INTEGRATION, the VLSI journal 42 (2009) 246–253

Contents lists available at ScienceDirect

INTEGRATION, the VLSI journal

journal homepage: www.elsevier.com/locate/vlsi

Block flipping and white space distribution for wirelength minimization

Chiu-Wing Sham [a,*,1], Evangeline F.Y. Young [b,*]

[a] Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong
[b] Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong

ARTICLE INFO

Article history:
Received 27 March 2008
Received in revised form
13 August 2008
Accepted 28 August 2008

Keywords:
Optimization
Floorplanning

ABSTRACT

Floorplanning plays an important role in the physical design of very large scale integration (VLSI) circuits. Traditional floorplanners use heuristics to optimize a floorplan based on multiple objectives. Besides traditional floorplanning approaches, some post-floorplanning steps can be applied to consider block flipping, pin assignment and white space distribution to optimize the performance. If we can consider the above three optimizations simultaneously as a post-floorplanning step, the total wirelength can be further reduced without modifying the original floorplan topology. Experimental results show that our approach can handle these issues simultaneously and wirelength can be further improved with a small penalty in runtime. Thus, this approach is highly desirable to be incorporated into a floorplanner as a post-processing step for wirelength optimization.

1. Introduction

Floorplanning plays an important role in physical design of very large scale integration (VLSI) circuits. The shapes and locations of the blocks on a chip are planned in this stage. The result of floorplanning will greatly affect the overall performance of the final circuit. As technology continues to scale down, a hierarchical approach is needed for circuit design to reduce runtime and to improve solution quality. In addition, the methodology of IP based design is widely adopted. This makes the role of floorplanning even more important. Wirelength minimization is an important objective in floorplanning. The total wirelength of a circuit layout can affect the timing closure and the routability. Thus, various wirelength-driven floorplanners [1–5] are proposed in recent years. The amount of research on this topic reflects the importance of floorplanning as a critical component for achieving timing closure in modern physical synthesis.

Besides traditional floorplanning approaches, some post-floorplanning steps can be applied to further optimize the performance. First, some movable blocks can be flipped or rotated in real circuits. we can apply such block flipping step to further reduce the total wirelength as a post-processing step while keeping the positions of all the blocks unchanged. Second, some pin positions on the blocks in floorplanning can be adjusted. The pin assignment problem of these floating pins in floorplanning can also be solved in order to minimize the wirelength. Third, the blocks are usually compacted to the lower-left (or upper-right) corner in traditional floorplanners in order to minimize area. However, the positions of the blocks can be adjusted to obtain a shorter total wirelength without changing the topology. This optimization on white space distribution can also be performed.

In this paper, we want to address the above mentioned post-floorplanning steps simultaneously. The optimal block flipping problem has been proven to be NP-complete [6]. However, many heuristics [7–13] were proposed to obtain sub-optimal solutions. A symbolic algorithm [14] based on Boolean decision diagram (BDD) was proposed that can solve the problem optimally to obtain the minimum wirelength for small-sized circuits, e.g. 20–30 blocks, but it cannot handle the cases with a large number of blocks because exhaustive searching is performed. A linear programming based approach [15] was proposed to solve this problem with more effectively. However, only fixed pins are considered in this approach. For the issue on white space distribution, an efficient min-cost flow based approach was proposed [16]. The minimum wirelength can be obtained in polynomial time while the minimum area of the given floorplan topology is maintained. Linear programming based approach was also proposed [17] for deadspace re-distribution but the shape of the blocks will be changed after applying this method.

We will present an optimal post-floorplanning approach to consider these three factors (block flipping, pin assignment and white space distribution) optimally and simultaneously. The problem can be formulated as a linear programming problem and it can be solved effectively. Although a simulated-annealing based approach can give an excellent result, we can still further

* Corresponding authors.
E-mail addresses: encwsham@poly.edu.hk (C.-W. Sham), fyyoung@cse.cuhk.edu.hk (E.F.Y. Young).

[1] The work described in this article was partially supported by the RGC Direct Allocation Fund from The Hong Kong Polytechnic University (Project no. A-PC0W).

0167-9260/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.vlsi.2008.08.001

reduce the wirelength (1.5–3.9%) with additional (0.3–2.3%) runtime. Thus, our post-floorplanning approach is still desirable to be applied in additional to the simulated-annealing based approaches.

This paper is organized as follows. First, some background information of floating pin assignment and white space distribution is given in Section 2. The formulation of the block flipping problem with floating pins is described in Section 3. The methodology of wirelength minimization by block flipping is discussed in Section 4. The adjustment of blocks by applying white space distribution is discussed in Section 6. Finally, the experimental results are shown in Section 8.

2. Background

There is no previous work addressing the block flipping problem with pin assignment simultaneously. The block flipping problem with floating pins (pin positions can be changed) is totally different from that with fixed pins only. With the same set of block positions, the optimal block orientations can be different due to different possible positions of the floating pins. An example is shown in Fig. 1. The half-perimeter metric is used to measure the wirelength. In Fig. 1(a), the pins are fixed, so their actual positions will only be affected by the corresponding block orientations. In this case, the shortest wirelength for the net connecting the three blocks can be obtained by flipping m_j only while keeping the orientation of m_i and m_k unchanged. In Fig. 1(b), the situation is similar except that the pins on all the blocks are floating (so they are marked by larger rectangles, which represent their feasible positions). In this case, the minimum wirelength is dependent not only on the block orientations, but also on the relative positions of the floating pins on the blocks. In this example, the smallest wirelength of this net can be obtained by flipping all the blocks. In addition, the corresponding pins should be placed on the rightmost position of m_i and the leftmost position of m_k in order to obtain a minimum wirelength. From this example, we can see that the block flipping problem with floating pins is totally different from the problem without floating pins.

Traditional floorplanners always give lower-left compacted floorplans. An example is shown in Fig. 2(a). Though the minimum area can be obtained, the total wirelength is not minimized. If we can adjust the position of the blocks, the topology of the floorplan can be maintained while the wirelength can be minimized such as in Fig. 2(b). The block m_i can be moved to the right and the block m_k can be moved up so that the total wirelength can be reduced. Notice that the total area and the topology of the floorplan are unchanged.

