Pin Assignment Approach to PGA Package and PCB Codesign

Guo-Chang Tzeng, Shuenn-Shi Chen, and Sao-Jie Chen

Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan, Republic of China

Abstract
Given a Pin-Grid-Array (PGA) package with a number of I/O pads around the boundary of the package and a number of devices (blocks) distributed on a printed-circuit-board (PCB), we propose an algorithm in this paper to find a pin assignment solution which eases the routing of a PGA and improves the nets routability of a PCB. We use a simulated-annealing algorithm to improve the solution by exchanging the pin assignment for some chosen nets on a package. The routing cost of the PGA package and PCB can be calculated separately for the pin assignment algorithm. Experiments show that routings produced by our pin assignment router with PCB routing consideration are better than the routings of that without considering PCB routing.

1. Introduction
In past years, the electronics industry has played a predominant role in one of the most fascinating, dynamic, and important industries because it has literally transformed the world and provides many products to affect our daily lives—for example, mobile telephones, personal computers, workstations, etc. One of the key technologies to helpfully make these products is focused on electronics packaging and assembly technology.

Packaging is an art based on the science of establishing interconnection levels ranging from either single-chip or multichip modules, from the chip die to chip packaging, from the chip to printed-circuit-board (or mother boards) and even from PCB to the whole system. As a result, developing a high-performance packaging tool becomes indispensable. Lately, many automated routers for PGA packages have been proposed in [1-4]. But in the state-of-the-art packaging tools do not properly offer an integrated developing environment to simultaneously consider both the PGA routing and PCB routing.

To cope with the situation above, we propose in this paper an algorithm to integrate both the PGA routing and PCB routing by choosing a better pin assignment of the PGA package. The conceptual representation of the problem of package pin assignment is shown in Figure 1. A large number of I/O pins distributed on the substrate of the package, should be mounted on the printed-circuit-board after the chip packaging. A net may start from one of the I/O pins, and be routed with other terminal pins of other blocks on the printed-circuit-board.

Figure 1. The conceptual representation of the problem of package pin assignment.

As the structure and conceptual routing environment of a PGA package is shown in Figure 2. Our proposed router is composed of three major phases: (1) PGA routing cost estimation. (2) PCB routing cost estimation. (3) A simulated-annealing technique [5] is employed to find a better pin assignment solution of the PGA package. Once the pin assignment of the PGA package has been made. The package routing and PCB routing are performed by using Chen’s [4] router and Maze router [6-7], respectively.

Figure 2. Routing environment of a PGA package.

The rest of this paper is organized as follows. Section 2 presents the problem formulation. Details of the algorithms are discussed in Section 3. The

---

*This work was supported by the National Science Council, Taipei, Taiwan, R.O.C., under Grant no. NSC 88-2216-E-002-017.
experimental results are reported in Section 4. Finally, we conclude this paper in Section 5.

2. Problem Formulation

Referring to Figure 1, all the devices and a package are mounted on the PCB that are called blocks. The pins of the blocks, excluding the package, are called terminal pins. The space that allows the routing wires to go through is called routing area, which may be divided into routing regions. As the routing wires to connect these terminal pins and I/O pins are called nets, which can be routed in 90 or 45 degrees on different layers through the vias. For convenience, the following symbols are defined and used in this paper.

• \( k \): the zone number, where \( k = 1, 2, 3, \) and 4 means the left, top, right, and bottom zones, respectively, as shown in Figure 3.

![Figure 3. Four regions of pin assignment.](image)

- \( T_{ki} \): the \( i \)-th I/O pads in the \( k \)-th zone.
- \( P_{ki} \): the \( i \)-th I/O pins in the \( k \)-th zone. The order of the I/O pins is from the inner- to outer-ring, as shown in Figure 2.
- \( TP \): the \( i \)-th terminal pins of the blocks on a printed-circuit-board.
- \( R_j \): the \( j \)-th routing region on a printed-circuit-board.

Formally stated, given a PGA package \( P = \{ P_{ki} \mid 1 \leq k \leq 4; i = 1, 2, \ldots \} \), I/O pads of a bare die chip \( T = \{ T_{ki} \mid 1 \leq k \leq 4, i = 1, 2, \ldots \} \) and a PCB design, the propose of our pin assignment problem is to find a mapping function \( F(T \rightarrow P) \), such that \( F(T_{ki}) = P_{ij} \) for some \( i, j \), \( 1 \leq k \leq 4 \); and \( F(T_{ki}) \neq F(T_{kj}) \) if \( k \neq g \) or \( i \neq j \).

Our object function is to minimize the total cost of such pin assignment

\[
\sum_{k,i} \lambda \cdot C(F(T_{ki})) + \sum_{j} \delta \cdot C(R_j)
\]

Where \( C(F(T_{ki})) \) is the cost of each net routed for the set \( T \) of I/O pads, for each \( k, i \). And \( C(R_j) \) is the cost of \( j \)-th region on a PCB. The \( \lambda \) and \( \delta \) are the weight values of the cost of packaging routing and PCB routing, respectively.

