

Pad Assignment for Die-Stacking System-in-Package Design*

Yu-Chen Lin[†], Wai-Kei Mak[†], Chris Chu[‡], and Ting-Chi Wang[†]

[†]Department of Computer Science, National Tsing Hua University, Hsinchu 300, Taiwan

[‡]Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011, USA
kenter78@gmail.com, wkmak@cs.nthu.edu.tw,
cnchu@iastate.edu, tcwang@cs.nthu.edu.tw

ABSTRACT

Wire bonding is the most popular method to connect signals between dies in System-in-Package (SiP) design nowadays. Pad assignment, which assigns inter-die signals to die pads so as to facilitate wire bonding, is an important physical design problem for SiP design because the quality of a pad assignment solution affects both the cost and performance of a SiP design. In this paper, we study a pad assignment problem, which prohibits the generation of illegal crossings and aims to minimize the total signal wirelength, for die-stacking SiP design. We first consider a variety of special cases and present a minimum-cost maximum-flow based approach to optimally solve them in polynomial time. We then describe an approach, which uses a modified left edge algorithm and an integer linear programming technique, to solve the general case. Encouraging experimental results are shown to support our approaches.

Categories and Subject Descriptors

B.7.2 [Integrated Circuits]: Design Aids

General Terms

Algorithms, Experimentation

Keywords

System-in-Package, Pad Assignment, Wire Bonding

1. INTRODUCTION

Comparing System-in-Package (SiP) with System-on-Chip (SoC), SiP is a more economical option than SoC in many consumer electronic products because of the high process complexity and cost associated with SoC. On the other hand, compared with traditional system integration where multiple dies with separate packaging are mounted on a PCB, SiP

*This work was partially supported by National Science Council of Taiwan under Grants NSC 98-2220-E-007-011, NSC 98-2220-E-007-012, and NSC 98-2220-E-007-013.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ICCAD '09, November 2-5, 2009, San Jose, California, USA
Copyright 2009 ACM 978-1-60558-800-1/09/11 ...\$10.00.

has the advantages of smaller size, lower cost, higher performance, lower power, and shorter time to market. So, today SiP is already widely used in consumer electronics such as cell phones. Currently, SiP design is mostly done by ad hoc methods and the quality of a design is heavily dependent on the designers' expertise. Tool support specific to SiP design is still inadequate [5, 6, 9].

Wire bonding is the most popular method to connect signals between different dies in SiP nowadays. As shown in Figure 1, a die-stacking SiP design using wire bonding has the following properties: (1) Dies with different sizes are stacked and pad signals are connected by bonding wires. (2) Die pads can only be located on die boundaries.

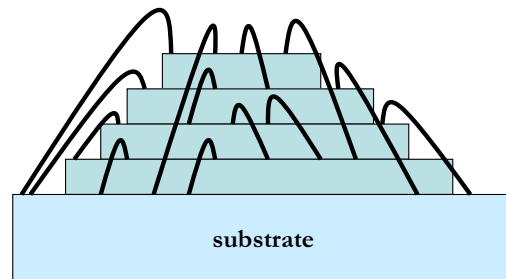


Figure 1: Wire bonding for die-stacking SiP design.

If a direct inter-die connection cannot be made for a signal between two dies, then we need to route the signal from the two dies to the package substrate by two bonding wires and use the substrate to complete its routing which will not only affect performance but is much more costly (because many SiPs use gold for bonding wires). It is desirable to maximize the number of direct inter-die connections and minimize the total bonding wirelength both for cost and performance consideration.

An important stage during the SiP physical design flow is *pad assignment* which assigns inter-die signals to die pads. Pad assignment is typically invoked after the partitioning of the components of a system (or sub-system) into different dies. After pad assignment, the subsequent floorplanning/placement and routing stages can then be carried out. When we do pad assignment, some wire crossings might be produced. Figure 2 shows a few types of crossings, where two wires are said to have a *crossing* if after projecting their pads to the $x-z$ plane, the two line segments connecting the pads intersect. Nevertheless, some types of crossings are in fact tolerable because it is relatively easy to prevent the two wires from actually touching each other when the bonding wires are made [1]. For example, Type 1 crossing in Figure 2 will not cause actual bonding wires to touch. On the other

hand, some types of crossings are *illegal*. The first type of illegal crossing is a crossing of two wires in which the upper pads of both wires are co-located on a die and the lower pads of both wires are also co-located on another die. Type 2 crossing in Figure 2 shows such a crossing. The other type of illegal crossing is Type 3 crossing in Figure 2 in which the four pads almost align into a straight line (after projecting them to the x - z plane). To make a more precise definition for Type 3 crossing, we introduce a user-specified constant dis such that if either segment l_1 or l_2 (see Figure 2) has length no more than dis , then we say the two wires has a Type 3 crossing. Note that the value of dis depends on the wire bonding rules. Since Type 2 and Type 3 crossings are likely to cause bonding wires to touch, we consider these two types of crossings as illegal ones, while treating the other types of crossing as legal ones during pad assignment. It should be pointed out that our pad assignment approaches (to be presented in Sections 3 and 4) are very flexible such that they can be directly applied or easily modified to handle any additional types of illegal crossings.