3. Problem formulation

In our approach, we consider the block flipping, assignment of floating pins and distribution of white space simultaneously to reduce the total wirelength. This will be applied as a post-processing step without changing the topology and the area of the floorplan obtained from a traditional floorplanner. The notations used in following sections are presented in Table 1.

Given a floorplan of a set of rectangular blocks M with both fixed pins (with fixed positions) and floating pins (with feasible positions within a rectangular region and all the rectangular regions are non-overlapping), we want to flip the blocks horizontally or vertically without moving the center of the block. Additionally, we want to fix the positions of the floating pins within their respective feasible regions in order to minimize the total wirelength, where the total wirelength is measured by the half perimeter bounding box (HPWL) metric. The goal of this block flipping problem is to find the optimal orientations for the blocks and the optimal positions for the floating pins such that the total wirelength is minimized. An example is shown in Fig. 3. In this example, the net connecting blocks m_i and m_k will have the shortest wirelength if both blocks m_i and m_k are flipped and the corresponding floating pins are placed at the appropriate positions in their feasible regions, respectively.

In addition, the problem on white space re-distribution can be considered simultaneously. After floorplanning, a horizontal (vertical) constraint graph can be obtained according to the

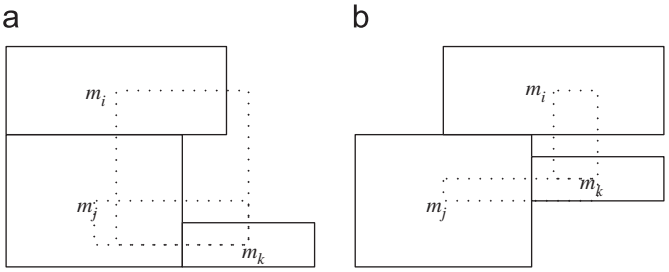

Fig. 2. Example of white space re-distribution.

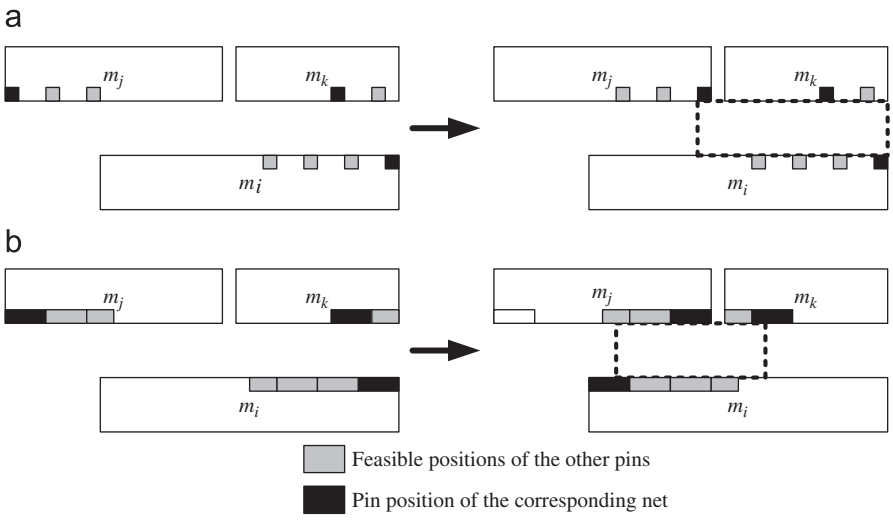

Fig. 1. Example of block flipping with floating pins.

Table 1
Notations

Notations	Descriptions
M	A set of rectangular blocks
N	A set of nets
m_i	ith block in M
n_j	jth net in N
W	Total HPWL wirelength
(w_i, h_i)	Width and height of block m_i
(x_i, y_i)	Center of block m_i
$(q_{i,j}, r_{i,j})$	Offset of a pin (center of possible pin position) on block m_i connecting to net n_j with respect to the center of m_i in the x and y directions
$(s_{i,j}, t_{i,j})$	Range of possible pin positions of a pin on block m_i connecting to net n_j in the x and y directions
$a_j(k)$	Index of the kth block connected by net n_j
z_j	Number of blocks connected by net n_j
$b_i(k)$	Index of the kth net connecting block m_i
p_i	Number of nets connecting block m_i
λ_i	Horizontal orientation of block m_i 1: flipped 0:not flipped
(L_j, L'_j)	Coordinates of the lower-left corner of the HPWL bounding box of net n_j
(R_j, R'_j)	Coordinates of the upper-right corner of the HPWL bounding box of net n_j
e_k	The edge on the constraint graph
e_k^s	The index of the block connected to the source of edge e_k
e_k^t	The index of the block connected to the sink of edge e_k

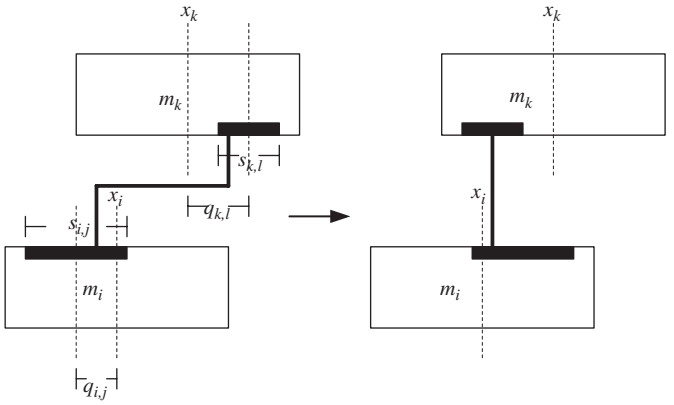

Fig. 3. Flipping blocks with floating pins.

positions of the blocks. Based on the constraint, the feasible ranges of adjustment on the positions of the blocks can be evaluated. The goal of white space re-distribution is to adjust the positions of the blocks (without change the width and height of the blocks and also the chips) in order to minimize the total wirelength while maintaining the topology of the original floorplan.

