*Technical Report NCSU-ERL-95-26 (will also appear in DAC 95)*

# Performance Driven Global Routing and Wiring Rule Generation for High Speed PCBs and MCMs

Sharad Mehrotra
IBM Corporation
11400 Burnet Road,Austin, TX 78758

Paul Franzon[1], Michael Steer
Department of Electrical & Computer Engineering
North Carolina State University, Raleigh, NC 27695-7911

## Abstract

**A new approach for performance-driven routing in highly congested high speed MCMs and PCBs is presented. Global routing is employed to manage delay, signal integrity and congestion simultaneously. Interconnect performance prediction models are generated through simulations. The global routing results and performance prediction models are used to generate bounds on the net lengths which can be used by a detailed router to satisfy constraints on interconnect performance.**

## 1 Introduction

In this paper, we address the problem of interconnect routing on a Printed Circuit Board (PCB) or MultiChip Module (MCM) under tight timing and noise ('signal integrity') constraints. The delay and noise characteristics of the interconnect are very sensitive to the interconnect topology and length of interconnect segments. Hence, in order to meet the timing and noise constraints, accurate constraints on topology and wire length ('wiring rules') must be produced for each net. Good analytical predictors, relating interconnect topology and net length to delay and noise parameters, are, in general not available (Some progress has been made for point to point net modeling in [4]). Hence the interconnect must be evaluated using circuit simulation to accurately predict the signal delay and noise. This implies that the task of generating constraints on net topology and wire length is quite difficult.

Most current commercial tools require users to specify the physical constraints up-front for each branch of each net and then use a post-routing simulator to adjust the constraints as necessary. This iterative approach is too time consuming and difficult in designs where there are many electrical constraints. Davidson & Katopis[1] suggest a technique in which polynomial equations are fitted to simulation results to predict interconnect delay. In addition, the designer specifies the net topology and the wirelength limits in order to manage signal integrity constraints. They point out the importance of using constrained net topologies (Figure 1) to control reflection noise. It is important to note that a Steiner tree is not one of these allowed topologies. A similar approach is also used in [**?**]. Lee et.al. [7] describe a technique in which wiring rules are generated by conducting a small number of simulations around a nominal design point, during design. An equation is fitted to these simulations and used to generate bounds for acceptable electrical behavior. Such 'on-line' simulation techniques require the use of the simplest driver models and lossless lines to achieve manageable simulation times. With more complex models, on-line simulation becomes prohibitive.

In this paper, we present a new technique to accurately transform timing and noise constraints into physical constraints ('wiring rules'). A simulation based circuit characterization technique is used to obtain easy to evaluate models of interconnect performance. Such models are generated *a priori* for each unique net topology in the design. A global routing strategy is employed to identify good routing trees for each net such that wiring congestion is manageable and the electrical constraints are likely to be satisfied. Wirelength bounds are then generated for these routing trees to meet the electrical constraints. These bounds may be used by a detailed router even if the detailed router can not follow the paths suggested by the global route. We begin with a discussion on the global routing technique.

## 2 Global Routing

The global routing problem is that of finding approximate paths for each net in the design, such that the paths are non-intersecting and satisfy the electrical constraints. The routing is performed on a *channel intersection graph*. The edges in the graph correspond to wiring channels, and the nodes correspond to intersection of channels and electrical pins. Since in PCBs and MCMs, the chip sizes and pin locations are fixed, such a

[1] supported by NSF under MIP-901704 and a Young Investigator award and by ARPA under contract DAAH04-94-G-003.

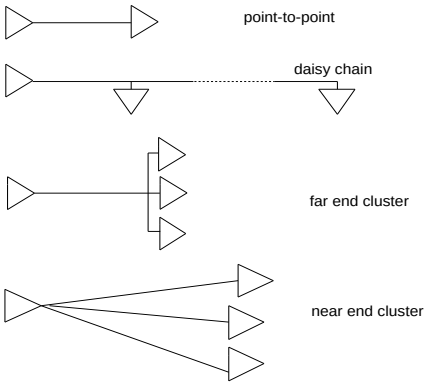


Figure 1: Net Topologies

graph can always be constructed. The edges in the graph have a wiring capacity associated with them, which determines the maximum number of wires that can be routed through the corresponding wiring channel in the given technology. Edge lengths in the routing graph are determined by Manhattan or Euclidean distances.

