Design Automation for Wire-bond Package Die Orientation and Placement

Geng-Shen Lin Yi-Yu Liu

Department of Computer Science and Information Engineering,
National Taiwan University of Science and Technology, Taipei City, Taiwan, R.O.C.
B10615049@mail.ntust.edu.tw yyliu@mail.ntust.edu.tw

Abstract— In IC design and manufacture industry, most chips must go through the process of packaging and testing before mounting on the PCB board. Owing to complex package design rules, the layouts of package substrate are manually designed by engineers. In order to reduce layout design time of package substrates, we develop a floorplanning methodology to automate the process in early design stage. Given the netlist, die pad and substrate bump ball locations, both die orientation and location can be simultaneously determined on the package substrate taking wire length into account.

I. Introduction

In the IC manufacturing process, each die has to go through the packaging assembly process as an encapsulated chip that can be mounted on the printed circuit board. In packaging design flow, the assembly house customers provides design specifications, such as the dimensions of a die and the corresponding substrates, the I/O pin locations, and the netlists which specify the interconnects between the die and the substrate pins. Once the specifications are fixed, package substrate engineer would design the substrate layout which include die placement and substrate routing so that all power and signal nets could be properly connected. Currently, the placed die orientation and location, the position of wire-bond fingers, and the wire routing in the package substrate are manually designed by package layout engineers.

During package substrate design stage, the die orientation and placement is the first design issue. This is because the position of the die must be decided before design wire bonding and substrate routing. With the advances in packaging technology and the increasing performance requirements, there might be multiple dies on a chip substrate. As a result, the complexity and difficulty of manual wire-bond finger placement and substrate routing are drastically increased. In order to make good use of package routing resource, the position of the die is very important for the subsequent package design. However, there is a missing design optimization gap between substrate die floorplanning and routing. In this paper, we propose a quadratic wire-length model for substrate die floorplanning taking die pad and solder ball netlist into account. With the quadratic objective function, we are capable of obtaining the best placement location and the rotation angle of the die. The quadratic solver minimizes the connection distance of power and signal from die pads to bump balls and improves the performance of the chip. In addition, the required substrate resource of wire-bond fingers and routing is reduced to ease the difficulty of design packaging and to shorten the time costs.

The rest of this paper is organized as follows. Section II gives the fundamental knowledge of related packaging technology. Section III presents the methodology of our research and mathematical programming. The experimental results are drawn in Section IV. Finally, Section V concludes our work.

II. Basic of packaging

A. Wire-bond Packaging Technology

Wire bonding is a design style to interconnect the I/O pins between an integrated circuit (IC) or other semiconductor device and the corresponding package substrate during semiconductor packaging stage. Wire-bond packaging is a mature technology in the IC packaging industry's manufacturing process. Owing to the most cost-effectiveness and flexible connection technology, wire bonding design style contributes to the majority ratio of semiconductor packages. [1, 2] Figure 1 is the structure of a typical wire-bond package comprises of a die, bonding wires, substrate, and solder ball grid array (BGA).

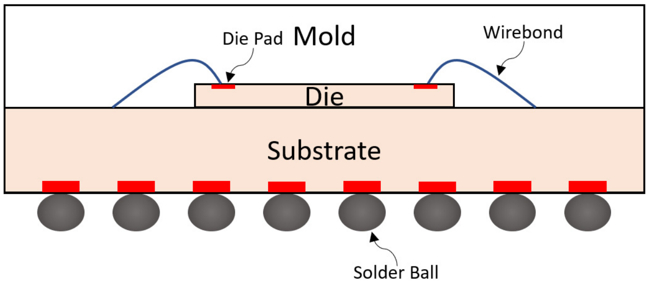
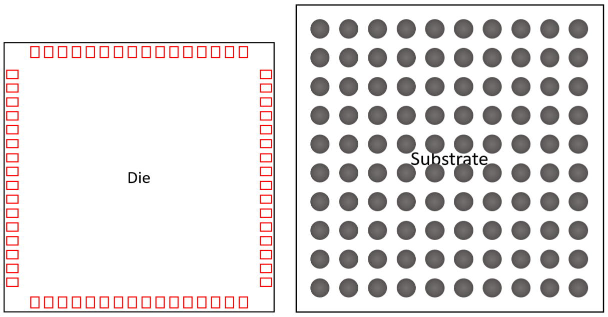


Fig. 1. The structure of a wire-bond package.