It is obvious that horizontal flipping and adjustment of blocks do not affect the y-coordinates of the pins and similarly, vertical flipping and adjustment of blocks do not affect the x-coordinates of the pins. Both dimensions of the flipping and block adjustment steps can be performed independently of each other. In the following sections, we will focus on the horizontal flipping and block adjustment steps while the vertical flipping and block adjustment steps can be applied similarly.

4. Block flipping

In this section, we will describe how to formulate the block flipping problem as a linear program (LP). Consider a net n_j connecting a set of z_j blocks, the coordinates of the pins on the blocks connecting to n_j can be represented by

$$(x_i + q_{i,j}(1 - 2\lambda_i) + g_{i,j}, \quad y_i + r_{i,j}(1 - 2\lambda'_i) + h_{i,j}) \tag{1}$$

for all $i \in \{a_j(1), a_j(2), \ldots, a_j(z_j)\}$.

Note that $\{m_{a_j(1)}, m_{a_j(2)}, \ldots, m_{a_j(z_j)}\}$ is the set of blocks connected by net n_j. λ_i tells whether block m_i will be flipped horizontally and λ'_i tells whether block m_i will be flipped vertically. If m_i is flipped horizontally, λ_i is one; otherwise, it is zero. Similarly, if m_i is flipped vertically, λ'_i is one; otherwise, it is zero. $g_{i,j}$ is the horizontal offset of the pin position with respect to the vertical center such that $-s_{i,j}/2 \leqslant g_{i,j} \leqslant s_{i,j}/2$ and $h_{i,j}$ is the offset of the pin position with respect to the center such that $-t_{i,j}/2 \leqslant h_{i,j} \leqslant t_{i,j}/2$.

We also denote the bounding box of net n_j by $(L_j, L'_j : R_j, R'_j)$ where (L_j, L'_j) and (R_j, R'_j) are the coordinates of the lower-left and upper-right corners of the bounding box, respectively. As all the pins connecting to n_j should be located inside the bounding box of n_j, we have

$$\begin{aligned} &x_i + q_{i,j}(1 - 2\lambda_i) + g_{i,j} \leqslant R_j \\ &x_i + q_{i,j}(1 - 2\lambda_i) + g_{i,j} \geqslant L_j \\ &y_i + r_{i,j}(1 - 2\lambda'_i) + h_{i,j} \leqslant R'_j \\ &y_i + r_{i,j}(1 - 2\lambda'_i) + h_{i,j} \geqslant L'_j \end{aligned} \tag{2}$$

for all $i \in \{a_j(1), a_j(2), \ldots, a_j(z_j)\}$.

Finally, in order to minimize the total wirelength (W) of the packing, we have the objective function as follows:

$$W = \sum_{j=1}^{|N|} (R_j - L_j + R'_j - L'_j) \tag{3}$$

5. Orientation fixing

Circuit designs of today are usually very complicated and involve a large number of blocks. If the block flipping problem with floating pins is directly formulated and solved as an LP, a large amount of memory and extremely long runtime will be needed. It is a good idea to reduce the number of blocks with variable orientations first before formulating the LP problem. A large portion of blocks can actually be fixed with their optimal orientations even in the presence of floating pins. If we can fix the orientations of many blocks before formulating the LP problem, the runtime and memory requirement to find the optimal orientations of the remaining blocks can be reduced significantly. According to the bounding region of each net, we can divide the blocks into three types: *independent blocks*, *solvable blocks* and *dependent blocks* as follows:

- *Independent blocks*: The orientations of these blocks will not affect the total wirelength.
- *Strictly or conditionally solvable blocks*: The orientations of these blocks will affect the total wirelength but they can be determined optimally independent of the orientations of the other blocks.
- *Dependent blocks*: The blocks that are neither independent, strictly nor conditionally solvable. Their optimal orientations are dependent on the orientations of the other "dependent blocks".

Details on independent blocks and solvable blocks will be discussed in the following sections.

5.1. Independent blocks

A block m_i is independent if

$$x_i - |q_{i,b_i(k)}| - \frac{s_{i,b_i(k)}}{2} > LR_{b_i(k)}$$
$$x_i + |q_{i,b_i(k)}| + \frac{s_{i,b_i(k)}}{2} < RL_{b_i(k)} \quad (4)$$

for all $k \in \{1, 2, \ldots, p_i\}$.

Note that $\{n_{b_i(1)}, n_{b_i(2)}, \ldots, n_{b_i(p_i)}\}$ is the set of nets connecting block m_i.

As the possible pin positions do not overlap with the bounding regions of the corresponding nets, changes in orientations of these blocks will not affect the total wirelength. An example is shown in Fig. 4. LL_j and LR_j are the minimum and maximum possible x-coordinate of the left boundary of the HPWL bounding box of net n_j, respectively. RL_j and RR_j are the minimum and maximum possible x-coordinate of the right boundary of the HPWL bounding box of net n_j, respectively. In this example, block m_1 is connected to net n_1 and n_2 as shown in Fig. 4(a) and (b), the possible pin locations on m_1 connecting to n_1 and n_2 do not lie inside the bounding regions of n_1 and n_2, respectively. Therefore, the orientation of m_1 will not affect the HPWL of n_1 and n_2 nor the total wirelength. Thus, m_1 is an independent block and its orientation can be fixed as flipped or not flipped.

5.2. Solvable blocks

For those blocks which are not dependent, their orientations will affect the total HPWL wirelength. But the orientations of some of them can still be determined optimally independent of the orientations of all the other blocks. We called such kind of blocks solvable blocks. There are two kinds of solvable blocks, strictly solvable and conditionally solvable. We will explain both of them in detail in the following sections.

5.2.1. Strictly solvable blocks

Consider a block m_i. If it is not an independent block, there must be at least a net n_j connecting m_i such that

$$x_i - |q_{i,j}| - \frac{s_{i,j}}{2} \leqslant LR_j \quad \text{or} \quad x_i + |q_{i,j}| + \frac{s_{i,j}}{2} \geqslant RL_j \quad (5)$$

Consider the set of nets $Q(m_i)$ connecting m_i and satisfying the above inequalities. If every net n_j in $Q(m_i)$ satisfies the condition that the possible positions of the pins connected by n_j (except the one on m_i) do not overlap with that of the pin on m_i connected by n_j, m_i is called strictly solvable and the optimal orientation of m_i can be determined immediately.

