Yield-Driven Multi-Project Reticle Design and Wafer Dicing

Andrew B. Kahng,[a] Ion Mandoiu,[b] Xu Xu[c] and Alex Zelikovsky[d]

[a]CSE and ECE Departments, University of California at San Diego
[b]CSE Department, University of Connecticut
[c]CSE Department, University of California at San Diego
[d]CS Department, Georgia State University

ABSTRACT

The aggressive scaling of VLSI feature size and the pervasive use of advanced reticle enhancement technologies has led to dramatic increases in mask costs, pushing prototype and low volume production designs to the limit of economic feasibility. Multiple project wafers (MPW), or "shuttle" runs, provide an attractive solution for such low volume designs, by providing a mechanism to share the cost of mask tooling among up to tens of designs. However, MPW reticle design and wafer dicing introduce complexities not encountered in typical, single-project wafers. Recent works on wafer dicing do not take in account several known degrees of freedom and requirements, which degrades the optimality and feasibility of the proposed solutions. Furthermore, the delay cost associated with schedule alignment has been completely ignored in all previous works.

In this paper we propose an enhanced MPW flow comprising four main steps: (1) schedule-aware project partitioning, (2) multi-project reticle floorplanning, (3) wafer shot-map definition, and (4) wafer dicing plan definition. The proposed project partitioning algorithm gives improved trade-offs between mask cost and schedule delay cost. Our reticle floorplaner combines hierarchical quadrisection with a simulated annealing framework to generate more "diceable" floorplans subject to given maximum reticle sizes. The round wafer shot-map definition step maximizes extraction of functional dies from partially printed reticle images. Finally, our dicing planner employs multiple side-to-side dicing plans for different wafers, as well as different reticle image rows/columns within a wafer. Experiments on industry testcases show that our methods significantly outperform not only previous methods in the literature, but also reticle floorplans manually designed by experienced engineers.

1. INTRODUCTION AND MOTIVATION

With the shrinking of VLSI feature size and the pervasive use of advanced reticle enhancement technologies such as Optical Proximity Correction (OPC) and Phase Shifting Masks (PSM), mask set 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 production volume. Multiple Project Wafers (MPW), or "shuttle" runs, provide an efficient method to reduce the cost.[7] Thus, from government sponsored programs allowing students to verify their design in silicon,[4] MPW has now become a commercial service offered by both independent providers such as MOSIS and CMP and semiconductor foundries such as TSMC and IBM.

Packing and dicing different dies on a multi-project wafer introduces complexities not encountered in typical, single-project wafers. Recently, several approaches have been proposed in the literature for addressing the MPW reticle floorplanning problem. Chen and Lynn[3] considered in this context the problem of finding the minimum area slicing floorplan, with 90-degree chip rotation allowed. They gave a "bottom-left fill" algorithm for constructing an initial solution, followed by enumeration based on B*-trees. Xu et al.[8] 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 Andersson et al.,[1] where the objective is to find a minimum area grid floorplan with at most one die per grid cell.

Kahng et al.[5] were the first to consider the side-to-side wafer dicing problem, and proposed a general multi-project reticle floorplanning method seeking to maximize dicing yield. Their method also allows maximum dicing margins to be specified for each die. Very recently, Kahng and Reda[6] revisited the grid-packing formulation in Andersson et al.[1] and proposed a new floorplanner with guaranteed yield. The approaches in[5,6] are based on the implicit assumption that all wafers use the same dicing plan. Xu et al.[9] combine the horizontal and vertical conflict graphs of Kahng et al.[5] into a

single conflict graph, and cut out from each wafer all dies receiving a certain color in a minimum coloring of the conflict graph. The implicit assumption for this approach is that exactly one horizontal (vertical) dicing plan is used for all reticle reticle image rows (columns) within a wafer. Finally, Xu et al.[10] give methods for MPW reticle floorplanning and dummy fill insertion to minimize topography variation after chemical-mechanical polishing. Balasinski[2] gives an overview of related multi-layer mask technologies, which rely on sharing the reticle space between multiple layers of the *same* design, typically via blading. These previous approaches fail to take into account (i) different production volume requirements for different dies,[6,9] (ii) the possibility of different dicing plans for different wafers,[5,6] or for different reticle reticle image rows/columns within the same wafer,[9] (iii) round wafer shape (by assuming a rectangular array of reticle images in the model),[5,6] or (iv) delay cost associated with schedule alignment.

