Timing-Constrained I/O Buffer Placement for Flip-Chip Designs

Zhi-Wei Chen[1] and Jin-Tai Yan[2]
[1]*College of Engineering,* [2]*Department of Computer Science and Information Engineering*
Chung-Hua University, Hsinchu, Taiwan, R.O.C
{m09402003, yan}@chu.edu.tw

***Abstract*—Due to inappropriate assignment of bump pads or improper placement of I/O buffers, the configured delays of I/O signals may not satisfy the timing requirement inside die core. In this paper, the problem of timing-constrained I/O buffer placement in an area-IO flip-chip design is firstly formulated. Furthermore, an efficient two-phase approach is proposed to place I/O buffers onto feasible buffer locations between I/O pins and bump pads with the consideration of the timing constraints. Compared with Peng's SA-based approach[7], with no timing constraint, our approach can reduce 71.82% of total wirelength and 55.74% of the maximum delay for 7 tested cases on the average. Under the given timing constraints, our result obtains higher timing-constrained satisfaction ratio(TCSR) than the SA-based approach[7].**

I. INTRODUCTION

With the increase of circuit complexity and the decrease of feature size, the dramatic demand for I/O counts becomes a significant issue in VLSI designs. An advanced packaging technology, flip-chip package[1], is introduced to meet the higher integration density and the larger I/O counts in modern VLSI designs. In general, a flip-chip design as shown in Fig. 1 describes the methodology of electrically connecting the die core to the package carrier based on the technology of the flip-chip package. Because of the reduced signal inductance and package footprint, the flip-chip technology has been widely used in high-performance circuit designs. However, the placement of I/O buffers is not exactly mapped onto the locations of bump pads in an area-I/O flip-chip design. Hence, an extra metal layer, *Re-Distribution Layer*(RDL), is used to redistribute the connections from I/O buffers on the die to the bump pads on the RDL.

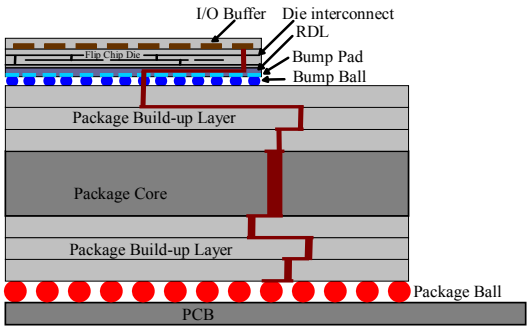

Fig. 1. Connection of IO signal in an area-I/O flip-chip design

In general, the performance in a chip design is evaluated by the timing effect inside the die core and most of works are concentrated on the timing delay inside the die core for the performance optimization. In an area-IO flip-chip design, I/O buffers can be placed onto white space inside the die to bridge I/O pins in circuit blocks to bump pads on RDL. After optimizing the performance inside the die core, the configured delay of I/O signals may not satisfy the timing requirement inside the die core due to inappropriate assignment of bump pads or improper placement of IO buffers. As a result, the final performance of the chip may be dominated by the timing effect at package stage.

The I/O buffer placement problem in conventional wire-bond designs[2, 3, 4, 5] has been extensively studied. However, these approaches are inapplicable in area-IO flip-chip designs. Recently, many works[6, 7, 8, 9, 10] on I/O buffer placement have been published in area-IO flip-chip designs. Hsieh et. al.[6] proposed a constructive approach with considering the total path delay and signal skew to place circuit blocks and I/O buffers simultaneously. A SA-based approach with the similar consideration was further introduced in [7]. However, these two approaches are concentrated on the skew minimization on the basis of wirelength. Xiong et al. [8] studied the I/O buffer placement with considering signal skew and other constraints. However, it is assumed that I/O buffers are placed at fixed locations for usage. Lai et. al.[9] considered the additional differential pairs into the IO buffer placement problem. Wang et al. [10] proposed an I/O buffer placement approach to take the routability issue for escape routing into consideration. However, all the works only considered how to place the circuit blocks and I/O buffers at the same time and the timing constraint did not be considered for I/O buffer placement.

For I/O buffer placement in an area-IO flip-chip design, the problem of timing-constrained I/O buffer placement in an area-IO flip-chip design is firstly formulated. Furthermore, an efficient two-phase approach is proposed to assign necessary I/O buffers for all the I/O signals such that the timing constraint of each I/O signal is satisfied. In the first phase, all I/O pins are assigned onto the feasible bump pads under the consideration of timing constraints. Furthermore, the I/O buffers are inserted to complete all the I/O signals from I/O pins to bump pads. Compared with Peng's SA-based approach[7], with no timing constraint, our approach can reduce 71.82% of total wirelength and 55.74% of the maximum delay for 7 tested cases on the average. Under the given timing constraints, our result obtains higher timing-constrained satisfaction ratio(TCSR) than the SA-based approach[7].

