

Crosstalk Constrained Global Route Embedding

Phiroze N. Parakh and Richard B. Brown

{phiroze, brown}@eecs.umich.edu

Advanced Computer Architecture Laboratory
Department of Electrical Engineering and Computer Science
University Of Michigan

Abstract - Route Embedding, a new method for mitigating the impact of crosstalk, is presented. It modifies a set of global-route structures to prevent timing and noise-margin violations caused by crosstalk, while maintaining routing constraints. An accurate and computationally-efficient empirical model for crosstalk impact is presented which by capturing noise and delay-changes on coupled conductors, permits a performance-driven approach to addressing crosstalk. Linearized crosstalk constraints are derived and satisfied for the expected noise and wire-delays at critical signal sinks. Unsatisfied constraints are resolved by inserting ground shields and by selective re-route through uncongested regions. Routing capacity constraints are enforced to guarantee a detailed routing solution.

1 Introduction

The inability of present-day CAD tools to handle what were historically second-order wiring effects, such as crosstalk-induced delay and noise, manifests itself in an increased number of design iterations. Timing errors and logic upsets due to crosstalk illustrate the severity of this problem. Most methodologies address crosstalk after generating a detailed routing. This is followed by analysis and rerouting where necessary. The complexity of detailed routing, parasitic extraction, crosstalk analysis and subsequent rip-up and reroute iterations, can be prohibitive for large designs. In addition, the outcome can be unpredictable, as this iterative process arises from the unconstrained creation of a detailed routing solution. Not only is this method compute-intensive, but it also puts a burden on the designer who must determine limits on coupling capacitances. Without knowledge of signal temporality, timing slack or receiver noise margins, this approach will be overly pessimistic and prone to solutions that require too much area.

Published approaches for minimizing crosstalk fall into three categories. The first of these modifies existing route structures to meet capacitance constraints [1, 2]. This methodology is incapable of rerouting, so is limited to local improvements. The second approach involves routing while minimizing coupling capacitance [4, 5, 7]. Limited by the complexity of detailed routing, this approach is forced to compromise on the quality of crosstalk analysis. Both of these methods are confined to pre-determined route topologies and lack temporal information about the signals. The third method addresses crosstalk by minimizing coupling capacitance during global [9] and area routing [8]. This approach relies upon rip-up and reroute to satisfy crosstalk constraints. The simplified crosstalk models used in the aforementioned methods preclude a performance-driven approach that could guaranteed convergence.

This paper introduces a new approach for minimizing crosstalk during global routing. It satisfies constraints that permit a trade-off between physical and temporal proximity, thereby

enabling a routing solution with no crosstalk violations. The paper begins with the description of an empirically-derived crosstalk model and continues with an introduction to route embedding. Next, the global-route embedding based methodology itself is presented. The paper concludes with experimental results.

2 Empirical Crosstalk Model

Bakoglu [10] showed that the wire-delay on a distributed RC line would contain an $R_W \cdot (C_S + C_C)$ time constant, where R_W is the interconnect resistance, and C_S and C_C are substrate and coupling (line-to-line) capacitances. This can be reduced to

$$R_W \cdot (C_S + C_C) = \frac{\rho \cdot l}{w \cdot t} \cdot \left(\frac{\epsilon_K \cdot l \cdot w}{h} + \frac{\epsilon_K \cdot l \cdot t}{s} \right), \quad (1)$$

where ρ is resistivity of the conductor, ϵ_K is the insulator dielectric constant, and w, t and h are the conductor's width, thickness and separation from the substrate respectively. Terms l and s represent the coupled length and spacing of the interconnect. If we define $h = \text{constant} \cdot s$, this equation suggests an l^2/s form for curve fitting. Devgan et al [11] showed with a distributed line model that the maximum induced noise due to crosstalk could be approximated by $\zeta_i = R_i \cdot I_c + \zeta_{i-1}$, where ζ_i is the amplitude of the induced noise at node i , R_i is the resistance driving that node, I_c is the total crosstalk current flowing into i and ζ_{i-1} is the amplitude of the induced noise at node $i-1$. A transformation similar to Eq. 1 again yields the l^2/s form.

The following section demonstrates the fitness of l^2/s for modeling crosstalk.

2.1 Simulating Crosstalk-Induced Delay and Noise

Figure 1 illustrates a coupled, distributed- RC circuit used to simulate crosstalk. The coupled RC circuit with specified design and technology parameters was simulated with SABER [12]. Each coupled line is built out of 40 RC components. Signal a was defined to be the aggressor and v , the victim. Parameters r_a and r_v are the aggressor and victim driver resistances. Parameters a_e and v_e are their edge-rates, with a_e constrained to be greater than v_e . Signals a and v were made to transition in the opposite directions to simulate the worst-case delay-increase scenario. Parameters a_a and v_a are the arrival times (as measured at the 50% points) of the aggressor and victim waveforms. To simulate various coupling-related events, the arrival times, edge rates, spacing and

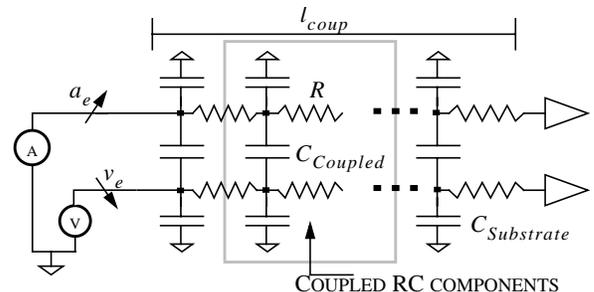


Figure 1. Circuit and parameters for simulating crosstalk.

