Performance-Aware CMP Fill Pattern Optimization

Andrew B. Kahng[1,2] and Rasit Onur Topaloglu[1,3]
University of California San Diego
[1]Computer Science and Engineering Department
[2]Electrical and Computer Engineering Department
[3]Advanced Micro Devices
abk@cs.ucsd.edu, rtopalog@cs.ucsd.edu, rasit.topaloglu@amd.com

Abstract

CMP fills are inserted to make metal density uniform and hence reduce post-polish height variations. Classical methods to insert fills focus on metal density uniformity, but do not take into consideration or are unable to minimize the impact of fills on circuit performance. In this paper, we develop a fill insertion method that heuristically minimizes coupling capacitance increase due to fill. Our optimization methodology builds on fill insertion guidelines previously developed in, e.g., [12] and [1]. Experiments show that the proposed optimization methods can reduce fill impact on coupling capacitances by up to 85% for 30% pattern density and up to 65% for 60% pattern density cases.

I. Introduction

CAD tools insert metal fills between interconnects to comply with metal density requirements induced by the chemical-mechanical polishing process during manufacturing. Maximal insertion of regularly-patterned floating fills is known to be inefficient from the standpoint of both pattern density variation control and post-fill electrical performance. Previous works establish guidelines for fill insertion that can reduce capacitive coupling [12] and show the importance of constraining fill impact on capacitance of timing-critical signal nets [11]. Nevertheless, traditional fill synthesis methods still use simplistic (e.g., Boolean) pattern operations and fail to provide necessary automation of such complex fill optimization guidelines. In this paper, we propose a novel energy-minimization model for fill synthesis, through which complex fill guidelines can be implemented.

In our fill synthesis framework, the possible locations for fill insertion have allocated energies. When a fill is to be inserted, the location with lowest energy is sought. Fills are inserted into a given region one by one, until a prescribed density constraint is satisfied. The energy network is designed so as to enable insertion according to a design guideline. In this framework, the design of the energy network becomes an important task, essentially migrating the fill synthesis problem from circuit design to the CAD algorithm domain. We propose energy models and give insights to ease the design of such energy networks.

The remainder of this paper is organized as follows. After reviewing previous work in this area, we describe our fill optimization methodology in Section III, where we also present the heuristics used in our method. Section IV gives experimental results comparing our fill synthesis results with traditional methods, according to static timing analysis and density histograms. The paper concludes in Section V.

II. Previous Work

Stine et al. [3] have targeted metal fill optimization to reduce dielectric thickness variations. Lee et al. [4] have analyzed the effects of certain parameters on parasitics. Recently, extraction of capacitance values for layouts which include floating fills have been important. Floating fills are preferable over grounded fills since they have less (albeit more variable) coupling and timing impact, do not interfere with power and ground network design, and yield more opportunity for performance optimization. Park et al. [5] and Cueto et al. [6] have developed a field solver which can take into account floating fills. Tugbawa et al. [15] have provided CMP models and a calibration methodology. Lee et al. [7] have proposed design guidelines for metal fill insertion and also proposed an RC extraction methodology. Kurokawa et al. [8] have presented a method for extraction in the presence of fills by mapping fill sizes to effective dielectric properties (thickness, permittivity). Kahng et al. [9] have proposed a window-based solution to the metal fill problem such that a density bound is preserved and variability across windows is reduced during metal fill insertion. Chen et al. [11] have introduced a performance-driven fill insertion method, either to minimize total delay impact or to maximize minimum slack. Kahng et al. [16] have proposed slotting algorithms in addition to filling. Xiang et al. have proposed a slot-based coupling-aware fill insertion method in [18]. Although mentioned as the first such algorithm for coupling awareness, this work was subsequent to [1]. Furthermore, no timing impact and density variation analyses are provided. Xiang et al. [19] have presented a layout density analysis method.

A number of fill layout guidelines have been suggested for designers in [12]. These guidelines include centralization of fill locations with respect to neighboring interconnects, positioning of fills toward edges of available layout regions, maximization of the number of fill columns and minimization of the number of fill rows (between two vertically-oriented interconnects). More detailed fill impact analysis, including inter-layer impacts, were provided in [2]. A hour-glass fill pattern for optimal intra-layer coupling was identified in [1]. However, known fill synthesis algorithms cannot effectively apply these rules, and manual application is impractical. Hence, new fill synthesis methods are still required.

