Layer Assignment for Reliable System-on-Package

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—The routing environment for the new emerging mixed-signal System-on-Package (SOP) technology is more advanced than that of the conventional PCB or MCM technology – pins are located at all layers of SOP packaging substrate rather than the top-most layer only. We propose a new interconnect-centric layer assignment algorithm named LA-SOP 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 layer assignment problem, and (iii) development of a fast and novel algorithm that considers the various design constraints unique to SOP. We review various approaches for the PCB, IC and MCM algorithms and investigate their applicability to the SOP model. Our related experimental results demonstrate the effectiveness of our algorithm LA-SOP.

I. INTRODUCTION

The increasingly higher integration of transistors at an increasingly lower cost per transistor has resulted in System-On-Chip (SOC) paradigm. ASIC foundries and EDA vendors see a promising new business opportunity in SOC, which extends ASIC design from component to system level. On the other hand, the systems integration community and electronics packaging design vendors see the systems market as an extension of their current business, via the System-On-Package (SOP) paradigm [1]. As they see it, SOP will increase their importance in the product supply chain linking electronics packaging directly to product specification, early design, and ASIC design. The SOP paradigm extends the role of electronics packaging from the later stages of the manufacturing process (in the current chip-centered design universe) to the front-end and conceptual phases of the design process.

The true potential of SOP 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 [11] 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 floorplan layers, and the floorplan layers can be use for local routing as well. Therefore, all layers are used for both floorplan 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. The number of layers used in SOP has a huge impact on the final manufacturing cost, so layer assignment problem is crucial in SOP physical layout. The single significant difference between SOP and the existing technologies is in-layer floorplan of *both* blocks and nets. This makes the routing aspect for SOP very interesting for research.

Layer assignment for IC, PCB and MCM is a well studied problem. The interested reader is referred to [10]. The layer assignment problem can be categorized into constrained and unconstrained. In the constrained form of layer assignment the net topology is fixed and the nets have to be assigned to the layers. The net topologies are undetermined in the unconstrained layer assignment. The popular approaches for solving layer assignment is tile based [7] and graph based [10]. However, layer assignment for SOP differs from that of PCB, IC or MCM due to its own unique issues. In this paper, we propose a new interconnect-centric layer assignment algorithm named LA-SOP 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 layer assignment problem, and (iii) development of a fast and novel algorithm that considers the various design constraints unique to SOP. We review various approaches for the PCB, IC and MCM algorithms and investigate their applicability to the SOP model. Our related experimental results demonstrate the effectiveness of our algorithm LA-SOP.

The organization of the paper is as follows. Section 2 introduces the routing resource model and layer assignment problem for SOP. Section 3 discusses our SOP layer assignment. Section 4 presents experimental results. Section 5 concludes the paper.

II. SOP LAYER ASSIGNMENT PROBLEM

A. Layer Structure in SOP

This work was supported in part by NSF and Packaging Research Center at Georgia Institute of Technology.

The layer structure in SOP is different from PCB or MCM—it has multiple floorplan 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 *floorplan 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 floorplan 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. The formal description of the SOP layer structure is given as follows:

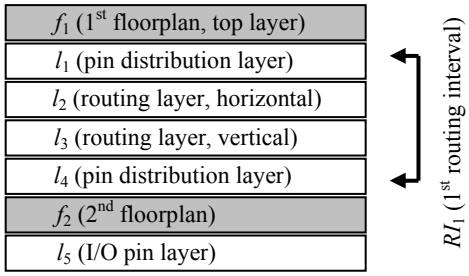

Figure 1. Illustration of SOP layer structure.

SOP Layer Structure: the layer structure for SOP is a set $LE=(F,RI)$, where $F=\{f_1,f_2,\ldots,f_k\}$ is a set of k ordered floorplan layers, and $RI=\{RI_1,RI_2,\ldots, RI_{k-1}\}$ is a set of k-1 routing intervals. Each floorplan $f_i=\{b_1(x_1,y_1,w_1,h_1),\ldots, b_j(x_j,y_j,w_j,h_j)\}$ is a set of blocks along with their location and dimension (width and height). The blocks are connected via a net $n=\{b_1,b_2,\ldots, b_j\}$, and $N=\{n_1,n_2,\ldots, n_n\}$ is a set of n nets. Each routing interval $RI_i=\{l_1,l_2,\ldots,l_j\}$ is a set of ordered routing layers (pin distribution and signal routing layers) in between two floorplan layers. The first and the last pin distribution layers are entry and exit layers, respectively. The total number of routing layers used in LE is $\sum|RI_i|$.

