Shuttle Mask Floorplanning with Modified Alpha-Restricted Grid

Royce L.S. Ching
Department of CSE
The Chinese University of Hong Kong
lsching@cse.cuhk.edu.hk

Evangeline F.Y. Young
Department of CSE
The Chinese University of Hong Kong
fyyoung@cse.cuhk.edu.hk

ABSTRACT

Multi Project Wafer (MPW) is an efficient method to share the mask cost among projects of different enterprisers for prototyping and low-volume manufacturing of IC designs. Designs from multiple customers can be put on one single mask substrate to produce MPW. Unlike traditional floorplanning, we need to consider the side-to-side wafer dicing constraint of the diamond sawing technology and different technology processes used in different projects for this problem. In our work, we use a branch and bound algorithm to solve this problem with a grid packing approach. We defined a special type of grid, called the modified α-restricted grid, to reduce the size of the searching solution space. Unlike many previous works, we consider non-zero margin width (but copies of the same design will have the same margin width), different technology processes of the projects, multiple copies of the same design on a mask, etc. In each searching step, our algorithm generates a grid and try to pack the dies into the grid by a Two Phase Packing(TPP) heuristic. We consider circular wafers and the objective is to minimize the total production cost. The experimental results are very promising and our approach can out-perform the most up-to-date works on this problem.

Categories and Subject Descriptors

B.7.2 [**Integrated Circuites**]: Design Aids—*Placement and routing*; J.6 [**Computer Applications**]: Computer-aided design—*Computer-aided design (CAD)*

General Terms

Algorithm, Design

Keywords

Multi-Project Wafers, Reticle Design

1. INTRODUCTION

Sub-wavelength lithography is one of the most important challenges brought by the deep sub-micron technology. In order to print those ever smaller features on a wafer as close to the original design as possible, advanced Reticle Enhancement Technologies (RET) are needed. The employment of RET and the huge rise in the number of features have increased dramatically the cost of mask manufacturing. The elevated mask costs will hinder effective prototyping and low-volume manufacturing.

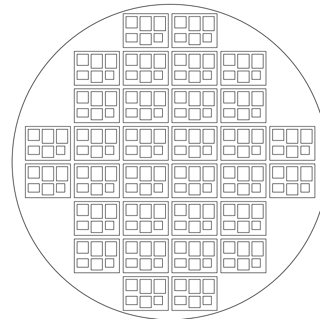

Figure 1: A Multi-Project Wafer

Various IC designs of same technology process can be fabricated on one Multiple Project Wafer (MPW) (Figure 1) to share the cost. The problem can be divided into two parts: reticle floorplanning and wafer dicing. The reticle will be repeatedly projected onto a wafer according to the floorplan and the die copies will then be diced out. As the cutting saw will only traverse the wafer horizontally and vertically from one side to another, some chips will be destroyed during the cutting process if the chips are not aligned properly. The problem of how to pack and cut the dies so that the total production cost is minimized is an interesting problem.

The participating projects may require different technology processes, e.g., 1P6M (1 poly and 6 metals) and 1P4M, etc. The cost can be further reduced if dies of different technology processes are allowed to be put on the same mask. However, dies can only be fabricated from wafers implemented with the required technology process. Other dies using different technology processes will be malfunctioned. For example, on a 1P4M wafer, only the dies requiring 1P4M can be extracted from the wafer, others such as 1P5M and 1P6M dies will be malfunctioned and discarded. In order to minimize the total production cost, we need to consider this technology process information.

During the sawing process, the margin between the cut line and the chip boundary should be limited and copies of the same design should have the same margin for effective packaging. We assume that this limit is given by the user as a ratio (r) of the width or height of the die. For example, for a die of width (height) d, it can have a margin in the range of $[0, d \times r]$ on the left (top) and right (bottom) side when it is diced out from the wafer.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
GLSVLSI'06, April 30–May 2, 2006, Philadelphia, PA, USA.
Copyright 2006 ACM 1-59593-347-6/06/0004 ...$5.00.

