An Efficient Sequential Quadratic Programming Formulation of Optimal Wire Spacing for Cross-Talk Noise Avoidance Routing

Paul B. Morton
Wayne Dai

UCSC-CRL-99-05
March 10, 1999

Jack Baskin School of Engineering
University of California, Santa Cruz
Santa Cruz, CA 95064 USA

ABSTRACT

In this paper we propose a new, and effective, approach to cross-talk noise avoidance routing. In our new approach we attack the cross-talk noise problem immediately following topological routing, which is the point in the routing process that gives the best trade off between the ability to detect cross-talk noise problems and the ability to correct the problems. We formulate the heart of this new approach as a convex, nonlinear, mathematical programming problem which determines an optimal set of wire spacings under cross-talk noise constraints. This new mathematical programming formulation is based on a detailed knowledge of the underlying cross-talk noise mechanisms and accounts for coupling capacitance, interconnect resistance, and aggressor net signal rise time on nets with arbitrarily complex tree topologies. Finally, by slightly restricting this programming problem we formulate it as a linearly constrained, convex, nonlinear, mathematical program which can be quickly and efficiently solved using sequential quadratic programming.

Keywords: cross talk, constrained routing, wire spacing, noise margin, sequential quadratic programming.

1 Introduction

One of the fundamental advantages of digital systems are their ability to reject noise through self restoring logic, that is, the output signal from a logic stage is closer to the ideal logic levels than the input signal. This is the main reason why the vast majority of electronic computing systems are digital.

Integrated circuit technology created a way to inexpensively mass produce very reliable and sophisticated digital electronic systems. Integrated circuit technology also brought with it the ability to make continuous and predictable incremental improvements in component density, speed and power consumption. This is accomplished by following a set of scaling rules which systematically reduced feature sizes and power supply levels while giving a high level of assurance that the shrunken devices will still operate correctly. Further density improvements were created through the use of novel gate designs, such as precharged logic. However, all of these techniques to improve density, speed, and power consumption were also systematically reducing the noise rejection ability of the integrated circuit technology. Recently these trends have elevated the process of noise management to the same level of importance as timing and power management. While there are several noise mechanisms that are being aggravated by these trends, this paper will focus on the cross-talk noise problem, where cross-talk noise is an unwanted signal induced on a net by a signal propagating along an adjacent net.

The cross-talk noise problem, unlike timing, power, and electromigration management problems has the added difficulty that in order to effectively detect the problem, we need a fairly detailed knowledge of the interaction between adjacent nets. Unfortunately, in a traditional routing technology, we are presented with two equally poor options for addressing this problem. We can either wait until late in the routing process, where we have an accurate picture of where the cross talk noise problems are, but have little flexibility in the routing to fix them, or we can try and fix the problems early in the routing process where we can easily modify the routing but cannot accurately predict the cross-talk noise problems.

2 Previous Work on Cross-Talk Noise Aware Routing

Since cross-talk noise is dependent on routing topology, it is most practical to deal with it during the routing phase of physical design. VLSI routing can generally be divided in to three phases. Global routing, detailed routing, and post route optimization.

During global routing, the very large chip routing problem is transformed into a set of smaller routing problems by restricting the topology of each net to pass through specific regions of the chip. Global routing cross-talk noise avoidance strategies have been proposed by Xue et. al. [1] and Zhou and Wong [2], however, since detailed net adjacency information is not available during this phase of the routing their strategies rely on some rather tenuous assumptions about which nets might be adjacent to each other.

During a traditional detailed routing, the rough global routes for each net are sequentially transformed into exact routing geometries. Strategies for cross-talk noise avoidance during a traditional detailed routing process have been proposed by Chen and Wong [3, 4], Miyoshi et. al. [5], and Zhou and Wong [6]. However, these strategies suffer from two problems. First, at the beginning of the detailed routing process, cross-talk noise driven routing decisions are based on very inaccurate net adjacency information, leading to questionable routing decisions. Second, near the end of the routing process very accurate net adjacency information has become available, but it can not be used effectively since the routing has been specified in too much detail to be easily adjusted.

The global and detailed routing phases strive to generate routes which meet certain constraints, such as timing, power consumption, and electromigration resistance. If all the constraints that have been placed on the routing have not been met by the routing produced by the global and detail router, then post route optimizations are applied to the routing to try and repair the short comings of the routing. Post route optimization strategies for cross-talk noise repair have been proposed by Gao and Liu [7, 8], Jhang et. al. [9], and Onozawa et. al. [10, 11]. In general each of these strategies suffers from the fact that the routing has been specified in too much detail and is now very difficult to adjust. Additionally, the strategies proposed in [7], [8], and [9] are only applicable to a style of routing which is no longer popular in state of the art VLSI designs, while the strategy outlined in [10] and [11] has been constructed based on the solution to an optimization problem that has been formulated with no knowledge of the underlying mechanisms which govern cross-talk noise.

3 Topological Routing Based Cross-Talk Noise Avoidance

In our new approach to cross-talk noise avoidance routing we take advantage of the fact that a detailed geometric routing can be broken into two components, a topological routing, and a set of branch spacings. Given a topological routing, such as that shown in Fig. 1(a), and a valid set of branch spacings, we can quickly and easily construct a geometric routing [12, 13], such as that shown in Fig. 1(b).

