A General Framework for Multi-Flow Multi-Layer Multi-Project Reticles Design

Andrew B. Kahng[a,b,c] and Xu Xu[b,c]

[a]UCSD ECE Department, La Jolla, CA
[b]UCSD CSE Department, La Jolla, CA
[c]Blaze DFM, Inc., Sunnyvale, California

ABSTRACT

The aggressive scaling of VLSI feature size and the pervasive use of advanced reticle enhancement technologies leads 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 designs, by providing a mechanism to share the cost of mask tooling among up to tens of designs of the same technology flow. However, delay cost associated with schedule alignment is ignored in previous work. The savings on mask cost may be easily surpassed by the profit loss due to forced schedule alignment. Therefore, Multi-Flow Multi-Layer Multi-Project Reticles (MFMLMPR) become a more viable mask-cost saving technique for low volume production since mask cost is shared between different layers of the same design and between designs of different technology flows. However, MFMLMPR design introduces complexities not encountered in traditional single-flow or single-layer reticles.

In this paper, we propose the first design flow for MFMLMPR aimed at minimizing the total manufacturing cost (including mask cost, wafer cost and delay cost) to fulfill given die production volumes. Our flow includes three main steps: (1) schedule-aware project partitioning with multi-flow embedding, (2) multi-frame reticle design, and (3) multi-project frame floorplanning. Our contributions are as follows. For the first step, a fast iterative matching algorithm is proposed to calculate the mask cost for multi-flow embedding with consideration of all practical manufacturing costs. We then propose an integer linear programming (ILP) based method for optimal manufacturing cost minimization. Since ILP suffers from impractically long runtimes when the number of projects is large, we propose a sliding time window heuristic to exhaustively search the solution space for the best tradeoff between mask cost and delay cost. For the second step, we propose an ASAP frame embedding heuristic to minimize the mask cost. Finally, for the third step, a "generalized chessboard" floorplan with simulated annealing is proposed to generate more dicing friendly frame floorplans for multi-flow projects, observing given maximum reticle sizes.

We have tested our flow on production industry testcases. The experimental results show that our schedule-aware project partitioner yields an average reduction of 58.4% in manufacturing cost. The reduction of mask cost is around 46.3% compared with use of traditional layer-by layer checking methods. Our generalized chessboard floorplanner leads to an average reduction of 22.8% in the required number of wafers compared to the previous best reticle floorplanner.

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 costs are predicted to reach $10 million by the end of the decade.[8] 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. Multiple-Project Wafers (MPW), or "shuttle" runs, provide an efficient method to reduce the cost.[9] Thus, MPW has now become a commercial service offered by both independent providers such as MOSIS and CMP and by semiconductor foundries such as TSMC and IBM. Recently, several approaches[1, 2, 4–6, 10–14] have been proposed in the literature for addressing the MPW reticle floorplanning and wafer dicing problems. However, one major practical limitation of the multi-project wafer is the delay cost associated with schedule alignment. Projects with early tapeout schedules must be delayed to align their schedules with the project having the latest tapeout

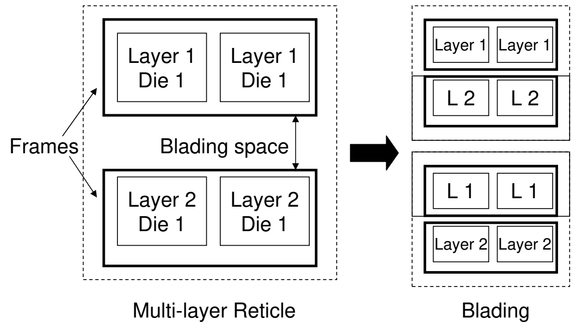

Figure 1. An example of Multi-Layer Reticle and blading.