Different approaches have been proposed to solve the problem of reticle floorplanning and wafer dicing. However, some of the previous works [1, 2] did not consider the side-to-side dicing technology. Some of them [1, 2, 3, 4, 5] did not aim at maximizing the wafer utilization but aim at minimizing reticle area. Kahng [6] revisited the MPW problem and tried to minimize the number of wafers used. The paper [7] considered dies of different technology processes. However, dicing margin was not allowed in both formulations of [6] and [7]. Recently, flexible chips (chips of varying dimensions) for MPW had also been considered [8].

A better solution can be obtained if dicing margin is allowed. Only one previous work by Kahng *et al.* [3] allowed dicing margin. However, they did not ensure that different copies of the same project obtained from different reticle projections will have the same margin width.

The organization of this paper is as follows. Details of our proposed algorithm will be discussed in Section 2. Section 3 will present the experiment results. The last section will be the conclusion.

2. OUR APPROACH

2.1 Problem Description

In our approach, we will try to produce one single mask only due to the high mask cost. Therefore, the objective is to minimize the total number of wafers needed to satisfy all project demands. We define the Shuttle Mask Floorplanning Problem (SMF) as follows:

Shuttle Mask Floorplanning Problem (SMF) *Given the radius of a wafer, the maximum reticle dimensions and a set of n projects, where each project i is specified by the width w_i and height h_i of the design, the type of technology process t_i used, the maximum margin width ratio r_i and the production demand D_i, we want to design a floorplan for the shuttle mask and the corresponding dicing plan such that the total number of wafers needed to satisfy all the production demands are minimized.*

In this paper, we will use a grid packing approach. A grid packing is a packing in which all the rooms (or cells) are arranged in rows and columns. In the following sections, we will denote a grid G by a triple (H, W, C), where H and W are two non-descending sequences of the row heights and column widths. We denote the set of cell dimensions in G by $P(G) = \bigcup_{a_i \in H}\{a_i\} \cup \bigcup_{b_i \in W}\{b_i\}$. Let $c_{p_i \times p_j}$ be the set of cells in G of which the dimensions are $p_i \times p_j$ where $p_i, p_j \in P(G)$. C denotes the set of all rectangular cells in G, i.e., $C = \bigcup_{p_i, p_j \in P(G)} c_{p_i \times p_j}$. An example is illustrated in Figure 2.

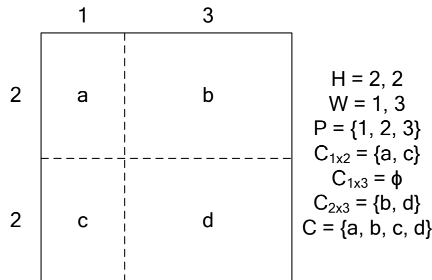

Figure 2: An Example of a Grid G(H, W, C)

Taking into consideration of the maximum margin width constraint, we aim at finding a floorplan such that all the chips of the same technology process can be diced out from a wafer without destruction, and the margins of different die copies of the same project will be the same. In our approach, we will generate a grid at each step and try to pack the dies into the grid.

2.2 An Overview

Given the production volumes of the participating projects, we can have more than one copy for each design on the mask in order to minimize the total production cost. We will follow the procedure below to decide on the number of copies for each project on the mask. At the beginning, each design has only one copy. We will do the packing and compute the production cost. Then, we will repeatedly add one copy for the most demanding project until all solutions found have exceeded the maximum reticle size.

Algorithm 1 Main Loop

1: **Procedure** *Main*
2: **for all** i **do**
3: $n_i \leftarrow 1$
4: **end for**
5: **repeat**
6: Find packing solution for the current numbers of die copies
7: $i \leftarrow$ index k that maximizes D_k/n_k
8: $n_k \leftarrow n_k + 1$
9: **until** no solution found
10: Output $\leftarrow$ best solution found in the repeat loop
11: **End procedure**

Let n_i be the number of copies for project i on the reticle. The number of reticle projections needed to meet all the demands will be at least $max_{1 \leq i \leq n}(\lceil \frac{D_i}{n_i} \rceil)$. If we want to reduce the number of wafers used, the replication number of the project with the maximum $\frac{D_i}{n_i}$ value should be increased. The project with a smaller die area will be chosen for tie breaking to favor masks with smaller areas. Now we can focus on step 6 which is, given a set of dies (some are replicated copies) to be packed on a mask, we want to design a shuttle mask floorplan such that the total number of wafers needed is minimized.

