Multi-Project Reticle Design and Wafer Dicing under Uncertain Demand*

Andrew B. Kahng, Ion Măndoiu†*, Xu Xu, and Alex Z. Zelikovsky*‡

CSE Department, UC San Diego, La Jolla, CA 92093, USA
†CSE Department, University of Connecticut, 371 Fairfield Rd., Storrs, CT 06269, USA
‡CS Department, Georgia State University, University Plaza, Atlanta, GA 30303, USA
{abk,xuxu}@cs.ucsd.edu, ion@engr.uconn.edu, alexz@cs.gsu.edu

Abstract

The pervasive use of advanced reticle enhancement technologies demanded by VLSI technology scaling leads to dramatic increases in mask costs. In response to this trend, multiple project wafers (MPW) have been proposed as an effective technique for sharing the cost of mask tooling among up to tens of prototype and low volume designs. Previous works on MPW reticle design and dicing have focused on the simple scenario in which production volumes are known a priori. However, this scenario does not apply for low- and medium-volume production, in which mask manufacturing is typically started when only rough estimates of future customer demands are available. In this paper we initiate the study of MPW use for production under demand uncertainty and propose efficient algorithms for two main optimizations that arise in this context: reticle design under demand uncertainty and on-demand wafer dicing. Preliminary experiments on simulated data show that our methods help reducing the cost overheads incurred by demand uncertainty, yielding solutions with a cost close to that achievable when a priori knowledge of production volumes is available.

1 Introduction and Motivation

With 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 prototype, low-volume, and low-price medium-volume production designs past the limits of economic feasibility since the costs cannot be amortized over the volume. In response to this trend, multiple project wafers (MPW) have been proposed as an effective technique for sharing the cost of mask tooling among up to tens of designs [3, 9].

Multi-project reticle design and wafer dicing have received much attention recently. Xu et al. [11] 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 was proposed in [2], where the objective is to find a minimum area grid floorplan with at most one die per grid cell. The grid-packing formulation in [2] was revisited by Kahng and Reda [7], who proposed a new floorplanner with guaranteed yield. Kahng et al. [5] considered the side-to-side wafer dicing problem, and proposed a general multi-project reticle floorplanning method seeking to maximize dicing yield. Xu et al. [12] combined the horizontal and vertical conflict graphs of [5] into a single conflict graph, and proposed dicing algorithms based on coloring of the combined conflict graph (all dies receiving a certain color can be diced in the same step). Wu et al. [14] proposed integer linear program (ILP) formulations for wafer dicing, further extending the graph coloring approach. In [15], Wu et al. gave ILP formulations for simultaneous floorplanning and dicing (which, unfortunately, cannot be solved in practical runtime even for small testcases) and proposed more practical algorithms for independent reticle floorplanning and wafer dicing, also incorporating chip replication in the former. Recently, Kang et al. [6] proposed (1) algorithms for balancing mask cost and schedule delay cost, (2) a new hierarchical quadrisection floorplanning algorithm based on simulated annealing, (3) shot-map optimization methods for maximizing the number of functional dies extracted from each wafer, and (4) new side-to-side dicing algorithms allowing multiple dicing plans for different wafers.

All previous works on MPW reticle design and dic-

*Work supported in part by Cadence Design Systems, Inc., the MARCO Gigascale Silicon Research Center, a Faculty Large Research Grant from the University of Connecticut Research Foundation, NSF Grant CCF-0429735, and CRDF Award #MOM2-3049-CS-03.

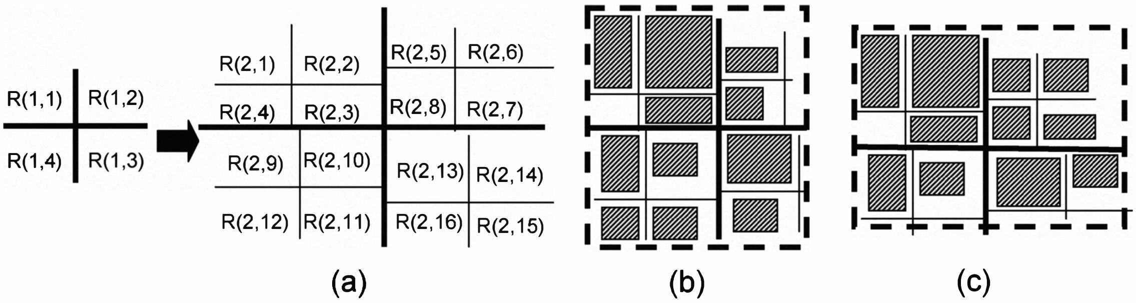

Figure 1: (a) 2-level hierarchical quadrisection floorplan mesh. (b)–(c) Two different floorplans obtained from different assignments of dies to mesh regions.

ing have focused on the simple scenario in which production volumes are fully known at reticle design time. While this assumption holds for prototype manufacturing, it may not hold for low- and medium-volume production. Due to the time-to-market pressure, the mask set must be manufactured as soon as possible, when only rough lower and upper bounds on customer demand are available. Once the mask set is available, lots of wafers can be manufactured in response to customer orders. To increase profitability, it is possible to manufacture larger wafer lots in anticipation of future customer demand, and then dice the wafers as customer orders arrive. Multi-project wafers become even more attractive in this context because, in addition to mask cost sharing, they allow reducing the risks associated with misprediction of customer demand.

