A Multi-Technology-Process Reticle Floorplanner and Wafer Dicing Planner for Multi-Project Wafers*

Chien-Chang Chen and Wai-Kei Mak
Department of Computer Science
National Tsing Hua University
Taiwan 300 R.O.C.

Abstract—As the VLSI manufacturing technology advances into the deep sub-micron(DSM) era, the mask cost can reach one or two million dollars. Multiple project wafers (MPW) which put different dies onto the same set of masks is a good cost-sharing approach. Every design needs to be produced by its desired technology process, such as 1 poly with 4 metal layers (1P4M), or 1 poly with 5 metal layers (1P5M). Dies with different desired manufacturing processes cannot be produced from the same wafer, but they can be put onto the same set of masks in order to reduce the total cost of the used masks and wafers. In this paper, we propose a novel integer linear programming (ILP)-based floorplanner for shuttle runs consisting of projects requiring different desired processes. Two simulated annealing-based side-to-side wafer dicing planners are also presented. Experimental results show that our approach achieves 28% wafer reduction on average compared to a previous simulated annealing-based reticle floorplanner.

I. Introduction

As the VLSI manufacturing technology advances into the deep sub-micron era, mask cost increases at an accelerating rate. The mask cost is around $700k dollars for 130nm and $1 million dollars for 90nm. Reticle enhancement technologies(RET) such as optical proximity correction (OPC) and phase shifting mask (PSM) cause the complexity and the cost of mask to grow dramatically [1, 2, 3]. Multiple project wafers(MPW), or shuttle run, allows customers to share the expensive cost of a common mask tooling set for an engineering-run and obtain their samples quickly for fast prototyping and low volume designs. MPW vendors like Taiwanese Semiconductor Manufacturing Company (TSMC) and IBM provide shuttle services to their customers.

MPW involves two key problems : (1) shuttle mask (reticle) floorplanning and (2) wafer dicing planning. Unlike the traditional floorplanning problem which is to pack the blocks as closely as possible in order to minimize the total area and minimize the wirelength between the blocks, the objective of shuttle mask floorplanning is to reduce the total cost of the used masks and wafers. A better shuttle mask floorplan provides better wafer yield, in other words, more dies can be produced by less wafers. A simple reticle floorplan example is shown in figure 1(a). The fabricated wafers must be cut to obtain the bare dies. Usually the dies on a wafer are cut out by a cutting saw that traverses the whole wafer horizontally and vertically. This is called side-to-side wafer dicing. A set of horizontal (vertical) cut lines, called a row (column) dicing plan must be assigned for each row (column) of reticles on the wafer as shown in figure 2. A die can be diced out successfully only when the cut lines are along the margins of the die in the reticle floorplan and no cut lines cut across the die. Only a fraction of dies can be diced out according to a dicing plan because cutting out one die may destroy another die. For example, in figure 1(b), only two dies are diced out. Packing the different dies on a reticle and choosing the right wafer dicing plans are two interesting and challenging problems.

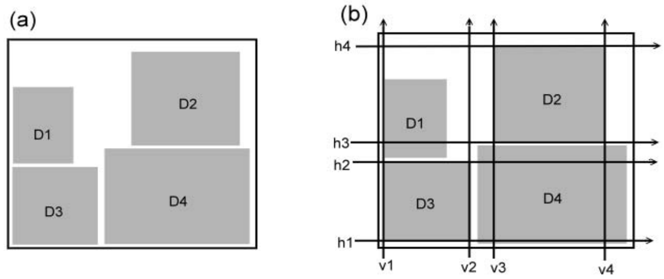

Fig. 1. (a) A reticle floorplan consisting of four dies. (b) $\{h1,\ h2,\ h3,\ h4\}$ is a row dicing plan and $\{v1,\ v2,\ v3,\ v4\}$ is a column dicing plan. Only D2 and D3 are diced out.

