Merits of Cellwise Model-Based OPC

Puneet Gupta[†], Fook-Luen Heng and Mark Lavin[‡]

[†]Electrical and Computer Engineering Dept., UCSD, La Jolla, CA
[‡] IBM T.J. Watson Research Lab, Yorktown Heights, NY.

ABSTRACT

One of the most compute intensive dataprep operations for 90nm PC level is the model-based optical proximity correction (MBOPC). The running time and output data size are growing unacceptably, particularly for ASICs and designs containing large macros built out of library cells (books). The reason for this growth is that the region-of-interest for MBOPC is approximately 600nm, which means that most library cells "see" interactions with adjacent books in the same row and also in adjacent rows.

In this paper, we investigate the merits of doing cellwise MBOPC. In its simplest form, the approach is to perform dataprep for each cell once per cell definition rather than once per placement. By inspection, this will reduce the computation time and output data size by a factor of P/D, where P is the number of book placements (100s to millions) and D is the number of book definitions.

Our preliminary finding indicates that there is negligible difference between nominal CD for cellwise corrected cells and chipwise corrected cells. We will present our finding in terms of average CD and contact coverage, as well as runtime reduction.

1. INTRODUCTION

Moore's law continues to drive higher performance with smaller circuit features. Consistent improvements in the resolution of optical lithography techniques have been a key enabler for 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 comparable to the minimum line dimensions to be printed. Optical lithography is being pushed to its limits with 193nm lasers being used to fabricate devices of dimensions of 90nm or less. The extension of optical lithography has been enabled by several developments such as chemically amplified photoresists and anti-reflective coatings. By predicting physical phenomena (especially diffraction and interference) behind optical systems and systematically compensating for them, the minimum feature and pitch that can be resolved are significantly extended. These Resolution Enhancement Techniques (RETs) are aimed at three major optical wave components, namely, direction, amplitude and phase. Optical Proximity Correction (OPC)is one such RET which makes small alterations to the mask features to reduce linewidth variation.

Traditionally, model-based OPC is applied after layout is completed and is done on the entire GDSII of the design. Such "flat" OPC, though accurate is very compute-intensive. In this paper, we propose a novel cell-wise OPC technique wherein the standard cell layouts are corrected at the library-design step rather than post-tapeout. This leads to huge savings in OPC runtime at little loss of accuracy. Predictability and reduction in layout database file sizes are other advantages of such an approach.

This paper is organized as follows. Next section motivates the need for cell-wise OPC in detail. Section 3 introduces the two metrics of designers' intent that we use to evaluate OPC results. Section 4 describes the cell-wise OPC flow. Experiments and results are described in Section 5. We conclude with suggestions for future work in Section 6.

Corresponding author: Puneet Gupta. Emails: Puneet Gupta (puneet@ucsd.edu), Fook-Luen Heng (heng@us.ibm.com), Mark Lavin (malavin@us.ibm.com).

Design and Process Integration for Microelectronic Manufacturing II, edited by
Lars W. Liebmann, Proc. of SPIE Vol. 5379 (SPIE, Bellingham, WA, 2004)
0277-786X/04/$15 · doi: 10.1117/12.535852