In this paper we propose an enhanced MPW flow aimed at minimizing the manufacturing cost to fulfill given die production volumes. Our flow includes four main steps: (1) schedule-aware project partitioning (2) multi-project reticle floorplanning, (3) wafer shot-map definition, and (4) wafer dicing plan definition. Our contributions are as follows. For the first step, we propose a branch and bound procedure to achieve the best tradeoff between mask cost and delay cost. For the second step, we propose an algorithm combining hierarchical quadrisection with simulated annealing to generate "diceable" floorplans observing 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 third step, which has not been previously considered in the context of MPW, we propose a simple algorithm that allows full utilization of 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 around 12% in the required number of wafers for a fixed reticle floorplan. For the fourth step, following[9] we assume that all rows and columns of reticle images within a wafer are diced using the same set of cuts and give an integer program for finding an *optimal* dicing plan in practical runtime. We also give a two-level optimization algorithm that simultaneously allows multiple side-to-side dicing plans for different wafers and for different reticle image rows/columns within a wafer. Finally, we show the advantages of partitioning each wafer into a small number of parts before individual die extraction. For a fixed reticle floorplan, the two-level optimization algorithm on average reduces the required number of wafers by 42%, 47%, or 63% without wafer partitioning and with wafer partitioning into 2 or 4 parts, respectively.

The rest of our paper is organized as follows. In the next section we consider the schedule-aware project partitioning problem. In Section 3, we give the new hierarchical quadrisection method for reticle floorplanning. Section 4 is devoted to the wafer shot-map definition problem and our proposed solution, while Section 5 describes the Multiple Dicing Plan (MDP) advantages and a new two-level optimization algorithm. Finally, in Section 6 we give experimental results that evaluate our proposed methods on industrial testcases. The comparisons are performed separately for the case when only side-to-side wafer dicing is allowed and when the wafer can be divided into halves or quarters before dicing.

2. SCHEDULE-AWARE PROJECT PARTITIONING

One major practical limitation of the multi-project wafer is the delay cost associated with schedule alignment. Projects with early tape-out schedules have to be delayed and the final MPW tape-out schedule depends on the project with the latest tape-out schedule.[2] The delay cost is too large to be ignored in practice, especially for low-volume production. In a simple delay cost model, for any single project the delay cost is equal to $c_0 \times T_d$, where c_0 is a constant and T_d is equal to the difference between its tape-out schedule date and the latest tape-out schedule date of the projects on the same reticle.

The *project partitioning problem* is formulated as follows:

Project Partitioning Problem (PPP). Given a maximum reticle size, a set of dies and their sizes, mask cost and tape-out schedule for each project, find a partition of projects into reticles such that the sum of the delay cost and the mask cost is minimized.

In this "front-end" reticle design stage, we assume that the wafer cost is ignorable compared with mask cost and delay cost. This assumption is reasonable for prototyping and low-volume production since the number of wafers to be used is small. We employ a greedy merge algorithm to solve this problem as shown in Figure 1. Line 1 gives the initial solution in which every die occupies an entire reticle. Then the reticles are sorted by tape-out schedules. Iterative merging reticles reduces the manufacturing cost (Lines 3-7). In each loop, we merge two neighboring reticles with the maximum positive cost reduction. A min-area reticle floorplanner is used to check feasibility of merging two reticles into a single reticle in Line 5.

Input: Mask cost and tape-out schedules of n dies, maximum reticle size
Output: Partition of the dies into m reticles
1. Start with each die in a separate reticle 2. Sort all reticles according to tape-out schedules 3. **while** (maximum cost reduction > 0) 4. **For** (every pair of neighboring reticles) 5. **If** (two reticles can be merged into a single reticle) 6. calculate the cost reduction 7. Merge the two reticles with the maximum cost reduction

Figure 1. Greedy merge algorithm for project partitioning.

3. 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 minimizing 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 (see Figure 2). The reticle is divided hierarchically into 4^l regions. At the l^{th} level, each region $R = R_{a_1 a_2 \ldots a_l} (a_i \in \{1,2,3,4\})$ contains at most one die. We denote the width of the region R as $W(R)$ and the height as $H(R)$. The hierarchical quadrisection allows computing height and width in a bottom-up manner using the following formulas.