Recently, a number of papers considered the reticle floorplanning problem and the wafer dicing problem [4]-[10]. Many works (eg. [4, 5, 6, 7, 9]) only considered how to minimize the reticle area and/or maximize the minimum number of successfully extracted copies of the same die type in a wafer without considering the actual demand of each type. Some works are based on the simple but inaccurate assumption that a wafer is rectangular when performing optimization [7, 9]. The use of highly expensive wafer dicing equipment other than a side-to-side wafer dicing equipment was considered in [6]. A simulated annealing-based floorplanner considering a weighted sum of various objectives was proposed in [8]. Wu et al. [10]

*This work was supported in part by NSC 94-2220-E-007-014 and MOEA 94-EC-17-A-01-S1-038.

0-7803-9451-8/06/$20.00 ©2006 IEEE.

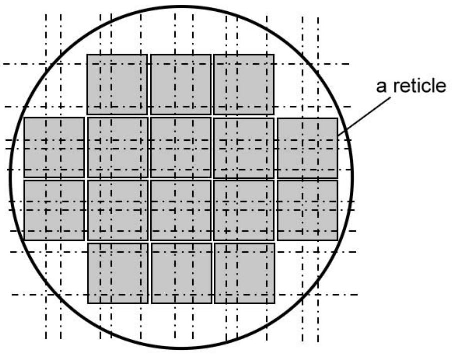

Fig. 2. A side-to-side wafer dicing plan.

considered the circular shape of the wafers and the production volume requirement of each die type when performing optimization, but their approach required a long run time.

In practice, in order to maximize the MPW utilization and reduce the total cost of the used masks and wafers, projects requiring different number of metal layers can be put on the same shuttle. If a wafer is used to fabricate the projects with 4 metal layers, the wafer is taken off before processing the 5th metal layers, so on and so forth. Non-1P4M projects that are diced out from 1P4M wafer will malfunction and must be discarded. In this paper, we propose a novel integer linear programming-based floorplanner for shuttle runs consisting of projects requiring different desired technology processes.

Our goal is to minimize the number of wafers needed to satisfy the demands of all die types. Our floorplanner incorporates die replication on a reticle to reduce the total number of wafers needed to meet the different demands of different die types. Moreover, we propose two simulated annealing-based wafer dicing planners to minimize the number of required wafers.

The rest of this paper is organized as follows. We formulate and analyze the problem in section II. A novel integer linear programming-based floorplanner is given in section III. A simulated annealing-based wafer dicing planner is presented in section IV. Experimental results are reported in chapter V.

II. Problem Analysis

The problem considered in this paper is as follows. Given (1) a set of N projects with their desired technology processes and demands, (2) the maximum dimensions of a reticle, (3) the wafer size, we want to find a reticle floorplan and a set of side-to-side wafer dicing plans in order to satisfy the demand of each die type while minimizing the required wafers.

We say that two dies on the reticle are in *vertical (horizontal) conflict* if no set of vertical (horizontal) cut lines can dice the two dies simultaneously. On the contrary, two dies are *conflict-free* if there exists a set of vertical cut lines and a set of horizontal cut lines to extract both dies simultaneously. Figure 3 illustrates these concepts. If many dies are in vertical/horizontal conflicts, the yield will be seriously affected.

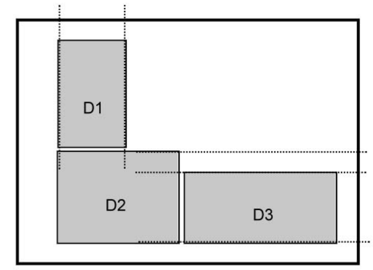

Fig. 3. D1 and D2 are in vertical conflict. D2 and D3 are in horizontal conflict. D1 and D3 are conflict-free.

In this paper, we consider the fact that the projects in a shuttle run may require different technology processes. We note that the dies of different technology processes cannot be extracted from the same wafer. Therefore, we have the following key observation.

