Enhanced Design Flow and Optimizations for Multiproject Wafers

Andrew B. Kahng, Ion I. Măndoiu, Xu Xu, *Student Member, IEEE*, and Alexander Z. Zelikovsky

***Abstract*—The aggressive scaling of very large-scale integration feature size and the pervasive use of advanced reticle enhancement technologies lead to dramatic increases in mask costs, pushing prototype and low-volume production designs to the limit of economic feasibility. Multiproject wafers (MPWs), or "shuttle" runs, provide an attractive solution for such designs by providing a mechanism to share the cost of mask tooling among up to tens of designs. However, MPW reticle floorplanning and wafer dicing introduce complexities that are not encountered in typical single-project wafers. Recent works on wafer dicing adopt one or more of the following assumptions to reduce problem complexity: 1) equal production volume requirement for all designs; 2) same dicing plan used for all wafers or for all rows/columns of reticle images on a wafer; 3) unrealistic wafer models such as a rectangular array of projections; and 4) fixed wafer shot-map. Although using one or more of the aforementioned assumptions makes the problem solvable, the performance of the solutions is degraded. In this paper, a comprehensive MPW flow aimed at minimizing the number of wafers needed to fulfill given die production volumes is proposed. The proposed flow includes two main steps: 1) multiproject reticle floorplanning and 2) wafer shot-map and dicing plan definition. For each of these steps, improved algorithms are proposed as follows. The proposed reticle floorplanner uses a hierarchical quadrisection combined with simulated annealing to generate "diceable" floorplans, observing given maximum reticle sizes. The proposed dicing planner allows multiple side-to-side dicing plans for different wafers and different reticle projection rows/columns within a wafer and further improves the dicing yield by partitioning each wafer into a small number of parts before individual die extraction. A wafer shot-map definition heuristic is also proposed in order to fully utilize round wafer real estate by extracting the maximum number of functional dies from both fully and partially printed reticle images. Experiments on industry test cases show that the proposed methods outperform significantly not only previous methods in the literature but also reticle floorplans manually designed by experienced engineers.**

***Index Terms*—Design automation, multiproject wafers, optimization methods, reticle design, wafer dicing.**

I. Introduction and Motivation

With the shrinking of very large-scale integration feature size and the pervasive use of advanced reticle enhancement technologies such as optical proximity correction and phase-shifting masks, mask costs are predicted to reach $10 million by the end of the decade. These high mask costs push prototyping and low-volume production designs to the limit of economic feasibility since the costs cannot be amortized over the volume. Multiproject wafers (MPWs), or "shuttle" runs, provide an efficient method to reduce the cost [10]. Thus, MPW has now become a commercial service offered by both independent providers such as MOSIS and CMP and semiconductor foundries such as the TSMC and IBM. An overview of related multilayer mask technologies, which rely on sharing the reticle space between multiple layers of the "same" design typically via blading, is given in [3].

Most previous papers on MPW reticle floorplanning rely on an "ideal dicing" model, which assumes either zero dicing loss [4] or arbitrary margins in the floorplan formulation. Chen and Lynn [5] considered the problem of finding the minimum area slicing floorplan, with 90° chip rotation allowed. Xu *et al.* [13] studied the MPW mask floorplanning under die-alignment constraints imposed by the use of die-to-die mask inspection. All these approaches assume that all dies can be obtained, which is impractical for current side-to-side wafer-dicing technology. A grid-packing formulation for MPW mask floorplanning is proposed in [1] and [2], with the assumption that an arbitrary blank area can be left on a die. However, in practice, arbitrary margins cannot be tolerated due to package requirement.[1]

Side-to-side dicing-based floorplanners consider the constraints imposed by the current side-to-side dicing technology. Due to the complexity of the general dicing problem, it is crucial to simplify the dicing problem and use a fast yet accurate wafer-cost evaluator in the floorplanners. According to the different dicing simplification methods, the current reticle floorplanners can be divided into two categories.

1) Single-wafer dicing plan (SWDP) assumption-based floorplanners assume that all wafers share the same dicing plan. Kahng *et al.* [7] were the first to consider the side-to-side wafer-dicing problem (SSWDP) with the SWDP assumption. They propose three optimal integer

[1]No margins are allowed for our industry test cases from CMP.

Manuscript received May 29, 2005; revised November 28, 2005 and April 10, 2006. This work was supported in part by Cadence Design Systems, Inc., in part by the MARCO Gigascale Silicon Research Center, in part by NSF Awards CCR-9988331, CCF-0429735, IIS-0546457, and DBI-0543365, and in part by U.S. CRDF Award MOM2-3049-CS-03. A preliminary version of the results in this paper has appeared in [8]. This paper was recommended by Associate Editor M. D. F. Wong.

A. B. Kahng is with the Departments of Computer Science and Engineering, and Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093-0114 USA (e-mail: abk@ucsd.edu).

I. I. Măndoiu is with the Department of Computer Science and Engineering, University of Connecticut, Storrs, CT 06269-2155 USA (e-mail: ion@engr.uconn.edu).

X. Xu was with the Department of Computer Science and Engineering, University of California at San Diego, La Jolla, CA 92093-0114 USA. He is now with Blaze DFM, Sunnyvale, CA 94089 USA (e-mail: xuxu@blaze-dfm.com).

A. Z. Zelikovsky is with the Department of Computer Science, Georgia State University, Atlanta, GA 30303 USA (e-mail: alexz@cs.gsu.edu).

Digital Object Identifier 10.1109/TCAD.2006.883922

linear programming (ILP) solutions and a fast heuristics for wafer-cost evaluation. The fast wafer-cost evaluator is used in a sequence-pair-based simulated annealing floorplanner. Recently, Kahng and Reda [9] proposed a grid floorplanner. The wafer cost of grid floorplans can be directly calculated with a closed-form formula. Therefore, it is even practical to apply a branch-and-bound algorithm to exhaustively search the whole solution space for small test cases. However, the closed-form wafer-cost calculation also depends on the impractical assumption that a wafer is a rectangular array of projections. In addition, the runtime of the proposed branch-and-bound algorithm may explode for large test cases.

2) Single-row and -column dicing plan (SRCDP)-assumption-based floorplanners employ the assumption that all rows and columns of reticle images within a wafer are diced using the same set of cuts. Xu *et al.* [14], [15] formulate the dicing problem as a minimum coloring problem. Wu and Lin [11] extend the minimum-coloring-based dicing approach by proposing three ILP formulations for optimal minimum coloring. In [12], they propose to perform chip replication and give integrated ILPs for simultaneous floorplanning and dicing, which are impractical even for small test cases due to large runtime.

