Multi-Project Reticle Floorplanning and Wafer Dicing[*]

Andrew B. Kahng, Ion Măndoiu[†], Qinke Wang, Xu Xu, and Alex Z. Zelikovsky[‡]

CSE Department, UC San Diego, La Jolla, CA 92093-0114

[†]CSE Department, University of Connecticut, 371 Fairfield Rd., Unit 1155, Storrs, CT 06269-1155

[‡]CS Department, Georgia State University, University Plaza, Atlanta, Georgia 30303

{abk,qiwang,xuxu}@cs.ucsd.edu, ion@engr.uconn.edu, alexz@cs.gsu.edu

ABSTRACT

Multi-project Wafers (MPW) are an efficient way to share the rising costs of mask tooling between multiple prototype and low production volume designs. Packing the different die images on a multi-project reticle leads to new and highly challenging floorplanning formulations, characterized by unusual constraints and complex objective functions. In this paper we study multi-project reticle floorplanning and wafer dicing problems under the prevalent side-to-side wafer dicing technology. Our contributions include practical mathematical programming algorithms and efficient heuristics based on conflict graph coloring which find side-to-side wafer dicing plans with maximum yield for a fixed multi-project reticle floorplan and given per-die maximum dicing margins. We also give novel shelf packing and simulated annealing reticle floorplanning algorithms for maximizing wafer-dicing yield. Experimental results show that our algorithms improve wafer-dicing yield significantly compared to existing industry tools and academic min-area floorplanners.

Categories and Subject Descriptors

B.7.2 [**Integrated Circuits**]: Design Aids; J.6 [**Computer Applications**]: Computer-Aided Engineering—*Computer-Aided Manufacturing*; F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems

General Terms

Algorithms, Design, Economics, Experimentation, Theory

Keywords

Multi-Project Wafers, Reticle Design, Wafer Dicing

[*]Work supported in part by Cadence Design Systems, Inc., the MARCO Gigascale Silicon Research Center, NSF Grant CCR-9988331, and by the State of Georgia's Yamacraw Initiative.

1. INTRODUCTION

As VLSI feature size continues to shrink in the sub-wavelength regime, mask costs are predicted to skyrocket, driven up by the pervasive use of such advanced Reticle Enhancement Technologies (RET) as Optical Proximity Correction (OPC) and Phase Shifting Masks (PSM) [1, 2, 3]. Indeed, mask cost is predicted to more than double at each technology node and reach $10 million by the end of the decade [3]. These high mask costs push prototyping and low volume production designs at the limit of economic feasibility, increasingly motivating the use of Multiple Project Wafers (MPW), or "shuttle" runs, which allow customers to share the cost of mask tooling [4].

MPWs have been introduced in the late 1970s and early 1980s as government sponsored programs allowing students at hundreds of universities throughout the world to verify their design in silicon. Current commercial MPW services MOSIS [10] and CMP [16] are a direct outgrowth of these early programs (see [5] for a detailed account of MPW pioneering efforts and an overview of major infrastructure MPW services today). More recently, semiconductor foundries such as the Taiwanese Semiconductor Manufacturing Company (TSMC) have aggressively started to promote their own shuttle services [12].

Packing the different die images on a multi-project reticle leads to new and highly challenging floorplanning formulations, characterized by unusual constraints and complex objective functions. Recently, several approaches have been proposed in the literature for addressing the MPW reticle floorplanning problem. Chen and Lynn [7] considered in this context the problem of finding the minimum area slicing floorplan, with 90 degree chip rotation allowed. They give a "bottom-left fill" algorithm for constructing an initial solution, followed by enumeration based on B*-trees. Xu et al. [6] studied the MPW mask floorplanning under die-alignment constraints imposed by the use of die-to-die mask inspection. A grid-packing formulation for MPW mask floorplanning is proposed in [8], where the objective is to find a minimum area grided floorplan with at most one die per grid cell. A number of companies [13, 17] provide tools to design multi-project reticles. For example, K2's MaskCompose [17] has an automated die cluster builder facility, which groups a number of circuits of various sizes into a cluster, and optimizes the layout of the cluster to enable wafer scribing with minimum loss of die. Xyalis' GTmuch [11] provides an automated tool which can optimize mask layout to minimize area or scribe lines in order to minimize destroyed dies during die sawing.

In this paper we propose new multi-project reticle floorplanning and wafer dicing formulations accurately capturing the constraints of the prevalent side-to-side wafer dicing technology (Figure 1). Previously published methods either assume the use of much more expensive "slicing" dicing plans [7, 6], or implement simplistic

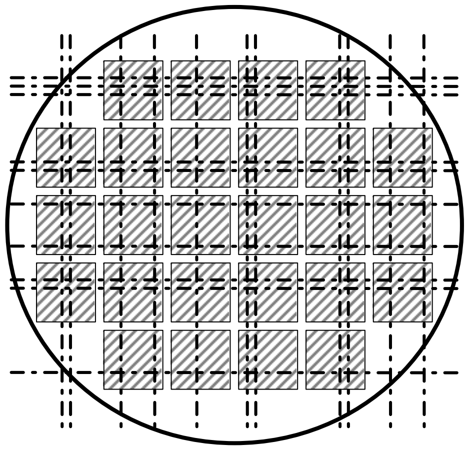

Figure 1: A multi-project wafer with a side-to-side wafer sawing plan consisting of dashed cuts. Each rectangle represents a multi-project reticle image.

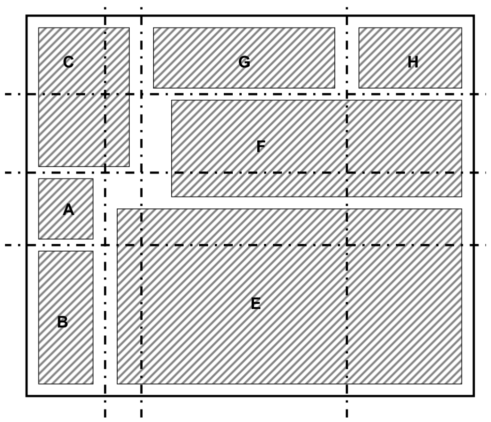

Figure 2: A multi-project reticle with a sample reticle dicing plan: the dashed cutting lines yield four usable dies: A,B,G, and H.

models of side-to-side wafer dicing, in which die-specific maximum bounds on dicing margins imposed by packaging decisions[1] are not taken into account [8].

