

A Coupling and Crosstalk Considered Timing-Driven Global Routing Algorithm for High Performance Circuit Design

Jingyu Xu, Xianlong Hong,

Tong Jing, Ling Zhang

Dept. of Computer Science & Technology, Tsinghua Univ., Beijing, 100084, P. R. China

Jun Gu

Dept. of Computer Science, Hong Kong Univ. of S & T, Hong Kong, P. R. China

Abstract - With the exponential reduction in scaling of feature size, inter-wire coupling capacitance becomes the dominant part of load capacitance. Two problems are introduced by coupling, delay deterioration and crosstalk. This paper presents a timing-driven global routing algorithm with consideration of coupling effects and crosstalk avoidance. Our work differs from the existing ones in that we design a global routing “framework” which performs well in routability, timing, and also facilitate the detailed routing in crosstalk avoidance. Experimental results on industrial circuits show that, the algorithm leads to substantial delay reduction and effective crosstalk elimination.

I. Introduction

As the CMOS technology enters the very deep sub-micron era, the shrinking of geometries brings the circuit designers a growing number of issues previously considered with the second order priority. One of the biggest concerns is interconnect *crosstalk*[1], which is usually due to the capacitive coupling between the “victim” net and one or more “aggressor” nets. For 0.18 micron designs, the coupling capacitance can exceed 70 percent of the total interconnect capacitance[2]. Ignoring such coupling effects may lead to significant deviations between actual and nominal timing responses, power consumptions and functional behaviors.

Previous works[3-6] have various contributions to performance-driven global routing based on conventional interconnect delay metrics[7, 8]. These algorithms are straightforward and efficient regarding specific test cases. However, with recent advances in VDSM technology, the switching cross-coupling experienced by critical wires becomes one of the unforeseen problems for these traditional approaches. Therefore, it is of increasing importance to consider and control coupling effects to guarantee a real reliable and high-performance design.

Most early works on crosstalk avoidance are focused on detailed routing[9-11], where the estimation of crosstalk is accurate but the flexibility to avoid it is restricted. A few recent works have developed coupling-aware techniques during global routing phase. In [12], global routing with crosstalk constraints has been studied. An extended global routing problem was addressed in [13] to consider

simultaneous shield insertion and net ordering with RLC crosstalk constraints. Typical methods also include post-global-route measures[14] to eliminate crosstalk. However, none of these works explicitly combine crosstalk elimination with the timing optimization problem, which remains one of the most important tasks for global routing to deal with.

A very accurate crosstalk calculation begins with detailed routing. But it is often difficult to find a crosstalk-feasible solution in detailed routing if global routing is crosstalk-blinded. The goal of this work is to develop a technique to deal with crosstalk in a global view for decreasing the difficulty in getting a crosstalk-free solution in detailed routing, while satisfy the timing constraints and routability at the same time. Our approach works as follows. A timing-relax method is applied in the earlier phase to eliminate congestion and optimize delay simultaneously. In the later phase, a crosstalk control method is devised to find the crosstalk-feasible solution. By utilizing a conservative crosstalk model, our approach produces the crosstalk reduction by releasing the regions of high coupling “risk” with topological optimization. Our work differs from the existing ones in that we design a global routing “framework” which performs well in routability, timing, and also facilitates the detailed routing in crosstalk avoidance. The algorithm achieves promising results on industrial circuits.

The remainder of this paper is organized as follows. In Section II, we formulate the global routing problem for symbolic analysis. In Section III, the timing analysis strategy is discussed. The global routing algorithm is given in detail in Section IV. Section V shows the experimental results and Section VI is an overall conclusion.

II. Problem Formulation

We first give the definition on notations. For SC layout design, the chip is divided into a rectangular array of $M_{row} \times M_{col}$ cells called global routing cells (GRCs). Global routing graph (GRG) is the dual graph, which is composed of gridlines and crossings. Let $G = (V, E)$ be the global routing

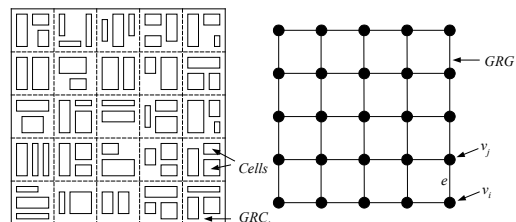


Fig. 1. Global routing graph (GRG)

*This work was supported partly by Hi-Tech Research and Development (863) Program of China under Grant No. 2002AA1Z1460, the National Natural Science Foundation of China(NSFC) 60121120706, NSFC under grant No.60373012, the National Natural Science Foundation of USA(NSF) CCR-0096383, the 973 Program of China G1998030403 and Key Faculty Support Program of Tsinghua Univ. [2002] 4.

graph shown by the solid line in Fig 1. Node v_i represents the center point of GRC_i . The edge that links v_i and v_j is a GRG edge (e) with length l denoting the distance between v_i and v_j . A non-negative number c_e , called edge capacity, is assigned to e , indicating the number of available tracks between the corresponding GRCs.