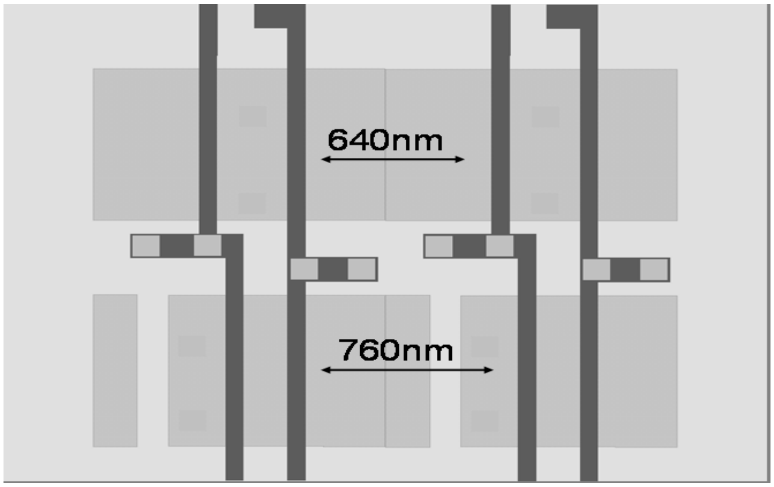

Figure 1. Placement of two NAND gates in 90nm technology with zero whitespace. Note that the minimum distance between NMOS or PMOS devices in neighboring cells is greater than 600nm which is the radius of influence.

2. MOTIVATING THE NEED FOR CELL-WISE OPC

OPC has traditionally been performed after tapeout during mask-dataprep. OPC tools operate on full design GDSII files. Correction can be rule-based or model-based. Rule-based OPC relies on simple geometric rules to determine the layout modifications which are to be applied to a feature. Rule-based OPC is very fast and is commonly used for the metal layers. For the polysilicon or the device layer, rule-based OPC is not accurate enough at the 130nm node and beyond as the required printing resolution at the device layer is much greater than the metal layers. Model-based OPC works by extensive lithography simulation and iteration to correct the layout. The entire design GDSII database is read into the OPC tool and the polygons are corrected in a flat manner. Such OPC is very accurate but is exceedingly slow. Typical runtime numbers quoted in industry are as bad as several CPU days for a modern 90nm design.[4] Such prohibitive runtimes as well as indispensability of model-based OPC for accuracy motivates alternative faster methods for OPC.

We propose a library-based method for cell-wise OPC. Instead of correcting the entire layout, stand-alone standard cell layouts can be corrected and the OPC'd versions of the cell layouts can be substituted after place and route. We believe that the loss of accuracy compared to traditional flat full-chip OPC is minimal. Optical diffraction effects tend to extend only up to a certain radius of influence beyond which they are negligibly small. For the currently in-use 193nm steppers, this radius of influence is believed to be about 600nm. Most commercial lithography simulators also do not look beyond this distance when evaluating the neighborhood of a geometry.[5] Moreover, due to various constraints in cell layout design (e.g., diffusion breaks) there are always large blocks of empty space near the periphery of a cell. For example for a 90nm standard cell library the distance between the outlying PMOS and NMOS devices and the cell outline tends to be 200-400nm. As a result the minimum spacing between an outlying device in a cell and nearest device in the neighboring cell in a placement is greater than 400nm. Coupled with existence of whitespace in the design, usually this inter-device spacing is greater than 600nm, i.e. the radius of influence.

The environment of gate-poly determines its correction by the OPC tool. Except for the (at most) four devices lying at the periphery of the cell, all internal devices in the cell layout have their environments determined by other devices *within* the cell. This is because the nearest neighbors of these internal gates belong to the cell itself while the next to nearest neighbors are too far ($\geq$600nm) to have any impact on printability of these devices. Moreover, as noted above the peripheral or outlying devices in a cell layout also usually have a known environment composed of empty space on one (or both) sides and a known device on the other. This environment is depicted in Figure 1 which shows two NAND gates in 90nm technology placed next to each other with zero whitespace. From this discussion we conclude that for a standalone cell, all non-peripheral devices in it can be OPC'd without any loss in accuracy compared to flat full-chip OPC as the environment for these devices does not depend on the cell placement. Moreover, there should be little loss in accuracy for the remaining peripheral devices as they also have their environment almost completely determined. Therefore, cell-wise OPC is likely to yield results comparable to full-chip OPC.

The advantages of the cell-wise OPC approach are obvious. They are listed below.

