POST GLOBAL ROUTING CROSSTALK RISK ESTIMATION AND REDUCTION

by

Tianxiong Xue, Ernest S. Kuh, and Dongsheng Wang

Memorandum No. UCB/ERL M96/13

10 April 1996

POST GLOBAL ROUTING CROSSTALK RISK ESTIMATION AND REDUCTION

by

Tianxiong Xue, Ernest S. Kuh, and Dongsheng Wang

Memorandum No. UCB/ERL M96/13

10 April 1996

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Abstract

Previous approaches for crosstalk synthesis are mainly localized optimization methods at the detailed routing level. Due to limited routing flexibility, they often fail to achieve satisfactory results. Furthermore, the problem of partitioning the risk tolerance bounds of nets among their routing regions, which is critical for constrained crosstalk optimization, has not been adequately addressed. This paper proposes a post global routing crosstalk optimization approach, which to our knowledge, is the first to estimate and reduce crosstalk risk at the global (instead of detailed) routing level. The crosstalk risk of each routing region is quantitatively defined and estimated using a graph-based theoretical approach. For accurate risk estimation, the risk tolerance bound of each net is partitioned appropriately among its routing regions via a two-phase integer linear programming. For high risk regions, net ripping-up and rerouting is applied to reduce their crosstalk risks. At the end of the entire optimization process, a risk-free global routing solution is obtained together with partitions of nets' risk tolerance bounds which reflect the crosstalk situation of the chip. These can greatly facilitate the generation of a risk-free final solution at later stages in the layout process. The proposed approach has been implemented and tested on CBL/NCSU benchmarks and the experimental results are very promising.

1 Introduction

Due to the scaling down of device geometry in deep-submicron technologies, interconnect wires are placed in increasingly closer proximity and higher density. As a result, the coupling capacitance between adjacent nets has increased significantly and the crosstalk noise it causes has become an important concern in high performance circuit design. If un-optimized, crosstalk may cause signal delay, logic hazards, and even malfunctioning of a circuit.

The crosstalk noise is routing-dependent, since the coupling between nets is determined by the routes of interconnects on the chip. Therefore, crosstalk risk estimation and reduction can only be carried out after a feasible routing solution of the chip is obtained. Previous approaches for crosstalk synthesis are mainly localized optimization methods at the detailed routing level[1, 2, 3, 4]. They adopt net-based approaches which estimate the crosstalk noise at each net in a region individually and reduce the coupling between adjacent nets via spacing[1, 2], track permutation [3] or track assignment[4]. Although these methods can achieve some reduction in crosstalk noise on a chip, they suffer from several drawbacks:

1. The optimization at the detailed routing level has very limited routing flexibility, since it can only adjust the routes of nets locally within a routing region, not globally among all regions on a chip. Consequently, its effectiveness depends heavily on the global routing solution, and it often fails to achieve satisfactory results especially for those regions having high densities of sensitive nets and limited routing resources. For example, it is impossible to avoid crosstalk among three nets that are sensitive to each other in a region having only four tracks at the detailed routing level.

2. Most previous approaches are not constraint-driven, but rather aim at coupling minimization. The crosstalk synthesis should be formulated as a constrained optimization process, since whether a net is subject to crosstalk violation depends not only on the couplings from its adjacent nets, but also on its *risk tolerance bound* - the maximum amount of crosstalk noise it can tolerate without affecting the functionality of circuit. The crosstalk noise at a net comes from all regions on its route, therefore, its risk tolerance bound must be partitioned appropriately among its routing regions. This risk tolerance bound partitioning problem is critical for constrained risk optimization and has not been adequately addressed so far.

In this paper, we present a post global routing crosstalk optimization approach, which to our knowledge, is the first to estimate and reduce crosstalk risk at the global (instead of detailed) routing level. Given a feasible global routing solution, sensitivities and risk tolerance bounds of nets, our approach produces a risk-free global routing solution in which all regions on the chip are free of crosstalk risks. In addition, it generates partitions of nets' risk tolerance bounds which reflect the crosstalk situation of the chip. These output can greatly facilitate the generation of a risk-free final solution at later stages in the layout process.

The entire optimization process iterates among three key components (Fig. 1): crosstalk risk estimation, risk tolerance bound partitioning and global routes adjustment. The region-based crosstalk risk estimation first constructs a crosstalk risk graph for each routing region representing its crosstalk situation based on initial partitions of risk tolerance bounds of nets. The crosstalk risk of the region, which indicates whether a risk-free routing solution is possible, is then quantitatively defined and estimated using a graph-based optimization approach. For accurate risk estimation, the impact of bound changes on regions' risks is analyzed and the current partitions of nets' risk tolerance bounds are adjusted via two-phase integer linear programming to minimize the regions' risks. If high risk regions still exist after bound partitioning, global routes adjustment is applied to reduce their crosstalk risks. First, nets whose removal leads to maximum risk reduction are identified, then they are ripped-up and rerouted with minimum cost alternative routes which consider both routing congestions and crosstalk risks of their routing regions. The entire iterative optimization process continues until a satisfactory solution is obtained.

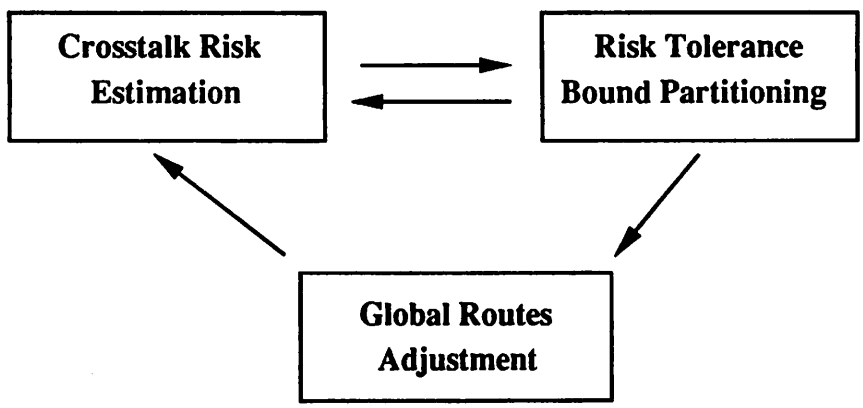

Fig. 1 Crosstalk Risk Estimation and Reduction Process

The rest of the paper is organized as follows: Section 2 discusses the region-based risk estimation method; Section 3 presents the risk tolerance bound partitioning algorithm; Section

4 explains the global routes adjustment approach; Section 5 shows experimental results which demonstrate the effectiveness of out approach; finally, Section 6 gives concluding remarks.