In this paper we initiate the study of MPW use for low- and medium-volume production under demand uncertainty and address the two main optimizations that arise in this scenario: reticle design under demand uncertainty and on-demand wafer dicing. Our contributions include novel simulated annealing (SA) algorithms for *robust* reticle floorplanning under demand uncertainty (Section 2). A key enabler of solution quality is the integration within the SA framework of *project replication*, or *cloning*. Unlike in [15], where project replication was proposed as a post-processing step intended to use the white space left on the reticle, our algorithm works with multiple copies (and can dynamically adjust their number) throughout the entire solution search process, therefore resulting into both a better use of reticle area and improved dicing properties. Our cloning strategy further allows full control of reticle size, and should also be useful for MPW reticle design in the case when production volumes are known. We also give algorithms for on-demand wafer dicing (Section 3). For both reticle design and on-demand dicing, our best algorithms require little or no knowledge about customer order distribution. Nevertheless, experiments on simulated data show that our algorithms are very effective in reducing overheads incurred by demand uncertainty, coming very close in solution quality to algorithms that rely on a priori knowledge of customer orders (Section 4).

2 Robust Reticle Floorplanning

Reticle floorplanning is perhaps the most important optimization step of any MPW flow. Compared with traditional chip floorplanning, the difficulty of MPW reticle floorplanning lies in the complex relationship between the reticle floorplan and overall manufacturing cost, via conflicting factors such as reticle area and dicing compatibility [15]. Manufacturing cost estimation becomes even more difficult when production volumes are uncertain. Two reticle floorplans that require the same number of wafers for satisfying a certain set of production volumes may require significantly different numbers of wafers even for slight changes in the production volumes. Ideally, we would like a reticle floorplan that leads to low production costs under most possible production requirements. This *robustness* objective for reticle design under production demand uncertainty can be formulated as follows:

Robust Reticle Design Problem (RRDP).
Given: Maximum reticle size, dies $\mathcal{D} = \{D_1, \ldots, D_n\}$, and probability distribution of customer orders for each die.
Find: a reticle floorplan for $\mathcal{D}$ that maximizes the *expected* number of wafers required to satisfy customer orders over the time horizon for which production is being planned.

Our RRDP algorithm uses the simulated annealing (SA) framework introduced in [6]. As in [6], a hierarchical quadrisection mesh is defined recursively as follows:

- At level 1, the full reticle area is divided via one horizontal line and one vertical line into 4 rectangular regions denoted $R(1,1)$, $R(1,2)$, $R(1,3)$, and $R(1,4)$, respectively.
- In general, each level $i \geq 1$ has 4^i regions denoted $R(i,j)$, $1 \leq j \leq 4^i$. Regions at level $i+1$ are obtained by dividing each region $R(i,j)$ at level i into

```
Input: Dimensions and volume requirement distributions of $n$ dies, parameter $\beta$, $0 \leq \beta < 1$
Output: Reticle floorplan and wafer dicing plan
Construct the hierarchical quadrisection floorplan mesh
Assign the $n$ dies to regions at random
If (floorplan width and heigh smaller than maximum reticle dimensions) then $FoundFeasible \leftarrow True$
Else $FoundFeasible \leftarrow False$
While (not converged and # of moves < $Move\_Limit$)
    Pick a move at random
    If (floorplan width and heigh smaller than maximum reticle dimensions) then
        $FoundFeasible \leftarrow True$; $\delta \leftarrow$New Objective Value - Old Objective Value
    Else, if ($FoundFeasible = False$) then $\delta \leftarrow$ New Area - Old Area, else $\delta \leftarrow \infty$
    If ($\delta < 0$) then accept the move, else accept the move with probability $e^{-\frac{\delta}{T}}$
    $T \leftarrow \beta T$
```

Figure 2: Hierarchical quadrisection floorplanning algorithm.

4 new regions, $R(i+1,4^{j-1}+1)$, $R(i+1,4^{j-1}+2)$, $R(i+1,4^{j-1}+3)$ and $R(i+1,4^{j-1}+4)$, respectively, via one horizontal line and one vertical line.

The number of levels l is chosen such that the number of regions at level l, 4^l, is greater than the number of dies. Figure 1(a) shows a mesh with 2 levels.