Once the routing graph is determined, the global routing problem can be formalized as follows [8]: An instance of the global routing problem consists of a routing graph $G = (V, E)$, with *vertices* $V$ and *edges* $E$, and a set of *supernets* $N$, where each super-net is a subset of $V$. A supernet is a collection of identical nets (same terminal nodes) that are treated as a single net with multiple wires. Each edge is labeled with a capacity $c : E \rightarrow R^+$ and edge length $l : E \rightarrow R^+$. Net $i$ has a multiplicity $d_i \geq 1$. For each supernet $i \in N$, there is a set of admissible routes, or *trees* $T_i^1, \ldots, T_i^{i_L}$. A solution to the global routing problem is a set of admissible trees, one or more for each net, such that the capacity $c(e)$ on each edge is not exceeded by the *traffic* on that edge. The *traffic* on an edge is defined by the weighted sum of all the routes that contain edge $e$:

$$U(e) \quad = \quad \sum_{i \in N, t \in i_L, e \in T_i^t} y(i,t) \qquad (1)$$

The weights $y(i,t)$ denote the number of wires in super-net $i$ that are routed using tree $t$. The objective function minimized over all such feasible solutions varies, depending on the design problem. Some formulations try to minimize wirelength, or the maximum ratio of the traffic on an edge to its capacity. The global routing objective in this paper is to satisfy as many of the electrical constraints as possible, subject to the wiring capacity constraints. To do this, a benefit function $b : T \rightarrow R$ is associated with each tree. $b(i,j)$ reflects the likelihood of satisfying the electrical constraints associated with net $i$ when routed using tree $T_i^j$. Hence the objective of the global routing is to maximize $B(T)$:

$$B(T) \quad = \quad \Sigma_{i \in N, j \in i_l} b(i,j) \qquad (2)$$

The global routing problem is optimally solved by finding a set of routing trees for each net in the design with high probability of meeting the electrical constraints, and then maximizing the routing objective function while satisfying the edge capacity constraints.

## 2.1 Integer Programming Formulation

The routing graph definition, the nets, routing trees for each net, and the associated benefit function fully specify a global routing problem. The routing problem can be formulated as an integer program, by associating an integer variable $y_{ij}$ with tree $T_i^j$. $y_{ij}$ is the number of wires in supernet $j$ routed with tree $i$. Then the global routing problem is given by the following integer program:

$$\begin{array}{lll} \text{Maximize} & \sum_{i=1}^{N} \sum_{j=1}^{i_L} b(i,j) y_{ij} & \\ \text{subject to} & \sum_{j=1}^{i_L} y_{ij} = d_i & i = 1, \ldots, N \\ & \sum_{i=1}^{N} \sum_{j=1}^{i_L} a_{ij}^k y_{ij} \leq c_k & k = 1, \ldots, M \\ \text{integer} & y_{ij} \geq 0 & \end{array}$$

Here $N$ is to total number of super-nets, $d_i$ is the number of wires in net $i$, $i_L$ is the number of routing trees for net $i$, $M$ is the number of edges in the graph, and $a_{ij}^k$ is a $(0,1)$ matrix that specifies whether or not tree $T_i^j$ uses the edge $k$.

## 2.2 Solution Methods

Integer programming is, in general, NP-Hard. There are numerous ways of solving integer programs, e.g. cutting plane algorithms and branch and bound [8]. One method that has been shown to be very effective for solving the global routing problem is a randomized rounding applied to the linear relaxation of the integer program [12]. The basic idea is to relax the integer constraint in the formulation, which makes it a linear program, and to solve the linear programming problem. If the solution of the linear program is integral, then we have an optimal global routing. If not, then we need to transform it into an integer solution by rounding the non-integral values. In our experience, almost all solutions turned out to be integer after solving the linear relaxation. This is primarily due to the high multiplicity of the nets. Hence there was usually no need to round to an integer solution. In the few cases with non-integer solutions, a simple rounding led to very few constraint violations, although the objective function might be sub-optimal.