2 Region-based Crosstalk Risk Estimation

2.1 Crosstalk risk representation

2.1.1 Definitions

Denote G as the global routing graph of a chip, E as the set of horizontal or vertical routing regions and N as the set of nets routed on the chip. Define $C(e)$ as the capacity (i.e., number of available routing tracks) of region $e \in E$ and $N(e)$ as the set of nets routed in e. The route of net $n \in N$, $route(n)$, is formulated as the embedding of topology of n on E, i.e., $route(n) \subseteq E$.

Although crosstalk noise between net pairs may cause delay and logic hazards in a circuit, a recent study[4] shows that the couplings between certain net pairs do not affect the proper functioning of the circuit due to the logical and temporal isolations. This implies that not every pair of nets is subject to crosstalk concern in crosstalk optimization and *crosstalk sensitivity*, S_{ij}, can be specified for each net pair (i, j). For digital circuits, $S_{ij} \in \{0, 1\}$ and $S_{ij} = 1$ implies that net i, j are subject to crosstalk concern during optimization. According to the sensitivities of net pairs, $N_s \subseteq N$ is defined as the set of nets that are sensitive to other nets on the chip, i.e., $N_s = \{i | \exists j \in N, s.t.\ S_{ij} = 1\}$, and $N_s(e) \subseteq N(e)$ is defined as the set of sensitive nets routed in region e, i.e., $N_s(e) = \{i | \exists j \in N(e), s.t.\ S_{ij} = 1\}$.

Since the coupling capacitance between a net pair (i, j) is directly proportional to their coupling wire length $len(i, j)$, the crosstalk noise $noise(i, j)$ between them can be measured by: $noise(i, j) = S_{ij} len(i, j)$. In our discussion, it is assumed that crosstalk noise exists only between net pairs routed in adjacent tracks and $Adj(i, e)$ is defined as the set of nets adjacent to net i in region e. For each sensitive net $i \in N_s$, its risk tolerance bound $Bound(i)$, is defined as the maximum amount of crosstalk noise it can tolerate without affecting the functionality of the circuit. Thus, net i is "safe" under crosstalk noises from its adjacent nets if and only if:

$$\sum_{e \in route(i)} \sum_{j \in Adj(i,e)} noise(i, j, e) = \sum_{e \in route(i)} \sum_{j \in Adj(i,e)} S_{ij} len(i, j, e) < Bound(i) \qquad (1)$$

where $len(i, j, e)$ and $noise(i, j, e)$ are the coupling length and crosstalk noise between net i, j in region e, respectively. Both sensitivities and risk tolerance bounds of nets can be extracted using temporal and functional analysis[4] and are given as input to our crosstalk optimization process together with a feasible global routing solution of the chip.

Obviously, the crosstalk noise at each net in a region e, determined by the couplings between its adjacent nets, can only be calculated exactly based on a detailed routing solution of e. But once a global routing solution of e is obtained and the nets routed in e are known, we can identify whether these nets can be placed adjacent to each other in the region free of crosstalk violation. Under global routing formulation, each net routed in region e counts for an entire track in the region, i.e., no two nets share the same track in e. Therefore, we define a *routing solution of region* e at the global routing level as a routing order of nets in $N(e)$ in adjacent tracks of e from one side of the region to the other. If there exists a routing order of region e according to which each net is free of crosstalk violation, it is denoted as a risk-free routing solution of e and e is defined as risk-free. If every region on the chip is risk-free, the current global routing solution of the chip is defined as risk-free. Therefore, the goal of our region-based crosstalk risk estimation process is to identify the existence of a risk-free routing solution for each region. Notice that the risk-free routing solution defined here at the global routing level is only used for our risk estimation purpose, it does not necessarily correspond to the final routing solution of the region which is to be generated at later stages in the layout process.

2.1.2 Crosstalk violations in global routing

Since crosstalk noise at net i comes from all routing regions on its route, $Bound(i)$ must be partitioned accordingly among $route(i)$ for crosstalk estimation. Denote $Bound(i, e)$ as the partitioned risk tolerance bound of net i in routing region $e \in route(e)$, the partition of $Bound(i)$ can then be expressed as:

$$Bound(i) = \sum_{e \in route(i)} Bound(i, e) \tag{2}$$

The partitioning of risk tolerance bounds for accurate risk estimation will be discussed in Section 3.

Since each net occupies an entire track in a region under global routing formulation, it can be adjacent to no more than two nets in its above and below tracks within the region. Therefore, crosstalk violation may occur at net i in region e only in the following two cases:

- Case 1. The noise from one of i's adjacent nets violates its risk tolerance bound in e, i.e., $\exists j \in Adj(i, e)$ s.t. $noise(i, j, e) \geq Bound(i, e)$

- Case 2. The summation of noises from both of i's adjacent nets violates its bound, i.e., $\exists j, k \in Adj(i, e)$ s.t. $noise(i, j, e) + noise(i, k, e) \geq Bound(i, e)$

Nets that cause crosstalk violations at i under these cases can not be placed adjacent to i in a risk-free routing solution of region e. These two cases are referred to as crosstalk violation Case 1 and 2 in later discussions.

2.1.3 Crosstalk representation

Based on the analysis above, two graphs are defined for each routing region e, representing its crosstalk situation under risk violation Case 1 and 2.

A. **Crosstalk risk graph**

Define $CRG(e) = (N_s(e), E_s(e))$ (Fig. 2(a)) as the *crosstalk risk graph* of region e, which represents the crosstalk noises between nets in $N_s(e)$. $CRG(e)$ is a weighted graph having $Bound(i, e)$ as the weight of node $i \in N_s(e)$ and $noise(i, j, e)$ as the weight of edge $(i, j) \in E_s(e)$. Each node i in $CRG(e)$ represents a sensitive net routed in e, and each edge (i, j) satisfies: $noise(i, j, e) < B(i, e)$ and $noise(i, j, e) < B(j, e)$,
i.e., the noise between i, j does not violate the risk tolerance bounds of both net i and j in region e. According to this noise constraint, $CRG(e)$ excludes crosstalk violations under Case 1 and it represents the compatibility between net pairs, i.e., each edge $(i, j) \in E_s(e)$ implies that net pair (i, j) can be placed in adjacent tracks in region e free of crosstalk violation under Case 1.

