ICCAD-2014 CAD Contest in Design for Manufacturability Flow for Advanced Semiconductor Nodes and Benchmark Suite

Rasit O. Topaloglu
IBM
Semiconductor R&D Center, Hopewell Junction, NY
e-mail: rasit@us.ibm.com

***Abstract*—We introduce the fill optimization problem and benchmarks. We provide two new hotspot definitions, slot line deviation and outliers, both of which pertain to yield. We provide the inputs, expected output, as well as objectives and constraints of the problem.**

I. Introduction

Design for manufacturability (DFM) is the science of characterization, modeling, and optimization of yield in semiconductor integrated circuits (ICs). It is a newer field as compared to more established VLSI design automation areas such as logic synthesis or place&route. Due to this, not all design models or flows are optimal yet. Due also to more stringent manufacturing challenges, criteria that has not been important in the past may start becoming important at newer technology nodes.

As circuit optimization requires simultaneous consideration of several circuit aspects, improving yield may end up with race conditions against criteria such as performance, runtime, or layout size. All of these are important in semiconductor IC manufacturing and should be accounted for in a DFM flow. Design steps that are post-route require stringent runtime restrictions. Impact of fills inserted post-route on timing may not be accounted for, and thereby should be minimized. As layout sizes are getting bigger and bigger, uploads to mask houses take longer time and long-term storage for post-manufacturing analysis becomes challenging. Thereby, layout size should also be controlled.

In the proposed problem, we tackle a traditional DFM problem [1], albeit with new metrics and hotspot definitions. In particular, we target fill insertion in VLSI.

II. Background

When a circuit is designed through either manual or automated placement and route of polygons that define devices, interconnects, or other passive devices, densities of these polygons within windows of interest may have a large variation across such windows in the rest of the layout. To balance the density of polygons, fills are inserted to bring the density of several windows to a uniform number which is picked by technology developers to be optimal.

As an input to the problem, participating teams have received layouts with regions identified for which fills can be inserted. Original design layouts are not provided. Teams have also been provided basic design rules that they have to adhere to. Violation of these rules results in their submission to fail.

While the latter restriction seems too strict for an academic competition, the importance of design rule-cleanliness property is paramount in real world. If a design rule error makes its way to the mask, most likely a circuit failure will occur. This issue will bring few months of delay to valuable learning as well as additional costs on the order of hundreds of thousands of dollars for a new mask.

The output is a layout with fills that have been optimally inserted into the regions provided. Insertion of fills outside the regions that are identified results in a submission to fail.

To enable a fair comparison, usage of GPU will not be permitted. However, multiple multicore processors are permitted. In the provided system until the beta submission, we have only measured the runtime and allowed full utilization of the computer resources. In future submissions, we plan to scale runtime to account for the use of number of cores and usage of maximum memory, using a logarithmic scale for the former.

A. Inputs

In this section, we desribe the inputs to the problem.

- Layout in GDS format with regions identified to fill is provided. Testcase s and testcase b have been available during alpha and beta phases of the contest. Additional hidden testcases may be tested in final stage.
- Density and average linewidth statistics for no-fill are given as ASCII data files. In the provided file format, first two columns indicate window location, third column is the no-fill density within that window, and the last column shows the linewidth average in micrometers. These statistics from no-fill option may need to be used towards the solution, however the non-fillable areas in the testcases could have been used to estimate these statistics until these files were provided. $20um$ non-overlapping square windows starting from the lower left corner are used for these computations.

978-1-4799-6278-5/14/$31.00 ©2014 IEEE

- Design rules relevant to the problem are given. Minimum metal width and space are 32nm, minimum metal area is $4800nm^2$. [1] Space rules apply in rectilinear directions.

B. Output

The output is a layout in GDS format with fills that adhere to the provided design rules.

C. Layout Restrictions

We summarize basic restrictions below:

- Fills use the same design rules as other polygons.
- Fills can only be rectangles.
- Fills can abut the fillable region boundaries.
- Lowest numbered layer is at the bottom of the stack.
- Layers are ordered by the layer numbers.
- Layers provided in the layout are neighbors to each other.
- Fills do not have any direction restrictions, they can be horizontal or vertical.

