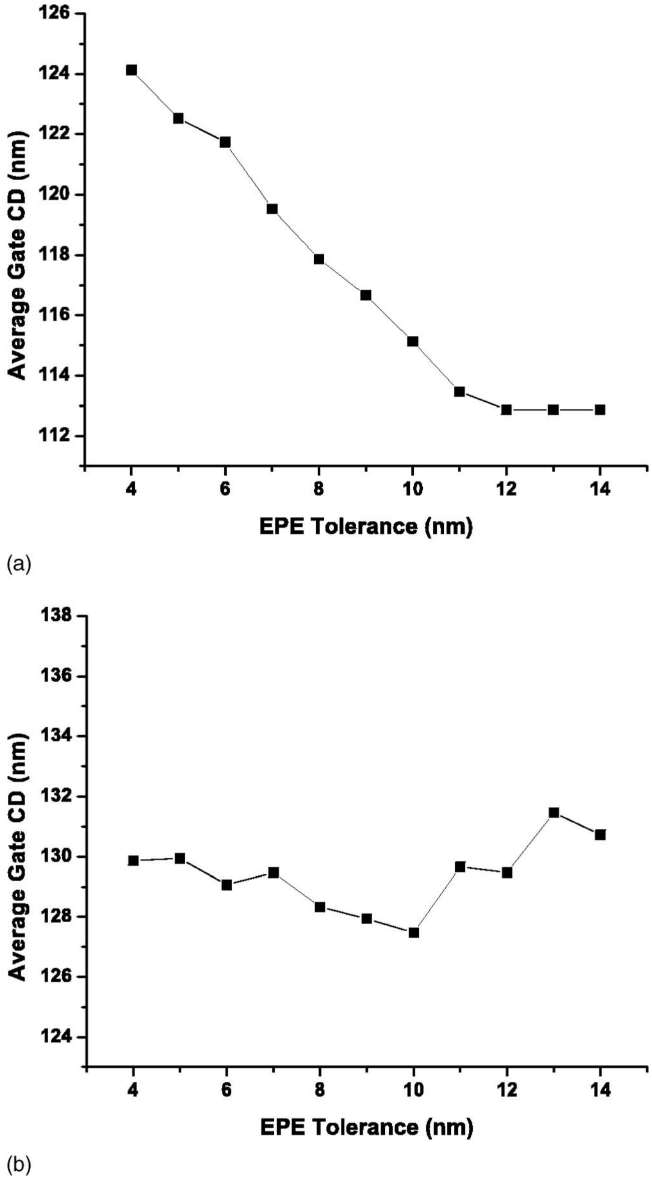

Fig. 6 Comparison of average printed gate CD with and without prebias for the cell macro NAND3X4: (a) without and (b) with prebias.

Table 3 Benchmark details.

Test case	Source	Cell count
c432	ISCAS85	337
c5315	ISCAS85	2093
c6288	ISCAS85	4523
c7552	ISCAS85	2775
alu128	Opencores	12,403
r4_sova	Industry	34,288

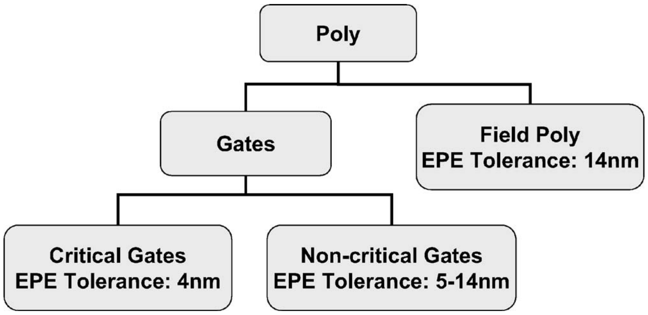

Fig. 7 Summary of EPE assigment for OPC level control.

MEBES data volume numbers for each (cell, tolerance) pair. Though the exact corrections applied to a cell will depend somewhat on its placement environment, standalone 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 Fig. 4 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 4 nm). When slack is distributed to various nodes, we extract the load capacitances that are used to identify entries in the sensitivity table. Cost change is most sensitive to delay changes when the load capacitance is small (this typically indicates a small driver and subsequently small amount of data volume) and the sensitivity numbers are on the order of 1 to 10× MEBES features per ps delay change.