- $W(R_{a_1 \ldots a_{l-1}}) = Max(W(R_{a_1 \ldots a_{l-1}1}), W(R_{a_1 \ldots a_{l-1}3})) + Max(W(R_{a_1 \ldots a_{l-1}2}), W(R_{a_1 \ldots a_{l-1}4}))$

- $H(R_{a_1 \ldots a_{l-1}}) = Max(H(R_{a_1 \ldots a_{l-1}1}), H(R_{a_1 \ldots a_{l-1}2})) + Max(H(R_{a_1 \ldots a_{l-1}3}), H(R_{a_1 \ldots a_{l-1}4}))$

The *wafer requirement* for each region R can be computed in the same recursive bottom-up manner. If we assume that single row and column dicing plan is used for all wafers, either all copies of die D on one wafer are obtained or no copies of die D are obtained. The *wafer requirement* of die D to satisfy the volume requirement is $\lceil \frac{N(D)}{Q(D)} \rceil$, where $N(D)$ is the volume requirement of the die D and $Q(D)$ is the number of die D per wafer. For a set S of dies in which any two dies can be simultaneously obtained, the *wafer requirement* is $MAX_{D \in S_1}(\lceil \frac{N(D)}{Q(D)} \rceil)$.

- For the region in the l^{th} level, the set $S_1(R_{a_1 \ldots a_l})$ includes the die in the region. The *wafer requirement* for S_1 is calculated. (The wafer requirement is zero for the empty set.)
- For the region in the $(l-i)^{th}$ level $R_{a_1 \ldots a_{l-i}}$, sort the 2^{i-1} sets in each of the four sub-regions according to wafer requirement. Then we can group the dies into 2^i sets: the first 2^{i-1} sets are $S_k = S_k(R_{a_1 \ldots a_{l-i}1}) \bigcup S_k(R_{a_1 \ldots a_{l-i}4})$ $(k = 1, \ldots 2^{i-1})$. It is obvious that any two dies in the same set are not in dicing conflict since all the dies in the region 1 are not in dicing conflict with the dies in the region 4. Similarly, the second 2^{i-1} sets are $S_{2^{i-1}+k} = S_k(R_{a_1 \ldots a_{l-i}2}) \bigcup S_k(R_{a_1 \ldots a_{l-i}3})$ $(k = 1, \ldots 2^{i-1})$.
- At the top level, we have 2^l sets and the final wafer requirement is the sum of the wafer requirement of all the 2^l sets.

Therefore, the reticle area and wafer requirement for the floorplan can be easily calculated.

We give a generic simulated annealing placement algorithm in Figure 3. Line 1 is the step to merge two dies with the same width w and volume requirement as one die whose width is w and whose height is the sum of the heights of the two dies. The algorithm starts with the floorplan with each die randomly placed in the 4^l regions as its initial placement. The objective value is calculated and recorded. In our implementation the objective function is the wafer requirement by assuming $Q(D) = \frac{wafer\_area}{reticle\_area}$ for all $D \in \mathcal{D}$. At each step we find a neighbor solution based on the following moves:

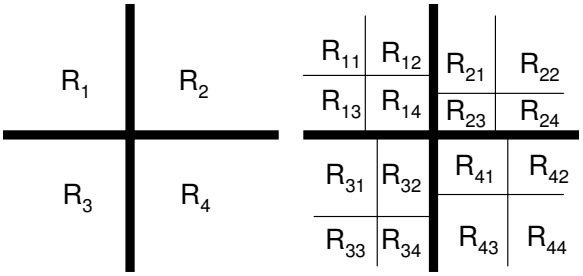

Figure 2. Two-level Hierarchical Quadrisection Floorplan.

Input: Dimensions of n dies, β: $0 \leq \beta < 1$
Output: Reticle floorplan and wafer dicing plan
1. Choose a random hierarchical quadrisection floorplan 2. **while** (not converge and # of move $<$ *Move_Limit*) 3. make a random move 4. calculate δ =New Objective Value - Old Objective Value 5. **If** $(\delta < 0)$ then accept the move 6. **Else** accept move with probability $e^{-\frac{\delta}{T}}$ 7. $T = \beta T$

Figure 3. Hierarchical Quadrisection Floorplan.

