Integrated Floorplanning and Interconnect Planning*

Hung-Ming Chen[1], Hai Zhou[1]†, F.Y. Young[1]‡, D.F. Wong[1], Hannah H. Yang[2], and Naveed Sherwani[2]
[1] Department of Computer Sciences, University of Texas at Austin, Austin, TX 78712
[2] Strategic CAD Labs, Intel Corporation, Hillsboro, OR 97124

Abstract

When VLSI fabrication enters the deep sub-micron era, communication between different components is significantly increased. Interconnect delay also becomes the dominant factor in total circuit delay. All these make it necessary to start interconect planning as early as possible. In this paper, we propose a method to combine interconnect planning with floorplanning. Our approach is based on the Wong-Liu floorplanning algorithm. When the positions, orientations, and shapes of the cells are decided, the pin positions and routing of the interconnects are decided as well. We use a multi-stage simulated annealing approach in which different interconnect planning methods are used in different ranges of temperatures to reduce running time. A temperature adjustment scheme is designed to give smooth transistions between different stages of simulated annealing. Experimental results show that our approach performs well.

1 Introduction

With VLSI fabrication entering the deep sub-micron (DSM) era, devices are scaled down to smaller sizes and placed at an ever increasing proximity. At the same time, with the increase of die dimensions, more functions are integrated into one chip. All these significantly increase the communication between different components, thus increasing the amount of interconnect on a chip. Moreover, the scaling down of fabrication geometry also makes interconnect delay a dominant factor in total circuit delay [1]. These trends make interconnect planning a neccessary step in DSM design [4].

Global interconnects have significant influence on system performance in DSM technologies. Floorplanning, the process of placing functional blocks on the chip, can significantly affect the global interconnect structure. Many floorplanning algorithms have been proposed in the past 20 years [5, 9, 2, 8, 3, 6]. All these algorithms focus on placing the circuit blocks using simple interconnect cost (e.g., total wire length) to guide the optimization. Without accurate interconnect planning during the floorplanning process, it is difficult for these algorithms to meet performance constraints due to unexpected "long" global interconnects resulted in the later routing stage.

In this paper, we propose a method to combine interconnect planning with floorplanning. Our approach is based on the Wong-Liu floorplanning algorithm [9]. Recall that the Wong-Liu algorithm uses Polish expressions to represent floorplans and searches for an optimal floorplan using simulated annealing by iteratively generating Polish expressions. Every time a Polish expression (i.e., a floorplan) is examined, the shape of the blocks are optimized and the total wire length is used as the interconnect cost. In this paper, instead of using the total wire length, we propose to perform careful interconnect planning with respect to the current floorplan being considered and obtain a much more accurate interconnect cost.

The interconnect planning step performs pin assignment and simple-geometry routing based on L-shaped and Z-shaped wires. Taking advantage of the nature of simulated annealing, we use different interconnect planning methods in different ranges of temperatures to reduce the running time. In particular, we use the conventional wirelength estimation by half-perimeter of net bounding box when temperature is high, and use a more accurate interconnect cost based on L-shaped routing when temperature is in the medium range, and finally use Z-shaped routing when temperature is low. In order to implement our multiple cost function scheme, we found that it was necessary to introduce a temperature adjustment method to cope with the intrinsic discontinuities resulted in the process of switching cost functions.

The rest of the paper is organized as follows. We introduce the algorithms for interconnect planning in Section 2. Section 3 discusses the multi-stage simulated annealing approach. Section 4 reports the experimental results for MCNC benchmarks and Section 5 concludes the paper.

2 Efficient Interconnect Planning

To simplify our discussion, we assume there are two layers for the routing of global interconnects – one layer for vertical wires and the other layer for horizontal wires. (However, our approach is applicable to designs with more than two layers.) We allow different layers to have different design rules, i.e., the minimum wire width and the minimum spacing in each layer are different. In order to estimate congestion/routability, we divide the floorplan into a number of bins by a grid the same way that it is typically done in global routing [7]. For each bin boundary, we define its *capacity* as the maximum number of nets that can cross it. Clearly, the capacity of each boundary can be easily computed based on its length (or width) and the design rules (i.e., minimum wire width and minimum wire spacing) for that layer. If the number of nets crossing a bin boundary exceeds the capacity

*This work was partially supported by a grant from the Intel Corporation and by the Texas Advanced Research Program under Grant No. 003658288.