D. Objectives

Objectives for the problem are given below:

- Minimize performance degradation (total area of fill and overlap of fill with neighboring (top and bottom averaged if both exist) layer non-fill regions as well as other fills that are inserted in these layers). Computation repeated for each layer and result is added up. This item accounts for 20% of total score with a breakdown of 5% for total area and 15% for overlap area.[2]
- Minimize variation, i.e., chip density standard deviations. Standard deviations per layer are added up towards the score for this part. This item accounts for 20% of total score.
- Minimize slot line deviation, i.e., variation of density per window from column average.[3]. Variations are added up for all columns and layers. This item accounts for 20% of total score with equal bias for each.
- Minimize outliers, i.e., variation of density per window from 3 sigma corners. Variations beyond 3 sigma corners are added up in absolute terms for each window and layer. Windows with density within 3 sigma of mean do not affect this score. This item accounts for 20% of total score.
- Minimize runtime. Accounts for 15% of total score.
- Minimize layout size (final GDS size). Accounts for 5% of total score.

[1] In actual technology there may be many more rules relevant, however we want to keep the rules simple so that teams could focus on the core problem.

[2] Readers may refer to [2] for improved characterization of fill impact and to [3] and [4] for ways to automate fill insertion while taking performance into account.

[3] This metric is an approximation of measuring deviations after lithographic tools implement an automated fix as described in [5]

III. Evaluation

Only open-source solvers can be used. Standalone tools that may require special licenses may not be used.

Compiled programs are run on a system with four Intel E5-2650 central processing units (CPU) with eight cores each. Each core has a 256KB L2 cache, and each CPU has a 20MB L3 cache. Core speed is 2.6GHz.[4] The processors run at 95W. Memory of the system is 64GB DDR3-1600. Until the beta release number of cores or maximum memory is not penalized, but we plan to monitor and account for these metrics in final submission to adjust the runtime score.

The score is computed based on the submitted solutions. We merge fills with original hidden design layers. Results with correct and design-rule clean outputs are considered for each portion of the score. A formula of $\frac{\beta_X - X}{\beta_X} > 0 ? \frac{\beta_X - X}{\beta_X} : 0$ is used for each X, where X can be any of P (performance), V (variation), R (runtime), S (layout size), L (slot line deviation) or O (outlier). Each term is multiplied by α_X and summed up to get the final score. Negative scores per item are set to 0, i.e., do not degrade the final score. To constitute the final score, all testcases furthermore are given equal weight out of a fraction of 1.[5]

testcase	β_P	β_V	β_R	β_S	β_L	β_O
s	5e4	0.1	50	30e6	2e3	0.35
b	5e6	1	500	300e6	2e6	350

testcase	α_P	α_V	α_R	α_S	α_L	α_O
b,s	0.2	0.2	0.15	0.05	0.2	0.2

IV. Conclusion

Benefits of this contest would be co-optimization of all real-world criteria at once in fill insertion using simplified metrics. Not only performance, i.e., timing, would be considered, but also layout size and runtime. New hotspot defitions that are important towards manufacturability are also considered.

References

[1] Y. Chen, A.B. Kahng, G. Robins, and A. Zelikovsky, "Practical iterated fill synthesis for CMP uniformity," *Proc. ACM/IEEE DAC*, 2000, pp. 671-674.

[2] A.B. Kahng and R.O. Topaloglu, "A DOE set for normalization-based extraction of fill impact on capacitances," *Proc. IEEE International Symposium on Quality Electronic Design*, 2007, pp. 467-474.

[3] R.O. Topaloglu, "Energy-minimization model for fill synthesis." *Proc. IEEE International Symposium on Quality Electronic Design*, 2007, pp. 444-451.

[4] H. Xiang, L. Deng, R. Puri, K.-Y. Chao, and M.D.F. Wong, "Dummy fill density analysis with coupling constraints," *Proc. ACM ISPD*, 2007, pp. 3-10.

[5] B. Liegl, B. Sapp, S. Greco, T.A. Brunner, N. Felix et al. "Predicting substrate-induced focus error," *Journal of Micro/Nanolith. MEMS MOEMS* 9(4), 041311, 2011.

[4] Specifications at intel.com indicate 2GHz with 2.8GHz at turbo boost for this processor model.

[5] Contest website should be visited for any updates to the information in this paper.