This is because only the nets in $Q(m_i)$ will affect the orientation of m_i. Consider a net n_j in $Q(m_i)$. The orientation of m_i will affect the position of the left or right boundary of n_j (since m_i and n_j satisfy inequalities (5)). However, no other blocks on net n_j overlap with m_i horizontally, so the effect of m_i's orientation on the wirelength of net m_j is clear, which is either $2 \times |q_{i,j}|$ or $-2 \times |q_{i,j}|$, depending whether m_i is on the left or the right boundary of n_j and the original orientation of m_i. Therefore, we can compute the total effect of m_i's orientation on W by summing up it's effects on each net in $Q(m_i)$ and determine its orientation λ_i (0 or 1).

Then, we can calculate the exact horizontal position of the floating pin on m_i connecting to n_j. If m_i overlaps with the right bounding region of n_j, the position should be $x_i + q_{i,j}(1 - 2\lambda_i) - s_{i,j}/2$. If m_i overlaps with the left bounding region of n_j, the position should be $x_i + q_{i,j}(1 - 2\lambda_i) + s_{i,j}/2$.

An example is shown in Fig. 5. We can look at block m_4 in Fig. 5. It overlaps with the bounding regions of both net n_3 and n_4, so it is not independent. However, it does not overlap with other blocks connected by those two nets. Therefore, it is a strictly solvable block. For net n_3, m_4 should not be flipped to obtain a smaller HPWL. For net n_4, it should be flipped to obtain a smaller HPWL. As $|q_{4,3}|$ and $|q_{4,4}|$ are 4 and 5, respectively, and $s_{4,3}$ and $s_{4,4}$ are 2, m_4 should be flipped. Hence, the horizontal position of the

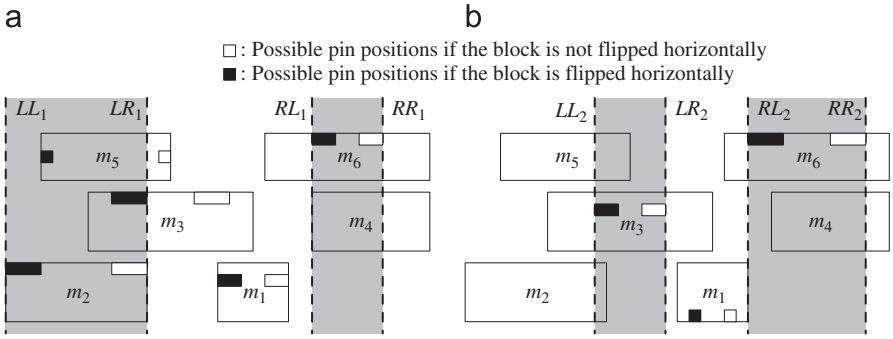

Fig. 4. An example of independent block: (a) a net n_1 connecting to m_1, m_2, m_3, m_5 and m_6 and (b) a net n_2 connecting to m_1, m_3 and m_6.

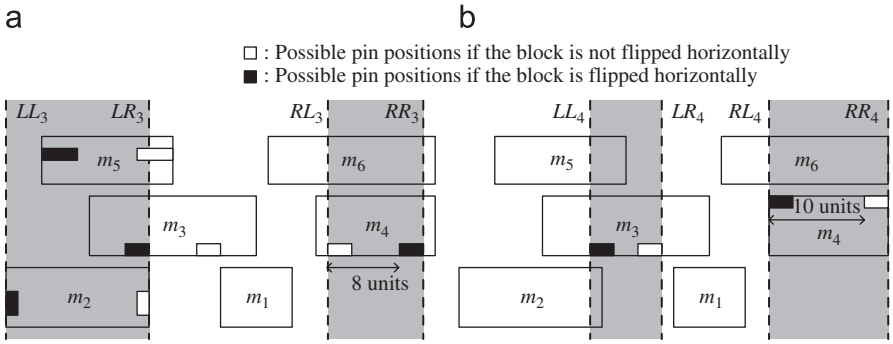

Fig. 5. An example of strictly solvable block: (a) a net n_3 connecting to m_2, m_3, m_4 and m_5 and (b) a net n_4 connecting to m_3 and m_4.

pin connecting m_4 to net n_3 and the pin connecting m_4 to net n_4 should be x_4+3 and x_4-6, respectively.

5.2.2. Conditionally solvable blocks

If a block m_i is not strictly solvable, there must be at least one net n_j in $Q(m_i)$ such that the possible positions of some pins connected by n_j overlap with that of the pin on m_i connected by n_j horizontally. Let $Q_1(m_i)$ be a subset of $Q(m_i)$ such that every net n_k in $Q_1(m_i)$ satisfies the condition that all the blocks (except m_i) connected by n_k do not overlap with block m_i, and we denote $Q(m_i)-Q_1(m_i)$ by $Q_2(m_i)$. Following the same argument for strictly solvable blocks, we can determine the effect of m_i's orientation on the total wirelength of the nets in $Q_1(m_i)$. Let this be $\delta_1(m_i)$. The value of $\delta_1(m_i)$ will determine a *potential* orientation X of m_i. Then, for each net n_j in $Q_2(m_i)$, we can determine the largest possible "adverse" effect on n_j if m_i follows the potential orientation X. This largest possible adverse effect on n_j will depend on whether m_i overlaps with the left or right bounding region of n_j and the potential orientation X. Then we can sum up all these adverse effects for each net n_j in $Q_2(m_i)$ to obtain $\delta_2(m_i)$. If $|\delta_2(m_i)|<|\delta_1(m_i)|$, block m_i is a conditionally solvable block and its optimal orientation should be X. Otherwise, m_i is not conditionally solvable and its orientation is to be determined in later steps.

