

Modeling OPC Complexity for Design for Manufacturability

Puneet Gupta^a, Andrew B. Kahng^{a,b,c}, Swamy Muddu^c, Sam Nakagawa^a and Chul-Hong Park^c

^aBlaze DFM Inc, Sunnyvale CA, USA 94089

^bCSE Department, UCSD, La Jolla CA, USA 92093

^cECE Department, UCSD, La Jolla CA, USA 92037

ABSTRACT

Increasing design complexity in sub-90nm designs results in increased mask complexity and cost. Resolution enhancement techniques (RET) such as assist feature addition, phase shifting (attenuated PSM) and aggressive optical proximity correction (OPC) help in preserving feature fidelity in silicon but greatly increase mask complexity and cost. Data volume increase with increasing mask complexity is becoming prohibitive for manufacturing. Mask cost is determined by mask write time and mask inspection time, which are directly related to the complexity of features printed on the mask. Aggressive RET increase complexity by adding assist features and by modifying existing features.

Passing design intent to OPC has been identified as a solution for reducing mask complexity and cost in several recent works^{1,2,3}. The goal of *design-aware* OPC is to relax OPC tolerances of layout features to minimize mask cost without sacrificing parametric yield. To convey optimal OPC tolerances for manufacturing, design optimization should drive OPC tolerance optimization using models of mask cost for devices and wires. Design optimization should be aware of impact of OPC correction levels on mask cost and performance of the design. Mask cost characterization (MCC) with different OPC tolerances is a critical step in linking design and manufacturing.

In this paper, we present a MCC methodology that provides models of mask cost of standard cells and wire patterns for use in design optimization. MCC for different tolerance combinations and layout contexts is very CPU intensive. To reduce OPC runtime for characterization, we build a limited set of test patterns, referred to as the library mask, with standard cells and wires patterns that capture all layout contexts in real designs. We then perform OPC followed by fracturing on the library mask. Based on the shot count data from OPC and fracturing of the library mask, we build statistical models of mask cost as a function of OPC tolerances and layout parameters.

Keywords: OPC, Fracturing, Mask cost, DFM

1. INTRODUCTION

Resolution enhancement techniques (RET) such as assist feature insertion and OPC are mandatory post tapeout steps to ensure printability of features in sub-100nm technology nodes. Doubling of layout data volume every technology node combined with aggressive RET are driving mask set costs higher. Transferring design intent to OPC process can reduce the increasing complexity of masks in sub-90nm technology nodes. Design intent-aware OPC applies different levels of OPC correction to different regions of a design based on their criticality.

There are two approaches for minimizing mask cost using design information. In the first approach, timing and power analysis are performed on the design to identify all critical layout features. OPC is then performed with tight tolerances on all critical features and with relaxed tolerances on all non-critical features to minimize mask cost. This approach proposed by Cote et.al¹ does not modify the design flow prior to the tapeout. However, relaxing OPC tolerance uniformly on all non-critical features does not lead to best possible mask cost reductions. In the second approach, tolerance optimization is performed by choosing OPC tolerance combination specific to the standard cell or wire pattern by analyzing its impact on mask cost and design performance simultaneously.² proposes such an approach for minimizing mask cost subject to meeting timing constraints by relaxing OPC tolerances on standard cells. Tolerance optimization is performed prior to the tapeout in this flow.

Characterization of mask cost and timing impact of different OPC tolerances is the basic step for a complete *design-aware* OPC flow. In this work, we characterize mask cost of devices and wires without performing OPC repeatedly with different tolerances. Based on the statistical analysis of shot count of patterns in the library mask, we construct models and lookup tables of mask cost that can be used during OPC tolerance optimization of the design. Specifically, we give models of mask cost with inner tolerance (IT), outer tolerance (OT) and starting side (SSIDE) of feature edges in the layout.

In current chip design and manufacturing flows, only RET teams at foundries perform OPC and fracturing of layout. To allow optimization of OPC tolerances by design teams, we use library mask as a medium to extract shot count data of standard cell and wire configurations from the process.

The main objectives of this work are

1. to present a methodology for construction of library masks by analysis of shot count trends in different layout contexts and
2. to construct a model of shot count of standard cells and wires as a function of OPC tolerance and layout parameters by statistical analysis of shot count data.

