Global Routing Paradigm for System-on-Package

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

Abstract

The true potential of three dimensional System-On-Package (SOP) technology lies in its capability to integrate both active and passive components into a single high speed/density multi-layer packaging substrate. We propose a new interconnect-centric SOP global routing algorithm that handles arbitrary routing topologies and produces near optimal results. The contribution of this work is threefold: (i) modeling of the SOP routing resource, (ii) formulation of the new SOP global routing problem, and (iii) development of a fast and novel algorithm that considers the various design constraints unique to SOP. Our related experimental results demonstrate the effectiveness of our algorithm.

1. Introduction

The true potential of three dimensional SOP [1] technology lies in its capability to integrate both active components such as digital IC, analog ICs, memory modules, MEMS, and opto-electronic modules, and passive components such as capacitors, resistors, and inductors all into a single high speed/density multi-layer packaging substrate. Since both the active and passive components are integrated into the multi-layer substrate, SOP offers a highly advanced three-dimensional mixed-signal system integration environment. Three-dimensional SOP packaging 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. Thus, innovative ideas in the development of CAD tools for multi-layer SOP technology is crucial to fully exploit the potential of this new emerging technology.

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.

Global routing for PCB, IC and MCM is a well studied problem. SLICE [12] and V4R [13] are the current state-of-the-art global routers for MCM. Layer Assignment which is a part of global routing has also enjoyed immense interest in the research community. The interested reader is referred to [10]. The popular approaches for solving layer assignment is tile based [7] and graph based [10]. In this paper, we propose a new interconnect-centric global routing paradigm that handles arbitrary routing topologies and produces near optimal results. The contribution of this work is threefold: (i) modeling of the SOP routing resource, (ii) formulation of the new SOP global routing problem, and (iii) development of fast and novel algorithms that considers the various design constraints unique to SOP. Our related experimental results demonstrate the effectiveness of our approach.

The organization of the paper is as follows: Section 2 presents the formulation of the global routing problem for SOP. Section 3 presents our SOP global routing algorithm. Section 4 presents the experimental results. Section 5 concludes the paper.

2. Problem Formulation

The layer structure in SOP is different from PCB or MCM—it has multiple placement layers and routing layers. Figure 1 shows an illustration of SOP layer structure. It has one *I/O pin layer* through which various components can be connected to the external pins. The *placement layers* contain the blocks, which from the point of view of physical design is just a geometrical object with pins. In some cases where these blocks are a collection of cells, the pins may not be assigned and pin assignment needs to be done to determine their exact location. The interval between two placement layers is called the *routing interval*. The routing interval contains a stack of *signal 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 it. We also allow routing to be done in the pin distribution layers.

We model the placement layer in the SOP as a floor connection graph [2]. The routing layer is modeled as a uniform grid graph. These two kinds of graphs are connected through via edges. The use of grid graph

facilitates development of simple and efficient algorithms. The advantage of our graph-based routing resource model is that we can consider layer/pin assignment and global routing simultaneously. The formal description of our graph-based SOP routing resource model is given as follows:

SOP Routing Resource Model: the routing resource for the SOP is represented by a graph $RS=(V,E,C,L)$, where $V=(BN,CN,LN,RN)$ is a set of vertices, $E=(VE,CE,PE)$ is a set of edges, C: $E\rightarrow I$ is the edge capacity function, and L: $V\rightarrow(x,y,l)$ is the vertex placement function. *BN*, *CN*, *LN*, and *RN* respectively denote the set of block, channel, layer-switch, and routing nodes. *VE*, *RE*, and *PE* respectively denote the set of via, routing, and pin assignment edges.

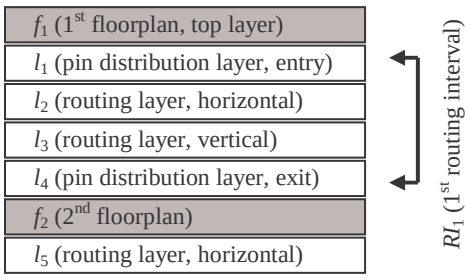

Fig. 1. Illustration of SOP layer structure.

