DRE: A Framework for Early Co-Evaluation of Design Rules, Technology Choices, and Layout Methodologies

Rani S. Ghaida, *Student Member,* IEEE, and Puneet Gupta, *Member,* IEEE

***Abstract*—Design rules have been the primary contract between technology developers and designers and are likely to remain so to preserve abstractions and productivity. While current approaches for defining design rules are largely unsystematic and empirical in nature, this paper offers a novel framework for early and systematic evaluation of design rules and layout styles in terms of major layout characteristics of area, manufacturability, and variability. The framework essentially creates a virtual standard-cell library and performs the evaluation based on the virtual layouts. Due to the focus on the exploration of rules at an early stage of technology development, we use first order models of variability and manufacturability (instead of relying on accurate simulation) and layout topology/congestion-based area estimates (instead of explicit and slow layout generation). Such a framework can be used to co-evaluate and co-optimize design rules, patterning technologies, layout methodologies, and library architectures.**

I. Introduction

The semiconductor industry is likely to see several radical changes in the fabrication and device technologies during this decade. On the patterning front, disruptive changes include the adoption of one or more of candidate next-generation lithography techniques such as nanoimprint, electron beam direct write, and extreme ultraviolet [1–4]. Each of these has challenging implications for layout methodologies and design rules (DRs). Resolution enhancement techniques (RETs) and other patterning solutions such as immersion and double-patterning technology (DPT), off-axis illumination (OAI), sub-resolution assist features (SRAFs), and phase-shift mask (PSM) require additional layout-restrictive DRs [5–11]. Therefore, *early assessment of design restrictions imposed by technological choices is absolutely essential.*

DRs are the biggest design-relevant quality metric for a technology. Even small changes in DRs can have significant impact on manufacturability [12] as well as circuit characteristics including layout area, variability, power, and performance [13, 14]. Unfortunately, even after DRs have existed for decades, design rule evaluation and exploration is largely unsystematic and empirical in nature. Several published works have done empirical "one-at-a-time" evaluation of design rules [12, 15]. For example, the work in [16] electrically evaluates line-end extension rule and conclude that it may be too conservative. Other recent works [17, 18] offer solutions to explore DRs from a pure printability perspective and do not examine the effects of DRs on circuit characteristics. Moreover, none of these methods account for layout topology changes that may happen when the DR values change significantly. They also ignore several practical constraints imposed on layouts by the standard-cell design methodology (e.g., cell width and height quantization). Finally, these approaches are based on explicit layout generation and lithography simulation, which makes them slow and dependent on the models accuracy.

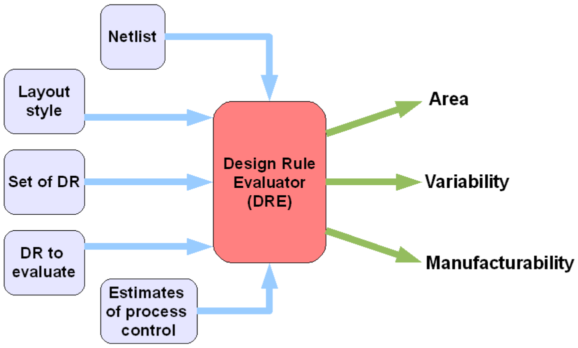
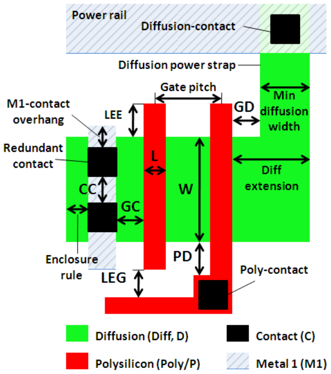


Figure 1. Overview-diagram of DRE framework.

In this paper, we extend our work presented in [19]. To the best of our knowledge, this work proposes the first framework to systematically and qualitatively explore area-manufacturability-variability tradeoffs in design rules. Rather than fine-tuning DRs, our goal is to make early decisions *before* exact process and design technologies are known. At this stage, accurate evaluation methods and models are unlikely to be available and the return on investment of using them is fairly low. Unlike other approaches that rely on layout generation or perturbation (e.g., [20, 21]), we use simple but justified approximations for manufacturability and variability. Because the search space of DRs is very large, we use fast layout topology generation methods to estimate area as opposed to full-blown layout generation. The accuracy of the former is surprisingly good and allows for explicit "layout style" guidelines, as we show later in this paper.

The structure of the proposed cell-level DR Evaluator (DRE) is depicted in Figure 1. The framework takes the following inputs: circuit netlists (e.g., SPICE) of cells (possibly scaled down from a previous technology generation), layout style and preferences (e.g., redundant contacts), DRs and their values (see Figure 2), estimates of process control (e.g., overlay error distribution), and benchmark designs (specified as cell usage statistics) to evaluate the rules on. In DRE, only the values of DRs to be evaluated are modified while all other rules remain unchanged. This modified set of DRs is then used to estimate the layout and determine major metrics of area, manufacturability, and variability[1].

[1] DR choices also affect delay, power, reliability, and designability. Evaluating these aspects of DRs is part of our future work.



DR	Value[nm]
L	≥ 50
M1 1/2 pitch	≥ 65
Gate pitch	≥ 190
Min diff width	≥ 90
LEE	≥ 55
LEG	≥ 75
GC	≥ 35
CC	≥ 75
GD, PD	≥ 50
M1-C overhang	≥ 35
Diff enclosure	≥ 5
Poly enclosure	≥ 5
Diff extension	≥ 70

Figure 2. Illustration of major DRs, their notations and values in FreePDK 45nm process [22].

We make the following contributions.

- We offer a framework for *fast*, *early* and *systematic* collective evaluation and exploration of DRs, layout styles, and library architectures. The framework makes DR generation and optimization easier and much faster. Rather than exploring the entire search space of DRs with conventional compute-expensive methods, the framework can be used to quickly eliminate poor DR choices.
- We evaluate some major DRs and layout style decisions such as: 1D and 2D poly, multiple and fixed-pitch poly, diffusion and metal 1 (M1) power-straps, and cell height.
- We demonstrate through case studies the use of the framework to explore DRs and compare processes from the design perspective.

The remaining paper is organized as follows. Section II describes the methods used for layout topology generation as well as metal-congestion estimation and its impact on the layout area. Sections III and IV provide details on the models and metrics used for manufacturability and variability. In Section V, comparative evaluations of several DRs are performed in a 45nm process. In addition, we analyze area-manufacturability-variability tradeoffs of a commercial standard and a low power 65nm process and illustrate the use of our framework for the collective exploration of DRs. Finally, Section VI summarizes our findings and presents directions of future research.

II. Area Estimation

The number of design rules is growing tremendously and design rule manuals (DRM) are becoming unmanageable as we move toward smaller feature sizes [23, 24]. In addition, DRs need to be evaluated individually as well as collectively over a wide range of values. As a result, our framework was designed for the *fast* evaluation necessary to enable DR exploration/optimization.

This section describes the methods used in the DRE framework for the fast layout topology generation and metal-congestion estimation.

A. Layout Topology Generation

Major transistor placement techniques used for layout-area reduction are highlighted in Figure 3. Transistor pairing consists of placing two inter-connected transistors, one pMOS and another nMOS, on the same column to minimize wire length and facilitate routing as well as to ensure more layout regularity. The coupled pMOS/nMOS transistors are referred to as transistor pairs. Transistor folding consists of replacing a large transistor by equivalent multiple transistors of smaller sizes connected in parallel. Transistor chaining is the process of abutting transistors of the same type by sharing the same diffusion area. Non-isolated transistors of the same active region form a transistor chain. A transistor stack refers to two transistors sharing a diffusion area that is not connected to any other parts of the circuit (i.e. contact-free diffusion).

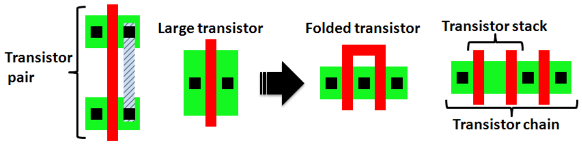


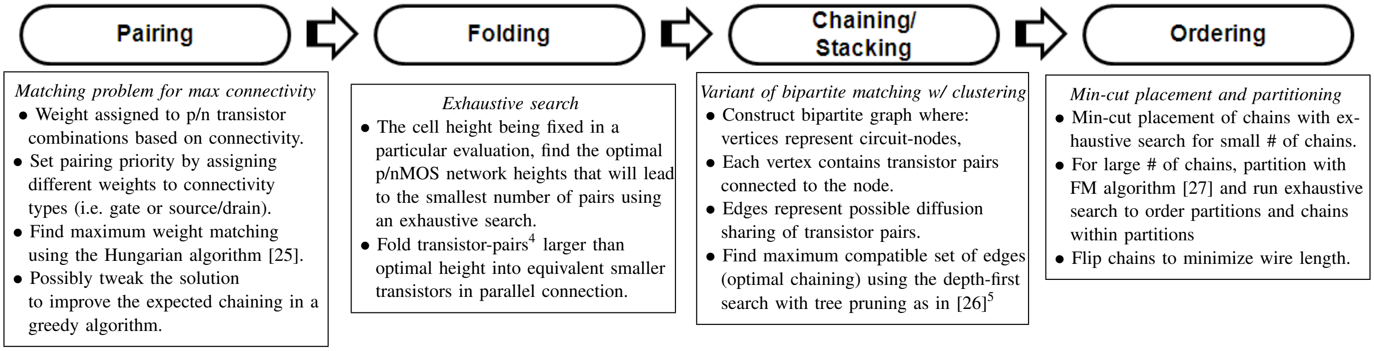
Figure 3. Techniques and notations used in layout topology generation.

Figure 4 outlines the flow of transistor placement used in our layout estimation and describes the algorithms used at each step. We illustrate the application of these steps on a standard-cell in Figure 5.

The first step is transistor pairing. A score is assigned to each pMOS/nMOS transistor combination based on the connectivity and the pairing problem is reduced to finding the matching with the maximum score. According to the layout style, different scores can be associated with different types of connections (i.e. gate or source/drain) to set the pairing priorities. Sharing of the gate signal is typically preferred over source/drain signals to save on contacts and the congestion they induce. This matching problem is solved in DRE optimally using the Hungarian algorithm [25].

Transistor folding is performed next. A transistor with width larger than its network (pMOS or nMOS) height must be folded into multiple transistors in parallel connection with the same total width. Therefore, the ratio of the pMOS network height to nMOS network height affects the total number of pairs after transistors are folded. Because the cell height is fixed and layout dimensions are quantized to the manufacturing grid size, there is a limited number of the possible pMOS/nMOS network-height ratio. So, we determine the total number of pairs associated with each ratio; the ratio leading to the minimum number of pairs is set for each cell. After the pMOS/nMOS network heights are decided, wide transistors that exceed the height of the corresponding network are actually folded.

