Thermal and Crosstalk-Aware Physical Design For 3D System-On-Package

Jacob Minz, Eric Wong, and Sung Kyu Lim
School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, GA 30332, USA
{jrminz, ewong, limsk}@ece.gatech.edu

Abstract

3D packaging via System-On-Package (SOP) is a viable alternative to System-On-Chip (SOC) to meet the rigorous requirements of today's mixed signal system integration. In this article, we present the first physical layout algorithm for 3D SOP that performs thermal-aware 3D placement and crosstalk-aware 3D global routing. Existing approaches consider thermal distribution and crosstalk issues as an afterthought, which may require expensive cooling scheme and additional routing layers. Our goal is to overcome this problem with our thermal and crosstalk-aware 3D layout tools. The traditional design objectives such as performance, area, wirelength, and via costs are considered simultaneously to ensure high quality results. Related experimental results demonstrate the effectiveness of our approach.

1. Introduction

Semiconductor industry is beginning to question the viability of System-On-Chip (SOC) approach due to its low-yield and high-cost problem. Recently, 3D packaging via System-On-Package (SOP) [1], [2], [3] has been proposed as an alternative solution to meet the rigorous requirements of today's mixed signal system integration.[1] The SOP is about 3D integration of multiple functions in a miniaturized package achieved by thin film embedding. The 3D SOP concept optimizes ICs for transistors and the package for integration of digital, RF, optical, sensor and others. It accomplishes this by both build-up SOP, similar to IC fabrication, and by stacked SOP, similar to parallel board fabrication. The uniqueness of 3D SOP is in the highly integrated or embedded RF, optical or digital functional blocks, and sensors, in contrast to stacked ICs and stacked package as illustrated in Figure 1.

Thermal issues cannot be ignored anymore in high performance 3D packages due to higher power densities and other issues. High temperatures not only require more advanced heat sinks, they also degrade circuit performance. Interconnect delay increases with temperature, which degrades circuit timing. If timing deteriorates enough, logic faults can occur. Hence thermal issues must be considered early-on in the design process. Moreover, due to the scaling down of device geometry in deep-submicron technologies, the crosstalk noise between adjacent nets has become a major concern in high performance packaging design. Increased coupling noise can cause signal delays, logic hazards and even malfunctioning of the designs. Thus controlling the level of crosstalk noise in 3D packages chip is an important task for the designers.

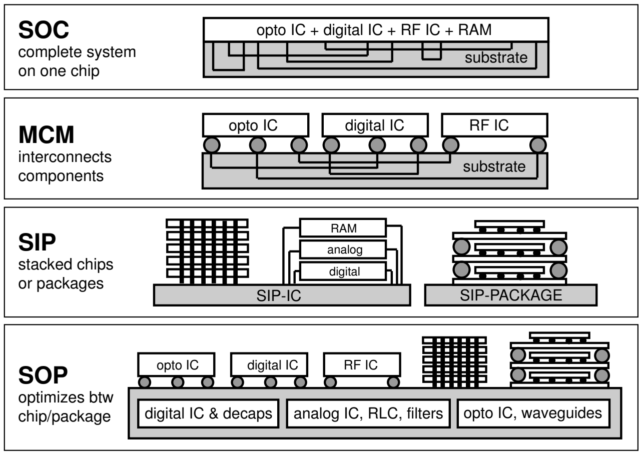

Fig. 1. Comparison among SOC (System-On-Chip), MCM (Multi-Chip Module), SIP (System-In-Package), and SOP (System-On-Package).

However, existing approaches consider these issues as an afterthought, which may require expensive cooling schemes and more routing layers. In addition, many time-consuming iterations are required between full-length thermal/crosstalk simulation and manual layout repair until we converge to a satisfactory result. We note that the placement of modules in 3D SOP design has huge impact on thermal distribution while the routing of the signal nets has direct impact on crosstalk. Therefore, the goal of this article is to present the first physical layout algorithm for 3D SOP that combines thermal-aware 3D placement and crosstalk-aware 3D global routing algorithm. The traditional design objectives such as performance, area, wirelength, and via costs are done simultaneously to ensure high quality results. Related experimental results demonstrate the effectiveness of our approach.