978-3-9810801-7-9/DATE11/©2011 EDAA

II. Problem Formulation

According to the design specification in an area-IO flip-chip design, I/O buffers can be placed on white space in a floorplan plane and bump pads must be placed at pre-defined locations on RDL. It is assumed that a set of blocks with some I/O pins is located in a floorplan plane and an array of bump pads is located on RDL. Generally speaking, any I/O pin must be assigned onto a unique bump pad and a necessary I/O buffer must be required to bridge the connection between the I/O pin and its corresponding bump pad. To assign an I/O buffer onto an available space in a floorplan plane to connect each I/O pin onto the corresponding bump pad, two essential constraints, *timing constraint* and *non-overlapping constraint*, must be satisfied on RDL.

In an area-IO flip-chip design, I/O signal is defined as an I/O connection to connect an I/O pin to a bump pad via an I/O buffer as shown in Fig. 2. In general, the delay of any I/O signal is constrained by a given timing delay of its I/O pin and divided into three parts: The first one is the delay between the I/O pin and the I/O buffer, the second part is the intrinsic delay of the I/O buffer, and the final part is the delay between the I/O buffer and the bump pad. Besides the intrinsic delay of the I/O buffer, the remaining delays are dependent on the lengths of the two connections. To satisfy the given timing delay, the *timing constraint* of each I/O connection must be considered in I/O buffer placement.

On the other hand, the overlapping relation between any I/O buffer and any block is not allowed. Hence, I/O buffers are only placed on white space in a floorplan plane. In general, white spaces in a floorplan plane can be obtained by performing the sweeping line algorithm[11]. Furthermore, all available locations for I/O buffers can be achieved according to the size of an I/O buffer and the size of each white space in the plane. To satisfy the valid locations of I/O buffers, the *non-overlapping constraint* must be must be considered in I/O buffer placement.

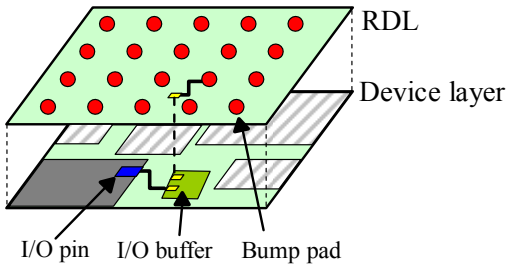

Fig. 2. I/O connection from an I/O pin to a bump pad via an I/O buffer

Given a set of m blocks, M = $\{B_1, B_2,...,B_m\}$, and a set of s buffer locations, S = $\{s_1, s_2,...,s_s\}$, in a floorplan plane, a set of n I/O pins, P = $\{p_1, p_2,...,p_n\}$, on the blocks, an array of r bump pads, B = $\{b_1, b_2,...,b_r\}$, on the given RDL and the timing constraint, t_i, each I/O pin, p_i, $1 \le i \le n$, the timing-constrained I/O buffer placement is to map each I/O pin onto a unique bump pad to construct an I/O signal and assign an I/O buffer on a feasible location for each I/O signal in the given floorplan plane under the timing and non-overlapping constraints. As illustrated in Fig. 3(a), given a set of 7 blocks, B_1, B_2,..., and B_7, and a set of 13 buffer locations, s_1, s_2,..., and s_{13}, in a floorplan plane, a set of 9 I/O pins, p_1, p_2,..., and p_9, on the blocks, a set of 9 bump pads, b_1, b_2,…, and b_9, on the given RDL and the timing constraint of each I/O pin, the 9 resultant I/O signals, (p_1, s_6, b_3), (p_2, s_2, b_2), (p_3, s_1, b_1), (p_4, s_8, b_5), (p_5, s_3, b_4), (p_6, s_9, b_6), (p_7, s_{13}, b_9), (p_8, s_{12}, b_7) and (p_9, s_{11}, b_8), with the location assignment of 9 I/O buffers can be obtained in timing-constrained I/O buffer placement and shown in Fig. 3(b).

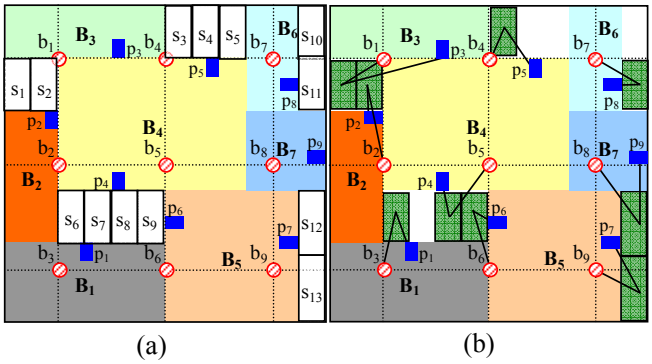

Fig. 3. Timing-constrained I/O buffer placement in an area-IO flip-chip design

III. Timing-Constrained I/O Buffer Placement

For timing-constrained I/O buffer placement in an area-IO flip-chip design, an efficient approach is proposed to assign I/O pins onto feasible bump pads to construct I/O signals and assign necessary I/O buffers onto feasible buffer locations for all I/O signals. Basically, the placement process can be divided into three main steps: *Extraction of timing-constrained routing length*, *I/O signal assignment* and *Timing-constrained I/O buffer placement* and the design flow for the proposed approach is illustrated in Fig. 4.