2.3 Modified α-Restricted Grid

In our approach, we will search for a grid to pack the dies by a branch and bound algorithm. For each candidate grid found, we can apply a heuristic to test if all the dies can be fit into the grid without violating the given constraints (feasibility check). The solution will then be the best grid (utilizing the least number of wafers) found during the search.

In order to reduce the search space, we have made use of a modified version of a special type of grid called the α-restricted grid proposed by Andersson[1]. Besides, having a finite solution space, this modified α-restricted grid structure is also good for controlling the margin widths of the diced chips. Copies of the same design should have the same margin width lying within a range given by the users. The original definition of an α-restricted grid given in [1] is as follows:

DEFINITION 1. *Let α be a real constant greater than one. An α-restricted grid $G(\alpha)$ is a grid where the width and height of each cell in the grid is an integral power of α, i.e.,*

the widths and heights of the rows and columns of the grid is α^i for some integer $i \geq 0$.

The area of an optimal α-restricted grid is proven to be at most α times the optimal grid size [1]. A modified α-restricted grid is obtained from an α-restricted grid and a given set of dies to be fit into the grid as follows:

DEFINITION 2. *Let D be the set of dimensions (heights and widths) of the dies to be fit into a grid and let α be a real constant greater than one. A modified α-restricted grid $G'(D, \alpha)$ is a grid in which the width and height of each cell in the grid is $\max_{d_j \in D \cap d_j \leq \alpha^i} d_j$ for some integer $i \geq 0$. We use $P'(D, \alpha)$ to denote the finite set of all possible cell dimensions (column widths and row heights) in a modified alpha-restricted grid $G'(D, \alpha)$.*

According to this definition, the unnecessary margin between the gridlines and the largest chip in a row or column will be removed. As a result, the total area of the grid is reduced without affecting the die placement. Notice that the set of all cell dimensions $P(G')$ of $G' = G'(D, \alpha)$ has at most $|D|$ values. We have the following lemma about the optimality of a modified α-restricted grid solution:

LEMMA 1. *Consider a set of die dimensions D. When $\alpha \leq d_i/d_j, \forall d_i > d_j$ and $d_i, d_j \in D$, an optimal solution based on a modified α-restricted grid $G'(D, \alpha)$ will be an optimal grid solution for the given set of dies when there is only one die in one cell.*

Proof: The optimal grid solution for a given set of dies D can be obtained by searching through all the possible combinations of different numbers of rows and columns where the row heights and column widths are the die dimensions. However, when $\alpha \leq d_i/d_j, \forall d_i > d_j$ and $d_i, d_j \in D$, the set of all possible cell dimensions in a modified α-restricted grid $G' = G'(D, \alpha)$ will be equal to the set of all possible die dimensions D.

This is because when $\alpha \leq d_i/d_j, \forall d_i > d_j$ and $d_i, d_j \in D$, consider any two die dimensions d_1 and d_2 where $d_1 < d_2$.
$\alpha^k \leq d_1 < \alpha^{k+1}$ for some k
$\alpha^{k+1} \leq \alpha d_1 \leq d_2$
$\therefore \alpha^k \leq d_1 < \alpha^{k+1} \leq d_2$
As a result, $P(G') = D$. □

2.4 Branch and Bound Algorithm

We use a branch and bound algorithm to find an optimal modified α-restricted grid to accommodate all the dies. In this section, the algorithm will be explained in details.