3.3 *Edge Placement Error Minimum Cost of Correction 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. As indicated in Fig. 7, we first separate the entire poly layer into gate-poly and field-poly components. The field-poly tolerance is taken to be ±14 nm, while gate-poly tolerance ranges from ±4 to ±14 nm. We tag the assigned EPE tolerance to cell names. In this way, we can track the EPE tolerance of each gate individually. We take 1 nm as our step size (step size is the minimum perturbation to an edge that a model-based OPC can make, and smaller step sizes lead to better correction accuracy at the cost of run time) 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.

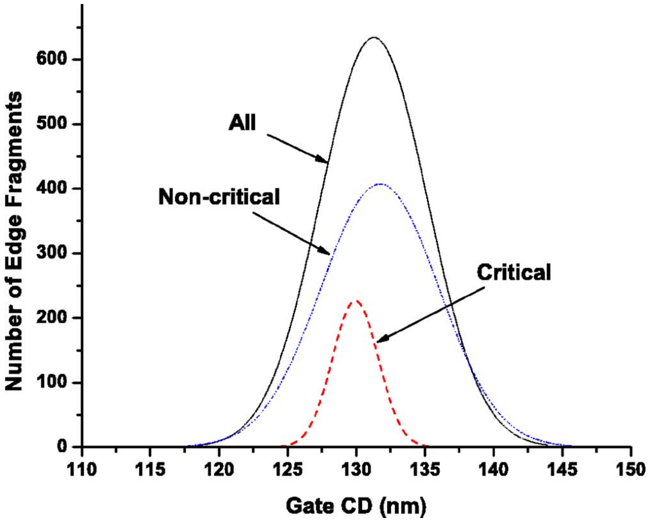

Fig. 8 Gate CD distribution for c432. Gates with budgeted 4-nm EPE tolerance are labeled critical gates, while others are labeled as noncritical. The *y* axis shows the number of fragments of gate edges with a given printed CD.

3.4 *Results*

We synthesize the benchmark circuits using a *Synopsys design compiler*. Place and route is performed using *Cadence Silicon Ensemble* (San Jose, CA) *Synopsys Primetime* (Mountain View, CA) is used to output the slack report of the top 500 critical paths (not true for the biggest benchmark r4_sova, where more paths are needed, as discussed later), as well as the load capacitance for each driving pin. As noted before, STA is run with a modified 134-nm (EPE tolerance tightest on gate poly and loosest on field poly) library with pin capacitances corresponding to 144 nm (loosest EPE tolerance) to remain conservative after slack budgeting. We use *CPLEX v8.1*[14] (Sunnyvale, CA) as the mathematical programming solver to solve the budgeting linear program. Two types of benchmarks are involved in our experiments: 1. large designs with a "wall" of critical paths, e.g., r4_sova in Table 3; and 2. circuits with fairly small sizes, e.g., benchmarks except r4_sova. For 2. a single iteration is efficient to solve the budgeting problem; for 1. however, more iterations may be necessary because some paths that are potentially critical but are not reported due to the constraint of maximum number of critical paths may become top critical later on, as they are not treated as optimization objects by the slack budgeting algorithm, resulting in performance degradation. One possible solution to this problem is to perform iterations to selectively include those paths that may cause performance degradation, as slack budgeting objects. Another simple but not as efficient option is to increase the constraint of maximum number of critical paths in the slack report. We deploy a hybrid way for r4_sova in our case, i.e., the constraint on the initial number of critical paths is increased from 500 to 10,000, then in each iteration 5000 more paths that are potentially critical are included for slack budgeting. After eight iterations, the performance degradation due to the selective OPC is reduced to less than 1% (first iteration gives 4.3% performance degradation).

The extracted CD variation for test case *c*432 after

Table 4 Impact of EPEMinCorr optimization on cost and CD. All run times are based on a 2.4-GHz Xeon machine with 2-GB memory running Linux.