In this paper, we propose a comprehensive MPW flow aimed at minimizing the number of wafers needed to fulfill the given die production volumes. Our flow includes two main steps: 1) multiproject reticle floorplanning, in which the reticle floorplan is designed for the given list of dies with fixed shot map and simplified dicing cost evaluation, and 2) wafer shot-map and dicing plan definition, in which the exact dicing plan and wafer-center location is determined for the floorplan generated in step 1. In step 2, the dicing plan generation algorithm is included in the wafer shot-map definition algorithm for accurate wafer-cost calculation. Our contributions are as follows. For the first flow step, we propose an algorithm based on a fixed hierarchical quadrisection structure, which is suitable for fast wafer-cost evaluation with simulated annealing to generate "diceable" floorplans, observing the given maximum reticle sizes. Our algorithm leads to an average reduction of 10%–20% in the required number of wafers compared to reticle floorplans manually designed by experienced industry engineers. For the second step, we give an integer program that can be used to find in practical time the "optimal" dicing plan under the SRCDP assumption. We also give a two-level optimization (TLO) algorithm that simultaneously allows multiple dicing plans (MDPs) for different wafers and for different reticle projection rows/columns within a wafer. We also show the advantages of partitioning each wafer into a small number of parts before individual die extraction. For a fixed reticle floorplan, the TLO algorithm is shown to give an average reduction in the required number of wafers of 42% without wafer partition and of 47% and 63% when partitioning into two and four parts, respectively. Finally, we propose to include wafer shot-map definition, which has not been previously considered in the context of MPW, in order to fully utilize the real estate on round wafers by extracting the maximum number of functional dies from both fully and partially printed reticle images. This optimization is shown to yield an average reduction of 13.6% in the required number of wafers for a reticle floorplan.

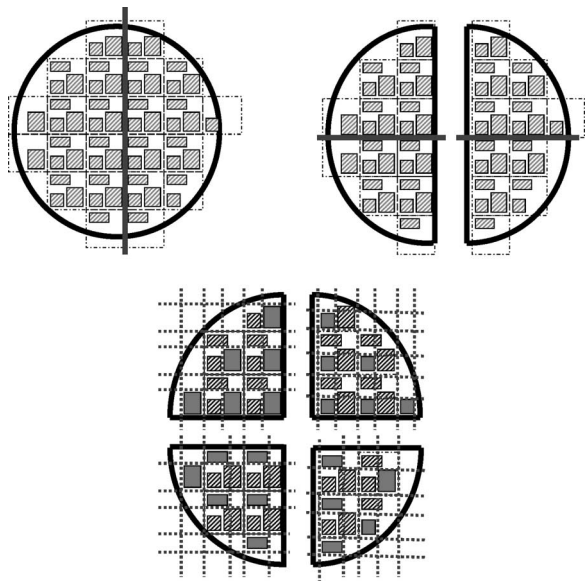


Fig. 1. Four quadrant dicing. The wafer is first divided into four quadrants. Then, each quadrant is diced independently using side-to-side cuts.

The rest of this paper is organized as follows. In the next section, we describe the basics of MPW with side-to-side wafer dicing. In Section III, a novel hierarchical quadrisection method is presented for reticle floorplanning. Section IV describes the MDP advantages and gives a novel TLO algorithm. Section V combines the wafer shot-map definition with dicing plan definition for further wafer-cost reduction. Finally, in Section VI, we give experimental results comparing the proposed methods on industrial test cases.

II. Preliminaries

A wafer consists of a number of reticle projections arranged in a number of reticle image "projection rows" and "projection columns." Each projection is a copy of the same reticle image. In the prevalent "side-to-side" wafer-dicing technology, the diamond blades cannot stop at arbitrary points during cutting; consequently, all projections in the same projection row (or column) will share the same horizontal (or vertical) cutlines. In this paper, we extend side-to-side dicing to allow preliminary partitioning of each wafer into a small number of parts (e.g., halves or quarters), as shown in Fig. 1, so that the side-to-side dicing plans for the parts can be independent from each other.

Following [7], two dies D and D' on a reticle are said to be in "vertical (horizontal) dicing conflict" if no set of vertical (horizontal) cuts can legally dice both D and D'. Let $\mathcal{D}$ denote the set of dies on a given reticle. The "vertical reticle conflict graph" $R_v = (\mathcal{D}, E_v)$ is the graph with vertices corresponding to the dies and edges connecting pairs of dies in vertical dicing conflict. The "horizontal reticle conflict graph" $R_h = (\mathcal{D}, E_h)$ is defined similarly. As usual, a set of vertices in a graph is called independent if they are pairwise nonadjacent. A "maximum horizontal (vertical) independent set" is a subset of $\mathcal{D}$,

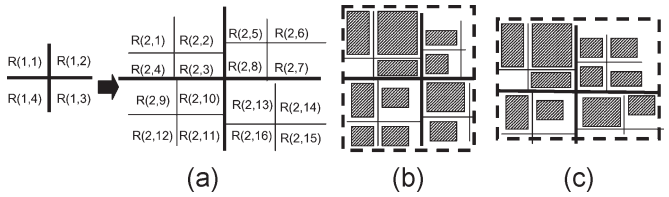


Fig. 2. Two-level hierarchical quadrisection floorplan.

which can be sliced out by a set of horizontal (vertical) cutlines; the set of cutlines used for a wafer is called as a "wafer-dicing plan." The "dicing yield" of die D is defined as the number of legally diced copies of D divided by its volume requirement. The "wafer-dicing yield" is defined as the minimum dicing yield over all dies $D \in \mathcal{D}$.

III. Reticle Floorplanning

In this section, we focus on the following MPW reticle floorplanning problem: Given a maximum reticle size and the size and required volume for each die, find a reticle floorplan (allowing die rotations) and a wafer-dicing plan that minimizes the number of used wafers.

Compared with other floorplanning problems, the main difficulty of the MPW reticle floorplanning problem lies in the wafer-cost calculation. To simplify and speed up the estimation of wafer cost and dicing plan yield, we use hierarchical quadrisection-based floorplanning. The reticle floorplan is based on a hierarchical quadrisection mesh, which is constructed in the following recursive way.

1) At level 1, the reticle area is divided into four regions with one horizontal line and one vertical line: $R(1,1)$, $R(1,2)$, $R(1,3)$, and $R(1,4)$, where $R(i,j)$ is the jth region for level i.
2) At level $i+1$, each region at level i $R(i,j)$ is divided into four regions with one horizontal line and one vertical line: $R(i+1,4^{j-1}+1)$, $R(i+1,4^{j-1}+2)$, $R(i+1,4^{j-1}+3)$, and $R(i+1,4^{j-1}+4)$.

Finally, there are 4^l regions at level l. Fig. 2(a) shows a mesh of level 2. The constructed mesh is "soft" since the dimensions of the regions are determined by the dies within the regions. The number of level l is chosen such that 4^l is greater than the number of dies.[2] Then, we place the dies in the regions of the level l mesh such that each region $R(l,j)(j=1,\ldots,4^l)$ contains at most one die. Different die placements lead to different reticle floorplans. Fig. 2(b) and (c) shows two different reticle floorplans for a set of ten dies based on the same mesh in Fig. 2(a). A simulated-annealing-based algorithm is used to find the best die placement.

We denote the width and height of region $R(i,j)$ as $W(R(i,j))$ and $H(R(i,j))$, respectively. The hierarchical quadrisection allows computing the height and width in a bottom-up manner.

