

OPC-Friendly Bus Driven Floorplanning

Hua Xiang
IBM T.J. Watson
Yorktown Heights NY
huaxiang@us.ibm.edu

Liang Deng
ECE Dept, UIUC
Urbana IL
ldeng@uiuc.edu

Li-Da Huang
Magma Corp.
Austin TX
lhuang@magma-da.com

Martin D. F. Wong
ECE Dept, UIUC
Urbana IL
mdfwong@uiuc.edu

Abstract

In this paper, we address the interconnect-driven floorplanning problem that integrates OPC-friendly bus assignment with floorplanning. Buses consist of a number of horizontal/vertical wires with identical widths. The positions of buses must be carefully designed so that the related blocks can connect to the buses with short connections. Meanwhile, as technologies march into deep sub-micron, sub-wavelength lithography causes many issues in lithographic processes. Off axis illumination (OAI) brings up the forbidden pitch issue, which could lower the yield substantially. In this paper, we first propose a litho model so that the optimal pitch can be efficiently computed for each bus. Then we present an exact algorithm to find the optimal position of a bus such that the total length of bus connections, i.e., connections from block centers to the buses, is minimized. The running time is $O(k \ln k)$ where k is the number of blocks that the bus connects. Next, we propose a linear programming based algorithm to exactly resolve overlaps among buses as well as minimizing bus connections. Furthermore, a fast heuristic approach is presented to speed up the overlap removal process. The bus assignment algorithms are smoothly integrated into the simulated annealing process of floorplanning to produce a compact floorplan with OPC-friendly buses. This work is the first one to consider litho impacts during the early floorplanning stage.

1. Introduction

As the complexity of chip design increases dramatically, chips become more congested and the connections among different macro blocks are huge. Floorplan, a very early stage in a design cycle, determines the block positions on a chip. The topology of the block placement heavily affects the connection routability and performance. This makes interconnect-driven floorplanning become a critical problem in physical design.

Buses take the major responsibility of communication among blocks, and bus routing has become more and more important due to increasingly demanding performance requirements. Since the positions of module blocks greatly affect the placement of buses, a good floorplanning strategy, which integrates bus planning with block packing, is highly desirable in the early stage of the physical design process.

In [17, 5], the authors proposed a bus-driven floorplanner. In a feasible floorplan solution, buses are realized as a 0-bend bus, i.e., a horizontal/vertical bus going through all blocks related to the bus. [9] extends the work to allow 0-bend, 1-bend and 2-bend buses. For certain critical buses, the bend constraints are good. But it is not necessary to apply the bend restrictions on all buses. Especially, these constraints may affect floorplan quality greatly. As shown in Figure 1, there are 9 blocks, and a bus connects all these blocks. Figure 1 (a) shows a placement without any deadspace. But no feasible 0-bend, 1-bend, or 2-bend bus can be found to connect

9 square blocks. In Figure 1 (b), we can assign a 2-bend bus, but the floorplan has large deadspace. Although the floorplan in Figure 1 (c) has zero deadspace, and a 0-bend bus can be assigned to realize the bus, the floorplan ratio (1 : 9) is not acceptable.

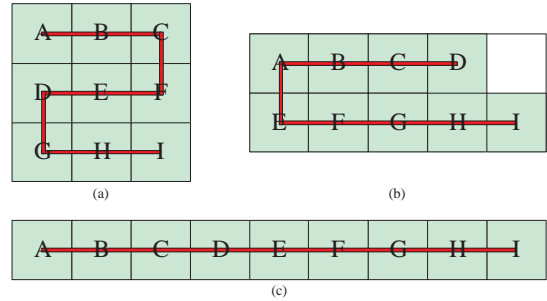


Figure 1: A floorplan problem with 9 blocks. One bus connects these 9 blocks. (a) A floorplan with zero deadspace. But the bus has 4 bends. (b) A floorplan with a 2-bend bus. But the deadspace is large. (c) A floorplan with a 0-bend bus. Although the deadspace is zero, the floorplan ratio is 1 : 9.