- Region exchange move, which exchanges the dies in two regions if at least one of the regions contains a die;
- 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 3). Note that the hierarchical quadrisection structure will be maintained during the process.

4. WAFER SHOT-MAP DEFINITION

The wafer shot-map definition step, which determines the position of reticle images printed on wafer, has been ignored in previous works on MPW. In both[6] and,[5] the wafer is modeled as a rectangular array of projects, which is obviously not true for actual round wafers. This simplification may lead to wrong dicing yield estimation since (i) the reticle image rows (columns) do not have equal contributions to the wafer dicing yield – e.g., the rows/columns near the center contain more reticle images, and (ii) fully printed dies within partial reticle image are ignored. For a round wafer with the radius r and center (x_0, y_0), a die image D is *on wafer* if and only if $(x-x_0)^2 + (y-y_0)^2 \leq r^2$ for all $(x,y) \in D$. Given a rectangular reticle image, a *reticle image plane* is a regular tiling of the plane with identical copies of the reticle. The *wafer reticle image problem* is formulated as follows:

Wafer Shot-Map Definition Problem (WSMDP). Given a reticle image plane and the wafer radius r, find the position of the wafer center minimizing the number of wafers required to meet the given production volumes.

Due to the periodicity of the reticle image lattice, we can prove that the optimal solution of WSMDP can be achieved when the location of the wafer center is restricted to be within one reticle reticle image L. The algorithm for MDP is summarized in Figure 4. Two tricks are employed in the algorithm to speed up the process. (1) We store all feasible sets whose wafer costs are calculated for comparison. In Line 6, if $F(g)$ is included in any stored set, g will be skipped to avoid redundant wafer cost calculation. (2) A threshold value α is used to determine whether the process should be continued. We can take the radial yield model (e.g., Teet's radial yield model) and defect models (e.g., Poisson, Murphy, Seeds, etc.) into account during the wafer cost evaluation.

Input: wafer radius r, reticle dimensions	
Output: placement of wafer center maximizing the given objective	
1.	Divide one projection L into $l \times l$ uniformly-spaced grid
2.	Find N_w and dicing yield y when the wafer center is at the first point g_0, store the feasible set $F(g_0)$
3.	$Min\_N_w \leftarrow N_w$; $Max\_yield \leftarrow y$
4.	**while** ($Max\_yield \geq \alpha$)
5.	Move to the next grid point g
6.	**If** $F(g)$ not included in any stored feasible set
7.	Find N_w and the dicing yield y, store $F(g)$
8.	**If** ($N_w < Min\_N_w$)
9.	$Min\_N_w \leftarrow N_w$; $Max\_yield \leftarrow y$

Figure 4. Wafer Shot-Map Definition Algorithm

5. MULTIPLE WAFER-DICING-PLAN DICING

In two works by Kahng et al.,[5,6] the authors assume that a single dicing plan (SDP) is used for all wafers. The wafer yield then is determined by the die with the minimum ratio of the number of copies sliced out to the volume requirement. When multiple dicing plans (MDP) are allowed, different wafers may contribute different number of copies of a die towards satisfying the total volume requirement. Thus, MDP can balance better the number of useful die copies extracted from different wafers, particularly for non-uniform production volume requirements. In this section we first describe how to extend the IASA SDP algorithm of Kahng et al.[5] to find MDPs. We then give a simple integer linear programming (ILP) approach to find optimal MDPs that are restricted as in Xu et al.[9] to use a single set of cuts for all reticle image rows/columns within a wafer. Finally, we conclude with a two-level optimization algorithm combining the first two approaches.

