Area Reduction by Deadspace Utilization on Interconnect Optimized Floorplan

CHIU-WING SHAM
The Hong Kong Polytechnic University
and
EVANGELINE F. Y. YOUNG
The Chinese University of Hong Kong

Interconnect optimization has become the major concern in floorplanning. Many approaches would use simulated annealing (SA) with a cost function composed of a weighted sum of area, wirelength, and interconnect cost. These approaches can reduce the interconnect cost efficiently but the area penalty of the interconnect optimized floorplan is usually quite large. In this article, we propose an approach called *deadspace utilization* (DSU) to reclaim the unused area of an interconnect optimized floorplan by linear programming. Since modules are not necessarily rectangular in shape in floorplanning, some deadspace can be redistributed to the modules to increase the area occupied by each module. If the area of each module can be expanded by the same ratio, the whole floorplan can be compacted by that ratio to give a smaller floorplan. However, we will limit the compaction ratio to prevent overcongestion. Experiments show that we can apply this deadspace utilization technique to reduce the area and total wirelength of an interconnect optimized floorplan further while the routability can be maintained at the same time.

Categories and Subject Descriptors: B.7.2 [**Integrated Circuit**]: Design Aids

General Terms: Performance

Additional Key Words and Phrases: Floorplanning, area reduction

ACM Reference Format:

Sham, C.-W. and Young, E. F. Y. 2007. Area reduction of dead space utilization on interconnect optimized floorplan. ACM Trans. Des. Automat. Elect. Syst. 12, 1, Article 3 (January 2007), 11 pages. DOI = 10.1145/1188275.1188278 http://doi.acm.org/ 10.1145/1188275.1188278

The work described in this article was partially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CUHK4188/03E).
Authors' Addresses: C.-W. SHAM, DE625, Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong; email: encwsham@eie.polyu.edu.hk; E. E. Y. Young, 1/F, Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong; email: fyyoung@cse.cuhk.edu.hk.
Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.
© 2007 ACM 1084-4309/2007/01-ART3 $5.00 DOI 10.1145/1188275.1188278 http://doi.acm.org/ 10.1145/1188275.1188278

1. INTRODUCTION

1.1 Motivations

Interconnect optimization has become the major concern in floorplanning. Due to the recent advances in VLSI technology, the number of transistors in a design are increasing rapidly, and interconnect has become a dominant factor in the overall performance of a circuit. Many floorplanning algorithms [Chen et al. 1999; Wang and Sarrafzadeh 2000; Wang et al. 2000; Chang et al. 2000; Lou et al. 2001; Sham and Young 2003; Lai et al. 2003; Ma et al. 2003] use simulated annealing (SA) with a cost function composed of a weighted sum of area, wirelength, and interconnect cost. While the cost function is being minimized in the annealing process, the area, wirelength, and routability are optimized accordingly. Although the interconnect cost can be reduced efficiently in this way, the penalty in area is usually quite large.

Traditional floorplanning algorithms assumed that the modules are rectangular in shape. Some floorplanners were further limited to mosaic or slicing floorplans for simplification. Actually, some simple rectilinear shapes such as L-shape or T-shape are acceptable because the modules are usually composed of a large number of small standard cells. By allowing other rectilinear shapes, the modules can be packed more closely together.

It will be very interesting if we can reduce the area of an interconnect optimized floorplan by making use of the flexibility of the module shapes while keeping the interconnect cost unchanged.

1.2 Related Work

Most existing floorplanning approaches only deal with rectangular modules. New approaches that can handle flexible and arbitrarily shaped modules are essential to achieve high-performance design. However, floorplanning with flexible and arbitrarily shaped rectilinear modules is a complicated problem.