In this paper, we extend the bus structure so that a bus is not required to go through the blocks that are related to the bus. Instead, a bus runs horizontally or vertically over the floorplan, and blocks connect to the bus with short connections. In some sense, buses act as “highways” such that macro blocks can reach the bus easily and efficiently. As shown in Figure 2, a bus (the red line) connects 6 blocks. The blocks are connected to the bus with the short blue lines. This kind of bus structure is simple and efficient, and can be easily adopted in later design stages. For each bus, the target is to shorten the bus length and minimize the total connection between buses and blocks. Since bus assignment is closely related to the block placement, careful design of buses during floorplanning stage is definitely needed in order to ease bus routing and avoid unnecessary iterations of the physical design cycles. In this paper, we propose efficient bus assignment algorithms to decide bus positions on a given floorplan.

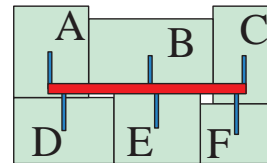


Figure 2: A floorplan with a bus connecting 6 blocks

Meanwhile, as CMOS technology is scaling down into the sub

100nm regime, it pushes the manufacturing to a limit, especially the photolithography. With the 193nm wavelength light, resolution enhancement techniques (RET) have to be intensively used for desired features. Though most RETs are now used for the poly layer, it will soon impact the design of higher metal layers. Moreover, features on metal layers are typically dense. To print better shapes, off-axis illumination (OAI) and Optical proximity correction (OPC) are adopted in metal layers [1]. However, OPC techniques greatly increase the mask cost. Layout designs obtained by tools unaware of OPC could lead to more corrections and thus higher costs. A maze routing algorithm is proposed in [7] to reduce the OPC cost. In [10], detail routing algorithm is proposed to reduce the edge placement error (EPE). Ripup and rerouting techniques are also used in post-routing optimization. All these techniques are based on the traditional on-axis illumination model.

With OAI and OPC used in sub-100nm technologies, “forbidden pitch” [14] becomes a critical issue. The interactions between features will not simply become better with a larger distance. If the pitch is in the range of forbidden pitches, the depth of focus (DOF) drops dramatically. Since the contrast and slope of light intensity are very poor, the lens aberration and the variations from photoresist will cause a large variation on the features. Ignoring OAI in early designs will lead to more expensive OPC, or even bridges and pinch errors.

Buses are usually implemented as a set of parallel wires with an identical width. Therefore, if we can carefully design the pitch for wires inside a bus to maximize the contrast and slope, we can get better printed wire features and minimize the variations. In this paper, we determine the optimal pitch for buses to derive OPC-friendly buses. This work is the first one to consider litho impacts during the early floorplanning design stage.

The rest of the paper is organized as follows. A formal definition of the problem is given in Section 2. In Section 3, a litho model is proposed to identify the optimal wire pitch in buses. Then the bus driven floorplanning algorithm is presented in Section 4. Experimental results are given in Section 5, and Section 6 concludes the paper.

2. Problem Formulation

We assume that there are two layers reserved for bus routing. One for horizontal buses, and the other for vertical buses. The OPC-friendly bus driven floorplanning problem can be defined as follows.

Given:

1. A set of n macro blocks $B = \{b_1, b_2, \dots, b_n\}$, where each block b_i has a width w_i and a height h_i ($i = 1, \dots, n$).
2. A set of m buses $U = \{u_1, u_2, \dots, u_m\}$. For each bus u_i , c_i is the number of wires inside u_i , t_i is the wire width, and block set B_i includes all of the blocks that u_i should connect. Let $|B_i| = k_i$.

The goal is to get a floorplan with OPC-friendly buses assigned.

The buses are the dominant connections among related blocks, and they are connected through short wires to blocks. For convenience, we call the connections between buses and blocks as bus connectors. The routing of bus connectors is not required to be strictly horizontal/vertical, and it can be easily handled by a router without introducing any routing violations. Since the bus connectors are short, a little bit detour does not hurt. Furthermore, at the

floorplanning stage, the pins on block boundaries may not be fixed, and the exact bus connector routing may not be applicable. We define the length of a bus connector as the distance between the block center and the center line of a bus.

Since buses include multiple identical width wires, we first apply litho analysis to find the optimal pitch of each bus. Then decide the position of macro blocks and buses such that the total chip area, the total bus length and the total bus connector length are minimized.

3. OPC-Friendly Bus

Bus structures are typically groups of long wires. If we ignore the small region at the ends of the lines, the main concern on manufacturing is to print better shapes of wires with smaller variations. Thus the post-layout or post-OPC simulation could be accurate enough to estimate the yield.