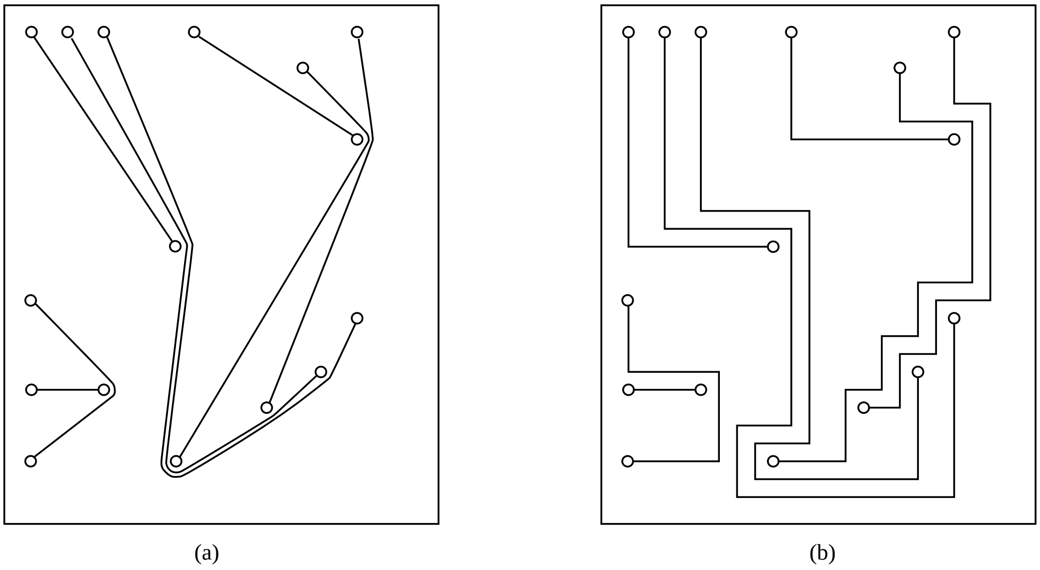


Figure 1: A topological routing (a), and its corresponding geometric routing (b).

There are two key ideas embedded in this view of a detailed routing. The first is that changes to the set of spacing variables can be made at virtually no cost. The only thing that we need to be careful about is that the changes don't produce an unroutable design, however, as we will see in sections 5 and 6 this restriction does not pose a serious problem. The second key idea is that all the information needed to make accurate cross-talk noise estimates can be extracted from the topological routing, since it is the routing topology which determines which nets are capacitively coupled.

From these ideas our new approach to cross-talk noise avoidance routing is:

1. **Identify cross-talk noise violations by estimating them from the topological routing.**
2. **Determine a set of branch spacing values which eliminate the violations.**
3. **Construct a geometric routing from the topological routing and the spacing values.**

It should be noted that the routing topology used in our new approach can either be extracted from an existing geometric routing, or be constructed directly form a net list.

This new approach to cross-talk noise avoidance routing can be based on an existing, gridless, topological routing system, called SURF [14]. This routing system constructs a detailed geometric routing by first constructing a topological routing and then transforming it into a geometric routing.

Ideally we would like to simultaneously determine the set of branch spacings for all nets with cross-talk noise violations. Unfortunately, given the enormous number of nets on todays chip designs, this is clearly an impractical approach. Because of this, we propose a greedy approach, that is, we order the nets with cross-talk noise violations according to the "severity" of their violations, then, determine a set of spacing values for each net in the list beginning with the net with the most severe violation. In order that we have adequate routing resources available to those nets at the end of the list, we need to determine each net's set of spacings such that they eliminate any cross-talk violations while using a minimum amount of routing resources. From this we see that the heart of our new noise avoidance strategy is an optimization problem which is the determination of a victim net's optimal wire spacing under cross-talk constraints. For brevity we will refer to this as "The Optimal Spacing Problem".

4 Cross-talk Noise Estimation Techniques

In order to formulate the optimal spacing problem we need to be able to estimate the cross-talk noise seen by any net in the design. We would like for this estimate to be as accurate as possible, however it must also be an estimate that can be used to compute the cross-talk noise on thousands of nets in a reasonable amount of time. Of equal importance, this estimate must be easily applicable to nets with complex topologies. Further, we would like for this estimate to be a reasonable upper bound on the cross-talk noise in order to be able to guarantee that any design changes that are based on this estimate will indeed produce a new design with fewer cross-talk problems.

The simplest, oldest, most widely used, and least accurate estimation technique is to use the total capacitive coupling between an aggressor and a victim net as an estimate of the peak cross-talk noise. The main advantage of this estimate is its simplicity. This estimate reduces the calculation of cross-talk noise to determining the ratio between coupling length and net spacing, which can be computed quickly on nets with complex topologies. The main disadvantage of the capacitive coupling estimate is also its simplicity, that is, it ignores interconnect resistance effects, the effect of the aggressor nets signal rise time, and the moderating effects of the victim nets ground capacitance. Of particular concern is the assumption that the victim net has negligible interconnect resistance given that we know the interconnect resistance is increasing as VLSI technology is scaled down. This assumption creates an estimate where the injected noise signal has the same effect on all sink pins of the victim net regardless of the distance of each sink pin form the point of injection. One important consequence of this simplification is that each net has to be characterized by a single noise margin, in particular, we are forced to choose the most conservative noise margin. A second important consequence of this simplification is that it does not account for the distance between the point of injection and the nets source pin. Specifically, the farther the point of injection is from the source pin, the harder it is for the source gate to "absorb" the noise signal, due to the larger resistance between the point of injection and the source pin.

By contrast, Sakurai's estimate [15] represents one of the most detailed cross-talk noise estimates. This estimate is derived by solving the differential equation which govern the behavior of two capacitively coupled RC lines. While this approach produces a very accurate estimate, it is very difficult to extend this technique to nets with complex topologies.

Another significant cross-talk noise estimation technique is the one proposed by Vittal and Marek-Sadowska [16]. In this technique expressions for peak cross-talk noise voltage and cross-talk noise pulse width are derived for a pair of capacitively coupled two pin nets. This technique takes into account coupling capacitance, driver strength, and ground capacitance, however, again, it is very difficult to extend this technique to nets with complex topologies.

One of the most obvious methods for estimating cross-talk noise would be to extract an RC network form a given layout and then use this extracted model in a simulation to determine the exact effects of cross-talk noise. While in theory this method looks promising, in practice this method would be far to slow.

Recently, Devgan [17] proposed a cross-talk noise estimate which provides an upper bound on peak cross-talk noise and is similar in form to the Elmore delay estimate [18, 19]. The victim and aggressor nets are modeled as capacitively coupled distributed RC networks, as shown in Fig. 2. The aggressor net's source gate is modeled as a saturated ramp voltage source in serious with a resistor. This source gate model is an improvement over the more traditional gate model, which uses a step voltage source in series with a resistor, due to its ability to more accurately model the shape of the wave form seen at the gate's output [20].