# 3 Circuit Characterization

The global routing methodology depends critically on having accurate measures of the interconnect performance that can be quickly evaluated at design time. A novel simulation based circuit characterization technique

is employed for this purpose. A separate characterization is performed for each unique circuit topology. The objective of the characterization is to obtain a *predictor function* that can be used to accurately and quickly obtain circuit responses of interest (e.g. delay and reflection noise) of a generalized interconnect circuit topology, over a range of certain *design variables*, such as net length.

To obtain a characterization, we conduct a computer experiment in which *samples* are taken at different points in the 'design space', the dimensions of which are the design variables. Taking a sample is akin to performing a circuit simulation. The sampling method is a heuristic, multistage experiment (see Figure 2) that uses Latin Hypercube Sampling [9] in each iteration. Resampling after the first iteration depends on the evaluation of the prediction error, measured by *cross-validation.*

The predictor function is a data interpolant. Interpolation is performed using Moving Least Square Interpolation [6], on the simulated points. The predictor function is given in the following form:

$$\phi(\boldsymbol{x}) = \Sigma_{j=1}^{n} a_j \beta_j(\boldsymbol{x}), \tag{3}$$

where $\boldsymbol{x}$ is the vector of design variables, $\beta_1(\boldsymbol{x}), \ldots, \beta_n(\boldsymbol{x})$ are $n$ linearly independent polynomials in $\boldsymbol{x}$ supplied by the user and the $a_j$'s are constants to be determined. Whenever $\phi(\boldsymbol{x})$ is evaluated, Moving Least Squares are used to determine the $a_j$. The $a_j$'s are chosen so that a weighted sum of the error of prediction at all sample points is minimized, and the predictor function exactly interpolates the sample points.

The flow of the multistage experiment is given in figure 2. Simulations are first performed for a specified initial number of samples. For each of the initially sampled points, predictive error is calculated using *cross-validation.* The cross-validation error is the difference between the actual response at a sample point $\boldsymbol{x}_i$, and the response predicted by interpolating all the other samples at $\boldsymbol{x}_i$ (assuming the response at $\boldsymbol{x}_i$ is not known). If the error is large, it implies that the sample points do not predict at $\boldsymbol{x}_i$. Therefore, it is desirable to sample more points near $\boldsymbol{x}_i$. These errors can be aggregated to identify regions where more samples are to be drawn. In the subsequent steps of the algorithm, Latin Hypercube sampling is carried out for all the regions where the error is larger than a specified threshold.

## 3.1 Benefit Function

The benefit function describes the likelihood of the signal integrity constraints being met by a certain routing tree of a given net. Since there is uncertainty in both the length estimates for the routing trees, as well as the performance estimates, the benefit function should account for these uncertainties.

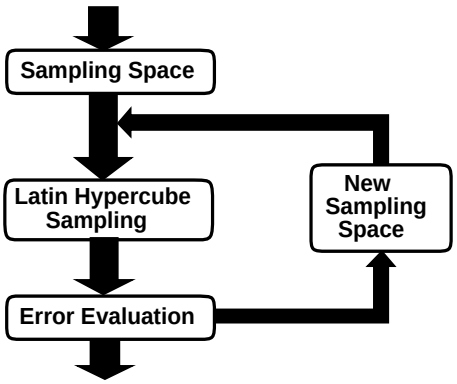


Figure 2: Steps in sequential sampling.