Both Kang and Dai [1998] and Xu et al. [1999] proposed that some integrated circuit components did not need to be rectangular. They extended the sequence-pair approach to arbitrarily sized and shaped rectilinear blocks. Young et al. [2000] showed that the area minimization problem with flexible modules can be solved optimally by geometric programming using Lagrangian Relaxation. However, the flexible modules were still restricted to a rectangular shape and the runtime was long. Metha and Sherwani [2000] presented three minimum-area floorplanning algorithms that assumed flexible and arbitrary rectilinear shapes. Those algorithms could minimize the area of the floorplan efficiently, but modules of long-snake shape could reselt. Yang et al. [2004] presented a virtual block floorplanning method to reduce the wirelength of a floorplan. They assumed that the modules can be expanded by occupying the deadspace block. Then they can assign the pin position of the modules to the expanded region such that the wirelengths of some nets can be reduced.

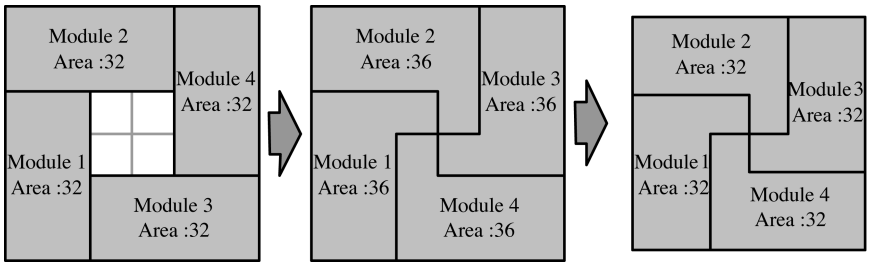

Fig. 1. An example of deadspace utilization.

1.3 Our Contributions

In this article, we assume an input floorplan that is first interconnect optimized. The deadspace in this input floorplan will be quite large. We will then apply a deadspace utilization technique to reduce the area and total wirelength. In floorplanning, most modules are not necessarily rectangular. Thus some deadspace can be redistributed to those soft modules in order to reduce the total chip area [Sham and Young 2004]. On the other hand, the shapes of the hard modules should remain unchanged and the relative positions between the modules should be maintained so that the routability will not be affected.

We devised a linear programming-based method to perform deadspace utilization (DSU). By deadspace utilization, the area and wirelength of the interconnect optimized floorplan can be further reduced subject to the constraint of keeping the routability and congestion of the original floorplan unchanged.

In this article, we will first give an overview of our design in Section 2. The implementation details of the deadspace utilization method will be described in Section 3. The experimental results will be shown in Section 4. Our conclusions will be offered in Section 5.

2. OVERVIEW OF OUR DESIGN

The modules in the floorplanning stage are not strictly rectangular. Deadspace utilization can be applied to reduce the floorplan area subject to the constraint of maintaining the routability of the original floorplan. An example is shown in Figure 1.

Given a floorplan, we will first find out all the deadspace blocks. Each deadspace block is surrounded by a number of modules. We will then assign room from the deadspace block to those modules in the surrounding to expand the occupying areas of the modules. If we can expand the occupying areas of all the modules by at least $\delta\%$ (see Table I for the definitions of the rotation used in this article), the whole floorplan can be compacted by $1 - \frac{1}{1+\delta\%}$. At the same time, the value of δ should be bounded by a congestion term because the routing congestion will be increased as the area is reduced. Thus δ should be bounded in order to maintain the routability of the floorplan. In the example of Figure 1, modules 1, 2, 3, and 4 can all be expanded by $\frac{4}{32}\% = 12.5\%$. It means that the whole floorplan can be compacted by $1 - \frac{1}{1.125} = 11\%$. This can be done by reducing the dimensions (width and length) of all modules by $(\sqrt{1.11} - 1)$.

