

A Flow Based Approach to the Pin Redistribution Problem for Multi-Chip Modules *

Douglas Chang

Teofilo F. Gonzalez

Oscar H. Ibarra

Department of Computer Science
University of California
Santa Barbara, CA 93106

Abstract

We investigate the pin redistribution problem (PRP) for multi-chip modules. A novel transformation to the max-flow problem is introduced. This approach provides an efficient algorithm for finding a 2-layer solution, whenever one exists. A greedy heuristic to find a k -layer solution is described. Our approach can find a minimum layer solution for two variants of the PRP; when each net can be routed on more than one layer, and when source and target terminals are drilled through all layers. Except for the heuristic procedure which takes $O(km^4 \log^2 m)$ time, our algorithms take $O(|S|km^2)$ time, where S is the set of source terminals, m is the number of rows and columns in the grid, and k is the number of layers required. One can show that generalizations of the k -layer PRP are NP-complete problems.

1 Introduction

The packaging between computer chips has become a greater factor in system performance as chip speeds have increased. Fifty percent of the delay in high-performance systems can be attributed to packaging, and this will likely increase in the future [Bak90]. Multi-Chip Modules (MCMs) have been introduced to reduce inter-chip delay by removing one layer of packaging. This improves system performance and reliability. In the MCM technology [She93], bare chips are placed on a common substrate called the *chip layer*. Directly below the chip layer there are a number of *pin redistribution layers*, and below them there are the *signal distribution layers* (see Figure 1). Some MCMs use the bottom signal distribution layer as a power

distribution layer. For simplicity, we omit this special layer, but our algorithms can be easily adapted to handle this situation. The pin redistribution layers are used to redistribute the chips' I/O pins to a set of pins with a minimum spacing, as required by the signal distribution layers. This redistribution can also be used to spread the pins uniformly over the MCM, which leads to fewer signal distribution layers, fewer vias, and minimal crosstalk [CLS92]. Lastly, the signal distribution layers are used to connect the appropriate (redistributed) chip I/O pins.

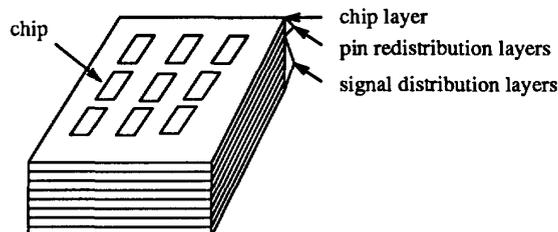


Figure 1: Multi-Chip Module.

An early example of MCM technology is IBM's ceramic multichip technology, used in the IBM 3081 processor in the late '70s [BB82]. More recent examples are IBM's glass-ceramic/copper module for the System 390/9000 and DEC's multilevel thin film for the VAX 9000 [Tum91]. A detailed discussion of various MCM technologies is given in [Tum89].

In this paper, we investigate the k -layer pin redistribution problem (PRP). This problem is to connect (redistribute) the (source) I/O pins on the chip layer to target locations on the bottom redistribution layer, using k redistribution layers. A wiring with the minimum number of redistribution layers (i.e., the optimization version of the PRP) can be obtained by solving a set of k -layer problems. Since an algorithm for

*Research partially supported by NSF Grant CCR-8918409.