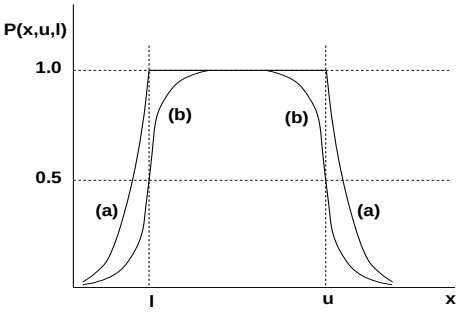


Figure 3: Benefit Functions

The performance can be predicted using Moving Least Square Interpolation (MLSI) on the set of simulated points. With MLSI, there is no estimation of the prediction error. The resampling threshold can be used as a measure of the uncertainty in the characterization. The expression for the benefit function is given as:

$$B(\boldsymbol{x}) = \Pi_{i=1}^{n} P(\phi_i(\boldsymbol{x}) \leq u_i, \phi_i(\boldsymbol{x}) \geq l_i) \tag{4}$$

The $P(\,)$ function can be defined as shown in Figure 3. The $P(\,)$ function labeled (a) is:

$$\begin{aligned} P_i(\boldsymbol{x}, u, l) &= exp(\tfrac{l_i - \phi_i(\boldsymbol{x})}{e_i}) && \phi_i(\boldsymbol{x}) \leq l_i \\ &= exp(\tfrac{u_i - \phi_i(\boldsymbol{x})}{e_i}) && \phi_i(\boldsymbol{x}) \geq u_i \\ &= 1 && l_i < \phi_i(\boldsymbol{x}) < u_i \end{aligned} \tag{5}$$

where $e_i$ is the error threshold in characterizing $p_i$. The $P(\,)$ function labeled (b) is:

$$P_i(\boldsymbol{x}, u, l) = \frac{1.0}{1.0 + exp(\frac{l_i - \phi_i(\boldsymbol{x})}{e_i})} \times \frac{1.0}{1.0 + exp(\frac{\phi_i(\boldsymbol{x}) - u_i}{e_i})} \tag{6}$$

Use of the benefit function represented in Equation 6 will result in more conservative management of delay and signal integrity than the use of the benefit function in Equation 5.

# 4 Tree Generation

The first step in solving the global routing problem is to find a set of routing trees for each net that satisfy the electrical constraints. There has been considerable research on tree generation algorithms in the past, primarily focussed on the *Steiner tree* generation problem, with the objective of minimizing total wirelength. Recently, timing driven Steiner tree generation algorithms have started to emerge [15]. These algorithms model the delay of the tree using a distributed RC model or the Elmore delay model. Again, these models are inadequate for predicting the delay for MCM and PCB interconnect, which display significant inductive or transmission line effects. It is quite difficult to have analytical expressions relating delay and noise to the tree topology and wirelength, since free form routing trees have unpredictable characteristic impedance and discrete discontinuities. Even for the simple wirelength minimization objective, the optimal Steiner tree construction problem is very hard. These two factors make the optimal tree construction problem totally intractable. Fortunately, the nature of the problem allows us to do quite well with heuristic solutions. Firstly, compared to on-chip nets, nets on a PCB or MCM have a smaller fanout. Secondly, the routing resource use does not have to be absolutely minimized. Hence the wirelength can be longer than optimal without resulting in a dramatic increase in resource requirement. Thirdly, the noise and delay problems are well controlled if nets are routed in restricted topologies. It turns out that generating routing trees in these restricted topologies is considerably easier from a computational standpoint. Hence the freedom allowed by relaxing the wirelength minimization objective, makes the tree generation task computationally tractable. The key to constructing feasible routing trees is then to generate routing trees in controlled topologies with small wirelength. These trees should then be checked against the characterizations of noise and delay to ensure that they meet the electrical constraints.

For point-to-point nets, there is only one topological way of constructing routing trees. Hence the shortest paths from the driver to the receiver need to be found. For multi-point nets, several topologies have been shown to have good delay and noise characteristics [1] as shown in Figure 1. Several optimal and heuristic algorithms for generating short wirelength trees for point-to-point and multi-point net topologies have been investigated. The details of these are not included because of space limitations. For details, refer to [10].