In the branch and bound algorithm, a modified α-restricted grid $G'(D, \alpha) = (H, W, C)$ is represented by its H and W. In the following pseudo code, we use $P'(D, \alpha) = \{p'_1, p'_2 \dots p'_m\}$ to denote the set of all possible cell dimensions in a modified α-restricted $G'(D, \alpha)$ (assuming $p'_i < p'_{i+1}$ for $i = 1 \dots m - 1$). During the search, the number of rows (columns) with height (width) p'_i are assigned sequentially from $i = m$ to 1. These numbers will be assigned from the largest to the smallest row (column) to explore all possibilities. This searching sequence will increase the grid area at a faster rate at the beginning such that pruning can be done earlier. The pseudo code of the main function of the branch and bound algorithm is shown in Algorithm 2. (Initially, $i = n - 1$, H and W of G' are empty and P' is $P'(D, \alpha)$)

Algorithm 2 The main function

```
 1: Procedure Grid_Search(i, G', P')
 2: if G' is redundant then
 3:    return
 4: end if
 5: if the dies cannot be all packed into G'_full then
 6:    return
 7: end if
 8: if i = −1 /* all row and column nos. are assigned */
    then
 9:    best_grid ← grid
10:    return
11: end if
12: for c_r ← 0 to n do
13:    for c_w ← 0 to n do
14:       Expand G' with c_r rows and c_w columns of height
          and width p'_i respectively
15:       if evaluate(G') < best then
16:          return
17:       end if
18:       Grid_Search(i − 1, G', P')
19:    end for
20: end for
21: End procedure
```

2.4.1 Redundancy Removal (Step 2)

There will be redundant grids generated during the search. Two grids $G_1 = \{H_1, W_1, C_1\}$ and $G_2 = \{H_2, W_2, C_2\}$ are identical if $H_1 = H_2$ and $W_1 = W_2$, or $H_1 = W_2$ and $W_1 = H_2$. We only need to explore one of them during the search. In order to remove redundant grids, we will only consider those grids $G' = (H, W, C)$ such that $H \leq W$ lexicographically. This is correct as swapping between H and W does not change the grid.

2.4.2 Conditions of Pruning (Step 5 & 15)

There are some other conditions for pruning during the search other than redundancy removal. They are the cases in which it is impossible for the current grid to be a part of the optimal solution, so that the current partial solution can be pruned away. There are two cases:

- If some of the dies cannot be packed into the grid even after expanding the grid to the fullest, the intermediate solution can be pruned. We can expand the current grid to the fullest G'_{full} by adding a maximum number (n) of rows and columns to the unassigned row heights and column widths. If the dies cannot be all packed in this G'_{full}, there is no feasible solutions in the sub-tree of the current search tree. The current partial solution can be pruned immediately. (step 5)
- If the number of reticle projections on a wafer of the current partial grid solution is smaller than that of the current best solution, this partial solution can be pruned as the number of the projections will only decrease further as the reticle area increases. (step 15)

2.4.3 Grid Evaluation

Our objective is to minimize the total number of wafers used to satisfy all the production demands. To compute the number of wafers used, we will place the reticles closely along the diameters of the wafer to compute the number of complete reticles on a wafer. Figure 3 shows the placement of reticles on a wafer.

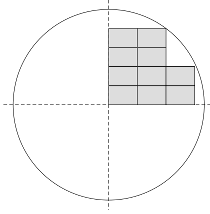

Figure 3: Reticles are Projected on the Wafer

2.5 Feasibility Check

Feasibility check is the step to check whether all the dies can be fit into a grid (step 5 in procedure *Grid_Search()*). When the number of copies from each project are all one, feasibility check can be done effectively by formulating the problem as a network flow problem [1]. However, when the number of copies from the projects are more than one, the problem is NP-complete as we need to make sure that different copies of the same project will be assigned to cells of the same size in order to have the same margin widths.

To test if all the dies can be fit into a grid, an exhaustive search or a heuristic method can be applied to do the checking. Exhaustive search is the simplest approach to solve the problem. However, expensive computational power is needed and it is only suitable for problems with only a few number of projects using one technology process. Exhaustive search is not practical and a heuristic is needed.