The first problem considered is the side-to-side wafer dicing problem for a given reticle floorplan. Note that under this model all reticles in the same row (column) on the wafer will be sawed by the same set of horizontal (resp. vertical) cut lines, but one can vary dicing plans (i.e., set of cut lines) between different columns and rows (see Figure 1). Which dies on a reticle will be usable and which will be destroyed depends solely on the dicing plans for the corresponding row and column (and dicing margin constraints).

Side-to-Side Wafer Dicing Problem (SSWDP). Given a multi-project reticle and maximum dicing margin constraints, find wafer dicing plans minimizing the number of wafers necessary to manufacture required production volumes for each project.

Second, we consider MPW reticle floorplanning for side-to-side wafer dicing. Dicing margins are a particularly important degree of freedom in reticle design for side-to-side dicing. In a typical minimum-area reticle floorplan (Figure 2), side-to-side dicing can only extract a small number of dies simultaneously, since many other dies have to be destroyed when extracting one die via side-to-side cuts around its edges. Inserting a small amount of space between dies and tolerating small dicing margins can greatly reduce the number of destroyed dice and increase the wafer-dicing yield.

Reticle Floorplanning and Wafer Dicing Problem (RFWDP). Given a set of designs with required production volumes, find a reticle floorplan and wafer dicing plans minimizing the number of wafers necessary to manufacture required production volumes for each project.

Our contributions are as follows:

1. We give practical non-linear and integer linear programming formulations for SSWDP, as well as a very efficient SSWDP heuristic based on conflict graph coloring.
2. We propose a fast algorithm (shelf packing and shifting) for RFWDP, which can improve the wafer-dicing yield by 83.3% at the expense of 5.96% increase in area compared with the min-area simulated annealing placer, Parquet. Compared with the commercial MPW tool, GTmuch, the algorithm improves wafer yield by 37.7% while *reducing* the area by 3.3%.
3. We propose a new simulated annealing algorithm for RFWDP, which can further improve the wafer yield of shelf packing and shifting by 30.5% while increasing the area by only 3.3%.

The rest of the paper is organized as follows. We formalize the Side-to-side Wafer Dicing problem in Section 2, and give exact mathematical programming algorithms and efficient near-optimal heuristics in Section 3. In Section 4 we describe a simulated annealing algorithm for the Reticle Floorplanning and Wafer Dicing problem. Finally, we present experimental results in Section 5, and conclude in Section 6 with directions for future research.

2. SIDE-TO-SIDE WAFER DICING

In this section we formalize the Side-to-side Wafer Dicing problem and give a simple heuristic based on conflict graph coloring.

A novel feature of our formulation is that it allows dicing margins up to given per-die limits. Ideally, each chip should be diced from the wafer exactly along its sides. In practice, this requirement can be relaxed. A certain margin width along some or all die sides may be acceptable as long as packaging requirements are still met. We will assume that a maximum margin width *not exceeding half of the die width* is specified for each of the four sides of the die.[2] We say that a die is *legally diced* by a set of side-to-side cuts if (1) no cut intersects the interior of the die, and (2) the margin widths on all its sides do not exceed specified maximums.

We say that two dies D and D' on a reticle are in *vertical (resp. horizontal) dicing conflict* if no set of vertical (resp. horizontal) cuts can legally dice both D and D' according to the maximum margin width requirements. 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 reticle 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

[1]In order to minimize packaging cost, a die is normally packaged in the smallest package which fits it and provides satisfactory heat dissipation. The package cavity size determines the maximum allowable dicing margins.

[2]Minimum margin widths, if any, can be simply included in the die dimensions. The assumption that the maximum margin width does not exceed half of the die width is required for establishing the key property in Lemma 1. This assumption is rarely restrictive in practice.

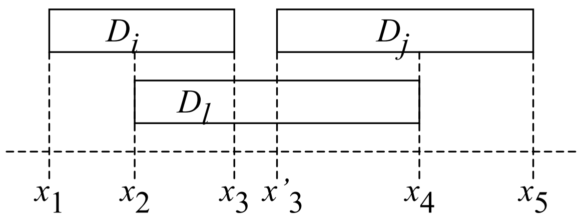

Figure 3: Horizontally non-overlapping dies D_i and D_j with a die D_l overlapping both of them. Since each margin width should be less than a half of die width, $(x_2 - x_1) < \frac{1}{2}(x_4 - x_2)$ and $(x_5 - x_4) < \frac{1}{2}(x_4 - x_2)$, therefore, $(x_5 - x_1) < 2(x_4 - x_2)$. On the other hand $(x'_3 - x_2) < \frac{1}{2}(x_5 - x'_3)$ and $(x_4 - x_3) < \frac{1}{2}(x_3 - x_1)$. Hence, $(x_4 - x_2) + (x'_3 - x_3) < \frac{1}{2}(x_5 - x_1) - \frac{1}{2}(x'_3 - x_3)$, and therefore $2(x_4 - x_2) < (x_5 - x_1)$, contradiction.

vertices in a graph is called independent if they are pairwise nonadjacent. The following lemma establishes the key property enabling the use of conflict graph coloring in finding legal dicing plans.

LEMMA 1. *Any set $\{D_1, \ldots D_k\}$ of independent dies in the vertical reticle conflict graph R_v can be simultaneously legally diced by a set of vertical side-to-side cuts.*

PROOF. Let D_i and D_j be two dies from the set that do not overlap horizontally. We claim that another die D_l cannot overlap horizontally with both D_i and D_j. Indeed, if D_l overlaps both D_i and D_j, then the pairs (D_i, D_l) and (D_l, D_j) cannot be both diced without leaving at least one margin wider than half of the corresponding die width (see Figure 3).