The peak cross-talk noise voltage on any node of the victim net can be computed as

$$V_n = V_{Par(n)} + R_n \sum_{\substack{i \in Des(n) \\ j \in Ag(i)}} C_{ij} \dot{U}_j \tag{1}$$

where

- $Des(n)$ is the set of victim net nodes that are descendants of node n.
- $Par(n)$ is the parent node of node n.
- $Ag(i)$ is the set of aggressor nets that are coupled into node i of the victim net.
- V_n is the maximum cross-talk noise voltage, induced on node n, by all down stream aggressor nets.
- R_n is the resistance connecting node n to node $Par(n)$.
- $\dot{U}_j$ is the slope of the voltage source driving aggressor net j.
- C_{ij} is the coupling capacitance between aggressor net j and the segment connecting node i to node $Par(i)$.

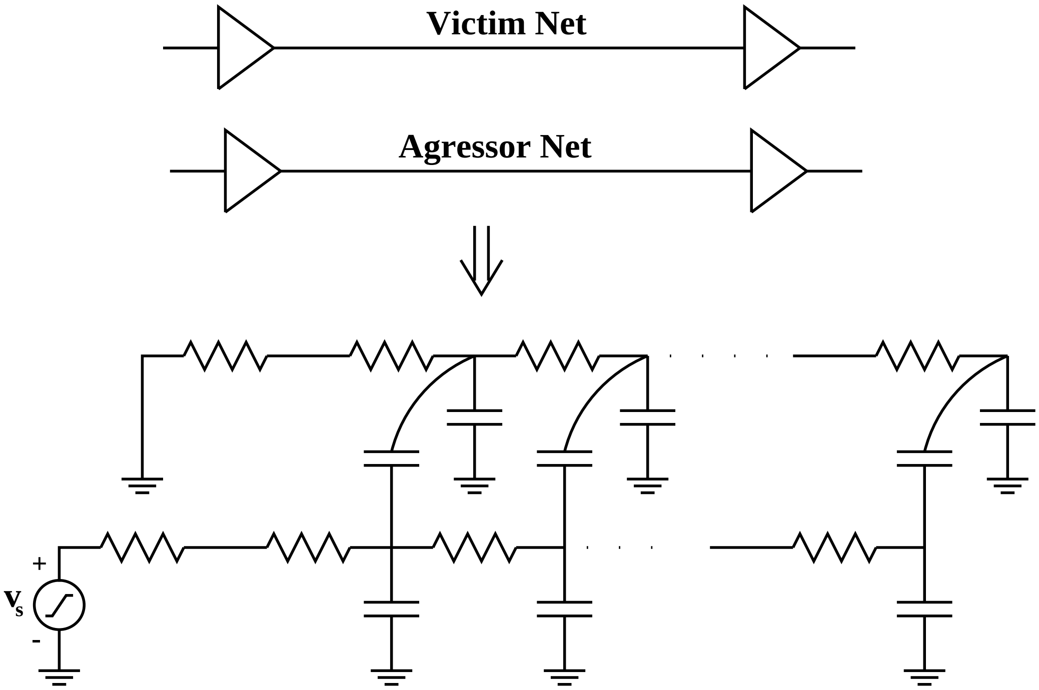


Figure 2: RC model for a coupled victim and aggressor net.

The main strengths of Devgan's estimate are its ability to include coupling capacitance, interconnect resistance, and aggressor net rise time in a closed form expression that can be used to quickly and easily analyze networks with complex topologies.

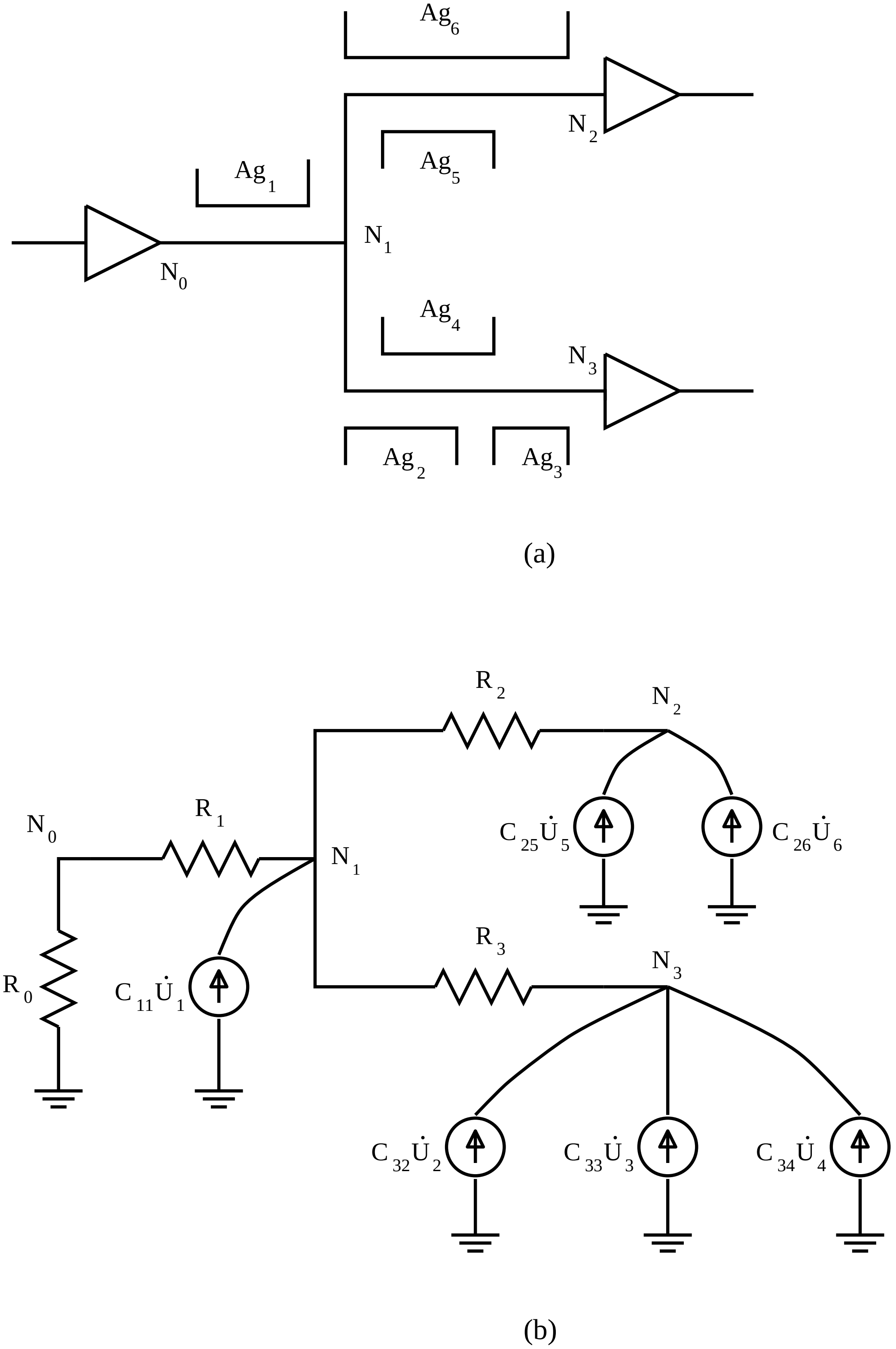
The key assumption behind Devgan's estimate is that the saturated ramp voltage source of each aggressor net can be approximated by an infinite ramp with the same slope. Applying and infinite ramp to the aggressor net guarantees that each of the victim net's ground capacitances will charge up to it's maximum steady state voltage. This guarantees that the estimate produces an upper bound on the cross-talk noise voltage at each node of the victim net.