Recent work on thermal-aware physical design algorithms include [4], [5], [6], [7], [8], [9]. Recently, physical design algorithms for 3D System-On-Package designs have been proposed including 3D placement [10], [11] and 3D routing [12], [13]. The remainder of this paper is organized as follows. We provide the problem formulation in Section 2. Our thermal-aware SOP placement algorithm is presented in Section 3. Section 4 presents our crosstalk-aware SOP global routing algorithm. Experimental results are presented in Section 5, and we conclude in Section 6.

[1] A special issue on SOP (vol. 27, issue 2, May 2004) provides a comprehensive survey of the state-of-the-art in SOP technology.

0-7803-8906-9/05/$20.00 ©2005 IEEE

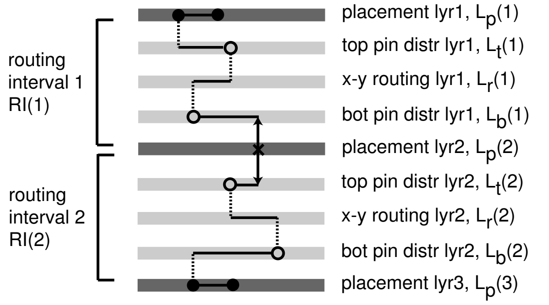

Fig. 2. Illustration of the layer structure and routing resource in SOP. The block and white dots respectively denote the original and redistributed pins. The "x" denotes a feed-through pin for an x-net to pass through a placement layer using a routing channel. The solid, dotted, and arrowed lines respectively denote signal wires, vias, and feed-through vias.

2. Problem Formulation

A. SOP Layer Structure

The layer structure in multi-layer SOP is illustrated in Figure 2. The *placement layers*[2] 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*. A 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. The pin distribution layers in each routing interval are used to evenly distribute pins from the nets that are assigned to this interval. Then these evenly distributed pins are connected using the routing layer pairs. Each placement layer consists of a pair of x-y routing layers, so routing is permitted. A *feed-through via* is used to connect two pin distribution layers from different routing intervals. Thus, the routing channels in each placement layer are used for two purposes: (i) accommodate feed-through vias, and (ii) perform local routing, where limited number of intra-layer connections are made.

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 terminals in different placement layers are *x-nets*. The i-nets can be routed in a single routing interval or indeed within the placement layer itself. On the other hand, the x-nets may span more than one routing interval.

B. SOP Placement and Routing Problem

The following are given as the input to our 3D SOP placement problem: (i) a set of blocks $B = \{B_1, B_2, \cdots, B_m\}$ that represent the various active and passive components in the given SOP design, (ii) width, height, and maximum switching currents for each block, (iii) a netlist $NL = \{n_1, n_2, \cdots, n_k\}$ that specifies how the blocks are connected via electrical wires, and (iv) K, the number of placement layers in the 3D packaging structure.

For each net n from a given netlist NL, let wl_n denote the wirelength of n. The wirelength wl_n is the sum of Manhattan distance in x, y, and z directions, where the z direction is the height of the associated vias. Let A^{tot} denote the final footprint area of the 3D placement. Let T^{max} denote the maximum temperature of the substrate. The goal of the *Thermal-aware SOP Placement Problem* is to find the location of each block in the placement layers such that the following cost function is minimized:

$$w_1 \cdot A^{tot} + w_2 \cdot \sum_{n \in NL} wl_n + w_3 \cdot T^{max} \qquad (1)$$

For each net n from a given netlist NL, let xt_n and v_n respectively denote the amount of crosstalk and via associated with n. Let $cl(n, m)$ denote the coupling length between n and m. We define xt_n as follows:

$$xt_n = \sum_{m \in NL,\ m \neq n} \frac{cl(n, m)}{|z(n) - z(m)|} \qquad (2)$$

where $z(n)$ denote the routing layer that contains net n. The formal definition of *Crosstalk-driven SOP Global Routing Problem* is 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 wire/via capacity constraints. Let L^{tot} denote the total number of routing layers. The objective is to minimize the following cost function:

$$w_4 \cdot L^{tot} + \sum_{n \in NL} (w_5 \cdot xt_n + w_6 \cdot wl_n + w_7 \cdot v_n) \qquad (3)$$