By the above claim, the set $\{D_1, \ldots D_k\}$ can be partitioned into subsets of pairwise-overlapping dies such that no two dies in different subsets overlap. Consider one such subset of pairwise-overlapping dies. Let x_L and x_R be the lowest and the highest x-coordinates of these dies, and let D_L and D_R be dies having these extreme coordinates. Let (x_1, x_2) be the projection on the horizontal axis of an arbitrary die D_i from the subset. Since D_i and D_L are not in conflict, $x_1 - x_L$ does not exceed the maximum left margin width for D_i. Similarly, since D_i and D_R are not in conflict, $x_R - x_2$ does not exceed the maximum right margin width for D_i. Thus, the two vertical cuts $x = x_L$ and $x = x_R$ legally dice all dies in the subset. Furthermore, by applying vertical cuts at the lowest and the highest x-coordinates of each subset of pairwise-overlapping dies, we get a set vertical cuts that legally dice all dies $D_1, \ldots D_k$. $\square$

A set of vertical and horizontal cuts is called a *reticle dicing plan*. Lemma 1 implies that any reticle dicing plan legally dices a set of dies which is independent in both the horizontal and vertical reticle conflict graphs.

A wafer contains a large number of reticles packed into a circular shape. Let $\mathcal{DW}$ denote the set of all dies on the wafer. Similarly to the reticle conflict graphs, we define the *vertical wafer conflict graph* $W_v = (\mathcal{DW}, EW_v)$, with edges corresponding to vertical conflicts, and the *horizontal wafer conflict graph* $W_h = (\mathcal{DW}, EW_h)$, with edges corresponding to horizontal conflicts. The analogously defined *wafer dicing plan* legally dices a set of dies which is independent in both the vertical and horizontal wafer conflict graphs.

Each die D on the given reticle should be produced in a specified volume $N(D_i)$. The *wafer dicing yield* of a wafer dicing plan P is defined as the minimum, over all dies $D \in \mathcal{D}$, of the number of legally diced copies of D divided by $N(D)$. The wafer dicing yield is inversely proportional to the number of wafers that should be manufactured in order to meet all production volumes, assuming that the same wafer dicing plan is used for all wafers.[3] This leads to the following formulation:

Side-to-Side Wafer Dicing Problem (SSWDP). Given a reticle with dies $\mathcal{D} = \{D_1, \ldots, D_n\}$, each die with specified maximum margin widths and required production volume, find a wafer dicing plan with maximum wafer dicing yield.

SSWDP can be reformulated as follows: Given two conflict graphs $R_v = (\mathcal{D}, E_v)$ and $R_h = (\mathcal{D}, E_h)$ on the same set of vertices $\mathcal{D}$ corresponding to the dies on the given reticle, assign an independent set of R_v (resp. R_h) to each column (resp. row) of reticles[4] on the wafer such that the yield of the resulting wafer dicing plan is maximum.

Next, we use this reformulation to develop a simple SSWDP heuristic based on conflict graph coloring. In next section we will describe mathematical programming algorithms for computing the optimal solution as well as a heuristic with improved practical performance.

Let c_v and c_h be the minimum number of colors needed to color the vertices of R_v and R_h, respectively. Clearly, each color in an optimum coloring of the vertical/horizontal reticle conflict graphs defines an independent set of vertices in the respective graph. The coloring-based heuristic uses in each column/row of reticles dicing plans derived from the color classes of 2 fixed optimal colorings of R_v and R_h. By Lemma 1, the corresponding independent sets of dies can be simultaneously legally diced by vertical/horizontal lines.

Assume that the wafer contains a rectangular array of $R \times T$ reticles. By assigning the independent set defined by each of the c_h (resp. c_v) colors in the optimal coloring of R_h (resp. R_v) to $\lfloor R/c_h \rfloor$ rows (resp. $\lfloor T/c_v \rfloor$ columns) of reticles on the wafer, we guarantee at least $\lfloor R/c_h \rfloor \times \lfloor T/c_v \rfloor$ legally diced dies for each design. Thus, we obtain:

THEOREM 1. *A constructive lower bound on the optimum wafer dicing yield is given by*

$$\frac{\max_{R,T} \lfloor R/c_h \rfloor \times \lfloor T/c_v \rfloor}{\max_{D \in \mathcal{D}} N(D)}$$

where the maximum in the numerator is taken over all pairs (R, T) such that an $R \times T$ rectangular array of reticles can be fit onto the wafer.

Remark. We note that in the important special case when maximum dicing margins are set to zero, R_v and R_h become interval graphs after merging vertices corresponding to dies that have identical vertical, respectively horizontal projections. It is well known that in this case c_v and c_h equal the maximum clique numbers in the respective interval graphs, and can thus be efficiently computed. In the general case R_v and R_h are not interval graphs (since, e.g., two dies with overlapping projections may still be legally diced simultaneously if maximum margins are not exceeded), and we do not

[3]For simplicity, we make this assumption for the rest of the paper. The algorithms in Section 3 can be easily extended to construct dicing plans for *lots of wafers*, thus allowing different dicing plans to be used for different wafers in a lot.

[4]Note that our formulation applies to the common case when a single reticle is uniformly stepped to manufacture the wafer. The methods extend to the case when multiple reticles are used in a repetitive regular pattern, e.g., equivalent to the uniform stepping of a "macro reticle" assembled from the smaller reticles.