- *OPC Runtime Saving.* Cell-wise OPC corrects each cell once per definition rather than once per instance. A single standard cell macro may be instantiated in the placement hundreds to hundreds of thousands times. Full-chip OPC corrects all these instances individually. Cell-wise OPC corrects each cell macro only once (say at the library design step). This results in orders of magnitude savings in OPC tool runtime.
- *Data Reduction.* After OPC, layout file size explodes. For instance, for a 5903 gate design, the GDSII file size increases from 59KB to 1396KB after model-based OPC. Maintaining and transferring such large files to the mask-shop is a major concern which has recently fuelled research into layout data compression[1] and new data formats.[2] With cell-wise OPC, only a library of corrected cell masters and uncorrected layout needs to be saved. This can lead to large reduction in data.
- *Predictability During Layout.* If OPC is performed before layout such that corrected versions of cell masters are instantiated, then lithography simulations can be performed before library characterization. Such simulations can predict with reasonable accuracy the actual printed geometries on the wafer. Printed dimensions rather than drawn dimensions can then be used during parasitic extraction and timing/power characterization. This can result in more accuracy and predictability during design and less guardbanding. A secondary benefit which may be explored due to design-time predictability of process effects is less aggressive correction that can lead to savings in mask costs.

3. METRICS FOR DESIGNER'S INTENT

Rising complexity of resolution enhancement (RET) and mask data preparation (MDP) lead to loss in design tool quality as well as design productivity. These have resulted in increased project uncertainty and manufacturing NRE. Designer, EDA, and process communities must cooperate and co-evolve to maintain the cost (value) trajectory of Moores Law. Limits of mask flow need to be passed back to design while functional intent needs to be fed forward to the mask flow.

Conventional OPC techniques and state of the art commercial OPC tools are driven by edge placement errors. I.e., they try to match the location of the edges drawn by designer on the layout exactly on the printed wafer. Edge-matching is an overconstraint and is oblivious of designer's real intent which is performance (measured as speed, power, etc). The real driver for performance at the device-level whether speed or power is the gate-length or critical dimension (CD) of the device. This is the region of overlap between the polysilicon and the diffusion layers. Similarly, the large number of features (e.g., serifs and hammerheads) to avoid corner rounding amount to over-correction. The goal should be to achieve acceptable overlap between the contact layer and the polysilicon layer. For example, driving OPC to get perfect corners on poly-overhangs is essentially OPC-ing the OPC. Sufficient contact coverage (as defined by the designer) is a good enough driver for correction. The two metrics are shown in Figure 2 for a simple NAND gate.

At the level of mask-data preparation where design is essentially a collection of polygons, CD and contact coverage are two metrics which capture designer's intent reasonably well. In this paper we will use these two metrics to measure the quality of results of OPC.

4. CELLWISE OPC FLOW

Cell-wise OPC is library-based rather than design-based. The cell layouts are corrected as a post-library layout step before physical design. This is in contrast to conventional OPC flow where RETs are applied after design layout has been completed.

In the cell-wise OPC regime, it is important to estimate the "representative" environment for every cell. Several design and mask rule violations can result if the cell layout is corrected as-is. Moreover, since the

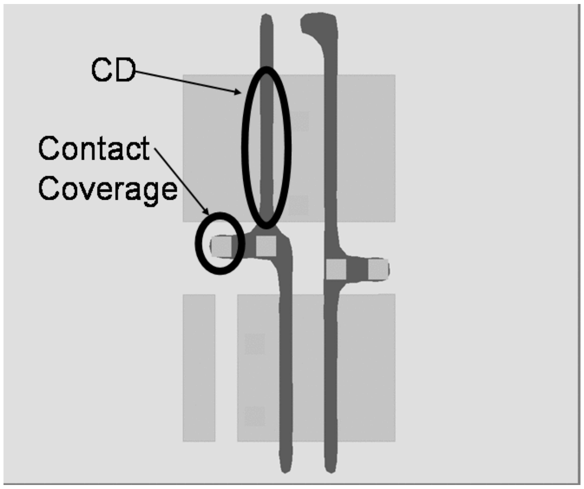

Figure 2: The two metrics of designers' intent: critical dimension (CD) or gate length and contact coverage.

typical environment of a cell in a full-chip layout may be completely different from an isolated cell, differences between cell-wise and full-chip OPC can be large. The choice of environment is more critical for contacts as the inter-contact spacing can be much smaller than the 600nm radius of influence in a full-chip layout. The three components of the environment of a cell that we consider important are listed below.