†Hai Zhou is now with Advanced Technology Group, Synopsys Inc.

‡F.Y. Young is now with Department of Computer Science and Engineering, The Chinese University of Hong Kong.

of the bin boundary, we say there is *overflow*. Each global routing solution gives us the number of nets crossing each bin boundary, thus giving us detailed congestion/overflow information. Our goal is to plan the interconnects to avoid congestion/overflow as much as possible.

2.1 Pin Assignment

The first step of interconnect planning is pin assignment. After module sizes and positions are fixed in a given floorplan, we determine the pin positions on each module. A simple strategy is used for efficiency. For each net, we connect the centers of the modules in this net and get the intersection points on the module boundaries as pin positions. This simple heuristic makes sense since it tries to minimize total wirelength. Note that each module boundary is partitioned into a number of boundary segments by the grid. Since each boundary segment can only accomodate a limited number of pins, we should make sure that the number of pins we assign to each boundary segment does not exceed its capacity. If segment overflow occurs, we redistribute some of the pins to neighboring segments. Another guideline for pin assignment is to evenly distribute the pins so that no boundary segments are too crowded.

2.2 Simple-Geometry Routing

After pin assignment, pin positions are known. We then perform simple-geometry based global routing to connect the pins. For a net with n pins where $n > 2$, we first construct a minimum spanning tree connecting the pins using the Manhattan distance metric. The net is then decomposed into a set of two-terminal nets which correspond to the edges of the minimum spanning tree. After that, we have a set of nets with only 2 pins. For each of them, we connect the two pins using simple-geometry routing based on L-shaped or Z-shaped wires. Since the algorithms for L-shaped routing and Z-shaped routing are similar, they will be described together. Before we do simple-geometry routing, we map the pin positions of the nets to the corresponding bins. We use a sequential routing approach, that is, we route one net at a time. There are two steps in our simple-geometry routing algorithms. The first step is to use a stochastic approach to obtain the initial global congestion information. The second step is to utilize the information from the first step to route nets one by one.

In the first step, we estimate the congestion on each bin boundary by the expected number of nets crossing that boundary. Consider a two-pin net with pins $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$. If only L-shaped routes are allowed, there are at most two routes to connect the two pins, as shown in Figure 1. Assume that each possible route is equally likely, we can add 1/2 to each bin boundary on the two routes as the net's contribution to the expected number of nets crossing that boundary. For Z-shaped routing, we compute the expected number of nets crossing each bin boundary as follows. Let m denote the total number of Z-shaped routes connecting p_1 and p_2. As we can see, if $x_1 = x_2$ or $y_1 = y_2$, then $m = 1$. Otherwise, m can be computed as follows.

$$m = |x_1 - x_2| + |y_1 - y_2|$$

For each bin boundary e, let m_e be the number of possible Z-shaped routes for the net to cross e. We again assume all routes are equally likely. Clearly, the net's contribution to the epxected number of nets crossing e is m_e/m. For the example shown in Figure 2 for Z-shaped routing, $m = 6$ and $m_e = 1$ where e is the right boundary of *bin(2,3)*. Thus the net's contribution to the expected number of nets crossing e is $m_e/m = 1/6$. Summarizing contributions from different nets, we can get the expected number of crossing nets on each boundary.