	Traditional OPC flow				EPEMinCorr flow							
	CD distribution					CD distribution						
	All gates (nm)		OPC runtime (h)	Delay (ns)	Budgeting run time (s)	All gates (nm)		Critical gates (nm)		OPC run time (h)	Delay (ns)	Normalized MEBES volume
Test case	Mean	σ				Mean	σ	Mean	σ			
c432	130.9	1.55	0.2643	1.33	1	131.3	3.90	129.9	1.67	0.2047	1.33	0.87
c5315	130.2	1.83	1.261	1.94	3	131.7	4.70	129.7	1.89	1.180	1.94	0.82
c6288	129.7	1.52	3.275	5.21	9	131.4	4.45	129.7	1.27	2.697	5.21	0.86
c7552	129.6	1.65	1.856	1.59	4	132.0	4.77	130.1	1.99	1.428	1.59	0.81
alu128	130.4	1.63	13.89	3.28	11	131.5	4.93	130.8	2.04	9.215	3.28	0.80
r4_sova	130.1	1.98	38.65	8.19	29,648	131.9	5.00	130.0	1.75	23.32	8.26	0.80

EPEMinCorr OPC is shown in Fig. 8. The distributions show that Calibre is able to enforce assigned tolerances very consistently. A tighter CD distribution for critical gates is achieved, while noncritical gates (which can tolerate a larger deviation from nominal) have a more relaxed (and hence less expensive to implement) gate length distribution. Table 4 compares the run time and data volume results for EPEMinCorr OPC and traditional OPC. For relatively small circuits, a single iteration of the budgeting approach ensures that there is no timing degradation going from the traditional to the EPEMinCorr flow, and the budgeting run times are negligibly small, ranging from 1 to 11 s. For large designs, especially those with a "wall" of critical paths, iterations may be required to avoid performance degradation and the sum of budgeting run times of each iteration may reach several hours (7 h for r4_sova). The important result is the amount of mask cost reductions achieved whether measured as run time of model-based OPC or fractured MEBES data volume. EPEMinCorr flow reduces MEBES data volume by 13 to 20%. Such reductions directly translate to substantial mask write time improvements. OPC run times are improved by 6 to 39%. These percentage numbers translate to huge absolute TAT savings. For instance, the EPEMinCorr flow saves 16.3 h compared to the traditional OPC flow on a 34000 gate benchmark.

4 Conclusions and Future Work

We propose and implement a practical means of reducing mask 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. An iterative linear-programming-based approach is used to perform slack budgeting in an efficient manner. This formulation results in a specific slack budget for each gate, which is then mapped to allowable critical dimensions in the standard cell. Finally, EPEs are generated from the CD budget and tags are placed on gates to indicate to the OPC insertion tool the appropriate level of correction. Our results on several benchmarks ranging from 300 to 34,000 cells show up to 20% reductions in MEBES data volume, which is frequently used as metric for RET complexity. Furthermore, the run time of the OPC insertion tool is reduced by up to 39% -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 future technologies, allowable CD tolerances may be set more by bounds on acceptable leakage power than by traditional delay uncertainty constraints. We plan to incorporate power constraints into our formulation. Moreover, we plan to extend the EPEMinCorr methodology for field-poly features. Impact of field polysilicon shapes on performance comes from their overlap with contact layers, so field-poly extensions to EPEMinCorr will have to evaluate error in terms of contact coverage area. Expensive masking layers include diffusion, contact, metal1 and metal2, besides polysilicon. The performance impact of OPC errors on these other layers can also be computed and consequently EPEMinCorr methodology extended.

Another direction of work is exploring other degrees of freedom in OPC besides EPE tolerance, which have a strong effect on mask cost. Two such parameters are fragmentation and minimum jog length.

In a follow-up work of an industrial scale of application[15] a methodology similar to EPEMinCorr was used to optimize mask cost for a big design block. The resulting OPC layout went through dummy mask write at a mask shop. The authors reported 25% shot count reduction and up to 32% reduction in mask write time.

References

1. International Technology Roadmap for Semiconductors, see http://public.itrs.net/ (2005).
2. C. Yang, "Challenges of mask cost and cycletime," *SEMATECH: Mask Supply Workshop*, Intel, Monterey, CA (2001).

