IEICE TRANS. FUNDAMENTALS, VOL.E92–A, NO.3 MARCH 2009

PAPER

Minimum Shield Insertion on Full-Chip RLC Crosstalk Budgeting Routing*

Peng-Yang HUNG[†], **Ying-Shu LOU**[††], ***Nonmembers***, ***and*** **Yih-Lang LI**[†††a)], ***Member***

SUMMARY This work presents a full-chip RLC crosstalk budgeting routing flow to generate a high-quality routing design under stringent crosstalk constraints. Based on the cost function addressing the sensitive nets in visited global cells for each net, global routing can lower routing congestion as well as coupling effect. Crosstalk-driven track routing minimizes capacitive coupling effects and decreases inductive coupling effects by avoiding placing sensitive nets on adjacent tracks. To achieve inductive crosstalk budgeting optimization, the shield insertion problem can be solved with a minimum column covering algorithm which is undertaken following track routing to process nets with an excess of inductive crosstalk. The proposed routing flow method can identify the required number of shields more accurately, and process more complex routing problems than the linear programming (LP) methods. Results of this study demonstrate that the proposed approach can effectively and quickly lower inductive crosstalk by up to one-third.

key words: *track routing, shield insertion, detailed routing, crosstalk optimization, global routing, VLSI layout optimization*

1. Introduction

Modern integrated circuit design and manufacturing technologies continue to grow towards rising clock frequencies and declining feature size. A side effect of this trend is that interconnections tend to be adversely affected by delays, since interconnection resistance is inversely proportional to wire width and height. Hence, wires are frequently designed with high height to width aspect ratios to alleviate this side effect. However, a high aspect ratio of a wire (tall wire) raises the plate area of its coupling capacitance with its adjacent wires, and produces capacitive crosstalk. As well as short-range capacitive coupling effects, long-range inductive coupling effects also become essential to successful high-speed circuit design. Therefore, the crosstalk budgeting interconnection optimization problem, as one vital source of signal integrity, needs to be addressed in high-performance VLSI design [1].

Net ordering algorithms have been demonstrated to separate sensitive nets to reduce crosstalk effect [2]–[4]. Shield utilization rate has been discussed in previous publications [5], [6], describing the simultaneous shield insertion and net ordering approach for lowering capacitive and inductive crosstalk. Xiong et al. first explored full-chip routing optimization with RLC crosstalk budgeting [7]. A three-phase method following a global router solves the full-chip routing optimization problem with RLC crosstalk budgeting for every sink. First, the crosstalk bound at every sink is distributed to every visited global region using linear-programming (LP) based scheme; second, a simulated annealing based shield insertion and net ordering (SINO) algorithm inserts shields and reorders the net segments in a global region to reduce the coupling effect, and finally, a local refinement algorithm removes previously inserted shields without inducing crosstalk violations. A large runtime of over ten thousands seconds for a circuit with less than 1,000 nets is the limitation of Xiong's work.

Conventional global routing and detailed routing design flow cannot effectively model the crosstalk optimization problem, since global routing has no track information, significantly lowering the accuracy of crosstalk calculation. Conversely, detailed routing determines the physical dimensions and locations of interconnections, but its high computation load makes additional constraints difficult to impose on the routing model. Thus, the purification of a simple two-stage routing model targeting crosstalk-budgeted routing optimization is essential. Shabbir et al. [8] developed track routing as an intermediate process between global and detailed routings. Routing is then performed in three stages, global routing, track routing, and detail routing. Global routers identify the routing regions (global cells) adopted in the detailed routing stage for every net. Conversely, every global cell knows the nets that will pass through it in the detailed routing stage. Track routing derives the track position of every net in a global cell. Since track routing simultaneously manages a series of global cells, called a panel, the path of a net is straighter using this method than by the maze routing algorithm. Furthermore, track routing fixes the track position of every processed net, making the coupling effect estimation more accurate than in global routing Therefore, coupling minimization methods developed in the track routing stage are realistic.

This investigation addresses the minimum shield insertion problem on full-chip RLC crosstalk budgeting routing based on three-stage routing flow. In global routing, crosstalk and congestion are simultaneously considered by determining the number of sensitive nets of the routed net and the space ratio of available tracks to total tracks. Track routing considers crosstalk minimization as well as track utilization. One difference between the proposed algorithm

Manuscript received November 7, 2007.
Manuscript revised October 27, 2008.

[†]The author is with Sunplus Corporation, Taiwan.
[††]The author is with Faraday Corporation, Taiwan.
[†††]The author is with National Chiao Tung University, Taiwan.

*This work was partially supported by the National Science Council of Taiwan by Grant Nos. NSC 96-2220-E-009-014 and 96-2220-E-009-011.

a) E-mail: ylli@cs.nctu.edu.tw

DOI: 10.1587/transfun.E92.A.880

and previous works is that IRoute is only marked as processed instead of being removed from the overlap graph, which helps track routing prefer to place the IRoutes close to the high density zone so as to increase track utilization. Finally, the minimum shield insertion problem is converted into a minimum column covering problem by creating an *LSK* reduction table containing critical regions for shield insertion, *LSK* slack on every crosstalk violation path, and *LSK* reduced value after shield insertion in every region. This method predicts the number of required shields more accurately than the linear programming method, and can efficiently process large designs.

2. Preliminaries

2.1 Sensitivity and Crosstalk Evaluation

The inductive crosstalk between two wire segments becomes increasingly significant as the operating clock frequencies of integrated circuits continues to rise. Xiong et al. [7] presented a simple yet efficient inductive crosstalk estimation model, called length-scaled K_{eff} (*LSK*) model. The coupling coefficient between two wire segments n_{it} and n_{jt} can be adopted to portray their inductive crosstalk, where n_{it} denotes the wire segment of net n_i in the routing region t and its value is the track number of n_{it} in the routing region t. The coefficient is defined as Eq. (1):

$$K_{it,jt} = \frac{M_{it,jt}}{\sqrt{L_{it} \cdot L_{jt}}}, \tag{1}$$

where $M_{it,jt}$ denotes the mutual inductance between n_{it} and n_{jt}, and L_{jt} and L_{jt} denote the self-inductance for n_{it} and n_{jt} under the loop inductance model [7]. In Fig. 1(a), s_{lt} and s_{rt} are shielding signals (such as power and ground signals) in the routing region t and their values are the track numbers of s_{lt} and s_{rt} in the routing region t. A simple yet effective *Length-Scaled* K_{eff} (*LSK*) model has been proposed for the application of inductive crosstalk estimation in computation intensive tasks, including routing problems [7]. The value of $K_{it,jt}$ ranges between 0 and 1, and is derived using Eq. (2):

$$K_{it,jt} = \frac{f(i,t) + g(j,t)}{2}, \tag{2}$$

where $f(i,t) = (n_{it} - s_{lt})/(n_{jt} - s_{lt})$ and $g(j,t) = (s_{rt} - n_{jt})/(s_{rt} - n_{it})$. Since a net segment may be sensitive to several net segments, the total amount of inductive crosstalk induced