A force-directed scheme to optimize fill shapes and locations was first proposed in [1]. Our present work is simpler (e.g., without an "atomistic" physical analogy), and validated through parasitic extraction and timing analysis of standard-cell designs. Force-directed schemes have been proposed for many years in the realm of VLSI standard-cell and module placement; see, e.g., [14] for an overview. In the VLSI placement context, the circuit netlist provides connectivity information: energy between cells is a function of separation in the layout, and inter-cell distances can be used in the energy minimization. By contrast, in the CMP fill placement problem, no such connectivity is inherently present. Rather, given a layout of signal interconnects, fills must be inserted according to a guideline, with each floating fill shape considered independently.[1]

Traditional Fill Insertion. Traditional fill insertion does not take into consideration the coupling between interconnects, despite a number of "timing-driven fill" or "intelligent fill" approaches having been proposed over the years [17]. Typically, a window-based "fixed-dissection" scheme is used, where overlapping windows span an entire layout, and density constraints must be observed in each window. Linear programming frameworks can be used to determine target amounts of fill to be added into each portion of each window, so that overall layout density is as uniform as possible [10], [17]. The actual fill pattern is then synthesized to realize these target amounts using, e.g., arrayed or staggered rectangular shapes; fill shapes and pattern may also be selected to help satisfy a given target density in a window. More naive flows simply insert as many fills as possible, as long as fill dimension and spacing constraints are satisfied. Such "maximum-fill" solutions give reasonably uniform post-fill layout density, but often at a much higher level than necessary; fill impact on timing is also higher than necessary.

III. Fill Optimization Methodology

We now describe our fill optimization methodology, covering the steps of region definition, grid definition, energy modeling within a grid, and fill insertion within a grid.

A. Adaptive Region Definition

Given a metal layer, we use a version of the scan-line algorithm to determine regions between facing interconnects with no other interconnect in between. A *region* consists of a maximal rectangular area such

[1] In the VLSI high-level synthesis literature, Paulin et al. [13] proposed a heuristic model using force-directed energy minimization for scheduling. Our method also targets energy minimization, but as one would expect the models are significantly different.

that two interconnects directly face each other. Each such region is converted into a *grid* after keep-off distances are stripped.[2] We illustrate region definition with four interconnects in Figure 1. For each region, we form a grid of rectangles into which fills will be placed. To obtain a uniform post-fill metal density while also considering performance impact of fill, a uniform target density can be heuristically used for each region. Our region definition is adaptive in the sense of following the specifics of the interconnect design, in contrast to methods that simply dissect the layout into uniform square regions.

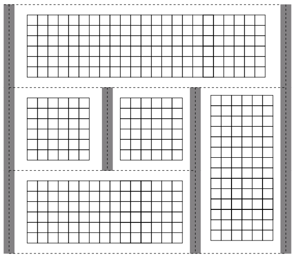


Fig. 1. Five regions are indicated by dashed lines and the grids of rectangles are shown.

B. Converting a Region to a Grid

In converting the region between two interconnects into a grid, we can define possible grid rectangles for fill insertion. An example is shown in Figure 2**(a)**, where dark lines are the interconnects and square boxes are grid rectangles into which metal fills can be placed.

To obtain a grid, we define an *auxiliary frame* at the keep-off distance away from any interconnects, and at half the keep-off distance away from edges of the region which do not overlap with any interconnect. The auxiliary frame forms the outer edge of a grid. The grid holds multiple rectangles in which fills may be inserted. The grid rectangles and the frame are connected together with *bonds* having adjustable energy values between them, as illustrated in Figure 2 **(b)**.

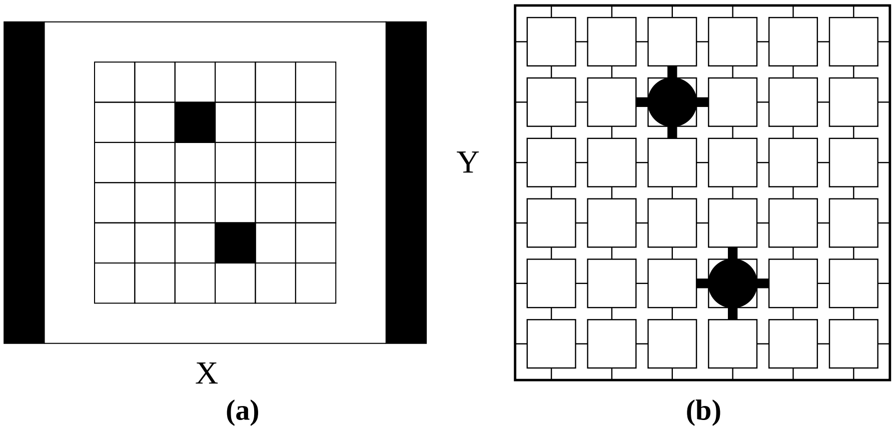


Fig. 2. **(a)** A region between two interconnects (dark thick lines) shown. A grid is formulated after keep-off distances are stripped out. There are 36 rectangles in this grid. Two fills are inserted. **(b)** An auxiliary frame (dark lines) holding grid rectangles (squares) with bonds (light lines) in between. Two fills are shown along with the incident bonds used for energy calculation (larger lines).