We model the blocks in the floorplan as *Block Nodes* (BN). The nets can cross over to the adjacent routing layers only through the regions in the channel. The channel itself is represented by *Channel Nodes* (CN). The actual blocks form blockages for the nets, which cannot be routed through them. The nets can switch from floorplan layer to the routing layer only through designated regions which are represented as *Layer-switch Nodes* (LN) in the resource graph. The LN in this case is simply four corners of the blocks. They denote regions rather than points through which nets will traverse to adjacent routing intervals. The routing layers are represented by a grid graph, each node specifying a region in the layer and edges representing the adjacency between regions. These nodes are called *Routing Nodes* (RN). The edges between channel nodes and block nodes are called *Pin Assignment Edges* (PE). This makes it possible to perform pin assignment during global routing. The pin assignment capacity is the maximum number of pins which can be assigned towards a particular channel. The edges between layer switch node and routing node is defined as *Via Edges* (VE). The capacity of this edge is the maximum number of nets which can cross between two regions in the two layers. The via edges also exist between two adjacent routing layers (actually layer pairs). The edges between routing nodes are *Routing Edges* (RE). The routing edge capacity is the number of nets which can pass through the routing regions.

In the SOP model the nets are classified into two categories. The nets which have all their terminals in the same floorplan are called i-nets, while the ones having terminal in different floorplans will be referred to as 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. The span of a net $[l, h]$ is determined by the lowest floorplan f_l and the highest floorplan f_h containing pins of the net. If l and h are equal for a particular net, the net is i-net else the net is x-net. The nets encountered in the MCM model are i-nets and nets with span utmost one. Finally, we define the SOP global Routing problem formally as follows:

SOP Global Routing Problem: given a set of floorplans $F=\{f_1,f_2,\ldots,f_k\}$, netlist $N=\{n_1,n_2,\ldots,n_n\}$, and the routing resource graph $RS=(V,E,C,L)$, generate the routing topology $T(n)$ 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 the capacity constraints in *RS*. The objective is to minimize the total number of routing layers used, wirelength, and crosstalk.

3. SOP Global Routing Algorithm

3.1. Overview of the Algorithm

We propose a divide-and-conquer based methodology to solve the SOP global routing problem. The approach seeks to convert the 3-D nature of the problem into a set of 2-D problems. The steps in the routing process are as follows:

1. Coarse Pin Distribution
2. Net Distribution
3. Detailed Pin Distribution
4. Topology Generation
5. 2-D Layer Assignment
6. Channel Assignment
7. Pin Assignment

Since x-nets span multiple floorplan layers, we need to determine the location of entry to and exit from the routing interval. In the 2-D refinement of the problem we treat this location as pins. The routing of i-nets deserves special attention. In routing intervals, except the first and last ones, we have the choice of placing those i-nets in a routing interval either on top or bottom of the floorplan. The objective is to minimize crosstalk and congestion in

the routing interval. This step is called *net distribution*. The pin information of the nets is required for efficient net distribution, but net distribution decides the number of pins (and their locations) at each routing interval. We solve this by using the results of *Coarse Pin Distribution* for net distribution.

Pins in all routing interval are projected to a single 2-D area and partitioning evenly distributes them over it. But the pins in different routing intervals may not be evenly distributed locally. After net distribution we perform *Detailed Pin Distribution* on each routing interval to minimize the estimated wirelength. Figure 2 shows an illustration of Net Distribution and Detailed Pin Distribution. After this step we have all the information needed for global routing in the routing interval. The topology of each net is generated during *Topology Generation*, and *2-D Layer Assignment* assigns different layer to conflicting nets. The *Channel Assignment* problem is to assign each pin in the pin distribution layers to a channel in the floorplan layers such that the routing layers and interconnect costs are minimized. The objective is to facilitate an efficient pin distribution on pin distribution layer with only minimal additional costs. The purpose of *Pin Assignment* is to assign a location to the pin on the block boundary on the floorplan layer while minimizing the connections between the pin and its "peer" on the channel, which was found out in the previous step. The peer is location in the floorplan which connects the net to rest of its interconnect in the routing layers.