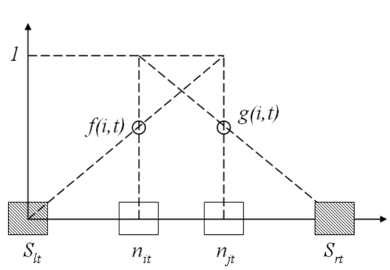


Fig. 1 Illustration of coefficient $K_{it,jt}$ in [7].

on the net segment n_{it} can be measured using Eq. (3):

$$K_{it} = \sum_{j \neq i} S_{ij} \times K_{it,jt}, \tag{3}$$

where $S_{ij} = 1$ (0) if net segment n_{jt} is (not) sensitive to net segment n_{it}. Since K_{it} is designed for fixed-length wire segments, the *LSK* value of net n_i at its jth sink is defined using Eq. (4):

$$LSK_{ij} = \sum_{t \in H_{ij}} l_t \times K_{it}, \tag{4}$$

where H_{ij} is the union of those passed routing regions by the route of the jth sink of net n_i, and l_t is the length of routing region t.

2.2 Noise-Bound Model

In this study, the crosstalk optimization problem is explored to create a capacitive crosstalk-free and bounded inductive crosstalk track routing result. Capacitive crosstalk induced by two sensitive nets is assumed to be eliminated by placing these two nets on two non-adjacent tracks. The inductive crosstalk induced by two net segments separated by a shielding wire is assumed to be small enough to be ignored. For net segments located between two shielding wires, the amount of inductive crosstalk for every source-to-sink path of a net cannot exceed a given LSK^{max} value.

2.3 Track Routing

Global routers partition routing regions into an array of global cells. The top section of Fig. 2(a) shows a 13×5 array of global cells. A panel is referred to as a series of global cells. For instance, every horizontal/vertical panel in Fig. 2(a) comprises 13/5 continuous global cells. A net

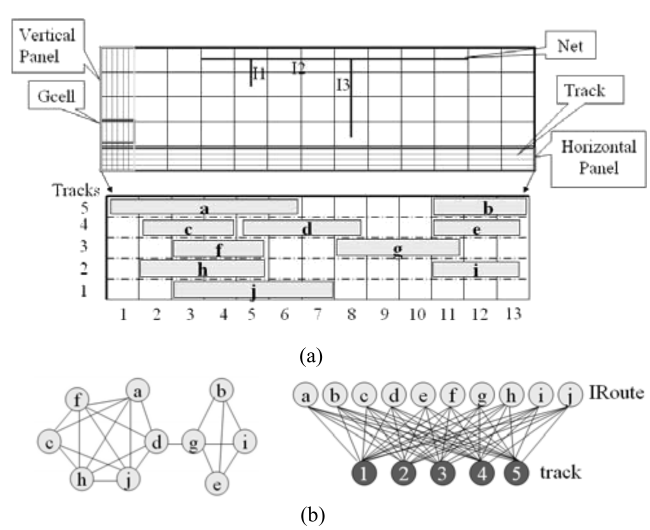


Fig. 2 (a) A routing region with 13 × 5 global cells and one horizontal panel contains 10 IRoute; (b) associated overlap graph and bipartite matching graph of the hirozontal panel in (a).

totally crossing over a global cell is named an IRoute and processed in track routing. Track routing completes the track assignment of one panel at a time [8]. The bottom figure in Fig. 2(a) displays an assignment of ten IRoutes in a five-track horizontal panel. Track routing first constructs an overlap graph. Every node of an overlap graph represents an IRoute, and an edge connects two nodes if their related IRoutes overlap. The left figure of Fig. 2(b) displays the overlap graph of the panel in Fig. 2(a). The assignability of an IRoute to every track in a panel can be modeled as a bipartite matching graph, where an edge connects an IRoute node and a track node only if this IRoute can be placed in this track, as shown in the right figure of Fig. 2(b). In Fig. 2(b), every IRoute node connects to every track node, since this panel has no pre-placed blockage. Two nets are regarded as sensitive nets for crosstalk if their IRoutes overlap, in which case placing them on adjacent tracks induces crosstalk effects. Placing two sensitive nets on non-adjacent tracks is assumed to eliminate capacitive crosstalk in this work. A sensitivity graph is constructed by connecting an edge between two nets whose coupling sensitivity is defined. The maximum clique of a sensitivity graph determines the lower bound of the required track number for producing a capacitive coupling-free track assignment.

In [9], the first proposed crosstalk-driven track routing repeatedly identifies a maximum clique, say CQ_{max}, from the overlap graph and then invokes the minimum weighted Hamiltonian path (minimum overlapping length) in CQ_{max} to place all IRoutes of the clique in the panel. All nodes in the maximum clique are removed from the overlap graph and subsequent iterations continue until all the nodes in the overlap graph are processed.