In general, the crosstalk between two parallel wires is proportional to their coupling length, and is inversely proportional to their separating distance. Given a net, the crosstalk effects from all neighboring wires may not happen at the same time. But characterizing all cases requires exhaustive timing. Considering the worst case, the summation of all effects from other nets is defined as the total crosstalk on a net i (Ct_i). That is,

$$Ct_j = \sum_{i=1, i \neq j}^{N_n} Ct_{ij} \quad (1)$$

where Ct_{ij} is the value of crosstalk on net j from net i and N_n the total number of nets within a design.

Without loss of generality, we tackle the timing optimization problem by minimizing the longest path delay from any input port to any output port of a circuit.

$$\text{Let } u_{ij} = \begin{cases} 1, & \text{when net } n_i \text{ passes edge } e_j \\ 0, & \text{elsewise} \end{cases}$$

Thus, the timing-driven global routing problem can be formulated as follows.

$$\text{Minimize } \max_i \text{delay}(P_i) \quad i \in \{1, 2, \dots, m\}$$

$$\text{Subject to } f_j = \sum_{i=1}^{N_n} u_{ij} \leq c_j$$

$$Ct_j = \sum_{i=1, i \neq j}^{N_n} Ct_{ij} \leq x_j$$

where m is the number of paths. $\text{delay}(P_i)$ denotes the delay of path P_i composed of gates and net wires. Two sets of constraints should be satisfied. Let f_j be the total demand of the nets using edge e_j . f_j should be no greater than the edge capacity c_j . And the total crosstalk on net j (Ct_j) should not exceed the corresponding threshold x_j .

III. Timing Analysis

For wire-load estimation, we apply a wire-load model that is useful for timing and noise analysis in an early design stage [15]. It is derived by simulation of metal wires in different layers to obtain the first cut parasitic numbers and then curve fitting. The largest error in estimation of parasitics of a metal wire is within 5% of the field solvers to measure

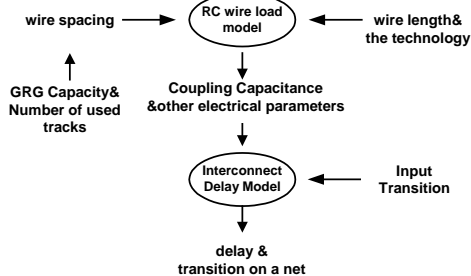


Fig. 2. Delay calculation considering coupling effects

these parasitics. With specified GRG capacity and number of used tracks, an estimated wire spacing and other geometrical information are the input of the wire load model. All capacitance around a given conductor can be extracted including the coupling capacitance.

Since the conventional interconnect delay metrics[7,8] do not take signal input rise/fall time into account or propagate signal transition time during delay evaluation, we apply more advanced delay estimation methods. The built-in delay evaluation engine uses congruence transformation technique [16] to reduce the order of a large RC net-list while stability and passivity can be guaranteed. Considering the trade-off between accuracy and speed, an adaptive method is used to control the order of net-list reduction. In benchmark testing, the accuracy can be within 1% of SPICE simulation. Fig 2 shows the delay estimation method of the algorithm utilizing above models, which takes coupling effects into consideration.

We adopt table-lookup model for gate delay estimation. In our work, the lookup tables are obtained from industrial circuit libraries.

IV. Global Routing Algorithm

Our approach mainly consists of two phases. In the earlier phase, we apply a timing-relax method. A heuristic algorithm is firstly developed to construct the initial delay-optimal solution, followed by an optimization algorithm, which utilizes coupling effect as a heuristic to optimize the circuit delay and congestion. Then, we devise the crosstalk control algorithm to minimize the total crosstalk.

A. The Initial Steiner Tree Algorithm

The initial solution is generated with two main concerns. (1)Timing performance is given first priority. (2)A time consuming initial solution is not desirable. With these considerations, we apply ITDT algorithm[17] in the first iteration of the routing tree construction for its simplicity.

Considering the longest path delay of the circuit, constructing minimum delay tree for each net respectively may not necessarily yield the best timing for the circuit. We then develop a short-term iteration algorithm to approach the optimal solution. The pseudo-code is shown in Fig 3.

