

Bus Assignment Considering Flexible Escape Routing for Layer Minimization in PCB Designs

Jin-Tai Yan[✉], Member, IEEE

Abstract—It is necessary for cost consideration to minimize the number of used layers in a PCB design. Traditionally, the number of used layers outside components in bus assignment depends on the locations of the bus pins inside components in escape routing in a PCB design. In this article, the concept of introducing the flexible escape directions in escape routing is considered in bus assignment outside components in a PCB design. Clearly, the flexible consideration of the escape directions inside components can lead to the reduction on the number of used layers in bus assignment outside components. Given a set of buses on a set of components in a PCB design, based on the introduction of the flexible escape directions in escape routing and the construction of the possible bus connections in bus assignment, the two upper bounds of the layer numbers inside and outside components can be first computed. By eliminating the redundant bus connections for the given buses, an integrated algorithm can be further proposed to minimize the number of used layers in a PCB design. Based on the assignment constraints from the intersection relations inside components, the physical connections of the given buses can be assigned onto a minimal set of used layers. Compared with the two-phase algorithm using Yan’s routing algorithm in direction-constrained rectangle escape routing and Yan’s assignment algorithm in bus assignment, the experimental results show that our proposed integrated algorithm uses reasonable CPU time to reduce 32.1% of the layer number for ten tested examples in a PCB design on the average.

Index Terms—Bus assignment, escape routing, layer minimization, PCB design.

I. INTRODUCTION

As IC technology advances, the shrinkage in die size and the increase in functional complexity [1] have made IC designs denser and complex. IC designs have made the pin count in a package increasing. However, new challenges cannot be effectively handled by using traditional routing approaches in high-speed PCB designs. Generally speaking, PCB routing can be divided into two phases: 1) *escape routing* and 2) *bus routing*. In escape routing, pins inside a pin array must be routed to the boundaries of the pin array. It is known that the increase of the pin count makes the escape routing difficult in PCB designs. In bus routing, all the nets inside a single bus are always routed on the same layer between two pin arrays such that the nets in bus routing are detoured to satisfy some length-matching or timing-matching constraints [2]. It is

Manuscript received 4 April 2020; revised 12 October 2020, 28 February 2021, and 10 May 2021; accepted 12 September 2021. Date of publication 15 September 2021; date of current version 19 July 2022. This article was recommended by Associate Editor S. Held.

The author is with the Office of Research and Development, Tainan National University of the Arts, Tainan 72045, Taiwan (e-mail: jtyan@tnnua.edu.tw).

Digital Object Identifier 10.1109/TCAD.2021.3112882

known that the complexity of the length-matching or timing-matching constraints makes bus routing more difficult in PCB designs. In addition, it is necessary for the consideration of the fabrication cost to minimize the number of used layers in a high-speed PCB design.

In the studies of the bus routing on modern high-speed PCB designs for the consideration of signal integrity, the assumption of routing a single bus on a single layer has been used in some published works [3]–[13], [15], [16]. In general, the bus assignment for layer minimization (BALM) in a PCB design can be divided into the *bus assignment inside components* and the *bus assignment outside components* in a PCB design. In the bus assignment inside components, the buses inside a component can be escaped onto the available boundaries on the component for layer minimization. As the pins on any escaped bus are fixed on the boundary of the corresponding component, all the escaped buses can be assigned onto a minimal set of layers in the bus assignment between components.

For the bus assignment inside one component, first, Kong *et al.* [3] proposed optimal algorithms for finding disjoint boundary rectangles on two, three, and four available boundaries inside a larger rectangle. However, the proposed algorithm cannot minimize the number of used layers in bus-oriented escape routing. In rectangle escape routing, Ma *et al.* [4] formulated the rectangle escape problem (REP) and proposed an approximation algorithm to solve the REP. Furthermore, the approximation algorithm of the REP with application to PCB routing is discussed by Ma and Wong [5] in detail. However, the proposed approximation algorithm takes more CPU time in rectangle escape routing. According to the location of the available boundaries in the REP, Yan *et al.* [6] divided the REP into five assignment problems for layer minimization. Furthermore, based on the optimal results in two simple problems, the other problems can be further solved by using two-phase density-reduction-oriented layer assignment. However, the proposed algorithms do not consider the desired escape directions for any bus inside a chip. In the consideration of the different escape directions for any bus, based on the maximal densities of the buses on four boundaries, Yan and Chen [7] proposed an iterative algorithm to reduce the maximal density of the buses escaped to four boundaries in direction-constrained rectangle escape routing. However, the maximal density of the buses escaped to any of four boundaries is only considered in one-directional viewpoint for layer assignment. The flexible selection of the buses under the consideration of one-directional maximum density may not lead to a good solution

for the 2-D layer assignment inside a larger chip. Recently, the problem of finding maximum disjoint set of boundary rectangles with application to PCB routing is further discussed by Ahmadinejad and Zarrabi-Zadeh [8]. Furthermore, a two-phase algorithm [9] is proposed to assign one feasible escape direction onto any bus to minimize the number of used layers in direction-constrained rectangle escape routing.

For the bus assignment outside components in a PCB design, all the buses are connected among components in a PCB design. Based on the bus decomposition and the escape routing result for a given set of buses, Kong *et al.* [10] proposed a bus planner to simultaneously complete the layer assignment of the given buses and the global bus routing. Furthermore, Wu *et al.* [11] also proposed an ILP-based bus planner to simultaneously solve the bus decomposition, the escape routing, the layer assignment and the global bus routing. Additionally, based on the concept of using a component connecting point (CCP) between two components and using the dynamic pin sequence (DPS) for the avoidance of any net crossing, Tsai *et al.* [12] proposed a global topological routing algorithm for single-layer routing. By using the topological routing algorithm layer by layer, the layer assignment of the given bus connections can be obtained. Furthermore, Chin *et al.* [13] introduced the concept of transforming the chords of a circle into the corresponding circle graph for DPS and used Supowit's algorithm [14] to find the maximum number of bus connections. By using Supowit's algorithm layer by layer, the layer assignment of the bus connections in a PCB design can be obtained. However, the DPS-based algorithm and Supowit's algorithm may not minimize the number of used layers for the layer assignment of the bus connections in a high-speed PCB design. Recently, based on the assumption of assigning all the nets inside a single bus on the same layer, Yan [15] proposed an efficient algorithm to minimize the number of used layers for the bus connections in a high-speed PCB design. However, the fixed locations of the bus pins on the component boundaries in escape routing may lead to more used layers in a PCB design.

Basically, the fabrication cost in a PCB design seriously depends on the number of used layers in a PCB design. It is necessary for the consideration of the fabrication cost to minimize the number of used layers in a PCB design. It is known that the number of used layers in a PCB design is from the number of used layers inside components in escape routing and the number of used layers outside components in bus assignment in a PCB design. If the flexible consideration of the escape direction for each given bus is used in escape routing, the number of used layers outside components in bus assignment can be further minimized in a PCB design.

The contributions of this article can be summarized in the following.

- 1) The concept of introducing the flexible escape directions in escape routing is considered in bus assignment in a PCB design. Clearly, the flexible consideration of the escape directions inside components in escape routing can lead to the reduction on the number of used layers in bus assignment outside components.

- 2) Given a set of buses on a set of components in a PCB design, based on the introduction of the flexible escape directions in escape routing and the construction of the possible bus connections in bus assignment, the upper bound of the layer number for the flexible escape connections inside components and the upper bound of the layer number for the possible bus connections outside components can be defined and computed.
- 3) By eliminating the redundant bus connections for the given buses, an integrated algorithm can be proposed to minimize the number of used layers in a PCB design. Based on the assignment constraints from the intersection relations inside components, the physical connections of the given buses can be assigned onto a minimal set of used layers.

The remainder of this article is organized as follows: Section II contains the motivation and problem formulation of the bus assignment considering flexible escape routing in a PCB design. In Section III, an integrated algorithm is proposed to minimize the number of used layers with considering flexible escape routing in a PCB design. In Section IV, the experimental results in the proposed algorithm are listed and compared with the published algorithm for the bus assignment in a PCB design. Finally, the conclusions and further works are summarized in Section V.

II. MOTIVATION AND PROBLEM FORMULATION

If there is no crossing condition between the connections of two given buses in a PCB design, the two buses can be assigned onto the same layer and the two buses can be defined to be *compatible* with each other. If any pair of buses on each available layer is compatible in a PCB design, the assignment of the given buses can be defined as one *valid bus assignment* in the PCB design. Hence, one valid bus assignment in a PCB design can be treated as the partition of the given buses onto some used layers.

It is known that the connections of two given buses that cross each other must be assigned onto two different layers and the connections of two given buses that do not cross each other may be assigned onto the same layer or two different layers on one valid bus assignment in a PCB design. To reduce the fabrication cost in a PCB design, the number of used layers must be considered in the bus assignment of a PCB design.

A. Motivations

In a PCB design, it is assumed that the routing region with IO pins must be located under one component for IO connections. Generally speaking, all the IO pins under one component can be grouped into some independent buses and all the I/O pins inside a given bus must be escaped to the same available boundary. For a given bus inside one component, the minimum rectangular region covering all the bus pins to be escaped to the same available boundary can be represented as a *pin rectangle*. Furthermore, the escape region of a given bus from its pin rectangle to its escaped boundary can be represented as a *projection rectangle* in its escaped direction. If the left, right, top, and bottom boundaries are the available boundaries

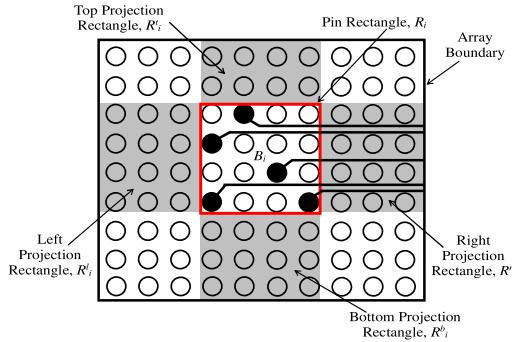


Fig. 1. Pin rectangle and projection rectangles for a given bus B_i .

for a given bus, B_i , the pin rectangle, R_i , and the four projection rectangles, R'^l_i , R'^r_i , R'^t_i , and R'^b_i , of the bus, B_i , can be obtained and illustrated in Fig. 1.