3. Thermal-Aware SOP Placement Algorithm

A. Overview of the Algorithm

Simulated Annealing is a very popular approach for module placement due to its high quality solutions and flexibility in handling various constraints. We extend the existing 2D Sequence Pair scheme [14] to represent our 3D module placement solutions. Simulated Annealing procedure starts with an initial multi-layer placement along with its cost in terms of area, wirelength, and maximum temperature. We then make random perturbation (move) to the initial solution to generate a new 3D placement solution and measure its cost. The algorithm does a one time set-up of the thermal matrices. These matrices are used during incremental temperature calculations to evaluate the thermal cost. The thermal modeling and evaluation is explained in section 3-B. If the new cost is lower than the old one, the solution is accepted; otherwise the new solution is accepted based on some probability that is dependent on temperature of the annealing schedule. We examine a pre-determined number of candidate solutions at each temperature. The temperature is decreased exponentially, and the annealing process terminates when the freezing temperature is reached.

[2] We use placement layer and device layer interchangeably.

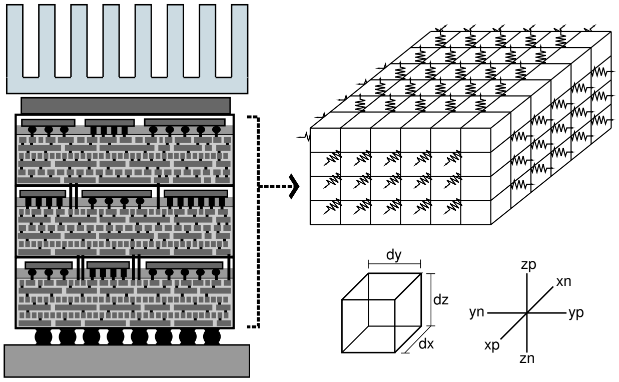

Fig. 3. Top: 3D grid of a SOP for thermal modeling

B. 3D Thermal Analysis

The linearized differential equation $(k \cdot \bigtriangledown^2 T + P = 0)$ for steady state heat flow was the basis of our thermal model, as described in [4]. In the equation, k is the thermal conductivity, T is the temperature, and P is the power density of heat sources. The chip is divided into a 3D grid to apply a finite difference approximation to the differential equation. (See Figure 3).

Figure 4 shows our thermal equations, where i is the current node and xn, xp, yn, yp, zn, zp are the nodes to the negative or positive x,y,z direction of i, k_{xn} is the thermal conductivity between node i and node xn, t is the temperature of the node, p_i the power density of node i, and dx, dy, dz are the dimensions of a node. When Equation (5) is written out for all nodes, they can be expressed as the following matrix equation: $G \cdot T = P$, where G is the thermal conductance matrix ($G_{i,j}$ is the thermal conductance between node i and node j), T is the temperature profile vector (T_i is the temperature of node i), and P is the power profile vector (P_i is the power dissipation of node i).

If the nodes are enumerated in such a way that all the active nodes take up the top m rows and the passive nodes take up the bottom n rows then the matrix equation takes the following form:

$$\begin{bmatrix} G_a & G_c^T \\ G_c & G_p \end{bmatrix} \begin{bmatrix} T_a \\ T_p \end{bmatrix} = \begin{bmatrix} P_a \\ 0 \end{bmatrix}$$

The passive nodes can be eliminated by defining matrix Y such that: $Y = G_a - G_c^T \cdot G_p^{-1} \cdot G_c$, and $Y \cdot T_a = P_a$. Inverting Y gives matrix R which is the thermal resistance matrix between the active nodes: $R = Y^{-1}$, $R \cdot P_a = T_a$. The temperature of all the active nodes can now be calculated from the power profile using a single matrix-vector multiplication.

