

Contents

List of Figures	iv
List of Tables	v
1 Introduction	1
1.1 Introduction	1
1.2 High Speed Circuits and Interconnect	3
1.2.1 MOS Delay Estimation	5
1.2.2 High Speed Interconnect Modeling	6
1.3 Circuit Characterization	10
1.4 Circuit Optimization	11
1.5 Simulation Based Design	13
1.5.1 Problem Complexity	14
1.6 Thesis Outline	18
2 Background and Existing Approaches	20
2.1 Macromodeling	20
2.2 Experimental Design	22
2.3 Circuit Optimization	25
2.3.1 Global Optimization Approaches	26

	iv
2.3.2	Algorithms based on Statistical Modeling 28
2.4	Summary 30
3	Thesis Statement 31
3.1	Research Goals 31
4	Characterization and Optimization Methodology 33
4.1	Characterization Methodology 33
4.2	Sequential Experimental Design 38
4.2.1	Characterization of Error in Prediction 39
4.2.2	Predictor Function 40
4.3	Implementation 42
4.3.1	Identifying the Design Variables and Initial Experimental Region 42
4.3.2	Identifying the next Experimental Region 44
4.3.3	Multiple Response Evaluation 46
4.3.4	Stopping Rules 46
4.4	Optimization Methodology 48
4.4.1	P-Algorithm 50
4.4.2	Implementation 51
4.4.3	Constrained Optimization 52
5	Global Routing and Rule Generation 55
5.1	High performance layout design 55
5.2	Global Routing 56
5.2.1	Formulation 58
5.2.2	Constrained Tree Generation 61
5.2.3	Benefit Function 72

5.2.4	Integer Programming Formulation	75
5.2.5	Solution Methods	76
5.3	Wiring Rule Generation	77
5.3.1	Formulation	79
5.4	Summary	84
6	Tools	85
6.1	Introduction	85
6.2	MetaSim	85
6.3	Characterization Tool	87
6.4	Global Router	90
7	Experimental Results	92
7.1	Characterization Experiments	92
7.1.1	Multi-Chip Module Interconnect	93
7.1.2	High Speed Data Latch	102
7.2	Optimization Experiments	108
7.2.1	Combinational Logic Element for Wave-pipelining	108
7.2.2	Clock Driver Circuit	111
7.2.3	Bistable Latch	117
7.2.4	High Speed Memory Bus	118
7.3	Global Routing and Wiring Rule Generation	123
7.3.1	Generation of Timing Constraints	126
7.3.2	MCC Design Example 1.	128
7.3.3	MCC Design Example 2.	134
7.3.4	Intel Pentium Board Design	137

	vi
7.4 Summary	142
8 Conclusion	147
8.1 Future Work	148
Bibliography	152
A MCC Designs	158
A.1 MCC1 Netlist	158
A.1.1 Coarse Graph	158
A.1.2 Fine Graph	167
A.2 MCC2 Netlist	190
A.3 Point to Point Net Description	200
B Equations	202
B.1 Stochastic Model Equations	202
C Intel Design	204
C.1 Netlist	204
D User Interface	209
D.1 The User Interface	209

List of Figures

1.1	Digital Circuit Timing	4
1.2	MOS equivalent model for timing analysis	6
1.3	RC-tree model	8
1.4	RLC-tree model	8
1.5	Signal Settling Time	10
1.6	Feasible Physical Design Space	14
2.1	Central Composite Design	24
2.2	Latin Hypercube Designs	25
4.1	Study Generator	43
5.1	Channel free design style	59
5.2	Channel Graph	60
5.3	Net Topologies	63
5.4	Example Graph	66
5.5	Adding a stub saves an escape path on a PGA	71
5.6	Benefit function	74
5.7	Benefit function	75
5.8	Lee's Approach to Rule Generation	78

5.9	Rule Generation Approach	80
6.1	Tools Hierarchy	86
6.2	MetaSim	88
6.3	Global Router	91
7.1	Net Topology	94
7.2	Signal Settling time	94
7.3	Partial characterization of MCM interconnect. $l_1 = 3\text{cm}$	96
7.4	Sampled characterization MCM interconnect. $l_1 = 3\text{cm}$	97
7.5	MCM Net Topology	99
7.6	Interconnect Cross-section	100
7.7	Waveform Parameters	100
7.8	Schematic of Latch Circuit	103
7.9	Signal Delay plot for data rise time of 0.3 ns: Actual response	104
7.10	Signal Delay plot for data rise time of 0.3 ns: Predicted response	105
7.11	Scatter plot of sample points. (a) Initial Sample (b) 2nd Sample	106
7.12	Cross coupled NAND gate	111
7.13	Circuit Block for Wave-pipelining	112
7.14	Possible input transitions	113
7.15	2 phase latching scheme	113
7.16	Clock Driver Circuit	114
7.17	Skew Definition	114
7.18	Bistable Latch Circuit	119
7.19	Latch Waveforms	120
7.20	Latch Waveforms for design in Sampler	121
7.21	Latch Waveforms for Optimal Point	122