Observation 1 : *Even if two dies with different desired processes are conflict-free in the reticle floorplan, they still cannot be produced at the same time.*

An example is shown in figure 4, where Die A must be produced by 1P4M wafers and Die B by 1P5M wafers. The two dies can be diced out at the same time from the reticle floorplan. But if a wafer is targeted to fabricate 1P4M dies, Die B extracted from this wafer will malfunction and must be discarded. On the other hand, if a wafer is targeted to fabricate 1P5M dies, Die A extracted from this wafer will malfunction and must be discarded.

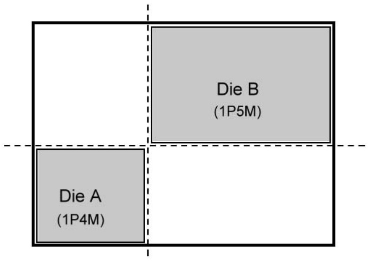

Fig. 4. Die A must be produced by technology process 1P4M, but Die B needs 1P5M.

In order to satisfy the different demands of the dies and reduce the total number of wafers needed, we have another observation as follows.

Observation 2 :*We can put multiple instances of the same die on a reticle during floorplanning depending on the demand of the die.*

III. Integer Linear Programming-Based Floorplanner

As explained in section II, if the dies are conflict-free on the reticle, they can be diced out at the same time.

However, we cannot put all the dies in conflict-free positions on the reticle due to the limitation on the size of the reticle. The above observation motivates us to place the dies of the same technology process in conflict-free positions on the reticle so that they may be extracted from the same wafer at the same time. In other words, we must minimize the horizontal conflict or vertical conflict situations for the dies of the same technology process but we are not concerned about the horizontal/vertical conflicts for the dies of different technology processes.

We assume that a grid structure with p rows and q columns is imposed on the reticle as shown in figure 5. We assume that there is at most one die allocated to each grid cell and the die is aligned with left-bottom corner in the grid cell. The width of column j in the grid structure is determined by the width of the widest die in column j. Similarly, the height of row i in the grid structure is determined by the height of the tallest die in row i.

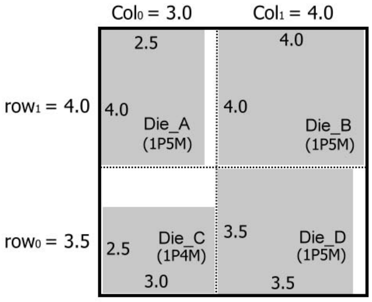

Fig. 5. A grid structure with two rows and two columns imposed on a reticle.

Variables used in our ILP formulation are:

- x_{ijk} denotes whether die k is allocated to row i and column j in the grid or not. $x_{ijk} = 1$ if die k is allocated to row i and column j, and $x_{ijk} = 0$ otherwise.
- r_k denotes whether die k is rotated or not. $r_k = 1$ if die k is rotated, and $r_k = 0$ otherwise.
- row_i denotes the height of row i in the grid. For the example in figure 5, $row_0 = 3.5$ and $row_1 = 4.0$.
- col_j denotes the width of column j in the grid. For the example in figure 5, $col_0 = 3.0$ and $col_1 = 4.0$.
- rc_i denotes the maximum number of the same technology process dies in row i. For the example in figure 5, $rc_0 = 1$ and $rc_1 = 2$.
- cc_j denotes the maximum number of same technology process dies in column j. For the example in figure 5, $cc_0 = 1$ and $cc_1 = 2$.

Constants for the ILP formulation are as follows,

- W_k, H_k, D_k denote the width, height and demand of die k, respectively, where $1 \leq k \leq N$.
- $Total\_Dmd$ denotes the total demand of the dies, i.e., $Total\_Dmd = \sum_{k=1}^{N} D_k$.
- T_m denotes the set of dies which must be produced by the same technology process 1PmM. For example in figure 5, $T_4 = \{Die\_C\}$ and $T_5 = \{Die\_A,\ Die\_B,\ Die\_D\}$
- Rw, Rh denote the given maximum width and height of the reticle.