Table I. Notations Used in This Article

Notation	Description
m_k	Module k
M	The set of all modules
M_s	The set of all soft modules
M_h	The set of all hard modules
R_i	The set of deadspace blocks that are surrounded by module m_i
d_k	Deadspace block k
w_{d_k}	The width of deadspace block k
h_{d_k}	The height of deadspace block k
$A(d_k)$	The area of deadspace block k
D	The set of all deadspace blocks
S_i	The set of modules surrounding deadspace block d_i
$E_{i,j}$	The possibly distributed area from the deadspace block d_j to module m_i where $m_i \in S_j$
$A(m_i)$	The area of module m_i
δ	The maximum area reduction ratio
G	The congestion constraint

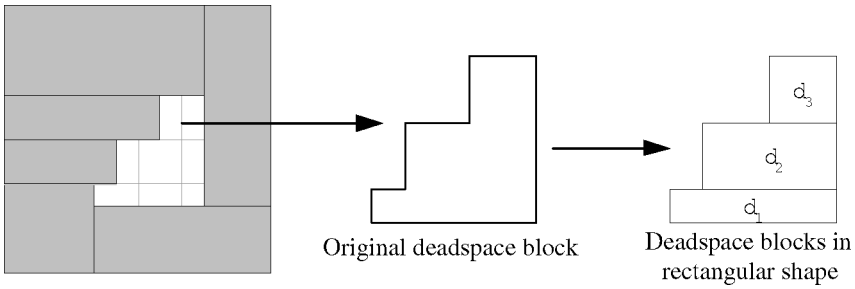

Fig. 2. Preprocessing of the deadspace blocks.

3. IMPLEMENTATION DETAILS OF DEADSPACE UTILIZATION

3.1 Preprocessing of the Deadspace Blocks

When we apply deadspace utilization, we need to find out all the deadspace blocks first. The deadspace blocks in the original floorplan may be of arbitrary rectilinear shapes. We will decompose the deadspace blocks into several smaller deadspace blocks that satisfy the following properties:

—rectangular in shape;
—each side surrounded by at most one module.

An example is shown in Figure 2. We have chosen to decompose the deadspace block into rectangular subblocks horizontally. Although there are some other decomposition methods, we believe that the result of the linear programming will only be affected slightly on average. After obtaining all these rectangular deadspace blocks, we can perform area minimization by linear programming.

Table II. Average Number of Equations for Each Test Case

Case	No. of Modules	Average Number of Equations
ami33	33	54.75
ami49	49	80.75
playout	62	92.5

3.2 Area Minimization by Linear Programming

In our design, we will try to find the maximum area reduction ratio $\delta\%$ by linear programming. First, we can at most use up the whole deadspace blocks. Second, we should ensure that all the modules can be expanded by $\delta\%$. In addition, the value of δ should be bounded by a congestion term G (the floorplan is assumed to be interconnect optimized, so we can obtain G by estimating the congestion of the floorplan) such that, when the whole floorplan is contracted by $1 - \frac{1}{1+\delta\%}$, the congestion at each tile is still below the maximum net capacity. This parameter G is given by the user according to the congestion situation. For example, if congestion will not be a problem, G can be set to infinity. Then the linear program is formulated as follows:

$$\begin{aligned}
&\text{Maximize} && \delta\% \\
&\text{Subject to} && \delta\% \le G \\
&&& \sum_{m_i \in S_k} E_{i,k} \le A(d_k) && \forall d_k \in D \\
&&& \sum_{d_i \in R_k} E_{k,i} \ge A(m_k) \times \delta\% && \forall m_k \in M
\end{aligned}$$

To solve the above linear programming problem, we will use the Simplex optimizer. The time complexity will depend on the number of equations. From the experiments (Table II), we can see that the average number of equations is linear to the number of modules in the floorplan. This is true because the number of modules is bounded by $|M|$ and the number of deadspace subblocks is bounded by $2 \times |M|$ (The upper-left corner of each deadspace subblock must touch either an upper-left or an upper-right corner of a module if we decompose the deadspace horizontally. In addition, each upper corner of a module can be touched by an upper-left corner of one deadspace subblock only. Therefore, the number of inequalities is bounded by $2 \times |M|$.) Hence, the number of inequalities is bounded by $2 \times |M|$. Thus the above linear programming problem can be solved efficiently.

After we calculate the value of δ, we can change the shapes and dimensions of the modules proportionally to compact the whole floorplan, and the total wirelength will also be reduced accordingly.