Traditionally, the bus assignment in a PCB design can be divided into two sequential steps: 1) *escape routing* inside components and 2) *bus assignment* outside components. In escape routing inside components, all the buses inside one component can be escaped to the available boundaries to minimize the number of used layers under the component. As all the given buses in a PCB design are escaped to the assigned boundaries, the escape directions of all the buses inside components can be fixed in escape routing. In bus assignment outside components, based on the escape directions of all the buses on the assigned boundaries, the connections of all the given buses outside components can be further assigned to minimize the number of used layers. After completing the escape routing inside components and the bus assignment outside components, the physical connection of a given bus between two components can be constructed from the combination of two escape connections inside two components and one topological bus connection between two sets of bus pins outside two components. It is known that the number of used layers in escape routing inside components and the number of used layers in bus assignment outside components can further decide the final number of used layers in a PCB design. Due to the fixed escape directions of all the buses in escape routing, the number of used layers in bus assignment can be restricted under the constraint of using the fixed escape directions.

Consider the specification of the bus assignment for a set of nine given buses, $B = \{B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_8, B_9\}$, on a set of four components, $C = \{C_1, C_2, C_3, C_4\}$, in a PCB design, C_0 , in Fig. 2(a), if there are four available escape directions for each given bus inside one component, based on the constraint of using only one escape direction for each given bus, the number of used layers inside the components, C_1, C_2, C_3 , and C_4 , can be minimized to be 1, 1, 1, and 1, respectively. Based on the assigned escape directions of the nine given buses, the number of used layers outside the components, C_1, C_2, C_3 , and C_4 , can be further minimized to be 5. Hence, the final number of used layers in a PCB design can be minimized as 5. In the assignment of the nine given buses, two buses, B_2 and B_8 , can be assigned onto the first layer, three buses, B_1, B_3 , and B_5 , can be assigned onto the second layer, two buses, B_6 and B_9 , can be assigned onto the third layer, one bus, B_4 , can be assigned onto the fourth layer, and

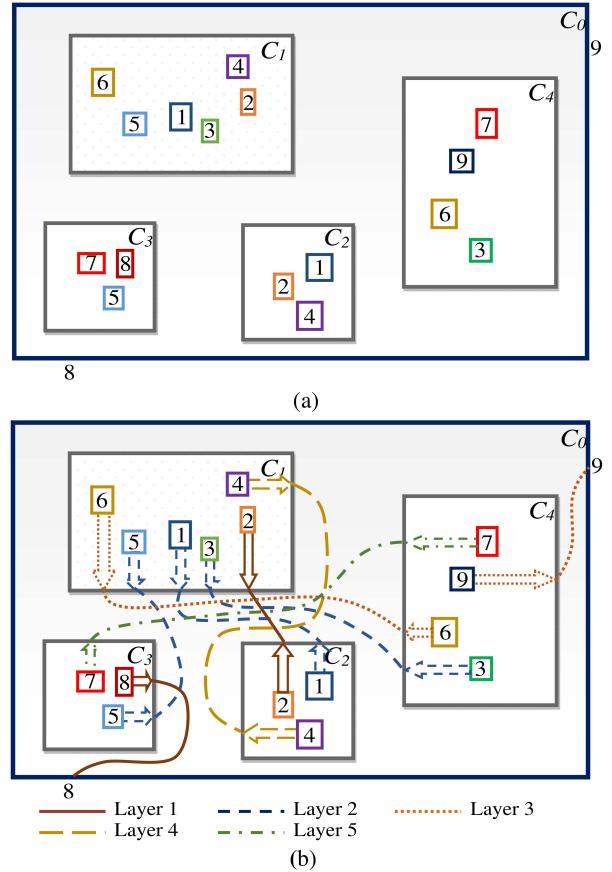


Fig. 2. Escape routing and bus assignment for nine given buses on four components in a PCB design. (a) Specification of bus assignment for nine given buses on four components. (b) Bus assignment onto five used layers for nine given buses.

one bus, B_7 , can be assigned onto the fifth layer as illustrated in Fig. 2(b).

B. Definition of Flexible Escape Directions on Buses

In general, there are four possible escape directions to four available boundaries for any given bus under one component. To satisfy the timing-driven requirement on bus routing between two components, it is desired that the bus connection can be routed by using its shortest path. Given a set of n given buses, $B = \{B_1, B_2, \dots, B_n\}$, on a set of components, $C = \{C_1, C_2, \dots, C_m\}$, in a PCB design, the *bus routing region* of a given bus, B_i , between two components, C_j and C_k , $1 \leq i \leq n$, $0 \leq j, k \leq m$, $j \neq k$ can be defined as a minimum rectangle covering the two pin rectangles, $R_{i,j}$ and $R_{i,k}$, under the components, C_j and C_k . Furthermore, the *flexible escape directions* on the bus, B_i , under the component, C_j or C_k , can be defined as one or two escape directions inside the routing region of the bus, B_i , under the component, C_j or C_k . If the component, C_j , is located on the left or bottom side of the component, C_k , one *flexible direction set*, $FD_{i,0}$, on the given bus, B_i , can be defined for the component, C_j , and the other *flexible direction set*, $FD_{i,1}$, on the given bus, B_i , can be defined for the component, C_k , $FD_{i,0}, FD_{i,1} \subset \{D_L, D_R, D_T, D_B\}$, $1 \leq i \leq n$, where D_L, D_R, D_T , and D_B are the escape directions to the left, right, top, and bottom boundaries, respectively. As a given bus, B_i ,

under the component, C_j , is escaped to one available boundary, the bus name can be labeled on its available boundaries by using the combination of its bus number and its escape direction.

Refer to the location distribution of nine given buses on four components in a PCB design in Fig. 2(a), the bus routing regions of the nine buses can be constructed as illustrated in Fig. 3(a). It is clear that there is only one flexible escape direction on the bus, B_2 , under the components, C_1 and C_2 , the bus, B_3 , under the component, C_4 , the bus, B_5 , under the components, C_1 and C_3 , the bus, B_6 , under the component, C_4 , the bus, B_7 , under the component, C_4 , and the bus, B_8 , under the component, C_3 , and there are two flexible escape directions on the bus, B_1 , under the components, C_1 and C_2 , the bus, B_3 , under the component, C_1 , the bus, B_4 , under the components, C_1 and C_2 , the bus, B_6 , under the component, C_1 , the bus, B_7 , under the component, C_3 , and the bus, B_9 , under the component, C_4 . Based on the definition of the flexible escape directions on two flexible escape sets for a given bus, the 18 flexible escape sets $FD_{1,0}=\{D_R, D_B\}$, $FD_{1,1}=\{D_L, D_T\}$, $FD_{2,0}=\{D_B\}$, $FD_{2,1}=\{D_T\}$, $FD_{3,0}=\{D_R, D_B\}$, $FD_{3,1}=\{D_L\}$, $FD_{4,0}=\{D_R, D_B\}$, $FD_{4,1}=\{D_L, D_T\}$, $FD_{5,0}=\{D_T\}$, $FD_{5,1}=\{D_B\}$, $FD_{6,0}=\{D_R, D_B\}$, $FD_{6,1}=\{D_L\}$, $FD_{7,0}=\{D_R, D_T\}$, $FD_{7,1}=\{D_L\}$, $FD_{8,0}=\{D_T\}$, $FD_{8,1}=\{D_B\}$, $FD_{9,0}=\{D_R, D_T\}$, and $FD_{9,1}=\{D_L\}$, for the nine buses on the four components in the PCB design can be obtained as illustrated in Fig. 3(b). In this figure, the name of each given bus under one component can be labeled on its available boundaries. For example, the names of the given bus B_4 under component C_1 can be labeled as 4R and 4B on the two available boundaries.

C. Problem Formulation

As mentioned above, the constraint of using the fixed escape directions in escape routing inside components may lead to the use of more layers in the bus assignment outside components in a PCB design. To minimize the number of used layers in a PCB design, the possible assignment of the flexible escape directions in escape routing can be further considered in the bus assignment.

Based on the definition of the flexible escape directions on two flexible escape sets for a given bus in the routing specification, given a set of n buses, $B = \{B_1, B_2, \dots, B_n\}$, on a set of components, $C = \{C_1, C_2, \dots, C_m\}$, in a PCB design, it is known that there are two flexible direction sets, $FD_{i,0}$ and $FD_{i,1}$, for each given bus, B_i , $1 \leq i \leq n$, $FD_{i,0}, FD_{i,1} \subset \{D_L, D_R, D_T, D_B\}$. In the bus assignment considering flexible escape routing in a PCB design, the problem of assigning the given buses in a PCB design is to minimize the number of used layers to obtain one valid assignment of n buses, B_1, B_2, \dots, B_n , in B on the m components, C_1, C_2, \dots, C_m , in C under the uniqueness constraint of assigning only one escape direction for each given bus and the noncrossing constraint of two different buses on the same layer.

Based on the assignment of the escape directions, $D0(B_i)$ and $D1(B_i)$, from the flexible direction sets, $FD_{i,0}$ and $FD_{i,1}$, and the assigned layer number, $LN(D0(B_i), D1(B_i))$, of the

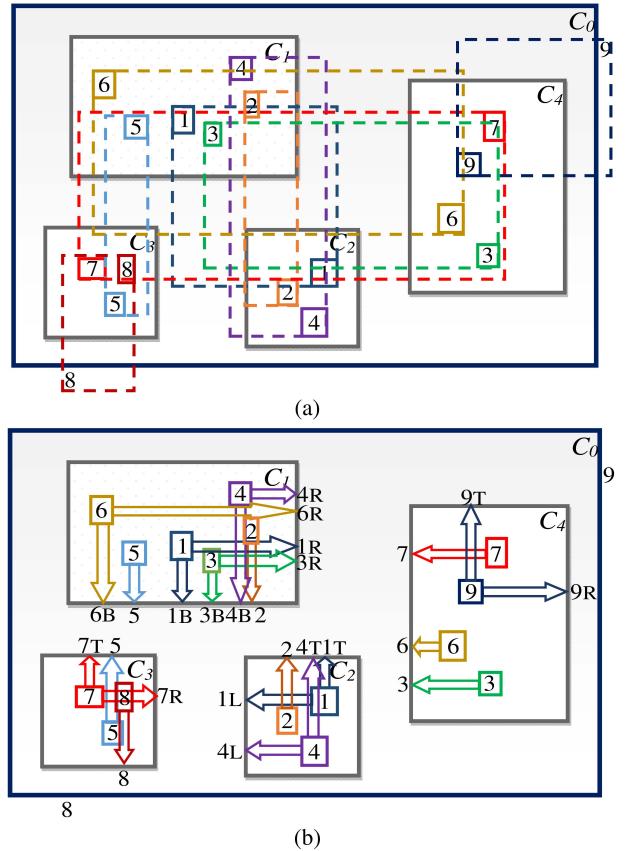


Fig. 3. Definition of flexible escape directions for nine given buses on four components. (a) Construction of nine bus routing regions for nine given buses. (b) Construction of 24 flexible escape directions for nine given buses on four components.

connection, $C(D0(B_i), D1(B_i))$, for each given bus, B_i , $1 \leq i \leq n$, the problem of assigning the given buses considering flexible escape routing in a PCB design can be formatted to minimize the number of used layers, $NL(B_1, B_1, \dots, B_n)$, for one valid assignment, $VA(C(D0(B_1), D1(B_1)), \dots, C(D0(B_n), D1(B_n)))$, of the n buses, B_1, B_2, \dots, B_n , in B as

$$\text{Min } NL(B_1, B_1, \dots, B_n)$$

subject to

- 1) *Uniqueness Constraints:*
 $D0(B_i) \in FD_{i,0}$, $D1(B_i) \in FD_{i,1}$, $1 \leq i \leq n$;
- 2) *Non-crossing Constraint:*
 $LN(D0(B_i), D1(B_i)) \neq LN(D0(B_j), D1(B_j))$,
for $C(D0(B_i), D1(B_i)) \cap C(D0(B_j), D1(B_j)) \neq \emptyset$, $1 \leq i, j \leq n$.