To convert target metal density to the number of fills to be inserted, we first compute the density in the grid.

$$D = (W \cdot A_r - A_i)/A_r \qquad (1)$$

[2] A keep-off distance is a design rule which states the minimum distance of a fill to the nearest interconnect.

Here, $0 \leq W \leq 1$ is the target *feature* density assigned to a region. Half the width of each interconnect belongs to the region, as shown by the dashed lines in Figure 1. A_r and A_i are the area of the region and total area of interconnects in the given region, respectively. $0 \leq D \leq 1$ is the target *fill* density within the *grid*, which consists of a region with interconnects and keep-off distances stripped. The fill area selection is described below.

C. Energy Modeling in a Grid

We have developed a parameterized model to choose bond values according to given guidelines. In the following, vertical and horizontal bonds refer to bonds in the vertical and horizontal directions, respectively, in Figure 2 **(b)**. According to the guidelines we have considered, fills should be (i) close to the ends of interconnects, and (ii) centralized with respect to interconnects in a manner reminiscent of a hourglass shape. Using 3D field solver experiments beyond those of [12], we have observed that these guidelines improve capacitance compared with other pattern guidelines. Hence, we have designed our models such that vertical bonds start from 0 around the middle of the interconnects and decrease as we approach the ends; energy minimization favors fill insertion closer to the ends of the grid in the vertical direction. We have also designed our models such that horizontal bonds start from 0 around the middle of the interconnects and increase as we approach the interconnects; energy minimization favors insertion closer to the center of the grid in the horizontal direction. For the vertical bonds, the bond energy along the *y*-direction is given as

$$\gamma + (|(i_{mid}) - i| * \zeta/(i_{mid})) \tag{2}$$

where i is the row number and real number i_{mid} is the number of rows divided by 2. For the horizontal bonds, energy along the *x*-direction is

$$\alpha + (|(j_{mid}) - j| * \beta/(j_{mid})) \tag{3}$$

where j is the column number and j_{mid} is the number of columns divided by 2. α, β, γ and ζ are fitting parameters. The use of separate vertical and horizontal energy models allows flexibility in capturing different fill guidelines.

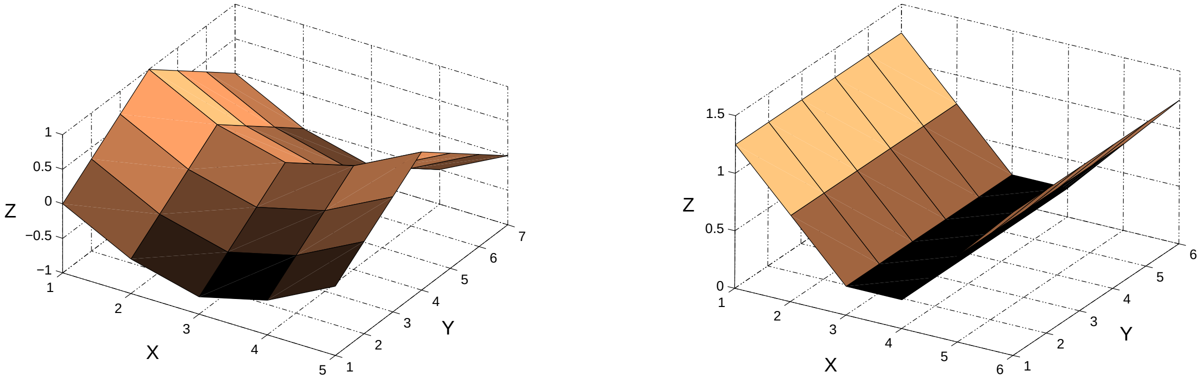
Figures 3 **(a)** and **(b)** show the energies assigned to bonds in the vertical and horizontal directions, respectively for an example assignment. X and Y labels correspond to the same labels as in Figure 2. The Z axis gives the bond values. As noted above, vertical bonds tend to receive higher energy values close to the interconnects and towards the center, as compared to at interconnect endpoints. Horizontal bond energies are simply designed to decrease toward the middle of two interconnects, in order to pull more fills into this region.

D. Insertion of Fills into a Grid

We have mapped the fill insertion problem to an energy-minimization problem. Fills are placed in available locations to minimize an energy criterion. In Figure 2 **(a)**, the rectangles are fixed in the grid, and hence the locations of bonds are fixed. Each bond is assigned an energy value, and we iteratively find the grid rectangle for fill insertion with minimum sum of energies of the incident bonds.