1) At level l, if there is a die in region $R(l,j)$, $W(R(l,j))$ is equal to the width of the die, and $H(R(l,j))$ is equal to the height of the die; otherwise, $W(R(l,j)) = H(R(l,j)) = 0$.
2) At level i, $W(R(i,j)) = \mathrm{Max}(W(R(i+1,4^{j-1}+1)), W(R(i+1,4^{j-1}+4))) + \mathrm{Max}(W(R(i+1,4^{j-1}+2)), W(R(i+1,4^{j-1}+3)))$. $H(R(i,j)) = \mathrm{Max}(H(R(i+1,4^{j-1}+1)), H(R(i+1,4^{j-1}+2))) + \mathrm{Max}(H(R(i+1,4^{j-1}+3)), H(R(i+1,4^{j-1}+4)))$.

There are two main advantages of the proposed floorplan structure. First, the structure is suitable for conflict elimination since there are no conflicts between dies located in diagonally opposite regions. Second, the wafer cost can be easily evaluated with the following lemma.

Lemma 1: All dies can be divided into at most 2^l conflict-independent sets of dies for the floorplan in a level l mesh such that any two dies in the same set are not in conflict.

Proof: The lemma is true for $l=1$ since the dies in $R(1,1)$ and $R(1,3)$ are not in conflict and the dies in $R(1,2)$ and $R(1,4)$ are not in conflict.

Suppose that the lemma is true for $l=i$, for $l=i+1$, the reticle is first divided into four regions $R(1,1)$, $R(1,2)$, $R(1,3)$, and $R(1,4)$, and each region is further divided into a level i mesh. Since the lemma is true for $l=i$, there are at most 2^i conflict-independent sets for each of the four regions. We denote the kth conflict-independent set of region $R(1,j)(j=1,\ldots,4)$ as $S(1,j,k)$. Since any die in $R(1,1)$ is not in conflict with any die in $R(1,3)$, we can have the 2^i combined conflict-independent sets $S(1,1,k) \bigcup S(1,3,k)(k=1,\ldots,2^i)$. Similarly, we can have another 2^i combined conflict-independent sets $S(1,2,k) \bigcup S(1,4,k)(k=1,\ldots,2^i)$. Therefore, there are at most 2^{i+1} conflict-independent sets. ■

It is obvious that all copies of the dies in the same conflict-independent set can be simultaneously sliced out since they are not in conflict. If we assume that only the dies of one conflict-independent set are obtained for each wafer, the "wafer requirement" for a conflict-independent set S is $\mathrm{MAX}_{D\in S}(\lceil N(D)/Q(D) \rceil)$, where $N(D)$ is the volume requirement of die D and $Q(D)$ is the number of copies of die D per wafer.[3] The total wafer requirement is the sum of the wafer requirements of all the conflict-independent sets.

We give a generic simulated annealing placement algorithm for finding reticle floorplans in Fig. 3. The algorithm starts with the floorplan, with each die randomly placed in the 4^l regions as its initial placement. First, the algorithm tries to minimize the floorplan area in order to find a feasible solution. After a feasible solution is found, the objective switches to minimizing the "total wafer requirement," whose calculation is specified in Lemma 1 and the paragraph following the proof of Lemma 1. Note that for speeding up the algorithm, quadrisection floorplan evaluation does not include the dicing plan and the shot-map

[2]It is sufficient to choose $l=3$ in practice since the case of putting more than 64 dies in one reticle is very rare, although we may choose $l = \lceil \log_4 \text{ number of dies} \rceil$ if the number of dies is larger than 64.

[3]In order to speed up the wafer-cost evaluation in the floorplanning step, we fix the wafer center at point (0, 0) and set $Q(D)$ to the number of dies D on the wafer (see Section V).

Input: Dimensions of n dies, β: $0 \leq \beta < 1$

Output: Reticle floorplan and wafer dicing plan

1. Construct the hierarchical quadrisection floorplan mesh
2. Assign the n dies to regions at random
3. If (floorplan width and height smaller than maximum reticle dimensions) then $FoundFeasible \leftarrow True$
4. Else $FoundFeasible \leftarrow False$
5. While (not converged and # of moves $<$ $Move\_Limit$)
6. Pick a move at random
7. If (floorplan width and height smaller than maximum reticle dimensions) then
8. $FoundFeasible \leftarrow True$; $\delta \leftarrow$New Wafer requirement - Old Wafer Requirement
9. Else, if ($FoundFeasible = False$) then $\delta \leftarrow$ New Area - Old Area, else $\delta \leftarrow \infty$
10. If ($\delta < 0$) then accept the move, else accept the move with probability $e^{-\frac{\delta}{T}}$
11. $T \leftarrow \beta T$
12. While ($\exists$ a die that can be inserted)
13. Sort all dies that can be inserted in descending order of $N(D)/A(D)$
14. For each die D_i do
15. If (D_i can be inserted) then insert it

Fig. 3. Hierarchical quadrisection floorplan.

definition. At each step, we find a neighbor solution based on the following moves:

1) region exchange move, which exchanges the dies in two regions if at least one of the regions contains a die;
2) orientation move, which rotates one die by 90° if the width and height of the die are different.

Each generated solution is evaluated and kept with a probability dependent on the current temperature (see Fig. 3). Finally, we may inset additional copies of dies if the reticle dimension is not increased (lines 12–15).[4,5]

IV. MDP Dicing

The following problem has been introduced in [7].

SSWDP: Given a reticle floorplan with dies $\mathcal{D} = \{D_1, \ldots, D_n\}$, required production volume for each die $N(D_i)$, $i = 1, \ldots, n$, and positions of the reticle projections of the wafer, find the minimum number of wafers N_w and the corresponding dicing plan for each wafer such that the wafer-dicing yield is at least one.

In [7] and [9], the authors adopt SWDP assumption, which limits the solution space. The Iterative Augment and Search Algorithm (IASA) method proposed for single dicing plan (SDP) in [7] can be extended to solve MDP by placing N_w wafers into one "superwafer" whose row (column) number is N_w times the initial row (column) number, as shown in Fig. 4. However, the runtime will increase rapidly when N_w is large since we need to check all rows and columns of the "superwafer" in each iteration. The large runtime makes it unsuitable to be used in our proposed flow since the wafer shot-map definition step requires the accurate wafer-cost calculation for each candidate wafer-center location.

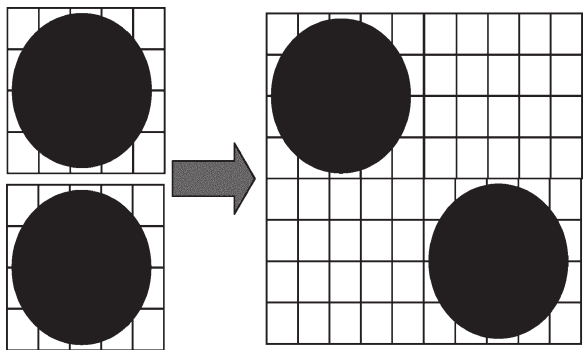


Fig. 4. Placing two wafers on one superwafer.