Assuming that the thermal conductivity of device blocks are similar (they are mostly silicon), swapping the location device blocks would not change the thermal resistance matrix R. This means that matrix R only needs to be computed once in the beginning. To calculate the temperature profile of a new block configuration, the power profile P needs to be updated and then multiplied by R. Alternatively, a change in power profile ΔP can be defined. Multiplying R and ΔP will give change in temperature profile ΔT. Adding ΔT to the old temperature profile will give the new temperature profile. These equations summarize the two methods: $T_{new} = R \cdot P_{new}$, $\Delta P = P_{new} - P_{old}$, $\Delta T = R \cdot \Delta P$, $T_{new} = T_{old} + \Delta T$. Swapping two blocks usually has a small effect on the power profile, so ΔP should be sparse. This reduces the number of multiplications used by the second method at the expense of doing extra additions and subtractions.

4. Crosstalk-Aware SOP Global Routing Algorithm

A. Overview of 3D Global Routing

Our 3D router, illustrated in Figure 5, 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 for each net is constructed and a layer pair is assigned to it during the topology generation step. 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

$$k_{xn}\frac{dydz}{dx}(t_{xn}-t_i)-k_{xp}\frac{dydz}{dx}(t_i-t_{xp})+k_{yn}\frac{dxdz}{dy}(t_{yn}-t_i)-k_{yp}\frac{dxdz}{dy}(t_i-t_{yp})+k_{zn}\frac{dxdy}{dz}(t_{zn}-t_i)- k_{zp}\frac{dxdy}{dz}(t_i-t_{zp})+dxdydz\cdot p_i=0 \quad (4)$$

$$t_{xn}(-k_{xn}\frac{dydz}{dx})+t_{xp}(-k_{xp}\frac{dydz}{dx})+t_{yp}(-k_{yp}\frac{dxdz}{dy})+t_{yn}(-k_{yp}\frac{dxdz}{dy})+t_{zn}(-k_{zn}\frac{dxdy}{dz})+t_{zn}(-k_{zn}\frac{dxdy}{dz})+ t_i(k_{xn}\frac{dydz}{dx}+k_{xp}\frac{dydz}{dx}+k_{yp}\frac{dxdz}{dy}+k_{yp}\frac{dxdz}{dy}+k_{zn}\frac{dxdy}{dz}+k_{zn}\frac{dxdy}{dz})=dxdydz\cdot p_i \quad (5)$$

Fig. 4. The Thermal Equations

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 [15] for coarse pin distribution. In [15], 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. Topology Generation

During this step, we visit each routing interval and generate Steiner tree for all nets distributed in this routing interval based on the 2D grid used for the prior detailed pin distribution step. We use an existing MSSA (Minimum Shortest Path Steiner Arborescence) algorithm [16] to construct the routing tree so that the paths from the source to all sink pins are always the shortest. This kind of tree guarantees the minimum source to