Refer to the specification of the bus assignment for a set of nine given buses, $B = \{B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_8, B_9\}$, on a set of four components, $C = \{C_1, C_2, C_3, C_4\}$ in a PCB design C_0 , in Fig. 1(a), if the flexible escape directions of the nine given buses in Fig. 3(b) are considered in bus assignment outside components, based on the assignment of using only one escape direction for each given bus, the number of used layers for the nine given buses outside components can be minimized to be 3 as illustrated in Fig. 4. Although the number of used layers for the nine given buses inside components can be obtained as 2, the final number of used layers in a PCB

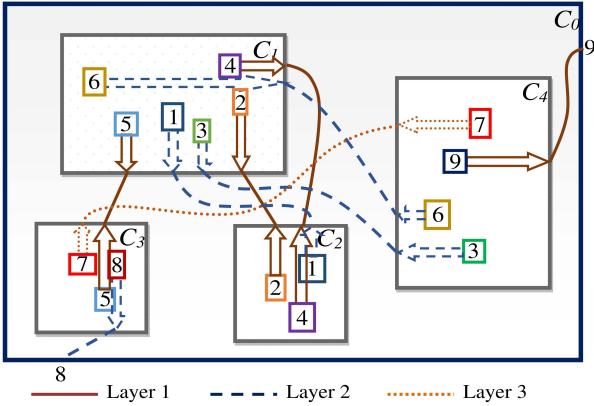


Fig. 4. Bus assignment considering flexible escape routing onto three used layers for nine given buses in a PCB design.

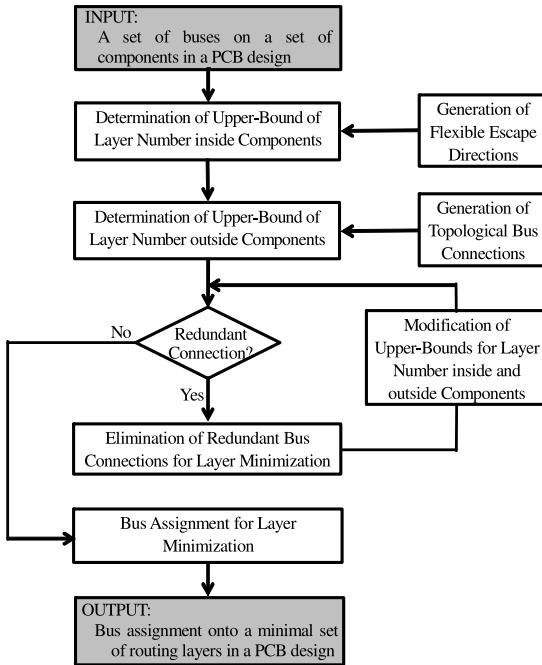


Fig. 5. Design flow of our proposed algorithm in bus assignment considering flexible escape routing for layer minimization.

design can be reduced from 5 to 3. In the assignment of the nine given buses, the four buses, B_2 , B_4 , B_5 , and B_9 , can be assigned onto the first layer, the four buses, B_1 , B_3 , B_6 , and B_8 , can be assigned onto the second layer and bus, B_7 , can be assigned onto the third layer.

III. BUS ASSIGNMENT CONSIDERING FLEXIBLE ESCAPE ROUTING FOR LAYER MINIMIZATION

In the bus assignment considering flexible escape routing, given a set of buses on a set of components in a PCB design, an integrated algorithm can be proposed to assign the given buses onto a minimal set of routing layers and the design flow of the proposed algorithm can be shown in Fig. 5. Basically, the proposed algorithm can be divided into four sequential steps: 1) determination of upper-bound of layer number for flexible escape directions inside components; 2) determination of upper-bound of layer number for possible bus connections

outside components; 3) iterative elimination of redundant bus connections (IERBCs) for layer minimization; and 4) *BALM*.

Based on the definition of the bus routing region for each given bus, the flexible routing directions of each given bus under two components can be defined and the possible connecting ports of the given bus can be treated as the possible bus pins of the given bus on the available boundaries. Under the consideration of using the flexible escape directions and the construction of the possible bus pins on the given buses, the upper bound of the layer number for the flexible escape directions inside components can be obtained. Based on the assignment of all the flexible escape directions inside components, the possible bus connections outside components can be further defined and the upper bound of the layer number for the bus connections outside components can be obtained.

By using the concept of reducing the maximum value of the two upper bounds inside and outside components on the layer number, an iterative elimination process can be proposed to eliminate the redundant escape directions inside components and the redundant bus connections outside components to minimize the number of used layers in a PCB design. Based on the assignment of the final escape directions inside components and the final bus connections outside components, all the given buses in a PCB design can be further assigned onto a minimal set of routing layers.

A. Determination of Upper-Bound of Layer Number for Flexible Escape Directions Inside Components

Given a set of r given buses, $B_{\alpha(1)}, B_{\alpha(2)}, \dots, B_{\alpha(r)}$, on one component, C_i , with their flexible direction sets, $1 \leq \alpha(j) \leq n$, $\alpha(j) \neq \alpha(k)$, $1 \leq i \leq m$, $1 \leq j, k \leq r$, based on the definition of the projection rectangles on the flexible escape directions for each given bus, B_j , under the component, C_i , the possible connection region, $PCR_{i,j}$, of the bus, B_j , under the component C_i , can be defined as the union of the pin rectangle and the projection rectangles of bus, B_j , using its possible flexible escape directions inside the component, C_i . Based on the intersection relation between the possible connection regions, $PCR_{i,j}$ and $PCR_{i,k}$, of two given buses, B_j and B_k , under the component, C_i , a conflict graph $G_i(V_i, E_i)$, on the component, C_i , can be constructed as follows: a vertex, v_j in V_i represents the possible connection region, $PCR_{i,j}$, of a given bus, B_j , and an edge, $e_{j,k}$, in E_i represents the intersection relation between the possible connection regions, $PCR_{i,j}$ and $PCR_{i,k}$, of two given buses, B_j and B_k . It is clear that each intersection relation in a conflict graph, $G_i(V_i, E_i)$, represents the conflict relation of assigning the escape directions of two given buses, B_j and B_k , on the same layer. Hence, the coloring number in a conflict graph, $G_i(V_i, E_i)$, can be used as the upper bound of the layer number for the flexible escape directions inside the component, C_i .

In finding the upper bound of the layer number for the flexible escape directions inside one component, it is known that the coloring number in the conflict graph from the pin and projection rectangles of the given buses under one component can be obtained from the bus assignment [9] in rectangle escape routing. In [9], based on the construction of the projection

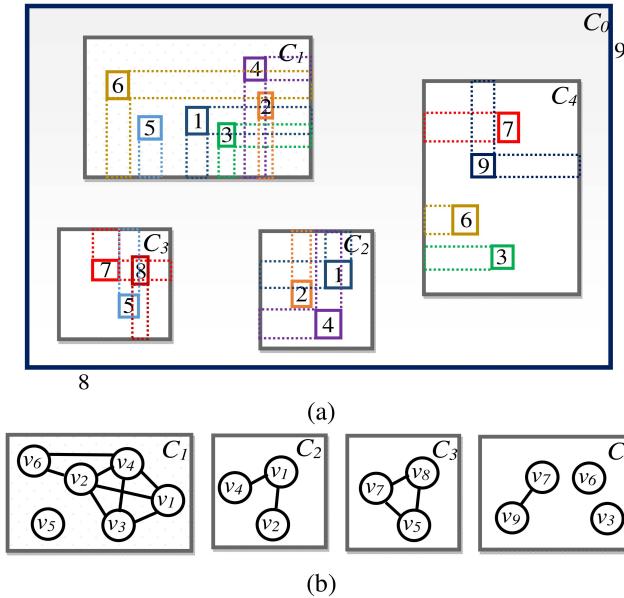


Fig. 6. Possible connection regions for nine given buses on four components in a PCB design. (a) Construction of possible connection regions for nine given buses under four components. (b) Construction of four conflict graphs on four components.

rectangles for the given buses under one component, the routing region under one component can be partitioned into a set of nonoverlapping routing subregions. It is clear that each nonoverlapping subregion may overlap the pin or projection rectangles of some given buses. Given a nonoverlapping subregion, the number of used layers can be obtained as the number of overlapping pin or projection rectangles. Hence, the upper bound of the layer number for the flexible escape directions inside one component can be obtained as the maximum value of the layer numbers for all the partitioned subregions inside one component. Furthermore, the upper bound of the layer number for the flexible escape directions inside components can be obtained as the maximum value of the layer numbers inside all the components.

Refer to the assignment of the 24 flexible escape directions for the nine given buses under four components in Fig. 3(b), the possible connection regions of nine given buses under four components can be constructed as illustrated in Fig. 6(a) and the four conflict graphs on the four components can be constructed as illustrated in Fig. 6(b). Based on the partition results of the routing regions under the four components can be illustrated in Fig. 6(a) and the numbers of the overlapping pin and projection rectangles on all the subregions, the upper bounds of the layer numbers for the flexible escape directions inside the components, \$C_1\$, \$C_2\$, \$C_3\$, and \$C_4\$, can be obtained as 4, 3, 3, and 2, respectively. Hence, the upper bound of the layer number for the flexible escape directions inside components can be obtained as 4.