2.5.1 A Heuristic for Feasibility Check

We proposed a Two Phase Packing(TPP) heuristic to pack the dies into a grid. The projects are considered one by one in a non-ascending order of their areas. They are considered in this order as larger dies will have less flexibility to be fit into the grid and should be considered earlier. In the first phase, we will try to fit only one die to a cell. We will test if there are enough cells of a particular size (that does not violate the maximum margin width constraint when cutting along the grid lines) to fit in all copies of a particular project. In this phase, the technology processes of different projects are not considered because each cell holds at most one die and there is no conflict between the dies. To place the dies of project i into a set of cells of size $p_j \times p_k$ in this phase, the following conditions have to be satisfied:

C1 There are enough empty cells (at least n_i) of size $p_j \times p_k$ to hold all die copies of project i.

C2 The maximum margin width constraint is not violated, i.e., $h_i \leq p_j \leq (1+r_i) \times h_i$ and $w_i \leq p_k \leq (1+r_i) \times w_i$.

When there are more than one set of cells found, ties are broken by picking the set of cells with the smallest total area. In the second phase, constraints are relaxed such that multiple dies (of the same or different projects) can be fit into a cell. In order to maintain 100% yield, i.e., all dies using the same technology process can be cut out simultaneously, a die of project i will be fit into a cell x only if the other dies in the same row or column of x (we call the column and row of cell x the *conflict area* of cell x) do not use the same technology as project i. If there are dies using same technology process in the *conflict area*, they are potentially in conflict with each other after placing project i. For example, in Figure 4, the shaded area is the conflict area of the labeled empty cells, and the dies of project i can be placed in the labeled empty cells only if all the dies in the shaded area are not using technology t_i.

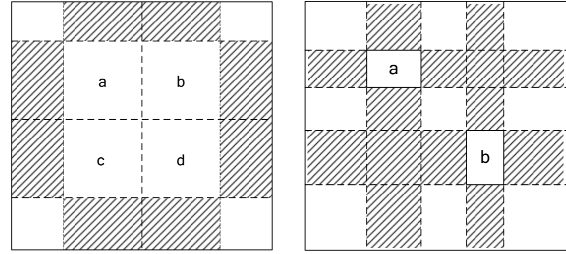

Figure 4: Conflict Area(Shaded) of the Labeled Cells

To place the dies of project i into a set of *same type* cells (cells of the same size $p_j \times p_k$ and holding the same project, or are both empty, after phase one) in this phase, the following conditions have to be satisfied:

C3 There are enough spaces in the set of cells to hold all die copies of project i.

C4 No dies in the conflict area of the set of cells are using technology process t_i.

C5 The dies are placed in such a way that they are not conflict with each other.