We formulate the shuttle mask floorplan problem as an integer linear program as follows,

$$min\ \ (\sum_{i=1}^{p} rc_i + \sum_{j=1}^{q} cc_j) - \sum_{k=1}^{N} (\frac{D_k}{Total\_Dmd} \sum_{i=1}^{p} \sum_{j=1}^{q} x_{ijk})$$

$s.t.$

$$\sum_{i=1}^{p} \sum_{j=1}^{q} x_{ijk} \geq 1 \qquad \forall k \quad (1)$$

$$\sum_{k=1}^{N} x_{ijk} = 1 \qquad \forall i, j \quad (2)$$

$$H_k(x_{ijk} - r_k) + W_k(x_{ijk} + r_k - 1) \leq row_i \qquad \forall i, j, k \quad (3)$$

$$W_k(x_{ijk} - r_k) + H_k(x_{ijk} + r_k - 1) \leq col_j \qquad \forall i, j, k \quad (4)$$

$$row_i \geq 0 \qquad \forall i \quad (5)$$

$$col_j \geq 0 \qquad \forall j \quad (6)$$

$$\sum_{i=1}^{p} row_i \leq Rh \quad (7)$$

$$\sum_{j=1}^{q} col_j \leq Rw \quad (8)$$

$$\sum_{k \in T_m} \sum_{j=1}^{q} x_{ijk} \leq rc_i \qquad \forall i, m \quad (9)$$

$$\sum_{k \in T_m} \sum_{i=1}^{p} x_{ijk} \leq cc_j \qquad \forall j, m \quad (10)$$

$$x_{ijk} \in \{0, 1\} \qquad \forall i, j, k (11)$$

$$r_k \in \{0, 1\} \qquad \forall k \quad (12)$$

(1) guarantees that each die type must be allocated at least one grid cell, in other words, there is at least one instance for each die type in the reticle. Moreover, multiple instances of the same die type can be assigned to multiple grid cells. (2) ensures that there is at most one die allocated to a grid cell. The variables x_{ijk} and r_k in the LHS of (3) and (4) have four possible combinations:

$$\begin{cases} case1: & x_{ijk} = 1,\ r_k = 1; \\ case2: & x_{ijk} = 1,\ r_k = 0; \\ case3: & x_{ijk} = 0,\ r_k = 1; \\ case4: & x_{ijk} = 0,\ r_k = 0. \end{cases}$$

If x_{ijk} is equal to 1, it means that die k is allocated to row i, column j in the grid. The LHS of (3) is equal to the width of die k if $r_k = 1$, otherwise it is equal to the height of die k. However, if x_{ijk}=0, no matter r_k=0 or 1, the LHS of (3) and (4) is negative. So, when x_{ijk}=0, constraints (3) and (4) will become non-binding. Row_i and col_j must be larger than or equal to zero by (5) and (6). If $row_i = 0$ ($col_j = 0$), it means that no die is allocated to row i (column j). (7) and (8) guarantee that the sum of all row heights and the sum of all column widths are not greater than the given maximum reticle dimensions. (9) and (10) are used to calculate the maximum number of dies of the same process in each row and column.

The objective function is set up according to two key factors, the maximum number of dies of the same process in each row(column) and the demand of each die type. By minimizing the maximum number of dies of the same process in each row(column), we can maximize the number of conflict-free dies of the same process. If more dies of the same technology process are arranged in conflict-free positions, more dies can be extracted at the same time. Secondly, we can put more than one instance of a die type on the reticle according to its demand. We set the weight for die type k as $\frac{D_k}{Total\_Dmd}$. The higher the demand of die type k, the larger is its weight. The objective function is set up to encourage replication of die types with higher weights.

After solving the integer linear programming problem, we get a reticle floorplan based on a grid graph. There may be much wasted reticle area because of the differing heights(widths) of the dies arranged on the reticle as shown in figure 6(a). The bottom-left grid cell of figure 6(a) contains a small die, Die_B, and resulted in much wasted area. In order to maximize the area utility of the reticle, we do the following refinement. We duplicate a die as many times as possible within its allocated grid cell as in figure 6(b).