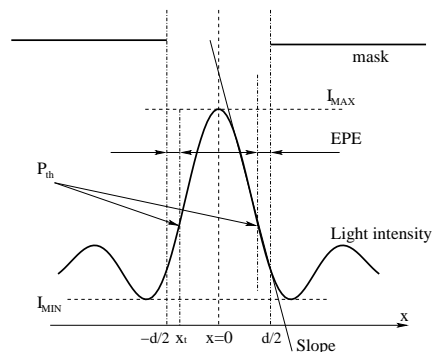


Figure 3: The mask and the aerial image. I_{MAX} , I_{MIN} , slope and EPE are measures for yield.

Edge placement error (EPE) is widely used to measure if the desired features are printed correctly. Reducing EPE can save OPC features needed on the mask, and thus reducing the mask cost substantially. However, potential printing issues are not only related to EPE. Full process window model and critical failure optical rule check (CFORC) [3] have been proposed to ensure real robust printed patterns. The maximum, minimum and slope of light intensity distributions are also needed for measuring the printability. A larger slope is preferred as it indicates higher contrast and larger DOF.

Off-axis illumination is introduced recently to improve the dense features. It allows high order frequency information of patterns pass through the illumination system. Therefore, the printability for a certain pitch can be improved. However, OAI brings the well-known forbidden pitch problem. Figure 4 shows the aerial image simulation from the Calibre [2]. The line width is set to be 80nm. Larger pitch makes the contrast worse. It is clear that 300nm pitch leads to much sharper images than 400nm pitch. With OAI, such as annular or quadrupole illumination, the process window of features becomes very small in some regions. DOF is even smaller than isolated features. This is due to the destructive interaction between neighbor features. So the width and spacing of bus wires should be carefully designed to avoid the forbidden pitch. To further reduce the geometry variation of wires, the aerial image of bus wires with higher contrast and larger slope is desired. However, since the width of bus wires varies for different buses, using Calibre [2] to simulate all possible pitches in a 2-D environment are quite time

consuming, and the long running time makes it hard to embed the litho simulation into floorplanning.

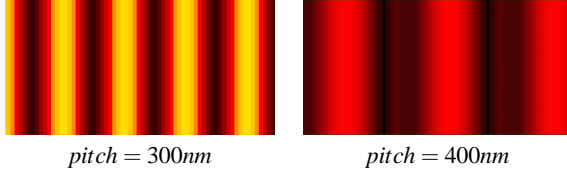


Figure 4: Simulation results for forbidden pitch from Calibre, when the wire width=80nm, $NA = 0.75$, $\lambda = 193nm$, and $\hat{f}_0 = 0.4$. Lighter region corresponds to larger light intensity.

To find the optimal pitch for a bus, we need to find out the light intensity distribution. Then the contrast and slope can be calculated to measure the printability of the aerial image. Since a bus typically consists of long wires. We can simplify the problem to one dimension. The mask spectrum of one wire considering OAI is $\hat{O}(\hat{f} - \hat{f}_0)$, where \hat{f}_0 corresponds to the off axis light source [15]. The electric field of aerial image can be obtained by convolution of the mask spectrum and the transfer function of optical system:

$$\begin{aligned}
 E_0(\hat{x}) &= \int_{-1}^1 \hat{O}(\hat{f} - \hat{f}_0) e^{-i2\pi\hat{f}\hat{x}} d\hat{f} \\
 &= \int_{-1}^1 \int_{-\infty}^{+\infty} \tilde{O}(\hat{x}) e^{i2\pi(\hat{f} - \hat{f}_0)\hat{x}} e^{-i2\pi\hat{f}\hat{x}} d\hat{x} d\hat{f} \\
 &= \int_{-1}^1 \hat{d} \text{sinc}((\hat{f} - \hat{f}_0)\hat{d}) e^{-i2\pi\hat{f}\hat{x}} d\hat{f}, \quad (1)
 \end{aligned}$$

where $\tilde{O}(\hat{x})$ is the mask pattern with normalized unit,

$$\tilde{O}(\hat{x}) = \begin{cases} 1 & \text{if } -\frac{\hat{d}}{2} < \hat{x} < \frac{\hat{d}}{2}, \\ 0 & \text{otherwise,} \end{cases}$$

This is the electric field of features at $x = 0$ with a width \hat{d} . It can be superimposed with those from other features. Thus, the electrical field $E(\hat{x}) = \sum_i E_i(\hat{x})$ for the whole region can be calculated. The light intensity of the aerial image is

$$I(\hat{x}) = |E(\hat{x})|^2 \quad (2)$$

The aerial image of a bus can be calculated by the equations discussed. With the light intensity distribution, we can easily get all I_{MAX} , I_{MIN} and slope of the aerial image. A bus consists of lines with the same width and pitch. Therefore, only the pitch needs to be decided for optimal lithography once the wire width is fixed. Since the optical system of lithography is usually optimized for the whole metal layer, we can change pitch to improve the yield for the bus.