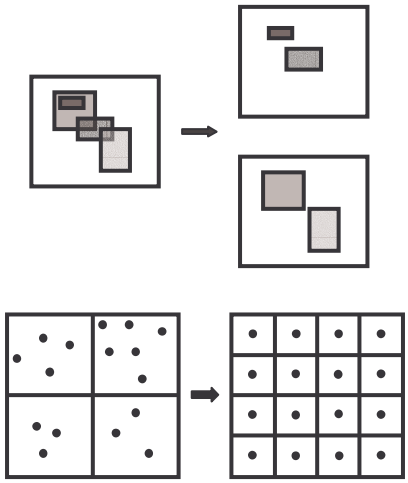

Fig. 2. Illustration of Net Distribution and Detailed Pin Distribution

Figure 3 gives the description of GROUTE algorithm for our SOP Global Routing problem. We define two types of nets, *current* and *propagated* nets. We visit the routing intervals sequentially from lowest to highest. The current nets are those which will be considered for layer assignment in the current routing interval. Current nets are the nets routed in the current routing interval. The current routing interval is the one currently processed by the algorithm. The propagated nets are nets "passed on" from this interval to be considered in the next routing interval. Net Distribution determines the number of current and propagated nets. For example x-nets will be propagated from its lowest level to the highest and will also be the current net for all the routing intervals in between. This is because we consider only a part (segment) of the x-net for routing in a particular routing interval (x-nets span multiple routing intervals). In the case of i-nets, the net is either current or propagated. The propagated nets form a subset of the nets to be routed in the next routing interval to be processed, since some i-nets may be included in the next routing interval.

Algorithm: SOP Global Routing (GROUTE)
Input: multi-layer SOP floorplan & netlist
Output: Routes of each net and the pin locations

1. Generate pins for all the nets.
2. Assign pins to a 2-D geometric partition. (Coarse Pin Distribution)
3. Initialize all nets with pins in the lowest floorplan as propagated nets.
4. Find the current nets for this routing interval and the propagated nets for the next routing interval. All propagated nets from previous interval will be current in this interval. Delete "old" i-nets and "finished" nets from propagated. (Net Distribution)
5. Find the entry/exit points for all the current nets in this routing interval. (Detailed Pin Distribution)
6. Generate net topologies for all the current nets on a parameterized grid graph. (Net Topology Generation)
7. Do 2-D Layer Assignment for the current nets
8. Goto step 2 unless this was the last of the ordered floorplans.
9. Do Channel Assignment for each routing intervals
10. Do Pin Assignment for each floorplan layer

Fig. 3. The Global Routing Algorithm in SOP

3.2. Summary of Our Previous Work

We have implemented the Coarse Pin Distribution and 2-D Layer Assignment [14] and use the existing RSA/G heuristic [6] among several others [3,4,5] for Topology Generation. Thus, the focus of this paper is Net Distribution and Detailed Pin Distribution.

Coarse Pin Distribution: In this step, we generate coarse locations for all pins of the nets in the routing interval. For the purpose of pin distribution we "flatten" the 3-D SOP

structure to 2-D and superimpose a AxB grid on it, were A and B are determined by the size of the circuit. We use GEO partitioning algorithm [11] to evenly distribute pins to all the partitions formed by this grid while keeping the wirelength minimum. Evenly distributing the pins among all partitions ensures efficient use of the routing resource provided by the single layer. The "coarse" location is the centre of the partition. After the partitioning the pins may not be uniformly distributed in the local routing interval. This partitioning algorithm is smart enough not to move the pins too far from their "initial" locations. The algorithm does iterative improvement until good results are obtained.

2-D Layer Assignment: We construct a Layer Constraint Graph (LCG) from the given global routing topology, where each node represents a net and 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. We use a fast node coloring heuristic algorithm to assign a color to the node such that no two nodes sharing an edge are assigned the same color. The algorithm is greedy in assigning colors but performs well and is fast. Close to optimal results are achieved because the heuristic tries to ensure that nodes with different colors have in fact an edge between them. The complexity of the algorithm is $O(n\log n)$, where n is the number of nets in the routing interval. The complexity is independent of the size of the grid used to compute the tree topologies. The capacity of the tiles determines the number of layers used. We use a simple formula to calculate this number (number of colors/capacity).