7.22	Circuit model for PCB Bus	125
7.23	Distribution of timing slacks for CMOS systems	127
7.24	Distribution of shortest path lengths for MCC1	128
7.25	Placement for MCC1	129
7.26	Coarse Channel Graph for MCC1	130
7.27	Fine Channel Graph for MCC1	131
7.28	Placement for MCC2	135
7.29	Channel Graph for MCC2	136
7.30	Placement for Intel Pentium Board	139
7.31	Channel Graph for Pentium Board	140

List of Tables

5.1	Short Path list for graph in figure 5.4	67
7.1	Error Statistics for the MCM interconnect characterization	95
7.2	Error Statistics for the 3 Terminal Net Characterization	97
7.3	Error Statistics for the 3 Terminal Net Characterization	98
7.4	Error Statistics for the 4 Terminal Net Characterization: Delay and Settling Delay	101
7.5	Error Statistics for the 4 Terminal Net Characterization: High and Low Undershoot	101
7.6	Error Statistics for the Latch Characterization	105
7.7	Results for Delay Controlled Element	111
7.8	Results for Clock Driver Circuit	114
7.9	Results for Bistable Latch	118
7.10	Simulated Points for PCB Bus	124
7.11	Routing Experiments for MCC1	133
7.12	Routing Results for MCC1	133
7.13	Routing Experiments for MCC2	136
7.14	Routing Results for MCC2	137
7.15	Constraints for Intel Pentium Design	142
7.16	Generated Rules for Intel Pentium Design	143

7.17	Generated Rules for Intel Pentium Design...	144
7.18	Generated Rules for Intel Pentium Design...	145
7.19	Generated Rules for Intel Pentium Design...	146

Chapter 1

Introduction

1.1 Introduction

Over the past few decades, the complexity and speed of operation of integrated circuits has increased dramatically. This has been brought about primarily due to the improvement in silicon processing technology and the miniaturization of device dimensions. In addition, highly structured design methodologies have been put into place and numerous computer aided design tools have been developed to assist the human designer in managing the design complexity. The design task is usually to convert a functional specification of the system to a detailed physical representation, e.g. masks for intergrated circuits. Of course, a given functional specification maps onto numerous physical implementations. It is desirable to find one that is optimal in some sense, e.g. speed of operation, power consumption, size etc. and is cheaply manufactured. The complexity of the design is managed by breaking down the design process into several stages. At each stage, a different representation of the system is considered, e.g. the architecture description stage, the logic design stage or the physical design stage. The performance goals are suitably modeled and a design is found that optimizes these performance goals and meets the functional specification. The

design percolates through the hierarchy of abstraction until a detailed physical implementation is devised. The performance of the system is then evaluated by prototyping and measurement. The design and prototyping process is extremely time consuming and expensive. It is not feasible to iterate on this process in order to produce a better design. Hence along with the design tools, *simulation* tools are employed that predict the performance of the system from a certain abstract representation. The simulation tools contain models of the behavior of the primitive elements in the representation and evaluate the response of the system to a certain stimulus. For example, a circuit level simulator e.g. SPICE [46], incorporates models of passive elements, transistors etc. and evaluates the voltages and currents in a circuit under the stimulus of voltage and current sources. The rapid advancement and increase in accuracy of circuit level simulation tools allows for the evaluation of the design without resorting to an expensive prototyping and measurement.

Though simulation tools help in identifying design problems, they can only be invoked once a full circuit representation is generated. Even the process of design and simulation is not very useful for improving system performance. This is because the design process and the simulation process are both quite time consuming and expensive. Hence it is necessary to consider circuit performance during design through suitable models, and be able to optimize the performance using these models.