Table 1: Crosstalk experiment parameters.

Parameter	Range-min	Range-max
Coupled length: (l)	100 μ m	3000 μ m
Wire spacing: s	0.21 μ m	0.42 μ m
Edge rate ratio: $ a_e/v_e $	1	0.2
Arrival time delta: $a_a - v_a$	$-v_e/2$	$+v_e/2$
Driver ratio: r_a/r_v	0.25	1.00

coupled wire-length of the signals, were swept through the ranges specified in Table 1. Technology constants were obtained from the SIA-1997 roadmap [13]. Next, the change in wire delay (or amplitude of injected noise), due to coupling is extracted. In the case of parameter sweeps this output was curve-fitted to a specified function by least squares error minimization.

The arrival time difference, normalized by the victim's edge rate

$$\delta a = \frac{a_a - v_a}{0.8 \cdot v_e} \quad (2)$$

is an indicator of the temporal proximity of the input transitions. Therefore, when $\delta a = -0.5$, the aggressor crosses its 50% point as the victim crosses its 20% point. At $\delta a = 0$, the two signals cross at their 50% points concurrently and at $\delta a = 0.5$, the aggressor crosses its 50% point as the victim crosses its 80% point. When $|\delta a| \leq 0.5$, the change in wire-delay τ , is measured. If the edges of the signals do not overlap ($|\delta a| \geq 0.5$), the victim's delay is unaffected, so instead of delay, the amplitude (at the receiver) of the induced noise ζ , on the quiescent victim line, is measured. Figures 2a and 2b illustrate the change in wire-delay and induced noise-amplitude due to coupling, as a function of coupled length and spacing.

2.2 Empirically Curve-Fitting to l^2/s

Empirical analysis of the wire-delay change (τ), and amplitude of the noise pulse (ζ), due to coupling, suggests the following equations:

$$\tau = \frac{\alpha \cdot l^m}{s^n} \text{ and } \zeta = \frac{\beta \cdot l^m}{s^n}, \quad (3)$$

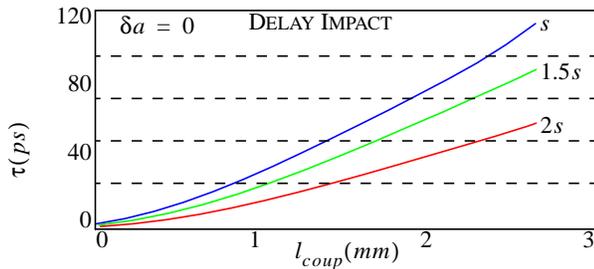


Figure 2a Delay impact vs. coupled length and spacing.

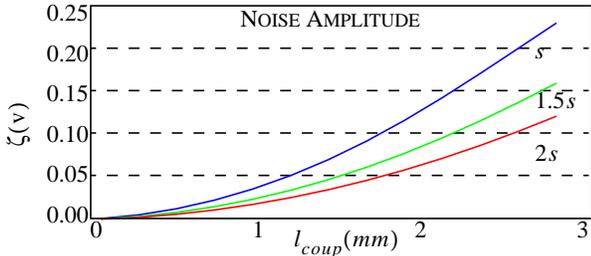


Figure 2b Noise amplitude vs. coupled length and spacing.

Table 2: Sample m and n parameters.

Spacing	Coupled length	Delay		Noise	
		m	n	m	n
s_0	$< 3500\mu$ m	1.98	0.96	1.97	0.97
$1.5 \cdot s_0$	$< 3500\mu$ m	1.88	0.93	1.91	0.94
$2.0 \cdot s_0$	$< 3500\mu$ m	1.80	0.90	1.87	0.92

where α and β are curve fit constants. Within bounds on l and s , the values of m and n were empirically observed to be near two and one respectively (Table 2). This allows Eq. 3 to be approximated as,

$$\tau = \frac{\alpha \cdot l^2}{s} \text{ and } \zeta = \frac{\beta \cdot l^2}{s} \quad (4)$$

which captures the l^2/s relationship in a very simple form. The errors in these approximations are plotted below. For a maximum coupled wire length of 3.5mm, Figure 3a shows τ to be accurate to within $\pm 8\%$. Figure 3b shows ζ to be accurate within $\pm 1\%$ for a similar maximum-length constraint. With extremely aggressive edge rates, the error in ζ increases to $\pm 10\%$.

