A Global Router for System-on-Package Targeting Layer and Crosstalk Minimization

Jacob R. Minz and Sung Kyu Lim
School of Electrical and Computer Engineering
Georgia Institute of technology, Atlanta, GA 30332-0250
{jrminz, limsk}@ece.gatech.edu

Abstract

In this paper, we formulate and solve a new 3D global routing problem for System-On-Package. Our divide-and-conquer algorithm provides an effective way to decompose the complex 3D problem into a set of 2D problems for simultaneous layer and crosstalk minimization.

I. Introduction

Three-dimensional SOP (System-On-Package) packaging [1] offers significant performance benefits over the traditional two-dimensional packaging such as PCB and MCM due to the electrical and mechanical properties arising from the new geometrical arrangement. The physical layout resource of SOP is multi-layer in nature–the top layer is mainly used to accommodate active components, the middle layers are mainly for passive components, and the I/O pins are located at the bottom of the SOP package. Routing layers are inserted in between these placement layers, and the placement layers can be used for local routing as well. Therefore, all layers are used for both placement and routing, and pins are now located at all layers rather than the top-most layer only as in PCB or MCM. Therefore, the existing routing tools for PCB or MCM can not be used directly for SOP routing. In this paper, we formulate and solve a new 3D global routing problem for SOP. Our divide-and-conquer algorithm provides an effective way to decompose the complex 3D problem into a set of 2D problems for simultaneous layer and crosstalk minimization. We show that the way we decompose the 3D problem has a huge impact not only on various design objectives such as layer and crosstalk, but also on the complexity of 2D sub-problems and the runtime needed to solve them.

II. Problem Formulation

The layer structure in multi-layer SOP contains one *I/O pin layer* through which various components can be connected to the external pins. The *placement layers* contain the blocks (such as ICs, embedded passives, opto-electric components, etc), which from the point of view of physical design are just rectangular blocks with pins along the boundary. The interval between two adjacent placement layers is called the *routing interval*. The routing interval contains a stack of *routing layers* sandwiched between *pin distribution layers*. These layers are actually X-Y routing layer pairs, so that the rectilinear partial net topologies may be assigned to them. We also allow routing to be done in the pin distribution layers as well as placement layers.

In the SOP model the nets are classified into two categories. The nets which have all their terminals in the same placement layer are called *i-nets*, while the ones having terminal in different placement layers are *x-nets*. The i-nets can be routed in the single routing interval or indeed within the placement layer itself. However, for high performance designs routing such nets in the routing interval immediately above or below the placement layer maybe desirable and even required. On the other hand, the x-nets may span more than one routing intervals.

We define the SOP global routing problem formally as follows: Given a 3D placement and netlist, generate a routing topology for each net n, assign n to a set of routing layers and assign all pins of n to legal locations. All conflicting nets are assigned to different routing layers while satisfying various capacity constraints. The objective is to minimize the total number of routing layers used, wirelength, and crosstalk.

III. SOP Global Routing Algorithm

A. Overview of 3D Global Routing

Our 3D router, illustrated in Figure 1, is divided into the following steps: (1) coarse pin distribution, (2) net distribution, (3) detailed pin distribution, (4) topology generation, (5) layer assignment, (6) channel assignment, and (7) pin assignment. The process of determining the location of entry/exit points of the nets for each routing interval is called the pin distribution step. The process of assigning nets to routing intervals is called the net distribution step. In the coarse pin distribution step, which is done before net distribution, we find a coarse location for the pins and use this information for the net distribution. After the net distribution, the detailed pin distribution step assigns finer location to all pins in each routing interval. A Steiner tree based routing topology [2] for each net is constructed and a layer pair is assigned to it during the topology generation step.