- *Border Poly.* In a typical standard cell placement, a cell instance will have two immediate neighbors. Both neighboring cells can be actual gates containing poly geometries or simple filler cells to cover whitespace. As a result, the gate poly shapes can have a variety of neighboring spaces depending on the space between the gate poly shape and the cell outline as well as the neighboring cell. To emulate a typical layout scenario, we insert dummy poly features at a predetermined spacing from the cell outline $before$ correcting the cell. These features are taken out from the layout after OPC. The border poly spacing has to be judiciously chosen. If it is taken to be too small, gate poly is corrected as "dense" lines and may end up printing smaller than intended.

- *Top-Bottom Poly.* Dummy features are inserted at top and bottom of the book also. These ensure that the poly in neighboring cell rows do not short or cause mask rule violations. These typically do not affect the CD but can have impact on contact coverage if there are contacts near the top or bottom boundary of the cell.

- *Contact Poly.* More dummy features may need to be inserted near the contacts so that the poly shapes after OPC in the region do not short or cause mask rule violations. These features can significantly reduce the contact coverage and should be inserted judiciously.

The three kinds of dummy features are shown in Figure 3 for a simple NAND gate.

After the correct environment for a cell is constructed, OPC is performed on it. Next, the dummy features added to construct the environment are removed to yield the corrected version of the cell layout. This layout may be simulated to yield the printed image of the cell for characterization.

5. EXPERIMENTS AND RESULTS

We conduct our experiments using IBM 90nm ASIC flow. Internal IBM tools are used for synthesis and placement. A commercial EDA software is used to perform model-based OPC as well as to obtain printimages of the layout. The optical and resist models used are actual plan-of-record models for the technology. All the results reported correspond to OPC without assist feature insertion. We test our cell-wise OPC flow on small

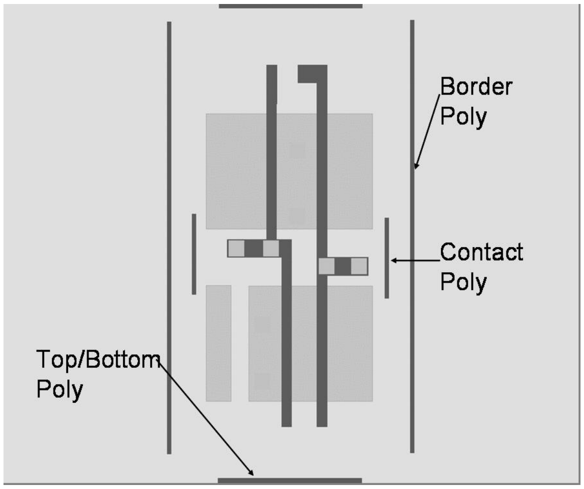

Figure 3: Three kinds of dummy poly features: border poly, top-bottom poly and contact poly.

ISCAS85 benchmark circuits. Moreover, we restrict ourselves to 10 cells from the library. These 10 cells are statistically the "most-used" combinational cells and we believe them to be fairly representative of the entire library. We investigate and compare two different OPC flows:

1. *OPC.* This is the conventional full-chip OPC flow where the entire layout is corrected.
2. *Cell-wise OPC.* This is the proposed cell-wise OPC flow. The cells are corrected standalone with a fixed environment but the image simulation is performed on the entire layout to measure CD and contact coverage.

In our results, we compare the cell-wise OPC and the OPC flow in terms of the metrics of designer's intent namely critical dimension and contact coverage.