The focus of this thesis is on considering the problem of designing high performance circuits and systems. Suitable evaluation models for these circuits have to be devised, such that the models can be quickly evaluated by a design tool in generating a circuit that meets performance goals. Performance models can be generated in several ways. It is argued in this thesis that the most accurate model of a circuit's performance is a circuit simulator. However, circuit simulation is computationally expensive. Hence an attempt is made to capture the information generated from cir-

cuit simulation in a model that is quickly evaluated, and retains the accuracy of the simulator. The main design problems considered here are those of directly optimizing the circuit performance by manipulating certain parameters in the physical design, and also finding suitable circuits that meet certain performance goals. The latter problem is of designing interconnect in PCBs and MCMs to meet signal integrity and delay requirements.

1.2 High Speed Circuits and Interconnect

The primary concern in this thesis is in devising methods to improve the operating speed of circuits. The operating speed of circuits is limited by the intrinsic delay in the switching of active circuit elements, e.g. MOS transistors, and either the time to charge the interconnect capacitance or the time of flight across the signal transmission medium. In synchronous digital systems, the constraints on maximum delay are imposed by the period of the clock that synchronizes the various storage elements. For example, consider the digital circuit shown in figure 1.1. The combinational circuit has inputs from two edge triggered flip-flops C1 and C2, and its outputs are captured by two flip-flops C3 and C4. All flip-flops are synchronized by a common clock CK. Timing constraints are imposed by the set up and hold time requirements of the flip-flops C3 and C4. The data to be latched must be stable before the set-up (referenced to a clock edge) and must continue to remain stable during the hold time. The time taken for data to propagate through the combinational circuits must observe constraints imposed by the set up and hold time requirements. For example, the set-up time requirements for C3 impose the following constraints:

$$t_{del_max} \leq T_{CK} - t_{skew} - t_{set-up} \quad (1.1)$$

Where T_{CK} is the clock period, t_{skew} is the maximum clock skew, and

t_{set-up} is the set-up time for the flip-flop C3. This is, of course, a simple example. The timing constraints are more complicated for multi-phase clocks, transparent latches and wave-pipelined systems. See [21] for a good discussion on various timing constraints.

The key to successful high performance digital system design, is then to obtain good models of the delays through transistor networks and the signal interconnections. Modeling circuit delay is a difficult problem. In the following two sections, some of the common methods for modeling the delay of CMOS circuits and interconnect structures respectively, are reviewed.

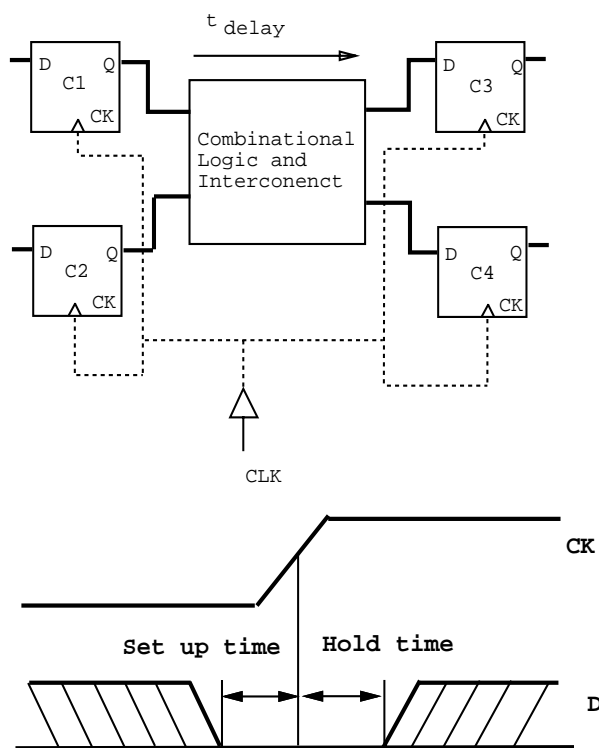


Figure 1.1: Digital Circuit Timing

1.2.1 MOS Delay Estimation

Sapatnekar [55] gives two alternatives to circuit delay estimation:

- The *micromodeling* approach where each transistor is replaced by an equivalent simplified model. Such a model is shown in figure 1.2 for a MOS transistor modeled as a voltage controlled switch with on-resistance R_{on} between drain and source and grounded capacitances from the drain, source and gate terminals, respectively C_d , C_s and C_g . The advantage of using such a model is that it offers a simple closed-form expression for the delay of a circuit. In addition, the delay of an entire circuit can be computed efficiently if the interconnect is also represented by an RC tree, by using the Elmore delay approximation [16]. Moreover, the circuit parameters C_d , C_s , C_g and R_{on} are easily related to the device dimensions. Hence performance optimization can be done by directly manipulating the physical device dimensions. The main drawback of this approach is the loss of accuracy. This is a severe limitation, especially since shrinking device geometries make several effects such as short channel effects, input slope dependence etc. that cannot be considered in this model.
- The other modeling approach is called *macromodeling*. Here the primitive circuit of interest is a logic gate. In macromodeling techniques, the delay of a gate is related directly to several parameters like device width, load capacitance, input rise time etc. This relationship is established through circuit simulation. Several ways of capturing this model are possible, the easiest of which is to build a look up table. Matson [44] fit simulation results to a non-linear equation to capture the delay model.

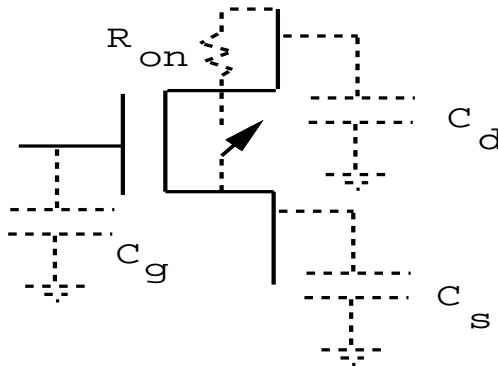


Figure 1.2: MOS equivalent model for timing analysis

The trade-off between the two models are obvious. When very large circuits are considered, the first model is more suitable because of the speed of evaluation and the tractability of the optimization formulations. This analysis, however, is not accurate at high operating frequencies or small geometry devices. The second approach is computationally more intensive, but has a considerable advantage in accuracy. The key to obtaining a middle ground is to generate simulation-based macromodels for design, while minimizing the number of simulations required for generating the models.

1.2.2 High Speed Interconnect Modeling

Modeling of high speed interconnect is an even more difficult task. Also, the line models depend heavily on the line cross section, and its resistivity of the line material. Interconnect modeling is done in several different ways, depending on the degree of accuracy required [63]:

- **Lumped capacitor model:** The interconnect is treated as a single lumped capacitor, with capacitance C equal to the total distributed capacitance of the interconnect. The interconnect delay is then estimated as $R_d C$, where R_d is the output impedance of the driver circuit. Such modeling is effective for VLSI interconnect with large driver output impedance and slow signal rise times.
- **RC tree model:** The line is modeled by several RC segments as shown in figure 1.3. This accounts for the resistance of the interconnect, and is useful for small geometry VLSI interconnect. The delay is usually modeled by the Elmore time constant [16].
- **RLC tree model:** The line is modeled by several RLC segments as shown in figure 1.4. such modeling is required when the signal rise time is comparable to the time of flight across the interconnect, and hence the interconnect inductance cannot be ignored. Also, if the driver impedance is smaller than the characteristic impedance of the line, the underdamped response of the line cannot be captured by an RC model. Several techniques, such as Asymptotic Waveform Evaluation [49] are being investigated to quickly estimate the delay of RLC trees.
- **Transmission line modeling:** In transmission line models, the modes of propagation of the signal in the interconnect are directly represented, e.g. using the telegrapher's equations, or the method of characteristics. Frequency dependent material properties can also be modeled. Transmission line modeling provides the most accurate method for determining signal characteristics. It is, however, computationally very expensive.

Designing high speed interconnect with distributed loading is a very complex task. The first choice to be made is that of a suitable electrical model, e.g. the