B. Determination of Upper-Bound of Layer Number for Possible Bus Connections Outside Components

In a PCB design, a given bus can be defined as a set of nets with the same circuit feature between two components. To satisfy the timing-matching constraint in a given bus, all the

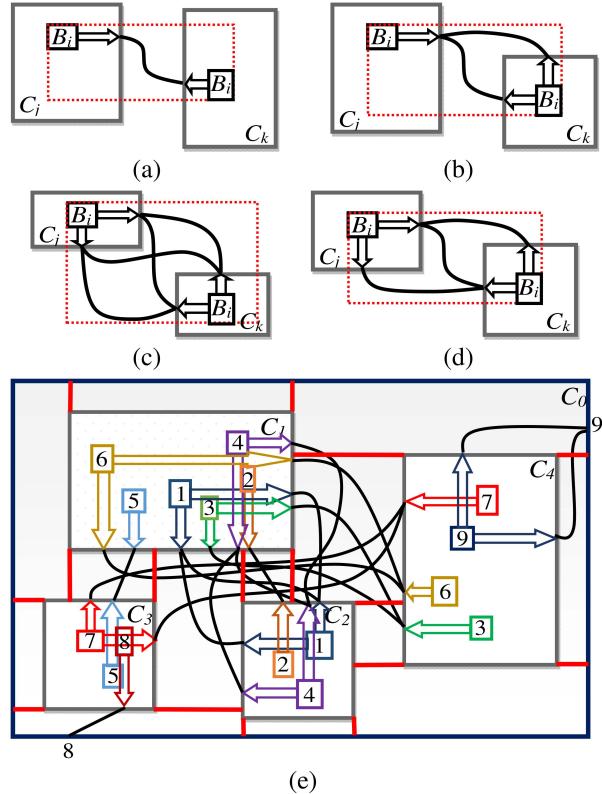


Fig. 7. Construction of possible topological bus connections for nine given buses. (a) Construction of one possible connection for a given bus \$B_i\$. (b) Construction of two possible connections for a given bus \$B_i\$. (c) Construction of four possible connections for a given bus \$B_i\$. (d) Construction of three possible connections for a given bus \$B_i\$. (e) Construction of 17 possible topological bus connections for nine given buses.

nets inside a given bus must be constrained to have similar global wiring paths with no via and assigned on the same layer. Because there are similar wiring characteristics in the nets inside a given bus, a given bus outside components can be treated as a *topological bus connection*. In our previous work [15], it is assumed that any independent net can be also treated as a given bus with only one net. In the bus assignment outside components, all the nets can be divided into a set of buses and the nets inside a given bus must be assigned onto the same layer.

Basically, all the topological bus connections in a PCB design can be globally routed according to the requirements of different area or delay. In constructing a bus connection outside components, first, the routing space among components in a PCB design can be partitioned into a set of rectangular subspaces. For a bus connection, two bus pins between two components can be routed by connecting an adjacent list of rectangular subspaces with the shortest distance. It is known that there are at most two flexible escape directions for one given bus under one component. Based on the locations of at most two bus pins on one given bus under one component, there are one, two, or four possible pin-to-pin topological bus connections for one given bus \$B_i\$, between two components, \$C_j\$ and \$C_k\$, as illustrated in Fig. 7(a)–(c), respectively. For any possible bus connection in one given bus, \$B_i\$, the bus connection can be named the combination of its bus number \$i\$ and

its used escape directions on the bus, B_i , between two components, C_j and C_k . For example, the topological bus connection in the given bus, B_4 , with the escape direction, 4B, under the component, C_1 , and the escape direction, 4L, under the component, C_2 , can be further named as N_{4BL} .

For one possible topological bus connection, N_j , with the special locations of the two bus pins, the bus connection, N_j , can be further deleted as follows: if the adjacent list storing one bus connection, N_i , is fully included in the adjacent list storing the bus connection, N_j , the bus connection, N_j , can be redundant because the crossings of the connection, N_j , covers all the crossings of the connection, N_i . Hence, the bus connection, N_j , cannot be considered in the bus assignment outside components. For example, one possible bus connection for one given bus, B_i , between two components, C_j and C_k , as illustrated in Fig. 7(d) cannot be considered in the bus assignment outside components. Hence, the number of possible bus connections for a given bus, B_i , can be reduced from 4 to 3. Referring to the assignment of the flexible escape directions for the nine given buses under the four components in Fig. 3(b), all the possible topological bus connections of the nine given buses can be constructed as illustrated in Fig. 7(e). It is clear that there is only one possible topological bus connection for the three given buses, B_2 , B_5 and B_8 , there are two possible topological bus connections for the four given buses, B_3 , B_6 , B_7 and B_9 , and there are three possible topological bus connections for the two given buses, B_1 , and B_4 .

As mentioned in our previous work [15], an efficient algorithm can be proposed to minimize the number of used layers and assign the given bus connections onto the used layers for the bus assignment in a PCB design. By using the efficient assignment algorithm [15] on all the possible bus connections outside components in a PCB design, the number of used layers for all the possible bus connections can be obtained by using the proposed assignment algorithm. Hence, the number of used tracks in the track assignment of the represent intervals with covering compatibility for all the possible bus connections can be used as the upper bound of the layer number for the possible bus connections outside components.

In the proposed assignment algorithm [15] on all the possible bus connections outside components, the assignment process can be divided into four sequential steps: 1) *construction of virtual walls*; 2) *transformation of bus connections*; 3) *iterative layer assignment of bus connections*; and 4) *layer assignment of eliminated bus connections*.

1) *Construction of Virtual Walls*: Given a set of p bus connections, $N=\{N_1, N_2, \dots, N_p\}$, on a set of q components, $C=\{C_1, C_2, \dots, C_q\}$, in a PCB design C_0 , an edge-weighted connection graph, $G_w(\{v_0\} \cup V_w, E_w)$, can be constructed in [15] as follows: the vertex, v_0 , represents the PCB design, a vertex, v_i , $1 \leq i \leq q$ in V_w represents one component, C_i , and an edge, $e_{i,j}$, $0 \leq i, j \leq q$, $i \neq j$ in E_w represents the adjacent relation between two components, C_i and C_j . In the connection graph G_w , the weight of an edge $e_{i,j}$, $0 \leq i, j \leq q$, $i \neq j$ in E_w is assigned as the number of bus connections crossing a virtual wall between two components C_i and C_j . To integrate the boundary pins on the given q components and the PCB into the boundary pins on a final component and minimize the number of bus connections crossing the constructed

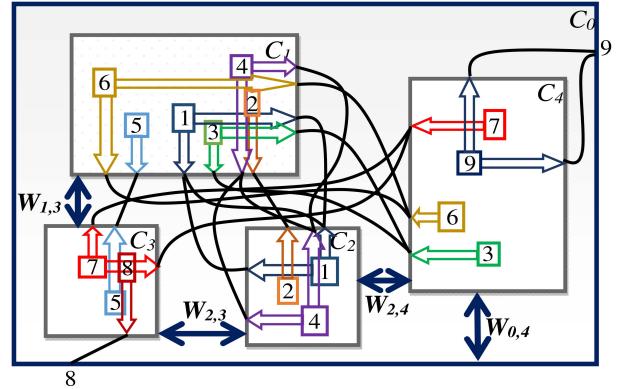


Fig. 8. Construction of four virtual walls for 17 possible bus connections on four components in a PCB design.

virtual walls, a set of virtual walls can be constructed to form a minimum-weight spanning tree of the q components and the PCB by eliminating some unnecessary adjacent relations in the connection graph.

Refer to the 17 possible bus connections, $N_{1RT}, N_{1BT}, N_{1BL}, N_2, N_{3RL}, N_{3BL}, N_{4RT}, N_{4BT}, N_{4BL}, N_5, N_{6RL}, N_{6BL}, N_{7TL}, N_{7RL}, N_8, N_{9T}$ and N_{9R} , on the four components, C_1, C_2, C_3 and C_4 , in a PCB design, C_0 , in Fig. 7(e), based on the adjacent relations on the four components in the PCB design, an edge-weighted connection graph, $G_w(\{v_0\} \cup V_w, E_w)$, can be constructed, where $V_w=\{v_1, v_2, v_3, v_4\}$ and $E_w=\{e_{0,1}, e_{0,2}, e_{0,3}, e_{0,4}, e_{1,2}, e_{1,3}, e_{1,4}, e_{2,3}, e_{2,4}\}$. Furthermore, four edges, $e_{0,4}$, $e_{1,3}$, $e_{2,3}$, and $e_{2,4}$, corresponding to the four virtual walls, $W_{0,4}$, $W_{1,3}$, $W_{2,3}$, and $W_{2,4}$, can be constructed in a minimum-weight spanning tree with the five vertices as illustrated in Fig. 8 and the total weight on the four virtual walls, $W_{0,4}$, $W_{1,3}$, $W_{2,3}$ and $W_{2,4}$, can be obtained as 0.

2) *Transformation of Bus connections*: In order to guarantee that no bus connection crosses any virtual wall in [15], if one bus connection crosses any virtual wall, the bus connection must be eliminated. After completing the assignment of the remaining bus connections in the PCB design, the eliminated bus connections can be further assigned onto the feasible routing layers. It is clear that the remaining bus connections can be fully included inside a closed region surrounded by the PCB boundary, the component boundaries and the virtual walls after some bus connections crossing the virtual walls are eliminated. In general, the boundary of a closed region can be treated as a circular boundary.

Based on the crossing relation between two bus connections inside a closed region, a crossing graph, $G_c(V_c, E_c)$, for the bus connections inside a closed region can be constructed in [15] as follows: a vertex, v_i , $1 \leq i \leq p$ in V_c represents a bus connection N_i inside a closed region and an edge, $e_{i,j}$, $0 \leq i, j \leq p$, $i \neq j$ in E_c represents the crossing relation between two bus connections, N_i and N_j , inside a closed region. It is known that the assignment problem for a set of bus connections inside a closed region is equal to the coloring problem [17] in the crossing graph. If the bus connections inside a closed region are exactly transformed into the bus connections on the boundary outside a closed region, the bus connections on the boundary can be further represented as a set of interval and the crossing graph for the bus connections on the boundary can

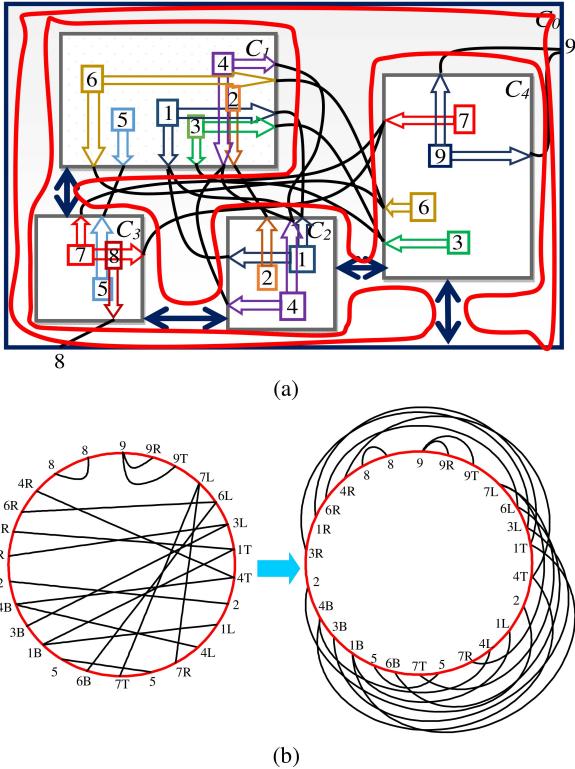


Fig. 9. Transformation of 17 possible bus connections inside a closed region. (a) Integration of boundary pins on a PCB boundary and boundary pins on four components. (b) Transformation of 17 possible bus connections inside a closed region into 17 possible bus connections outside a closed region.

be further treated as an interval graph. Hence, the assignment problem for the bus connections inside a closed region can be solved in polynomial time. Based on the transformation of two crossing or noncrossing bus connections inside a closed region in [15], all the bus connections inside the closed region can be transformed into the connections on the boundary outside the closed region. As a result, the crossing relations on all the bus connections inside the closed region can be exactly maintained on the boundary outside the closed region.