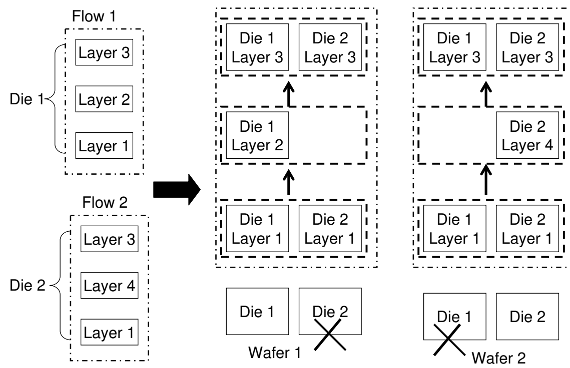

Figure 2. An example of Multi-Flow Reticles and their corresponding wafer manufacturing process.

schedule.[3] Even worse, schedule delay of the test chip manufacturing will in turn lead to schedule delay of the final product manufacturing. That is, the delay due to schedule alignment will result in profit loss which is too large to be ignored in practice. Unfortunately, this important profit loss due to delay (delay cost) is ignored in most previous papers. Balasinski[3] introduced the concept of the delay cost and Kahng et al.[7] first took the delay cost into consideration in their "schedule-aware" project partitioning algorithm. However, they use a simple delay model which assumes only a delay cost per week, and which does not reflect the actual product profit loss of the final product due to the delayed schedule of test chip manufacturing.

Since the savings of mask cost with MPW can be easily surpassed by the profit loss due to schedule alignment, a new multi-layer mask technology, *Multi-Layer Reticles*, which relies on sharing the reticle space between multiple layers of the *same* design, typically via blading, is proposed by Balasinski.[3] The delay cost can be minimized since the mask cost is amortized between the layers of the same design. As shown in Figure 1, the mask images of the same layer are placed in one rectangular area, which is referred to as a *frame*. A multi-layer reticle is shared by several frames separated by a certain distance which is referred to as the *blading space*. Exactly one frame will be exposed in photolithography while other regions of the reticle will be covered; this is referred to as *blading*. Therefore, the reticle images of different layers can be printed on the wafer one by one.

A more aggressive mask-sharing technique is to share the mask cost between designs of different technology flows. In the example shown in Figure 2, two designs of different technology flows can share the mask cost by placing same layers in one reticle and placing different layers in separated reticles. These reticles are exposed in two different ways onto two wafers such that exactly one design is successfully printed on each wafer. Therefore, two reticles can be saved at the expense of doubled wafer cost. This tradeoff is highly desired for low volume production since the current mask set cost (around \$ $1M$) is much larger than the wafer cost (around \$ $3.5K$).

In this paper, we propose to use Multi-Flow Multi-Layer Multi-Project Reticles (MFMLMPR) which combine all available mask cost saving technologies to achieve the maximum manufacturing cost saving. An example of MFMLMPR is shown in Figure 3, where Die 1 and Die 2 are two dies of different technology flows. To our

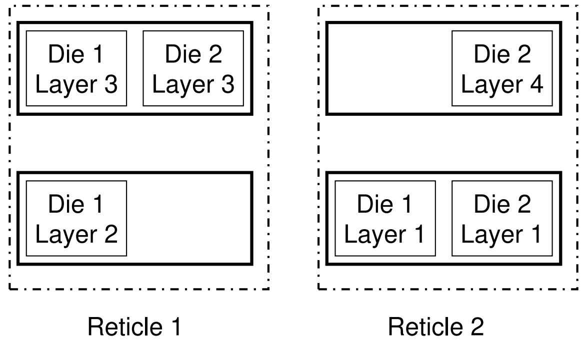

Figure 3. An example of Multi-Flow Multi-Layer Multi-Project Reticles.

best knowledge, we are the first to consider design optimization for MFMLMPR. Our flow includes three main steps: (1) schedule-aware project partitioning with multi-flow embedding, (2) multi-frame reticle design, and (3) multi-project frame floorplanning. Our contributions are as follows. For the first step, we propose a fast iterative matching procedure to find the mask cost estimation for multi-flow embedding. A sliding timing window heuristic is proposed to achieve the best tradeoff between mask cost and delay cost. This optimization is shown to yield an average reduction of around 58.4% in the manufacturing cost. For the second step, we propose an ASAP frame embedding heuristic to minimize the mask cost. The reduction of mask cost is around 46.3% compared with traditional layer-by layer checking methods. Finally, for the third step, we propose a "general chessboard" floorplan with simulated annealing to generate more "diceable" frame floorplans for multi-flow projects observing given maximum reticle sizes. Our algorithm leads to an average reduction of 22.8% in the required number of wafers compared to the previous best reticle floorplanner.[7]

2. PRELIMINARIES AND DESIGN FLOW

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.