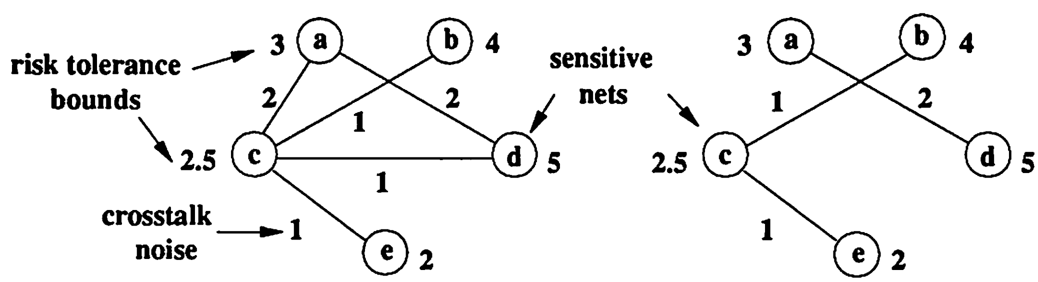

Fig. 2 (a) $CRG(e)$ (b) $CRG_{csp}(e)$

B. **Constrained simple path sub-graph**

The net compatibility represented in $CRG(e)$ is only pair-wise, i.e., the fact that net j, k are compatible with net i separately does not guarantee they can be placed adjacent to i at the same time, since the summation of noises from j, k may cause crosstalk violation at i under Case 2. For example, although net a, e are both compatible with c in Fig. 2(a), the total noise from a, e violates the risk tolerance bound of c.

For further crosstalk representation, $CRG_{csp}(e) = (N_s(e), E_p(e))$ (Fig. 2 (b)) is defined as the *constrained simple path sub-graph* of $CRG(e)$ containing simple path segments only(isolated nodes are regarded as special path segments). $E_p(e) \subseteq E_s(e)$ and $degree(i) \leq 2$ holds for every node i in $CRG_{csp}(e)$. Furthermore, each simple path segment p in $CRG_{csp}(e)$ satisfies: $noise(i, j, e) + noise(i, k, e) < Bound(i, e),\ j, k \in Adj(i, e), \forall i \in p$,
i.e., the total noise at each node i from its two adjacent nodes is less than its risk tolerance bound. According to this noise constraint, crosstalk violation under Case 2 is also excluded from $CRG_{csp}(e)$. Notice that $CRG_{csp}(e) \subseteq CRG(e)$ is not unique, the construction of $CRG_{csp}(e)$ having maximum number of edges will be discussed in Section 2.3.2.

2.2 Region-based crosstalk risk definition

2.2.1 Risk-free routing solution

According to the definition of $CRG_{csp}(e)$, each simple path segment $p = (n_1, \ldots, n_k) \in CRG_{csp}(e)$ corresponds to a risk-free routing order of nets on p. In other words, nets $n_1, \ldots, n_k$ are free of crosstalk violations under Case 1 and 2 if they are routed in region e in the same order as they appear on p. For example, path segment $p = (b, c, e)$ in Fig. 2(b) corresponds to a risk-free routing order of net b, c, e in the region.

In graph theory, a *Hamiltonian path* in graph G is defined as a simple path that visits every node in G exactly once. $CRG_{csp}(e)$ is equivalent to a Hamiltonian path if it contains just one simple path segment. According to the above analysis, a Hamiltonian path in $CRG_{csp}(e)$ corresponds to a risk-free routing solution of region e. Therefore, region e is identified as risk-free if a Hamiltonian path exists in $CRG_{csp}(e)$.

Proposition 1 *A routing region e is risk-free if $CRG_{csp}(e)$ has a Hamiltonian path.*

2.2.2 Shields

It is not always possible to find a $CRG_{csp}(e) \subseteq CRG(e)$ which contains a Hamiltonian path. When multiple simple path segments exist in $CRG_{csp}(e)$, the end nodes of these path segments can not be adjacent to each other in region e due to crosstalk violations under Case 1 and 2. To generate a risk-free routing solution of the region, we introduce the concept of *shield*. The shields in e are the non-sensitive nets or empty tracks in the region, each having zero crosstalk with other nets and infinite risk tolerance bound. This implies that shields can be used to separate the end nodes of those simple path segments so that they are no longer subject to crosstalk violations. In other words, each shield s can "connect" two disjoint path segments, p_1 and p_2 in $CRG_{csp}(e)$ into a longer path segment, $p_1 \cup \{s\} \cup p_2$, which corresponds to a risk-free routing order of nets on both p_1 and p_2. Therefore, a risk-free routing solution of region e exists if and only if there are enough shields in e to connect all simple path segments in $CRG_{csp}(e)$ into one Hamiltonian path. An example of shield application is shown in Fig. 3:

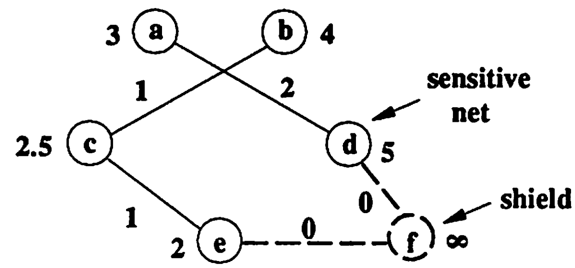

Fig. 3 Construction of Hamiltonian path using shield

Here, two disjoint path segments (a, d) and (b, c, e) are connected together by a shield f to form a Hamiltonian path, which corresponds to a risk-free routing solution of the region.

2.2.3 Crosstalk risk definition

Denote $P(e)$ as the number of simple path segments in $CRG_{csp}(e)$, $S_{avail}(e)$ as the number of shields available in region e and $S_{need}(e)$ as the number of shields needed in e to generate a risk-free routing solution of e. According to shield definition, $S_{avail}(e)$ equals the total number of empty tracks and non-sensitive nets in e and can be expressed as:

$$S_{avail}(e) = C(e) - |N_s(e)| \tag{3}$$

$S_{need}(e)$ is determined by the configuration of $CRG_{csp}(e)$ and can be calculated according to the following proposition:

Proposition 2 $S_{need}(e) = |N_s(e)| - |E_p(e)| - 1$, *where* $N_s(e), E_p(e)$ *are node and edge set of* $CRG_{csp}(e)$, *respectively.*

Proof:

Since each edge in $CRG_{csp}(e)$ links two path segments (including isolated nodes) into one, $P(e)$ can be computed by:

$$P(e) = |N_s(e)| - |E_p(e)| \tag{4}$$

By definition, $S_{need}(e)$ is the number of shields needed in e to connect $P(e)$ simple path segments into a Hamiltonian path. Since each shield adheres two disjoint simple path segments into one and reduces the number of path segments in $CRG_{csp}(e)$ by 1, $S_{need}(e)$ can be expressed as:

$$S_{need}(e) = P(e) - 1 = |N_s(e)| - |E_p(e)| - 1 \tag{5}$$

□

For risk estimation, $S_{need-min}(e)$ is denoted as the minimum number of shields needed in region e to generate a risk-free routing solution. Its corresponding $CRG_{csp}(e)$ having maximum number of edges, $|E_{p-max}(e)|$, is denoted by $CRG_{csp-max}(e)$.