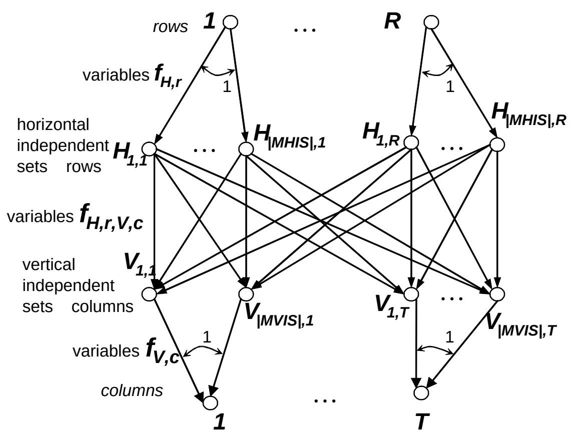

Figure 4: Relationship between variables of (ILP1).

know an efficient algorithm for computing exactly c_v and c_h. Efficiently computable lower-bounds on optimum c_v and c_h can be obtained by interval graph coloring by noting that the conflict graphs R_v and R_h are subgraphs of the interval graphs corresponding to the zero-margin case.

3. SSWDP ALGORITHMS

The independent sets used in the coloring-based heuristic given in previous section are generally not be maximal. Note that we may always extend the independent set used for a row/column to a maximal independent set in the corresponding reticle conflict graph, as this may only increase the wafer yield. In this section we give several algorithms that use exclusively maximal independent sets to define dicing plans for the rows/columns. All maximal independent sets can be enumerated based on Lemma 1. Indeed, there are at most 2^{2n} different sets of side-to-side cuts; in practice the number of maximal independent sets is much smaller (see Section 5).

Throughout this section we use the following notations:

- $R =$ number of rows of reticles on the wafer
- $T =$ number of columns of reticles on the wafer
- $MHIS =$ set of all maximal independent sets in the horizontal reticle conflict graph
- $MVIS =$ set of all maximal independent sets in the vertical reticle conflict graph

3.1 Non-Linear Programming Formulation

For each independent set $H \in MHIS$, let f_H denote the number of rows of reticles which use the dicing plan defined by H, and for each independent set $V \in MVIS$, let g_V denote the number of columns of reticles which use the dicing plan defined by V. Using an additional variable z to denote the wafer dicing yield of the resulting dicing plan, SSWDP can be formulated as the following non-linear program:

Maximize z (NLP)
subject to

$$\Big(\sum_{D\in H} f_H\Big)\Big(\sum_{D\in V} g_V\Big) \geq N(D)z, \quad \forall H \in MHIS, V \in MVIS$$
$$\textstyle\sum_H f_H = R$$
$$\textstyle\sum_V g_V = T$$
f_H, g_V nonnegative integers

3.2 Integer Linear Program Formulations

In next subsection we give a sequence of 3 integer linear programming formulations for SSWDP. The most compact of these formulations was found in our experiments to be more stable and often more efficient than the non-linear formulation given in previous subsection.

The first formulation uses 0/1 variables $f_{H,r}$ for every horizontal independent set $H \in MHIS$ and wafer row r and $f_{V,c}$ for every vertical independent set $V \in MVIS$ and wafer column c. In addition, we have 0/1 variables $f_{H,r,V,c}$ for every $H \in MHIS$, $V \in MVIS$, wafer row r and column c. The variables used in the ILP correspond to the edges of the graph in Figure 4. The formulation is the following:

Maximize z (ILP1)
subject to

$$\begin{aligned}
\sum_{H\in MHIS} f_{H,r} &= 1, & &\forall\, r\\
\sum_{V\in MVIS} f_{V,c} &= 1, & &\forall\, c\\
f_{H,r,V,c} &\leq (f_{H,r}+f_{V,c})/2, & &\forall\, H,r,V,c\\
\sum_{H,V} q^D_{H,V}\left(\sum_{r,c} f_{H,r,V,c}\right) &\geq N(D)z, & &\forall\, D\in\mathcal{D}\\
f_{H,r} &\in \{0,1\}, & &\forall\, H,r\\
f_{H,r,V,c} &\in \{0,1\}, & &\forall\, H,r,V,c\\
f_{V,c} &\in \{0,1\}, & &\forall\, V,c
\end{aligned}$$

Here and in the following ILPs, $q^D_{H,V}$ is a constant equal to 1 if die D belongs to both H and V, and to 0 otherwise.

Note that (ILP1) has $1+|MHIS|\cdot R+|MVIS|\cdot T+|MHIS|\cdot R\cdot|MVIS|\cdot T$ variables and $|\mathcal{D}|+R+T+|MHIS|\cdot R\cdot|MVIS|\cdot T$ constraints. The large number of variables comes mainly from the fact that we use a complete graph between each possible instance of a horizontal independent sets and each possible instance of a vertical independent set.

A more compact ILP is obtained next by avoiding explicit representation of all instances of the horizontal independent sets. Instead, we use arbitrary integer variables to represent the number of times a horizontal independent set is used. More precisely, we introduce integer variables $f_{H,V,c}$, and require that their value does not exceed $\sum_r f_{H,r}$, i.e., the number of rows for H is being used. The improved ILP formulation is the following (the graph representing the updated set of variables is given in Figure 5):

Maximize z (ILP2)
subject to

$$\begin{aligned}
\sum_{H\in MHIS} f_{H,r} &= 1, & &\forall\, r\\
\sum_{V\in MVIS} f_{V,c} &= 1, & &\forall\, c\\
f_{H,V,c} &\leq \sum_r f_{H,r}, & &\forall\, H,V,c\\
f_{H,V,c} &\leq Q\cdot f_{V,c} & &\forall\, H,V,c\\
\sum_{H,V} q^D_{H,V}\left(\sum_{c} f_{H,V,c}\right) &\geq N(D)z, & &\forall\, D\\
f_{H,r} &\in \{0,1\}, & &\forall\, H,r\\
f_{H,V,c} &\in \mathbb{Z}_+, & &\forall\, H,V,c\\
f_{V,c} &\in \{0,1\}, & &\forall\, V,c
\end{aligned}$$

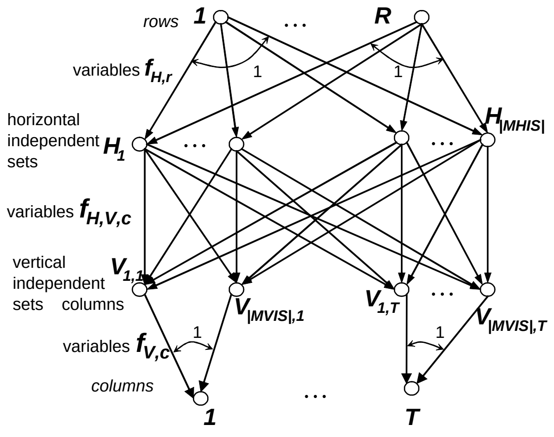

Figure 5: Relationship between variables of (ILP2).