Refer to the 17 possible bus connections, \$N_{1RT}\$, \$N_{1BT}\$, \$N_{1BL}\$, \$N_2\$, \$N_{3RL}\$, \$N_{3BL}\$, \$N_{4RT}\$, \$N_{4BT}\$, \$N_5\$, \$N_{6RL}\$, \$N_{6BL}\$, \$N_{7TL}\$, \$N_{7RL}\$, \$N_8\$, \$N_{9T}\$ and \$N_{9R}\$ on the four components, \$C_1\$, \$C_2\$, \$C_3\$ and \$C_4\$, and the four constructed virtual walls, \$W_{0,4}\$, \$W_{1,3}\$, \$W_{2,3}\$ and \$W_{2,4}\$, in a PCB design, \$C_0\$, in Fig. 8, it is clear that no bus connection crosses any virtual wall. Hence, no bus connection must be eliminated and the closed boundary can be obtained from the PCB boundary, the four component boundaries and the four constructed virtual walls. Furthermore, the boundary pins on the PCB boundary and the boundary pins on the four components can be integrated into a set of pins on the boundary of a closed region as illustrated in Fig. 9(a). Based on the feasible transformation of two crossing or noncrossing bus connections inside a closed region, the 17 possible bus connections inside the closed region can be transformed into the connection of the 17 possible bus connections on the boundary outside the closed region as illustrated in Fig. 9(b).

3) Iterative Layer Assignment of Bus Connections: For the possible bus connections on a circular boundary, each possible

bus connection on a circular boundary can be represented as a *circular curve* with a fixed radian between two connecting pins. By assigning the location with the least crossings between one pair of two adjacent pins on the circular boundary as a splitting point, the circular boundary can be split to form a linear boundary and any possible bus connection covering the splitting point can be eliminated. As a result, the remaining possible bus connections on the circular boundary can be treated as the remaining possible bus connections on a linear boundary. For the possible bus connections on a linear boundary, each possible bus connection can be further represented as a *horizontal interval* between two connecting pins on a linear boundary, that is, any possible bus connection \$N_i\$ on a linear boundary can be represented as a horizontal interval \$I_i\$ on a linear boundary. Hence, the assignment of the possible bus connections in a PCB design on the available layers under some assignment constraints can be treated as the assignment of the represented intervals on the available tracks with some corresponding assignment constraints.

If each possible bus connection is represented as a horizontal interval on a linear boundary, the number of assigned tracks can be minimized for the represented intervals without covering compatibility by using the left-edge algorithm [18]. In the assignment of the possible bus connections in a PCB design, if there is no crossing relation between two possible bus connections, the two bus connections can be assigned onto the same layer. In the interval representation of the possible bus connections on a linear boundary, it is known that two represented intervals with a covering relation mean that there is no crossing relation between the two corresponding possible bus connections. Hence, two represented intervals with a covering relation can be assigned onto the same track and the covering compatibility of two represented intervals must be further considered in the track assignment of the represented intervals. Based on the covering relation between two represented intervals, a directed covering graph, \$G_t(V_t, E_t)\$, for the represented intervals can be constructed as follows: a vertex, \$v_i\$, in \$V_t\$ represents a represented interval, \$I_i\$, and a directed edge, \$e_{i,j}\$, in \$E_t\$ represents a covering relation from the represented interval, \$I_i\$, to the represented interval, \$I_j\$.

Refer to the 17 possible bus connections on the circular boundary in Fig. 9(b), by assigning the location between the two pins, 2 and 3R, as a splitting point, the 17 possible bus connections on the circular boundary can be treated as the 17 possible bus connections on a linear boundary as illustrated in Fig. 10(a). Based on the track assignment of the possible 17 represented intervals on ten used tracks using the left-edge algorithm [18] in Fig. 10(b), a directed covering graph, \$G_t(V_t, E_t)\$, for the 17 represented intervals can be constructed as illustrated in Fig. 10(c), where \$V_t = \{v_{1RT}, v_{1BT}, v_{1BL}, v_2, v_{3RL}, v_{3BL}, v_{4RT}, v_{4BT}, v_{4BL}, v_5, v_{6RL}, v_{6BL}, v_{7TL}, v_{7RL}, v_8, v_{9T}, v_{9R}\}\$ and \$E_t = \{e_{1RT,6RL}, e_{1BT,5}, e_{1BL,5}, e_{2,1BL}, e_{2,4BL}, e_{3RL,6RL}, e_{3BL,1BT}, e_{3BL,1BL}, e_{4RT,8}, e_{4RT,9T}, e_{4RT,9R}, e_{4BT,1BL}, e_{4BL,5}, e_{6RL,8}, e_{6RL,9T}, e_{6RL,9R}\}\$.

In the track assignment of the represented intervals with covering compatibility, an iterative modified left-edge algorithm in [15] can be proposed to minimize the number of used tracks and assign all the represented intervals onto the used

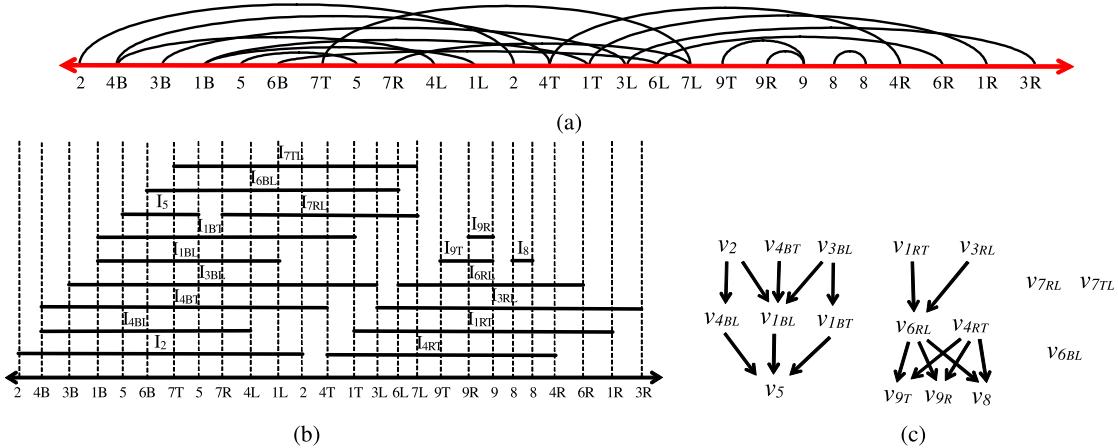


Fig. 10. Construction of a covering graph for 17 represented intervals. (a) Transformation of 17 possible bus connections on a circular boundary. (b) Track assignment of 17 represented intervals. (c) Construction of a directed covering graph for 17 represented intervals.

tracks. Basically, the useful space of each track can be treated as a *track interval*. Based on the track assignment of the represented intervals in the left-edge algorithm, a covering graph for the represented intervals can be first constructed. In each iteration, the vertices with in-degree 0 in the covering graph can be selected and a modified left-edge algorithm is used to assign the selected represented intervals onto a set of the available track intervals or the assigned represented intervals with covering compatibility. In the track assignment of the represented intervals, a hierarchical covering tree with the covering relations of the represented intervals can be further constructed. After assigning the selected intervals, the selected vertices with in-degree 0 must be deleted from the covering graph. Until the covering graph is empty, the iterative assignment will stop and the final assignment of the represented intervals on the assigned tracks will be obtained. According to the assignment result of all the represented intervals on the assigned tracks, all the possible bus connections represented by the intervals can be also assigned onto the used layers.

Refer to the covering graph in Fig. 10(c), by using a modified left-edge algorithm in [15], the nine represented intervals, I_{1RT} , I_2 , I_{3RL} , I_{3BL} , I_{4RT} , I_{4BT} , I_{6BL} , I_{7TL} and I_{7RL} , can be selected in the first iteration and the nine represented intervals can be assigned on seven available track intervals with covering compatibility. Furthermore, the four represented intervals, I_{1BT} , I_{1BL} , I_{4BL} and I_{6RL} , can be selected in the second iteration and the four represented intervals can be assigned on four assigned represented intervals with covering compatibility. Finally, the four represented intervals, I_5 , I_8 , I_{9T} and I_{9R} , can be selected in the third iteration and the four represented intervals can be assigned on three assigned represented intervals with covering compatibility. As a result, the 17 represented intervals can be fully assigned onto seven minimized tracks as illustrated in Fig. 11(a) and the corresponding hierarchical covering tree of the 17 represented intervals can be constructed as illustrated in Fig. 11(b). It is clear that the 17 bus connections represented by the 17 intervals can be assigned onto 7 minimized layers.