In our algorithm, each one of the n given dies is assigned to a different region of the hierarchical quadrisection mesh. The mesh itself is "soft" since the dimensions of each mesh region are determined by the die assignment. As shown in Figure 1 (b) and (c), different die assignments lead to different region dimensions and therefore to different reticle floorplans. We denote the width of the region $R(i,j)$ by $W(R(i,j))$ and the height by $H(R(i,j))$. For a given die assignment, region heights and widths can be computed in a bottom-up manner, as follows. At level l, if there is a die in the region $R(l,j)$, then $W(R(l,j))$ is set to the width of the die and $H(R(l,j))$ is set to the height of die; otherwise, $W(R(l,j)) = H(R(l,j)) = 0$. At level $i < l$, $W(R(i,j)) = \max\{W(R(i+1,4^{j-1}+1)), W(R(i+1,4^{j-1}+4))\} + \max\{W(R(i+1,4^{j-1}+2)), W(R(i+1,4^{j-1}+3))\}$ and $H(R(i,j)) = \max\{H(R(i+1,4^{j-1}+1)), H(R(i+1,4^{j-1}+2))\} + \max\{H(R(i+1,4^{j-1}+3)), H(R(i+1,4^{j-1}+4))\}$.

A main advantage of the proposed hierarchical quadrisection mesh structure is that it guarantees dicing compatibility between dies placed in certain regions. In particular, dies located in diagonally opposing sibling regions can always be diced together. This property is used by our simulated annealing algorithm to quickly evaluate the quality of a reticle floorplan.

We call a set of mesh regions *independent* if the dies in these regions can be diced simultaneously. For an independent set S the *expected wafer requirement* is upperbounded by

$$W(S) = \int \max_{D \in S} \left\lceil \frac{N(D)}{Q(D)} \right\rceil \mathrm{d}N \tag{1}$$

where $N(D)$ is the random variable denoting the volume requirement for die D, and $Q(D)$ is the number of copies of die D per wafer for the evaluated floorplan. In our implementation we use a Monte-Carlo simulation to evaluate (1): we generate 100 random production volume vectors N according to the given distributions, and then average $\max_{D \in S} \left\lceil \frac{N(D)}{Q(D)} \right\rceil$ over these vectors.

To evaluate a given hierarchical quadrisection floorplan, we start by creating 4^l sets, each consisting of at most one die, corresponding to the level-l regions of the floorplan mesh. We then merge, in bottom-up order, pairs of sets coming from diagonally opposing sibling regions, thus ensuring that the merged sets remain independent. Note that the wafer requirement of a merged pair is the *maximum* of the wafer requirements of its two sets. Thus, to ensure the smallest total wafer requirement, when merging the sets coming from two diagonally opposing sibling regions we sort the sets in each region according to their wafer requirement, merge the sets with highest wafer requirement in each region, then merge the sets with second highest wafer requirement, and so on. Since at each level the merging process decreases the number of sets in half, we end up with 2^l independent sets covering all dies at level 1. The objective function used by our simulated annealing algorithm to evaluate floorplan changes is the sum of wafer requirement upperbounds computed using (1) for these 2^l independent sets.

The algorithm (Figure 2) starts by assigning each die randomly to one of the 4^l regions of the hierarchical quadrisection mesh. The objective function is calculated and recorded for this initial floorplan. At each step we find a neighbor solution based on the following moves:

- **Region exchange move:** exchange the dies in two regions; if one of the regions is empty this amounts to moving a die from one region 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.

2.1 Die Cloning

Most previous works on MPW reticle design assume that the reticle contains a single copy of each die. This is appropriate for prototype manufacturing, when only a small number of wafers is produced, since it minimizes reticle area (and hence total cost). For low- and medium-production volume the number of manufactured wafers is larger, and die cloning (i.e., using multiple copies of a die in the reticle) may be justified even when it leads to some increase in reticle area, since cloning can improve dicing yield and thus decrease the number of required wafers.

A simple die cloning method was proposed in [15], based on insertion of additional die copies within the white space available in a floorplan constructed starting with a single copy of each die. However, this post-processing approach has limited potential for improvement since typically there is not much empty space left on the reticle. Here, we propose a comprehensive approach to die cloning, which involves making cloning decisions *before*, *during*, as well as *after* running the simulated annealing algorithm in Figure 2.

First, we set the initial number of copies c_D for die D with average volume requirement V_D to

$$c_D = \max\{1, \beta f(V_D)\}$$

Here, $f(V_D)$ is a monotonically increasing function of V_D. We used $f(V_D) = V_D$ and $f(V_D) = \sqrt{V_D}$ in our experiments, the resulting algorithms are referred to as SA-clone1 and SA-clone2, respectively. Parameter β is a scaling factor chosen such that

$$\sum_{D \in \mathcal{D}} c_D a_D \leq \alpha A$$

where, a_D denotes the area of die D, A denotes the maximum reticle area and $\alpha \leq 1$ is a maximum reticle utilization factor, which was set to 0.6 in our experiments. To facilitate dicing, all copies of a die are arranged in a $k \times l$ "clone array" which is always assigned to a single floorplan mesh region.