The layout topology generation continues with the step of transistor chaining. The fast method discussed in [26] is implemented to perform the chaining. In this method, the cell circuit is represented as a bipartite graph. Vertices represent nodes in the circuit and each vertex contains all transistor pairs connected to its corresponding node. Edges represent possible abutments of transistor pairs. A depth-first search with tree pruning is used to find the maximum compatible set of edges, which corresponds to the optimal chaining. Solutions with the higher upper bound on the number of realizable abutments are examined first and we found that the optimal solution (i.e. the same chaining as in actual layouts) is reached in almost every case after examining the first few solutions.



Pairing

Matching problem for max connectivity

- Weight assigned to p/n transistor combinations based on connectivity.
- Set pairing priority by assigning different weights to connectivity types (i.e. gate or source/drain).
- Find maximum weight matching using the Hungarian algorithm [25].
- Possibly tweak the solution to improve the expected chaining in a greedy algorithm.

Folding

Exhaustive search

- The cell height being fixed in a particular evaluation, find the optimal p/nMOS network heights that will lead to the smallest number of pairs using an exhaustive search.
- Fold transistor-pairs[4] larger than optimal height into equivalent smaller transistors in parallel connection.

Chaining/ Stacking

Variant of bipartite matching w/ clustering

- Construct bipartite graph where: vertices represent circuit-nodes,
- Each vertex contains transistor pairs connected to the node.
- Edges represent possible diffusion sharing of transistor pairs.
- Find maximum compatible set of edges (optimal chaining) using the depth-first search with tree pruning as in [26][5]

Ordering

Min-cut placement and partitioning

- Min-cut placement of chains with exhaustive search for small # of chains.
- For large # of chains, partition with FM algorithm [27] and run exhaustive search to order partitions and chains within partitions
- Flip chains to minimize wire length.

Figure 4. Flow of layout topology generation in the DRE framework.

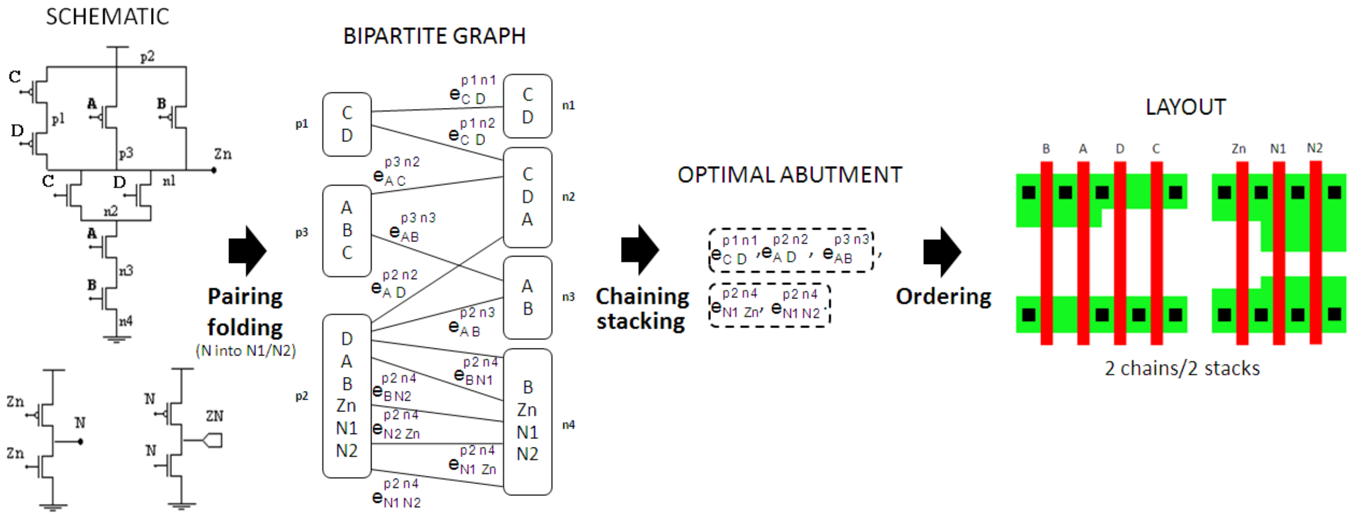


Figure 5. Example that illustrates our layout topology generation for a 4-input OAI standard-cell.

Thus, we have limited the number of iterations[2] to make the algorithm run faster. Folds of the same transistor are treated as independent transistors and, in some cases, might end up abutted to different transistors and separate from each other to improve the chaining solution. When transistors are folded into large number of folds however, this practice no longer improves the chaining solution and makes the algorithm run much slower. As a result, we cluster large number of folds belonging to the same transistor into groups that we treat as single transistors during chaining. Transistor stacking is considered a special type of chaining. Stacks have an advantage over regular chaining in that they do not need a contact and, consequently, might improve the layout density in some process technologies[3]. Therefore, if multiple chaining solutions have the maximum number of abutments, we pick the one with the maximum number of stacks.

The ordering of transistors within chains is inferred from the abutments associated with the chaining solution that was picked. Chains are then ordered linearly in a row following the familiar 1D placement. The problem is formulated as a *min-cut placement* to minimize the overall wire length. In case the number of chains is small, we run exhaustive search to find the optimal solution; otherwise, we partition the graph of chains using the Fiduccia-Mattheyses (FM) algorithm [27] and run exhaustive search to first find the optimal order of partitions and, then, the optimal order of chains within each partition. Once the ordering is complete, chains are possibly flipped to reduce the overall wire length further.

The exact transistor and pin locations along the horizontal direction are then determined based on minimum DR dimensions. As for transistor locations along the vertical direction, we consider three possibilities: (a) as near as possible to power rails, (b) exactly in the center of p/n networks, and (c) as near as possible to p/n interface. The choice of vertical location of transistors is regarded as a layout style, which can also be evaluated by the DRE framework[4].

B. Tweaking Pairing

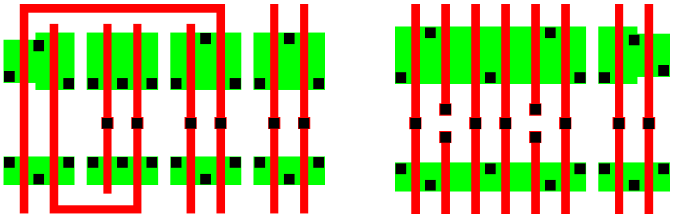
The pairing step results in pMOS/nMOS pairs with the largest number of shared signals and preference to the sharing of the gate signal. In practice, layout designers may introduce small tweaks on pairing to improve the chaining solution (i.e. reducing the front-end area) as shown in the example of Figure 6. Therefore, we introduce an additional step just after the first pairing to perform such tweaks automatically.

Tweaking of the pairing solution is performed using a greedy algorithm. Given the initial pairing solution, we precompute for each pair the number of connections with the

[2] Twenty eight iterations for cells with more than 20 transistors and six hundred iterations for smaller cells.

[3] The minimum gate pitch is typically smaller than the contacted gate pitch unless a fixed-pitch poly style is adopted.

[4] This decision has implications on M1-congestion as well as the impact of stress and well-proximity effect on performance [13, 14].



(a): Perfect pairing, area=3.192 (b): Imperfect pairing, area=2.66

Figure 6. Example illustrating imperfect pairing and its associated tradeoffs for a DLH_X2 cell layout.

other pairs that can be performed in both nMOS and pMOS sides, i.e. the number of possible abutments with the other pairs. We then check if the switching of the transistors of any combination of pairs can improve the number of possible abutments. The switch with the best improvement is performed and the involved pairs are prevented from future switching. This process is repeated until all switches with improvement are performed or until all pairs have been switched.

The downside of imperfect pairing is that it requires more spacing between the pMOS and nMOS transistors than in the case of perfect pairing (see Figure 6). This extra spacing requirement can result in higher number of folds in some cases. Therefore, we determine, based on the target pMOS to nMOS transistor height ratio of the library, the expected number of folds before and after a transistor switch is made. The switch is prevented if it is expected to cause a larger number of folds. It is worth noting that the extra spacing requirement for imperfect pairing reduces the available wiring tracks in the top and bottom channels of the Poly (or horizontal local interconnect) layer as shown in Figure 6.

C. Routing Estimation

Once transistor placement is complete, locations of gates and contacts to the gates and transistor source/drain (S/D) terminals are determined. S/D contacts connected to power supply are located as close as possible to the power rail without violating DRs. All other S/D contacts are located near the p/n interface to reduce the length of wires necessary to connect transistors from the nMOS network to transistors from the pMOS network. Contacts to gates (poly contacts) are placed at the p/n interface of the cell (y-coordinate) and the same horizontal locations (x-coordinate) of the gates that they connect to.

Rather than performing actual routing, we estimate the routes and model metal congestion with the goal of considering its effect on layout area. Estimating routes is preferred over performing actual routes for three reasons. First, different automated tools and layout designers can reach completely different routes and a small change in the DRs may result in very different routing solutions. On the other hand, estimated routing is generic, meant to assess the quality of rules, and is not affected by small DR changes. Second, performing actual routing is very time consuming and can be a runtime bottleneck for our automated evaluation. Third, introducing new rules or layout styles may require a significant reimplementation of a router; this problem is much less severe for smart congestion estimation. Hereafter, the term “routing” denotes “estimated routing” and not actual routing.

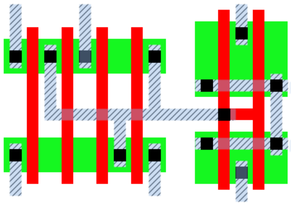


Figure 7. S/D-to-gate interconnections may be routed on M1 or poly layers and S/D-to-S/D interconnections may be routed on M1 or M2 layers. We assume a single-trunk Steiner tree for routing.

Transistor interconnections, i.e. intra-cell routes, are assumed to be performed using polysilicon (poly), diffusion for power connections (i.e. power straps) if dictated by the layout methodology, the first metal (M1) layer, and possibly the second metal layer (M2) if accessible for cell design. There are three types of connections: gate-to-gate, S/D-to-gate, and S/D-to-S/D. The way gate-to-gate connections are performed depends on poly-routing restrictions, which are characterized by the layout style.