Above, Q is an upperbound on the number of copies that can be produced from a single wafer for any of the projects (e.g., Q can be set to $R \cdot T$.)

(ILP2) has $1 + |MHIS| \cdot R + |MVIS| \cdot T + |MHIS| \cdot |MVIS| \cdot T$ variables and $|\mathcal{D}| + R + T + |MHIS| \cdot |MVIS| \cdot T$ constraints. For non-square reticles, $|MHIS| \cdot |MVIS| \cdot T$ can be replaced by $|MHIS| \cdot |MVIS| \cdot \min\{R, T\}$ by using the best between the above construction and the one obtained by swapping the role of horizontal and vertical cuts.

For the case when the number of maximal independent sets is large, it is possible to get an even more compact formulation using the following variables (see Figure 6):

- $f_{H,r} \in \{0,1\}$ for every horizontal independent set $H \in MHIS$ and wafer row r
- $f_{H,V} \in \mathbb{Z}_+$ for every horizontal independent set $H \in MHIS$ and every vertical independent set $V \in MVIS$
- $f^D_{V,c} \in \mathbb{Z}_+$ for every vertical independent set $V \in MVIS$, wafer column c, and die $D \in \mathcal{D}$
- $f_{V,c} \in \{0,1\}$ for every vertical independent set $V \in MVIS$ and wafer column c

Maximize z (ILP3)
subject to

$$\begin{aligned}
\sum_{H \in MHIS} f_{H,r} &= 1, && \forall\, r \\
\sum_{V \in MVIS} f_{V,c} &= 1, && \forall\, c \\
f_{H,V} &\le \sum_r f_{H,r}, && \forall\, H, V \\
f^D_{V,c} &\le \sum_H q^D_{H,V} f_{H,V}, && \forall\, D, V, c \\
\sum_{V,c} f^D_{V,c} &\ge N(D) z, && \forall\, D \\
f^D_{V,c} &\le Q \cdot f_{V,c} && \forall\, D, V, c \\
f_{H,r} &\in \{0,1\}, && \forall\, H, r \\
f_{H,V} &\in \mathbb{Z}_+, && \forall\, H, V \\
f^i_{V,c} &\in \mathbb{Z}_+, && \forall\, i, V, c \\
f_{V,c} &\in \{0,1\}, && \forall\, V, c
\end{aligned}$$

(ILP3) has $1 + |MHIS| \cdot R + |MVIS| \cdot T + |MHIS| \cdot |MVIS| + |MVIS| \cdot T \cdot |\mathcal{D}|$ variables and $|\mathcal{D}| + R + T + |MHIS| \cdot |MVIS| + 2|MVIS| \cdot T \cdot |\mathcal{D}|$ constraints. If $|MVIS| \cdot T > |MHIS| \cdot R$, we can use the the symmetric construction in which we swap the role of horizontal and vertical directions to further reduce the size of the ILP.

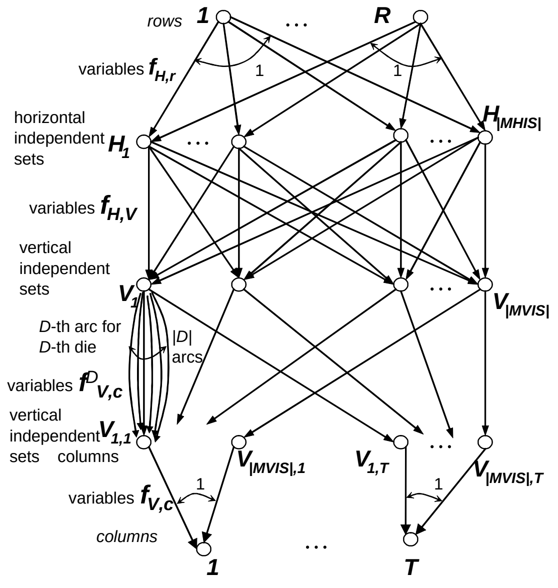

Figure 6: Relationship between variables of (ILP3).

3.3 Iterative Augment and Search Algorithm

In order to rapidly find a near optimal solution for SSWDP, we propose the heuristic shown in Figure 7 which starts by using conflict graph coloring to obtain dicing plans for the first c_h rows and c_v columns. Then we choose cut lines for a new row or column in each loop. In Lines 04-11, we first check the solutions which are obtained by changing the dicing plans for one row or column. If a solution with higher wafer-dicing yield is found, we replace the current dicing plan with the better dicing plan for that row or column. Then in Lines 12-15, we try to find the best dicing plan for a new row or column. We calculate the wafer-dicing yield z for dicing plans corresponding to all possible maximal horizontal/vertical independent sets, and find one with the largest yield z.

4. RFWDP ALGORITHMS

In this section, we focus on the following optimization problem: Given a reticle size, a set of dies and their sizes, required volume and maximum margin widths for each die, find a die floorplan within the boundary of the reticle (allowing die rotations) and a wafer dicing plan such that the resulting wafer-dicing yield is maximum.