The largest possible adverse effect $\delta_2(m_i,n_j)$ on a net $n_j\in Q_2(m_i)$ if m_i follows a potential orientation X depends on whether m_i overlaps with the left or right bounding region of n_j and the potential orientation X of m_i. There are three possible cases. In the first case, m_i overlaps with the right bounding region of net n_j only. In this case, if X is flipping m_i to the left (more exactly, it should be flipping the pin on m_i connecting to n_j to the left), $\delta_2(m_i,n_j)=0$. Otherwise, $\delta_2(m_i,n_j)$ can be obtained by assuming that all the other blocks on n_j that overlap with m_i are flipped in such a way to minimize the wirelength of n_j and $\delta_2(m_i,n_j)$ will be the difference in the wirelength of n_j between the case of flipping m_i to the right and flipping m_i to the left. The second case of m_i overlapping with the left bounding region of net n_j only can be considered similarly. In the third case, m_i overlaps with both the left and right bounding regions of net n_j. In this case, $\delta_2(m_i,n_j)$ will be the difference in the wirelength of n_j between the case when m_i follows the orientation X and all the other blocks are flipped in such a way to maximize the wirelength and the case when m_i follows the opposite orientation of X and all the other blocks are flipped in such a way to minimize the wirelength.

If a block is conditionally solvable, we can determine its orientation λ_i (0 or 1). Then, we can calculate the exact horizontal position of the floating pin on m_i connecting to n_j. In the first case, m_i overlaps with the right bounding region of net n_j only. The pin position should then be $x_i+q_{i,j}(1-2\lambda_i)-s_{i,j}/2$. In the second case, m_i overlaps with the left bounding region of net n_j only. The pin position should be $x_i+q_{i,j}(1-2\lambda_i)+s_{i,j}/2$. In the third case, m_i overlaps with both the left and right bounding regions of net n_j. Because the positions of the other pins inside the bounding regions are not known, we cannot determine the best possible position of the corresponding floating pin yet at this stage. In this case, we will only decide on the block orientation (λ_i) while the exact pin position will be left to be determined by the MILP or LP.

An example of a conditionally solvable block is shown in Fig. 6. We can look at block m_5 in Fig. 6. It overlaps with the bounding regions of both net n_5 and n_6, so it is not an independent block. It does not overlap with the other blocks on n_6 but overlaps with block m_2 on n_5. Thus, we have $Q_1(m_5)=\{n_6\}$ and $Q_2(m_5)=\{n_5\}$. It is obvious that m_5 should not be flipped to give a smaller HPWL when considering n_6. Thus, the value of $\delta_1(m_5)$ for this potential orientation X of not flipping is $2\times|q_{5,6}|=9$. Then, we need to determine the largest possible adverse effect on $n_5\in Q_2(m_5)$ if m_5 does not flip. The block m_5 overlaps with the left bounding region of n_5. Since the potential orientation X will put the pin on m_5 connecting to net n_5 to the left, $\delta_2(m_5,n_5)$ will be the difference in wirelength between not flipping and flipping m_5 when the orientation of m_2 is such that the wirelength of n_5 is minimized, i.e., flipping the pin connecting m_2 to net n_5 to the right. This value is 6 in this example. Finally, we have $|\delta_2(m_5)|<|\delta_1(m_5)|$, so the block m_5 is a conditionally solvable block and its optimal orientation should be "not flipped". The horizontal position of the pin connecting block m_5 and net n_5 and the pin connecting block m_5 and net n_6 should be $x_5-2.5$ and $x_5+3.5$, respectively.

After identifying the types of blocks [15], the orientations of those independent and solvable blocks can be fixed. Only dependent blocks will be considered in the LP formulation.

6. Distribution of white space

After traditional floorplanning approaches, the positions of the blocks can be obtained. Based on the positions, constraint graphs

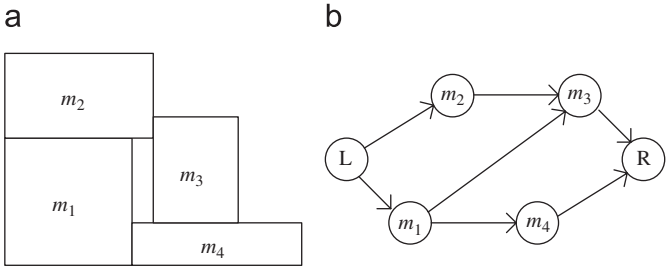

Fig. 7. Example of constructing a constraint graph for a floorplan. (a) Floorplan; (b) horizontal constraint graph.

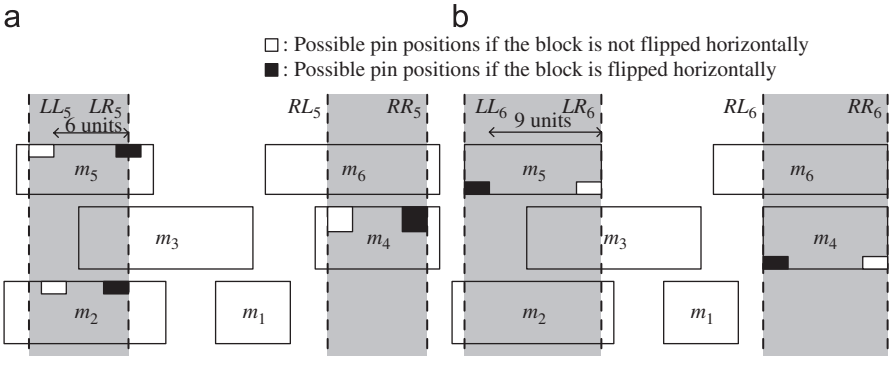

Fig. 6. An example of conditionally solvable block m_5: (a) a net n_5 connecting to m_2, m_4 and m_5 and (b) a net n_6 connecting to m_4 and m_5.

can be constructed. An example is shown in Fig. 7. The floorplan is shown in Fig. 7(a). For each pair of blocks, if they do not overlap horizontally, an edge is added on them to reflect the horizontal relationship. Thus, a horizontal constraint graph, $G_H = (V_H, E_H)$, can be constructed as in Fig. 7(b). The vertical constraint graph can also be constructed similarly.