3. Flow and Algorithm

The proposed routing flow is composed of three stages: (1) crosstalk and congestion-driven global routing, (2) crosstalk-driven track routing, and (3) minimum column covering based shield insertion

3.1 Crosstalk- and Congestion-Driven Global Routing

Xiong et al. [7] demonstrated that the number of inserted shielding wires of a net in a region rises with its sensitivity rate in this region while the inductive crosstalk bound is fixed, where the sensitivity rate of a net in a region is given by the ratio of the number of its aggressors in this routing region to the total number of nets in this routing region. Thus the number of required shielding wires in a global cell for a global routing result is also proportional to the number of sensitive nets arranged to cross the global cell. Restated, seeking a global routing using the minimum number of shielding wires can be transformed into seeking a global routing with the minimum number of sensitive nets arranged in every global cell. The total wire length required to cross global cells without sensitive nets, is probably significantly greater than that with sensitive nets. To balance routing quality and number of shielding wires, the cost function of crosstalk- and congestion-driven global routing in evaluating a passing routing region is defined as:

$$Cost = \alpha \times N_s + \beta \times \frac{N_t}{N_f} + \gamma \times L + C_v + C_p, \tag{5}$$

where N_s is the number of signals that interfere with the subject signal in the region; N_t is the total number of tracks in the routing region; N_f is the number of available tracks in the region; L is the length of the routing region; C_v denotes the via cost, and C_p denotes the routing cost in the preferred direction. The first term of the equation uniformly disperses sensitive signals to all routing regions; the second term distributes congestion over all routing regions, and the last three terms evaluate routing quality.

3.2 Crosstalk-Driven Track Routing

Crosstalk-driven track routing focuses on generating a capacitive crosstalk-free and minimum inductive crosstalk track routing result. The first crosstalk-driven track routing algorithm iteratively discovers a maximum clique and assigns the IRoutes within it [9]. A clique can be regarded as a zone in routing region. The zone-based algorithm may lower track utilization and fail in the completion of track routing since the vicinal region of CQ_{max} is probably of high density too and should have a high priority to be processed. In this work, we proposed an improved zone-based track routing algorithm to better the assignment order of the IRoutes in the zones next to CQ_{max}. Traditional zone-based algorithm eliminates the nodes in CQ_{max}. The nodes next to the nodes in CQ_{max} get decreasing degree and tend to be skipped in the next iteration. In Fig. 3(a), traditional zone-based algorithm is employed to complete the track assignment problem in Fig. 2(a). The top part of Fig. 3(b) displays the process of zone identification. The first CQ_{max} contains nodes a, c, f, h, and j. The dotted edges in the overlap graph are the edges of CQ_{max} After assigning all nodes in CQ_{max}, all edges connecting these five nodes are eliminated as shown in the top right part of Fig. 3(b), where the nodes with dotted borders represent the IRoutes that have been well placed. Thus second maximum clique identification finds the clique containing nodes b, e, g, and i. Under the sensitivity constraint in Fig. 3(a), the assignment order (g, b, e, i) is one possible solution with minimum coupling capacitance and shown in Fig. 3(b). However, IRoute g occupies the resource of the remaining unassigned IRoute d. Therefore, to remedy this problem, the nodes of CQ_{max}, say NO_{max_c}, are only marked as processed rather than being eliminated from the overlap graph. If a node in NO_{max_c} only links to the other nodes in NO_{max_c}, then all the edges of this node are removed to avoid influencing further zone identification since its connecting neighbors have been processed. The adaptive zone-based algorithm gives the neighbors of NO_{max_c} high priority for subsequent processing if they are also of high degree. Figure 3(c) displays the assignment result of the proposed enhanced zone-based algorithm. After

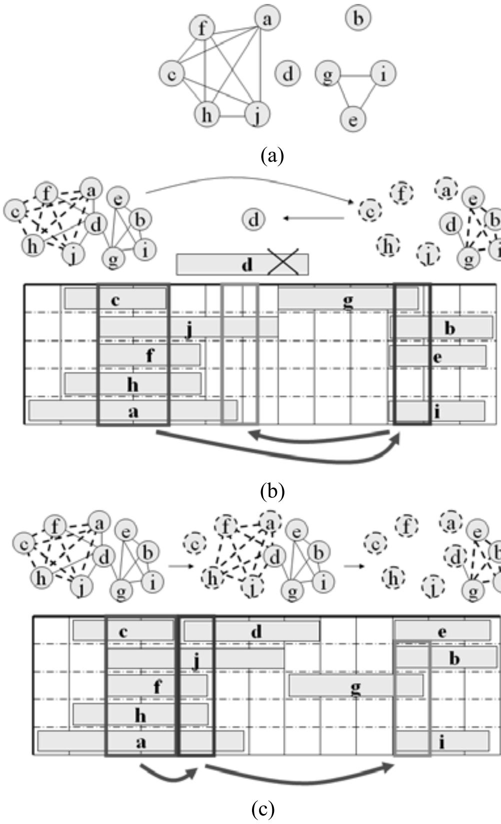


Fig. 3 (a) The sensitivity graph of IRoute in Fig. 2; (b) assignment result using traditional zone-based algorithm. IRoute d is an incomplete IRoute; (c) assignment result using the proposed enhanced zone-based algorithm. The assignment all IRoutes is complete.

the assignment of the first clique (IRoutes a, c, f, h, and j), only the edges connecting to IRoute c are removed because all the connected edges of IRoute c connects to the other IRoutes in CQ_{max}. All nodes in CQ_{max} (IRoutes a, c, f, h, and j) are marked as already placed. The second identified clique contains IRoutes a, d, f, h, and j, where only IRoute d is incomplete and can be placed in the topmost track, as shown in Fig. 3(c). Finally the last clique (IRoutes b, e, g, and i) is found and all IRoutes in the clique are successfully distributed to the panel.

In addition to the proposed zone-based algorithm, the assignment method of CQ_{max} in this work considers sensitivity graph while sensitivity graph is not used in Ho's work [9]. The sensitivity graph of a set of nets is defined in a similar way to the overlap graph. The connectivity of two IRoutes is determined by their overlap relation, and fixed by their sensitivity relation. The sensitivity graph of all IRoutes, called S_{IR}, is constructed first. The IRoute assignment of CQ_{max} is processed as follows:

1. Identify a sub-graph of S_{IR}, say S_s, such that S_s and CQ_{max} have the same nodes.
2. Determine the maximum clique, says SCQ_{max}, in S_s. Notably every IRoute in SCQ_{max} is sensitive to the other IRoutes in SCQ_{max}, while every IRoute in S_s overlaps the other IRoutes in S_s.
3. Process IRoutes in SCQ_{max} in decreasing order of their node degree and assign each IRoute to its bottommost feasible track. A feasible track is an assignable track and is placed non-adjacent to any previously placed IRoutes in SCQ_{max} to avoid inducing coupling capacitance. For instance, if SCQ_{max} has three IRoutes and six tracks are all available to three IRoutes, then the three IRoutes are assigned to Tracks 1, 3 and 5
4. Remove SCQ_{max} from S_s and go to step 2 if S_s is not empty.