Following,[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 is called a *wafer dicing plan.* The *dicing yield* of a 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 $D \in \mathcal{D}$, which needs to be at least 1.

In order to print a die image onto the wafer, we need a set of reticles, each of which contains a *layer* image of the die. The main properties of a layer include grade, tonality, cost (the minimum reticle cost of the layer) and mask process type. Two layers are *compatible* if and only if they can share one reticle.* One reticle may have several non-overlapping *frames*, each of which is a rectangular area with some compatible layer images. In lithography, only the layer images of one frame will be exposed. The distance between any two frames should be larger than the *blading space* to avoid interference between frames. In practice, there is a limit on the number of

*In general, a layer of higher grade is more expensive. If two layers have the same tonality and mask process type, they can share one reticle whose grade is the higher grade of the two layers. Two layers with different tonality can share one reticle only if the grade of one layer is much smaller than the grade of the other layer.

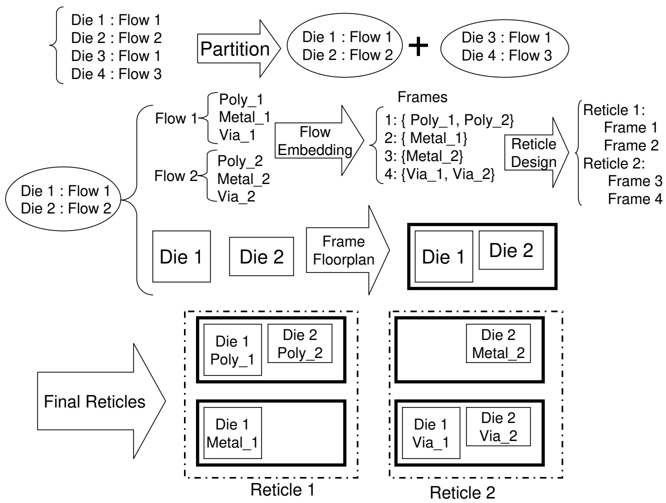

Figure 4. An example of MFMLMPR design flow.

frames one reticle.[†] As shown in Figure 2, a *flow* contains a sequence of *layers* to be printed onto the wafer in order to obtain one die with the flow. Obviously, each frame cannot contain more than one layer of a flow.

The following schemes give details of three main steps in our proposed MFMLMPR design flow: (1) schedule-aware project partitioning with multi-flow embedding, (2) multi-frame reticle design, and (3) multi-project frame floorplanning. As shown in Figure 4, the projects are partitioned into several sets in the first step to minimize the sum of delay cost and mask cost. The projects of each set are simultaneously manufactured. The frames for the projects of different flows are designed with multi-flow embedding. Then we divide the frames into several sets such that the frames of the same set can be placed on one reticle in Step 2. Finally, we need to design the frame floorplan to minimize the number of wafers used.

3. SCHEDULE-AWARE PROJECT PARTITIONING

We propose the following delay model for any project d:
$Delay\_cost(d) = T(d) * P(d) * N(d)$, where $T(d)$ is equal to the difference between the tapeout schedule date of d and the latest tapeout schedule date of the projects on the same reticles, $P(d)$ is the profit of d equal to the difference between sale price and manufacturing cost of final product, and $N(d)$ is the number of final products to be manufactured per week.

The *project partitioning problem* is formulated as follows:

Project Partitioning Problem (PPP). Given a set of projects of different flows S and tapeout schedule for each project, find a partition of projects into sets such that the sum of the delay cost and the manufacturing cost is minimized.

To simplify the mask cost evaluation of a set, we can assume that the manufacturing cost is equal to the flow embedding cost, and that the wafer cost is ignorable compared with mask cost and delay cost.[‡] If we use a binary variable x_i for each subset of S to denote whether the subset S_i is used in the partition solution, PPP can be formulated as the following integer linear program (ILP):

$$\text{Minimize } \sum_i c_i x_i \qquad \text{(ILP)}$$

[†]A small frame will result in too many exposures and wafer over-heating.