Figure 5 shows the results from our model. We can clearly see the contrast and slope change with pitch identically. They share the same maximum point. Moreover, the curve is concave near the optimal pitch. So we can find the optimal pitch very efficiently. From Figure 5, it is easy to tell that 300nm pitch is a better choice than 400nm for a 80nm wide wire as shown in Figure 4.

4. Bus Driven Floorplanning

Simulated annealing (SA) is used to search for a solution. The candidate solution is evaluated based on the floorplan area/ratio, the

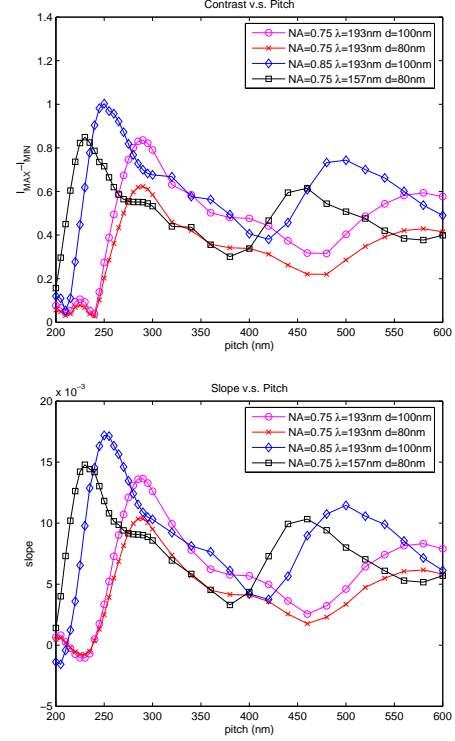


Figure 5: Contrast and Slope v.s. Pitch

total bus length and the total bus connector length. Therefore, the cost function is defined as follows.

$$Cost = \alpha \cdot A + \beta \cdot R + \gamma \cdot B + \lambda \cdot C,$$

where A is the floorplan area, R is the floorplan ratio, B is the bus length, and C is the bus connector length, and α , β , γ and λ are defined by users. There are also some other regular objectives such as wirelength and power, and they can be simultaneously handled with known techniques by using more terms in the cost function. Therefore, we only focus on the new issues, i.e., the bus assignment.

Figure 6 outlines the algorithm. The first step is to generate a floorplan which can be derived by a floorplan representation such as slicing tree [13], normalized Polish expression (NPE) [16], sequence pair [11], bounded-sliceline grid (BSG) [12], O-tree [8], B*-tree [4], corner block list (CBL) [6]. Once a floorplan is obtained, the bus assignment algorithm can be applied to find the bus positions. Then a cost is calculated for evaluation.

According to the litho model, we can get the optimal wire pitch for each bus. Therefore, the width of a bus can be easily derived. (A little extra space is added to the boundary of each bus so that no disturbance can be induced when the bus is placed close to other features.) In the following sections, we present algorithms to assign buses on a given floorplan.

4.1 Single Bus Assignment

A bus is either horizontal or vertical. If a bus is horizontal, then all the related blocks should be able to connect the bus vertically, i.e., a vertical connection from the block centers to the center line of the bus. The total bus connector length is the sum of all these vertical connections. The similar rule applies for vertical buses.

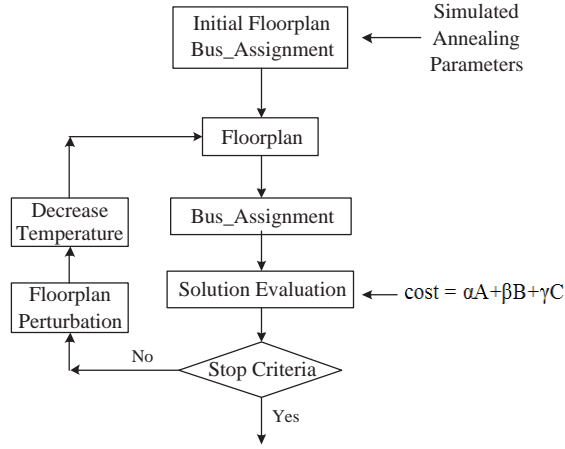


Figure 6: Bus driven floorplanning flow chart.