5.1. Side-to-Side Wafer Dicing

A wafer consists of a number of reticle reticle images arranged in a number of reticle image *reticle image rows* and *reticle image columns*. Each reticle image is a copy of the same reticle image. In the prevalent "side-to-side" wafer dicing technology, the diamond blades can not stop at arbitrary points during cutting; consequently, all reticle images in the same reticle image row (or column) will share the same horizontal (or vertical) cutlines. Following Kahng et al.,[5] two dies D and D' on a reticle are said to be in *vertical (resp. horizontal) dicing conflict* if no set of vertical (resp. 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 (or vertical) independent set* is a subset of $\mathcal{D}$ which can be sliced out by a set of horizontal (or vertical) cutlines; the set of cutlines used for a wafer are called as a *wafer dicing plan*. The following problem formulation extends the formulation of Kahng et al.[5] by allowing different dicing plans to be used for different wafers:

Side-to-Side Multi-Wafer Dicing Problem (SSMWDP). Given a reticle with dies $\mathcal{D} = \{D_1, \ldots, D_n\}$, required production volume for each die $N(D_i)$, $i = 1, \ldots, n$, and the positions of the reticle reticle images of the wafer, find the minimum number of wafers N_w and the corresponding dicing plan for each wafer such that the required production volume for each die is satisfied.

The *dicing yield* of a multi-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 $N(D)$. Note that SSMWDP requires the minimum number of wafers (and the associated dicing plans) such that the dicing yield is at least 1. In our present work, we extend SSMWDP to allow preliminary partitioning of each wafer into a small number of parts (e.g., halves or quarters) so that the side-to-side dicing plans for the parts can be independent from each other.

5.2. Extended IASA

The IASA method proposed by Kahng et al.[5] can be easily extended to solve MDP by placing N_w wafers into one "super-wafer" as shown in Figure 5. Then we can use IASA for SDP to produce a dicing plan for the N_w wafers. However, the runtime will increase rapidly when N_w is large since we need to check all rows and columns of the "super-wafer" in each iteration.

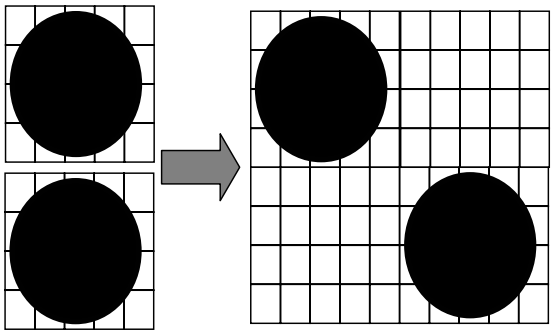

Figure 5. Placing two wafers on one "super-wafer".

5.3. Integer Linear Program for Restricted MDPs

Xu et al.[9] assume that each wafer uses exactly one horizontal dicing plan and one vertical dicing plan for all reticle image rows/columns within a wafer. This assumption allows them to use a coloring-based heuristic which gives good results for testcases with large volume requirement. In this section we give an integer linear programming formulation which allows finding optimal MDPs restricted in this way.

As in,[9] 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 $\mathcal{D}$ which 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 graph. For each independent set $C \in MCIS$, let f_C denote the number of wafers which use the dicing plan defined by C, MDP can be formulated as the following integer linear program:

Minimize N_w (ILP1)
subject to

$$\begin{aligned} &\sum_{D \in C} Q(C,D)\, f_C \geq N(D), && \forall D \in \mathcal{D} \\ &\textstyle\sum_C f_C = N_w \\ &f_C \in \mathbb{Z}_+, && \forall C \in MCIS \end{aligned}$$

where $Q(C,D)$ is a constant which 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 6, the runtimes of ILP are within 0.03 second in all the experiments on industry testcases with up to 30 dies.

5.4. Two-Level Optimization 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 Two-level Optimization (TLO) heuristic shown in Figure 6. We first solve ILP1 to obtain an upper bound on N_w. Then we gradually reduce the number until the dicing yield becomes smaller than 1. In Lines 04-08, we assume all rows (columns) of each wafer using the same horizontal (vertical) dicing plan. The dicing plan for each wafer are obtained by solving the following ILP:

Minimize Y (ILP2)
subject to

$$\begin{aligned} &N(D) - \sum_{D \in C} Q(C,D)\, f_C \leq y_D, && \forall D \in \mathcal{D} \\ &\textstyle\sum_C f_C = N_w \\ &\textstyle\sum_D y_D = Y \\ &f_C \in \mathbb{Z}_+, && \forall C \in MCIS \\ &y_D \in \mathbb{Z}_+, && \forall D \in \mathcal{D} \end{aligned}$$