We also modified the simulated annealing algorithm in Figure 2 by adding four new moves: addition/deletion of a row/column of copies from a clone array. Finally, after the completion of the algorithm, we try to insert additional rows or columns into the clone arrays without increasing reticle size.

3 On-Demand Wafer Dicing

A wafer consists of reticle projections ("flashes") arranged in a number of *projection rows* and *projection columns*. Each projection is an image of the reticle, which includes one or more copies of each die. Although all die copies on a wafer can be recovered by using expensive dicing technologies such as laser cutting [10], with the prevalent side-to-side wafer dicing technology some die copies will be destroyed. Under the side-to-side dicing model all reticles in the same row (column) on the wafer are sawed by a single set of horizontal (resp. vertical) cut lines, as the diamond blades cannot stop at arbitrary points during cutting.

3.1 Single Batch Dicing

Dicing a single batch of customer of orders can be formulated as the following *Side-to-Side Wafer Dicing Problem* (SSWDP): Given a multi-project reticle and required production volumes for each die, find the minimum number of wafers and a set of wafer dicing plans yielding for each die a number of copies equal to or greater than the required production volume. Xu et al. [12] assumed that each wafer uses exactly one horizontal dicing plan and one vertical dicing plan for all reticle image rows/columns. This assumption allowed them to use a coloring-based heuristic giving good results for testcases with large volume requirement. In [6] we have given an integer linear programming formulation which allows finding an *optimal* set of dicing plans restricted in this way. Next we refine the formulation in [6] to change the objective from pure minimization of the number of wafers to the minimization of the combined wafer *and* dicing cost, where the former is proportional to the number of wafers and the latter is proportional to the number of different wafer dicing plans used to fulfill the batch of orders.

As in [12], 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* 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 cut lines. 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 be an integer variable denoting the number of wafers which use the dicing plan defined by C. Also, let x_C be a 0/1 variable which is set to 1 if and only if the dicing plan defined by C is used to dice at least one wafer. Denoting by α the cost of a wafer and

```
Input: reticle floorplan with dies $\mathcal{D} = \{D_1, \ldots, D_n\}$, wafer shot-map,
       customer orders $Q_i$, $1 \le i \le m$
Output: number of wafers $N_i$ to be diced after receiving each order $Q_i$
For each $k = 1, \ldots, n$, $in\_stock(k) \leftarrow 0$
For $i = 1, \ldots, m$ do
    For each $k = 1, \ldots, n$
        If $in\_stock(k) \ge Q_i(k)$ then
            $in\_stock(k) \leftarrow in\_stock(k) - Q_i(k)$
            $Q_i(k) \leftarrow 0$
        Else
            $Q_i(k) \leftarrow Q_i(k) - in\_stock(k)$
            $in\_stock(k) \leftarrow 0$
    If $Q_i \ne 0$
        Run the SSWDP algorithm in Section 3.1 with production volumes given by $Q_i$
        Let $N_i$ be the number of wafers required by the algorithm, and
         $yield(k)$ be the resulting number of copies of die $D_k$
        For each $k = 1, \ldots, n$
            $in\_stock(k) \leftarrow in\_stock(k) + yield(k) - Q_i(k)$
    Else $N_i \leftarrow 0$
Return $N_i$, $i = 1, \ldots, m$
```

Figure 3: Greedy ODSSWDP algorithm.

by β the cost of reprogramming the dicing machine, we obtain the following integer linear program:

$$\text{Minimize } \alpha N_w + \beta N_{dp}$$

subject to

$$\begin{array}{ll}
\sum_{C \in MCIS} Q(C,D) f_C \ge N(D), & \forall D \in \mathcal{D} \\
N x_C \ge f_C, & \forall C \in MCIS \\
N_w = \sum_{C \in MCIS} f_C & \\
N_{dp} = \sum_{C \in MCIS} x_C & \\
f_C \in \mathbb{Z}_+, \; x_C \in \{0,1\} & \forall C \in MCIS
\end{array}$$

where $Q(C,D)$ is a constant which represents the number of copies of die D obtained from a wafer diced according to C and $N = \max_{D \in \mathcal{D}} N(D)$. This integer program can be optimally solved efficiently for practical SSWDP instances since there are only $O(|MCIS|)$ variables and $O(|\mathcal{D}| + |MCIS|)$ constraints.