Three configurations of poly-routing are allowed: no poly-routing (1D), limited poly-routing, and unrestricted poly-routing (2D). In case no poly-routing is allowed, poly is used only to connect dual gates (i.e. gates of same transistor-pair). Connections between any other gates need to be performed with metal layers. In case poly-routing is limited, poly is used to connect adjacent gates in the same network (pMOS or nMOS) in addition to dual gates. In case poly-routing is not restricted, all gate interconnections are performed on the poly layer unless routing is infeasible due to congestion or blocking active layer. Since routing resources are limited, we give priority for routing longer nets to maximize poly utilization. Horizontal wiring on poly uses the available tracks of p/n routing channels at the top and bottom of the cell with the exception of wiring used to connect adjacent gates of folded transistors (a.k.a. fingers), which are assumed to occupy the routing-channel at the p/n interface in the center of the cell. Excluding finger interconnection, there are three cases for gate-to-gate routing involving horizontal wiring not to be possible on the poly layer. The first case is when diffusion power-straps block both the top and bottom routing channels. The second case is when the access to the gate from the top and bottom channels is blocked by nets previously routed on poly. The last case of infeasible poly routing occurs if at any location between source and destination along both top and bottom channels all horizontal poly tracks are occupied.

Hypernets involving S/D-to-gate connections are decomposed into a single metal segment and one or more poly segments depending on poly-routing restrictions. In this case, a single poly-contact is added per poly segment and it is placed along the y-coordinate of the p/n interface at the same x-coordinate of the nearest gate it connects to (see the poly contact in the example of Figure 7.

Metal segments of S/D-to-gate and S/D-to-S/D connections are made with metal layers. If all pins have the same x or y-coordinates, the route is performed using a vertical or horizontal wire connecting all pins. For nets involving pins at different x or y-coordinates, we assume they can be routed using a single-trunk Steiner tree as shown in Figure 7. Single-trunk Steiner tree routing is common in real layouts and we

Table I
SHAPE COUNT AND LOCATIONS ASSUMPTIONS FOR NETS THAT CANNOT BE CONNECTED WITH A STRAIGHT LINE.

Shape	**Count**	**Location**
Tip	# of pins	Fixed at pin locations with bounding box including the tip in its all possible orientations
L-shape	two	Anywhere in net's bounding box
T-shape	# of pins minus two minus # of crosses	Anywhere in net's bounding box
Cross	Special case of T-shape, detected based on x coordinates of pins	Anywhere in net's bounding box
Line	One horizontal and one vertical if space permits	Anywhere in net's bounding box

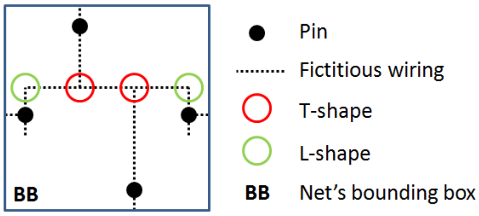


Figure 8. Illustrating example showing how the shape count is determined for a 4-pin net based on the assumption of single-trunk Steiner tree routing.

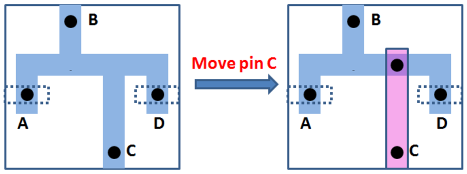


Figure 9. Illustrating example showing the move of a pin from M1 to M2 to resolve congestion on M1.

avoid fixing the trunk to an exact location to keep the routing estimation generic. With this assumption, we can determine what shapes are involved in each route based on the number of pins. The shape count is summarized in Table I and an illustration example is shown in Figure 8. The wire length is estimated as the half-perimeter of the bounding box.

There are three configurations for metal layer assignment:

1) 2D M1 with prohibited access to M2 layer for intra-cell routing;
2) 2D M1 with use of M2 to resolve M1 congestion;
3) and 1D M1 in one direction and 1D M2 in the orthogonal direction.

In case (1), when M1 is congested, the cell-area is increased to accommodate all the wiring. In case (2), certain segments are assigned to M2 to resolve M1 congestion as illustrated in Figure 9 and the cell-area is increased only if M1 remains congested after all the available space on M2 is exploited. The number of segments and the utilization of M2 are minimized during the segment assignment to M2. This minimization is done while meeting the maximum utilization allowed on M2 and discounting any segment assignment that introduces congestion in the orthogonal direction. The algorithm used for the layer assignment of segments is described in Figure 10.

```
Parameters definitions
    $\Delta C^*$: needed congestion reduction
    $\Delta C_{orth,max}$: allowed increase of congestion in orthogonal direction without
        an area increase
    $L_{M2,max}$: maximum allowed total wire length on M2
    $\Delta C_i$: congestion reduction by moving segment $i$
    $L_{M2,i}$: length of the segment when moved to M2
    $\Delta C_{orth,i}$: induced change in congestion when segment $i$ is moved to M2
    $L_{M2}$: current total wire length on M2
    $\Delta C_{orth}$: current change of congestion in orthogonal direction induced so far
Algorithm
    Determine $\Delta C^*$
    Determine $\Delta C_{orth,max}$
    Determine $L_{M2,max}$
    for all segment $i$ sorted from largest $\Delta C_i$ to smallest do
        Determine $L_{M2,i}$
        Determine $\Delta C_{orth,i}$
        if $L_{M2}$ + $L_{M2,i}$ > $L_{M2,max}$
            OR $\Delta C_{orth}$ + $\Delta C_{orth,i}$ > $\Delta C_{orth,max}$ then
            Skip segment $i$
        end if
        Move segment $i$ to M2
        if $\Delta C$ + $\Delta C_i$ >= $\Delta C^*$ then
            exit loop
        else
            $\Delta C \leftarrow \Delta C + \Delta C_i$
            $L_{M2} \leftarrow L_{M2} + L_{M2,i}$
            $\Delta C_{orth} \leftarrow \Delta C_{orth} + \Delta C_{orth,i}$
        end if
    end for
```

Figure 10. Overview of the algorithm used to determine the segments that need to be moved from M1 to M2 to resolve congestion on M1.

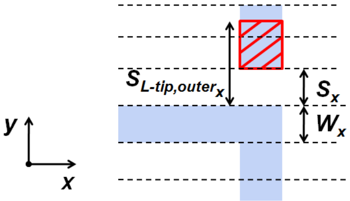


Figure 11. Example illustrating blockage model for an instance of L-shape with a single tip facing its outer corner.

D. Congestion Estimation

Once all routes are estimated, we calculate M1/M2 wire length in x and y directions including via/contact-landing pads for the cell I/O pins. Occupied track-length (OTL) in a particular routing direction is then determined as the sum of wire length and blocked track-length from different patterns as well as wires in the orthogonal direction. Specifically, OTL in y direction is calculated as follows (similar expression for x direction):

$$OTL_y = WL_y + \sum Block_y + \sum_{seg}\left(\left\lceil \frac{WL_{seg,x} + Block_y}{w_y + s_y} \right\rceil - IS_{seg}\right) \times (w_x + s_x). \tag{1}$$

The parameters of Equation 1 are defined in Table II. IS is included to prevent counting blockage for actual intersections that form corner connections between vertical and horizontal wires. $Block_{x,y}$ models the extra spacing requirement of rules that exceed the minimum spacing (e.g., tip-to-tip). We estimate the number of occurrences of patterns that invoke each of these rules and each occurrence contributes to $Block_{x,y}$ factor by the required spacing minus the minimum spacing. For an illustrating example, consider the pattern of Figure 11, which consists of an instance of L-shape with a single tip facing its outer corner. For this pattern, $Block_y$ is the L-shape to tip spacing, $S_{L-tip,outer_x}$, minus the minimum spacing, S_x. The pattern crosses three tracks and has a single intersection. Therefore, the term in the first parenthesis of the second summation of Equation 1 evaluates to 2 for this pattern. Estimating the number of occurrences and determining $\sum Block_{x,y}$ are

Table II
PARAMETER DEFINITION FOR EQUATION 1.

Symbol	Description
OTL_y	Occupied track-length in y direction
WL_y	Total wire length in y direction
$WL_{seg,x}$	Wire length of a segment in x direction
$Block_y$	Blockage in y direction due to spacing rules that exceed the minimum spacing (e.g., tip to tip)
w_x	Minimum wire-width in x direction
w_y	Minimum wire-width in y direction
s_x	Minimum line-to-line spacing rule in x direction
s_y	Minimum line-to-line spacing rule in y direction
IS_{seg}	Number of actual intersections in a particular segment (i.e. number of L/T-shapes and crosses)

Table III
SPACING RULES CONSIDERED IN THE DRE FRAMEWORK FOR BOTH X AND Y DIRECTIONS.

Tip-to-tip min spacing
Tip-to-line min spacing
L-shape to *outer* tip min spacing
L-shape to *inner* tip min spacing
L-shape to line min spacing
T-shape to *outer* tip min spacing
T-shape to *inner* tip min spacing
T-shape to line min spacing
Cross to tip min spacing rule

performed using the algorithm in Figure 12 and a summary of the spacing rules considered in the DRE framework is given in Table III.

Track congestion in one direction is defined as the ratio of occupied to available track-length (i.e., number of tracks times length of the track).

E. Area Increase Due to Congestion

In case congestion (denoted by C) exceeds a certain threshold, the cell-area is increased or M2 layer is used to accommodate all the wiring. This threshold depends on the intra-cell routing efficiency and empty space required on M1 to access the cell I/O pins. Furthermore, routing efficiency is a function of the proportion of non-preferred direction wire length to the total wire length. If wires are mostly in one direction, routing is efficient and increasing the cell-area is only necessary for very high congestion. In contrast, if wires are evenly distributed in the two directions, routing is difficult and increasing cell-area is expected for relatively low M1-congestion. To capture these effects, we model track-congestion threshold as follows:

$$C_{threshold} = \alpha + \left| \frac{U_x - U_y}{U_x + U_y} \right| \times \beta, \tag{2}$$

where U_x and U_y are the track utilization in x and y directions. Here, track utilization is defined as the ratio of the occupied track length *without consideration for track blockage from the orthogonal direction wiring* (i.e. Equation (1) with $WL_{seg,x} = 0$ for all segments), to the available track-length. α and β parameters, with typical values of 0.6 and 0.2 respectively, are a function of intra-cell routing efficiency. The values of all these parameters are set by the user based on the router specifications.