0-7803-667-1/04/$20.00 © 2004 IEEE

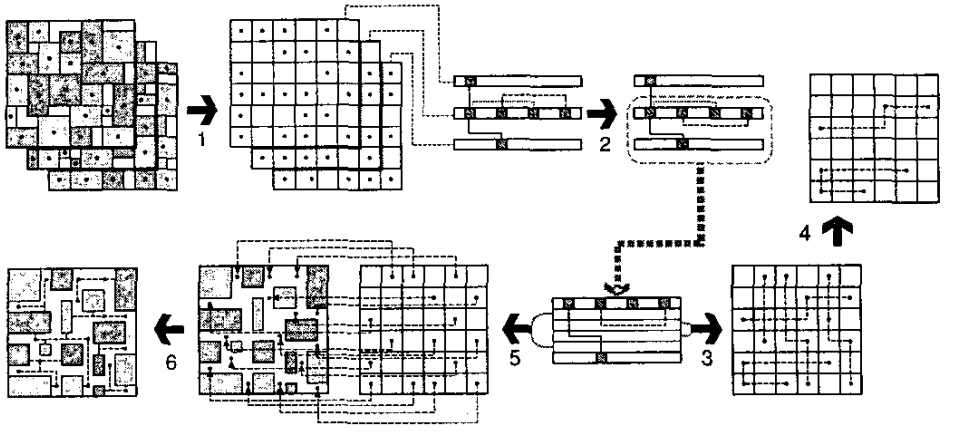

Fig. 1. Overview of the global routing process. 1=pin distribution, 2=net distribution, 3=topology generation, 4=layer assignment, 5=channel assignment, 6=pin assignment.

The conflict among the nets for routing resources is resolved and layer pairs are assigned during the layer assignment step. The channel assignment problem is to assign each pin in the pin distribution layers to a channel in the placement layers. The purpose of pin assignment is to finish connection between the pins in the routing channel and the pins along the block boundary. The pin and net distribution are performed while considering all routing intervals simultaneously. During topology generation and layer assignment, we visit each routing interval sequentially from bottom to top. During channel and pin assignment, we visit placement layers sequentially from bottom to top.

B. Coarse Pin Distribution

During 3D placement, we assume pins are located at the center of the modules (= soft modules) or at the boundary of the modules (= hard module). Thus, the pin location is highly localized and not evenly distributed. Since our plan is to use pin distribution layers and routing layers in combination to finish routing in each routing interval, one of the important steps is to evenly distribute pins in the pin distribution layer so that routing in the routing layers is done more evenly. This greatly helps reduce the number of routing layers used as well as crosstalk among nets. However, pin distribution cannot be accurate without knowing which nets are assigned to each routing interval. In addition, our net distribution needs to be based on newly distributed pin location for more accurate crosstalk measurement. Consequently, we need to iterate between pin distribution and net distribution until we converge to a good solution. We solve this issue with our three-stage effort: coarse pin distribution, net distribution, and detailed pin distribution.

During our coarse pin distribution step, we superimpose all placement layers onto a single 2D layer with $m \times n$ grid and perform pin distribution so that each pin is assigned to one of the slots. We extend the mincut-based global placement algorithm [3] for coarse pin distribution. In [3], hypergraph nodes are partitioned into $m \times n$ grid while minimizing the number of inter-partition connections (= cutsize) as well as their estimated wirelength. In our new heuristic algorithm, our cost function is based on (i) how far the new pin location is from the initial location, (ii) how evenly distributed the pins are, (iii) cutsize and wirelength, and (iv) how evenly distributed the inter-partition connections are. More specifically, we construct the initial $m \times n$ placement according to the initial pin location. We then compute the *move gain* for each pin so that it represents how much the cost is improved if moved to another partition. We then perform a sequence of pin moves based on the gain until the quality of the solution is not improved.

C. Net Distribution

Net distribution problem is to assign nets to routing intervals. Net distribution for some nets is straight forward–all nets having their pins in the lowest placement are assigned to the routing interval right above it. The nets having pins in the top-most placement are assigned the routing interval right below it. In case of an x-net, all routing intervals that this net spans are used. However, the net distribution of i-nets involves decision since they can be assigned to the routing interval right above or below. In our heuristic algorithm, the objective is to reduce crosstalk, where we use the amount of overlap among bounding boxes of the nets as a measure of crosstalk.

The net distribution problem is modeled with an undirected graph, where each net becomes a node and two nodes are connected via an edge if there is crosstalk between the two corresponding nets. The weight of the edges denotes the amount of crosstalk between the nets which is calculated by the amount of overlap among the bounding boxes of the nets. The problem can then be seen as a restricted graph partitioning problem, where each partition represents a routing interval. The nodes

that represent i-net can be partitioned into one of the two predetermined partitions (routing interval right above or below), whereas the nodes that represent x-net segments are fixed during the partitioning. Our heuristic algorithm is gain-based iterative improvement approach, where each movable node maintain stwo cost functions, *up_cost* and *down_cost* to represent how much the crosstalk is reduced if partitioned to routing interval right above or below. Once a node is moved to another routing interval, the cost of all its neighboring nodes are dynamically updated.