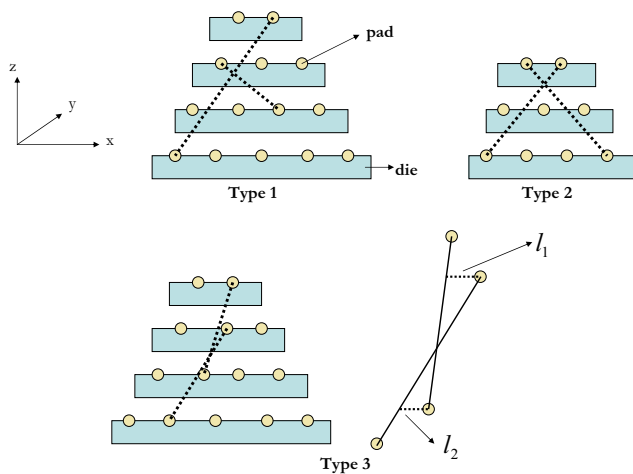


Figure 2: Some types of crossings.

In this paper, we study a pad assignment problem which prohibits the generation of illegal crossings and aims to minimize the total signal wirelength. We first consider a variety of special cases and present a minimum-cost maximum-flow [4] based approach to optimally solve them in polynomial time. We then describe an approach, which consists of two stages, to solve the general case. In the first stage, the left edge algorithm (which was originally designed for the classical channel routing problem) [7] is modified to assign as many signals as possible to die pads. If there are remaining signals whose pads cannot be determined in the first stage, our approach proceeds to assign them to die pads using the integer linear programming (ILP) technique [8] in the second stage. Extensive experiments are conducted and encouraging results are reported to support our approaches. To the best of our knowledge, our work is the first one which addresses a pad assignment problem for die-stacking SiP design.

The rest of this paper is organized as follows. Section 2 gives the problem assumptions and the problem formulation. Section 3 describes our minimum-cost maximum-flow based approach to a variety of special cases, and Section 4 gives the details of our two-stage approach for the general case. The experimental results are reported in Section 5, and we conclude this paper in Section 6.

2. ASSUMPTIONS AND PROBLEM FORMULATION

2.1 Assumptions

A die has a rectangular shape, and its pads are positioned along its four sides. For a SiP design, we assume that dies are arranged as a stack and their order is fixed. The size of an upper die is smaller than that of a lower die (and thus no dummy die is inserted in between). In addition, the number of pads on each side of an upper die is less than or equal to that on the same side of a lower die.

We also assume each signal needs to be assigned to exactly two pads on different dies, and each die has adequate pads to accommodate associated signals. Note that it is possible that some signal may only have a pad on a die which needs to be connected to a finger on the substrate. For this case, we can treat the substrate as a die sitting at the bottom of the die stack and each finger as a pad. A pad assignment result for a signal must be one such that both upper and lower pads assigned to the signal are located on the same side but on different dies. Therefore, the pad assignment result for the one shown on the left of Figure 3 is feasible for the signal while the one on the right is disallowed. For a signal involving more than two dies, we assume that it has been converted to two-pad signals in advance.

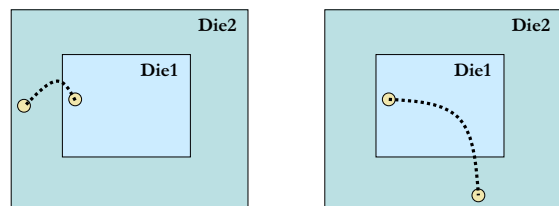


Figure 3: Allowed and disallowed assignments.

2.2 Problem Formulation

The pad assignment problem considered in this paper has the following inputs. A set of dies, numbered by $1, 2, \dots$, from top to bottom in the stack, and a set of signals, w_1, w_2, \dots , are given. Each signal w_i is associated with two die numbers U_i and L_i , representing the dies on which its upper pad and lower pad should be located. For each die, a set of pads on each side and their associated locations in the 3-dimensional space are also given.

The pad assignment problem asks to assign each signal w_i to two pads on the same side of dies U_i and L_i such that no illegal crossing is created and the sum of the wirelengths of all signals (i.e., the total signal wirelength) is minimized. After pad assignment, we know which two pads are assigned to a signal and therefore the wirelength of this signal can be calculated. For simplicity, we use the Euclidean distance between the two pads to approximate the actual length of the bonding wire.