4.1 Simulated Annealing Algorithm

In order to reduce the die loss during the die-sawing process, we propose the shelf-packing and shifting method detailed in Figure 8. A shelf is defined as a rectangle with fixed width. The height of a shelf is determined by the height of the highest die in the shelf and all dies must be placed on the bottom of the shelf. In this algorithm, we first sort all dies according to their height. Then we set the shelf width w, and insert dies one by one to the lower-most shelf in the shelf packing stage in lines 3-4. In the shifting stage shown in lines 5-12, we find the widest shelf l, and align the dies in other levels with dies in level l if there is room to push some dies to the right side. The above steps are repeated for all the possible shelf widths

```
Input: $MHIS$, $MVIS$, $c_h$, $c_v$
Output: dicing plan with the largest wafer-dicing yield $z$
01. Choose dicing plan for $c_h$ rows and $c_v$ columns according to
        conflict graph coloring
02. Calculate the wafer-dicing yield $z$
03. For ($i$=0; $i < R+T-c_h-c_v$; $i$++)
04.     improve=true
05.     While (improve==true)
06.         improve=false
07.         For (each row and column whose dicing plan is chosen)
08.             try the dicing plan corresponding to each
                  maximal horizontal/vertical independent set
09.             If (wafer-dicing yield $z$ increases)
10.                 Replace the current dicing plan
11.                 improve=true
12.     If ($\exists$ rows or columns whose dicing plans are not chosen )
13.         For (each dicing plan)
14.             Calculate wafer-dicing yield $z$
15.     Choose the dicing plan for one row or column which
          produces the largest yield $z$
```

Figure 7: Iterative Augment and Search Algorithm

```
Input: Dimensions of Dies
Output: Die floorplan and dicing plan
1. Sort dies according to height
2. For (all possible shelf widths $w$)
3.     For (all dies $D$ from the highest to the shortest)
4.         Insert $D$ onto the lower-most shelf onto which die $D$ will fit
             or create a new shelf if no shelf has sufficient room
5.     Find the shelf $l$ with the maximum total die width
6.     For each other shelf $l'$
7.         sort dies according to width and place them onto the shelf
8.         slack[$l'$] = $w$ - total width of the dies in $l'$
9.         While (slack[$l'$] $> 0$)
10.            choose the leftmost die whose left edge is not aligned
                 with the dies in shelf $l$
11.            shift the die and all dies on its right side by $i$ to
                 align it with the dies in the shelf $l$
12.            slack[$l'$] = slack[$l'$] - $i$
13.    calculate the wafer-dicing yield $z$ using IASA
14. Find the shelf placement corresponding to the largest yield $z$
```

Figure 8: Shelf-packing and Shifting Method.

and the best die placement with the largest wafer-dicing yield z is chosen.

The generic simulated annealing placement algorithm is given in Figure 9. The algorithm starts with the floorplan found by the shelf-packing and shifting method as its initial placement. The objective value is calculated and recorded. In our implementation the objective function is computed as $(1-\alpha)$ (reticle area) + $\alpha(R\cdot T - z)$ where $\alpha \in \{0.1, 0.2, ..., 0.9\}$. (We run the algorithm with each value of α, and report the placement with maximum wafer-dicing yield z.) At each step we find a neighbor solution based on the following moves:

- x or y move, which changes the position of a single die;
- Aspect ratio move, which changes positions of multiple dies in order to reduce reticle aspect ratio;
- Orientation move, which rotates one die by 90 degrees.

Each generated solution is evaluated and kept with a probability dependent on the current temperature (see Figure 9).

5. EXPERIMENTAL RESULTS

Testcases for evaluating the performance and scalability of the SSWDP algorithms are randomly generated and placed by the Rectangle Packing Code of Javier Arevalo [14]. We choose ten random testcases with different numbers of dies, $|MHIS|$ and $|MVIS|$. Relevant parameters of the testcases are listed in Table 1. We set the required production volume to 40 for all dies and assumed that the number of reticle rows/columns on the wafer are $R = 10$, respectively $T = 10$. We used CPLEX 8.100 Mixed Integer Optimizer [18] with a runtime limit of 1 hour (3600 seconds) to solve the (ILP1)–(ILP3). A local optimal solution was obtained for the non-linear formulation (NLP) in Section 3.1 using LINGO 6.0 [19]. We also implemented the IASA heuristic (Figure 7) in C. All tests are run on an Intel Xeon 2.4GHz CPU.

```
Input: Dimensions of $n$ Dies, $\beta$: $0 \le \beta < 1$
Output: Reticle floorplan and wafer dicing plan
1. Get a shelf packing floorplan as the initial floorplan
2. Calculate Objective Value
3. while (not converge and # of move $< Move\_Limit$)
4.     choose a uniform random number $r \in [0,1]$
5.     make a random move according to $r$
6.     calculate $\delta$ =New Objective Value - Old Objective Value
7.     If ($\delta < 0$)
8.         Accept the move
9.     Else
10.        Accept the move with probability $e^{-\frac{\delta}{T}}$
11.    $T = \beta T$
```

Figure 9: Simulated Annealing Placement.

Case No.	# Dies	$\|MHIS\|$	$\|MVIS\|$	c_h	c_v
1	5	4	3	2	2
2	6	4	3	2	2
3	7	6	2	3	2
4	8	5	8	3	3
5	9	11	6	3	3
6	10	5	8	2	2
7	11	10	4	3	2
8	14	18	10	3	3
9	16	14	12	3	3
10	21	12	9	2	3

Table 1: Testcase parameters.

Results in Table 2 convincingly show that (ILP3) is more efficient than (ILP1) and (ILP2). On average, CPLEX can find the optimal solution with (ILP3) 1000 times faster than with (ILP1) and over 20 times faster than with (ILP2). For some testcases for which $|MHIS|$ and $|MVIS|$ are small, Lingo can also find the local optimal solutions with the NLP formulation quite quickly. IASA finds in all cases an optimum solution within 0.01 second, although in general the solution is not guaranteed to be optimal.

In order to evaluate RFWDP algorithms, we generated ten testcases to simulate industry MPW runs. For each testcase: (a) the reticle area limit is set as 300 with a maximum reticle dimension of 25, (b) the number of dies is between 9 and 20, and (c) the dimensions of the dies are between 1 and 8, with a total die area of at least 200.