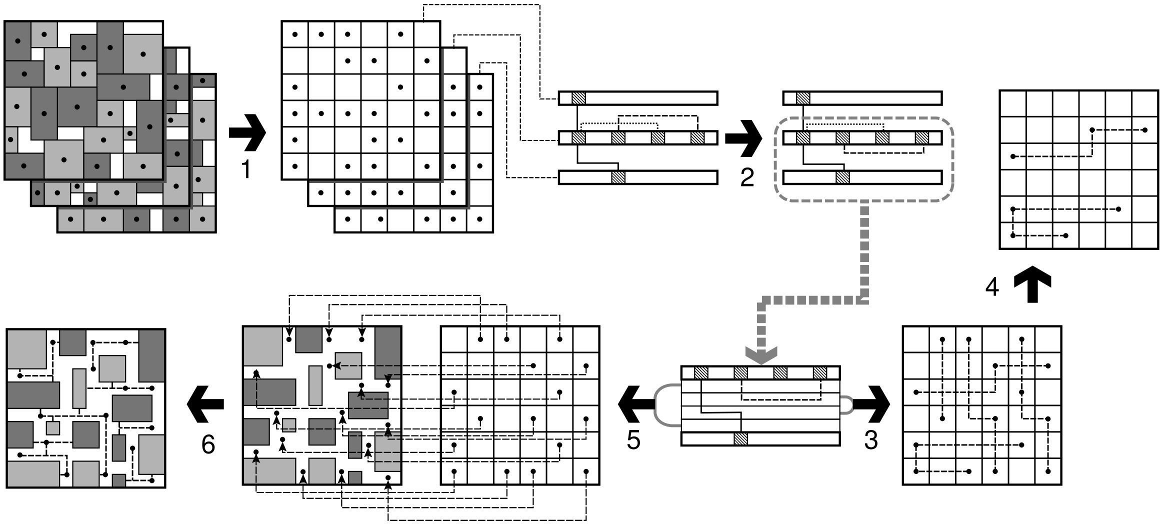

Fig. 5. Overview of the global routing process. 1=pin distribution, 2=net distribution, 3=topology generation, 4=layer assignment, 5=channel assignment, 6=pin assignment.

sink delay and provide a reasonable wiring length (= total wire capacitance).

F. 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) [17] 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.

G. 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.

H. 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) [18]. 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 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.

5. Experimental Results

We implemented our algorithms in C++/STL and ran our experiments on Linux Beowulf clusters. We tested our algorithms with two sets of benchmarks. The first set is from the standard GSRC floorplan circuits. The second set, named the GT benchmark, was synthesized from the IBM circuits [19], where we use our multi-level partitioner [20] to divide the gate-level netlist into multiple blocks. The GSRC benchmarks are small to medium-sized in terms of both the number of blocks and nets.[3] The GT benchmarks contain medium to large number of blocks with *dense* netlists. We report the average runtime for each benchmark measured in seconds.

A. SOP Placement Results

For our thermal analysis we used a grid size of 10 x 10 x 7 in the x, y and z direction. The number of active layers was four with three passive layers in between them. The conductivity of the substrate was chosen to be that of silicon ($0.1W/mmC$). The conductivity of the sides of the package was fixed at $0.01W/mmC$. The heat sink was assumed to be at the top of the package with a conductivity of $0.5W/mmC$ and the conductivity of the bottom of the package was chosen to be $0.2W/mmC$. The power density of the blocks varied from $10W/mm^2$ to $300W/mm^2$ depending on the switching current demands. The switching current demands were generated using a formula using a random number and the size of the block.

We note from Table I that our thermal-driven floorplanning achieved a 21% improvement on maximum temperature results over the baseline (= area/wirelength-driven) at the cost of 76% area and 28% wirelength increase. The runtime is comparable in this case. The area/wirelength optimized floorplan with a thermal constraint of 90C and a tolerance of 10C improved the area and wirelength results at the cost of temperature increase. The CPU times of the algorithm depend on the number of candidate solutions evaluated. The times reported in the table directly reflect the subset of the configuration space spanned by the algorithm. The constraint version ran much faster, making a small number of moves because fewer moves were accepted.

TABLE II

SOP PIN REDISTRIBUTION RESULTS USING OUR COARSE AND DETAILED PIN DISTRIBUTION ALGORITHMS. WE REPORT THE WIRELENGTH BETWEEN THE ORIGINAL AND THE NEW LOCATION (DW), TOTAL WIRELENGTH (WL), CROSSTALK (XT) AND TOTAL NUMBER OF LAYER PAIRS USED (LYR) IN THE ROUTING LAYERS.