To illustrate the use of this model, consider the routing depicted in Fig. 3(a), containing four nodes, N_0 through N_3, and six aggressor nets, Ag_1 through Ag_6. This can be reduced to the network shown in Fig. 3(b), where R_0 represents the victim net's source resistance and it is assumed that all coupling capacitances are lumped at the nearest down stream Steiner point. Using (1) we see that the noise voltage for each node can be written as:

$$\begin{aligned}
V_0 &= R_0\left(C_{11}\dot{U}_1 + C_{25}\dot{U}_5 + C_{26}\dot{U}_6 + C_{32}\dot{U}_2 + C_{33}\dot{U}_3 + C_{34}\dot{U}_4\right) \\
V_1 &= R_1\left(C_{11}\dot{U}_1 + C_{25}\dot{U}_5 + C_{26}\dot{U}_6 + C_{32}\dot{U}_2 + C_{33}\dot{U}_3 + C_{34}\dot{U}_4\right) + V_0 \\
V_2 &= R_2\left(C_{25}\dot{U}_5 + C_{26}\dot{U}_6\right) + V_1 \\
V_3 &= R_3\left(C_{32}\dot{U}_2 + C_{33}\dot{U}_3 + C_{34}\dot{U}_4\right) + V_1
\end{aligned}$$

5 Formulating the Optimal Spacing Problem

Our formulation of this problem will assume we are starting with a routable topological routing from which we can determine routing resource availability and accurately estimate branch length and net adjacency information. The formulation will be based on Devgan's



(a)

(b)

Figure 3: Victim net with six adjacent aggressor nets.

cross-talk noise estimate. Its ability to incorporate a detailed knowledge of the cross-talk noise mechanisms on arbitrary net topologies will allow us to efficiently allocate the available routing resources to those points along the victim net where the resources will have the greatest impact on the net's cross-talk noise problems.

Since Devgan's estimate assumes that all aggressor net voltage sources are ramps, we have

$$\dot{U}_j = 0.8\,\frac{V_{DD}}{t_j} \tag{2}$$

where V_{DD} is the supply voltage and t_j is the 10% to 90% rise time of the voltage source driving aggressor net j. Further, we have that

$$C_{ij} = C_T\,\frac{L_{ij}}{S_{ij}} \tag{3}$$

and

$$R_n = R_T\,L_n \tag{4}$$

where

- L_{ij} is the estimated length of the adjacency between aggressor net j and the branch connecting node i to node $Par(i)$.
- S_{ij} is the distance separating the adjacent branches.
- C_T is a proportionality constant determined from the technology.
- L_n is the estimated length of the branch connecting node n to $Par(n)$.
- R_T is a proportionality constant determined from the technology.

Making the conservative assumption that all capacitive coupling for a branch of the routing is lumped at its down stream node, as was done in the example shown in Fig. 3, and substituting (2), (3), and (4) into (1) we have

$$V_n = V_{Par(n)} + 0.8\,V_{DD}\,C_T\,R_T\,L_n \sum_{\substack{i \in Des(n)\\ j \in Ag(i)}} \frac{L_{ij}}{S_{ij}\,t_j} \tag{5}$$

The routing resources consumed by a net can be measured by the spacing area the net consumes

$$A = \sum_{\substack{i \in N\\ j \in Ag(i)}} L_{ij}\,S_{ij} \tag{6}$$

where N is the set of all victim net nodes.

Using (5) and (6) we can formulate the optimal spacing problem as the following constrained minimization problem:

$$\min_{\{S_{ij}\ \mid\ \forall\ i \in N, j \in Ag(i)\}} \Big\{ \sum_{\substack{i \in N\\ j \in Ag(i)}} L_{ij}\,S_{ij} \Big\} \tag{7}$$

subject to

$$\begin{aligned} S_{ij} &\geq S_{min} & \forall\, i \in N,\ j \in Ag(i) && (8)\\ f_{ij}(\overline{S}) &\leq S_{ij-max} & \forall\, i \in N,\ j \in Ag(i) && (9)\\ V_n &\leq M_n & \forall\, n \in Pins(N) && (10) \end{aligned}$$

Where

- (8) are lower bounds imposed by the technology.
- (9) are "upper bounds" imposed by the routing topology.
- (10) are the cross-talk noise constraints.
- S_{min} is the minimum allowable spacing between adjacent wires.
- $Pins(N)$ is the subset of victim net nodes connected to sink pins.
- M_n is the noise margin for node n of the victim net.
- $\overline{S}$ is a vector of all spacing variables.