3.2 Dicing Multiple Batches

The SSWDP formulation is appropriate in the context of current shuttle services, which are focusing on serving the prototyping needs of independent design companies. In this context, there is no uncertainty on the number of prototype copies required for each project since these are specified by the customers before reticle design, and all dicing can be done in a single batch. However, for low- and medium production the exact customer demand may not be known a priori. When a single company owns all designs on the MPW, it is advantageous to manufacture a large wafer lot in anticipation of future customer demand, and then dice the wafers only in response to incoming customer orders. This motivates the study of the following *on-demand* version of the dicing plan optimization problem:

On-Demand Side-to-Side Wafer Dicing Problem (ODSSWDP).
Given:

- A multi-project reticle floorplan with dies $\mathcal{D} = \{D_1, \ldots, D_n\}$,
- A wafer shot-map, i.e., the position of reticle images on the wafer, and
- A sequence of customer orders Q_i, $1 \le i \le m$, where each Q_i is an n-dimensional vector of non-negative integers

Find: number of wafers N_i to be diced after receiving each order Q_i and corresponding dicing plans
Such that:

- Each customer order is satisfied before receiving the next order, i.e., for every $k \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, m\}$, the number of copies of die D_k that result from dicing the first $\sum_{i=1}^{j} N_i$ wafers is at least $\sum_{i=1}^{j} Q_i(k)$ (we assume that excess die copies obtained in a dicing step are stored at no cost and can be used to satisfy customer orders in subsequent step.)
- Dicing decisions are made on-demand, i.e., for every i, the number of wafers N_i and the associated dicing plans are chosen *without* any knowledge of Q_j for $j > i$, and
- The total wafer and dicing cost is minimized.

```
Input: reticle floorplan with dies $\mathcal{D} = \{D_1, \ldots, D_n\}$, wafer shot-map,
       customer orders $Q_i$, $1 \le i \le m$
Output: number of wafers $N_i$ to be diced after receiving each order $Q_i$
01. For each $k = 1, \ldots, n$, $in\_stock(k) \leftarrow 0$, $past\_demand(k) \leftarrow 0$
02. For $i = 1, \ldots, m$ do
03.     For each $k = 1, \ldots, n$
04.         $past\_demand(k) \leftarrow past\_demand(k) + Q_i(k)$
05.         If $in\_stock(k) \ge Q_i(k)$ then
06.             $in\_stock(k) \leftarrow in\_stock(k) - Q_i(k)$
07.             $Q_i(k) \leftarrow 0$
08.         Else
09.             $Q_i(k) \leftarrow Q_i(k) - in\_stock(k)$
10.             $in\_stock(k) \leftarrow 0$
11.     If $Q_i \neq 0$
12.         $\alpha \leftarrow \max\{Q_i(k)/past\_demand(k) \mid past\_demand(k) \neq 0\}$
13.         For each $k = 1, \ldots, n$
14.             $Q'(k) \leftarrow \max\{0, \lceil \alpha \cdot past\_demand(k) \rceil - in\_stock(k)\}$
15.         Run the SSWDP algorithm in Section 3.1 with production volumes given by $Q'$
16.         Let $N_i$ be the number of wafers required by the algorithm, and
17.             $yield(k)$ be the resulting number of copies of die $D_k$
18.         For each $k = 1, \ldots, n$
19.             $in\_stock(k) \leftarrow in\_stock(k) + yield(k) - Q_i(k)$
20.     Else $N_i \leftarrow 0$
21. Return $N_i$, $i = 1, \ldots, m$
```

Figure 4: History-based ODSSWDP algorithm.

We remark that, although we refer to the demand vectors Q_i as "customer orders", they may represent customer orders aggregated over certain periods of time (e.g., daily, weekly, etc.) In Section 4.2 we will use this flexibility to gauge the benefits of batching customer orders for dicing purposes.

A simple greedy ODSSWDP algorithm is given in Figure 3. The algorithm keeps track of the existing die stock, which changes after each dicing step. For every incoming customer order, the algorithm tries first to use existing dies to satisfy as much as possible of the order. If the order is fulfilled using existing stock, no additional wafers are diced. Otherwise, the SSWDP algorithm in Section 3.1 is invoked with the remaining order balance as required production volume to determine how many additional wafers to dice (and what dicing plans to use for them). Finally, the die copies thus obtained are used to complete the order, and any leftover copies are stored for future use.

The algorithm in Figure 3 is attractive for its simplicity, but has a number of weaknesses. The SSWDP instances solved in Line 11 of the algorithm in Figure 3 will typically require only a few wafers. This means that a large fraction of the resulting die copies will end up being stocked for future use. These die are chosen by the SSWDP algorithm without considering the already existing stock or the demand trends that can be inferred from past customer orders. An improved ODSSWDP algorithm correcting these weaknesses, which we call "history-based", is given in Figure 4. In addition to tracking the existing stock, the improved algorithm also tracks the past order history. When a customer order cannot be fulfilled using the existing stock, instead of calling the SSWDP algorithm with production volumes given by the remaining balance, we call it with a vector of production volumes given by the past demand scaled down as much as possible while still ensuring that we can satisfy the remaining order balance, and further adjusted by subtracting existing stock quantities (Lines 12-14).

Testcase	# dies	Min Vol.	Max Vol.	Die area(cm^2)
Ind1	12	200	20000	1.13
Ind2	14	100	10000	1.36
Ind3	24	200	20000	1.82
Ind4	31	100	20000	1.62
Ind5	14	100	10000	0.86
Ind6	24	100	3000	2.26

Table 1: CMP testcases parameters.