3.3. Net Distribution

As has been mentioned earlier in the paper, proper distribution of the nets is required to ensure end results are close to optimal. Net assignment for some nets is straight forward. When the floorplans are visited bottom to top, all nets having their pins in the lowest floorplan are assigned to the routing interval above it. The nets having pins in the top-most floorplan are assigned to the routing interval right below it. If the net is an x-net it is propagated through every layer until its topmost floorplan is reached. The net distribution of the i-nets is interesting. The objective of this step is to reduce crosstalk. We use the amount of overlap of bounding boxes of the nets as a measure of crosstalk. We also studied the case when no i-nets are propagated. We have developed a sophisticated heuristics wherein we partition the nets so that the overall crosstalk is minimized. Figure 4 gives the pseudocode of the algorithm used.

The net distribution problem is modeled as a graph with each i-net in the routing interval as node and the crosstalk interaction as edges. The weight of the edges denotes the amount of crosstalk between the nets. The crosstalk is calculated by the overlap of the bounding boxes of the net. The coarse pin distribution is used as the approximate location of the pins. It is assumed that nets in different interval are crosstalk shielded, which means no crosstalk exist between nets in different interval. The problem can then be seen as a restricted graph partitioning problem where some of the nodes can only go to one of two predetermined partitions. All nodes have two cost functions, *up_cost* and *down_cost* which are the costs of putting the net in either top or bottom routing interval. These costs are calculated based on the crosstalk induced by fixed nets and also the "movable" nets, which are nets assignable to the top or bottom routing interval. A probability of .5 is assigned to each movable nets. Once the nodes are moved to their routing intervals, the costs of all neighboring nodes are updated. In order to achieve better results iterative techniques similar to ones in [11] are used. The complexity of the algorithm is O(V+E) where V is number of nodes and E is the number of edges in the graph model.

Algorithm: Net Distribution (NETDIST)
Input: multi-layer SOP floorplan, netlist
Output: routes of each net and the pin locations.

1. Break x-nets into partial nets for each routing interval.
2. Get initial costs (crosstalk) for each net
3. Get up_cost and down_cost for each i-nets
4. Assign x-nets to the routing interval which minimizes cost for the net in a round-robin fashion.
5. Iterate with the updated cost for each nets until the solution cannot be improved further.

Fig. 4. Net Distribution Algorithm in SOP

3.4. Detailed Pin Distribution

This is an important step in the global routing flow of SOP. The purpose of this step is to legalize the location of the pins while respecting the coarse pin assignment and optimizing wirelength. The results of the coarse pin assignment are used for force-directed placement of the pins in the pin distribution layers. Since we did not consider the layer in which the pin was located in the coarse pin redistribution, it may be possible that the pins exceed the capacity of the partitions local to the routing interval. However our algorithm handles this by moving the pins from such location to the closest available position. The pins are placed in locations near the centre of the net. The pins furthest from its center of the net in coarse assignment, gets placed in the best location

(location nearest to the center) in the local partition. Figure 5 is the pseudocode used for detailed pin distribution.

The algorithm uses the "approximate" position of the nets as found by coarse pin distribution and the net distribution results to determine the initial location and routing interval of the pin. The position of the nets is stored as the grid location of the coarse pin distribution. The center of each net is calculated from this position of the nets. The displacement vector is calculated by taking the difference of the position of the center of net and the pin. A pair of numbers (a,b) such that $0 < a < 1$, $0 < b < 1$ is added to the position of the pins. The numbers reflect the scaled magnitude of the displacement vector. The variables a and b are less than 1 so that we can still keep track of the partitions of the pins. The pins in each routing interval are sorted according to their new positions. The pins are then sequentially assigned to grids previously determined.