	DPD				CPD+DPD			
ckt	dw	wl	xt	lyr	dw	wl	xt	lyr
n30	0.10	0.03	0	5	0.15	0.04	0	5
n50	0.15	0.04	5	7	0.24	0.05	5	6
n100	0.26	0.07	10	10	0.41	0.07	10	8
n200	0.51	0.16	51	15	0.80	0.15	52	14
n300	0.69	0.21	219	14	1.17	0.21	220	16
gt50	1.50	0.41	92	38	2.19	0.38	90	33
gt100	3.01	0.84	441	73	4.34	0.78	446	57
gt300	3.94	1.19	2200	74	6.14	1.07	2186	66
gt1000	7.45	2.21	13365	89	11.46	2.06	13496	87
gt1500	8.76	2.67	14296	87	13.38	2.33	14366	94
TIME	579				588			

B. SOP Routing Results

In Table II we compare pin redistribution results. Under the DPD columns, we perform DPD (= detailed pin distribution) only, where we skip CPD (= coarse pin distribution) and assign all i-nets to the routing interval below for net distribution. Under the CPD+DPD, we perform CPD using the algorithm, assign all i-nets to the routing interval below for net distribution, and perform DPD. DPD serves as our baseline, where CPD+DPD respectively demonstrate the impact of our coarse pin distribution. In all cases, we perform detailed pin distribution to legalize the pin location, i.e., remove overlaps among the pins. We use the following metrics to evaluate our solutions: wirelength between the original and the new location (dw), total wirelength (wl), crosstalk (xt) and total number of layer pairs used (lyr) in the routing layers. The displacement (dw) and wirelength (wl) results are scaled by 10^6, and the time reported is the average runtime among the GSRC/GT circuits. From the comparison between DPD and CPD+DPD, we note that the displacement result (dw) increases by an average of 50%. However, CPD lowers the total wirelength (wl) consistently by 10% on average and the number of layers (lyr) by 10% on average.

In table III we show our topology generation (RSA/G) and layer assignment (LAYER) results. We used the technology parameters for 0.13μ process for Elmore delay computation. Specifically, the driver resistance of $29.4k\Omega$, input capacitance of $0.050fF$, unit-length resistance of $0.82\Omega/\mu m$ and unit-length capacitance of $0.24fF/\mu m$ are used. We report the total wirelength (wl), Elmore delay of the nets with maximum sink delay (dly), and the lower bound and the actual number of layers used for the top-most routing interval. In general, GSRC benchmarks have bigger delay than GT benchmarks due to the larger average wirelength. Our layer assignment algorithm is able to achieve results very close to the lower bounds. For the GT circuits, the layer assignment results are within 10% of the lower bound. For the GSRC circuits, we were able to achieve the results equal to the lower bound.

Our channel assignment results are shown in Table IV. The

[3] The GT benchmark circuits are available for download at our website: www.gtcad.gatech.edu.

TABLE I

AREA/WIRE-DRIVEN VS THERMAL-DRIVEN VS AREA/WIRE-DRIVEN UNDER THERMAL CONSTRAINT.

ckts		area/wire-driven			thermal-driven			thermal-constraint		
name	size	area	wire	temp	area	wire	temp	area	wire	temp
n50	50	22107	2.66	87.25	37710	3.59	68.90	22020	2.97	81.82
n50b	50	25509	2.51	75.53	38121	3.42	63.25	26768	2.74	73.70
n50c	50	21233	2.35	80.66	31406	3.01	65.11	21942	2.86	71.71
n100	100	31551	6.66	86.56	49376	8.41	69.83	37142	7.50	83.78
n100b	100	28017	4.83	104.86	50151	6.95	78.23	27867	4.83	104.86
n100c	100	32845	6.21	90.94	52908	8.43	73.87	37561	7.43	87.67
n200	200	56012	1.71	96.46	107736	2.45	76.21	55802	1.72	96.47
n200b	200	55624	1.82	97.77	103160	2.56	76.25	67203	2.14	96.11
n200c	200	54259	1.69	100.96	113549	2.43	76.16	54174	1.69	100.96
n300	300	84610	2.86	100.16	131052	3.88	86.65	84263	2.86	100.12
gt100	100	19156	1.32	71.00	47466	2.04	52.34	20743	1.68	70.90
gt300	300	23898	1.96	93.21	52809	2.80	72.19	24487	2.26	90.43
gt400	400	27017	2.81	114.05	36290	3.70	89.25	26665	2.81	114.05
gt500	500	31665	3.03	99.78	54125	3.85	80.34	32926	3.68	99.48
gt600	600	47541	6.08	115.62	77795	7.65	84.60	47268	6.08	115.62
RATIO		1.00	1.00	1.00	1.76	1.28	0.79	1.05	1.10	0.98
TIME			563			512			191	

