Simultaneous Constrained Pin Assignment and Escape Routing Considering Differential Pairs for FPGA-PCB Co-design

Seong-I Lei and Wai-Kei Mak

***Abstract*—With the increasing complexity of circuit design in recent years, the pin assignment and escape routing problems for field-programmable gate array (FPGA) on a printed circuit board (PCB) have become greatly difficult due to the fast increase in pin count and density. Most existing works only focus on either the FPGA pin assignment problem or the PCB escape routing problem independently, but cannot handle them simultaneously. In this paper, we propose an integer linear programming based method to simultaneously solve the problem of pin assignment and escape routing for FPGA-PCB co-design. Moreover, differential pairs and single-ended signals are handled together optimally to minimize the total wirelength in escape routing. Encouraging experimental results are shown to support our approach showing reduction in the number of PCB routing layers and/or wirelength.**

***Index Terms*—Differential pairs, escape routing, field-programmable gate array (FPGA)-printed circuit board (PCB) co-design, pin assignment.**

I. Introduction

As THE very large scale integration technology shrinks down to nanometer range, more and more functionalities can be integrated into a chip. The pin count is increasing at the same time. The problem of pin assignment and escape routing for field-programmable gate array (FPGA) on a printed circuit board (PCB) has become greatly difficult due to the fast increase in pin count and density. Although there are some works addressing the FPGA pin assignment problem and the PCB escape routing problem independently, there has never been any attempt to unify these two problems. In order to obtain a feasible pin assignment and an escape routing solution that can reduce the system cost and improve system performance, it is desirable to consider them together.

FPGA pin assignment is complicated due to the various assignment constraints imposed by the FPGA architecture. Several previous works [2]–[4] considered the constrained pin assignment problem. In [2], a heuristic approach was proposed to deal with the requirement of constrained input/output (I/O) placement for FPGAs. Unfortunately, there is no guarantee that the heuristic approach can find a feasible I/O placement solution, even when feasible I/O placement solutions exist. When the approach fails to place all the I/Os, the designer needs to place them manually. An integer linear programming (ILP) based approach is used in [3] to solve the problem but the formulation is not general enough to be applicable to all FPGA devices. Subsequently, a more general ILP based approach is proposed in [4]. Even though [4] associates different costs for assigning an I/O signal to different banks in their objective function, escape routing for the FPGA's I/O signals on the PCB is still not considered.

Escape routing is a key problem in PCB routing. The objective of escape routing is to route signals from all terminal pins inside a pin array component to the boundaries of the component. Three types of escape routing have been considered in the literature. Free escape routing [5], [11], [12] considers single component escape. The pins inside the component are to be routed to the boundaries of the component without any constraint on the pin order. Usually the wirelength is optimized in free escape routing. Ordered escape routing [6], [7] also considers single component but it requires the escape routing solution to satisfy a specified order along the boundary. Simultaneous escape routing [8]–[10] considers escape routing of two components. The pin orders along the boundaries of the two components are required to match each other such that there is no crossing in the area routing between the components. However, all these works except [10] only focus on escape routing without handling the assignment of the I/O signals inside the pin array. The assignment of the I/O signals will greatly affect the routability of the design because the escape routing stage is performed after the pin assignment stage. It will increase the difficulty and increase the wirelength of escape routing if the pin assignment has not been done well. Note that the pin assignment problem addressed in [10] does not have any constraint, which is not true for FPGA devices.

For modern high-speed PCB designs, high frequency signals are usually transmitted through differential pairs. A differential pair consists of two complementary signals for transmitting one signal. The advantages of using differential pairs include better noise immunity and higher reduction of electromagnetic interference. In differential-pair pin assignment and escape

Manuscript received January 10, 2013; revised June 14, 2013; accepted July 25, 2013. Date of current version November 18, 2013. This work was supported by the National Science Council under Grant NSC 102-2220-E-007-013. A preliminary version of this paper appeared in the Proceedings of FPL'2011 [1]. This paper was recommended by Associate Editor C. C.-N. Chu.

The authors are with the Department of Computer Science, National Tsing Hua University, HsinChu 300, Taiwan (e-mail: d9762804@oz.nthu.edu.tw; wkmak@cs.nthu.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCAD.2013.2276536

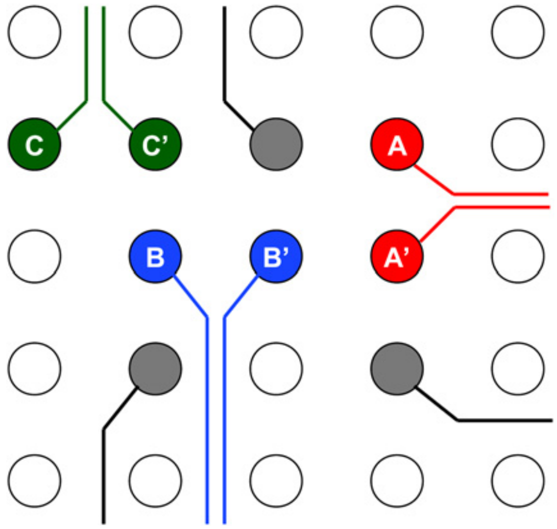
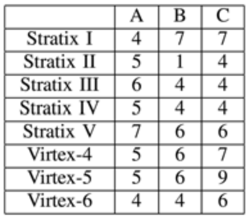


Fig. 1. Escape routing of six signals. (A, A'), (B, B') and (C, C') are three differential pairs.

routing, we prefer the two pins of a differential pair to be assigned to adjacent pins because the two wires from a differential pair can be merged immediately so that the parallel routing of a differential pair can cancel most part of crosstalk or electromagnetic interference. The escape routes of the two pins of a differential pair should be as close as possible in order to take the full advantages of differential signalling. Fig. 1 shows an example of escape routing including three differential signals (A, A'), (B, B') and (C, C').

Unfortunately, there are only a few works [13]–[15] on escape routing considering differential pairs. Yan *et al.* [13] presented an optimal algorithm for routing a single differential pair. For multiple differential pairs, it proposed a scheme combining network flow, and rip-up and reroute. Tai *et al.* [14] considered length matching besides wirelength minimization in escape routing. However, neither [13] nor [14] can guarantee to obtain an optimal routing solution of multiple differential pairs. Moreover, their methods mainly focus on the escape routing of differential signals while single-ended signals are largely ignored. Fang *et al.* [15] considered differential pairs and single-ended signals in routing for chip-package-board co-design. But there are a few limitations. First, they assume that each differential pair must be routed within a specified region of the pin array and only construct the routing network for a differential pair inside the region in order to reduce the complexity, but it will affect the optimality. Second, their method cannot allow a differential pair and a single-ended signal to pass through the same tile. Third, in their routing network, for each differential pair they need to introduce two exclusive nodes for them in each interval within its bounding box for routing. If the number of differential pairs is large, the runtime will become enormous because a large number of nodes will be needed in each interval of their routing network. Fourth, the routing model used in their approach is an over-simplified model, which is shown to be unable to model the diagonal capacity correctly [5].

TABLE I
NUMBERS OF CLASS A, B, AND C SIGNALS SUPPORTED BY ALTERA STRATIX I TO V FPGAS AND XILINX VIRTEX-4 TO VIRTEX-6 FPGAS



	A	B	C
Stratix I	4	7	7
Stratix II	5	1	4
Stratix III	6	4	4
Stratix IV	5	4	4
Stratix V	7	6	6
Virtex-4	5	6	7
Virtex-5	5	6	9
Virtex-6	4	4	6

The contributions of our work are summarized as follows.

1) Previous works only focus on either the pin assignment problem, or escape routing problem, but cannot handle them simultaneously. To the best of our knowledge, this is the first work that simultaneously considers the pin assignment problem and escape routing problem for FPGA on a PCB. We present an integer linear programming based approach that can simultaneously handle the pin assignment and escape routing of differential pairs and single-ended signals to minimize the total wirelength in escape routing. Our routing network model is very flexible and allows a differential pair and single-ended signals to pass through the same tile at the same time while [15] cannot.
2) Besides the exact formulation, we also present a faster alternative formulation.
3) Our ILP formulation can be extended to handle multiple layers.
4) Experimental results demonstrate that our approach can reduce the number of PCB layers required and use shorter wirelength when compared to competitive two-stage approaches.

The rest of this paper is organized as follows. In Section II, we discuss FPGA I/O standards and constrained pin assignment. Then, we formulate the problem of constrained FPGA pin assignment and escape routing for FPGA-PCB co-design. In Section III, we show that this problem can be neatly modelled as a 0-1 integer linear program. Experimental results are reported in Section IV. Section V concludes the entire work.

II. PRELIMINARIES & PROBLEM FORMULATION

A. FPGA I/O Standards and I/O Bank Constraints

Different I/O standards are used in IC design and they have different voltage requirements. Some I/O standards require a specific supply voltage (V_{cco}) to power either or both the input and output signals. On the other hand, some I/O standards require the use of a differential amplifier input and an external reference voltage (V_{ref}) must be provided to the amplifier.

I/O signals can be divided into four major classes.

1) Class A: I/O signals that require a V_{cco} voltage but not a V_{ref} voltage.