Algorithm: Detailed Pin Distribution (DPD)
Input: coarse pin distribution & net distribution
Output: exact pin location in the routing layers

1. Calculate the center of net for each net.
2. Extract the pins for the routing interval from coarse partitioning and net distribution results.
3. Calculate the displacement vector of the pins from the center of the net.
4. Add a value (a,b) proportional to the displacement vector to the position (x,y) of the pin. (0<a <1, 0<b<1)
5. Sort the pins according to their new position values.
6. Assign the pins a unique location according to its rank in the list.

Fig. 5. Detailed Pin Distribution Algorithm

4. Experimental Results

We implemented our algorithm GROUTE in C++/STL and ran experiments on a Dell Dimension 8800 Linux box. Our test cases are generated using our multi-layer SOP floorplanner on GSRC benchmark circuits. The number of layer is fixed to four. Table 1 shows the characteristics of the benchmark circuits. Our layer usage results are based on the tile density w=10. The RSA/G-based global routing trees are generated based on 10×10 unless otherwise specified. Table 1 presents the characteristics of the benchmarks used in our study.

Table 1. Benchmark characteristics

ckts	blocks	pins	nets	i-nets	x-nets
n10	10	248	118	31	87
n30	30	723	349	97	252
n50	50	1050	485	76	409
n100	100	1873	885	189	696
n200	200	3599	1585	297	1288
n300	300	4358	1893	339	1554

The experiments were designed to study the effect of Net Distribution on the crosstalk of various circuits and the impact of Detailed Pin Distribution on the wirelength. The aim was to study how important parameters of global routing such as number of layers were affected with the combination of various schemes. All the benchmarks completed in less than one minute. So we do not explicitly report the runtimes.

Table 2. The impact of the Net Distribution (NETDIST) on crosstalk minimization.

ckt	RI1	RI2	RI3	total
n10	0	0	0	0
n30	0	0	0	0
n50	-0.57	5.12	-7.03	1.18
n100	6.94	-0.18	-9.94	1.10
n200	2.31	7.15	-11.5	0.92
n300	-1.35	8.15	-7.93	0.08

We present the results of our Net Distribution algorithm in Table 2. Since all circuits have 4 floorplan layers, we have 3 routing intervals for all of them. We show the % reduction of crosstalk after using NETDIST for net distribution for each routing interval. The base case is the random distribution of nets in the adjacent routing intervals. The results show that the algorithm achieve 1% improvement in crosstalk which is not insignificant because only 10% of the i-nets are actually eligible for consideration in net distribution because bulk of this nets are already fixed to their routing intervals, that is the top and the bottom floorplan and we start with a random distribution of nets and i-nets form a small percentage of the total number of nets.

Table 3. The impact of various Detailed Pin Distribution schemes using NETDIST algorithm for net distribution. The routing capacity *w=10* for each circuit.

ckt	CPD #layers	CPD WL	RAND #layers	RAND WL	DPD #layers	DPD WL
n10	3	22	3	1111	3	1016
n30	6	1100	4	3889	4	3393
n50	4	807	5	5725	5	4553
n100	13	2999	6	8779	7	6893
n200	27	11424	12	18395	11	14020
n300	12	8627	13	20508	13	16169

In Table 3 we report the number of layers required to complete routing and the total wirelengh (WL) for various Detailed Pin Distribution schemes such as CPD where no detailed pin distribution was carried out. The pins were assigned a location in the centre of their coarse partition without legalization. The algorithm RAND randomizes pin locations while respecting the coarse partitions of the pins. DPD is our wirelength oriented detailed Pin Distribution Algorithm. We include CPD since the wirelength can be seen as a tight lower bound for other schemes. We used NETDIST as the 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.

Table 4. The impact of various net distribution schemes with best algorithm for detailed pin distribution.

ckts	RUP layer	RUP WL	RD layer	RD WL	ND layer	ND WL
n10	3	1014	3	1016	3	1016
n30	4	3391	5	3398	4	3393
n50	5	4551	6	4557	5	4553
n100	7	6892	6	6901	7	6893
n200	11	14020	11	14020	11	14020
n300	13	16169	13	16186	13	16169

In Table 4, we study how the number of routing layers and wirelength (WL) change with various Net Distribution approaches. We used very simple heuristics such as assigning all i-nets to the routing interval above its floorplan (RUP) and below its floorplan (RD). We compare the results with the ones achieved by our algorithm NETDIST (ND). The detailed pin distribution approach used in all cases was DPD. We notice that ND preserves the number of layers, with only nominal differences in wirelength. One observation is that the wirelength is the average of RUP and RD in most cases.