In general, the upper bound on each S_{ij} is dependent on other spacing variables. In order to accommodate this behavior, the upper bound on each S_{ij} is written, in (9), as a function, f_{ij}, of all the spacing variables. In general, each S_{ij-max} is a positive constant and the functions f_{ij} are linear functions with positive coefficients, as we will see in section 6 The exact form of each f_{ij} is determined by the routing topology and the value of the coefficients are determined from the state of the routing just prior to the optimization process.

6 Formulating the Upper Bound Constraints

To determine the set of upper bound constraint equations represented by (9) we take advantage of Maley's routability theorem [12], which states that a topological routing is routable if and only if all shortest straight cuts between all pairs of visible features are safe. Feature, in this theorem, is used to mean any object through which a branch cannot be routed, excluding other branches. A straight cut, in this theorem, is a line segment that starts on one feature and ends on another. While the shortest straight cut between two visible features is the shortest straight line segment that starts on one feature, ends on another, and does not intersect any other features. From now on we will refer to the shortest straight line cut between two visible features as a cut.

In order to determine if a cut is safe we need to determine if there is enough routing space across the cut (between the two features) to accommodate all of the branches that need to cross the cut. The capacity of a cut is a measure of the routing space across a cut. For a Manhattan routing style the capacity of a cut, c, whose end points are (x_1, y_1) and (x_2, y_2) is defined to be

$$capacity(c) = max\{|x_1 - x_2|, |y_1 - y_2|\} \tag{11}$$

The flow of a cut, $flow(c)$, is a measure of the amount of routing resources needed to route all the branches that need to cross the cut. The flow must include the width of each branch, as well as the spacing needed to separate each pair of adjacent branches, and any spacing needed to separate the branches from the two features which define the end points of the cut.

Using the definitions of flow and capacity we can define the slack of a cut, c, to be

$$slack(c) = capacity(c) - flow(c) \tag{12}$$

and from this we say that a cut, c, is safe if and only if $slack(c) \geq 0$. From this we see that (9) can be written as

$$flow(c) \leq capacity(c) \quad \forall\, c \in C \tag{13}$$

where C is the set of all cuts that the victim net crosses.

7 Solving the Optimal Spacing Problem

Because (10) is a nonlinear function of S_{ij}, the optimal spacing problem, (7) through (10), forms a nonlinear programming problem. However, (10) is also convex over the region defined by $S_{ij} \geq S_{min}$, and thus (7) through (10) forms a convex programming problem. While this mathematical programming problem could be directly solved using a generalized sequential quadratic programming (SQP) techniques [21], similar to that used by Menezes et. al. in [22], this technique is significantly complicated by the nonlinear constraints. However, by slightly restricting the optimal spacing problem we can transform it into a linearly constrained, convex, nonlinear programming problem which can be quickly and efficiently solved using SQP.

We will restrict the problem by determining a single fixed budget for each S_{ij}. This can be done by evenly dividing the *slack* for each *cut* the victim net crosses, among all the spacing variables involved with that cut. This process determines a set of potential budgets for each S_{ij}. For each S_{ij} we then select the most restrictive budget to form the upper bound constraints on S_{ij}

$$S_{ij} \leq S_{ij-budgit} \quad \forall\, i \in N,\; j \in Ag(i) \tag{14}$$

We can now convert this restricted version of the problem to a convex program with linear constraints by substituting

$$q_{ij} = \frac{1}{S_{ij}} \tag{15}$$

Using this substitution we see that the objective function (7) becomes

$$\sum_{\substack{i \in N \\ j \in Ag(i)}} \frac{L_{ij}}{q_{ij}} \tag{16}$$

which is a convex function of q_{ij}. Substituting (15) into (5) gives us

$$V_n = V_{Par(n)} + 0.8\, V_{DD}\, C_T\, R_T\, L_n \sum_{\substack{i \in Des(n) \\ j \in Ag(i)}} q_{ij} \frac{L_{ij}}{t_j} \tag{17}$$

which is now a linear equation in q_{ij}.

From this restriction and substitution we now have the following linearly constrained, convex, nonlinear, mathematical program:

$$\min_{\{q_{ij} \,\mid\, \forall\, i \in N, j \in Ag(i)\}} \Big\{ \sum_{\substack{i \in N \\ j \in Ag(i)}} \frac{L_{ij}}{q_{ij}} \Big\} \tag{18}$$

subject to

$$q_{ij} \leq \frac{1}{S_{min}} \qquad \forall\, i \in N,\; j \in Ag(i) \tag{19}$$

$$q_{ij} \geq \frac{1}{S_{ij-budgit}} \qquad \forall\, i \in N,\; j \in Ag(i) \tag{20}$$

$$V_n \leq M_n \qquad \forall\, n \in Pins(N) \tag{21}$$

Lemma 1: *By moving the nonlinearity from the set of constraint equations to the objective function, the optimal spacing problem becomes a linearly constrained, convex, nonlinear, mathematical program that can be efficiently solved using sequential quadratic programming.*

An SQP algorithm converts a nonlinear programming problem into a sequence of quadratic programming (QP) subproblems. In our case, the objective function of each QP subproblem is constructed by approximating the nonlinear objective function with a quadratic approximation about the solution to the previous QP subproblem, while the constraints are those of the original optimization problem. The SQP algorithm terminates when a convergence criterion is met.

Our implementation is constructed in MATLAB, and utilizes MATLAB's QP solver. The algorithm terminates when the change in the objective function in (18), between two successive iterations of the SQP algorithm, is smaller than 0.01%. The initial feasible point is chosen to be

$$q_{ij} = \frac{1}{S_{ij-budgit}} \quad \forall\, i \in N,\ j \in Ag(i) \tag{22}$$

This point was chosen since it is guaranteed to be in the feasible region if a feasible region exists. Additionally, by checking this point to see if it is feasible, we can quickly and easily determine if there is no solution to a programming problem.

It should be noted that for the purposes of this paper we chose to use MATLAB's general purpose QP solver, however, since the objective function in (18) is separable, and thus has a diagonal Hessian, and since the vast majority of the constraint equations are of the form $q_{ij} \leq constant$ (19) or $q_{ij} \geq constant$ (20), we can extend the technique proposed by Chu and Wong in [23] to implement a more efficient QP solver for our problem.

