An efficient RDL routing for flip-chip designs

Tu-Hsiung Tsai and Hung-Ming Chen, National Chiao Tung University, and Hung-Chun Li and Shi-Hao Chen, Global Unichip Corp - August 20, 2013

Engineers use a redistribution layer (RDL) in flip-chip designs to redistribute I/O pads to bump pads without changing the I/O pad placement. However, traditional routing capacity may be insufficient to handle sizable designs, in which the RDL may be very congested and especially when there is a less-than-optimal I/O-bump assignment. As a result, routing may not be completed within a single layer even with manual routing.

As demand for more input/output (I/O) increases, traditional wire bond packaging may not effectively support thousands of I/Os. Flip-chip assembly is commonly used in place of wire bond because it reduces chip area while supporting many more I/Os. It also greatly reduces inductance, allows high-speed signals, and possess better heat conductivity properties. The flip-chip ball grid array (FCBGA) is also growing in popularity as an alternative methodology for high I/O count chips.

Figure 1. Flip chip cross section: Signal traces travel through three interfaces including RDLs.

The redistribution layer (RDL) is the interface between chip and package for flip-chip assembly (Fig. 1). An RDL is an extra metal layer consisting of wiring on top of core metals that makes the I/O pads of the die available for bonding out other locations such as bump pads. Bumps are usually placed in a grid pattern and each one is molded with two pads (one on the top and one on the bottom) that are then attached to the RDL and package substrate respectively. The RDL, therefore, serves as the layer connecting I/O pads and bump pads.
Flip Chip Structures & Pad Assignments

Previous research has identified two flip-chip structures and two pad assignment methods, both shown in Fig. 2. Free-assignment (FA) and pre-assignment (PA) are two pad assignment methods. Peripheral-I/O (PI/O) and area-I/O (AI/O) are two flip-chip structures.

The two pad assignment methods are defined by whether or not the mapping between bump pads and I/O pads has been given as input. For FA problems, each I/O pad is free to assign to any bump pad, so assignment is considered together with routing. For PA problems, each I/O pad must connect with a specified bump pad, thus solving complex crossing issues. PA problems are more difficult than FA problems but, at the same time, are more convenient for designers.

Two flip-chip structures represent patterns of I/O placement. AI/O and PI/O challenges place I/Os in the central area and on the periphery of die respectively. PI/O is more popular today because of its simplicity and low design cost, although AI/O theoretically provides better performance.

An example of PI/O is shown in Fig. 3. The green rectangles on the periphery are I/O pads. The red and yellow circles are power and ground bumps, while the blue circles are signal bumps. Some power/ground bumps, located in the center of die, are sorted by mesh type. Signal bumps are sorted by grid type.
Figure 3. RDL top view. Bump pads in a grid pattern and I/O pads on the periphery.

All aforementioned works focus on single-layer routing. They restrict routes to one metal layer on which every net must be routed. The common objective is wire-length minimization. The optimization schemes are done under a prerequisite that routability is 100%. This has proven to be very successful for each type of RDL routing problem, providing that a single layer solution exists.

**Practical RDL Routing**

Both RDL routing and bump assignment are additional implementation tasks to migrate designs from wire bonding to flip-chip. Bump assignment means assigning each bump to a specific I/O pad. Since I/O pads for most designs are on the die periphery, the fly-lines and signal routing look like nets running from the chip's center to its boundary.

Figure 3 shows a real scale design example using two RDL layers. Metal 10 (M10) and Metal 9 (M9) route all signal nets and implement the power/ground (PG) mesh and power routing respectively. Usually, there are a large number of signal nets that require routing. Bump pads feature large areas and are treated as obstacles in the routing stage.
Fig. 4(a) shows an example of a congested RDL, where six nets, netA, netB, ..., netF, are shown in fly-lines. Such a design is so congested that 100% routability can't be achieved on single layer (ex: M10). One solution is to increase the RDL (ex: M10) area. This is equivalent to increasing the die-size, as shown in Fig. 4(b). Another solution is to add an extra RDL layer (ex: M11), as shown in Fig. 4(c). While practical from an engineering standpoint, many times neither solution is acceptable from a cost standpoint.

An Alternative Framework
A more practical option is a concept called Pseudo Single-Layer Routing that appropriates a small area from an already-existing metal layer (ex: M9). This is practical and cost-effective, provided that the appropriated area is used for less performance-critical functions.

In Fig. 4(d), some areas of M9 (the pink area) are appropriated to complete routing. Here we assume that the area between a boundary track (the dotted grey line) and the border of the die is used to assist routing. Pseudo single-layer routing avoids cost problems and relieves congested routing. While previous work focuses on single-layer routing, pseudo single-layer routing uses two-layer routing within a small area.

This is applicable to RDL because M9 is traditionally used to connect PG from I/O pads and because the most important M9 function is to evenly distribute power to every logic gate in the core. Consequently, M9's peripheral area is relatively less important than the central area enabling signal nets to share M9 peripheral area with PG nets.
Figure 5. First and second RDLs are on layer 9 and layer 10 respectively. PG meshes are placed in $M_{\text{inner}}^L9$. Routable regions are $M_{\text{outer}}^L10 \cup M_{\text{inner}}^L10 \cup M_{\text{outer}}^L9$.

The problem of RDL routing is to connect net $N_i$ between the bump pad $B_i$ and the input/output pad $O_i$. First and second RDLs are M9 and M10 respectively, as shown in Fig. 5. We name the area as inner/outer region with respect to the boundary track. The entire RDL is partitioned into four sectors: $M_{\text{inner}}^L9$, $M_{\text{outer}}^L9$, $M_{\text{inner}}^L10$, and $M_{\text{outer}}^L10$.