Initial_solution_optimization_algorithm()

1) For(each net) do

Let the sink pin t_i yielding maximum $L(s, t)$ be the target critical node t .
ITDT_algorithm(t); /*find best timing tree with respect to delay(s, t)*

2) Check each path and find the longest delay path P_i .

3) $dp = \text{pathdelay}(P_i)$; /*the temporary longest path delay*/

Improve = 0; /*denote whether the solve has been improved*/

4) For(each net, on P_i) do

If(critical node $t_i \notin P_i$) and (net ID not marked) do

Get the actual end pin t_a , $t_a \in V(\text{net}_i)$ and $t_a \in P_i$

Find the biggest sub-tree rooted at v for $\forall v \notin \text{path}(s, t_a)$

Minimize the total capacitance of Steiner tree below v ;

Update longest delay path and calculate new dp ;

If($\Delta dp \geq 0$)

Set: Improve = 0;

Restore the initial tree of net _{i} ;

else

Set: Improve = 1;

Mark the net ID;

5) If(Improve == 0), stop;

6) Else, go step2), iterate;

Fig. 3. Optimization algorithm for the initial routing solution

When minimizing the total capacitance of Steiner tree below a node v to reduce $\text{delay}(s, t_a)$ in step 4, we either increase the wire spacing to reduce the inter-wire coupling capacitance or construct minimum wire length sub-tree to reduce the ground capacitance. The proof for effectiveness of the optimization algorithm in step 4 can be found in [18].

B. Timing Optimization Considering Coupling Effects

Based on the initial solution, we optimize the network topology of the circuit to adjust most congested areas. To maintain timing performance of the initial solution, the most critical paths should be kept in the original topology. Applied simultaneously with the congestion optimization, the basic idea of timing optimization is to perform coupling effects transference, which directs the coupling effects to transfer to areas that are not critical for circuit delay. An extended congestion weight is used to denote the combination of both coupling weight and congestion weight.

Definition 1 The extended congestion(EC) of a segment is the combination of coupling and congestion evaluation.

The EC weight of segment i can be defined by the following formulae:

$$\tilde{w}_i = \alpha_1 w_{congi} + \alpha_2 w_{coupi} \quad (\alpha_1 + \alpha_2 = 1) \quad (2)$$

$$w_{congi} = (f_i + 0.5\varepsilon)/(c_i + \varepsilon) \quad (3)$$

where w_{congi} is contributed merely by congestion of the segment and w_{coupi} by coupling effects. α_1 was experimentally set to be 0.8. ε is a small constant.

As the coupling capacitance is inversely proportional to the spacing between neighboring wires, the coupling weight can be defined as:

$$w_{coupi} = \frac{C_{ci}}{C_{mi}} \quad (4)$$

where coupling capacitance C_{ci} depends on spacing and wire length of the segment, C_{mi} is the maximum coupling capacitance calculated under minimum spacing condition.

With the intention to optimize the circuit delay, we naturally note that adjusting the route on the critical path would result in a smaller circuit delay. The EC weight of segment i on the longest delay path is given by:

$$\tilde{w}_i = \alpha_1 w_{congi} + \mu \alpha_2 w_{coupi}, \quad \mu > 1 \quad (5)$$

Since $\mu \alpha_2 > \alpha_2$, the extended congestion weight of segment on critical path is magnified. Thus when a net has alternative routes available in rerouting, it is more likely to choose the GRG segment on non-critical path because it has comparatively lower “congestion”. In this way, the routing demands on critical path transfer to non-critical paths, which may increase their path delay. However, the coupling effects transference is directed to facilitate a delay reduction if considering the circuit timing as a whole, and consequently guarantees space of further delay optimization.

C. Crosstalk Control

The purpose of the crosstalk control algorithm is to find a crosstalk-feasible solution based on some estimation method. In practice, designers may want to control the crosstalk among logically sensitive nets, for which a switching event