A. Integer Linear Program for Restricted MDPS

Xu *et al.* [14] assume that each wafer uses exactly one horizontal dicing plan and one vertical dicing plan for all projection rows/columns within a wafer. This assumption allows them to use a coloring-based heuristic that gives good results for test cases with a large volume requirement. In this section, we give an ILP formulation that allows finding optimal MDPs that are restricted in this way.

As in [14], two dies D and D' on a reticle are said to be in "dicing conflict" if they are either in horizontal dicing conflict or vertical dicing conflict. The "conflict graph" $R_c = (\mathcal{D}, E_c)$ is the graph with vertices corresponding to the dies and edges connecting pairs of dies in dicing conflict. A "maximum conflict independent set" is a subset of $\mathcal{D}$ that can be sliced out by a set of horizontal and vertical cutlines. We use $MCIS$ to denote the set of all maximal independent sets in the conflict

[4]Whether die D can be inserted is decided by finding a free room for D on the reticle. We place the left bottom corner of D and its 90° rotation at the corners of each die in the reticle and check whether D overlaps with other dies in the reticle.

[5]In practice, there is not too much empty space left in the reticle even if the number of dies is substantially smaller than the grid number of the mesh due to two reasons: 1) if there is no die in a region of level l, the region area is zero and 2) if a floorplan has too much empty area, its dimension will exceed the maximum reticle dimension, and this floorplan will be discarded.

graph.[6] For each independent set $C \in MCIS$, let f_c denote the number of wafers that use the dicing plan defined by C. MDP can then be formulated as the following integer linear program:

$$\text{Minimize} \quad N_w \qquad \text{(ILP1)}$$

subject to

$$\sum_{D \in C} Q(C,D) f_C \geq N(D) \qquad \forall D \in \mathcal{D}$$

$$\sum_{c} f_C = N_w$$

$$f_C \in \mathbb{Z}_+ \qquad \forall C \in MCIS$$

where $Q(C,D)$ is a constant that represents the number of copies of die D obtained from a wafer diced according to C. The ILP can be optimally solved in a short time since there are only $|MCIS|$ variables and $|\mathcal{D}|+1$ constraints. As shown in Section VI, the runtimes of ILP are within 0.03 s in all the experiments on industry test cases with up to 30 dies.

B. TLO Algorithm for MDP

Although the ILP method can solve the MDP problem quickly, its performance will be degraded for the small-volume-requirement cases. Extended IASA for MDP can produce a good solution but suffers from large runtime with large N_w. In order to rapidly find a near optimal solution for MDP, we propose the TLO heuristic shown in Fig. 5. We first solve ILP1 to obtain an upper bound on N_w. Then, we gradually reduce the number until the yield becomes smaller than one. In lines 4–8, we assume all rows (columns) of each wafer to be using the same horizontal (vertical) dicing plan. The dicing plan for each wafer is obtained by solving

$$\text{Minimize} \quad Y \qquad \text{(ILP2)}$$

subject to

$$N(D) - \sum_{D \in C} Q(C,D) f_c \leq y_D \qquad \forall D \in \mathcal{D}$$

$$\sum_{c} f_C = N_w$$

$$\sum_{D} y_D = Y$$

$$f_C \in \mathbb{Z}_+ \qquad \forall C \in MCIS$$

$$y_D \in \mathbb{Z}_+ \qquad \forall D \in \mathcal{D}$$

where Y is the total number of unsatisfied volume requirement and y_D is the number of unsatisfied volume requirement for die D. Since one maximal conflict independent set may belong to several maximal horizontal (vertical) independent sets, we use y_D as the weight of D and choose the maximal horizontal (vertical) independent set with the maximum total weight for each wafer. Then, we perform "row-and-column level" dicing plan replacement in lines 10–13 to improve the yield.[7] A "candidate pool" is employed to speed up the process. Since the wafer yield depends on the dies with the minimum dicing yield, the dicing plans that can slice out at least one of these dies are put into the candidate pool. Only the dicing plans in the candidate pool will be tried in each iteration. The candidate pool will be updated with change of minimum yield dies. However, this process is greedy, requiring yield increase with each dicing plan replacement. If die D does not belong to any chosen horizontal or vertical dicing plan, we need to simultaneously change a horizontal and a vertical dicing plan to obtain one copy of D and increase the yield. Therefore, a "cross-selection" process in lines 14–17 is used to choose the dicing plans for one row and one column simultaneously. Since the cross-selection process is extremely time consuming, we do it only for the center row and column of each wafer.

Input: *MHIS*, *MVIS*, *MCIS*
Output: N_w and dicing plan for N_w wafers
01. Solve ILP1 to obtain the N_w upper bound
02. while (yield$\geq$ 1)
03. $N_w - -$
04. Solve ILP2 and choose one set C for each wafer
05. Set the weight of each die D as y_D
06. **For** (each wafer)
07. Choose max horizontal (vertical) independent set
08. **While** (improve==true)
09. **While** (improve==true)
10. **For** (each row and column)
11. try other horizontal (vertical) dicing plans
12. **If** (wafer-dicing yield increases)
13. Replace the current dicing plan
14. **For** (the center row and column of each wafer)
15. *Simultaneously* try other pairs of dicing plans
16. **If** (wafer-dicing yield increases)
17. Replace the current dicing plan

Fig. 5. TLO heuristic.

V. Wafer Shot-Map Definition

In the previous section, we have fixed reticle images in order to reduce the problem complexity. However, if we allow the reticle images position to freely move on the wafer, then the wafer

[6] $MCIS$ can be found as follows: We denote the $MCIS$ for i dies as $MCIS(i)$.

1) Sort all dies according to $\max\_x$.
2) $MCIS(1) \leftarrow \{D_1\}$.
3) For $(i = 2; i \leq n; i++)$.
4) Find the last die D_j that satisfies $\max\_x(D_j) < \min\_x(D_i)$.
5) Add D_i to every set in $MCIS(j)$ and $MCIS(i) \leftarrow MCIS(j) \cup MCIS(i-1)$.

[7] In the process of yield and wafer-cost evaluation, we may take the dicing operation setup cost and lithography cost into consideration. Here, yield improvement is equal to total manufacturing cost reduction.

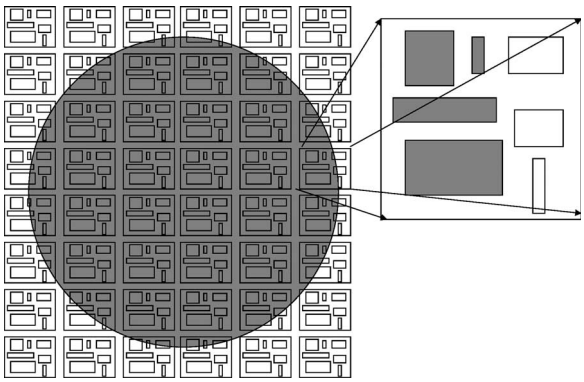


Fig. 6. Periodic shot-map with a dark circular wafer. A partially printed reticle contains completely dark printed projects.