Term α captures the effect of temporal proximity when computing wire-delay changes (Figure 4a). When measuring noise, Term β captures the dependence on the strength of the aggressor waveform (Figure 4b). Simulations show that α diminishes as a function of increasing temporal separation ($|\delta a|$). Furthermore, the magnitude of α is dependent on the strength of the aggressor (a_e/v_e). Since α and β are non-linearly related to the arrival time and edge rates of the interfering signals, a table-lookup approach is implemented. A β value is obtained by indexing the β -table with a_e and r_v if the nets are temporally orthogonal (i.e. their transitions do not overlap: $|\delta a| \geq 0.5$). If the transitions on the adjacent nets interact, then an α value is determined by indexing an α -table with δa and the edge-rate ratio a_e/v_e . The edge rates are bound to an acceptable range, as determined by designers and the technology (e.g., between 200ps and 800ps). This bounding condition applies for both tables. For the α -table, δa is bound by -0.5 and 0.5. Values within the tables ranges are determined by a linear-interpolation. A computationally efficient table lookup will therefore predict the curve fit parameters for all crosstalk behavior

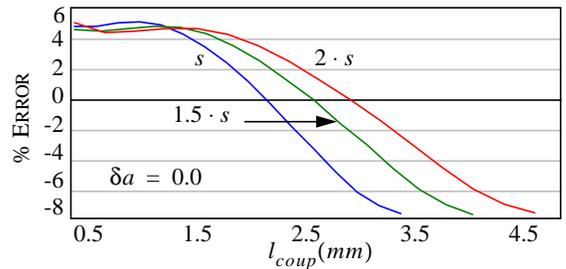


Figure 3a Crosstalk delay-impact: approximation error.

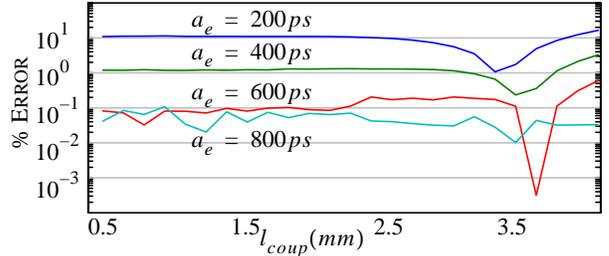


Figure 3b Crosstalk noise-amplitude: approximation error.

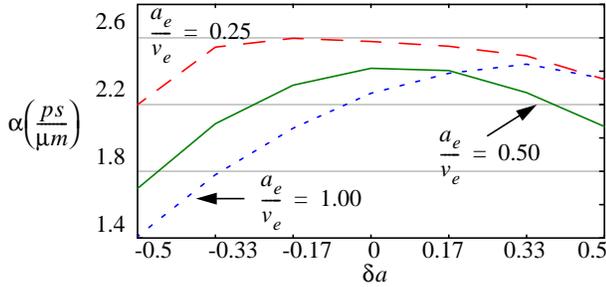


Figure 4a Effect of temporal proximity on α .

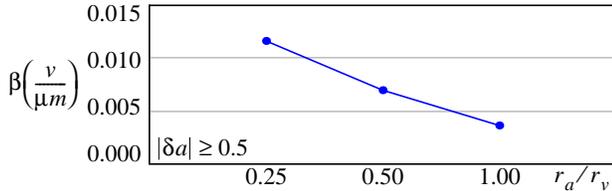


Figure 4b Effect of aggressor-victim drive strength on β .

under these predefined conditions, and enable a performance-driven approach for addressing crosstalk.

This empirical model is sufficiently accurate and flexible for a variety of scenarios. A table lookup for α and β makes the process computationally efficient. Furthermore, it enables a performance-driven approach for addressing the impact of crosstalk. The popular method of minimizing coupling capacitance [5, 7] precludes a performance-driven approach. Figure 5 compares the empirical model to the capacitive metric. For a given constraint (target), the linear model can be made to match the empirical model. However, it will permit greater coupling length for delays above the target and over-constrain the coupling length for delays below the target. Similarly, for a given coupling length constraint, the linear model will incorrectly estimate the crosstalk impact.

3 Global Route-Embedding

The complexity of detailed routing [14], makes concurrently minimizing crosstalk constraints computationally prohibitive. This complexity is evidenced by a sampling of the tasks performed by a detailed router. These include creating minimum length connections, adhering to design rules and electrical correctness, minimizing the number of layers, bends and vias, enforcing wire widths, and following the global tree topologies while avoiding blockages.

Channel and switchbox routing are used to simplify route topology while satisfying crosstalk constraints [4, 5, 2], but these rigid routing approaches have limited applicability in modern design methodologies. Other approaches attempt to dictate a detailed routing through assignment or adjustment of physical adjacency [1, 3]. These schemes are prone to increasing routing area because they do not consider wireability.