4 Experimental Results

To evaluate the performance and scalability of the proposed algorithms, we used six industry testcases from CMP [1], each having between 12 and 31 dies with varying sizes. For each die, we assumed upper and lower bounds on the production volume requirements as given in Table 1.

4.1 Reticle Design

We included in our comparison several floorplans:

- The industry floorplan designed by CMP engineers (CMP);
- The floorplan obtained by running the hierarchical quadrisection algorithm in [6] (HQ) with production volumes set to the median point of the expected distribution, i.e., to the average between the minimum and the maximum expected customer orders;
- The floorplan obtained by running the simulated annealing algorithm in Figure 2 driven by the uniform, respectively normal distributions for customer orders (SA-unif and SA-norm); and
- The floorplans obtained by running the simulated annealing algorithm with cloning, using an initial number of clones which is proportional to the average production volume (SA-clone1), respectively to the square root of the average production volume (SA-clone2). In these versions of the algorithm we used a simpler SA objective instead of the Monte Carlo simulation used in SA-unif and SA-norm, namely the number of wafers needed to satisfy the average production volumes.

Table 2 gives the observed average and standard deviation of the number of wafers required to fulfill 100 random production requirement vectors generated for each die according to uniform, respectively normal probability distributions. Dicing was done in all cases using the integer linear programming SSWDP algorithm in Section 3.1 with $\alpha = 1$ and $\beta = 0$. For comparison, we include in the table the observed average and standard deviation for the number of wafers obtained by independently running the hierarchical quadrisection algorithm of [6] for each of the 100 random production volume requirements (HQ*). HQ* can be used to estimate the reticle floorplanning suboptimality incurred due to demand uncertainty, since in its computation we allow selecting an individual floorplan for every production volume vector. (Note that HQ* is not a true lower-bound since HQ does not guarantee optimality.) The results show that the two cloning based SA algorithms give the best results, often better even than HQ*, despite the unfair advantage of the latter. The cloning algorithm which starts with a number of clones proportional to the square root of average production volumes gives best results, improving over CMP floorplans by an average of 33% for production volumes generated from uniform distributions, and by an average of 28% for production volumes generated from normal distributions. Most of the improvement is due to the use of cloning, as can be seen from the fact that the SA algorithms without cloning give significantly worse results (although still better than CMP when driven by the correct distribution).

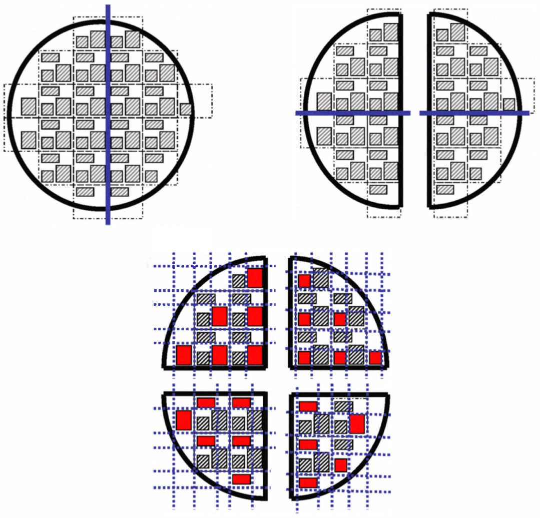

Figure 6: Four quadrant dicing: the wafer is first divided into four quadrants, then each quadrant is diced independently using side-to-side cuts.

In Figure 5 we explore another measure of floorplan robustness. Here, we plot the tradeoff curve between the number of wafers and the probability of satisfying customer orders generated from the underlying distributions. To efficiently determine these tradeoffs, we used the following Monte-Carlo simulation:

- First, we generated a large number Q of random production volume vectors according to the given distributions.
- Then, for each production volume vector q we computed the minimum number of wafers $N(q)$ required to satisfy it using the integer linear programming SSWDP algorithm in Section 3.1
- Finally, for every number of wafers N, we estimate the probability of satisfying an arbitrary customer order as the ratio between $|\{q \mid N(q) \leq N\}|$ and Q.

The tradeoff curves show that cloning based floorplanning yields the highest success probability *over the entire range of number of wafers*. Besides showing the intrinsic qualities of a selected multi-project floorplan, estimates of success probability in Figure 5 could be useful, e.g., in determining how many wafers to manufacture in order to maximize expected profit.