We have implemented metal fill optimization using greedy optimization. We write our fill insertion algorithm as:

```
Initialize $t_{min} = INF$ $e_{min} = INF$
do
  do
    $t \leftarrow t_{next}$
    $e = \sum b_t$
    if ($e \leq e_{min}$)
      Update $e_{min}$ and $t_{min}$
```



(a) **(b)**

Fig. 3. Plots of energies assigned to bonds; the Z axis shows the energy value of a bond at the corresponding location. **(a)** Energies assigned to vertical bonds. **(b)** Energies assigned to horizontal bonds.

```
[7]       until all $t \in G$ are evaluated
[8]       Place fill at $t_{min}$
[9]  until all fills are inserted
```

The input to the algorithm consists of a grid and the number of fills to be inserted. The output is a grid with the fills placed. In the algorithm, t is the current grid rectangle location, t_{next} is the next grid rectangle, e is energy, b_t is the set of bonds incident to grid rectangle t, e_{min} is minimum stored energy, t_{min} is the grid rectangle location with minimum energy, and G is the grid itself. Lines 1 to 5 search for a grid rectangle with minimum energy, and the fill is inserted in Line 8. This algorithm is run for all regions identified by the scanline algorithm in a given metal layer. Hence, overall runtime is dependent on the number of interconnects (initial scanline-based processing) as well as on the total grid area.

Grid and Fill Size Selection. Assuming square fills, we compute the fill size, spacing and grid rectangle size as follows.

$$FW = (1+\varepsilon)\sqrt{D} \tag{4}$$

$$FS = \sqrt{1-D} \tag{5}$$

In Equation (4), FW is a parameter correlated to the fill width. $0 < D < 1$ is the target fill density. $0 < \varepsilon < 1$ is an adjustment parameter which helps select a fill size that will leave empty space in a region to enable movement of fills for improved optimization.

From Equation (5), FS is proportional to the fill spacing. FW and FS are scaled up by 50% at a time while the fill width to fill spacing ratio is preserved, until the design rules for a given layer (i.e., fill width, spacing, and area rules) are satisfied. For square fill size in the traditional fill flow, we set the fill width over spacing proportional to $\sqrt{D}/(1-\sqrt{D})$, and set ε equal to 0. We did not necessarily use 50% for the traditional case, but we provide the final sizes that we have used in Table III.

For rectangular fills, both the proportionality constant and the fill aspect ratio are considered, and legality is enforced with selection of the fill spacing.

The grid rectangle size is selected as $1.5^n * (FW+FS)$, where n is the number of times FW and FS are scaled up. The number of fills to be inserted per region is then computed as $D * A_r$ divided by the area of the grid rectangle.

TABLE I

ANALYSIS FLOW

1. Place, synthesize clock network and route design.
2. Using GDS from Step 1, insert fills into GDS.
3. Use Perl scripts to obtain DEF containing fills.
4. Extract .spef parasitics from DEF containing fills.
5. Run static timing analysis using .spef file from Step 4.
6. Use Calibre scripts to obtain density histograms.
7. Use Perl scripts to obtain slack histograms.

TABLE II

DESIGN RULES

	Min Width	Min Spacing	MinDensity	Max Density	Min Area
M2-6	0.1μm	0.1μm	0.15	1.0	0.04μm^2
M7-8	0.4μm	0.4μm	0.20	1.0	0.056μm^2

IV. EXPERIMENTAL RESULTS

In the following, we describe our experimental setup and results.

A. Experimental Setup and Protocol

Our experiments validate the efficiency and relevance of the proposed fill synthesis method by comparing layouts filled with the proposed method against those filled with a traditional fill method. We have implemented the proposed method using approximately 4000 lines of C++ code. Input is a GDSII (Graphic Data Stream) file converted into OpenAccess format, and output is the fill-optimized GDSII and DEF (Design Exchange Format).

In our flow, we use *Cadence SOC Encounter v5.2* for placement, clock tree synthesis and routing, and *Synopsys Star RCXT 2006.06* to extract post-fill parasitics. We compare layouts filled with the proposed scheme versus a *Calibre 2007.1_34*-implemented traditional fill scheme.[3] We then run the *Synopsys Primetime 2006.12* static timing analysis tool and compare the slack distributions reported by timing analysis. Our C++ code uses a subset of the *OpenAccess 2.2.4* API.

Our analysis flow is given in Table I. For the optimized case, Step 2 uses our code. We use Calibre scripts for traditional fill insertion. We bypass Step 2 for the no-fill case. The $.spef$ file on Line 4 is in Standard Parasitic Exchange File format, and the DEF file is in Design Exchange File format.