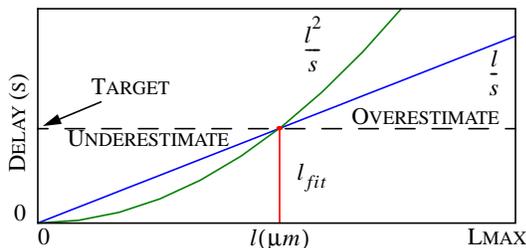


Figure 5. The empirical model vs. a capacitive crosstalk

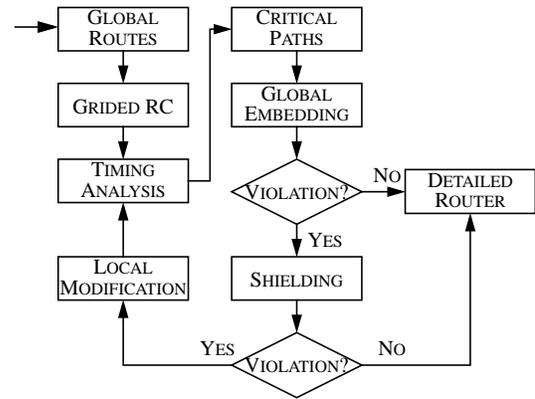


Figure 6. Methodology.

Route embedding is a means of modeling a detailed router without undertaking the entire complexity of detailed routing. Instead of simplifying the crosstalk analysis for insertion into a detailed router, the detailed router itself is abstracted. In this way, only the fundamental task of generating non-overlapping route structures needs to be modeled. Route embedding works by attempting to express this non-overlap requirement of a detailed router with global routes. Abstraction allows the route embedder to target its analysis toward accurately satisfying timing and noise constraints. This is preferable to “searching” for a valid solution using rip-up and re-route procedures. Lastly, a route embedder simultaneously processes a set of route trees, permitting a global view of the task.

An implementation of route embedding with crosstalk constraints is presented in the next sub-section. Route embedding, which is applicable at the global routing stage, expresses a routing solution through a set of desired spacings for multiple critical nets. It accurately trades-off physical or temporal proximity to permit the creation of a routing with no slack and noise-margin violations due to crosstalk.

3.1 Crosstalk Constrained Route Embedding Methodology

The methodology depicted in Figure 6 begins by overlaying a global routing solution with a grid. Each element of the grid is called a gcell. A static timing analyzer capable of modeling these global trees is invoked to obtain slack values at each sink. Additionally, each net is annotated with waveform parameters (delay and edge-rate) at each gcell. The RC trees are modeled using an exponential input extension [6] to the explicit three-pole model by Tutuianu et al [17]. Temporal information for each net at each gcell is used to derive the expected impact at each sink. The criticality of these sinks is determined by their magnitude of slack and noise margin violations. Timing-annotated nets are processed by a linear system solver, the program of which is constrained by the timing and noise margins at all critical sinks. Capacity constraints are enforced to guarantee a routing solution. The solution will define spacing constraints that satisfy the timing and noise margins. Nets may be rerouted if routing capacity constraints are breached. If crosstalk constraints are not satisfied, the expected impacts are refined, and a new linear program is generated and solved. This process continues until timing and capacity constraints are satisfied. The resultant spacings and gcell refinements represent a route embedding that satisfies crosstalk constraints.

3.2 Global Route Embedding

3.2.1 Initialization

The global embedder begins by overlapping the placement expression with an $\langle n \times m \rangle$ grid of gcells. The global trees provided by the placement engine are then traced over the grid, assigning nets to gcells. Let ϕ_c be the set of nets passing through a gcell c and $|\phi_c|$ be the number of nets passing through c . For each net traced, the RC -tree equivalent is constructed by assigning a constant resistance (R_{gc}) and capacitance (C_{gc}) at each gcell as shown in Figure 7. These values are calculated as:

$$R_{gc} = \rho^y \cdot \frac{l_{gc}}{A_{net}} \text{ and } C_{gc} = \epsilon_\kappa \cdot l_{gc} \cdot \left(\frac{w_{net}}{h_0^y} + 2 \cdot \frac{t_0^y}{s_0^y} \right), \quad (5)$$

where ρ^y , s_0^y , h_0^y and t_0^y are the resistivity, minimum spacing, height and thickness of the expected layer y for the net. A_{net} and w_{net} are the cross-sectional area and width of the net (passing through the gcell). l_{gc} defines the length and width of the gcell. This RC circuit, when loaded with sink capacitances is processed by a timing engine to determine the arrival time and edge rate at each gcell. The timing engine also computes the slack at each sink.

3.2.2 Expected Crosstalk Delay Impact

The interaction of a net with its neighbors as it passes through a set of gcells is modeled by extending Eq. 5 to compute the expected delay impact at each sink. The expectation value of a function $f(x)$ ¹, is defined to be

$$\langle f(x) \rangle = \sum_x f(x) \cdot P(x). \quad (6)$$

Where $P(x)$ is the Probability Distribution Function (PDF) for x . Define term π_k^n as the set of gcells that describe the path from its source to sink k for a given net n . For example, in Fig. 7 $\pi_{k_2}^n = \{g_{0,0}, g_{1,0}, g_{2,0}\}$, where $g_{i,j}$ is the gcell indexed at row i and column j . Define term L_k^n as the path length of π_k^n (in gcells) from source to sink k . Let term Φ_k^n be the set of nets passing through all the gcells in π_k^n . The delay impact is derived from the subset of nets $\Phi_{\tau,k}^n \subset \Phi_k^n$, which switch with n . From Eq. 5 and 6,