3. AN APPROACH TO A VARIETY OF SPECIAL CASES

In this section, we consider a variety of special cases which can be solved optimally in polynomial time. We first consider a special case where only two dies are involved, and present a minimum-cost maximum-flow [4] based approach for it. We then explain how to extend this approach to other special cases.

3.1 Two-die case

For the case with only two dies, there is only one wire type, i.e., each wire is from the top die to the bottom die. We can formulate the two-die pad assignment problem as a minimum-cost maximum-flow problem as follows.

Suppose we are given k signals, and the pad sets P and Q on the two dies, respectively. We will construct a flow network $G = (V, E)$, where V and E are the node and edge sets, respectively. For each pad in $P \cup Q$, there is a node in V . A source node s , a sink node t , and two intermediate nodes m_1 and m_2 are also added to V . For each pad $p_i \in P$ and each pad $q_j \in Q$ (i.e., a pair of pads on different dies), if they are on the same side, there is a directed edge from p_i to q_j in E with the capacity being 1 and the cost being the wirelength between p_i and q_j . In addition, there is a directed edge from m_1 to each pad p_i in P and there is a directed edge from each pad q_j in Q to m_2 ; the capacity and cost of each of these edges are 1 and 0, respectively. Finally, there is a directed edge from s to m_1 and there is a directed edge from m_2 to t ; the capacity and cost of each of the two edges are k and 0, respectively. Figure 4 shows the flow network for an instance of the two-die pad assignment problem, where each node p_i ($1 \leq i \leq 12$) represents a pad on the upper die, each node q_j ($1 \leq j \leq 16$) represents a pad on the lower die, k is the number of signals, and $WL(p_i, q_j)$ denotes the wirelength between two pads p_i and q_j . In this example, it is also assumed that pads in the set $\{p_1, p_2, p_3, q_1, q_2, q_3, q_4\}$ ($\{p_4, p_5, p_6, q_5, q_6, q_7, q_8\}$, $\{p_7, p_8, p_9, q_9, q_{10}, q_{11}, q_{12}\}$, $\{p_{10}, p_{11}, p_{12}, q_{13}, q_{14}, q_{15}, q_{16}\}$, respectively) are on the same side.

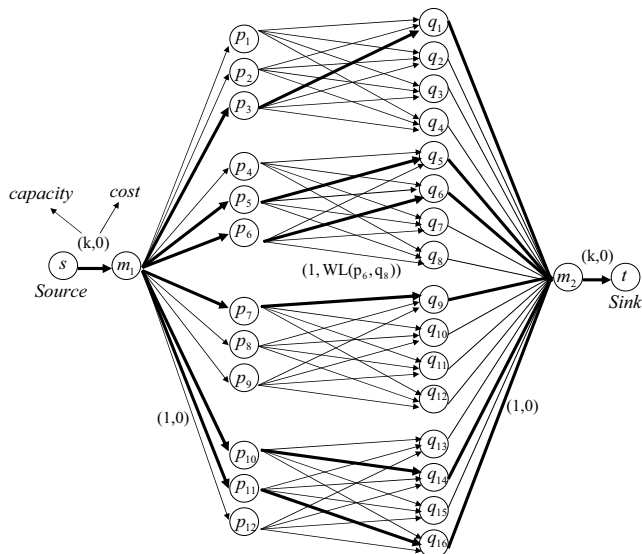


Figure 4: The flow network of a two-die instance.

After the flow network G is built, we proceed to find a minimum-cost maximum-flow of G [4]. It is well known that for any minimum-cost maximum-flow problem instance with integral edge capacities, there is always an integral optimum solution and the network simplex algorithm is guaranteed to find an integral optimum flow f in polynomial time. Since we assume that each die has adequate pads to accommodate associated signals, it is not hard to see that f must have flow value equal to the number of signals, k . We now explain how to produce the corresponding pad assignment solution from f . Because the flow value of f is k and the capacity of each edge except (s, m_1) and (m_2, t) is 1, there will be k paths of the form: $m_1 \rightarrow p_i \rightarrow q_j \rightarrow m_2$ such that all edges on each path are saturated (i.e., having a flow of 1);

in addition, except the starting node m_1 and ending node m_2 , these paths are all node-disjoint. As a result, we can find k node-disjoint saturated edges (p_i, q_j) 's from these paths, and assign the given k signals to the corresponding pads of these edges one by one in an arbitrary order. For example, in Figure 4, if there are 6 signals to be assigned, and the bold edges are saturated edges found in a minimum-cost maximum-flow, then we can assign the 6 signals to the pairs of pads (p_3, q_1) , (p_5, q_5) , (p_6, q_6) , (p_7, q_9) , (p_{10}, q_{14}) , (p_{11}, q_{16}) .