The above process ensures that no IRoute is placed next to its sensitive IRoutes. If a sensitive-free placement cannot be realized with current track resource, then IRoute is placed on the track that results in the minimum overlapping length. The assignment of a CQ_{max} is displayed using the IRoutes in Fig. 2 and their sensitivity graph in Fig. 3(a). The first maximum clique CQ_{max} contains IRoutes a, c, f, h and j. IRoute set (a, c, f, h, j) is its related sensitive sub-graph S_s, then the first maximum clique SCQ_{max} from S_s is also (a, c, f, h, j). Since the number of available tracks equals to the number of IRoutes in SCQ_{max}, minimum weighted Hamiltonian path algorithm is employed to determine the minimum-coupling assignment order of SCQ_{max}. The found order is a-h-f-j-c (from bottom to top). The second CQ_{max} is (a, d, f, h, j) and only IRoute d is unprocessed. The last CQ_{max} is (b, e, g, i) and related sensitive sub-graph S_s contains a clique of degree 3 (IRoutes e, g and i) and a separate node (IRoute b). Thus SCQ_{max} contains IRoutes e, g and i and are distributed to tracks 5, 3 and 1 respectively. The initial track assignment result is thus obtained and shown in Fig. 3(c).

The initial track routing does not yield minimum inductive crosstalk; it can be further refined with Tabu search [10], [11]. Tabu search is an optimization algorithm similar to simulated annealing except that it stores all local optimal solutions found so far in a list to avoid trapping in the same local optimality in subsequent operation. The local optimum is defined as the best solution identified within a predefined number of iterations. Since the time taken by this searching process lengthens significantly as the number of problem instances rises, Tabu search is designed to be invoked after the assignment of each maximum clique instead of each panel to lower the problem instance size. If Tabu search is employed after the initial track routing, then the required track routing time for the whole routing region becomes very large thus slowing down the track routing. Therefore, the original flow of track routing is updated by placing the Tabu search behind the assignment of each maximum clique. The Tabu search only randomly moves newly placed IRoutes. The cost function is defined as:

$$Cost_{tabu} = \sum_{i} \sum_{j \neq i} S_{ij} \cdot \left(nb_{ij} \cdot SC_{\infty} + \overline{nb_{ij}} \cdot LSK_{ij} \right), \quad (6)$$

where $S_{ij} = 1$ (0) if IRoutes i and j are (not) sensitive to each other; $nb_{ij} = 1$ (0) and $\overline{nb_{ij}} = 0$ (1) if IRoutes i and j are (not) placed in adjacent tracks; SC_{∞} is a large constant for penalizing poor placement of two sensitive IRoutes in

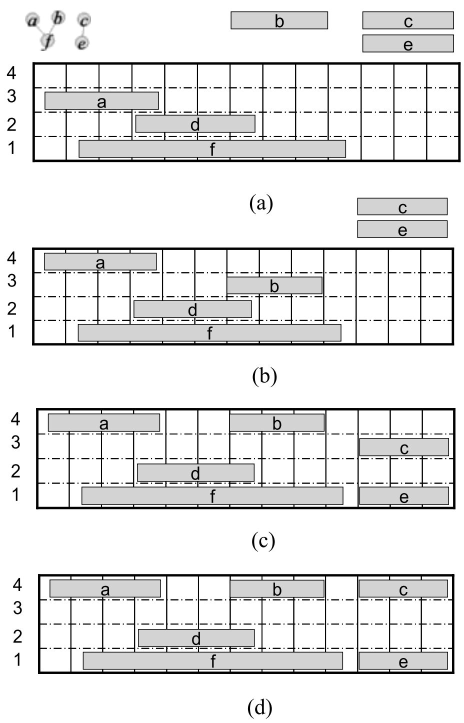


Fig. 4 (a) Initial assignment of the clique (a, d, f); (b) move IRoute a from track 3 to 4 after Tabu search and assign clique (b, d, f); (c) move IRoute b from track 3 to 4 after Tabu search and assign clique (c, e); (d) move IRoute c from track 3 to 4 after Tabu search.

adjacent tracks, and LSK_{ij} is the inductive crosstalk effect derived from the LSK model in Eq. (4).

The Tabu search gains better effects for the panels with more empty tracks. For the congested panels, less available move can be applied and thus coupling decrease is not available. The assignment in Fig. 3(c) is an example of congested panel. Figure 4 shows another example of loose panel that illustrates the proposed crosstalk-driven track routing, where its sensitivity graph is displayed at the left top part of Fig. 4(a). Figure 4(a) displays the track routing result of the first maximum clique (a, d, f) Tabu search moves IRoute a rather than IRoute d upwards to yield a minimum $Cost_{tabu}$ since d is not sensitive to f. The next identified maximum clique is (b, d, f) and only IRoute b needs to be placed. The bottommost available track is track 3, as illustrated in Fig. 4(b). IRoute b is the only candidate for random move in the following Tabu search and is moved to track 4 (Fig. 4(c)). Figure 4(c) shows the initial track routing result of the last maximum clique (c, e). Tabu search moves IRoute c from track 3 to track 4 (Fig. 4(d)).