4) Layer Assignment of Eliminated Bus Connections: After completing the assignment of the possible bus connections in

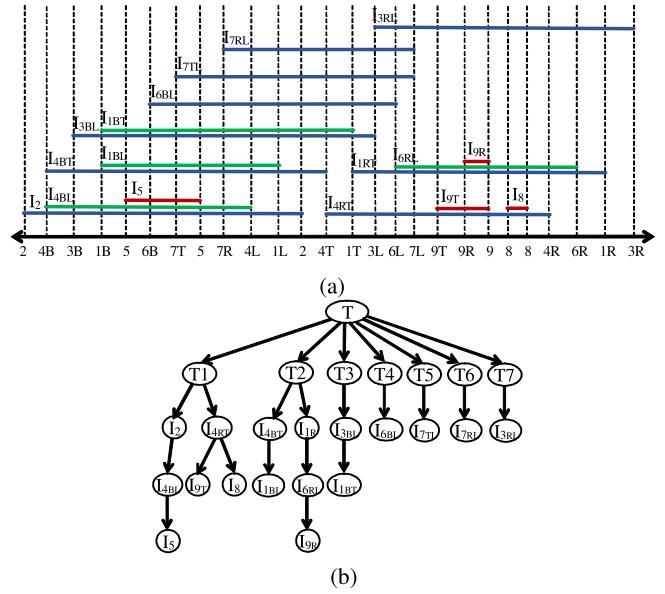


Fig. 11. Assignment of 17 represented intervals with covering compatibility in seven minimized tracks. (a) Track assignment of 17 represented intervals. (b) Construction of a hierarchical covering tree with 17 represented intervals.

a PCB design in [15], the eliminated bus connections crossing the virtual walls and covering the selected splitting point must be further assigned onto the available layers. For the layer assignment of any eliminated net, if the eliminated connection does not cross any possible bus connection inside a used layer, the eliminated net can be further assigned onto the used layer. In contrast, if the eliminated connection is not assigned onto any used layer, a new layer must be created and the eliminated connection can be further assigned onto the new layer. As a result, all the possible bus connections in a PCB design can be successfully assigned onto the used layers.

C. Iterative Elimination of Redundant Bus Connections for Layer Minimization

Basically, the physical connection of a given bus B_i , between two components, C_j and C_k , $1 \leq i \leq n$, $1 \leq j \leq m$, $k \neq j$ can

be defined as the union of the two escape connections using two necessary escape directions, $D0(B_i)$ and $D1(B_i)$, $D0(B_i) \in FD_{i,0}$, $D1(B_i) \in FD_{i,1}$ inside the components, C_j and C_k , and one necessary bus connection, N_i , between the components, C_j and C_k , for the bus, B_i . Based on the definition of the flexible escape directions inside components and the construction of the possible bus connections outside components, it is known that there are at most two possible escape directions inside one component and there are at most four possible bus connections between two components for a given bus. In the construction of the physical connection for a given bus, it is desired that only one necessary escape direction inside one component and only one necessary bus connection between two components can be used for any given bus. Hence, some redundant escape directions inside one component and some redundant bus connections between two components for some buses must be eliminated in the construction of the physical connections of the given buses in a PCB design.

To minimize the number of used layers in a PCB design, an iterative elimination algorithm can be proposed to minimize the maximum value of the two upper bounds of the layer numbers inside and outside components by eliminating the redundant escape directions inside components and the redundant bus connections outside components. Initially, if only one possible escape direction is used for a given bus inside one component, the possible escape direction can be set as one *necessary escape direction* for the bus under the component. If only one possible bus connection is used for a given bus between two components, the bus connection can be set as one *necessary bus connection* for the bus between two components. In addition, the upper bound of the layer number, U_{in} , for the flexible escape directions inside components can be obtained from the conflict graphs under the components in a PCB design and the upper bound of the layer number U_{out} for the possible bus connections outside components can be obtained from the track assignment of the intervals representing the possible bus connections between the components in a PCB design. Based on the determination of the two values, U_{in} and U_{out} , one elimination process can be used to eliminate some redundant escape directions inside components or some bus connections outside components in each iteration. Until there is no redundant escape direction inside components and no redundant bus connection outside components for the given buses, the iterative algorithm will stop.

In each iteration, the elimination process for layer minimization can be divided into two different conditions: 1) $U_{out} > U_{in}$ and 2) $U_{out} \leq U_{in}$. In the first condition, if all the represented intervals on one used track in the track assignment are deleted to reduce the upper bound, U_{out} , the possible bus connections represented by the deleted intervals can be eliminated. On the other hand, if all the intervals on one used track in the track assignment are not deleted to reduce the upper bound, U_{out} , the possible bus connection with the most crossings on the necessary bus connections can be eliminated. After completing the eliminating process of some possible bus connections, the possible bus connection can be set as one necessary bus connection if one possible bus connection is the only one possible bus connection for a given

bus. Additionally, the unused escape direction under the component must be eliminated if one escape direction inside one component is not used for any possible bus connection. Based on the elimination of the unused escape directions and the possible bus connections, the two upper bounds, U_{in} and U_{out} , in a PCB design must be further modified.

In the second condition, if the elimination of one escape direction on a given bus under one component is used to reduce the upper bound, U_{in} , the escape direction on the bus can be eliminated and the other escape direction on the bus can be set as one necessary escape direction on the bus. On the other hand, if the elimination of one escape direction on a given bus under one component is not used to reduce the upper bound, U_{in} , one escape direction on a given bus with the most intersection relations can be eliminated and the other escape direction on the bus can be set as one necessary escape direction on the bus. After eliminating one escape direction, the unused bus connections using the eliminated escape direction must be also eliminated. Based on the elimination of one escape direction and the unused bus connections, the two upper bounds, U_{in} and U_{out} , in a PCB design must be further modified.

In the bus assignment considering flexible escape routing for layer minimization, based on the determination of the upper bounds of the layer numbers for the flexible escape directions inside components and the possible bus connections outside components in a PCB design, the iterative algorithm, IERBCs, can eliminate the redundant escape directions inside components and the redundant bus connections outside components to minimize the number of used layers in a PCB design and be described in Algorithm 1.

Refer to the four conflict graphs on the four components for the 24 flexible escape directions of the nine given buses in Fig. 6(b) and the track assignment of the 17 intervals for the 17 possible bus connections in Fig. 11(a), the two upper bounds, U_{in} and U_{out} , can be initially obtained as 4 and 7, respectively.

Due to $FD_{2,0} = \{D_B\}$, $FD_{2,1} = \{D_T\}$, $FD_{3,1} = \{D_L\}$, $FD_{5,0} = \{D_T\}$, $FD_{5,1} = \{D_B\}$, $FD_{6,1} = \{D_L\}$, $FD_{7,1} = \{D_L\}$, and $FD_{8,1} = \{D_B\}$, the eight flexible escape directions can be eight necessary escape directions on six buses, B_2 , B_3 , B_5 , B_6 , B_7 and B_8 , under four components and the three possible bus connections represented by the intervals, I_2 , I_5 and I_8 , in the track assignment can be three necessary bus connections. Hence, the eight corresponding vertices in the four conflict graphs can be marked by setting their red circles and the three intervals I_2 , I_5 , and I_8 , representing the possible bus connections of the three buses, B_2 , B_5 , and B_8 , in the track assignment can be marked by setting the red color onto the three represented intervals as illustrated in Fig. 12(a).

In the iteration elimination of the redundant bus connections for layer minimization, initially, it is known that $U_{in} = 4$ and $U_{out} = 7$. In the first iteration, the possible bus connection represented by the interval, I_{3RL} , can be eliminated to reduce the upper bound, U_{out} , from 7 to 6 as illustrated in Fig. 12(b). Hence, the unused escape direction, D_R , on the bus, B_3 , under the component, C_1 , can be eliminated to reduce the upper bound, U_{in} , from 4 to 3 and the escape direction, D_B on the bus, B_3 , under the component, C_1 can be set as the

Algorithm 1 IERBC

Input: B – a set of n buses in a PCB design;
 C – a set of m components in a PCB design;

Output: Bus connections onto a minimal set of layers for the buses in B ;

$(G_1, G_2, \dots, G_m) \leqslant$ Construct m conflict graphs considering flexible escape directions on m components in C ;

$U_{in} =$ Determinate the upper bound of the layer number for the flexible escape directions inside components in B ;

$N_b =$ Construct the possible bus connections considering flexible escape directions on m components for buses in B ;

$T_b \leqslant$ Construct a hierarchical covering tree for the track assignment of the intervals representing the possible bus connections in N_b ;

$U_{out} =$ Determinate the upper bound of the layer number for the possible bus connections in N_b outside components;

$NN_b \leqslant$ Find an initial set of necessary bus connections in B ;

while there is any redundant bus connection in N_b **do**

if $U_{out} > U_{in}$ **then**

$IS \leqslant$ Find one track with all the unnecessary intervals on the same track in T_b ;

if $IS \neq \emptyset$ **then**

$T_b = T_b - IS$;

$U_{out} = U_{out} - 1$;

else

$I_{mc} \leqslant$ Find the unnecessary bus connection with the most crossings on NN_b ;

$T_b = T_b - \{I_{mc}\}$;

end if

else

$d_{in} \leqslant$ Find one unnecessary escape direction for the reduction of the upper bound, U_{in} ;

if $d_{in} \neq \emptyset$ **then**

 Eliminate the escape direction, d_{in} , from (G_1, G_2, \dots, G_m) ;

$U_{in} = U_{in} - 1$;

else

$d_{mc} \leqslant$ Find the unnecessary escape direction with the most intersection relations on G_1, G_2, \dots or G_m ;

 Eliminate the escape direction, d_{mc} , from (G_1, G_2, \dots, G_m) ;

end if

end if

 Update N_b and NN_b ;

 Modify (G_1, G_2, \dots, G_m) and T_b ;

end while

return Bus connections onto a minimal set of layers for the buses in B ;

necessary escape direction on the bus, B_3 , under the component C_1 . Furthermore, the possible bus connection represented by the interval, I_{3BL} , will be set as the necessary bus connection on the bus, B_3 .

In the second iteration, it is known that $U_{in} = 3$ and $U_{out} = 6$. In this condition, the possible bus connections represented by the interval, I_{7RL} , can be eliminated to reduce the upper bound, U_{out} , from 6 to 5 as illustrated in Fig. 12(c). Hence, the unused escape direction, D_R , on the bus, B_7 , under the component, C_3 , can be eliminated and the escape direction, D_T , on the bus, B_7 , under the component, C_3 , can be set as the necessary escape direction on the bus, B_7 , under the component, C_3 . Furthermore, the possible bus connection represented by the interval, I_{7TL} , will be set as the necessary bus connection on the bus, B_7 .

In the third iteration, it is known that $U_{in} = 3$ and $U_{out} = 5$. In this condition, the possible bus connection represented by the interval, I_{6BL} , can be eliminated to reduce the upper bound, U_{out} , from 5 to 4 as illustrated in Fig. 12(d). Hence, the unused escape direction, D_B , on the bus, B_6 , under the component, C_1 , can be eliminated and the escape direction, D_R , on the bus, B_6 , under the component C_1 , can be set as the necessary escape direction on the bus, B_6 . Furthermore, the possible bus connection represented by the interval, I_{6RL} , will be set as the necessary bus connection on the bus, B_6 .