This paper is organized as follows. In Section 2 we present MCC methodology for standard cells and wires separately. Since layout topology of standard cells is complicated than that of wires, library mask construction and MCC is different for both. Standard cell library MCC is explained in Section 2.1 and wire MCC is explained in Section 2.2. Section 3 gives experimental results of MCC in the 90nm technology node for TSMC and UMC libraries. Section 4 gives a summary of MCC and presents future directions.

2. MASK COST CHARACTERIZATION

OPC of layout features and hence their shot count depend on layout context. Use of shot count of standard cells or wires in isolated context can lead to incorrect results. On the other hand, it is computationally expensive to perform OPC and fracturing of every standard cell with its layout context for different OPC tolerances during tolerance optimization.

In this work, we perform MCC for model-based OPC (MBOPC) only. MBOPC adjusts edge placement of features in the layout according to tolerance combination within the maximum number of iterations. In addition to tolerance combination, the final fracture count of OPC'ed layout depends on the convergence criterion of OPC algorithm and edge length parameters of fragments. Since OPC algorithm is iterative, modeling edge placements of features and fracture count as a function of number of iterations and fragmentation parameters is very difficult. Particularly, this is more complicated for standard cells than for wires and hence we perform MCC separately for standard cells and wires. To build mask cost models, we assume that shot counts are generated with sign-off OPC recipes and optical models. Unless otherwise mentioned, shot count of a standard cell refers to shot count of polysilicon layer only, since it dominates the cost of entire mask set. In this work, we do not consider assist feature insertion during MCC. Details of standard cell MCC and library mask construction are given in Section 2.1. Details of wire MCC presented in Section 2.2.

2.1. LIBRARY MCC (LMCC)

Shot count of a standard cell is a function of OPC tolerances (IT, OT, SSIDE) and its layout context. In the absence of any layout feature within the optical radius of influence, shot count depends entirely on IT, OT and SSIDE. We refer to the absence of features within the distance of optical radius as isolated context. MCC for isolated context can be performed by running OPC followed by fracturing on individual standard cells for different IT, OT and SSIDE. But in the presence of other standard cells, MCC with IT, OT and SSIDE variation is CPU intensive. In real layouts, standard cells exist in many different layout contexts with other standard cells. Apart from the different types of standard cells surrounding a given cell, the placement of cells within the optical radius also impact the shot count. Running OPC and fracturing on all possible contexts with different spacings between standard cells is practically infeasible. To characterize mask cost of standard cells in

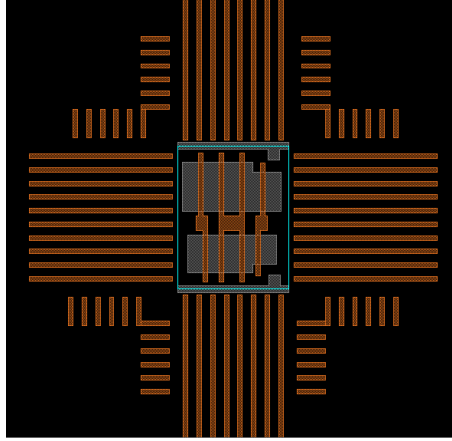


Figure 1. Worst-case “fence-context” surrounding the standard cell has the maximum shot count among all possible contexts.

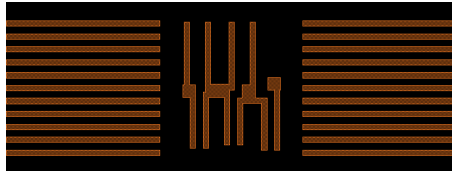


Figure 2. Worst-case left and right “fence-context” to test horizontal isolation of standard cell during OPC. A similar context with top and bottom neighbors is used for testing vertical isolation.

a real layout context, we first identify different layout parameters that impact shot count. We then construct the library mask with artificial contexts of standard cells that vary the identified parameters systematically.

To identify context parameters that impact shot count in a layout context, we first verify the following hypotheses. Each of the following is verified by using an artificial layout context around the standard cell that results in maximum shot count for any given tolerance combination. To create such a *worst-case* context around a standard cell, we place minimum-pitch, minimum-width polysilicon lines on all four sides of the standard cell. To consider the impact of diagonal neighbors we place polysilicon features in all four corners. The typical *worst-case* context for a standard cell is show in Figure 1.