We compared the performance of our algorithms with the two state-of-art floorplanners: Parquet [9] and GTmuch [11]. We downloaded Parquet 030224 [20], a simulated annealing min-area placer, from the web. We use Parquet as a min-area floorplanner and evaluate the yield of generated placement by using the (ILP3) specified in Section 3.2. A demo version of GTmuch 030925, a commercial placer with the objective of minimizing both reticle area and die loss during die sawing, is obtained from Xyalis. The runtime information is not given in the demo version. In our experiments, we use

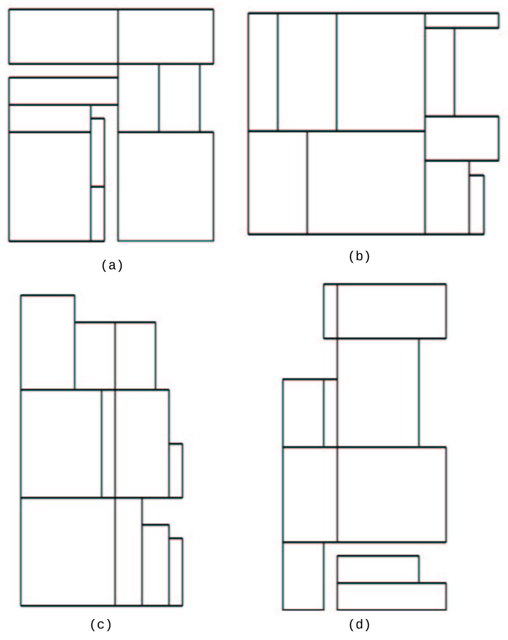

Figure 10: Placements for testcase 1: (a) Xyalis GTmuch (b) Parquet (c) Shelf packing+shifting (d) Simulated Annealing with the objective of minimizing *z*.

the "best optimization" option of GTmuch. We implemented the shelf packing and shifting algorithm as a C code and used it as the initial floorplan for the simulated annealing algorithm. The objective function for the simulated annealing algorithm was computed as $(1-\alpha)$ (reticle area) + $\alpha(R \cdot T - z)$ where $\alpha \in \{0.1, 0.2, ..., 0.9\}$. We ran the algorithm with all 9 different values of α, and report the placement with maximum wafer-dicing yield z. The move limit of 30000 is set for both Parquet and our simulated annealing placer.

The results for $R = 10$ and $T = 10$ given in Table 3 show that, when compared to Parquet, our shelf packing and shifting method can improve wafer-dicing yield by 83.3% at the expense of increase of reticle area by 5.96%. Compared to GTmuch, the shelf packing and shifting method can improve wafer-dicing yield by 37.7% while *reducing* reticle area by 3.3%. By using the simulated annealing code, we can further improve wafer-dicing yield by 30.5% in less than 1020 second at the expense of an increase of area by 3.3% compared to shelf packing and shifting method. The runtime of the simulated annealing algorithm is much longer than for the other algorithms due to the need to compute the wafer-dicing yield z for each move. However, the runtime remains practical for problem instances of typical size. The final placement solutions of the four methods for testcase 1 are shown in Figure 10. The side-to-side wafer sawing plan for testcase 1 generated by the simulated annealing placer is shown in Figure 11. The dies yielded by the sawing are pattern filled.

6. CONCLUSIONS AND FUTURE WORK

Multi-project wafers have recently become a popular way for minimizing mask manufacturing cost for low volume designs. In this paper we have considered multi-project reticle floorplanning and wafer dicing problems under the prevalent side-to-side wafer dicing technology. We have given practical mathematical programming formulations and efficient heuristics that maximize wafer-dicing yield while taking into account requirements on maximum margin widths.

In future work we will validate the proposed methods on industry testcases, and extend them to handle more constraints, such as user specified minimum die-to-die distances, rotation constraints (for some die rotation may not be allowed), and die-to-die alignment constraints. We will explore exact methods such as branch and bound for reticle design; such methods can be very useful since in practice the number of dies in a multiproject wafer is often small. We will also explore important degree of freedom such as partitioning of given set of projects into multiple reticles and assigning of multiple copies of the same design to the same reticle.[5] Finally, we plan to address alternative dicing technologies, which, although more expensive than side-to-side dicing, by enabling the use of more compact reticles may lead to lower overall production costs.

7. REFERENCES

[1] John, and B. Lin,"Mask Cost and Cycle Time Reduction". `http://www.sematech.org/public/resources/litho/mask/msw1001/E_TSMC.PDF`

[2] C. Yang, "Challenges of Mask Cost & Cycle Time". `http://www.sematech.org/public/resources/litho/mask/msw1001/K_Mask_cost_Intel.pdf`

[3] M. LaPedus, "The IC industry heading to the $10 million photomask?", *Semiconductor Business News*, Oct. 7, 2002, `http://www.siliconstrategies.com/story/OEG20021007S0053`

[4] R. D. Morse, "Multiproject Wafers: not just for million dollar mask sets", *Proc. SPIE*, Vol 5043, 2003, pp. 100-113.

[5] B. Courtois, "Infrastructure for Wducation and Research: from National Initiatives to Worldwide Development". Paper presented at Tech. Univ. Darmstadt Infrastructure Overview, available at `http://vlsi1.engr.utk.edu/ece/msn/courtois.pdf`

[6] G. Xu, R. Tian, D.F. Wong, and A. Reich, "Shuttle Mask Floorplanning", *Proc. SPIE*, Vol 5256, to appear.

[7] S. Chen and E. C. Lynn, "Effective Placement of Chips on a Shuttle Mask", *Proc. SPIE*, Vol 5130, 2003, pp. 681-688.

[8] M. Andersson, C. Levcopoulos and J. Gudmundsson, "Chips on Wafers", *Proc. WADS (Workshop on Algorithms and Data Structures)*, August 2003.

[9] S. N. Adya and I. L. Markov, "Fixed-outline Floorplanning Through Better Local Search", *Proc. International Conference On Computer Design*, pp. 328-333, August 2001.