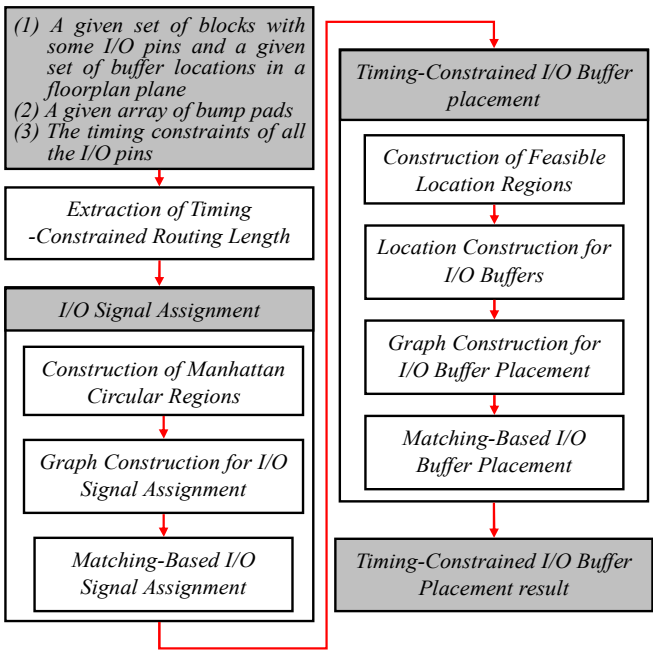

Fig. 4. Design flow of timing-constrained I/O buffer placement

A. *Extraction of Timing-Constrained Routing Length*

For any I/O pin in a floorplan plane, its timing constraint determines all the feasible bump pads to construct an I/O signal. In other words, the location of any valid bump pad must connect from the I/O pin to the pad to satisfy the given timing constraint. To find all the valid bump pads under the timing constraint, the maximal routing length of the I/O pin must be extracted from the given timing constraint.

Basically, an I/O connection can be modeled as a RC circuit and the delay of the RC circuit can be computed by using Elmore delay model. In Elmore delay model, the delay, t, of any I/O connection with length, l, can be obtained as

$$t = \left(\frac{r_0 c_0}{2}\right) l^2 + (Rc_0 + r_0 C) l + RC,$$

where R is the driver strength, C is the driving load, r_0 is the unit resistance and c_0 is the unit capacitance.

It is assumed that t_{max} is the maximum timing delay of a given I/O connection with length, l. It is known that the delay, t, of the I/O connection with length, l, must satisfy the timing constraint, $t \le t_{\max}$. By using the delay expression in Elmore delay model, the length, l, of the I/O connection must satisfy the following constraint.

$$l \le \frac{\sqrt{c_0^2 R^2 + r_0^2 C^2 + 2 r_0 c_0 t_{\max}} - c_0 R - r_0 C}{r_0 c_0},$$

Hence, the maximal length, l_{max}, of the I/O connection can be obtained as

$$l_{\max} = \frac{\sqrt{c_0^2 R^2 + r_0^2 C^2 + 2 r_0 c_0 t_{\max}} - c_0 R - r_0 C}{r_0 c_0},$$

According to the length extraction of any I/O connection under a given timing constraint, t_i, for I/O pin, p_i, the maximum routing length, l_i, of the I/O pin can be extracted.

B. *I/O Signal Assignment*

According to the length extraction under a given timing constraint, the maximum routing lengths of all I/O pins in a floorplan plane can be extracted. In a Manhattan routing model, the collection of points within a fixed distance of a given point is called a *Manhattan circular region*(MCR) whose boundary is composed of two line segments with slope +1 and two line segments with slop -1. Furthermore, the radius of the MCR is the Manhattan distance between the given point and its boundary. For any I/O pin, p_i, with its timing constraint, t_i, its feasible pad candidates must be inside the Manhattan circular region, MCR_i, whose center is the coordinate, (x_i, y_i), of the I/O pin, p_i, and whose radius is the maximum routing length, l_i, under the timing constraint, t_i. As illustrated in Fig. 5, I/O pin, p_k, constructs its MCR_k based on the coordinate, (x_k, y_k), of the I/O pin as the MCR center and the maximum routing length, l_k, as the MCR radius. It is assumed that there are four bump pads, $b_{i,j}$, $b_{i+1,j}$, $b_{i,j+1}$, and $b_{i+1,j+1}$, around the MCR_k. Clearly, the bump pads, $b_{i+1,j}$ and $b_{i+1,j+1}$, are not feasible to be assigned onto the pin, p_k, because the two bump pads are not located inside MCR_k. In contrast, the two bump pads, $b_{i,j}$ and $b_{i,j+1}$, are feasible to be assigned onto the pin, p_k, because the two bump pads are located inside MCR_k. Hence, the two pads, $b_{i,j}$ and $b_{i,j+1}$, are assigned as the pad candidates of the pin, p_k.