B. Routing Resource Model

We model the floorplan layer in the SOP as a floor connection graph [2]. The routing layer is modeled as a uniform grid[1] 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:

[1] For large SOPs memory will be a concern. An alternative idea for routing layer resource representation is as a collection of net entry/exit points. The routing will be done by area router, which will intuitively result in finer routing assignment, but at the expense of larger runtimes.

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, $E=(VE,RE,PE)$ 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.

The issues which must be taken into account in the model are the regions through which the nets can be routed and coarse location from where the nets can originate. To this end, 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 concepts are illustrated in figure 2, which shows the various views and types of nodes used in SOP.

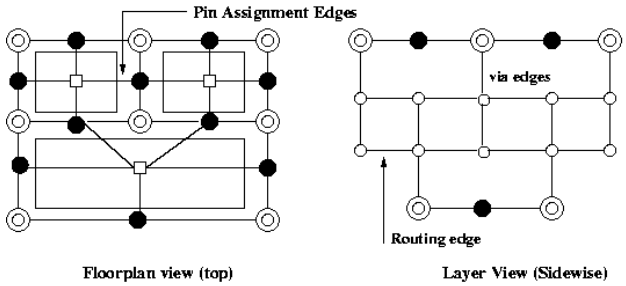

Figure 2. The floorplan and the layer view of the resource graph with node and edge types. Double circle, square, black dot, and white dot respectively denote the layer switch, block, channel, and routing nodes.

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.

C. SOP Layer Assignment Problem Formulation

Our routing resource model is designed for SOP global routing. However, the number of routing layers between the floorplan layers must be determined before performing global routing. SOP Layer Assignment (SLA) problem seeks an estimation of the smallest total number of layers between each pair of floorplan layers to complete the routing as well as the assignment of the nets to the layers. The objective is to use the routing layers optimally to connect all the nets in the netlist. We define the SLA problem formally as follows:

SOP Layer Assignment (SLA) Problem: given a set of floorplans $F=\{f_1,f_2,\ldots,f_k\}$, netlist $N=\{n_1,n_2,\ldots,n_n\}$, routing topology $T(n)$ for each net n in N, and the routing resource graph $RS=(V,E,C,L)$, assign each net to a set of routing layers such that the total number of routing layers used ($=\sum|RI_i|$) is minimized and all conflicting nets are assigned to a different routing layer while satisfying the capacity constraints C in RS.

D. Categorization of SOP Nets

In the SOP model the nets are classified into two categories. The nets which have all their terminals in the same floorplan layer are called i-nets, while the ones having terminal in different floorplan layers will be referred to as x-nets. The i-nets can be routed in the single routing interval or indeed within the floorplan layer itself. However, for high performance designs routing such nets in the routing interval immediately above or below the floorplan layer maybe desirable and even required, to avoid the detours in the floorplan layer. On the other hand, the x-nets may span more than one routing intervals. The only case where one routing interval may suffice is when the terminals of the net are located in either of the floorplan layers immediately above or below the routing interval.

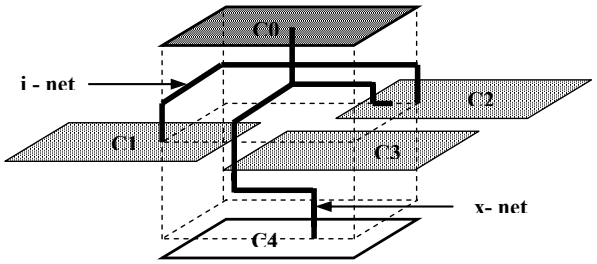

Figure 3. The routes of i-nets and x-nets.

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 difference between the two is the span of the net. Greater the span of the net, more number of routing intervals (and floorplan layers) the nets must go, which leads to increased demands in the actual number of layers required per routing intervals. The nets encountered in the MCM model are i-nets and nets with span utmost one. The SOP algorithms must handle x-nets in all levels of physical design. For example, one of the objectives of the SOP floorplanner should be to reduce the span of the nets while assigning blocks to different floorplan layers. In Figure 3, sample routes for these two types of nets are shown.