Figure 2(a) presents a model of a wire-bond die and die pads which are typically located at the periphery of the die. [3] Figure 2(b) presents a model of a package substrate and the corresponding BGA solder balls. To interconnect the die to outer devices, the power/ground and signals of the die are transmitted from die pad to package substrate via bonding wires followed by substrate routing wires to transmit the power/ground



(a) A periphery die model (b) A substrate and BGA model

Fig. 2. The component model of a wire-bond package design.

and signals to underlying solder balls. Finally, molding compound encapsulates the bonding wires and the dies to provide physical protection and to avoid contamination.

III. Mathematical Programming Modeling

A. Research Method

In IC packaging design, there are 2 important factors that would affect the position and the rotation angle of a die:

- The netlists of die pads to solder balls.
- The netlists of die pads to the other die pads.

Therefore, we convert these netlist relations to mathematical formulas. We sum up all the quadratic distance of netlists as the objective function, and set the position and rotation angle of die as variables. Finally, we use an optimization solver to derive the minimum objective function.

B. Definition

- Solder Ball:
  Assume there are m solder balls on bottom-side of the chip package. The location of each BGA ball is given and denoted as $(B_{ix},\ B_{iy})$ in Cartesian coordinates system.
- Die:
  - Assume there are n dies in a package where n is a positive number. Each die has three variables: the center of the die, denoted as $(c_{ix},\ c_{iy})$, in Cartesian coordinate system, and the rotation angle, denoted as α_i, in Polar coordinate system.
  - The die pads, also known as bond pads, are the entrance that the die communicate with outside world. As the die orientation and location changed, the die pad positions and rotation angles changed correspondingly. In order to indicate the rotation angle of die α_i in the position of die pad. We denote the j-th die pad of the i-th die as $(r_{i,j}, \theta_{i,j} + \alpha_i)$ in Polar coordinate system. Consequently, each die pad position can be easily derived by transforming the Polar coordinate system to the Cartesian coordinate system:

$$(P_{i,j\ x}, P_{i,j\ y}) = \\ [c_{ix} + r_{i.j}cos(\theta_{i,j} + \alpha_i),\ c_{iy} + r_{i.j}sin(\theta_{i,j} + \alpha_i)] \quad (1)$$

- Netlist:
  Since there are die-to-substrate and die-to-die connections, we separate the objective functions into two parts. In order to obtain an overall convex function for optimization solver to start with, the quadratic pin-to-pin straight line distance is used in our objective function. The respective mathematical formulas are as follows:
  - Netlist of die pads to solder balls
    The square of straight line distance of i-th solder ball and k-th die pad of the j-th die:

$$(B_{ix} - P_{j,k\ x})^2 + (B_{iy} - P_{j,k\ y})^2 \quad (2)$$

  - Netlist of die pads to other die pads
    The square straight line distance of the j-th die pad of the i-th die and the l-th die pad of the k-th die:

$$(P_{i,j\ x} - P_{k,l\ x})^2 + (P_{i,j\ y} - P_{k,l\ y})^2 \quad (3)$$

C. Mathematical Programming Function

In this paper, we adopt the concept of analytical placement [4, 5] to determine the die orientation and placement location. A quadratic die orientation and placement objective function, denoted as QOP, is developed for wire length optimization. The reason for this is the quadratic objective function has better results and can highlight the importance of long distance connections by magnifying their distance. The optimization solver could subsequently minimize such long-wire connections.

- Objective function

$$min \sum^{m} \left[(B_{ix} - P_{j,k\ x})^2 + (B_{iy} - P_{j,k\ y})^2\right] + \\ \sum^{n} \left[(P_{i,j\ x} - P_{k,l\ x})^2 + (P_{i,j\ y} - P_{k,l\ y})^2\right] \quad (4)$$

- Subject To

$$-\pi \leq \alpha \leq \pi$$

IV. Experimental Results

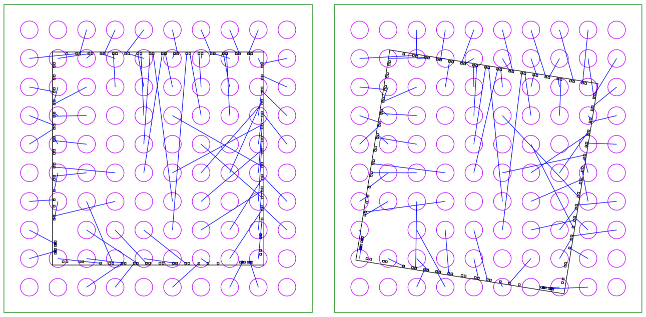
We develop our formulation using C++ programming language on a Linux Ubuntu18.04 workstation with 2.50 GHz Intel i5-7200U CPU and 8 GB memory. Our QOP formulation tool is supplemented by Gtk and Cairo libraries [6, 7] for user-interface design and the overall substrate and die drawing. We use interior point algorithm of "fmincon" function in MATLAB R2019b to obtain the optimal QOP result. [8, 9]

TABLE I
WIRE LENGTH REDUCTION & DIE ROTATION OF SINGLE DIE DESIGNS.