Figure 13 depicts one method to extract α and β parameters either from trial routes of few cells or from cells of a previous generation library. Every single cell implementation adds lower and upper bound lines that narrow down the feasible solution space. Therefore, the more cells are used, the more precise the solution is. If a cell is not congested, we can add one upper bound line derived from $C_{threshold} < 1$ and one lower bound line derived from $C_{threshold} > C$ (by plugging in Equation (2) in both cases). If a cell is congested, we can only add one upper bound line that is derived from $C_{threshold} < C$. In the end, exact values of α and β can be approximated by the coordinates of the feasible region's geometric centroid.

```
Construct list of fixed shapes constituting of tips, power supply wires, input metal
pins, and straight-line connections
Construct list of non-fixed shapes as the complementary of the list of fixed shapes
$\sum Block_{x,y} \leftarrow 0$
{Step 1: check all fixed shapes in the cell against each others}
for all combination of two fixed shapes that are not processed yet do
    if $DR^*$ involving the two shapes is violated then
        $\sum Block_{x,y}$ = $\sum Block_{x,y}$ + $DR^*$ - $S_{x,y}$
        Mark both involved shapes as processed
    end if
end for
{Step 2: check all fixed shapes in the cell against non-fixed shapes}
for all non-processed fixed shapes do
    for all Nets do
        if Fixed shape interacts with the net's bounding box then
            Find worst $DR^*$ that is violated
            $\sum Block_{x,y}$ = $\sum Block_{x,y}$ + $DR^*$ - $S_{x,y}$
            Mark both involved shapes as processed
        end if
    end for
end for
{Step 3: check all non-fixed shapes in the cell against each others}
for all Nets do
    for all Nets do
        if Bounding boxes of both nets interact then
            Find worst $DR^*$ that is violated
            $\sum Block_{x,y}$ = $\sum Block_{x,y}$ + $DR^*$ - $S_{x,y}$
            Mark both involved shapes as processed
        end if
    end for
end for
```

Figure 12. Overview of the algorithm used to determine blockage from rules that exceed the minimum spacing ($\sum Block_{x,y}$).

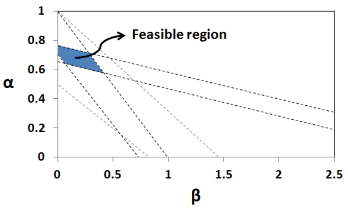
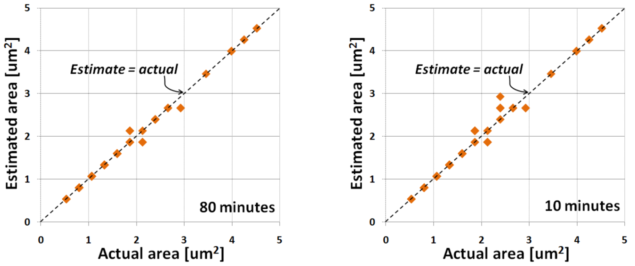


Figure 13. Illustrating example for extraction of α and β parameters of Equation 2 from M1 congestion data.

Another method to extract α and β is through an automated control loop that runs the DRE framework for a bunch of cells from a previous generation (or cells with trial routes) and fine-tune α and β until the estimated cell-area is very close to the actual cell-area.

F. Runtime and Validation of Area Estimation

In order to validate our layout estimation method and its efficiency, we use the DRE framework to estimate the topology of the entire Nangate 45nm Open Cell Library [28] (96 cells) and estimate cell-area. The comparison between the estimated and the actual areas is depicted in Figure 14(a). The results show very good accuracy of the layout estimation method; for 89 out of 96 cells, the estimated areas match exactly with the actual areas and, only for 7 cells, the estimated areas are off by a single poly pitch. This corresponds to an absolute error of less than 1% on average. DRE area estimation has also



(a) (b)

Figure 14. DRE estimated cell area versus actual cell area of the Nangate Open Cell Library [28] (96 standard cells) with a runtime of 80 minutes and average absolute error less than 1% (a) as well as a runtime of 10 minutes and average absolute error of 2% (b).

been validated in [29] by comparison with actual layouts of a commercial 32nm standard-cell library.

The runtime of the evaluation procedure for the entire cell-library is roughly 80 minutes in real time on a single processor of 2GHz clock speed and 2MB cache. This runtime can be reduced to 10 minutes by sacrificing a fraction of the quality of the layout estimation (average absolute error increases to 2%) as depicted in Figure 14(b)[5]. In the experiments of this work, we use the DRE setup with the better accuracy.

III. Manufacturability

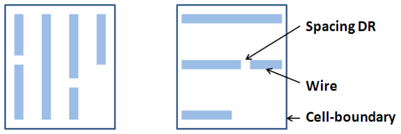
Our manufacturability index for evaluating DRs is the functional yield from three sources of failure[6]:

1) overlay error (i.e. misalignment between layers) coupled with lithographic line-end shortening (a.k.a. pull-back);
2) contact-hole failure;
3) random particle defects.

Hence, the overall yield is given by

$$Y = Y_{overlay} \times Y_{contacts} \times Y_{particles}. \tag{3}$$

The yield from overlay, $Y_{overlay}$, is equal to the probability of survival (POS) from the overlay error coupled with the lithographic line-end shortening. Overlay vector components in x and y directions are described by a normal distribution with zero mean and process-specific 3σ estimate. We compute POS from overlay causing: failure to connect between contact and poly/M1/diffusion, gate-to-contact short defect, and always-on device caused by poly-to-diffusion overlay error. Connection failure at contacts occurs when the area of overlap with top/bottom connecting layers is smaller than a certain threshold-value. Thus, we consider overlay in both x and y directions in this analysis. In gate-related failure analysis, overlay in just one direction is considered since gates are presumably unidirectional. Moreover, we assume all layers are aligned to a reference alignment mark on substrate[7] and overlay between different layers and the reference layer to be independent[8]. The overall POS from overlay is then calculated as the product of POS from independent overlay errors. If overlay is assumed to be completely a die-to-die variation, then POS of the die is p (equal to POS of the most overlay-critical spot in the layout). On the other extreme, if overlay is completely random within-die variation, then POS of the die is p^n, where n is the total number of critical spots in the design. Reality is closer to the former situation (since field and wafer level components dominate intra-field components [30]), which is our assumption in this paper.



(a) (b)

Figure 15. Virtual artwork representation for (a) horizontal and (b) vertical M1 wires.

Because contact-hole failure is a random process, we model $Y_{contacts}$ using the Poisson model (as in [31]). The average number of contact defects (λ) is equal to the number of non-redundant contacts in the layout (N_c) times contact-hole failure rate (D_f). In case contact-redundancy is implemented, duplicated contacts are assumed to always yield since the probability for *two* contacts connected to the *same* pin to fail is negligible. Thus,

$$Y_{contacts} = e^{-\lambda} = e^{D_f \times N_c}. \tag{4}$$

To capture failure caused by random particles, we perform critical area analysis for open and short defects at M1/poly/contact layers and short defects between gates and diffusion-contacts. For fast analysis, we use the virtual artwork approach proposed in [32]. Poly and contact layers are represented by strips separated by spacing-DRs; whereas for the M1 layer, this separation corresponds to the spacing that makes the wires as far apart as possible (see example of Figure 15). The virtual artwork representation allows quick calculation of critical area as a function of defect size by applying a closed-form model. The average critical area (A_c) for all defect sizes is then determined for each layer while using the following defect size distribution model [33, 34]:

$$f_s(r) = \begin{cases} \frac{2(n-1)r}{(n+1)r_0^2} & \text{if } 0 \leq r \leq r_0, \\ \frac{2(n-1)r_0^{n-1}}{(n+1)r^n} & \text{if } r > r_0. \end{cases} \tag{5}$$

where r is the defect size, r_0 is the defect size with peak density (a.k.a. critical defect size), and n is a parameter related to the cleanliness of the fabrication process and ranges between 2 and 4. Finally, $Y_{particles}$ is calculated using the widely adopted negative binomial model [35] as follows:

$$Y_{particles} = \prod_{l=1}^{L} Y_{particles,l} \tag{6}$$

$$Y_{particles,l} = \prod_{j=1}^{k} \left(1 + \frac{A_{c,j} \times D_0}{\alpha}\right)^{-\alpha}, \tag{7}$$

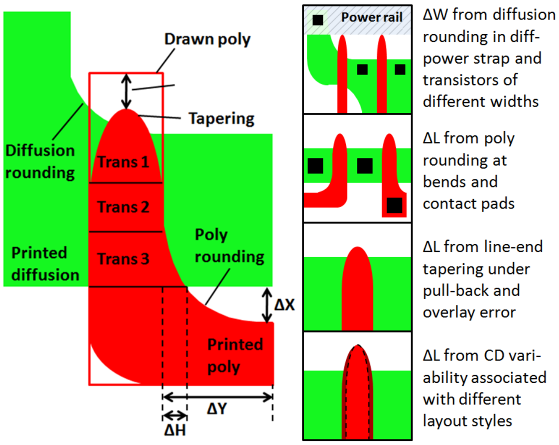
where $Y_{particles,l}$ is the yield from particle defects at layer l, k is the type of defect (e.g., open circuit, short circuit), $A_{c,j}$

[5] These area and runtime results are obtained by reducing the maximum number of iterations in the chaining algorithm to 18 iterations.

[6] In this work, we do not model the yield loss from lithography induced systematic failures. In future work, we plan to employ a 2D printability to account for such failures.

[7] This can be modified to conform with the process alignment strategy.

[8] In reality, overlay of different layers with the reference layer have some degree of correlation. This can be dealt with by reducing the amount of overlay (i.e. use smaller 3σ for overlay distribution).



Drawn poly
Tapering
Diffusion rounding
Trans 1
Trans 2
Trans 3
Poly rounding
Printed diffusion
ΔX
Printed poly
ΔY
ΔH
Power rail
ΔW from diffusion rounding in diff-power strap and transistors of different widths
ΔL from poly rounding at bends and contact pads
ΔL from line-end tapering under pull-back and overlay error
ΔL from CD vari-ability associated with different layout styles

Figure 16. Illustration of slicing model, rounding model parameters, and the sources of gate length and width variability considered in the DRE framework. Here, models for tapering and corner-rounding, rather than actual lithography simulation, are used to estimate the contours.

is the average critical area for defect type j, D_0 is the average defect density, and α is the defect clustering parameter.

IV. Variability

In sub-wavelength lithography regime, three sources of printing imperfection causing gate-dimension variation are dominant [36] (depicted in Figure 16):

- diffusion and poly corner rounding;
- line-end tapering under overlay error and line-end pull-back;
- CD variability associated with different patterning restrictions.

The contribution of each source to gate length and width variations (ΔW and ΔL) is modeled independently. First, we estimate the geometric change in gate length and width from each source. The estimated gates dimensions are then used to determine the overall variability. Our variability index for evaluating and comparing DRs is the total change in drive current, which we calculate using the following equation:

$$\Delta(\frac{W}{L}) = \frac{\sum_{allgates} \left|\Delta(\frac{W}{L})_i\right|}{(\frac{W_{tot}}{L})_{ideal}}, \tag{8}$$

where i represents the source of variability[9].