$$\langle \tau_k^n \rangle = \frac{\langle \alpha^{n,k} \rangle \cdot (L_k^n \cdot l_{gc})^2}{\langle s^{n,k} \rangle} \quad (7)$$

is defined as the expected delay impact at the sink. $\langle \alpha^{n,k} \rangle$ is the expected α -parameter and $\langle s^{n,k} \rangle$ is the expected spacing that would mimic the multiple couplings. Figure 7 motivates the translation of Eq. 7 to account for multiple coupling, at each gcell, along π_k^n . This is achieved by,

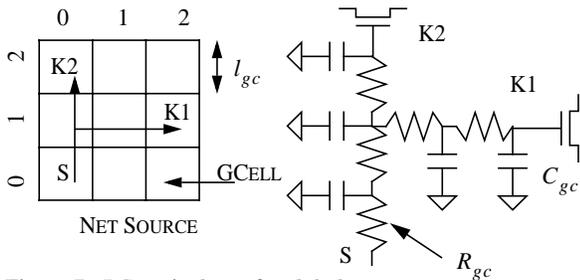


Figure 7. RC equivalent of a global tree.

1. For discrete x only. If x is continuous then
 $\langle f(x) \rangle = \int f(x) \cdot P(x) \cdot dx$

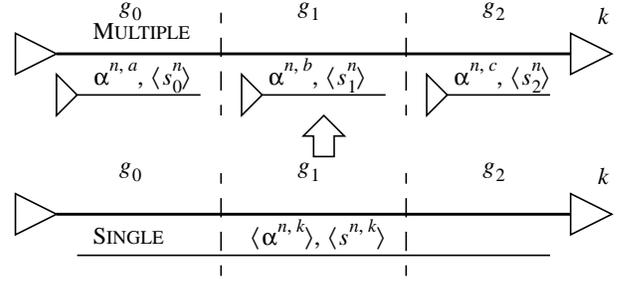


Figure 8. Approximating multiple coupling events.

$$\langle \tau_k^n \rangle = \left(\sum_{c \in \pi_k^n} \frac{\langle \alpha_c^n \rangle}{\langle s_c^n \rangle} \cdot \frac{1}{L_k^n} \right) \cdot (L_k^n \cdot l_{gc})^2 \quad (8)$$

$$\langle \tau_k^n \rangle = L_k^n \cdot \left(\sum_{c \in \pi_k^n} \frac{\langle \alpha_c^n \rangle}{\langle s_c^n \rangle} \right) \cdot l_{gc}^2$$

where $\langle \alpha_c^n \rangle$ is the expected α -parameter for the net n within the gcell c , and $\langle s_c^n \rangle$ is the expected spacing around the net in the gcell c . It is expected that each gcell will contain only one edge of n and therefore $\langle \alpha_c^n \rangle$ is not required to differentiate with k . Since the nets within a gcell are equally likely to be adjacent, the impact PDF for any net n , passing through gcell c will be the inverse of the total number of nets. Hence,

$$\langle \alpha_c^n \rangle = \sum_{e \in \phi_c} \alpha^{n,e} \cdot \frac{1}{|\phi_c|} = \frac{\overline{\alpha^{n,\phi_c}}}{|\phi_c|} \quad (9)$$

states that the expected impact within a gcell is the average of the α -impact factors between net n and all other nets j passing through c . These α values are obtained through the appropriate α -table lookup. With the inclusion of a factor of two, to account for two possible adjacencies per entry within the gcell, Eq. 8 can now be reduced to

$$\langle \tau_k^n \rangle = L_k^n \cdot \left(\sum_{c \in \pi_k^n} \frac{2 \cdot \overline{\alpha^{n,\phi_c}}}{\langle s_c^n \rangle} \right) \cdot l_{gc}^2. \quad (10)$$

This equation is initially evaluated with the expected spacing for each net n , through each gcell c with $\langle s_c^n \rangle = s_0^y$. The process begins with all nets minimally spaced apart on their expected layer y . Let the slack at each sink be S_k^n . The delay-criticality of this sink is computed by $S_k^n / \langle \tau_k^n \rangle$. A sink is critical if this ratio is less than one. Define Θ as the set of critical sinks. Furthermore, let Γ be the set of gcells covered by the paths to the sinks in Θ .