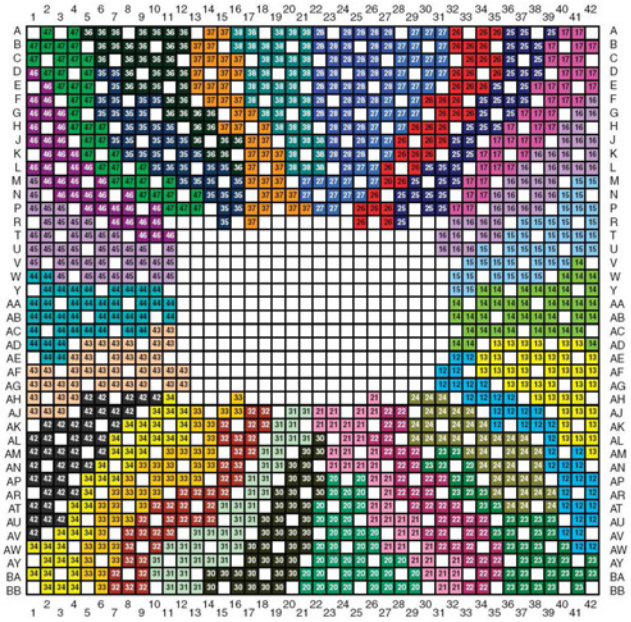


Fig. 2. Xilinx Virtex-6 FF1760 LX760 FPGA I/O bank diagram.

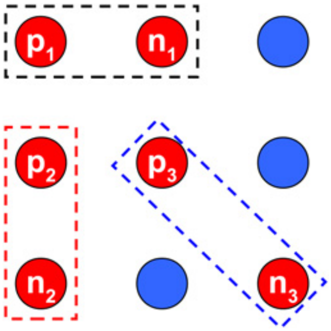
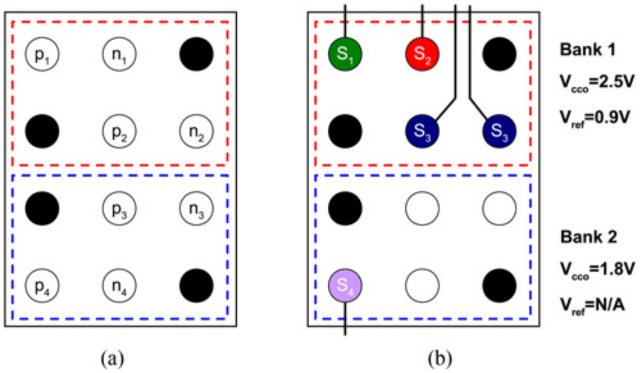


Fig. 3. Example with three differential pin-pairs.

2) Class B: I/O signals that require a V_{ref} voltage but not a V_{cco} voltage.
3) Class C: I/O signals that require both a V_{cco} voltage and a V_{ref} voltage.
4) Class D: I/O signals that require neither a V_{cco} voltage nor a V_{ref} voltage.

Different FPGA families support different sets of I/O standards. The numbers of class A, class B and class C signals supported by Altera Stratix I to V FPGAs and Xilinx Virtex-4 to Virtex-6 FPGAs are listed in Table I. For instance, for a Stratix I FPGA, I/O signals of class A can be divided into A_1 to A_4 according to their V_{cco} values, those of class B can be divided into B_1 to B_7 according to their V_{ref} values. There are seven combinations of V_{cco} and V_{ref} for class C I/O signals. We denote the V_{cco} voltage used by A_i as $V_{Ci}(i = 1, ..., 4)$ and the V_{ref} voltage used by B_j as $V_{Rj}(j = 1, ..., 7)$.

The commercial FPGAs that support multiple I/O standards all use a banked I/O organization similar to the one shown in Fig. 2. The physical I/O pins are organized into a number of I/O banks. Each I/O bank is served by a single V_{cco} voltage and a single V_{ref} voltage which can be set by the user. Hence, no two I/O signals that require different V_{cco} voltages or different V_{ref} voltages can be assigned to the same bank. However, I/O signals of different standards with no conflict in their V_{cco} voltage and V_{ref} voltage requirements can be assigned to I/O pins in the same bank by setting the V_{cco} and V_{ref} voltages of the bank accordingly.



(a) (b)

Fig. 4. (a) FPGA package pinout. (b) Feasible solution of the constrained pin assignment and escape routing in the FPGA.

Some specific pairs of user I/O pins can be used by differential signals. The pins in these pairs are either horizontial, vertical, or diagonal neighbors depending on the specification of the FPGA. Fig. 3 shows an example with three pin-pairs. Pins p_i and n_i represent the positive and negative side of a differential pair. Note that the predefined differential pin-pairs of a FPGA are mutually exclusive and do not share any pin.

B. Problem Formulation

For constrained pin assignment of a design mapped to a FPGA, we need to assign all the I/O signals of the mapped design to the physical I/O pins of the FPGA and determine the V_{cco} voltage and the V_{ref} voltage for each bank in the FPGA such that the V_{cco} voltage and V_{ref} voltage requirements for all signals are satisfied. Therefore, a class A_i I/O signal can only be assigned to a pin in a bank which has the V_{Ci} supply voltage. A class B_j I/O signal can only be assigned to a pin in a bank which has the V_{Rj} reference voltage. A class C_{ij} I/O signal can only be assigned to a pin in a bank which has the V_{Ci} supply voltage and the V_{Rj} reference voltage.

We use the example in Fig. 4 to illustrate the constrained pin assignment and escape routing for FPGA-PCB co-design. There are two banks and four pin-pairs in the FPGA. Nonuser I/O pins are denoted by black pins. Suppose we need to assign and route two class A_1 single-ended signals S_1 and S_2, one class B_1 differential signal S_3 and one class A_2 single-ended signal S_4 where $V_{C1} = 2.5V$, $V_{C2} = 1.8V$ and $V_{R1} = 0.9V$. Fig. 4(b) shows a feasible solution of the pin assignment and escape routing with this FPGA. The V_{cco} voltage and the V_{ref} voltage of bank 1 are set to $2.5V$ and $0.9V$. The V_{cco} voltage of bank 2 is set to $1.8V$. Note that signals S_1, S_2 and S_3 can be assigned to bank 1 at the same time because there is no conflict in their voltage requirement.

The problem of constrained pin assignment and escape routing for FPGA-PCB co-design is defined as follows. Given a FPGA package with k banks, and a set of I/O signals to be assigned to the I/O pins and routed to the package boundary,

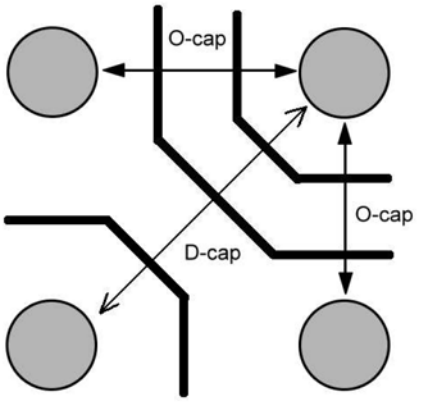


Fig. 5. Orthogonal and diagonal capacity constraints. In this example, O-cap = 2 and D-cap = 3.

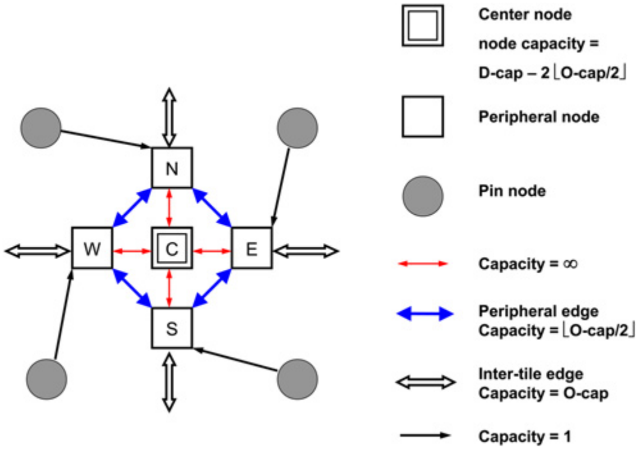


Fig. 6. Single-ended signal routing network model for a tile.

determine the supply voltage V_{cco} and the reference voltage V_{ref} for each bank in the FPGA, and assign all I/O signals to the physical user I/O pins on the FPGA satisfying the V_{cco} and V_{ref} voltage requirements and obtain a legal escape routing solution without violating the capacity constraints from PCB design rules. Our objective is to minimize the total number of PCB routing layers required and the total wirelength in escape routing of the I/O signals.

III. Simultaneous Approach

In this section, we propose a simultaneous approach to solve the problem. We first introduce the routing networks for escape routing. Then we present our ILP formulation.

A. *Escape Routing Networks*

In a pin array, a tile is a square formed by four adjacent pins (Fig. 5). The separation between adjacent pins in a tile limits the number of wires that can pass through two orthogonal or diagonal adjacent pins. The capacities of the four sides of a tile are equal to the orthogonal capacity (O-cap), and the capacities of its two diagonals are equal to the diagonal capacity (D-cap). For a legal escape routing solution, the O-cap and D-cap constraints have to be satisfied everywhere in the pin array. Fig. 5 shows a legal escape routing with O-cap = 2 and D-cap = 3.