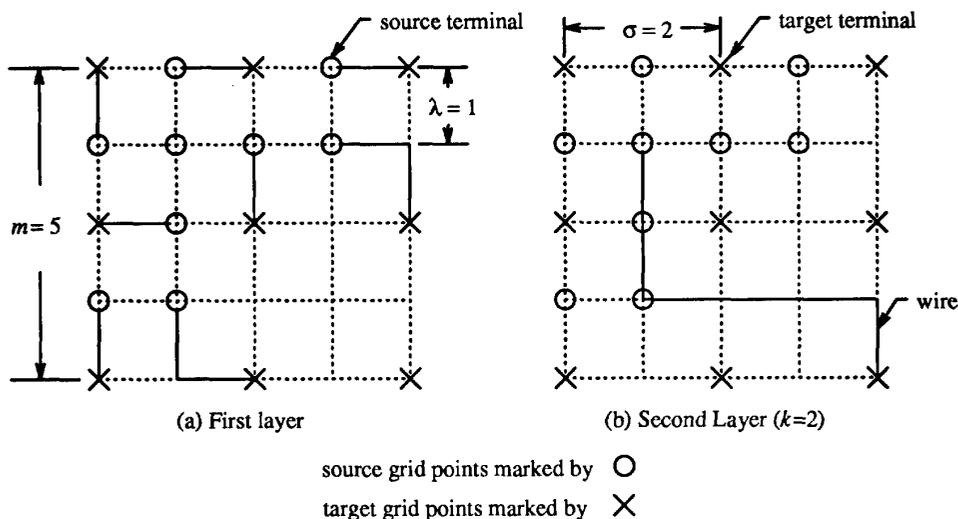


Figure 2: PRP Instance.

one of these problems can be easily adapted from the other, we will refer to both of these problems as the PRP.

1.1 The Pin Redistribution Problem

The basic model used is the k -layer routing model as described in [Len90]. In this model, the routing graph consists of k stacked grid graphs (each representing one layer). The grid graphs (or layers) are numbered in increasing order from top (1) to bottom (k). Each edge in the graph can accommodate one wire segment. Vertical vias are available at each grid intersection.

Given a set of source terminals on grid 1 (the top grid) and a set of target terminals on grid k (the bottom grid), the PRP is to connect each source terminal to a different target terminal by a wire in *only one* layer (grid) such that no two wires on the same layer (grid) intersect. Note that a source terminal does not have to be connected to a specific target terminal; it just has to be connected to some target terminal. This is the main difference between the PRP and a conventional *two pin routing problem*.

A source (target) grid point is defined as an (r, c) grid point where there is a source (target) terminal on grid 1(k). Note that the only vias that can be used are at the source and target grid points, and these vias can only be used by the corresponding source or target terminal. This is because each net has to be routed on *only one* layer. Another observation is that if the net

for the source (target) terminal (r, c) is routed in layer i then the grid point (r, c) in any layer $j > i$ ($j < i$) can be used to route another net in layer j . However, the grid point (r, c) in any layer $j \leq i$ ($j \geq i$) cannot be used by another net. An instance and solution of a PRP is shown in Figure 2.

Two variations of the PRP are also discussed. In one variation nets are allowed to be routed on more than one layer, and vias are allowed at any subset of grid points. Under this model, stack vias are allowed. The second variation of the PRP is when source and/or target terminals are drilled through all the redistribution layers. These variations can be solved efficiently by our technique.

1.2 Previous Work

Cho and Sarrafzadeh [CS] introduce and formalize the PRP. In their formulation, the PRP is a 6-tuple, $(k, m, S, T, \lambda, \sigma)$ (see Figure 2), where

1. k is the number of layers available for pin redistribution, including the chip layer.
2. m is the number of rows and columns in the grid.
3. S is the set of grid points on the chip layer where the source terminals are located.
4. T is the set of grid points on the bottom redistribution layer where the target terminals are located.

5. λ is the minimum legal distance between two parallel wires on the same layer.
6. σ is the distance between adjacent target terminals (i.e., target terminals are spread uniformly over grid k).

A solution to the PRP is a wiring connecting all the source terminals to the target terminals in the k -layer grid, such that each net is wired on one layer, no two wires intersect, and the minimum distance λ between two parallel wires on the same layer is maintained. Our PRP formulation differs in two ways: (1) there is no restriction on the placement of target terminals, and (2) the value of λ is equal to 1.