The above-mentioned minimum-cost maximum-flow based approach is called MCMF. It can be proved that the pad assignment solution produced by MCMF has minimum wirelength and does not have any crossing. Therefore we have the following theorem.

THEOREM 1. *The two-die pad assignment problem can be optimally solved by MCMF.*

3.2 Extensions to other cases

In fact, MCMF can be easily extended to handle other special cases with more than two dies. For example, Figure 5 gives a three-die pad assignment problem instance where for simplicity, we only show one side for each die. Each of the 5 signals a, b, c, d, e has to be assigned to a pad located on a common die, i.e., Die 2. For this example, we can construct the flow network as shown in Figure 6, where three intermediate nodes m_1, m_2 , and m_3 (instead of two in the two-die case) are created and there is no edge between each pair of p_i and r_j (because there is no signal connecting Die 1 and Die 3). The capacities of edges (s, m_2) , (m_1, t) , and (m_3, t) are set to 5, 2, and 3, respectively since Dies 2, 1, and 3 have 5, 2, 3 associated signals. Similar to the two-die case, we can also prove that an optimal pad assignment solution to this example can be produced by using the network simplex algorithm to find a minimum-cost maximum-flow from the corresponding flow network.

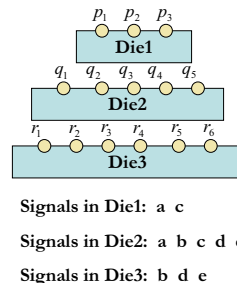


Figure 5: A three-die pad assignment problem instance.

We can generalize the 2-die and 3-die cases to any n -die case as long as each of the signals in the n -die case has to be assigned to a pad located on a common die. Consequently we have the following theorem.

THEOREM 2. *For any n -die pad assignment problem, where $n \geq 2$ and each signal has to be assigned to a pad located on a common die, it can be transformed into a minimum-cost maximum-flow problem and solved optimally.*

4. AN APPROACH TO THE GENERAL CASE

In this section, we describe an approach, which consists of two stages, to the general case. To reduce the problem complexity, our approach only focuses on a certain subset of solution space in the first stage such that the left edge algorithm [7] can be modified and applied to assign as many signals as possible to die pads. If there are remaining signals

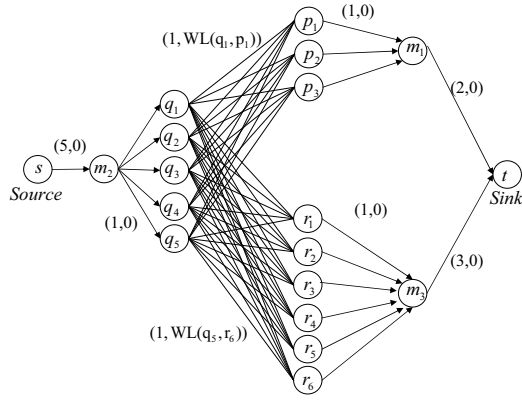


Figure 6: The flow network of the example in Figure 5.

whose pads cannot be determined in the first stage, an integer linear programming (ILP) based method is invoked in the second stage. Both stages are guaranteed not to generate any illegal crossing. This approach is called MLE+ILP, and the details of each stage are explained in the next two subsections.

4.1 The first stage

For each side of a die, we label all its pads from the center towards the two ends, starting with 1 (see Figure 7). The pads on different dies but on the same side and with the same label will form an *imaginary track*, and the length of the track is determined by the difference of the largest and smallest die numbers among all the dies covered by this track (see Figure 7). In the first stage, our approach is to assign as many signals as possible to these imaginary tracks. That is, a signal can be assigned only to two pads with the same label.

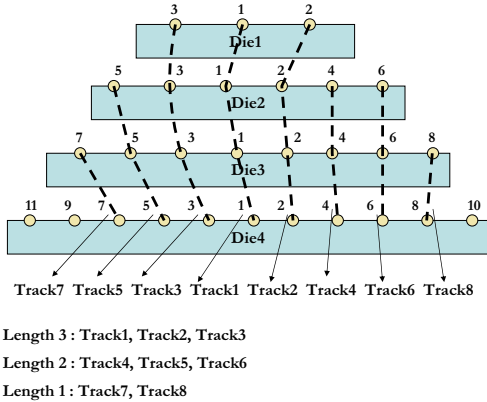


Figure 7: Labeling pads and forming imaginary tracks.