3.3 Room Assignment of Deadspace Block

In order to assign room from a deadspace block to the modules in the surroundings, we will first divide the deadspace block into four subblocks (because there are at most four neighboring modules) and assign each subblock to one neighboring module. Initially, these subblocks may be triangular or trapezoidal in

Table III. Calculations of (x_1, y_1) and (x_2, y_2)

When $(E_{L,j} + E_{R,j} \geq \frac{1}{2} \times (E_{L,j} + E_{R,j} + E_{B,j} + E_{T,j}))$:
$x_1 = \quad w_{d_j} \times \frac{E_{L,j}}{E_{L,j}+E_{R,j}}$
$x_2 = \quad w_{d_j} \times \frac{E_{L,j}}{E_{L,j}+E_{R,j}}$
$y_1 = \quad E_{B,j} \times 2/w_{d_j}$
$y_2 = \quad h_{d_j} - E_{T,j} \times 2/w_{d_j}$
When $(E_{T,j} + E_{B,j} > \frac{1}{2} \times (E_{L,j} + E_{R,j} + E_{B,j} + E_{T,j}))$:
$x_1 = \quad E_{L,j} \times 2/h_{d_j}$
$x_2 = \quad w_{d_j} - E_{R,j} \times 2/h_{d_j}$
$y_1 = \quad h_{d_j} \times \frac{E_{B,j}}{E_{B,j}+E_{T,j}}$
$y_2 = \quad h_{d_j} \times \frac{E_{B,j}}{E_{B,j}+E_{T,j}}$

shape. Transformation will be done to these subblocks to change their shapes to rectilinear. Each deadspace block is surrounded by at most four modules and each side of the deadspace block can be surrounded by at most one module. First of all, we will compute two points (x_1, y_1) and (x_2, y_2) in order to obtain the sizes and locations of the four subblocks. We use m_L, m_R, m_T, and m_B to denote the modules on the left, right, top, and bottom of a deadspace block d_j, respectively. The coordinates (x_1, y_1) and (x_2, y_2) with respect to the lower-left corner of d_j can be calculated according to Table III. Notice that if one side of the deadspace subblock is not surrounded by any module, the value of the expanded area will become zero.

After we have computed the coordinates (x_1, y_1) and (x_2, y_2), we can divide the deadspace block into four subblocks. They are denoted by sb_R, sb_L, sb_T, and sb_B. An example is shown in Figure 3. We can see that each subblock will abut with one module only and will later be assigned to that module. If $E_{L,j} + E_{R,j}$ is larger than or equal to half of the area of d_j (when the deadspace block is fully occupied), the subblocks will be divided as shown in Figure 3(a). Otherwise, the subblocks will be divided as shown in Figure 3(b). If the deadspace block is not fully occupied, the subblocks can be compacted proportionally.

In the second step, we will transform the subblocks into rectilinear shapes. After the first step, four slanted lines will be formed. We can simply change the slanted lines to Z-shaped lines to transform the subblocks into rectilinear shape. An example is shown in Figure 4. We can see that we can obtain the rectilinear-shaped subblocks by changing all the slanted lines to Z-shaped lines.

3.4 Handling Hard Modules

A circuit may consist of hard macro cells which have fixed shapes (hard modules). In order to apply deadspace utilization, we can expand the hard modules proportionally first (same aspect ratio) and then obtain the maximum area reduction ratio $\delta\%$ of the given packing by linear programming. Because the hard modules have been expanded, these modules can be compacted to restore the actual size without changing the shapes or occupying the deadspace blocks if the