In the fourth iteration, it is known that $U_{in} = 3$ and $U_{out} = 4$. In this condition, the three possible bus connections represented by the three intervals, I_{1BL}, I_{4BT} and I_{1RL} , can be eliminated to reduce the upper bound U_{out} from 4 to 3 as illustrated in Fig. 12(e). Hence, the unused escape direction, D_R , on the bus, B_1 , under the component, C_1 , and the unused escape direction, D_L , on the bus, B_1 , under the component, C_2 , can be eliminated, the escape direction, D_B , on the bus, B_1 , under the component, C_1 , and the escape direction, D_T , on the bus, B_1 , under the component, C_2 , can be set as the necessary escape direction on the bus, B_1 . Furthermore, the possible bus connection represented by the interval, I_{1BT} , will be set as the necessary bus connection on the bus, B_1 .

In the fifth iteration, it is known that $U_{in} = 3$ and $U_{out} = 3$. In this condition, the escape direction, D_B , on the bus, B_4 , under the component, C_1 , can be eliminated to reduce the upper bound, U_{in} , from 3 to 2 as illustrated in Fig. 12(f). Hence, the unused escape direction, D_L , on the bus, B_4 , under the component, C_2 , and the possible bus connection represented by the interval, I_{4BL} , can be eliminated, the escape direction, D_R , on the bus, B_4 , under the component, C_1 , can be set as the necessary escape direction on the bus, B_4 , under the component, C_1 , and the escape direction, D_T , on the bus, B_4 , under the component, C_2 , can be set as the necessary escape direction on the bus, B_4 , under the component, C_2 . Furthermore, the possible bus connection represented by the interval, I_{4RT} , will be set as the necessary bus connection on the bus, B_4 .

In the sixth iteration, it is known that $U_{in} = 2$ and $U_{out} = 3$. In this condition, the possible bus connection represented by the interval, I_{9T} , can be eliminated as illustrated in Fig. 12(g). Hence, the unused escape direction, D_T , on the bus, B_9 , under the component, C_4 , can be eliminated and the escape direction, D_R , on the bus, B_9 , under the component, C_4 , can be set as the necessary escape direction on the bus, B_9 , under the component, C_4 . Furthermore, the possible bus connection represented by the interval, I_{9R} , will be set as the necessary bus connection on the bus, B_9 .

After eliminating the redundant escape directions and the redundant bus connections on the six given buses B_1, B_3, B_4, B_6, B_7 and B_9 , the upper bound, U_{in} , of the layer number inside components can be obtained as 2 and the upper bound, U_{out} , of the layer number outside components can be obtained as 3.

D. Bus Assignment for Layer Minimization

After eliminating the redundant bus connections for the given buses, the final conflict graphs inside components and

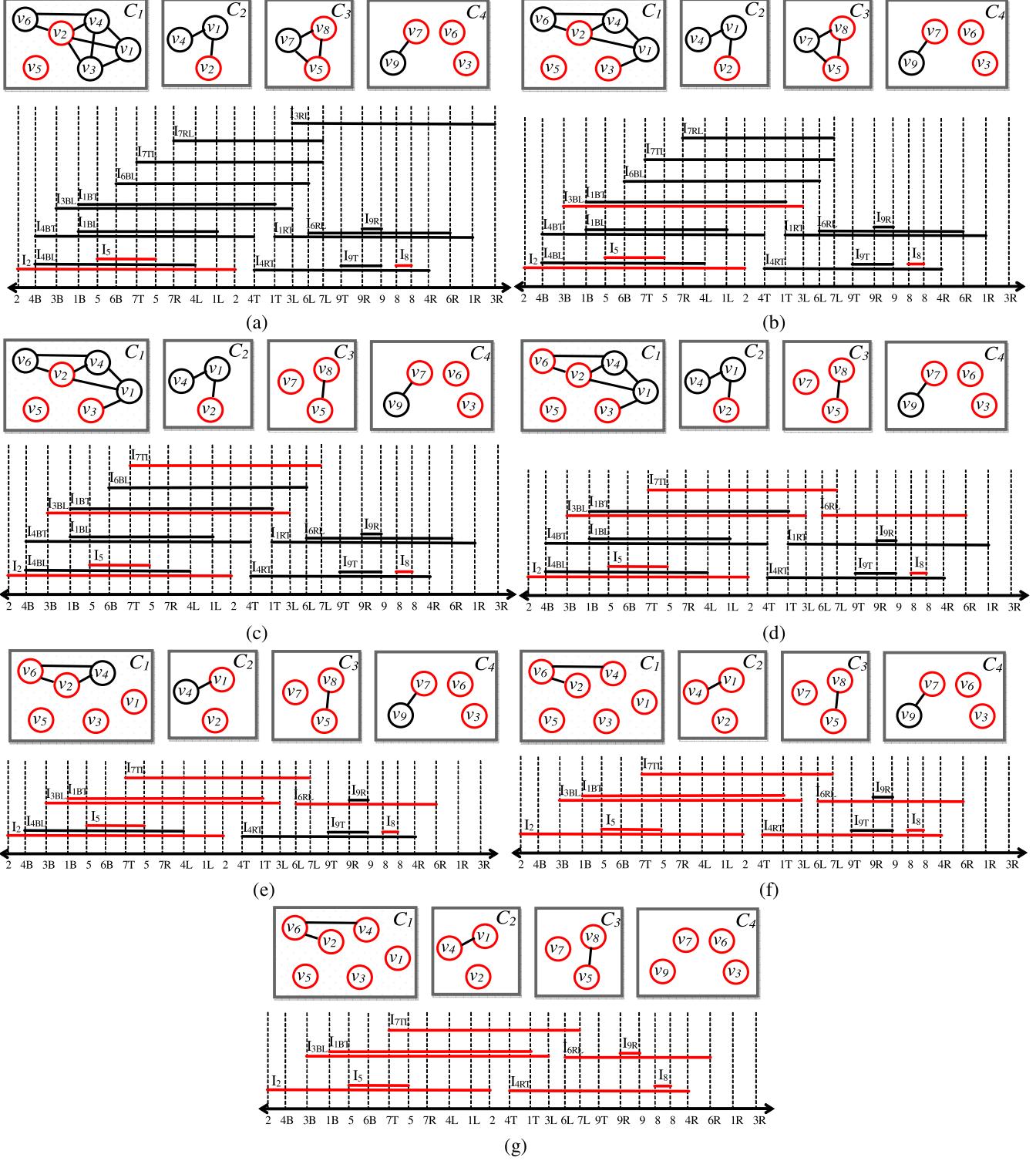


Fig. 12. IERBC for nine given buses. (a) Initialization of four conflict graphs and track assignment of 17 intervals on seven used tracks with covering compatibility. (b) Elimination of one redundant bus connection represent by one interval I_{3RL} . (c) Elimination of one redundant bus connection represent by one interval I_{6RL} . (d) Elimination of one redundant bus connection represent by one interval I_{6RL} . (e) Elimination of three redundant bus connections represent by three intervals I_{1RT}, I_{1BL} , and I_{4BT} . (f) Elimination of one redundant escape direction D_B . (g) Elimination of one redundant bus connection represent by one interval I_{9T} .

the track assignment of the intervals representing the final bus connections outside components in a PCB design can be obtained. In the integration of the BALM, the intersection relations inside components can be further treated as the constraints on the bus assignment outside components in

a PCB design. Based on the track assignment of the intervals representing the final bus connections outside components, a new covering graph for the final represented intervals can be constructed. Furthermore, the iterative modified left-edge algorithm in [15] on a set of assignment constraints can be

Algorithm 2 BALM

Input: B – a set of n buses in a PCB design;
 AC – a set of assignment constraints inside components;
Output: A set of physical connections for the given buses in B in a minimal set of layers;
Sort the intervals representing the final bus connections in B ;
 $G \leqslant$ Construct a covering graph for the intervals representing the final bus connections in B ;
while $G \neq \emptyset$ **do**
 $V_0 \leqslant$ Select the vertices with in-degree 0 in G ;
 Run the modified left-edge algorithm [15] under the assignment constraints in AC for the intervals represented by vertices in V_0 ;
 $G = G - V_0$;
end while
return A set of physical connections for the given buses in B in a minimal set of layers;

proposed to minimize the number of used tracks and assign all the represented intervals onto the used tracks with satisfying the assignment constraints. Hence, the physical connections of the given buses including the necessary escape directions inside components and the necessary bus connections outside components can be assigned onto a minimal set of used layers.

In the BALM, based on the assignment constraints from the intersection relations in the conflict graphs, the algorithm, BALM, can assign the physical connections of the given buses onto a minimal set of used layers in a PCB design and be described in Algorithm 2.

Refer to the four conflict graphs inside four components and the track assignment of the 9 represented intervals with covering compatibility in Fig. 12(g), two intersection relations under the component C_1 , one intersection relation under the component C_2 , and one intersection relation under component C_3 can be treated as the four assignment constraints on the assignment of the nine given buses outside components as illustrated in Fig. 13(a).

In the assignment of the nine given buses for layer minimization, first, the nine intervals representing nine bus connections on three used tracks can be sorted in an increasing order according to the location of the left end point inside one interval. Furthermore, a new covering graph for the nine intervals can be constructed as illustrated in Fig. 13(b). Based on the four assignment constraints in Fig. 13(a), the nine sorted intervals with covering compatibility can be assigned onto three used tracks by using the iterative modified left-edge algorithm in [15] as illustrated in Fig. 13(c). As a result, the nine given buses in a PCB design can be assigned onto three used layers. It is clear that the three given buses B_2 , B_4 , and B_5 can be assigned onto the first layer, the five given buses B_1 , B_3 , B_6 , B_8 , and B_9 can be assigned onto the second layer and the given bus B_7 can be assigned onto the third layer, respectively.

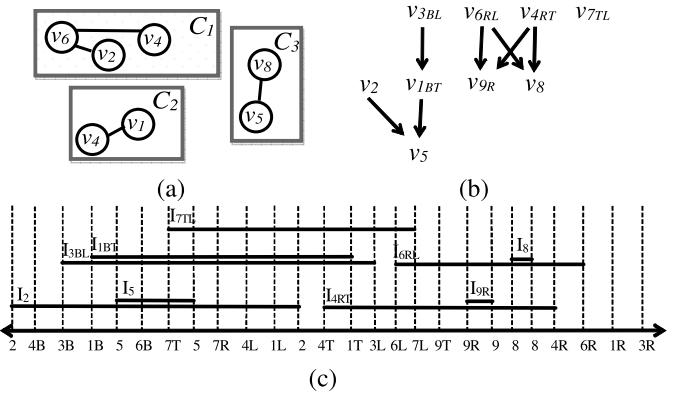


Fig. 13. Assignment of nine intervals with four assignment constraints for layer minimization. (a) Generation of four assignment constraints from four intersection relations. (b) Construction of a covering graph for nine represented intervals. (c) Track assignment of nine represented intervals under four assignment constraints.