# 5 Rule Generation

*Wiring Rules* are explicit constraints on the geometry of the net, for example, a maximum and minimum constraint on each branch in the routing tree. If the electrical performance can be captured in a piece-wise linear function, then the wiring rule can be generated directly [13]. However, such a global rule tends to be fairly conservative and does not lead to routing completion [14]. The global routing solution gives a good starting point for wiring rule generation as it gives a minimum length that is highly likely to produce a feasible route. If the global route is feasible, then a wiring rule can be generated by expanding the design space around the global routed solution.

Lee et.al. [7] have proposed a novel method for generating bounds on net lengths to meet electrical constraints. Their approach can be summarized as follows: First, for all electrical constraints, a feasible solution is found using semi-empirical formulas. Then a simulation technique based on AWE [11] is used to calculate the sensitivities of the electrical performance to the net lengths. Then the electrical performance is approximated by a Taylor series expansion where initial values and the partial derivatives were obtained from simulation. The largest value of the net lengths is solved for by linear programming for each performance separately, and then the solution spaces are intersected to determine the intervals of consistency.

The main drawback of this technique is that the simple equations cannot account for complex driver/receiver models, and thus the initial solution may not be feasible. The net length bounds are generated disregarding the constraints induced by the board placement. Nevertheless, the approach is quite attractive and can be extended to overcome these shortcomings, using the global routing described earlier to arrive at the initial solution.

Moving Least Square Interpolant gives us a local estimate of the electrical performance. Recall that the form of the Interpolant is:

$$\phi(\boldsymbol{x}) = \sum_{i=1}^{n} a_i(\boldsymbol{x})\beta_i(\boldsymbol{x}) \tag{7}$$

where $\phi(\boldsymbol{x})$ is the predicted value for some electrical performance for net length vector $\boldsymbol{x}$ and $\beta_i$'s are polynomial basis functions. If the basis functions are chosen to be linear in $\boldsymbol{x}$, then the form of $\phi(\boldsymbol{x})$ is that of a local linear approximation. By choosing $\boldsymbol{x}$ as the global routed net length, we have a linear approximation of the interconnect performance near the likely design point. Note that any electrical performance can be approximated in this manner. So the minimum length constraints obtained from the global routed length and the constraints specified by equating the linear approximations of the performance to the electrical constraints, describe a polytope over the space of net lengths. To obtain absolute bounds on the net lengths a largest hyperrectangle has to be fitted in this polytope. This can be achieved by linear

programming as presented in the next section.

### 5.1 Formulation

Suppose that the net is described by a vector of *physical design variables* $(x) = (x_1, \ldots, x_d)$. There are $m$ electrical performances of interest $p_1, \ldots, p_m$, with upper and lower bounds, given by $u_1, \ldots, u_m$ and $l_1, \ldots, l_m$. The global routed solution is the point $x_c = (x_{c1}, \ldots, x_{cd})$. The electrical performances $p_j$ is approximated by the predictor function $\phi_j = \sum_{i=0}^{d} a_{ij} x_i$ where $x_0 = 1$. The wiring rule generation problem is to fit a maximal hypercube in the polytope defined by the linear inequalities:

$$\begin{array}{ll} x_i \geq x_{ci} & i = 1, \ldots, d \\ \sum_{i=0}^{n} a_{ij} x_i \leq u_j & j = 1, \ldots, m \\ \sum_{i=0}^{n} a_{ij} x_i \geq l_j & j = 1, \ldots, m \end{array}$$

The constraints bounding the polytope define hyperplanes in $R^d$. The hyperplane corresponding to the $j$th constraint is denoted $\pi_j$. For example the hyperplane corresponding to the constraint $\sum_{i=0}^{d} a_{ij} x_i \leq u_j$ is $\pi_j \equiv \sum_{i=1}^{d} a_{ij} x_i = u_j - a_{0j}$. Now the problem of fitting a maximal hypercube in the polytope defined by the constraints is given as:

$$\begin{array}{ll} \text{maximize} & r \\ x^0, r & \\ \text{subject to the constraints} & \\ d_n(x^0, \pi_j) \geq r, & j = 1, \ldots, d + 2m \end{array}$$

There are $d + 2m$ hyperplanes corresponding to the $d$ lower bound constraints from global routing and $m$ upper bound performance constraints and $m$ lower bound performance constraints.