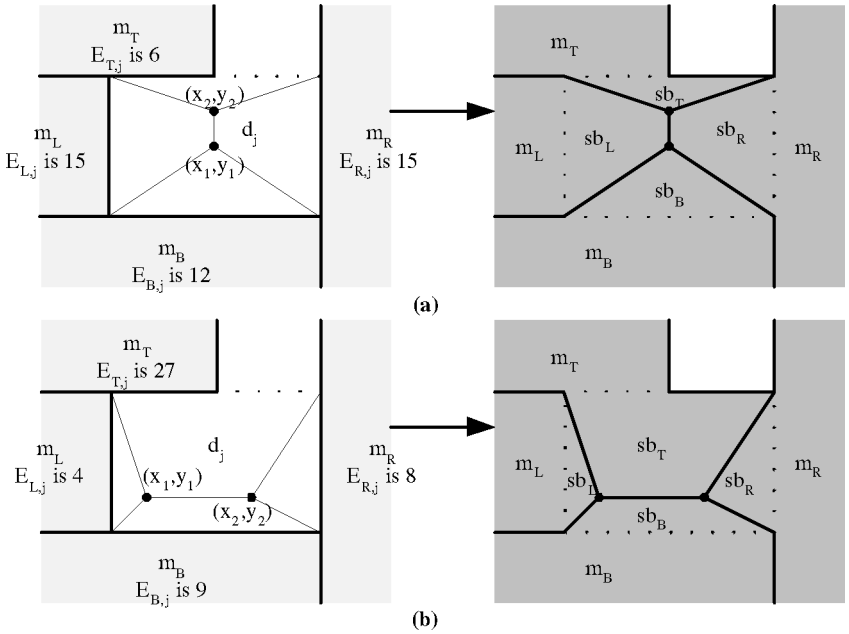

Fig. 3. Initial room assignments of a deadspace block.

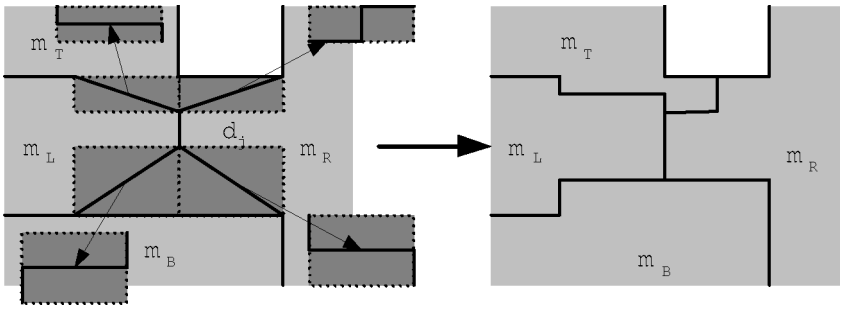

Fig. 4. Final room assignments of a deadspace block.

compaction ratio ($\delta\%$) is smaller than the expansion ratio of the hard modules. The detail design flow is shown in Figure 5

$$
\begin{aligned}
&\text{Maximize } \delta\% \\
&\text{Subject to } \delta\% \le G \\
&\qquad \sum_{m_i \in S_k} E_{i,k} \le A(d_k) \qquad \forall d_k \in D \\
&\qquad \sum_{d_i \in R_k} E_{k,i} \ge A(m_k) \times \delta\% \qquad \forall m_k \in M_s \\
&\qquad \sum_{d_i \in R_k} E_{k,i} = 0 \qquad \forall m_k \in M_h
\end{aligned}
$$