Since the resulting ΔW and ΔL are not across the entire gate, we quantify their contribution to $\Delta(\frac{W}{L})$ by modeling devices as parallel slices of transistors[10].

Diffusion rounding at corners formed by diffusion power-straps and unleveled abutment of transistors (as depicted in Figure 16) induces width variation at the gate edge. In addition, poly corner rounding in bends and contact-pads near the gate represents an important source of gate-length variation. The shape of the rounding is a function of the corner dimensions and is modeled as $\Delta H = K_1 \Delta Y \Big/ \sqrt[n]{1 + (\frac{\Delta Y}{K_2})^n}$

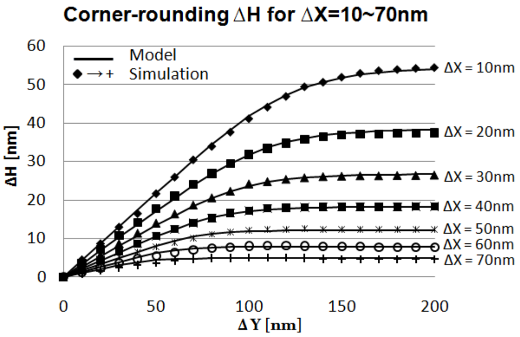


Figure 17. Rounding model fitted to give $< 0.8nm$ ΔH error with measured data from printed-image simulations on a fairly wide range of practical corner-dimensions ($\Delta X = 30 \rightarrow 70nm$ and $\Delta Y = 10 \rightarrow 200nm$).

where $K_1 = Ce^{D\Delta X}$ and $K_2 = A\Delta X + B$. In this model, ΔX, ΔY, and ΔH are depicted in Figure 16; A, B, C, D, and n parameters are fitted to give $< 0.8nm$ ΔH error with measured data from printed-image simulations on a fairly wide range of practical corner-dimensions[11] as shown in Figure 17. Simple geometric approximations are then used to infer the gate-length and gate-width variations from the ΔH values caused by the rounding of each corner (in the diffusion and poly layers). It is worth noting that approximate predictive rounding-models fitted from tentative simulation models, which are typically available in early stages of technology development, could be used in lieu of the current model.

Line-end tapering can affect the length of the gate at its edge. This effect becomes pronounced when considering line-end pull-back and poly-to-diffusion overlay error. The tapered shape and gate length at the transistor edge are described using the model offered in [16][12] while accounting for line-end pull-back (mean value) and overlay errors (from distribution). Line-ends are assumed to extend beyond the gate as far as possible unless the user enforces minimum line-end extension (LEE) rule for the entire layout.

CD uniformity (CDU) is another major contributor to the change in drive current. In our framework, CDU is described by a distribution, which captures the dependency on dose and focus variations. Pattern dependency is captured by using different CDU 3σ values for each poly-patterning style including 1D/2D patterning and multiple/fixed pitch, which can seriously impact CDU [11, 41, 42].

After determining all $\Delta(\frac{W}{L})$ terms from different sources, we compute the absolute sum of all terms for the entire layout with the intention of highlighting the actual gate variability. Finally, the drive current variability index is calculated using Equation 8.

[9] We realize that this estimate is approximate as effects from different sources can interfere. Nevertheless, it is a good indicator of worst-case variability and process control requirement.

[10] More accurate slicing models of [37–40] can also be embedded in the framework if they are available.

[11] The fitting of the model is performed only once per technology node. The model can be fitted to early printed-image simulations or actual silicon data from early testing. Models for our printed-image simulations, which we used to fit the corner-rounding model, were calibrated using CalibreOPC and 45nm OPC models.

[12] $L_i = 2a\big(1 - |\frac{h_i - k}{b}|^n\big)^{\frac{1}{n}}$, where l_i is the gate-length at i location in the line-end extension, h_i is the distance from i to gate-edge, a is half the nominal gate-length, b is the line-end extension, and k and n parameters describe the taper-shape. In our experiments, we use $k = 0$ and $n = 3$.

Table IV
BENCHMARK DESIGNS USED IN OUR EXPERIMENTS AND THEIR CORRESPONDING NUMBER OF CELL INSTANCES AND UNIQUE CELL TYPES.

Circuit	Description	Cell instances	Cell types
nova	video compression decoder	43156	81
vga	VGA/LCD controller core	36097	60
mips	processor core	17032	54
ae18	processor core	4358	50

Table V
PROCESS CONTROL PARAMETERS USED IN OUR EXPERIMENTS.

Parameter	45nm	65nm
Avg defect density [$faults/m^2$]	1395	1757
Critical defect size [nm]	34	45
Max defect size [nm]	250	250
Fab cleanliness parameter	3	3
Clustering parameter (α)	2	2
Contact-holes rate [ppm]	0.00004	0.00004
Overlay (3σ) [nm]	13	15
Line-end pull-back (mean) [nm]	10	14
Gate CDU (3σ) [nm]	2.6	3.3
Critical M1 line-width [nm]	10	15
Critical poly line-width [nm]	15	20
Critical contact-width [nm]	10	15

V. Experimental Setup and Results

In this section, we evaluate and analyze major contentious DRs and layout styles for 45nm open-source FreePDK process [22]. The DRE framework is also used to compare standard and low power 65nm process from a commercial vendor as well as study the density impact of alternative technologies for the M1 layer at the 14nm node. In another experiment, we collectively explore two gate-spacing related DRs.

A. Testing Setup

Throughout the experiments, we use four benchmark designs from [43] synthesized using Nangate 45nm Open Cell Library (scaled for testing with 65nm process). Table IV describes all designs and lists their cell counts and number of unique cell types.

The experiments were performed using 45nm open-source FreePDK process and 65nm process from a commercial vendor. Estimates of process control parameters associated with each process are summarized in Table V. We use projected values from ITRS technology roadmap [1] and typical values for critical M1 and poly line-width and critical contact-width, which represent the minimum acceptable width for the defect not to be considered a failure. CDU value in the table is for 2D-poly patterning. For 1D fixed-pitch poly, we use CDU 3σ improvement factor of 47% over 2D-poly reported by IBM in [11] and assume that half the improvement is from poly being unidirectional and the other half is from the poly pitch being fixed.

α and β parameters of the congestion threshold model (Equation 2) are fine-tuned in a control loop to minimize the error in the estimated area as discussed in Section II-D. Because these parameters model the routing efficiency, the tuning needs to be done just once and only for a small group of cells. We used a couple of cells from the Nangate library that covers the different routing schemes including: highly congested layout with an area increase due to the routing, highly congested layout without area increase, and highly congested layout in a single direction.

Because the area of the benchmark designs is relatively small, we normalize POS values to a $100mm^2$ chip area. We determine for the base case in each experiment the number of design copies that can fit in $10 \times 10mm$ chip size with 80% cell-area utilization and find the corresponding number of contacts and critical areas.

The results of the DR evaluations are a strong function of the base set of rules, layout styles, library architecture, and design type and, hence, they are *not generalizable*. First, we perform studies on 45nm FreePDK process and later we perform studies on a 65nm commercial process as an example.

The number of possible case studies that DRE framework can perform is huge. For brevity, we only show studies of some important DRs and layout styles including: 1D/2D-poly, multiple/fixed pitch poly, diffusion/M1 power-straps, and 8/10/12-track cell heights. Our baseline experiment unless otherwise specified is with the following setup:

- limited routing fixed-pitch poly,
- M1 power-straps,
- and 10-track cell height.

B. Evaluation of Poly-Patterning Restrictions

Five configurations of poly-patterning styles are investigated:

- unrestricted poly, i.e. 2D-poly,
- limited wrong-way poly, i.e. limited poly routing,
- no poly routing, i.e. 1D-poly,
- limited routing fixed-pitch poly,
- and fixed-pitch 1D poly.

In the cases of 1D poly configuration, poly is used only to connect dual gates (i.e. gates of same transistor-pair). In the cases of limited poly routing, poly is also used to connect adjacent gates in the same p or n network. In the case of 2D poly, poly is used to perform all gate interconnections unless it is blocked by previous routing or diffusion power-straps.

Figure 18 shows area, manufacturability, and variability tradeoffs associated with the five configurations of poly-patterning styles on a 45nm process with M1 power-straps and a 10-track cell height.

We observe that 2D poly has a considerable 15% area benefit compared to limited poly routing. On the downside, 2D poly leads to roughly 3× larger variability compared to 1D poly, which is mainly caused by CDU improvement associated with unidirectional patterning. On the other hand, limited poly routing has only 3% area benefit compared to 1D poly and leads to a much larger variability. Thus, allowing small notches on poly (H, U, and Z shapes) with RET complications does not bring much benefits.

Fixed-pitch 1D poly implementation leads to 37% less variability compared to multiple-pitch 1D poly implementation and almost the same area. The area overhead of the fixed-pitch poly restriction is small because the minimum gate pitch (of two stacked gates) is equal to the contacted-gate pitch in FreePDK process and, consequently, a gate-spacing increase is necessary only for isolated gates (with a diffusion gap between the gates).

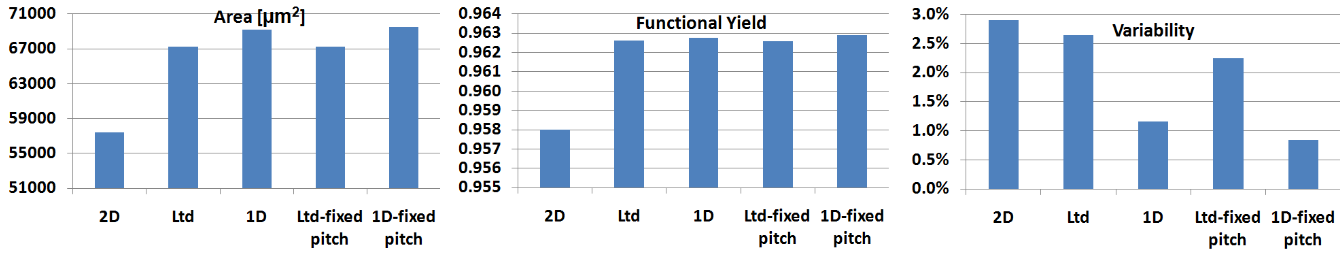


Figure 18. Evaluation of restrictive poly-patterning styles on 45nm FreePDK process[13].