3. Algorithm Descriptions

The design flow of our algorithm is refined into the following subsections.

3.1 PGA routing cost estimation

In the package routing [4], nets are routed from the inner- to outer-ring in a multi-ring routing environment. Therefore, we must estimate some routing factors like crossing number, detour length, and capacity in the package routing cost for the pin assignment solution. The estimation algorithm is based on the notation of an inversion-table similar to Knuth [8]. We can find two sets of \( V, D, \) and \( C \) values and store them in an \( R \)-Step inversion table and a \( L \)-Step inversion table, respectively. Here, the \( V, D, \) and \( C \) represent the inversion value, distance, and capacity, respectively.

Given a pin set \( P_i \), and a pad set \( T \) in a PGA package. Let \( (P_{ki}, P_{k(n-i)+1}, \ldots, P_{k4}) \) be a permutation of sorted data \((n, n-1, \ldots, 1)\), then \((dn, da-1, \ldots, d1)\) represent the right (left) inversion values of \( (P_{ki}, P_{k(n-i)+1}, \ldots, P_{k4}) \), where each \( di \) represents the number of elements located at the right (left) side of \( P_{ki} \) and greater (less) than \( P_{ki} \). For instance, given a permutation \( (3, 2, 4, 1, 5) \), its right and left inversion values are \((2, 2, 1, 1, 0)\) and \((0, 0, 2, 0, 4)\), respectively. Again, assume pin \( P_{ki} \) have an inversion point \( P_{kj} \) at a ring, where an inversion point of net \( P_{ki} \) in an \( R \)-Step (a \( L \)-Step) table is the first element greater (less) than \( P_{ki} \) scanning from right to left (from left to right). The inversion distance of net \( P_{ki} \) is defined as \( D = m - n \), where \( m \) and \( n \) are the pin positions of \( P_{ki} \) and its inversion point \( P_{kj} \), respectively. On the other hand, the inversion capacity \( C \) of pin \( P_{ki} \) is equal to the number of nets which may pass through between \( P_{ki} \) and \( P_{kj} \). That is \( C = |P_{ki} - P_{kj}| - 1 \).

Finally, the weight of each net is calculated as \( W = C(F(T_{ki})) = k_i V + k_i D + k_i C \), for each \( k, i \); where \( k_1, k_2, \) and \( k_3 \) are constants. The nets having minimum weights in the \( L \)-Step inversion table are assigned to one layer and the others are assigned to another layer. This process is called layer assignment. As shown in Figure 4, nets (8, 5, 3, 7) have to be assigned to layer 1 and nets (4, 6, 2, 1) to layer 2. This process is repeatedly run until the number of routing layers of package are exhausted.

![Figure 4. Estimation of package routing cost.](image)
3.2 PCB routing cost estimation

In PCB routing, almost all the spaces (even under blocks) can be used for routing of all the nets. In our routing model of PCB, we construct the whole area of a PCB into rectangular routing channels and their corresponding channel connection graph (CCG) is shown in Figure 5, where a vertex and an edge in CCG represent a routing channel and the connection of the two channels, respectively. Here, we define the capacity of the channel as the number of routing lines that can pass through this channel. Furthermore, since we can determine the connection of these channels in computing their capacity, we represent CCG as an $L \times M \times N$ dimension matrix, which is called channel connection matrix (CCM). An element in CCM represents a vertex in CCG.

![Routing channels and CCG](image)

(a) Routing channels  
(b) CCG

Figure 5. The routing model of PCB.

For the estimation of congestion in each routing channel, we assign a cost for each of them to evaluate the selection of a routing path by using Maze router [6-7]. When a channel element in CCM is initialized, its capacity is set to the maximum number of nets that a channel may contain minus the space occupied by obstacles. Therefore, the maximum capacity of each channel is formulated as follows:

$$C_M = \sum_i \left[ \text{Width} \left( L_{\text{max}} + S_{\text{max}} \right) \right] \cdot \left[ \text{Height} \left( L_{\text{max}} + S_{\text{max}} \right) \right]$$

where $C_M$ is the maximum capacity, Width is the width of a channel, Height is the height of a channel, $L_{\text{max}}$ is the maximum width of the routing line, and $S_{\text{max}}$ is the maximum space between two routing lines. If we define our channel as a square area in each layer, the above equation can be simplified to

$$C_M = 2l \left[ \text{Width} \left( L_{\text{max}} + S_{\text{max}} \right) \right]$$

Because all the channels in our PCB routing model have fixed dimension, if the upper bound capacity (< 2l) is fixed, the congestion of a channel can be represented by the capacity. In the filling phase of Maze router, we do not care about the congestion of the channel, thus we find the minimum path as possible. In the retracing phase of Maze router, we try to lower the average congestion of each channel. Under such consideration, a path may bend several times to avoid the congested channels. After the above process, the routing cost of R(j) (described in Section 2) is obtained.