Test	Manhattan Distance			Euclidean Distance			$(\Delta x, \Delta y)$	Rotation	Run Time
Case	*Manual*	*QOP*	Ratio	*Manual*	*QOP*	Ratio		Angle	(sec)
Ind1	6.134E+04	6.034E+04	1.63%	4.880E+04	4.664E+04	4.43%	(383.8, 93.6)	-4.40°	2.23
Ind2	9.629E+04	8.156E+04	15.29%	7.770E+04	6.754E+04	13.07%	(-254.6, -303.9)	-9.18°	1.74
Ind3	9.548E+05	9.530E+05	0.19%	7.471E+05	7.459E+05	0.16%	(2.9, 118.5)	-2.60°	3.68
Ind4	3.007E+05	2.914E+05	3.10%	2.319E+05	2.276E+05	1.82%	(41.7, 177.9)	-5.05°	3.00
Ind5	1.698E+06	1.685E+06	0.79%	1.287E+06	1.276E+06	0.86%	(-207.9, 146.9)	7.80°	3.11
Ind6	1.880E+06	1.879E+06	0.04%	1.438E+06	1.436E+06	0.16%	(141.6, 236.8)	-1.46°	3.66
Average			3.51%			3.42%			

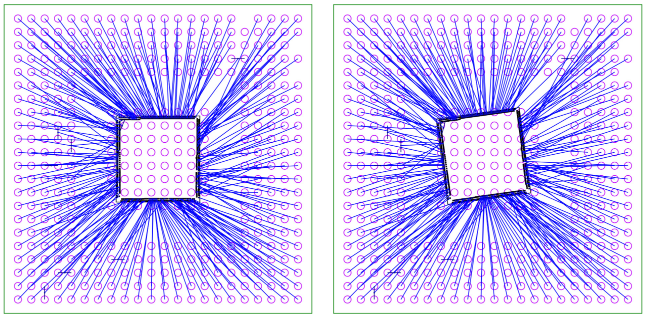
There are six single-die industrial designs in our experiments. There are two kinds of connection relationship, the power and signal connections. Since most of the nets are one-to-one signal nets and the number of power nets are relatively small with multiple pins, in our experiment, we only consider the signal nets. The power connection is not considered. Notice that the signal connection of die pads to solder balls consists of two parts, the bonding wires and the substrate routing wires. Neither bonding wire nor substrate routing wire is straight line. Consequently, we report both Manhattan distance and Euclidean distance after the die orientation and placement are determined.

Table I lists the results of the single die designs and only considers the signal connections relationship. The column of *Manual* is the floorplan designed by substrate layout engineers with the die placed on the center of the package substrate for simplicity. The column of *QOP* is the floorplan derived by using our proposed QOP objective function. The calculation of the *Ratio* column is $[(Manual - QOP) \div Manual] \times 100\%$. In single-die designs, the total distance of the signal connection relations is reduced by using our QOP function. Figure 3 and Figure 4 compare the partial IC packaging design of some single die test data, in which Figure 3(a) and Figure 4(a) are designed by package substrate layout engineers, and the position and rotation angle of die in Figure 3(b) and Figure 4(b) are obtained by QOP function.



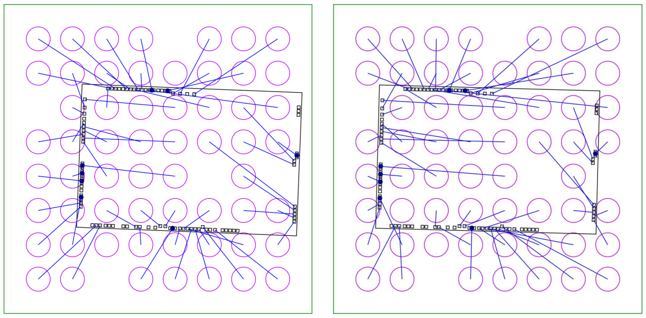
(a) Manual Design of Ind2 (b) QOP Design of Ind2

Fig. 3. The floorplan of the Industrial2.