The following lemma gives the optimal position of a bus.

Lemma 1. *Suppose bus u connects blocks b_1, b_2, \dots, b_k . The center point of block b_i is (x_i, y_i) . If u is horizontal, then the bus spans $[\min\{x_i\}, \max\{x_i\}]$ ($i \in [1..k]$). If k is an odd number, the optimal position of the bus is the median of y_i , i.e., $y_{\frac{k+1}{2}}$. If k is an even number, any position between $[y_{\frac{k}{2}}, y_{\frac{k+1}{2}}]$ gives an optimal position.*

Similarly, if the bus is vertical, the bus spans $[\min\{y_i\}, \max\{y_i\}]$ ($i \in [1..k]$). If k is odd, the optimal position is $x_{\frac{k+1}{2}}$; otherwise, the optimal position falls in $[x_{\frac{k}{2}}, x_{\frac{k+1}{2}}]$.

Figure 7 shows examples of optimal positions of horizontal buses. The red points represent the center of macro blocks. In Figure 7 (a), there are 7 points, and their y -positions are from y_1 to y_7 , respectively. The bus spans from x_1 to x_3 . y_4 is the median of the seven y -values. When a bus is placed at y_4 as shown with the blue line, we can get the minimum total bus connectors. In Figure 7 (b), there are 6 points. Then any position between y_3 and y_4 gives the optimal position of the bus.

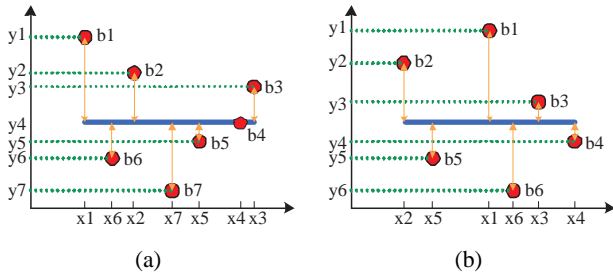


Figure 7: (a) Optimal bus position when a bus connects 7 blocks; (b) Optimal bus position when a bus connects 6 blocks.

After sorting the block positions, it takes only constant time to find the optimal bus position. Therefore, the running time of the single bus assignment is $O(k \ln k)$ where k is the number of blocks that the bus connects.

Given a floorplan, each bus can be placed in two ways: horizontal and vertical. For either orientation, it is easy to find the optimal

bus position according to the above lemma. Our target is to shorten the bus and minimize the bus connector lengths. Therefore, we can define a bus cost as

$$Cost_{bus} = \mu \cdot B_b + \nu \cdot B_c$$

where B_b is the bus length, B_c is the total bus connector length, and μ and ν are defined by users. Then for each bus, we try the two options, and compare their costs. The one with less cost is selected.

4.2 Bus Overlap Removal

In the above section, we have discussed how to decide the orientation and position of a bus. Since buses have a width, it is very likely that two buses overlap with each other. In this section, we proposed a linear programming approach to remove bus overlaps.

Given a floorplan, we decide the orientations and the initial bus positions according to the lemma. This step also decides the ordering of buses as well. During bus overlap removal, the adjustment does not change the bus length and orientation. But the bus may not be in its optimal position any more, and the length of bus connectors may increase. So the adjustment targets to minimize the total length of bus connectors.

Suppose we have M horizontal buses. For each bus u_i ($i = 1, \dots, M$), let w_i be the bus width, x_i^1 and x_i^2 be the x -coordinates of the two end points of u_i , respectively, and p_i be the initial bus position. For any two buses u_i and u_j , if $[x_i^1, x_i^2] \cap [x_j^1, x_j^2] \neq \emptyset$, then if $p_i > p_j$ or $p_i = p_j (i > j)$, we say u_i is above of u_j . During bus position adjustment, we maintain the ordering of buses, i.e., if initially u_i is above u_j , then u_i is still above of u_j after adjustment. We can formulate the bus overlap removal problem as follows.