[‡]This assumption is reasonable for prototyping and low-volume production since the number of wafers to be used is small.

Input: A set of projects S to be partitioned; Properties of each project d (flow, tapeout schedule, $P(d)$ and N_d)
Output: Partition of the dies into m sets
1. Sort all reticles according to tapeout schedules 2. Place the first $2k$ projects into a "window" 3. **While** (the last project is not in the window) 4. Solve the ILP for the projects in the window 5. Sort partitioned sets according to the latest schedule 6. Delete partitioned sets in sequence from the window until the project number in the window $\leq k$ 7. Place the next k projects in the window 8. Solve the ILP for the projects in the window

Figure 5. Sliding Window Partition Algorithm.

subject to

$$\sum_{D \in S_i} x_i = 1, \quad \forall D \in S$$
$$x_i \in \{0,1\} \qquad\qquad \forall i$$

where c_i is equal to the sum of the delay cost of the subset S_i and the mask cost of all the flows of the projects in S_i which can be calculated with the iterative matching heuristics specified in the next section. The constraint means that one project belongs to exactly one subset in the partitioning solution. We observe that the projects of the same flow and tapeout schedule should always belong to the same set, and hence they can be merged into one "combined project" to simplify the problem.

Although PPP can be solved optimally with the ILP, the runtime becomes impractical when the number of projects in S is large since the number of subsets is $2^{|S|}$. Therefore, we propose the *Sliding Window Partition Algorithm* to solve this problem as shown in Figure 8. The reticles are first sorted by tapeout schedules from the earliest to the latest. Then we use a "window" to include all the projects to be partitioned. We limit the project number to $2k$ to enable an optimal partition in practical time. Initially, we put the first $2k$ projects in the window. In each iteration, we solve the ILP for the projects in the window to obtain the optimal set partition. We sort partitioned sets according to the latest schedule of the projects in the sets, and delete the partitioned sets in sequence until the number of projects in the window is at most k. Then we "slide" the window to include the next k projects. The process ends when all projects are partitioned into m sets; the projects of each set are simultaneously manufactured.

4. MULTI-FLOW EMBEDDING

One challenge of the MFMLMPR design is to partition layers of different flows to be simultaneously manufactured into multiple frames to minimize the mask cost. Let there be f flows, with each flow i containing a_i layers $L(i,j)$ $j = 1,\ldots,a_i$. The *layer compatibility graph* $G = (\mathcal{L}, E)$ is the graph with flow nodes corresponding to the layers of the flows $L(i,j)$ $i = 1,\ldots,f$ and $j = 1,\ldots,a_i$; and edges connecting pairs of compatible layers with weights equal to the larger cost of the two layers. Figure 6 (a) shows the layer compatibility graph for the example in Figure 2. We define a *multi-flow embedding* as a partition of layers of n flows into m sets such that each set contains at most one layer of each flow and the layers of each set are compatible. The cost of each set is the maximum cost of all layers in the set and the *total embedding cost* is the sum of the costs of all the sets. We may formulate the multi-flow embedding problem as follows.

Multi-Flow Embedding Problem (MFEP). Given a layer compatibility graph $G = (\mathcal{L}, E)$, find a multi-flow embedding to minimize the total embedding cost.[§]

The current industry method adopts a "layer-by-layer" checking heuristic which follows the physical layer sequence. That is, we first label the layers from bottom for all the flows, then we partition the i^{th}-layers into

[§]With the assumption that all products of all the flows can be placed within one frame and that the number of frames is proportional to the number of reticles, the total embedding cost is proportional to the mask cost.