1. Horizontal and vertical isolation: Left neighbors do not impact right side OPC and vice-versa. Similarly, top neighbors do not impact bottom OPC and vice-versa. The impact of OPC is measured in terms of shot count of the standard cell. This hypothesis is illustrated in Figure 2. To verify this hypothesis, we perform the following experiment.

Experiment: To verify horizontal isolation,

- (a) Place the standard cell in the worst-case left context and perform OPC and fracturing for all tolerance combinations.
- (b) Now place the cell in the worst-case right context and perform and OPC and fracturing to measure shot count for all tolerance combinations.
- (c) Place the cell in the worst-case right and left contexts and perform OPC and fracturing.
- (d) Verify that the shot count for any tolerance combination from Step(3) is additive from Step(1) and Step(2).

Results of the experiment are summarized in Table ??.

- Adjacent side independence: Left neighbors do not impact OPC due to top or bottom contexts and vice-versa. Similarly, top neighbors do not impact OPC due to left or right neighbors and vice-versa. This hypothesis is illustrated in Figure 3.



Figure 3. Worst-case left-top and left-bottom “fence-context” to test adjacent side independence of standard cell after OPC.

Experiment: To verify adjacent side independence, we perform an experiment similar to that of horizontal and vertical isolation. Results of the experiment are summarized in Table ??.

- Irrelevance of diagonal neighbors: Diagonal neighbors do not impact OPC. Note that diagonal neighbors may impact OPC of top/bottom or left/right neighbors of the standard cell. However, we only consider the direct impact of diagonal neighbors. This hypothesis is illustrated in Figure 4.

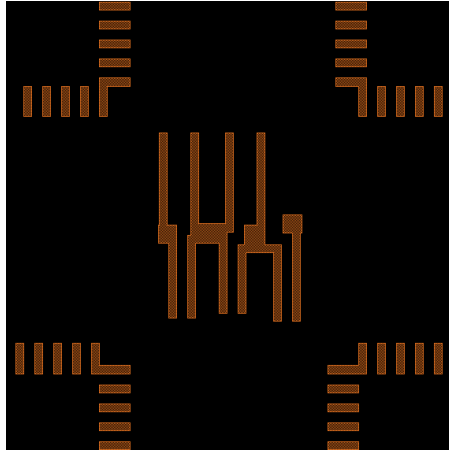


Figure 4. Worst-case diagonal “fence-context” to test impact of diagonal features on standard cell shot count after OPC.

Experiment: To verify irrelevance of diagonal neighbors, we construct a cell context by placing ”L”-shaped polysilicon features at all four diagonal ends and perform OPC followed by fracturing. Results of the experiment are summarized in Table ??.

We performed the above experiments for verifying hypotheses (1) - (3) in 90nm technology using TSMC and UMC standard cell libraries. From the results of the experiments, we find that all the hypotheses hold

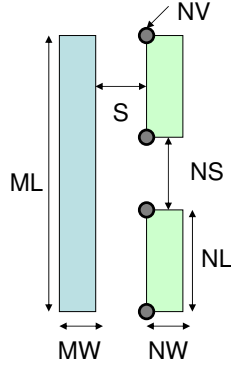


Figure 6: A example of context for WMC glossary.

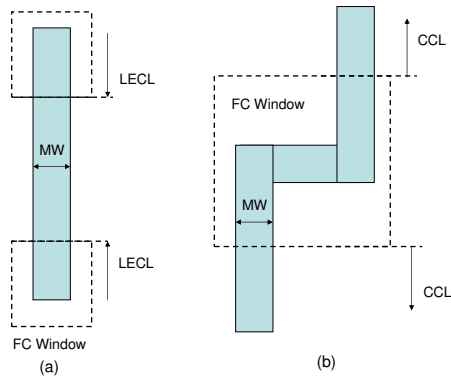


Figure 7: Test structures for (a) line-end characteristic length (LECL) and (b) corner characteristic length (CCL).

Dimension (μm)	0.3	0.65	1.3	1.5
FC for S	101	79	79	79
FC for NS	158	138	124	124
FC for NL	86	93	101	101
FC for LECL	47	59	60	60
FC for CCL	69	71	71	71

Table 2. Saturation points characteristics: $S_0 = 0.65\mu\text{m}$, $NS_0 = 0.13\mu\text{m}$, $NL_0 = 0.13\mu\text{m}$, $LECL_0 = 0.65\mu\text{m}$ and $CCL_0 = 0.65\mu\text{m}$.