D. Detailed Pin Distribution

After coarse pin distribution and net distribution are finished, we know which set of nets are assigned to each routing interval as well as their (evenly distributed) entry/exit points in pin distribution layers. However, the coarse pin distribution is done based on the 2D grid that merged all multiple placement layers into one. The even pin distribution in this 2D grid offers a good enough reference points for net distribution. But, it does not consider even pin distribution in each individual routing interval. In addition, it is also possible that pin capacity for each partition in each routing interval may be violated. Therefore, it is possible that pin distribution in *each routing interval* is still not even and may violate pin capacity constraint. Therefore, the goal of detailed pin distribution is to address these problems in each routing interval so that the subsequent topology generation and layer assignment truly benefit from this even pin distribution.

Since the layer and crosstalk minimization are addressed during the prior steps, the major focus of our heuristic algorithm is on (i) how far the new location is from the original location obtained from coarse pin distribution, and (ii) the total wirelength. We use the same grid we used for coarse pin distribution. Our force-directed heuristic algorithm encourages all of the pins from the same net to be placed closer to the center of mass while minimizing the distance between the old and new pin location.

E. Layer Assignment

Given a set of entry/exit locations of the nets in a routing interval and their routing topologies in 2D grid, the layer assignment problem is to assign each net to a routing layer so that nets do not overlap and the number of routing layers used is minimized. We construct a Layer Constraint Graph (LCG) [4] from the net topology as follows: corresponding to each net we have a node in the LCG. Two nodes in the LCG have an edge between them if corresponding net segments of same orientation (horizontal or vertical) share at least one tile in the routing grid. In other words, an edge between the nodes denotes conflict. Then we use a node coloring algorithm to assign a color to the node such that no two nodes sharing an edge are assigned the same color. It is easily seen that the nodes having same color can be assigned to the same routing layer.

In our coloring heuristic algorithm, we first sort all nodes in LCG in a decreasing order of the number of their neighbors. Let $fin[n]$ denote the neighbors of n that are already colored. When we visit a node n from the sorted list, we compute the set of all colors that are used in $fin[n]$. In case there exists a color that is used for some node not in $fin[n]$, we assign this color to n. Otherwise, we introduce a new color and assign it to n. In spite of its simplicity, this greedy algorithm provides results that are very close to a tight lower bound on total number of colors used.

F. Channel Assignment

The pins in the routing interval have to be connected to their corresponding blocks in the placement layer using pin distribution layers and vias. Since vias can only be accommodated in the routing channels in the placement layer, we assign pins to routing channels while satisfying the channel capacity. The channel assignment result has a direct impact on the number of pin distribution layers used, so layer minimization is the primary goal. Our secondary objective is to reduce the number of bends which would necessitate the use of secondary vias. We assume a straight or L-shaped routing of nets to their assigned channel. This reasonable assumption simplifies the evaluation of the wirelength. We observed that the congestion of pin connections and wire crossings on a particular channel would increase the layer count. Our cost model for the problem captures these issues.

Our heuristic algorithm assigns pins to channels based on the cost of the assignment. When we select a pin to be assigned, we seek a channel with the best assignment cost. This cost is a combination of (i) the sum of L-distance between pin and channel, (ii) the channel density, and (iii) the bending penalty. In order for a channel assignment to be legal, the via capacity of each channel should not be violated. Our sequential pin assignment approach requires updates on channel capacity as well as congestion upon each assignment. Since the pins in each routing interval have been distributed evenly by our pin distribution steps, our sequential approach with no particular ordering of the pins does not degrade solution quality too much.

G. Pin Assignment

The final step in our proposed methodology finishes connection between pins in the channel and block boundaries. The pin assignment is done entirely in the routing channels of the placement layer. In case the boundary information is not available, we determine pin locations along the boundary as well. The channel pins are actually the entry/exit points to the routing interval. We model the placement layer with a FCG (Floorplan Connection Graph) [5]. The pin is now either a block node or channel node, and edge weight denotes the routing capacity of the channel. It is possible that the pins from the same multi-terminal

TABLE I

LAYER ASSIGNMENT, PIN DISTRIBUTION, AND CHANNEL ASSIGNMENT RESULTS

	layer assignment and pin distribution						channel assignment							
	RAND		CPD		DPD		best wirelength				combined cost			
ckt	lyr	wl	lyr	wl	lyr	wl	ly	wl	bnd	vl	ly	wl	bnd	vl
n10	3	1111	3	822	3	1016	5	6963	0	55	5	14174	34	0
n30	4	3889	6	1100	4	3393	6	11288	0	144	6	24726	122	0
n50	5	5725	4	807	5	4553	6	14826	3	282	6	32059	189	0
n100	6	8779	13	2999	7	6893	6	16430	3	413	7	36331	382	0
n200	12	18395	27	11424	11	14020	6	24078	1	845	8	61485	917	0
n300	13	20508	12	8627	13	16169	6	28469	3	1000	8	65189	1029	0

x-net are assigned to multiple channels in the same placement layer. In this case, we need to determine which of these pins to connect to a block in the placement layer if such a connection is desired. This process is called pin selection.