$$\begin{array}{ll} \text{maximize} & r \\ x^0, r & \\ \text{subject to the constraints} & \\ x_i^0 - x_{ci} \geq r, & i = 1, \ldots, d \\ \sum_{i=1}^{d} a_{ij} x_i^0 + r \sum_{i=1}^{d} a_{ij} \leq u_j - a_{0j}, & j = 1, \ldots, m \\ \sum_{i=1}^{d} a_{ij} x_i^0 - r \sum_{i=1}^{d} a_{ij} \geq l_j - a_{0j}, & j = 1, \ldots, m \end{array}$$

If this program has a solution, the maximum and minimum constraints on the variable $x_j$ are simply given as:

$$\begin{aligned} x_{ju} &= x_j^0 + r \\ x_{jl} &= x_j^0 - r \end{aligned}$$

This problem has a similar flavor as the design centering problems considered in [2]. Hence a lot of the extensions therein, e.g. variable scaling, are applicable to our formulation.

# 6 Results

In this section, several design examples are given to show the effectiveness of the global routing procedure and the wiring rule generation methodology. The netlist for two of the design examples were obtained from MCC and one from Intel Corporation. Each of these designs is done with only two signal wiring layers. Hence the global routing procedure is quite applicable to these designs. The MCC design examples are for MCMs and the Intel example is for a PCB. The Intel design has timing constraints and also wiring rules given in [5]. There are no available timing constraints for the MCC design examples. Hence these constraints were generated using statistical arguments. The only information available about these designs is the placement and a netlist. Hence the interconnect and driver/receiver models had to be assumed. Since the timing constraints were generated based on these same models, there is a fair basis for evaluating how well the global routing procedure is able handle performance constraints.

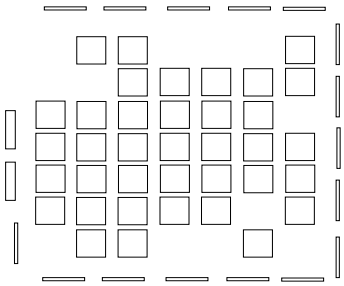

Figure 4: Placement for MCC1

### 6.1 MCC Design Example 1

This routing example has 37 gate arrays chips and 18 high density connectors on a 6 x 6 inch substrate. The net list contains 7114 signal nets and 14659 pins. There are two available signal layers. The placement for this example is shown in Figure 4. Two graph models were generated for this design. The first mapped each chip and edge connector to one vertex in the channel graph, and had 64 vertices, 112 edges and 307 supernets. The second model maps each chip edge to a separate vertex, and each edge connector to one vertex in the channel graph. The resulting graph has 332 vertices, 378 edges and 1200 super nets.

Almost all nets in this examples have only two or three terminals. No information about the driver/receiver models, package parasitics or the interconnect models was available for this design, and hence had to be assumed. The interconnect model was a buried microstrip with 50 ohms characteristic impedance. Settling delay [3] to withing 8% of the supply voltage was chosen as the performance measure. Two different sets of timing constraints were generated. Settling delay for the two and three terminal nets was characterized using 75 and 150

samples respectively. Several routing experiments were run by varying the graph model routing topologies, number of routing trees, benefit function (ABS is the function shown in figure 3(a) and PROB is the one in figure 3(b)), channel capacities, and the timing constraints (Loose or Tight).

Table 1 gives a description and results of several studies performed for this design. The second last column gives the value of the objective function. The global routed lengths for each of the nets was simulated and the simulated delay was compared against the constraints. The last column shows the number of nets for which the simulated delay met the constraints. The total number of nets is 7114.

Table 1: Routing Experiments for MCC1

| No. | Model | Slacks | Benefit | Objective | Good Nets |
|---|---|---|---|---|---|
| 1 | small | Loose | ABS | 7108 | 7102 |
| 2 | small | Tight | ABS | 7104.1 | 7102 |
| 3 | small | Tight | PROB | 4808.3 | 7104 |
| 4 | large | Tight | PROB | 4665.3 | 7085 |
| 5 | large | Tight | ABS | 7093.5 | 7086 |

The routing results indicate that the combination of the global routing and characterization gives a very good indication of successful design completion under signal integrity and congestion constraints. In the worst-case 98.5% of the nets were successfully routed.