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 (dicing yield≥ 1 )
03.     N_w − −
04.     Solve ILP2 and choose one set C ∈ MCIS for each wafer
05.     Set the weight of each die D as y_D
06.     For (each wafer)
07.         Choose maximal horizontal (vertical) independent set which include C and maximizes the total weight of dies
08.         Use the corresponding dicing plans for each row (column)
09.     While (improve==true)
10.       While (improve==true)
11.         For (each row and column)
12.             try other horizontal (vertical) dicing plans
13.             If (dicing yield increases)
14.                 Replace the current dicing plan
15.       For (the center row and the center column of each wafer)
16.           Simultaneously try other pairs of horizontal and vertical dicing plans
17.           If (dicing yield increases)
18.                 Replace the current dicing plan
```

Figure 6. Two-level Optimization Heuristic

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 1. CMP testcase parameters.

where Y is the total number of unsatisfied volume requirement and y_D is the number of unsatisfied volume requirement for the die D. We choose the horizontal and vertical dicing plan for each wafer which maximizes the total weight, and then we perform the row and column level check in Lines 11-14 to improve the yield by replacing the dicing plan for one row or column. Since the dicing plans for all rows and columns are chosen, we do not have the *iterative augment* process of IASA in our heuristic. Instead, we use a *cross selection* process in Lines 15-18 to choose the dicing plan for one row and one column simultaneously. Since the "cross selection" process is time-consuming, we do it only for the center row and column of each wafer.

6. EXPERIMENTAL RESULTS

We used six industry testcases from CMP[11] to evaluate the performance and scalability of the proposed algorithms. Each testcase has between 12 and 31 dies, with varying sizes and production volume requirements. For the wafer shot-map and wafer dicing problems, we use the reticle floorplan of the actual industry MPW runs, which were manually designed by an experienced engineer. The basic parameters of the six testcases are listed in Table 1.

Project Partitioning. Our algorithm for the schedule-aware project partitioning problem is implemented in C++. We assume that $c_0 = 150000$ per week and the mask cost is 500000. The tape-out schedules for all projects are randomly generated between zero and ten weeks. The project partitioning results are summarized in Table 2. Here "Without MPW" denotes the sum of mask cost and delay cost for project partitioning without MPW, i.e., each project occupies one reticle, "Schedule-blind" is the mask cost driven partitioner which aims to minimize the number of reticles without considering delay cost, and "Greedy Partitioner" is our proposed greedy merge algorithm. The results show that our proposed greedy merge algorithm can reduce cost by 63.8% compared with the traditional project partitioning. On the other hand, ignoring

Cases	Without MPW	Schedule-blind	Greedy Partitioner
Ind 1	6M	8.3M	2.95M
Ind 2	7M	9.05M	3.25M
Ind 3	12M	15.8M	3.9M
Ind 4	15.5M	20.75M	4.3M
Ind 5	7M	9.05M	3.25M
Ind 6	12M	15.8M	3.9M
Total	59.5M	78.75M	21.55M
Reduction (%)	0	-32.35	63.8

Table 2. Project partitioning results for six testcases.

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		