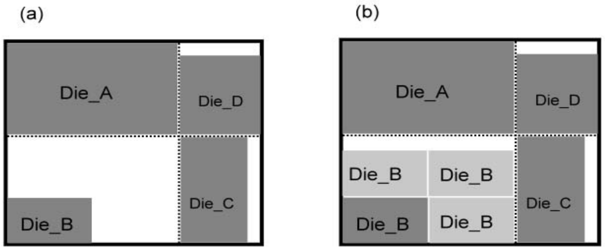

Fig. 6. An example of die replication after floorplanning.(a) The bottom-left grid cell has much wasted area. (b) If we duplicate Die_B in its allocated grid cell, the area utility of the reticle is increased.

IV. Simulated Annealing-Based Wafer Dicing

In this section, we propose two simulated annealing-based dicing planners. A set of horizontal (vertical) cut lines, called a row (column) dicing plan must be assigned for each row (column) of reticles on the wafer as shown in figure 2. We note that different technology processes must use different wafers. Instead of using a single wafer dicing plan for all wafers, we choose to optimize the dicing plans of different technology process wafers independently. For example, when we compute the wafer dicing plan for 1P4M wafer, our objective is to be able to use the minimum number of 1P4M wafers to satisfy the demands of all 1P4M projects.

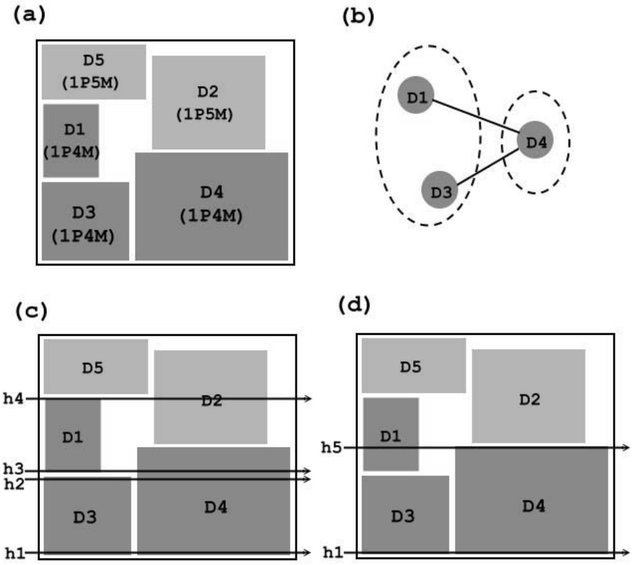

Fig. 7. Finding candidate row dicing plans. (a) A reticle floorplan. (b) A horizontal conflict graph of technology process 1P4M corresponding to the floorplan in (a). {D1, D3} is a maximal independent set and {D4} is another maximal independent set. (c) D1 and D3 are simultaneously dicable by row dicing plan {h1, h2, h3, h4}. (d) D4 is dicable by row dicing plan {h1, h5}.

We find all the row (column) dicing plans for each technology process m by computing all the maximal independent sets in a horizontal (vertical) conflict graph. Figure 7 shows an example for finding row dicing plans of 1P4M wafer.

Each die of technology process m on a reticle is represented by a node in the horizontal (vertical) conflict graph, there is an edge between two nodes if the corresponding dies are in horizontal (vertical) conflict in the reticle floorplan. An independent set in a conflict graph is a set of nodes without any edge between them. A maximal independent set is an independent set such that no node can be added to the set without breaking the independence property. The dies corresponding to the nodes in the same maximal independent set can be diced out at the same time. We construct an initial wafer dicing plan by assigning a row (column) dicing plan to each row (column) of reticles on the wafer randomly. Then we apply simulated annealing to search for a good wafer dicing plan with the following moves:

- Exchange the row (column) dicing plans of two rows (columns) where the number of printed reticle images of the rows (columns) are not equal.
- Change the row (column) dicing plan of a row (column).

We propose two slightly different wafer dicing planners as follows.