3.2.3 Expected Crosstalk Noise Impact

Devagan et al., showed that the total noise on a line is upper-bounded by a superposition of the individual noise sources. It is clear then that a bound the total expected noise impact to sink k is the summation of all expected crosstalk noise, due to the nets $\Phi_{\zeta,k}^n \subset \Phi_k^n$, along the path π_k^n .

$$\langle \zeta_k^n \rangle = \sum_{c \in \pi_k^n} \langle \zeta_c^n \rangle \quad (11)$$

Similar to Eq. 9, the expected noise-impact $\langle \zeta_c^n \rangle$, on the net n in gcell c is dependent on the noise impact PDF. Therefore,

$$\langle \zeta_c^n \rangle = \sum_{e \in \phi_c} \zeta^{n,e} \cdot \frac{1}{|\phi_c|} = \zeta_c^{n,\phi_c} = 2 \cdot \frac{\overline{\beta_c^{n,\phi_c}} \cdot l_{gc}^2}{\langle s_c^n \rangle}. \quad (12)$$

As before, this equation is initially evaluated with the expected spacings set to s_0^y . The β factors are obtained through appropriate lookups into the β -table. The factor of two accounts for the two

possible adjacencies. Define M_k^n as the noise-margin of sink k on net n . All sinks with $M_k^n \geq \langle \zeta_k^n \rangle$ are added to the set of critical sinks (Θ).

3.2.4 Computing an Embedding

User defined parameters S_k^n and M_k^n are the acceptable slack and noise-margins at sink k of net n . The crosstalk constraints $\langle \tau_k^n \rangle \leq S_k^n$ and $\langle \zeta_k^n \rangle \leq M_k^n$ must be satisfied. This result is achieved when a linear program (LP) solves for the expected spacings while satisfying crosstalk and capacity constraints. Each $\langle s_c^n \rangle$ represents a linear variable in the LP. The objective function minimizes the total spacing required to satisfy the LP:

$$\min \left\{ \forall k \in \Theta, \sum_{c \in \pi_k^n} \langle s_c^n \rangle \right\}. \quad (13)$$

For each path π_k^n to the sinks in set Θ , crosstalk constraints are asserted by

$$L_k^n \cdot \left(\sum_{c \in \pi_k^n} \frac{2 \cdot \overline{\alpha}_c^{n, \phi_c}}{\langle s_c^n \rangle} \right) \cdot l_{gc}^2 \leq S_k^n \text{ and} \quad (14)$$

$$\left(\sum_{c \in \pi_k^n} \frac{2 \cdot \overline{\beta}_c^{n, \phi_c}}{\langle s_c^n \rangle} \right) \cdot l_{gc}^2 \leq M_k^n. \quad (15)$$

Eq. 14 and 15 guarantee the slack (S_k^n) and noise margin (M_k^n) requirements at each critical sink. Each gcell with a critical path is constrained by its available supply of routing tracks:

$$\forall c \in \pi_k^n, \sum_{n \in \Phi_c} (\langle s_c^n \rangle + w_c^n) \leq l_{gc} \quad (16)$$

where w_c^n is the width of net n passing through gcell c which measures l_{gc} on a side. Finally, each spacing variable is trivially constrained by $\forall (n, c), \langle s_c^n \rangle \geq s_0^y$.

Both Eq. 14 and Eq. 15 are non-linear constraints due to the $1/\langle s_c^n \rangle$. They are linearized by a first-order Taylor's series approximation to $1/x$ centered at s_0^y , $(4 \cdot s_0^y)/3$ and $2 \cdot s_0^y$. By introducing σ_c^n as a slack variable for each $\langle s_c^n \rangle$, Eq. 14 and Eq. 15 can be transformed into:

$$L_k^n \cdot \left(\sum_{c \in \pi_k^n} \overline{\alpha}_c^{n, \phi_c} \cdot \sigma_c^n \right) \cdot l_{gc}^2 \leq S_k^n \text{ and} \quad (17)$$

$$\left(\sum_{c \in \pi_k^n} \overline{\beta}_c^{n, \phi_c} \cdot \sigma_c^n \right) \cdot l_{gc}^2 \leq M_k^n$$

with the approximation to $1/\langle s_c^n \rangle$ represented by three linear constraints:

$$\sigma_c^n \geq \frac{2 \cdot (x \cdot s_0) - \langle s_c^n \rangle}{(x \cdot s_0)^2} \cdot \langle \alpha_c^n \rangle \quad ; x \in \left\{ 1, \frac{4}{3}, 2 \right\}. \quad (18)$$

This LP formulation requires $\forall (n, k) \in \Theta: 2 \sum L_k^n$ linear variables. The linearization of each $\langle s_c^n \rangle$ requires three constraints. Each path will have one slack and one noise margin constraint. In addition, each gcell is bounded by one capacity constraint. This results in a total of

$$\forall (n, k): 3 \cdot \sum L_k^n + 2 \cdot |\Theta| + |\Gamma| \quad (19)$$

linear constraints.

