A DESIGN METHODOLOGY FOR ADDRESSING CROSSTALK IN INTEGRATED CIRCUITS

TECHNICAL REPORT NO. SSEL-291

1998

By

Phiroze N. Parakh

SOLID-STATE
ELECTRONICS
LABORATORY

THE UNIVERSITY OF MICHIGAN
ARTES · SCIENTIA · VERITAS
1817

DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
THE UNIVERSITY OF MICHIGAN, ANN ARBOR

This report has also been submitted as a dissertation in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the University of Michigan, 1998.

REPORT DOCUMENTATION PAGE		Form Approved OMB NO. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comment regarding this burden estimates or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.		

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE 1998	3. REPORT TYPE AND DATES COVERED Technical

4. TITLE AND SUBTITLE A Design Methodology for Addressing Crosstalk in Integrated Circuits	5. FUNDING NUMBERS
6. AUTHOR(S) Phiroze N. Parakh	DAAH04-94-G-0327

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(ES) University of Michigan Department of Electrical Engineering 1301 Beal Ave. Ann Arbor, MI 48109-2122	8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army Research Office P.O. Box 12211 Research Triangle Park, NC 27709-2211	10. SPONSORING / MONITORING AGENCY REPORT NUMBER ARO 33790.74-EL-

11. SUPPLEMENTARY NOTES

The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution unlimited.	12 b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This dissertation focuses on a design methodology for addressing capacitive crosstalk. Crosstalk is a severe problem in the field of VLSI design where aggressive scaling of interconnect pitch has led to increased capacitance between adjacent traces, causing non-linear interactions evidenced as timing violations and erroneous circuit activity. New process technologies will achieve tighter metallization, increased clock frequencies, smaller voltage swings and longer interconnect. Estimates show these trends will double the impact of crosstalk during the next decade.

A physical design methodology that accounts for crosstalk with accurate and consistent estimates of wiring constraints throughout the design flow is presented. By maintaining a consistent view across the design flow, violations due to crosstalk become predictable, and therefore, avoidable. A case is made for estimating crosstalk using an empirical model, avoiding crosstalk using congestion-driven placement, and reducing crosstalk via a global-route embedder.

14. SUBJECT TERMS			15. NUMBER IF PAGES
			16. PRICE CODE

17. SECURITY CLASSIFICATION OR REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL

ABSTRACT

A DESIGN METHODOLOGY FOR ADDRESSING CROSSTALK IN INTEGRATED CIRCUITS

by

Phiroze N. Parakh

Chair: Richard B. Brown

This dissertation focuses on a design methodology for addressing capacitive crosstalk. Crosstalk is a severe problem in the field of VLSI design where aggressive scaling of interconnect pitch has led to increased capacitance between adjacent traces, causing non-linear interactions evidenced as timing violations and erroneous circuit activity. New process technologies will achieve tighter metallization, increased clock frequencies, smaller voltage swings and longer interconnect. Estimates show these trends will double the impact of crosstalk during the next decade.

A physical design methodology that accounts for crosstalk with accurate and consistent estimates of wiring constraints throughout the design flow is presented. By maintaining a consistent view across the design flow, violations due to crosstalk become predictable, and therefore, avoidable. A case is made for estimating crosstalk using an empirical model, avoiding crosstalk using congestion-driven placement, and reducing crosstalk via a global-route embedder.

Accurate models for crosstalk interactions are required to achieve timing convergence. A computationally efficient empirical model for crosstalk impact that captures noise and delay-changes on coupled conductors is presented. It permits a performance-driven approach that is superior to the popular method of minimizing

adjacent capacitance for addressing crosstalk. Simulations show the model to be accurate within 8% for wire-delay and 10% for noise.

A strong correlation between crosstalk and wiring congestion is demonstrated. Experiments on designs ranging from large standard-cell groups to a microprocessor show a majority of the critical nets have more than 50% of their length passing through regions of high congestion. Through interplay between a wiring-engine and quadratic placement, a global treatment of congestion is demonstrated. Experiments show up to 20% improvements in the wireability of designs.

A new method for mitigating the impact of crosstalk, route embedding, is presented. It simultaneously modifies a set of route structures to satisfy timing and noise constraints while maintaining routing constraints. Linearized crosstalk constraints are derived and satisfied for the expected noise and wire-delay at critical sinks. This is achieved by computing new spacings and inserting ground shields. Routing capacity constraints are enforced to guarantee a detailed routing solution. The methodology results in embeddings that satisfy a range of crosstalk constraints for all test cases.

PREFACE

List of Symbols that frequently occur in Chapter 2 and Chapter 5.

$a(t)$ Aggressor signal.

α Empirically derived constant for wire-delay change due to crosstalk.

a_a Aggressor signal arrival-time.

a_e Aggressor signal edge-rate.

$\alpha^{n,e}$ α-parameter between nets n and e.

α_c^n α-parameter within gcell c on net n

$\overline{\alpha_c^{n,\phi_c}}$ Average α-parameter for n in gcell c. Average of $\alpha^{n,e}$ for all $e \in \phi_c$.

β Empirically derived constant for noise amplitude due to crosstalk.

$\beta^{n,e}$ β-parameter between nets n and e.

β_c^n β-parameter within gcell c on net n

$\overline{\beta_c^{n,\phi_c}}$ Average β-parameter for n in gcell c. Average of $\beta^{n,e}$ for all $e \in \phi_c$.

c A gcell.

C_C Line-to-line (coplanar) capacitance. Also crosstalk capacitance.

δa Temporal Proximity: Arrival time difference, normalized by v_e.

ε_κ Dielectric constant multiplied by the permitivity of free space.

f_i Tangent for linearization of empirically derived crosstalk model.

Φ_k^n Set of nets passing through all gcells in π_k^n.

$g_{i,j}$ gcell at index i, j.

Γ Union of all π_k^n; $k \in \Theta$.

h Distance of conductor from substrate.

k A sink.

k_i i^{th} residue for RC circuit.

l Coupled length between two adjacent lines.

L_k^n	Cardinality of π_k^n ($\left\|\pi_k^n\right\|$).
λ_1	Maximum value for l, when spacing is minimum.
λ_j	Maximum value for l, when spacing is s_j.
m_i	i^{th} circuit moment.
M_k^n	Noise Margin at sink k of net n.
n	A net.
$\omega_i(u_i, s_j)$	Offset (y-intercept) for f_i.
p_i	i^{th} pole for RC circuit.
π_k^n	Path (set of gcells) from net n's source to sink k.
ρ	Conductor Resistivity.
Θ	Set of critical sinks.
r_a	Aggressor driver output resistance.
r_v	Victim driver output resistance.
s	Line-to-line spacing.
s_0	Minimum spacing of two adjacent lines.
s_j	j times minimum spacing (s).
S_k^n	Slack at sink k of net n.
t	Thickness of conductor.
τ	Change in wire-delay due to crosstalk.
τ_k^n	Change in wire-delay at sink k on net n due to crosstalk.
u_i	Point on crosstalk curve at which f_i is tangent.
$v(t)$	Victim signal.
v_e	Victim signal edge-rate.
v_a	Victim signal arrival-time.
w	Line width.
x_i	Point at which f_i and f_{i+1} intersect.
$\|x\|$	Cardinality (size) of set x.
$\langle x \rangle$	Expectation value for x.
y	A routing layer.
ζ_k^n	Noise amplitude at sink k on net n due to crosstalk.
ζ	Crosstalk induced noise amplitude.

TABLE OF CONTENTS

LIST OF TABLES

CHAPTER I

INTRODUCTION

Scaling of VLSI process technology has invalidated many of the simplifying assumptions in VLSI CAD. The inability to handle historically second-order wiring effects, such as crosstalk-delay and noise, manifests itself in increased numbers of design iterations. Timing errors and logical upsets due to crosstalk, illustrate the severity of this problem, and in addition the need for a proper solution. The inaccurate models and reactive back-end processes for addressing crosstalk in current design methodologies are inadequate. A physical design methodology that accounts for crosstalk, with accurate and consistent estimates of wiring constraints, throughout the design process, is required to eliminate the need for multiple design iterations.

1.1 Overview

The semiconductor industry is fueled by the cost-effectiveness of scaling. Designs are routinely implemented with $0.18\mu m$ feature sizes and are predicted to reach $0.08\mu m$ by 2006. Deep sub-micron (DSM) metallization has effectively invalidated assumptions that have previously shaped CAD for VLSI. Historically, the analysis of second and third-order effects such as crosstalk, RC-delay and electromigration were reserved for clock nets, long busses and power rails. However, the model of interconnect as a zero-delay (or capacitive) connection between the driver and receiver is no longer valid, causing many CAD methodologies to produce unacceptable solutions. The following quote succinctly illustrates the concerns of the VLSI physical-design industry:

"What people really mean when they refer to the deep sub-micron problem is that old abstractions are no longer adequate. The physics that were so neatly packaged by these abstractions are coming back with a vengeance as new technologies cause the relative importance of various phenomena to change. We have reached the point that wires matter more than gates in layout and power consumption. Should we continue to focus on logic minimization? With current ASIC methodologies beginning to fail it is crucial that we embark on the development of new abstractions and methodologies that make the right trade-offs for today's technologies and focus on speed and power rather than area." - [86]

Currently, CAD tools do little to mitigate signal integrity problems [58]. However, it is imperative to address these issues as a significant percentage of nets in a 0.25μm process (and lower) may be affected by one or more DSM issues:

- **Crosstalk:** A dynamic phenomenon that cannot be simulated by static analyzers. It can cause unpredictable glitches, timing violations and erroneous operation.

- **Noise:** The trend to reduce power supply voltages and threshold voltages has made circuits more susceptible to power supply noise.

- **Temperature:** Thermal gradients in large chips result in gate threshold differences and, therefore, reduced noise margins. It effects timing and reliability.

- **Processing variations:** Extremely thin gate-oxides (10-12 Å) amplify the impact of impurities and variations in thickness, causing noticeable threshold shifts, manifest as cross-chip gradients. This is of prime concern in analog design, DRAM's and other sensitive circuitry.
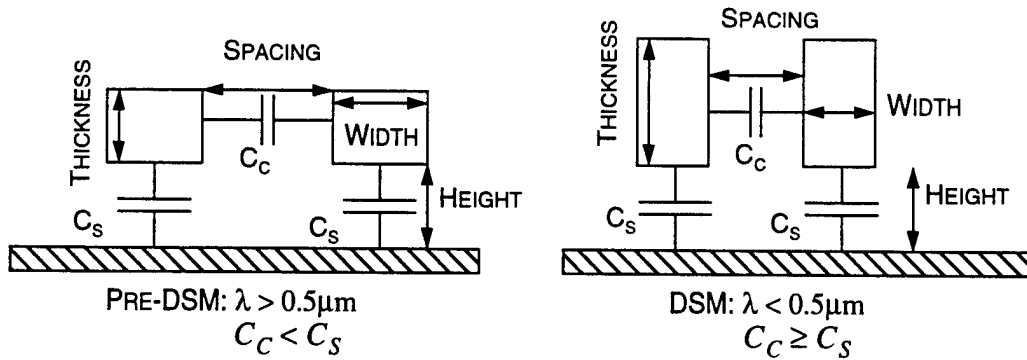
Figure 1.1   Metalization cross sections.

Term λ is the minimum feature size of the technology. Terms C_c and C_s are the coupling and substrate capacitances respectively.

Of these, capacitive crosstalk is receiving the most attention as it is proving to be a major reason for increased design iterations and failed timing constraints. The following quote by an Intel manager emphasizes the need for a design methodology to address crosstalk.

"With 4.5 million transistors, 0.35μm rules and a complex multilayer structure, the P55C is a prime example of deep-submicron design. Only a small part of the 4.5 million transistors are in highly regular structures like the cache. Most are in data paths and control logic. If there is a land mine in the design flow for a deep-submicron chip, it is crosstalk". [57]

The significance of these issues in all stages of VLSI CAD, has prompted much research into effective modeling of capacitively coupled RC delay and the mitigation of crosstalk through a physical design flow.

1.2   Defining Crosstalk

Capacitive crosstalk, also referred to as capacitive coupling, becomes significant when the spacing between conductors becomes comparable to (and less than) the thickness of the conductors. From Figure 1.1 it can be seen that DSM metallization is

characterized by high aspect ratios and dense interconnect. This implies a higher coupling-capacitance to substrate capacitance ratio. The larger, parasitic coupling capacitance between two such conductors will impact signal delay and integrity. Accurately predicting delay (and noise) along a path becomes problematic as the delay depends upon the substrate and coupling capacitances. Which, in turn, is a function of the input data, logic and timing behavior and structure of the neighboring interconnect. While crosstalk between adjacent wires can lead to timing errors, more catastrophic events such as logic upsets can occur due to crosstalk between nets and sensitive nodes. In a multi-layer metallization process, each layer adds uncertainty by adding its parasitic capacitances. The cost of an increased number of design iterations due to these uncertainties, motivates the need for predictive control of crosstalk.

Current solutions to crosstalk include post-routing optimization by spacing adjacent conductors, switching metal layers, and in some cases by selectively re-routing critical nets. No place-and-route strategies for minimizing noise on critical nodes have been published in the literature for digital VLSI. Furthermore, there is no comprehensive methodology for addressing these issues at a global level in VLSI CAD.

1.3 Technology Roadmap

Driven by the cost effectiveness of greater integration, shrinking metallization leads to narrower wires (lower capacitance to substrate) and reduced spacing. Sematech, a government-sponsored consortium of semiconductor and Electronic Design Automation (EDA) companies, provides a roadmap that predicts this continual scaling. The primary driver of this Semiconductor Industry Association (SIA) roadmap [1] is the continuation of Moore's Law [83]. The roadmap predicts shrinking metallization, increasing clock frequencies (faster edge rates), smaller voltage swings and longer interconnect. Tighter metallization implies reduced pitches, narrower conductors and therefore rising wire

Table 1.1 Interconnect Parameters: 1995 and 1998 SIA Roadmap.

The reduction in interconnect resistance (R) and coupling capacitance (Cc), between the SIA-1995 and SIA-1998 technology roadmap is due to the introduction of copper metalization and low-k dielectrics.

SIA YEAR:		1995	1998	1995	1998	1995	1998
Year	Pitch (nm)	R (Ω/sq)		Cc (fF/μm)		Aspect Ratio	
1999	420	0.290	0.044	0.19	0.08	2.0	1.8
2001	360	0.290	0.046	0.21	0.07	2.5	2.0
2003	300	0.840	0.053	0.24	0.06	3.0	2.1
2006	240	1.340	0.057	0.27	0.06	4.0	2.4

resistances [Table 1.1]. In an effort to control the rising resistance of the wires (and maintain the RC product), the aspect ratio is increased, thereby increasing the inter-wire (coupling) capacitance. Longer interconnect increases the dominance of interconnect delays and the susceptibility to capacitive coupling effects. Finally, lower voltages compounded by reducing thresholds will make designs increasingly sensitive to supply and crosstalk induced noise.

This analysis can be analytically interpreted. If all dimensions are uniformly scaled by S, then the following claims can be made:

1. Coupling capacitance will scale by S

2. Capacitance to ground will scale by $1/S$

3. Crosstalk per unit length will scale by S^2

However, due to fringe effects, the lateral (side wall) capacitance does not scale as effectively as the ground capacitance, and the actual scaling factor may be S^n, where n is between one and two [40]. Figure 1.2 plots the ratio of coupling to substrate capacitance. Note that the ratio is predicted to be around 4:1 by the year 2001 and rises to 6:1 by 2012.
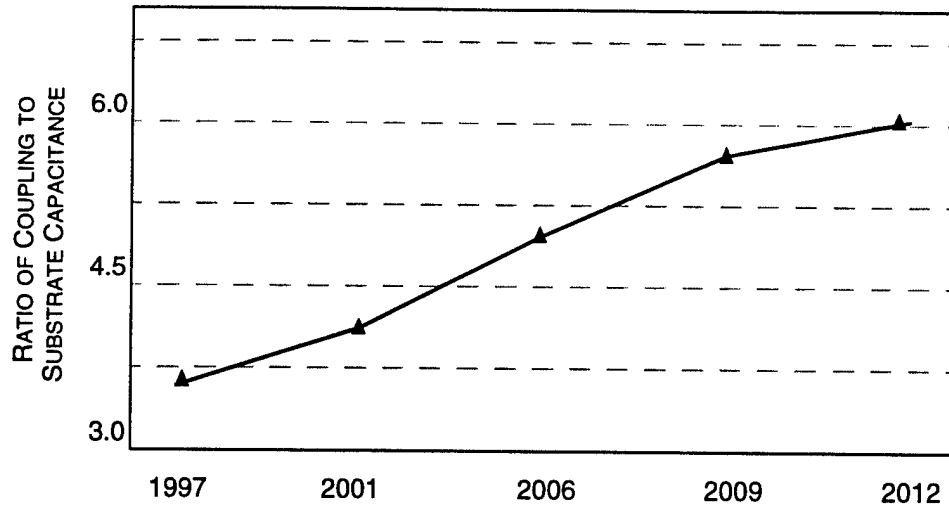
Figure 1.2   1998 SIA roadmap increase in capacitive crosstalk.

The rise in interconnect-resistance, due to interconnect scaling, is compensated by increasing interconnect-thickness. Thus an increase in the ratio of coupling to substrate capacitance is predicted.

To study the effect of technology scaling on interconnect delay, two coupled RC lines were scaled and simulated using SABER for each of the SIA-1998 projected technologies. The lines were coupled for $1000\mu m$ and driven by drivers of varying relative strength. The amplitude of noise, induced by crosstalk is measured when one signal is kept quiescent
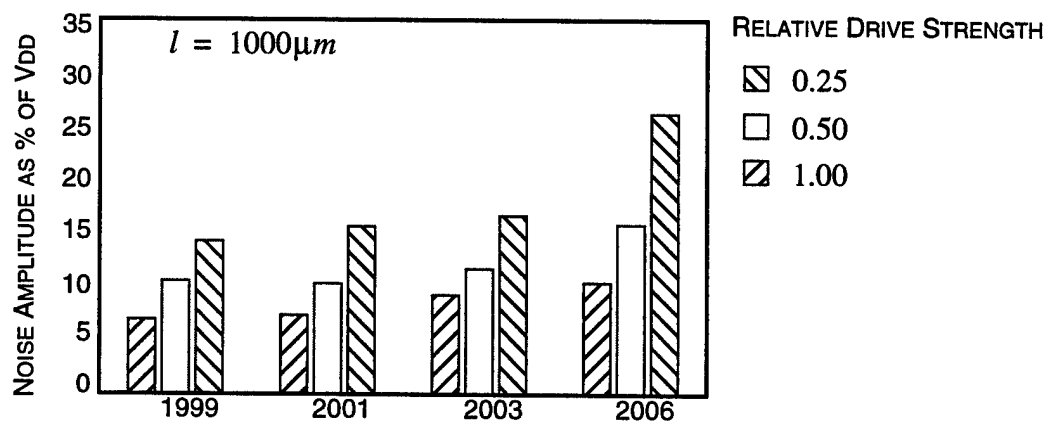
Figure 1.3   Crosstalk noise impact due to technology generations.

This chart was generated using the simulation methodology described in Section 2.3.1 on page 43. The increase in noise-amplitude, over the projected technologies, points to the severity of crosstalk.
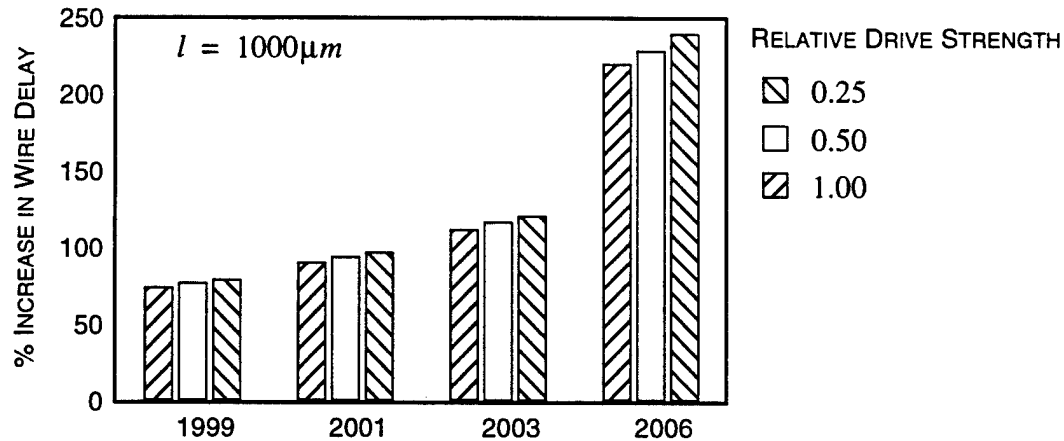
Figure 1.4   Crosstalk delay impact due to technology generations.

This chart was generated using the simulation methodology described in Section 2.3.1 on page 43. The increase in crosstalk-induced wire delay, over the projected technologies, points to the severity of crosstalk.

during the other's transition. By comparing their coupled delays, when driven by opposing waveforms, to the delay of a signal without coupling, the change in delay due to coupling is determined. The simulated results, shown in Figure 1.3 and Figure 1.4, show the impact of coupling doubling by the next decade. Thus, interconnect behavior will become strongly dependent on the switching activity of adjacent conductors. The unpredictable dependence on the physical location of wires will increase the number of timing violations. This points towards the severity of crosstalk and the need to address it.

1.4   The Current State of CAD

Traditional physical design methodologies are centered around the cornerstones of CAD: placement and routing. Both are driven by the desire to minimize die size. Typical constraints on a design include timing requirements and power budgets. The incessant march of technology has increased the importance of these constraints to the point where they now enjoy greater importance than the original objective of minimizing area. Timing convergence has become the singular driving (and selling) force for most methodologies.

A design will not be fabricated until it has met its timing requirements; this is true for ASIC's and high-performance microprocessors alike.

To avoid the need for achieving timing-convergence through multiple design-iterations, major efforts in synthesis and post-route optimizations have been made. As shown in Figure 1.5, these include wire-modeling, performance driven buffering, RC-tree modeling, wire-sizing, performance driven routing and numerous other schemes. Wire-modeling during synthesis is critical to achieving timing convergence. An inaccurate model will require resizing or even re-synthesis. Again, during placement, performance driven slack allocation schemes attempt to drive a placement towards timing convergence. However, the validity of the solution is again determined by the wiring load model. Finally, a design will not achieve timing convergence, if the assumptions made by the placement and synthesis tools are not satisfied during detailed routing. Post-optimizations such as wire-sizing, repeater insertion and buffer sizing help direct a design towards timing convergence. However, the lack of attention paid to crosstalk can invalidate the aforementioned wire models and cause these steps to iterate with no guarantee of convergence.

In the following section, a critical analysis of recent place and route algorithms is given. The merits and concerns with the various methodologies are listed. Finally, an analysis of the CAD methodology implemented for the University of Michigan high-performance microprocessor project (PUMA) is presented.

1.4.1 Placement

Placement assigns physical locations for the instances of a netlist while minimizing wiring cost. Violated routing-congestion and timing-constraints are the primary causes of iteration during placement. For a given netlist, the total active area is
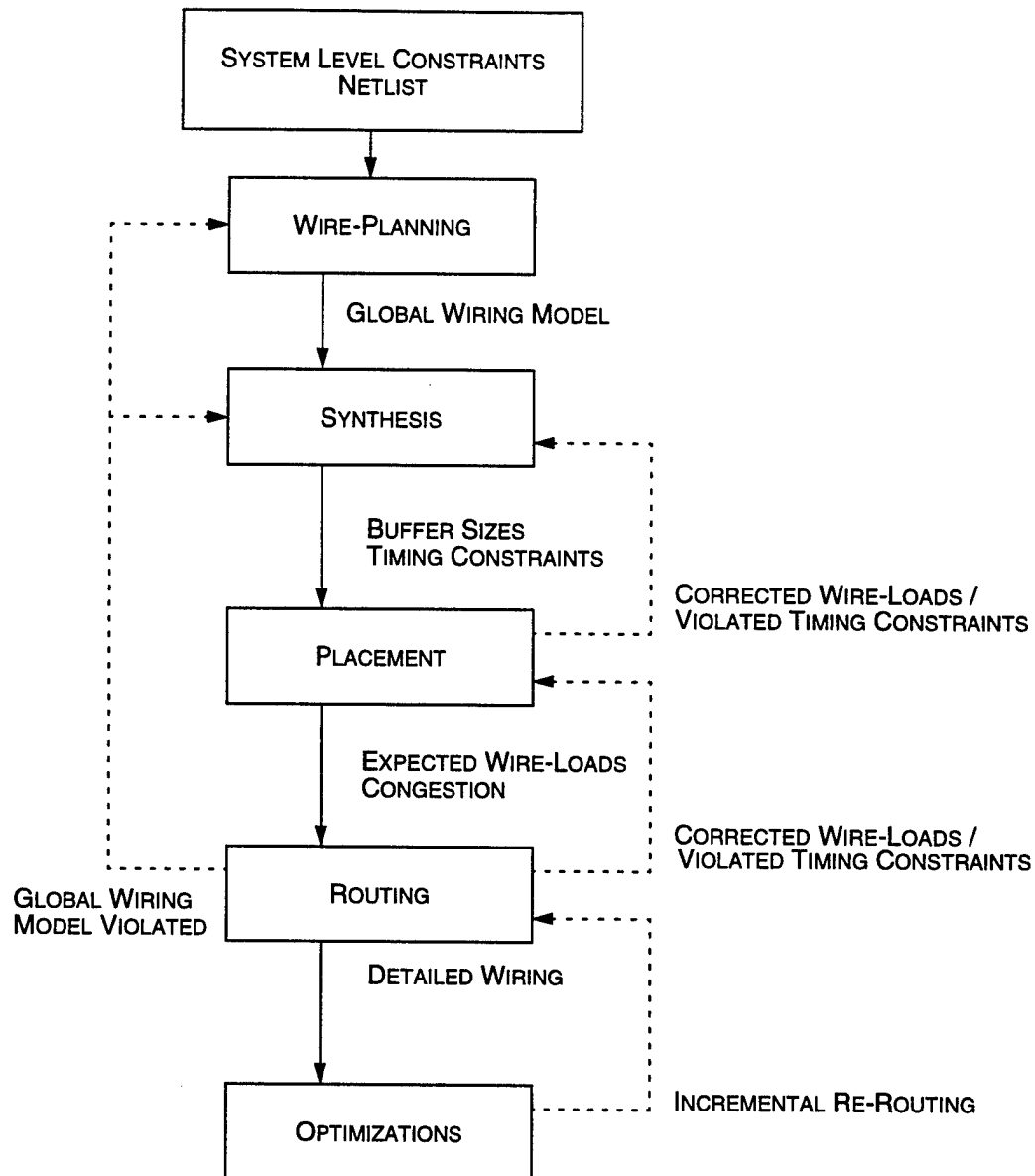
Figure 1.5   Typical CAD design methodology with iterative flow.

Iterations are indicated by dashed lines. The cause of the iteration is indicated adjacent to the line. Each iteration serves to improve the wiring-model and thus guide the design to meet timing specifications.

fixed, providing a lower bound for a placement expression. Any increase in area from this minimum occurs due to routing. For example, in a standard-cell centered design methodology, channel heights directly impact the total design area. Therefore, the objective of most placement algorithms is to reduce the total interconnect length. In the general sense, reducing wire-length lessens the demand imposed on routing resources,

REPEATEDLY PARTITION

WHILE MINIMIZING NETCUT
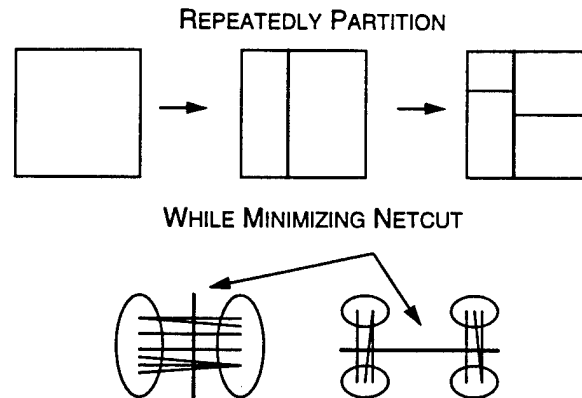
Figure 1.6   Constructive placement minimizing netcut cost.

A net-based graph is partitioned by minimizing its cut-cost. This results in two partitions. Again, using a minimum cut-cost metric, each is independently partitioned to produce a total of four partitions. This process continues until each partition contains one node of the graph.

thereby reducing area. The lack of information related to wiring supply will require iterations to satisfy congestion. Performance-driven placement addresses the critical path issue by weighting nets with area and timing costs [37, 52, 138]. Critical to these methodologies are the ability to predict a route topology for a given net and then estimate its delay. It is obvious that improper abstractions will not aid in timing convergence.

Most placement schemes are classified as either constructive (net based) or iterative (path based). Constructive algorithms start with an unplaced netlist and then build towards a complete placement expression. Cluster-growth and partition-based placement algorithms are examples of constructive placement schemes [130,75,6]. They typically make use of a net-based graph and attempt to minimize the netcut cost (Figure 1.6). This allows the cost-to-route (area) from one sub-block to the next to be minimized. For the general case, this cost is a function of the number of partitions it spaned by a net. Typical enhancements to this function include modeling absorption cost, clique cost and quadratic cost. The various net models used by these placers epitomize the heuristic nature of this process. Unfortunately, these cost functions are poor abstractions of the rectilinear tree

generated by a global and detailed router. An improved cost function is a minimum spanning tree approximation [80].

The concept of placing highly connected blocks within a partition was originally formulated for PCB designs. It is understood that, placing highly connected blocks together improves placement quality by reducing the connection cost, making netcut cost an abstraction of wiring for placement. However, an unconstrained netcut cost that places highly connected blocks together without supply-demand metrics will increase congestion and the burden on the router. Various devices such as quadrisection and terminal propagation have been developed to better model the creation of partitions and direct the netcuts. Suaris, et al. [136], developed the first quadrisection scheme by partitioning a network along two perpendicular cut lines, into four parts of predetermined sizes such that a cost function based on the crossing patterns is minimized. However, this cost function involved only the cost of the current partition level and failed to address this cost globally. Terminal propagation developed by Dunlop, et. al. [38], allowed the placement of modules into partitions to be influenced by external connections or IO pads. This approach, which adds dummy cells at fixed locations to influence the inclusion of cells within a partition, appears to be crucial in reducing congestion with netcut based schemes. Furthermore, the quality of the final routed solution is similar to the iterative schemes at a much lower computational cost.

As opposed to constructive schemes, iterative algorithms approach placement globally. Figure 1.7 depicts an iterative algorithm for incrementally updating a complete placement expression to minimize a cost function. Placement engines use annealing, genetic algorithms or numerical methods to minimize the cost function. Total wire-length is widely accepted as a metric to guide iterative placement algorithms. Reducing the global wire length helps reduce the total wiring demand. However, it does not account for wiring supply, and may exacerbate the local demand/supply ratio. It is common for a
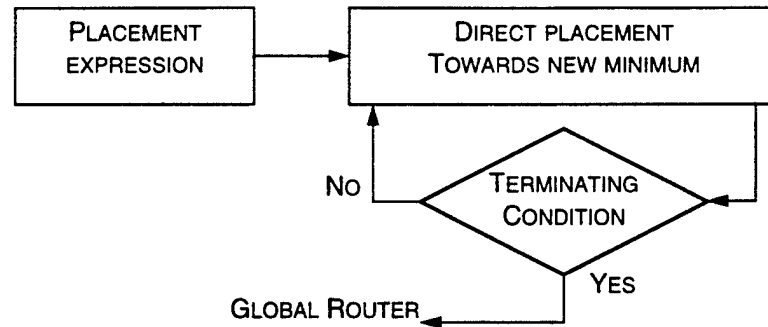
Figure 1.7   Iterative placement.

An iterative placement algorithm will repeatedly modify the current placement until a terminating condition is met. Each modification is directed towards improving a global objective function.

minimum wire length approach to require more tracks for routing through a region than are available or necessary. This is because total wire-length only models half of the congestion equation (demand) and does not model the actual behavior of the router. The quadratic placement solver GORDIAN, by Sigl et al. [68], uses numerical methods to solve sets of constrained linear equations. These linear equations represent the squared wire-length metric $(\Delta x^2 + \Delta y^2)$ constrained by center-of-gravity equations. Despite their global outlook on wire-length quadratic placers are unable to account for congestion. Quadratic placement is currently the industry norm for placement dealing with 100K+ instances and is frequently used as a seed for low temperature annealers that reduce local congestion [17].

The current problem in determining placement stems from the fact that most algorithms fail to correctly account for routing resources. With small examples, a wire length model might suffice, as the strain on resources is minimal. However, with the rising size of netlists, this model fails to achieve minimal area because it incorrectly accounts for routing resources. Placement engines driven by an accurate wiring model can better address congestion. The challenge lies in modeling the router, since the inclusion of a detailed route algorithm within the inner-loop of a placement algorithm is impractical. Innovations, and compromises, have lead to methods that permit routing during

placement. Mayrhofer et al [80], apply a multi-partitioning heuristic to solve congestion. After assigning sets of cells to partitions, wire assignment using Steiner trees [ref] is performed. By pre-computing the Steiner trees, they were able to trivially evaluate the cost of the partition. Furthermore, by constraining the assignment, they substantially reduced and equalized congestion. However, the time and space complexity needed to precompute the Steiner trees made the scheme unsuitable for generating more than 16 partitions. Huang, et al. [56], enhanced the original quadrisection-based placer by implementing a new general gain update scheme that captured detailed placement objectives on a per-net basis. By using a minimum spanning tree (MST) cost for each net, they were able to better represent the cost of each net. Of importance is the fact that they were able to globally place and route. The loose global route performed after each quadrisection was used to guide the next quadrisection, creating results competitive with the leading numerical solvers. This concept of updating the next iteration using the previous iterations' congestion data is used within the congestion-driven quadratic placement algorithm presented later.

Route estimation is crucial to placement. Most methods mentioned above, lacked an adequate expression for the physical manifestation of routing. This leads to a lack of understanding, prediction and control of congestion, and eventually to crosstalk hazards.The placement paradigm which makes routing a function of placements not valid. The need to drive placement with an accurate cost of routing congestion is evident.

1.4.2 Routing

As illustrated in Figure 1.5, routing has traditionally followed placement. Routing determines the physical structure of the wires that connect the placed cells. The complexity of routing is exemplified by the multiple cost components in a typical objective function. These components include minimizing routing area and maximizing

utilization of active area, minimizing wire-length, via-count, number of layers and guaranteeing signal delay. Furthermore, the final routing is required to guarantee electrical and design rule correctness.

The predominant methods for routing are two-phased and area routing. The classical two-phased approach is accomplished by dividing the problem into global and detailed routing, each having no influence on placement. Global routing determines the behavior of each net's path through a coarse grid overlaying a given placement, while satisfying constraints such as total wire length, congestion and delay [98, 70]. This step approximates the final routing and is typically implemented in a hierarchical manner (Figure 1.8). Non-hierarchical Integer Linear Program (ILP) implementations of global routing can produce optimal congestion-constrained global route solutions for small netlists. Wang et al [29], demonstrate the concept of supply and demand as a means of driving global routers to satisfy congestion by maximizing route uniformity and minimizing maximum route density.

Detailed routers define the structure of routes, by assigning nets to tracks and layers. The classical grided router makes use of (grided) tracks to transform the task of routing from the geometrical domain to a graph domain [72]. Typically the task of routing is subdivided into simpler instances of switch box or channel routing with the global

LEVEL 1   LEVEL 2   LEVEL 3
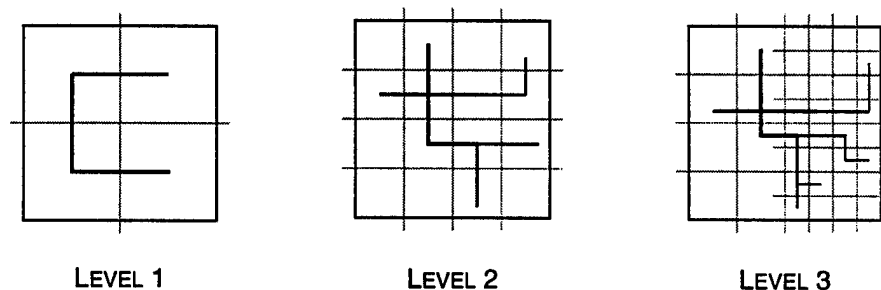
Figure 1.8   Hierarchical Global Routing.