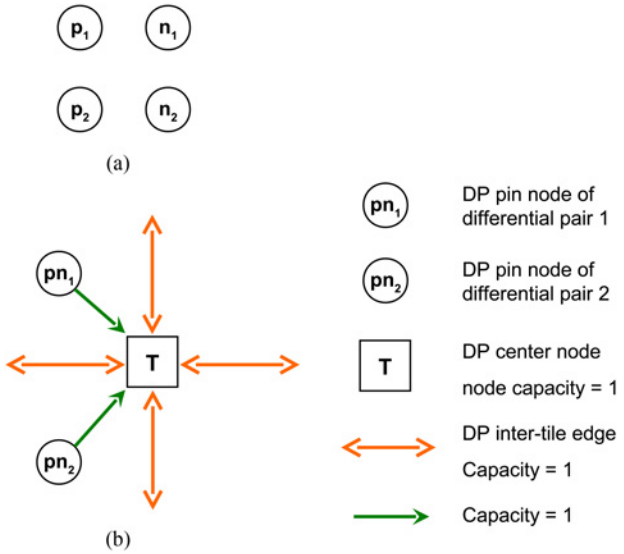


Fig. 7. (a) Two pin-pairs form a tile. (b) Differential-pair routing network model for the tile. Note that we only use one DP pin node to represent a differential pair.

For escape routing, we employ the routing network model shown in Fig. 6 for single-ended signals and we propose another routing network model shown in Fig. 7 for differential pairs. Two networks are constructed for the same FPGA pin array, one for single-ended signals and one for differential pairs. The size of each network will be a constant times the size of the FPGA pin array. We will compute the routing of single-ended signals and differential pairs in their respective network concurrently in our ILP formulation. We use some ILP constraints to guarantee that there is no conflict in the pin assignment and escape routing between single-ended signals and differential pairs. We discuss these two routing network models in details below.

1) *Network for Single-ended Signal Escape Routing:* A routing network model that correctly captures the capacity constraints of O-cap and D-cap for escape routing is proposed in [5]. Each tile is modeled as in Fig. 6. There are four pin nodes in each tile and the directed edge from a pin node on the corner to a peripheral node has capacity 1. The four peripheral nodes E, S, W and N represent four sides of the tile. The center node C has capacity D-cap$-2\cdot\lfloor$O-cap$/2\rfloor$. There are bidirectional edges between every peripheral node and the center node, these edges have infinite capacity. Furthermore, there are bidirectional edges called peripheral edges between peripheral node pairs (N,E), (E,S), (S,W), (W,N). The peripheral edges have capacity $\lfloor$O-cap$/2\rfloor$. Intertile edges are bidirectional edges that connect between tiles and have capacity O-cap.

2) *Network for Differential-Pair Escape Routing:* We can construct a simpler routing network model for differential-pair escape routing. A differential pair uses two pins for transmitting one differential signal. Without loss of generality, we assume that there are two pin-pairs within a tile and the

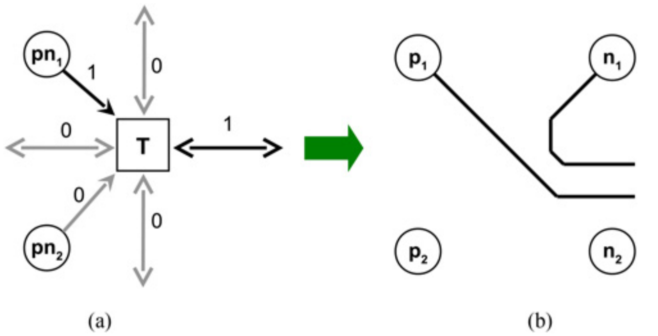


Fig. 8. (a) Solution in differential-pair routing network. (b) Corresponding escape routing of the differential pair.

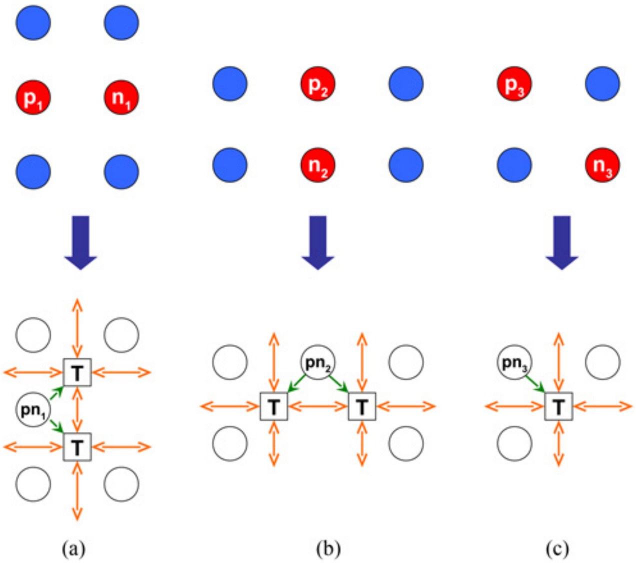


Fig. 9. Corresponding differential-pair routing networks for the three types of pin-pair arrangements. (a) Horizontal. (b) Vertical. (c) Diagonal.

TABLE II
PARAMETERS AND SETS USED IN THE ILP FORMULATION

Parameters	
U^k	the number of regular user I/O pins in bank k.
$\lvert\dot{A}_i\rvert$	number of class A_i single-ended signals to be assigned.
$\lvert\dot{B}_j\rvert$	number of class B_j single-ended signals to be assigned.
$\lvert\dot{C}_{ij}\rvert$	number of class C_{ij} single-ended signals to be assigned.
$\lvert\dot{D}\rvert$	number of class D single-ended signals to be assigned.
$\lvert\ddot{A}_i\rvert$	number of class A_i differential signals to be assigned.
$\lvert\ddot{B}_j\rvert$	number of class B_j differential signals to be assigned.
$\lvert\ddot{C}_{ij}\rvert$	number of class C_{ij} differential signals to be assigned.
$\lvert\ddot{D}\rvert$	number of class D differential signals to be assigned.
Sets	
N	set of integers.
P_k	set of pin-pairs in bank k that can be assigned to differential signals.
$\dot{E}_{inter}$	set of inter-tile edges.
$\dot{E}^b_{inter}$	set of inter-tile edges that escape to the package boundaries.
$\dot{E}_{peri}$	set of peripheral edges.
$\dot{E}_{C'C''}$	set of edges that connect nodes C' and C'' for any center node C as in Fig. 10.
$\dot{E}_B$	set of edges that directly escape to the package boundaries from boundary pin nodes.
$\ddot{E}_{inter}$	set of DP inter-tile edges.
$\ddot{E}^b_{inter}$	set of DP inter-tile edges that escape to the package boundaries.
$\ddot{E}_{T'T''}$	set of edges that connect nodes T' and T'' for any DP center node T.
E_v^{in}	set of incoming edges of node v.
E_v^{out}	set of outgoing edges of node v.

top two pins form one pair and the bottom two pins form another pair as shown in Fig. 7(a). Pin p_i represents the positive side of a differential pair while pin n_i represents the negative side of the same differential pair. Fig. 7(b) shows the corresponding routing network model for differential-pair escape routing. Here we only use one DP center node for each tile and one DP pin node pn_i to represent the differential pair with pins p_i and n_i. The node capacity of the DP center node is 1. There is a directed edge of capacity 1 from a DP pin node of a differential pair to a DP center node if the DP pin node and the DP center node belong to the same tile. DP intertile edges are bidirectional edges that connect between tiles and also have capacity 1.

In the differential-pair routing network, one unit of flow represents two wires from one differential pair. For example, if we get a solution in the differential-pair routing network as shown in Fig. 8(a). Fig. 8(b) shows the corresponding escape routing of the differential pair.

Fig. 9(a)–(c) show the corresponding routing network for the three types of pin-pair arrangement as mentioned before. For the horizontal and vertical arrangements, the differential pair can escape initially through one of the two adjacent tiles, so there are two directed edges connecting the DP pin node to the DP center nodes of these two tiles. For the diagonal arrangement, the differential pair can only escape initially through its own tile, so there is only one directed edge connecting the DP pin node to the DP center node of its own tile. Subsequently, we want the complementary signals of a differential pair to be routed together so that zero skew routing can be obtained for each differential pair.

B. ILP with Single-Ended Signals Only

We show that the constrained FPGA pin assignment and PCB escape routing problem can be cast as a 0-1 integer linear program. Our formulation can be applied to different FPGA

TABLE III
VARIABLES USED IN THE ILP FORMULATION

$\dot{a}_i^k$	number of class A_i single-ended signals assigned to bank k.
$\dot{b}_j^k$	number of class B_j single-ended signals assigned to bank k.
$\dot{c}_{ij}^k$	number of class C_{ij} single-ended signals assigned to bank k.
$\dot{d}^k$	number of class D single-ended signals assigned to bank k.
$\ddot{a}_i^k$	number of class A_i differential signals assigned to bank k.
$\ddot{b}_j^k$	number of class B_j differential signals assigned to bank k.
$\ddot{c}_{ij}^k$	number of class C_{ij} differential signals assigned to bank k.
$\ddot{d}^k$	number of class D differential signals assigned to bank k.
$f(e)$	number of signals passing through edge e.
α_i^k	0-1 integer variable that $\alpha_i^k = 1$ if bank k is configured with a V_{cco} equal to V_{Ci}, and $\alpha_i^k = 0$ otherwise.
β_j^k	0-1 integer variable that $\beta_j^k = 1$ if bank k is configured with a V_{ref} equal to V_{Rj}, and $\beta_j^k = 0$ otherwise.
σ_v	$\sigma_v = 1$ if pin v is assigned to a single-ended signal and $\sigma_v = 0$ otherwise.
δ_r	0-1 integer variable that $\delta_r = 1$ if pin-pair r in the FPGA is assigned to a differential signal and $\delta_r = 0$ otherwise.
h_t	0-1 integer variable that $h_t = 1$ if tile t is reserved for a differential pair and $h_t = 0$ otherwise.

devices with different voltage levels of V_{cco} and V_{ref} and different number of banks. To better present our formulation, the notations used in the formulation are defined in Tables II and III.