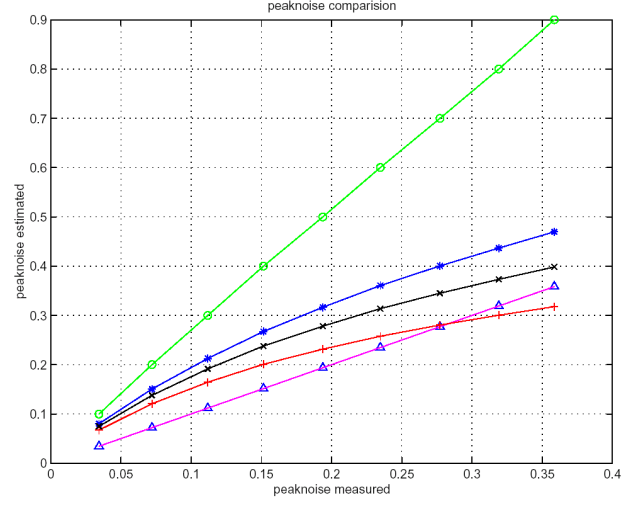


Fig. 4. Comparison of three noise estimation metrics. The x coordinate is the peak noise measured by SPICE. The y coordinate is the peak noise estimated by crosstalk metrics. The victim was not switching in all cases.

on one net causes the other to malfunction. Their relation is characterized by the use of a sensitivity graph [19]. If the sensitivity graph is not available, we assume all the nets are sensitive to adjacent nets.

The crosstalk constraints are defined through the definition of “crosstalk risk bound” for each net. Our approach produces the crosstalk reduction by releasing the regions of high crosstalk “risk”. For example, suppose there are two wires routed in parallel over a number of GRG edges. One wire may violate corresponding crosstalk constraint if they remain being placed in parallel over the whole distance. We shall reroute part of the “risky” wire to change the topology to produce a crosstalk reduction. The violations can be checked during detailed routing as guiding information to tune our measurements.

• Crosstalk Model and Basic Assumptions

As described in Section II, crosstalk is a function partially depending on the parallel length and spacing between neighboring wires. Calculation of the noise waveform can accurately determine the crosstalk volume, while it would be time-consuming for global routing crosstalk estimation. Some previous works adopt the formula in [20], where crosstalk is given by:

$$C = \alpha \frac{\text{length}}{\text{distance}^\beta}$$

where β is an experimentally estimated constant. Considering more direct factors such as the impact of coupling capacitance and driver strengths, etc, we adopt a noise metric to estimate crosstalk effects. Given a victim net j , the crosstalk on net j from net i can be formulated as follows:

$$V_{\text{crosstalk}_{ij}} = \frac{R_{\text{driver}_j} C_{\text{coupling}_{ij}} + R_{\text{wire}_j} C_{\text{coupling}_{ij}}}{R_{\text{driver}_j} C_{\text{total}_j} + R_{\text{wire}_j} C_{\text{total}_j} + R_{\text{driver}_i} C_{\text{total}_i} + R_{\text{wire}_i} C_{\text{total}_i}} \quad (6)$$

where R_{driver_i} and R_{driver_j} is the driver resistance of net i and j , respectively. R_{wire_i} and R_{wire_j} are the wire resistances. $C_{\text{coupling}_{ij}}$ is the coupling capacitance between net i and j , C_{total_j} includes all interconnect ground capacitance, coupling

capacitance and load capacitance of net j . The definition of C_{total_i} is similar.

Our metric can be used as an upper bound for real peak noise. To testify we carried out a simple noise analysis and meanwhile compared our metric with another two widely used noise estimation metrics (denoted as *Metric A* and *Metric B*). They are given as follows:

Metric A:

$$V_{crosstalk_ij} = \frac{C_{coupling_ij}}{C_{total_i} + C_{total_j}}$$

Metric B:

$$V_{crosstalk_ij} = \frac{R_{driver_j} C_{coupling_ij}}{R_{driver_i} C_{total_i} + R_{driver_j} C_{total_j}}$$

The results are shown in Fig 4. The curve at the bottom (magenta color, triangle mark) is the peak noise from SPICE simulation. Therefore it is a straight line with slope equal to 1. The green curve with “o” mark is the coupling ratio defined as $C_{coupling_ij} / C_{total_j}$, where net j is the victim and net i the aggressor. The blue curve with “*” mark is the noise estimation using metric A. The red curve with “+” mark is the noise estimation of metric B considering driver resistance. The black curve with “x” mark is the noise estimation of our metric considering driver resistance and wire resistance. All three metrics can be used as the upper bound of real peak noise, while it is clear that our metric lowers the bound and is also conservative.

Without knowing the detailed routing information, some general assumptions are made first:

Assumption 1 The coupling length (overlapping length between adjacent wires) of wire segments in GRG edge i is the length of the edge l_i .

Assumption 2 Each wire segment has crosstalk with at most two adjacent wires (wires above and below, or wires left and right) within a GRG edge.

Assumption 3 The spacing between adjacent wire segments is the average spacing within a GRG edge.