A hierarchal global router will generate global routes on a coarse grid. Each cell of the grid is individually subdivided and its routes are refined. The process terminates at a predetermined level.
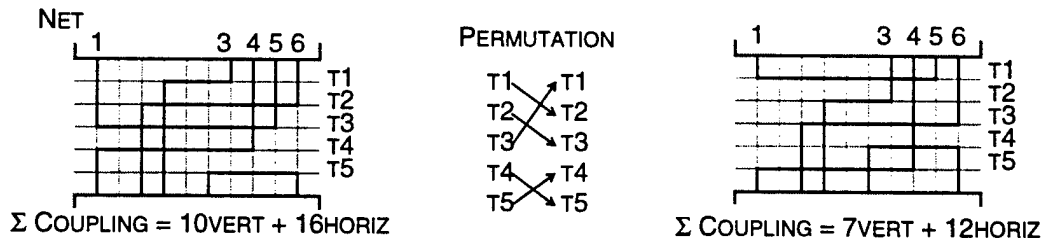
Figure 1.9 Channel Crosstalk Optimiser: Sample permutation.

By permuting the original channel assignment, the total adjacent capacitance for the horizontal tracks is reduced by four units. The total adjacent capacitance for the vertical tracks is reduced by three units.

routing paths used for guidance. A detailed router's singular purpose is to create electrical connections between two points. However, the complexity of this objective makes the task of including other objectives such as crosstalk minimization quite daunting. If the global routing graph is made to represent the routing region in complete detail, then the global routing algorithm becomes an area router. This allows an area router to behave both like a global and a detailed router. However, the memory requirements of the routing grid can make area routing impractical.

Since wire adjacencies and signal transitions generate crosstalk, most methods for resolving crosstalk occur concurrent to, or after, detailed routing. Current crosstalk minimization schemes for IC routing have been formulated predominantly for specific route architectures. In the case of grided channel routers, Jhang, et al. [64, 63] modeled the problem as routing track reassignment (permutations) to minimize adjacency (Figure 1.9). By defining slack as the difference between the maximum allowable coupling length and the coupling length of the net being routed, they transformed the problem into one of maximizing the minimum slack, while forcing it to be positive. They showed track permutation to be NP-complete and made use of a mixed ILP solver to deduce the optimal permutation. Unfortunately, this form of crosstalk minimization may call for an increase in the number of tracks (routing area) to achieve a valid solution. Furthermore, the authors

assume that all coupling is of similar quality (invariant with signal arrival time). Therefore, their process cannot be used to satisfy timing constraints.

Other approaches to reducing crosstalk involve post processing the layout to space wires apart [47]. Onozawa et al, define "spacing" as a technique to adjust the distances between wire segments to improve the performance on a critical-path while keeping the layout design-rule correct and minimally increasing the area [88]. Figure 1.10 illustrates the process on a contrived set of routes. This approach will only be successful if the routing demand is less than the supply. If not, a trade-off between area and delay must be made. To enhance the flexibility of long routes, Onozawa inserted jogs but achieved only marginal improvements with substantial increase in computation time.

Layer reassignment is another valid approach to minimizing crosstalk, as demonstrated by Thakur et al [104]. It is unknown whether their scheme could be extended to area-routers; if so, this could be a fruitful avenue of research. Dutta et al, formulated routing as an optimization problem [99]. They departed from algorithms tailored toward specific routing styles, which allowed a flexibility that permitted complex cost functions. Using an annealer, they were capable of addressing crosstalk while routing. This method is attractive in that it can model arbitrary functions, but suffers from all the shortcomings of using an annealer to minimize a non-linear objective.
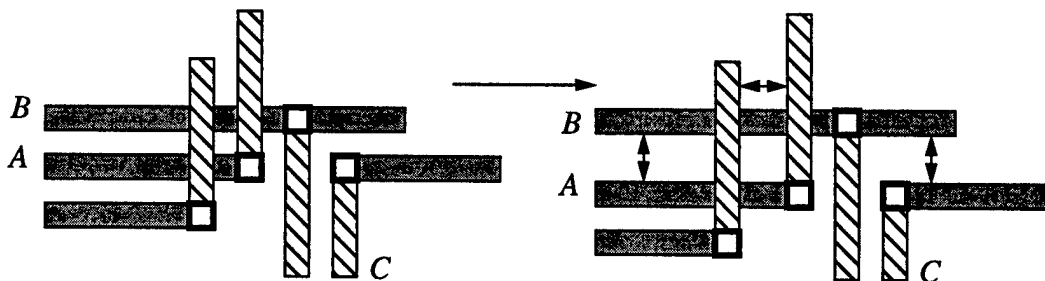
Figure 1.10   Spacing wires to minimize crosstalk.

Spacing apart wires *A*, *B* and *C*, reduces their coupling (adjacent) capacitance. The physical ordering of the routes is fixed.

Xue et al, demonstrated a crosstalk aware global router [116]. They were able to modify routes at the global level while satisfying congestion and crosstalk constraints. However, they used an over simplified model for the crosstalk that would fail to accurately drive the global router. Furthermore, they attempted to satisfy crosstalk constraints by suggesting a physical ordering of the nets. This may over constrain a detailed router and will cause the router to not complete.

A unique approach to addressing crosstalk, presented by D. P. LaPotin [32], made use of expected crosstalk behavior for an MCM global router. Overall coupled noise was minimized by selecting regions with lower expected activity. Furthermore, by dividing each physical region into "time buckets," the analysis was able to include temporal locality. It has served to provide a basis of the global route embedder to be presented later in this work.

1.4.3 Interconnect Modeling

Timing analyzers (for mid-frequencies) using RC trees can model physical interconnect to within 10% of a SPICE simulation at 1000x the speed. However, these models break down for high frequency DSM technologies, where physical interconnect delay dominates. Asymptotic Waveform Evaluation (AWE) was developed to accurately model the transient responses of RLC networks by matching initial boundary conditions and the first $2q - 1$ moments of the exact response of the tree to a lower order q-pole model in the Laplace (s) domain [97]. While the modeling and numerical methods employed for AWE provide accuracy, they are cumbersome for "on demand" timing analysis. Reduced moment models, derived from AWE theory, provide an excellent trade-off of delay with accuracy.

Many attempts have been made to model capacitive crosstalk and coupling effects. Efforts have ranged from analytical models to simplified SPICE simulators. Moll et al,

presented a simplified model for crosstalk amplitude, due to capacitive coupling, between two adjacent conductors [82]. Ironically, its shortcoming serves to define the requirements of a crosstalk aware timing model. The model must account for driver impedance, signal arrival time and edge rates. The model should treat the coupled conductors as distributed coupled RC lines and determine both the delay and noise impact of coupling. Chandramouli et al, introduced a wire-delay model for coupled, distributed URC (uniform-RC) transmission lines which provided the required accuracy, at far lower computational cost, than standard moment matching based wire-delay approximations [87]. The model confirmed that the value of temporal separation causing minimum delay is a function of line length, input transition time, edge rate ratios and receiver load. The model was accurate to within 5% of SPICE simulations. Recently, models for approximating the impact of coupling capacitance on interconnect and gate delays have been developed. Dartu, et. al, present a gate level model that predicts bounds on noise and wire-delay changes for general RC loading conditions [40]. This is achieved by extending the "effective capacitance" problem to include coupling capacitances.

Current metallizations and advanced DSM technologies make use of five or more routing layers. This allows three or four layers to be used for routing and the rest for power, clock and bus route topologies. The capacitive model used by most crosstalk driven methodologies would inaccurately estimate the parasitic capacitances of a multi-layer structure [84]. A solution for restructuring crosstalk to internal nodes and between nets, using a realistic model of these parasitic capacitances, will be required for the next generation of crosstalk minimizing routers.

1.4.4 The PUMA CAD Methodology

Researchers at the University of Michigan, working on the PowerPC Ultra-Fast MCM Architecture (PUMA) project, have developed three microprocessors of varying
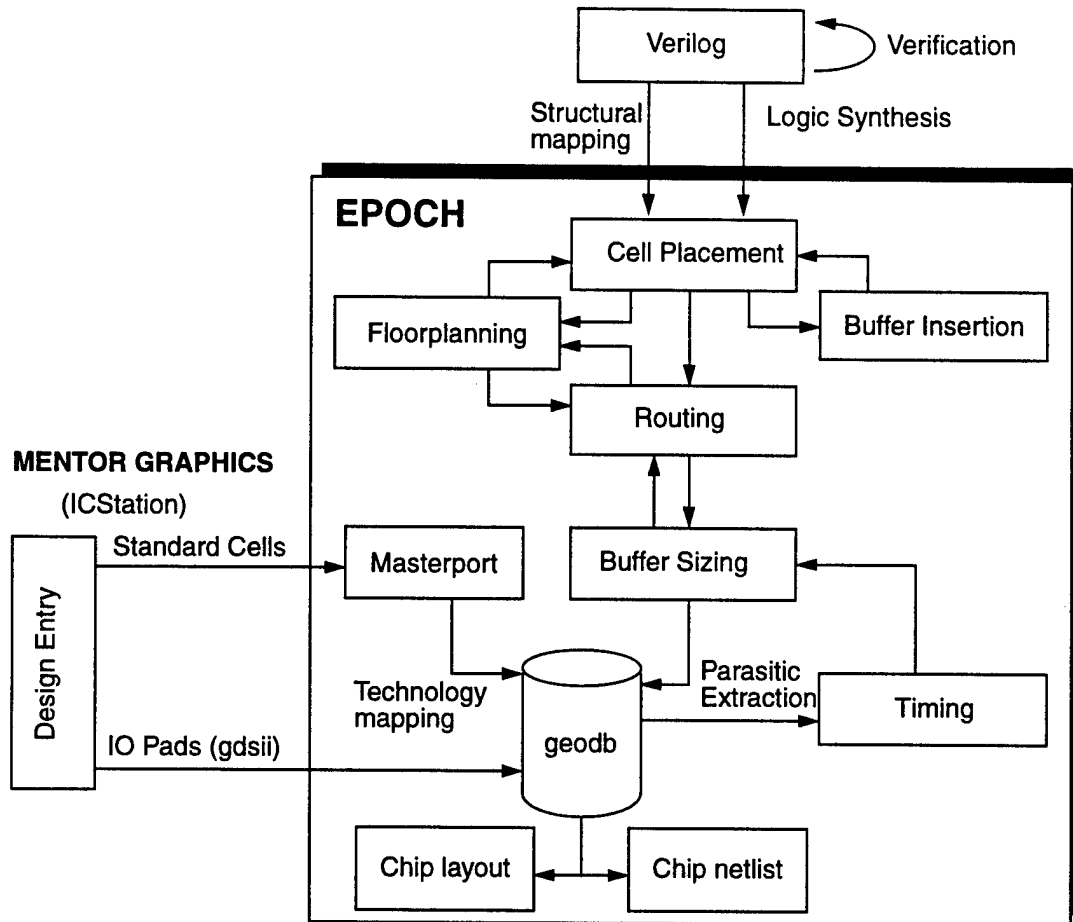
Figure 1.11   PUMA IC physical design methodology using EPOCH.

Netlists are placed and routed in the EPOCH framework. Floorplanning between the place and route iterations permits modification of wiring resources. Performance-driven buffer-sizing is used to meet timing specifications.

complexity. The design methodology, shown in Figure 1.11, used to implement these processors is fairly representative of a typical ASIC flow. Of relevance, to this chapter, is the physical design flow through Placement and Routing. To achieve timing convergence, the methodology supports timing driven placement and performance driven buffering. Both procedures are initiated after detailed routing. Thus a design, by default, will iterate at least once. The design algorithms are geared towards route completion. Frequently this occurs at the expense of increased route-area. A floorplanner, invoked after placement, can

be used to change and allocate wiring resources. This path is frequently utilized to minimize the routing cost. The methodology does not address crosstalk.

1.5 Summary

The current generation of VLSI process technologies have invalidated some of the foundational assumptions for CAD. Historically second-order wiring effects, such as crosstalk-delay and noise, need to be accounted for throughout the physical design flow. Timing errors and logical upsets due to crosstalk point to the severity of this problem and in particular, the need for a solution. Upcoming process technologies are driving towards tighter metallization, increased clock frequencies, smaller voltage swings and longer interconnect. Each of these trends will increase the impact of capacitive crosstalk. Estimates on coupling show this crosstalk impact will double by the next decade. Current generations of CAD tools achieve targeted interconnect reliability goals through back-end analysis and design iteration. The very nature of this solution will require an increased number of design iterations to satisfy timing constraints.

Traditional physical design methodologies attempt to minimize die size, power and design time. Of these, reducing design time (quick timing convergence) has become the singular driving (and selling) force for DSM methodologies. This chapter summarized various physical design algorithms and methodologies that attempt to achieve timing convergence. In most cases, unaccounted crosstalk constraints will cause these schemes to miss timing convergence. Current methodologies that do address crosstalk are iterative and inaccurately model crosstalk. Therefore, these methodologies can lead to timing failures.

In order to eliminate the need for multiple design iterations, a physical design methodology that accounts for crosstalk, with accurate and consistent estimates of wiring constraints throughout the design process, is required.

CHAPTER II

MODELING CROSSTALK

Aggressive scaling of interconnect pitch has led to increased coupling capacitance,
which has caused non-linear interactions in adjacent conductors. These are revealed by
delay variations, noise-pulses and non-monotonic waveforms. Accurate models of this
interaction are required to achieve timing convergence. This Chapter examines the
interaction between capacitively coupled conductors, and develops an empirical model for
crosstalk-impact. The model captures noise and delay changes on coupled conductors. It
can be linearized, making possible a trade-off between delay, noise and area while
mitigating crosstalk.

2.1 Overview

Figure 2.1 represents a lumped capacitance approximation of two parallel coupled
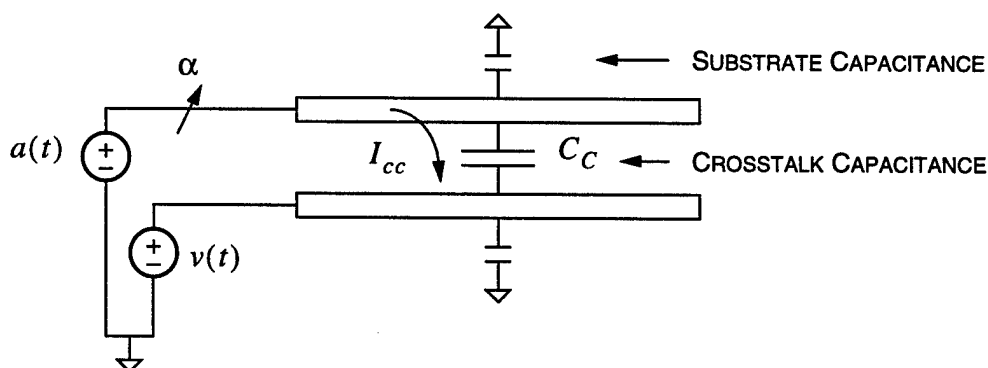
Figure 2.1   Crosstalk capacitance between parallel adjacent wires.

Term I_{cc} represents the current required to charge the crosstalk capacitance
between the adjacent conductors. This current influences the waveforms pro-
duced by $a(t)$ and $v(t)$.

lines. Pillage et al [96] state that opposite transitions with edge rate ratio α will cause the signals to become non-monotonic, thereby increasing delay. If the lines transition in the same direction (both rising or falling), a decrease in delay is observed. The default case, with only one line switching, has a delay between the former two cases. This change in delay (performance) is due to the increased amount of current (I_{cc}), required to drive the coupling capacitance (C_c)

$$I_{cc} = C_c \cdot \frac{dV_{cc}}{dt}. \qquad (2.1)$$

Where $\frac{dV_{cc}}{dt} = \frac{d}{dt}a(t) - \frac{d}{dt}v(t)$. Using Eq. 2.1, the opposite transition case requires $I_{cc} = C_c(\alpha + 1)$, the single transition requires less with $I_{cc} = C_c \cdot \alpha$, and the similar transition case requires the least current with $I_{cc} = C_c(\alpha - 1)$. The worst-case increase in delay is observed when crosstalk, due to opposite transitions, occurs in the high gain region of the drivers $a(t)$ and $v(t)$. It has been noted that a $vdd/10$ bump during high-gain coupling could produce a noticeable delay change [7]. Noise pulses will be observed if one of the lines is quiescent during the transition of the other. Intuitively, any change in the interfering waveforms is a function of the current needed to charge the coupling capacitance. In turn, this current is determined by the voltage across it and the geometry of the coupled conductors. Determining the impact of coupling requires knowledge of the temporal and physical proximity of the signals. The type of interaction (wire-delay change or noise), and its magnitude are dependent on the arrival time and edge rates of the interfering signals. This accounts for the $\frac{dV_{cc}}{dt}$ term in Eq. 2.1 and represents the temporal proximity of the signals. Physical proximity is represented by the coupling capacitance and is a function of the spacing and coupling length of the conductors.

A function for estimating crosstalk-impact would require as input, the coupled length and spacing of the adjacent conductors, their signal edge rates and signal arrival times to describe the range of coupling related possibilities. This relationship is difficult to capture

due to the interdependency of each signal upon the other. Furthermore, there could be multiple signals coupling with a given signal. The nature of this phenomenon makes the problem of determining the exact change in delay difficult.

There are multiple models for approximating crosstalk, ranging from the use of relative coupling factors [141] to simpler estimates using coupling length [47]. A common model adds twice the coplanar (adjacent line-to-line) capacitance to the ground capacitance, $2 \cdot C_c + C_s$, results in an inaccurate estimate of delay. This is because the system needs to account for the aggressor and victim input waveforms and interconnect resistance. For the same reasons, a charge sharing model ($\frac{C_c}{C_c + C_s}$) for noise fails. Chandramouli et. al [110], described a model for coupled wire delay in the presence of temporal proximity. The model quantified delay (due to coupling) as a function of driver impedance, signal proximity, coupling length and spacing. This model could not be easily linearized. Dartu et al [40], showed how to approximate the impact of crosstalk by extending the "effective capacitance" problem [41, 59] to include coupling capacitances. This is derived from the assertion that the added current needed to charge the coupling capacitance can be modeled by a capacitive load. While the model is capable of accurately modeling multiple coupled RC-trees, they do not present a method for obtaining waveform data for the internal nodes of a net.

A timing model for approximating the crosstalk-impact requires the following characteristics. First, it should compute the delay and noise impact between multiple coupled RC-tree structures. Second, it must account for the aggressor and victim waveforms. Finally, the model needs to be linearizable. This allows a linear solver to make the trade-off between physical and temporal proximity. Obviously, the model should be accurate and computationally efficient.

The analytical examination of coupled nodes by Moll et al only accounted for noise on a quiescent victim line [82]. The following section expands on that analysis to account for
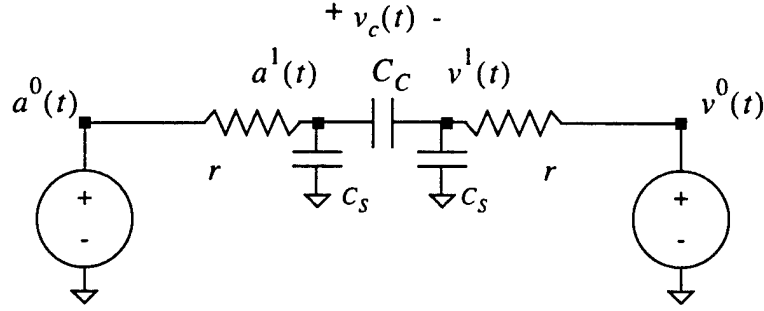
Figure 2.2   A lumped *RC* circuit model for capacitive crosstalk.

Each adjacent line Fig. 2.1 is approximated by a resistor r and substrate capacitance C_s. The lines are coupled by their adjacent capacitance C_c.

all crosstalk interaction (delay and noise). The next section details the derivation of an empirical model for crosstalk. The chapter concludes with a section on delay models for uncoupled *RC*-trees.

2.2   An Analysis of Capacitively Coupled Nodes

In Figure 2.2, $a^0(t)$ is the aggressor signal and $v^0(t)$ the victim. a^1 and v^1 are the coupled nodes. The aggressor is modeled as a rising exponential:

$$a^0(t) = 1 - e^{-t \cdot c}. \tag{2.2}$$

The positive constant c models the edge-rate of the aggressor. This is a departure from the generally used ramp or step signals. However, the exponential input has been noted to provide greater accuracy [110] and has the added advantage of being analytically malleable. The KCL equations for nodes a^1 and v^1, in the Laplace domain, generate the following system of equations:

$$\begin{bmatrix} s \cdot C_T + \dfrac{1}{r} & -s \cdot C_C \\[2mm] -s \cdot C_C & s \cdot C_T + \dfrac{1}{r} \end{bmatrix} \cdot \begin{bmatrix} A^1(s) \\[2mm] V^1(s) \end{bmatrix} = \begin{bmatrix} \dfrac{A^0(s)}{r} + C_S \cdot a^1(0) + C_C \cdot v_c(0) \\[2mm] \dfrac{V^0(s)}{r} + C_S \cdot v^1(0) - C_C \cdot v_c(0) \end{bmatrix} \tag{2.3}$$

where $v_c(t)$ is the voltage across the coupling capacitor C_C and $C_T = C_C + C_S$. Let

$$M = \begin{bmatrix} s \cdot C_T + \dfrac{1}{r} & -s \cdot C_C \\ -s \cdot C_C & s \cdot C_T + \dfrac{1}{r} \end{bmatrix}. \tag{2.4}$$

Using Cramer's rules [127], the following expression for $A^1(s)$ is obtained:

$$A^1(s) = \frac{\begin{vmatrix} \dfrac{A^0(s)}{r} + C_S \cdot a^1(0) + C_C \cdot v_c(0) & -s \cdot C_C \\ \dfrac{V^0(s)}{r} + C_S \cdot v^1(0) - C_C \cdot v_c(0) & s \cdot C_T + \dfrac{1}{r} \end{vmatrix}}{|M|}. \tag{2.5}$$

Similarly,

$$V^1(s) = \frac{\begin{vmatrix} s \cdot C_T & \dfrac{A^0(s)}{r} + C_S \cdot a^1(0) + C_C \cdot v_c(0) \\ -s \cdot C_C & \dfrac{V^0(s)}{r} + C_S \cdot v^1(0) - C_C \cdot v_c(0) \end{vmatrix}}{|M|}. \tag{2.6}$$

Defining

$$H(s) = V^1(s)/A^1(s) \tag{2.7}$$

to be the transfer function for modeling the impact of the aggressor on the victim permits the analysis of the four different interactions between $a^1(t)$ and $v^1(t)$. The interactions include

1. Coupling to a quiescent victim,
2. Same direction coupling,
3. Opposite direction coupling, and
4. Non-monotonic coupling.
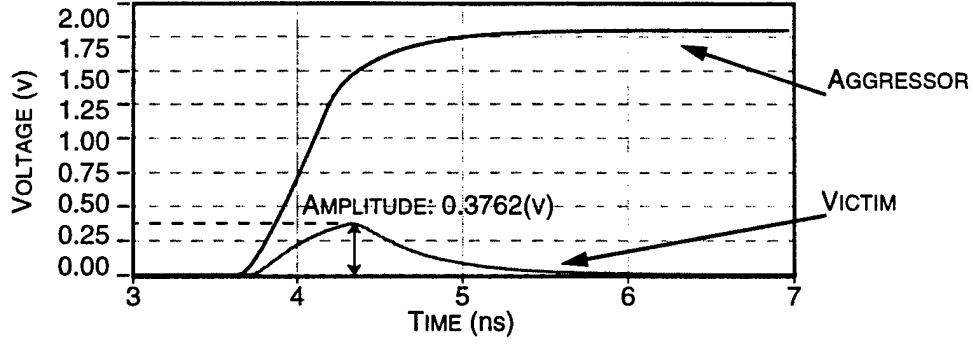
Figure 2.3 Simulation of a noise pulse on a quiescent victim line.

The circuit in Figure 2.2 is simulated in SABER. The values of r, C_C and C_S model a lumped approximation to 3000μm of SIA-1999 coupled interconnect. The victim line is held at ground potential. The transition on the aggressor induces a noise-pulse through the coupling capacitance.

2.2.1 Quiescent Victim

With a quiescent victim signal, $v^0(t) = 0$, and $a^1(0) = 0$, Eq. 2.7 reduces to

$$H(s) = \frac{s \cdot C_C \cdot r}{s \cdot r \cdot C_T + 1} \tag{2.8}$$

The inverse laplace transform of $H(s) \cdot A^1(s)$ produces (Figure 2.3):

$$v^1(t) = \frac{r \cdot c \cdot C_C \cdot \left(e^{-\frac{t}{\tau}} - e^{-t \cdot c} \right)}{1 - c \cdot \tau} \tag{2.9}$$

where $\tau = r \cdot C_T$. The peak occurs when the derivative of $v^1(t)$ is equal to zero. The value of the peak can be obtained by substituting $t = \frac{\tau \cdot \ln(c \cdot \tau)}{c \cdot \tau - 1}$ into Eq. 2.9.

2.2.2 Identical Signals

When $v(t) = a(t)$, $H(s)$ becomes the identity function as expected, and therefore, the signals do not interfere. However, since $v_c(t) = 0$, a reduction in signal
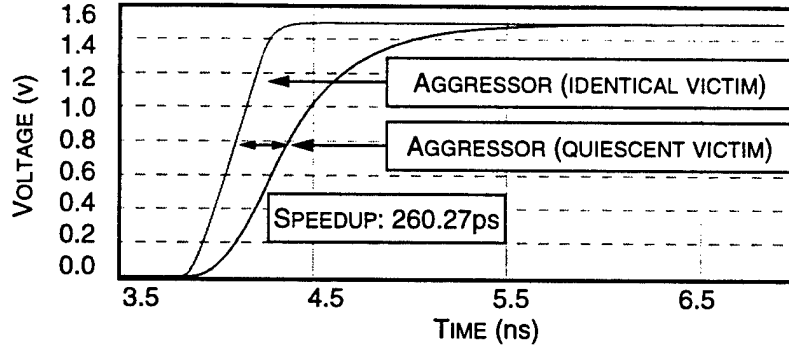
Figure 2.4   Signal speedup with identical signals on coupled lines.

The circuit in Figure 2.2 is simulated in SABER. The values of r, C_C and C_S model a lumped approximation to 3000μm of SIA-1999 coupled interconnect. The victim and the aggressor simultaneously transition in the same direction. A reduction in delay is observed because no current is required to charge the coupling capacitance.

delay from the general case is observed (Figure 2.4). This reflects the fact that $a^0(t)$ and $v^0(t)$ need charge only their substrate capacitances.

2.2.3 Opposing Signals

If $v(t) = 1 - e^{-c \cdot t}$, and $a(t) = 1 - v(t)$, $V^1(s)$ from Eq. 2.6 can be inverse transformed to:

$$v^1(t) = \frac{(1 - e^{-t \cdot c}) + c \cdot \tau \cdot \left(1 - e^{-\frac{t}{\tau}}\right)}{1 - c \cdot \tau} \tag{2.10}$$

where $\tau = r(C_T + C_S)$. For present-day VLSI technologies, $1 - c \cdot \tau \cong 1$. Therefore Eq. 2.10 reduces to a form that shows $v^1(t)$ "delayed" by $c \cdot \tau$ (Figure 2.5).

2.2.4 Non-Monotonic Waveforms

In the preceding equations, $v(t)$ and $a(t)$ possessed constant output impedance. This is not the case with actual drivers. A signal's driver will transition through a high-gain region where its output impedance may increase by orders of magnitude [115]. If a
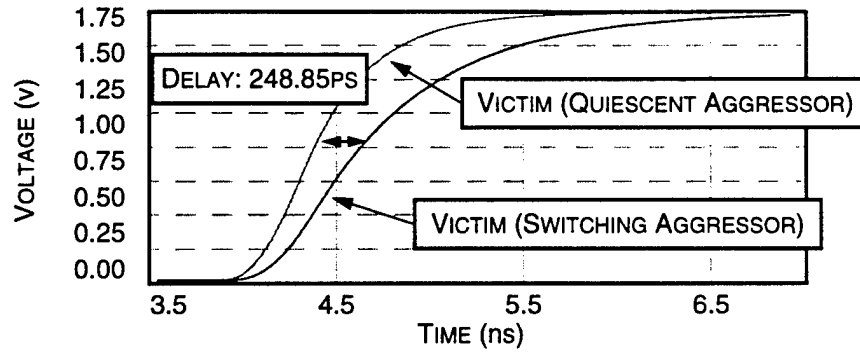
Figure 2.5   Delay increase with opposing signals on coupled lines.

The circuit in Figure 2.2 is simulated in SABER. The values of r, C_C and C_S model a lumped approximation to 3000μm of SIA-1999 coupled interconnect. The victim and the aggressor simultaneously transition in opposite directions. The current required to charge the coupling capacitance results in a delay increase.

fast aggressor interferes with a victim line during such a transition, then the victim signal becomes non-monotonic (Figure 2.6). A designer can avoid the need to account for this scenario by choosing appropriate driver strengths and wiring rules [32].

2.2.5   Summary of the Coupled Node Model

The previous model has shortcomings that result from two dominating simplifications. The first simplification eliminated the component of arrival time that is
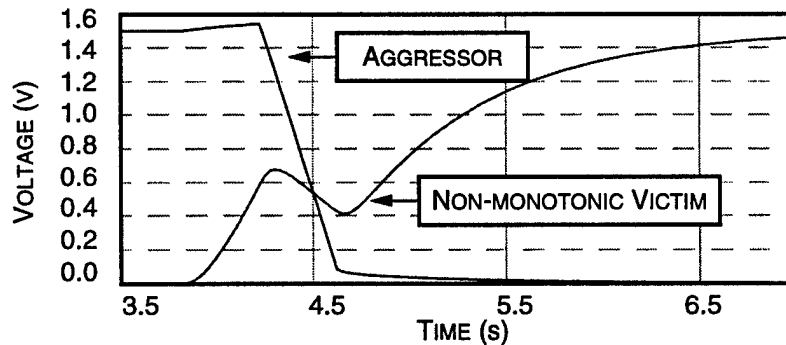
Figure 2.6   Simulation of non-monotonic waveforms on coupled lines.

The circuit in Figure 2.2 is simulated in SABER. The values of r, C_C and C_S model a lumped approximation to 3000μm of SIA-1999 coupled interconnect. The victim and the aggressor simultaneously transition in opposite directions. The weaker victim-driver is unable to provide sufficient current to overcome the rapid transition of the aggressor.

crucial to crosstalk. Experiments in the next section indicate that crosstalk-induced wire-delay varies significantly with temporal separation. Secondly, the model uses a lumped RC approximation for what should be modeled as a distributed RC circuit.

The next section addresses these shortcomings and derives an empirical model for crosstalk.

2.3 Empirical Derivation

Section 2.2.3 suggested that the delay change due to coupling would contain an $R \cdot (C_S + C_T)$ time constant. This can be factored into:

$$R \cdot (2 \cdot C_S + C_C) = \frac{\rho \cdot l}{w \cdot t} \cdot \left(2 \cdot \frac{\varepsilon_\kappa \cdot l \cdot w}{h} + \frac{\varepsilon_\kappa \cdot l \cdot t}{s} \right),$$
(2.11)

where ρ is resistivity of the conductor, ε_κ is the dielectric constant, and w, t, h are the conductor's width, thickness and separation from the substrate, respectively. Terms l and s, represent the coupled length and spacing. Defining $h = $ constant $\cdot s$ suggests an l^2/s form for curve fitting crosstalk. Devgan et al [34, 33] showed with a distributed line model that the maximum induced noise, due to crosstalk could be approximated by:

$$\zeta_i = R_i \cdot I_c + \zeta_{i-1}.$$
(2.12)

Where ζ_i is the amplitude of the induced noise at node i, R_i is the resistance driving that node and I_c is the total crosstalk current flowing into i and ζ_{i-1} is the amplitude of the induced noise at node $i-1$. The dependence of ζ_i on ζ_{i-1} accounts for multiple noise sources on a given conductor. A transformation similar to Eq. 2.11, yields the l^2/s form.

The following sections present a set of experiments designed to determine the fitness of the l^2/s form.
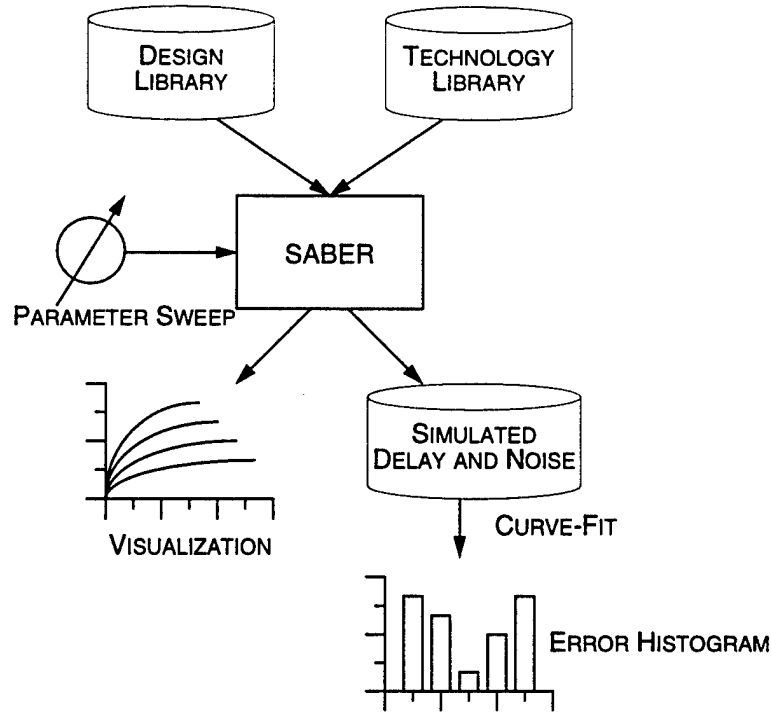
Figure 2.7   Test-bed to investigate crosstalk.

The test bed uses SABER to simulate various coupling-related events. Measured results from these simulations are fitted to functions by satisfying a least-squares formulation. The resultant error distribution determines the appropriateness of the specified functions.

2.3.1   Simulating Crosstalk-induced Delay and Noise

Figure 2.7 illustrates the interaction between the various components of the test bench. The use of the circuit simulator, SABER [129], is central to the functional flexibility of the test setup. First, a coupled distributed-RC circuit with specified design and technology parameters is simulated through SABER. Next, the change in wire delay is extracted and reported. In the case of parameter sweeps, this output is curve-fitted to a specified function by satisfying a least squares formulation. Figure 2.8 illustrates the test bed circuit and its parameters used to simulate crosstalk. Each coupled line is built out of 40 RC components. As in the previous sections, a is defined to be the aggressor signal and v as the victim. Terms r_a and r_v are the aggressor and victim driver resistances.
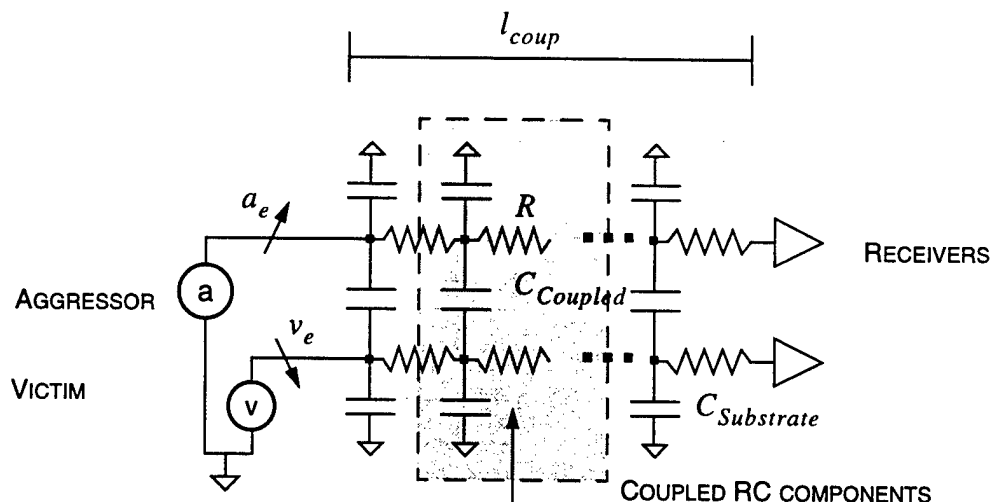
Figure 2.8 Circuit and parameters for simulating crosstalk.