8 Results

We tested our SQP algorithm using MATLAB 5.2 running on a Sun Enterprise 450 (300 MHz Ultra SPARC-II CPU) with 1 GB of memory under SunOS 5.6. We tested the algorithm on 100, nontrivial, randomly generated nets. By nontrivial we mean that the constraint set formed by (19) through (21) was not empty, and the point

$$q_{ij} = \frac{1}{S_{min}} \quad \forall\, i \in N,\ j \in Ag(i) \tag{23}$$

was not in the feasible region (note that if (23) is in the feasible region then it must be the optimal solution).

The technology parameters used in our experiments are based on the 0.18 μm technology specified in the SIA road map [24]. Specifically, the minimum spacing between wires is 0.33 μm. The wire resistivity is 0.291 $\Omega/\mu m$. The capacitive coupling between adjacent wires separated by the minimum wire spacing is 0.745 $fF/\mu m$. The supply voltage is $1.5V$. The noise margin for each sink pin is randomly selected between $0.25V$ and $0.5V$. The aggressor net rise times are randomly selected between 20 pS and 500 pS. The source gates output resistance is randomly selected between 20 Ω and 400 Ω. The budget for each spacing variable, S_{ij}, is randomly selected between 0.385 μm and 3.08 μm.

The lengths of the nets ranged from 0.165 mm to 2 mm. These nets can be further characterized by the graphs in Fig. 4 and Fig. 5 which show the number of sink pins and the number of adjacent aggressor nets for a given net length, respectively.

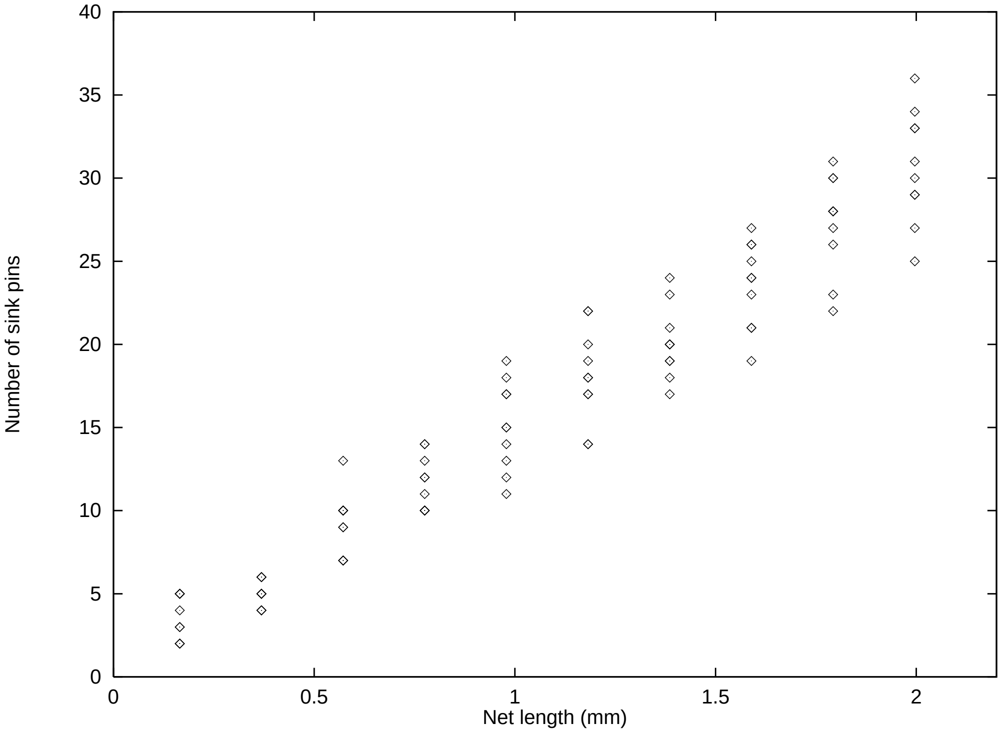


Figure 4: Number of sink pins.

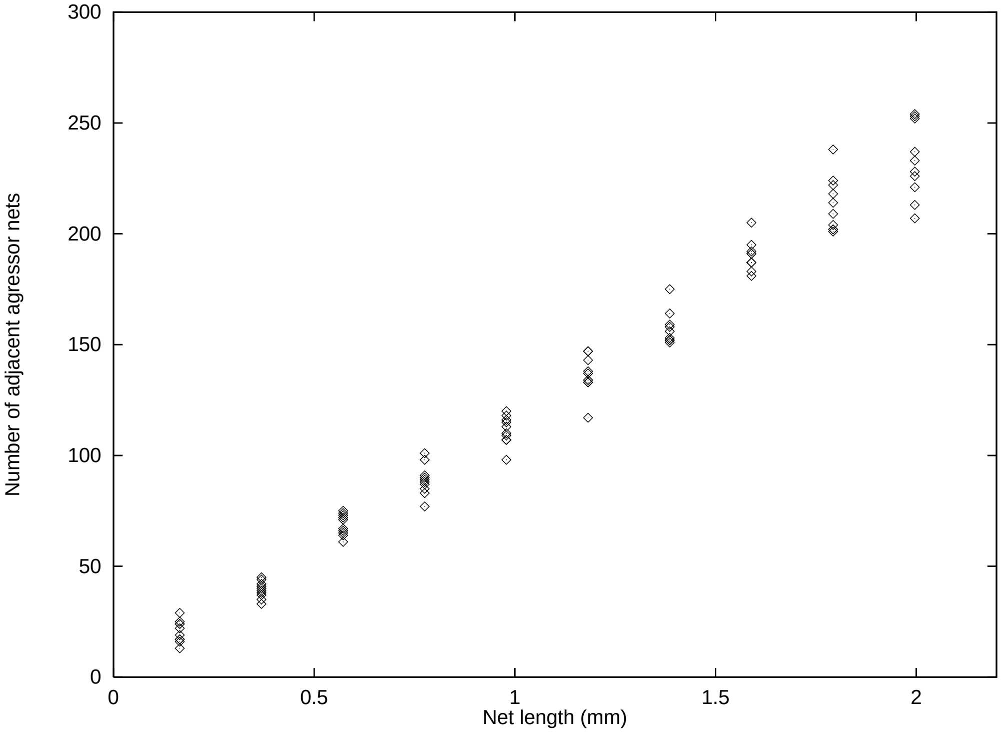


Figure 5: Number of adjacent aggressor nets.