Cho and Sarrafzadeh present three heuristics to solve the PRP. Their first heuristic is based on concurrent maze routing. The other two heuristics are based on finding a global routing, and then performing the detailed routing. They also show that given a special global routing with density two (at most two wires can be assigned to each grid point), a 2-layer solution can be found in polynomial time. However this special global routing does not always exist, so in the worse case, they need to double the routing area to generate a 2-layer solution.

We have taken a different approach to solving the PRP. Our approach is based on reducing the PRP to the maximum flow (max-flow) problem, which can be solved efficiently [Eve79]. We show that given the restriction of $\lambda = 1$, a 2-layer solution can be found quickly, whenever one exists. The time complexity of our algorithm is $O(m^2|S|)$, which is less than the algorithms in [CS]. We then present a heuristic procedure, based on the 2-layer algorithm to find a suboptimal solution to the optimization version of the PRP. Note that in most practical cases a solution using at most three layers exists [CS]. We also present an algorithm for the k -layer PRP when nets are allowed to be routed on more than one layer. Lastly, we show that if we restrict each source (target) grid point to be used only by the wire connecting that source (target) terminal, a k -layer solution can be found efficiently, whenever one exists. All of our algorithms take $O(|S|km^2)$ time, except for the heuristic procedure which takes $O(km^4 \log^2 m)$ time.

2 Flow Solution

An input to the max-flow problem is a directed graph G , called the flow graph, with two special nodes labeled s (source) and t (sink). Each arc in the flow

graph has a positive real capacity associated with it. A feasible flow F is any assignment of flow values to each of the arcs in the flow graph such that the flow along each arc is between 0 and the flow capacity of the arc, and the flow at each node is conserved (i.e., the flow into a node must equal the flow out of it). The max-flow problem consists of finding a maximum feasible flow from the source s to the sink t . There are a number of efficient algorithms to solve the max-flow problem [AMO93]. It is well known that when all the capacities are integers, a maximum flow in which all the arcs have integer flows exists, and algorithms such as Ford and Fulkerson's or Dinic's generate such a flow [Eve79].

Let us now consider the following reduction from the k -layer PRP to the max-flow problem. We map the routing grid to a flow graph as follows. Each grid point is represented by a flow cell (see Figure 3). Each flow cell has a subset of the *in* arcs (IN, IE, IS, IW), *out* arcs (ON, OE, OS, OW), *layer* arcs (IA, OB), and the *inner layer* arc F. Each arc has capacity 1. A flow cell with all the possible arcs is shown in Figure 3a. Flow cells corresponding to adjacent grid points are connected as follows. Flow cell X immediately to the west of flow cell Y has a correspondence between arcs $X.OE$ and $Y.IW$. There is also a correspondence between arcs $X.IE$ and $Y.OW$. A similar arrangement holds for flow cells immediately to the east, north, and south, as shown in Figure 3b. The inner layer arc is present in every flow cell. The only flow cells with layer arcs are those representing source and target grid points (called source and target flow cells). If flow cells X and Y correspond to the same source or target grid point in layers i and $i + 1$, respectively, then $X.OB$ corresponds to $Y.IA$.

There are two additional nodes, the source s and the sink t . The IA arcs of all the source flow cells on grid 1 emanate from s , and all the OB arcs of the target flow cells on grid k enter t (see Figure 4).

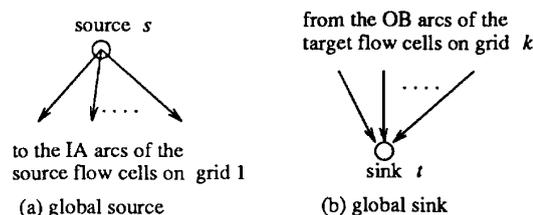


Figure 4: Global source and sink (all arcs have capacity 1).