Assumption 4 The crosstalk between wire segments in different GRG edges can be ignored.

The purpose of assumption 1 is to make a conservative estimation on the crosstalk volume, illustrated in Fig 5. Assumption 2 and 3 assume that each wire segment within a GRG edge has equal status. Assumption 4 assumes coupling capacitance existing between wire segments in different GRG edges can be ignored because in practice, the length of a GRG edge is usually a few times of the height of a cell.

• Heuristics

The basic idea of the crosstalk control algorithm is based

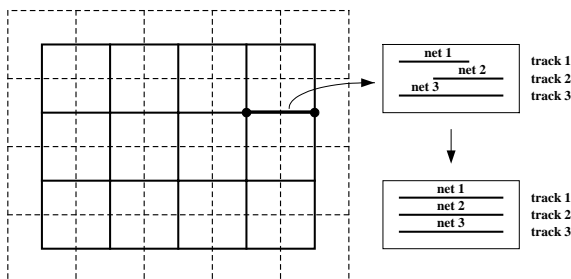


Fig. 5. An assumption in crosstalk control.

on rip up and iteration. According to formula (1), the total crosstalk on a net j can be given by:

$$Ct_j = \sum_{i=1, i \neq j}^{N_n} Ct_{ij} = \sum_{k=1}^{S_n} \sum_{i=1, i \neq j}^{N_n} Ct_{ijk} \quad (7)$$

where S_n is the total number of GRG edges that net j routed through, Ct_{ijk} is the value of crosstalk on segment k of net j from net i . As we assume that crosstalk Ct_j below a risk bound x_j will not affect the proper functioning of the circuit, the “overflow” crosstalk on net j and the total “overflow” crosstalk of solution S can be defined by the following formulae:

$$\tilde{Ct}_j = Ct_j - x_j \quad (8)$$

$$\tilde{Ct}(S) = \sum_{j=1}^{N_n} \tilde{Ct}_j = \sum_{j=1}^{N_n} (Ct_j - x_j) \quad (9)$$

Given a global routing solution S , the total overflow crosstalk and crosstalk on each net $Ct_j(S)$ can be calculated. If crosstalk violations are found, the nets having violations are ripped up. Since the rerouted nets will introduce extra crosstalk not only on itself but also on other nets, with which it shares GRG edges, the cost can be calculated by crosstalk increases of these two parts. The total overflow crosstalk introduced by rerouting of net j is formulated as follows:

$$\begin{aligned} \Delta \tilde{Ct} &= k_1 \Delta \tilde{Ct}_j + k_2 \sum_{i=1, i \neq j}^{N_n} \Delta \tilde{Ct}_i \\ &= k_1 \left(\sum_{s_i \in T'_j} \Delta \tilde{Ct}_j^{s_i} + \sum_{s_i \in T_j} \Delta \tilde{Ct}_j^{s_i} \right) + k_2 \sum_{i=1, i \neq j}^{N_n} \sum_{s_i \in T'_j \cup T_j} \Delta \tilde{Ct}_i^{s_i} \end{aligned} \quad (10)$$

where T_j and T'_j is the routing tree for net j before and after rerouting, respectively. $\Delta \tilde{Ct}_j^{s_i}$ is the difference of overflow crosstalk introduced on GRG edge s_i .

During crosstalk elimination, we also need to consider timing performance. One thing benefiting from our timing optimization algorithm is that the critical path has relatively low crosstalk due to coupling effects transference. In order to prevent the critical path from delay deterioration, the overflow crosstalk of a segment s_i on critical path introduced by rerouting of net j is given by the following expression:

$$\Delta \tilde{Ct}_j^{s_i} = \beta (Ct_j^{s_i} - Ct_j^{s_i}) \quad \beta > 1 \text{ if } (Ct_j^{s_i} - Ct_j^{s_i} > 0) \quad (11)$$

where $Ct_j^{s_i}$ and $Ct_j^{s_i}$ denote the crosstalk on net j before and after rerouting, respectively.

Each newly constructed routing tree will be recorded into the corresponding net. When a new solution is constructed, we compare it with all the historical solutions for minimal crosstalk cost. To expand the space of greedy search, the sequence of the nets having crosstalk violations will be randomized in rerouting to search for the global optimum. The heuristic algorithm is described in Fig 6.