Fig. 6 show the CPU time needed to compute the set of optimal spacings for each experimental net. From this we can see that an optimal set of spacings could be calculated in under 650 *sec* in the worst case. From Fig. 7, which is a plot of the *log* of the net length vs the *log* of the CPU time, we see that the time to compute the optimal set of spacings exhibits a roughly cubic, $O(n^{3.33})$, dependency on the length of the net. It should be noted that data for the shortest nets were disregarded for the purposes of determining the algorithm's computational complexity since the CPU time measurements for these nets was smaller than the margin of error for CPU time measurements. Finally, we found that in all cases that our SQP algorithm converged to a solution in fewer than eight iterations.

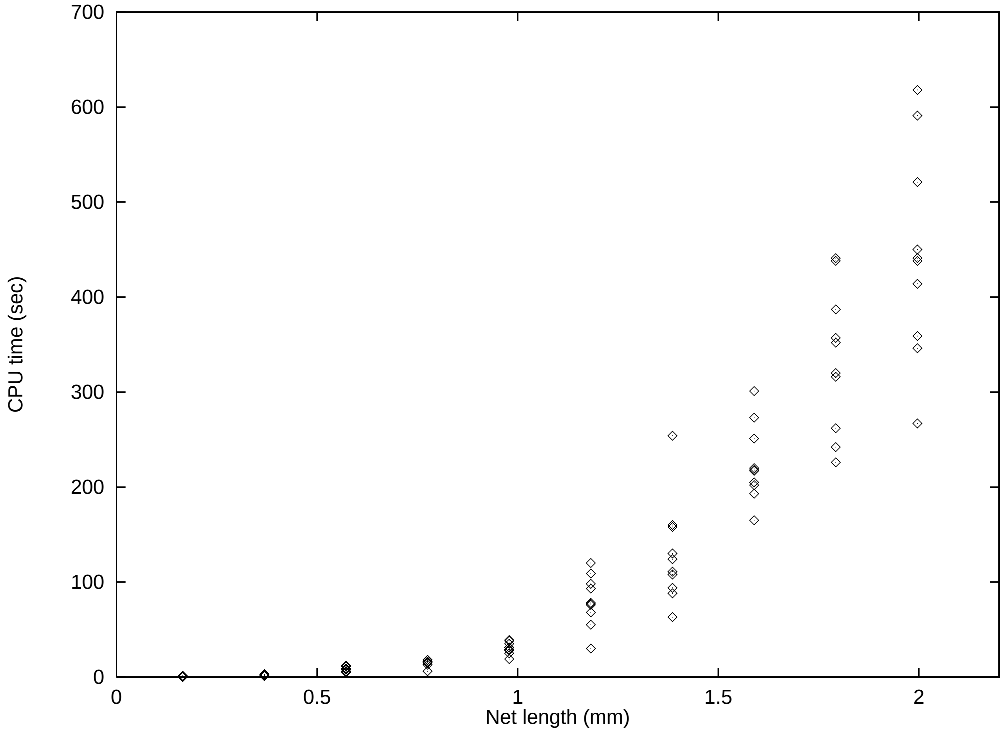


Figure 6: CPU time to compute spacing sets.

9 Conclusion

In this paper we have presented a new, and effective, cross-talk noise avoidance routing strategy. This strategy attacks the cross-talk noise problem immediately following topological routing, which is the point in the routing process which gives us the best trade off between the ability to quickly and accurately detect cross-talk noise violations and the ability to adjust the routing to correct the violations. The heart of this new strategy has been formulated as a convex, nonlinear, mathematical programming problem. This programming problem is a new formulation which determines an optimal set of wire spacings which eliminates cross-talk noise violations while using a minimum amount of routing resources. This formulation is based on a detailed knowledge of the underlying cross-talk noise mechanisms for nets with arbitrary routing topologies. In particular, this formulation accounts for the effects of coupling capacitance, interconnect resistance, and aggressor net signal rise time. Finally, we have shown that through a slight restriction we can transform

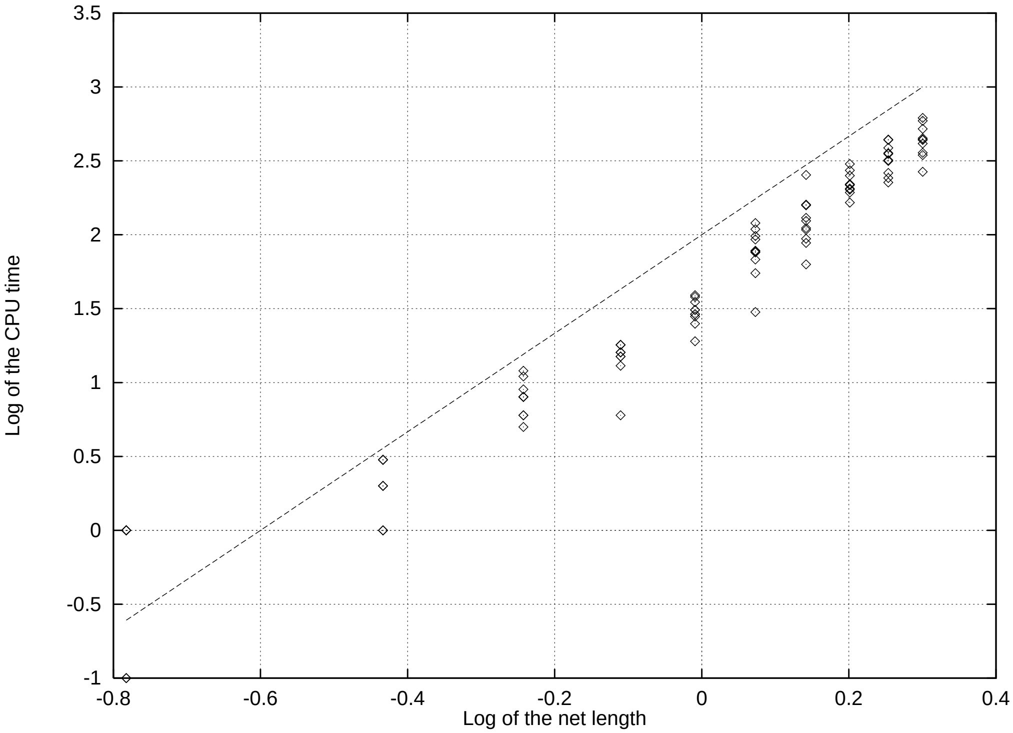


Figure 7: Log-log plot of CPU time.