III. SOP LAYER ASSIGNMENT ALGORITHM

A. Overview of the Algorithm

We propose a divide-and-conquer based algorithm named LA-SOP to solve the SOP layer assignment problem. LA-SOP seeks to convert the 3-D layer assignment problem into a set of 2-D problems. The steps in LA-SOP are as follows:

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

It is important to note the difference between the real pins and the entry/exit points. The real pins are assigned to the block boundaries, whereas the number and locations of the entry and the exit points of the net to the routing interval must also be determined. By definition, the pins distributed just below the floorplan layer are called *entry points*, whereas the pins assigned just above the floorplan layer are called *exit points*. For congestion issues, we may choose to project and distribute the pins in the routing interval. The x-nets will have both entry and exit points for the routing interval, whereas the i-nets will have only one of the two. The number of entry/exit points can exceed the actual number of pins of the net, since the entry and exit point of an x-net to adjacent and non-adjacent routing intervals must be determined. This process of determining the location of entry/exit points for each routing interval is called *Pin Distribution.* The assignment of nets to the routing intervals affects the number of layers that will be used. In case of a x-net with $[l,h]$ span, all routing intervals between h^{th} and l^{th} floorplan will be used. In case of an i-net, a decision must be made to assign the net either below or above the floorplan, except in the case of the lowest and highest floorplan in the set. This process of assigning nets to routing intervals is called *Net Distribution*.

The pin distribution and net distribution depend on each other. Net distribution cannot be accurate unless the locations of the pins are known for the nets that are distributed. Likewise pin distribution cannot be done efficiently without the final assignment of nets to the routing intervals. We solve this problem by arranging a compromise between the two steps. 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. This is more accurate than assigning a "approximate" location of the pins to the center of the blocks. After the net distribution, *Detailed Pin Distribution* step assigns "finer" location to all pins.

After the three steps just described, we have an estimate of the number of nets and their entry/exit points for each routing interval. The layer assignment problem can now be solved on the grid graph for each routing interval independently. A

routing topology for the net is constructed and a layer pair is assigned to it. In case we have many nets, there will be conflicts among the nets for routing resources which must be resolved and layer pairs assigned to the different nets. This procedure is called *2-D Layer Assignment*. The problem is 2-D because all computations use the 2-D grid graph for resource and layer estimations.

Algorithm: LA-SOP
Input: multi-layer SOP floorplan, netlist Output: total # of layers, net assignment to layers
1. Initialize all nets with pins in the lowest floorplan as propagated nets. (coarse pin distribution) 2. 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) 3. Find the entry/exit points for all the current nets in this routing interval. (detailed pin distribution) 4. Generate net topologies for all the current nets on a parameterized grid graph. 5. Solve 2-D LA problem for the current nets. 6. Goto step 2 unless this was the last of the ordered floorplans.

Figure 4. LA-SOP algorithm for layer assignment in SOP

B. SOP Layer Assignment Algorithm

Figure 4 shows the description of LA-SOP algorithm for our SOP layer assignment problem. The nets may be provided as connections between blocks in which case we may have to do pin assignment on the blocks. Pin distribution can be done irrespective of whether or not pins have been assigned on the block because all that is relevant to the problem at hand is the entry/exit points to the routing interval. For the purpose of the algorithm we define two types of nets, *current* and *propagated* nets. We visit the floorplan layers from the lowest to the highest level, in other words we consider 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. 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. It's only a subset because some i-nets from the next routing interval may also be included for routing. The decision of which and how many i-nets to include for routing in a routing interval, eventually determines the quality of the solution. Net distribution is important because putting more nets in a congested routing interval may increase the total number of layers and decrease effective utilization of routing resource. Figure 4 outlines the overall algorithm for SLA.

The algorithm is designed to handle arbitrary tree topologies. The traditional predetermined topologies may underestimate the number of layers by being very optimistic about net conflicts. We examined a variety of candidates for representative topologies [3, 4, 5]. For our implementation, we have used RSA/G heuristic [6] to generate the net topologies at each routing interval, since it is fast and simple. The minimum arborescence is a good representative for the topology of a net in a high performance design.