The observations in our study can be summarized as follows.

1. NETDIST achieves significant reduction in crosstalk.
2. Using DPD we are able to reduce wirelength by 20% over random pin distribution, while not increasing the number of layers required and we are not too far from the lower bound projected by CPD. The number of layers is *reduced* in most cases.
3. NETDIST doesn't increase number of layers and wirelength significantly while reducing crosstalk.

5. Conclusions

In this paper we have introduced a new paradigm for global routing for SOP which looks into various aspects such as crosstalk, wirelength and layer minimization. We advocate a modular approach towards global routing because it facilitates handling of various objective functions efficiently. We have shown the impact of various algorithms such as coarse pin distribution, net distribution and detailed pin distribution on the overall global routing flow of SOP. Net Distribution is unique to SOP CAD and has a huge role to play in crosstalk and layer minimization. Our experimental results show that our algorithms efficiently handle the various objectives.

Our future work would include channel assignment and pin assignment of the global routing flow which we were unable to include in this paper due to time and space constraints. As emphasized in the paper, there is need for a fresh approach towards the SOP physical design to handle issues unique to this emergent technology. Our work is an attempt towards this direction.

References

[1] Rao Tummala and Vijay Madisetti, "System on Chip or System on Package?" IEEE Design & Test of Computers, pp 48-56, 1999.

[2] J. Cong, Pin assignment with global routing for general cell designs, Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on , Volume: 10 Issue: 11 , Nov 1991 , Page(s): 1401 -1412

[3] C. J. Alpert, T. C. Hu, J. H. Huang, A. B. Kahng, and D. Karger, Prim-Dijkstra tradeoffs for improved performance-driven routing tree design, IEEE Trans. on CAD 14(7), July 1995, pp.890-896

[4] M. Borah, R. M. Owens, and M. J. Irwin, An edge-based heuristic for Steiner routing, in IEEE Trans. on CAD, vol. 13, no. 12, Dec. 1994, pp.1563-1568,

[5] A. B. Kahng and G. Robins, "A New Class of Iterative Steiner Tree Heuristics with Good Performance", IEEE Trans. on CAD 11(7), July 1992, pp. 893-902

[6] J. Cong, A.B. Kahng, K Leung, Efficient Algorithms for the Minimum Shortest Path Steiner Arborescence Problem with Applications to VLSI Physical Design, IEEE Trans. on CAD, vol. 17, no.1, January 1998
[7] Ho, M. Sarrafzadeh, G. Vijayan, C. Wong, "Layer assignment for multi-chip modules", IEEE Transactions on Computer-Aided Design, Vol. 9, No. 12, December 1990, pp. 1272-1277
[8] M. Sriram and S. M. Kang, "Detailed Layer Assignment for MCM Routing," Proc. of ICCAD, pp. 386-389, 1992.
[9] J. Cho, M. Sarrafzadeh, M. Sriram, S. Kang, High Performance MCM Routing, IEEE Design & Test of Computers, 1993.
[10] Joy, D.A.; Ciesielski, M.J., "Layer assignment for printed circuit boards and integrated circuits", Proceedings of the IEEE, Vol 80, pp 311-331, Feb 1992.
[11] Jason Cong and Sung Kyu Lim, "Physical Planning with Retiming", IEEE International Conference on Computer Aided Design, p2-7, 2000.
[12] K.-Y. Khoo and J. Cong, "A fast multilayer general area router for MCM designs," *IEEE Trans. Circuits and Syst. II*, Nov. 1992, vol.39, pp. 841-51
[13] K.-Y. Khoo and J. Cong, "An efficient multilayer MCM router based on four-via routing," *IEEE Trans. Computer-Aided Design*, Oct. 1995, vol.14, pp. 1277-90
[14] Removed for blind review, "Layer Assignment for System on Packages", July, 2003.