## 6.2 MCC Design Example 2

The second design example is another MCM design from MCC. It consists of 6 chips, 765 I/O pins and contains 799 signal nets, two power and one ground net. There are 2496 pins total, 2043 of which are signal pins. There are numerous 3 to 7 pin nets. There are two signal layers, and separate power and ground layers. Figure 5 shows the placement for this example. The channel graph has a total of 69 vertices and 78 edges. The number of super nets is 119. The routing problem is particularly difficult because of the multi-pin nets in the design. The interconnect, package and driver/receiver models used were the same as those for the first example. Several routing experiments were undertaken, as summarized in table 2

From the routing results, it is seen that the design constraints were quite well satisfied with the global routing procedure. In the worst case, 92.2% of the nets were successfully routed. Also notice, that the routing results improve considerably when stubs are introduced in the routing trees in a controlled manner. There is, however, a direct tradeoff present here. Stubs do improve routing congestion, and create shorter routing trees. However, the time taken for characterization increases considerably, as each stub introduces an extra characterization variable. The important fact is that the characterizations are done *a priori*, and do not prolong the design time.

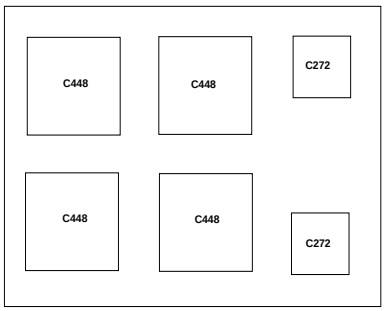


Figure 5: Placement for MCC2

Table 2: Routing Experiments for MCC2

| No. | Slacks | Trees | Objective | Good Nets |
|---|---|---|---|---|
| 1 | Tight | chain | 767.25 | 734 |
| 2 | Loose | chain | 786.32 | 760 |
| 3 | Tight | stubs | 765.69 | 753 |
| 4 | Loose | stubs | 796.0 | 793 |

## 6.3 Intel Pentium Board Design

The last design example is of the Intel Pentium Board Design. Figure 6 shows the component placement for the PCIset ISA Reference Design PCB Layout, and figure 7, the corresponding channel graph. Only the Pentium chip, the Local Bus Accelerator chips(LBXs) , the Cache SRAMs and the PCMC are shown in this placement. This is the only part of the layout that has the high speed (66 MHz) signals on it.

Table 3 shows the design space for each of the net classes, and the maximum settling delay in this design space from simulation. Design rules were generated for each net using the rule generation methodology. 3 of

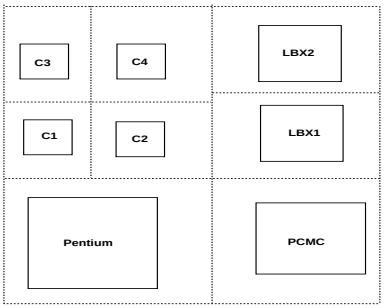


Figure 6: Placement for Intel Pentium Board

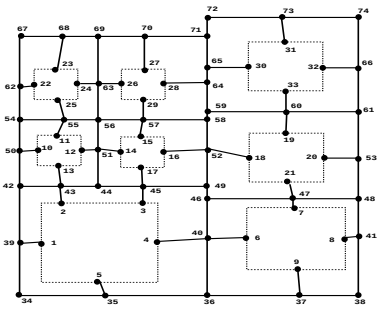

Figure 7: Channel Graph for Pentium Board

the nets in this design have no rules. This is because the global routed length for these nets did not meet the delay constraint. The reason for this is that the edge lengths in this graph were such that they may overestimate the actual routed lengths in some cases. The efficacy of the design rules was measured by the safeness coefficient, as described in [13]. Almost all design rules were 100% safe, with the worst coefficient being 72% . For more details, please see [10].