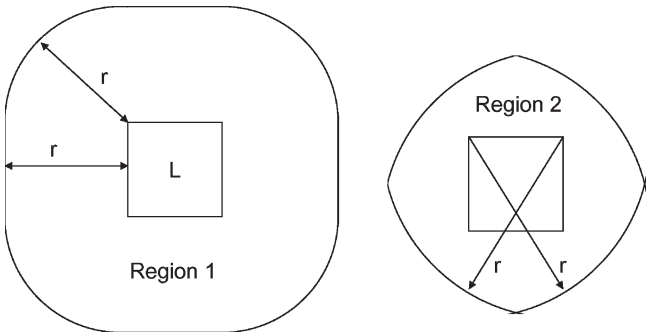


Fig. 7. Regions 1 and 2 for projection L.

Input: wafer radius r, reticle dimensions, one projection L
Output: wafer center location minimizing manufacturing cost
1. $L_0 \leftarrow L$
2. **For** (level =0; level $< l$; level++)
3. Divide L_0 into $k \times k$ uniformly-spaced grid
4. **For** (all the grids)
5. choose the grid center p as the wafer center
6. **If** ($F(p)$ not included in any stored feasible set)
7. calculate N_w with TLO, store $F(p)$
8. Find the grid g with the minimum N_w
9. $Min\_N_w \leftarrow N_w$; $L_0 \leftarrow g$;

Fig. 8. Hierarchical wafer shot-map definition algorithm.

cost can be reduced even more. The wafer shot-map definition step, which determines the position of reticle images printed on the wafer, was previously investigated for general wafers to maximize the wafer yield [6]. However, it was ignored in the previous papers in MPW context. In both [7] and [9], the wafer is modeled as a rectangular array of projects, which is not true for actual round wafers. This simplification may lead to wrong dicing yield estimation since: 1) the projection rows (columns) do not have equal contributions to the wafer-dicing yield—the rows/columns near the center contain more reticle images—and 2) fully printed dies within partial reticle projection are ignored. For a round wafer with radius r and center (x_0, y_0), die image D is "on wafer" if and only if $(x - x_0)^2 + (y - y_0)^2 \leq r^2$ for all $(x, y) \in D$ (see Fig. 6). Given a rectangular reticle image, a shot-map is a regular tiling of the plane with identical copies of the reticle. The corresponding problem of wafer position with respect to shot-map is formulated as follows.

1) Wafer Shot-Map Definition Problem (WSMDP): Given a projection plane and wafer radius r, find the position of the wafer that minimizes the number of wafers required to meet the given production volumes.

The periodic property of the projection plane imply the following lemma.

Lemma 2: The optimal solution of WSMDP can be achieved when the location of the wafer center is restricted to be within one reticle projection L.

Proof: Let the reticle width and height be R_w and R_H, respectively, and the optimal solution of WSMDP can be achieved when the wafer center is located at $(i \times R_w + x, j \times R_H + y)$, where i, j are integers and $0 \leq x < R_w, 0 \leq y < R_H$. It is obvious that for any copy of a die located in the wafer centered at $(i \times R_w + x, j \times R_H + y)$, there is a corresponding copy of the same die located in the wafer centered at (x, y) and vice versa. Therefore, the optimal solution can also be achieved when the wafer center is located at (x, y). ■

Therefore, the wafer center is constrained in one projection. The wafer-center location is further constrained by the following lemma:

Lemma 3: The optimal solution of WSMDP can be achieved when at least two die corners located on the circular boundary of the wafer and the dies having these corners are located within the wafer.

Proof: Suppose that the optimal solution of WSMDP can be achieved when the wafer center is located at (x, y). Let S be the set of the four corner coordinates of all dies on the wafer. One fact is that one die is on the wafer if and only if its four corners are in the wafer, so the solution remains optimal if all the points in S are in the wafer. If no points in S are on the wafer boundary, then for any point $(x_i, y_i) \in S$, $(x_i - x)^2 + (y_i - y)^2 < r^2$. Let $t_i = x_i - x + \sqrt{r^2 - (y_i - y)^2}$. It is easy to prove that $t_i > 0$ and $(x_i - x - t_{i,1})^2 + (y_i - y)^2 = r^2$ (intuitively, this equals moving the wafer center to the right by t_i to make the point (x_i, y_i) on the boundary). Let t be the smallest value of all t_i values, and move the wafer center to $(x + t, y)$. Then, at least one point in S will be located on the wafer boundary. In addition, if any point $(x_i, y_i) \in S$ is out of the wafer, then $(x_i - x - t)^2 + (y_i - y)^2 > r^2$, $(x_i - x - t)^2 + (y_i - y)^2 > (x_i - x)^2 + (y_i - y)^2 \Rightarrow r^2 - (y_i - y)^2 < (x_i - x - t)^2$, and $t > 2(x_i - x) \Rightarrow t_i < x_i - x + \sqrt{x_i - x - t)^2} = t$. However, this is impossible since t is the smallest value. Therefore, all points in S are still on the wafer when the wafer center is located at $(x + t, y)$.

Next, if there is only one die corner (x_1, y_1) on the boundary, we can perform coordinate transformation such that (x, y) in the original coordinates becomes $((x + t - x_1)\cos\phi + (y - y_1)\sin\phi,\ (y - y_1)\cos\phi - (x + t - x_1)\sin\phi)$ in the new coordinates, where $\phi = \arctan(y - y_1/x + t - x_1)$. It is easy to prove that points (x_1, y_1) and $(x + t, y)$

TABLE I
CMP TEST-CASE PARAMETERS

Cases	# dies	Total volume	Max Vol.	Min Vol.	Die area(cm^2)	$\|MCIS\|$	$\|MHIS\|$	$\|MVIS\|$
Ind 1	12	330	40	25	1.13	19	32	36
Ind 2	14	275	25	6	1.36	19	15	50
Ind 3	24	775	67	25	1.82	56	280	200
Ind 4	31	755	30	8	1.62	242	450	1008
Ind 5	14	250	25	12	0.86	18	63	40
Ind 6	24	625	35	25	2.26	127	588	1080

TABLE II
RETICLE FLOORPLAN RESULTS FOR SIX INDUSTRY TEST CASES. CMP IS THE ORIGINAL INDUSTRY FLOORPLAN USED IN CMP, "IASA+SA" IS THE SDP-DRIVEN FLOORPLANNER USED IN [9], AND HQ IS OUR PROPOSED HIERARCHICAL QUADRISECTION FLOORPLAN ALGORITHM