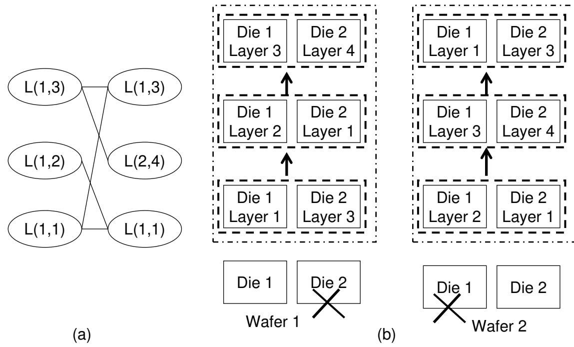

Figure 6. An example of layer compatibility graph (a) and its optimal multi-flow embedding (b).

the minimum number of sets such that all layers of the same set are compatible. As shown in Figure 2, the first layers of the two flows are placed in one reticle since they are compatible; the second layers of the two flows are placed in two different reticles since they are not compatible. However, it is unnecessary to follow the physical layer sequence as long as layers at one level are not compatible, since the dies of different flows will be successfully printed on different wafers with different frame exposure sequence.[¶] The optimal embedding is shown in Figure 6(b) in which only three sets (reticles) are needed.

We propose a new *Iterative Matching Heuristic (IMH)* as shown in Figure 7, to minimize the total cost of multi-flow embedding. For the given layer compatibility graph G, we merge two flows into a "combined flow" in every iteration (Lines 3-7). We always choose two flows with the most edges between them. (Intuitively, they have more layers to be shared.) Line 4 constructs a subgraph G' with the nodes of the two flows and edges between them. Then we add pseudo nodes to G' in order to reduce the embedding problem to the perfect matching problem, which can be solved optimally in Line 6.[‖] From the matching solution, we can merge the two flows into a "combined flow" in Line 7 by deleting all edges between two flows and merging two matched nodes into one node. In the end, all flows are merged into one combined flow and every node in G represents a set of layers.

5. MULTI-FRAME RETICLE DESIGN

For a given set of frames to be manufactured, we need to place them onto reticles to minimize the total mask cost. The *Multi-Frame Reticle Design*is formulated as follows:

Multi-Frame Reticle Design (MFRD). Given a set of frames and the maximum number of frames on one reticle, find the partition of frames into reticles such that all frames on one reticle are compatible and the total reticle cost is minimized.

We use a simple yet efficient algorithm to design the reticles which utilizes the fact that the price of the frames with higher grade is more expensive. All frames are first sorted according to grade. The first frame is placed on one reticle. In each iteration, we try to place one frame in sequence. We first search all existing reticles which are not full to try to find a place for the frame in Lines 4-6. The frame will be placed on a new reticle if it cannot be placed on any existing reticle.

6. MULTI-PROJECT FRAME FLOORPLANNING

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

[¶] If two flows have the same number of layers and all corresponding layers are compatible, we can treat them as one flow.

[‖] We can prove that the optimal matching solution of G' leads to the minimum total cost for two flows. The proof is omitted for brevity.

Input: Layer compatibility graph $G = (\mathcal{L}, E)$ with f flows

Output: m sets of layers with minimum total cost

1. Initially, each node in G represents one layer
2. **For** $(k = 1; k < f; k++)$
3. Find two flows i, j with the most edges between them
4. Obtain G' with $a_i + a_j$ nodes and edges between i and j
5. Add $a_i + a_j$ pseudo nodes to G', for each pseudo node:
   (1) add one edge of 0 weight to each other pseudo node;
   (2) add one edge to each flow node whose weight is equal to the cost of the corresponding layer.
6. Find the min-weight perfect matching solution of G'
7. Merge two flows into one "combined flow":
   (1) Delete the edges between the two flows from G
   (2) If two flow nodes n_1 and n_2 are matched in G', replace them with a new node n_3 in G which represents all layers of n_1 and n_2;
   The cost of n_3 is equal to the higher cost of n_1 and n_2.
   For any node $n \notin G'$: if and only if n is compatible with all layers of n_3, add one edge between n and n_3 whose weight is the higher cost of n_3 and n.
8. In the end, every node in G represents a set of layers

Figure 7. Iterative Matching Heuristics for Embedding.

Input: A set of projects S to be partitioned;
Properties of each project d (flow, tapeout schedule, $P(d)$ and N_d)

Output: Partition of the dies into m sets

1. Sort all frames according to grade
2. Place the frame with the highest grade on one reticle
3. **For** (all frames f)
4. **For** (all existing reticles which are not full)
5. **If** (f is compatible with all frames on the reticle)
6. Place the frame in the reticle and break
7. **If** (f is not placed)
8. Place f on a new reticle

Figure 8. ASAP Reticle Design Algorithm.