Coarse Pin Distribution: In this step, we generate coarse locations for all pins of the net in the routing interval. For the purpose of pin distribution we "flatten" the 3-D SOP structure to 2-D and superimpose a 4x4 grid on it. We use GEO partitioning algorithm [12] 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. This partitioning algorithm is smart enough not to move the pins from their "initial" locations, that is given an initial partitioning the algorithms does iterative improvement until good results are obtained.

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 floorplan layers are visited bottom to top, all nets having their pins in the lower floorplan layer are assigned to the routing interval above it. If the net is an x-net it is propagated through every layer until its topmost floorplan layer is reached. The net distribution of the i-nets is interesting. We have implemented a very simple heuristic where in we evenly distribute the nets to the routing intervals. This means we keep half the i-nets as current and propagate the other half to the next routing interval. We also studied the case when no i-nets are propagated. Smarter heuristic needs to consider the degree of congestion caused by the nets in the adjoining interval and assign it to an interval with lower congestion. The concept of net interference can be used which can consider various metrics to do net distribution.

Detailed Pin Distribution: The approach that we follow in our tool is a very simple heuristic. We evenly assign the nets originating from a particular block to its four corners in the floorplan layer and project the "assigned" pins to the routing interval as entry/exit points. More sophisticated heuristics can be developed. For instance, the points could be distributed in such a way to provide good "coverage" of the routing plane while ensuring that these points are not too far

away from their actual pins. Other issue to be taken in to consideration is capacity constrained pin distribution wherein more than certain number of pins cannot be assigned to a particular region.

C. 2-D Layer Assignment

The problem of 2-D Layer Assignment (2-D LA) of nets (maybe multi-terminal) is to minimize v (the number of plane-pairs) given a set of entry/exit locations of the net and their corresponding routing topologies in the plane grid. No restrictions are placed on the configuration of the topologies. Hitherto, the problems of the same sort in the literature place a restriction on the shape of the two terminal nets to either L shaped and multi-terminal nets to be comb-shaped [7,8].

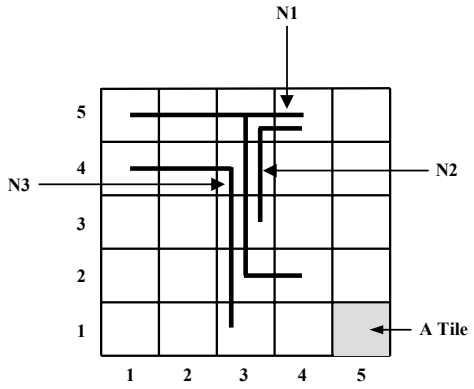

Figure 5. A 5×5 routing grid

In the 2-D layout environment let $t(i,j)$ denote the tile located at row i and column j. The number of nets passing horizontally through $t(i,j)$ where $1 \leq i, j \leq R$ in a $R \times R$ grid is denoted as $h(i,j)$, and $v(i,j)$ is the number of nets passing vertically through the tile. We define $d=\max(h(i,j),v(i,j))$ to be the density of the problem. If w is the capacity of the tile, clearly $v \geq \text{celing}(d/w)$. This equation provides the theoretical lower bound on the number of layer-pairs required.

We propose an approximation algorithm to solve an arbitrary instance of 2-DLA. The approach used here is similar to one outlined in [9]. We construct a Layer Constraint Graph (LCG) 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. However, the metric for conflict can be generalized to include constraints due to other effects such as cross-talks etc. 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 layer pair. Hereafter, when we say the node has been colored k, we mean that the net corresponding to that node has been assigned layer pair P_k. 2-DLA is NP-complete since, node coloring problem is. Figure 5 gives an example of the net topologies in the planar grid. The corresponding LCG has three nodes N1 to N3 and they are connected via 3-clique. An interesting thing to note is that d (the density of the problem) is the lower bound of the maximum LCG clique size. The number of colors needed for coloring the LCG will actually depend on the maximum clique size. For the example in figure 5, the graph will require 3 colors and the corresponding nets will be assigned to three layer pairs if w=1. If w>1, the net colored k can be assigned to layer q such that $k=qw+r$. In our research, we concentrate our efforts on lowering the total number of colors used for the nodes without explicitly considering routing capacities in our algorithms. This is justified since the routing capacities are largely dependent on specific design rules for different process technologies.