Two distributed RC lines are coupled and driven to simulate coupling-related events. Each line is composed of 40 RC components which can be scaled to appropriately model a given coupled wire-length (l_{coup}).

terms a_e and v_e are their edge-rates, with a_e constrained to be greater than v_e. a and v are 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 coupled wire-length of the signals, were swept through the ranges specified in Table 2.1.

Table 2.1 Crosstalk experiment parameters.

Ranges for arrival times, edge rates, spacing, relative driver strength and coupled wire-length to simulate various coupling events are listed below.

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

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

$$\delta a = \frac{a_a - v_a}{0.8 \cdot v_e}$$

(2.13)

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. Figure 2.9 shows the change in wire delay as a function of δa. The y axis of these contour maps represent coupled length. Through the contour maps, one concludes that wire-delay impact diminishes as a function of increasing $|\delta a|$. Furthermore, the magnitude of this impact is dependent on the strength of the aggressor as shown by the ratio of the edges rates (a_e/v_e). For example, on a $2000\mu m$ wire, a 12% change in δa results in a 50% change in wire-delay when the edge rates of the interfering signals are equal. However, a 25% change in δa is required to induce a 50% change in wire-delay when $a_e/v_e = 0.25$.

For $\delta a = 0$, Figure 2.10 illustrates the change in wire-delay τ, as a function of coupled length and spacing. It is obvious that the relationship with l and s is monotonic and non-linear. When the edges of the signals do not overlap ($|\delta a| \geq 0.5$), the delay is not affected, so instead of the delay, the amplitude of the induced noise ζ, on the quiescent victim line, is measured at the receiver (Figure 2.11). Empirical analysis of the delay change (τ), and amplitude of the noise pulse (ζ), due to coupling, suggests the following equations:

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

(2.14)
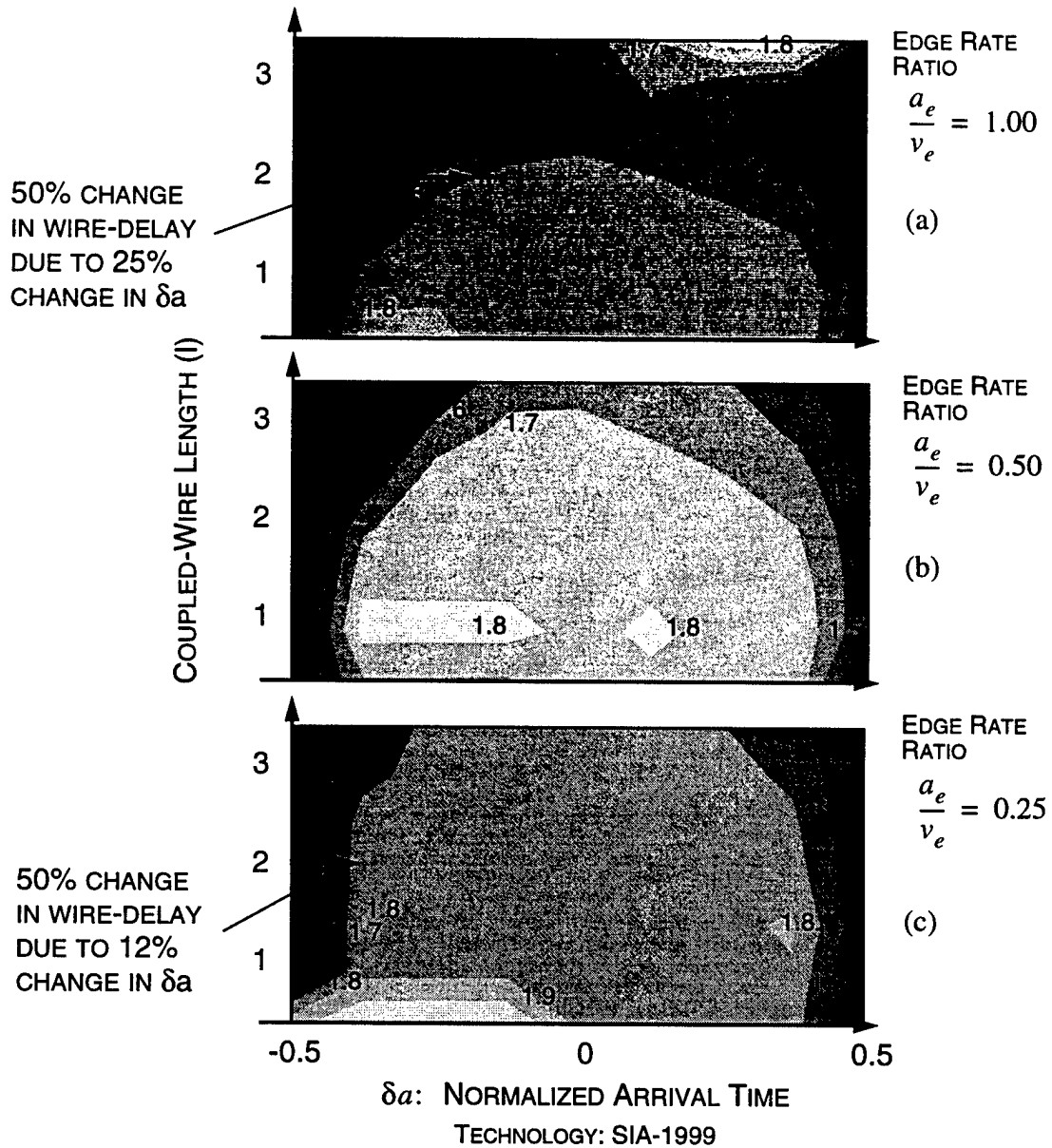
Figure 2.9 The impact of arrival time on wire-delay of a victim line.

The circuit in Figure 2.8 is simulated by SABER with the SIA-1998 technology parameters for 1999. The contour maps depict the change in wire delay as a function of temporal proximity and coupled wire-length. The numbers on the map represent the magnitude of the change in wire-delay. Simulation data for three different aggressor-victim edge-rate ratios are presented in (a), (b) and (c). Decreasing the edge-rate ratios increases the change in wire-delay. The effect of temporal proximity is most severe at the center ($\delta a = 0$), and reduces towards either extreme.
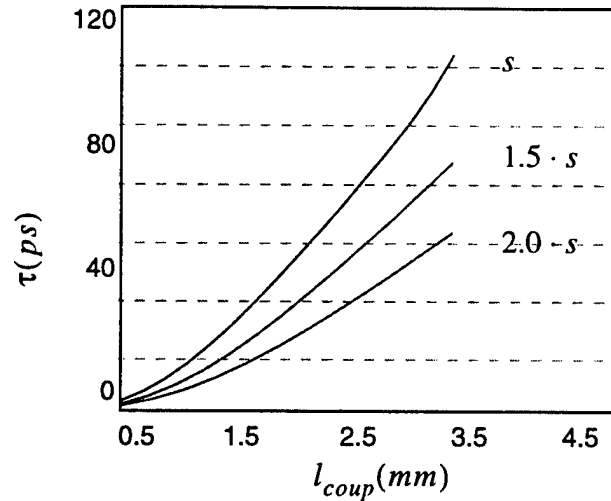
Figure 2.10   Crosstalk-delay impact vs. coupled length and spacing.

The circuit in Figure 2.8 is simulated by SABER with the SIA-1998 technology parameters for 1999. Observe that crosstalk-delay impact increases quadratically with coupled wire-length (l_{coup}). Furthermore, increasing the spacing (s) reduces the impact.

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 2.2). This allows Eq. 2.14 to be approximated as:
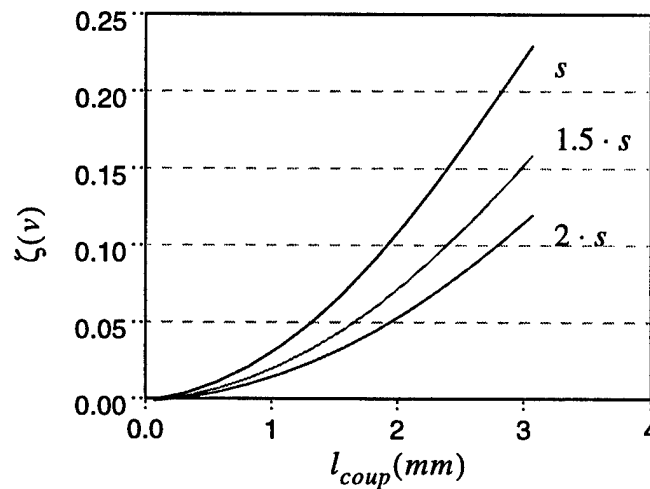
Figure 2.11   Crosstalk noise peak vs. coupled length and spacing.

The circuit in Figure 2.8 is simulated by SABER with the SIA-1998 technology parameters for 1999. Observe that the amplitude of crosstalk induced noise increases quadratically with coupled wire-length (l_{coup}). Furthermore, increasing the spacing (s) reduces the amplitude.

Table 2.2 Sample m and n parameters for Eq. 2.14.

The circuit in Figure 2.8 is simulated by SABER with the SIA-1998 technology parameters for 1999. The total coupled wire-length does not exceed 3000μm.

		Delay		Noise	
s	l_{coup}	m	n	m	n
s_0	< 3000μm	1.98	0.96	1.97	0.97
$1.5 \cdot s_0$	< 3000μm	1.88	0.93	1.91	0.94
$2.0 \cdot s_0$	< 3000μm	1.80	0.90	1.87	0.92

$$\tau = \frac{\alpha \cdot l^2}{s} \ and \ \zeta = \frac{\beta \cdot l^2}{s}. \qquad (2.15)$$

Eq. 2.15 captures the l^2/s relationship in a very simple form. The errors is these approximations are plotted in Figure 2.12 and Figure 2.13. For a maximum coupled wire length of 3.5mm, Figure 2.12 shows τ to be accurate to within ± 8 %. Figure 2.13 shows ζ to be accurate within ± 1 % for a similar maximum-length constraint. With extremely aggressive edge rates, the error in ζ increases to ± 10 %.
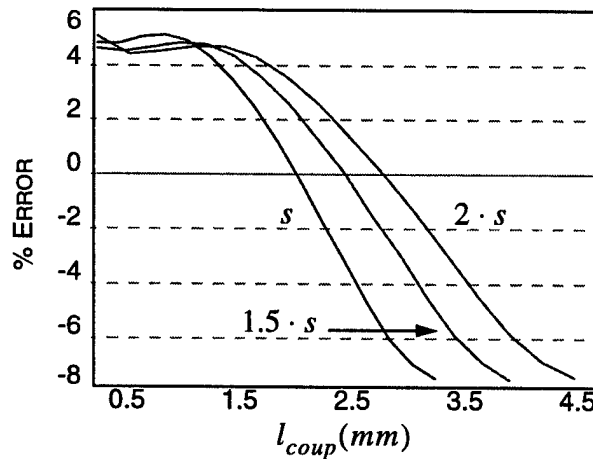
Figure 2.12 Crosstalk delay impact: approximation error.

The error in wire-delay (τ) by setting $m = 2$ and $n = 1$ in Eq. 2.14 is shown for a range of spacings.
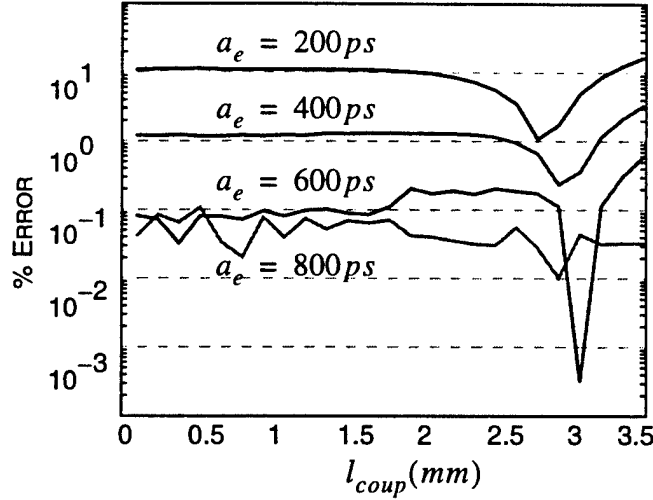
Figure 2.13   Crosstalk noise amplitude: approximation error.

The error in approximating wire-delay (ζ) by setting $m = 2$ and $n = 1$ in Eq. 2.14 is shown for a range of aggressor edge-rates (a_e). The error of the approximation increases for faster aggressor edge-rates.

Parameter α captures the effect of temporal proximity when computing wire-delay

changes (Figure 2.14). When measuring noise, β captures the dependence on the strength

of the aggressor waveform (Figure 2.15). Since α and β are non-linearly related to the
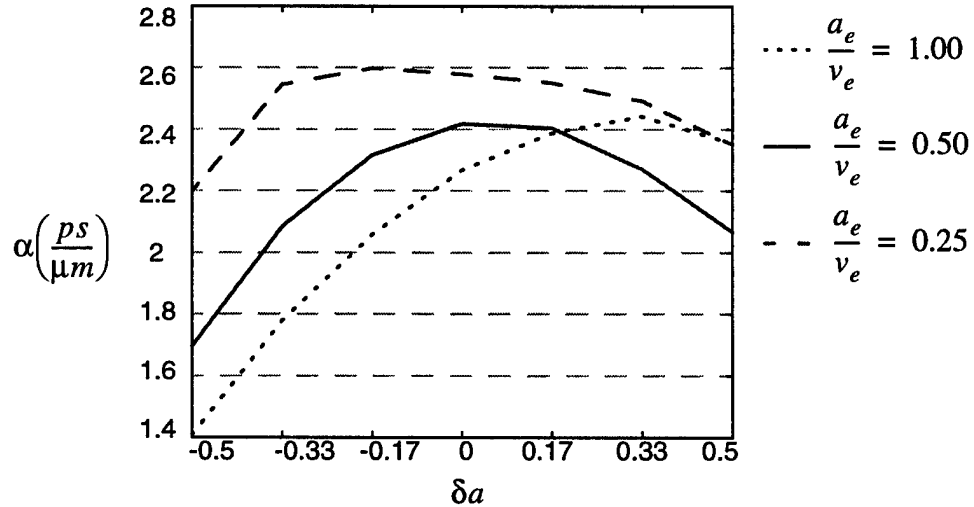
Figure 2.14   Effect of temporal proximity on α.

Increasing the temporal separation between the aggressor and the victim decreases α. This is true across a range of aggressor-victim edge-rate ratios.
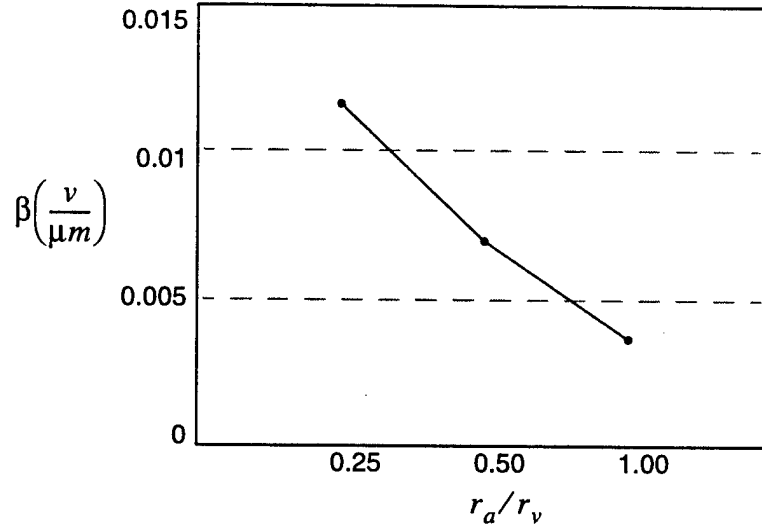
Figure 2.15   Effect of aggressor-victim relative drive strength on β.

The circuit in Figure 2.8 is simulated in SABER with the SIA-1998 technology parameters for 1999. The victim line is held at ground. Driver resistances are obtained from Table 2.1. The value for β is obtained by a least-squares curve fit of Eq. 2.15 to the measured amplitude of crosstalk-noise. Increasing the victim line's driver-strength (reducing r_v) reduces β.

arrival time and edge rates of the interfering signals, a table-lookup solution 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). For the α-table, δa is bound by $-0.5...0.5$. Values within the table are determined by linear interpolation. The bounding condition applies for the α and β tables. A computationally efficient table lookup will, therefore, approximate the curve fit parameters for all crosstalk behavior 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.
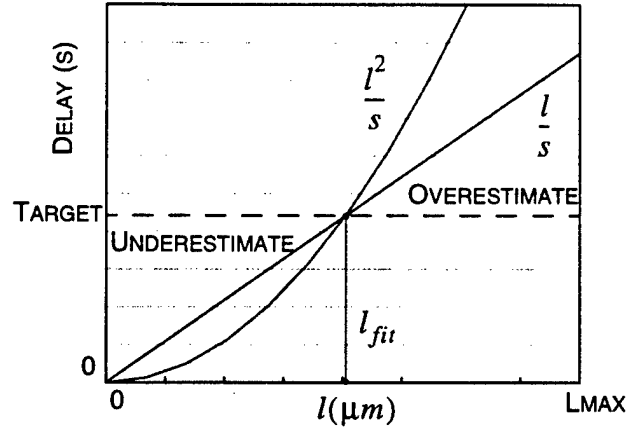
Figure 2.16   The empirical model vs. a capacitive crosstalk model.

The capacitive (linear) model is fitted to correctly estimate the impact at a pre-determined coupled length (l_{fit}). Above this length, the model will overestimate the allowable coupled wire-length for a given delay constraint. Below l_{fit} it will underestimate the allowable coupled wire-length for a given delay constraint.

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 [63, 122] precludes a performance-driven approach. Figure 2.16 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.

The following section defines a linearization of the empirical crosstalk model that permits a linear solver to satisfy crosstalk constraints by trading-off temporal and physical proximity.
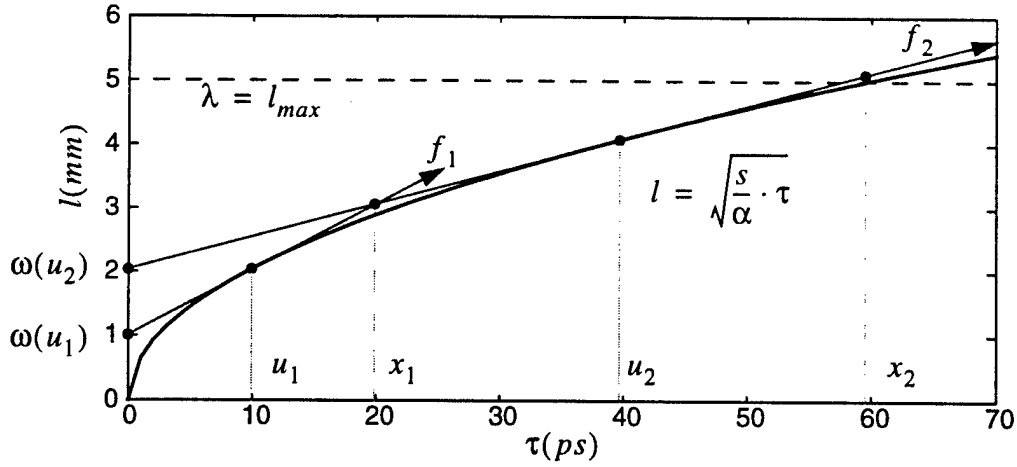
Figure 2.17   Linearizing crosstalk delay: two tangent approximation.

An approximation to $l(\tau, s)$ using two tangents f_1 and f_2 at positions u_1 and u_2 is shown. Location x_1 marks the intersection of f_1 and f_2. Parameter λ is the upper bound for $l(\tau, s)$.

2.3.2   Linearizing the Empirical model

Let s_1 be the minimum spacing between coupled lines and $s_j = j \cdot s_1$ $(j \geq 1.0)$. From Eq. 2.15,

$$l(\tau, s_j) = \sqrt{s_j \cdot \tau/\alpha} \text{ and } l(\zeta, s_j) = \sqrt{s_j \cdot \zeta/\beta} \qquad (2.16)$$

are derived. This transformation of Eq. 2.15 determines the allowable coupling length for a given delay change or noise amplitude. Eq. 2.16 is a non-linear function of s_j and τ (or ζ). The goal of linearizing this model is to derive a set of constraints that allows a linear solver to trade-off coupling length, spacing and delay (or noise). Figure 2.17 shows how an n-point piece-wise-linear trapezoidal approximation to Eq. 2.16 can be derived. The curve representing the wire-delay impact (or noise amplitude) can be approximated by a series of tangential lines. For brevity, only the linearization of delay-impact is derived. The identical conditions hold for linearizing the noise-amplitude equation.

Let f_i be one of the n lines, drawn tangential to $l(\tau, s_j)$ at points u_i:

$$f_i(\tau, s_j) = \omega_i(u_i, s_j) + \tau \cdot \frac{d}{d\tau}l(u_i, s_j) \qquad (2.17)$$

where $\omega_i(u_i, s_j)$ and $\frac{d}{dt}l(u_i, s_j)$ are the offsets and slopes of the n tangents. Two issues with this tangential linearization must be addressed:

1. Minimization of the approximation error, and

2. The linearization of $\omega_i(u_i, s_j)$ and $\frac{d}{d\tau}l(u_i, s_j)$

Error minimization is first addressed. From Figure 2.17, let adjacent f_{i+1}, f_i intersect at x_i. Using Eq. 2.17, we derive $x_i = \sqrt{u_{i+1} \cdot u_i}$. The error of the approximation attributed to a single trapezoid (bound by x_{i+1}, x_i) will therefore be

$$\int_{x_i}^{x_{i+1}} (f_i(\tau, s_j) - l(\tau, s_j))d\tau. \qquad (2.18)$$

The total error is a summation of Eq. 2.18 applied to all f_i. A minimization of the total error produces

$$\sum_{i=1}^{n} (\sqrt{u_{i-1}} - \sqrt{u_{i+1}}) \cdot (\sqrt{u_{i-1}} - \sqrt{u_i} \cdot \sqrt{u_{i+1}}) = 0. \qquad (2.19)$$

The initial condition of $u_0 = 0$, deduces

$$u_i = i^2 \cdot u_1. \qquad (2.20)$$

Let λ_1 be the upper bound for $l(\tau, s_1)$. Setting $l(x_{i+1}, s_1) = \lambda_1$, implies

$$u_1 = \frac{\lambda_1^2 \cdot \alpha}{n(1+n) \cdot s_1}. \qquad (2.21)$$

As shown, Eq. 2.20 and Eq. 2.21 define the u_i for a minimum-error linear approximation to $l(\tau, s_1)$ using n tangents.

The linearity of $\omega_i(u_i, s)$ and $\frac{d}{d\tau}l(u_i, s)$ in Eq. 2.17, with respect to s, is now addressed. It is obvious that $\frac{d}{d\tau}l(u_i, s_j)$, the slope of f_i, is a non-linear function of s_j. It can be made

invariant with spacing (i.e: constant for a given u_i) by making λ_1 (originally a user defined constant) scale with s_j. This reflects the fact that for greater spacings, an increase in coupling length is required to produce a similar impact. Therefore, λ_j is the upper bound for $l(\tau, s_j)$. Asserting

$$\frac{d}{d\tau}l(u_i, s_1) = \frac{d}{d\tau}l(u_i, s_j) \tag{2.22}$$

results in

$$\lambda_j = \frac{s_j \cdot \alpha_1 \cdot \lambda_1}{s_1 \cdot \alpha_j}, \tag{2.23}$$

which makes $\frac{d}{d\tau}l(u_i, s_j)$ a constant. Now, all that remains of linearizing f_i is the offset (y-intercept). Through empirical observation and curve fitting, it was noted that $\omega_i(u_i, s_j)$ (the y-intercept) could be re-written with negligible error in a linear form as $\gamma_i + v_i \cdot s_j$. This leads to Eq. 2.17 being expressed as a union of n tangents (f_i):

$$l \cong \bigcup_{i=1}^{n} (\gamma_i + v_i \cdot s + d_i \cdot \tau), \tag{2.24}$$

with $d_i = \frac{d}{dt}l(u_i, s_j)$, which is linear in τ, s_j and l. This formulation allows a linear solver to trade-off crosstalk impact for coupling length and spacing. For each edge pair, n constraints representing the tangential approximation are required. The values for γ_i, v_i and d_i can be pre-calculated for the various δa and aggressor-victim edge-rate ratios.

2.4 Modeling Interconnect in the Absence of Crosstalk

The previous sections showed that crosstalk (delay, noise) and its magnitude are sensitive to the temporal and physical proximity of the interfering signals. Therefore, accurate estimates of interconnect delay and signal transition rates are crucial to determining the impact of crosstalk. This dependence requires an RC model that provides
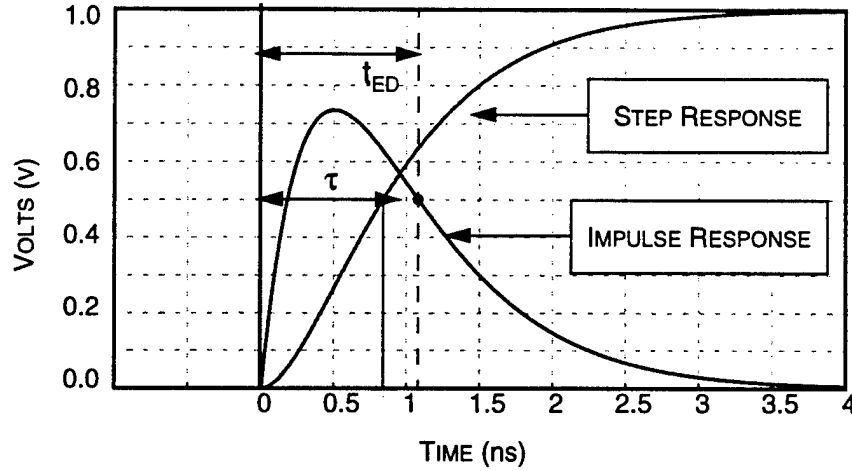
Figure 2.18   Impulse vs. step response for an RC circuit.

Term t_{ED} is the first moment of the impulse response. It is used to approximate the 50% crossing point of the step response (τ).

accurate waveform estimates at each node of the interconnect structure. The following sections describe an enhancement to an explicit two-pole RC model by Tutuianu et al [109], to accept exponential inputs.

2.4.1   Moment Matching Approximations for Interconnect Delay

To account for multiple interactions on a net, precise waveform information needs to be maintained at each point of interaction. Lumped capacitive delay models are inappropriate for DSM technologies as they do not account for wire resistance. Instead, advanced timing analyzers use moment-matching based transformations for RC interconnect structures. This methodology has been shown to be extremely accurate and computationally efficient. The original moment-based model was presented by Elmore et al [113]. The Elmore Delay (ED) of an RC circuit is an approximation of the step response of a circuit by the first moment of the impulse response ($h(t)$) (Figure 2.18):

$$ED = \int_0^\infty t \cdot h(t) \cdot dt \tag{2.25}$$

Penfield et al [92], showed that for any linear RC interconnect structure the ED could be computed recursively by a path tracing algorithm. If R_e is the driving resistance for node e, and C_e is the total downstream capacitance rooted at e, the ED to any node m can be computed as:

$$ED = \sum_{e \in Path(m)} R_e C_e, \qquad (2.26)$$

where $Path(m)$ is the path from source to node m. The ED model is inaccurate as it does not include the higher moments of the impulse response, which contain RC components of sub-trees not on $Path(m)$. Therefore, it cannot be used to predict waveform shapes for crosstalk analysis.

Asymptotic Waveform Estimation (AWE) [97] with a Padé approximation approximates the Laplacian of the transfer function $h(t)$ as a summation of exponentials:

$$h(t) = \sum_{i=1}^{n} k_i \cdot e^{-p_i t} \qquad (2.27)$$

The p_i and k_i are the poles and residues of the approximated transfer function. This is achieved through a Taylor Series expansion of e^{-st} about $s = 0$ of the Laplacian of $h(t)$.

$$H(s) = \int h(t) \cdot e^{-st} \cdot dt \qquad (2.28)$$

$$H(s) = \int h(t) \cdot \sum_{i=1}^{n} \frac{(-st)^{i-1}}{i!} \cdot dt$$

$$H(s) = 1 + m_1 s + m_2 s^2 + \dots + m_n s^n$$

Therefore, the i circuit-response moments are expressed as:

$$m_i = \frac{-1^i}{i!} \cdot \int_0^{\infty} t^i \cdot h(t) \cdot dt \qquad (2.29)$$

The nth moment can be calculated by scaling each of the downstream capacitances by the $n-1^{\text{th}}$ moment [92]:

$$m_i^{(n)} = -\sum_{e \in Path(i)} R_e \cdot \sum_{k \in C_e} C_k \cdot m_k^{(n-1)} \qquad (2.30)$$

which is an extension of Eq. 2.26. $t_{ED} = -m_1$ is derived by comparing Eq. 2.29 to Eq. 2.25. A rational expression for $H(s)$ is obtained through the Padé[1] approximation to Eq. 2.28:

$$\sum_{i=0}^{n} (m_i s)^i \cong \frac{N(s)}{D(s)} = P(s) \qquad (2.31)$$

$D(s)$ is a polynomial of power n while $N(s)$ is one of power $n-1$. Therefore the Padé approximation of $H(s)$ will possesses n poles and residues allowing it to be re-written as:

$$P(s) = \frac{N(s)}{\displaystyle\prod_{i=1}^{n} (s + p_i)}, \qquad (2.32)$$

where p_i, the dominant poles of $H(s)$, are the n roots of the polynomial $D(s)$. This factoring of $D(s)$ allows Eq. 2.32 to be decomposed into

$$P(s) = \sum_{i=1}^{n} \frac{k_i}{s + p_i} \qquad (2.33)$$

where the k_i are the residues. Eq. 2.33 has the Laplace inverse as required by Eq. 2.27.

AWE approximations for a RLC circuit suffer from a few drawbacks. For a given RC circuit, it is difficult to determine the number of poles *a priori*. This is because the dominant poles are unknown. Choosing a predetermined number of poles is inappropriate, due to the instability created by to the Padé approximation. To combat this instability, the

1. Let $P^n(x)$ be a polynomial of power n. The fundamental theorem of algebra states that $P^n(x)$ will have n roots. Padé approximation states that $P^n(x)$ can be approximated by $N^m(x)/Q^n(x)$. This allows $P^n(x) \cdot Q^n(x) \approx N^m(x)$ to be solved by comparing the like powers of x.

Padé via Lanczos process [44] or moment shifting techniques [5] can be employed. Both occur at the expense of increased computation.

2.4.2 Explicit Two-Pole model

Tutuianu et al, derive a stable, two-pole and three moment delay approximation for RC trees [109]. The approximation makes use of the fact that the ratio of successive moments of an RC tree forms a series that converges to the dominant pole

$$\lim_{i \to \infty} \left(\frac{m_i}{m_{i-1}} \right) = p_1. \qquad (2.34)$$

Therefore, the first pole is estimated as

$$p_1 = -\frac{m_2}{m_3}. \qquad (2.35)$$

A two-pole, one-zero Padé approximation, to the three moment approximation of $H(s)$ is explicitly computed, leading to derivations for the second pole and both residues:

$$p_2 = p_1 \frac{\begin{vmatrix} \dfrac{1}{m_1} & -\dfrac{m_1}{m_2} \\ \dfrac{m_1}{m_2} & -\dfrac{m_2}{m_3} \end{vmatrix}}{}, \; k_1 = p_1^2 \cdot \frac{1 - m_1 p_2}{p_1 - p_2} \; and \, k_2 = -p_2^2 \cdot \frac{1 + m_1 p_1}{p_1 - p_2}. \qquad (2.36)$$

Analytical equations for the output waveform $y(t)$, using both step and ramp inputs, are derived. The time to reach a given target voltage (e.g.: 50%) can be determined through Newton-Raphson (NR) iterations. By deriving an approximation for the initial value of the NR solver, typically no more than one iteration is required.

This two-pole model is an accurate approximation of an RC-tree driven by step or ramp inputs, without the computational complexity of general AWE. It is shown to be within 5% of a four-pole AWE model which provides extremely accurate results for RC-trees, and within 0.5% when the signals are driven by realistic driving resistances.

The following section expands Tutuianu's model to account for exponential inputs.

2.4.3 Exponential Inputs

Ramp and step input waveforms are simplifications of real waveform shapes. By modeling the input waveform as an exponential, greater predictability is obtained. The explicit stable model for an RC-tree structure is augmented to account for exponential inputs. As in AWE with Padé, $H(s)$ for a given node in the global tree is represented as

$$H(s) = \sum_{i=0}^{n} (m_i \cdot s)^i. \tag{2.37}$$

Next, define

$$x(t) = 1 - e^{-c \cdot t} \quad \text{, and its laplacian} \tag{2.38}$$

$$X(s) = \frac{1}{s} - \frac{1}{s+c}$$

as the input function with parameter c that determines the 20% to 80% edge rate. The Laplacian form (for the voltage at any node on the tree) can then be represented as

$$Y(s) = X(s) \cdot P(s) \tag{2.39}$$

$$Y(s) = \left(\frac{1}{s} + \frac{1}{c+s}\right) \sum_{i=1}^{2} \frac{k_i}{s+p_i}$$

The Inverse Laplace Transform, using a two pole $H(s)$, yields the following equation:

$$y(t) = c\left(\frac{k_1(1-e^{p_1 t})}{p_1(c-p_1)} + \frac{k_2(1-e^{p_2 t})}{p_2(c-p_2)}\right) + (e^{-ct} - 1)\left(\frac{k_1}{c-p_1} + \frac{k_2}{c-p_2}\right). \tag{2.40}$$

In order to compute the delay to specific voltage, a NR scheme is used. With a ramp or step input, the initial approximation for the NR method is derived with the knowledge that
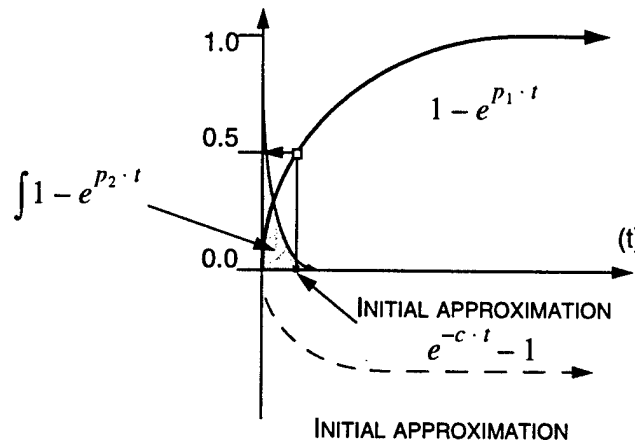
Figure 2.19 Components of an RC response to an exponential input.

The first exponent, due to the first pole p_1, dominates the RC response. The next exponent, due to the second pole p_2, can be replaced by a constant (shaded-area). The final exponent (dashed) is a scaling of the input waveform. In this figure, the initial approximation is derived from the dominant exponent.

the exponent term with p_2 can replaced by a constant (Figure 2.19) (p_1 is dominant).

However, the inclusion of the e^{-ct} term requires an extension to this initial condition. The

limit, as $t \to \infty$, of the contributions of the individual exponents in $y(t)$ are:

$$\lim_{t \to \infty} c \frac{k_1(1 - e^{p_1 t})}{p_1(c - p_1)} = \frac{c \cdot k_1}{p_1(c - p_1)}, \tag{2.41}$$

$$\lim_{t \to \infty} c \frac{k_2(1 - e^{p_2 t})}{p_2(c - p_2)} = \frac{c \cdot k_2}{p_2(c - p_2)}, \text{ and} \tag{2.42}$$

$$\lim_{t \to \infty} (e^{-ct} - 1)\left(\frac{k_1}{c - p_1} + \frac{k_2}{c - p_2}\right) = \frac{k_1}{c - p_1} + \frac{k_2}{c - p_2}. \tag{2.43}$$

If Eq. 2.41 is greater than Eq. 2.43 then p_1 will dominate c, and the initial estimate for t can be derived by solving

$$\tilde{y}(t) = \frac{c \cdot k_1(1 - e^{p_1 t})}{p_1(c - p_1)} - \frac{c \cdot k_2}{p_2(c - p_2)} + \frac{k_1}{c - p_1} + \frac{k_2}{c - p_2}. \tag{2.44}$$

Conversely,

$$\tilde{y}(t) = \frac{c \cdot k_1}{p_1(c - p_1)} + \frac{c \cdot k_2}{p_2(c - p_2)} + (e^{-ct} - 1)\left(\frac{k_1}{c - p_1} + \frac{k_2}{c - p_2}\right), \qquad (2.45)$$

is the initial estimate if c dominates. Using this initial condition, successive NR iterations (typically no more than two), yield an accurate estimate of t at which $y(t)$ is the specified target voltage.