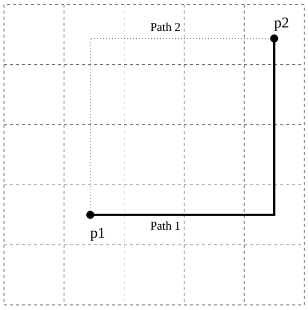

Figure 1: L-shaped Routing

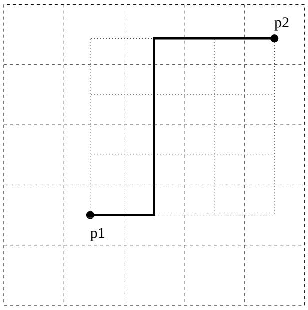

Figure 2: Z-shaped Routing

In the second step, we route one net at a time. When routing a net, we first remove its contribution from the expected number of crossing nets at each bin boundary. Then we determine a routing path with minimum crossing cost. The cost of crossing a bin boundary depends on a few factors. We use OF_e to represent the overflow amount on bin boundary e, RD_e to represent the difference between the current crossing (no overflow yet) and the capacity of e, and OVL to represent the overlapping length with previously routed wires belonging to the same (multi-pin) net. We determine a routing path which minimizes the following quantity: $\alpha \sum OF_e^2 + \beta \sum 1/RD_e^2 - \gamma OVL^2$. The first part is a penalty term, meaning that the global router is penalized because of going through the congested bin boundary. The second term is a prevention term, that is, the global router prevents from taking the path that is reaching saturation of the capacity. The third term is a reward that the router follows previous routes for those two-terminal nets within a multi-terminal net. After routing a net, if the route crosses a bin boundary e, we add 1 to the expected number of nets crossing e to reflect the real route. If the current crossing of the bin boundary exceeds the capacity, mark this net to be riped-up and re-routed. For all nets that need to be riped-up and re-routed, we process them in the order from the most congested net, which is the net crossing the maximum number of congested bin boundaries, to the less congested ones trying to remove overflow as much as possible. Then we examine the results by getting the total square overflow terms of all bin boundaries. If the current overflow status exceeds the former one, we recover the net to its original route.

3 Multi-Stage Simulated Annealing

Among our two interconnect planning approaches, Z-shaped routing is more accurate than L-shaped routing. But Z-shaped routing is also more expensive than L-shaped routing. Using Z-shaped routing all along will give the most accurate estimation. However, based on the characteristics of simulated annealing, we can speed up the procedure without sacrificing the quality of solutions.

The Wong-Liu floorplanning algorithm [9] is based on simulated annealing which is a technique for solving general optimization problems. The algorithm moves from one solution to another, tring to find the optimum solution. It accepts a move with the probability $e^{-\Delta C/T}$, where ΔC is the increase of cost by that move and T is the current temperature. When the temperature is very high, different estimation methods for the cost will not show much difference on $-\Delta C/T$. This means it does not affect much in performance if we use rough cost function at the beginning of annealing. When temperature gradually decreases, we use more accurate cost estimation. As we can see, the L-shaped routing estimation is more accurate than the simlple center-to-center or half-perimeter estimation. Similarly, the Z-shaped routing is more accurate than the L-shaped routing estimation. Therefore, we will start with the the center-to-center or half-perimeter estimation, gradually transfer to the L-shaped routing, and finally reach the Z-shaped routing. This multi-stage approach is very effective in reducing total running time.

In fact, multi-stage simulated annealing is just a method to combine different approaches together in one process. It should be reasonable if those different approaches used in multi-stage simulated annealing are not totally different, which means they have a certain degree of correlation. In this paper, we use a three-stage simulated annealing approach. The first stage is to get a good initial solution by using only the half-perimeter wire length estimation. The second stage is to estimate interconnect cost by using L-shaped global routing. The third stage is to estimate interconnect cost by using Z-shaped global routing. The transitions between stages are not very abrupt since they evolve from simple to complex, from rough to accurate. However, even for very similar estimations, we still need to find a way to take care of any possible discontinuity in switching cost functions.