According to the above analysis, the existence of a risk-free routing solution of region e is determined by the difference between $S_{need-min}(e)$ and $S_{avail}(e)$. Thus, our region-based approach defines the risk of region e, $Risk(e)$, as:

$$Risk(e) = S_{need-min}(e) - S_{avail}(e) = 2|N_s(e)| - |E_{p-max}(e)| - C(e) - 1 \tag{6}$$

For example, Fig. 3 corresponds to a $CRG_{csp-max}(e)$ of region e with $C(e) = 6$, $|N_s(e)| = 5$ and $|E_{p-max}(e)| = 3$. Thus, $S_{avail}(e) = S_{need-min}(e) = 1$ and $Risk(e) = 0$.

$Risk(e)$ indicates whether region e is risk-free. If $Risk(e) \leq 0$, there are more than enough shields in region e to generate a risk-free routing solution of e. If $Risk(e) > 0$, $Risk(e)$ is the number of extra shields needed in e, which should be minimized during the risk reduction phase discussed later. Based on this analysis, the following proposition can be established.

Proposition 3 *The current global routing solution of the chip is risk-free if and only if* $Risk(e) \leq 0$ *holds for every routing region* e *on the chip.*

2.3 Crosstalk risk estimation

2.3.1 Problem definition

According to Eqn (6), the key to the crosstalk risk estimation of region e is to construct the largest sub-graph of $CRG(e)$, $CRG_{csp-max}(e)$. The construction of $CRG_{csp-max}(e)$ can be formulated as a generalized approach for finding a Hamiltonian path in a graph and the following proposition holds:

Proposition 4 *The crosstalk risk estimation problem is NP-complete.*

Proof:

A Hamiltonian path in graph G is the largest possible maximum simple path sub-graph G_{sp-max} of G, i.e., if a Hamiltonian path exists in G, it is also a G_{sp-max} of G and can be found via G_{sp-max} construction. Therefore, the Hamiltonian path problem can be reduced in polynomial time to the problem of constructing $CRG_{csp-max}(e)$ from $CRG(e)$ by setting $CRG(e)$'s node weights to infinity and edge weights to 1, which effectively eliminates the noise constraints for $CRG_{csp-max}(e)$.

Since finding a Hamiltonian path in a graph is known to be NP-complete, the construction of $CRG_{csp-max}(e)$, i.e., the risk estimation of region e is also NP-complete.

□

2.3.2 Crosstalk risk estimation algorithm

There may exists multiple $CRG_{csp-max}(e)$s of $CRG(e)$, all having the maximum number of edges, $|E_{p-max}(e)|$. For crosstalk risk estimation, we aim at calculating the value of $|E_{p-max}(e)|$, rather than finding a specific $CRG_{csp-max}(e)$ of $CRG(e)$. In other words, we are interested in the existence of a risk-free routing solution of e, finding a specific one is the task of later stages in the layout process.

Due to the NP-complete nature of the crosstalk estimation problem, we develop a two-step algorithm for $CRG_{csp-max}(e)$ construction: first, an initial $CRG_{csp-max}(e)$ is constructed by sequentially removing minimum number of edges from $CRG(e)$, then the graph is iteratively improved to avoid locally optimal solution.

A. Initial $CRG_{csp-max}(e)$ construction

Define the degree of edge (i, j), $degree(i, j)$, as the summation of its node degrees in $CRG(e)$, i.e., $degree(i, j) = degree(i) + degree(j)$.

For the construction of $CRG_{csp-max}(e)$, edges are removed sequentially from $CRG(e)$ until the degree of each node is no more than 2 and the noise constraints for $CRG_{csp-max}(e)$ are satisfied. To minimize the number of edges that have to be removed, we adopt the following two heuristics:

1. Remove edges with largest degrees first.

2. Among edges with same degree, remove those having the largest weight (noise) first.

The initial $CRG_{csp-max}(e)$ is then constructed as follows:

1. While there exists node i with $degree(i) > 2$:

   1.1 Compute degrees of edges in current $CRG(e)$.

   1.2 Remove edges from $CRG(e)$ according to Heuristics 1 and 2.

2. While there still exists crosstalk violations at nodes:
   Remove edges according to Heuristics 2.

The set of removed edges during $CRG_{csp-max}(e)$ construction is denoted as $E_{rem}(e)$.

B. Iterative $CRG_{csp-max}(e)$ improvement

At Step A, the initial $CRG_{csp-max}(e)$ is constructed via sequential edge removal in a greedy fashion. To avoid local optimal solution, we design a two-phase improvement process which iteratively increases the number of edges in $CRG_{csp-max}(e)$.

Phase I:

Since edges are removed sequentially from $CRG(e)$ during the initial construction step, we check if any previously removed edges in $E_{rem}(e)$ can now be added back to $CRG_{csp-max}(e)$ without violating its node degree and noise constraints. The complexity of this phase is bounded by the number of edges in $CRG(e)$, which is $O(|E_s(e)|)$.

Phase II:

To further improve the locally optimal solution obtained in Phase I, we apply the so-called *k-Opt* heuristics in Phase II, which checks if more than k previously removed edges can be added back to $CRG_{csp-max}(e)$ when k edges randomly picked from the current $CRG_{csp-max}(e)$ are removed. If *k-Opt* heuristics is applied with k ranging from 1 to $|E_s(e)|-1$, a globally optimal $CRG_{csp-max}(e)$ can be found. However, this is not feasible in practice due to the $O(|E_s(e)|^{k+1})$ complexity of $k-Opt$. In our implementation, *1-Opt* and *2-Opt* are used and experiments show that they yield good results for $CRG_{csp-max}(e)$ construction. Phase I and II iterates until no further increase in $|E_{p-max}(e)|$ can be obtained.

2.4 Crosstalk risk reduction

Once the crosstalk risks of regions are estimated, those regions having positive risks can be identified. By risk definition, the total positive risk of these regions, P_{sum}, equals the total number of extra shields needed to generate a risk-free global routing solution of the chip. The basic goal of crosstalk risk reduction in global routing is to eliminate those positive risk regions so that no extra shields are needed and every routing region on the chip is risk-free.

Due to the one-to-one correspondence between a region's crosstalk risk graph and its risk estimation, crosstalk risk reduction can be achieved by modifying the configuration of CRG. According to Eqn (6), the two adjustable variables in $Risk(e)$ of region e are:

1. The number of edges in $CRG_{sp-max}(e)$, $|E_{p-max}(e)|$, determined by the routing solution of e, net sensitivity $S_{ij}s$ and partitioned risk tolerance bounds of nets $Bound(i, e)$s.

2. The number of sensitive nets $|N_s(e)|$ routed in region e, determined by the global routing solution of the chip.

These point to two ways of reducing the risks of those high risk regions on the chip:

- Increase the $|E_{p-max}(e)|$s of their $CRG_{sp-max}(e)$s by appropriately partitioning the risk tolerance bounds of nets among their routing regions. This is discussed in Section 3.

- Reduce their $|N_s(e)|$s by globally adjusting the routes of nets via net ripping-up and rerouting. This is discussed in Section 4.