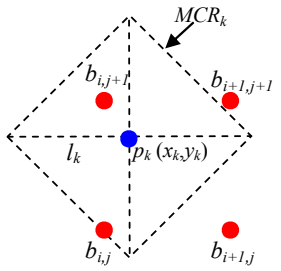

Fig. 5. Bump pad candiates for I/O pin

Refer to the set of 9 I/O pins, p_1, p_2,..., and p_9, and the set of 9 bump pads, b_1, b_2,..., and b_9, in Fig. 3(a), based on the extraction of the maximum routing lengths of the I/O pins, all the MCRs can be achieved and illustrated in Fig. 6(a). According to the locations of all the bump pads and all the MCRs, the pad candidates of each I/O pin can be obtained and all the possible I/O signals can be constructed as illustrated in Fig. 6(b). Clearly, I/O pin, p_1, has a set of pad candidates, b_2, b_3 and b_6, I/O pin, p_2, has a set of pad candidates, b_1 and b_2, I/O pin, p_3, has a set of pad candidates, b_1 and b_4, I/O pin, p_4, has a set of pad candidates, b_2, b_3, b_5 and b_6, I/O pin, p_5, has a set of pad candidates, b_4 and b_7, I/O pin, p_6, has a set of pad candidates, b_5 and b_6, I/O pin, p_7, has a set of pad candidates, b_8 and b_9, I/O pin, p_8, has a set of pad candidates, b_7 and b_8, and I/O pin, p_9, has a pad candidate, b_8.

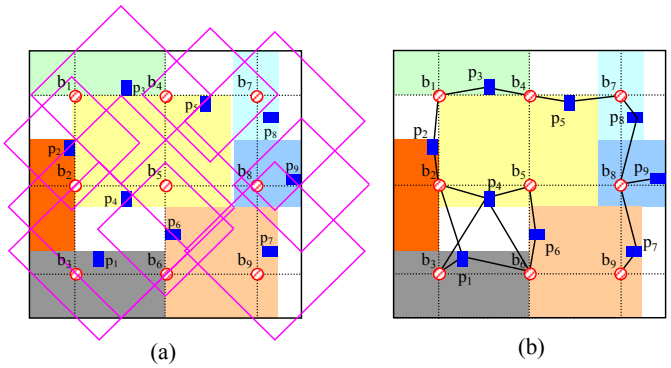

Fig. 6. Distribution of Manhattan circular regions for I/O pins and all the possible I/O signal for 9 I/O pins

According to the possible I/O signals for all the I/O pins, a bipartite graph, $G=(V_p, V_b, E)$, can be constructed as follows: the vertex set in G can be divided into a vertex set, V_p, for I/O pins and a vertex set, V_b, for pad candidates. For v_i in V_p and v_j in V_b, there exists an edge, $e_{i,j}$, in E if the connection between the corresponding I/O pin in v_i and the corresponding pad candidate in v_i is a possible I/O signal under a given timing constraint. Clearly, an assignment solution can be obtained if the number of pad candidates is greater than or equal to the number of I/O pins and each I/O pin has at least one pad candidate. To make each I/O pin to assign a feasible pad candidate, the I/O pins with only one pad candidate must have the highest priority to construct I/O signals. Therefore, a

matching-based I/O signal assignment can be applied as follows: Firstly, any vertex in V_p with degree 1 and its connecting vertex in V_b are selected to form an I/O signal and the original bipartite graph is modified by deleting the selected vertex pair in V. If there is no vertex in V_p with degree 1, the vertex in V_p with the smallest degree and its nearest connecting vertex in V_b are selected to form an I/O signal and the original bipartite graph is modified by deleting the selected vertex pair in V. Until the vertex set, V_p, is empty or there is no edge in G, the assignment process will stop and the final I/O signal assignment result will be obtained.