We now present the coloring heuristic used to assign layer pairs to the nets in the current routing interval. After the trees have been generated a LCG is constructed as mentioned in the earlier section. We use a fast and efficient heuristic to assign color to the nodes. The algorithm is outlined below in Figure 6.

Algorithm: Node Coloring
Input: Layer Constraint Graph Output: number of colors and node colors
1. Sort the nodes by the number of its adjacent edges. Assign fin[n] = 0 for all node in LCG. Set min_color=0. 2. Remove the highest node from the list. Let the node be u. 3. Find out the set of colors used by the colored neighbors of u. 4. If the node u cannot be colored by the lowest available color, increase the minimum number of colors and assign this color to this node. 5. For all nodes v which are neighbors of u, set fin[u] = fin[u] + 1. 6. If there are nodes remaining in the list goto 2 else stop.

Figure 6. Algorithm for Node Coloring

The algorithm colors the nodes of LCG in an order determined by the node degrees. The variable *fin*[n] tracks the neighbors of node n, which are already colored. If the neighbors are assigned a set of colors equal to the current number of colors available, the node is colored by a unique color. This node cannot be colored by the available colors since it conflicts with nodes which have been assigned all the available colors. The algorithm is greedy in assigning colors but performs well and is fast. Close to lower bounds 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.

IV. EXPERIMENTAL RESULTS

We implemented our algorithm LA-SOP in C++/STL and ran 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. The number of x-nets and i-nets has great impact on the number of layer-pairs used in the routing interval. The x-nets are expensive for the layer assignment. The benchmark circuits have much higher number of x-nets than i-nets leading to a high demand of routing layer pairs. For each benchmark circuit, we measure the horizontal and vertical densities and the number of colors and layers required. Since the routing capacities are dependent on the process technologies (the width of the nets, minimum separation between the nets, other design rules), the colors provide a good comparison metric for our algorithms. 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. 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

Table 2 shows LA-SOP results under grid size 10×10 with w=10. We report the maximum horizontal and vertical density, number of net segments, color, and layers used for each routing interval. The ratio is between the actual color used and the theoretical lower bound (= max(hx,vx)). The number of net segments is higher than actual number of nets due to the x-nets that need segmentation in each routing interval. We use the i-net distribution discussed in Section III.B, but the partitioning-based coarse pin distribution is not used for results shown in Table 2. We note from the results that LA-SOP obtains solutions that are very close to theoretical minimum. In almost all cases except for the smallest circuit, LA-SOP gives the ratio of 1.00. LA-SOP is very fast—it finishes all circuits less than a minute. In fact, all of our main algorithms in LA-SOP are fast: partitioning-based coarse-pin assignment, RSA/G based global routing, net distribution algorithm, and 2D coloring heuristic.

The theoretical minimum for the 2-DLA problem is not necessarily the indicator for the theoretical lower bound of the SLA problem. The sum of the theoretical lower bounds for the entire decomposed 2-DLA problem is in fact not an actual tight lower bound for the SLA. However, we could measure the efficiency of our algorithm with respect to the solution for 2-DLA. It can be seen from the results that the number of layers assigned were actually optimal in a number of cases and very near to optimal in others, within the routing intervals. However, this optimality is only local to each routing intervals. As we mentioned earlier the routing density is a lower bound on the maximum clique size of the LCG. In cases where routing density is equal to the clique in the LCG, the results obtained are optimal and the number of layer pairs needed is the theoretical lower bound for a particular routing interval. The fact that the routing demand is near to the maximum clique size can be attributed to the choice of the routing topology. The results show that the RSA/G heuristic is a very good candidate tree representation in this respect. At this point, we can only claim that we have a near-optimal solution for our 2-DLA subproblem in our SLA framework. The steps prior to this one in our methodology strives to further decrease the lower bounds for 2-DL. Even better results can be obtained by employing smarter net and pin distribution algorithms which will decrease the routing demands and net conflicts and hence the number of routing layers required, helping to achieve results closer to global optimum.