Suppose that a problem instance I of the PRP has

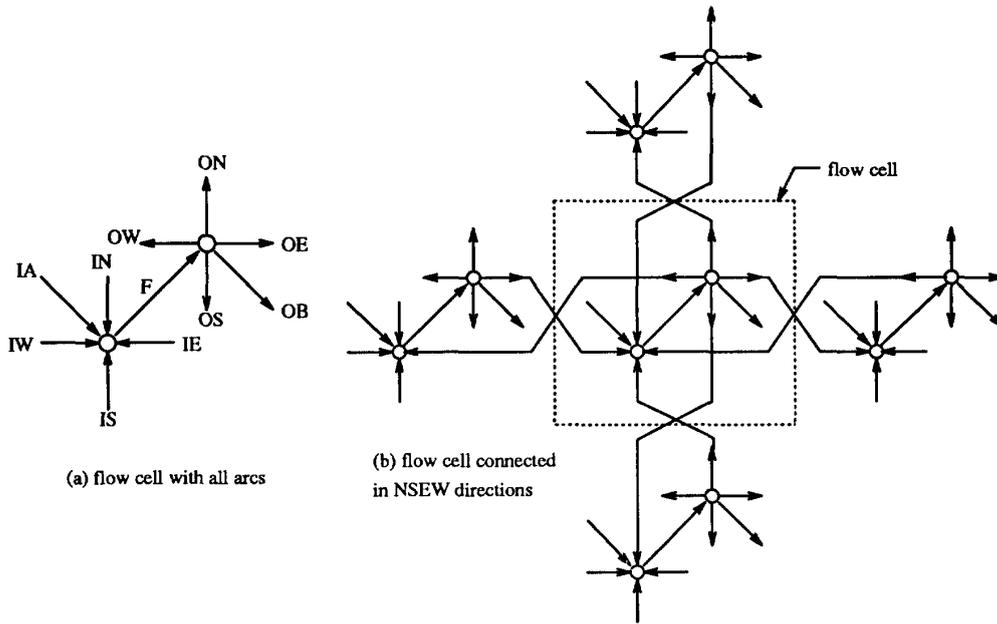


Figure 3: Flow Cell (all arcs have capacity 1).

a k -layer routing. We claim that the max-flow problem instance, $M(I)$, generated by our reduction for problem instance I has a maximum flow from source to sink equal to $|S|$. Let R be any valid routing for I . Now consider net j routed on layer i in I . For this net we construct a flow of one unit from source to sink (i.e., its flow path) as follows. Starting at the source s send a flow of one unit through the source flow cells for the net until you reach the part of the flow graph representing layer i . Then send a flow of one unit from that flow cell in layer i to the flow cell that represents the target grid point for net j along the path corresponding to the route followed by the wire connecting net j in the layout R . Then from that target flow cell to the sink, a flow of one is sent that goes through only the corresponding target flow cells for the net. The flows for all the nets can be easily combined into a valid flow from s to t with value $|S|$.

To prove the converse, i.e., show that if there is a flow with value $|S|$ from source to sink in $M(I)$ then the PRP has a routing, is not possible. This is because the flow might imply a routing for a net in the PRP in more than one layer rather than just one. An example of this is shown in Figure 5. For this simple example there obviously is a 2-layer solution. For more complex examples there are no 3-layer solutions, but an illegal 3-layer solution is found by the algorithm (i.e., some

nets are routed in two or more layers). However, for the case when $k = 2$, the converse claim always holds. For this case, when the flow value is $|S|$, a flow path cannot be on two layers (i.e., each flow path uses the in arcs and out arcs in only one layer). This is because if a flow path is on both layers it must be blocking another flow path, thus the flow value would be less than $|S|$.

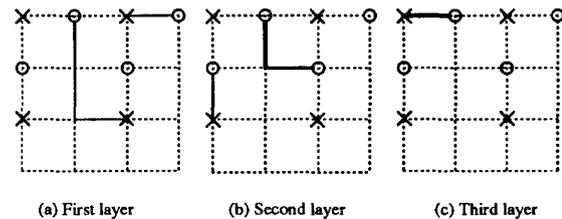


Figure 5: Illegal wiring (thick line) that could occur with 3-layer PRP.