Refer to all the possible I/O signals in Fig. 6(b), a bipartite graph, $G=(V_p, V_b, E)$ can be built as illustrated in Fig. 7, where $V_p=\{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9\}$, $V_b=\{b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9\}$ and $E=\{e_{1,2}, e_{1,3}, e_{1,6}, e_{2,1}, e_{2,2}, e_{3,1}, e_{3,4}, e_{4,2}, e_{4,3}, e_{4,5}, e_{4,6}, e_{5,4}, e_{5,7}, e_{6,5}, e_{6,6}, e_{7,8}, e_{7,9}, e_{8,7}, e_{8,8}, e_{9,8}\}$. By performing a matching-based I/O signal assignment, the edge, $e_{9,8}$, is firstly selected to form an I/O signal and the edges, $e_{7,8}$ and $e_{8,8}$, are deleted because the pad candidate, b_8, is assigned onto the pin, p_9. Furthermore, for the vertices in V_p with degree 1, the edges, $e_{7,9}$, $e_{8,7}$, $e_{5,4}$, $e_{3,1}$ and $e_{2,2}$, are sequentially selected to form I/O signals. Finally, for the vertices in V_p with the smallest degree, the edges, $e_{6,6}$, $e_{1,3}$ and $e_{4,5}$, are sequentially selected to form I/O signals and the assignment result of 9 I/O signals is illustrated in Fig. 8.

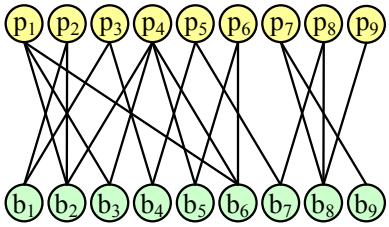

Fig. 7. Construction of a bipartite graph for all the I/O signals of 9 I/O pins

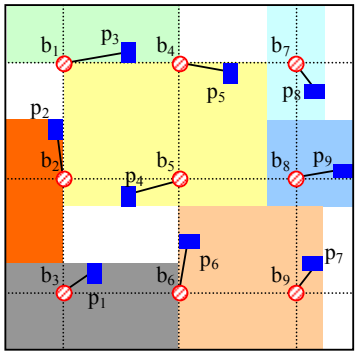

Fig. 8. I/O signal assignment for 9 I/O pins

C. *Timing-Constrained I/O Buffer Placement*

After a set of I/O signals is constructed in I/O signal assignment, necessary I/O buffers must be further placed onto feasible buffer locations to complete I/O connections between IO pins and bump pads. As illustrated in Fig. 9, according to the size of an I/O buffer, the whitespace in a floorplan plane can be assigned as 13 buffer locations. On the other hand, if the delay of an I/O signal from its I/O pin to its bump pad via an I/O buffer violates the given timing constraint, the placement of the I/O buffer for the I/O signal is not allowed. Hence, the location of the placed buffer for an I/O signal must be one available buffer location under its timing constraint.

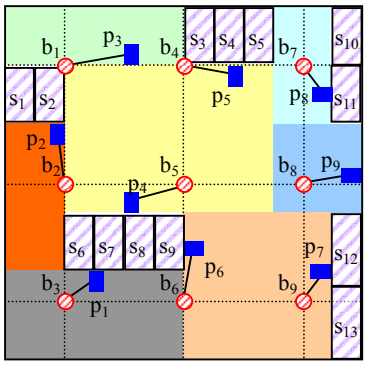

Fig. 9. Available locations for I/O buffer placement

For any I/O signal from I/O pin, p_i, to bump pad, b_j, based on the construction of a Manhattan circular region, two Manhattan circular regions, MCR^p_i, and MCR^b_j, must be built from I/O pin, p_i, to the placed I/O buffer and from the placed I/O buffer to the corresponding bump pad, b_j, respectively. Furthermore, the *feasible location region*, FLR, of the placed I/O buffer for the corresponding I/O signal can be defined as the intersection region between two MCRs, MCR^p_i, and MCR^b_j, as illustrated in Fig. 10, where the center of the MCR^p_i is the coordinate, (x_i, y_i), of the I/O pin, p_i, and the radius, l^p_i, of the MCR^p_i is the maximum routing length under the given timing delay, t^p_i, and the center of the MCR^b_j is the coordinate, (x_j, y_j), of the bump pad, b_j, and the radius, l^b_j, of the MCR^b_j is the maximum routing length under the given timing delay, t^b_j.