Let p'_i be the new position of bus u_i . Suppose that the macro blocks connected by bus u_i are defined in the block set B_i , and (x_i, y_i) is the center of the macro block b_i . Then our target is

$$\min \sum_{i=1}^M \left(\sum_{b_j \in B_i} |p'_i - y_j| \right)$$

subject to

$$p'_s - p'_t \geq (w_s + w_t)/2 \quad \text{if } [x_s^1, x_s^2] \cap [x_t^1, x_t^2] \neq \emptyset \text{ and } u_s \text{ is above } u_t$$

Let $t_{ij} = |p'_i - y_j|$. Then the above problem can be transformed to a linear programming problem.

$$\min \sum_{i=1}^M \left(\sum_{b_j \in B_i} t_{ij} \right)$$

subject to

$$\begin{aligned} p'_i - y_j &\leq t_{ij}; \\ -t_{ij} &\leq p'_i - y_j; \\ t_{ij} &\geq 0; \quad i = 1, \dots, M; \quad b_j \in B_i \\ p'_s - p'_t &\geq (w_s + w_t)/2; \quad \text{if } [x_s^1, x_s^2] \cap [x_t^1, x_t^2] \neq \emptyset \text{ and } u_s \text{ is above } u_t \end{aligned}$$

The results of the linear programming problem return the optimal position of the M buses such that the total length of bus connectors is minimized, and no buses have overlap.

4.3 Fast Bus Assignment

During the high temperature period of the simulated annealing process, the solution candidate is far from optimal. Usually the floorplan is not well packed and it has large deadspace. Therefore, the cost is dominated by the floorplan area. (A large floorplan may also lead to longer buses and bus connectors.) Although a precise bus assignment can help reduce the total cost, the impact is limited and the primary target at this stage is to minimize the chip area. Therefore, we propose a fast heuristic approach for bus assignment to speed up the execution.

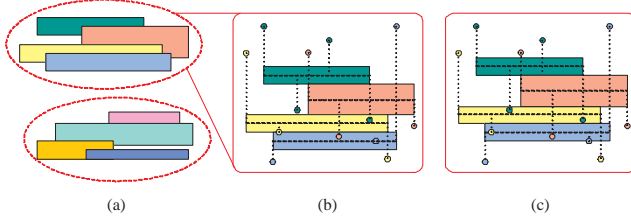


Figure 8: (a) Partition buses into two groups. (b) Spread buses to resolve overlaps. (c) Shift buses to reduce the connections between buses and blocks.

Given a floorplan, we first get the initial bus positions. If there are no overlaps between any two buses, the solution is good. Otherwise, some steps have to be taken to resolve the overlaps. If any two buses overlap, they are regarded as one group. Therefore, the first step is to partition buses into groups according to the overlap relationship. Then for each group, we resolve the overlap by placing the buses next to the other. The bus ordering obeys the original bus ordering. In this step, the relative positions of these buses are set. Next, we find the optimal position for the group so that the bus connectors are minimized. Figure 8 illustrates an example. In Figure 8 (a), there are 8 horizontal buses. They are clustered into two groups according to their overlap relationship. Figure 8 (b) spreads the four buses in one group to resolve overlaps. Then the four buses shift upwards to minimize the total bus connector length as shown in Figure 8 (c). The algorithm is summarized as follows.

Algorithm Fast_Bus_Assignment()

1. Assign bus initial positions;
2. While (overlap exists)
 3. Partition overlap buses into groups;
 4. For each group
 5. Spread buses to remove overlaps;
 6. Adjust buses to minimize connectors;
 7. Endfor
8. Endwhile

Suppose there are M horizontal buses in a group. For a bus u_i ($i = 1, \dots, M$), the width is w_i , the initial position is p_i and the x -coordinates of the two end points are x_i^1 and x_i^2 . Assume u_1 is the bus on the bottom, i.e., all other buses are above u_1 . Then we fix u_1 , and spread buses from bottom to top. The new position u_i is $p'_i = p'_{i-1} + (w_{i-1} + w_i)/2$ ($i = 2, \dots, M$). This step removes all overlaps among buses. At the same time, the distance between u_i and u_1 is set. Let $d_i = p'_i - p_1$ ($i = 1, \dots, M$). Then the target is to

minimize the bus connectors.

$$L_c = \sum_{i=1}^M \left(\sum_{b_j \in B_i} |p'_i - y_j| \right),$$

where B_i is the macro block set related to bus u_i , and y_j is the y -coordinate of the center of block b_j . Then we have

$$L_c = \sum_{i=1}^M \left(\sum_{b_j \in B_i} |(p_1 + d_i) - y_j| \right) = \sum_{i=1}^M \left(\sum_{b_j \in B_i} |p_1 - (y_j - d_i)| \right)$$

Both y_j and d_i are fixed. So p_1 is the only variable. To minimize L_c , the above problem can be transformed to a single bus assignment problem. In the equivalent single bus assignment problem, a bus need connect $K = \sum_{i=1}^M |B_i|$ macro blocks, where $|B_i|$ is the number of blocks inside B_i . The y -coordinate of the center of these blocks is $y_j - d_i$. According to the lemma, when K is an odd number, the median gives the optimal value of p_1 ; when K is an even number, any value between the two middle points returns the minimum value of L_c .