We first present an ILP formulation without considering differential pairs. The formulation considering differential pairs is presented in Section III-C. Our ILP formulation with single-ended signals only is as follows:

$$min \sum_{e \in \dot{E}_{inter}} f(e)$$

$$s.t. \sum_i \dot{a}_i^k + \sum_j \dot{b}_j^k + \sum_{i,j} \dot{c}_{ij}^k + \dot{d}^k \le U^k \quad \forall \text{bank } k \tag{1}$$

$$\sum_k \dot{a}_i^k = |\dot{A}_i| \quad \forall i \tag{2}$$

$$\sum_k \dot{b}_j^k = |\dot{B}_j| \quad \forall j \tag{3}$$

$$\sum_k \dot{c}_{ij}^k = |\dot{C}_{ij}| \quad \forall i, j \tag{4}$$

$$\sum_k \dot{d}^k = |\dot{D}| \tag{5}$$

$$\dot{a}_i^k \le min(|\dot{A}_i|, U^k)\alpha_i^k \quad \forall i, k \tag{6}$$

$$\dot{b}_j^k \le min(|\dot{B}_j|, U^k)\beta_j^k \quad \forall j, k \tag{7}$$

$$\dot{c}_{ij}^k \le min(|\dot{C}_{ij}|, U^k)\alpha_i^k \quad \forall i, j, k \tag{8}$$

$$\dot{c}_{ij}^k \le min(|\dot{C}_{ij}|, U^k)\beta_j^k \quad \forall i, j, k \tag{9}$$

$$\sum_i \alpha_i^k = 1 \quad \forall \text{bank } k \tag{10}$$

$$\sum_j \beta_j^k \le 1 \quad \forall \text{bank } k \tag{11}$$

$$\dot{a}_i^k, \dot{b}_j^k, \dot{c}_{ij}^k, \dot{d}^k \ge 0 \quad \forall i, j, k \tag{12}$$

$$\alpha_i^k, \beta_j^k \in \{0, 1\} \quad \forall i, j, k \tag{13}$$

$$\sum_i \dot{a}_i^k + \sum_j \dot{b}_j^k + \sum_{i,j} \dot{c}_{ij}^k + \dot{d}^k = \sum_{\text{pin node } v \in \text{ bank } k} \sigma_v \quad \forall k \tag{14}$$

$$\sigma_v = \sum_{e \in E_v^{out}} f(e) \quad \forall \text{pin node } v \tag{15}$$

$$\sum_{e \in E_v^{in}} f(e) = \sum_{e \in E_v^{out}} f(e) \quad \forall \text{peripheral node } v \tag{16}$$

$$\sum_{e \in E_v^{in}} f(e) = \sum_{e \in E_v^{out}} f(e) \quad \forall \text{center node } v \tag{17}$$

$$\sum_{e \in \dot{E}_{inter}^b} f(e) + \sum_{e \in \dot{E}_B} f(e) = \sum_i |\dot{A}_i| + \sum_j |\dot{B}_j| + \sum_{i,j} |\dot{C}_{ij}| + |\dot{D}| \tag{18}$$

$$0 \le \sigma_v \le 1 \quad \forall \text{pin node } v \tag{19}$$

$$f(e) \le D\text{-}cap - 2\lfloor \frac{O\text{-}cap}{2} \rfloor \quad \forall e \in \dot{E}_{C'C''} \tag{20}$$

$$f(e) \le O\text{-}cap \quad \forall e \in \dot{E}_{inter} \tag{21}$$

$$f(e) \le \lfloor \frac{O\text{-}cap}{2} \rfloor \quad \forall e \in \dot{E}_{peri} \tag{22}$$

$$f(e) \ge 0 \quad \forall e. \tag{23}$$

Constraint (1) corresponds to the supply constraint for each bank such that the number of I/O signals consumed in a bank does not exceed the number of pins available. Constraints (2)–(5) correspond to the demand constraints for each class of I/O signals such that all I/O signals of a class are assigned to I/O banks and none of them is left unassigned. Constraints (6)–(9) correspond to the allowance constraints for each class of I/O signals in a bank. Constraint (6) ensures that a class A_i I/O signal is only allowed in a bank if the V_{cco} voltage of this bank is equal to the V_{cco} voltage used by the class A_i I/O signal. Constraints (7)–(9) are the allowance constraints for class B_j and class C_{ij} I/O signals. Constraint (10) corresponds to the V_{cco} selection constraint for each bank such that each bank is assigned exactly one V_{cco} voltage. Similarly, constraint (11) corresponds to the V_{ref} selection constraint for each bank such that each bank is assigned exactly one V_{ref} voltage or no V_{ref} voltage.

Constraints (14)–(23) are for escape routing and the routing network model in Fig. 6 is assumed. Constraint (14) corresponds to the pin assignment constraint for the routing network. The number of used pins in bank k for single-ended signals must equal to the total number of single-ended signals assigned to bank k. Constraints (15)–(17) correspond to the flow conservation constraints for the pin nodes, the peripheral nodes, and the center nodes, respectively. The amount of incoming flow must equal to that of outgoing flow for each

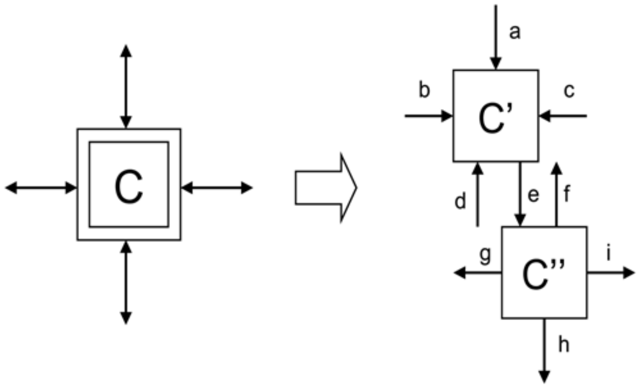


Fig. 10. Center node C can be split into nodes C' and C''.

node. Because the center node C has node capacity, we split it into nodes C' and C'' as shown in Fig. 10. The capacity of the edge e (edge $C'C''$) is equal to the node capacity of the original center node C. Constraint (18) guarantees that the number of nets which escape to the boundaries must be equal to the total number of single-ended signals. Constraint (19) bounds each variable σ_v. Constraints (20)–(22) correspond to the capacity constraints for the center node, the intertile edge and the peripheral edge in the routing network, respectively. Finally, constraint (23) ensures the flow in all edges in the routing network are nonnegative.

In the objective function, the number of signals passing through the intertile edges is minimized and thus the total wirelength can be minimized. We note that if the 0 to 1 integer variables α_i^k and β_j^k are fixed, our formulation reduces to a minimum cost network flow program. So, by imposing integer constraints on α_i^k and β_j^k only, it will automatically yield an optimal solution with integral values for $\dot{a}_i^k$, $\dot{b}_j^k$, $\dot{c}_{ij}^k$, $\dot{d}^k$, σ_v and $f(e)$ using any ILP solver based on the simplex algorithm.

It should be noted that a problem instance can be infeasible. If the ILP solver reports that an instance is infeasible, there are two possible reasons. One is failure in pin assignment and the other one is failure in escape routing. To identify which kind of failure has occurred, we can simply solve the problem only with constraints (1)–(13) related to the pin assignment. If failure happens, this means that the design is infeasible on the current FPGA device and we need to use a larger device with more I/O pins and/or banks. If no failure happens, this means that the escape routing resource is not enough. To fix the problem, we can increase the number of PCB routing layers.

C. ILP Incorporating Differential Pairs

Next, we show how to revise our ILP formulation to take differential pairs into account. The objective function is changed as follows:

$$min \sum_{e \in \dot{E}_{inter}} f(e) + 2 \sum_{e \in \ddot{E}_{inter}} f(e).$$

We minimize the total wirelength in the objective function. The total wirelength includes the wirelength of single-ended signals and the wirelength of differential signals, so the number of wires passing through the intertile edges and the DP intertile edges is minimized. Recall that in the differential-pair escape routing, one unit of flow represents two wires. So, we need to multiply the second term by two in the objective function.