The original left edge algorithm [7] is to solve the channel routing problem, and therefore in order to modify and apply it to solve the pad assignment problem in the first stage, we need to decide the set of available tracks and the set of wires to be routed. In our pad assignment problem, each die has four sides, and therefore we will consider pad assignment for four sides simultaneously. For the example in Figure 8, it will form 12 tracks with length 3, 4 tracks with length 2, and 4 tracks with length 1 at the beginning, as shown in Figure 9, when four sides are considered together. Since each signal w_i is to be assigned to two pads on dies U_i and L_i , the signal

forms a wire to be routed with its two end points on dies U_i and L_i . Throughout the rest of this paper, signal and wire will be used interchangeably. For the example in Figure 8, signal b is on Die 1 and Die 3, so there will be a wire from Die 1 to Die 3. Figure 9 shows all tracks and all wires of the example in Figure 8. Note that the channel routing problem assumes that the available tracks are unlimited and have equal length, and the left edge algorithm tries to route the wires using as few tracks as possible. However, in our pad assignment problem, the tracks are limited and could have different lengths, thus making our problem different from the channel routing problem.

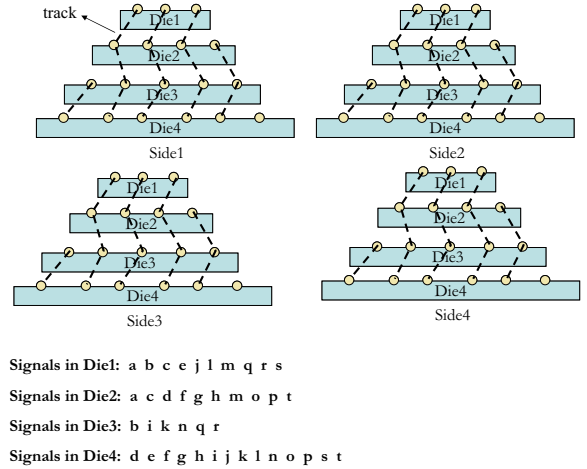
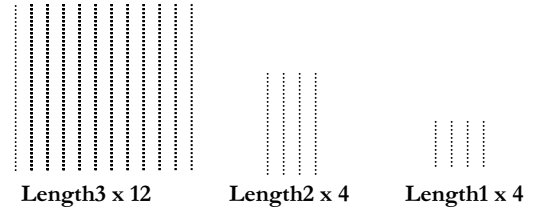


Figure 8: An example.

Tracks:



Wires:

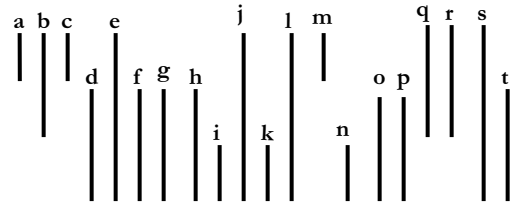


Figure 9: Deciding tracks and wires.

After all the tracks and wires are created, we sort all the wires to form an order for assignment. A wire with a higher upper end point (i.e., with a smaller die number at the upper end point) has a higher priority to assign. We also sort all the tracks in a non-increasing order of their lengths.

We start the assignment process one wire at a time according to the sorted order of wires. The tracks are considered for wire assignment according to the sorted order of tracks, and if the current track has no capacity for a wire, we will consider the next track. If all the tracks are considered and

have no capacity for the current wire, the wire fails to be assigned in this stage. Every time after a wire from Die i to Die j ($i < j$) is assigned to a track, the pads from Die $(i + 1)$ to Die $(j - 1)$ in the track can still be used later for some wires, and therefore we create a new track from Die $(i + 1)$ to Die $(j - 1)$ and add it to the end of the set of tracks. Note that the original left edge algorithm does not create such a track. After all the wires are processed, we need to check if there are any illegal crossings and remove them whenever necessary by making some assigned wires become unassigned.

The wire assignment result produced for the example in Figure 8 is shown in Figure 10. When the wire assignment is finished, we will distribute all the assigned wires to four sides uniformly, and meanwhile the wires assigned to the same track will be assigned to the same side and to the pads with same label. Figure 11 shows the pad assignment result produced for the example in Figure 8. We call the method used in this stage as the modified left edge algorithm. As shall be seen in Section 5, the majority of the signals can have their die pads assigned in the first stage.

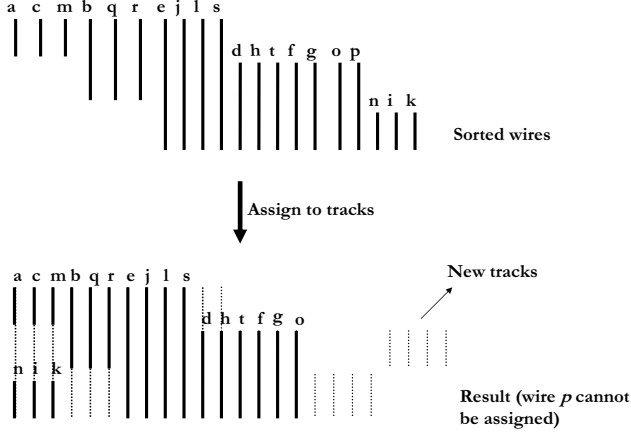


Figure 10: Wire assignment by the modified left edge algorithm.