4.2 On-Demand Wafer Dicing

We have implemented in the C++ language the greedy and history-based algorithms for on-demand side-to-side wafer dicing as described in Section 3.2. In the

Testcase	CMP	HQ	SA-unif	SA-norm	HQ*	SA-clone1	SA-clone2
Uniform distribution							
ind1	166.9/48.0	138.7/38.7	138.7/38.7	154.2/42.2	128.6/31.8	115.7/22.2	114.0/23.6
ind2	116.7/19.8	107.3/17.8	103.1/13.1	132.9/27.6	101.7/16.2	93.9/15.7	85.9/13.6
ind3	352.2/52.0	348.6/50.5	347.2/48.2	417.7/67.4	257.6/35.0	247.9/46.3	210.0/38.4
ind4	101.9/22.1	103.7/20.5	100.2/20.7	108.3/19.9	89.7/19.6	68.6/11.1	60.8/6.6
ind5	107.3/15.8	104.8/16.0	99.6/15.7	107.3/21.3	84.3/12.4	88.9/16.1	75.3/13.1
ind6	80.9/9.6	85.6/11.6	77.3/10.2	78.9/9.9	72.8/8.4	74.3/9.4	74.0/8.2
Improvement	0	4.02%	6.46%	-7.93%	20.65%	25.55%	33.04%
Normal distribution							
ind1	137.3/22.7	144.2/24.0	144.2/24.0	131.5/20.9	127.0/22.3	116.5/16.4	108.4/15.2
ind2	149.7/19.6	135.3/15.7	161.1/24.8	127.3/16.1	114.9/14.9	115.3/14.9	113.7/14.7
ind3	375.0/36.6	355.6/27.3	361.8/26.7	352.8/31.3	270.3/19.2	281.1/38.1	272.4/21.5
ind4	120.2/28.9	97.0/10.4	104.6/11.5	95.3/10.4	88.2/9.9	80.9/19.7	81.0/26.2
ind5	119.5/15.9	96.0/10.4	93.2/10.0	81.0/8.4	75.7/8.3	74.8/8.5	73.6/8.3
ind6	105.3/7.9	78.4/5.6	89.2/7.4	76.1/5.8	74.1/4.7	76.8/6.4	73.2/5.6
Improvement	0	9.98%	5.25%	14.20%	25.50%	25.98%	28.27%

Table 2: Average and standard deviation of the number of wafers assuming fixed whole wafer dicing.

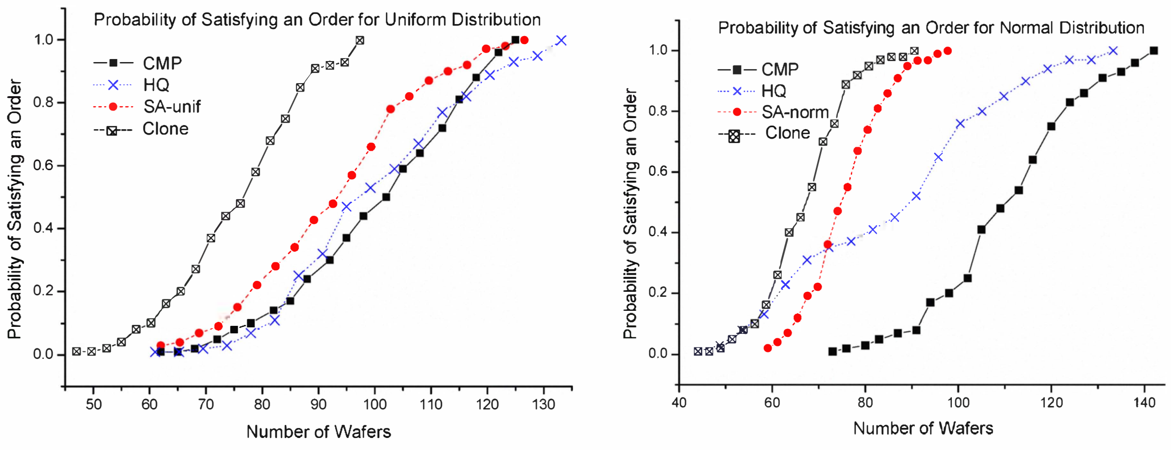

Figure 5: Tradeoff curves between the probability of satisfying an order and the number of wafers for CMP testcase “Ind5” with production volumes generated from the (a) uniform and (b) normal distributions.

basic ODSSWDP formulation we assumed that side-to-side cutting is done on whole wafers. As reported in [6], when production volumes are known, significant dicing yield improvement can be achieved by first partitioning each wafer into four equal quadrants, then dicing each quadrant independently using side-to-side cuts ([6], see Figure 6). To explore the advantages of quadrant-based methods in an on-line dicing environment we have also implemented quadrant-based versions of the two algorithms in Section 3.2.

To compare the quality of the two on-demand dicing algorithms, we generated for each die 100 individual customer orders with quantities coming from either a uniform or a normal distribution, and then randomly permuted their arrival order. Tables 3 and 4 give (in the columns for batch size of 1) the number of wafers required by the two algorithms for each of the 6 industry testcases when run on individual customer orders. To gauge the benefits of batching customer orders we also include in the two tables results obtained by running the algorithms on “batched orders” that combine groups of 10 or 100 consecutive customer orders. Finally, to estimate the overhead in the number of wafers due to demand uncertainty during dicing, we ran the two algorithms on a single batch combining all orders (both algorithms reduce to running the integer programming SSWDP algorithm in Section 3.1 on the production volume totals in this case, and therefore both result in the same number of wafers).