For $|P_k| \neq 0$[1], we replace constraint (1) by the following constraint:

$$\begin{aligned} &2\sum_i \ddot{a}_i^k + 2\sum_j \ddot{b}_j^k + 2\sum_{i,j} \ddot{c}_{ij}^k + 2\ddot{d}^k + \\ &\sum_i \dot{a}_i^k + \sum_j \dot{b}_j^k + \sum_{i,j} \dot{c}_{ij}^k + \dot{d}^k \leq U^k \quad \text{if } |P_k| \neq 0. \end{aligned} \tag{24}$$

Furthermore, the following constraints are added to the original ILP formulation:

$$\sum_{k \in K} \ddot{a}_i^k = |\ddot{A}_i| \quad \forall i \tag{25}$$

$$\sum_{k \in K} \ddot{b}_j^k = |\ddot{B}_j| \quad \forall j \tag{26}$$

$$\sum_{k \in K} \ddot{c}_{ij}^k = |\ddot{C}_{ij}| \quad \forall i, j \tag{27}$$

$$\sum_{k \in K} \ddot{d}^k = |\ddot{D}| \tag{28}$$

$$\ddot{a}_i^k \leq min(|\ddot{A}_i|, |P_k|)\alpha_i^k \quad \forall i, k \tag{29}$$

$$\ddot{b}_j^k \leq min(|\ddot{B}_j|, |P_k|)\beta_j^k \quad \forall j, k \tag{30}$$

$$\ddot{c}_{ij}^k \leq min(|\ddot{C}_{ij}|, |P_k|)\alpha_i^k \quad \forall i, j, k \tag{31}$$

$$\ddot{c}_{ij}^k \leq min(|\ddot{C}_{ij}|, |P_k|)\beta_j^k \quad \forall i, j, k \tag{32}$$

$$\sum_i \ddot{a}_i^k + \sum_j \ddot{b}_j^k + \sum_{i,j} \ddot{c}_{ij}^k + \ddot{d}^k = \sum_{r \in P_k} \delta_r \quad \forall k \tag{33}$$

$$\delta_r + \sigma_v \leq 1 \quad \forall k; \forall r \in P_k; \forall v \in r \tag{34}$$

$$\delta_r = \sum_{e \in E_v^{out}} f(e) \quad \forall k; \forall r \in P_k; \forall \text{DP pin node } v \in r \tag{35}$$

$$\sum_{e \in E_v^{in}} f(e) = \sum_{e \in E_v^{out}} f(e) \quad \forall \text{DP center node } v \tag{36}$$

$$\sum_{e \in \ddot{E}_{inter}^b} f(e) = \sum_i |\ddot{A}_i| + \sum_j |\ddot{B}_j| + \sum_{i,j} |\ddot{C}_{ij}| + |\ddot{D}| \tag{37}$$

$$f(e) \leq 1 \quad \forall e \in \ddot{E}_{inter} \cup \ddot{E}_{T'T''} \tag{38}$$

$$\delta_r \in \{0, 1\} \quad \forall k; \forall r \in P_k. \tag{39}$$

Constraint (24) ensures that the total number of single-ended signals and differential signals consumed in a bank does not exceed the capacity of this bank. Constraints (25)–(28) are the demand constraints for differential signals. Similarly, constraints (29)–(32) are the allowance constraints for differential signals.

[1] For example, in Altera Stratix I EP1S80 B956 FPGA, differential pairs are only allowed to be placed in bank 1,2,5,6. So, $|P_1|, |P_2|, |P_5|, |P_6| \neq 0$. In Xilinx Virtex-6 FF1760 LX760 FPGA, differential pairs are allowed to be placed in all banks. So, $|P_1|, ..., |P_{30}| \neq 0$

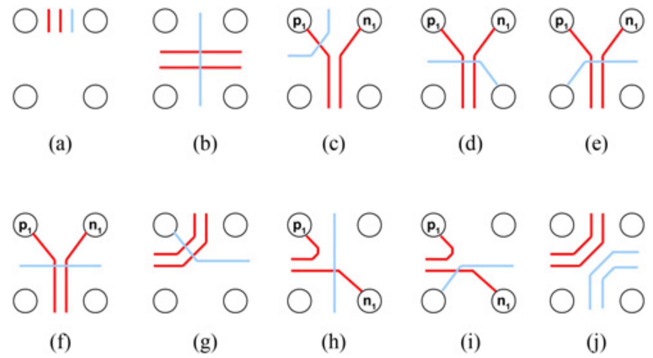


Fig. 11. Illegal cases in escape routing of single-ended signals and differential pairs if we assume that O-cap = 2 and D-cap = 3. The red lines denote the routing of a differential pair while each light blue line denotes the routing of a single-ended signal.

Constraint (33) corresponds to the differential-pair pin assignment constraint for each bank. We introduce a 0 to 1 integer variable δ_r for each pin-pair r that can be assigned to a differential signal in the FPGA. $\delta_r = 1$ if the two pins of the pin-pair r are assigned to a differential signal, otherwise $\delta_r = 0$. Constraint (33) ensures that the number of pin-pairs used in bank k for differential signals must equal to the total number of differential signals assigned to bank k. Constraint (34) prevents conflict in the pin assignment for single-ended signals and differential signals. Constraint (34) guarantees that if the two pins of a pin-pair are assigned to a differential signal, then none of them can be assigned to a single-ended signal. On the other hand, if one of the two pins of a pin-pair is assigned to a single-ended signal, these two pins cannot be assigned to a differential signal. Constraints (35) and (36) correspond to the flow conservation constraints for the DP pin nodes and the DP center nodes, respectively. Constraint (37) guarantees that the number of differential signals which escape to the boundaries must be equal to the total number of differential signals.

Finally, constraint (38) is the capacity constraint for DP intertile edges and DP center node T. Similar to the center node shown in Fig. 10, we split the DP center node T in Fig. 7 into node T' and node T''. The node capacity of DP center node T is equal to the capacity of edge $T'T''$.

As single-ended signals travel in one network and differential pairs travel in another, the escape routing solution may not be legal. Hence, we must exclude the illegal escape routing solution by imposing additional constraints. Fig. 11 shows the illegal cases in escape routing of single-ended signals and differential pairs that we need to avoid. Without loss of generality, we only discuss one orientation for each case but it is not hard to see that other orientation can be handled in the same way. Here we assume O-cap = 2 and D-cap = 3 as this is a common combination of capacity in general PCB escape routing [9]. We can add the following constraints to eliminate the illegal escape routing solutions shown in Fig. 11:

$$f(\dot{e}_1) \leq 2 - 2f(\ddot{e}_1) \tag{40}$$

$$f(\dot{e}_1) \leq 2 - f(\ddot{e}_2) - f(\ddot{e}_4) \tag{41}$$

$$f(\dot{e}_3) \leq 2 - f(\ddot{e}_2) - f(\ddot{e}_4) \tag{42}$$

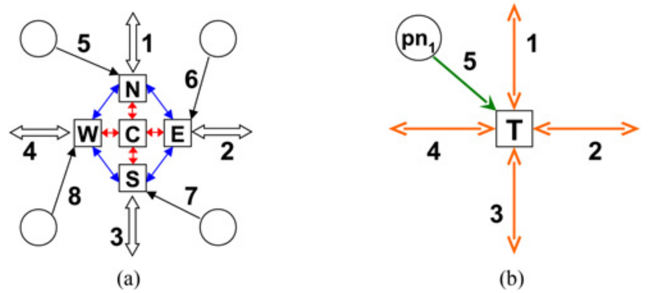


Fig. 12. (a) Single-ended signal routing network model with labeled edges. (b) Differential-pair routing network model with labeled edges.

$$f(\dot{e}_1) \leq 2 - 2f(\ddot{e}_5) \tag{43}$$

$$f(\dot{e}_4) \leq 2 - f(\ddot{e}_5) - f(\ddot{e}_3) \tag{44}$$

$$f(\dot{e}_2) \leq 2 - f(\ddot{e}_5) - f(\ddot{e}_3) \tag{45}$$

$$f(\dot{e}_5) \leq 2 - f(\ddot{e}_1) - f(\ddot{e}_4) \tag{46}$$

$$f(\dot{e}_3) \leq 2 - f(\ddot{e}_5) - f(\ddot{e}_4) \tag{47}$$

$$f(\dot{e}_8) \leq 2 - f(\ddot{e}_5) - f(\ddot{e}_4) \tag{48}$$

$$f(e) \in N \quad \forall e \in \dot{E}_{inter} \cup \ddot{E}_{inter}. \tag{49}$$

We use $\dot{e}_i$ to denote the edge labeled with i in the single-ended signal routing network model shown in Fig. 12(a) and $\ddot{e}_i$ to denote the edge labeled with i in the differential-pair routing network model shown in Fig. 12(b). Constraint (40) eliminates the illegal routing solution for O-cap violation as shown in Fig. 11(a). If the interval of a tile is passed through by a differential pair, single-ended signal cannot pass through this interval again. Therefore, if $f(\ddot{e}_1) = 1$, constraint (40) will force $f(\dot{e}_1) \leq 0$. Constraints (41) and (42) eliminate the crossing in the routing between a differential pair and a single-ended signal as shown in Fig. 11(b). If $f(\ddot{e}_2) = f(\ddot{e}_4) = 1$, constraint (41) will force $f(\dot{e}_1) \leq 0$ and constraint (42) will force $f(\dot{e}_3) \leq 0$. Similarly, constraints (43)–(45) eliminate the crossing in the routing for the cases shown in Figs. 11(c)–(e), respectively. The illegal case that is similar to the case (d) or (e) but the single-ended signal is routed horizontally from the left to the right shown in Fig. 11(f) can also be eliminated by either constraint (44) or (45). Constraint (46) eliminates the case shown in Fig. 11(g). When a differential pair is routed from the top to the left in a tile, single-ended signal cannot escape to this tile from the top left pin in the tile. Fig. 11(h) and (i) show the crossing in the merging tile of a differential pair which has diagonal arrangement and escapes to the left of this tile. For case (h), we use constraint (47) to prevent a single-ended signal passing through the bottom of this tile. For case (i), we use constraint (48) to prevent a single-ended signal escaping to this tile from the bottom left pin in the tile. We do not need to add additional constraint to eliminate the D-cap violation shown in Fig. 11(j) because constraint (42) can also eliminate it by forcing only one single-ended signal to pass through the diagonal of this tile. After introducing constraints (40)–(48), the flow in each edge is no longer guaranteed to be integral, so we need to impose integer constraint (49) for the flow variables.