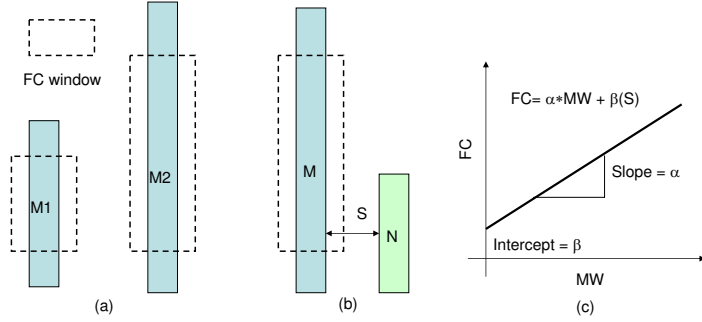


Figure 8. Line-body model characteristics:(a) Test patterns for α calculation, (b) test patterns of β LUTs calculation and (c) line-body model graph ($FC = \alpha ML + \beta(S, EPE_{tol})$).

EPE_{tol}	LUT (Space: nm)			FC of example (a)		FC of example (b)	
	200	400	600	Experiment	Simulation	Experiments	Simulation
0.002	23	10	3	123	124	114	111
0.003	24	12	0	127	126	109	114
0.005	15	6	0	109	108	101	99
0.010	13	6	0	103	104	98	97

Table 3. LUT for line-body model and comparison of FC with simulation and experimental results for two examples: maximum difference of FC is 5.)

$$S_o, LECL_o, CCL_o = OR \quad (1)$$

$$NS_o, NL_o = 2 * OR \quad (2)$$

FC window size, a unit of FC measurement, is determined based on saturation points, i.e., FC window size is larger than S_0 if the main (or primary) pattern has a neighbor. By above equations, one saturation point such as S_0 automatically determine other points so that the number of test pattern is reduced.

Figure 8 shows patterns for line-body model. To calculate $\alpha = [FC(M2) - FC(M1)]/[M2 - M1]$ which represents slope of line-body function, there are first two lines with different lengths as shown in Figure 8 (a). A pair of patterns with main (primary) and neighboring patterns generates β LUT as a function of S and EPE_{tol} *. FC of a main pattern within FC window is predicted as a function of $FC = \alpha ML + \beta(S, EPE_{tol})$. In other words, FC of line-body depends on length of main pattern, spaces between a main pattern and neighbors. To verify the model, we generate α and β LUT and predicts FC of two test patterns as shown in 10. The LUT with EPE_{tol} and comparison of FC with simulation and experimental results for two examples are summarized in Table 3. FC models used in two examples are (a) $FC = \alpha ML + \beta(S1, EPE_{tol})$ and $FC = \alpha ML + \beta(S1, EPE_{tol} + \beta(S2, EPE_{tol}))$, respectively. Our model can exactly predict FC even when the two neighbors are located at left- and right-hand sides of a main pattern. The maximim FC difference is 5.

LUT for line-end model is consisted of two types of test patterns such as single neighbor and double neighbors as shown in Figure 9. Left space in LUT with double neighbors is same as right space. However, if real wire pattern has asymmetric space in left and right neighbors, FC for line-end of pattern uses LUT ($S2$) if $S1 < S2$ because the longer space dominates the total line-end FC. WMC model for corner context generates LUTs with EPE_{tol} of a jog pattern as shown in Figure 7 (b). FC for corner is determined by multiplying the LUT and the number of jog.

* EPE_{tol} is the allowed maximum edge placement error for a given feature.

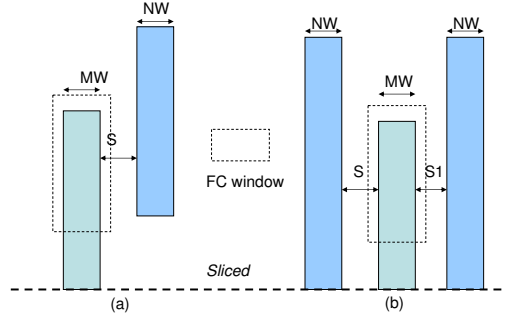


Figure 9. Pattern examples for line-end model: (a) line-end with single neighbor and (b) line-end with double neighbors.