3. M. L. Rieger, J. P. Mayhew, and S. Panchapakesan, "Layout design methodologies for sub-wavelength manufacturing," *Proc. Design Auto. Conf. (DAC)*, pp. 85–92, IEEE/ACM (2001).
4. L. Liebmann, "It's not just a mask," *Proc. SPIE* **4409**, 23–32 (2001).
5. S. Murphy, "Dupont photomask," *SEMATECH: Mask Supply Workshop*, Monterey, CA (2001).
6. K. Wampler, ASML MaskTools, private Communication (Mar. 2003).
7. "*Optical lithography cost of ownership*," *Final Report*, see http://www.sematech.org/docubase/document/4014atr.pdf.
8. C. Spence, S. Goad, P. Buck, R. Gladhiu, R. Cinque, J. Preuninger, U. Griesinger, and M. Blöcker, "Mask data volume—historical perspective and future requirements," *Proc. SPIE* **6281** (2006).
9. R. Nair, C. L. Berman, P. S. Hauge, and E. J. Yoffa, "Generation of performance constraints for layout," *IEEE Trans. Comput.-Aided Des.* **8**(8), 860–874 (1989).
10. C. Chen, E. Bozorgzadeh, A. Srivastava, and M. Sarrafzadeh, "Budget management with applications," *Algorithmica*, pp. 261–275 (2002).
11. E. Bozorgzadeh, S. Ghiasi, A. Takahashi, and M. Sarrafzadeh, "Optimal integer delay budgeting on directed acyclic graphs," *Design Auto. Conf. (DAC)*, pp. 920–925, IEEE/ACM (2003).
12. See http://www.mentor.com.
13. See http://www.opencores.org.
14. See http://www.ilog.com.
15. Y. Zhang, R. Gray, O. S. Nakagawa, P. Gupta, H. Kamberian, G. Xiao, R. Cottle, and C. Progler, "Interaction and balance of mask write time and design RET strategies," *Proc. SPIE* **5853**, 614–618 (2005).

Puneet Gupta received the BTech degree in electrical engineering from the Indian Institute of Technology, Delhi in 2000. He joined the electrical and computer engineering department at University of California (UC), San Diego, in 2001, where he is currently a PhD candidate. He has been at Blaze DFM Incorporated since 2004 as cofounder and product architect. His research has focused on building high-value bridges between physical design and semiconductor manufacturing for lowered cost, increased yield, and improved predictability of integrated circuits. He has authored more than 40 papers and is a recipient of an IBM PhD fellowship. He holds one U.S. patent and has 12 pending. He has given tutorial talks at ICCAD, WesCon, CMP-MIC, UC Santa Cruz, and UC San Diego, and was a short-course instructor at SPIE's Advanced Lithography 2007.

Andrew B. Kahng is professor of computer science and engineering and electrical and computer engineering at UC San Diego. He holds an AB degree in applied mathematics from Harvard College, and MS and PhD degrees in computer science from UCSD. He received a 1992 National Science Foundation (NSF) Young Investigator Award, and has approximately 300 publications in the VLSI CAD area, including five best paper awards and six other best paper nominations. Since 1997, his research in VLSI design for manufacturing has pioneered methods for automated phase-shift mask layout, CMP fill synthesis, and parametric yield-driven, cost-driven, and variation-aware optimizations.

Dennis Sylvester received the BS degree in electrical engineering summa cum laude from the University of Michigan, Ann Arbor, in 1995. He received the MS and PhD degrees in electrical engineering from University of California, Berkeley, in 1997 and 1999, respectively. His dissertation research was recognized with the 2000 David J. Sakrison Memorial Prize as the most outstanding research in the UC-Berkeley EECS department. He is now an associate professor of electrical engineering and computer science at the University of Michigan, Ann Arbor. He is also a visiting associate professor of electrical and computer engineering at National University of Singapore during the 2006 to 2007 academic year. He has published numerous articles along with one book and several book chapters in his field of research, which includes low-power circuit design and design automation techniques, design-for-manufacturability, and on-chip interconnect modeling. He received an NSF Career award, the 2000 Beatrice Winner Award at the International Solid State Circuits Conference (ISSCC), an IBM Faculty Award, a Semiconductor Research Corporation (SRC) Inventor Recognition Award, and several best paper awards and nominations. He is the recipient of the Association for Computing Machinery (ACM) Special Interest Group on Design Automation (SIGDA) Outstanding New Faculty Award, the 1938E Award for teaching and mentoring, and the Vulcans Education Excellence Award from the College of Engineering, and the University of Michigan Henry Russel Award.

Jie Yang received the BS degree in microelectronics from Peking University, Beijing, China in 2001. She joined the electrical engineering and computer science department at University of Michigan, Ann Arbor, in 2001, where she is currently a PhD candidate. She has been with Advanced Micro Devices (AMD) since 2004. Her research has focused on the physical design optimizations for manufacturability, including printability verification EDA tool and flow, proximity effect embedded timing flow, intra-die and inter-die process variation modeling, cost-effective design rule exploration, and timing driven manufacturability enhancements. She holds one U.S. patent, has three pending, and one best paper nomination from the design automation conference.