D. Speeding Up the ILP

We note that constraints (1) and (24) are not necessary constraints but they can reduce the runtime for solving the ILP. They are the upper bound constraints for bounding the number of signals that can be assigned in each bank. They are not necessary constraints because constraints (14), (33)–(34) will guarantee that the total number of signals consumed in a bank must not exceed the capacity of this bank.

We may solve the problem of simultaneous pin assignment and escape routing for single-ended signals and differential pairs more efficiently with little loss of quality if we assume that a tile can only be passed through by either a differential pair or single-ended signals in escape routing. In this way, we do not need to consider the illegal cases shown in Fig. 11 and we can replace constraints (40)–(49) by the following constraints:

$$f(e) \leq (1-h_t)\textit{O-cap} \quad \forall\text{tile } t; \forall e \in \dot{E}_{inter} \text{ s.t. } e \text{ is in } t \quad (50)$$

$$f(e) \leq h_t \quad \forall\text{tile } t; \forall e \in \ddot{E}_{T'T''} \text{ s.t. } e \text{ is in } t \quad (51)$$

$$\begin{aligned} &f(e) \leq 1-h_t \quad \forall\text{tile } t; \forall e \in E_v^{out} \\ &\text{s.t. } v \text{ is a pin node and } e \text{ is in } t \end{aligned} \quad (52)$$

$$h_t \in \{0, 1\} \quad \forall\text{tile } t. \quad (53)$$

We introduce another 0 to 1 integer variable h_t for each tile t. If $h_t = 1$, it means tile t is reserved for a differential pair, so the capacity of the intertile edges in tile t of the single-ended signal routing network will be forced to be zero by constraint (50) while the capacity of the edge $T'T''$ in tile t of the differential-pair routing network will become one by constraint (51). On the other hand, $h_t = 0$ means tile t is reserved for single-ended signals, so the capacity of the intertile edges in tile t of the single-ended signal routing network will be O-cap by constraint (50) while the capacity of the edge $T'T''$ in the differential-pair routing network will be forced to be zero by constraint (51). Constraint (52) is not a necessary constraint but it can further reduce the runtime. It means that if a tile is reserved for a differential pair, we immediately forbid any single-ended signal to escape in this tile from the four pins that form the tile.

When all 0 to 1 integer variables α_i^k, β_j^k, δ_r and h_t are fixed, this simplified formulation reduces to a minimum cost network flow program. It will automatically yield an optimal solution with integral values for $\dot{a}_i^k$, $\dot{b}_j^k$, $\dot{c}_{ij}^k$, $\dot{d}^k$, $\ddot{a}_i^k$, $\ddot{b}_j^k$, $\ddot{c}_{ij}^k$, $\ddot{d}^k$, σ_v and $f(e)$ using any ILP solver based on the simplex algorithm. This simplified formulation is expected to be more efficient as it requires fewer integer variables and constraints than the exact formulation.

E. Handling Multiple Layers

In PCB escape routing, using only a single layer may not be able to successfully escape all the I/O signals if the number of I/O signals is large. Consequently, using multiple layers is a choice for handling such a problem. In this section, we will show that our ILP formulation can be extended to handle multiple layers.

Suppose there are multiple layers, the routing network is the same for each layer. We can simply rewrite constraints (15) and (35) as follows:

$$\sigma_v = \sum_{e^m \in E_v^{out}} f(e^m) \quad \forall\text{pin node } v \quad (54)$$

$$\delta_r = \sum_{e^m \in E_v^{out}} f(e^m) \quad \forall k; \forall r \in P_k; \forall\text{DP pin node } v \in r \quad (55)$$

where e^m is the edge e in layer m and $f(e^m)$ is the number of signals passing through edge e in layer m. Constraints (16)–(18), (20)–(23), (36)–(38) are the same except that $f(e)$ is replaced by $f(e^m)$. Constraints (50)–(53) are the same except that $f(e)$ is replaced by $f(e^m)$ and h_t is replaced by h_t^m. Here, h_t^m corresponds to the tile t in layer m.

For multiple layers solution, the maximum number of nets that can escape in one layer can be calculated by the following equation:

$$N = t \times \textit{O-cap} \quad (56)$$

where N is the maximum number of nets that can escape in one layer and t is the number of the escape intervals on the boundaries of the FPGA device. An escape interval is an interval between two adjacent boundary pins.

The lower bound of the number of layers required can be calculated as follows:

$$L = \lceil \frac{n-p}{N} \rceil \quad (57)$$

where L is the lower bound of the number of layers, n is the number of I/O signals and p is the number of user I/O pins on the boundaries of the FPGA device.

The complexity of the ILP formulation depends on the number of constraints and the number of variables. Assume the size of the FPGA pin array is $x \times y$. There are k banks and r pin-pairs in the FPGA. The number of class A, class B and class C single-ended and differential signals supported by the FPGA are i, j and g. The number of routing layers is z. The number of edges for each tile in the single-ended signal routing network and the differential-pair routing network are no more than 29 and 11, respectively. For the exact formulation, the number of constraints is no more than $5k+2(i+j+g+1)+2(i+j+g)k+3r+2xy+55xyz$ and the number of variables is no more than $(3i+3j+2g+2)k+r+xy+40xyz$. For the simplified formulation, the number of constraints is less than the exact formulation by $6xyz$ while the number of variables is more than the exact formulation by xyz.

IV. Experimental Results

We implemented our methods in the C++ programming language on a Linux machine with two 3.4GHz Intel Xeon CPUs and 2GB memory. The Gurobi optimizer [16] is employed as our ILP solver. Extensive experiments were done on four different types of FPGAs.

TABLE IV

COMPARISON WITH A TWO-STAGE APPROACH ON ALTERA STRATIX I FPGA

Testcase	#single-ended signals	Quartus II 10.0 + [5]			Ours			
		min. #layers	WL	runtime (s)	min. #layers	WL	WL Imp.(%)	runtime (s)
T1	297	2	2096	11+0.06	1	575	72.6	2.22
T2	314	2	2201	16+0.07	2	624	71.6	6.02
T3	427	2	3560	16+0.08	2	1343	62.3	5.51
T4	461	2	3632	29+0.07	2	1639	54.9	19.10
T5	466	2	3686	118+0.08	2	1684	54.3	27.59
T6	481	2	3693	20+0.07	2	1833	50.4	5.05
T7	501	2	3814	24+0.07	2	2143	43.8	37.25
						Average:	58.6	

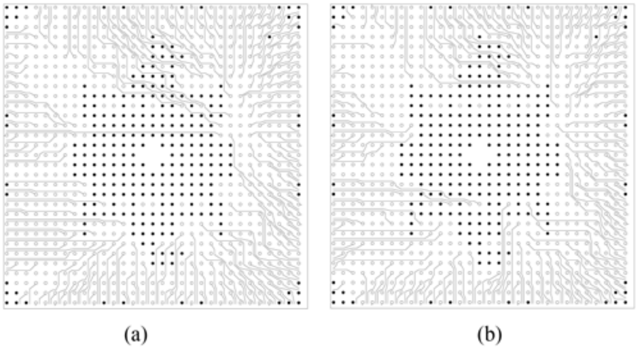

(a) (b)

Fig. 13. Results for T5 by a two-stage approach. (a) Layer 1. (b) Layer 2. Non-user I/O pins are denoted by solid black circles.

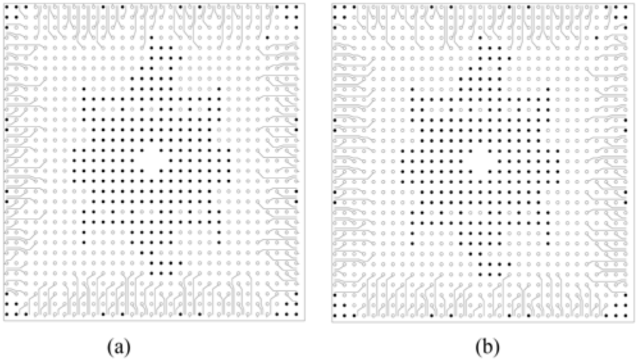

(a) (b)

Fig. 14. Results for T5 by our algorithm. (a) Layer 1. (b) Layer 2. Nonuser I/O pins are denoted by solid black circles.