3.1 Cost Function Transitions

The cost function used in [9] is $A + \lambda W$, where A is the total area of the packing, W is the half-perimeter estimation of the interconnect cost, and λ is a constant which controls the relative importance of these two terms and is usually set such that the area term and the interconnect term are approximately balanced. In this paper, we use the cost function $\psi = \alpha A + \beta W + \gamma OF$, where A and W are the same as in [9] and OF is the sum of the square of overflow in routings. Although the format of cost function is identical for three stages of the process, the content of each term is different. The term W in stage 1 is obtained by applying half-perimeter method of bounding boxes; the term W in stages 2 and 3 are obtained by applying pin assignment and summing the net length from pin positions. The term OF in stage 1 is zero; the term OF in stages 2 and 3 is applying simple-geometry routing and computing the congestion/routability estimation of bin boundaries. As we mentioned earlier, in order to cope with the discontinuities resulted in the process of switching cost functions, we introduce a temperature adjustment method, which is described in next sub-section.

3.2 Temperature Adjustment

Simulated annealing uses temperature to control the probability in accepting uphill moves. We use a temperature schedule of the form $T_k = r * T_{k-1}, k = 1, 2, 3, ...$ The initial temperature T_0 is determined by performing a sequence of random moves and computing the quantity Δ_{avg}, the average value of the magnitude of change in cost per move. We should have $e^{-\Delta_{\text{avg}}/T_0} = P \approx 1$ so that there will be a high probability of acceptance at high temperatures. This suggests that $T = -\Delta_{\text{avg}}/\ln P$ is a good choice for T_0.

In [9], a single cost function is used to evaluate the quality of a solution. However, in this paper, we use different cost funcitons in different stages. We know that one major term to decide the acceptance of a solution in simulated annealing is $e^{-\Delta C/T}$. Take the transition between the first stage and the second stage as an example, the difference of cost in the second stage is relatively larger than that in the first stage. That is, $-\Delta C_{\text{old}} >> -\Delta C_{\text{new}}$. Therefore, the acceptance ratio in the iterative-based process will drop suddenly and the simulated annealing process will end prematurely. For example, in the experiment of ami33 benchmark, when we encounter the stage transfer from half-perimeter estimation to L-shaped routing, the average ΔC_{old} is 46.5 and the average ΔC_{new} is 97.5. The acceptance ratio in the first stage is 0.809 and is 0.639 in the second stage.

Our approach is that not only we calculate the initial temperature of the first stage, we also determine the initial temperature of using L-shaped and Z-shaped routing to calculate cost with the same approach. When we reach the transition between first and second stages or between second and third stages, we compute the initial temperature of the second or the third stage T' by getting the new average value of the magnitude of change in cost per move, and using the current acceptance ratio, $P_{\text{curr\_acpt}}$, as a reference probability: $T' = -\Delta'_{\text{avg}}/\ln P_{\text{curr\_acpt}}$.

Although we use the current acceptance ratio to compute the new initial temperature during transition, the acceptance ratio will rise. The reason is that for the very first initial temperature estimation, we measure the term by random walks, but there exists very few random walk when transition occurs. We handle this by reducing the temperature much faster than the usual cooling ratio, until the acceptance ratio goes back on track. Experiments show that this approach is really helping the continuity of the simulated annealing process and the quality of performance.

4 Experimental Results

We have tested our approach on some MCNC building blocks examples. All experiments were carried out on a 300MHz Pentium II Intel Processor. In order to compare the performance of the interconnect planning approach with that

		Our Floorplanner				Floorplanner in [9]			Improvement	
Data	n	Time (sec)	Dead Space(%)	Total Vios(μm)	Max Vio(μm)	Dead Space(%)	Total Vios(μm)	Max Vio(μm)	Total Vios(%)	Max Vio(%)
apte	9	277.6	0.99	0.51	0.27	0.86	10.31	3.45	95	92
xerox	10	589.7	0.14	0.0	0.0	0.07	23.92	8.88	100	100
hp	11	141.2	0.30	0.76	0.68	0.61	15.16	3.34	95	80
ami33	33	2220	3.66	1.55	0.64	5.68	15.96	2.64	90	76
ami49	49	4041	2.93	7.68	2.75	3.21	38.62	6.75	80	59

Table 1: Performance of our approach on MCNC examples; compared with the method in [9]

of the conventional approach [9] in terms of routability, we perform pin assignment and use Z-shaped routing to route the nets in the final floorplans produced by the conventional approach. We can see the significant difference in Figure 3 and Figure 4 for ami49 benchmark, and the packing areas are about the same (Table 1). The dashed line are the grid lines and the thickness of line in the boundaries denotes the degree of overflow. For the five MCNC benchmarks, we observe that the new approach produces floorplans which are much more routable than the ones produced by the original floorplanner. In fact, the new method achieves a significant percentage of improvement in maximum violation and total violations without any area overhead.