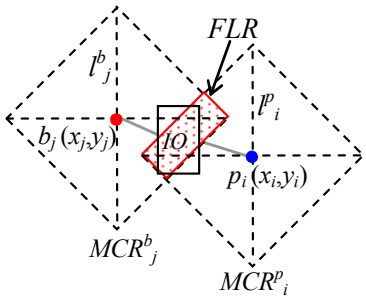

Fig. 10. Feasible location region for an I/O buffer

To satisfy the given timing constraint of any I/O pin, p_i, the FLR between MCR^p_i, and MCR^b_j, for the placed I/O buffer must be constrained into two necessary conditions. Firstly, the timing constraint of the I/O signal for the I/O pin, p_i, must be satisfied as

$$t^p_i + t_{buf} + t^b_j \le t_{max},$$

where t^p_i is the timing delay from the I/O pin, p_i, to the placed I/O buffer, t_{buf} is the intrinsic delay of the placed I/O buffer, and t^b_j is the timing delay from the placed I/O buffer to the bump pad, b_j. Based on the length, l^p_i, between the I/O pin, p_i, and the placed I/O buffer and the length, l^b_j, between the placed I/O buffer and the bump pad, b_j, t^p_i and t^b_j can be obtained as

$$t_i^p = \left(\frac{r_0 c_0}{2}\right) l_i^{p\,2} + (Rc_0 + r_0 C_b) l_i^p + RC_b \text{ and}$$

$$t_j^b = \left(\frac{r_0 c_0}{2}\right) l_j^{b\,2} + (R_b c_0 + r_0 C) l_j^b + R_b C,$$

where R_b and C_b are the output resistance and input capacitance of an placed I/O buffer, respectively. On the other hand, to form the intersection region between two MCRs, the total length of both radiuses must be greater than or equal to the distance, d_{ij}, between the points, (x_i, y_i) and (x_j, y_j), that is, $l^p{}_i + l^b{}_j \geq d_{ij}$.

According to all the available locations of I/O buffers for any I/O signal in a floorplan plane, a bipartite graph, $G(V_s, V_b, E)$, can be constructed as follows: the vertex set in G can be divided into a vertex set, V_s, for I/O signals and a vertex set, V_b, for available buffer locations. For v_i in V_s and v_j in V_b, there exists an edge, $e_{i,j}$, in E if the corresponding I/O signal in v_i can use the corresponding buffer location in v_i to place an I/O buffer under a given timing constraint. Clearly, a placement solution can be obtained if the number of buffer locations is greater than or equal to the number of I/O signals and any I/O signal has at least one buffer location to place an I/O buffer. To make each I/O signal to place an I/O buffer, the I/O signals with only one buffer location must have the highest priority to place I/O buffers. Therefore, a matching-based I/O buffer placement can be applied as follows: Firstly, any vertex in V_s with degree 1 and its connecting vertex in V_b are selected to place an I/O buffer and the original bipartite graph is modified by deleting the selected vertex pair in V. If there is no vertex in V_s with degree 1, the vertex in V_s with the smallest degree and its nearest connecting vertex in V_b are selected to place an I/O buffer and the original bipartite graph is modified by deleting the selected vertex pair in V. Until the vertex set, V_s, is empty or there is no edge in G, the placement process will stop and the final I/O buffer placement result will be obtained.

Refer to the set of 9 I/O signals, n_1, n_2,..., and n_9, and the set of 13 buffer locations, s_1, s_2,..., and s_{13}, in Fig. 9, a bipartite graph, $G(V_s, V_b, E)$ can be built as illustrated in Fig. 11, where V_s={n_1, n_2, n_3, n_4, n_5, n_6, n_7, n_8, n_9}, V_s={s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}, s_{11}, s_{12}, s_{13}} and E={$e_{1,3}$, $e_{1,4}$, $e_{2,2}$, $e_{3,1}$, $e_{3,2}$, $e_{4,5}$, $e_{4,6}$, $e_{5,7}$, $e_{5,8}$, $e_{6,5}$, $e_{6,6}$, $e_{7,13}$, $e_{8,10}$, $e_{8,11}$, $e_{9,11}$, $e_{9,12}$}. By performing a matching-based I/O buffer placement, the edges, $e_{2,2}$ and $e_{7,13}$, are selected to place two I/O buffers and the edges, $e_{3,2}$, is deleted because the location, s_2, is used for the I/O signal, n_2. Furthermore, for the vertices in V_s with degree 1, the edge, $e_{3,1}$, is selected to place an I/O buffer. Finally, for the vertices in V_p with the smallest degree, the edges, $e_{1,3}$, $e_{4,5}$, $e_{6,6}$, $e_{5,7}$, $e_{8,11}$, *and* $e_{9,12}$, are sequentially selected to place I/O buffers and the placement result of 9 I/O buffers is illustrated in Fig. 12.