Although there are several approaches for multi-project reticle floorplanning,[5–7] they are based on the single-flow scenario. For multi-flow frame floorplanning, we need to consider the fact that only the dies of *one* flow can be successfully printed on each wafer. Therefore, it is unnecessary to avoid dicing conflicts between dies of different flows since they cannot be simultaneously obtained from one wafer. In this section, we propose a new "generalized chessboard mesh" based frame planning algorithms. The generalized chessboard mesh is a $g \times g$ grid mesh. As shown in Figure 9(a), the grids are labeled with numbers 1 to f for dies of f flows in the following way: the grids of the first row are repeatedly labeled from 1 to f; the label of the $j + 1^{st}$ grid of the $(i+1)^{st}$ row is the label of the j^{th} grid of the i^{th} row. Then we place the dies into the mesh such that the dies of flow j can only be placed in the grid labeled with j and only one die can be placed in each grid. The constructed mesh is "soft" since the height of each row is equal to the height of the highest die of the row and the width of each column is equal to the width of the widest die of the column. Figure 9(b) shows two different floorplans for the same mesh in Figure 9(a). There are two main advantages of the proposed mesh structure. First, the number of conflicts between dies of the same flow is minimized since there are no conflicts between dies located in diagonal grids. Second, the wafer cost can be easily evaluated due to the small number of maximal independent sets.[**]

[**] To evaluate the wafer cost, we can use the integer linear programming method in[7] by changing the definition of a *maximum conflict independent set* to a maximal set of dies *of the same flow* which can be sliced out by a set of horizontal

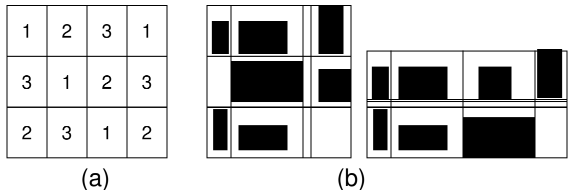

Figure 9. An example of the generalized chessboard mesh for three flows (a) and two different frame floorplans based on the mesh (b).

Cases	# dies	Total Vol.	Max Vol.	Min Vol.	Die area(cm^2)	# layers
Ind 1	12	330	40	25	1.13	23
Ind 2	14	275	25	6	1.36	22
Ind 3	24	775	67	25	1.82	23
Ind 4	31	755	30	8	1.62	24
Ind 5	14	250	25	12	0.86	23
Ind 6	24	625	35	25	2.26	24

Table 1. CMP testcase parameters.

The generalized chessboard mesh floorplanning algorithm (Figure 10) starts by assigning each die randomly to one grid in the mesh. The mesh size g is chosen such that the grid number of each flow is greater than its die number. The wafer cost is calculated and recorded for this initial floorplan. At each step we find a neighbor solution based on the following moves:

- **Grid exchange move:** exchange the dies in two grids of the same flow; if one of the grids is empty, this amounts to moving a die from one grid to another.
- **Orientation move:** rotate one die by 90 degrees if die width and height are different.

After each move, we evaluate the objective function for the resulting floorplan. To enforce the given maximum reticle dimensions, the objective value is set to infinity when the evaluated floorplan's dimensions exceed allowed maximums (unless we have not yet found a feasible floorplan, in which case the algorithm switches to using floorplan area as objective function). As in any simulated annealing algorithm, improving moves are always accepted, while remaining ones are accepted with a probability exponentially decreasing with the objective value increase and the current annealing temperature. If the floorplan is not feasible after the simulated annealing process, we can use the floorplan compaction, which removes the empty space of the rows exceeding the width limit and the empty space of the columns exceeding the height limit. The chessboard mesh structure may be destroyed during the compaction process. Finally, we we may greedily insert additional copies of dies if there is empty space left (Line 17).