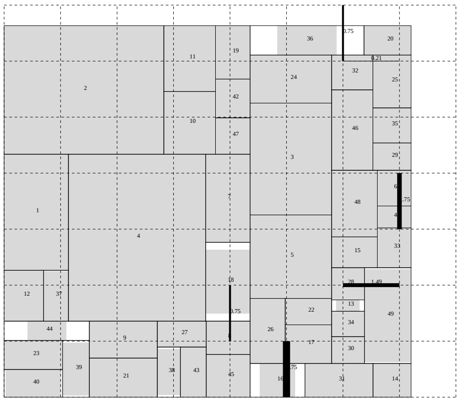

Figure 3: Result packing of ami49 using interconnect planning approach

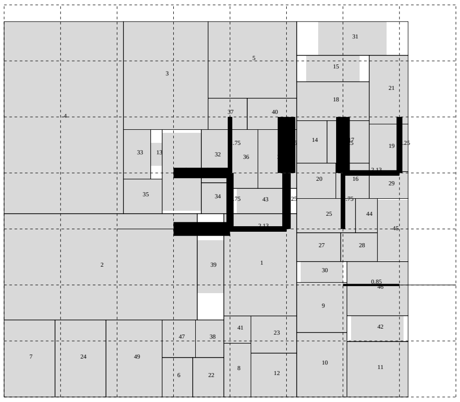

Figure 4: Result packing of ami49 using conventional approach

5 Conclusion

This paper presents a method to integrate floorplanning with interconnect planning. Simple-geometry routing is used to efficiently plan wires during module packing. A congestion cost is combined into the Wong-Liu simulated annealing based floorplanner, and a multi-stage simulated annealing strategy is used to effectively reduce the running time. We further develop a temperature adjustment approach to cope with the discontinuity resulting from switching cost functions. Experimental results show that our approach works well.

References

[1] J. Cong, L. He, K. Y. Khoo, C. K. Koh, and Z. Pan. "Interconnect Design for Deep Submicron ICs". In *Proceedings IEEE International Conference on Computer-Aided Design*, pages 478–485, 1997.

[2] H. Murata, K. Fujiyoushi, S. Nakatake, and Y. Kajitani. "Rectangle-Packing-Based Module Placement". In *Proceedings IEEE International Conference on Computer-Aided Design*, pages 472–479, 1995.

[3] S. Nakatake, K. Fujiyoushi, H. Murata, and Y. Kajitani. "Module Placement on BSG-Structure and IC Layout Applications". In *Proceedings IEEE International Conference on Computer-Aided Design*, pages 484–491, 1996.

[4] Ralph H.J.M. Otten and Robert K. Brayton. "Planning For Performance". In *Proceedings ACM/IEEE Design Automation Conference*, pages 122–127, 1998.

[5] R.H.J.M. Otten. "Automatic Floorplan Design". In *Proceedings ACM/IEEE Design Automation Conference*, pages 261–167, 1982.

[6] B. Preas and W. VanCleemput. "Placement Algorithms for Arbitrary Sahped Blocks". In *Proceedings ACM/IEEE Design Automation Conference*, pages 474–480, 1979.

[7] Naveed Sherwani. *"Algorithms for VLSI Physical Design Automation"*. Kluwer Academic Publishers, 1995.

[8] T. Tamanouchi, K. Tamakashi, and T. Kambe. "Hybrid Floorplanning Based on Partial Clustering and Module Restructuring". In *Proceedings IEEE International Conference on Computer-Aided Design*, pages 478–483, 1996.

[9] D.F. Wong and C.L. Liu. A new algorithm for floorplan design. In *Proceedings ACM/IEEE Design Automation Conference*, pages 101–107, 1986.