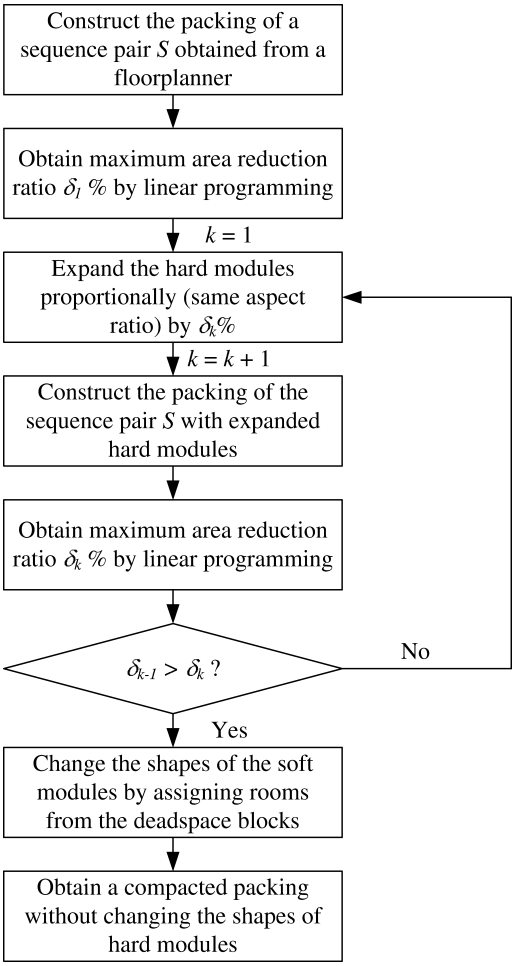

Fig. 5. Design flow for handling hard modules.

Table IV. Information of the Test Cases

Cases	No. of Cells	No. of Nets	No. of 2-pin Nets
*ami*33	33	123	305
*ami*49	49	408	526
playout	62	1611	2138

4. EXPERIMENTAL RESULTS

In the experiments, we show the improvement in area and wirelength by using the deadspace utilization (DSU) technique. We have implemented the linear programming method to perform deadspace utilization. All programs were written in the C language and run on a machine with an Intel Xeon 2-GHz processor and 1-GB memory.

The test cases used were the MCNC benchmark circuits *ami*33, *ami*49, and *playout*. We first used a floorplanner to obtain four different floorplans for each test case. The detailed information of the testing circuits are shown in Table IV.

Table V. The Improvements on Area and Wirelength Using Deadspace Utilization

Percentage of Hard Modules	Cases	δ%	Area ($10^3 \mu m^2$)		Wirelength ($10^3 \mu m$)	
			Before DSU	After DSU	Before DSU	After DSU
0%	*ami*33	11	626.3 [1.00]	550.7 [0.88]	22.08 [1.00]	20.71 [0.93]
	*ami*49	10	71957.9 [1.00]	64463.8 [0.89]	495.60 [1.00]	469.08 [0.94]
	playout	9	1798.1 [1.00]	1623.5 [0.90]	451.25 [1.00]	428.79 [0.95]
5%	*ami*33	6	626.3 [1.00]	594.0 [0.94]	22.08 [1.00]	21.42 [0.97]
	*ami*49	1	71957.9 [1.00]	70878.5 [0.99]	495.60 [1.00]	490.66 [0.99]
	playout	9	1798.1 [1.00]	1636.3 [0.91]	451.25 [1.00]	428.82 [0.95]
10%	*ami*33	3	626.3 [1.00]	607.5 [0.97]	22.08 [1.00]	21.75 [0.98]
	*ami*49	1	71957.9 [1.00]	70882.8 [0.99]	495.60 [1.00]	491.08 [0.99]
	playout	7	1798.1 [1.00]	1672.1 [0.93]	451.25 [1.00]	433.20 [0.96]

Table VI. The Number of Unroutable Net and the Run-Time of Deadspace Utilization ($G = 20$)

Percentage of Hard Modules	Cases	δ%	No. of Unroutable Net		Wirelength ($10^3 \mu m$)		Run-Time s
			Before DSU	After DSU	Before DSU	After DSU	
0%	*ami*33	11	0.0	4.0	24.18	—	0.66
	*ami*49	10	0.0	2.0	505.36	—	0.43
	playout	9	0.0	0.0	473.32	465.64	0.20
5%	*ami*33	6	0.0	0.0	24.48	23.67	0.39
	*ami*49	1	0.0	0.0	512.63	502.98	0.41
	playout	9	0.0	0.0	496.53	467.37	0.43
10%	*ami*33	3	0.0	0.0	24.32	23.09	0.39
	*ami*49	1	0.0	0.0	564.81	542.22	0.41
	playout	7	0.0	0.0	501.17	491.78	0.65

We applied the deadspace utilization technique to each floorplan. The value of the congestion term G was set to 10 and 20 (It means that the area can be reduced at most by 10% and 20%). This can be estimated according to the net densities of the original floorplan. Finally, we used a maze router (Labyrinth [Kastner et al. 2002]) to perform global routing. During global routing, we used the center of each module (original center of the module before deadspace utilization step) as the corresponding pin position. The wirelength (center-to-center estimation) and routability were obtained from the global router before and after the deadspace utilization step.