For a routing wire in PCB routing, it may connect multi-pin nets. To route a multi-pin net, we may divide these pins into several two-terminal nets. First, we pick up two of terminal pins in the pin set heuristically. In our algorithm, the I/O pins of the package are always picked up first. We use a Maze router to route these two pins. Then the channels along the found path are all marked as “T”. The third pin is then routed until it reaches the “T” channel in the filling phase of Maze router. After each pin of the same net is routed, the channels along the new found path are also marked. As shown in Figure 6, the pin A and B are first to be routed and then pin C is later.

![Routing a multi-pin net](image)

Figure 6. Routing a multi-pin net.

3.3 Pin Assignment

Many studies [9-10] on the pin assignment problem of PCB have been presented. But they do not consider the cost estimation methods of the package and PCB routing. In this paper, we employ a simulated annealing like technique [5] to make the solutions of package pin assignment and estimate their costs.

The algorithm of pin assignment takes the results of PCB design and package design as input. The I/O pads as well as the I/O pins of a package are divided into four regions as shown in Figure 3. First, we define three types of pin exchange, where these types represent the chosen pins from one region are exchanged with those in another region. We calculate their costs and compare them to choose the best one.

The choice of the pins in each region is according to the Bonus of each net. The Bonus of each net is defined as the cost change before and after the corresponding pin changes its assignment. We sort the Bonus in decreasing order, and then pick top $M$ (positive number) pins in each region. Among the three types of exchanging, the pins are exchanged one-to-one, i.e., a pin with the highest (lowest) Bonus in the one region is exchanged with that in another region. The three costs are calculated after all the three types of pin exchange are made. If the lowest cost is lower than the original one, we pick it as the new original cost in the next repeated step.

Finally, the whole process will be terminated and then our object function (described in Section 2) is also satisfied. Therefore, a better solution of pin assignment of the package is obtained.
4. Experimental Results

The routing system was implemented on a Pentium II-300 PC in Java language running Windows-98 operating system. Since no this type of benchmarks are available from the literature, a set of testing examples created by the authors are used to evaluate the efficiency of our router.

In our experiments, there having nets in some of PCB circuits can’t be completely routed after the process of our pin assignment because the limited four-layer is available routing on PCB circuits. As can be seen in Table 1, routings produced by our pin assignment router with PCB routing consideration have on the average 30% better than the routings of that without considering PCB routing. The first-layer routing result of a PCB (Circuit_1) is plotted in Figure 8.

5. Conclusions

An innovative windows-based routing tool for the codegen between PGA package routing and PCB routing has been presented in this paper. A simulated-annealing based algorithm is employed to solve the problem of pin assignment of the package. As the results of pin assignment depend on the routing cost estimation methods of the package and PCB. Experiments show that careful pin assignment of the package indeed eases the routing of a PGA package and improves the nets routability of a PCB.

The algorithms of routing cost estimation of the package and PCB are implemented in software components and then they can be run concurrently through the mechanism of remote method invoking (RMI) of Java language.

References

![Figure 8. The wiring result on layer 1 of PCB (Circuit_1) after pin assignment.](image)

<table>
<thead>
<tr>
<th>PCB name</th>
<th>number of pins</th>
<th>number of nets</th>
<th>number of layers</th>
<th>unrouted nets</th>
<th>improved percentage</th>
<th>cpu-time (second)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Circuit_1</td>
<td>464</td>
<td>220</td>
<td>4</td>
<td>6</td>
<td>2.72%</td>
<td>100</td>
</tr>
<tr>
<td>Circuit_2</td>
<td>464</td>
<td>220</td>
<td>4</td>
<td>114</td>
<td>9</td>
<td>63</td>
</tr>
<tr>
<td>Circuit_3</td>
<td>528</td>
<td>220</td>
<td>4</td>
<td>114</td>
<td>9</td>
<td>51.82%</td>
</tr>
<tr>
<td>Circuit_4</td>
<td>592</td>
<td>229</td>
<td>4</td>
<td>78</td>
<td>2</td>
<td>33.19%</td>
</tr>
<tr>
<td>Circuit_5</td>
<td>592</td>
<td>229</td>
<td>4</td>
<td>8</td>
<td>1</td>
<td>3.06%</td>
</tr>
<tr>
<td>Circuit_6</td>
<td>592</td>
<td>229</td>
<td>4</td>
<td>110</td>
<td>13</td>
<td>42.36%</td>
</tr>
</tbody>
</table>

Note: the w-p and w+p are represented without and with pin assignment, respectively.