3.3 Minimum Column Covering Based Shield Insertion

Every processed source-to-sink path in the track routing has its accumulated LSK value after the track routing stage. If the LSK value of a source-to-sink path is larger than LSK^{max}, then a crosstalk violation occurs and this path has a negative LSK slack value, which is defined as the difference between LSK^{max} and LSK. This section presents a shielding wire insertion algorithm to reduce a large LSK to below LSK^{max}. The inserting policy is to first lower the LSK of the most serious crosstalk violation net among all inductive-crosstalk-violation nets. Since not every source-to-sink path of currently processed net has crosstalk violation, an analysis on every source-to-sink path reports the paths with negative LSK slack value, called P_{si}. The problems now are where to insert shielding wires and how to insert the minimum number of shielding wires such that the diminished LSK value on every path of P_{si} at least equals the absolute value of its LSK slack value. To determine the routing regions into which shielding wires should be inserted, the maximum K value (K_{max}) of the paths P_{si} is identified as a basis. One possible approach is to iteratively insert shielding wires in the routing region containing the wire segment of the path(s) of P_{si} with the maximum K value until the LSK value of currently processed net falls below LSK^{max}. The potential limitation of this approach is that shielding wires might be inserted into routing regions used by single source-to-sink path of P_{si} rather than in the routing regions shared by source-to-sink paths of P_{si}, where shielding wires insertion is more efficient in the latter condition than in the former condition owing to the realization of LSK reduction on multiple paths of P_{si}. Thus, the routing region selection approach must favor routing regions that are shared by multiple paths of P_{si}, as well as having high K values. The following formula simultaneously addresses these two factors.

$$K_{it} \times \left(1 + \frac{N_u}{N_o}\right) \geq K_{threshold}, \tag{7}$$

where K_{it} is the K value of the ith path of P_{si} in routing region t; N_u is the number of paths of P_{si} sharing the ith path of P_{si}; N_o is the total number of paths in P_{si}, and $K_{threshold}$ is the threshold K value derived by scaling down K_{max} with a constant near and less than 1. The value of 0.75 was adopted herein. All routing regions whose K values conform to this inequality are called *critical regions* and are recognized as the candidates to accommodate shielding wires in this iteration. This selection approach chooses the routing regions whose K values are very close to or equal to K_{max} and the routing regions that are shared by multiple paths of P_{si} and have high K values.

Figure 5 shows a net example containing three source-to-sink paths (A→B, A→C, and A→D) all with LSK values exceeding LSK^{max}. Assume that routing regions 1, 2,

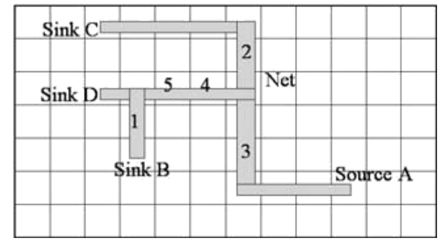


Fig. 5 Example of a net containing three source-to-sink paths (A *to* B), (A → C), and (A → D), where LSK values all exceed LSK^{max}.

Sink & LSK value / Routing region	B	C	D
	-27	-5	-10
1	8		
2		3	
3	7	7	7
4	10		10
5	3		3

Fig. 6 A *LSK* reduction table containing five critical routing regions and three source-to-sink paths with their inductive capacitance slacks.

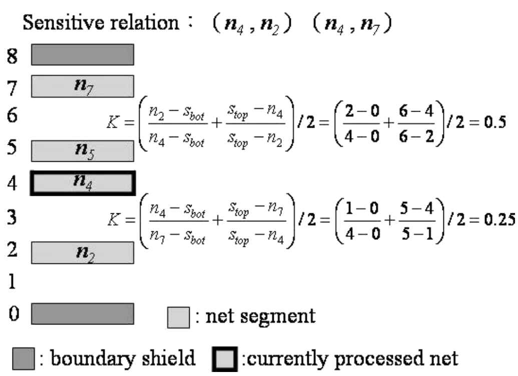


Fig. 7 Example of simulating shielding wire insertion on a routing region with three empty tracks (tracks 1, 3 and 6), wire segments of four nets (n_2, n_4, n_5 and n_7) and two sets of sensitive relation.

3, 4 and 5 are selected for shielding wire insertion based on Eq. (7). Routing regions 1 and 2 are applied by a single path (A→B); routing regions 4 and 5 are shared by two paths (A→B and A→D), and routing region 3 is shared by all three paths. Moreover, assume that the *LSK* slack values of these three paths are −27, −5 and −10. The minimum shielding wire insertion problem can then be transformed into a minimum column covering problem by putting in a table, called *LSK reduction table*, all information, including the *LSK* slack values, the path set P_{si}, and all critical regions. Figure 6 displays an *LSK* reduction table comprising all information in the top two rows and the left column, where the left column shows the critical regions; the topmost row illustrates the path set P_{si}, and the data below the path set P_{si} indicate the *LSK* slack values of the paths. An empty item in the row of routing region *i* and the column of path *j* denotes the reduced amount of *LSK* on path *j* resulting from placing a shielding wire in routing region *i*. If path *j* does not pass by routing region *i*, then the related item is always blank. Apart from the blank item, all items in a row have the same value since their values corresponds to the reduced amount of *LSK* in a routing region.

To determine the reduced amount of *LSK* in every routing region, the reduced amount of *LSK* in every possible insertion case is derived by Eq. (3). Figure 7 shows an example for estimating the reduced amount of *LSK* in a routing region. Four nets pass by this routing region (n_2, n_4, n_5 and n_7) and *LSK* reduction is intended for net n_4, which is sensitive to nets n_2 and n_7. The available tracks for accommodating shielding wires are tracks 1, 3 and 6. Since putting

```
while (there is a net with LSK > LSK^MAX && available
region for shield insertion) {
    Find the most serious crosstalk violation net；
    while (there is source-to-sink crosstalk violation path)
    {
        Identify P_si of currently processed net;
        Identify critical regions based on Eq. 7;
        Create LSK reduction table；
        while (there is uncovered column && un-selected
        row) {
            Apply row selection scheme to select a row for
            inserting a shielding wire;
            Update LSK slack values in the columns with valid
            item in the selected row;
        }
    }
}
```

Fig. 8 Minimum column covering algorithm.

shielding wire on track 1 does not separate any two sensitive nets, this track is not considered for inserting shielding wire. The new *K* value of net n_4 in the case of putting shielding wire on track 3 is 0.25 while that placing it on track 6 is 0.5. For instance, if shielding wire is placed on track 3, then the track number of every track above the new shielding wire must be decreased by 3, as illustrated in Fig. 7. In this case, shielding wire is placed on track 3 and the reduced *K* value is 1.04 − 0.25 = 0.79.