The above procedure resolves the overlaps inside one group. However, after bus spreading, new overlaps among buses may be introduced as shown in Figure 9. In Figure 9 (a), there are two groups. After resolving overlaps within each group, new overlaps are introduced between two groups as illustrated in Figure 9 (b). Therefore, all these buses are regarded as one group and the spreading process is applied to resolve overlaps. Figure 9 (c) shows the final solution. Vertical buses can be handled similarly.

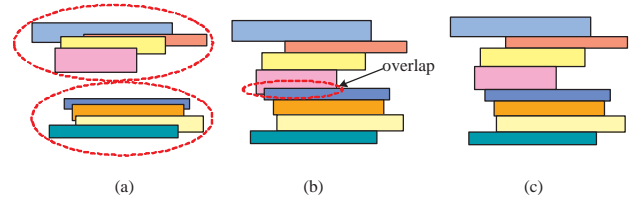


Figure 9: (a) Buses in two groups. (b) New overlap is introduced when resolving overlaps inside groups. (c) Remove overlap.

In Fast_Bus_Assignment, the initial position of a bus u_i can be obtained in $O(mn \cdot \ln n)$, where m is the number of buses and n is the number of blocks. To remove overlaps, it takes at most $O(m)$ iterations. For each iteration, the running time can be bounded by $O(m \cdot N \cdot \ln N)$, where N is the total number of bus connectors. Therefore, the total running time of Fast_Bus_Assignment is $O(m^2 \cdot N \cdot \ln N)$.

5. Experimental Results

Our algorithm was implemented in C on a PC workstation (1.8GHz) with 1.5GB memory. The test cases are derived from the MCNC benchmarks for floorplanning. Different numbers of buses are added to the benchmarks. We compared our algorithms with the algorithm in [17]. Our algorithms also use sequence pair for floorplan representation as [17]. Since [17] does not calculate bus widths, we first use our litho model to get bus widths, then apply the BDF algorithm in [17]. The optical wavelength is 193nm, and the line width and space are based on 90nm process.

The test results are listed in Table 1. The last four test cases include some buses that connect a larger number (8 or 9) of blocks.

Table 1: Experimental Results

File	Block	Bus	Bus Width CPU Time (s)	[17]				Our work			
				Assigned Buses	Dead Space	BusLength (μm)	Time (s)	Assigned Buses	Dead Space	BusLength (μm)	Time (s)
apte	9	5	12	5	3.33%	1597.0	14	5	3.25%	1362.1	4
xerox	10	6	14	6	3.06%	1441.0	23	6	2.42%	1212.1	10
hp	11	14	34	14	4.47%	1512.7	154	14	4.47%	1519.0	116
ami33-1	33	8	21	8	7.85%	610.4	59	8	4.01%	531.7	17
ami33-2	33	18	44	18	7.71%	422.1	171	18	7.56%	498.4	48
ami49-1	49	9	22	9	6.35%	2437.1	153	9	4.45%	2537.5	139
ami49-2	49	12	29	12	6.99%	3290.8	56	12	3.65%	3985.1	24
ami49-3	49	15	36	15	7.92%	417.1	111	15	2.87%	4136.3	44
ami33-3	33	9	22	6	9.28%	-	219	9	7.70%	818.0	81
ami33-4	33	7	17	5	5.71%	-	181	7	5.70%	517.0	50
ami49-4	49	8	18	6	4.65%	-	212	8	3.61%	3176.6	35
ami49-5	49	10	24	9	17.54%	-	283	10	4.33%	4265.8	49

For all test cases, our algorithm can assign all buses with smaller deadspace in shorter running time, while the BDF algorithm [17] has difficulty in finding a floorplan with all buses assigned. The bus lengths are also comparable to those of [17]. The bus length of the last four test cases is not provided since not all buses are assigned. As an illustration, Figure 10 displays the final packing result with 12 buses.