Crosstalk_control_algorithm()
while (iteration-time < LIMIT), do
 for (each net_j), calculate $Ct_j(S)$;
 calculate total overflow crosstalk $\tilde{Ct}(S)$;
 find the set of crosstalk violated nets Vt ;
 generate a randomize subset Vs of Vt ;
 For (each net_j ∈ Vs)
 Find the most violated n segments;
 reroute segments $k_1 \sim k_n$ with minimal $\sum_{i=1}^n \Delta \tilde{Ct}_j^{s_{k_i}}$;

record the solution;
For each historical solution of net_i
calculate $\Delta \tilde{C}_t$;
determine routing tree T_j ;
reroute net_k yielding minimal $\Delta \tilde{C}_t$ with corresponding routing tree T_k ;

Fig. 6. The crosstalk control algorithm

An example of the procedure is given in Fig 7. The original solution is shown in (a), where the regions having crosstalk risk are identified by red color. After calculation of $\Delta \tilde{C}_t$, net_1 is chosen and rerouted. Consequently a wire segment is released from crosstalk risk on net_2 and net_5 , respectively, which is shown in (b). (c) and (d) illustrate the operations of steps followed.

Since the coupling length is conservatively estimated, the resulting total crosstalk value may appear to be relatively high. Nevertheless, the assumptions herein are general enough to reflect the extent of “crosstalk risk” for each net as the guide information for detailed routing. Thus, the crosstalk risk bound can be flexibly defined.

D. The Global Routing Algorithm Description

The core algorithm for optimization is a 3-step (step 3, 4, 5, in Fig 8) iteration algorithm based on rip-up and reroute method. Evaluation functions are used with respect to single rerouted tree and overall routability. In our optimization process, the evaluation function varies with changing of routing phases, by which we may take purposive control.

Optimization_algorithm()

1. Construct initial timing tree for all the nets.
2. Evaluate global congestion of GRG.

Evaluation functions are defined in table I and table II:

TABLE I

Single Routing Tree Extended Congestion Evaluation

Variables	Definition
$w_t = \sum_{e_i \in t} \tilde{w}_i$	Weighted Congestion of tree t \tilde{w}_i : Weighted Congestion of edge e_i
$E_t = \alpha w_t / \max_{x \in T_n}(w_x) + \beta l_t / \max_{y \in T_n}(l_y) + \gamma b_t / \max_{z \in T_n}(b_z)$	α, β, γ : non-negative constant, l_t : wire length of tree t , b_t : number of bends for tree t , T_n : Steiner trees ever solved for net n , E_t : evaluation for tree t

TABLE II

Overall Extended Congestion Evaluation

Variables	Definition
$C_{\max} = \max(\tilde{w}_i)$	Largest EC among all the GRG edges
$C_s = \sum \tilde{w}_i^2$	Sum of squared EC for congested GRG edges
$E_{cg}(\alpha) = \{e_i \tilde{w}_i > \alpha\}^*$	The set of congested GRG edges
$C_{overflow}(\alpha) = \sum_{e_i \in E_{cg}(\alpha)} (\tilde{w}_i - \alpha)$	The total overflows of GRG edges
$C_{so}(\alpha) = \sum_{e_i \in E_{cg}(\alpha)} \tilde{w}_i (\tilde{w}_i - \alpha)$	Weighted total overflow of GRG

* α is a constant, normally set to 1.0.

3. Determine the longest delay path P_r .
4. Simultaneous random optimization & Sequential random optimization.
 - a) for each segment S_j of net_i : calculate congestion & coupling weight,
 - b) for each S_j of net_i on P_r : recalculate weighted coupling of S_j ,
 - c) Simultaneous random optimization,

- c.1) select congested nets by extended congestion evaluation, reroute congested segments and evaluate new tree according to table I,
- c.2) after rerouting of all selected nets, refresh the weight of GRG,
- c.3) evaluate overall routability by function $(C_{\max}, C_{overflow}, C_s)$, record the best solution S ,
- c.4) if $E_{cg}(1.0) = \emptyset$, go to step 5,
- c.5) if iterations < M1, go to step 3, do iteration of 3-4c),
- d) Sequential random optimization,
 - d.1) select congested nets list and randomize the sequence,
 - d.2) reroute a congested net, refresh the weight of GRG serially,
 - d.3) repeat d.2) for all selected nets.
 - d.4) evaluate overall routability by function C_{so} , record the best solution S .
 - d.5) if $E_{cg}(1.0) = \emptyset$, go to step 5,
 - d.6) if iterations < M2, go to step 3, do iteration of 3-4d),
- 5) Call crosstalk control algorithm.

*If the constraints are not satisfied after iterations of step 5, do the outmost iteration of step 3-4-5 again.

Fig. 8. Pseudo-code of the optimization algorithm.