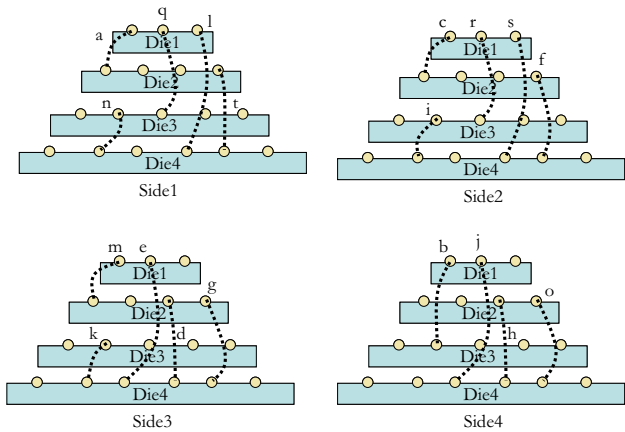


Figure 11: Pad assignment by the modified left edge algorithm.

4.2 The second stage

If there are wires which cannot be assigned to die pads after the first stage, we will assign them by an ILP based

method in the second stage. In this ILP formation, we have the following constants and variables.

- Constants

- $T_i: 1 \leq i \leq n$
The i -th wire type. The wire type of a signal is determined by its two associated die numbers. If two signals have the same associated die numbers, they have the same wire type. n is the total number of different wire types for the remaining signals.
- $N_{T_i}: 1 \leq i \leq n$
The number of remaining signals with wire type T_i .
- $C_{T_i}: 1 \leq i \leq n$
The number of wire candidates for wire type T_i . This amount can be determined from the remaining die pads. If a wire candidate causes an illegal crossing with an already assigned wire, then we will not create this wire candidate.
- $W_j^{T_i}: 1 \leq j \leq C_{T_i}, 1 \leq i \leq n$
The wirelength of the j -th wire candidate for wire type T_i .
- $P_k: 1 \leq k \leq m$
The k -th unassigned pad. Here we only consider those pads which may be used by unassigned signals. m is the total number of such unassigned pads.
- $E_j^{T_i}: 1 \leq j \leq C_{T_i}, 1 \leq i \leq n$
 $E_j^{T_i}$ is the set of the two pads which are used to realize the j -th wire candidate for wire type T_i .

- Variables

- $x_j^{T_i}: 1 \leq j \leq C_{T_i}, 1 \leq i \leq n$
 $x_j^{T_i} \in \{0, 1\}$. If $x_j^{T_i}$ is 1 in an ILP solution, it means that the j -th wire candidate of wire type T_i is selected; if $x_j^{T_i}$ is 0, the wire candidate is not selected.

For each wire type T_i , it needs to select exactly N_{T_i} wire candidates, so we have the following constraints:

$$\sum_{j=1}^{C_{T_i}} x_j^{T_i} = N_{T_i}, 1 \leq i \leq n$$

Each unassigned pad can only be used by a wire candidate or none, so we have the following constraints:

$$\sum_{\forall E_j^{T_i}: P_k \in E_j^{T_i}} x_j^{T_i} \leq 1, 1 \leq k \leq m$$

For any two wire candidates, say the j -th wire candidate of wire type T_i and the j' -th wire candidate of wire type $T_{i'}$, if they cause an illegal crossing, we need to add the following constraint to avoid the two wire candidates to be selected simultaneously:

$$x_j^{T_i} + x_{j'}^{T_{i'}} \leq 1$$

The objective is to minimize the total wirelength and hence is stated as follows:

$$\min \sum_{i=1}^n \sum_{j=1}^{C_{T_i}} (x_j^{T_i} \times W_j^{T_i})$$

It is worth mentioning that this ILP formulation exactly models our pad assignment problem, if the first stage of our approach is skipped.

To further reduce the time complexity of the ILP problem, we will reduce the number of wire candidates generated by each unassigned pad. For the wire candidates of a wire type that can be generated by a pad, we only keep at most R wire candidates which have shorter wirelengths than the others. Now if we can find a solution to this new ILP problem, it must also be a solution to the original ILP problem but may not be optimal. On the other hand, if there is no solution found for this new ILP problem, the value of R will be increased by a fixed amount to get another new ILP problem to be solved. The whole process is iterated until a solution is found, or the value of R reaches the upper limit but no solution is found.

5. EXPERIMENTAL RESULTS

Our approaches were implemented in C++ and run on an Intel 2.4GHz Linux machine with 8GB memory. We used CPLEX [2] to solve the ILP instances and LEDA [3] to solve the minimum-cost maximum-flow instances.