The results show that, compared to the simple greedy algorithm, the history-based algorithm reduces wafer overhead by an average of 19.2% (respectively 8.9%) for the uniform (normal) distributions. This consistent improvement suggests that the proposed history tracking scheme is effective in “learning” the demand distribution. As expected, regardless of the algorithm used, batching leads to significant reduction in wafer overhead, i.e., the required number of wafers required by on-demand dicing gets closer to the number of wafers required when knowing all orders in advance. Some-how surprisingly, the results show that the more complex quadrant-based dicing does not help on-line dicing unless using large batch sizes.

5 Conclusions and Future Work

In this paper we have explored the use of multiple project wafers for production under demand uncertainty. We have proposed novel algorithms and methodologies for robust multi-project reticle floorplanning and on-demand wafer dicing, and have shown that our algorithms come close in solution quality to algorithms relying on a priori knowledge of production volumes. In ongoing work we investigate the use of multi-layer reticles for further reductions in manufacturing costs.

References

[1] `http://cmp.imag.fr`

[2] M. Andersson, C. Levcopoulos and J. Gudmundsson, "Chips on Wafers, or packing rectangles into grids," *Computational Geometry* 30, pp. 95–111, 2005. Preliminary version in *Proc. WADS (Workshop on Algorithms and Data Structures)*, August 2003.

[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*, pp. 70-77, April 2004.

[6] A.B. Kahng, I.I. Măndoiu, X. Xu, and A. Zelikovsky. Yield-driven multi-project reticle design and wafer dicing. In *Proc. 25th Annual BACUS Symposium on Photomask Technology*, 2005.

[7] 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.

[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] D. Perrottet, J.-M. Buchilly, B. Richerzhagen, and W. Kröninger. "WaterJet-Guided Laser Achieves Highest Die Fracture Strength," *Future Fab International* 18, 2005, pp. 157–159.

[11] G. Xu, R. Tian, D.F. Wong, and A. Reich, "Shuttle Mask Floorplanning," *Proc. SPIE*, Vol 5256, pp. 185-194.

[12] 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.

[13] 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.

[14] M.-C. Wu and R.-B. Lin, "A Comparative Study on Dicing of Multiple Project Wafers", *Proc. IEEE Symp. on VLSI*, pp. 314–315, 2005.

[15] M.-C. Wu and R.-B. Lin, "Reticle Floorplanning and Wafer Dicing for Multiple Project Wafers", *Proc. Intl. Symposium on Quality Electronic Design*, pp. 610–615, 2005.

#Parts	Whole Wafer				4 Quadrants			
Batch size	1	10	100	all	1	10	100	all
Test	Greedy							
ind1	64	62	54	54	64	59	48	47
ind2	356	289	268	253	356	272	250	239
ind3	228	186	153	144	211	178	146	135
ind4	72	68	57	51	69	62	50	45
ind5	185	160	156	148	185	152	148	141
ind6	86	84	76	76	86	83	70	67
Overhead	36.5%	16.9%	5.2%	0	44.1%	19.6%	5.6%	0
Test	History-based							
ind1	59	57	55	54	58	53	50	47
ind2	301	272	282	253	294	264	267	239
ind3	182	161	157	144	177	161	150	135
ind4	63	58	53	51	61	57	52	45
ind5	167	156	163	148	159	156	162	141
ind6	80	77	79	76	81	75	73	67
Overhead	17.3%	7.6%	8.6%	0	23.1%	13.6%	11.8%	0

Table 3: On-demand wafer dicing results for six industry testcases with customer orders generated from a uniform distribution.

#Parts	Whole Wafer				4 Quadrants			
Batch size	1	10	100	all	1	10	100	all
Test	Greedy							
ind1	215	183	177	177	215	176	167	167
ind2	123	108	108	104	123	107	104	97
ind3	375	354	338	322	372	361	325	306
ind4	96	95	93	87	92	90	88	83
ind5	118	107	90	83	118	101	88	80
ind6	99	89	81	81	101	98	77	74
Overhead	20.1%	9.6%	3.9%	0	26.5%	15.6%	5.2%	0
Test	History-based							
ind1	193	177	185	177	187	174	175	167
ind2	110	105	108	104	106	102	102	97
ind3	354	344	339	322	346	343	326	306
ind4	94	95	87	87	90	91	87	83
ind5	107	103	85	83	99	96	86	80
ind6	92	86	82	81	93	86	81	74
Overhead	11.2%	6.5%	3.7%	0	14.1%	10.5%	6.2%	0

Table 4: On-demand wafer dicing results for six industry testcases with customer orders generated from a normal distribution.