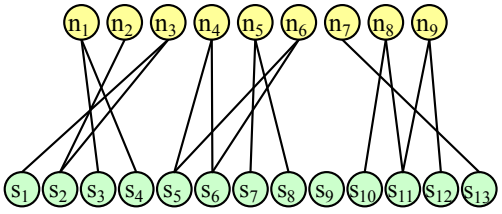

Fig. 11. Construction of a bipartite graph for 9 I/O signals and 13 available buffer locations

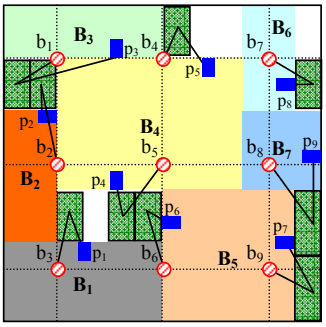

Fig. 12. Timing-constrained I/O buffer placement for 9 I/O pins

D. *Analysis of Time Complexity*

Based on the timing constraints of all I/O pins, it is clear that the time complexity of extracting the timing-constrained routing lengths of all the pins in Elmore delay model is O(n), where n is the number of I/O pins in a floorplan plane. Furthermore, the time complexity of constructing the Manhattan circular regions(MCRs) for all the pins under their timing constraints is O(n). Based on all the possible I/O signal, it is clear that the time complexity of constructing a bipartite graph is O(nr) and the time complexity in a matching-based I/O signal assignment is O(n^2), where r is the number of bump pads in the area-I/O flip-chip design.

In the placement of I/O buffers, firstly, the time complexity of finding the possible buffer locations for all the I/O signals under their timing constraints is O(n^2). Furthermore, the time complexity of constructing a bipartite graph is O(ns). Finally, the time complexity in a matching I/O buffer placement is O(n^2). Hence, the time complexity of timing-constrained I/O buffer placement is O(n^2+nr+ns). As a result, the time complexity in the proposed approach is O(nr+ns) because the number of bump pads is generally greater than the number of I/O pins.

IV. Experimental Results

For timing-constrained I/O buffer placement, our proposed approach has been implemented by standard C++ language and run on an Intel Core2 Quad 2.6GHz machine with 3.5G memory. We take seven industrial circuits in [7] as tested cases. Due to no timing constraint in these tested cases, we set the timing delay of the distance between two adjacent bump pads as a unit delay and the timing constraint for all I/O pins as the window size of the delay of multiple units. The parameters in 0.18μm process from NTRS'97 are used to compute the delay of a RC circuit. In the experiments, the driver strength and the driving load are set as 180(Ω) and 23.4(pF), respectively, r_0 is assigned as 0.0076(Ω/μm) and c_0 is assigned as 0.118(pF/μm). Besides that, one kind of I/O buffers with input capacitance of 23.4(pF), output resistance of 180(Ω) and the intrinsic delay of 36.4(ps) is applied for the tested cases. Here, the timing-constrained satisfaction ratio(TCSR) is defined as the ratio between the number of I/O signals which satisfy the given timing constraints and the total number of I/O signals in an area-I/O flip-chip design.

Two experiments are set up to evaluate the TCSR and the results in our proposed approach are compared with those in

Peng's SA-based approach[7]. In the first experiment, no timing constraint is considered in 7 tested cases and the experimental results including the total wirelength and the maximum delay are shown in Table I. The experimental results in Table I show that our approach can reduce 71.82% of the total wirelength and 55.74% of the maximum delay for 7 tested cases on the average. Under the various timing constraints from the window sizes(W.S) of the delay of 6, 12 and 18 units, the tested cases are used to evaluate the TCSR in our second experiment and the experimental results are shown in Table II. Compared with Peng's SA-based approach[7], the experimental results show that the SA-based approach only has low TCSR in 7 tested cases to satisfy the given timing constraints. Under the first timing constraint, our proposed approach obtains 86.5% TCSR in 7 tested cases in on the average. As the timing constraint is modified as the window size of the delay 18 units, all the tested cases are completely assigned by our proposed approach.

V. CONCLUSIONS

The problem of timing-constrained I/O buffer placement in an area-IO flip-chip design is firstly formulated. Given a set of blocks with some I/O pins and the timing constraints of the I/O pins in a floorplan plane, our proposed approach is effectively to assign all the I/O signals and place the necessary I/O buffers for the assigned I/O signals with satisfying the given timing constraints. Experimental results present that our proposed approach reduces the total wirelength and the maximum delay with no timing constraint and achieves higher TCSR with the given timing constraints than the previous work.

REFERENCES

[1] P. Dehkordi and D. Bouldin, "Design for packageability: the impact of bonding technology on the size and layout of VLSI dies," *Multi-chip Module Conference*, pp. 153-159, 1993.

[2] P. Buffet, J. Natonio, R. Proctor, Y. Sun and G. Yasar, "Methodology for I/O cell placement and checking in ASIC designs using area-array power grid," *IEEE Custom Integrated Circuits Conference*, pp. 125-128, 2000.

[3] A. E. Caldwell, A. B. Kahng and I. L. Markov, "Can recursive bisection alone produce routable placement?," *ACM/IEEE Design Automation Conference*, pp. 477-482, 2000.