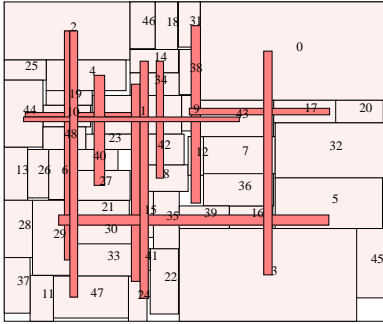


Figure 10: Floorplan for ami49-2.

6. Conclusion

In this paper, we propose an OPC-friendly bus driven floorplanning algorithm. A litho model is presented to calculate the optimal bus pitches. This helps produce OPC-friendly buses so that great efforts can be saved from fixing litho problems in the post design stage. On the other hand, bus positions must be well designed so that the related blocks can connect to the buses with short connections. An optimal algorithm is proposed to identify the position of a bus such that the total connections between blocks and buses are minimized. The running time is $O(k \ln k)$ where k is the number of blocks that a bus connects. Two bus removal algorithms are presented. One is based on linear programming, and it exactly resolves overlaps among buses as well as minimizing bus connector lengths. The other is fast with running time $O(m^2 \cdot N \cdot \ln N)$ where m is the number of buses and N is the total number of bus connectors. The bus assignment algorithms can be smoothly integrated into the simulated annealing process of floorplanning. Experimental results demonstrate the efficiency and effectiveness of our algorithm.

7. References

- [1] ITRS. International technology roadmap for semiconductors 2004. Technical report, 2004.
- [2] Mentor Graphics, Calibre Model-Based OPC Users Manual, 2004.
- [3] A. Borjon, and et. al., High accuracy 65nm OPC verification: full process window model vs. critical failure ORC, Proc. SPIE, vol 5754, pp.1190-1201, 2005.
- [4] Y.C. Chang, Y.W. Chang, G.M. Wu, and S.W. Wu, B*-trees: A new representation for non-slicing floorplans, Proc. DAC, pp. 458-463, 2000.
- [5] T.C. Chen, and Y.W. Chang. Modern floorplanning based on fast simulated annealing, ACM International Symposium on Physical Design, pp 104-112, San Francisco, CA, April, 2005.
- [6] X. Hong, G. Huang, T. Cai, J. Gu, S. Dong, C.K. Cheng, and J. Gu, Corner Block List: An effective and efficient topological representation of non-slicing floorplan, Proc. ICCAD, pp. 8-12, 2000.
- [7] L. Huang, and M.D.F. Wong, Optical proximity correction (OPC)-friendly maze routing, Proc. DAC, pp. 186-191, 2004.
- [8] P.N. Guo, C.K. Cheng, and T. Yoshimura, An O-Tree representation of non-slicing floorplan and its applications, Proc. DAC, pp. 268-273, 1999.
- [9] J.H.Y. Law, and E.F.Y. Young. Multi-bend bus driven floorplanning, ACM International Symposium on Physical Design, pp 113-120, San Francisco, CA, April, 2005.
- [10] J. Mitra, P. Yu, and D.Z. Pan, RADAR: RET-aware detailed routing using fast lithography simulations, Proc. DAC, pp. 369-372, 2005.
- [11] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani, Rectangle-packing based module placement, Proc. ICCAD, pp. 472-479, 1995.
- [12] S. Nakatake, K. Fujiyoshi, H. Murata, and Y. Kajitani, Module placement on BSG-Structure and IC layout applications, Proc. ICCAD, pp. 484-491, 1996.
- [13] R.H.J.M. Otten, Automatic floorplan design, Proc. DAC, pp.261-267, 1982.
- [14] R. Socha, M. Dusa, L. Capodiceci, J. Finders, F. Chen, D. Flagello, and K. Cummings. Forbidden pitches for 130nm lithography and below. In Proc. of SPIE, volumn 4000, pp 1140-1155, 2000.
- [15] A. K. K. Wong, Resolution Enhancement Techniques in Optical Lithograph, SPIE Press, 2001.
- [16] M.D.F. Wong, and C.L. Liu, A new algorithm for floorplan design, Proc. DAC, pp. 101-107, 1986.
- [17] H. Xiang, X. Tang, and M.D.F. Wong. Bus-driven floorplanning, IEEE/ACM International Conference on Computer Aided Design, pp 66-73, San Jose, CA, Nov. 2003.