The index of two blocks connected by edge e_k are denoted as e_k^s and e_k^t, respectively. Block $m_{e_k^s}$ should be on the left of $m_{e_k^t}$. Hence, the constraints on the horizontal positions of the blocks can be represented by the following inequalities:

$$x_{e_k^s} + \frac{w_{e_k^s}}{2} - \left(x_{e_k^t} - \frac{w_{e_k^t}}{2}\right) \leqslant 0 : \forall e_k \in E_H \tag{6}$$

We can see that the problem of block flipping, floating pin assignment and white space re-distribution is a mixed-integer linear programming problem (MILP) with a set of linear inequality constraints (Eqs. (2) and (6) and the ranges of $g_{i,j}$ and $h_{i,j}$) and a linear objective function (Eq. (3)). We can solve it optimally by invoking an MILP solver. For efficiency purpose, we can also obtain approximate solutions by solving it as an LP.

If we solve the MILP directly, the problem can be solved separately in the horizontal and vertical directions. We assume that the x-direction in the following discussion and the problem in the y-direction can be handled similarly. The MILP in the x-direction is formulated as follows:

Minimize:

$$W = \sum_{j=1}^{|N|}(R_j - L_j) \tag{7}$$

Subject to

$$\lambda_k = 0 \text{ or } 1 \quad \forall k \in \{1, 2, \ldots, |M|\}$$

$$-\frac{s_{i,j}}{2} \leqslant g_{i,j} \leqslant \frac{s_{i,j}}{2}$$

$$x_i + q_{i,j}(1 - 2\lambda_i) + g_{i,j} \leqslant R_j$$

$$x_i + q_{i,j}(1 - 2\lambda_i) + g_{i,j} \geqslant L_j$$

$$\forall j \in \{1, 2, \ldots, |N|\} \text{ and } \forall i \in \{a_j(1), a_j(2), \ldots, a_j(z_j)\}$$

$$x_{e_k^s} + \frac{w_{e_k^s}}{2} - \left(x_{e_k^t} - \frac{w_{e_k^t}}{2}\right) \leqslant 0 \quad \forall e_k \in E_H \tag{8}$$

7. LP-based optimization

According to the above MILP formulation, λ_i's are integers, $x_{e_k^s}$'s, $x_{e_k^t}$'s, $g_{i,j}$'s, R_j's and L_j's are real values, and $q_{i,j}$'s and $s_{i,j}$'s are real constants. We can use the same formulation but the constraints of λ_i's are released to any real values between 0 and 1. The problem can hence be solved as an LP instead. When there are real numbers in the solution, we will simply round the solution to 1 or 0 according to following equations:

$$\lambda_i = 1 \quad \text{if } \lambda_i > 0.5$$
$$\lambda_i = 0 \quad \text{otherwise} \tag{9}$$

We have done the experiment by both MILP and LP approaches. From the experimental results, we found that more than half of the λ_i's will be set to either 1 or 0 after solving the LP. It means that most of the blocks can be determined as "should be flipped" or "should not be flipped" by LP. Similar to the results shown in [15], the results are quite close to the optimal. In addition, results also show that the optimized total wirelength obtained by LP is at most 0.1% longer than the optimized total wirelength obtained by MILP. By solving the problem in this way, the runtime is observed to be much shorter. It is because the complexity of solving an MILP problem is much higher than solving an LP problem and the performance of the wirelength optimization can be maintained.

According to the LP formulation, the complexity of the this LP problem can be evaluated easily. There are $|M|\lambda_i$'s, $|M|$ $x_{e_k^s}$'s, $|M|$ $x_{e_k^t}$'s, $|N|$ L_j's, $|N|$ R_j's, $k|N|$ $q_{i,j}$'s and $k|N|$ $s_{i,j}$'s (k is the average pin number of each net). Thus, the total number of variables is $3|M| + (2k+2) \times |N|$. Additionally, the number of inequalities is equal to $2k|N| + |M|^2$. $2k|N|$ and $|M|^2$ are the number of $q_{i,j}$'s and $s_{i,j}$'s and the total number of relationships between any two blocks (at most $|M|^2$), respectively.

If the orientation fixing step is applied, around half of blocks can be fixed. In addition, some of pin positions can also be fixed accordingly. Thus, the total number of variables and the total number of inequalities can be reduced by half. In general, we can also run this algorithm on ISPD 2006 benchmarks with orientation fixing step (without consideration of white space distribution) with a machine having 2 GB RAM [15]. Without orientation fixing step, most cases of ISPD 2006 benchmarks cannot be completed because of insufficient memory.

8. Experimental results

In order to evaluate our proposed methods, comprehensive experiments were performed on floorplanning benchmarks. We used CPLEX9.0 as the LP solver. All programs were written in C and were run on a Sun Blade 2500 workstation with a 1.6 GHz UltraSPARC IIIi CPU and 2 GB RAM. Details of the experimental results are discussed below.

In the experiments, the test cases used are the MCNC and GSRC benchmarks. Detailed information of the testing circuits are presented in Table 2. The pre-placed circuits are obtained from the floorplanner Parquet 4.5 [5] with "*minWL*" turned on ("*areaWeight*" = 0.3 and "*wireWeight*" = 0.7). This is to ensure that the initial floorplan is already well-optimized in terms of wirelength by the floorplanner. We applied our approaches (MILP) on all the floorplanning results to see whether our post-floorplanning process can improve the wirelength. We evaluated the effect on wirelength reduction when the circuits contain floating pins. We assumed that the range, $s_{i,j}$ and $t_{i,j}$, of each floating pin are smaller than w_i/p_i and h_i/p_i, respectively, (p_i is the number of nets connecting block m_i). It can ensure that the final pin positions will not overlap with each other.