B. Discussion of Results

We have used TSMC 65nm GPlus 8-layer metal technology in our experiments. Table II provides the relevant design rules. We have applied our tool to the ISCAS89 s38417 benchmark circuit using a 2GHz computer. This benchmark contains 8 metal layers, 8,514 gates, and 43,076 interconnect stubs, i.e. pieces of

[3] It is important to note that henceforth, while we use "Calibre" to denote the traditional fill pattern, this reflects only on the merits of the traditional strategy. The *Calibre* tool itself is a very powerful physical verification platform that is a standard vehicle for post-layout design tasks, including the insertion of metal fill.

TABLE III

FILL SIZES FOR TRADITIONAL FILL

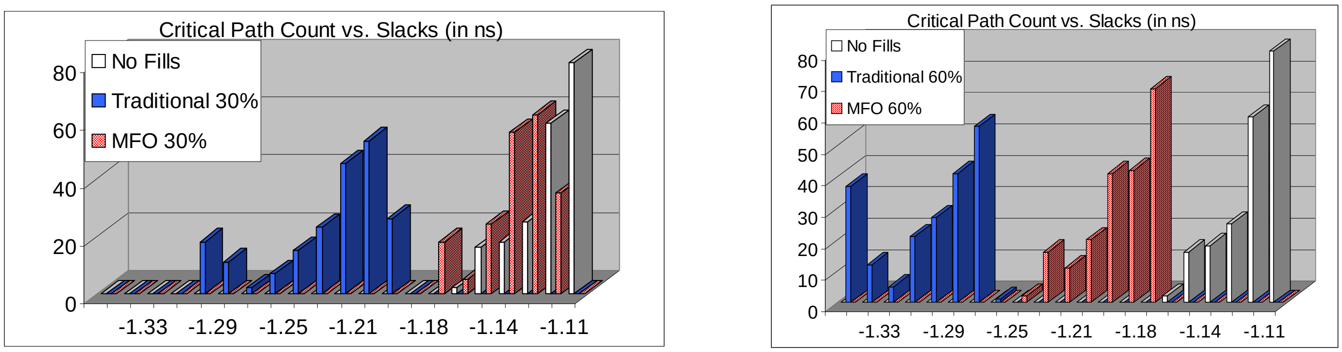
	Fill Width 30%	Fill Spacing 30%	Fill Width 60%	Fill Spacing 60%
M2-5	0.24μm	0.2μm	0.67μm	0.2μm
M6-8	0.48μm	0.4μm	1.34μm	0.4μm

interconnects between gates, ports and flip-flops which may be on different metal layers. Our program uses up to 160 MB of RAM for the *s*38417 benchmark. We have applied fill optimization for this benchmark (layers M2 through M8) and compared it with fills inserted traditionally and with no-fill conditions.

Comparison. To facilitate an equal comparison, we use a fixed target density. For the traditional fills, we have adjusted the fill size and spacing such that a given density such as 30% or 60% will be attained. We set the parameter ε defined in Section III to 0.5, as we have seen that this value gives a reasonable tradeoff between final density uniformity and timing optimality. A smaller value of ε (closer to 0) reduces the number of available spaces, restricting the movement of fills during optimization. A larger value of ε increases the fill size, which may increase coupling through the fills and degrade timing, but at the same time, reduces the inserted number of fills. We use square fills in our analysis, although our tool can place rectangular and even more complex fill shapes.

For traditional fills inserted using Calibre, we have used the parameters given in Table III. These parameters are optimized to provide a given density, i.e., 30% or 60% in our experiments. For the bond network energies, we use $\alpha = 0$, $\beta = 1$, $\gamma = 0$ and $\zeta = -1$.

We note that keep-off distance is the same as the fill spacing in our experiments. It is possible to see in actual designs that keep-off distances can be up to 5 times larger than minimum fill spacing. Although originally introduced to reduce intra-layer coupling, such a design rule may in fact result in an increase of the inter-layer coupling components, since intra-layer flux is no longer pulled away by the intra-layer floating fills but by the inter-layer fills.



(a)

(b)

Fig. 4. Timing slacks for s38417 benchmark for no-fill, traditional and optimized fill cases. **(a)** 30% density case. **(b)** 60% density case.

Figure 4 **(a)** shows the worst 200 critical path setup slacks for *s*38417 benchmark in the three cases of no-fill, traditional fill, and optimized fill patterns. *Slack* is a parameter which gives the difference between the actual signal arrival time at a flip-flop input or primary output, and the corresponding required arrival time. A large negative slack indicates that timing is not met by a particular critical net, and that hence either the clock period must be increased (i.e., frequency decreased), the clock skew between sequentially adjacent flip-flops must be adjusted, or the combinational cells on the critical timing path must be sped up. We observe from Figure 4 **(a)** that metal fill pattern optimization using our tool *MFO (Metal Fill Optimizer)* performs almost as well as the no-fill condition. *MFO* results outperform the traditional fill results by 0.1*ns* through consideration of interconnect coupling during fill insertion. We have observed that hold timing slacks were similar in all three cases.