First, pin assignment and escape routing without differential pairs is performed for the Altera Stratix I EP1S80 B956 FPGA. It has 956 pins in total and 691 user I/O pins in 12 I/O banks. It supports four V_{cco} and seven V_{ref} voltage levels. The test cases T1-T7 for single-ended signals are generated by a test case generator provided by Altera. Table IV shows the experimental results for comparing with the traditional two-stage approach and our method. In the two-stage approach, we use an industrial FPGA CAD tool Altera Quartus II 10.0 to do the FPGA pin assignment and then use an optimal escape routing tool [5] to perform the PCB escape routing for wirelength minimization. O-cap = 2 and D-cap = 3 are used in this experiment.

The minimum number of layers required, total wirelength and runtime are reported in Table IV. We can see from the experimental results that our method outperforms the two-stage approach by a large margin. The wirelength by our method is much shorter in all test cases and we can achieve an average 58.6% wirelength improvement over the two-stage approach. Our method can obtain a solution with one less layer for T1. In addition, our method is efficient and each test case can be done successfully within one minute. Fig. 13 and 14 show the pin assignment and escape routing results of the two-stage approach and our algorithm for case T5.

Our second experiment is on test cases with both single-ended signals and differential pairs. The Xilinx Virtex-6 FF1760 LX760 FPGA which has over one thousand user I/O pins is targeted. The FPGA has 1760 pins in total and 1200 user I/O pins. It supports four V_{cco} and five V_{ref} voltage levels. It has 30 I/O banks and differential pairs are allowed to be placed in any bank. Fig. 2 shows the I/O bank diagram for Xilinx Virtex-6 FPGA.

Table V shows the experimental results on Xilinx Virtex-6 FPGA considering differential pairs. We use O-cap = 2 and D-cap = 3 in this experiment. We sort the test cases according to the number of pins used in ascending order. The average result of every ten test cases is shown in each row. In addition, we implemented a competitive two-stage approach for comparison. In the two-stage approach, we first solve the ILP with constraints (1)–(14) and (24)–(35) only to get a pin assignment for both single-ended and differential signals. Here, we minimize the total distance of each assigned pin to the nearest boundary in the objective function during the pin assignment as a means to reduce the escape routing wirelength. After that, we apply the negotiation based routing scheme proposed in [13] to perform the escape routing for both single-ended and differential signals to minimize the total escape routing wirelength. For correctly modeling the D-cap, we adopted the routing network model of [5] in the escape routing phase of this two-stage approach. Columns 2–4 show the numbers of single-ended signals, differential pairs and pins used in each group of test cases. The average minimum number of layers, WL and runtime are reported for both the two-stage approach and our approach. Moreover, we also implemented the simplified formulation under the assumption that a tile can only be passed through by either a differential pair or single-ended signals. We found that for all cases that we tested, the optimal solution of the simplified formulation is as good as the optimal solution of the exact formulation. So we only report the runtime of the simplified formulation (denoted as runtime*). From the experimental results, our simultaneous approach can achieve better result than the two-stage approach in both minimum number of layers required and WL with reasonable runtime overhead.

TABLE V

COMPARISON WITH A TWO-STAGE APPROACH ON XILINX VIRTEX-6 FPGA USING BENCHMARKS WITH DIFFERENTIAL PAIRS

				Pin assignment + [13]			Ours			
Testcase	#single-ended signals	#DPs	#pins used	avg. min. #layers	avg. WL	avg. runtime(s)	avg. min. #layers	avg. WL	avg. runtime(s)	avg. runtime(s)*
R1-R10	480-616	51-110	700-736	2.90	3159.10	33.94	2.00	3126.60	199.82	118.64
R11-R20	461-584	82-139	737-752	3.00	3394.00	26.70	2.00	3831.60	308.61	118.41
R21-R30	514-656	50-130	756-780	3.00	3666.70	25.52	2.90	3652.90	127.80	168.32
R31-R40	534-666	61-125	780-791	3.00	3839.00	26.72	3.00	3765.50	145.56	176.87
R41-R50	533-670	67-126	792-814	3.00	4074.40	27.91	3.00	4000.40	127.13	79.36
R51-R60	520-671	80-152	819-836	3.00	4395.70	29.06	3.00	4306.20	125.28	67.72
R61-R70	507-707	66-170	838-857	3.00	4659.90	46.81	3.00	4552.60	152.43	140.95
R71-R80	573-676	100-145	857-886	3.00	5000.10	74.02	3.00	4856.40	274.89	247.48
R81-R90	609-828	47-149	892-949	3.00	6192.10	157.64	3.00	5618.40	225.48	146.52
R91-R100	666-893	87-166	956-1125	3.60	8774.90	214.56	3.50	8234.80	424.14	299.75
Overall average:				3.05	4715.59	66.29	2.84	4594.54	211.11	156.40
Overall average of the cases with the same #layers:				3.05	5054.73	76.72	3.05	4816.28	195.21	162.60

TABLE VI

EXPERIMENTAL RESULTS OF OUR ALGORITHM ON XILINX VIRTEX-6 FPGA WITH O-CAP = 3 AND D-CAP = 5

Testcase	avg. min. #layers	avg. WL	avg. runtime(s)
R1-R10	2.00	3001.90	96.68
R11-R20	2.00	3303.80	49.87
R21-R30	2.00	3591.30	98.43
R31-R40	2.00	3767.60	768.25
R41-R50	2.20	3998.60	69.45
R51-R60	2.50	4306.20	122.29
R61-R70	3.00	4552.60	103.71
R71-R80	3.00	4856.40	294.24
R81-R90	3.00	5577.00	81.61
R91-R100	3.10	7813.70	149.87
Overall average:	2.48	4476.91	183.44

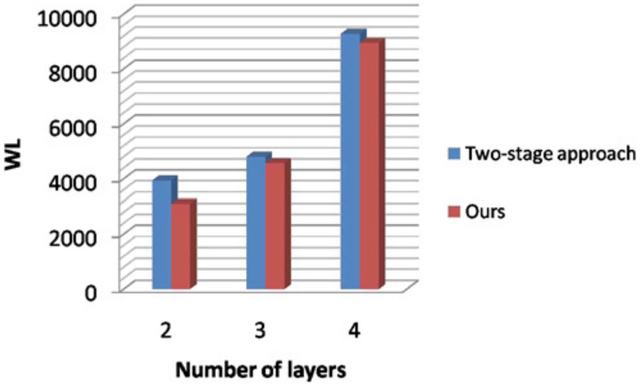


Fig. 15. Wirelength comparison under the test cases with the same number of layers in the experiment shown in Table V.

Among the 100 test cases, we routed 21 test cases using one less layer than the two-stage approach. For a fair comparison of WL, we compare the average WL of the test cases which require the same number of layers. We can achieve an average 4.72% WL improvement over the two-stage approach. For the group R81-R90, the average WL improvement and maximum WL improvement are 9.26% and 15.97%, respectively. We observe that the number of pins used in the test cases of this group is near the maximum number of pins which can be successfully escaped using only three layers. The escape routing of these test cases is very dense and many wires need to detour for a successful escape. Therefore, escape routing will critically depend on the pin assignment for these test cases, and our simultaneous approach can achieve a much better WL than the two-stage approach. We also compare the average WL of the test cases according to the number of layers required in the chart shown in Fig. 15. For the cases which require 2, 3, and 4 layers, we can achieve an average 21.28%, 4.69% and 3.51% WL improvement over the two-stage approach.

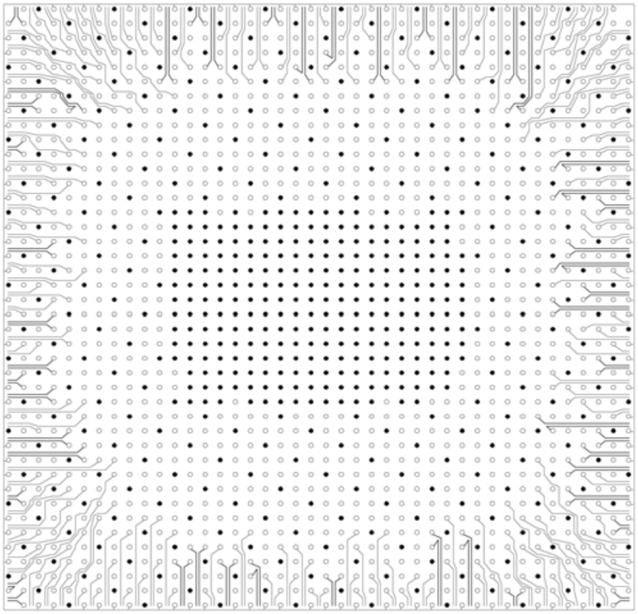


Fig. 16. Layer 1 results with O-cap = 2 and D-cap = 3 of R19 which has 93 differential pairs and 565 single-ended signals. The routing of differential pairs is denoted by bold lines. Nonuser I/O pins are denoted by solid black circles.