3 Risk Tolerance Bound Partitioning

3.1 An example

For crosstalk risk estimation, the risk tolerance bound of each net should be partitioned among its routing regions as stated in Eqn (2): $Bound(i) = \sum_{e \in route(i)} Bound(i, e)$
Given a global routing solution and sensitivities of nets, the crosstalk risk graphs of routing regions are determined by the partitions of risk tolerance bounds of nets. Different bounds partitions may result in different CRG configurations and risk estimations as illustrated by the following example.

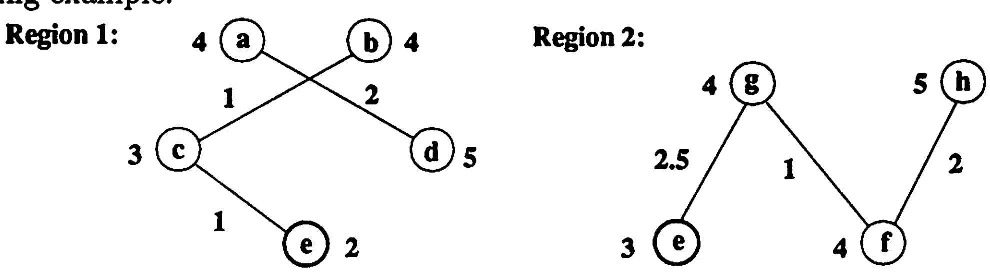

Fig. 4 (a) Partition One of $Bound(e) : Bound_1(e, 1) = 2, Bound_1(e, 2) = 3$

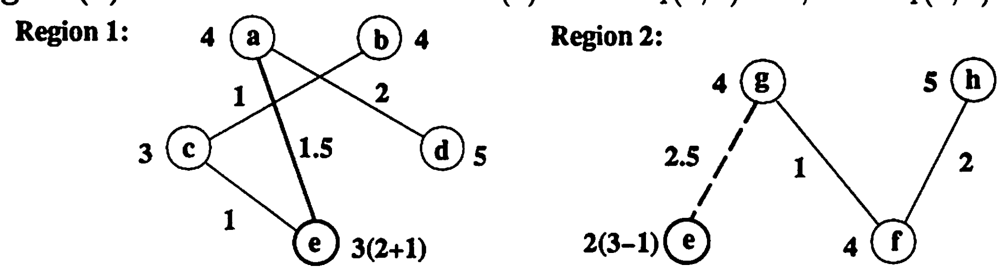

Fig. 4 (b) Partition Two of $Bound(e) : Bound_2(e, 1) = 3, Bound_2(e, 2) = 2$

Fig. 4 shows the CRG_{sp-max}s of Region 1 and 2 under two different partitions of risk tolerance bound of net e, which is routed in both regions. Suppose that $C(1) = C(2) = 5$, $Bound(e) = 5$, and the risk bounds partitions of other nets routed in the two regions are fixed. Under Partition One of $Bound(e)$ (Fig. 4(a)), $Bound(e) = Bound_1(e, 1) + Bound_2(e, 2) = 2 + 3$, $Risk(1) = 2 * 5 - 3 - 5 - 1 = 1 > 0$ and $Risk(2) = 2 * 4 - 3 - 5 - 1 = -1 < 0$, i.e., Region 1 is not risk-free under Partition One. Under Partition Two (Fig. 4(b)), $Bound(e, 1)$ is increased from 2 to 3, while $Bound(e, 2)$ is reduced from 3 to 2. As the result, edge (a, e) which violates $Bound_1(e, 1)$ (violation Case 2) under Partition One can now be added into $CRG_{sp-max}(1)$ without crosstalk violation and $Risk(1) = 2 * 5 - 4 - 5 - 1 = 0$ is reduced from 1 to 0. On the other hand, edge (e, g) is removed from $CRG_{sp-max}(2)$ since it violates

$Bound_2(e, 2)$ (violation Case 1) under Partition Two, still, $Risk(2) = 2 * 4 - 2 - 5 - 1 = 0$ is non-negative, since there is one empty track in Region 2 which can be used as a shield to connect the two path segments (e) and (g, f, h). Therefore, both regions are risk-free under Partition Two of $Bound(e)$.

This example shows that the risks of those positive risk regions can be reduced under appropriate partitions of risk tolerance bounds. Since $Bound(i)$ is a constant for every net i, the increases in $Bound(i, e)$s in some regions are at the expense of decreases in $Bound(i, e)$s in other regions on $route(i)$. Our goal for risk tolerance bound partitioning is to partition the bound of each net appropriately among its routing regions to reflect their crosstalk situations so that the total positive risk of the chip is minimized and an accurate estimation of regions' risks can be obtained.

3.2 Impact of bound changes on region's risk

For risk tolerance bound partitioning, we first analyze the impact of risk bound changes on the crosstalk risk graphs and risk estimations of routing regions. An edge (i, j) cannot be included in $CRG_{sp-max}(e)$ of region e if placing i, j in adjacent tracks results in crosstalk violations in the following two cases:

1. Crosstalk violation under Case 1 or 2 happens at both i and j, and edge (i, j) is denoted as "locked".
2. Crosstalk violation happens at only one of i and j, and edge (i, j) is denoted as "half-locked".

Each edge (i, j) in $CRG_{sp-max}(e)$ is free of risk violation at both ends and is denoted as "free".

According to its definition, the crosstalk violation at net i, j can be eliminated by increasing $Bound(i, e)$ and $Bound(j, e)$ appropriately, which switches edge (i, j) from being "locked" to "half-locked" or from being "half-locked" to "free". In the later case, (i, j) may become a new edge in $CRG_{sp-max}(e)$ and $Risk(e)$ can be reduced without global routes adjustment. On the other hand, decrease in bounds may result in fewer edges in $CRG_{sp-max}(e)$ and increase in $Risk(e)$. The change in $Risk(e)$ due to the adjustments in bounds is characterized by the following proposition:

Proposition 5 *The change in $Risk(e)$ of region e caused by adjusting $Bound(i, e)$ of net $i \in N_s(e)$ alone equals one of* $\{-2, -1, 0, 1, 2\}$.

Proof:

Since $CRG_{sp-max}(e)$ consists of simple path segments only, $degree(i) \leq 2$ holds for every node i in $CRG_{sp-max}(e)$. Therefore, the increase (decrease) in $Bound(i,e)$ can at most add (remove) two edges connecting to node i in $CRG_{sp-max}(e)$, i.e., the change in edge number of $CRG_{sp-max}(e)$, $|E_{p-max}(e)|$, equals one of $\{-2,-1,0,1,2\}$. According to Eqn (6), $Risk(e) = 2|N_s(e)| - |E_{p-max}(e)| - C(e) - 1$. Therefore, the change in $Risk(e)$ also equals one of $\{-2,-1,0,1,2\}$.