2.5   Summary

Accurate models of crosstalk interactions are required to achieve timing convergence in a design flow. An analytical examination of coupled nodes is detailed. Through simulation of distributed coupled conductors, the dependence and sensitivity on waveform parameters such as arrival time and edge-rates is shown. Simulations on coupled wires, using the SIA predicted technology-parameters for 1999, show that a 12% change in arrival time difference can cause a 50% change in wire delay. The importance of accurately modeling the interacting waveforms is justified. Analysis on the interaction between capacitively coupled conductors disclosed a relatively simple, yet extensible model capable of predicting both crosstalk noise-amplitude and wire-delay changes. It permits a performance-driven approach to addressing crosstalk. The model is transformed into a set of linear constraints. With a linear solver, the trade-off between temporal proximity and spatial proximity can be made.

The explicit two-pole RC model is extended for exponential inputs. It provides accurate waveform estimates at each node of an interconnect structure.

CHAPTER III

CROSSTALK CONSTRAINED DESIGN METHODOLOGY

This research advocates the use of signal proximity, in both the spatial and temporal domains, to manage crosstalk. Place and route tools are embedded within a wiring engine to address congestion and crosstalk. This reduces costly design iterations by improving the predictability of the design. The ensuing sections will present a case for the avoidance of crosstalk using congestion-driven placement, and the reduction of crosstalk via an appropriately enabled route embedder.

3.1   Overview

Traditional CAD methodologies are driven by objective functions that minimize area while meeting timing requirements. Crosstalk manifests itself as timing and noise-margin violations that are only detectable after detailed routing. Resolving these may require multiple iterations through the design flow. Figure 3.1 depicts reactive methods to address crosstalk in a typical design methodology. After multiple timing-driven place and route iterations, an accurate parasitic extraction is performed. This is followed by timing analysis, during which crosstalk-related timing errors and noise-margin violations are detected. Some timing analyzers fail to adequately address the dynamics of crosstalk, requiring expensive circuit simulation of the critical paths. The detected violations are resolved by back-annotating the extracted parasitics and directing the entire methodology towards another solution which may possess a reduced number of crosstalk violations. This involves selective ripup and re-route with spacing constraints or ground shielding to
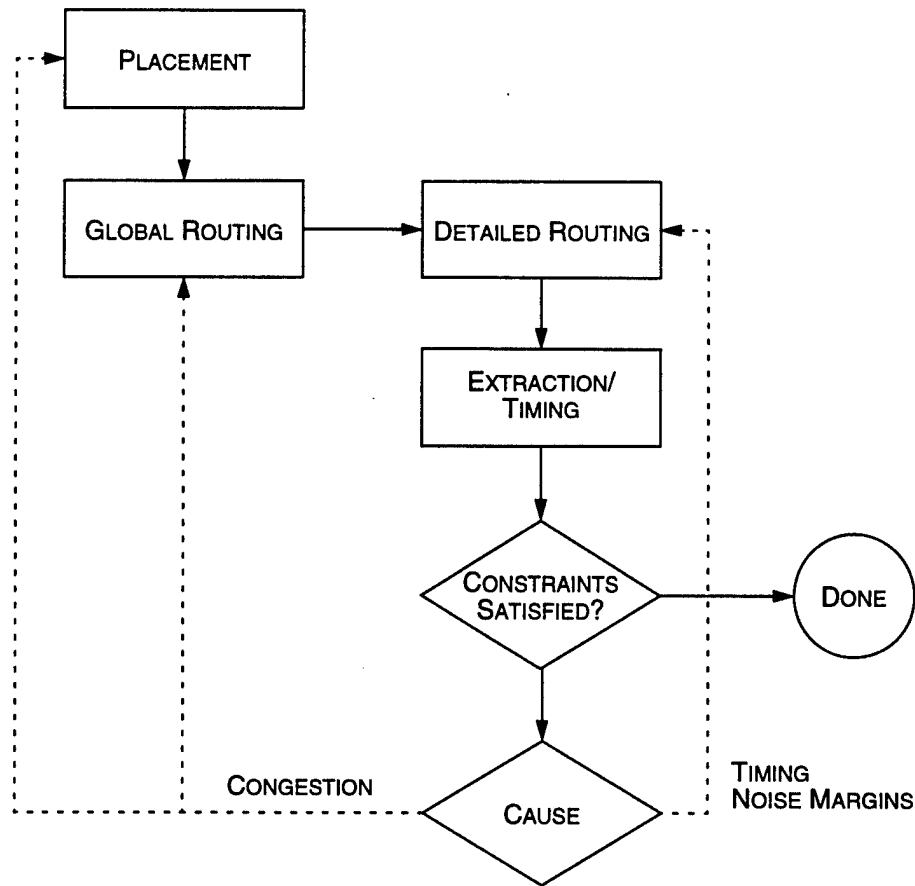
Figure 3.1   Iterative methods to address crosstalk.

After detailed routing, parasitics are extracted and a detailed timing analysis is performed. The design flow iterates to correct any unsatisfied design constraints (dashed paths). Timing and noise-margin violations are corrected by selective re-routing. Congested regions are re-designed by refining the placement and global-routing.

bound the parasitic line-to-line capacitance. In some cases, manual intervention is required. Occasionally, a design cycle may return to placement to help find a better, less congested solution that can be successfully routed to satisfy crosstalk constraints. Achieving timing convergence is a lengthy and difficult process requiring multiple iterations. While extracted parasitics do aid in directing the design iterations, other information such as congestion, global route structure and receiver slack and noise margins needs to be accounted to reduce the number of design iterations.

Currently, solutions for addressing crosstalk fall into three broad classes:

1. Crosstalk capacitance minimization by modifying defined route structures.

2. Satisfying noise-margins through spacing constraints.

3. Global/area routing with crosstalk constraints.

A detailed router, already burdened by aggressive multi-layer metallization, blockages, pre-allocated routing channels, performance constraints (wire sizing, via-minimization, etc.) is further complicated by the need to satisfy crosstalk constraints. Restricting the problem of routing to specific, well defined instances (e.g., two-layer channel routing) permits a concurrent approach to routing with crosstalk-constraints [64, 63]. However, this narrow definition of routing makes the approach inappropriate for the general case. The second approach of solving spacing constraints circumvents the complexity of detailed routing. However, its success is limited by the input detailed route structure. A compromise between these two approaches is needed. Recently, global and area-routing methods that incorporate a crosstalk metric within the algorithm have been demonstrated [108, 121]. These rely on iterative rip-up and re-route of critical nets to satisfy crosstalk constraints. Zhou et. al, demonstrated an MCM maze-router that satisfied crosstalk constraints using lagrangian relaxation [122]. However, the complexity of their formulation precludes its applicability for IC routing. Unfortunately, the crude crosstalk models implemented in these examples will fail to correctly detect and account for timing and signal-integrity violations. The simplified models of crosstalk for the aforementioned methods will not permit a performance driven approach.

Over the last decade, the average product life of an integrated circuit has shrunk from three years to one, while the time taken to complete a design has increased from three months to six. This increase is due to rising complexity and failure to achieve timing convergence. The pressure to complete designs in order to meet time-to-market (TTM) goals has risen six fold [93]. TTM considerations are forcing buyers to choose tools that boast quicker
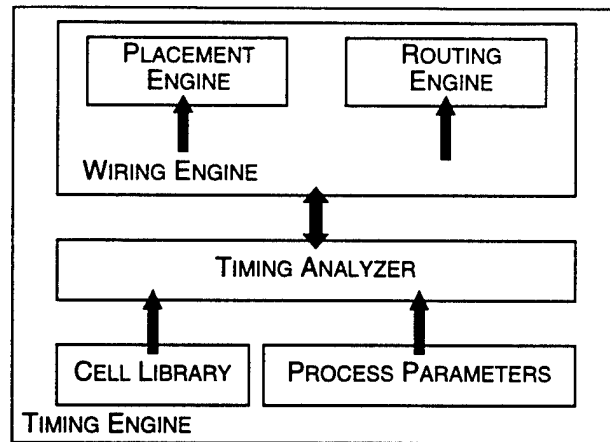
Figure 3.2   Hierarchy for proposed methodology.

The placement and route engines obtain all wiring related information through the wiring engine. The wiring engine communicates with the timing analyzer to obtain signal-waveforms and determine crosstalk impact. The cell library and process determine the parameters for the timing analyzer.

timing convergence, motivating the need for methodologies and theory to address crosstalk in VLSI.

3.2   Proposed Methodology

The crosstalk impact between two signals is a measure of their physical and temporal adjacency. Violations due to crosstalk are caused by inconsistent views of routing through the design. This permits the derivation of inaccurate net loads and wiring requirements, rendering invalid any estimates of physical or temporal proximity. Therefore, a consistent view of routing which maintains physical and temporal information must be generated and adhered to throughout the design flow.

Signal proximity in spatial and temporal domains can be used to manage crosstalk. The methodology shown in Figure 3.2 defines place and route tools as components of a Wiring Engine. The wiring engine directs the place and route algorithms to address crosstalk by providing a consistent view of routing. In the ensuing sections, a case for addressing
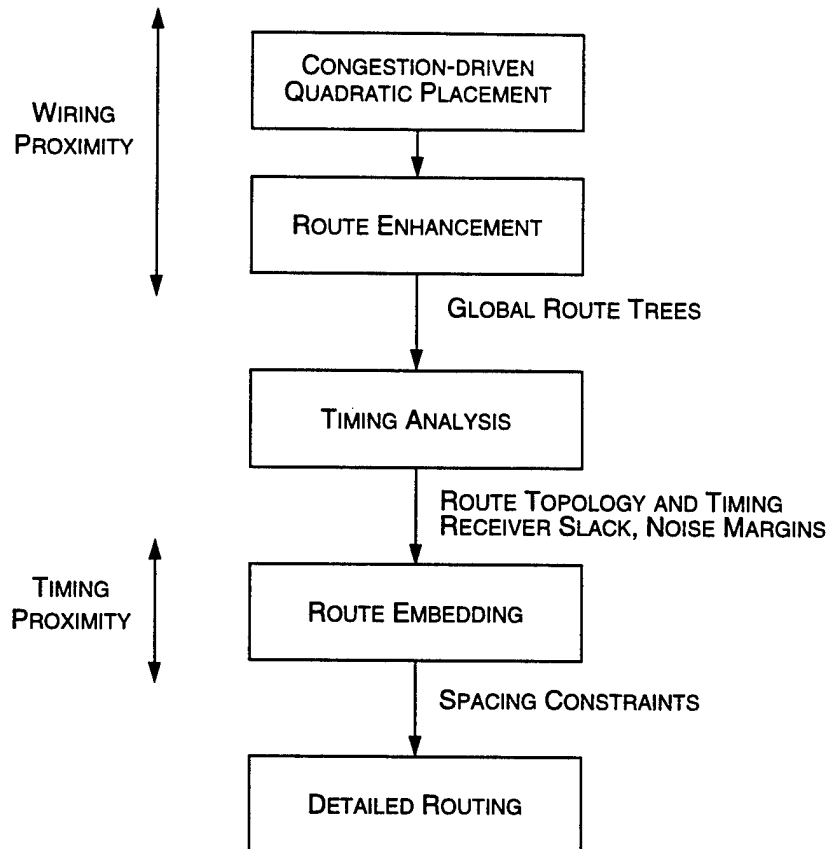
Figure 3.3   Design Methodology.

Wiring proximity is addressed through a congestion-driven quadratic placement methodology. Global route trees, used to generate the congestion estimates, are analyzed by the timing engine and then processed by the route embedder to satisfy crosstalk constraints.

crosstalk through the use of congestion-driven placement and route embedding will be made.

The design methodology flow, illustrated in Figure 3.3, begins with a congestion-driven placement algorithm. Most placement engines allocate space to account for wiring, using estimates that are not determined by actual route topologies. The congestion-driven quadratic placer uses the wiring engine to generate congestion estimates using accurate wiring estimates derived from global routing. This methodology innovates by enforcing these global routing structures throughout the design flow. The classical steps of

placement followed by global routing are replaced by Congestion-driven Quadratic Placement followed by Route Enhancement. During route enhancement, the global routes are minimally modified to meet detailed wiring requirements. Following this, the enhanced global routes are processed by a timing analyzer to generate detailed estimates of each net's waveform (delay and edge-rate). Slack constraints at each sink are also derived. The Route Embedder uses this temporal information to derive the expected crosstalk impact at each sink. A solution to mitigate this impact, specified as spacing constraints for a detailed router, is generated. These spacing allocations satisfy the expected delay and noise constraints at each sink.

The following sections present each component of the methodology. The detailed interactions between the components are illustrated in Figure 3.8 on page 74.

3.2.1 The Wiring Engine

Through the use of place and route tools, the wiring engine guides an implementation towards low spatial and temporal congestion (see Figure 3.4). It reduces
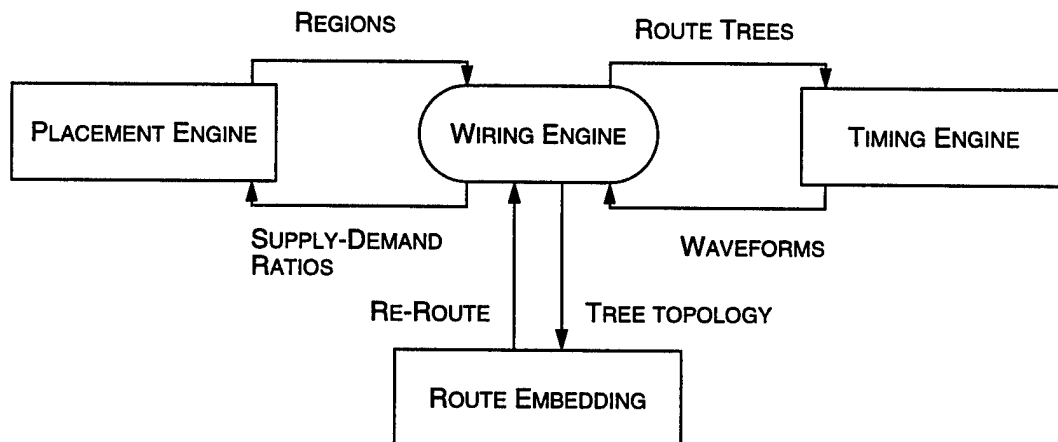
Figure 3.4   Wiring Engine services.

The wiring engine generates global-routes for congestion estimation in the placement engine. It invokes the timing engine to generate waveform estimates for the global routes. The wiring engine modifies existing global-route topologies for the route embedder.

the potential sites for crosstalk by addressing congestion during placement. Working concurrently with the placer, it efficiently and accurately models congestion for a given placement expression. Internal route estimation heuristics and region-based global routing help determine an accurate estimate of wiring demand. This information is used to compute supply-demand ratios, which are used by the placement engine to transform regions. The transformations are directed at equalizing routing resource demand to the given supply. After placement, the wiring engine minimally modifies the region-trees to eliminate wiring congestion. Using the timing engine, an initial timing estimate of the route structures (and the design) is generated. These timing-annotated route trees are delivered to the route embedder to determine a solution that satisfies the timing and noise-margin requirements at each sink. The route embedder can request local modifications of route structures through the wiring engine.

The wiring engine serves as the focal point for all wiring related computation, and enforces a routing topology on the placement, timing and routing engines.

3.2.2 The Timing Engine

The timing engine provides waveform estimates for a given RC-tree to the wiring engine. The change in delay and amplitude of induced noise on an RC segment, due to crosstalk, is dependent on the temporal proximity and shape of the interfering waveforms. RC timing models that do not derive the output waveform delays and edge rates, are unacceptable for addressing crosstalk. The timing engine also provides feedback to the route embedder about delay-changes and noise due to crosstalk (see Figure 3.5). These estimates of crosstalk account for physical and temporal proximity over a range of design parameters.
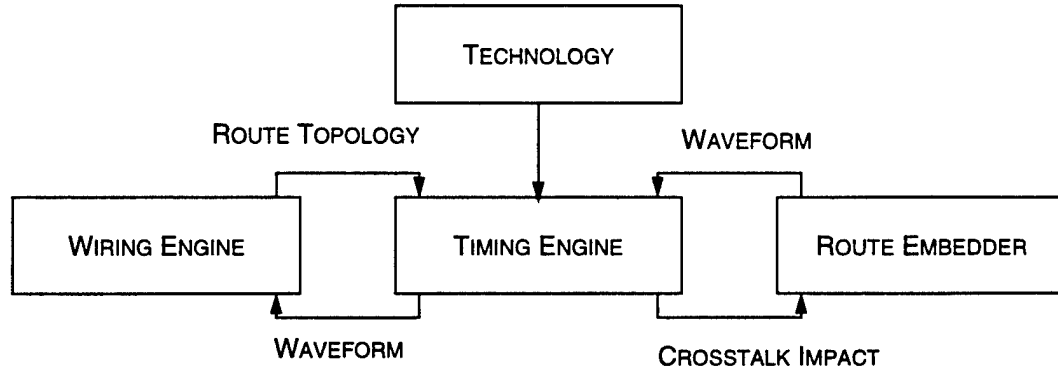
Figure 3.5   Timing Engine services.

The timing engine calculates waveform estimates for the route topologies provided by the wiring engine. It determines the crosstalk-impact for the route embedder.

Repeated invocation of the timing models, within the inner loops of the wiring engine and the route embedder, requires that they be simple and computationally efficient. The Elmore Delay (ED) metric shows high fidelity for RC-trees, implying it can accurately predict if one RC path will have greater delay than another. Yet it is inaccurate [113]. The sensitivity of crosstalk-impact to signal proximity and edge rates renders ED unacceptable. Asymptotic Waveform Evaluation (AWE) [39, 40], is frequently used in lieu of ED to determine accurate waveform estimates. However, AWE suffers from the algorithmic complexity of numerical computation and the stability of its results [44, 5]. The timing engine implements an explicit three moment, two pole model, proposed by Tutuianu et al [109], for waveform estimation.

Crosstalk impact is determined through an empirically derived model that accounts for temporal and physical proximity. Using an efficient table-lookup scheme to compute impact parameters, delay changes due to adjacent interacting waveforms and the amplitude of injected noise are efficiently determined.

3.2.3 The Placement Engine

Minimizing congestion has the dual benefit of reducing potential sites for crosstalk and providing flexibility (area) for a detailed router to address crosstalk. In order to model congestion during placement, the wiring model used by the placer needs to approximate the routes produced by the routing engine. The objective functions of classical placement algorithms fail to adequately address congestion because they are inaccurate abstractions of wiring. Cheng et al, demonstrated a wiring model that accounted for wiring demand as a function of the net topology [17]. The model effectively reduced congestion when inserted into the cost function of an annealing based placement engine.

Figure 3.6 depicts a framework that enables a quadratic placer to relieve congestion using actual route topologies while simultaneously solving for minimal wire length. Published
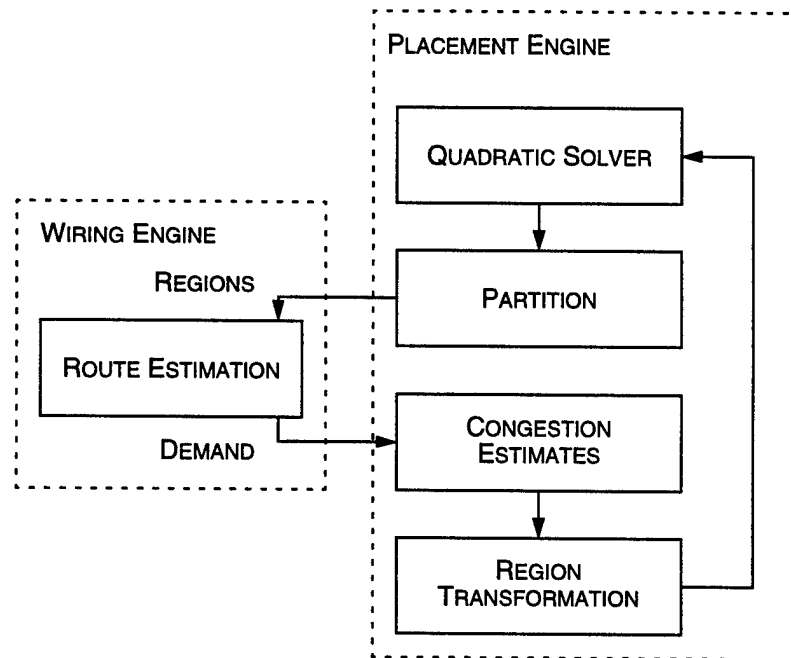
Figure 3.6   Congestion Driven Quadratic Placement.

The placement from the quadratic solver is partitioned into regions. The wiring engine generates global and internal routes, for these regions, to enable congestion estimation. Through these estimates, and the region transformation process, the quadratic solver accounts for congestion and refines the placement for the next iteration.

results by Parakh et al, demonstrate the effectiveness of the methodology in reducing congestion [90]. The wiring engine computes supply-demand weights through appropriate modeling of routing resources on a region-based routing graph. Repeated partitioning during quadratic placement allows this graph to converge upon a dense routing grid. The dynamic behavior of this routing grid supports increasingly accurate supply-demand modeling at each iteration of the algorithm. The wiring engine computes global demand by creating global routes between regions and derives internal demand estimates through heuristics. It provides the flexibility to enforce different route models and thus generate different congestion-driven placements. Further reductions in congestion, wire-length and area are obtained by incorporating relaxed pin constraints in the quadratic placement formulation.

3.2.4   Route Embedding

Instead of simplifying the crosstalk analysis for insertion into a detailed router, a route embedder attempts to model the detailed router. By doing so, the route embedder is able to simultaneously satisfy timing and noise constraints on a set of critical sinks. Furthermore, this is achieved without the combinatorial complexity of detailed routing. This is preferable to methods that effectively "search" for a valid routing solution using inaccurate crosstalk models.

As depicted in Figure 3.7, timing-annotated RC-trees are input to the route embedder. The route embedder then computes the expected crosstalk at each sink on each net due to neighboring activity. The embedder makes use of the timing engine's ability to differentiate between noise and delay-change events. The utilization of a sink's slack (or noise margin), due to the expected crosstalk impact from source to sink, is used to determine the criticality of each sink. To satisfy crosstalk constraints, the expected spacing along the RC path to each critical sink is determined. This is achieved by solving a linear
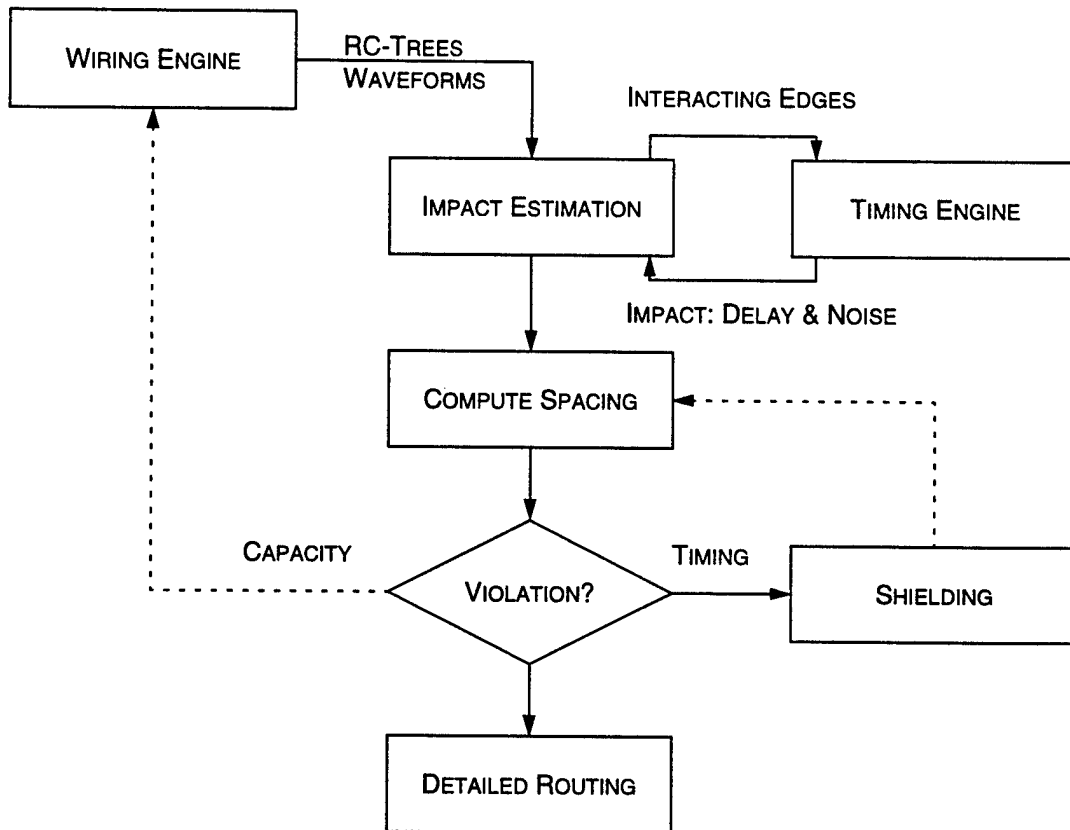
Figure 3.7   Global Route Embedding for Crosstalk.

The route embedder uses the timing engine to compute the expected crosstalk at each sink on each net due to neighboring activity. To satisfy crosstalk constraints, spacings for the critical paths are determined. Violations are iteratively resolved through shielding and limited re-route.

program. A crosstalk-violation occurs if a path does not meet its slack or noise-margin requirements. These paths are analyzed and the cause of their violation is determined. Timing violations are resolved through spacing wires and the introduction of ground shields. Capacity violations are satisfied by local routing modifications supported by the wiring engine. The resulting expected spacings and route refinements represent a route embedding that satisfies crosstalk constraints.

3.3 Summary

This chapter introduces a design methodology in which crosstalk is addressed by managing a consistent view of signal spatial and temporal proximity. By maintaining this view, timing violations due to crosstalk become predictable and, therefore, avoidable. The Congestion-driven Quadratic Placement engine and Route Embedder are the primary components for satisfying crosstalk constraints. Their interplay is directed by a wiring engine that maintains and enforces physical and temporal constraints throughout the design flow. By addressing wiring congestion, the methodology is able to reduce potential crosstalk sites and relieve the burden on the route embedder. The wiring engine provides accurate wiring estimates derived from global routing for the placement engine. The methodology enforces these global routing structures throughout the flow. These route structures are annotated with timing and processed by the route embedder. The temporal information is calculated by an efficient and accurate timing engine, capable of modeling RC-trees and crosstalk impact. The route embedder uses the timing engine to derive the expected crosstalk impact at each sink. The timing engine accounts for noise and delay-changes through an empirically derived formulation. A solution to mitigate this impact, specified as spacing constraints for a detailed router, is generated by solving a linear program. These spacing allocations will satisfy the expected delay and noise constraints at each sink.
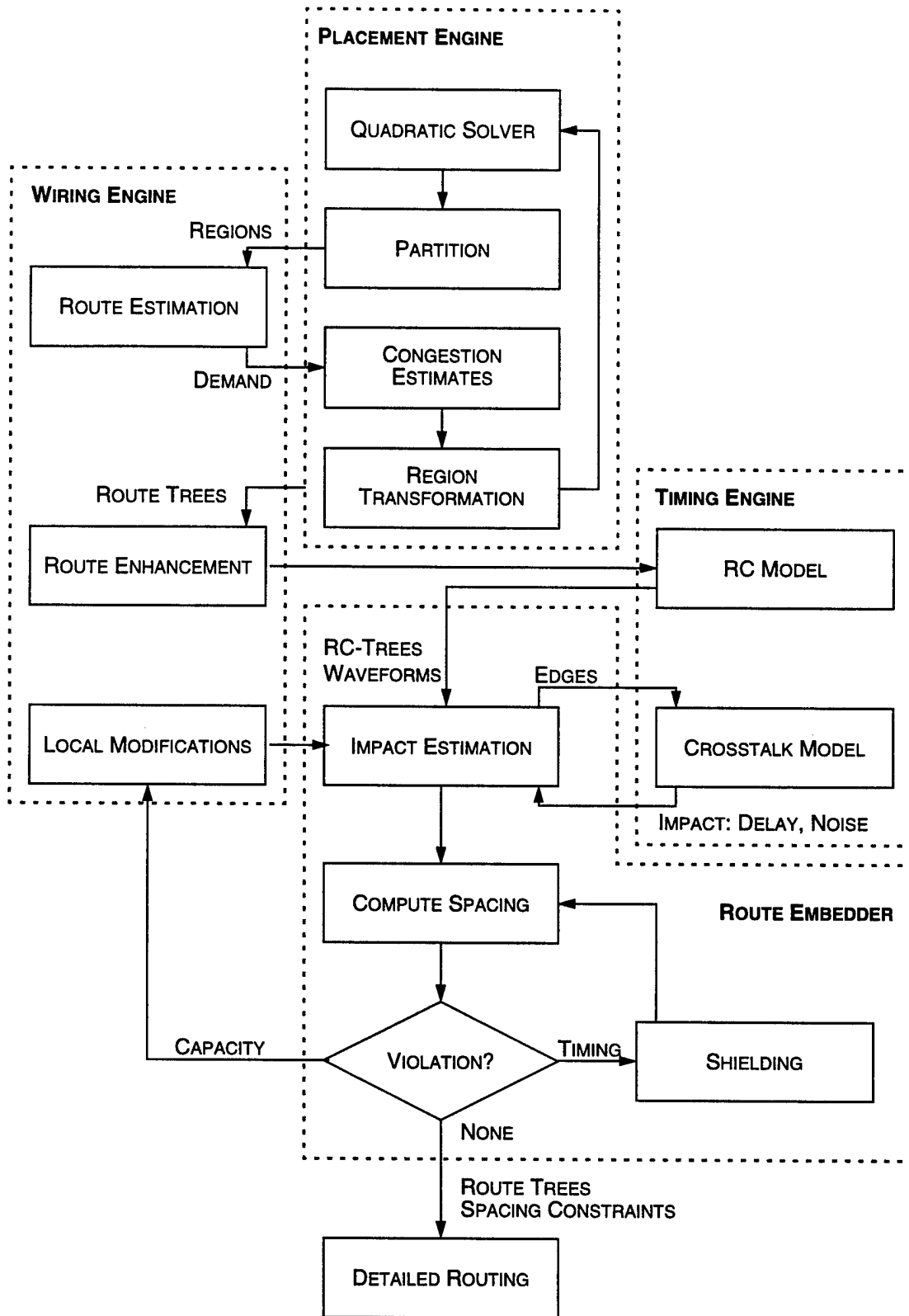
Figure 3.8 Design Methodology (detailed)

CHAPTER IV

CONGESTION DRIVEN QUADRATIC PLACEMENT

This chapter introduces the relationship between crosstalk and wiring congestion. Wiring congestion, in a region $\Re$, is defined by the ratio of routing demand to routing supply. Regions with congestion values greater than one (greater demand than supply) cannot be routed. With lower congestion values, a router will be able to provide adequate space around critical nets to reduce the impact of crosstalk, thereby helping to meet timing constraints. A variety of large designs are analyzed in this chapter to show the correlation between crosstalk and congestion.

This chapter introduces and demonstrates an extension to quadratic placement that accounts for wiring congestion. The algorithm uses an A* router and line-probe heuristics on region-based routing graphs to compute routing cost. The interplay between routing analysis and quadratic placement using a growth matrix permits global treatment of congestion. Further reduction in congestion is obtained by the relaxation of pin constraints. Experiments described in the end of this chapter show improvements in wireability over standard quadratic placement.

4.1 Correlating Congestion with Crosstalk

A PowerPC Fixed-point Unit (FXU) [105] designed in the PUMA project at the University of Michigan was used as a vehicle for evaluating the correlation between congestion and crosstalk. The FXU, designed in a three metal 0.35μm CMOS process,
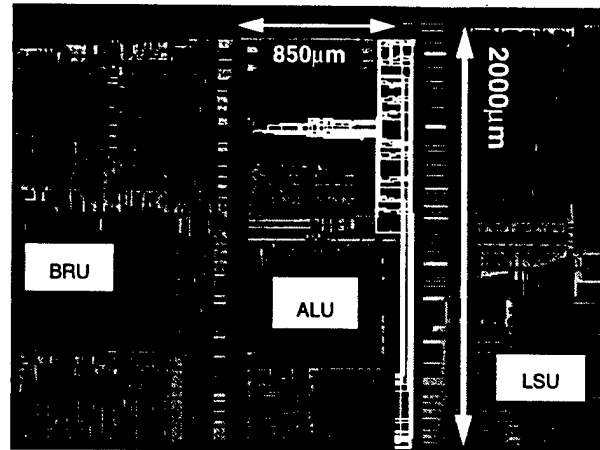
Figure 4.1   A critical bus shown in white passing though congestion.

The background shows a layout-capture of the execution unit in a fixed point
microprocessor designed in a 0.35μm CMOS process for the PUMA project
[89]. A critical bus overlays the layout. It passes horizontally through 850μm
of congested cells and then vertically through 2000μm of congested routing
channels.

was initially inspected for correlations between congestion and crosstalk on critical paths.

For example, Figure 4.1 explicitly shows a highlighted critical path flowing through

850μm of congested cells and 2000μm of congested channel area in the FXU's Execution

Unit (EXU). The average coupled length on this path is of the same total magnitude. Static

timing analysis predicted a standard deviation of 200ps for the edge generation times on

the bus, and an average of 100ps slack on the receivers. There is no doubt that crosstalk

could trigger timing errors. Therefore, reducing congestion will reduce the number of

timing violations by enabling the router to space traces apart so that coupling does not

extend propagation time through the wires. Restricting congestion analysis to the vicinity

of critical paths can reduce the complexity of congestion management.

In Figure 4.2, a congestion graph is generated for the FXU. The FXU is superimposed

with a grid. The congestion at each cell is determined by the global routes passing through

them. Cells with demand-supply ratios closer to one are congested, while cells with ratios
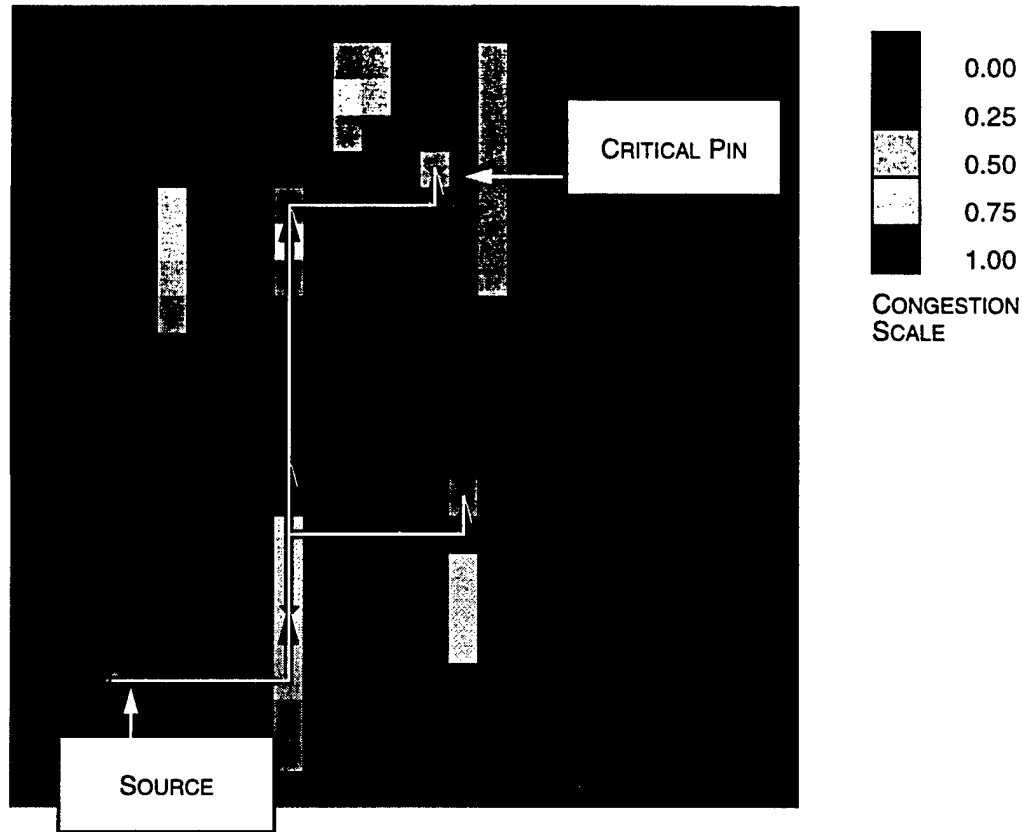
Figure 4.2   Critical net passing through congested regions in the FXU.