To show that a lower target density has not caused the superior results for *MFO*, we have also repeated the experiments with a 60% target density. The results are shown in Figure 4 **(b)**. Although the optimized case has deviated from the no-fill case, it still is superior as compared to traditional fills.

Again to ensure fairness of comparisons, we have conducted an additional set of experiments, summarized

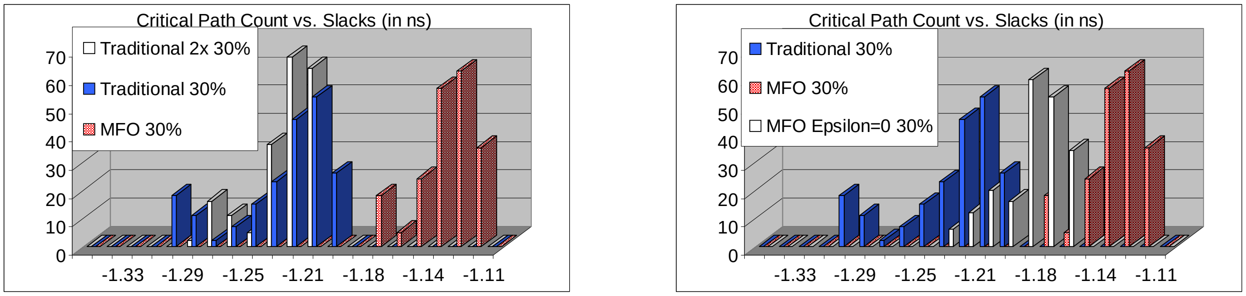
TABLE IV

Number of Fills Inserted

MFO 30%	Traditional 30%	Traditional 2x 30%	MFO 60%	Traditional 60%
952,487	555,466	111,251	160,505	209,495

in Figure 5. Slack data for *MFO* and traditional fills inserted with Calibre are replicated from Figure 4 **(a)**. To determine whether the number of fills make a strong difference, we have increased both the fill width and fill spacing by a factor of 2, and repeated the experiment. This increase effectively yields the same target density of 30%, but results in a smaller number of fills. Again, we use Calibre to generate traditional fills. This data is labeled as 2*x* in Figure 5 **(a)**. The fill numbers of the experiments are given in Table IV. MFO 30% creates more fill shapes than traditional 30%, because the MFO-selected fill width and spacing are smaller, which can be observed in Figure 7. Comparing traditional versus traditional 2x results, the latter has somewhat fewer than one-fourth the number of fills as the former. This is because when fill width and spacing are doubled, certain empty areas can no longer accept any fill shape (i.e, spacings smaller than fill width *plus* twice the fill spacing). Finally, the comparison of MFO versus traditional fill for the 60% case indicates that the former can insert fewer fill shapes than the latter, again because of the computed fill width and spacing. As shown in Figure 5 **(a)**, fill number reduction through fill size and width increase minimally reduces the slacks even though the fill number decreases significantly.

The improvement over Calibre fills comes at the expense of increased density variability, though a standard deviation of 3% as can be observed from Figure 6, is not highly significant. This variation is likely due to the use of scanline algorithm and a region-based filling instead of a traditional window-based filling. The former restricts fill insertion between short facing interconnect stubs, but makes energy network modeling much easier. Choices of ε for 0.5 and 0 cases are shown in Figure 5 **(b)**.



(a) **(b)**

Fig. 5. **(a)** Timing slacks for *s*38417 benchmark for 30% traditional, 30% traditional with large widths (2x) and 30% optimized fill cases. **(b)** Comparison of the impact of *epsilon*.

Density Uniformity. To evaluate density uniformity, we compare the density histograms between proposed and traditional fill patterns. The density distributions are shown in Figures 6 **(a)** and **(b)** for the *M*5 layer for optimized and traditional cases, respectively. 20*µm* windows stepped by 10 *µm* are used for the density analysis. The total area of the layout is 175 by 169*µm*. Both fills achieve a mean close to 30% for the case when the target density is 30%. MFO-optimized fill results in 28.87% density with 2.55% standard deviation. While the traditional fill shows a mean of 29.54% and a tighter density standard deviation of 0.002%, the MFO-optimized fill still achieves a reasonable density range. Greater density variability is observed when fill sizes are increased to 2x for the traditional fill approach: standard deviation of density increases to 2.13%, while the density mean decreases to 26.85% from 29.54%. We note that density variations are typically defined by the design rather than by fills. For example, scribe lines in between the chips, memory modules, on-chip decoupling capacitances and analog circuitry next to digital circuitry can create much larger density variations. Hence, for a fill insertion algorithm, coupling minimization may be a higher-priority objective than zeroing out the density variability.