Table 2. Layer assignment results under grid size 10×10 with w=10. We report the maximum horizontal and vertical density, number of net segments, color, and layers for each routing interval. The ratio is between the actual color used and the theoretical lower bound (= max(hx,vx)).

ckt	RI	hx	vx	nets	cols	lyr	ratio
n10	1	38	26	87	46	5	1.21
	2	22	26	55	27	3	1.03
	3	12	13	40	16	2	1.23
n30	1	52	100	262	100	10	1
	2	61	56	170	62	7	1.01
	3	16	36	119	36	4	1
n50	1	39	93	344	93	10	1
	2	65	83	303	83	9	1
	3	37	31	198	37	4	1
n100	1	110	110	562	110	11	1
	2	60	160	524	160	16	1
	3	55	92	368	93	10	1.01
n200	1	249	177	1003	249	25	1
	2	158	226	1003	226	23	1
	3	80	155	702	155	16	1
n300	1	257	172	1121	257	26	1
	2	165	163	1177	165	17	1
	3	117	87	873	117	12	1

Table 3 compares various setting for LA-SOP. n×n refers to the grid size for tree topology generation. "i-net" refers to our i-net propagation heuristic, and "part" refers to our partitioning-based coarse pin distribution. The grid sizes for the coarse pin distribution are scaled up with the circuit size—it ranges from 2×2 for the smallest to 8×8 for the largest circuit. We turn on and off our i-net propagation and coarse pin distribution heuristics and measure the impact on the layer usage. We also change the grid size for tree topology generation and observe the impact. We summarize our observations as follows:

1. the impact of i-net propagation (column 2 *vs* 3) is visible for n50, where the number of layers used

decreased by 7. However, the impact is not significant in overall.

2. when we increase the grid size for tree topology generation from 10×10 to 20×20 (column 3 *vs* 4), we observe improvement on layer usage for several circuits. This is expected since the finer routing grids lead to the less net conflicts—pin locations can be assigned more accurately in the proximity of its actual location. However this accuracy is achieved at the cost of runtime.
3. the impact of coarse pin distribution is significant. The number of layers used is reduced up to 90% for the biggest circuit n300 and by 70% on the average. The pin locations are assumed to be at the center of the blocks before coarse pin distribution, which causes significant amount of congestion in local zones. This in turn increases the number of layers used. Thus, pin distribution helps reduce the congestion and promote better utilization of routing resource.

Table 3. Layer usage under various settings. *n*×*n* refers to the grid size for tree topology generation. "i-net" refers to our i-net propagation heuristic, and "part" refers to our partitioning-based coarse pin distribution.

ckts	10×10 no i-net no part	10×10 i-net no part	20×20 i-net no part	10×10 i-net part
n10	9	10	9	3
n30	20	21	18	4
n50	30	23	21	3
n100	36	37	36	5
n200	60	64	46	15
n300	55	55	44	7

V. CONCLUSIONS

In this paper, we have emphasized the need for new techniques and models to solve the new and emerging SOP technology. The physical design of SOPs is significantly different from the traditional way in which the physical design is done for the existing technologies such as PCB and MCM. Although conventional approaches can still be used to solve some of the problems, there is huge scope for achieving higher efficiency by independently investigating the issues involved. We have introduced the SOP routing resource model which can be used for global routing. We have formulated the SOP Layer Assignment problem and proposed a methodology to solve it. We have obtained very encouraging results by using a smart coloring heuristic. The comparisons of routing densities and LCG gave us some insights on when the theoretical lower bounds on number of layer pairs can be achieved.

There are several ways in which we can improve the methodology proposed in the paper. We are working on the net distribution problem, which considers crosstalk and congestion of the routing intervals. Another important issue we'll be investigating is legalization, which is legal placement of the pins respecting design rules and via constraints. The complete package will also contain pin assignment routines to assign pins to the block boundaries in the floorplan layer. It would also be interesting to study the unconstrained version of the problem and its comparison with the solution of constrained version of layer assignment.

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] Rao Tummala, "Multichip packaging-a tutorial", Proceedings of the IEEE, pp 1924 -1941, 1992.

[12] Jason Cong and Sung Kyu Lim, "Physical Planning with Retiming", IEEE International Conference on Computer Aided Design, p2-7, 2000.