The FXU is overlaid with its congestion graph. The colors in the graph indicate the density of the underlying routes. A path on a critical-net, from source to critical pin is shown (in white) passing through regions of high congestion (red).

near zero are not. The figure shows a critical path traversing highly congested cells. It would be difficult to guarantee timing at the sink as it will not be possible to increase the space surrounding this net in the congested regions.

Global routes from six different designs were investigated to determine the relationship between routing congestion and crosstalk. s5378 and s15850 were selected from the ISCAS-89 benchmarks [21] and placed using Quadratic Placement (QP). These standard-cell groups were also placed with Linearized QP to produce s5378_1 and s15850_1. The EXU and FXU are block level netlists. Only their top-level routing was analyzed. Each placed netlist was superimposed with a grid. The congestion (ratio of demand to

Table 4.1 Probability of a critical path encountering congestion

The probabilities of a critical path encountering varying amounts of congestion are listed. For each design, a significant amount of the critical paths (>95%) are congested by more than 25%. On average, nearly half of the critical paths have half of their lengths passing through congested regions.

Design	Avg. l	Ratio of path congestion to path length			
		0.10	0.25	0.50	0.75
s5378	8.66	1.00	0.96	0.23	0.00
s5378_1	4.44	1.00	0.97	0.75	0.31
s15850	5.25	1.00	0.99	0.96	0.66
s15850_1	4.15	1.00	0.97	0.67	0.30
EXU	11.72	0.99	0.94	0.45	0.04
FXU	14.97	0.93	0.70	0.51	0.40

supply) imposed by the global routes, on the grid cells, is computed. The value of congestion in each cell is bound between zero (no demand) and one (full demand). Therefore, the total congestion encountered by a path of length l, computed as the sum of congestion in each cell traced through by the path, cannot exceed l. The paths were then ordered by their timing criticality and the top 10% of these were selected for analysis. The total congestion encountered by these paths is then computed and normalized by path length. This ratio of path congestion to path length is a measure of the congestion encountered by a path. Table 4.1 lists the probabilities of these critical paths encountering congestion, varying from 0.1 to 0.75. The table shows that placing s15850 with Linearized QP resulted in a 45% reduction in the number of critical paths that are congested by more than 75% (0.75). However, for s5378_1 an increase in congestion is noted. This is indicative of the fact that minimizing wire-length does not necessarily address congestion. In these netlists, nearly 100% of the critical paths pass through some amount of congestion. On average, nearly 50% of these critical paths have 50% of their lengths passing through congested regions.

(a) (b)
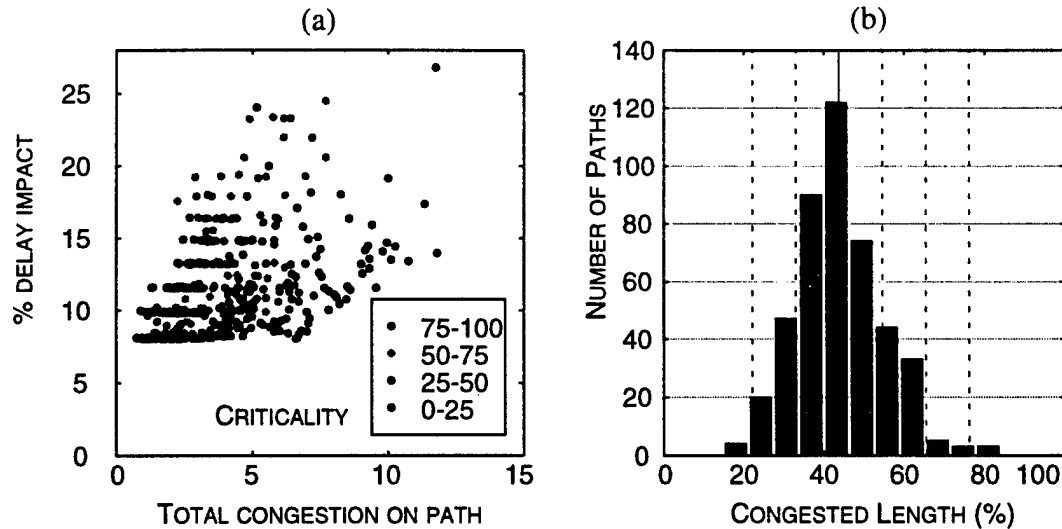
Figure 4.3   Correlating congestion with crosstalk on s5378.

The scatter chart (a) plots the total congestion on a critical path versus its crosstalk-induced delay impact. The criticality of individual paths (points) is color coded by quartiles. Paths with greater congestion and higher criticality are impacted by crosstalk with greater severity. The histogram (b) shows the distribution of congestion encountered by the critical paths.

Figures 4.3 to 4.5 correlate congestion with crosstalk, on these critical paths. The left plots compare the impact of crosstalk on a path with the encountered path congestion. The criticality of paths is color-coded by quartiles. The top fourth are colored blue, while the bottom are colored black. The histograms at the right present the data as distributions. Much about the structure of the designs can be determined from these charts. Pure standard cell designs have a continuous distribution of paths, and continuous distribution in congestion. The top level routing for the FXU, shown in Figure 4.5, is dominated by bus structures that disrupt the smooth distribution of the path lengths. However, a larger fraction of these routes (10%) have 90% of their paths traveling through highly congested regions. These results show that timing-critical paths tend to pass through congested regions, and as a result are subject to crosstalk effects. Therefore, these regions may become sources for timing convergence failure as they do not offer much freedom to a crosstalk conscious route methodology.
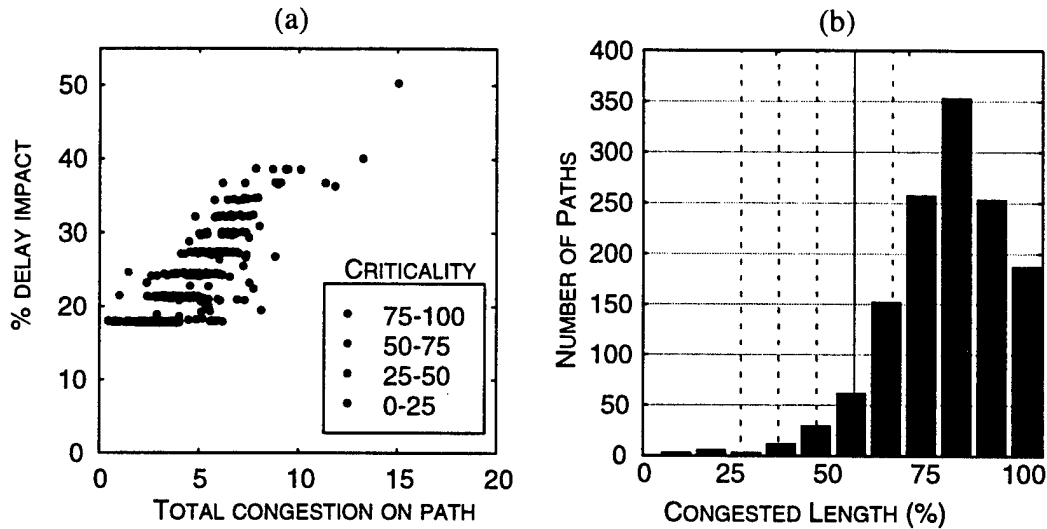
Figure 4.4 Correlating congestion with crosstalk on s15850.

The scatter chart (a) plots the total congestion on a critical path versus its crosstalk-induced delay impact. The criticality of individual paths (points) is color coded by quartiles. Paths with greater congestion and higher criticality are impacted by crosstalk with greater severity. The histogram (b) shows the distribution of congestion encountered by the critical paths.
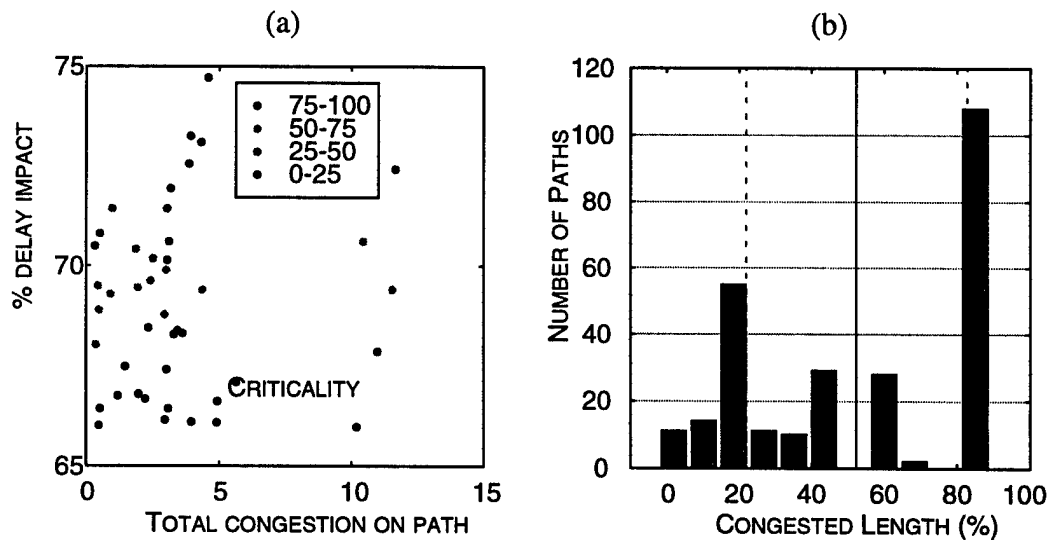
Figure 4.5 Correlating congestion with crosstalk on the FXU.

The scatter chart (a) plots the total congestion on a critical path versus its crosstalk-induced delay impact. The criticality of individual paths (points) is color coded by quartiles. Paths with greater congestion and higher criticality are impacted by crosstalk with greater severity. The histogram (b) shows the distribution of congestion encountered by the critical paths.

The demonstrated correlation among congestion, timing criticality and crosstalk justify the need for a congestion-driven placement algorithm. The following sections detail two enhancements for addressing congestion within quadratic placement.

4.2   Congestion Driven Quadratic Placement

Deep sub-micron metallization is proving previous wiring-abstractions as inappropriate. The focus of CAD has changed from optimization of gates to the management of routing resources. A design that effectively uses its resources requires smaller area, while inefficient utilization leads to greater number of wire crossings, increased wire length and poor channeling. Since the advent of overcell routing, the goal of place and route methodologies has been to utilize all available active area to prevent spilling of routes into channels. It is this overflow of routes that accounts for an increase in area. Congestion-aware global routers avoid this by constructing a set of routing trees for all nets in the design that match the global supply of wiring tracks. Heuristics are then applied to manage local congestion and to improve the final route quality. Wang et al, favorably demonstrate the concept of supply and demand to guide global routers [29]. The objective of their global router is defined in terms of route uniformity and route density. This scheme, and others like it, are unlikely to achieve optimality because the quality of a routing solution is largely determined by the starting placement, with congestion occurring where demand exceeds supply. Thus rises the need to model congestion within placement.

Typical placement objectives involve reducing net-cut costs or minimizing wire length. Due to their constructive nature, min-cut based strategies minimize the number of net crossings but fail to uniformly distribute them [130]. Quad-partitioning schemes, as first demonstrated in [136] did account for some global routing resources, but did not address supply nor account for internal routing. Congestion driven placement based on multi-

partitioning was proposed in [80], but was limited in the number of partitions due to the use of pre-computed Steiner trees. To the author's knowledge, congestion driven placement has not used information about congestion at a global level to update global placement. Alternatives to min-cut based schemes attempt to minimize wire length. The use of minimal wire length as a metric to guide placement has been successful but, it indirectly models congestion and the behavior of the router. Reducing the global wire length helps reduce the wiring demand, but it is oblivious to wiring supply. It is entirely possible for a minimum wire length solution to require more tracks for routing through a region than are available. Therefore, schemes such as quadratic placement (including its linearized form) which are based solely on wire length minimization cannot adequately account for congestion. A simulated annealer in [17] used wire distribution functions and net bounding boxes to model routing resource demand, but the complexity of its wire model precluded its use at a global level and forced the analysis to be performed in smaller blocks.

The quadratic formulation for squared wirelength can be solved for a global minimum. However, the solution contains much cell overlap which has to be resolved by partitioning and appropriate constraints. Therefore, it is incapable of producing sufficient non-overlap to account for routing resources. This section develops a framework that drives quadratic placement to relieve congestion while simultaneously solving for minimal wire length. Supply-demand weights are computed through appropriate modeling of routing resources on a region-based routing graph. Repeated partitioning during quadratic placement allows the analysis to converge upon a dense routing grid. The dynamic behavior of this graph supports increasingly more accurate supply-demand modeling at each iteration of the algorithm. The current implementation of the supply-demand algorithm creates global routes between regions using an A* search [98] algorithm and computes internal costs using a line-probe [27] heuristic. However, this component of the algorithm can be
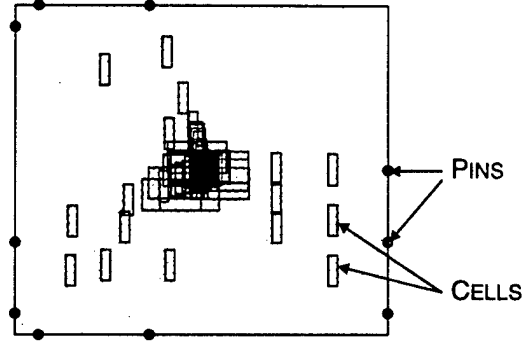
Figure 4.6 First iteration of Quadratic placement.

The positions of cells (rectangles) and pins (circles) of circuit s298 (Table 4.2) after the first iteration of quadratic placement is shown. The positions of the cells are determined by solving Eq. 4.2.

replaced with any appropriate route model. Further reductions in congestion, wire-length and area are obtained by the inclusion of relaxed pin constraints into the formulation.

The next section presents the key concepts in quadratic placement. This is followed by a section that describes region-updating during quadratic placement along with the necessary framework for congestion analysis on a region-based grid. Results of this new methodology are then presented.

4.2.1 Quadratic Placement

Quadratic placement engines [52, 68, 51, 135, 101] solve an unconstrained minimization problem whose objective function $\Phi_q(x, y)$ is the squared wirelength

$$\Phi_q(x, y) = \sum_{i, j} a_{ij}(x_i - x_j)^2 + \sum_{i, j} a_{ij}(y_i - y_j)^2 \qquad (4.1)$$

where x and y are vectors of cell coordinates and $a_{ij} \in \{0, 1\}$ represents connectivity between cells i, j. The optimal placement solution is found by solving:

$$\nabla \Phi_q(x, y) = 0 \qquad (4.2)$$

This can be represented in two systems of linear equations, separable in x and y. By expansion and proper assimilation of common terms, Eq. 4.1 can be rewritten in the more familiar form of:

$$\Phi_q(x) = \frac{1}{2}x^T Q x + d_x^T x$$

$$\Phi_q(y) = \frac{1}{2}y^T Q y + d_y^T y$$

(4.3)

where Q is the $n \times n$ Laplacian matrix, with $q_{ij} = -a_{ij}$ on the non-diagonal entries and $\sum_{j=1}^{n} a_{ij}$ on the diagonal entries. The vectors d_x and d_y originate from the locations of the coordinates of the pins and can be adjusted to include port locations on the cells.

The solution for Φ_q tends to be grouped around the origin, as seen in Figure 4.6. In order to disperse the cell locations and resolve overlaps, the solutions are partitioned and iterated upon. Quadratic placement algorithms differ in their methodology with respect to partitions, i.e., their iterative approach. The PROUD [101] algorithm, calls its quadratic solver on individual regions, while treating the rest as fixed. Thus its solution quality is dependent on the order with which regions are selected. The GORDIAN algorithm [68], creates placement via global optimization. At each iteration, it minimizes the global wire length while constraining the regions to unique centers of gravity. This scheme is favored due to its global outlook on optimization. However, the squared wirelength metric has been show to be inferior because of the inconsistency with which it handles net lengths [51]. Longer wires get overly penalized, while shorter ones get under penalized. The GORDIAN-L [135] algorithm minimizes linear wirelength through iterative refinement after each call to the GORDIAN solver. By weighting each edge a_{ij}, with the inverse of the current iteration's wirelength, successive calls to the quadratic placer minimize for linear wire length. However, the computational cost of repeatedly weighting nets, with the inverse of their linear distance, until convergence is prohibitively expensive for large designs.

$$\begin{bmatrix} d & 0 & \alpha & 0 & \beta & 0 \\ 0 & e & 0 & \gamma & 0 & \eta \end{bmatrix} \cdot x = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$$

$$H \qquad\qquad\qquad b$$

REGION 1

REGION 2

Figure 4.7   Center of gravity constraints.

Locations z_1 and z_2 are the centers of Region 1 and 2 respectively. Cells d, α and β belong to Region 1, while e, γ and η belong to Region 2. Matrix H forces the center-of-gravity of the cells in Region 1 and Region 2 to coincide with z_1 and z_2 respectively (b).

The quadratic placer used in this work makes use of the GORDIAN algorithm. Using the circuit's connectivity, a sparse set of linear equations is constructed, and solved, to obtain an (x_i, y_i) position for each cell. This is immediately followed by a partitioning phase which introduces a new center of gravity constraint for each region. The process is repeated, alternating in horizontal and vertical directions, until each region represents an individual cell. Generating center-of-gravity constraints for each region has the effect of distributing the cells. With each iteration, there is a doubling of the number of constraints, until each constraint represents an individual cell with a unique center of gravity. Thus, at a given level (l) of iteration, there may be $q \leq 2^l$ regions, each containing a unique subset of cells. The center of gravity constraints on the q regions are written as:

$$H_x x = b_x \qquad H_y y = b_y \qquad\qquad (4.4)$$

Each row, r, of the $\langle q \times n \rangle$ H (Figure 4.7), contains the normalized area of the modules within region r (each row sum is equal to one). Thus, the area-weighted mean of the coordinates within a region is required to correspond to the center of gravity of that region. Note that the columns of H contain one non-zero entry each. At each iteration, these q linear constraints restrict the movement of the modules to a $\Re^{n-q}$ subspace of $\Re^n$, by

fixing the location of one module per region, while permitting the rest a full freedom of motion. Therefore, $x \in \mathfrak{R}^n$, can be partitioned into q dependent (x_d), and $n - q$ independent variables (x_i), which in turn partitions H into $\langle q \times q \rangle$ dependent (D) and $\langle q \times n - q \rangle$ independent (B) matrixes:

$$Hx = \left[D | B \right] \text{ and } x = \begin{bmatrix} x_d \\ x_i \end{bmatrix} \tag{4.5}$$

x is now rewritten as:

$$x_i = Zx_d + x_0 \tag{4.6}$$

with $Z = \begin{bmatrix} -D^{-1}B \\ I \end{bmatrix}$ and $x_0 = \begin{bmatrix} -D^{-1}u \\ 0 \end{bmatrix}$. Substituting Eq. 4.6 into Φ produces an $\mathfrak{R}^{n-q}$

unconstrained quadratic problem in x_I:

$$\min \left\{ \psi(x_i) = \frac{1}{2} x_i^T Z^T C Z x_i + c^T x_i \right\} \tag{4.7}$$

with $c^T = (Cx_0 + d_x) \cdot Z^T$. C is the connectivity matrix with entry $c_{i,j}$ set to one if modules i and j are connected and zero if not. $\psi(x_i)$ can be minimized by setting the $\nabla \psi(x_i) = 0$, and solving the $\langle n - q \times n - q \rangle$ linear equation system $Z^T C Z x_i = -c$. Back substituting x_i into Eq. 4.6 produces x. The entire process is repeated for the y dimension.

These x and y solution vectors are used to aid the partitioner in generating new regions. G is partitioned into ρ' and ρ'' by choosing half of the cells for ρ' and the other half for ρ'', with ordering on the cells imposed by the solution vectors. Partition size and cut-costs are accounted for to improve the quality of the final placement. These partitions are maintained by adding a center of gravity constraint to each region. With further levels of

INDEPENDENT MODULES

Δx 1 Δx 1 Δx 1 2 $c\Delta x$
a   b   c   d
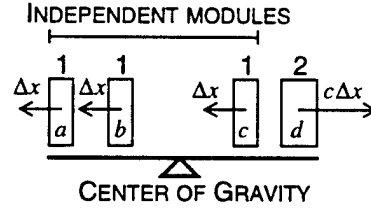
CENTER OF GRAVITY

Figure 4.8   Center of gravity constraint example.

Objects a, b and c are independent modules and of equal size (1). Object d is the dependent module (size 2). Moving the independent modules by Δx to the left forces d to the right, by $3\Delta x/2$, to maintain the center of gravity.

partitioning, a non-overlapping distribution of the cells, influenced by the global outlook of the minimization function, will be obtained. However, since the numerical solver must guarantee the constraint, it can force certain cells to move contrary to minimizing Φ. This happens when the dependent cells (x_d) have to guarantee a constraint through Eq. 4.6. In Figure 4.8, if the independent cells $\{a, b, c\}$ were moved left by Δx, the dependent cell d would be forced right by $(3\Delta x)/2$. Such events increase the separation between the cells, without minimizing Φ. Quadratic placement would be more appropriate if the constraints on Φ maintained cells around their regional centers, yet allowing them to gravitate without unduly impacting the placement of their dependent cells.

Finally, quadratic placement algorithms require pin locations, else they return a trivial solution with all cells at the origin. Graphically, the pins stretch the network of cells to induce a valid placement solution, and therefore, they have a strong impact on the solution. While it may be prudent to assign pin locations based on floorplanning, or through other top-down pin-placement approaches, the location (or order) of these pins may aggravate the placement. Consider the fact that incorrect pin ordering could twist the design and thus create congestion. In order to reduce congestion, the placement should be able to influence the pins.

```
┌─────────────────────────────┐
│      QUADRATIC SOLVER       │◄──────┐
└─────────────────────────────┘       │
               │                       │
               ▼                       │
┌─────────────────────────────┐       │
│          PARTITION          │       │
└─────────────────────────────┘       │
               │                       │
               ▼                       │
┌─────────────────────────────┐       │
│        REGION ROUTER        │       │
└─────────────────────────────┘       │
               │                       │
               ▼                       │
┌─────────────────────────────┐       │
│       INTERNAL ROUTER       │       │
└─────────────────────────────┘       │
               │                       │
               ▼                       │
┌─────────────────────────────┐       │
│        CONGESTION           │───────┘
│        ESTIMATOR            │
└─────────────────────────────┘
```
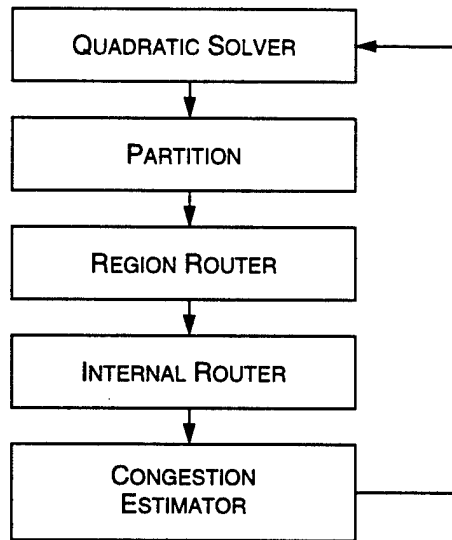
Figure 4.9   Congestion driven Placement methodology.

A placement from the quadratic solver is partitioned into regions. Then region based global and internal routes are generated. These routes permit the estimation of congestion. Using these estimates, and a region transformation process, the quadratic solver accounts for congestion and refines the placement for the next iteration.

4.2.2   Congestion Driven Placement

Figure 4.9 illustrates the congestion-driven placement methodology. Fundamental to this technique is the need to model congestion and to influence the quadratic placer with that metric. As in the GORDIAN algorithm, placement produced by minimizing the wire length metric is iteratively partitioned and re-placed with new center of gravity constraints. Before each successive placement, internal route estimation and a region-based global route are performed on each region to estimate supply-demand ratios. These ratios are used to influence the quadratic placer into growing (or shrinking) regions based on resource demand. Furthermore, congestion induced by incorrect pin ordering is relieved by relaxing the pin constraints from being fixed in both x and y, to being bound in a single dimension. Computing the internal and external routes takes much less time than a single minimization of Φ.

EXPANSION   COMPRESSION

REGION   REGION   REGION
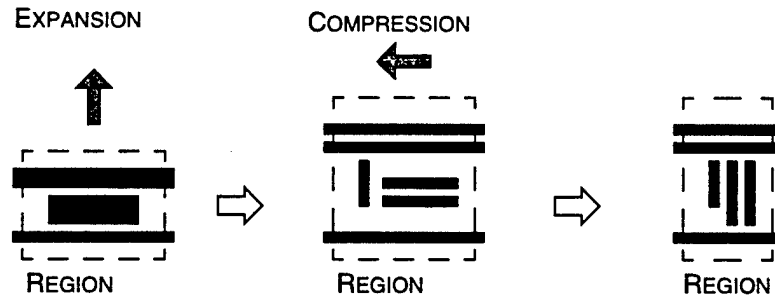
Figure 4.10   Example of region growth relieving congestion.

The initial vertical expansion converts some of the internal routes (gray) into vertical ones. A further compression may convert the remaining horizontal routes into vertical ones. The horizontal external routes (black) are not influenced by this process.

4.2.3   Growing and shrinking regions

Reduction in congestion occurs due to the ability of the model to transform horizontal routes into vertical routes and vice-versa. In Figure 4.10, if a region is deemed vertically congested, then a vertical expansion is performed. At this point, internal horizontal nets ($\Delta x > \Delta y$) could become vertical ($\Delta x < \Delta y$), thus relieving congestion. During the next iteration, if the region is shrunk horizontally, more nets from the previous iteration could become vertical, further relieving congestion.

The positions of the independent cells can be affected by supply-demand ratios returned from the congestion estimator. This allows resource limited regions to be expanded (or reduced) to account for the wiring demand imposed upon them. The horizontal ratio is used to stretch the region in y, while the vertical ratio stretches the region along x. In Figure 4.11, vertical growth of region 1 is achieved by a growth factor of two. It is interesting to observe that the growth factor does not simply disperse the modules within a region. They have influenced cell positions while permitting the solver to minimize Φ.

(a)   (b)

GROWTH: $W_1 = 2.0$
$W_2 = 1.0$
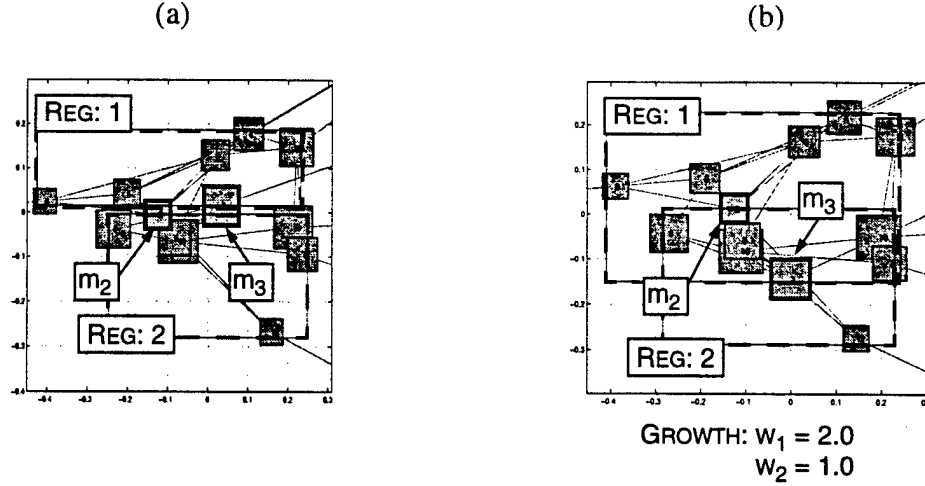
Figure 4.11   Region growth due to weights.

Region 1 is influenced by a vertical growth factor of two. Region 2 is not transformed. The result of the transformation is shown in (b).

To introduce the awareness of congestion into the algorithm, an $\langle n - q \times n - q \rangle$ diagonal growth matrix, G, with entry g_{ii} equal to the region weight for the independent cell x_i is constructed. Define $x_i' = Gx_i$. A global minimum of Φ that incorporates an awareness of congestion is obtained by substituting for x_i in Eq. 4.7 and solving:

$$(G^{-1}Z)^T C(ZG^{-1}) \cdot x_i' = -c'$$

(4.8)

with $c'^T = (Cx_0 + d_x)^T G^{-1} Z$. It is known that Z forms a basis for the positive definite C [68]. Since G is a diagonal matrix with positive diagonal entries, $G^{-1}Z$ will have the same form as Z and thus $(G^{-1}Z)^T C(ZG^{-1})$ will be positive definite. Standard successive over-relaxation (SOR) or Krylov subspace solvers [45, 39] can be used to solve the linear system.

The incorporation of the growth matrix into the quadratic placement algorithm is the key contribution of this methodology.

4.2.4 Computing the Growth Matrix

After each quadratic placement, a new set of regions is generated by the partitioner. For each congestion analysis, a new routing graph is constructed using these regions. An important characteristic of this routing graph is its dynamic behavior. With each successive placement and partitioning, the granularity of the routing graph decreases, thus increasing the quality of the analysis, until a detailed routing graph is achieved. Furthermore, since the regions reflect well-connected cells, they form more appropriate routing graphs than those produced for an arbitrary pitch. This approach provides the appropriate amount of supply and demand analysis at each iteration.

The supply created by each region R is separated into horizontal and vertical components: R_s^h and R_s^v, and cannot exceed the capacity for the region, which is determined by the horizontal and vertical metal pitch (g^v and g^h). Internal blockages, reflecting pre-allocated routing channels and port locations on cells diminish the supply. Each cell $\mu \in R$ contributes μ_f^v and μ_f^h, to the total number of vertical and horizontal feed-thrus for the region. This amount is normalized to the width R_w (or height R_h) of the region, thus producing the incident supply of a region:

$$R_s^v = \frac{\sum_{\mu \in R} \mu_f^v}{(\sum_{\mu \in R} \mu_w)/R_w} \qquad (4.9)$$

where μ_w is the cell width. The horizontal component is expressed identically.

Demand on each region is comprised of external, $E(R)$, and internal, $I(R)$, components:

$$R_d^v = E^v(R) + I^v(R) \qquad (4.10)$$
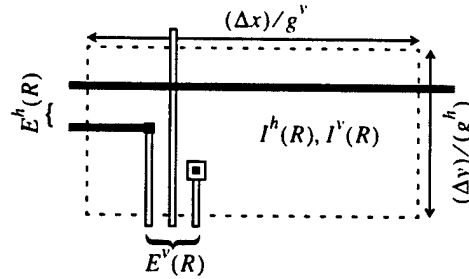
Figure 4.12 Region external supply and demand components.

The horizontal ($E^h(R)$) and vertical routes ($E^v(R)$) comprise the external horizontal and vertical demand on region R. The horizontal supply is computed by dividing the height of the region (Δy) by the vertical pitch (g^v). Vertical supply is similarly determined.

The external cost originates from nets that span multiple regions and is computed by the region router. It is the sum of all region-routes that intersect it (Figure 4.12) and includes routes that terminate within it. In order to compute the internal-route cost, a heuristic that mimics a line-probe router is used. Using Figure 4.13 as an example, the internal cost for route α is one vertical track and three horizontal, while β costs one horizontal and two vertical tracks.

VERTICAL ROUTE

(a)
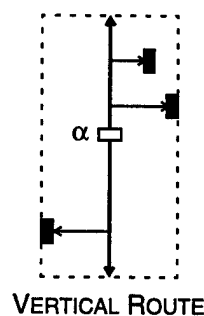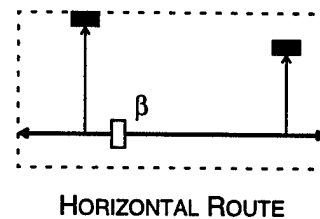
HORIZONTAL ROUTE

(b)

Figure 4.13 Internal routing model.

Due to the vertical aspect ratio of net α (a), the internal route model will generate one vertical trunk and three horizontal branches (one per sink). Net β is implemented with one horizontal trunk and two vertical branches (b).
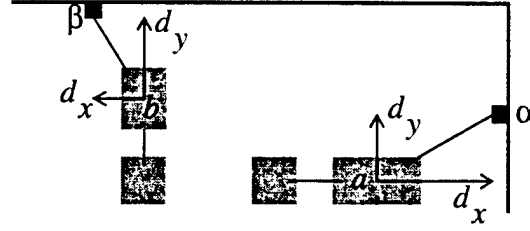
Figure 4.14   Effect of fixed pin locations.

Pin α imparts a vertical and horizontal force on module a. Similarly β imparts a vertical and horizontal force on module b. The vertical force on module a and the horizontal force on module b unduly influence the placement.

Once the supply and demand values for a region are determined, their difference determines the number of extra routing tracks required and therefore, the amount of region growth (w_R^v and w_R^h) required to satisfy demand.

$$w_R^v = 1 - \frac{(R_s^v - R_d^v) \cdot g^v}{\Delta x_R}$$

$$w_R^h = 1 - \frac{(R_s^h - R_d^h) \cdot g^h}{\Delta y_R}$$

(4.11)

4.2.5   Relaxed pins

The task of optimally assigning pin locations to a large block of standard cells is NP-complete. Consider the fact that the optimal pin location permits an optimal placement which is a known NP-complete problem. While classical top-down floorplanning does aid in establishing the locations of pins on a block, it does not consider the resultant effect on the block. A poor ordering of pins could, therefore, unduly "torque" the block thus creating congested regions. This effect can be very pronounced in quadratic placers, where the placement solution, ultimately, is derived from the locations of the pins. From Eq. 4.3, it is known that $\Phi(x)$ and $\Phi(y)$, are functions of $d_x^T x$ and $d_y^T y$ respectively, with each fixed pin imparting a d_x and d_y component to the layout. Thus pin α in Figure 4.14 pulls module a up as well to the right. It is obvious that module a will be unduly moved to
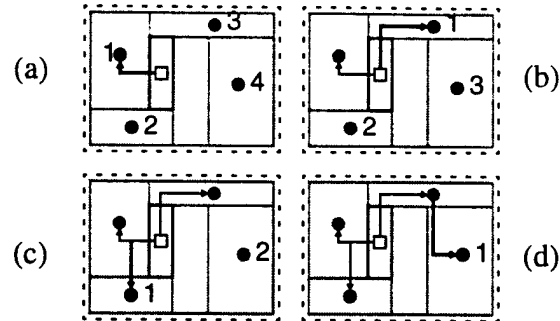
Figure 4.15 A* Router (in region center mode).

The A* router initially orders the sinks by their distance to the source (a). The first sink is selected and routed to the source. The ordering is redetermined and the next sink is routed to the existing tree (b). This process continues (c), and terminates when all sinks are connected (d).

minimize the vertical force on it. This effect can be nullified by setting the d_y component due to α (and likewise the d_x component for β) to zero. This has the effect of allowing α to track a in the y direction, and β to track b in the x direction. In experiments, this approach produced reductions wirelength of as much as 7% for quadratic placement and 10% for linearized quadratic placement. Similar reductions in area were achieved, thus showing a reduction in congestion.

4.3 Implementation and Results

4.3.1 The Region Router

Potential region router algorithms are numerous. The router should accurately estimate supply and demand while being computationally efficient. Furthermore, it is critical that the region router used in this methodology match the characteristics of the detailed router. Discrepancies between the characteristics of the two can lead to incorrect assumptions, causing the placer to create an inappropriate placement for the final route. In this preliminary implementation, routing on the region-based graph was performed by an A* algorithm [11]. The algorithm initially sorts the sinks in order of manhattan distance to
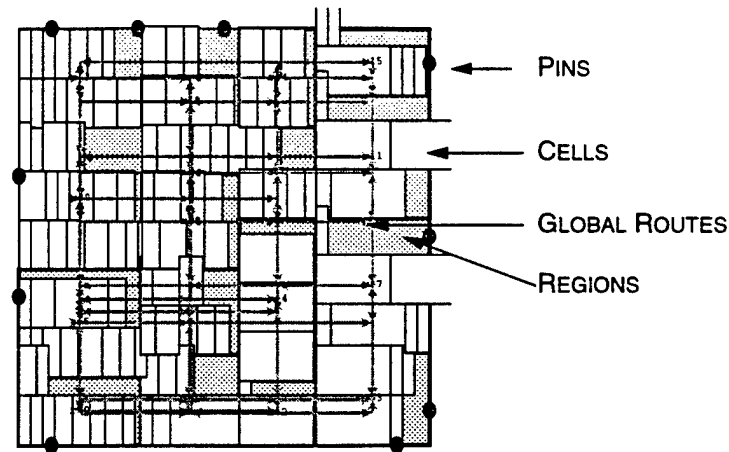
Figure 4.16 Global router on s298 in region center mode.