After calculating the reduced amount of *LSK* in every routing region, the *LSK* reduction table looks like the table in Fig. 6. The shielding wire insertion then becomes a minimum column covering problem. Whenever a row is chosen (a shielding wire is inserted in this region), the *LSK* slack values of the item value is added to the columns with a non-empty table item in this row. For instance, region 3 is chosen to place a shielding wire and the shielding effect contributes to all three paths, whose *LSK* slack values are incremented by 7. Furthermore, path *C* becomes free from crosstalk violation since its *LSK* slack value becomes positive. If a path becomes crosstalk-violation-free, then its associated column is said to be *covered*. Row selection scheme obeys the following two rules.

Rule 1. If a column has only one non-empty table item, then the row containing this item must be selected.

Rule 2. The row selection approach first selects the rows that cover the maximum number of columns. If there is a tie, then the rows with the maximum number of non-empty table items are selected. If there is still a tie, the scheme selects the rows that reduce most *LSK*.

This process repeats until all columns are covered or all rows are selected. If all critical routing regions have been employed, and at least one column is not covered, then the next round of critical region selection based on Eq. (7) and minimum column covering operation is performed. The entire flow ends if all columns are covered or no region is available for shielding wire insertion. In Fig. 6, region 3 is selected with *Rule* 2 in the first round to make column *C* covered. In the second round, region 4 is selected with *Rule*

2 such that column *D* becomes covered.

In the third round, region 1 is selected with *Rule* 2 to achieve the maximum *LSK* reduction. Region 5 is selected with *Rule* 2 to cover column *B* in the last round. Figure 8 illustrates the shielding wire insertion algorithm.

4. Experimental Results

All algorithms were implemented with the C++ programming language. One set of benchmarks was adopted and run on a SunBlade 2000 workstation with 1 GHz CPU and 2G RAM. Table 1 lists the statistics of six benchmark circuits. In these benchmark circuits, the sensitivity rate of each net was assumed to reach 50%, i.e., every signal was sensitive to half the number of other nets. The sensitive nets of every net were randomly generated. The *LSK* bound adopted by Xiong et al. [7] was 1000. This work applied an *LSK* bound of 5000 for the first two circuits and 10000 for the others because the circuit complexity measured by the number of nets was 10–20 times that in [7].

4.1 Statistics of Track Routing and Shield Insertion

Table 2 displays the comparison of track routing between the proposed work and that in [9]. The platform used in [9] is the same as ours. Track routing in this work includes the proposed enhanced zone-based algorithm and Tabu search algorithm. It is worth of noting that the goals of these two works are different. In the work of [9], track routing is to identify a solution with minimum coupling length between any two adjacent IRoutes, and minimum weighted Hamiltonian path algorithm is employed to every found IRoute clique for determining the IRoute ordering in the panel. As a result, the required runtime is relative large in [9]. On the other hand, this work adopts the sensitivity constraints defined by users and separates two IRoutes with sensitivity relation on non-adjacent tracks. Thus a simple and fast algorithm based on the decreasing order of node degree in SCQ_{max} is applied to the IRoutes with sensitivity constraints in the identified clique. As a result, the number of applying minimum weighted Hamiltonian path algorithm in this work is much less than that in [9], and the required runtime is also much less than that in [9]. The proposed enhanced zone-based algorithm also betters the completion rate of IRoute assignment. A 100% completion rate of IRoute assignment for all circuits is achieved in this work.

Table 3 lists the results of the proposed routing flow. Under the pre-designed constraints, the proposed routing flow yielded a crosstalk-violationfree result in every circuit. Every routing case can be completed in twenty seconds with satisfying crosstalk constraints.

To discriminate the *LSK* variations of crosstalk violation paths, Fig. 9 displays the variation of *LSK* values of circuit s15850 before and after applying shielding wire insertion, where the *y*-coordinate denotes the *LSK* value and the *x*-coordinate denotes the path number of every crosstalk violation path. Xiong et al. [7] modeled the relationship among the number of shielding wires, the sensitivity rate and the noise bound as a linear property, which were obtained by experimenting on 10000 randomly generated routing results. Figures 10 and 11 depict the relationship between the number of shielding wires and *LSK* bound in circuit s38417 (pink curves). One common observation in all cases is that the reduced rate of the number of required shields was relatively significant before the *LSK* bound reaches a certain threshold value. The rate of decrease of the number of required shields became small when the *LSK* bound exceeded this threshold. This is because most *LSK* values of crosstalk violation paths were not much larger than 5000 (the *LSK* bound of the first two benchmarks) and 10000 (the *LSK*

Table 1 Benchmark circuit statistics.

Circuits	Size(μm)	# of Layer	# of Nets	# of pins
s5378	4330×2370	3	1694	4734
s9234	4020×2230	3	1486	4185
s13207	6590×3640	3	3781	10562
s15850	7040×3880	3	4472	12566
s38417	11430×6180	3	11309	32210
s38584	12940×6710	3	14754	42589

Table 2 Statistics of track routing.

		*S*5378	*S*9234	*S*13207	*S*15850	*S*38417	*S*38584
Time (sec)	Ours	0.84	0.28	1.95	2.81	5.18	8.84
	[9]	10.6	8.1	22.6	62.6	71.3	255.6
C.R. (%)	Ours	100	100	100	100	100	100
	[9]	99.8	99.9	99.8	99.7	99.8	99.8

C.R.: completion rate = (placed IRoutes/total IRoutes)×100

Table 3 Shield insertion results.

50% sensitivity rate Circuit	Net no.	Source-to-sink path no.	Xtalk violation no.	*LSK* Bound	No. of shielding wires
s5378	1693	3124	149	5000	57
s9234	1485	2774	64	5000	23
s13207	3781	6995	102	10000	69
s15850	4471	8321	224	10000	113
s38417	11308	21035	809	10000	199
s38584	14753	28177	1618	10000	514

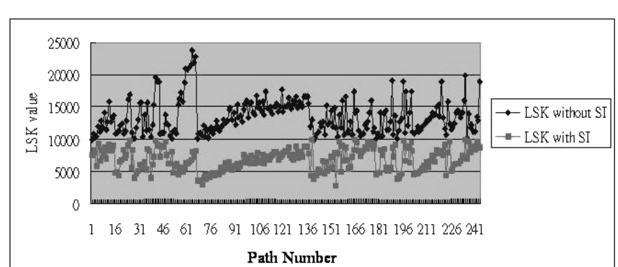


Fig. 9 Variation of *LSK* values before and after shielding wire insertion for every crosstalk violation path of s15850.