- **D1**: Use the same wafer dicing plan for the wafers of the same technology process. And we use simulated annealing to search for a single wafer dicing plan for each technology process m. For each technology process m, the objective is to search for a wafer dicing plan such that $\max_{k \in T_m} \frac{D_k}{d_k}$ is minimized (i.e., the number of technology process m wafers needed is minimized), where d_k denotes the number of die k obtained from each wafer.
- **D2**: Use different wafer dicing plan for each wafer (even for wafers of the same technology process). For example, for the first wafer of technology process m, we use simulated annealing to compute a wafer dicing plan to minimize $\sum_{k \in T_m} D'_k$ where $D'_k = D_k - d_k$ if $d_k \leq D_k$, and $D'_k = 0$ otherwise. If $\sum_{k \in T_m} D'_k = 0$, then the demands of all dies k of technology process m have been met. Otherwise, there is still some die k of technology process m whose demand has not been met and we need another wafer of technology process m. In this case, we repeat the step above after updating the demand D_k of die k to D'_k. This can be repeated until the demands of all dies k of technology process m are met.

V. Experimental Results

We compared our ILP-based floorplanner with the simulated annealing-based floorplanner in [7]. The characteristics of ten benchmarks are shown in Table I. The smaller benchmarks are provided by [11] and we combined the projects in the smaller benchmarks randomly to generate the larger benchmarks. The demands of the dies in each benchmark vary from 100 to 200. We employed lp_solve 5.1[12] to solve the integer linear programs and implemented the SA-based floorplanner of [7] in C++. All experiments were run on an AMD Opteron processor with 4GB memories. While performing ILP-based floorplanning, we set up the grid size with the smallest p such that p=q and $p * q \geq N$.

In the first experiment, we wanted to compare the performance of our floorplanner and the floorplanner in [7] if all dies could use the same technology process. So, we computed the number of wafers needed using the simulated annealing-based method in [7], and we also computed the number of wafers required using our floorplanner. The results are reported in Table II. For all benchmarks, the minimum required wafers based on our ILP-based_floorplanner is less than the floorplanner in [7]. The maximum wafer reduction is 52% and the average reduction is 30%. Moreover, the run-time of our ILP-based_floorplanner is also faster than the previous method.

In the second experiment, we assumed that different technology processes must be used for the dies in each benchmark as shown in Table I. The minimum required wafers for each benchmark is reported in Table III. Our ILP-based_floorplanner is again better than the simulated annealing-based floorplanner and resulted in 50% maximum wafer reduction and 28% reduction on average. The run-time of our floorplanner is again faster.

Finally, we tried two different mechanism of wafer dicing. We used the same floorplanner (ILP-based_floorplanner) with the two dicing planners D1 and D2. The minimum required wafers for each benchmark is given in Table IV. The results of using a different wafer dicing for each wafer is clearly better than the results of using the same wafer dicing for the same process wafers.

VI. Conclusion

In this paper, we proposed an integer linear programming (ILP)-based floorplanner for shuttle runs consisting of projects requiring different desired processes. Our floorplanner incorporate die replication on a reticle during floorplanning to reduce the total number of wafers needed to meet the different demands of different die types. Two simulated annealing-based wafer dicing planners were also presented. Experimental results show that our approach achieves 28% wafer reduction on average compared to a previous SA-based floorplanner and our method is computationally efficient.

References

[1] W. Gorbman, R. Boone, C. Phibin, and B. Jarvis, "Reticle Enhancement Technology Trends: Resource and Manufacturability Implication for the Implementation of Physical Design" in *Proc. of ACM/IEEE on ISPD*, pp. 45-51, 2001

[2] John, and B. Lin, "Mask Cost and Cycle Time Reduction" http://www.sematech.org/resources/litho/meetings/mask/20011001/E_TSMC.PDF

[3] C. Yang, "Challenges of Mask Cost & Cycle Time" http://www.sematech.org/resources/litho/meetings/mask/20011001/K_Mask_cost_Intel.pdf

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

[5] M. Andersson, C. Levcopoulos, and J. Gudmundsson, "Chips on Wafers" in *Proc. Workshop on Algorithms and Data Structures*, 2003.