By simultaneous random optimization of step 4, we can approach some local optimum quickly but roughly. Sequential optimization then helps to find better route by accurate rerouting and weight-refreshing. The crosstalk control algorithm will be applied in case crosstalk violations are found.

V. Experimental Results

We have implemented the timing-driven global routing algorithm on a Sun Enterprise 450 in C language and tested it with three industrial circuits extracted from microprocessor design. They are under $0.13 \mu m$ technology and provided with corresponding look-up tables containing the gate timing arc information. Note that the routing resource constraints had all been satisfied in the following experimental results unless specified.

Two sets of experimental data are provided. The first set of data (given in Section A) is the comparison on timing optimization capability of two methods, which can be applied before crosstalk elimination. The resulting delay values are all actual delay with consideration of worst-case coupling capacitance. The second set of data demonstrates the effect of our crosstalk control algorithm, which is given in Section B.

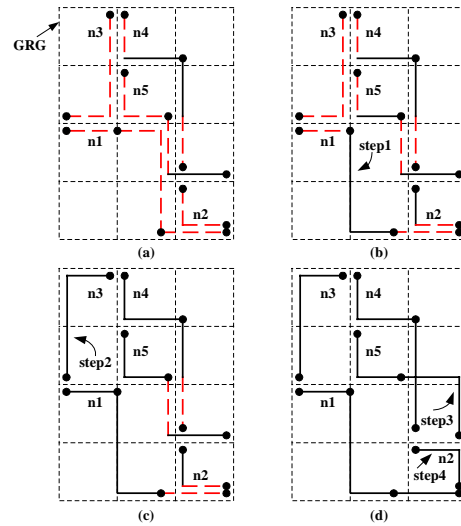


Fig. 7. An example showing the crosstalk control algorithm.

A. Comparison on Circuit Delay

Table III compares the experimental results with respect to two different global routing methods. In Method 1, we first apply our initial routing tree construction algorithm. Then, optimization is applied based on the initial solution but no coupling directed optimization is carried out. In Method 2, we utilize coupling as a heuristic in guiding timing optimization, after which the crosstalk control algorithm will be applied.

Row index of Table III is the circuit name and size of net list. Comparisons of delay performance and run time are shown in column 3 to 5, column 6 to 7 respectively. We can see from Table III that applying our coupling directed delay optimization, Method 2 has made an up to 12% improvement over the timing performance of Method 1 within very slightly increased run time.

B. Experimental Results on Crosstalk Control

In order to measure the effectiveness of the crosstalk control algorithm, we first run our global router for timing optimization merely, which we had discussed as Method 2 in Section A. The experimental results are compared with those obtained after applying crosstalk control algorithm. We give the comparison in Table IV. It is clear from Table IV that after applying our crosstalk control method, the total overflow crosstalk of the circuits have been suppressed with very slight deterioration on timing performance. Meanwhile the number of nets having violations has been successfully reduced.

VI. Conclusions and Future Work

Our study has shown that it is important to consider coupling effects into VLSI interconnect optimization for deep sub-micron designs. We adopt advanced delay estimation models to include the lateral coupling capacitance for delay calculation. We develop a timing-driven global routing algorithm for high performance circuit design. By utilizing coupling as a heuristic in guiding the optimization, substantial delay reduction has been achieved. In addition, effective crosstalk control algorithm helps to lead to an overall feasible solution for later layout phases.

Many problems in crosstalk elimination have shown to be NP-hard. We will consider integrated shielding, net ordering and power grid design during global routing phase as part of our future work.

References

[1] H. B. Bakoglu, "Circuits, Interconnections and Packaging for VLSI",

Addison-Wesley, 1990.