Snapshots of the GDS layouts we have evaluated for *MFO* and traditional fill patterns are shown in Figures 7 **(a)** and **(b)**, respectively.

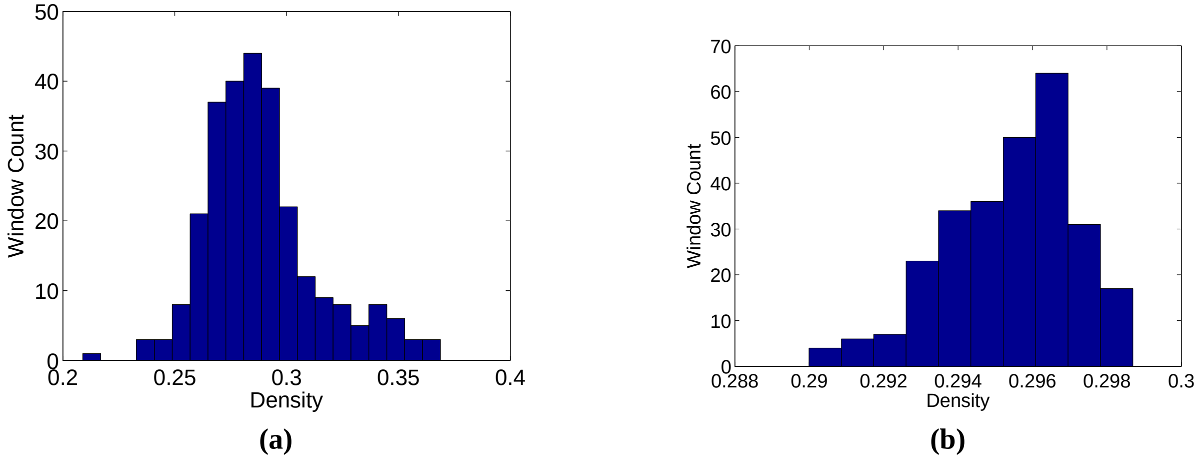


Fig. 6. Density histograms for M5 for benchmark *s*38417. **(a)** *MFO* fill. Although somewhat suboptimal with respect to traditional fill, density variations on layout are most of the time defined by design rather than fill patterns.**(b)** Traditional fill.

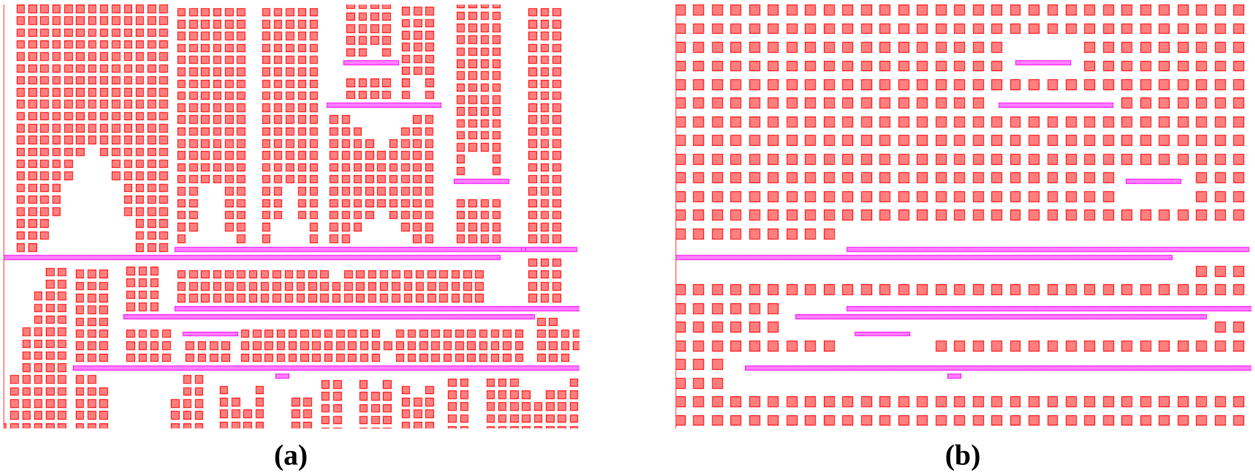


Fig. 7. Snapshot of the 30% optimized fills in the s38417 benchmark layer M5. Window width is 10 μm. **(a)** *MFO* fill. Fill widths selected large so that resulting empty space can be utilized for optimization. **(b)** Traditional fill. Fill sizes selected are the minimum acceptable by design rules to facilitate a uniform density.