A summary of the improvement in HPWL for different portions of floating pins without white space re-distribution is presented in Table 3. P is the ratio of the floating pin number to the total number of pins. The optimal improvement is about 2.01% on average for fixed pins. The variation on the improvement for different test cases is quite large. By our investigation, Parquet can always give appropriate orientations for solvable and independent blocks. If there are more dependent blocks in the floorplan

Table 2
Information of the testing circuits

Test cases	No. of blocks	No. of pins	No. of nets	No. of pads
apte	9	97	264	73
xerox	10	209	696	2
hp	11	83	214	45
ami33	33	123	480	42
ami49	49	408	931	22
n10	10	118	179	69
n30	30	349	511	212
n50	50	455	841	209
n100	100	885	1539	334
n200	200	1585	3035	564
n300	300	1893	3789	569

Table 3
Wirelength improvement for different portions of floating pins without re-distribution of white space

Test cases	Wirelength improvement (%)			
	$P = 0\%$	$P = 25\%$	$P = 50\%$	$P = 100\%$
apte	15.91	16.24	16.57	17.20
xerox	5.31	5.50	5.68	6.05
hp	0.83	1.42	1.95	3.03
*ami*33	0.04	0.56	1.07	2.18
*ami*49	0.00	0.59	1.16	2.35
*n*10	–	0.55	1.08	2.22
*n*30	–	0.33	0.66	1.33
*n*50	–	0.32	0.64	1.31
*n*100	–	0.29	0.57	1.14
*n*200	–	0.19	0.38	0.77
*n*300	–	0.20	0.41	0.82
Average	2.01	2.38	2.74	3.49
w.r.t. $P = 0\%$	1.00	1.19	1.37	1.74

Table 4
Runtimes for different portions of floating pins without re-distribution of white space

Test cases	Runtime (s)					
	$P = 0\%$	$P = 25\%$	$P = 50\%$	$P = 100\%$	Parquet	[14]
apte	0.04	0.01	0.04	0.04	4.08	0.09
xerox	0.09	0.09	0.09	0.09	7.85	2.10
hp	0.04	0.04	0.05	0.05	5.17	0.90
*ami*33	0.05	0.05	0.05	0.05	30.48	2.30
*ami*49	0.08	0.09	0.14	0.15	71.38	19.00
*n*10	0.01	0.01	0.02	0.02	4.32	–
*n*30	0.01	0.01	0.02	0.04	35.99	–
*n*50	0.01	0.01	0.04	0.07	85.32	–
*n*100	0.04	0.06	0.07	0.09	337.84	–
*n*200	0.06	0.09	0.13	0.16	1395.26	–
*n*300	0.07	0.11	0.14	0.20	2771.45	–
w.r.t. Parquet (%)	0.32	0.32	0.37	0.38	100.0	9.13

solution (such as *apte* and *xerox* in this experiment), the possible improvement will be larger. Notice that the given pin positions of the GSRC benchmarks are at the block centers. Thus, the improvements are 0% for all the GSRC benchmarks when there is no floating pins, i.e., $P = 0$, since the wirelength is not affected by the block flipping step. The average wirelength improvement increases with the value of P. It means that if we can make use of the feasibility of pin location adjustment, the wirelength can be reduced significantly. It means that we can reduce the total wirelength significantly by exploiting representational symmetries with our approach.

Table 4 presents the runtime of our approach for different portions of floating pins without white space re-distribution. We also show the runtimes of the floorplanner Parquet to see the relative amount of runtime needed by our post-processing step. In addition, the runtime (running on Intel Pentium4, 3 GHz CPU with 2 GB memory) of the BDD-based approach [14] is also shown. The runtime of our approach is very fast. It is less than 1% of the runtime of the floorplanning step. Additionally, we should have similar wirelength compared with the BDD-based approach as we should obtain optimal total wirelength if the same intermediate floorplan results are used. Results show that our runtime of all the cases are much faster. It is because we can reduce the problem size effectively by applying the orientation fixing step when white space distribution is not considered. Second, the problem is formulated as LP which can be solved effectively.

The summary of the improvement in HPWL for different portions of floating pins with white space re-distribution is presented in Table 5. The optimal improvement is about 3.15% on average for fixed pins. It means that we can further reduce the total wirelength if the distribution of white space is considered simultaneously. The average wirelength improvement also increases with the value of P. An example of whitespace re-distribution for test case *n*10 is shown in Fig. 8. We can see that the movement of the blocks is small but the improvement on wirelength optimization is still significant.

Table 6 presents the runtime of our approach for different portions of floating pins with white space re-distribution. We also show the runtimes of floorplanner. The runtime is fast but it is slower than the one without considering the re-distribution of white space. It is around 2% of the runtime of the floorplanning step. In general, the floorplanner can give very good results. There is 4% reduction of wirelength with only around 2% additional runtime. This approach is desirable to be incorporated into a floorplanner as a post-processing step for wirelength optimization.

Table 5
Wirelength improvement for different portions of floating pins with re-distribution of white space

Test cases	Wirelength improvement (%)			
	$P = 0\%$	$P = 25\%$	$P = 50\%$	$P = 100\%$
apte	16.24	16.33	16.66	17.28
xerox	7.16	7.35	7.54	7.91
hp	4.46	5.11	5.70	6.91
*ami*33	1.33	1.91	2.42	3.54
*ami*49	1.85	2.44	3.01	4.22
*n*10	0.70	1.18	1.66	2.19
*n*30	0.59	0.91	1.24	1.57
*n*50	0.24	0.55	0.86	1.19
*n*100	0.47	0.75	1.03	1.31
*n*200	1.02	1.21	1.40	1.59
*n*300	0.54	0.73	0.94	1.14
Average	3.15	3.50	3.86	4.61
w.r.t. $P = 0\%$	1.00	1.32	1.63	1.95

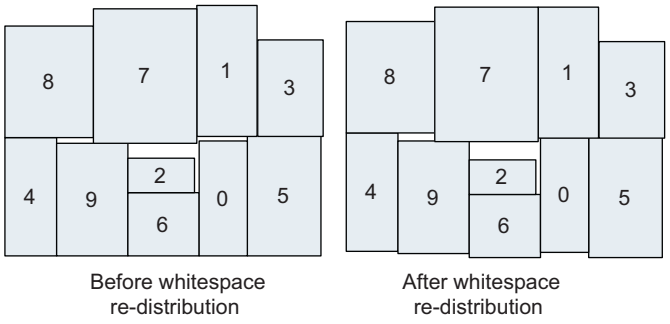

Fig. 8. Example of white space re-distribution.