Table 3: Constraints for Intel Pentium Design

| Net Class | Design Space (m) | Delay (ns) |
|---|---|---|
| Pentium-PCMC | $0.0 \leq l_0 \leq .165$ | 7.49 |
| PCMC-Pentium | $.058 \leq l_0 \leq .12$ | 7.82 |
| PCMC-LBX | $0.0 \leq l_0 \leq .20$ | 10.61 |
| | $0.0 \leq l_1 \leq .14$ | 11.6 |
| Data Bus | $0.0 \leq l_0 \leq .12$ | 1.94 |
| | $0.0 \leq l_1 \leq .11$ | 1.42 |
| Addr. Bus | $0.0 \leq l_0 \leq .11$ | 6.58 |
| | $0.0 \leq l_1 \leq .12$ | 10.15 |

# 7 Disucssion

The strength of the approach described above lies in developing models of interconnect performance offline. The global routing and rule generation procedures are typically quite fast. The tree generation procedure was the most time consuming, taking about 10 CPU minutes for the largest routing problem discussed above. The time taken for the characterization methodology described here depends on the nature of the response and also the number of design variables. The largest net in the examples above required several hundred simulations for an accurate model. Again, the characterization is performed before detailed design, and can be reused across several designs.

In conclusion, the combination of global routing, circuit characterization and rule generation described here, provides a complete solution path for meeting tight performance constraints in layout design for MCMs and high speed PCBs.

# Acknowledgments

The authors wish to thank Slobodan Simovich for his contributions to the circuit characterization work.

# References

[1] E.E. Davidson and G.A. Katopis. Package Electrical Design. In R.R. Tummala and E.J. Rymaszewski, editors, *Microlectronics Semiconductor Handbook*, chapter 3. Van Nostrand Reinhold, 1989.

[2] S. W. Director, W. Maly, and A. J. Storjwas. *VLSI Design for Manufacturing : Yield Enhancement.* Kluwer Academic Publishers, 1990.

[3] P. D. Franzon. Chapter 11. In D. A. Doane and P. D. Franzon, editors, *Multichip Module Technologies and Alternatives: The Basics.* Van Nostrand Reinhold, 1992.

[4] R. Gupta and L. T. Pillage. OTTER: Optimal termination of transmission lines excluding radiation. In *Proceedings of the 31st DAC*, pages 640–645, 1994.

[5] Intel Corporation. *Pentium Processor/82430 PCIset Open Design Guide*, 1993.

[6] Peter Lancaster and Kestutis Salkauskas. *Curve and surface fitting : An introduction.* Academic Press, 1986.

[7] J. B. Lee, E. B. Shragowitz, and D. Poli. Bounds on net lengths for high speed PCBs. In *Proc. of ICCAD*, pages 73–76, 1993.

[8] Thomas Lengauer. *Combinatorial Algorithms for Integrated Circuit Layout.* John Wiley and Sons, 1990.

[9] M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, May 1979.

[10] Sharad Mehrotra. *Automated synthesis of high speed digital circuits and package-level interconnect.* PhD thesis, North Carolina State University, 1994.

[11] L. T. Pillage and R. A. Rohrer. Asymptotic Waveform Evaluation form timing analysis. *IEEE Trans. on CAD*, 9:pp. 352–366, 1990.

[12] P. Raghavan and C. D. Thompson. Multiterminal global routing: A deterministic approximation. *Algorithmica*, 6:73–82, 1991.

[13] S. Simovich, S. Mehrotra, P. Franzon, and M. Steer. Delay and reflection noise macromodeling for signal integrity management of pcbs and mcms. *IEEE Trans. on CPMT*, 17(1):15–21, 1994.

[14] Slobodan Simovich. *A methodology for automated simulation-based electrical characterization and design of high-speed systems.* PhD thesis, North Carolina State University, 1994.

[15] M. Sriram and S. M. Kang. *Phyiscal Design of Multi Chip Modules.* Kluwer Academic Press, 1993.