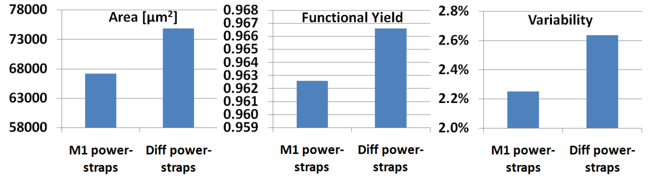
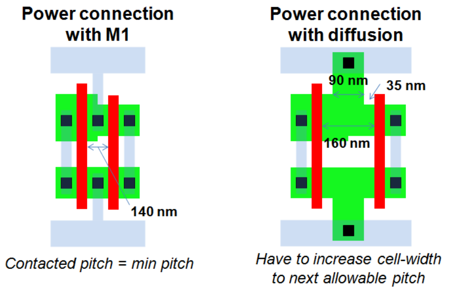


Figure 19. Evaluation of M1/diffusion power-strap styles on 45nm FreePDK process[13].


Power connection with M1
140 nm
Contacted pitch = min pitch
Power connection with diffusion
90 nm
35 nm
160 nm
Have to increase cell-width to next allowable pitch

(a) (b)

Figure 20. Example of a layout with M1 power-strap (a) and with a diffusion power-strap (b).

C. Evaluation of Layout Styles

Figure 19 shows area, manufacturability, and variability tradeoffs associated with M1/diffusion power-straps on 45nm process with limited routing fixed-pitch poly and 10-track cell height. The diffusion power-strap style results in a much larger variability than in the case of the M1 power-strap style (79% larger), which manifests the intensity of the diffusion corner-rounding effect. The reason for this large effect is the fact that cells are packed in the horizontal direction to minimize the cell width and minimum DRs are used. In contrast, poly corner-rounding and line-end tapering effects are usually less important because cells are normally relaxed in the vertical direction (cell-height being fixed).

Furthermore, 11% area overhead is associated with the diffusion power-strap style. This overhead is due to the extra gate separation required to drop the power strap as illustrated in Figure 20. The required gate separation at diffusion power straps is even larger when the fixed-pitch poly style is adopted. On the good side, diffusion straps reduce M1 congestion and, consequently, the area of some of the congested cells. In another experiment (not shown in Figure 19) with a smaller cell-height (8 tracks), diffusion power-strap style leads to a smaller area overhead (9.6%) than in the case of 10-track cell height, which is because M1 congestion affects the cell area seriously when the cell height is small.

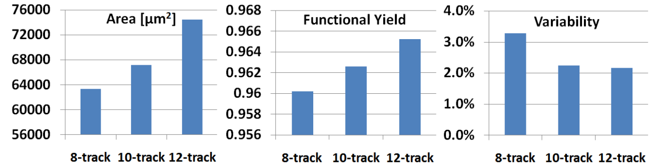


Figure 21. Evaluation of 8/10/12-track cell height on 45nm FreePDK process[13].

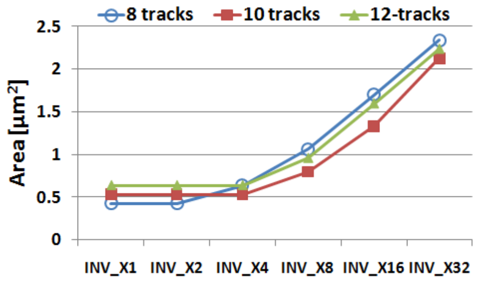


Figure 22. Increasing area with increasing transistor width for 8/10/12-track cell height.

Diffusion power-straps have some manufacturability benefits. Gate-to-contact shorts are reduced and contact redundancy for power connections is implemented at no cost since these contacts are placed on the power-rail in this case.

We also investigate different cell-height decisions. Figure 21 shows area, manufacturability, and variability tradeoffs associated with 8/10/12-track cell heights on the FreePDK 45nm process with limited routing fixed-pitch poly and M1 power-straps style. The results show a considerable improvement of variability (32%) when the number of tracks is increased from 8 to 10, but only a slight improvement (4%) when the number of tracks is increased from 10 to 12. This is because poly corner-rounding and line-end tapering are aggravated when cells are packed in the vertical direction in the case of a small cell height. The smallest cell-area of the benchmark designs is achieved with 8-track cell height. However, this is not true for all cells as a large cell height is more suitable for cells with wide transistors (as Figure 22 shows), i.e. high-performance designs.

D. Assessment of Technologies and Wiring Schemes

In the previous experiments, we assumed a single metal layer (M1) for the wiring of transistors. Here, we study the effect of allowing an extra metal layer. We run the baseline experiment in the case where M1 is bidirectional and M2 is used only to resolve M1 congestion as well as the case of 1D

[13] The Y-axis showing the functional yield does not start from the zero value to emphasize differences in results (although the differences are tiny).

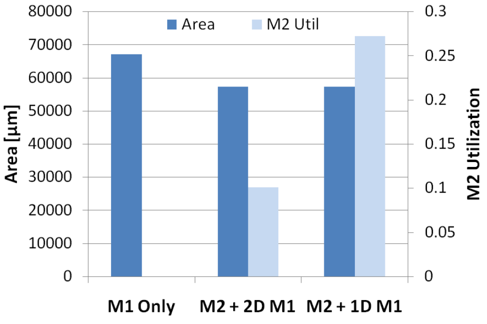


Figure 23. Layout area and M2 utilization results when M2 is used only in case of congestion on M1 and when M1 is unidirectional.

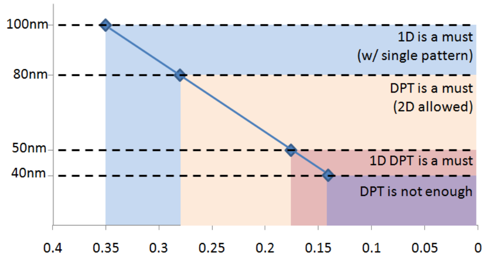
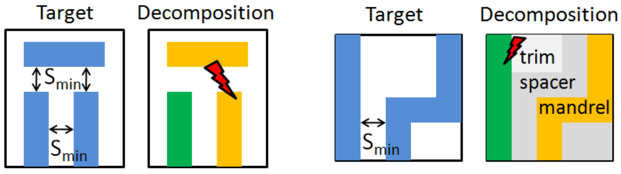


Figure 24. Wiring pitch as a function of the k_1 factor and the limits of patterning technologies and directionality (based on [44]).



(a)

(b)

Figure 25. Examples of a PS-DPT forbidden pattern because of a coloring conflict (a) and a SADP forbidden pattern because a line-side that cannot be defined except with the trim exposure (b)[15].

layout style for M1 and M2. Figure 23 depicts the cell area and the M2 utilization associated with each wiring scheme. Allowing M2 in the cell layout reduces the area by 17% on average across all benchmark designs. In case of unidirectional M1 and M2, M2 utilization reaches 27%; whereas in case of bidirectional M1 and M2 used only when M1 is congested, M2 utilization reaches just 10% (2.7× smaller than 1D M1). The downside of higher M2 utilization in the cell layouts is more blockages for the routing at the chip level, which may cause a larger chip area or the need for a larger number of routing layers.

The DRE framework can also be used to assess design implications of patterning technologies. We will show this through an example. Let us consider the patterning for the M1/M2 layers at the 14nm technology node where the alternative technologies are: Single+Trim Exposure and unidirectional M1 (STE) as well as Double-Patterning Technology (DPT) including Pitch-Split Double-Patterning Technology (PS-DPT) and Self-Aligned Double Patterning (SADP), a.k.a. Sidewall Image Transfer (SIT).

In STE, the assumed process consists of forming a grating of unidirectional M1 at fixed pitch with a single exposure followed by a trim exposure to form line-ends. PS-DPT consists of two separate exposure and etch steps, essentially splitting the layout patterns into two separate masks so that the pitch on the mask is relaxed. SADP consists of forming a first pattern at a relaxed pitch, depositing a sidewall-spacer around the first pattern, and, lastly, defining a second pattern based on the combination of the sidewall-spacer and a trim exposure[14]. On one hand, SADP has typically higher fabrication cost than PS-DPT because it involves more processing steps. On the other hand, when the trim exposure of SADP is allowed to define line-ends but not line-sides (to prevent overlay of trim to mandrel from translating into line-width variation), SADP is more favorable than PS-DPT because of its better overlay performance.

Using immersion lithography and presuming a numerical aperture (NA) equal to 1.35, the limit of bidirectional resolution is at k_1 factor of 0.35 and the limit of unidirectional resolution is at k_1 factor of 0.28 [44]. Therefore, the best pitch that can be achieved with STE and unidirectional M1 is roughly 80nm. With DPT (PS-DPT or SADP), the k_1 limits for bidirectional and unidirectional patterning are roughly one half that of single patterning presented earlier [44]. So, the best achievable pitch with DPT while maintaining bidirectional patterning is 50nm and the best achievable pitch with unidirectional patterning is 40nm. Figure 24 shows the wiring pitch as a function of the k_1 factor and the limits of patterning technologies and directionality.

PS-DPT requires the decomposition of the layout into a first and a second mask layout. Features assigned to the same mask layout must meet the minimum spacing rule of single exposure, which is typically 2× the minimum spacing in the complete layout. For the decomposition to be successful (i.e. without violations), the layout must be adapted for PS-DPT. The layout can be adapted either in a construct-by-correction approach with post-layout perturbations or in a correct-by-construction approach during the design of the layout. One possible method of the correct-by-construction approach is to use conservative spacing rules that, if met, prevent any violations and shield the layout designer from the complexity in dealing with double-patterning violations. For our study, we evaluate the latter method and the minimum spacing of single exposure to be 2× the minimum spacing in the layout. To guarantee *almost zero* double-patterning violations for 2D layouts, we set all rules that involve a tip as well as the L-shape-to-line spacing to the minimum spacing of single exposure. Similarly to PS-DPT, SADP requires layout adaption. Unlike PS-DPT violations, SADP violations are too complex to be prevented with simple geometric rules. 1D layout however, are guaranteed to be SADP decomposable. So for SADP, we assume a 1D M1/M2 in our study. Table VI gives a summary list of layout styles and DRs assumptions made for each technology in our study.

We study the density impact of the different alternative technologies available at the limits of the k_1 factor shown in Figure 24. At 80nm wiring pitch, we can either have a 1D layout and use STE or enable 2D layout with PS-DPT. At 50nm wiring pitch, we can either have a 2D layout with PS-DPT or a 1D layout with SADP. Finally, for 40nm wiring

[14] In a sidewall-is-dielectric process, the first and second patterns are lines; in a sidewall-is-metal process, the first and second patterns are spaces.

[15] S_{min} is the minimum spacing in the layout. Here, we show a sidewall-is-dielectric process for SADP with a trim mask not allowed to define line-sides.