Because the capacities are integer, from max-flow theory we know there is always a flow with integer values on all the edges, and such a flow is obtained by most maximum flow algorithms. This flow can be seen as consisting of a set of edge disjoint flow paths, and these flow paths can be easily translated to a routing of each one of the nets in exactly one of the two layers.

Therefore, we claim that for the 2-layer case, there is a solution to the PRP if and only if the maximum flow is equal to the number of source terminals ($|S|$). Furthermore, the layout can easily be constructed from the maximum flow.

For this special type of flow graph, one can show that both Ford and Fulkerson, and Dinic's algorithms take time $O(|S|m^2)$. This is also faster than Even and Tarjan's algorithm for flow graphs with all arc capacities equal to one which has time complexity $O(m^3)$ for our special type of graph [ET75]. Let us now establish our time complexity bound. Ford and Fulkerson's algorithm begins with a flow of zero. Then in $O(m^2)$ time a flow augmenting path can be constructed, whenever one exists, that when added to the previous flow increases the flow value by one. Clearly, no more than $|S|$ of such flow augmentations need to be performed, thus the time complexity becomes $O(|S|m^2)$. Dinic's algorithm is more sophisticated, because at each iteration a set of augmenting paths may be found, and that set normally has cardinality greater than one. Each of the phases takes $O(m^2)$ time. For a large number of problem instance Dinic's algorithm will be faster than Ford and Fulkerson's algorithm; however, we were able to construct flow graphs for which Dinic's algorithms requires $|S|$ phases to find a maximum flow.

We propose the following strategy for the $k > 2$ PRP. Connect the maximum number of nets in the first two layers, while allowing the unconnected source terminals to reach layer three. Then repeat the same operation for layers three and four, and so on. This can be done by modifying the previous flow graph to have the OB arcs of both the source and target flow cells on the second grid enter t . Then a cost of 1 is placed on each arc in the flow graph, except for the arcs from source flow cells to t , which get a cost of 2. Now the PRP has been transformed to the minimum cost flow problem which can be solved efficiently [AMO93]. This gives the maximum number of connections on two layers, and source terminals not connected on the first two layers are available for connecting on lower layers. When we reapply the algorithm on lower layers the grid points of the target terminals that have been connected above are no longer available for wires, so their inner layer arc is removed. The time complexity of this algorithm is $O(km^4 \log^2 m)$.

2.1 Variations of the PRP

Two interesting variations of the PRP can be solved using the flow strategy. One variation is allowing nets to be wired on more than one layer. A k -layer solution

```

begin
  create appropriate flow graph for 1-layer PRP;
  apply max-flow to the flow graph;
  while (flow value < |S|)
    begin
      add one layer to the bottom of the flow graph;
      extend flows that have already been found;
      through the added layer;
      apply max-flow to the new flow graph;
    end
  end
  output the flow paths found;
end

```

Figure 6: Algorithm for PRP variations

for this variation is found by the max-flow algorithm on our original flow graph construction. Note that the PRP solution in Figure 5 would be a legal wiring under this variation. The flow graph can easily be modified to allow vias at locations other than the source and target grid points. If vias are allowed at grid point (r, c) then the corresponding (r, c) flow cell will have the layer arcs. Note that the solution may include stacked vias. The algorithm given in Figure 6 finds a minimum layer solution for this variation of the PRP.

Note that the flows found in iteration i are *not* discarded in iteration $i + 1$. Thus the total number of augmenting paths that need to be found is $|S|$. The size of the final flow graph is $O(km^2)$. This leads to a time complexity bound of $O(|S|km^2)$.