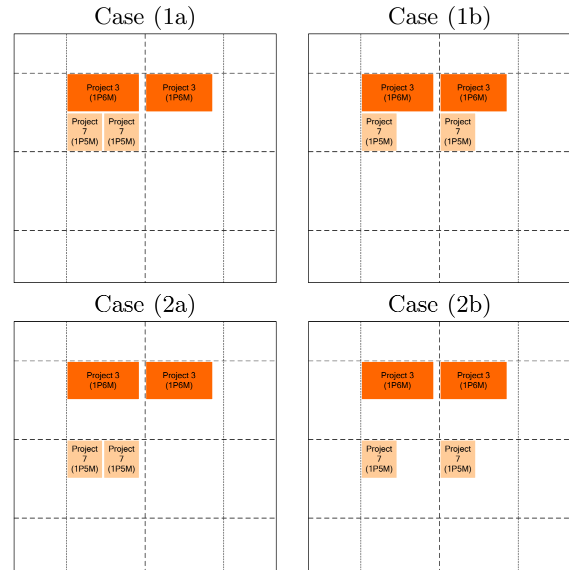

Figure 5: Some examples of Fitting Dies in Phase 2

Some examples of fitting dies into the grid in phase two are shown in Figure 5. In this example, we assume that the two dies of project 3 are already placed in the cells during phase one. Now we want to fit into the two dies of project 7. Assuming that no dies in the conflict area of the four middle cells are using 1P5M, so the two dies of project 7 can be fit into these four cells. Notice that we can place multiple die copies into a cell and also share a cell among different projects. We will exhaust all cases. The upper two cells are treated as same set of cells as they both contain a die of project 3 and the lower two are the other set. In this example, case (1a) is chosen finally as there is least white space left in the cells occupied by the dies of project 7.

```
1: Procedure TPP(N, G'(H, W, C))
2: Sort the projects in N in a non-ascending order of their areas
3: for all i ∈ N do /*first phase*/
4:   for all c_{p_j×p_k} ∈ C do
5:     if Condition C1 and C2 are satisfied then
6:       Remember c_{p_j×p_k} as a possible set of cells
7:     end if
8:   end for
9:   Put the dies of project i into the best found set of cells
     with the minimum total area
10: end for
11: Sort the projects which are not placed yet in a non-
    ascending order of their areas and store in project
12: for all i ∈ project do /*second phase*/
13:   for all c_{p_j×p_k} ∈ C do
14:     if Condition C3, C4 and C5 are satisfied then
15:       Remember c_{p_j×p_k} as a possible set of cells
16:     end if
17:   end for
18:   Put the dies of project i into the best found set of
      cells with the minimum white space left
19: end for
20: End procedure
```

2.6 Dicing Plan

As we ensure that no dies are in conflict, taking into consideration the technology processes used, when fitting dies into the grid. Therefore, all the dies of the same technology process can be diced out from a wafer simultaneously. The dicing lines are simply the gridlines, or along the boundaries of the dies with the largest width or height in a column or row (for the cases when there are multiple dies in a cell). An example is shown in the Figure 6.

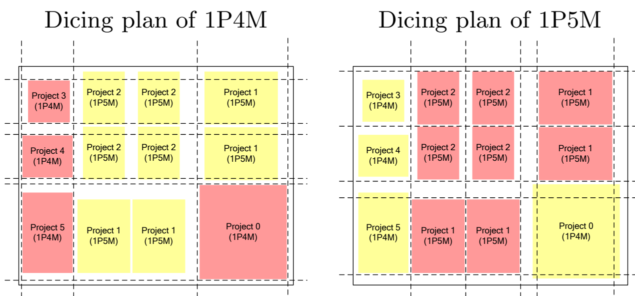

Figure 6: An Example of Dicing Plan. The darker dies are diced out.

3. EXPERIMENTAL RESULT

Our program was implemented in C++. All the experiments were conducted on a Nix dual Intel Xeon 2.4GHz Linux Workstation with 2GB RAM. Notice that $\frac{\alpha-1}{2}$ has to be smaller than $min\{r_i|1 \leq i \leq n\}$. Otherwise, some dies may not be able to fit into any grid cell due to the maximum margin width constraint. In the comparisons with the results of the other papers, we use the minimum possible α such that the solution quality of TPP is better.

We have compared our results with one of the most up-to-date published work [5] by Kahng and Reda which also uses grid packing. We have implemented the algorithm according to the pseudo code in their paper. We compared our results with five randomly generated test cases. The characteristics of the test cases are listed in Table 1. We assume that all projects use the same technology process in this experiment as the work in [5] did not consider different technology processes. Meanwhile, we have modified our algorithm such that the objective is to minimize the reticle area instead of minimizing the number of wafers since this is the original objective in [5]. The results are shown in Table 2. The run time of both implementations are also listed for reference.