Our heuristic algorithm first performs pin selection, where the shortest distance between the pins–which are already assigned to a channel–and the destination block is used. We then perform maze routing, where a weighted shortest path in FCG is found for each channel-to-block connection. The edge weight in FCG represents the current usage of the channel, which is dynamically updated upon routing of each connection. Therefore, a detour is made for a connection that needs to go through a congested channel.

IV. EXPERIMENTAL RESULTS

We implemented our algorithms in C++/STL and ran experiments on a Dell Dimension 8800 Linux box. Our test cases are generated using our multi-layer SOP placer [6] on GSRC benchmark circuits [7]. The number of placement layers used is four. Our layer usage results are based on the boundary/channel capacity of 10 nets. The global routing trees are generated based on 10×10 grid. Due to a space limit, we could not include all results in this paper. Interested readers are referred to [8], [9], [10], [6] for more details.

In Table I we report the number of layers required to complete routing (lyr) and the total wirelengh (wl) for various pin distribution schemes. CPD is where no detailed pin distribution was carried out. The pins were assigned a location in the center of their coarse partition without legalization. RAND randomizes pin locations while respecting the coarse partitions of the pins. DPD is our wirelength oriented pin distribution Algorithm. We include CPD since the wirelength can be seen as a tight lower bound for other schemes. We used our crosstalk driven net distribution algorithm for all schemes. The results show that DPD achieves the lowest wirelength for all circuits, while also decreasing the number of layers. Other related experiments [9] show that our net distribution algorithm is effective in reducing crosstalk.

We present channel assignment results in Table I. Our baseline algorithm–not usable due to channel capacity violation–is performing wirelength minimization only, and via capacity violation was allowed. We use the following metric: number of layers, wirelength, bends, and number of pins violating channel via capacities. In channel assignment, we are close to the number of layers predicted by the best case. The increase in layers is due to increased routing density in the channels. Interestingly, the violations in the best case and the number of bends reported by our algorithm are very close, suggesting that violations were fixed by bending the interconnections.

V. CONCLUSION

In this paper, we presented a global routing algorithm for 3D packaging via System-On-Package (SOP). We provided an effective way to decompose the complex 3D problem into a set of 2D problems for simultaneous layer and crosstalk minimization. We are curently working on detailed routing for SOP.

REFERENCES

[1] R. Tummala and V. Madisetti, "System on chip or system on package," *IEEE Design & Test of Computers*, pp. 48–56, 1999.

[2] J. Cong, A. Kahng, and K. Leung, "Efficient algorithms for the minimum shortest path steiner arborescence problem with applications to vlsi physical," *IEEE Trans. on Computer-Aided Design*, pp. vol. 17, no.1, January 1998.

[3] J. Cong and S. K. Lim, "Physical planning with retiming," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, 2000, pp. 2–7.

[4] J. D. Cho, S. Raje, M. Sarrafzadeh, M. Sriram, and S. M. Kang, "Crosstalk-minimum layer assignment," in *IEEE Custom Integrated Circuits Conference*, 1993.

[5] J. Cong, "Pin assignment with global routing for general cell designs," *IEEE Trans. on Computer-Aided Design*, pp. 1401–1412, 1991.

[6] P. H. Shiu, R. Ravichandran, S. Easwar, and S. K. Lim, "Multi-layer floorplanning for reliable system-on-package," in *Proc. IEEE Int. Symp. on Circuits and Systems*, 2004.

[7] Http://www.cse.ucsc.edu/research/surf/GSRC/GSRCbench.html.

[8] J. Minz and S. K. Lim, "Layer assignment for system on packages," in *Proc. Asia and South Pacific Design Automation Conf.*, 2004.

[9] J. Minz, M. Pathak, and S. K. Lim, "Net and pin distribution for 3D package global routing," in *Proc. Design, Automation and Test in Europe*, 2004.

[10] R. Ravichandran, J. Minz, M. Pathak, S. Easwar, , and S. K. Lim, "Physical layout automation for system-on-packages," in *IEEE Electronic Components and Technology Conference*, 2004.