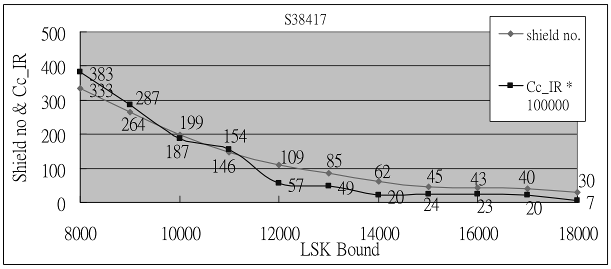


Fig. 10 The pink curve represents the relation between the number of shields and *LSK* bound, while the blue curve represents the relation between the increasing rate of coupling capacitance and *LSK* bound, both in s38417.

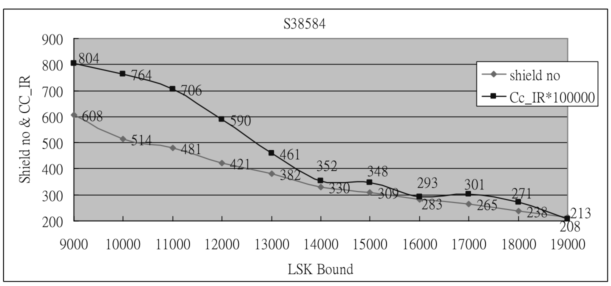


Fig. 11 The pink curve represents the relation between the number of shields and *LSK* bound, while the blue curve represents the relation between the reduction rate of coupling capacitance and *LSK* bound, both in s38584.

bound for the other benchmarks), i.e., most *LSK* values are less than the threshold value; thus many crosstalk violation paths become free of crosstalk violation since the *LSK* bound is less than the threshold value.

Since the benchmark circuits used in [7] and [12] are different from those in this work, it is infeasible to compare the quality among these three works. As runtime comparison, the maximum numbers of processed nets in [7] and [12] are 814 and 64 respectively and their runtimes are about 13106 seconds and 5 seconds. The maximum number of processed nets in this work is 14754 and the required runtime is about 20 seconds. It is obvious that this work is superior to the works in [7] and [12] in speed.

4.2 Increased Coupling Capacitance by Inserted Shields

Shield insertion can effectively decrease coupling inductance, but the coupling capacitance may be raised by the newly induced coupling capacitance between shields and their nearby IRoutes. To estimate the variation of total coupling capacitance before and after shield insertion, the crosstalk model used in [13]–[15] is employed to calculate the total coupling capacitances of every circuit before and after shield insertion. The coupling capacitance between IRoutes i and j is defined as:

$$C_c(i, j) = \alpha \cdot f_{i,j} \cdot \frac{l_{i,j}}{d_{i,j}^{\beta}}, \tag{8}$$

where α and β are technology-dependent constants, $l_{i,j}$ is the coupling length, $d_{i,j}$ is the wire spacing between IRoutes i and j and $f_{i,j}$ is the switching factor for IRoutes i and j. Two IRoutes whose coupling length is larger than zero are evaluated using Eq. (8) for their coupling capacitance. The blue curves in Figs. 10 and 11 represents the relations between the increasing rate of total coupling capacitance caused by inserted shields and *LSK* bound for circuits $S38417$ and $S38584$. Since the increasing rates are very small, all increasing rates are multiplied by 100000 in Figs. 10 and 11. Table 4 lists the statistics of the increasing rate of total coupling capacitance caused by inserted shields and the number of inserted shields, where five different *LSK* bounds are conducted on every circuit. For every *LSK* bound, the increasing rate of total coupling capacitance and the number of inserted shields are listed in the left and right parts respectively. For every circuit, the bold increasing rate represents the largest increasing rate in all *LSK* bounds. The largest increasing rate happens as the maximum number of shields is inserted in five out of six cases. In circuit $S9234$, many of the early inserted shields are distributed to loose global cells and the early inserted shields lower the required number of shields inserted in congested global cells that are selected as routing regions to insert shields in subsequent iteration when the *LSK* bound is 3000. On the contrary, as the *LSK* bound is set as 5000, some paths that are regarded as inductive crosstalk violation under the *LSK* bound of 3000 become legal paths, and congested global cells are first selected as routing regions to insert shields such that the induced coupling capacitance increases significantly. However, the maximum increasing rate of total coupling capacitance in all cases is 0.8% and relatively small.

Table 4 Statistics of coupling capacitance increasing rates and inserted shield number.

sensitivity rate:50%	*LSK* Bound									
Circuit	Coupling capacitance increasing rate (%)					Shield no.				
s5378	4000		6000		8000		9000		10000	
	0.17	77	0.15	41	0.09	19	0.02	13	0.09	9
s9234	3000		5000		7000		9000		10000	
	0.26	34	**0.40**	14	0.05	4	0.02	3	0	1
s13207	6000		8000		10000		12000		14000	
	0.67	167	0.34	100	0.15	64	0.04	30	0.05	19
s15850	10000		12000		14000		16000		18000	
	0.32	113	0.21	71	0.10	40	0.06	22	0.003	12
s38417	8000		10000		12000		14000		16000	
	0.38	333	0.19	199	0.06	109	0.02	62	0.02	43
s38584	9000		11000		13000		15000		17000	
	0.8	608	0.7	481	0.46	382	0.35	309	0.3	265

♦Coupling capacitance increasing rate (%) = ((coupling capacitance after shield insertion – coupling capacitance before shield insertion) / coupling capacitance before shield insertion) × 100.