[10] MOSIS, `http://www.mosis.org.`

[11] XYalis GTsuite, `http://www.xyalis.org.`

[12] R. Wilson, "Economics Hold Industry Back From Plunge into 90 nm". *EE Times*, May 17, 2002, `http://www.eetimes.com/semi/news/OEG20020517S0074.`

[13] WaferYield Inc., `http://www.waferyield.com.`

[14] Javier Arevalo, Rectangle Packing Code, `http://www.iguanademos.com/Jare/docs/RectPlace.htm.`

[15] IntelliMask, `http://www.intellisense.com/productsservices/products/intellimask.asp.`

[16] Curcuits Multi-Projects, `http://cmp.imag.fr.`

[17] K2 Technologies, `http://www.k2tech.com/products/`

[18] CPLEX Mixed Integer Optimizer, `http://www.cplex.com/`

[19] LINGO, LINDO Systems Inc., `http://www.lindo.com/`

[20] `http://vlsicad.eecs.umich.edu/BK/parquet/`

[5]As a preliminary result on the latter problem, we have shown that, for projects of identical dimensions, the optimum is to assign a number of copies proportional to the square root of the required production volume of each die.

Case No.	ILP1		ILP2		ILP3		NLP		IASA	
	40z	CPU(s)	40z	CPU(s)	40z	CPU(s)	40z	CPU(s)	40z	CPU(s)
1	25	1215.7	25	41.4	25	0.3	25	0.3	25	0.00
2	24	3600	25	1.2	25	0.2	25	0.2	25	0.00
3	21	12.3	21	0.4	21	0.1	21	1.4	21	0.00
4	10	3600	14	73.1	14	6.7	14	27.1	14	0.00
5	14	2013.4	14	57.3	14	2.5	14	40.7	14	0.00
6	20	3600	25	3514.3	25	4.3	25	0.2	25	0.00
7	16	3600	18	6.73	18	2.7	18	4.1	18	0.00
8	0	3600	14	3600	16	51.4	16	294.1	16	0.00
9	0	3600	10	3600	12	91.4	12	1236.4	12	0.00
10	0	3600	18	3600	20	179.7	20	12.2	20	0.00

Table 2: Results for SSWDP algorithms. $R = 10, T = 10$. 40z represents the minimum number of die copies diced from the wafer under the respective dicing plan.

Case No.	# Die	Die area	GTmuch		Parquet			Shelf+shift			SA+IASA		
			40z	area	40z	area	CPU(s)	40z	area	CPU(s)	40z	area	CPU(s)
1	10	231	18	255	14	255	0.03	16	276	0.00	28	288	80
2	18	226	16	285	5	252	0.07	21	253	0.16	25	270	783
3	11	252	15	280	10	272	0.03	16	280	0.01	30	294	132
4	9	203	18	221	14	220	0.03	25	224	0.01	30	288	118
5	10	226	12	272	15	247	0.02	25	280	0.00	25	260	94
6	15	227	10	285	6	252	0.06	18	238	0.00	18	238	226
7	15	234	14	285	9	252	0.06	15	288	0.01	20	285	782
8	14	232	20	285	9	255	0.03	20	288	0.00	20	288	152
9	10	231	20	285	14	255	0.03	16	276	0.00	28	288	161
10	20	215	6	304	6	238	0.09	15	255	0.01	20	260	1020
Total		2277	149	2757	102	2518		187	2668		244	2757	

Table 3: Results for RFWDP algorithms. $R = 10, T = 10$. 40z represents the minimum number of die copies diced from the wafer under the respective dicing plan.

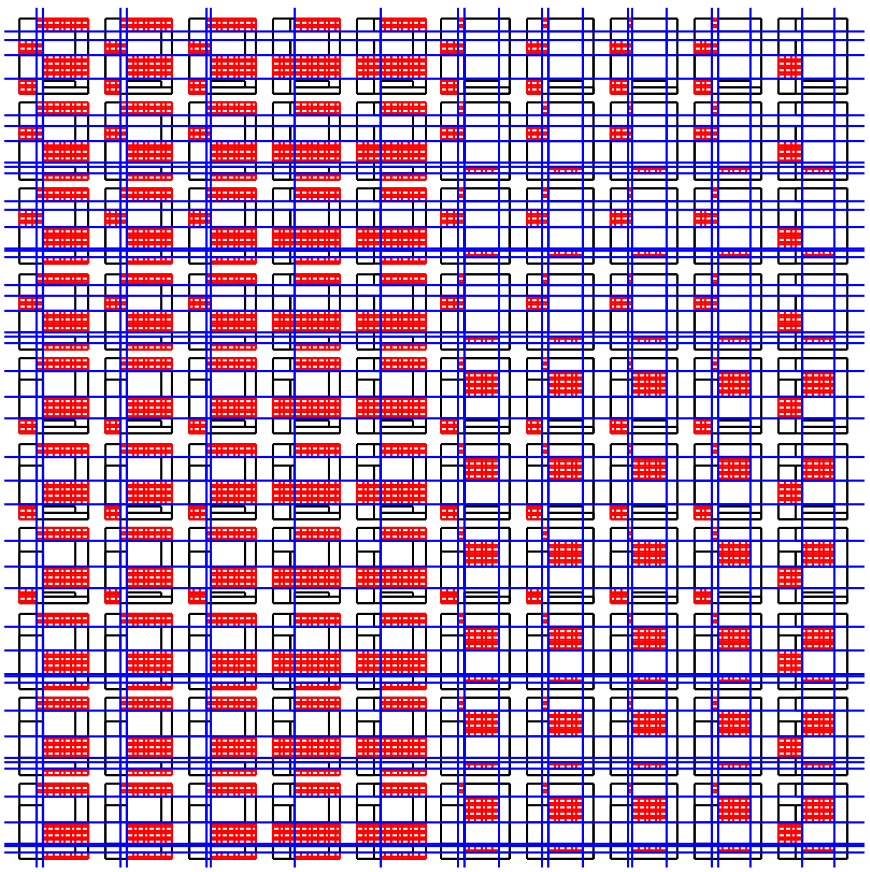

Figure 11: The side-to-side wafer sawing plan for testcase 1 generated by the simulated annealing placer.