The sink locations for the global routes (in green) are determined by their region centers. Individual cells are snapped into the regions for clarity.

source. The farthest sink is selected and routed to the nearest edge on the current tree (Figure 4.15 and 4.16). The tree is then updated with the new edges and the ordering of the remaining nodes is redetermined. The process terminates when all nodes are connected. In this methodology, it is not essential for the router to be congestion aware, as the placer will yield solutions that permit direct routes.

Two implementations of the sink lists can be considered: region-based and cell based. In the former, region centers are the sink locations for inter-region nets, while, in the latter, cell locations are used. An advantage of using region centers is the potential reduction in the number of sinks. However, this is accompanied by a loss of detail, as certain regions may not be intersected. This effect is reduced at finer levels of partitioning until the region centers correspond very closely to the cell locations. In the current implementation, routes are initially performed to cell locations. At later stages, a switch to region mode occurs, thus maintaining accuracy and efficiency.

4.3.2 Implementing the Supply-Demand Algorithm

The following algorithm lists the steps required to compute the congestion on a region based routing grid.

Inputs: Netlist, Regions and cell coordinates

Outputs: Region weights

```
1.    sort cells by region number
2.    foreach region R do
3.            snap region's cells to region
4.            compute feedthroughs
5.            foreach internal route r do
6.                    if horizontal route
```
7. $$R_s^h -= 1 , \quad R_s^v += |r| - 1$$
```
              else
```
8. $$R_s^v -= 1 , \quad R_s^h += |r| - 1$$
```
              endif
          endfor
9.    endfor
10.   generate external routes: E
11.   foreach external route e do
12.           foreach region r ∈ e ∩ R
13.                   update external route cost
          endfor
      endfor
14.   foreach region R do
15.           compute region weights
      endfor
16.   return region weights
```

4.3.3 Results

The congestion-driven algorithm, implemented in C, was incorporated into an existing quadratic placer provided by Cascade Design Automation[1]. The numerical routines for sparse matrix operations [128, 45] were obtained through use of the Meshach

---

1. Now owned by DUET Corporation.

Table 4.2 Benchmark information.

The following netlists were obtained from the ISCAS-89 benchmark suite [21].

Module	cells	edges	Module	cells	edges
s298	133	404	s1494	653	2057
s444	202	595	s5378	2958	7527
s641	398	974	s9234	5825	14251
s1238	526	1602	s13207	8620	21122
s1423	731	2042	s15850	10369	25207
s1488	659	2057	s35932	17793	49517

library [39]. A subset of netlists from the ISCAS-89 benchmark (Table 4.2) were placed and subsequently analyzed.

Global flow uniformity is expressed as the average supply-demand excess over all edges, and local flow uniformity as its standard deviation. When comparing two placements, a lower average reflects lower global congestion and a tighter standard deviation (smaller) is representative of greater uniformity. The goal of a routing resource driven placement algorithm should be to minimize the global, local and maximum flow excess. The final analysis is performed on a uniform grid. Δ_{DS}, the demand-supply excess is computed for each grid node and is processed to determine the global maximum: $max\{\Delta_{DS}\}$, average: $\overline{\Delta_{DS}}$ and standard deviation: $\sigma(\Delta_{DS})$. $wlen$ is the total route length as determined by the A* global router. For the final analysis, the A* router generates internal routes as well.

Table 4.3 lists the improvements in congestion, as measured by $max\{\Delta_{DS}\}$, $\overline{\Delta_{DS}}$ and $\sigma(\Delta_{DS})$ over the benchmark set. With larger designs such as s15850 and s35932, the local reductions in congestion tend to have less effect on the global average and uniformity. Some benchmarks show an increase in congestion. This is due to the inaccuracy of the internal line-probe route model.

Table 4.3 Improvement with congestion analysis over quadratic placement.

Each circuit is overlaid with a grid and demand-supply excess (Δ_{DS}) is computed for each grid-cell. The improvement in total wire-length, maximum Δ_{DS}, average Δ_{DS} and variance of Δ_{DS} over general quadratic placement is shown. The average improvement over the entire suite is show in the last row.

Benchmark	Percent Improvement			
	$wlen$	$max\{\Delta_{DS}\}$	$\overline{\Delta_{DS}}$	$\sigma(\Delta_{DS})$
s298	0%	0.5%	2.2%	3.7%
s444	4%	14.9%	26.5%	17%
s641	6%	5.0%	21.8%	15.6%
s1238	2%	9.1%	18.7%	15.6%
s1423	2%	15.7%	6.6%	12.4%
s1488	2%	8.5%	11.1%	-1.9%
s1494	2%	7.8%	-6.0%	6.8%
s5378	5%	8.1%	8.3%	3.0%
s9234	2%	2.4%	2.5%	4.1%
s13207	4%	-8%	7%	3.0%
s15850	-1%	5.8%	0.4%	-0.8%
s35932	0%	12.8%	-0.1%	4.4%
Average:	2.3%	6.8%	8.2%	6.9%

Table 4.4 shows the effectiveness of relaxed pins in wire length reduction on a sample set of small to medium-sized netlists. The effect of the enhancement on both quadratic and linearized quadratic placement is shown. The wirelength and area reductions were obtained after detailed routing using an area router provided by Cascade Design Automation.

4.4 Summary

The relationship between crosstalk and wiring congestion was investigated. Various designs, ranging from large standard-cell groups to full microprocessors, were evaluated. For a given placement, the correlation between congestion and timing criticality

Table 4.4 Improvement in area and wirelength due to relaxed pins.

The effect of the relaxed-pins on both quadratic and linearized quadratic-placement [135] is shown. Wirelength and area statistics were obtained after using an area router provided by Cascade Design Automation.

Module	Quadratic placement		Linear Placement	
	$wlen_q(\mu)$	$area_q(mils)$	$wlen_l(\mu)$	$area_l(mils)$
s298	0.8%	-8.4%	9.3%	6.5%
s641	-2.2%	0.2%	10.4%	3.1%
s1238	7.1%	5.7%	-3.7%	-8.8%
s1488	3.6%	2.5%	1.9%	2.3%
s1494	2.1%	-2.8%	-1.9%	2.4%
s5378	7.2%	4.2%	3.2%	-1.1%
s9238	7.0%	4.5%	3.9%	-2.4%

was determined; in all cases, a high degree of correlation was observed. For the benchmark set, 50% of the critical paths have more than 50% of their length passing through regions of high congestion.

A methodology for integrating congestion into quadratic placement is presented. The inclusion of a growth matrix into the quadratic placement formulation permits the use of a routing model to resolve congestion while concurrently minimizing wirelength. Further reduction in congestion, reflected in wirelength and area reduction, is obtained by the relaxation of pin constraints. This implementation with an A* search router and line-probe heuristics showed up to 20% reductions in demand-supply excess over quadratic placement without congestion analysis.

CHAPTER V

CROSSTALK CONSTRAINED ROUTE EMBEDDING

A route embedder minimally modifies a set of route structures while optimizing area and maintaining the form of the original route topology. A Linear Program formulation of route embedding for global routes with crosstalk constraints is implemented in this chapter. By satisfying timing and noise constraints at individual critical sinks, it permits the creation of a routing solution having no crosstalk violations. The standard method of addressing crosstalk by uniformly minimizing adjacent capacitance is rejected in favor of a method which uses an accurate crosstalk model that satisfies slack and noise margins, at the critical sinks. This is achieved by appropriating space or ground shields around critical traces. Routing capacity constraints are enforced to guarantee a routing solution.

A detailed route embedder is also presented. The theoretical limits of crosstalk constrained route embedding are investigated through it. Implemented as a mixed linear program, it allows the solver to explicitly capture the behavior of a detailed router with linearized crosstalk constraints.

5.1 Overview

Most methodologies address crosstalk after generating a detailed routing. This is followed by analysis and re-routing where necessary. However, the complexity of detailed routing, parasitic extraction and crosstalk analysis followed by subsequent rip-up and re-route 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 and receiver noise margins, these solutions will be overly pessimistic and prone to suggesting an extreme area-delay trade-off.

Published approaches for minimizing crosstalk fall into three distinct categories. The first approach modifies existing route structures to meet capacitance constraints [47, 88]. Unfortunately, this methodology is incapable of re-routing, so is limited to local improvements. The solution, therefore, will be limited by the initial detailed route. The second approach involves routing while minimizing coupling capacitance [64, 63]. Limited by the complexity of detailed routing, this approach is forced to compromise on the quality of crosstalk analysis. Both are confined to pre-determined route topologies and lack temporal information about the signals. The final method addresses crosstalk by minimizing coupling capacitance during global [121] and area routing [108]. They rely upon rip-up and re-route to satisfy crosstalk constraints. The simplified models of crosstalk for the aforementioned methods preclude a performance driven approach with guaranteed convergence.

This chapter introduces a new approach to minimizing crosstalk. It satisfies constraints that permit a trade-off between physical and temporal proximity, thereby enabling a routing solution with no crosstalk violations. The chapter begins with an introduction to route embedding, followed by a brief description of the empirically derived crosstalk model and its applicability to route embedding. Next, the route embedding-based methodology itself is presented, followed by specific sections on the global and detailed route embedding. The chapter concludes with experimental results on various benchmark circuits and route topologies.

5.2   Route Embedding

The complexity of detailed routing [72], makes concurrently minimizing crosstalk constraints prohibitive. This complexity is evidenced by a sampling of the tasks performed by a detailed router, which include:

1. Creating minimum length connections,

2. Adhering to design rules and electrical correctness,

3. Minimizing the number of layers, bends and vias,

4. Enforcing wire widths, and

5. Following the global tree topologies while avoiding blockages.

Simplifying the route topology through the use of channel or switchbox routing, for example, is the predominant method used to minimize the complexity of generating a detailed routing, while satisfying crosstalk constraints. This does, however, limit the applicability of the router in a design methodology. Other solutions attempt to dictate a detailed routing through assignment or adjustment of physical adjacency. These schemes are prone to increasing routing area because they do not consider wireability.

Route embedding is a means of effectively 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. This way, only the fundamental task of generating non-overlapping route structures needs to be modeled. Route embedding works by attempting to express a routing solution without overlapping wires. This feature can be exploited to make route embedding computationally cheaper than a detailed routing. Abstraction allows the route embedder to target its analysis toward accurately satisfying timing and noise constraints. This is preferable to the previous methods that effectively "searched" for a valid solution using

rip-up and re-route procedures. Lastly, the route embedder simultaneously processes a set of route trees, permitting a global view of the task.

In this chapter, two implementations of route embedding with crosstalk constraints are investigated. The first, applicable at the global routing stage, expresses a routing solution through a set of desired spacings for multiple critical nets. The second expresses non-overlap between route-trees through an zero-one Linear Program (LP) with linearized crosstalk constraints. Both models of route embedding accurately trade-off physical and temporal proximity to permit the creation of crosstalk free solutions.

5.3   Crosstalk Impact Model

For a given coupling length l and spacing s, Figure 5.1 shows that the change in wire delay (τ) and noise (ζ) can be empirically curve-fitted to

$$\tau = \frac{\alpha \cdot l^2}{s} \; and \; \zeta = \frac{\beta \cdot l^2}{s} \tag{5.1}$$

where α and β are curve-fitted technology parameters that are dependent on the coupling waveforms. r_v is the victim-driver's output impedance. a_e and v_e are the edge rates of the waveforms. Let δa be the arrival time difference (as measured between the 50% point of the signals) between the aggressor a and victim v normalized by the victim's edge rate. α and β are non-linearly related to the arrival time and edge rates; we tabulate these values for table-lookup. When the nets are temporally orthogonal (i.e. their transitions do not overlap: $|\delta a| \geq 0.5$), noise becomes the limiting factor and a β value is obtained by indexing the β-table with a_e and r_v. When the transitions on the adjacent nets do interact, then there will be an impact on delay, so 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) and δa is
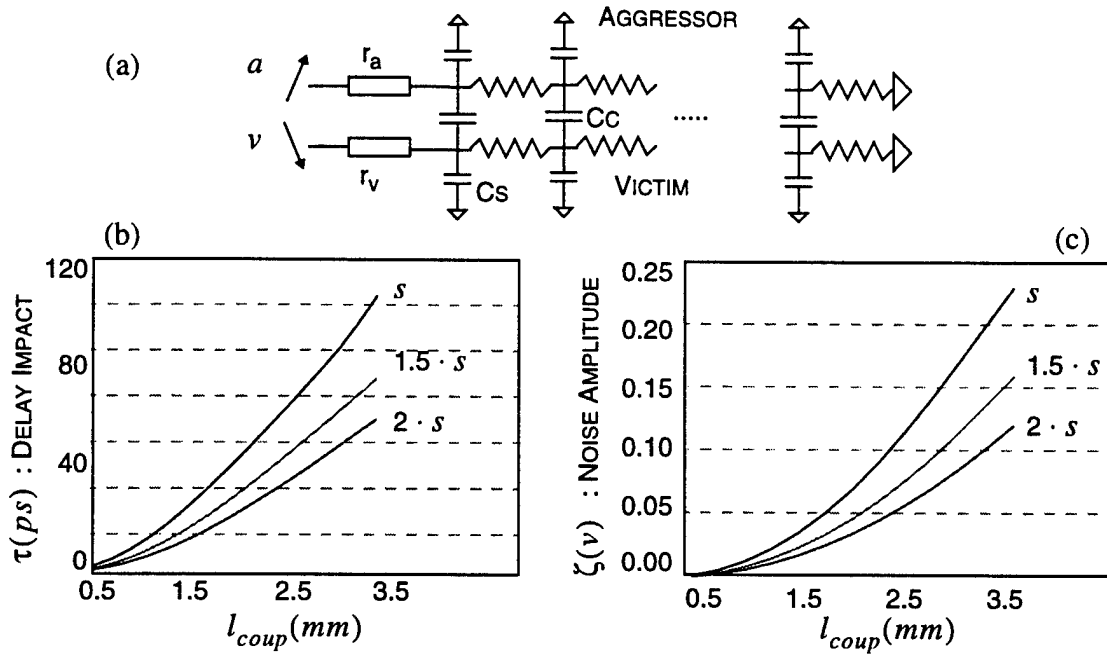
Figure 5.1 Model for crosstalk with simulated results.

The schematic in (a) represents two coupled distributed-RC conductors. Simulation of the delay change in the victim line, due to the aggressor is plotted in (b). Simulation of the amplitude of noise injected onto the victim by the aggressor is plotted in (c). Section 2.3.1 on page 43 describes the simulation methodology in detail.

bound to $-0.5\ldots0.5$. The bounding condition applies for both tables. Values within the tables are determined by a linear-interpolation. A computationally efficient table lookup will therefore predict the curve fit parameters for all crosstalk behavior under these predefined conditions.

5.4 Crosstalk Constrained Route Embedding Methodology

The proposed methodology, depicted in Figure 5.2, 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, edge-rate) at each gcell. As described in Section 2.4.3, the RC trees are modeled using an exponential

```
┌─────────────────┐           ┌─────────────────┐
│  GLOBAL ROUTES  │─────┐  ┌─▶│  CRITICAL PATHS  │
└─────────────────┘     │  │  └─────────────────┘
         │              │  │           │
         ▼              │  │           ▼
┌─────────────────┐     │  │  ┌─────────────────┐
│    GRIDED RC    │     │  │  │ GLOBAL EMBEDDING │
└─────────────────┘     │  │  └─────────────────┘
         │              │  │           │
         ▼              │  │           ▼
┌─────────────────┐     │  │         ◇ VIOLATION? ◇────No──▶┌─────────────────┐
│ TIMING ANALYSIS │─────┘  │                               │  DETAILED ROUTER │
└─────────────────┘        │           │                   └─────────────────┘
         ▲                 │          YES                           ▲
         │                 │           │                            │
┌─────────────────┐        │  ┌─────────────────┐                   │
│LOCAL MODIFICATION│       │  │    SHIELDING    │                   │
└─────────────────┘        │  └─────────────────┘                   │
         ▲      Yes        │           │                            │
         └──────────────◇ VIOLATION? ◇──────No────────────────────┘
```
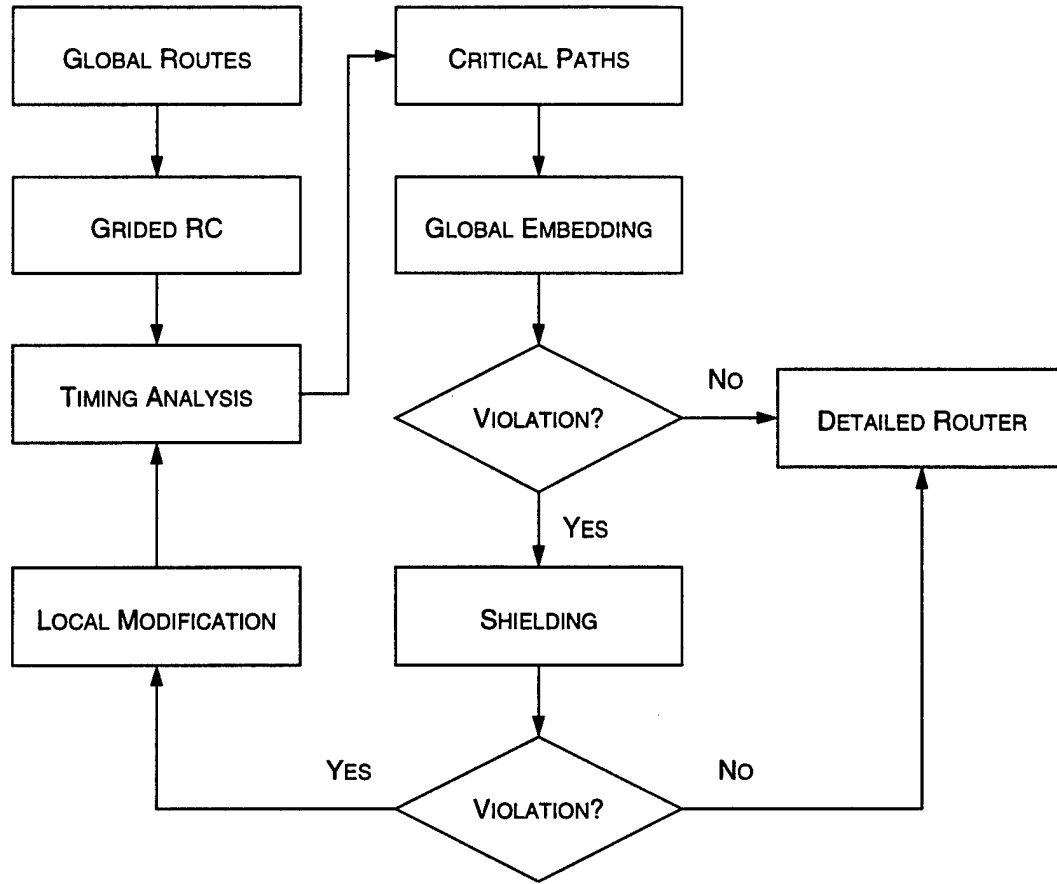
Figure 5.2   Crosstalk Constrained Route Embedding methodology.

After timing analysis of the global routes, paths that violate their specifications (timing and noise-margins) are processed by the global route embedder to satisfy crosstalk constraints. Unsatisfied paths are resolved through appropriate shield insertion and local modifications (re-routing).

input extension to an explicit three-pole model [109]. 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 solver in-turn processes a Linear Program (LP) 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 re-routed if routing capacity constraints are breached. If crosstalk constraints are not satisfied, the expected
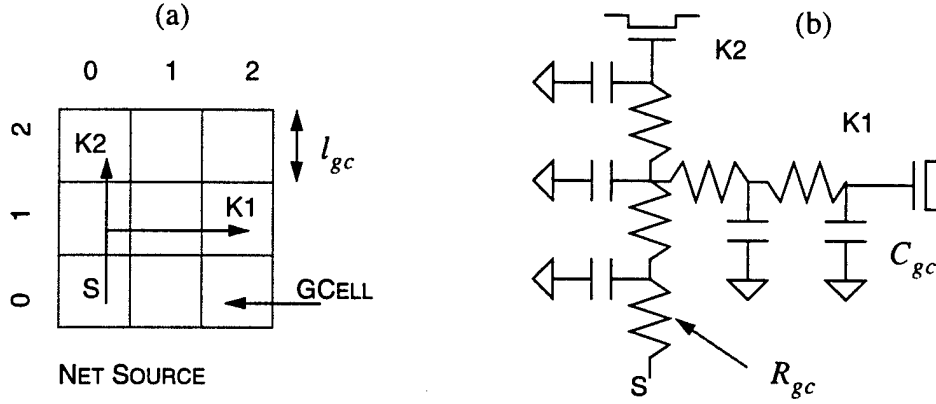
Figure 5.3   Constructing the *RC* equivalent of a global tree.

The net in (a) is overlaid with a gcell based grid. Each gcell traced by the net is replaced by an *RC* component ($R_{gc} \cdot C_{gc}$) (b).

impacts are refined, and a new LP 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.

5.5   Global Route Embedding

5.5.1   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 5.3. These values are calculated as:

$$R_{gc} = \rho^y \cdot \frac{l_{gc}}{A_{net}} \text{ and } C_{gc} = \frac{\varepsilon_\kappa \cdot l_{gc} \cdot w_{net}}{h_0^y} + 2 \cdot \frac{\varepsilon_\kappa \cdot l_{gc} \cdot t_0^y}{s_0^y} \tag{5.2}$$

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-section 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 the timing engine to determine the arrival time and edge rate at each gcell. The timing engine also computes the slack at each sink.

5.5.2 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. 5.1 to compute the expected delay impact at each sink. The expectation value of a function $f(x)$ [1], is defined to be

$$\langle f(x) \rangle = \sum_x f(x) \cdot P(x) \tag{5.3}$$

where $P(x)$ is the Probability Distribution Function (PDF) for x and $\sum_x P(x) = 1.0$. Define 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. 5.3 $\pi_{k_2}^n = \{g_{0,0}, g_{1,0}, g_{2,0}\}$, where $g_{i,j}$ is the gcell indexed at row i and column j. $L_k^n = |\pi_k^n|$ is the path length from source to sink k. Let 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. 5.1 and 5.3,

$$\langle \tau_k^n \rangle = \frac{\langle \alpha^{n,k} \rangle \cdot (L_k^n \cdot l_{gc})^2}{\langle s^{n,k} \rangle} \tag{5.4}$$

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 coupling at each gcell along π_k^n.

---

1. For discrete x only. If x is continuous then $\langle f(x) \rangle = \int f(x) \cdot P(x) \cdot dx$
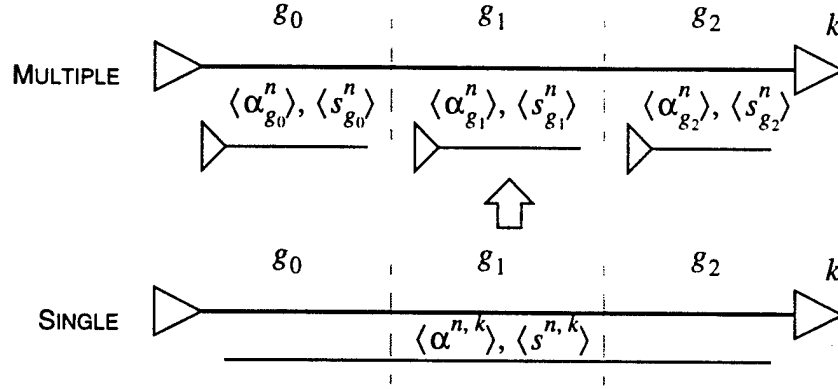
Figure 5.4   Using expected values to approximate multiple coupling.

The expected delay-impact parameter $\langle \alpha^{n,k} \rangle$ is the average of the gcell impact parameters.

Figure 5.4 motivates the translation of Eq. 5.4 to account for these multiple couplings. This is achieved by,

$$\langle \tau_k^n \rangle = \left(\sum_{c \in \pi_k^n} \frac{\langle \alpha_c^n \rangle}{\langle s_c^n \rangle} \cdot \frac{1}{L_k^n} \right) \cdot (L_k^n \cdot l_{gc})^2 \qquad (5.5)$$

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

where $\langle \alpha_c^n \rangle$ is the expected α-parameter for the net within the gcell c, and $\langle s_c^n \rangle$ is the expected spacing around n 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,

$$\langle \alpha_c^n \rangle = \sum_{e \in \phi_c} \alpha^{n,e} \cdot \frac{1}{|\phi_c|} = \overline{\alpha_c^{n,\phi_c}} \qquad (5.6)$$

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. 5.5 can now be reduced to

$$\langle \tau_k^n \rangle = L_k^n \cdot \left(\sum_{c \in \pi_k^n} \frac{2 \cdot \overline{\alpha_c^{n, \phi_c}}}{\langle s_c^n \rangle} \right) \cdot l_{gc}^2 \qquad (5.7)$$

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.

5.5.3 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 [33]. It becomes clear that 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.

$$\langle \zeta_k^n \rangle = \sum_{c \in \pi_k^n} \langle \zeta_c^n \rangle \qquad (5.8)$$

Similar to Eq. 5.6, the expected noise-impact $\langle \zeta_c^n \rangle$, on the net n in gcell c is dependent on the noise impact PDF. Therefore,

$$\langle \zeta_c^n \rangle = \sum_{e \in \phi_c} \zeta^{n, e} \cdot \frac{1}{|\phi_c|} = \overline{\zeta_c^{n, \phi_c}} = 2 \cdot \frac{\overline{\beta_c^{n, \phi_c}} \cdot l_{gc}^2}{\langle s_c^n \rangle} \qquad (5.9)$$

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 nose-margin of sink k on net n. All sinks with $M_k^n / \langle \zeta_k^n \rangle \geq 1.0$ are added to Θ.

5.5.4 Computing An Embedding

S_k^n and M_k^n as the acceptable slack and noise-margin 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 an 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:

$$min\left\{ \forall k \in \Theta, \sum_{c \in \pi_k^n} \langle s_c^n \rangle \right\} \tag{5.10}$$

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

$$L_k^n \cdot \left(\sum_{c \in \pi_k^n} \frac{2 \cdot \overline{\alpha_c^{n, \phi_c}}}{\langle s_c^n \rangle} \right) \cdot l_{gc}^2 \leq S_k^n \text{ and} \tag{5.11}$$

$$\left(\sum_{c \in \pi_k^n} \frac{2 \cdot \overline{\beta_c^{n, \phi_c}}}{\langle s_c^n \rangle} \right) \cdot l_{gc}^2 \leq M_k^n \tag{5.12}$$

Eq. 5.11 and 5.12 guarantee the slack (S_k^n) and noise margin (M_k^n) requirements at each sink. Each gcell with a critical path is constrained by its available supply of routing tracks:

$$\forall c \in \pi_k^n, \sum_{n \in \phi_c} (\langle s_c^n \rangle + w_c^n) \leq l_{gc} \tag{5.13}$$

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 $\forall (n, c)$, $\langle s_c^n \rangle \geq s_0^y$.
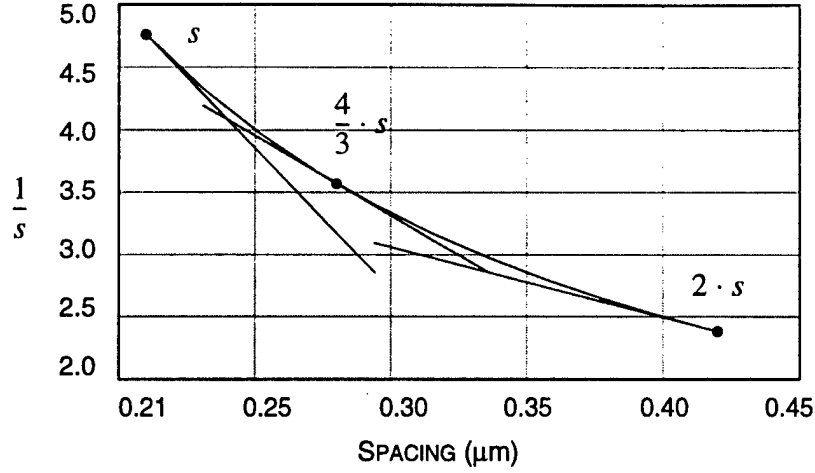
Figure 5.5   Linearization of $1/\langle s_c^n \rangle$.

The $1/\langle s_c^n \rangle$ curve is linearized through three tangents. The tangents are derived from the first order Taylor's approximation to $1/x$ centered at $\{1, 4/3, 2\}$.

Both Eq. 5.11 and Eq. 5.12 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, $\frac{4}{3} \cdot s_0^y$ and $2 \cdot s_0^y$ (see Figure 5.5). By introducing σ_c^n as a slack variable for each $\langle s_c^n \rangle$, Eq. 5.11 can be transformed into

$$L_k^n \cdot \left(\sum_{c \in \pi_k^n} \overline{\alpha_c^{n, \phi_c}} \cdot \sigma_c^n \right) \cdot l_{gc}^2 \le S_k^n \ \text{ and} \tag{5.14}$$

$$\left(\sum_{c \in \pi_k^n} \overline{\beta_c^{n, \phi_c}} \cdot \sigma_c^n \right) \cdot l_{gc}^2 \le M_k^n \tag{5.15}$$

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

$$\sigma_c^n \ge \frac{2 \cdot (x \cdot s_0) - \langle s_c^n \rangle}{(x \cdot s_0)^2} \cdot \langle \alpha_c^n \rangle \qquad ;x \in \left\{ 1, \frac{4}{3}, 2 \right\} \tag{5.16}$$

This LP formulation requires $2 \cdot \sum_{\forall (n, k) \in \Theta} L_k^n$ 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 leads to a total of

$$\forall (n, k): \qquad 3 \cdot \sum L_k^n + 2 \cdot |\Theta| + |\Gamma| \qquad (5.17)$$

linear constraints.

5.5.5 Resolving an Unfeasible Embedding

Figure 5.6 illustrates the process by which an unfeasible global embedding LP is resolved. A violation occurs when a path does not meet its slack or noise margin requirements. 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 $\overline{\alpha_c^{n, \phi_c}}$ and $\overline{\beta_c^{n, \Phi_c}}$ set to zero. p_0^y is
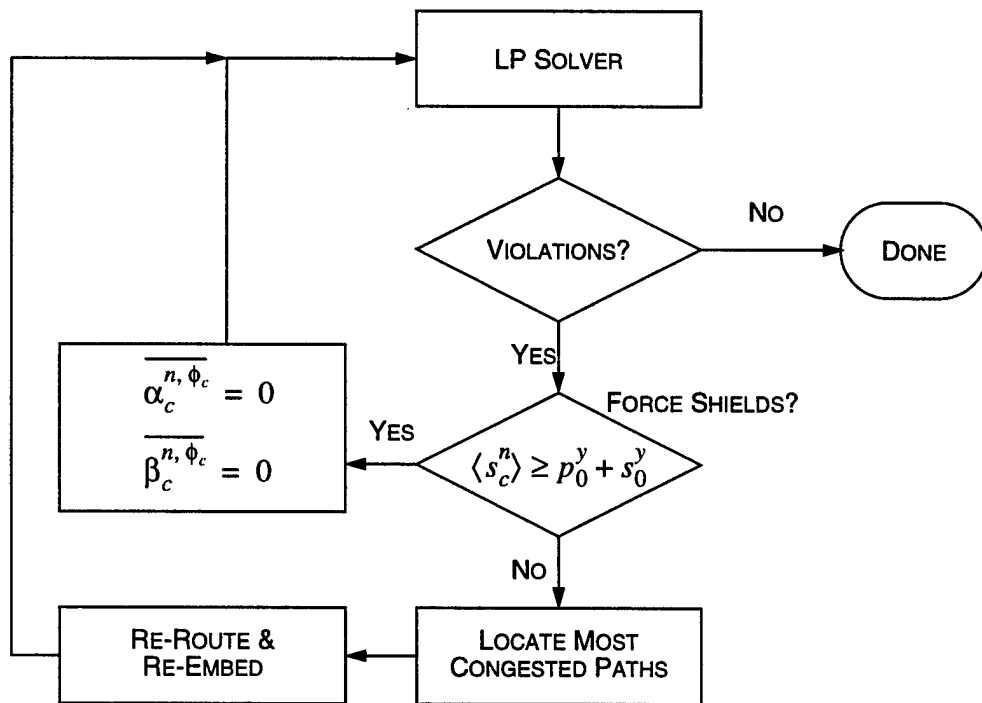
Figure 5.6   Iterating to satisfy violated crosstalk constraints.

A violation occurs when a path does not meet its slack or noise margin require-
ments. Shields are inserted to reduce the impact of crosstalk on unsatisfied
paths. The Linear program is updated and solved. Paths that do not meet con-
straints are minimally re-routed through un-congested gcells. Finally, a new
embedding is generated and solved.

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. Then slack and noise-margin constraints are re-evaluated. If constraints are still not met, sections of violated paths, passing through regions of congestion are minimally re-routed through un-congested gcells. After re-timing, the impact parameters are updated and a new embedding LP is generated and solved for. This process continues until the crosstalk constraints are satisfied.

5.6 Detailed Route Embedding

This section investigates the theoretical limits of route embedding and determines its complexity and behavior.

Any description of routing which contains route-edge relationships (adjacencies) is of invaluable aid in addressing crosstalk. In the previous section, we showed that this information could be approximated (modeled) by computing expected spacings. The calculations that lead to expected impacts are a result of a "smoothing function" (the impact PDF). Since it only models the probable adjacency, an explicit trade-off between noise and delay cannot be made.

The detailed route embedder, on the other hand, models the combinatorial complexity of a detailed router by limited reconfiguration of the input route topologies. By doing so, it is able to explicitly model the adjacencies and therefore create an exact impact PDF. The detailed route embedder is formulated as a Mixed Integer Linear Program (MIP). This allows the solver to explicitly capture the behavior of a detailed router. Linearized crosstalk constraints are appended to the formulation. Any feasible solution of this MIP will, by definition, satisfy all crosstalk constraints. All integer variables are binary (0/1). Each change in a binary variable represents an attempt to reconfigure the route topologies. Permuting through the possibilities effectively captures the combinatorial behavior of a

ALTERNATIVE I   ALTERNATIVE II

ORIGINAL ROUTE TOPOLOGY

VERTICAL OVERLAP

MINIMAL CHANGE TO SATISFY CROSSTALK

ALTERNATIVE III

HORIZONTAL OVERLAP

MINIMAL CROSSTALK BUT MAXIMUM CHANGE
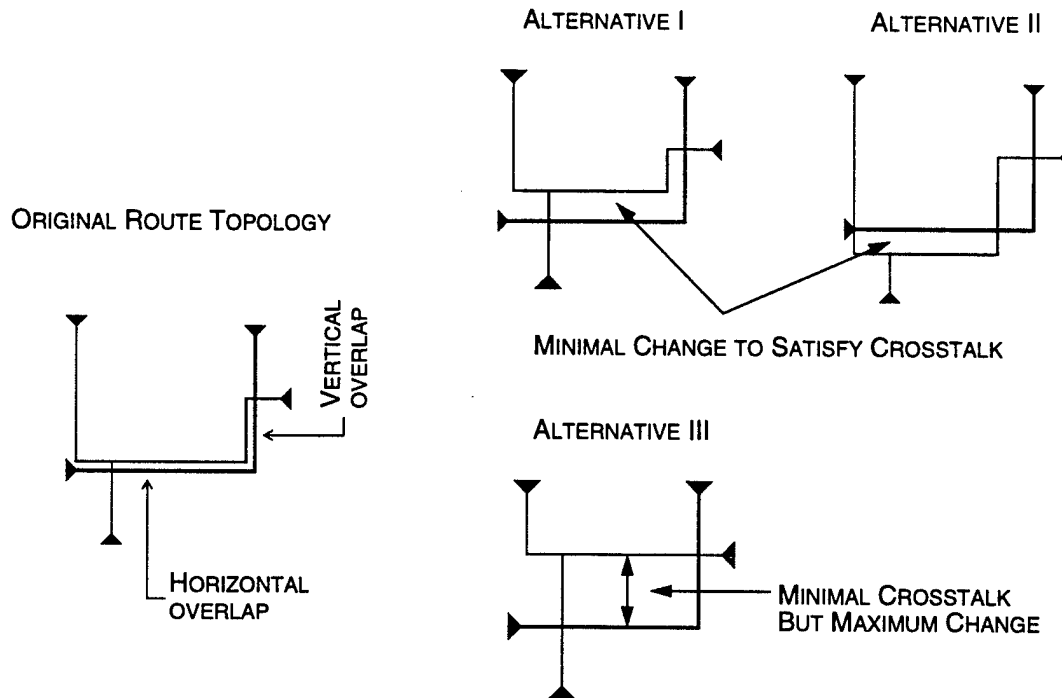
Figure 5.7   Modifying route topology while satisfying constraints.