5. Conclusions and Discussions

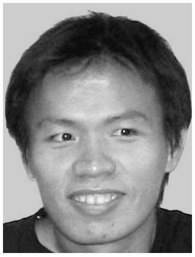
This study addresses the minimum shield insertion problem on full-chip RLC crosstalk budgeting routing according to a three-stage routing flow. In global routing, crosstalk and congestion are simultaneously considered by deriving the number of sensitive nets of the routed net and the space ratio of available tracks to total tracks. Track routing considers crosstalk minimization as well as track utilization. One difference between the proposed and previous algorithms is that the proposed algorithm simply marks IRoute as processed rather than removing it from the overlap graph, enabling the positioning of IRoutes near the high density zone so as to increase track utilization. Finally, the minimum shield insertion problem is transformed into a minimum column covering problem by constructing an *LSK* reduction table maintaining details of critical regions in which columns denote critical regions, rows denote crosstalk violation paths and *LSK* slack values, and every table item is the *LSK* reduced value after shield insertion in the critical region. Experimental results demonstrate that the proposed algorithms can yield crosstalk violation-free track routing results with shielding wires more rapidly than the method of Xiong et al. [7].

The *LSK* inductance model adopted in this work is a parallel coplanar structure. The inductive effect decreases slowly with wire separation, and also appears between wires in different layers. Accurate analysis of RLC interconnection can be obtained with Partial Element Equivalent Circuit (PEEC) method [16]. Besides, some researches, such as the work in [17], proposed a 3-D model that first individually establishes horizontal and vertical 2-D inductance models and then jointly integrates 2-D inductance models to form a complete 3-D inductance model. However the induced high computation complexity lowers its availability, especially in optimization problems. Furthermore, the scheme of shield insertion is hard to be employed for lowering inductive coupling between wires in different layers. For two wires in adjacent layers, no available space is reserved for shields. For two wires in different and non-adjacent layers that have the same preferable routing direction, such as L_i and L_{i+2} the routing plane in layer L_{i+1} can not accommodate shields for wires in layers L_i and L_{i+2} because the inserted shields have a common direction that is orthogonal to the preferable routing direction in layer L_{i+1}. Although with the limited application, the proposed algorithms can fast and efficiently minimize the coupling capacitance and solve inductance constraint violations in coplanar structure.

References

[1] K.L. Shepard and V. Narayanan, "Noise in deep submicron digital design," Proc. Int. Conf. Computer-Aided Design, pp.524–531, 1996.
[2] T. Gao and C.L. Liu, "Minimum crosstalk channel routing," Proc. Int. Conf. Computer-Aided Design, pp.692–696, Nov. 1993.
[3] D.A. Kirkpatrick and A.L. Sangiovanni-Vincentelli, "Techniques for crosstalk avoidance in the physical design of high-performance digital systems," Proc. Int. Conf. Computer-Aided Design, Nov. 1994.
[4] S.S. Sapatnekar, "A timing model incorporating the effect of crosstalk on delay and its application to optimal channel routing," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.19, no.5, pp.550–559, May 2000.
[5] L. He, N. Chang, S. Lin, and O.S. Nakagawa, "An efficient inductance modeling for on-chip interconnects," Proc. CICC, pp.457–460, 1999.
[6] L. He and K.M. Lepak, "Simultaneous shielding insertion and net ordering for capacitive and inductive coupling minimization," Proc. Int. Symp. Phys. Design, pp.55–60, 2000.
[7] J. Xiong and L. He, "Full-chip routing optimization with RLC crosstalk budgeting," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.23, no.3, pp.366–377, March 2004.
[8] S. Batterywala, N. Shenoy, W. Nicholls, and H. Zhou, "Track assignment: A desirable intermediate step between global routing and detailed routing," Proc. Int. Conf. Computer-Aided Design, pp.59–66, Nov. 2002.
[9] T.-Y. Ho, Y.-W. Chang, S.-J. Chen, and D.T. Lee, "Crosstalk- and performance-driven multilevel full-chip routing," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.24, no.6, pp.869–878, June 2005.
[10] F. Glover, "Future paths for integer programming and links to artificial intelligence," Computers and Operations Research, vol.13, no.5, pp.533–549, 1986.
[11] L. Zhang, T. Jing, X. Hong, J. Xu, J. Xiong, and L. He, "Performance and RLC crosstalk driven global routing," Proc. International Symposium on Circuits and Systems, 2004.
[12] K.M. Lepak, M. Xu, J. Chen, and L. He, "Simultaneous shield insertion and net ordering for capacitive and inductive coupling minimization," ACM Trans. Design Automation of Electronic Systems (TODAES), vol.9, no.3, pp.290–309, July 2004.
[13] H. Zhou and D.F. Wong, "Global routing with crosstalk constraint," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.18, no.11, pp.1683–1688, 1999.
[14] H.-P. Tseng, L. Sheffer, and C. Sechen, "Timing- and crosstalk-driven area routing," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.20, no.4, pp.528–544, 2001.
[15] D. Wu, J. Hu, M. Zhao, and R. Mahapatra, "Timing driven track routing considering coupling capacitance," Proc. Asia and South Pacific Design Automation Conference, pp.1156–1159, 2005.
[16] A.E. Ruehli, "Equivalent circuit models for three-dimensional multiconductor systems," IEEE Trans. Microw. Theory Tech., vol.MIT-22, no.3, pp.216–221, March 1974.
[17] T. Lin, M.W. Beattie, and L.T. Pileggi, "On the efficiency of simulated 2D on-chip inductance models," Proc. Design Automation Conference, June 2002.

Peng-Yang Hung received the B.S. and M.S. degrees in Computer and Information Engineering from Chung Hua University and National Chiao Tung University in 2003 and 2005, respectively. His research interests include layout optimization and embedded software design. He has been with the Embedded Software department, Sunplus Corporation since 2005.



Ying-Shu Lou received the B.S. and M.S. degrees in Computer and Information Engineering from National Central University and National Chiao Tung University in 2003 and 2005, respectively. His research interest includes layout optimization. He is currently with Faraday Corporation, where he is responsible for physical synthesis of DSB designs.

Yih-Lang Li received the B.S. degree in Nuclear Engineering in 1987, and the MS and Ph.D. degrees in Computer Science, in 1990 and 1996, respectively, all from National Tsing Hua University (NTHU), Taiwan. In 1995–1996, he was with Springsoft Corp. as a software engineer where he was responsible for Verilog parser and circuit information extraction. After completing his military service, he returned to Springsoft in 1998 He led the place and route group until he joined the faculty in the Department of Computer and Information Science, National Chiao Tung University in 2003. His interests include physical design automation and quantum-dot cellular automata.