For most groups of test cases, the simplified formulation is more efficient than the exact formulation as it involves fewer integer variables and constraints than the exact formulation. However, for groups R21-R30 and R31-R40, the exact

TABLE VII
COMPARISON WITH A TWO-STAGE APPROACH ON ALTERA STRATIX V FPGA AND XILINX VIRTEX-5 FPGA USING BENCHMARKS WITH DIFFERENTIAL PAIRS

		Pin assignment + [13]			Ours		
Testcase	FPGA	avg. min. #layers	avg. WL	avg. runtime(s)	avg. min. #layers	avg. WL	avg. runtime(s)
R1-R10	Stratix V	3.00	11093.80	234.23	3.00	10158.50	322.88
R11-R20	Stratix V	3.00	12203.00	258.00	3.00	11234.10	328.13
R21-R30	Stratix V	3.00	13071.20	254.56	3.00	12344.80	426.35
R31-R40	Stratix V	3.00	13722.00	266.31	3.00	13031.90	351.37
R41-R50	Stratix V	3.00	14592.40	269.62	3.00	13916.50	450.46
R51-R60	Virtex-5	3.00	4428.60	65.52	3.00	4303.40	81.60
R61-R70	Virtex-5	3.00	4682.10	83.79	3.00	4543.10	231.29
R71-R80	Virtex-5	3.00	5003.30	80.71	3.00	4851.10	290.61
R81-R90	Virtex-5	3.00	5807.50	158.54	3.00	5607.70	177.05
R91-R100	Virtex-5	3.50	8623.60	258.47	3.50	8215.70	295.00
Overall average:		3.05	9322.75	192.98	3.05	8820.68	295.47

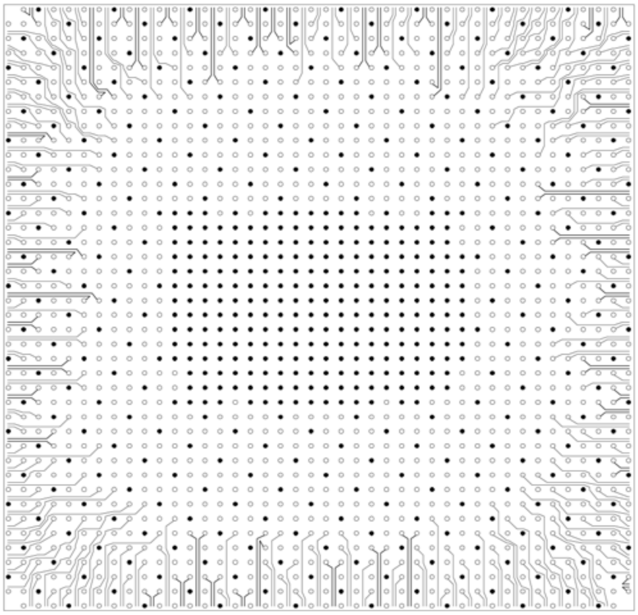


Fig. 17. Layer 2 results with O-cap = 2 and D-cap = 3 of R19 which has 93 differential pairs and 565 single-ended signals. The routing of differential pairs is denoted by bold lines. Nonuser I/O pins are denoted by solid black circles.

formulation is a little faster than the simplified formulation. One reason is that the total number of variables in the simplified formulation is more than the exact formulation so we cannot absolutely claim that the simplified formulation is always faster than the exact formulation. On average the simplified formulation is 25.92% faster to solve than the exact formulation. Figs. 16 and 17 show the pin assignment and escape routing results on layers 1 and 2 of our algorithm on Xilinx Virtex-6 FPGA for case R19.

In addition, we show the results of our simplified formulation when O-cap and D-cap are increased to 3 and 5 in Table VI. We change the node capacity of the DP center node to be 2 and rewrite constraint (51) to be $f(e) \leq 2h_t$. We observe that the average number of layers required and the average WL are smaller than those under O-cap = 2 and D-cap = 3 as shown in Table V. It is because larger O-cap and D-cap will increase the amount of routing resources per layer, and in some cases it can reduce the number of detours in escape routing as more signals are allowed to pass through each tile.

In order to show that our method is equally good for different types of FPGAs, we further tested it on another two very large FPGAs and show the results in Table VII. For the Altera Stratix V 5SEE9F45 FPGA, it has 20 I/O banks, 1932 pins in total and 840 user I/O pins. For the Xilinx Virtex-5 FF1760 LX330 FPGA, it has 32 I/O banks, 1760 pins in total and 1200 user I/O pins. For both FPGAs, differential pairs are allowed to be placed in any bank. We assume O-cap = 2 and D-cap = 3. As the Stratix V FPGA has 840 user I/O pins only, experiments on it were done with R1-R50. For larger test cases R51-R100, we ran them on the Virtex-5 FPGA. We compare our algorithm with the same two-stage approach as before. Comparing with the competitive two-stage approach, our algorithm can reduce the wirelength by 5.39% on average with reasonable runtime overhead.

V. CONCLUSION

In this paper, we proposed a method to simultaneously solve the problem of constrained pin assignment and escape routing for FPGA-PCB co-design. To the best of our knowledge, this is the first work that simultaneously considers the pin assignment problem and the escape routing problem for FPGA on a PCB. Moreover, both differential pairs and single-ended signals are handled together. Experimental results demonstrate that our method can reduce the number of PCB layers required and/or use shorter wirelength when compared with competitive two-stage approaches.

REFERENCES

[1] S. I. Lei and W. K. Mak, "Simultaneous constrained pin assignment and escape routing for FPGA-PCB codesign," in *Proc. FPL*, 2011, pp. 435–440.

[2] J. Anderson, J. Saunders, S. Nag, C. Madabhushi, and R. Jayaraman, "A placement algorithm for FPGA designs with multiple I/O standards," in *Proc. Int. Conf. Field-Programmable Logic Appl.*, (LNCS 1896), 2000, pp. 211–220.

[3] W. K. Mak, "I/O placement for FPGAs with multiple I/O standards," *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 23, no. 2, pp. 315–320, Feb. 2004.

[4] W. K. Mak and C. L. Lai, "On constrained pin-mapping for FPGA-PCB codesign," *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 25, no. 11, pp. 2393–2401, Nov. 2006.

[5] T. Yan and M. D. F. Wong, "A correct network flow model for escape routing," in *Proc. DAC*, 2009, pp. 332–335.

[6] L. Luo and M. D. F. Wong, "Ordered escape routing based on Boolean satisfiability," in *Proc. ASPDAC*, 2008, pp. 244–249.

[7] L. Luo and M. D. F. Wong, "On using SAT to ordered escape problems," in *Proc. ASPDAC*, 2009, pp. 594–599.

[8] L. Luo, T. Yan, Q. Ma, M. D. F. Wong, and T. Shibuya, "B-Escape: A simultaneous escape routing algorithm based on boundary routing," in *Proc. ISPD*, 2010, pp. 19–25.

[9] Q. Ma, T. Yan, and M. D. F. Wong, "A negotiated congestion based router for simultaneous escape routing," in *Proc. ISQED*, 2010, pp. 606–610.

[10] H. Kong, T. Yan, and M. D. F. Wong, "Optimal simultaneous pin assignment and escape routing for dense PCBs," in *Proc. ASPDAC*, 2010, pp. 275–280.

[11] J. W. Fang and Y. W. Chang, "Area-I/O flip-chip routing for chip-package co-design," in *Proc. ICCAD*, 2008, pp. 518–522.

[12] J. W. Fang, I. J. Lin, Y. W. Chang, and J. H. Wang, "A network-flow-based RDL routing algorithm for flip-chip design," *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 26, no. 8, pp. 1417–1429, Aug. 2007.

[13] T. Yan, P. C. Wu, Q. Ma, and M. D. F. Wong, "On the escape routing of differential pairs," in *Proc. ICCAD*, 2010, pp. 614–620.

[14] T. H. Li, W. C. Chen, X. T. Cai, and T. C. Chen, "Escape routing of differential pairs considering length matching," in *Proc. ASPDAC*, 2012, pp. 139–144.

[15] J. W. Fang, K. H. Ho, and Y. W. Chang, "Routing for chip-package-board co-design considering differential pairs," in *Proc. ICCAD*, 2008, pp. 512–517.

[16] Gurobi. *Gurobi optimizer*. [Online]. Available: http://www.gurobi.com

Seong-I Lei received the B.S. and M.S. degrees in computer science from National Tsing Hua University, Hsinchu, Taiwan, in 2006 and 2008, respectively. He is currently pursuing the Ph.D. degree at the same university.

His current research interests include pin assignment, escape routing, and related very large-scale intergrated physical design automation.

Wai-Kei Mak received the B.S. degree from the University of Hong Kong, Hong Kong, in 1993, and the M.S. and Ph.D. degrees in computer science from the University of Texas at Austin, TX, USA, in 1995 and 1998, respectively.

He is currently a Professor with the Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan. From 1999 to 2003, he was an Assistant Professor with the Department of Computer Science and Engineering, University of South Florida, Sarasota, FL, USA. His current research interests include VLSI physical design automation, and CAD for field-programmable technologies.

Dr. Mak's lab won the first place at the FPT 2008 Logic Block Clustering Contest, the third place at the IEEE CEDA PATMOS 2011 Timing Analysis Contest, and the second place at the TAU 2013 Variation-Aware Timing Analysis Contest. He has served on the Program and/or the Organizing Committees of Asia South Pacific Design Automation Conference, the International Conference on Field Programmable Logic and Applications, and the International Conference on Field-Programmable Technology (FPT). He was the Technical Program Chair of FPT in 2006, and the General Chair of the same conference in 2008. Since 2009, he has been a Steering Committee Member of the International Conference on Field-Programmable Technology.