E. Analysis of Time Complexity

Given a set of n buses on a set of m components in a PCB design, our proposed algorithm can be divided into four sequential steps: 1) determination of the upper bound of the layer number for flexible escape directions inside components; 2) determination of the upper bound of the layer number for possible bus connections outside components; 3) IERBCs for layer minimization; and 4) BALM.

In the determination of the upper bounds of the layer number for the flexible escape directions inside components, the time complexity of finding the upper bound of the layer number by using the maximum density of the partitioned subspace under one component is $O(n^2)$. Hence, the time complexity of finding the upper bound of the layer number inside components in a PCB design is $O(mn^2)$.

Refer to the analysis of the time complexity for the assignment of the bus connections in [15], given a set of n bus connections in a PCB design, the time complexity of completing the assignment of the bus connections in a PCB design is $O(n^2)$. Since the number of all the possible bus connections outside components in a PCB design is in $O(n)$, the time complexity of completing the assignment of all the possible bus connections in a minimal set of routing layers is $O(n^2)$. Hence, the time complexity of finding the upper bound of the layer number outside components is $O(n^2)$ in the determination of the upper bounds of the layer number for the possible bus connections outside components.

In the iterative elimination of the redundant bus connections for layer minimization, the time complexity of eliminating the redundant bus connections in a PCB design in each iteration is $O(mn)$. Since the number of iterations is in $O(n)$, the time complexity of eliminating all the redundant bus connections in a PCB design is $O(mn^2)$.

In the BALM, the time complexity of sorting the intervals representing the necessary bus connections is $O(n\log n)$ and the time complexity of constructing a covering graph is $O(n^2)$. Furthermore, the time complexity of running an iterative modified left-edge algorithm under a set of assignment constraints is $O(n)$. Hence, the time complexity of assigning the physical

TABLE I
EXPERIMENTAL RESULTS FOR BUS ASSIGNMENT CONSIDERING FLEXIBLE ESCAPE ROUTING IN PCB DESIGNS

Examples	#Components	#Buses	Two-Phase Algorithm				Our Integrated Algorithm		
			Yan's Routing Algorithm [9]		Yan's Assignment Algorithm [15]		#ELayer	#BLayer	Time(s)
			#ELayer	ETime(s)	#BLayer	BTime(s)			
Board01	6	27	2	0.03	4(100.0%)	0.05	2	3(75.0%)	0.07
Board02	10	62	2	0.07	5(100.0%)	0.11	2	4(80.0%)	0.21
Board03	13	79	2	0.08	6(100.0%)	0.16	3	5(83.3%)	0.31
Board04	19	92	3	0.11	7(100.0%)	0.21	4	5(71.4%)	0.42
Board05	21	105	3	0.15	8(100.0%)	0.30	4	6(75.0%)	0.49
Board06	32	147	3	0.21	10(100.0%)	0.43	4	7(70.0%)	0.68
Test01	35	200	3	0.31	13(100.0%)	0.63	4	10(76.9%)	0.98
Test02	40	300	4	0.41	16(100.0%)	1.12	4	12(75.0%)	1.65
Test03	45	400	3	0.63	17(100.0%)	1.45	4	12(70.6%)	2.17
Test04	50	500	4	0.78	20(100.0%)	1.95	5	14(70.0%)	3.04

connections of the given buses onto a minimal set of routing layers is in $O(n^2)$.

As a result, the time complexity of completing the bus assignment considering flexible escape routing in a PCB design is $O(mn^2)$, where m is the number of components and n is the number of given buses in a PCB design.

IV. EXPERIMENTAL RESULTS

In the bus assignment considering flexible escape routing for layer minimization in a PCB design, our proposed integrated algorithm has been implemented by using standard C++ language and run on an Intel Core i7-3770 CPU 3.40-GHz machine with 8-GB memory.

Six tested examples: 1) Board01; 2) Board02; 3) Board03; 4) Board04; 5) Board05; and 6) Board06 are obtained from the real PCB designs in industry and used in [15]. Additionally, four larger tested examples: 1) Test01; 2) Test02; 3) Test03; and 4) Test04, from 200 buses on 35 components to 500 buses on 50 components are randomly generated in [16] and globally routed by using the Allegro PCB router [19]. In Table I, *#Component* denotes the number of components in a tested example, *#Bus* denotes the number of given buses in a tested example, *#ELayer* denotes the number of the used layers in escape routing for the given buses under the components in a tested example, *#BLayer* denotes the number of used layers in the bus assignment for the given buses outside the components in a tested example, *ETime* denotes the execution time in escape routing for the given buses under the components in a tested example, *BTime* denotes the execution time in bus assignment for the given buses outside the components in a tested example, and *Time* denotes the execution time for the bus assignment considering flexible escape routing in a tested example.

To compare the number of used layers in the bus assignment in a PCB design with the published works, Yan's algorithm [15] in the bus assignment in a PCB design after implementing Yan's algorithm [9] in direction-constrained rectangle escape routing is further implemented in the experiment. In the combination of Yan's routing algorithm [9] and Yan's assignment algorithm [15] in a PCB design, given a set of buses on a set of components in a PCB design, the pins of the given buses inside any component can be first escaped

to the available boundaries to minimize the number of used layers by using Yan's routing algorithm [9] inside one component. Furthermore, based on the escape routing results of all the given buses inside all the components, all the bus connections can be assigned onto a minimal set of routing layers by using Yan's assignment algorithm [15] outside components in a PCB design. As a result, the experimental results of the bus assignment for ten tested examples can be also listed in Table I.

Compared with the two-phase algorithm using Yan's routing algorithm [9] in direction-constrained rectangle escape routing and Yan's assignment algorithm [15] in bus assignment, the experimental results show that our proposed integrated algorithm uses reasonable CPU time to reduce 32.1% of the layer number for ten tested examples on the average, respectively. From the viewpoint of the fabrication cost in the experimental results, the integration of the escape routing process inside components and the bus assignment process outside components in a PCB design can leads to the reduction on the number of used layers in a PCB design.

V. CONCLUSION

Traditionally, the bus assignment outside components is based on the locations of the bus pins in escape routing inside components. Based on the concept of introducing the flexible escape directions inside components in escape routing, an integrated algorithm can be proposed to minimize the number of used layers by eliminating the redundant bus connections and assign all the physical connections of the given buses onto the available layers. Experimental results showed that our proposed integrated algorithm is efficient on the number reduction of the used layers in a PCB design.

REFERENCES

- [1] C. F. Coombs, Ed., *Printed Circuits Handbook*, 6th ed. New York, NY, USA: McGraw-Hill, 2007.
- [2] M. M. Ozdal and M. D. F. Wong, "A length-matching routing algorithm for high-performance printed circuit boards," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 12, pp. 2784–2794, Dec. 2006.
- [3] H. Kong, Q. Ma, T. Yan, and M. D. F. Wong, "An optimal algorithm for finding disjoint rectangles and its application to PCB routing," in *Proc. Design Autom. Conf.*, 2010, pp. 212–217.

- [4] Q. Ma, H. Kong, M. D. F. Wong, and E. F. Y. Young, "A provably good approximation algorithm for rectangle escape problem with application to PCB routing," in *Proc. Asia South-Pac. Design Autom. Conf.*, Yokohama, Japan, 2011, pp. 843–848.
- [5] Q. Ma and M. D. F. Wong, "NP-completeness and an approximation algorithm for rectangle escape problem with application to PCB routing," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 31, no. 9, pp. 1356–1365, Sep. 2012.
- [6] J.-T. Yan, J.-M. Chung, and Z.-W. Chen, "Density-reduction-oriented layer assignment for rectangle escape routing," in *Proc. ACM Great Lakes Symp. VLSI*, 2012, pp. 275–278.
- [7] J.-T. Yan and Z.-W. Chen, "Direction-constrained layer assignment for rectangle escape routing," in *Proc. Int. SOC Conf.*, Niagara Falls, NY, USA, 2012, pp. 254–259.
- [8] A. Ahmadinejad and H. Zarabi-Zadeh, "Finding maximum disjoint set of boundary rectangles with application to PCB routing," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 36, no. 3, pp. 412–420, Mar. 2017.
- [9] J.-T. Yan, "Direction-constrained rectangle escape routing," *ACM Trans. Design Autom. Electron. Syst.*, vol. 23, no. 3, p. 34, 2018.
- [10] H. Kong, T. Yan, and M. D. F. Wong, "Automatic bus planner for dense PCBs," in *Proc. Design Autom. Conf.*, San Francisco, CA, USA, 2009, pp. 326–331.
- [11] P.-C. Wu, Q. Ma, and M. D. F. Wong, "An ILP-based automatic bus planner for dense PCBs," in *Proc. Asia South-Pac. Design Autom. Conf.*, Yokohama, Japan, 2011, pp. 181–186.
- [12] T.-Y. Tsai, R.-J. Lee, C.-Y. Chin, C.-Y. Kuan, H.-M. Chen, and Y. Kajitani, "On routing fixed escaped boundary pins for high speed boards," in *Proc. Design Autom. Test Eur. Conf.*, Grenoble, France, 2011, pp. 461–466.
- [13] C.-Y. Chin, C.-Y. Kuan, T.-Y. Tsai, H.-M. Chen, and Y. Kajitani, "Escaped boundary pins routing for high-speed boards," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 3, pp. 381–391, Mar. 2013.
- [14] K. J. Supowit, "Finding a maximum planar subset of a set of nets in a channel," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 6, no. 1, pp. 93–94, Jan. 1987.
- [15] J.-T. Yan, "Efficient layer assignment of bus-oriented nets in high-speed PCB designs," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 35, no. 8, pp. 1332–1344, Aug. 2016.
- [16] J.-T. Yan, "Layer assignment of buses and nets with via-count constraint in high-speed PCB designs," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 38, no. 3, pp. 512–525, Mar. 2019.
- [17] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. Cambridge, MA, USA: MIT Press, 2009.
- [18] A. Hashimoto and J. Stevens, "Wire routing by optimizing channel assignment within large apertures," in *Proc. 8th Design Autom. Workshop*, 1971, pp. 155–169.
- [19] *Allegro SPB v17.2*, Cadence, San Jose, CA, USA, 2016. [Online]. Available: <https://www.cadence.com>



Jin-Tai Yan (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer and information science from National Ciao-Tung University, Hsinchu, Taiwan, in 1988, 1990, and 1995, respectively.

From 1999 to 2019, he was a Faculty Member with the Department of Computer Science and Information Engineering, Chung-Hua University, Hsinchu. He is currently a Professor Researcher with the Office of Research and Development, Tainan National University of the Arts, Tainan, Taiwan. His current research interests include VLSI design automation, PCB designs, synthesis of quantum circuits, digital IC design, computer architecture, and combinational optimization.