□

According to this proposition, the amount of increase in $Bound(i,e)$, $Inc(e)$, can be characterized by:

$Inc_0(i,e)$: the amount of increase in $Bound(i,e)$ that does not affect $Risk(e)$ but may switch some edges from "locked" to "half-locked" or from "half-locked" to "free".

$Inc_{1,2}(i,e)$: the minimum amount of increase in $Bound(i,e)$ that can reduce $Risk(e)$ by 1 and 2 respectively.

Clearly, $Inc_0(i,e) < Inc_1(i,e) < Inc_2(i,e)$. Notice that it is not always possible to specify $Inc_1(i,e)$ and $Inc_2(i,e)$ since the edges connecting to i are also constrained by bounds at other nodes and may not be released by increasing $Bound(i,e)$ alone.

Similarly, the amount of decrease in $Bound(i,e)$, $Cut(e)$, can also be characterized as:

$Cut_0(i,e)$: the maximum amount of decrease in $Bound(i,e)$ that does not affect $Risk(e)$ but may switch some edges from "free" to "half-locked" or from "half-locked" to 'locked".

$Cut_{1,2}(i,e)$: the maximum amount of decrease in $Bound(i,e)$ that increases $Risk(e)$ by 1 and 2 respectively.

Again, $Cut_0(i,e) < Cut_1(i,e) < Cut_2(i,e)$. Since the reduction in $Risk(e)$ caused by decrease in $Bound(i,e)$ can be no more than 2, $Cut_2(i,e) = Bound(i,e)$.

3.3 Risk tolerance bound partitioning

3.3.1 Problem formulation

Based on the above characterization of risk bound adjustment, the bound partitioning problem can be formulated as:

Given initial partitions of risk tolerance bounds of nets, adjust the partitioned bounds of each net among its routing regions by Incs and Cuts so that the total positive risk of the chip is minimized and the chip's risk estimation becomes accurate.

Notice that $Risk(e)$s are discrete functions of $Bound(i,e)$s, i.e., the risks of regions do not change continuously with respect to adjustments in bounds. Due to this discrete nature, the risk tolerance bound partitioning problem is formulated as an iterative two-phase integer linear programming (ILP) with the following objectives:

Phase I: Release maximum number of "locked" edges to "half-locked" so that they may become "free" in Phase II.

Phase II: Minimize the total positive risk of the chip by switching "half-locked" edges to "free".

3.3.2 ILP formulation for release of "locked" edges

Since the release of "locked" edges does not affect the risks of routing regions, the adjustment in $Bound(i,e)$ should be no more than $Cut_0(i,e)$ and $Inc_0(i,e)$. Denote $Inc_{0k}(i,e) \leq Inc_0(i,e)$ as the minimum amount of increase in $Bound(i,e)$ that can release k "locked" edges connecting to i, $Cut_{0k}(i,e) \leq Cut_0(i,e)$ as the maximum amount of decrease in $Bound(i,e)$ that switches k edges from "half-locked" to "locked" status (the edges switching for "free" to "half-locked" do not need to be considered). In this ILP formulation, $u_k(i,e)$ and $v_k(i,e)$ are defined as binary variables indicating whether $Bound(i,e)$ is increased by $Inc_{0k}(i,e)$ or reduced by $Cut_{0k}(i,e)$ respectively. The ILP optimization aims at maximizing the increase in total number of "half-locked" edges and is formulated as:

Maximize $\sum_e E(e)$

Subject to:

$$\sum_{i \in N_s(e)} \sum_k k u_k(i,e) - \sum_{i \in N_s(e)} \sum_k k v_k(i,e) = E(e) \qquad \forall e, Risk(e) > 0$$

$$\sum_{e \in route(i)} \sum_k Inc_{0k}(i,e) u_k(i,e) \leq \sum_{e \in route(i)} \sum_k Cut_{0k}(i,e) v_k(i,e) \qquad \forall i \in N_s$$

$$0 \leq \sum_k u_k(i,e) + \sum_k v_k(i,e) \leq 1, \;\; u_k(i,e), v_k(i,e) \in \{0,1\} \qquad \forall e \in route(i), \forall i \in N_s$$

The first constraint defines $E(e)$ as the increase in the number of "half-locked" edges for those positive risk regions after bound adjustment. Notice that $E(e)$ is an approximation of the actual increase, since bound adjustments at different nodes are not independent of each other. The second constraint specifies the "supply" and "demand" relation for each sensitive net e, i.e., the increases in $Bound(i,e)$s in some regions must be balanced by the decreases in $Bound(i,e)$s in other regions on $route(e)$. The third constraint indicates that $Bound(i,e)$ can only be increased or decreased by a certain amount once at a time.

Although only portions of Inc_0s and Cut_0s are used for bound adjustment, the new bounds partitions after optimization may result in changes in regions' risks. This is due to the fact that Incs and Cuts are estimated for each node individually, while bounds at different nodes in the region are adjusted at the same time. In an actual implementation, this ILP phase can be integrated with the risk minimization phase described below.

3.3.3 ILP formulation for risk minimization

Define $x_{1-2}(i,e)$ and $y_{0-2}(e)$ as binary variables indicating whether $Bound(i,e)$ is increased by $Inc_{1-2}(i,e)$ or decreased by $Cut_{0-2}(i,e)$ respectively during bound adjustment. This ILP phase aims at minimizing the total positive risk of the chip and is formulated as:

Minimize $\qquad \sum_e R(e)$

Subject to:

$$Risk(e) + \sum_{i \in N_s(e)} (y_1(i,e) + 2y_2(i,e) - x_1(i,e) - 2x_2(i,e)) = R(e) \qquad \forall e, Risk(e) > 0$$

$$\sum_{e \in route(i)} (Inc_1(i,e)x_1(i,e) + Inc_2(i,e)x_2(i,e))$$

$$\leq \sum_{e \in route(i)} (Cut_0(i,e)y_0(i,e) + Cut_1(i,e)y_1(i,e) + Cut_2(i,e)y_2(i,e)) \qquad \forall i \in N_s$$

$$0 \leq x_1(i,e) + x_2(i,e) + y_0(i,e) + y_1(i,e) + y_2(i,e) \leq 1 \qquad \forall e \in route(i), \forall i \in N_s$$

$$x_1(i,e), x_2(i,e), y_0(i,e), y_1(i,e), y_2(i,e) \in \{0,1\}$$

Similar to the ILP formulation in Phase I, the first constraint defines $R(e)$ as the updated risk of region e after bound adjustment. The second constraint enforces that the "demands" for risk bounds can be no more than the "supplies" for each sensitive net. The third constraint indicates that $Bound(i,e)$ can be updated only once at a time. Like $E(e)$, $R(e)$ is a linearized approximation of the actual risk of region e under the updated bounds partitions because of the simultaneous adjustments of $Bound(i,e)$s of nets in e. Nevertheless, minimizing $R(e)$ points to the right direction of bound adjustment for positive risk minimization and the accurate risks of regions can be estimated after each round of ILP during optimization.