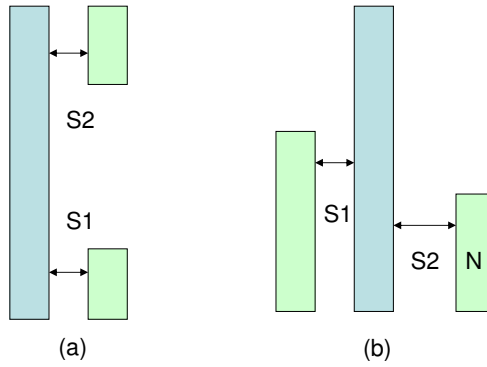


Figure 10. Two examples for validation of line-body model: (a) two neighbors with same space ($S1 = 200nm$) and (b) two neighbors with different spaces of ($S1 = 200nm$ and $S2 = 400nm$)

2.2.2. FC Prediction

A major function of FC predictor divide all patterns of a metal wire into three contexts of L, LE and C. Then FC predictor parses the LUTs of line-body, line-end and corner generated by WMC Model. FC predictor finally estimates total FC of a wire as calculating the sum of FC for each context.

2.2.3. WMC Flow

The total flow of WMC including Model generator and FC Predictor is shown in Figure 11. For each wire, various steps are performed as follows: (1) generate a master test layout that captures the three context of a wire. (2) extract optical radius (OR) using a test pattern. (3) construct fracturing window based on OR. (4) run OPC and fracturing on the master test layout and then record FCs of contexts. (5) fit model parameters of α and β based on FCs and then populate LUTs. (6) FC Predictor decomposes a wire to three sub-sections: line, line-end and corner contexts. (7) parse LUTs generated by WMC model and then estimate FC of the contexts. (8) add FC of each context to obtain the total FC.

3. EXPERIMENTAL RESULTS

Describe experimental flow and present details of libraries, optical models, flow assumptions and tool versions. Present results of fracture count comparison between traditional flow and LMCC/WMCC for different OPC tolerances and demonstrate runtime improvement.

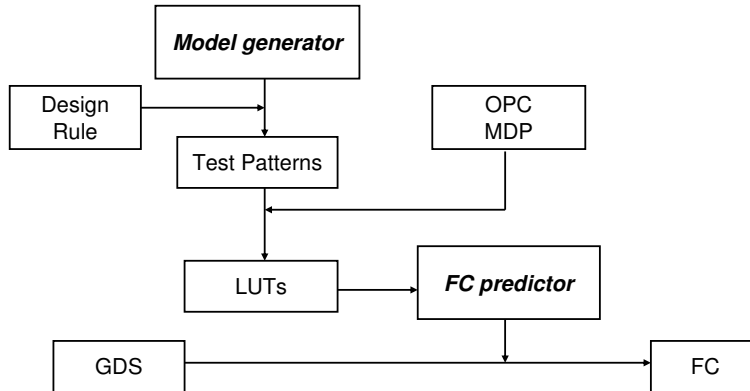


Figure 11: Total flow of WMC.

4. CONCLUSIONS

REFERENCES

1. M. L. Cote, P. Hurat, A. Miloslavsky, D. Goinard and M. L. Rieger, "Mask Cost Reduction and Yield Optimizing Using Design Intent", *Proc. SPIE on Design and Process Integration for Microelectronic Manufacturing*, 2005, pp. 389-396
2. P. Gupta, A. B. Kahng, D. Sylvester and J. Yang, "A Cost-Driven Lithographic Correction Methodology Based on Off-the-Shelf Sizing Tools", *Proc. ACM/IEEE Design Automation Conf.*, 2003, pp. 16-21
3. M. E. Mason, "Rising cost of RETs: Understanding the Value Proposition", *Proc. SPIE on Design and Process Integration for Microelectronic Manufacturing*, 2004, pp.10-19
4. M. L. Rieger, *Mask EDA Workshop*, 2001, http://www.sematech.org/resources/litho/meetings/mask/20010711/G_AVANTI.pdf
5. S. F. Schulze and J. Word, "Interaction of RET and MDP: Optimization for Reducing the Mask Writing Time", *Proc. SPIE on Design and Process Integration for Microelectronic Manufacturing*, 2004, pp. 170-181
6. Y. Zhang, S. Chou, B. Rockwell, G. Xiao, H. H. Kamberian, R. Cottle and C. J. Proglar, "Mask Cost Analysis via Write-time Estimation", *Proc. SPIE on Design and Process Integration for Microelectronic Manufacturing*, 2005, pp. 313-318