TABLE III

TOPOLOGY GENERATION AND LAYER ASSIGNMENT RESULTS. WE REPORT THE TOTAL WIRELENGTH (WL), ELMORE DELAY OF THE NETS WITH MAXIMUM SINK DELAY (DLY), AND THE LOWER BOUND (LOW) AND THE ACTUAL NUMBER (LYR) OF LAYERS USED.

	RSA/G		LAYER	
ckt	wl	dly	low	lyr
n30	391.8	2.779	2	2
n50	429.3	3.045	3	3
n100	384.7	2.728	7	7
n200	400.5	2.841	7	7
n300	506.7	3.599	9	9
gt50	202.7	1.436	16	16
gt100	232.3	1.645	29	29
gt300	259.2	1.837	27	31
gt1000	291.4	2.064	32	37
gt1500	289.5	2.054	40	49
TIME	90		150	

TABLE IV

SOP CHANNEL ASSIGNMENT RESULTS. WE REPORT THE LAYER USAGE, WIRELENGTH, AND VIA FOR THE BASELINE (WIRELENGTH MINIMIZATION ONLY) AND MULTI-OBJECTIVE ALGORITHMS.

	wl-only			lyr+wl+via		
ckt	lyr	wl	via	lyr	wl	via
n30	9	0.023	103	8	0.033	132
n50	10	0.035	167	8	0.039	239
n100	12	0.039	317	9	0.049	484
n200	13	0.058	603	10	0.081	1244
n300	12	0.066	686	12	0.086	1314
gt50	11	0.710	7033	9	0.833	11538
gt100	18	1.320	11770	14	1.503	21249
gt300	26	1.925	17938	17	2.087	34418
gt1000	67	4.048	43029	29	4.448	77237
gt1500	74	4.517	48299	35	5.033	87506
TIME		194			210	

baseline case is where we optimize the wirelength only. We then compare it to our multi-objective channel assignment algorithm that simultaneously optimizes the wirelength, via, and layer usage. Our comparison indicates that the number of layers is consistently and significantly reduced especially for the bigger GT benchmarks, where an average improvement of 48% is observed. In case of the second largest benchmark gt1000, we achieved 57% improvement. This saving on the layer usage comes at the cost of increase in wirelength and vias. The average increase in wirelength is 12% and 30% for GSRC and GT benchmarks, respectively. The average increase in via usage is 63% and 79% for GSRC and GT benchmarks, respectively. We noted that the channel assignment result is very sensitive to the weighting constants among the objectives used in our cost function. This indicates that the solution space of the channel assignment problem offers many useful tradeoff points.

Table V reports our local routing results. We report the wirelength, maximum and average routing demand as well as the standard deviation. Our baseline is the local routing optimized for wirelength only. We then compare it against our multi-objective local routing algorithm that simultaneously optimizes wirelength and routing demands. In both cases, the same pin demand constraint is imposed. We note that the improvement of our multi-objective algorithm over the baseline is significant, especially for GSRC circuits—the routing demands were reduced by 33% on average while the wirelength increased by only 10%. In addition, we reduced the routing demands for the GT benchmarks by 41% on average, with wirelength increase by 20%. In our biggest benchmarks (gt1500), our routing demand reduction is the largest (53%), which comes with the maximum increase in wirelength (23%). This again indicates that the local routing result is very sensitive to the weighting constants among the objectives used in our cost function. The lower standard deviation of our multi-objective algorithm indicates that the routing demand is more evenly distributed (= lower congestion) compared to the wirelength-only case.