Table 1: Five Test Cases for Comparison with [5]

	No. of Projects	Total No. of Dies
Case 1	3	10
Case 2	6	10
Case 3	5	9
Case 4	8	15
Case 5	9	17

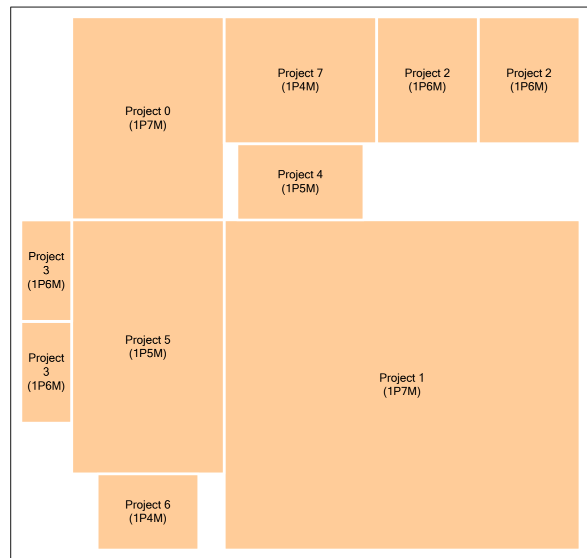

Figure 7: A Solution Obtained from Our Algorithm ($r = 0$, dead space $= 12.77\%$)

Our algorithm with no margin ($r = 0$) performs exactly the same as the algorithm in [5] with 100% yield in all cases. This is because we both are finding the optimal grid with zero margin. When margin is allowed, we can produce smaller reticle of 100% yield in all cases even when comparing with their result with only 50% yield. We found that their algorithm does not perform well when the dimensions of the input die copies are very distinct. In their algorithm with 100% yield, only dies with the same height (width) can be placed in the same row (column). In those cases, the results will be better if margin is allowed. Therefore, allowing some margin can result in a much better solution.

We have also evaluated our algorithm with the benchmarks provided by the authors of [7]. The characteristics of the data sets are shown in Table 3 and the results are listed in Table 4. The run time of their approach is quoted from their paper [7]. Their experiments were run on an AMD Opteron processor with 4GB RAM. When $r = 0$, both of us consider zero margin and different technology processes are taken into account. We can see from the table that we outperform [7] in both run time and solution quality in most cases. However, our approach can allow even more flexibility by exploring non-zero margin and the result can be further improved by taking larger margin ($r = 0.1, 0.2$ and 0.25).

Table 2: Result in Comparison with the Approach in [5]

Test Case	Die Area	TPP (100% Yield)($r=0$) Reticle Area	Dead Space	Time (s)	TPP (100% Yield)($r=0.25$) Reticle Area	Dead Space	Time (s)	[5] (100% Yield) Reticle Area	Dead Space	Time (s)	[5] (50% Yield) Reticle Area	Dead Space	Time (s)
1	80	220	63.8%	0.1	118	32.6%	0.1	220	63.8%	48	231	47.6%	173
2	283	598	52.7%	0.2	353	19.8%	0.2	598	52.7%	18.9	377	24.9%	222
3	224	680	67.1%	0.1	392	42.7%	0.1	680	67.1%	0.2	430	48.0%	92
4	180	528	65.9%	0.3	331	45.7%	1.8	528	65.9%	0.7	341	47.3%	269
5	359	976	63.2%	1.4	480	25.2%	2.3	976	63.2%	8	592	39.4%	472

Table 3: Characteristics of the Benchmarks from [7]

Benchmark	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
No. of Die Types	10	10	14	15	15	16	18	18	20	20
No. of Technology processes	4	5	4	4	4	4	4	3	4	5
Max. Reticle Dimensions	15x15	20x20	20x20	15x15	20x20	20x20	20x20	15x15	20x20	15x15

Table 4: Result in Comparison with [7]

	M1		M2		M3		M4		M5	
	#Wafer	Time (s)	#Wafer	Time (s)	#Wafer	Time (s)	#Wafer	Time (s)	#Wafer	Time (s)
TPP($r=0.25$)	4	0.1	5	0.1	8	2.7	7*	0.9	8	2.7
TPP($r=0.2$)	5	0.2	5	0.3	8	3.6	7*	1.2	8	3.6
TPP($r=0.1$)	5	0.2	5	0.3	8	3.3	7*	1.2	8	8.6
TPP($r=0$)	5	0.2	5	0.3	8	3.2	7*	1.5	8	7.8
ILP of [7]	7	393	8	65	10	2	9	587	11	6