3.3.4 Risk tolerance bound partitioning algorithm

The risk tolerance bound partitioning algorithm is designed as an iterative process. Initially, nets' risk bounds are partitioned uniformly among their routing regions. At each iteration, the current bounds partitions are adjusted for positive risk minimization and the regions' risks are updated accordingly. This process continues until the total positive risk of the chip is minimized.

Risk tolerance bound partitioning algorithm

1. Initial bound partitioning:
   Partition the risk tolerance bound of each net uniformly among its routing regions.

2. Estimate the crosstalk risk of each region on the chip.

3. While reduction in positive risk is possible:

   3.1 Calculate Incs and Cuts for the current partitions of risk tolerance bounds.

   3.2 Solve two-phase ILP optimization for risk minimization.

   3.3 Update crosstalk risk graphs and regions' risks.

The regions' crosstalk risks may be over estimated initially, since uniform bounds partitions do not reflect the actual crosstalk situation of the chip. After risk tolerance bound partitioning, the total positive risk of the chip is minimized, indicating fewer regions and nets are subject to global routes adjustment for crosstalk risk reduction. This speeds up the generation of a risk-free global routing solution of the chip.

4 Global Routes Adjustment

Once an accurate estimation of crosstalk risk situation on the chip is obtained after risk tolerance bound partitioning, the regions with positive risks can be identified. According to Eqn (6), the crosstalk risk of a region can be decreased by reducing the number of sensitive nets routed in it. Since adjusting routes of nets globally may affect the quality of the current global routing solution in terms of routing density, total wire length, number of vias and timing properties, the number of nets whose routes have to be adjusted for risk reduction should be minimized, and the global routes adjustment problem is formulated as:
Generate a risk-free global routing solution of the chip by ripping up and rerouting minimum number of nets from current positive risk regions.

4.1 Net ripping-up

For each positive risk region e, we define $N_r(e) \subseteq N_s(e)$ as the smallest set of nets to be ripped-up from e for risk reduction, i.e., the removal of nets in $N_r(e)$ from e reduces $Risk(e)$ to 0. The relation between risk reduction and net ripping-up is stated by the following proposition:

Proposition 6 *The reduction in $Risk(e)$ of region e caused by ripping-up net i from e, $Risk_{dec}(i, e) \in \{0, 1, 2\}$; more precisely, $Risk_{dec}(i, e) = 2 - degree(i)$, where $degree(i)$ is the degree of node i in $CRG_{sp-max}(e)$.*

Proof:

Ripping up net i from region e deletes node i and its connecting edges from $CRG_{sp-max}(e)$. As a result, $|N_s(e)|$ is reduced by 1 and $|E_{c-max}(e)|$ is reduced by $degree(i)$. Thus, the reduction in $Risk(e)$ can be expressed as $Risk_{dec}(i, e) = 2 - degree(i)$, according to the definition of $Risk(e)$ in Eqn (6). Since $CRG_{sp-max}(e)$ is a simple path sub-graph, $degree(i) \in \{0, 1, 2\}$, which implies $Risk_{dec}(i, e) \in \{0, 1, 2\}$.
□

Ripping up a net from region e frees one track in e which can be used as a shield. According to the above proposition, removal of node with degree 0 and 1 in $CRG_{sp-max}(e)$ can reduce $Risk(e)$ by 2 and 1 respectively, while removal of nodes with degree 2 does not change $Risk(e)$. Thus, $N_r(e)$ can be constructed as follows:

1. Choose nodes with degree 0 while they exist.

2. Choose a node with degree 1, break ties by selecting one which connects to node also with degree 1. Go back to Step 1.

Nodes with degree 0 are chosen first at Step 1 since their removal can result in maximum reduction in $Risk(e)$. Each node i with degree 1 connects to another node j having degree 1 or 2 in $CRG_{sp-max}(e)$. At Step 2, priority is given to node i which connects to a node j with $degree(j) = 1$, since j can become a new 0 degree node after the removal of i and its connecting edge. This iterative net selecting process continues until $\sum_{i \in N_r(e)} Risk_{dec}(i, e) \geq Risk(e)$, i.e., ripping up nets in $N_r(e)$ from e reduces $Risk(e)$ to 0.

4.2 Net rerouting

Once nets in $N_r(e)$ are identified, they are ripped-up from region e and rerouted through other regions on the chip with a minimum cost alternative route. To this end, we adopt a modified version of the global router developed in [5]. The original router is extended to take the regions' crosstalk risks into consideration, in addition to other concerns in global routing such as densities, wire lengths, number of vias, timing constraints, etc. Analogous to net ripping-up, the rerouting of nets may result in increase in risks of regions on their new routes. To minimize the increase of positive risk, our router reroutes those ripped-up nets through regions having the lowest risks so that least number of new positive risk regions are created and few iterations are required to generate a risk-free routing solution of the chip.

4.3 Global routes adjustment algorithm

The global routes adjustment is formulated as an iterative optimization process, which updates the regions' risks and partitions of risk tolerance bounds after each round of net ripping-up and rerouting.

Global Routes Adjustment Algorithm

While there exists region e on the chip with $Risk(e) > 0$:

1. Identify set of nets $N_r(e)$ to be ripped up from region e for risk reduction.

2. Reroute the ripped-up nets with minimum cost alternative routes.

3. Redo risk estimation and bound partitioning.

5 Experimental Results

Our post global routing crosstalk risk estimation and reduction approach has been implemented and tested on a DEC 5000/125 workstation. Four test circuits constructed from the CBL/NCSU building-block benchmarks, ami33, hp, xerox and ami49 are used. The specifications of these circuits are listed in Table 1, where G_{size} refers to the the size of global routing graph of the chip.

Table 1. Benchmark specifications

Circuit	# macro cells	# nets	# pins	G_{size} (row x col)
ami33	33	123	442	28 x 23
hp	11	83	309	289 x 228
xerox	10	203	696	24 x 24
ami49	49	408	953	184 x 139

The feasible global routing solution of these chips are generated by the performance-driven global router[5]. In our experiments, circuit ami33 and xerox are each tested under two different placement/global routing solutions, denoted as $*.1$ and $*.2$ respectively. The ILPs for bound partitioning are solved by *lp-solve* optimization tool. Since there are no standard benchmarks having net sensitivity information, our crosstalk optimization approach is tested under all possible values of both net sensitivity ratio, which is the percentage of net pairs in the circuit that are subject to crosstalk risk concern, and risk tolerance bound of each net, specified as the percentage of total net length allowed for coupling with other nets.