6. Conclusions and Ongoing Works

In this article, we presented the first physical layout algorithm for 3D SOP designs that includes thermal-aware 3D placement and crosstalk-aware 3D routing algorithm. We

TABLE V

SOP local routing results for the baseline (wirelength minimization only) and multi-objective algorithms. We report the wirelength (wl), maximum (max) and average (ave) routing demand as well as the standard deviation (dev).

	wl-only				wl+rd			
ckt	wl	max	avg	dev	wl	max	avg	dev
n30	0.080	47	7.85	8.82	0.082	38	7.69	7.63
n50	0.129	66	10.29	11.61	0.133	64	10.10	9.63
n100	0.247	183	14.83	19.21	0.276	112	14.77	14.73
n200	0.512	305	22.25	28.68	0.593	144	22.82	19.45
n300	0.786	297	22.50	29.59	0.920	145	23.23	19.28
gt50	1.417	3423	296.39	456.73	1.684	2089	311.30	356.83
gt100	2.803	4564	343.74	611.19	3.338	3335	367.08	465.06
gt300	5.069	6504	327.32	596.98	6.197	3629	358.67	435.62
gt1000	8.162	6492	263.65	500.58	9.698	3730	304.08	382.46
gt1500	9.666	9221	256.60	506.93	11.899	4313	294.06	378.35
TIME	547				573			

extended the thermal models to 3D and used them to guide our 3D module placement. Our routing process is divided into pin redistribution, topology generation, layer assignment, channel assignment, and local routing steps. Our major objective is to reduce the amount of crosstalk and layers while satisfying various constraints on routing resource. Our ongoing work includes SOP detailed routing.

Acknowledgment

This research is partially supported by the grants from National Science Foundation and the Packaging Research Center at Georgia Institute of Technology.

References

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

[2] R. Tummala, “SOP: What is it and why? a new microsystem-integration technology paradigm-moore’s law for system integration of miniaturized convergent systems of the next decade,” *IEEE Transactions on Advanced Packaging*, 2004.

[3] R. T. et al, “The SOP for miniaturized, mixed-signal computing, communication, and consumer systems of the next decade,” *IEEE Transactions on Advanced Packaging*, 2004.

[4] C. Tsai and S. Kang, “Cell-level placement for improving substrate thermal distribution,” *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 2000.

[5] C. N. Chu and D. F. Wong, “A matrix synthesis approach to thermal placement,” *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 1998.

[6] B. Goplen and S. Sapatnekar, “Efficient thermal placement of standard cells in 3D ICs using a force directed approach,” in *Proc. IEEE Int. Conf. on Computer-Aided Design*, 2003.

[7] S. Das, A. Chandrakasan, and R. Reif, “Timing, energy, and thermal performance of three-dimensional integrated circuits,” in *Proc. Great Lakes Symposum on VLSI*, 2004.

[8] J. Cong, J. Wei, and Y. Zhang, “A thermal-driven floorplanning algorithm for 3D ICs,” in *Proc. IEEE Int. Conf. on Computer-Aided Design*, 2004.

[9] K. Skadron, M. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, “Temperature-aware microarchitecture,” in *Proc. IEEE Int. Conf. on Computer Architecture*, 2003, pp. 2–13.

[10] P. 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.

[11] 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.

[12] J. Minz and S. K. Lim, “A global router for system-on-package targeting layer and crosstalk minimization,” in *Proc. IEEE Electrical Performance of Electronic Packaging*, 2004.

[13] 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.

[14] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani, “Rectangle packing based module placement,” in *Proc. IEEE Int. Conf. on Computer-Aided Design*, 1995, pp. 472–479.

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

[16] J. Cong, A. B. Kahng, and K.-S. Leung, “Efficient algorithms for the minimum shortest path steiner arborescence problem with applications to VLSI physical design,” *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, pp. 24–39, 1999.

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

[18] J. Cong, “Pin assignment with global routing for general cell design,” *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1401–1412, 1991.

[19] C. J. Alpert, “The ISPD98 circuit benchmark suite,” in *Proc. Int. Symp. on Physical Design*, 1998, pp. 80–85.

[20] J. Cong and S. K. Lim, “Edge separability based circuit clustering with application to multi-level circuit partitioning,” *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, no. 3, pp. 346–357, 2004.