Alternatives I and II depict minimal topological changes that satisfy crosstalk constraints in the original route topology. The third alternative minimizes the impact of crosstalk at the expense of maximum topological change. The first two alternatives are preferred.

detailed router attempting to define non-overlapping adjacencies. The MIP uses a "minimal change" objective function which constrains the solution space and speeds up the search (Figure 5.7).

As with any linear program, the detailed embedder shown in Figure 5.8 is implemented as a set of linear constraints on a linear objective function. The linear program solver CPLEX [15] is used to solve the system. The constraints are classified into topological and crosstalk constraints. Topological constraints express the connectivity of the route trees which must be binary and Manhattan (only 90° angles). Through binary variables, the topological constraints permit the expression of the various feasible route permutations. The crosstalk constraints are sub-divided into edge and path-based constraints. The edge-based crosstalk constraints are pivotal in constraining the overlap and spacing between
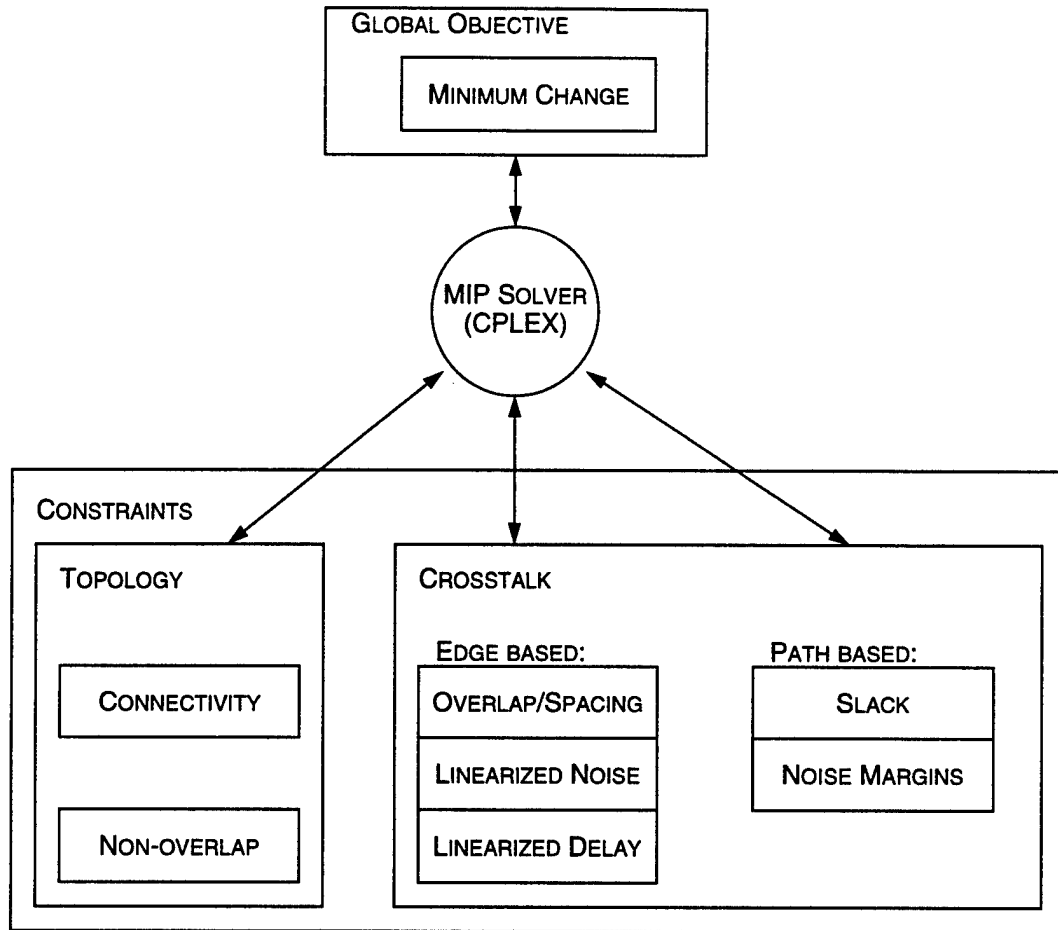
Figure 5.8   Structure of the detailed embedder.

CPLEX is used to satisfy the linear program for detailed embedding. The minimum change objective function is constrained by topological and crosstalk constraints. Topological constraints enforce the tree structures and guarantee non-overlap of edges. The crosstalk constraints enforce timing and noise-margin specifications by constraining the overlap and spacing between edges.

edges to satisfy linearized crosstalk constraints. The path-based crosstalk constraints ensure that the sink slack and noise margin requirements are met. The following sections describe the setup and generation of these constraints in greater detail.

5.6.1   Expressing Route Structures

As illustrated in Figure 5.9, each route tree is expressed as a set of constraints on edges: $T = \{C, E\}$. In the case of the example tree, $C = \{c_1, c_2, f_1, f_2, f_3\}$ and
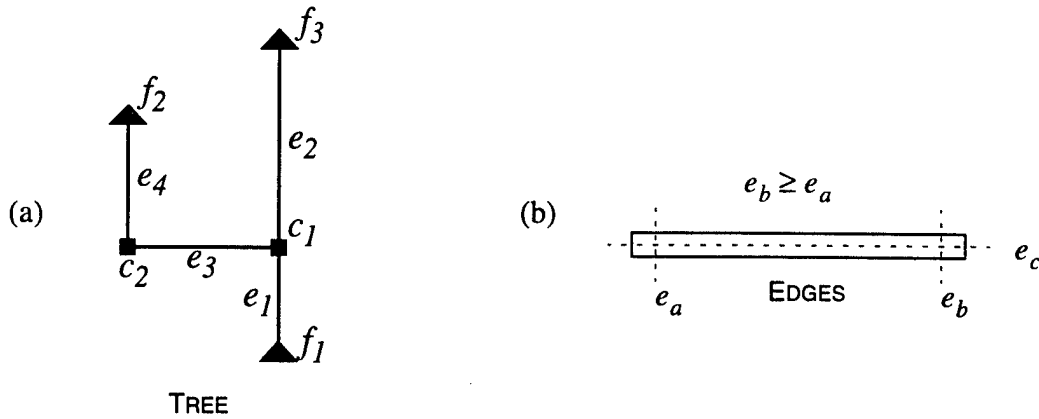
Figure 5.9 Expressing trees and edges in the detailed embedder.

The structure of the tree in (a) is described by the connectivity constraints $\{f_1, f_2, f_3, c_1, c_2\}$ on the edges $\{e_1 ... e_4\}$.

Terms e_a and e_b represent the variable dimension (x) of the horizontal edge in (b). Term e_c varies in the vertical dimension (y).

$E = \{e_1, e_2, e_3, e_4\}$. The edges of the route tree correspond to physical wires in the design. Since the edges describe manhattan geometries, each edge $e \in E$ can be represented by four variables: $\langle e_a, e_b, e_c, e_w \rangle$. A distinction between horizontal and vertical edges is made. Variables e_a and e_b describe the varying dimension of the edge. e_c is the value of the constant dimension and therefore represents the y co-ordinate when e is a horizontal and the x co-ordinate when e is vertical. e_w is the width of the edge. The route model is gridless, thus permitting greater flexibility for precise solutions to the crosstalk constraints. Each edge possesses a default edge constraint $e_b \geq e_a$. This simplifies the connectivity constraints and the overall formulation.

Each connectivity constraint permits some freedom of movement to its edges, allowing the tree to take on different but similar topologies. Three types of connectivity constraints, which are sufficient for expressing all manhattan route structures (when rotations are taken into account), are defined. They are: F, L and T (see Figure 5.10). For example, the L constraint limits a horizontal/vertical pair of edges (h, v), by asserting $h_a = v_c$ and
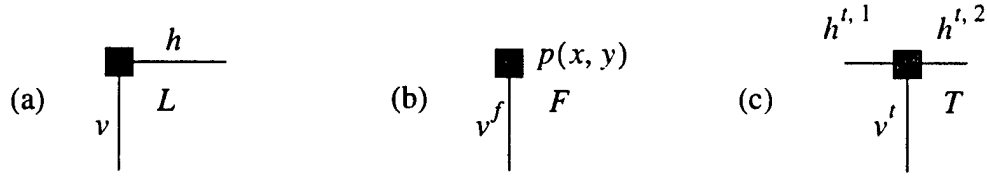
Figure 5.10 Connectivity constraints for the detailed embedder.

The L constraint (a) asserts $h_a = v_c$ and $v_b = h_c$. The F constraint (b) fixes v_c^f to $p(x)$ and v_b^f to $p(y)$. The T constraint (c) asserts that $h^{t,1}$ and $h^{t,2}$ be horizontal aligned and $v_c^t = h_a^{t,2}$.

$v_b = h_c$. The F constraint fixes one end of an edge (source/sink) and the T constraint ensures connectivity between two orthogonal edges. The flexibility of the topology can be increased by the insertion of L jogs, but, such insertions only increase the problem complexity while offering marginal improvements due to problem constraints.

5.6.2 Enforcing Non-Overlap

To the solver, non-overlap of edges is expressed by representing edge overlap as an unfeasible solution. As shown in Figure 5.11, non-overlap between an edge pair (g, h) can be expressed by asserting $(g_c = h_c) \wedge ((g_b < h_a) \vee (g_a > h_b))$, which states that two
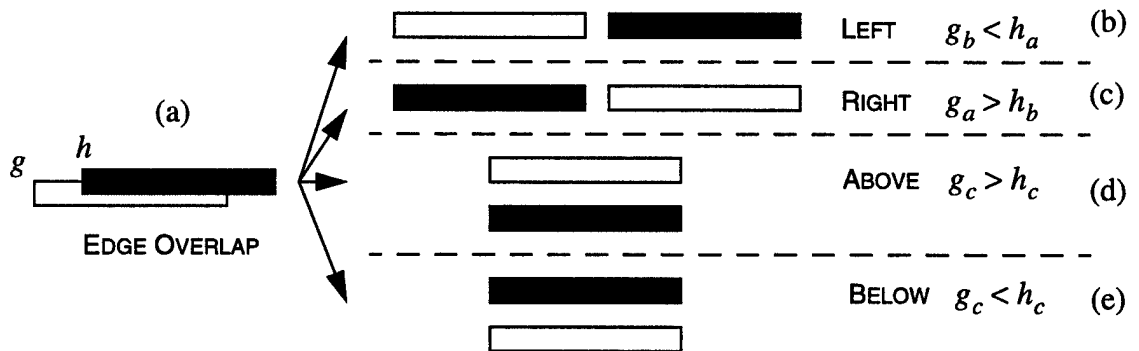
Figure 5.11 Enforcing non-overlap of edges.

The overlap of edges g and h (shaded) is an infeasibility (a). The edges can only possess the same track $(g_c = h_c)$ if g is to the left (b) or right of h (c). Alternatively, g is required to be above (d) or below h (e).

edges can occupy a given track if they are to the right or left of each other. This can be asserted with the aid of four binary variables and the following linear constraints:

$$0 \leq cgt[g, h] + clt[g, h] \leq 1$$
$$0 \leq agb[g, h] + bla[g, h] \leq 1 \qquad (5.18)$$
$$1 \leq cgt[g, h] + clt[g, h] + agb[g, h] + bla[g, h] \leq 2$$

where $cgt[g, h] = g_c > h_c$, $clt[g, h] = g_c < h_c$, $agb[g, h] = g_a > h_b$ and $bla[g, h] = g_b < h_a$. Eq. 5.18 asserts that if $cgt[g, h] + clt[g, h] = 0$ ($g_c = h_c$), then $agb[g, h] + bla[g, h]$ is forced to one. This equation requires g to be left or right of h by constraining $agb[g, h] + bla[g, h] = 1$. The assignment of the four binary variables is a non-linear process which is linearized using the "big-M" notation. For example, $cgt[g, h]$ is set by asserting:

$$g_c - h_c \geq \Delta[g, h] - (1 - cgt[g, h]) \cdot M \quad and \qquad (5.19)$$
$$h_c - g_c - \varepsilon \geq -cgt[g, h] \cdot M$$

where $\Delta[g, h]$ is the spacing between edge g and h. M is a large enough number such that setting $cgt[g, h]$ to one(zero) makes the second(first) equation trivial. Similar constraints are setup for the rest of the binary variables.

Fixed blockages (modules) can now be introduced into the formulation. To the solver, they represent solution infeasibilities for the edges. Similar to the non-overlap of edges, each edge-block pair will require four additional binary variables. Only if the edges can be embedded around blockages are the binary variables for blockages introduced. In each case, the trees are pre-processed to determine the bounds (range) of each edge. The process begins by bounding the range of each edge by propagating the connectivity constraints through its tree. For example, in Figure 5.9, the horizontal edge e_3 is vertically bound between f_1 and f_2. Therefore its vertical range is expressed as $R_v[e_3] = [f_{1x}, f_{2x}]$. Define Ω_h to be the set of horizontal edge-pairs with non-overlap constraints. Each pair of $(h_i, h_j) \in \Omega_h$, satisfies the condition

$$R_v[h_i] \cap R_v[h_j] \neq \varnothing \qquad\qquad (5.20)$$

and are constrained with the non-overlap constraints. Similar sets, Ω_v, and conditions $R_h[v_i] \cap R_h[v_j] \neq \varnothing$ are generated for vertical edges. The edge-blockage pairs are placed in set Φ. Let $\Xi_h \subset \Omega_h$ and $\Xi_v \subset \Omega_v$ be the set of horizontal (vertical) edges that are tagged for crosstalk constraints.

5.6.3 Linearizing Crosstalk Impact

The LP implementation for crosstalk is divided into edge based and path based constraints. The edge based constraints allow the solver to trade-off temporal and physical proximity between a pair of edges. From Section 2.3.2, the linearized expression for crosstalk impact between two coupled lines is implemented as

$$\forall i \in \{1...n\}: \qquad \lambda[g,h] = \gamma_i + \nu_i \cdot \Delta[g,h] + d_i \cdot \tau[g,h] + f_s \cdot M \qquad (5.21)$$

where the three variables Δ, λ and τ represent the spacing, overlap and delay impact between two edges $\{g,h\} \in \Xi_h \cup \Xi_v$. f_s is a binary variable that invalidates the constraint if g and h are separated by more than twice the minimum pitch. This allows for shielding between g and h. Δ is constrained to be greater than the minimum spacing and τ is defined to be greater than zero. γ_i, ν_i, d_i are constants, described in detail in Section 2.3.2. $\lambda[g,h]$, the edge overlap variable, is a non-linear variable defined in Figure 5.12. Its linearization requires two additional binary variables. For brevity, the linearization of λ is omitted. Noise constraints between the crosstalk edge pairs are added using a form identical to that of Eq. 5.21, after assigning the appropriate constraints and substituting ζ for τ.

The path-based crosstalk constraints guarantee the total impact at a sink will not violate slack and noise constraints. Let π_h^k and π_v^k represent the set of horizontal and vertical

$$g$$

$$h$$

| $g_b - h_a$ | $g_b - g_a$ | $h_b - h_a$ | $h_b - g_a$ |

$$\lambda[g, h] = min(g_b, h_b) - min(g_a, h_a)$$
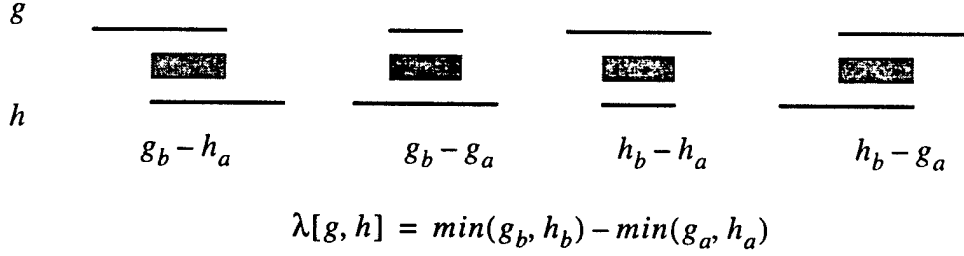
Figure 5.12   Edge overlap as defined by the detailed route embedder.

The shaded area represents the overlap between different configurations of adjacent edges g and h. Term $\lambda[g, h]$ is the length of the overlap region.

edges, respectively, that define the path to sink k. The following constraints assert the slack and noise margin requirements:

$$\sum_{h \in \pi_h^k} \sum_{\{h, e\} \in \Omega_h} \tau[h, e] + \sum_{v \in \pi_v^k} \sum_{\{v, e\} \in \Omega_v} \tau[v, e] \leq S_k \qquad (5.22)$$

$$\sum_{h \in \pi_h^k} \sum_{\{h, e\} \in \Omega_h} \zeta[h, e] + \sum_{v \in \pi_v^k} \sum_{\{v, e\} \in \Omega_v} \zeta[v, e] \leq M_k \qquad (5.23)$$

where S_k and M_k are the slack and noise-margin bounds at sink k.

5.6.4   The Detailed Embedding Objective Function

While there may be a number of feasible non-overlapping solutions, a relative few will be minimally changed from the original route structures. The route-embedder solves for the edge-variables while minimizing their change (Δe_c) as follows:

$$min\left\{ \sum_{e \in E} \Delta e_c \right\} \text{ such that} \qquad (5.24)$$

$$\Delta e_c \geq e_c - \tilde{e}_c \quad \text{and} \qquad (5.25)$$
$$\Delta e_c \geq \tilde{e}_c - e_c$$

where $\tilde{e}_c$ is the original value of e_c. This forces the embedder to adhere to the original route topologies while satisfying crosstalk constraints.

5.6.5 Complexity

The complexity of the detailed embedder formulation lies in the combinatorial cost of permuting edge locations to find a route embedding that satisfies both timing and noise constraints. This becomes apparent through the branch and bound method used by the solver to find a feasible integer assignment for the binary variables. Reducing this combinatorial complexity allows less-optimal solutions to be quickly found.

For each edge-pair, four binary variables are required to implement non-overlap and four more to avoid blockages. Each edge-pair tagged for crosstalk constraints requires two binary variables to help calculate the overlap length and one more to enforce shielding. A quick breakdown of these variables shows that eight binary variables are required to perform the embedding while just three are needed for crosstalk analysis. Furthermore, the binary variables for crosstalk analysis exist solely to cope with the changing state of the routing. Table 5.1 lists the number of these variables by type and constraint-type. Thus, there are $4|\Omega| + 4|\Phi|$ binary variables to guarantee non-overlap. With N nets and an average of $\tilde{e}$ overlapping edges per net, this can be modeled as $O((N \cdot \tilde{e})^2)$. There are $3|\Xi|$ binary variables to assist in computing the crosstalk constraints. As before, $\bar{x}$ edges

Table 5.1 Breakdown of variables in the Detailed Embedder.

Term $|E|$ is the number of edges processed by the detailed embedder. Term N is the number of nets, each having an average of $\tilde{e}$ edges needing non-overlap constraints. Term $\bar{x}$ is the number of edges per net requiring crosstalk constraints.

	Variable Type	
Constraint	**Linear**	**Binary**
Routing	$O(|E|)$	$O((N \cdot \tilde{e})^2)$
Crosstalk	$O((N \cdot \bar{x})^2)$	$O((N \cdot \bar{x})^2)$

per net requiring crosstalk constraints need $O((N \cdot \bar{x})^2)$ binary variables. Each edge $e \in E$, requires three variables $\langle e_a, e_b, e_c \rangle$ to be solved. However, analysis of the connectivity constraints shows that it is sufficient to solve for the e_c variables only. Continuing, for each edge pair in Ξ, variables for overlap-length, spacing and timing (or noise) are required. Therefore, the detailed route embedding formulation requires $O(3|\Xi| + |E|)$ real and $O(N^2 \cdot (\bar{e} + \bar{x})^2)$ binary variables.

The following section documents an attempt to systematically reduce the number of binary variables.

5.6.6 Hierarchical Reduction

A hierarchical approach can be taken to reducing the number of binary variables. First, a weighted undirected impact graph $I = (N, \Gamma)$ is created. Figure 5.13 shows the process on a sample graph. Each node $n \in N$, of I is a net. Each edge of the graph $(n_i, n_j) \in \Gamma$, is weighted with the coupling cost of its two nodes. The lack of an edge implies that the nets never overlap. The coupling cost between two nets n_i, n_j is the
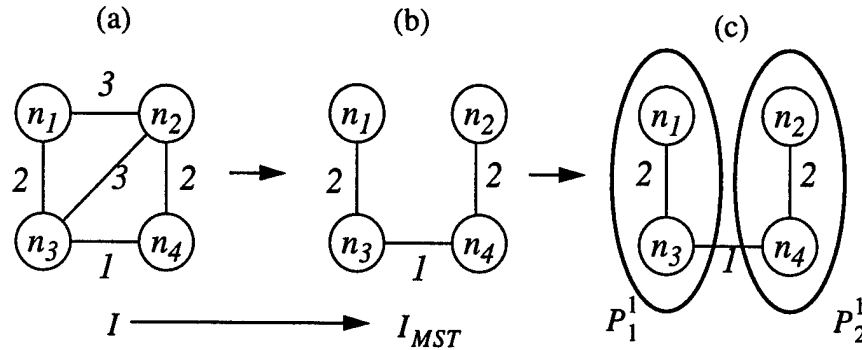
Figure 5.13 Partitioning an Impact graph.

Nodes $n_1 \ldots n_4$ represent nets being embedded. The weight of each edge in the impact graph (a) is the crosstalk cost between its two nodes. I_{MST} is a minimum spanning tree for I (b). A partitioning algorithm then subdivides I_{MST} into P_1^1 and P_2^1 (c). Partition P_1^1 contains n_1 and n_3, while partition P_2^1 contains n_2 and n_4. Each is individually embedded.
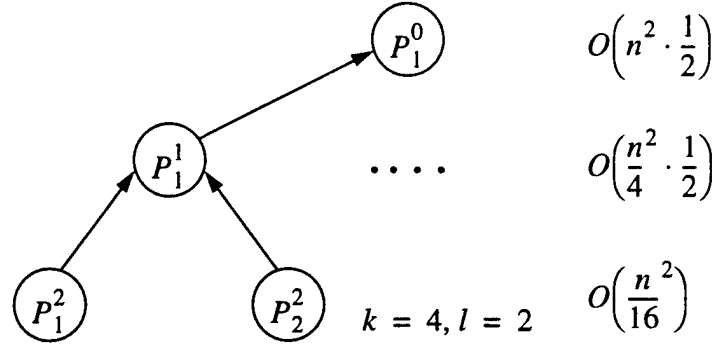
Figure 5.14 Complexity at each level of a bottom-up rebuild.

A subset of the rebuild process is shown. First, at the lowest level ($l = 2$), the four partitions $P_1^2 ... P_4^2$ are individually embedded. At the next level, P_1^2 and P_2^2 are merged to produce P_1^1. Eventually the embedding of P_1^1 is merged with the embedding of P_2^1 to create the final input (P_1^0) for the embedder.

increase in delay (or magnitude of noise) if the two nets were routed maximally adjacent to each other. Then a minimum spanning tree of I, I_{MST}, is calculated. I_{MST} is partitioned into k groups, such that an optimal embedding for each group can be obtained in reasonable time. Therefore, at the lowest level l, each partition P_i^l, will need a fraction ($1/k$) of the binary variables. Each partition is then embedded and the relative ordering of the edges (the values of the binary variables) is saved. The groups, chosen two at a time, are merged to create the next level $l-1$, $P_i^l \cup P_{i+1}^l \rightarrow P_{(i+1)/2}^{l-1}$. Thus from Figure 5.14, P_1^2 and P_2^2 are merged to produce P_1^1. By allowing the groups to merge, while maintaining their own original order at higher levels, refined solutions are permitted to develop. The complexity, as a function of binary variables at each level, is reduced by noting that the relative ordering of the edges within each group is maintained by fixing the values of the binary variables from the previous groups. This requires each level to carry only half the number of binary variables. For large n, this is still unacceptable. However, if the levels are abutted and not merged, an obvious speedup will be achieved. The complexity of this implementation will be $O(N^2 \cdot (\bar{e} + \bar{x})^2/k)$, and can be linear in N with the appropriate selection of k.
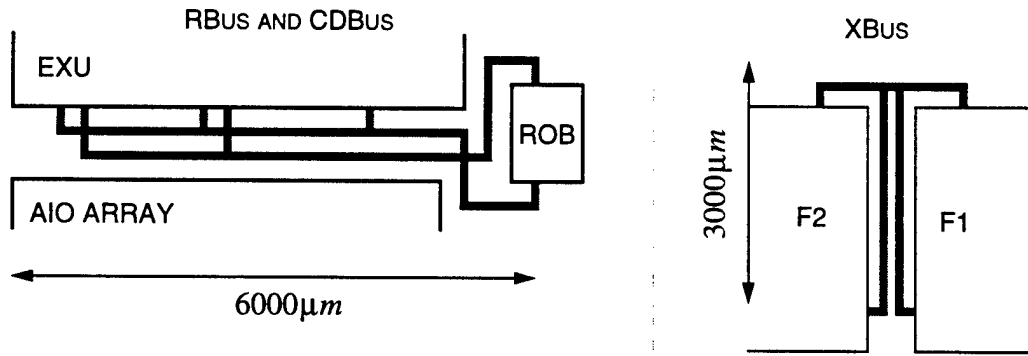
Figure 5.15 Detailed embedding benchmarks.

The topologies of the result (RBus) and common data bus (CDBus) were extracted from the FXU. The figures are not drawn to scale. Each bus structure is 16-bits wide. Timing data (arrival time and slack) for the embedder was generated by the TACTIC static timing analyzer.

5.6.7 Experiments with Sample Route Structures

Figure 5.15 depicts two 16-bit bus structures from the FXU [89] that were

processed by the detailed embedder. The detailed embedder generated valid solutions that

satisfied a range of slack and noise margins. Table 5.2 lists the time (on a Sun Ultra-1)

taken to solve the linear system and the required routing area to satisfy the crosstalk

constraints. The slack values at the sinks and departure times at the source were obtained

through the static timing analyzer (TACTIC) by Cascade Design Automation. The slack

Table 5.2 Detailed Embedding Results.

The increase in area is relative to the starting conditions of 1ns slack for the RBus and CDBus pair, and the starting condition of 400ps slack for the XBus example. The embedding was performed on a Sun Ultra-1. An increase in processing-time and area to satisfy the tighter constraints is observed.

RBus and CDBus			XBus		
Slack	Time (m)	Area (μm^2)	Slack	Time (m)	Area (μm^2)
1ns	7.3	94090	400ps	4.3	112978
300ps	9.2	+3.91%	300ps	5.2	+2.37%
200ps	12.1	+4.35%	200ps	7.4	+3.40%

values were scaled to observe the trade-off made to satisfy crosstalk constraints. For the XBus, a 50% tightening of slack requires a 3.4% increase in routing area. For the RBus and CDBus, an 80% reduction of slack was satisfied by a 4.35% increase in routing area. The large amounts of time required by the MIP solver are primarily attributable to its inefficiency in handling binary constraints. The solver relaxes the binary attributes of the variables in Eq. 5.18, treating the expressions as linear equations and not logical statements.

5.6.8 Summary of the Detailed Embedder

A detailed embedder with linearized crosstalk constraints was developed. Through this model, the $O(N^2)$ complexity of detailed route-embedding with crosstalk constraints is realized. A hierarchical methodology to address this complexity is demonstrated. Experiments with two 16-bit bus structures from the FXU demonstrate the method. The detailed embedder produces a minimum-change minimum-area embedding for a range of performance bounds. The inefficiency of the solver in processing pure binary constraints limits the scope of the formulation.

5.7 Global Embedder Implementation

A global embedder (groute) was implemented in C++. Its user-interface was built using tcl/tk. The interface enables the user to generate routes, perform selective rip-up and re-route procedures and general route analysis.

Figure 5.16 shows the user interface for the global embedder. After loading the placement and global route information, congested gcells are appropriately highlighted using their supply/demand ratio. The timing engine can be invoked to perform a timing analysis of the netlist. It first annotates the route trees with their RC-delays (to each sink), and then
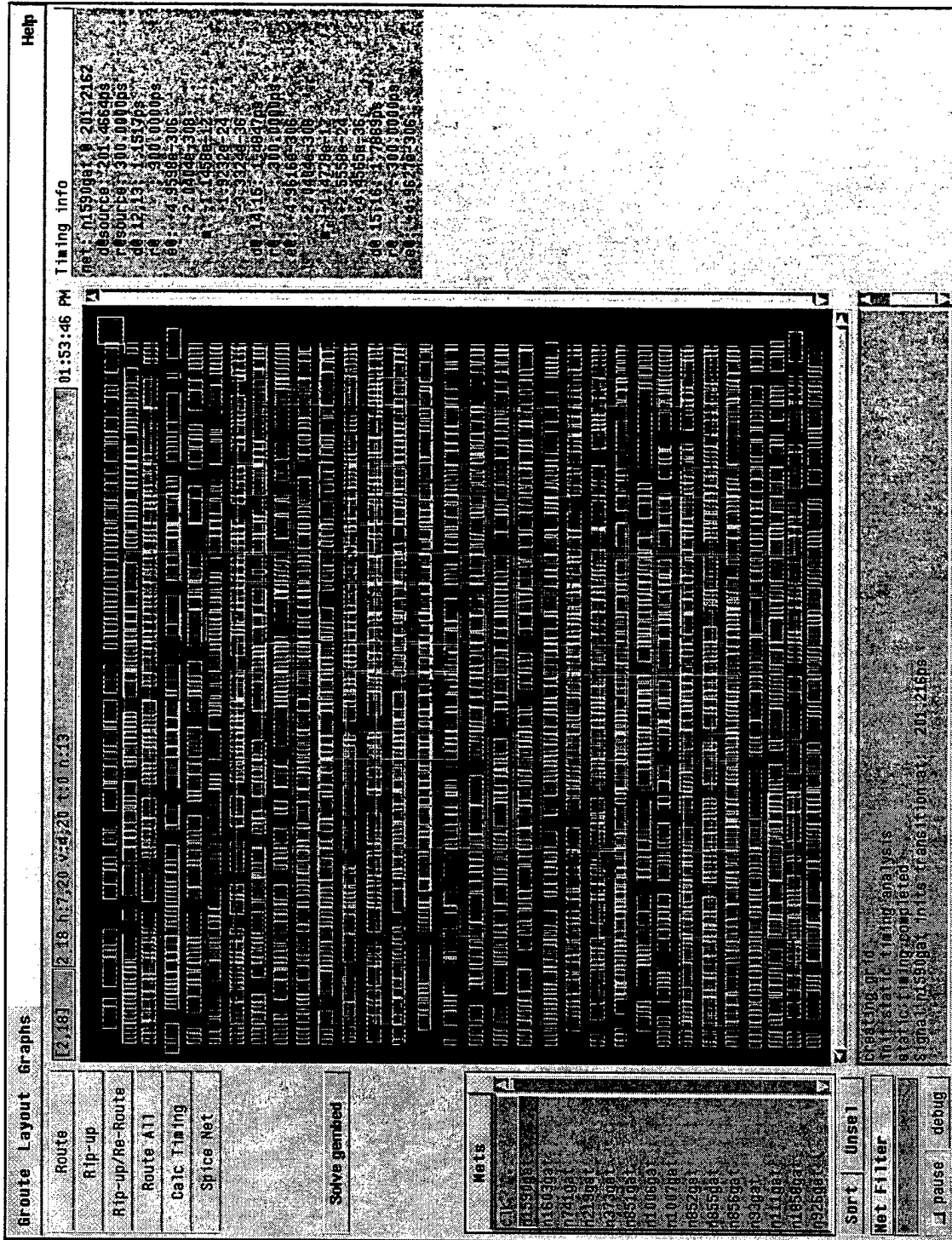
Figure 5.16 `groute`: Graphical User Interface.
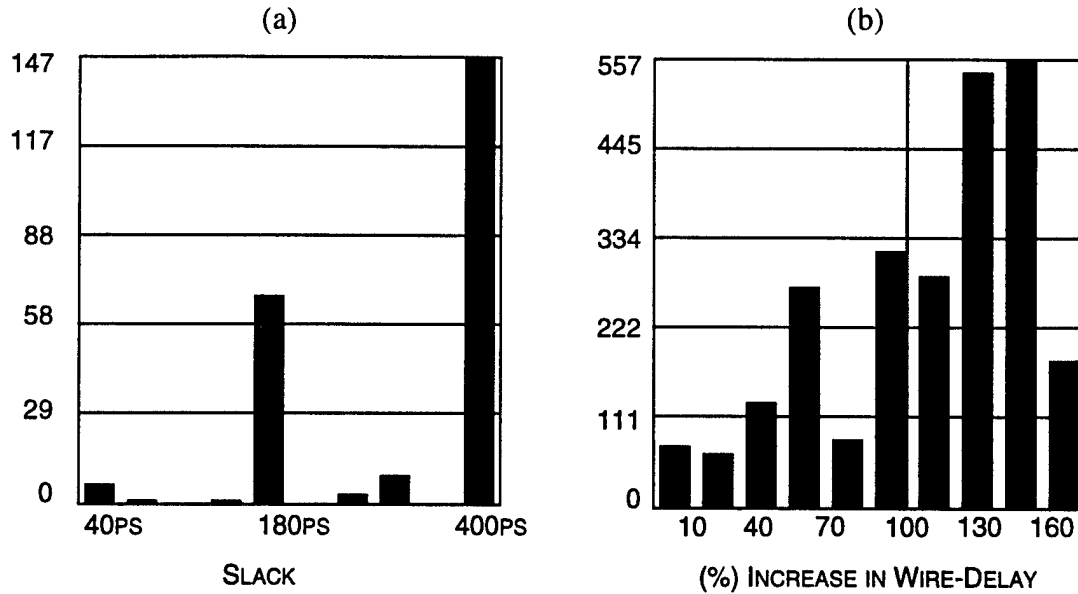
Figure 5.17 groute: Histograms for the FXU.

These sample histograms were generated by groute. The histogram in (a) shows the distribution of slack on the FXU. The cycle time was set to 1ns. The histogram in (b) shows a distribution of wire-delay change, due to crosstalk. More than 50% of the nets will double their delay due to crosstalk.

proceeds to use these delays for slack computation. Timing information related to the selected nets are displayed on the left panel and on the layout. The global embedder also displays various statistics for a given routing solution. For example, Figure 5.17 shows the net slack and crosstalk impact histogram for the top-level routing of the FXU. A majority of the nets have 400ps of slack, however, there are about 80 nets with less than 200ps of slack. The second histogram shows that a large number of these nets will have their delays substantially increased due to crosstalk.

5.8   Global Embedding Results

Table 5.3 lists the pertinent statistical data for the benchmark circuits. The designs were embedded using the SIA-1999 predicted technology [1]. The benchmark designs were selected for their size as larger netlists tend to have longer paths that require crosstalk constraints. s15850 was processed by the placement-engine to produce a lesser

Table 5.3 Global Embedder Benchmark statistics (1999 SIA Technology).

The following designs were embedded to satisfy crosstalk constraints. The designs may intermix standard cells with block objects. The total number of nets, unique paths (from source to sinks) and cells is listed. Design s15850 was obtained from the ISCAS benchmark. The other designs were obtained via the PUMA group [89].

Design	Type	Nets	Paths	Cells	Size (mm^2)	Avg. critical path l (μm)
s15850	Cell	10369	13928	8620	1.21	110.3
s15850_c	Cell	10369	13928	8620	1.20	87.2
MS8	Cell & Block	423	777	250 + 2	0.10	332.6
NNC	Cell & Block	15239	31465	14825 + 2	4.10	424.6
EXU	Cell & Block	4461	9951	3822 + 3	9.83	492.2
FXU	Block	2322	2322	17	47.8	1257.5

congested placement in s15850_c. Design NNC is the core of a image-recognition neural-net chip. The MS8 is a micro-controller. The EXU (Execution Unit) is a large functional block within the Fixed-point Unit (FXU) of the PUMA project [89]. Both contain less paths than the standard-cell groups, however, their average path-length is greater.