[4] A. R. Agnihotri, M. C. Yildiz, A. Khatkhate, A. Mathur, S. Ono and P. H. Madden, "Fractical cut: improved recursive bisection placement," *IEEE/ACM International Conference on Computed-Aided Design*, pp. 307-310, 2003.

[5] S. N. Adya, I. L. Markov and P. G. Villarrubia, "On whitespace in mixed-size placement and physical synthesis," *IEEE/ACM International Conference on Computed-Aided Design*, pp. 311-318, 2003.

[6] H. Y. Hsieh and T. C. Wang, "Simple yet effective algorithms for block and I/O buffer placement in flip-chip design," *IEEE International Symposium on Circuits and Systems*, pp.1879-1882, 2005.

[7] C. Y. Peng, W. C. Chao, Y. W. Chang and J. H. Wang, "Simultaneous block and I/O buffer floorplanning for flip-chip design", *IEEE Asia and South Pacific Design Automation Conference*, pp.213-218, 2006.

[8] J. Xiong, Y. C. Wong, E. Sarto and L. He, "Constraint driven I/O planning and placement for chip-package co-design," *IEEE/ACM Asia and South Pacific Design Automation Conference*, pp.207-212, 2007.

[9] M. F. Lai and H. M. Chen, "An implementation of performance-driven block and I/O placement for chip-package codesign,", *IEEE International Symposium on Quality Electronic Design*, pp.604-607, 2008.

[10] C. Y. Wang and W. K. Mak, "Signal skew aware floorplanning and bumper signal assignment technique for flip-chip," *IEEE/ACM Asia and South Pacific Design Automation Conference*, pp.341-346, 2009.

[11] M. de Berg, O. Cheong, M. van Kreveld and M. Overmars, *Computational Geometry: Algorithms and Applications*, 3rd edition, Springer-Verlag, 2005.

TABLE I. EXPERIMENTAL RESULTS FOR I/O BUFFER PLACEMENT WITH NO TIMING CONSTRAINT

Ex	#Pin	#Block	#Bump	Chip area	Peng's SA-based Approach[7]			Our Approach			
					#WL(μm)	*TCSR*	*Max. delay(ps)*	*#WL(μm)*	*TCSR*	*Max. delay(ps)*	*Time(s)*
case1	25	6	25	1040x1040	16270(100%)	100%	23.18 (100%)	12130(75.20%)	100%	21.41(92.36%)	0.016
case2	168	12	289	3440x3440	710450(100%)	100%	122.51(100%)	162360(22.85%)	100%	62.92(51.35%)	0.562
case3	320	23	441	4240x4240	1779840(100%)	100%	170.03(100%)	339040(19.04%)	100%	69.61(40.93%)	1.970
case4	384	28	484	4440x4440	1906720(100%)	100%	147.79(100%)	252260(13.23%)	100%	54.43(36.82%)	4.094
case5	384	28	484	4440x4440	2210200(100%)	100%	173.89(100%)	417000(18.86%)	100%	73.46(42.24%)	2.547
case6	384	28	529	4480x4480	2339220(100%)	100%	345.91(100%)	385340(16.47%)	100%	68.65(19.84%)	3.937
case7	384	28	529	4480x4480	2209740(100%)	100%	330.95(100%)	697260(31.55%)	100%	87.12(26.32%)	3.188

TABLE II. EXPERIMENTAL RESULTS FOR TIMING-CONSTRAINED I/O BUFFER PLACEMENT UNDER THREE WINDOW SIZES = 6, 12 AND 18

Ex	W.S = 6				W.S = 12				W.S = 18			
	Peng's SA-based Approach[7]		*Our Approach*		*Peng's SA-based Approach[7]*		*Our Approach*		*Peng's SA-based Approach[7]*		*Our Approach*	
	TCSR	*#WL(μm)*	*TCSR*	*#WL(μm)*	*TCSR*	*#WL(μm)*	*TCSR*	*#WL(μm)*	*TCSR*	*#WL(μm)*	*TCSR*	*#WL(μm)*
case1	100%	16270	100%	12130	100%	16270	100%	12130	100%	16270	100%	12130
case2	N/A	N/A	89.2%	111620	N/A	N/A	100%	162800	0.6%	3550	100%	162360
case3	N/A	N/A	82.5%	194860	N/A	N/A	100%	332280	N/A	N/A	100%	339040
case4	N/A	N/A	98.4%	245840	N/A	N/A	100%	257940	N/A	N/A	100%	252260
case5	N/A	N/A	83.0%	253700	N/A	N/A	100%	412280	N/A	N/A	100%	417000
case6	N/A	N/A	86.1%	243660	3.9%	30660	99.2%	376840	10.2%	105400	100%	385340
case7	1.04%	4300	66.4%	196720	6.2%	42140	94.2%	533860	18.2%	182240	100%	697260