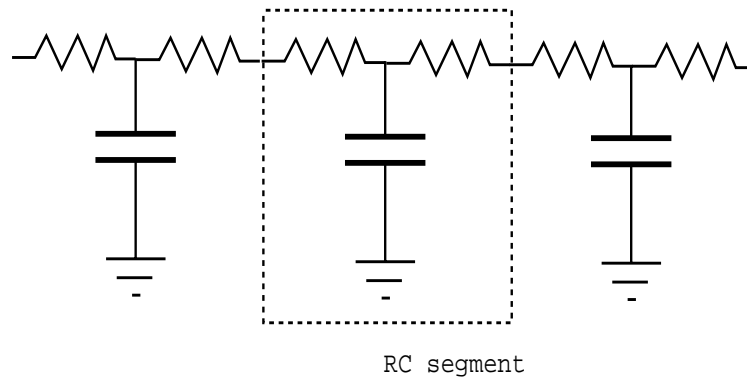


Figure 1.3: RC-tree model

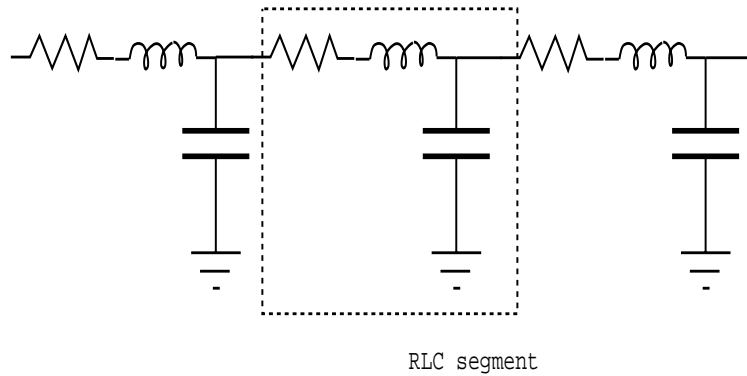


Figure 1.4: RLC-tree model

RLC tree or a transmission line. Additionally, parasitic capacitances and inductances, induced by vias, chip attaches and bond pads must be properly modeled. The driver and receiver circuits should also be suitably modeled. One problem with obtaining circuit level models of drivers and receivers is that most such design are proprietary. The IBIS [23] model is a step towards alleviating this problem. The IBIS model is a standardized representation of driver and receiver circuits and their electrical be-

havior is captured by I-V tables. Another major concern is the various noise sources, such as reflections, simultaneous switching and cross-talk. The cumulative effect of these noise sources is that the signal transitions are non-monotonic. Hence modeling the propagation delay is not enough. In a latch based design, as shown in figure 1.1, the signal should have safely settled above a certain noise threshold before it can be sampled by the flip flop to determine the logic level, as in figure 1.5. This is dictated by the noise immunity characteristics of the receiver circuit. See Doane & Franzon [18] for a good discussion on the relationship between delay and noise.

Davidson and Katopis [11] present a comprehensive methodology for managing noise and determining signal delay in high speed nets. This approach is entirely dependent on circuit simulation to predict signal waveforms. From the simulation results, a set of **wiring rules** is developed to constrain the geometry and topology of the interconnect structure. Obeying these wiring rules ensures that all receivers in a multi drop net switch at the first incidence of the signal waveform and subsequent reflections from the discontinuities do not alter their state. Within the confines of the wiring rule, a **delay equation** is generated by fitting a polynomial to a set of simulation results.

More traditionally, the wiring rules are specified as **rules of thumb** and the delay equations are generated from *analytical modeling*. The rules of thumb are based on a designer's experience and are used to constrain the geometry and loading on nets to ensure proper electrical behavior. An example of a rule of thumb is a constraint placed on the length of a stub wire attaching a receiver to a signal line for controlling reflection noise. Analytical equations like those for time-of-flight, and loaded time-of-flight across a signal line relate signal delays to the loading, length and material properties of the interconnect. Rules of thumb and analytical equations have been extensively used in the past [71]. However, for modern high speed designs, this

approach is inadequate. This is because analytical equations are generated under simplifying assumptions that severely degrade their accuracy. The rules of thumb tend to be extremely overconstraining, resulting in a failure to complete designs.

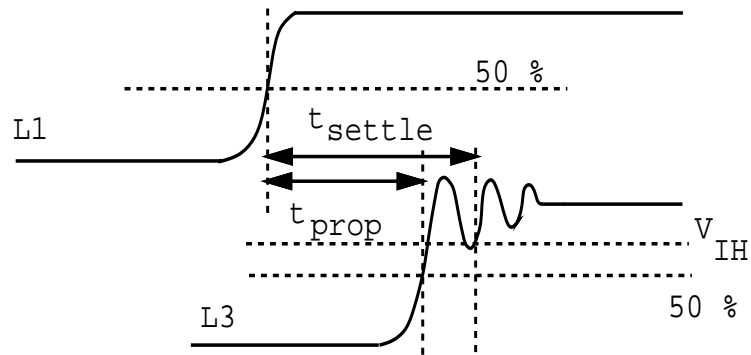


Figure 1.5: Signal Settling Time

1.3 Circuit Characterization

From the previous discussions, it is apparent that the design of high speed systems requires the careful analysis of the system based on circuit simulations. Benkoski [3] terms the process of *observing* the *behavior* of a circuit under different *conditions* and building a *simplified model* that exhibits similar characteristics, *circuit characterization*. The emphasized keywords in this definition require further elaboration. The *behavior* of the system that is of interest are the performance parameters such as circuit delay, power consumption, noise etc. The *observation* is performed by running a simulation on a circuit model, and extracting the relevant performance parameters from the voltage and current waveforms. The *conditions* are a set of circuit parameters that are varied over a range that represents the range of circuit designs that can be