First, we compared the efficiency of our minimum-cost maximum-flow approach MCMF against an ILP based method on several instances of the special case with two dies only. The ILP method is to run our MLE+ILP approach directly from the second stage without setting the value of R to control the amount of wire candidates, and hence is also an optimal method. The test cases were randomly generated assuming the pad pitch is 50um, the thickness of a die is 6 mil (1 mil=25.4um), and the thickness of the film between adjacent dies is 1 mil. The details of the test cases and the results are shown in Table 1, where the WL columns show the total wirelength results. An optimal pad assignment solution produced from MCMF could be computed quickly in less than a second for every instance. On the other hand, when the pad assignment problem was modeled as an integer linear program, too many ILP constraints were generated and exceeded the memory limit of our system except for the three smallest instances.

Second, we experimented on three real SiP designs obtained from the industry. We note that they are examples of the special cases discussed in Section 3.2, so their pad assignments can be optimally determined by the proposed minimum-cost maximum-flow approach. The details of the three designs are given in Table 2, where wire type (i, j) means that it is from Die i to Die j . Table 2 also compares the assignment results by the minimum-cost maximum-flow approach against the original assignments. Column Original shows the original wirelengths. Column MCMF shows the results by the minimum-cost maximum-flow method. It can be seen that MCMF efficiently reduced the total wirelength by up to 36.2% (see the results in the Imp. column of Table 2). The run time of MCMF in each case was well under a second.

We also experimented on ten randomly generated instances for the general case with characteristics listed in Table 3. Table 4 shows the detailed results in each stage of our MLE+ILP approach for the instances. Stage 1 is assignment by the modified left edge algorithm and stage 2 is additional assignment by ILP. After stage 1, the majority of the wires were assigned to pads and the assignment ratio

was 78.44% on average. The run time for stage 1 was less than 0.01 second for all cases. The remaining wires were all assigned to pads in stage 2 by ILP. In each case, we created an initial ILP instance with the wire candidate range $R = 5$. And if an ILP instance was infeasible, we would increase R by 2 iteratively until a feasible assignment could be found. The number of iterations taken for all cases were listed in the last column of Table 4. The second stage was also very fast and could finish in a few seconds for all cases.

Finally, we checked if it was computationally feasible to perform pad assignment by ILP directly without reducing the problem size by the modified left edge algorithm for the test cases in Table 3. We tried the ILP approach without limiting the wire candidate range R . We also tried the ILP approach with R set to 5 initially and increased it by 2 iteratively if the corresponding ILP was infeasible. The results are shown in Table 5. Column ILP shows the results when no range was used while column ILP-R shows the results with range. When no range was used, the number of ILP constraints were very large and we ran out of memory in seven out of ten cases. We set a time-limit of 10,000 seconds for the ILP approach with no range and also for each iteration of ILP-R. Experimental results show that without the help of the modified left edge algorithm, the ILP instances with or without range were often too large for an optimal solution to be found in a reasonable amount of time. Time-outs occurred for most of the cases and we report the best solutions found before timeouts in Table 5. Comparing with our two-stage approach, the total wirelength generated by ILP-R was only 0.85% smaller on average. Hence, the two-stage approach is significantly more efficient than the ILP only approach with little loss of quality.

6. CONCLUSIONS

In this paper, we have presented novel pad assignment approaches for SiP design. The experimental results have been provided to demonstrate the efficiency and effectiveness of our approaches.

7. ACKNOWLEDGMENTS

We would like to thank Dr. Wang-Jin Chen at Faraday Technology Corporation for providing us with some technical suggestions and test cases.

8. REFERENCES

- [1] Bond wire modeling standard. [Online]. Available: <http://www.jedec.org/download/search/jesd59.pdf>.
- [2] Cplex. [Online]. Available: <http://www.ilog.com>.
- [3] Leda package. [Online]. Available: <http://www.algorithmicsolutions.com>.
- [4] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, 1993.
- [5] S. Al-sarawi, D. Abbott, and P. Franzon. A review of 3-d packaging technology. *in IEEE Trans. on Components, Packaging, and Manufacturing Technology - Part B*, 21(1):2-14, Feb 1998.
- [6] L. Golick, J. Goodelle, and T. Shilling. Sip modules call for right blend of tech. *EE Times*, May 11, 2004.
- [7] A. Hashimoto and J. Stevens. Wire routing by optimizing channel assignment within large apertures. *In Proceedings of the 8th Workshop on Design Automation*, pages 155-169, 1971.
- [8] R. J. Vanderbei. *Linear Programming: Foundations and Extensions*. Springer, 2007.
- [9] R. Wilson. Sips form good soc alternative, but designer beware. *EE Times*, May 11, 2004.