The global embedder processed each design from Table 5.3. The results of the first iteration through the solver are listed in Table 5.4. Data for the total 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 violation is presented. The second iteration following rip-up and re-route satisfied all constraints.

The correlation between crosstalk and congestion, demonstrated in Section 4.1, is validated 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 to introduce more spacing (and shields), to satisfy crosstalk constraints, in

Table 5.4 First iteration through Global Route Embedding.

Data for the total 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 violation is listed. The crosstalk constraints for each design were scaled to simulate the impact of tighter design specifications. Tighter constraints were satisfied by increasing the total spacing and number of shields.

| 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
MS8	1.0	92.1	0	0	0
	0.9	100.9	0	0.8	0.1
	0.8	113.5	9	1.1	0.7
	0.7	129.7	16	1.4	1.7
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

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.
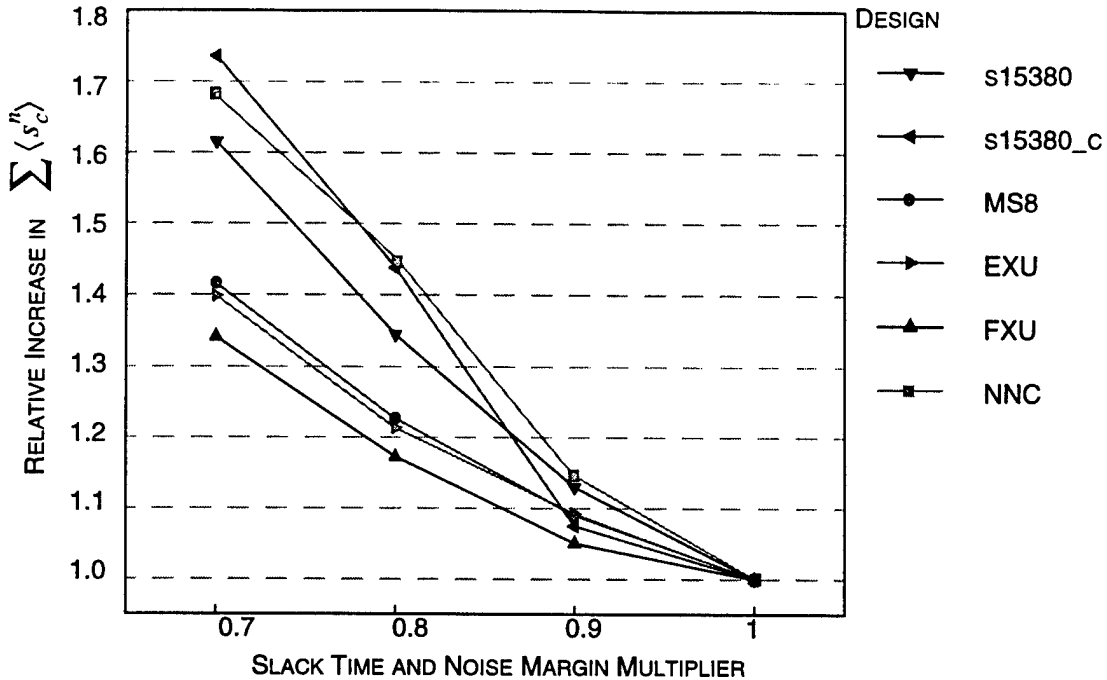
Figure 5.18 Trade-off between area and crosstalk severity.

The global embedder satisfies tighter crosstalk constraints by increasing the total spacing $\sum \langle s_c^n \rangle$.

The constraints on the critical paths are scaled to simulate the impact of tighter design specifications. 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 re-routing for each design. Increasing the severity of the crosstalk constraints causes the global embedder to further allocate spacing and insert shields. Figure 5.18 displays the trade-off in area made by the global embedder as the severity of the crosstalk is increased. 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 a negligible 2% on the first pass of the embedder.

5.9 Summary

Route embedding is presented as a new method for addressing crosstalk. This approach minimally modifies a set of routes to satisfy timing and noise for individual sinks. The traditional method of addressing crosstalk by minimizing coupling capacitance is rejected in favor of an accurate crosstalk model. Coupled with this new crosstalk model, a global route embedder with performance driven crosstalk constraints is implemented. The global embedder satisfies slack and noise margins, at the critical sinks, by computing expected spacings, inserting shields and selective re-route through uncongested regions. This approach, implemented as a Linear Program, precludes the creation of a routing solution with crosstalk violations. Capacity constraints guarantee a detailed routing solution. Experiments on large standard-cell groups and the top level routing of a microprocessor validate 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 FXU required a 34% increase in routing area. A correlation between crosstalk and congestion is validated. With uncongested designs, the global embedder is better able to limit the number of paths requiring re-routing and their average crosstalk-violation.

Through the detailed embedder the theoretical limits of crosstalk constrained route embedding are investigated. Implemented as a mixed linear program, it allows the solver to explicitly capture the behavior of a detailed router with linearized crosstalk constraints. The $O(N^2 \cdot (\bar{x} + \bar{e})^2)$ complexity of this formulation is addressed by a partition and re-build scheme. Experiments with 16-bit bus structures from the FXU demonstrate the capability of the detailed embedder. Minimum-change and minimum-area embeddings for a range of performance bounds are generated. In one case, the detailed embedder resolved a 50% tightening of crosstalk constraints through a 3.4% increase in routing area.

CHAPTER VI

CONCLUSIONS

The dissertation focuses on a design methodology for addressing capacitive crosstalk. Crosstalk is a severe problem in the field of Very Large Scale Integrated-circuit (VLSI) design where aggressive scaling of interconnect pitch has lead to increased capacitance between adjacent traces, causing non-linear interactions evidenced as timing violations and erroneous circuit activity. Upcoming process technologies are driving towards tighter metallization, increased clock frequencies, smaller voltage swings and longer interconnect. Each of these trends will increase the impact of capacitive crosstalk. Estimates on coupling show this impact will double during the next decade.

This dissertation presents a physical design methodology that accounts for crosstalk with accurate and consistent estimates of wiring constraints throughout the design flow. By maintaining a consistent view, the methodology makes violations due to crosstalk predictable, and therefore, avoidable.

This final chapter summarizes the contributions of the research and submits suggestions for future work.

6.1   Contributions

This work makes contributions in the field of crosstalk-constrained physical design. It is the first to address crosstalk throughout the physical design process. It has

made advances in crosstalk modeling, design methodologies, routing congestion, quadratic placement, and route embedding.

6.1.1 Crosstalk Modeling

The dependence and sensitivity of crosstalk impact on waveform parameters such as arrival time and edge-rates was determined. A simple and extensible model capable of predicting both crosstalk noise and delay changes for coupled RC lines, is presented. The model is shown to be accurate to within 8% for wire-delay, and 10% for noise. Use of this model is more effective at satisfying performance constraints that the ubiquitous metric of minimizing adjacent capacitance. A linearized form of this crosstalk model, with accurate bounds on the linearizing-error, is derived. This allows a linear solver to trade off physical and temporal proximity when satisfying performance-driven crosstalk constraints.

6.1.2 Crosstalk Constrained Design Methodology

The concept of a wiring engine that maintains and enforces a consistent view of routing is introduced. It enforces wiring constraints throughout the design flow, making violations due to crosstalk predictable, and thus avoidable. By addressing wiring congestion, the methodology is able to reduce potential crosstalk sites and relieve the burden on the later routing steps. The methodology innovates by enforcing these global routing structures throughout the flow. These route structures are annotated with timing information and processed by the route embedder. A solution which mitigates crosstalk on critical nets is generated by computing spacing and inserting ground shields. These resource allocations satisfy the expected wire-delay and noise constraints at each critical sink.

6.1.3 Correlation between Congestion and Crosstalk

The correlation between congestion and crosstalk was shown on designs ranging from large standard-cell groups to the top-level routing of a VLSI microprocessor. In each case, a high degree of correlation is observed. On average, 50% of the critical paths in these benchmark designs have more than 50% of their length passing through regions of high congestion. Minimizing congestion reduces potential sites for crosstalk and provides flexibility (area) for a detailed router to use in addressing crosstalk. The correlation between crosstalk and congestion motivates the need for congestion-driven placement when addressing crosstalk.

6.1.4 Congestion Driven Quadratic Placement

A methodology for integrating congestion into quadratic placement was presented. The inclusion of a growth matrix into the quadratic placement formulation permitted the use of a routing model to reduce congestion while concurrently minimizing wirelength. A further reduction in congestion, reflected in wirelength and area reduction, was obtained by relaxing pin constraints. Together, these approaches showed up to 20% reductions in congestion.

6.1.5 Global Route Embedding

A new method of mitigating crosstalk-impact, route embedding, was presented. Route embedding modifies a set of route structures to satisfy timing and noise constraints, while minimizing area and maintaining the form of the original route topology. Equations for the expected timing and noise impact at critical sinks were derived. The global embedder satisfies slack and noise constraints by computing expected spacings and inserting grounded shields. Routing capacity constraints are enforced to guarantee a routing solution. The trade offs between area (spacing) and crosstalk impact were

presented for designs ranging from large standard-cell groups to the top-level routing of a VLSI microprocessor. The formulation was tested by tightening the crosstalk constraints down to 70% of their original values. In all cases, the methodology generated an embedding that satisfied crosstalk constraints. Tightening the crosstalk constraints for the microprocessor by 30%, resulted in a 34% larger embedding and required a re-routing of 6% of the critical paths. These paths did not satisfy their crosstalk constraints by a negligible 2% on the first pass of the embedder. All crosstalk constraints were met after a re-routing of the unsatisfied paths through lesser congestion.

6.1.6 MIP Detailed Route Embedding

A Mixed-Integer Linear Program (MIP) that implements a model of route embedding with crosstalk constraints was developed. The formulation minimally modifies route structures while satisfying crosstalk constraints. The binary variables in the MIP define an upper-bound $O(N^2)$ complexity of routing with crosstalk constraints. Two multi-sink bus structures, extracted from the top-level routing of a microprocessor, were embedded. The embedder satisfied a 50% tightening of crosstalk-constraints through a 4% increase in routing area.

6.2 Future Work

This work described in this dissertation not only contributes original ideas to the area of crosstalk-constrained physical design, but also exposes potential avenues for further research in congestion-driven placement and detailed routing.

6.2.1 Crosstalk-constraints within Congestion-driven Placement

The Congestion-driven Quadratic Placement algorithm makes use of global routes provided by a wiring engine to estimate the global demand of wiring resources. The

demand imposed by the global routes could be made to include the expected spacings needed to satisfy crosstalk constraints. The global route algorithm could be extended to account for crosstalk by accounting for signal activity in each region. This would involve merging the global route embedder into the wiring engine for the congestion-driven quadratic placer. This would eliminate the need for rip-up and re-route in the global embedder.

6.2.2 A Gridless Crosstalk-Constrained Detailed Router

The global embedder would benefit from an gridless area-router. The grided area router, presented by Tseng et al in [108], could be converted to a gridless one for the purposes of constructing the solution determined by the global embedder. This would enhance the current methodology by generating the final routing solution. It would provide insight into the accuracy of the global embedder's estimates of expected crosstalk-impact.

6.3 Summary

This dissertation is the first to address crosstalk throughout the physical design process. It proposes a new crosstalk model that enables performance driven crosstalk constraints. The correlation between crosstalk and routing congestion is shown and a Congestion-driven Quadratic Placement methodology is presented. Finally, a new method of mitigating crosstalk, Route Embedding, is presented. It satisfies crosstalk constraints on critical paths by computing expected spacings and inserting grounded shields.

This dissertation has presented a physical design methodology to address crosstalk. Accurate estimates of wiring and crosstalk-constraints are maintained throughout the design flow. This consistent view makes violations due to crosstalk predictable and therefore, avoidable.

BIBLIOGRAPHY

BIBLIOGRAPHY

Table 6.1 Listed Conferences and Proceedings.

Acronym	Conference
DAC	Design Automation Conference
ASPDAC	Asia and South Pacific Design Automation Conference
ICCD	International Conference on Computer Design
ICCAD	International Conference on Computer Aided Design
ISCAS	International Symposium on Circuits and Systems
CICC	Custom Integrated Circuits Conference
ASPCAS	Asia and South Pacific Conference on Circuits and Systems
ISPD	International Symposium on Physical Design
TCAD	Transaction on Computer Aided Design
TCAD-ICS	Transaction on Computer Aided Design of Integrated Circuits and Systems

[1]   1997 Semiconductor Industry Association Roadmap, http://www.sematech.org.

[2]   Alexander. M. J, Robins. G, "New performance-driven FPGA routing algorithms". *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol.15, no.12, pp. 1505-1517.

[3]   Alpert. C. J, Cong. J, Kahng. A. B, Robins. G, Sarrafzadeh. M, "Minimum density interconnection trees". *1993 IEEE ISCAS*, vol.3, pp. 1865-1868.

[4]   Alpert. C. J, Hu. T. C, Huang. J. H, Kahng. A. B, Karger. D, "Prim-Dijkstra trade-offs for improved performance-driven routing tree design". *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol.14, no.7, pp. 890-896.

[5]   Anastasakis. D. F, Gopal. N, Kim. S. Y, Pillage. L. T, "On the Stability of Moment-Matching Approximations in Asymptotic Waveform Estimation", *DAC-29*. pp. 207-212.

[6]   Areibi. S, Vannelli. A, "An efficient clustering technique for circuit partitioning". *1996 IEEE ISCAS*, vol.4, pp. 671-674.

[7]   Backoglu. H. B, "Circuits, Interconnections, and Packaging for VLSI," Addison-Wesley Publishing Company, 1990.

[8]   Basso. T, "A Microarchitecture for High-Speed, Resource-Limited, Superscalar Microprocessors". Ph. D. Thesis. University of Michigan, 1998.

[9]   Bazaraa. S. M, Jarvis. J. J, Sherali. H. D, "Linear Programming and Network Flows," John Wiley and Sons, Chichester, 1990.

[10]   Bevington. P. R, Robinson. D. K, "Data Reduction and Error Analysis for the Physical Sciences," McGraw-Hill, Inc., 1992.

[11]   Boese. K. D, Kahng. A. B, McCoy. B. A, Robins. G, "Near-optimal critical sink routing tree constructions". *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol.14, no.12, pp. 1417-1436.

[12]   Boese. K. D, Kahng. A. B, McCoy. B. A, Robins. G, "Rectilinear Steiner trees with minimum Elmore delay". *31st DAC*, pp. 381-386.

[13]   Borah. M, Owens. R. M, Irwin. M. J, "Recent developments in performance driven Steiner routing: an overview". *Proceedings*, pp. 137-142.

[14]   Brasen. D. R, Bushnell. M. L, "MHERTZ: a new optimization algorithm for floorplanning and global routing". *27th ACM/IEEE DAC*, pp. 107-110.

[15]   CPLEX, http://www.cplex.com/.

[16]   Chaudhary. K, Onozawa. A, Kuh. E. S, "Performance driven spacing algorithms using attractive and repulsive constraints for submicron LSIs".*IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol.14, no.6, pp. 707-719.

[17]   Cheng. C. L. E, "RISA: accurate and efficient placement routability modeling". *1994 IEEE/ACM ICCAD*, pp. 690-695.

[18]   Cho. J. D, Raje. S, Sarrafzadeh. M, Sriram. M, Kang. S. M, "A multilayer assignment algorithm for interference minimization". *Workshop Proceedings 4th ACM/ SIGDA Physical Design Workshop Layout Synthesis for the New Generation of VLSI ASIC Technologies*, pp. 63-67.

[19]   Cho. J. D, Raje. S, Sarrafzadeh. M, Sriram. M, Kang. S. M, "Crosstalk-minimum layer assignment". *Proceedings of IEEE CICC 1993*.

[20]   Chu C-Y, Horowitz M. A, "Charge-Sharing Models for Switch-Level Simulation," *IEEE TCAD*, vol 6, no 6. Nov 1996.

[21]   Collaborative Benchmarking Lab, http://www.cbl.ncsu/.

[22]   Cong. J, Kahng. A. B, Robins. G, Sarrafzadeh. M, Wong. C. K, "Provably good performance-driven global routing". *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol.11, no.6, pp. 739-752.

[23]   Cong. J, Madden. P. H, "Performance driven routing with multiple sources". *1995 IEEE ISCAS*, vol.1, pp. 203-206.

[24]   Cong. J, Preas. B, "A new algorithm for standard cell global routing". *Integration 1992*, vol.14, no.1, pp. 49-65.

[25]   Cong. J, Xu. D, "Exploiting signal flow and logic dependency in standard cell placement". *Proceedings of ASP-DAC'95*.

[26]   D. K. Boese, B. A. Kahng, B. Robins, "High-performance routing trees with identified critical sinks". 30th DAC, 1993, pp 182-187.

[27] D. W. Hightower, "A solution to Line Routing Problems on the Continuous Plane", 6th Design Automation Workshop 1969, pp. 172-174.

[28] D. Wang, Kuh. E. S, "A new timing-driven multilayer MCM/IC routing algorithm". *1997 IEEE Multi-Chip Module Conference*, pp. 89-94.

[29] D. Wang, Kuh. E. S, "Performance-driven interconnect global routing". *The Sixth Great Lakes Symposium on VLSI*, 1996. pp. 132-136.

[30] Dai. W. W. M, Kong. R, Jue. J, Sato. M, "Rubber band routing and dynamic data representation". *1990 IEEE ICCAD*, pp. 52-55.

[31] Dasgupta. P. S, Sen. A. K, Nandy. S. C, Bhattacharya. B. B, "Geometric bipartitioning problem and its applications to VLSI". *Proceedings of the Ninth International Conference on VLSI Design*, pp. 400-405.

[32] David. P. Lapotin, "Early Assessment of Design, Packaging and Technology Trade-offs". *International Journal of High Speed Electronics 1991*, vol. 2, No. 4, pp. 209-233.

[33] Devgan A, "Efficient Coupled Noise Estimation for On-Chip Interconnects," *IEEE-ICCAD*, pp. 147-153. 1997.

[34] Devgan. A, Alpert. C. J, Quay. S. T, "Buffer Insertion for Noise and Delay Optimization," *DAC-35*. pp. 362-367. 1998.

[35] Diallo. O, Lucke. L, Cameron. G, Hassoun. M, Jerdee. A, Melvin. C, "Multi-objective based placement for custom macro-cells". *Proceedings of the 39th Midwest Symposium on Circuits and Systems*, vol.1, pp. 447-450.

[36] Doll. K, Johannes. F. M, Antreich. K. J, "Iterative placement improvement by network flow methods". *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol.13, no.10, pp. 1189-1200.

[37] Donath. W. E, Norman. R. J, Agrawal. B. K, Bello. S. E, Han. S. Y, Kurtzberg- JM, Lowy. P, McMillan. R. I, "Timing driven placement using complete path delays". *27th ACM/IEEE DAC*, pp. 84-89.

[38] Dunlop. A. E, Kernighan. B. W, "A procedure for placement of standard-cell VLSI circuits". *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, no.1, pp. 92-98.

[39] E. S. David, L. Zbigniew, "Meshach: Matrix Computations in C". *Proceedings of the Center for Mathematics and its Applications. Australian National University*, vol. 32, 1994.

[40] F. Dartu, L. T. Pileggi, "Calculating Worst-Case Gate Delays Due to Dominant Capacitance Coupling", *DAC-1997*, pp. 46-51.

[41] F. Dartu, N. Menezes, L. T. Pillage, "Performance computation for precharacterized CMOS gate with RC-loads," *IEEE TCAD-ICS*, vol. 15, pp. 544-553, May 1996.

[42] F. J. Liu, J. Lillis, C. K. Cheng, "A New Layout-Driven Timing Model for Incremental Layout Optimization", *TAU-1997*, pp. 127-131.

[43] F. J. Liu, Lillis. L, C. K. Cheng, "A new layout-driven timing model for incremental layout optimization". *Proceedings of the ASP-DAC'97*, pp. 127-131.

[44] Feldmann. P, Freund. R. W, "Efficient linear circuit analysis by Pade approximation via the Lanczos process," *IEEE TCAD-ICS*, vol.14, no.5, pp 639-649, May 1995.

[45] G. H. Golub, F. V. C. Loan, "Matrix computations", The Johns Hopkins University Press, 2nd ed.

[46] Ganley. J. L, Cohoon. J. P, "Rectilinear Steiner trees on a checkerboard". *ACM Transactions on Design Automation of Electronic Systems*, vol.1, no.4, pp. 512-522.

[47] Gao. T, Liu. C. L, "A spacing algorithm for performance enhancement and cross-talk reduction". *1993 IEEE/ACM ICCAD*, pp. 697-702.

[48] Garbers. J, Korte. B, Promel. H. J, Schwietzke. E, Steger. A, "VLSI-placement based on routing and timing information". *EDAC*, pp. 317-321.

[49] Guruswamy. M, Wong. D. F, "Echelon: a multilayer detailed area router". *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol.15, no.9, pp. 1126-1136.

[50] H. P. Tseng, Sechen. C, "A gridless multi-layer router for standard cell circuits using CTM cells". *Proceedings of ED & TC 97*, pp. 319-326

[51] Hagen. L, Kahng. A. B, Tsai. Y. T, dLuna. L. J, Lee. P. PK, "Improving the quadratic objective function in module placement". *Proceedings of Fifth Annual IEEE International ASIC Conference and Exhibit*, pp. 42-45.

[52] Hamada. T, Cheng. C. K, Chau. P. M, "Prime: A timing-driven placement tool using a piecewise linear resistive network approach". *30th DAC*, pp. 531-536.

[53] Hom. I, Granacki. J, "Estimation of the number of routing layers and total wire-length in a PCB through wiring distribution analysis"., pp. 310-315.

[54] Hong. X, Xue. T, Kuh. E. S, Cheng. C. K, Huang. J, "Performance-driven Steiner tree algorithms for global routing". *30th DAC*, pp. 177-181.

[55] Hossain. M, Thumma. B, Ashtaputre. S, "A new faster algorithm for iterative placement improvement". *Proceedings of The Sixth Great Lakes Symposium on VLSI*, pp. 44-49.

[56] Huang. J, Hong. X. L, Cheng. C. K, Kuh. E. S, "An efficient timing-driven global routing algorithm". *30th DAC*, pp. 596-600.

[57] *Intel Pushes the Design Envelope*, EETIMES-1997, issue 936.

[58] *Interconnect Analysis Must Move to 3-D*, EETIMES-1997, issue 980.

[59] J. Quain, S. Pullela, L. T. Pillage, "Modeling the 'effective capacitance' of the RC-interconnect." *IEEE TCAD-ICS*, vol. 13, pp. 1526-155, December 1994.

[60] J. D. Cho, Sarrafzadeh. M, "A buffer distribution algorithm for high-performance clock net optimization". *IEEE Transactions on Very Large Scale Integration-*, vol.3, no.1, pp. 84-98.

[61] J. Song, H. K. Choo, W. Zhuang, "A new model for general connectivity and its application to placement". *The Sixth Great Lakes Symposium on VLSI*, pp. 60-63.

[62] Jackson. M. A. B, Kuh. E. S, Sadowska. M, "Timing-driven routing for building block layout". *1987 IEEE ISCAS*, vol.2, pp. 518-519.

[63] Jhang. K. S, Ha. S, Jhon. C. S, "COP: a Crosstalk OPtimizer for grided channel routing". *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol.15, no.4, pp. 424-429.

[64] K. S. Jhang, S. Ha, C. S. Jhon, "A segment rearrangement approach to channel routing under the crosstalk constraints". *ASPCAS-94*, pp. 536-541.

[65] Kang. S. M, "Performance-driven layout of CMOS VLSI circuits". *1990 IEEE ISCAS*, vol.2, pp. 881-884.

[66] Kannan. L. N, Suaris. P. R, Hong. Gee. Fang, "A methodology and algorithms for post-placement delay optimization". *31st DAC*, pp. 327-332.

[67] Klein. P, Plotkin. S, Stein. C, Tardos. B, "Faster approximation algorithms for the unit capacity concurrent flow problem with applications to routing and finding sparse cuts". *SIAM Journal on Computing*, vol.23, no.3, pp. 466-487.

[68] Kleinhans. J. M, Sigl. G, Johannes. F. M, Antreich. K. J, "GORDIAN: VLSI placement by quadratic programming and slicing optimization". *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol.10, no.3, pp. 356-365.

[69] Koide. T, Ono. M, Wakabayashi. S, Nishimaru. Y, Yoshida. N, "A new performance driven placement method with the Elmore delay model for row based VLSIs". *Proceedings of the ASP-DAC 95*, pp. 405-412.

[70] Kuwahara. N, Asami. S, Takano. N, Nomura. M, "A routing system for high-performance computer systems". *ICCAD 86*, pp. 250-253.

[71] L. Scheffer, "A roadmap of CAD tool changes for sub-micron interconnect problems". ISPD-97, pp. 104-109.

[72] Lengaur. T, "Combinatorial Algorithms for Integrated Circuit Layout," John Wiley and Sons, Chichester, 1990.

[73] Lienig. J, "A parallel genetic algorithm for two detailed routing problems". *1996 IEEE ISCAS*, vol.4.

[74] Lillis. J, C. K. Cheng, T. T. Y. Lin, "Simultaneous routing and buffer insertion for high performance interconnect". *Proceedings*, pp. 148-153.

[75] Lin. I, Du. D. HC, "Performance-driven constructive placement". *27th ACM/IEEE DAC*, pp. 103-106.

[76] Litovski. V, Randjelovic. Z, Damnjanovic. M, "Routing in standard cells". *1995 20th International Conference on Microelectronics*, vol.2, pp. 461-466.

[77] Liu. L. C, Tseng H. P, Sechen. C, "Chip-Level Area Routing". *1998 International Symposium on Physical Design*, pp. 197-204.

[78] Mackey. C. A, Carothers. J. D, "Performance-driven macrocell placement". *Conference Proceedings of the 1996 IEEE Fifteenth Annual International Phoenix Conference on Computers and Communications*, pp. 427-433.

[79] Matsubayashi. A, Ueno. S, "On the complexity of embedding of graphs into grids with minimum congestion". *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, no.4, pp. 469-476.

[80] Mayrhofer. S, Lauther. U, "Congestion-driven placement using a new multi-partitioning heuristic". *1990 IEEE ICCAD*, pp. 332-335.

[81] McMurchie. L, Ebeling. C, "PathFinder: a negotiation-based performance-driven router for FPGAs"., pp. 111-117.

[82] Moll. F, Isern. E, Sicard. E, Rubio. A, Michael. S, "Analysis of cross-talk effects on logic cell delays in CMOS integrated circuits". *Proceedings of the 34th Midwest Symposium on Circuits and Systems,* vol.1, pp. 387-390.

[83] Moore, G. "Are We Really Ready for VLSI," *Caltech Conference on VLSI*, 1979.

[84] Nabors. K, White. J, "FastCap: a multipole accelerated 3-D capacitance extraction program," *IEEE TCAD-ICS*, vol. 10, no. 11. pp 1447-1459, Nov 1991.

[85] Natesan. V, Bhatia. D, "Clock-skew constrained cell placement". *Ninth International Conference on VLSI Design,* pp. 146-149.

[86] *Observations on CAD research*, NSF Workshop-CAD 1996.

[87] Okada. K, Onodea. H, Tamaru. K, "A global routing algorithm for analog circuits using a resistor array model". *1996 IEEE ISCAS*, vol.4.

[88] Onozawa. A, Chaudhary. K, Kuh. E. S, "Minimum crosstalk channel routing". *1993 IEEE/ACM ICCAD*, pp. 692-696.

[89] PUMA Project, http://www.eecs.umich.edu/UMichMP.

[90] Parakh. P, Brown. R, Sakallah. K, "Congestion Driven Quadratic Placement," *DAC-35*. 1998.

[91] Parker. B. H, Ray. A. K, Ghassemlooy. Z, "Crosstalk in the interconnection bus for a high-speed digital logic circuit". *International Journal of Electronics*, vol.76, no.2, pp. 265-269.

[92] Penfield. P, Rubenstein. J, "Signal Delay in RC-tree networks," *DAC-19*, 1981.

[93] Personal conversations with D. Boyle and R. Blair.

143

[94]   Peters. L, "Pursuing the Perfect Low-k Dielectric," *Semiconductor International*, Sept. 1998, pp 64-74.

[95]   Peters. I, Molitor. P, Weber. M, "Vertical floating pins in OTC routing [VLSI layout]". *APCCAS 1996*, pp. 385-388.

[96]   Pileggi. L, "Coping with RC(L) interconnect design headaches". *1995 IEEE/ACM ICCAD*, pp. 246-253.

[97]   Pillage. L. T, Rohrer. R. A, "Asymptotic waveform evaluation for timing analysis". *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol.9, no.4, pp. 352-366.

[98]   Prasitjutrakul. S, Kubitz. W. J, "A timing-driven global router for custom chip design". *1990 IEEE ICCAD*, pp. 48-51.

[99]   R. Dutta, A. Roy, R. Rao, "Multi Layer Area Routing as an Optimization Problem". *1990 IEEE CICC*, pp 27.4.1-27.4.4.

[100] R. Fourer, D. M. Gay, B. W. Kernighan, "A Modeling Language for Mathematical Programming", *Management Science 1990*, vol. 36, no. 5, pp. 519-554.

[101] R. S. Tsay, E. S. Kuh, C. P. Hsu, "PROUD: A Fast sea-of-gates placement algorithm". 25th DAC 1988, pp 318-323.

[102] Tellez. G. E, Knol. D. A, Sarrafzadeh. M, "A performance-driven placement technique based on a new budgeting criterion". *IEEE ISCAS*, vol.4, pp. 504-507.

[103] Terai. M, Shirota. H, Shibatani. S, Sato. K, "A fast routing method for channel-less sea-of-gates arrays with three routing layers". *Transactions of the Information Processing Society of Japan*, vol.38, no.3, pp. 657-668

[104] Thakur. S, K. Y. Chao, Wong. D. F, "An optimal layer assignment algorithm for minimizing crosstalk for three layer VHV channel routing". *1995 IEEE Symposium on Circuits and Systems*, vol.1, pp. 207-210.

[105] Theune. D, Thiele. R, Lengauer. T, Feldmann. A, "HERO: Hierarchical EMC-constrained routing". *1992 IEEE/ACM ICCAD*, pp. 468-472.

[106] Tsay. R. S, Chang. S. C, Thorvaldson. J, Tsai. Y. T, Luna. L. J, Lee. P. PK, "Early wireability checking and 2D congestion-driven circuit placement". *Proceedings of Fifth Annual IEEE International ASIC Conference and Exhibit*, pp. 50-53

[107] Tsay. R. S, Koehl. J, "An analytic net weighting approach for performance optimization in circuit placement". *28th ACM/IEEE DAC*, pp. 620-625.

[108] Tseng H. P, Scheffer. L, Sechen. C, "Timing and Crosstalk Driven Area Routing," *DAC-98*, pp 378-381.

[109] Tutuianu. B, Dartu. F, Pileggi. L, "An Explicit RC-Circuit Delay Approximation Based on the First Three Moments of the Impulse Response," *DAC-1996*, pp 611-616.

[110] V. Chandramouli, A. I. Kayassi, K. A. Sakallah, "Signal Delay in Coupled Distributed RC Lines in the Presence of Temporal Proximity". 17th ARVLSI-1997, pp. 32-46.

[111] Vittal. A, M. Sadowska, "Crosstalk reduction for VLSI". *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol.16, no.3, pp. 290-298.

[112] Vittal. A, Sadowska. M, "Minimal delay interconnect design using alphabetic trees". *31st DAC*, pp. 392-396.

[113] W. C. Elmore, "Transient Response of Damped Linear Networks with Particular Regard to Wideband Amplifiers", *Journal of Applied Physics 1948*, vol. 19(1).

[114] Watanabe. T, Oda. T, Onaga. K, "A congestion-cost-directed router for VLSI switchboxes". *1990 IEEE ISCAS*, vol.3, pp. 1684-1687.

[115] X. Zhang, "Coupling effects on Wire Delay". *IEEE Circuits and Devices 1996*, pp. 12-18.

[116] Xue. T, Kuh. E. S, Wang. D, "Post global routing crosstalk risk estimation and reduction". *1996 IEEE/ACM ICCAD*, pp. 302-309.

[117] Y. N. Kim, H. D. Lee, S. Y. Hwang, "An interconnect allocation algorithm for performance-driven datapath synthesis". *Journal of Circuits, Systems and Computers*, vol.6, no.4, pp. 403-423.

[118] Y. W. Chang, D. F. Wong, C. K. Wong, "FPGA global routing based on a new congestion metric". *Proceedings of ICCD*, pp. 372-378.

[119] Yuh. Sheng. Lee, Wu. A. C. H, "A performance and routability-driven router for FPGAs considering path delays". *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol.16, no.2, pp. 179-185.

[120] Zamani. M. S, Hellestrand. G. R, "A stepwise refinement algorithm for integrated floorplanning, placement and routing of hierarchical designs". *1995 IEEE Symposium on Circuits and Systems*, vol.1, pp. 49-52.

[121] Zhou. H, Wong D. F, "Global Routing with Crosstalk Constraints," *DAC-98*. pp 374-377.

[122] Zhou. H, Wong. D. F, "Crosstalk-Constrained Maze Routing Based on Lagrangian Relaxation," *Proceedings ICCD*. pp. 628-633, 1997.

[123] Ratzlaff L. C, Gopal N, Pillage T. L, "RICE: Rapid Interconnect Circuit Evaluator," *DAC-28*, pp 555-560, 1991.

[124] Ratzlaff L. C, Pillage T. L, "RICE: Rapid Interconnect Circuit Evaluation Using AWE," *IEEE TCAD-ISCS*, vol 13, no. 6, pp 763-776. June 1994.

[125] Rohrer. R, "The absolute need for physical verification and analysis of submicron circuits". *WESCON/95 Conference Record*, pp. 112-114.

[126] Roy. K, B. Guan, Sechen. C, "A sea-of-gates style FPGA placement algorithm". *VLSI Design*, vol.4, no.4, pp. 293-307.

[127] Sedra. S. A, Smith. C. K, "Microelectronic Circuits,", Saunders College Publishing, 1991

[128] S. Yousef, "Iterative methods for sparse linear systems". PWS Publishing Company.

[129] SABER. http://www.analogy.com/.

[130] Saab. Y, "A fast clustering-based min-cut placement algorithm with simulated-annealing performance". *VLSI Design*, vol.5, no.1, pp. 37-48.

[131] Sarrafzadeh. M, K. F. Liao, C. K. Wong, "Single-layer global routing". *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol.13, no.1, pp. 38-47.

[132] Scales. L. E, "Introduction to Non-Linear Optimization," Springer-Verlag New York Inc., 1985.

[133] Sengupta. M, Lipa. S, Franzon. P, Steer. M, "Crosstalk driven routing advice". *1994 Electronic Components and Technology Conference.*

[134] Shirota. H, Shibatani. S, Terai. M, "A new rip-up and reroute algorithm for very large scale gate arrays". *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences 1997*, no.3, pp. 506-513.

[135] Sigl. G, Doll. K, Johannes. F. M, "Analytical placement: a linear or a quadratic objective function?". *28th ACM/IEEE DAC*, pp. 427-432.

[136] Suaris. P. R, Kedem. G, "A quadrisection-based combined place and route scheme for standard cells". *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol.8, no.3, pp. 234-244.

[137] Sutanthavibul. S, Shragowitz. E, Lin. R. B, "An adaptive timing-driven placement for high performance VLSIs". *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol.12, no.10, pp. 1488-1498.

[138] Sutanthavibul. S, Shragowitz. E, "An adaptive timing-driven layout for high speed VLSI". *27th ACM/IEEE DAC*, pp. 90-95.

[139] Suzuki. T, Koide. T, Wakabayashi. S, Yoshida. N, "A timing-driven global routing algorithm considering channel density minimization for standard cell layout". *1996 IEEE ISCAS*, vol.4, pp. 424-427.

[140] Systems Optimization Laboratory, http://www.-leland.stanford.edu/group/SOL/solsoft.html.

[141] T. Stohr, M. Alt, A. Hetzel, J. Koehl, "Analysis, Reduction and Avoidance of Crosstalk on VLSI Chips," *ISPD-98*, pp 211-218.

146