3.2.5 Resolving an Infeasible Embedding

Figure 9 illustrates the process by which infeasibility in the linear program for global embedding is resolved. A violation occurs when a path does not meet its slack or noise margin require-

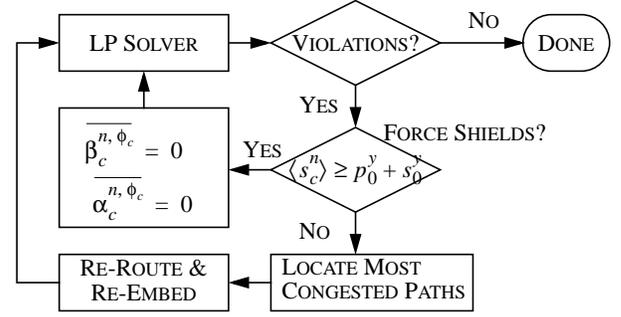


Figure 9. Iterating to satisfy violated crosstalk constraints.

ments. The paths and variables that do not satisfy constraints are analyzed. First, all $\langle s_c^n \rangle$ greater than $p_0^y + s_0^y$ have their impact parameters set to zero, where p_0^y is the minimum pitch on layer y . This is equivalent to running a ground shield or a signal with orthogonal temporal locality adjacent to the net within the gcell c . Then slack and noise-margin constraints are re-evaluated. If constraints are still not met, sections of violated paths passing through regions of high wiring-congestion are minimally rerouted through un-congested gcells. After re-timing, the impact parameters are updated and a new embedding LP is generated and solved. This process continues until the crosstalk constraints are satisfied.

4 Results

Table 3 lists pertinent statistical data for the benchmark circuits on which global route embedding was evaluated. The designs were embedded using the SIA predicted technology rules for 1999. The designs were selected for their size, since larger netlists tend to have longer paths that require crosstalk constraints. Netlist s15850 was processed by a congestion driven quadratic placer [16] to produce a less-congested placement in s15850_c. Design NNC is the core of a image-recognition neural-net chip. The EXU (Execution Unit) is a large functional block within the Fixed-point Unit (FXU) of the PUMA processor [15]. Both contain fewer paths than the standard-cell benchmarks, but, their average path-length is greater.

The embedder processed each design from Table 3. The results of the first iteration through the solver are listed in Table 4. Data for the total routing area required to satisfy crosstalk ($\sum \langle s_c^n \rangle$), the number of shields inserted, the percent of unsatisfied paths and the average percentage of violations are presented. The second iteration following rip-up and re-route through uncongested gcells satisfied all constraints.

The constraints on the critical paths are scaled to simulate the impact of tighter crosstalk constraints. For example, a constraint multiplier of 0.7 reduces the slack and noise-margins on all critical-sinks by 30%. With the multiplier set to 1.0, the global route-embedder generated solutions that did not require rerouting. Tightening the crosstalk constraints causes the global embedder to allocate more space and insert more shields. Figure 10 displays the trade-off in area made by the global embedder as the timing and

Table 3: Benchmark design statistics (1999 SIA technology).

Design	Type	Nets	Paths	Cells + Blocks	Size mm^2	Avg. critical path l (μm)
s15850	Cell	10369	13928	8620 + 0	0.31	110.3
s15850_c	Cell	10369	13928	8620 + 0	0.30	87.2
NNC	Cell & Block	15239	31465	14825 + 2	4.1	424.6
EXU	Cell & Block	4461	9951	3822 + 3	2.47	492.2
FXU	Block	2322	2322	0 + 17	11.9	1257.5

Table 4: First iteration through Global Route Embedding.

Design	Constraint multiplier	$\sum \langle s_c^n \rangle$	Shields	Specification Violations (%)	
				Paths not satisfied (%)	Average violation
s15850	1.0	1156.6	0	0.0	0.0
	0.9	1306.8	36	1.8	6.5
	0.8	1555.5	32	1.7	17.5
	0.7	1868.5	24	2.1	25.4
s15850_c	1.0	998.6	0	0.0	0.0
	0.9	1073.8	63	1.2	4.3
	0.8	1435.7	72	1.3	6.7
	0.7	1733.4	96	1.5	8.2
NNC	1.0	2335.6	0	0.0	0.0
	0.9	2693.3	0	0.8	0.1
	0.8	3335.1	9	1.1	0.7
	0.7	3925.4	16	1.4	1.7
EXU	1.0	2634.2	128	0.0	0.0
	0.9	2877.9	247	0.0	0.0
	0.8	3197.8	229	0.0	0.0
	0.7	3684.6	385	0.0	0.0
FXU	1.0	978.3	155	0.0	0.0
	0.9	1027.7	130	0.0	0.0
	0.8	1147.4	147	0.0	0.0
	0.7	1312.5	135	6.0	2.0

noise specifications are tightened. This is accompanied by an increase in the number of unsatisfied paths, and the magnitude of their violation. In the case of the FXU, with a multiplier of 0.7, 6% of the paths did not satisfy their constraints by an average of 2% on the first pass of the embedder.

A correlation between crosstalk and wiring-congestion is noted by comparing s15850 to s15850_c. A reduction in the number of unsatisfied paths and the average amount of their violation is noted. This is because the global embedder can introduce more spacing (and shields), while requiring less reroute, to satisfy crosstalk constraints, in an uncongested design. This explains the increase in $\sum \langle s_c^n \rangle$ for s15850_c when the multiplier is scaled to 0.8.