Cases	# part	CMP		IASA+SA			HQ		
		N_w	area	N_w	area	CPU(s)	N_w	area	CPU(s)
Ind 1	1	3	1.13	3	1.58	24.2	3	1.42	0.00
Ind 2	1	3	1.36	3	1.83	39.2	2	1.65	0.00
Ind 3	1	4	1.82	7	1.96	1031	4	2.26	0.01
Ind 4	1	4	1.62	5	2.72	2351	4	1.82	0.01
Ind 5	1	2	0.86	2	1.77	51.7	2	1.19	0.00
Ind 6	1	6	2.26	6	3.60	795	5	2.66	0.01
Total		22		26			20		
Red.(%)				-18.2			9.1		
Ind 1	2	2	1.13	2.5	1.58	24.2	1.5	1.42	0.00
Ind 2	2	2	1.36	2	1.83	39.2	1.5	1.65	0.00
Ind 3	2	3	1.82	4	1.96	1031	3	2.26	0.01
Ind 4	2	3.5	1.62	3.5	2.72	2351	2.5	1.82	0.01
Ind 5	2	1.5	0.86	1.5	1.77	51.7	1.5	1.19	0.00
Ind 6	2	5	2.26	6	3.60	795	3	2.66	0.01
Total		17		19.5			13		
Red.(%)				-14.7			23.5		
Ind 1	4	1.5	1.13	1.75	1.58	24.2	1.25	1.42	0.00
Ind 2	4	1.5	1.36	1.75	1.83	39.2	1.5	1.65	0.00
Ind 3	4	2.75	1.82	4	1.96	1031	2.75	2.26	0.01
Ind 4	4	2.75	1.62	3.25	2.72	2351	2.25	1.82	0.01
Ind 5	4	1	0.86	1.25	1.77	51.7	1	1.19	0.00
Ind 6	4	4.5	2.26	4.5	3.60	795	3	2.66	0.01
Total		14		16.5			11.75		
Red.(%)				-17.8			16.1		

become (0, 0) and $(r, 0)$, respectively, in the new coordinates. Let (x'_i, y'_i) be the new coordinates of the points $(x_i, y_i) \in S$, $(x'_i - r)^2 + y'^2_i < r^2 \Rightarrow x'^2_i + y'^2_i < 2rx_i$. Let $\theta_i = \arcsin(\sqrt{x'^2_i + y'^2_i}/2r) - \arcsin(x'_i/\sqrt{x'^2_i + y'^2_i})$. It is easy to prove that $\theta_i < 0$ and $(x'_i - r\cos\theta_i)^2 + (y'_i - r\sin\theta_i)^2 = r^2$ (intuitively, this step equals rotating the wafer center around point (0, 0) by θ_i to make the point (x'_i, y'_i) on the boundary). Let θ be the largest value of all θ_i values, and rotate the wafer center around point (0, 0) by θ to $(r\cos\theta, r\sin\theta)$. Then, at least two points in S will be located on the wafer boundary: (0, 0) and point (x'_i, y'_i), whose θ_i is θ. If any point $(x'_i, y'_i) \in S$ is out of the wafer, i.e., $(x'_i - r\cos\theta)^2 + (y'_i - r\sin\theta)^2 > r^2 \Rightarrow \sqrt{x'^2_i + y'^2_i}/2r > \sin(\theta + \arcsin(x'_i/\sqrt{x'^2_i + y'^2_i})) \Rightarrow \theta_i > \theta$. However, it is

TABLE III
WAFER DICING RESULTS FOR SIX TEST CASES. IASA IS THE ALGORITHM PROPOSED IN [7], E-IASA IS OUR EXTENDED IASA HEURISTIC, ILP IS THE PROPOSED INTEGER LINEAR PROGRAMMING APPROACH, AND TLO REFERS TO OUR TLO ALGORITHM

Cases	# part	IASA		E-IASA		ILP		TLO	
		N_w	CPU(s)	N_w	CPU(s)	N_w	CPU(s)	N_w	CPU(s)
Ind 1	1	4	0.9	3	21.4	6	0.0	3	0.14
Ind 2	1	3	0.9	3	20.9	5	0.01	3	0.18
Ind 3	1	9	4.8	5	617	5	0.03	4	4.59
Ind 4	1	7	26.1	4	1631	8	0.03	4	73.6
Ind 5	1	2	1.9	2	15.5	4	0.0	2	0.21
Ind 6	1	13	13.2	6	2634	8	0.00	6	3.57
Total		38		23		36		22	
Red.(%)				39.5		5.3		42.1	
Ind 1	2	3	2.6	2.5	37.0	3	0.0	2	0.05
Ind 2	2	3	2.3	2	18.8	2.5	0.0	2	0.06
Ind 3	2	7	16.8	4.5	1485	3.5	0.01	3	3.98
Ind 4	2	5	76.9	3.5	3041	4	0.02	3.5	0.76
Ind 5	2	2	5.7	1.5	17.7	2	0.0	1.5	0.21
Ind 6	2	9	37.4	5	4457	5	0.02	5	0.04
Total		29		18.5		20		17	
Red.(%)		23.7		51.3		47.4		55.3	
Ind 1	4	2	6.5	1.75	31.4	1.75	0.01	1.5	0.02
Ind 2	4	2	6.3	1.75	29.9	2.25	0.0	1.5	0.02
Ind 3	4	7	44.8	3.75	2246	3	0.01	2.75	0.17
Ind 4	4	4	225	3	6176	3.25	0.03	2.75	0.72
Ind 5	4	1	13.6	1	17.9	1	0.0	1	0.01
Ind 6	4	9	91.6	4.75	10606	4.75	0.02	4.5	0.82
Total		25		16		16		14	
Red.(%)		34.2		57.9		57.9		63.2	

impossible since θ is the largest value. Therefore, all points in S are still in the wafer; then, the wafer center is located at $(r\cos\theta, r\sin\theta)$. ■

If two points (x_1, y_1) and (x_2, y_2) on the wafer boundary are known, the wafer center is located at either $((x_1 + x_2 - (y_1 - y_2)t)/2,\ (y_1 + y_2 + (x_1 - x_2)t)/2)$ or $((x_1 + x_2 + (y_1 - y_2)t)/2,\ (y_1 + y_2 - (x_1 - x_2)t)/2)$, where $t = \sqrt{4r^2/((y_1 - y_2)^2 + (x_1 - x_2)^2) - 1}$.

As in Fig. 7, when the wafer center is constrained in one projection L, all dies within region 1 can be on the wafer. All dies within region 2, which is the intersection of four cycles with radius r and whose centers are located at the four corners of L, will be within the wafer no matter where within L the wafer center is. We define the set S as the set of dies which are in region 1 and not in region 2. From Lemma 3, it is sufficient to consider the points in L, which is determined by at least two corners of dies in S when the corners are on the wafer boundary. The number of these points is at most $O(|S|^2)$. An optimal solution can be achieved by checking all these points.