Table 3. Reticle floorplan results for six industry testcases. CMP is the original industry floorplan used by the CMP multi-project wafer service, "IASA+SA" is the SDP-driven floorplanner used in [5] and HQ is our proposed hierarchical quadrisection floorplan algorithm.

the delay cost leads to an increase of the cost by 32.35%, which indicates that delay cost cannot be ignored in project partitioning.

Reticle Floorplanning. We implemented our hierarchical quadrisection floorplan algorithm in C++. The maximum reticle dimension is set to be 2cm. 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 3. Here "CMP" denotes the original industry floorplan used by CMP, "IASA+SA" is the SDP driven floorplanner used by Kahng et al.,[5] 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.

Wafer Shot-Map Definition. Our algorithm for the wafer shot-map definition problem is implemented in C++. We choose

Cases	# part	1×1		10×10		100×100	
		N_w	CPU(s)	N_w	CPU(s)	N_w	CPU(s)
Ind 1	1	3	0.14	3	0.14	2	1534
Ind 2	1	3	0.18	2	8.3	2	1.15
Ind 3	1	4	4.59	4	4.6	4	4.6
Ind 4	1	4	73.6	4	73.7	4	73.7
Ind 5	1	2	0.21	2	0.3	2	0.3
Ind 6	1	6	3.57	5	200	5	343
Total		22		20		19	
Red.(%)				9.1		13.6	
Ind 1	2	2	0.05	2	0.1	2	0.1
Ind 2	2	2	0.06	2	0.1	2	0.06
Ind 3	2	3	3.98	3	3.97	3	3.95
Ind 4	2	3.5	0.76	3	4908	3	2915
Ind 5	2	1.5	0.21	1.5	0.3	1	1382
Ind 6	2	5	3.57	4	223	4	1001
Total		17		15.5		15	
Red.(%)				8.8		11.8	
Ind 1	4	1.5	0.02	1.5	0.1	1.25	641
Ind 2	4	1.5	0.02	1.25	0.5	1.25	4.62
Ind 3	4	2.75	0.17	2.75	0.16	2.5	55017
Ind 4	4	2.75	0.72	2.5	170	2.5	1456
Ind 5	4	1	0.01	1	0.01	0.75	1877
Ind 6	4	4.5	0.82	4	1250	4	5230
Total		14		13		12.25	
Red.(%)				7.1		12.5	

Table 4. Cost efficiency of wafer shot-map definition step for six industry testcases.

the number of grid points as 1×1, 10×10 and 100×100 and use TLO as the dicing heuristic. We choose $\alpha = 1.15$ in our experiments. The wafer cost and runtime results are summarized in Table 4. The results show that the wafer cost can be reduced by 9.1% and 13.6% by using 10×10 and 100×100 grid, respectively, at the expense of increased runtime. Similar improvements are observed for two- and four-part dicing.

Wafer Dicing. We implement the wafer dicing algorithms in the C++ language. We set the wafer diameter to be six inches and use a fixed wafer shot-map for all testcases. The number of wafers used (N_w) and runtime of four methods are shown in Table 5, where IASA is the SDP method used by Kahng et al.,[5] E-IASA is the extended IASA in Section 3.1, ILP is the integer linear programming restricted MDP method specified in Section 3.2 and TLO is the proposed two-level MDP optimization method. Each method was run without any wafer partition, and with wafer partition into 2 or 4 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 can reduce the cost by 57.9% for four parts. Therefore, ILP is more efficient for multiple part dicing. TLO achieves the best solution quality in a short time, reducing wafer cost by 63.2% for four parts.

To investigate the impact of volume requirement on all dicing methods, we multiply the volume requirement of each die by a coefficient. The coefficient is chosen from 0.5 to 16 for the testcase “Ind 3”. The results shown in Table 6 suggest that Extended IASA gives good results but needs prohibitively long runtime for large required volumes. The ILP solution can always find a solution very quickly. Its performance is not as good as TLO for small volume requirements, but is comparable to that of TLO for large volume requirements.