CD is measured from the overlap between the polysilicon and the diffusion regions in the device under consideration. Since poly does not print as drawn, the gate length is nonuniform across the gate width. We compute the "average CD" as the poly-diffusion overlap area divided by the device width. This averaging is reasonable for predicting delay (as dependence of device delay on gate length is almost linear) but may not be as accurate for predicting other performance metrics (e.g. leakage). In a similar way, contact coverage is measured as the area of overlap between the contact and the poly.*

Table 1 shows the results for measured CD. About half of the devices have less than 1% error compared to full-chip OPC while almost all devices fall within 6% error. A sample CD error distribution is shown in Figure 4 from which it is clear that the majority of the devices have close to zero error. Similar results and plots for contact coverage simulation are shown in Table 2 and Figure 5. The error results for contact coverage are not as impressive as those for CD. We believe this to be due to exceedingly conservative construction of top-bottom and contact poly dummy features to avoid any mask rule violations. Further investigation of these errors is part of our ongoing work.

Runtime benefits of cell-wise OPC are also obvious from Table 1. The runtime of cell-wise OPC is just 90 seconds which corresponds to correction of 10 cell masters. Full-chip OPC runtime ranges from 185 seconds to 747 seconds for these small benchmark designs. The runtime reduction is likely to be much larger for state of the art designs.

Reduction in GDSII file sizes is another advantage of cell-wise OPC. For cell-wise OPC the layout data which needs to be maintained consists of 1) corrected layouts of the library cell masters; and 2) uncorrected

*We do not simulate the contact features but compute the coverage based on ideal contact shapes. We believe this issue is orthogonal since OPC is performed on a single layer at a time.

Testcase	$N_{1\%}$	$N_{3\%}$	$N_{6\%}$	Max -ve Error (%)	Max +ve Error (%)	Runtime (s)
C1355	58	83	97	7.8	15	477
C2670	45	78	96	9	15	747
C3540	40	77	96	10.2	14.7	1131
C432	35	76	97	8	13.2	185
C499	54	79	96	8	15	495

Table 1. Comparison of cell-wise OPC and full-chip OPC for measured CD. $N_{i\%}$ denotes the % of devices in cell-wise result which print with less than i% absolute difference from the full-chip OPC result. The total OPC runtime for 10 cells used for cell-wise OPC is 90 seconds.

Testcase	$N_{10\%}$	$N_{20\%}$	$N_{40\%}$	Max -ve Error (%)	Max +ve Error (%)
C1355	53	70	95	54	16
C2670	51	68	96	54	16
C3540	48	70	99	54	17
C432	44	64	100	0	42
C499	50	66	97	54	17

Table 2. Comparison of cell-wise OPC and full-chip OPC for measured contact coverage. $N_{i\%}$ denotes the % of devices in BBOPC result which print with less than i% absolute difference from the full-chip OPC result.

layout of the design as opposed to the corrected layout of the entire design as is the case with full-chip OPC. As OPC adds a large number of features to the uncorrected layout, cell-wise OPC can achieve large data volume reductions. The results for the small ISCAS85 benchmarks are shown in Table 3. The predictability aspects of the cell-wise OPC are used and explored in another work.[3]

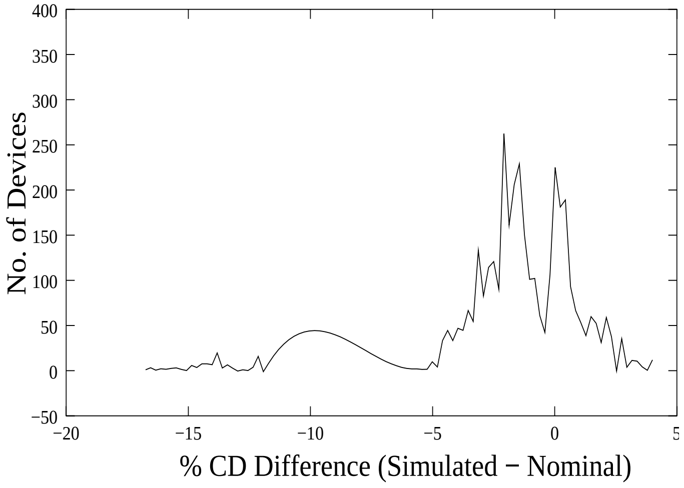

Figure 4: CD error distribution for C3540 benchmark.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a novel library-based cell-wise model-based OPC methodology. The key advantages of our approach lie in reduction in OPC runtime as well data volume. We see up to 8X runtime reduction even with small ISCAS85 circuits. Similarly, data volume reductions range from 2.8X to 12.2X. Impact of OPC can be accurately predicted and incorporated into timing with cell-wise OPC being performed before library characterization. This aspect is further explored in another work.[3]