V. Conclusions

In this work, we have introduced a fill synthesis framework that heuristically accepts and realizes complex, performance-driven CMP fill guidelines. Our heuristic approach uses an energy minimization framework to achieve metal fill insertion in a given region. We have presented a parameterized bond-energy model, along with insights to guide modeling when guidelines change due to newer technology or design styles. With the proposed framework, complex guidelines or traditional fill insertions can be implemented efficiently. Experimental results with the proposed optimization methods show that with 65nm testcases, it is possible to reduce the fill impact on coupling capacitances by up to 85% for 30% pattern density, and by up to 65% for 60% pattern density.

REFERENCES

[1] R. O. Topaloglu, “Energy-Minimization Model for Fill Synthesis”, *Proc. ISQED*, 2007, pp. 444-451.

[2] A.B. Kahng and R.O. Topaloglu, ”A DOE Set for Normalization-Based Extraction of Fill Impact on Capacitances”, Proc. ISQED, 2007, pp. 467-474.

[3] B.E. Stine et al., “The Physical and Electrical Effects of Metal-Fill Patterning Practices for Oxide Chemical-Mechanical Polishing Processes”, *IEEE Trans. Electron Devices*, 45(3), 1998, pp. 665-679, 1998.

[4] W.-S. Lee et al., “Investigation of the capacitance deviation due to metal-fills and the effective interconnect geometry modeling”, *Proc. ISQED*, 2003, pp. 373-376.

[5] J.-K. Park, K.-H. Lee, J.-H. Lee, Y.-K. Park and J.-T. Kong, “An Exhaustive Method for Characterizing the Interconnect Capacitance Considering the Floating Dummy-Fills by Employing an Efficient Field Solving Algorithm”, *Proc. SISPAD*, 2002, pp. 98-101.

[6] O. Cueto, F. Charlet and A. Farcy, “An Efficient Algorithm for 3D Interconnect Capacitance Extraction Considering Floating Conductors”, *Proc. SISPAD*, 2002, pp. 107-110.

[7] K.-H. Lee et al., “Analyzing the Effects of floating Dummy-Fills: From Feature Scale Analysis to Full-Chip RC Extraction”, *Proc. IEDM*, 2001, pp. 685-688.

[8] A. Kurokawa, T. Kanamoto, A. Kasebe, Y. Inoue and H. Masuda, “Efficient Capacitance Extraction Method for Interconnects With Dummy Fills”, *Proc. CICC*, 2004, pp. 485-488.

[9] A. B. Kahng, G. Robins, A. Singh, H. Wang and A. Zelikovsky, “Filling Algorithms and Analyses for Layout Density Control”, *IEEE Trans. on CAD* 18(4), 1999, pp. 445-462.

[10] Y. Chen, A. B. Kahng, G. Robins and A. Zelikovsky, “Area Fill Synthesis for Uniform Layout Density”, *IEEE Trans. on CAD* 21(10), 2002, pp. 1132-1147.

[11] Y. Chen, P. Gupta and A. B. Kahng, “Performance-Impact Limited Area Fill Synthesis”, *Proc. Design Automation Conf.*, 2003, pp. 22-27.

[12] A. B. Kahng, K. Samadi and P. Sharma, “Study of Floating Fill Impact on Interconnect Capacitance”, *Proc. ISQED*, 2006, pp. 691-696.

[13] P. G. Paulin and J. P. Knight, “Force-Directed Scheduling for the Behavioral Synthesis of ASICs”, *IEEE Transactions on Computer-Aided Design* 8(6), 1989, pp. 661-678.

[14] T. Chan, J. Cong and K. Sze, “Multilevel Generalized Force-Directed Method for Circuit Placement”, *Proc. ISPD*, 2005, pp. 185-192.

[15] T. Tugbawa et al., “A Mathematical Model of Pattern Dependencies in Cu CMP Processes”, *Proc. Electrochemical Society Meeting*, Oct. 1999.

[16] A. B. Kahng, G. Robins, A. Singh, H. Wang and A. Zelikovsky, “Filling and Slotting: Analysis and Algorithms”, *Proc. ACM/IEEE Intl. Symp. on Physical Design*, 1998, pp. 95-102.

[17] A. B. Kahng and K. Samadi, “CMP Fill Synthesis: A Survey of Recent Studies,” to appear in *IEEE Trans. on CAD*, 2007.

[18] H. Xiang, L. Deng, R. Puri, K.-Y. Chao and M. D. F. Wong, “Dummy Fill Density Analysis with Coupling Constraints,” *Proc. ACM/IEEE International Symposium on Physical Design*, 2007, pp. 3-9.

[19] H. Xiang, K.-Y. Chao, R. Puri and M. D. F. Wong, “Is Your Layout Density Verification Exact? - A Fast Exact Algorithm For Density Calculation,” *Proc. ACM/IEEE International Symposium on Physical Design*, 2007, pp. 19-26.