However, obtaining the optimal solution is impractical due to the large number of points to be checked (as shown in Table IV) and the relatively long runtime of wafer-cost calculation procedure TLO. Therefore, we propose a hierarchical wafer shot-map definition algorithm, as summarized in Fig. 8, which only calls TLO dicing procedures at a few hierarchically selected locations. We first divided the projection L into several grids and then run TLO while wafer centering at each grid center. The "best" grid will be chosen for the next iteration. The following trick is employed in the algorithm to speed up the process. For each candidate wafer-center location p, there is a "feasible set" of dies $F(p)$ on the wafer when the wafer center is at p. Obviously, the wafer cost will not be reduced if $F(p)$ is a subset of one feasible set whose wafer cost is already calculated. We store all feasible sets whose wafer costs are calculated for comparison. In line 6, if $F(p)$ is included in

TABLE IV
COST EFFICIENCY OF WAFER SHOT-MAP DEFINITION STEP FOR SIX INDUSTRY TEST CASES. "# points" IS THE NUMBER OF POSSIBLE WAFER CENTER LOCATIONS TO BE CHECKED TO OBTAIN AN OPTIMAL SOLUTIONS. l IS THE NUMBER OF LEVELS, AND k IS THE GRID SIZE USED IN EACH LEVEL

Cases	# part	# points	Fixed		$l=1,k=10$		$l=1,k=100$		$l=3,k=10$	
			N_w	CPU(s)	N_w	CPU(s)	N_w	CPU(s)	N_w	CPU(s)
Ind 1	1	1023940	3	0.14	3	13.7	2	2496	2	31.7
Ind 2	1	1226510	3	0.18	2	32.9	2	2790	2	68.5
Ind 3	1	2513061	4	4.59	4	442.7	4	39763	4	1069
Ind 4	1	5433742	4	73.6	4	7455	4	635342	4	18674
Ind 5	1	1717290	2	0.21	2	20.9	2	1762	2	48.3
Ind 6	1	2144469	6	3.57	5	1024	5	96521	5	2581
Total			22		20		19		19	
Red.(%)					9.1		13.6		13.6	
Ind 1	2	281189	2	0.05	2	4.87	2	372	2	9.73
Ind 2	2	341819	2	0.06	2	5.73	2	461	1.5	46.8
Ind 3	2	685663	3	3.98	3	376	3	28909	3	937
Ind 4	2	1491077	3.5	0.76	3	1937	3	93402	3	4852
Ind 5	2	463405	1.5	0.21	1.5	17.6	1	3594	1	79.8
Ind 6	2	605159	5	3.57	4	479	4	38721	4	971
Total			17		15.5		15		14.5	
Red.(%)					8.8		11.8		14.7	
Ind 1	4	56205	1.5	0.02	1.5	1.76	1.25	634	1.25	16.8
Ind 2	4	75445	1.5	0.02	1.25	3.51	1.25	337	1	39.1
Ind 3	4	152140	2.75	0.17	2.75	19.4	2.5	45827	2.5	673
Ind 4	4	316718	2.75	0.72	2.5	173	2.5	14523	2.5	483
Ind 5	4	98545	1	0.01	1	0.97	0.75	1877	0.75	13.5
Ind 6	4	133111	4.5	0.82	4	567	4	30469	4	1235
Total			14		13		12.25		12	
Red.(%)					7.1		12.5		14.3	

any stored set, p will be skipped to avoid redundant wafer-cost calculation.

VI. EXPERIMENTAL RESULTS

We used six industry test cases from CMP [16] to evaluate the performance and scalability of the proposed algorithms, each having between 12 and 31 dies with varying sizes and production volume requirements. For the wafer shot-map and wafer-dicing problem, we used the reticle floorplan of the actual industry MPW runs, which were manually designed by an experienced engineer. The basic parameters of the six test cases are listed in Table I.

1) Reticle Floorplanning: We implemented our hierarchical quadrisection floorplan algorithm in C++. The maximum reticle dimension is set as 2 cm. After the placement, we use a fixed wafer shot-map and TLO dicing method to generate the dicing plans for all the wafers. The reticle floorplan results are summarized in Table II. Here, "CMP" denotes the original industry floorplan used by CMP, "IASA+SA" is the SDP-driven floorplanner used in [7], and HQ is our proposed hierarchical quadrisection floorplan algorithm. The results show that our proposed hierarchical quadrisection floorplan can reduce wafer cost by 9.1%, 23.5%, and 16.1% for one part, two parts, and four parts compared with the original industry floorplan. On the other hand, IASA+SA increases the wafer cost by 18.2%, 14.7%, and 17.8%, which indicates that IASA+SA is not a good choice for MDP on round wafers.

2) Wafer Dicing: We implemented the wafer-dicing algorithms in the C++ language. We set the wafer diameter as 6 in and use a fixed wafer shot-map for all test cases. The number of wafers used N_w and runtime of four methods are shown in Table III, where IASA is the SDP method used in [7], E-IASA is the extended IASA in Section IV, ILP is the ILP-restricted MDP method specified in Section IV-A, and TLO is the two-level MDP optimization method proposed in

Section IV-B. Each method was run without any wafer partition and with wafer partition into two or four parts prior to dicing. The results show that compared with the original IASA with one part, the wafer cost can be reduced by 34.2% by using four parts. E-IASA can reduce the wafer cost by 39.5% for one part at the expense of long runtime. ILP can reduce the cost by 5.3% for one part and 57.9% for four parts. Therefore, ILP is more efficient for multiple-part dicing. TLO achieves the best solution quality in a short time. TLO reduces the wafer cost by 63.2% for four parts.

3) Wafer Shot-Map Definition: Our algorithm for WSMDP was implemented in C++.

The wafer cost and runtime results are summarized in Table IV. "# points" is the number of possible wafer-center locations that need to be checked to obtain an optimal solution. l is the number of levels, and k is the grid size used in each level. Compared with the fixed shot-map, the wafer cost can be reduced by 9.1% by using a 10×10 grid at the expense of increased runtime. Runtime will significantly increase when N_w is reduced since there will be more dicing plan replacement iterations in TLO procedure. Although using a 100×100 grid can further reduce the wafer cost (13.6%), the runtime becomes impractical (more than $100\times$). However, a good trade-off between solution quality and runtime can be achieved by using our proposed hierarchical wafer shot-map definition algorithm with $l = 3$ and $k = 10$. The wafer cost is reduced by 13.6%, while the runtime is within $2.5\times$ compared with using 10×10 grid.

VII. Conclusion and Future Work

In this paper, we proposed improved algorithms for multiproject reticle floorplanning, wafer shot-map definition, and wafer dicing. Experiments on industry test cases show that our methods outperform significantly previous methods in the literature as well as floorplans manually designed by experienced engineers. Our methods can also be extended to handle additional constraints such as die-alignment constraints imposed by the use of die-to-die mask inspection [13] by merging two copies of a die in a single "superdie." With ongoing work, we investigate the use of multiple die copies on the reticle and multilayer reticles for further reductions in the manufacturing cost of given die production volumes.