- [2] D. Sylvester, C. M. Hu, O. S. Nakagawa, S. Y. Oh, "Interconnect Scaling: Signal Integrity and Performance in Future High-speed CMOS Designs", *In: Proceedings of VLSI Symposium on Technology*, Honolulu, HI, USA, pp.42-43, 1998.
- [3] M. Rose, M. Wiesel, D. Kirkpatrick, N. Nettleton, "Dense, Performance Directed, Auto Place and Route", *In: Proceedings of IEEE CICC*, Rochester, NY, pp.11.1.1-11.1.4, 1988.
- [4] X. L. Hong, T. X. Xue, J. Huang, C. K. Cheng, E. S. Kuh, "TIGER: An Efficient Timing-Driven Global Router for Gate Array and Standard Cell Layout Design", *IEEE Trans. on CAD*, 16(11), pp.1323-1330, 1997.
- [5] J. Hu, S. S. Sapatnekar, "A Timing-constrained Algorithm for Simultaneous Global Routing of Multiple Nets", *In: Proceedings of IEEE/ACM ICCAD*, San Jose, CA, pp.99-103, 2000.
- [6] T. Jing *et al*, "A Novel and Efficient Timing-Driven Global Router for Standard Cell Layout Design Based on Critical Network Concept", *In: Proceedings of IEEE ISCAS*, Scottsdale, USA, pp.1165-1168, 2002.
- [7] W. C. Elmore, "The Transient Response of Lumped Linear Networks with Particular Regard to Wideband Amplifiers", *J. of Applied Physics*, 19(1): pp.55-59, 1948.
- [8] T. Sakurai, "Approximation of Wiring Delay in MOSFET LSI", *IEEE J. of SSC*, 18(4): pp.418-426, 1983.
- [9] H. Zhou and D. F. Wong, "An Optimal Algorithm for River Routing with Crosstalk Constraints", *In: Proceedings of IEEE/ACM ICCAD*, San Jose, CA, pp. 310-315, 1996.
- [10] S. S. Sapatnekar, "A Timing Model Incorporating the Effect of Crosstalk on Delay and its Application to Optimal Channel Routing", *IEEE Trans. on CAD*, Vol. 19, No.5, pp.550-559, 2000.
- [11] K. C. Hsu, Y. C. Lin, P. X. Chiu, and T. M. Hsieh, "Minimum Crosstalk Channel Routing with Dogleg", *In: Proceedings of IEEE ISCAS*, Geneva, pp.73-76, 2000.
- [12] H. Zhou, and D. F. Wong, "Global Routing with Crosstalk Constraints", *IEEE Trans. on CAD*, Vol. 18, No. 11, pp. 1683-1688, 1999.
- [13] James D. Z. Ma and Lei He, "Towards Global Routing with RLC Crosstalk Constraints", *In: Proceedings of IEEE/ACM DAC*, New Orleans, Louisiana, pp. 669-672, 2002.
- [14] T. Xue, E. S. Kuh, and D. Wang, "Post global routing crosstalk risk stimation and reduction", *In: Proceedings of IEEE/ACM ICCAD*, pp. 302-309, 1996.
- [15] X. D. Yang, A Reduced Order Modeling and Analysis for VLSI RLC Interconnect, Ph.D. thesis, CSE Dept., UCSD, USA, 2000.
- [16] A. Odabasioglu, M. Celik, L. T. Pileggi, "PRIMA: Passive reduced-order interconnect macromodeling algorithm", *In: Proceedings of IEEE/ACM ICCAD*, San Jose, CA, USA, pp.58-65, 1997.
- [17] J. Y. Xu, X. L. Hong, T. Jing, Y. C. Cai, J. Gu, "A Novel Timing-Driven Global Routing Algorithm Considering Coupling Effects for High Performance Circuit Design", *IEICE Trans. Fundamentals*, Vol E86-A, No.12 December 2003.
- [18] J. Y. Xu, X. L. Hong, *et al* "An Efficient Hierarchical Timing-Driven Steiner Tree Algorithm for Global Routing", *In IEEE/ACM ASP-DAC*, Bangalore, India, pp.473-478, 2002.
- [19] L. He and K. M. Lepak, "Simultaneous shielding insertion and net ordering for capacitive and inductive coupling minimization", *In: IEEE/ACM ISPD*, pp.56-61, 2000.
- [20] T. Sakurai and K. Tamaru, "Simple formulas for two and three dimensional capacitance", *IEEE Trans. on Electron Devices*, pp. 183-185, 1983..

TABLE III. Comparison on Total Delay and Run Time

Testcase	Number of Nets	Method 1 Delay(ns)	Method 2 Delay(ns)	% Delay Optimized	Method 1 CPU Time(s)	Method 2 CPU Time(s)
GDC	1568	5.4832	4.8711	11.16%	41	49
BIU	943	8.0667	7.4398	7.77%	14	16
PAT	4674	19.6060	18.0656	7.86%	45	51

TABLE IV. Experimental Results of Crosstalk Control

Testcase	Number of Nets	GR without CC			GR with CC		
		#Crosstalk Overflow	#Nets violations	Circuit Delay(ns)	#Crosstalk Overflow	#Nets violations	Circuit Delay(ns)
GDC	1568	14.54	194	4.87	0.23	17	4.96
BIU	943	4.91	58	7.44	0.42	7	7.61
PAT	4674	27.02	309	18.07	0.60	21	18.22