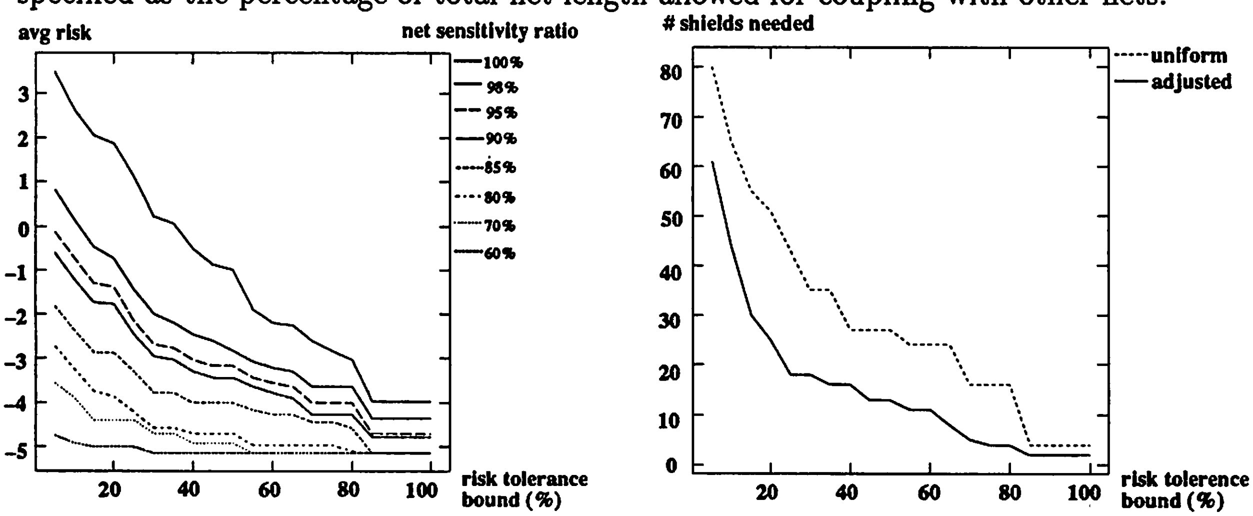

Fig. 5 (a) Risk estimation of Chip (b) Risk bound partitioning

Fig. 5 shows the testing results on ami33.1. Fig. 5 (a) illustrates how the average risk of regions on the chip varies with different net sensitivity ratios and risk tolerance bounds (partitioned uniformly in this test). It can be observed that crosstalk risk decreases as bound increases and sensitivity ratio decreases. This is due to the fact that nets having larger risk tolerance bounds are less vulnerable to crosstalk violation, and fewer shields are needed when fewer net pairs are subject to crosstalk concern.

Fig. 5 (b) compares the total number of extra shields needed (i.e., total positive risk) on the chip for a risk-free global routing solution under two different partitions of risk tolerance bounds: uniform and adjusted by our bound partitioning algorithm. Here, the results are measured under 100% sensitivity ratio. It can be seen that the risk estimation becomes more accurate under adjusted bounds partitions, and the number of shields needed is reduced drastically by over 50% for the entire range of bound specifications.

Table 2. Estimation of High Risk Regions Before Global Routes Adjustment

Testing Circuit	# positive risk regions			total # shields needed			# nets to be ripped-up		
	uniform	adjusted	-%	uniform	adjusted	-%	uniform	adjusted	-%
ami33.1	7	4	43	27	13	52	15	8	47
ami33.2	13	0	100	17	0	100	13	0	100
hp	39	39	0	105	59	44	72	48	33
xerox.1	12	5	58	44	10	77	24	5	79
xerox.2	53	43	19	175	88	50	103	60	42
ami49	214	166	22	375	270	28	232	166	28

Table 3. Estimation of High Risk Regions After Global Routes Adjustment

Testing Circuit	# positive risk regions		total # shields needed		# nets to be ripped-up	
	uniform	adjusted	uniform	adjusted	uniform	adjusted
ami33.1	0	-	0	-	0	-
hp	0	-	0	-	0	-
xerox.1	11	0	25	0	14	0
xerox.2	15	0	38	0	23	0
ami49	24	0	48	0	24	0

For crosstalk risk reduction, our main focus is on regions with positive risks on the chip. Table 2 and 3 show estimations of positive risk regions under uniform and adjusted partitions of risk tolerance bounds before and after global routes adjustment, respectively. Here, results are measured under 100% net sensitivity ratio and risk tolerance bound at 50% of net wire length.

When applied before global routes adjustment (Table 2), bound partitioning reduces the numbers of positive risk regions, extra shields needed and nets to be ripped up for risk reduction by an average of 40%, 59% and 55% respectively, which means fewer nets need to be ripped-up and rerouted based on the accurate risk estimation. In case of circuit ami33.2, global routes adjustment is avoided since bound partitioning eliminates high risk regions on the chip. When applied after net ripping-up and rerouting (Table 3), bound partitioning reduces all those three numbers to 0 (for circuit ami33.1 and hp, nets' bounds partitions do not need to be adjusted), indicating that only one round of global routes adjustment is needed to generate a risk-free global routing solution for each circuit. This also implies that our net ripping-up and rerouting method is very efficient for risk reduction. Our experiments show that there were little changes in routing densities and wire lengths of nets in the global routing solutions due to limited global routes adjustments.

6 Conclusions

This paper presents the first approach for crosstalk risk estimation and reduction at the post global routing level. It estimates risk of each routing region using a graph-based optimization approach and globally adjusts routes of nets for risk reduction. It produces a risk-free global routing solution of the chip together with appropriate partitions of risk tolerance bounds of nets which reflect the crosstalk situation of the chip. These can greatly facilitate the generation of a risk-free final solution at later stages in the layout process.

References

[1] H. Chen and C. Wong, "Wiring and Crosstalk Avoidance in Multi-Chip Module Design", *Proc. CICC 92*, pp. 28.6.1-28.6.4, 1992.

[2] K. Chaudhary, A. Onozawa and E.S. Kuh, "A Spacing Algorithm for Performance Enhancement and Cross-Talk Reduction", *Proc. ICCAD 93*, pp. 697-702, 1993.

[3] T. Gao, C. Liu, "Minimum Crosstalk Switchbox Routing", *Proc. ICCAD 94*, pp. 610-615, 1994.

[4] D. Kirkpatrick, A. Sangiovanni-Vincentelli, "Techniques for Crosstalk Avoidance in the Physical Design of High-Performance Digital Systems", *ICCAD 94*, pp. 616-619, 1994.

[5] D. Wang, E.S. Kuh, "Performance-Driven Interconnect Global Routing", *Proc. 6th Great Lakes Symp. on VLSI*, pp. 132-136, 1996.