7. EXPERIMENTS

We used six industry testcases from CMP[15] 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. We assume that they belongs to four different flows with 22 to 25 layers. 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++. The sale price, manufacturing cost and production volume of the final products are assumed based on the typical industry values shown in.[3] The tapeout schedules for all projects are randomly generated between zero and twelve weeks. The project partitioning results are summarized in Table 2. Here "Single Project" denotes the

---

and vertical cutlines.

Input: Dimensions and volume requirement of n dies, $0 \leq \beta < 1$
Output: Frame floorplan and wafer printing and dicing plan
1. Construct the $g \times g$ generalized chessboard mesh 2. Assign the n dies to grids of its flow at random 3. **If** (floorplan width and height < maximum dimensions) 4. $\quad FoundFeasible \leftarrow True$ 5. **Else** $FoundFeasible \leftarrow False$ 6. **While** (not converged and # of moves < $Move\_Limit$) 7. $\quad$ Pick a move at random 8. $\quad$ **If** (floorplan width and height < maximum dimensions) 9. $\qquad FoundFeasible \leftarrow True$; $\delta \leftarrow$wafer cost difference 10. $\quad$ **Else if** ($FoundFeasible = False$) 11. $\qquad \delta \leftarrow$ area difference 11. $\qquad$ **Else** $\delta \leftarrow \infty$ 12. $\quad$ **If** ($\delta < 0$) then accept the move 13. $\quad$ **Else** accept the move with probability $e^{-\frac{\delta}{T}}$ 14. $\quad T \leftarrow \beta T$ 15. **If** ($FoundFeasible = False$) 16. $\quad$ Floorplan compaction 17. Greedily insert additional copies to fill empty space

Figure 10. The generalized chessboard mesh floorplanning algorithm.

Cases	Single Project	Proposed
Ind 1	\$ $6M$	\$ $3.3M$
Ind 2	\$ $7M$	\$ $3.7M$
Ind 3	\$ $12M$	\$ $4.35M$
Ind 4	\$ $15.5M$	\$ $4.7M$
Ind 5	\$ $7M$	\$ $4.05M$
Ind 6	\$ $12M$	\$ $4.65M$
Total	\$ $59.5M$	\$ $24.75M$
Reduction (%)	0	58.4

Table 2. Project partitioning results for six testcases.

sum of mask cost and delay cost for single-project wafers with zero delay cost, and "Proposed" is our proposed sliding window partition algorithm. The results show that our proposed greedy merge algorithm can reduce cost by 58.4% compared with the traditional project partitioning.

Flow Embedding and Reticle Design. Our algorithm for the flow embedding and reticle design problem is implemented in C++. The layers and their manufacturing costs are assumed based on the typical industry values. The flow embedding and reticle design results are summarized in Table 3. Here "Layer-by-layer Checking" denotes the sum of mask cost for the traditional layer-by-layer checking heuristic according to the physical layer sequence, "Proposed IMH" is our proposed iterative matching based algorithm. The results show that our proposed iterative matching based algorithm can reduce mask cost by 46.3% compared with the traditional layer-by-layer checking heuristic.

Frame Floorplan. Our proposed generalized chessboard floorplanner is implemented in C++. The frame floorplan results are summarized in Table 4. Here "CMP" denotes the original industry floorplan used by CMP, "HQ" denotes the wafer cost generated by the hierarchical quadrisection algorithm proposed in,[7] "Generalized Chessboard" is our proposed multi-flow frame floorplanning algorithm. N_w is the number of used wafers. The results show that our proposed multi-flow frame floorplanning algorithm can reduce wafer cost by 43.2% compared with the original CMP floorplan, and by 22.8% compared with the hierarchical quadrisection algorithm.

Cases	Layer-by-layer Checking	Proposed IMH
Ind 1	\$ 2.42M	\$ 1.32M
Ind 2	\$ 2.33M	\$ 1.16M
Ind 3	\$ 3.62M	\$ 1.73M
Ind 4	\$ 3.41M	\$ 1.64M
Ind 5	\$ 2.63M	\$ 1.46M
Ind 6	\$ 3.55M	\$ 2.33M
Total	\$ 17.96M	\$ 9.64M
Reduction (%)	0	58.4

Table 3. Flow embedding and reticle design results for six testcases.