Besides minimizing the wirelength, another advantage of this approach is that the topology of the floorplan can be maintained. During the floorplanning stage, a good topology can be found to optimize objectives like power, routability, timing, etc., then we can further reduce the wirelength without modifying the topology by using the proposed method.

9. Conclusion

To conclude, we have presented a detailed study on the block flipping, floating pin assignment and white space re-distribution in

Table 6
Runtimes for different portions of floating pins with re-distribution of white space

Test cases	Runtime (s)				
	$P = 0\%$	$P = 25\%$	$P = 50\%$	$P = 100\%$	Parquet
apte	0.05	0.09	0.10	0.14	4.08
xerox	0.51	0.54	0.59	0.62	7.85
hp	0.29	0.32	0.32	0.35	5.17
*ami*33	0.20	0.20	0.24	0.29	30.48
*ami*49	1.01	1.09	1.11	1.19	71.38
*n*10	0.06	0.06	0.06	0.06	4.32
*n*30	0.55	0.56	0.56	0.60	35.99
*n*50	0.64	0.65	0.66	0.72	85.32
*n*100	1.71	1.72	1.77	1.99	337.84
*n*200	6.24	6.47	6.48	6.54	1395.26
*n*300	9.87	10.25	10.40	10.48	2771.45
w.r.t. Parquet (%)	1.85	2.05	2.14	2.37	100.0

floorplanning. We consider the above three issues simultaneously as a post-floorplanning process, and the total wirelength can be further reduced while maintaining the topology. Experimental results show that our approach can handle these three issues simultaneously and wirelength reduction can be resulted with just a small penalty in runtime. Only around 0.4–2.3% of additional runtime can result in 2–4% (or even more for some cases) reduction in wirelength. This approach is highly desirable to be incorporated into a floorplanner as a post-processing step for wirelength optimization.

References

[1] H. Murata, K. Fujiyoshi, S. Nakatake, Y. Kajitani, VLSI module placement based on rectangle-packing by the sequence-pair, IEEE Trans. Comput. Aided Des. Integrated Circuit Syst. (1996) 1518–1524.

[2] Y.C. Chang, Y.W. Chang, G.M. Wu, S.W. Wu, B*-trees: a new representation for non-slicing floorplans, in: Proceedings of the 37th ACM/IEEE Design Automation Conference, 2000, pp. 458–463.

[3] X.P. Tang, R.Q. Tian, D.F. Wong, Fast evaluation of sequence pair in block placement by longest common subsequence computation, in: Proceedings of Design, Automation and Test in Europe Conference and Exhibition, 2000, pp. 106–111.

[4] J.M. Lin, Y.W. Chang, Tcg: a transitive closure graph-based representation for non-slicing floorplans, in: Proceedings of the 38th ACM/IEEE Design Automation Conference, 2001, pp. 764–769.

[5] S.N. Adya, I.L. Markov, Fixed-outline floorplanning: enabling hierarchical design, in: Proceedings of International Conference on Computer Design, 2003, pp. 1120–1135.

[6] C.K. Cheng, S.Z. Yao, T.C. Hu, The orientation of modules based on graph decomposition, IEEE Trans. Comput. (1991) 774–780.

[7] M. Yamada, C.L. Liu, An analytical method for optimal module orientation, in: Proceedings of IEEE International Symposium on Circuits and Systems, 1988, pp. 1679–1682.

[8] R. Libeskind-Hadas, C.L. Liu, Solutions to the module orientation and rotation problems by neutral computation network, in: Proceedings of the 26th ACM/IEEE Design Automation Conference, 1989, pp. 400–405.

[9] S. Nahar, E. Shragowitz, S. Sahni. Simulated annealing and combinatorial optimization, in: Int. J. Comput. Aided VLSI Des. (1989) 1–23.

[10] K. Chong, S. Sahni, Flipping modules to minimize maximum wire length, in: Proceedings of International Conference on Computer Design: VLSI in Computers and Processors, 1991.

[11] K. Chong, S. Sahni, Minimizing total wire length by flipping modules, in: Proceedings of International Conference on VLSI Design, 1992, pp. 25–30.

[12] K. Chong, S. Sahni, Minimizing total wire length by flipping modules, IEEE Trans. Comput. Aided Des. Integrated Circuits Syst. (1993) 167–175.

[13] K. Chong, S. Sahni, Flipping modules to improve circuit performance and routability, in: Proceedings of International Conference on VLSI Design, 1994, pp. 127–132.

[14] X. Hao, F. Brewer, Wirelength optimization by optimal block orientation, in: Proceedings of IEEE International Conference on Computer-Aided Design, 2005.

[15] C.-W. Sham, E.F.Y. Young, C. Chu, Optimal cell flipping in placement and floorplanning, in: Proceedings of the 43th ACM/IEEE Design Automation Conference, 2006, pp. 1109–1114.

[16] X.P. Tang, R.Q. Tian, M.D.R. Wong, Optimal redistribution of white space for wire length minimization, in: Proceedings of ASP-ACM/IEEE Design Automation Conference, 2005, pp. 412–417.

[17] C.-W. Sham, E.F.Y. Young, Area reduction by deadspace utilization on interconnect optimized floorplan, ACM Trans. Des. Autom. Electron. Syst. (2007) 3:1–3:11.

Chiu-Wing Sham received the B.Eng.(Hons.), M.Phil. and the Ph.D. degrees in computer engineering, in 2000, 2002 and 2006, respectively, from the Chinese University of Hong Kong, Shatin, NT, Hong Kong. Currently, he is a Lecturer with the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University. His research interests include digital design, algorithms and design automation of very-large-scale integration circuits. He is now working on interconnect-driven floorplanning, flexible objects placement problems and routing.

Evangeline F.Y. Young received the B.Sc. and M.Phil. degrees in computer science from the Chinese University of Hong Kong, Shatin, NT, Hong Kong, in 1991 and 1993, respectively, and the Ph.D. degree from the University of Texas, Austin, in 1999. Currently, she is an Associate Professor with the Department of Computer Science and Engineering, Chinese University of Hong Kong. Her research interests include algorithms and computer-aided design of very-large-scale integration circuits. She is now working actively on floorplan design optimization, circuit partitioning, circuit retiming and routing.