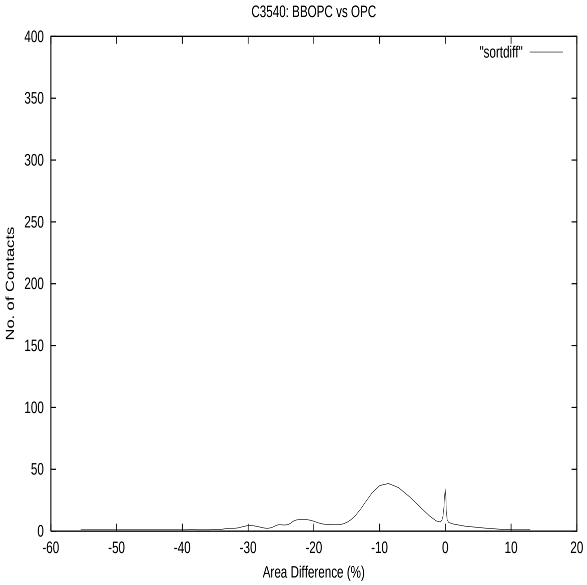

Figure 5: Contact coverage error distribution for C3540 benchmark.

Testcase	No. of Gates	**GDSII File Size (KB)**		
		Full-Chip OPC	Cell-wise OPC	Reduction
C1355	2058	422	75	5.6X
C2670	3655	756	92	8.2X
C3540	5903	1396	114	12.2X
C432	968	208	73	2.8X
C499	1728	464	80	5.8X

Table 3. Data volume reduction achieved by cell-wise OPC. The total data volume for the OPC'd versions of the 10 cells is 55KB.

Ongoing work includes the following.

- Our current experiments do no include assist features. Intelligent inclusion of assists in the cell-wise OPC flow can make the environment of devices very similar and hence further reduce the discrepancy between full-chip and cell-wise OPC flows. We intend to validate the cell-wise OPC approach with SRAFs.
- We are investigating the use of specially designed filler cells to make cell-wise OPC more effective. Having dummy poly geometries in filler cells can lead to restriction on the maximum poly-to-poly spacing in the design and thus help in getting a relatively uniform environment for peripheral poly geometries.
- We are also investigating ways to use less conservative contact and top-bottom poly so that the error in contact coverage can be reduced.

ACKNOWLEDGMENTS

The authors would like to thank Jin-Fuw Lee, Jim Culp and Yuping Cui for providing help with the OPC software. We also like to thank Kafai Lai and Ron Gordon for insightful discussions and Daniel L. Ostapko for reviewing the manuscript.

REFERENCES

1. A.B. Kahng, R. Ellis, and Y. Zheng, "Compression Algorithms for Dummy Fill Layout Data", *Proc. SPIE Conf. on Design and Process Integration for Microelectronic Manufacturing*, Feb. 2003, to appear.

2. New Standards Specification for Open Artwork System Interchange Standard, Dec. 11, 2002. http://www.semi.org/web/wcontent.nsf/url/stds_blueballot
3. F.-L. Heng, P. Gupta, K. Lai, R.L. Gordon and J.-F. Lee, "Taming Pattern and Focus Variation in VLSI design", *Proc. SPIE Conf. on Design and Process Integration for Microelectronic Manufacturing*, Feb. 2004, to appear.
4. G.E. Sery, "Approaching the One Billion Transistor Logic Product: Process Design Challenges", *Proc. SPIE Conf. on Design and Process Integration for Microelectronic Manufacturing*, 2002.
5. A. Wong and R. Ferguson, "Kernel-Based Fast Aerial-Image Computation for a Large-Scale Design of Integrated Circuit Patterns", US Patent 6223139, 2001.