Table 1: Comparison of MCMF and ILP on two-die test cases; “-” denotes out of memory

test case	# of dies	# of wires	# of pads	MCMF		ILP	
				WL (um)	Time (s)	WL (um)	Time (s)
case1	2	40	96	7181.96	<0.01	7181.96	0.31
case2	2	80	176	14363.92	<0.01	14363.92	4.48
case3	2	160	336	28727.84	0.04	28727.84	103.7
case4	2	240	496	43091.76	0.07	-	-
case5	2	320	656	57455.68	0.13	-	-
case6	2	400	816	71819.60	0.20	-	-
case7	2	480	976	86183.52	0.27	-	-

Table 2: Results on real test cases

test case	# of dies	# of wires	# of pads	# of wires of each wire type			Original	MCMF		
				(1,2)	(1,3)	(2,3)	WL(um)	WL(um)	Imp.	Time(s)
case8	3	58	197	26	0	32	55483.88	42416.03	23.6%	0.02
case9	3	38	155	15	23	0	212760.68	135641.88	36.2%	0.01
case10	3	141	483	41	0	100	221244.62	169771.38	23.3%	0.08

Table 3: Detailed information of general test cases

test case	# of dies	# of wires	# of pads	# of wires of each wire type															
				(1,2)	(1,3)	(1,4)	(1,5)	(1,6)	(2,3)	(2,4)	(2,5)	(2,6)	(3,4)	(3,5)	(3,6)	(4,5)	(4,6)	(5,6)	
case11	4	160	352	19	22	35	0	0	31	26	0	0	27	0	0	0	0	0	0
case12	4	320	672	43	58	55	0	0	58	58	0	0	48	0	0	0	0	0	0
case13	4	640	1312	87	114	110	0	0	115	115	0	0	99	0	0	0	0	0	0
case14	5	160	340	16	11	11	14	0	17	16	11	0	12	28	0	24	0	0	0
case15	5	320	660	32	28	26	30	0	37	27	28	0	26	41	0	45	0	0	0
case16	5	640	1300	69	61	50	61	0	56	66	61	0	70	73	0	73	0	0	0
case17	6	160	336	4	10	6	8	8	14	14	5	7	12	8	8	12	16	28	28
case18	6	320	672	12	23	18	20	19	30	27	12	19	22	16	17	20	29	36	36
case19	6	640	1296	46	30	49	34	37	35	51	34	38	30	56	57	42	48	53	53
case20	6	800	1632	59	38	65	45	45	44	60	46	51	41	64	69	52	58	63	63

Table 4: Detailed results of each stage of MLE+ILP

test case	stage 1				stage 2				
	Time (s)	# of assigned wires	assigned %	WL (um)	Time (s)	# of assigned wires	assigned %	WL (um)	# of Iter
case11	<0.01	134	83.75	40578.07	0.02	26	16.25	10002.07	1
case12	<0.01	262	81.88	77206.07	0.16	58	18.12	21450.04	2
case13	<0.01	521	81.40	153514.39	7.69	119	18.60	42734.46	5
case14	<0.01	131	81.88	41655.37	0.01	29	18.12	12600.69	1
case15	<0.01	264	82.50	85285.77	0.03	56	17.50	25334.00	1
case16	<0.01	508	79.38	165185.07	0.08	132	20.62	59052.82	1
case17	<0.01	130	81.25	43091.76	0.03	30	18.75	15211.93	1
case18	<0.01	252	70.00	91390.44	0.78	68	30.00	35945.53	2
case19	<0.01	455	71.09	170930.64	6.55	185	28.91	100829.31	6
case20	<0.01	570	71.25	215638.34	10.53	230	28.75	123260.75	6
Average			78.44				21.56		

Table 5: Comparison between ILP, ILP-R, and MLE+ILP on general test cases; “-” denotes out of memory

test case	ILP		ILP-R			MLE+ILP	
	WL (um)	Time (s)	WL (um)	# of Iter	Time (s)	WL (um)	Time (s)
case11	49972.38	161.55	49972.38	1	0.79	50580.14	0.02
case12	-	-	98217.35	1	588.39	98656.11	0.16
case13	-	-	195913.38	1	10000	196248.85	7.69
case14	54122.13	10000	54122.13	1	10000	54256.06	0.01
case15	-	-	110422.41	1	10000	110619.77	0.03
case16	-	-	223861.00	1	10000	224237.89	0.08
case17	58056.77	10000	58059.08	1	10000	58303.69	0.03
case18	-	-	126149.74	1	10000	127335.97	0.78
case19	-	-	264067.02	2	20000	271759.95	6.55
case20	-	-	332138.26	1	10000	338899.09	10.53
Ratio			1.0			1.0085	