(a) Manual Design of Ind5 (b) QOP Design of Ind5

Fig. 4. The floorplan of the Industrial5.



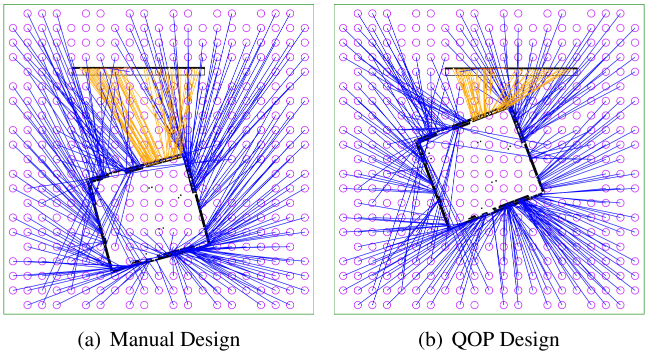
(a) QOP with no constraint (b) QOP with boundary constraint

Fig. 5. Boundary constraint issue in Industrial1.

In Industrial2, both the total Manhattan distance and Euclidean Distance are reduced by more than 10%. This is because in the manual design, the connections at the right side and below of the die are oblique. Using QOP, these connections are made nearly horizontal and vertical to dramatically reduce the wire length. The column of $(\Delta x, \Delta y)$ is the difference of the manual die center and optimized die center. The column of Rotation Angle is the die's rotation angle, and it could be positive or negative. The rotation of die show that if we want to reduce the line length even more, we need to rotate a little bit the angle of the die.

In the partial package design of single die, we can see that the die of some design are too close to the substrate boundary, such as figure 5(a), which may cause non-compliance with the package design rules. The current design can be constrained by limiting the range of optimally feasible solutions by MATLAB optimization solver. Figure 5(b) shows the results with boundary limits in the design of Industrial1.

Our objective function is not limited to one die packaging design. Figure 6(a) is a sample design of multi-die packaging design was designed by industry engineers. The design contains two dies, die1 and die2. The die1 is at the bottom of the substrate, and the die2 is at the top of the substrate. Figure 6 compares the partial design of the package from the multi-die test data. The die1 and die2 positions and rotation angles of Figure 6(b) are obtained by solving the target function. The result of the Figure 6 shows that our function can also be used in multi-die.


(a) Manual Design

(b) QOP Design

Fig. 6. The floorplan of an industrial multi-die design.

V. Conclusion and Future Work

We have developed a floorplanning methodology to automate the early-stage package substrate design. We propose a QOP function to optimize the die orientation and placement simultaneously. The optimized floorplan effectively reduces the overall wire length, which can also reduce the required substrate resource for subsequent wire-bond finger placement and substrate routing. If the die is too close to the substrate boundary, additional formulations are used to avoid design rule violations. Besides of the near-boundary and die overlapping design issues, the multiple-pin power signal are not considered yet since the properties of the power nets are different from those of the signal nets. We plan to discuss these issues in subsequent studies.

References

[1] *Wire bounding - Wikipedia*, Available: `https://en.wikipedia.org/wiki/Wire_bonding`.

[2] *PACKAGING: Wirebond to wafer level? No problem |EE Times*, Available: https://www.eetimes.com/packaging-wirebond-to-wafer-level-no-problem/.

[3] Farhang Yazdani, *Foundations of Heterogeneous Integration: An Industry-Based, 2.5D/3D Pathfinding and Co-Design Approach*, San Jose, CA: Springer, Cham, 2018.

[4] G. Sigl, K. Doll and F. M. Johannes, *Analytical placement: A Linear or Quadratic Objective Function?*, Proc. DAC, pp427-432, 1991.

[5] N. Viswanathan and C. C. -N. Chu, *FastPlace: Efficient Analytical Placement using Cell Shifting, Iterative Local Refinement and a Hybrid Net Model*, ISPD 2004.

[6] *The GTK Project - A free and open-source cross-platform widget toolkit*, Available: `https://www.gtk.org/`.

[7] *Cairo Tutorial*, Available: `https://www.cairographics.org/tutorial/`.

[8] *R2019b - Updates to the MATLAB and Simulink product families - MATLAB & Simulink*, Available: `https://www.mathworks.com/products/new_products/release2019b.html`.

[9] *Find minimum of constrained nonlinear multivariable function - MATLAB fmincon*, Available: `https://www.mathworks.com/help/optim/ug/fmincon.html`.