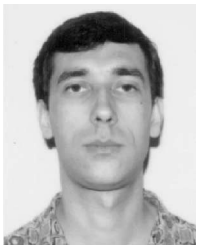
References

[1] M. Andersson, C. Levcopoulos, and J. Gudmundsson, "Chips on wafers," in *Proc. WADS*, Aug. 2003, pp. 412–423.
[2] ——, "Chips on wafers," *Comput. Geom.-Theory Appl.*, vol. 30, no. 2, pp. 95–111, 2005.
[3] A. Balasinski, "Multi-layer and multi-product masks: Cost reduction methodology," in *Proc. 24th BACUS Symp. Photomask Technol.*, 2004, vol. 5567, pp. 351–359.
[4] A. Caprara, A. Lodi, and M. Monaci, "An approximation scheme for the two-stage, two-dimensional bin packing problem," in *Lecture Notes in Computer Science*, vol. 2337. New York: Springer-Verlag, 2002, pp. 315–328.
[5] S. Chen and E. C. Lynn, "Effective placement of chips on a shuttle mask," *Proc. SPIE*, vol. 5130, pp. 681–688, 2003.
[6] C. Chien and J. Deng, "Optimization of wafer exposure patterns using a two-dimensional cutting algorithm," *Int. Trans. Oper. Res.*, vol. 8, no. 5, pp. 535–545, 2001.
[7] A. B. Kahng, I. I. Măndoiu, Q. Wang, X. Xu, and A. Zelikovsky, "Multi-project reticle floorplanning and wafer dicing," in *Proc. Int. Symp. Phys. Des.*, Apr. 2004, pp. 70–77.
[8] A. B. Kahng, I. I. Măndoiu, X. Xu, and A. Zelikovsky, "Yield-driven multi-project reticle design and wafer dicing," *Proc. SPIE*, vol. 5992, pp. 1247–1257, 2005.
[9] A. B. Kahng and S. Reda, "Reticle floorplanning with guaranteed yield for multi-project wafers," in *Proc. Int. Conf. Comput. Des.*, Oct. 2004, pp. 106–110.
[10] R. D. Morse, "Multiproject wafers: Not just for million dollar mask sets," *Proc. SPIE*, vol. 5043, pp. 100–113, 2003.
[11] M.-C. Wu and R.-B. Lin, "A comparative study on dicing of multiple project wafers," in *Proc. Symp. VLSI (ISVLSI)*, 2005, pp. 314–315.
[12] ——, "Reticle floorplanning and wafer dicing for multiple project wafers," in *Proc. Int. Symp. Quality Electron. Des.*, 2005, pp. 610–615.
[13] G. Xu, R. Tian, D. F. Wong, and A. Reich, "Shuttle mask floorplanning," *Proc. SPIE*, vol. 5256, pp. 185–194, 2003.
[14] G. Xu, R. Tian, D. Z. Pan, and M. D. F. Wong, "A multi-objective floorplanner for shuttle mask optimization," *Proc. SPIE*, vol. 5567, pp. 340–350, 2004.
[15] ——, "CMP aware shuttle mask floorplanning," in *Proc. ASPDAC*, 2005, pp. 1111–1114.
[16] Circuits Multi-Projets (CMP). [Online]. Available: http://cmp.imag.fr/

Andrew B. Kahng received the A.B. degree in applied mathematics from Harvard College, Cambridge, MA, and the M.S. and Ph.D. degrees in computer science from University of California, San Diego (UC San Diego).

From 1989 to 2000, he was a member of the University of California, Los Angeles (UCLA) Computer Science Faculty. He is currently a Professor of CSE and ECE with UC San Diego. Since 1997, he has defined the physical design roadmap for the International Technology Roadmap for Semiconductors (ITRS), and from 2000 to 2003, he chaired the U.S. and international working groups for design technology for the ITRS. He has been active in the MARCO Gigascale Silicon Research Center since its inception. He has published more than 300 papers in the VLSI CAD literature. His research interests include physical design and performance analysis of VLSI, the VLSI design manufacturing interface, combinatorial and graph algorithms, and large-scale heuristic global optimization.

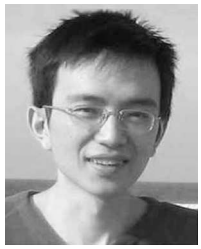
Dr. Kahng was also the Founding General Chair of the ACM/IEEE International Symposium on Physical Design and a Cofounder the ACM Workshop on System-Level Interconnect Planning. He received four Best Paper Awards and an NSF Young Investigator Award.



Ion I. Măndoiu received the M.S. degree from the University of Bucharest, Bucharest, Romania, in 1992, and the Ph.D. degree from Georgia Institute of Technology, Atlanta, in 2000, both in computer science.

From 2000 to 2003, he was a Postdoctoral Researcher and then a Research Scientist with the University of California, Los Angeles, and the University of California, San Diego. He is currently an Assistant Professor with the Department of Computer Science and Engineering, University of Connecticut, Storrs. He has published more than 50 papers in international journals and refereed conference proceedings. His research interests include the design and analysis of exact and approximation algorithms for NP-hard optimization problems, particularly, in the areas of VLSI and biochip design automation, bioinformatics, and ad hoc wireless networks.

Dr. Măndoiu is a member of the Association for Computing Machinery and the International Society for Computational Biology. He is a founding Cochair of the ACIS International Workshop on Self-Assembling Wireless Networks (SAWN) and a Program Committee Cochair of the 2007 International Symposium on Bioinformatics Research and Applications (ISBRA). He also serves on the Editorial Board of the International Journal of Bioinformatics Research and Applications and is a Guest Editor for the IEEE/ACM TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS, the *International Journal of Wireless and Mobile Computing,* the IEEE TRANSACTIONS ON NANOBIOSCIENCE, and the *Journal of Universal Computer Science*. He is a recipient of the Best Paper Award at the joint Asia-South Pacific Design Automation/VLSI Design Conferences in 2003, the Best Poster Award at the Annual BACUS Symposium on Photomask Technology in 2005, and the U.S. National Science Foundation's Faculty Early Career Development (CAREER) Award in 2006.



Xu Xu (S'01) received the B.S. degree from the Special Class for Gifted Young, University of Science and Technology of China, Hefei, China, in 1998, and the Ph.D. degree from the Department of Computer Science and Engineering, University of California at San Diego, La Jolla, in 2006.

He was with Ammocore Technology, Inc., Santa Clara, CA, in 2002, and Synopsys, Inc., Mountain View, CA, in 2004. He joined Blaze DFM, Sunnyvale, CA, in 2006. He is the author of more than 20 refereed publications on VLSI physical design, DNA array design, statistical static timing analysis, and IC manufacturing cost minimization.

Alexander Z. Zelikovsky received the Ph.D. degree in computer science from the Institute of Mathematics, Belorussian Academy of Sciences, Minsk, Belarus, in 1989.

From 1989 to 1995, he was with the Institute of Mathematics, Kishinev, Moldova. From 1992 to 1995, he visited the University of Bonn, Bonn, Germany, and the Institut fur Informatik, Saarbrueken, Germany. He was a Research Scientist with the University of Virginia, Charlottesville, from 1995 to 1997, and a Postdoctoral Scholar with the University of California, Los Angeles, from 1997 to 1998. In 1999, he joined the Department of Computer Science, Georgia State University, Atlanta, where he is currently an Associate Professor. He is the author of more than 120 refereed publications. His research interests include VLSI physical layout design, ad hoc wireless networks, discrete and approximation algorithms, combinatorial optimization, and computational biology.