The final reticle floorplan and wafer dicing plans for the CMP testcase “Ind 2” are shown in Figures 7 and 8.

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	

Table 5. Wafer dicing results for six testcases. IASA is the algorithm proposed in [5]; E-IASA is our extended IASA heuristic; ILP is the proposed integer linear programming approach; and TLO refers to our two level optimization algorithm.

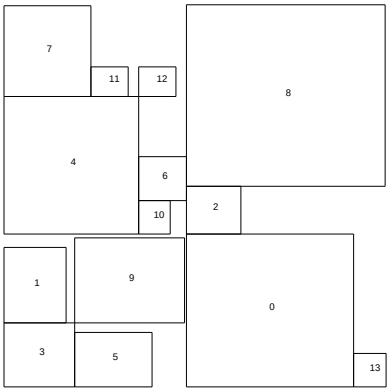

Figure 7. The reticle floorplan for testcase “Ind 2”.

7. CONCLUSIONS AND FUTURE WORK

In this paper we have proposed improved algorithms for schedule-aware project partitioning, multi-project reticle floorplanning, wafer shot-map definition, and wafer dicing. Experiments on industry testcases show that our methods significantly outperform 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,[8] by merging two copies of a die in a single “super-die”. In ongoing work we investigate the use of multiple die copies in the reticle, as well as multi-layer reticles, for further reductions in the manufacturing cost of prescribed die production volumes.

coeff	# part	IASA+SDP		IASA+MDP		ILP		TLO	
		N_w	CPU(s)	N_w	CPU(s)	N_w	CPU(s)	N_w	CPU(s)
0.5	1	5	4.8	3	141	5	0.01	3	2.92
1	1	9	4.8	5	617	5	0.01	4	4.59
2	1	17	4.8	8	3054	7	0.01	6	4.53
4	1	34	4.8	13	13796	12	0.01	11	0.53
8	1	68	4.8	23	74173	21	0.01	21	0.16
16	1	135	4.8	45	494657	41	0.01	40	1.73
0.5	2	4	16.8	2.5	256	2.5	0.00	2	3.83
1	2	7	16.8	4.5	1485	3.5	0.01	3	3.98
2	2	13	16.8	7	3187	6	0.0	5.5	0.29
4	2	25	16.8	13	24419	10.5	0.0	10	15.8
8	2	50	16.8	23.5	242752	20.5	0.0	20	1.38
16	2	100	16.8	–	–	40	0.01	39.5	2.26
0.5	4	4	44.8	2	406	1.5	0.01	1.5	0.01
1	4	7	44.8	3.75	2246	3	0.01	2.75	0.17
2	4	13	44.8	6	7978	5.25	0.0	5.25	0.0
4	4	25	44.8	11.5	51930	10.25	0.0	10.25	0.0
8	4	50	44.8	23.0	472487	20.25	0.0	20.25	0.0
16	4	100	44.8	–	–	40.5	0.0	40.25	3.17

Table 6. Wafer dicing results for the testcase “Ind 3” with different volume coefficient.

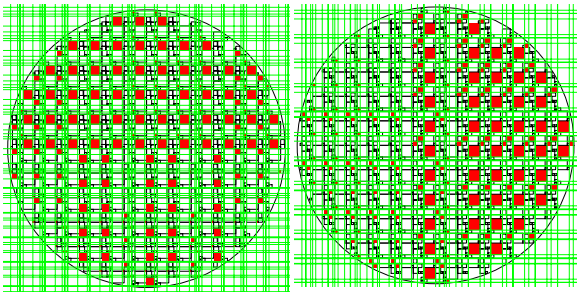

Figure 8. The wafer dicing plans for testcase “Ind 2”.

REFERENCES

1. M. Andersson, C. Levcopoulos and J. Gudmundsson, “Chips on Wafers”, *Proc. WADS (Workshop on Algorithms and Data Structures)*, August 2003.
2. A. Balasinski, “Multi-layer and multi-product masks: cost reduction methodology”, *Proc. 24th BACUS Symp. on Photomask Technology*, Proc. SPIE, Vol 5567, 2004, pp. 351–359.
3. S. Chen and E. C. Lynn, “Effective Placement of Chips on a Shuttle Mask”, *Proc. SPIE*, Vol 5130, 2003, pp. 681-688.
4. 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`
5. A. B. Kahng, I. I. Mandoiu, Q. Wang, X. Xu, and A. Zelikovsky, “MultiProject Reticle Floorplanning and Wafer Dicing”, *Proc. Intl. Symp. on Physical Design*, pp. 70-77, April 2004.
6. A. B. Kahng and S. Reda, “Reticle Floorplanning With Guaranteed Yield for Multi-Project Wafers”, *Proc. International Conference On Computer Design*, pp. 106-110, October 2004.
7. R. D. Morse, “Multiproject Wafers: not just for million dollar mask sets”, *Proc. SPIE*, Vol 5043, 2003, pp. 100-113.
8. G. Xu, R. Tian, D.F. Wong, and A. Reich, “Shuttle Mask Floorplanning”, *Proc. SPIE*, Vol 5256, pp. 185-194.
9. G. Xu, R. Tian, D. Z. Pan and M. D. F. Wong “A Multi-objective Floorplanner for Shuttle Mask Optimization”, *Proc. SPIE*, Vol 5567, 2004, pp. 340-350.
10. G. Xu, R. Tian, D. Z. Pan and M. D. F. Wong “CMP Aware Shuttle Mask Floorplanning”, *Proc. Asia South Pacific Design Automation Conference (ASPDAC)*, 2005.
11. Curcuits Multi-Projects, `http://cmp.imag.fr.`