Table VI
SUMMARY OF PATTERNING STYLES AND RULES ASSUMPTIONS MADE FOR EACH TECHNOLOGY IN OUR STUDY.

Tech	Assumptions
SIT	• Unidirectional patterning only • Pitch = 80nm • All spacing rules = 40nm
PS-DPT	• Bidirectional patterning • Pitch from 80 to 50nm (SADP more favorable below 50nm) • Spacing rules = half pitch except rules involving tips and L-to-L-shape spacing, which are equal to 2× the half-pitch
SADP	• Unidirectional patterning only • Pitch from 80 to 40nm • All spacing rules = half pitch

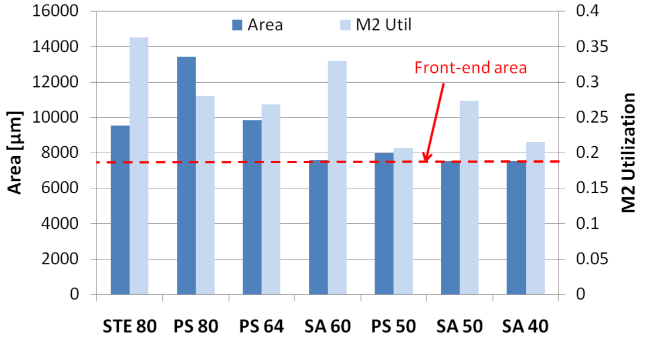


Figure 26. Layout area and M2 utilization results for STE, PS-DPT (PS), and SADP (SA) for M1/M2 pitch between 80 and 40nm. Front-end area denotes the area of diffusion, poly, and contacts layers.

pitch, only a 1D layout with SADP is possible. For STE, we assume a tip-to-tip spacing rule equal to the minimum spacing in our study[16]. PS-DPT and SADP impose peculiar layout restrictions, however, and many patterns cannot be formed with these technologies (see examples of Figure 25).

We run DRE for the designs of Table IV with the three patterning technologies at M1/M2 while assuming all other layers are patterned the same way (i.e. the DRs at all other layers are kept the same in all runs). The results of the evaluation are shown in Figure 26. PS-DPT at M1/M2 pitch of 80nm leads to 29% larger area than that achieved with STE at the same pitch. To achieve the same area as with STE at 80nm pitch, the wiring pitch of PS-DPT must be less than 64nm. Hence, a correct-by-construction approach through conservative spacing rules to enable M1/M2 layouts for PS-DPT may not be satisfactory (given the associated area overhead). On the good side, because PS-DPT allows 2D M1, the M2 utilization with PS-DPT is considerably smaller than that with the other technologies (43% smaller than STE and 24% smaller than SADP). SADP in a correct-by-construction approach with 1D layout seems to be the best alternative in terms of cell area. It can achieve almost the minimum possible area (i.e. the front-end area), which corresponds to 21% smaller area than that of STE, at the wiring pitch of 60nm.

Another example of technology assessment using the DRE framework is the assessment of Shift-Trim Double-Patterning Lithography (ST-DPL), a new double-patterning technology that we propose in [45]. ST-DPL essentially consists of applying a translational mask shift to re-use the same photomask for both exposures of DPT and removing extra printed features using a non-critical trim exposure. To validate the new technique and study its impact on layout density, we migrated in [45] a small set of standard cells from an existing library so that they become compatible with ST-DPL. Because the automated generation of actual layouts that are ST-DPL compatible is not currently available, the migration of cells was performed manually. Manual layout generation is time-consuming however; only a limited number of layouts can be actually generated and just few layout styles can be tried in practice. Moreover, specific rules are required to simplify the trim mask and exposure in ST-DPL and evaluating the impact of these rules with manual layout generation is practically impossible.

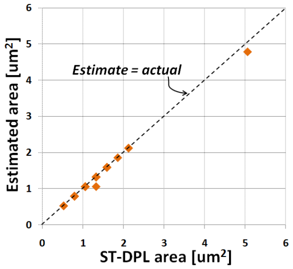


Figure 27. DRE estimated cell area versus the actual area of ST-DPL compatible cells designed manually (41 standard cells).

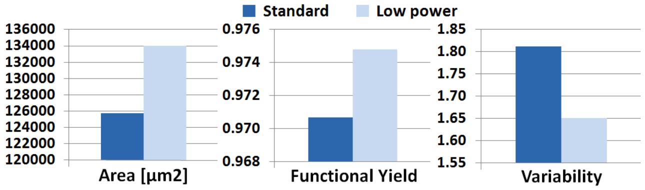


Figure 28. Comparison between a standard and a low power 65nm process from the same commercial vendor.

An efficient alternative to manual layout generation is the use of DRE to evaluate the impact of ST-DPL on the design. In Figure 27, we compare the area of ST-DPL-compatible cells that are manually generated with the cell area estimated by DRE. For 39 cells, the estimated areas match exactly with the actual areas and, for only two cells, the estimated areas are off from the actual areas by a single poly pitch. The good accuracy of the results imply that DRE can be used instead of manual layout generation to obtain an estimate of the density impact of ST-DPL.

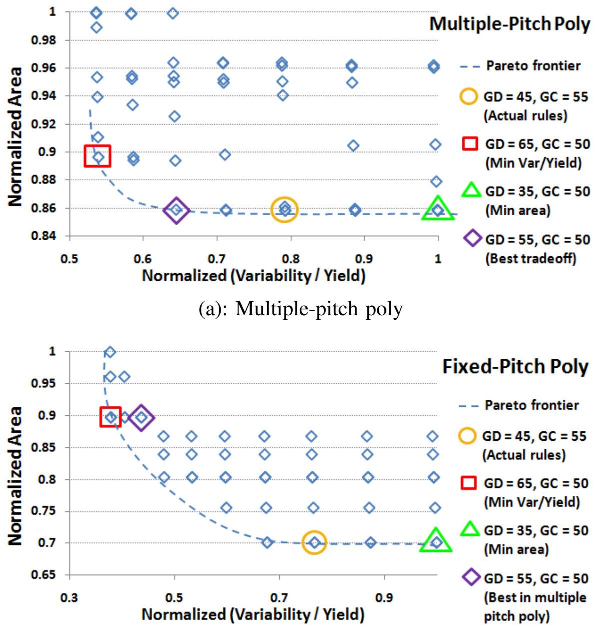
E. DR Comparison of Different Processes

Comparison of DR sets of different processes is another application of the DRE framework. Here, we compare DRs of a standard and a low power 65nm process from the same commercial vendor. We perform this comparison with the layout styles of the baseline experiment. The results depicted in Figure 28 show an advantage of low power over standard process in terms of variability and manufacturability; on the other hand, standard process is more area-efficient (7.9% less area).

F. DR Exploration

The DRE framework is used for the collective exploration of gate-to-diffusion (GD) and gate-to-contact (GC) spacing rules

[16] Implying that the minimum linewidth of the trim mask is the same as that of the first exposure mask.



(a): Multiple-pitch poly

(b): Fixed-pitch poly

Figure 29. Co-exploration of GC/GD rules (see figure 2) in a commercial 65nm process with diffusion power-straps and limited routing.

in the 65nm commercial process. We perform the study for all benchmark designs of Table IV and use diffusion power-straps and limited routing multiple-pitch poly styles that were common at the 65nm node.

The results are depicted in Figure 29(a). Each data point correspond to unique combination of GD/GC values. The Y-axis represents the normalized area and the X-axis represents the normalized variability over yield ratio (average values across all benchmark designs). The point corresponding to the process GD/GC actual values falls on the Pareto frontier and very near the solution with the "best tradeoff" (i.e. smallest variability to yield ratio among solutions with almost the smallest area).

We repeat the same experiment with a limited routing fixed-pitch poly style and show the results in Figure 29(b). The solution with the "best tradeoff" in the previous experiment shifts away from the Pareto optimal frontier and is associated with a large area in this case. Yet, the point corresponding to the process GD/GC actual values falls again on the Pareto optimal frontier and very near the *new* "best tradeoff" solution.

Although quite simplistic, this example provides compelling evidence of our evaluation metrics fidelity and validates our approach. Moreover, the outcomes of this experiment suggest that the optimality of DRs depend strongly on the layout methodology that is in use (layout styles and library architecture) and DR exploration and optimization should be performed across the different layout methodologies that may be used with the process. This example also shows that the DRE framework can be used as a first-level filter in a DR optimization loop. Rather than exploring the entire search space of DRs with conventional runtime-expensive methods, DRE can be used to quickly eliminate poor DR choices.

VI. Conclusions and Future Work

We proposed a novel framework for *fast*, *early* and *systematic* evaluation and exploration of design rules and technology decisions (**available for download at http://nanocad.ee.ucla.edu/Main/DownloadForm**). By using first order models of circuit characteristics and layout topology and metal congestion-based area estimation, our framework can evaluate big decisions *before* exact process and design technologies are known. In this paper, we illustrated the potential applications of our framework for the collective evaluation and exploration of DRs as well as the quantitative comparison of DRs from different processes and different technology alternatives. The framework makes DR generation and optimization easier and much faster. Rather than exploring the entire search space of DRs with conventional runtime-expensive methods, the framework can be used as a first-level filter to quickly eliminate poor DR choices. To the best of our knowledge, this is the first work that includes all area, manufacturability, and variability metrics in the evaluation of DRs. Nevertheless, this is just the first step and our ongoing work pursues the following directions:

- address design rule effects on other layout and circuit characteristics including performance, power, reliability, and some notion of designability;
- introduce a 2D printability model (not based on field simulation), for example, derived from [46–48];
- extrapolate the DR evaluation to the chip level and include intermediate and global metal and via layers;
- study interactions and tradeoffs of variability and area, as in [49] for example.

Acknowledgements

This work was generously supported in part by IMPACT UC Discovery Grant (http://impact.berkeley.edu) contract number ele07-10291, Semiconductor Research Corporation contract number 2008-TJ-1816, and NSF CAREER Award number 0846196. The authors would like to acknowledge the fruitful discussions with Dr. Lars Liebmann and Dr. Praveen Elakkumanan from IBM and Dr. Chul-Hong Park from Samsung. The authors would also like to thank Mr. Tuck-Boon Chan from UCSD for providing the corner-rounding model and Mr. Santiago Mok from UCLA for his help with the implementation of the netlist parser.

References

[1] International Technology Roadmap for Semiconductors, Report 2007.

[2] I. Yoneda *et al.*, "Study of nanoimprint lithography for applications toward 22nm node cmos devices," in *Proc. SPIE*, vol. 6921, 2008, p. 692104.