5 Conclusions

Route Embedding is presented as a new method for addressing crosstalk. This approach minimally modifies a set of routes to satisfy timing and noise-margin constraints at individual sinks. The traditional method of addressing crosstalk by minimizing coupling capacitance is rejected in favor of a performance-oriented crosstalk

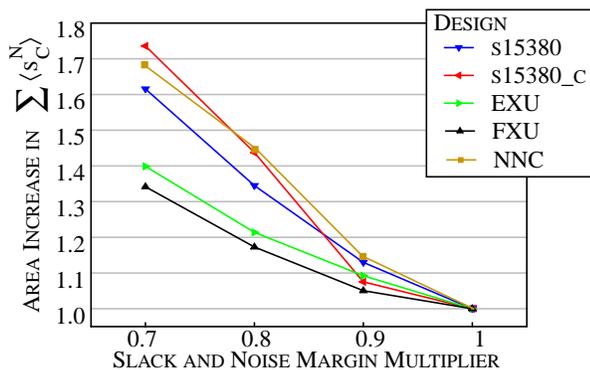


Figure 10. Trade-off between area and crosstalk severity

model. Analysis of the interactions between capacitively-coupled conductors disclosed a relatively simple, yet extensible model capable of predicting both crosstalk-induced noise-amplitude and wire-delay changes. The model captures the dependence and sensitivity of crosstalk on temporal proximity and edge-rates. Simulations show the model to be accurate within 8% for wire-delay changes and 10% for noise amplitude. Furthermore, the model is linearized, permitting a trade-off between temporal proximity and spatial proximity in a linear solver.

Coupled with this crosstalk model, a global-route embedder with performance-driven crosstalk constraints is implemented. It satisfies slack and noise-margins at the critical sinks by computing expected spacings, inserting shields and selectively rerouting unsatisfied paths through uncongested regions. This approach, implemented as a linear program, guarantees that timing and noise specifications will be met. Routing capacity constraints are enforced to guarantee that a detailed routing solution is feasible. Experiments on large standard-cell groups and the top level routing of a microprocessor demonstrate the method. The global embedder satisfied crosstalk constraints at critical sinks for a range of performance goals, demonstrating a trade-off between area (spacing and shields) and crosstalk. A 30% tightening of constraints on the microprocessor required a 34% increase in critical-path routing area. Finally, a correlation between crosstalk and congestion is demonstrated. With uncongested designs, the global embedder is better able to meet their timing and noise specifications and limit the number of paths requiring rerouting.

6 References

- [1] T. Gao, C. L. Liu, "A spacing algorithm for performance enhancement and cross-talk reduction". *1993 IEEE/ACM ICCAD*, pp. 697-702.
- [2] A. Onozawa, K. Chaudhary, E. S. Kuh, "Minimum crosstalk channel routing". *1993 IEEE/ACM ICCAD*, pp. 692-696.
- [3] K. Chaudhary, A. Onozawa, E. S. Kuh, "Performance driven spacing algorithms using attractive and repulsive constraints for submicron LSIs". *IEEE TCAD-ICAS*, vol.14, no.6, pp. 707-719.
- [4] K. S. Jhang, S. Ha, C. S. Jhon, "A segment rearrangement approach to channel routing under the crosstalk constraints". *ASPCAS-94*, pp. 536-541.
- [5] K. S. Jhang, S. Ha, C. S. Jhon, "COP: a Crosstalk Optimizer for grided channel routing". *IEEE TCAD-ICAS*, vol.15, no.4, pp. 424-429.
- [6] P. N. Parakh, "A Design Methodology for Minimizing Crosstalk," *Ph.D Dissertation*. University Of Michigan, 1998.
- [7] H. Zhou, D. F. Wong, "Crosstalk-Constrained Maze Routing Based on Lagrangian Relaxation," *Proceedings ICCD*. pp. 628-633, 1997.
- [8] H. P. Tseng, L. Scheffer, C. Sechen, "Timing and Crosstalk Driven Area Routing," *DAC-98*.
- [9] H. Zhou, D. F. Wong, "Global Routing with Crosstalk Constraints," *DAC-98*.
- [10] H. B. Bakoglu, "Circuits, Interconnections, and Packaging for VLSI," Addison-Wesley Publishing Company, 1990.
- [11] A. Devgan, C. J. Alpert, S. T. Quay, "Buffer Insertion for Noise and Delay Optimization," *DAC-35*. pp. 362-367, 1998.
- [12] SABER. <http://www.analogy.com/>.
- [13] 1997 Semiconductor Industry Association Roadmap, <http://www.sematech.org/>.
- [14] T. Lengaur, "Combinatorial Algorithms for Integrated Circuit Layout," John Wiley and Sons, Chichester, 1990.
- [15] PUMA Project, <http://www.eecs.umich.edu/UMichMP>.
- [16] P. N. Parakh, R. Brown, K. Sakallah, "Congestion Driven Quadratic Placement," *DAC-35*, 1998.
- [17] B. Tutuianu, F. Dartu, L. Pileggi, "An Explicit RC-Circuit Delay Approximation Based on the First Three Moments of the Impulse Response," *DAC-33*, pp 611-616, 1996.