Cases	CMP		HQ[7]			Generalized Chessboard		
	N_w	area	N_w	area	CPU(s)	N_w	area	CPU(s)
Ind 1	6	1.13	5	1.42	0.00	4	1.49	0.00
Ind 2	6	1.36	5	1.65	0.00	4	1.76	0.01
Ind 3	8	1.82	6	2.26	0.01	4	2.54	0.02
Ind 4	8	1.62	6	1.82	0.01	5	2.01	0.03
Ind 5	4	0.86	3	1.19	0.00	2	1.35	0.00
Ind 6	12	2.26	7	2.66	0.01	6	2.82	0.04
Total	44		32			25		
Red.(%)	0		27.2			43.2		

Table 4. Frame floorplan results for six testcases. "CMP" is the original industry floorplan used in CMP, "HQ" is the proposed hierarchical quadrisection floorplan algorithm proposed in,[7] and "Generalized Chessboard" is our proposed multi-flow frame floorplanning algorithm. N_w is the number of used wafers.

8. CONCLUSION AND FUTURE WORK

In this paper we propose a framework for optimization of Multi-Flow Multi-Layer Multi-Project Reticles (MFMLMPR), which combines all available mask cost-saving technologies to achieve the maximum manufacturing cost saving. Experiments on industry testcases show that our methods can significantly reduce the manufacturing cost. In on-going work we investigate the use of multiple die copies on the reticle for further reductions in the manufacturing cost of given die production volumes.

REFERENCES

1. M. Andersson, C. Levcopoulos and J. Gudmundsson, "Chips on Wafers", *Proc. WADS (Workshop on Algorithms and Data Structures)*, August 2003, pp. 412-423.
2. M. Andersson, C. Levcopoulos and J. Gudmundsson, "Chips on Wafers", *Computational Geometry - Theory and Applications*, 30(2), 2005, pp. 95-111.
3. 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.
4. S. Chen and E. C. Lynn, "Effective Placement of Chips on a Shuttle Mask", *Proc. SPIE*, Vol 5130, 2003, pp. 681-688.
5. A. B. Kahng, I. I. Mandoiu, Q. Wang, X. Xu and A. Zelikovsky, "Multi-Project Reticle Floorplanning and Wafer Dicing", *Proc. Intl. Symp. on Physical Design*, April 2004, pp. 70-77.
6. A. B. Kahng and S. Reda, "Reticle Floorplanning With Guaranteed Yield for Multi-Project Wafers", *Proc. International Conference On Computer Design*, October 2004, pp. 106-110.
7. A. B. Kahng, I. I. Mandoiu, X. Xu and A. Zelikovsky, "Yield-Driven Multi-Project Reticle Design and Wafer Dicing," *Proc. SPIE*, Volume 5992, 2005, pp. 1247-1257.
8. M. LaPedus, "The IC industry Heading to the \$10 Million Photomask?", *Semiconductor Business News*, Oct. 7, 2002, `http://www.siliconstrategies.com/story/OEG20021007S0053`
9. R. D. Morse, "Multiproject Wafers: Not Just For Million Dollar Mask Sets", *Proc. SPIE*, Vol 5043, 2003, pp. 100-113.
10. M.-C. Wu and R.-B. Lin, "A Comparative Study on Dicing of Multiple Project Wafers", *Proc. ISVLSI*, 2005, pp. 314-315.
11. M.-C. Wu and R.-B. Lin, "Reticle Floorplanning and Wafer Dicing for Multiple Project Wafers", *Proc. Intl. Symposium on Quality Electronic Design*, 2005, pp. 610-615.

12. G. Xu, R. Tian, M. D. F. Wong and A. Reich, “Shuttle Mask Floorplanning”, *Proc. SPIE*, Volume 5256, 2003, pp. 185-194.
13. 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.
14. G. Xu, R. Tian, D. Z. Pan and M. D. F. Wong “CMP Aware Shuttle Mask Floorplanning”, *Proc. Asia South Pacific Design Automation Conference*, 2005, pp. 1111-1114.
15. CMP: Circuits Multi-Projets. `http://cmp.imag.fr/`