In Table V, the improvements in area and wirelength brought by the deadspace utilization technique are shown. From the experimental results, we can see that the area and wirelength can both be reduced by the deadspace utilization step. As the area is reduced, the distances between the modules are shortened and the wirelength can thus be reduced accordingly. The results also show that the improvements in area and wirelength are very significant (more than 10%) when the size of the floorplan is large. We have evaluated the performance of the deadspace utilization step by having hard modules (5% and 10%) in the experiments. When the number of hard modules increases, the improvement on area and wirelength is reduced.

In Tables VI and VII, the effects on routability (including the wirelength after global routing) of deadspace utilization are shown. In Table VI, the value of G is set to 20, we can see that unroutable nets resulted after deadspace utilization

Table VII. The Number of Unroutable Net and the Run-Time of Deadspace Utilization ($G = 10$)

Percentage of Hard Modules	Cases	δ%	No. of Unroutable Net		Wirelength ($10^3 \mu m$)		Run-Time s
			Before DSU	After DSU	Before DSU	After DSU	
0%	*ami*33	10	0.0	0.0	24.18	24.17	0.18
	*ami*49	10	0.0	0.0	505.36	506.42	0.19
	playout	9	0.0	0.0	473.32	465.64	0.20
5%	*ami*33	6	0.0	0.0	24.48	23.67	0.39
	*ami*49	1	0.0	0.0	512.63	502.98	0.41
	playout	9	0.0	0.0	496.53	467.37	0.43
10%	*ami*33	3	0.0	0.0	24.32	23.09	0.39
	*ami*49	1	0.0	0.0	564.81	542.22	0.41
	playout	7	0.0	0.0	501.17	491.78	0.65

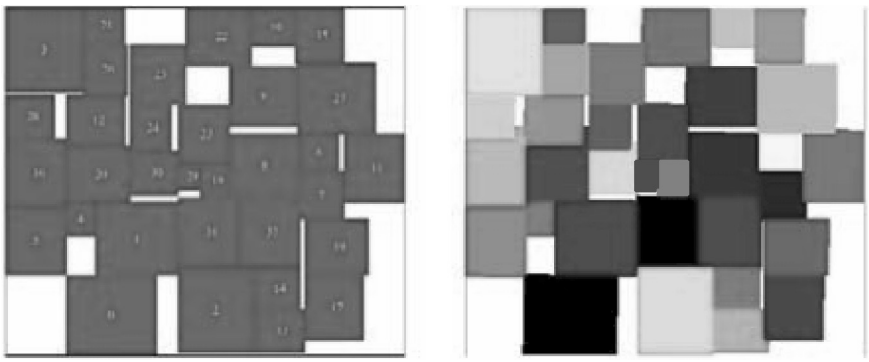

Fig. 6. Example of packings before and after deadspace utilization.

for some data sets because the circuit after compaction may have been too congested. We need to set the value of G appropriately to avoid congestion. However, if G is set to 10, the area reduction is upper bounded by 10%. This is a tradeoff between area and routability.

Although the shapes of some modules are changed, their shapes will not be very irregular and no snake-like shapes will result. An example is shown in Figure 6. In general, most of them are L-shaped and T-shaped. This is because most modules will get new spaces from one deadspace block only according to the solution of the linear program. In addition, the routability of the floorplans can be maintained. That is because the reduction ratio is bounded by the congestion term G in the linear program.

5. CONCLUSION

To conclude, most existing interconnect-driven floorplanning approaches may lead to a large penalty in chip area. However, most modules in the floorplanning stage are not strictly rectangular in shape. We propose a new approach called *deadspace utilization* to reduce the area of an interconnect optimized floorplan by making use of the flexibility of the module shapes. Experiments have shown that we can apply the deadspace utilization technique to reduce the

area and wirelength of the original floorplan further, subject to the constraint of maintaining the routability and congestion of the floorplan.