[3] T. Maruyama *et al.*, "Ebdw technology for eb shuttle at 65nm node and beyond," in *Proc. SPIE*, vol. 6921, 2008, p. 69210H.

[4] H. Meiling *et al.*, "Euvl system: moving towards production," in *Proc. SPIE*, vol. 7271, 2009, p. 727102.

[5] K. Lai *et al.*, "32 nm logic patterning options with immersion lithography," in *Proc. SPIE*, vol. 6924, 2008, p. 69243C.

[6] C. A. Mack, "Seeing double," *IEEE Spectrum*, pp. 46–51, 2008.

[7] G. Bailey *et al.*, "Double pattern eda solutions for 32nm hp and beyond," in *Proc. SPIE*, vol. 6521, 2007, p. 65211K.

[8] A. Hazelton *et al.*, "Double patterning requirements for optical lithography and prospects for optical extension without double patterning," in *Proc. SPIE*, vol. 6924, 2008, p. 69240R.

[9] A. Balasinski *et al.*, "Patterning techniques for next generation ic's," in *Proc. SPIE*, vol. 6798, 2008, p. 679803.

[10] L. W. Liebmann, "Layout impact of resolution enhancement techniques: impediment or opportunity?" in *Proc. IEEE Intl. Symp. on Physical Design*, 2003, pp. 110–117.

[11] L. W. Liebmann *et al.*, "High-performance circuit design for the ret-enabled 65nm technology node," in *Proc. SPIE*, vol. 5379, 2004, pp. 20–29.

[12] Y. Zhang, J. Cobb, A. Yang, J. Li, K. Lucas, and S. Sethi, "32nm design rule and process exploration flow," in *Proc. SPIE*, vol. 7122, 2008, p. 71223Z.

[13] V. Joshi, B. Cline, D. Sylvester, D. Blaauw, and K. Agarwal, "Leakage power reduction using stress-enhanced layouts," in *Proc. ACM/IEEE Design Automation Conference*, 2008, pp. 912–917.

[14] Y. Sheu *et al.*, "Modeling well edge proximity effect on highly-scaled MOSFETs," in *Proc. of IEEE CICC'05*, 2005, pp. 831–834.

[15] L. Capodieci, P. Gupta, A. B. Kahng, D. Sylvester, and J. Yang, "Toward a methodology for manufacturability-driven design rule exploration," in *Proc. IEEE/ACM Design Automation Conference*, 2004, pp. 311–316.

[16] P. Gupta, K. Jeong, A. B. Kahng, and C. Park, "Electrical metrics for lithographic line-end tapering," in *Proc. SPIE*, vol. 7028, 2008, p. 70283A.

[17] V. Dai, L. Capodieci, J. Yang, and N. Rodriguez, "Developing DRC Plus rules through 2D pattern extraction and clustering techniques," in *Proc. SPIE*, vol. 7275, 2009, p. 727517.

[18] S. Chang *et al.*, "Exploration of complex metal 2D design rules using inverse lithography," in *Proc. SPIE*, vol. 7275, 2009, p. 72750D.

[19] R. S. Ghaida and P. Gupta, "A framework for early and systematic evaluation of design rules," in *Intl. Conf. on Computer-Aided Design*, 2009, pp. 615–622.

[20] A. R. Subramaniam, R. Singhal, C. Wang, and Y. Cao, "Design rule optimization of regular layout for leakage reduction in nanoscale design," in *Proc. Asia and South Pacific Design Automation*, 2008, pp. 474–479.

[21] S. Kobayashi *et al.*, "Yield-centric layout optimization with precise quantification of lithographic yield loss," in *Proc. SPIE*, vol. 7028, 2008, p. 70280O.

[22] FreePDK. [Online]. Available: http://www.eda.ncsu.edu/wiki/FreePDK

[23] L. Liebmann, L. Pileggi, J. Hibbeler, V. Rovner, T. Jhaveri, and G. Northrop, "Simplify to survive, prescriptive layouts ensure profitable scaling to 32nm and beyond," in *Proc. SPIE*, vol. 7275, 2009, p. 72750A.

[24] J. Yang, L. Capodieci, and D. Sylvester, "Layout verification and optimization based on flexible design rules," in *Proc. SPIE*, vol. 6156, 2006, p. 61560A.

[25] J. P. S. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, pp. 83–97, 1955.

[26] C. Hwang, Y. Hsieh, Y. Lin, and Y. Hsu, "A fast transistor-chaining algorithm for cmos cell layout," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 9, no. 7, pp. 781–786, 1990.

[27] C. M. Fiduccia and R. M. Mattheyses, "A linear time heuristic for improving network partitions," in *Proc. ACM/IEEE Design Automation Conference*, 1982, pp. 175–181.

[28] Nangate open cell library v1.2. [Online]. Available: http://www.si2.org/openeda.si2.org/projects/nangatelib

[29] C.-H. Park, D. Z. Pan, and K. Lucas, "Exploration of vlsi cad researches for early design rule evaluation," in *IEEE Asian and South Pacific Design Automation Conference*, 2011, pp. 405–406.

[30] B. Eichelberger *et al.*, "32nm overlay improvement capabilities," in *Proc. SPIE*, vol. 6924, 2008, p. 69244C.

[31] Eyes semiconductor yield estimation tool. [Online]. Available: http://www.icyield.com/yieldmod.html

[32] W. Maly, "Modeling of lithography related yield losses for CAD of VLSI circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. CAD-3, no. 3, 1985.

[33] J. P. de Gyvez, "Yield modeling and beol fundamentals," in *Proc. IEEE/ACM System Level Interconnect Prediction SLIP'01*, 2001, pp. 135–163.

[34] C. H. Stapper, "Modeling of integrated circuit defect sensitivities," *IBM Journal of Research and Development*, vol. CAD-3, no. 3, pp. 549–557, 1983.

[35] I. Koren and Z. Koren, "Defect tolerance in vlsi circuits: techniques and yield analysis," in *Proc. of the IEEE*, vol. 86, no. 9, 1998, pp. 1819–1837.

[36] T.-B. Chan, R. S. Ghaida, and P. Gupta, "Electrical modeling of lithographic imperfections," in *Intl. Conf. on VLSI Design, VLSID'10*, 2010, pp. 423–428.

[37] P. Gupta, A. B. Kahng, Y. Kim, S. Shah, and D. Sylvester, "Investigation of diffusion rounding for post-lithography analysis," in *Proc. Asia and South Pacific Design Automation*, 2008, pp. 480–485.

[38] R. Singhal, A. Balijepalli, A. Subramaniam, F. Liu, S. Nassif, and Y. Cao, "Modeling and analysis of non-rectangular gate for post-lithography circuit simulation," in *Proc. of Design Automation Conference*, 2007, pp. 823–828.

[39] P. Gupta, A. B. Kahng, S. Nakagawa, S. Shah, and P. Sharma, "Lithography simulation-based full-chip design analyses," in *Proc. SPIE*, vol. 6156, 2006, p. 61560T.

[40] S. X. Shi, P. Yu, and D. Z. Pan, "A unified non-rectangular device and circuit simulation model for timing and power," in *Proc. IEEE/ACM Int'l Conf. on Computer-Aided Design*, 2006.

[41] M. C. Smayling, H. Liu, and L. Cai, "Low k1 logic design using gridded design rules," in *Proc. SPIE*, vol. 6925, 2008, p. 69250B.

[42] L. Pileggi, H. Schmit, A. Strojwas, P. Gopalakrishnan, V. Kheterpal, A. Koorapaty, C. Patel, V. Rovner, and K. Tong, "Exploring regular fabrics to optimize the performance-cost trade-off," in *Proc. IEEE/ACM Design Automation Conference*, 2003.

[43] [Online]. Available: http://www.opencores.org/

[44] T. Wallow. (2009, July) Logic double patterning at pitches below 80 nm. Presentation in Sokudo Lithography Forum Semicon West. [Online]. Available: http://www.sokudo.com/event/images/090715/SOKUDO_LBF2009_GLOBALFOUNDARIES.pdf

[45] R. S. Ghaida, G. Torres, and P. Gupta, "Single-mask double-patterning lithography," *IEEE Transactions on Semiconductor Manufacturing*, vol. 24, no. 1, pp. 93–103, Feb 2011.

[46] A. B. Kahng, C. Park, and X. Xu, "Fast dual graph-based hotspot detection," in *Proc. SPIE*, vol. 6349, 2006, p. 63490H.

[47] M. Cho, K. Yuan, Y. Ban, and D. Z. Pan, "ELIAD: efficient lithography aware detailed router with compact post-opc printability prediction," in *Proc. Design Automation Conference*, 2008, pp. 504–509.

[48] B. Yenikaya and A. Sezginer, "A rigorous method to determine printability of a target layout," in *Proc. SPIE*, vol. 6521, 2007, p. 652112.

[49] K. Jeong, A. B. Kahng, and K. Samadi, "Quantified impacts of guardband reduction on design process outcomes," in *Proc. Intl. Symp. Quality Electronic Design ISQED'08*, 2008, pp. 790–797.

Rani S. Ghaida (S'03) is a final-year PhD candidate and researcher at the department of Electrical Engineering at UCLA. He earned the Master's degree in Computer Engineering from the University of New Mexico in 2008. In 2010, Rani was on internship at IBM Austin Research Lab, where he worked on the design enablement of double-patterning lithography. In 2011, He was with IBM T. J. Watson Research Center where he worked, in collaboration with IBM Semiconductor R&D Center in East Fishkill, on developing a platform for exploring design rules and patterning strategies for the 14nm node. His research work has been focused on Design-Centric Assessment of Technology, Design/Technology Co-Optimization, and Design for Manufacturability. He has several conference and journal publications as well as five pending patents in his field. Rani is a student member of the IMPACT research center headquartered in UC Berkeley and the Semiconductor Research Corporation (SRC).

Puneet Gupta (S'02–M'08) is currently a faculty member of the Electrical Engineering Department at UCLA (http://nanocad.ee.ucla.edu). He received the B. Tech degree in Electrical Engineering from Indian Institute of Technology, Delhi in 2000 and Ph.D. in 2007 from University of California, San Diego. He co-founded Blaze DFM Inc. (acquired by Tela Inc.) in 2004 and served as its product architect till 2007. He has authored over 80 papers, 15 U.S. patents, and a book chapter. He is a recipient of NSF CAREER award, ACM/SIGDA Outstanding New Faculty Award and SRC Inventor Recognition Award. Dr. Gupta's research has focused on building high-value bridges across application-architecture-implementation-fabrication interfaces for lowered cost and power, increased yield and improved predictability of integrated circuits and systems.