the optimal wiring spacing problem into a linearly constrained, convex, nonlinear, mathematical programming problem which can be quickly and efficiently solved using sequential quadratic programming. Our experiments have shown that the time to compute an optimal set of spacings exhibits a roughly cubic, $O(n^{3.33})$, dependency on net length.

10 Acknowledgment

This work was supported in part by DARPA under contract number DAAL01-96-K-0113.

References

[1] T. Xue, E. Kuh, and D. Wang, "Post Global Routing Crosstalk Risk Estimation and Reduction," in *Proc. IEEE/ACM Int. Conf. on Computer Aided Design*, pp. 302–309, 1996.

[2] H. Zhou and D. Wong, "Global Routing with Crosstalk Constraints," in *Proc. Design Automation Conf.*, pp. 374–377, 1998.

[3] H. Chen and C. Wong, "63-Layer TCM Wiring with Three-Dimensional Crosstalk Constraints," *International Journal of High Speed Electronics and Systems*, vol. 6, no. 3, pp. 497–508, 1995.

[4] H. Chen and C. Wong, "Wiring and Crosstalk Avoidance in Multi-Chip Module Design," in *Proc. IEEE Custom Integrated Circuits Conference*, pp. 28.6.1–28.6.4, 1992.

[5] T. Miyoshi, S. Wakabayashi, T. Koide, and N. Yoshida, “An MCM Routing Algorithm Considering Crosstalk,” in *Proc. IEEE Int. Symposium on Circuits and Systems*, pp. 211–214, 1995.

[6] H. Zhou and D. Wong, “Crosstalk-Constrained Maze Routing Based on Lagrangian Relaxation,” in *Proc. IEEE Int. Conf. on Computer Design*, pp. 628–633, 1997.

[7] T. Gao and C. Liu, “Minimum Crosstalk Channel Routing,” *IEEE Trans. Computer-Aided Design*, vol. 15, no. 5, pp. 465–474, 1996.

[8] T. Gao and C. Liu, “Minimum Crosstalk Switchbox Routing,” in *Proc. IEEE/ACM Int. Conf. on Computer Aided Design*, pp. 610–615, 1994.

[9] K. Jhang, S. Ha, and C. Jhon, “A Segment Rearrangement Approach to Channel Routing Under the Crosstalk Constraints,” in *IEEE Asia-Pacific Conf. onf Circuits and Systems*, pp. 536–541, 1994.

[10] A. Onozawa, K. Chaudhary, and E. Kuh, “Performance Driven Spacing Algorithms using Attractive and Repulsive Constraints for Submicron LSI’s,” *IEEE Trans. Computer-Aided Design*, vol. 14, no. 6, pp. 707–719, 1995.

[11] K. Chaudhary, A. Onozawa, and E. Kuh, “A Spacing Algorithm for Performance Enhancement and Cross-talk Reducton,” in *Proc. IEEE/ACM Int. Conf. on Computer Aided Design*, pp. 697–702, 1993.

[12] F. Maley, *Single-Layer Wire Routing and Compaction.* Cambridge, Mass: The MIT Press, 1989.

[13] D. J. Staepelaere, “Geometric transformation for a rubber-band sketch,” Master’s thesis, University of California Santa Cruz, Septemeber 1991.

[14] D. J. Staepelaere, J. Jue, T. Dayan, and W. W.-M. Dai, “Surf: a rubber-band routing system for multichip modules,” in *Proc. IEEE Design and Test of Computers*, 1993.

[15] T. Sakurai, “Closed-Form Expressions for Interconnection Delay, Coupling, and Crosstalk in VLSI’s,” *IEEE Trans. on Electron Devices*, vol. 40, no. 1, pp. 118–124, 1993.

[16] A. Vittal and M. Marek-Sadowska, “Crosstalk Reduction for VLSI,” *IEEE Trans. Computer-Aided Design*, vol. 16, no. 3, pp. 290–298, 1997.

[17] A. Devgan, “Efficient coupled noise estimation for on-chip interconnects,” in *Proc. IEEE/ACM Int. Conf. on Computer Aided Design*, pp. 147–151, 1997.

[18] W. Elmore, “The Transient Response of Damped Linear Networks with Particular Regard to Wideband Amplifiers,” *Journal of Applied Physics*, vol. 19, no. 1, pp. 55–63, 1948.

[19] J. Rubinstein, P. Penfield, and M. Horowitz, “Signal Delay in RC Tree Networks,” *IEEE Trans. Computer-Aided Design*, vol. 2, no. 3, pp. 202–211, 1983.

[20] F. Dartu, N. Menezes, J. Qian, and L. Pillage, “A Gate-Delay Model for High-Speed CMOS Circuits,” in *Proc. Design Automation Conf.*, pp. 576–580, 1994.

[21] P. Gill, W. Murray, and M. Wright, *Practical Optimization.* London: Harcourt Brace and Co., 1981.

[22] N. Menezes, R. Baldick, and L. Pileggi, “A Sequential Quadratic Programming Approach to Concurrent Gate and Wire Sizing,” *IEEE Trans. Computer-Aided Design*, vol. 16, no. 8, pp. 867–881, 1997.

[23] C. Chu and D. Wong, “A New Approach to Simultaneous Buffer Insertion and Wire Sizing,” in *Proc. Design Automation Conf.*, pp. 614–621, 1997.

[24] Semiconductor Industry Association, *National Technology Roadmap for Semiconductors.* 1994.