A second variation is restricting each source and target grid point to be used only by the wire connecting the corresponding source or target terminal. In other words, the source and target terminals are drill through all the layers. This is similar to many printed circuited board (PCB) technologies. In this variation, we can also obtain a k -layer solution efficiently, whenever one exists. The modification of the original flow graph needed is removing the in arcs for each source flow cell, and removing the out arcs for each terminal flow cell. This means that no wire can go through the source and target grid points. Then the max-flow algorithm is applied. A flow of $|S|$ can be achieved in the graph if and only if there is a k -layer layout. A minimum layer solution can be found in time $O(|S|km^2)$ by using the algorithm given in Figure 6. The only difference from the first variation is in the flow graph construction.

3 Generalized PRP

Our algorithms can easily be adapted to handle the case when the grid has holes in it, where routing is not possible. This could occur, for example, if areas are reserved for routing power and ground. Our algorithms can also handle a more general PRP in which the grid graph is replaced by an arbitrary graph. We can show (proof omitted) that this generalized PRP is NP-complete for $k > 2$ layers. As before, we can obtain a k -layer solution (whenever one exists) for the generalized PRP, under the two variations discussed above. However if we weaken the second variation by restricting only the source grid points or only the target grid points instead of both, the problem becomes NP-complete again.

4 Conclusion

We introduced a novel way of solving the PRP based on maximum flow and minimum cost flow algorithms. This led to an efficient algorithm for the 2-layer PRP. We also proposed a greedy heuristic for the k -layer PRP. Two variations on the k -layer PRP were solved by the flow technique. This technique also can be used for the generalized k -layer PRP. We can show this problem is NP-Complete for $k > 2$. Currently we are studying the case when $\lambda > 1$, and trying to extend the NP-Completeness result to the grid PRP.

References

- [AMO93] Ahuja, R. K., Magnanti, T. L., and Orlin, J. B., *Network Flows: Theory, Algorithms, and Applications*. Englewood Cliffs, NJ: Prentice Hall, 1993.
- [Bak90] Bakoglu, H. B., *Circuits, Interconnections, and Packaging for VLSI*. The VLSI systems series, Reading, Mass.: Addison-Wesley Pub. Co., 1990.
- [BB82] Blodgett, A. J. and Barbour, D. R., "Thermal Conduction Module: A High-Performance Multilayer Ceramic Package," *IBM Journal of Research and Development*, vol. 26, pp. 30-36, January 1982.
- [CLS92] Cho, J. D., Liao, K. F., and Sarrafzadeh, M., "Multilayer Routing Algorithm for High Performance MCMs," in *Fifth Annual IEEE International ASIC Conference and Exhibit*, pp. 226-229, 1992.
- [CS] Cho, J. D. and Sarrafzadeh, M., "The pin redistribution problem in multi-chip modules." to appear in the Special Issue of Mathematical Programming B on Applications of Discrete Programming in Computer Science. A preliminary version is in *Proc. of the Fourth Annual IEEE Inter. ASIC Conference and Exhibit, 1991*.
- [ET75] Even, S. and Tarjan, R. E., "Network Flow and Testing Graph Connectivity," *SIAM Journal on Computing*, vol. 4, pp. 507-518, December 1975.
- [Eve79] Even, S., *Graph Algorithms*. Computer software engineering series, Potomac, MD: Computer Science Press, 1979.
- [Len90] Lengauer, T., *Combinatorial Algorithms for Integrated Circuit Layout*. Applicable theory in computer science, New York: John Wiley & Sons, 1990.
- [She93] Sherwani, N. A., *Algorithms for VLSI Physical Design Automation*. Boston: Kluwer Academic Publishers, 1993.
- [Tum89] Tummala, R. R., *Microelectronics Packaging Handbook*. New York: Van Nostrand Reinhold, 1989.
- [Tum91] Tummala, R. R., "Electronic Packaging in the 1990's — A Perspective from America," *IEEE Transactions on Components, Hybrids, and Manufacturing Technology*, vol. 14, pp. 262-271, June 1991.