[6] G. Xu, R. Tian, D. F. Wang, and A. Reich "Shuttle Mask Floorplanning" in *Proc. of SPIE*, Vol. 5256, 2003, pp.185-194.

[7] A. B. Kahng, I. Mandoiu, Q. Wang, X. Xu, and A. Zelikovsky, "Multi-Project Reticle Floorplanning and Wafer Dicing" in *Proc. of ACM/IEEE on ISPD*, 2004, pp. 70-77.

[8] G. Xu, R. Tian, D. Z. Pan, and D. F. Wang "A Multiple-objective Floorplanner for Shuttle Mask Optimization" in *Proc. of SPIE*, Vol. 5567 , 2004, pp.185-144.

[9] A. B. Kahng, and S. Reda, "Reticle Floorplanning With Guaranteed Yield for Multi-Project Wafers" in *Proc. of ACM/IEEE on ICCD*, 2004, pp. 106-110.

[10] M. C. Wu and R. B. Lin, " Reticle Floorplanning and Wafer Dicing for Multiple Project Wafers" in *Proc. of ISQED*, 2005.

[11] Global UniChip Corp. http://www.globalunichip.com/

[12] M. Berkelaar, K. Eikland, P. Notebaert, *lp_solve*, available from http://groups.yahoo.com.tw/group/lp_solve

TABLE I
THE CHARACTERISTICS OF EACH BENCHMARK.

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	5	3	5	4

TABLE II
THE EXPERIMENTAL RESULTS OF EACH BENCHMARK IF ALL DIES USED THE SAME TECHNOLOGY PROCESS. THE REQUIRED WAFERS MEANS THE NUMBER OF 200MM WAFERS REQUIRED TO SATISFY THE DEMAND OF EACH DIE IN EACH BENCHMARK.

Benchmark	**SA**-based_floorplanner+D1		**ILP**-based_floorplanner+D1		Reduction of Required wafers (%)
	Required wafers	run-time(sec)	Required wafers	run-time(sec)	
M1	9	325	7	18	22
M2	14	132	11	2	21
M3	16	122	14	11	12
M4	17	426	8	24	52
M5	26	281	14	26	46
M6	16	409	12	13	25
M7	17	646	10	217	41
M8	17	941	12	104	29
M9	17	1229	16	105	5
M10	25	1136	14	656	44

TABLE III
THE EXPERIMENTAL RESULTS OF EACH BENCHMARK FOR THE NEW MPW PROBLEM WHICH CONSIDER THE DESIRED TECHNOLOGY PROCESS OF EACH DIE.

Benchmark	**SA**-based_floorplanner+D1		**ILP**-based_floorplanner+D1		Reduction of Required wafers (%)
	Required wafers	run-time(sec)	Required wafers	run-time(sec)	
M1	11	229	8	25	27
M2	14	345	14	2	0
M3	24	359	14	24	41
M4	24	401	12	107	50
M5	28	461	14	26	50
M6	32	248	21	243	37
M7	18	853	15	717	16
M8	18	988	13	693	27
M9	18	944	17	95	5
M10	26	1348	18	703	30

TABLE IV
THE EXPERIMENTAL RESULTS OF EACH BENCHMARK WITH THE TWO WAFER DICING PLANNERS

Benchmark	ILP-based_floorplanner+**D1**		ILP-based_floorplanner+**D2**		Reduction of Required wafers (%)
	Required wafers	run-time(sec)	Required wafers	run-time(sec)	
M1	8	25	7	393	12
M2	14	2	8	65	42
M3	14	24	10	2	28
M4	12	107	9	587	25
M5	14	26	11	6	21
M6	21	243	19	14	10
M7	15	717	10	395	33
M8	13	693	13	2814	0
M9	17	95	15	487	11
M10	18	703	16	723	11