REFERENCES

CHANG, C. C., CONG, J., PAN, D. Z., AND YUAN, X. 2000. Interconnect-driven floorplanning with fast global wiring planning and optimization. In *Proceedings of the SRC Tech. Conference*.

CHEN, H. M., ZHOU, H., YOUNG, F. Y., WONG, D., YANG, H. H., AND SHERWANI, N. 1999. Integrated floorplanning and interconnect planning. In *Proceedings of the IEEE International Conference on Computer-Aided Design*. 354–357.

KANG, M. Z. AND DAI, W. W. M. 1998. Arbitrary rectilinear block packing based on sequence pair. In *Proceedings of the IEEE International Conference on Computer-Aided Design*. 259–266.

KASTNER, R., BOZORGZADEH, E., AND SARRAFZADEH, M. 2002. Pattern routing: Use and theory for increasing predictability and avoiding coupling. *IEEE Trans. Comput.-Aid. Des. Integrat. Circ. Syst. 21*, 7, 777–790.

LAI, S. T. W., YOUNG, E. F. Y., AND CHU, C. C. N. 2003. A new and efficient congestion evaluation model in floorplanning: Wire density control with twin binary trees. In *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition*.

LOU, J., KRISHNAMOORTHY, S., AND SHENG, H. S. 2001. Estimating routing congestion using probabilistic analysis. In *Proceedings of the International Symposium on Physical Design*. 112–117.

MA, Y. C., HONG, X. L., DONG, S. Q., CHEN, S., CAI, Y. C., CHENG, C. K., AND GU, J. 2003. Dynamic global buffer planning optimization based on detail nlock locating and congestion analysis. In *Proceedings of the ACM/IEEE Design Automation Conference*. 806–811.

MEHTA, D. P. AND SHERWANI, N. 2000. On the use of flexible, rectilinear blocks to obtain minimum-area floorplans in mixed block and cell designs. *ACM Trans. Des. Automat. Electron. Syst. 5*, 1 (Jan.), 82–97.

SHAM, C.-W. AND YOUNG, E. F. Y. 2003. Routability-driven floorplanner with buffer block planning. *IEEE Trans. Comput.-Aid. Des. Integrat. Circ. Syst. 22*, 4, 470–480.

SHAM, C. W. AND YOUNG, E. F. Y. 2004. Area reduction on interconnect optimized floorplan using deadspace utilization. In *Proceedings of the IEEE International Midwest Symposium on Circuits and Systems*. I445–I448.

WANG, M. AND SARRAFZADEH, M. 2000. Modeling and minimization of routing congestion. In *Proceedings of the ASP-ACM/IEEE Design Automation Conference*. 185–190.

WANG, M., YANG, X., AND SARRAFZADEH, M. 2000. Congestion minimization during placement. *IEEE Trans. Comput.-Aid. Des. Integrat. Circ. Syst. 19*, 10, 1140–1148.

XU, J., GUO, P.-N., AND CHENG, C.-K. 1999. Sequence-pair approach for rectilinear module placement. *IEEE Trans. Comput.-Aid. Des. Integrat. Circ. Syst. 18*, 4, 484–493.

YANG, C. Q., HONG, X. L., YANG, H. H., AND LU, Y. Q. 2004. An effective floorplanning algorithm in mixed mode placement integrated with rectilinear-shaped optimization for soft blocks. In *Proceedings of the IEEE International Midwest Symposium on Circuits and Systems*. (II)433–(II)436.

YOUNG, E. F. Y., CHU, C. C. N., LUK, W. S., AND WONG, Y. C. 2000. Floorplan area minimization using lagrangian relaxation. In *Proceedings of the International Symposium on Physical Design*. 174–179.

Received March 2005; revised March 2006, July 2006; accepted September 2006