	M6		M7		M8		M9		M10	
	#Wafer	Time (s)	#Wafer	Time (s)	#Wafer	Time (s)	#Wafer	Time (s)	#Wafer	Time (s)
TPP($r=0.25$)	7	25.3	5	81.3	3	21.5	6	1012	7	114
TPP($r=0.2$)	7	36.3	5	86.1	5	90.2	7	1486	7	132
TPP($r=0.1$)	7	20.6	5	102	6	70.9	8	844	7	96
TPP($r=0$)	7	11.1	6	70.8	6	3	9	91.8	8	6.5
ILP of [7]	19	14	10	395	13	2814	15	487	16	723

*Reticle size is 16×16 instead of 15×15.

Table 5: Tradeoff between Runtime and Quality When $r = 0.25$

	M6		M7		M8		M9		M10	
	#Wafer	Time (s)	#Wafer	Time (s)	#Wafer	Time (s)	#Wafer	Time (s)	#Wafer	Time (s)
$\alpha = 1.4$	10	0.3	8	5.3	6	0.5	10	4.5	8	0.5
$\alpha = 1.2$	7	1.6	7	8.0	6	0.6	10	7.9	7	0.8
minimum α	7	25.3	5	81.3	3	21.5	6	1012	7	114

Due to our 100% yield guarantee, the number of wafers needed can be greatly reduced. However, we sometimes (e.g., for data set M4) cannot find a solution with the given maximum reticle size. But we can find a good solution (using less number of wafers) with just a slightly bigger size. For test case M4, the size of our wafer is 16×16 instead of 15×15.

In addition, we have run some experiments on some test-cases using $r = 0.25$ to demonstrate the tradeoff between solution quality and run time by adjusting the value of α. The result is shown in Table 5. When α is increased, the search space is reduced exponentially. Therefore, the run time can be improved a lot, but with a trade off in the solution quality. In order to obtain the best solution, the smallest possible α (as suggested by Lemma 1) should be used.

4. CONCLUSIONS

In this paper, we presented an algorithm to solve the multi-technology-process reticle floorplanning problem by grid packing approach with non-zero margin allowed. Our algorithm guarantees that no dies of the same technology process will be destroyed in the sawing process. Also, we defined a modified α-restricted grid to reduce the size of the search space. It also allows us to adjust the tradeoff between solution quality and run time. With non-zero margin, flexibility is increased in die placement and the solution quality can be improved further. Our algorithm allows us the flexibility to use zero or non-zero margin. Experimental results have shown its effectiveness in comparison with previous works in both zero margin or non-zero margin.

5. REFERENCES

[1] M. Andersson, J. Gudmundsson, and C. Levcopoulos. Chips on wafers. In *Proceedings of Workshop on Algorithms and Data Structures*, 2003.

[2] G. Xu, R. Tian, D.F. Wong, and A. Reich. Shuttle mask floorplanning. In *Proceedings of SPIE*, volume 5256, pages 185–194, 2003.

[3] A.B. Kahng, I. Mandoiu, Q. Wang, X. Xu, and A. Zelikovsky. Multi-project reticle floorplanning and wafer dicing. In *Proceedings of ISPD*, pages 70–77, 2004.

[4] G. Xu, R. Tian, Z. Pan, and D.F. Wong. A multi-objective floorplanner for shuttle mask optimization. In *Proceedings of SPIE*, volume 5567, 2004.

[5] A.B. Kahng and S. Reda. Reticle floorplanning with guaranteed yield for multi-projects wafers. In *Proceedings of ACM/IEEE on ICCD*, pages 106–110, 2004.

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

[7] C.C. Chen and W.K. Mak. A multi-technology-process reticle floorplanner and wafer dicing planner for multi-project wafers. In *ASPDAC*, 2006.

[8] Meng-Chiou Wu and Rung-Bin Lin. Reticle floorplanning of flexible chips for multi-project wafers. In *ASPDAC*, 2006.