Term Definitions

- Routable region (pseudo single layer): $M_{\text{outer}}^L10 \cup M_{\text{inner}}^L10 \cup M_{\text{outer}}^L9$
- Outer region: $M_{\text{outer}}^L10 \cup M_{\text{outer}}^L9$
- Inner region: $M_{\text{inner}}^L9 \cup M_{\text{inner}}^L10$

The pseudo single-layer RDL routing problem is to physically connect $B_i$ and $O_i$ of net $N_i$ in the routable region and to minimize the area of inner region. This also means that the boundary track is not fixed. The solution is to determine the location of boundary track.

Our pseudo single-layer routing algorithm is defined by four steps: Step 1 is regional layer allocation, movable pin assignment, and layout abstraction. Step 2 is to route a net from a bump pad to a pin. Step 3 is to decide which track to use. Step 4 is to route from the I/O pad to the pin. The flow is shown with a simple example in Fig. 6 for making movable pin assignments. Step 1 is the most important step. Good movable pin assignment minimizes the RDL tracks.
Figure 6. This simple example explains the routing flow: (a) Regional layer allocation, assignment of movable pins, and layout abstraction. Steps (b) to (d) illustrates which track to use and routing from I/O pad to the pin using channel route. (e) Represents the routing results remapped into the original layout.
Figure 7. Two versions of movable pin assignment: (a) Movable pin assignment ordered from single side. (b) Movable pin assignment using a bump pin selection algorithm. The bump pin selection algorithm achieves routing results with fewer tracks.

There are two movable pin assignment methodologies shown in Fig. 7. The first version makes a movable pin assignment for each bump row from the same side so that the pin sequence and bump sequence are the same. This methodology quickly makes the movable pin assignments but the disadvantage is that the sequence is fixed by the bump row. If the bump sequence is not ideal, the result is a large number of routing tracks.

The second and recommended methodology is a pin selection algorithm as shown in Fig. 8. Step 1 generates all the possible movable pin sequence and the routes from bumps to the pin sequence without any cross nets. Step 2 consists of choosing a movable pin sequence from the first step with the least number of crosses. The bump selection algorithm ensures bump-to-pin connections without any crosses and pin-to-pad with the least number of crosses. After using the bump select algorithm, a channel routing algorithm finishes the route from pins to I/O pads, determines the number of tracks and distributes track resource. Finally, the results are remapped to the original layout and the RDL routing with the pseudo single layer is finished.
Figure 8. Bump selection algorithm. (a) Generate the movable pin sequence. (b) Choose the pin sequence that minimizes crossing between movable pin and I/O pads.

Verifying Effectiveness

Our framework has been implemented on a sizable commercial project. First, the chip was divided into four sectors: W, N, E, and S. Each sector contained more than 100 signal bumps. For each sector, our router generated the results and dumped command scripts in less than five seconds. By sourcing these scripts in Encounter Digital Implementation (EDI), wires are physically routed. This can also be obtained by any pin-to-pin router since all the pin positions are allocated. Design Rule Checking (DRC) determined that all results are clean. The routing results are shown in Fig.6 and Fig. 7, and summarized in Table I, where fcroute is flip-chip router and p2proute is point-to-point router in defined EDI. Due to a non-disclosure agreement, only partial results are shown.

<table>
<thead>
<tr>
<th>#</th>
<th>List</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>12 21</td>
</tr>
<tr>
<td>3</td>
<td>123 312 321 213</td>
</tr>
</tbody>
</table>

TABLE I Summary Routing Results
### Conclusions

We introduced a methodology for RDL routing on a pseudo single-layer that can be used in situations when congestion is so severe that even manual routing cannot find a single-layer solution. Pseudo single-layer routing provides a viable alternative to adding an extra metal layer or increasing the die-size. The keys to success are regional layer allocation, movable pin assignment and layout abstraction. These techniques transform the RDL routing problem into a classic channel routing problem. One hundred percent of the wires are routed and the area of two layer routing is minimized.

### REFERENCES


Tu-Hsiung Tsai received his B.S. degree in electronic engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2006. He is currently working toward a Ph.D. degree in the Department of Electronic Engineering. His interests include the placement of digital circuits, physical design methods, and co-design flow of system design.

Hung-Ming Chen received his B.S. degree in computer science and information engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1993. He received his M.S. and Ph.D. degrees in computer sciences from University of Texas at Austin in 1998 and 2003, respectively. He is currently an Associate Professor in the Department of Electronics Engineering at National Chiao Tung University. Dr. Chen has served as a technical program committee member for ACM/IEEE ASP-DAC, IEEE SOCC, VLSI-DAT, and ACM ISPD. His research interests include physical design automation in digital and analog circuits, beyond-die integration (off-chip EDA), and 3D IC design methodology.

Hung-Chun Li worked for Cadence Design as an R&D Architect in the IC Digital Design Group and has held other technical responsibilities with Broadcom’s Network Switching Division, TSMC’s Design Service Division and ISSI’s Nonvolatile Memory Division. He is currently a Director of the Design Service Division, GUC (Global Unichip Corporation), Hsinchu, Taiwan. He leads the GUC’s design flow R&D team that develops IC design methodology for advanced technology nodes (40nm/28nm).

Shi-Hao Chen received his B.S. and M.S. degrees from the Department of Electronic Engineering, Chung Yuan Christian University, Chungli, Taiwan, in 1996 and 1998, respectively, and his Ph.D. degree from the Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan,
in 2012. He is currently a Deputy Director, Design Service Division, GUC (Global Unichip Corporation), Hsinchu, Taiwan. His research interests include design flow automation, physical design, and low power design methodologies.