manufactured. These circuit parameters might be the geometrical parameters of the devices and interconnect, or the process parameters used to manufacture the devices. Finally the *simplified model* is a suitable abstract representation that captures the relationship between circuit parameters and the performance parameters as closely as possible. A key feature of this model is that it should be computationally much cheaper to evaluate than a circuit simulation. In certain cases, there might be other restrictions on the analyticity of the model to ensure its suitability for mathematical programming.

1.4 Circuit Optimization

The process of circuit optimization is quite similar to characterization. In this case we are more interested in finding a single circuit representation, that exhibits optimal performance. For high speed systems, it is once again necessary to resort to an expensive circuit simulation, each time the performance of a circuit needs to be evaluated. One method of circuit optimization is to set up an interaction between a non-linear optimization tool, and a circuit simulator. A parametric circuit simulation is performed each time the optimization tool needs an evaluation of the circuit performance. There are considerable difficulties in employing this simplistic approach directly:

1. Most non-linear optimizers require the gradient of the objective function to be computed. Gradient information is very difficult to obtain from simulation. Though there are numerical optimization techniques which do not require explicit gradient information, these techniques tend to be slow. Also, they try to evaluate the gradient through perturbation. This implies a further increase

in the number of objective evaluations.

2. Most numerical optimization techniques operate on a continuous parameter range. Some circuit parameters, e.g. transistor widths, can only be varied in fixed minimum quanta. Hence the final solution might be an infeasible sizing scheme. Moving the solution to the closest feasible size may lead to a sub-optimal solution.
3. The user has no direct control over the optimizer, i.e., the optimization task is not interactive. Usually an experienced engineer has considerable insight into the behavior of the circuit being optimized. An interactive optimization method is much more likely to converge to a good solution quickly compared to a fully automated method.
4. The optimization routines look for strict local minima. Usually, the designer is interested only in obtaining a rough approximation to a globally optimal solution. To achieve the global minimum, the optimizer has to be run from multiple, random, initial solutions. Even then, there are not even theoretical guarantees of achieving a globally optimal solution, except for some very restricted problems.

In view of these difficulties a new approach to circuit characterization and optimization is required. Part of this thesis is devoted to developing such a methodology. In the next section, the characterization and optimization problems are rigorously defined as a **simulation based design** problem.

1.5 Simulation Based Design

The simulation based design problem can be formalized as follows. The physical system under consideration is represented by an electrical circuit model, which has a fixed structure and some given element types. The system is manipulated by a certain set of *physical designable parameters*. These parameters are such that the performance of the circuit is directly affected by a change in these parameters. These parameters are represented by a vector x^d where $x^d \in R^d$. Of course, there are limits on the size of these design parameters. These limits are called the *physical design space* A , where $A \subset R^d$. In most design problems, A is a closed convex space. In some instances, A might be a finite subset of R^d , for example, when the physical design parameters are transistor widths that can be varied only in fixed minimum quanta. In addition to the design parameters, there are several *performance parameters*. These are represented by the vector y^p , where $y^p \in R^p$. The mapping $x^d \rightarrow y^p$ is established only through a circuit simulation. We presume that the simulation gives the most accurate information y^p we can hope to get.

Two kinds of design problems are of interest. In the first, we are interested in finding which circuits meet certain bounds on the performance parameters. For example, figure 1.6, shows a hyperrectangular subset of R^p called E^p which represents acceptable performance bounds for a circuit. These bounds are achieved only by a certain subset of the physical design space. This region is called the *feasible physical design space* A_c as shown in figure 1.6. This region can, in general, be non-convex and even disconnected. Given an element of A it is required to determine if it lies within A_c , and if it does, then a closed convex space around it, that lies entirely within A_c and is maximal in some measure. To solve this problem, it is first necessary to find a close approximation of the mapping $x^d \rightarrow y^p$, which is relatively much cheaper to

evaluate than a full circuit simulation. This approximation is termed a *macromodel*.

The other design problem is to find a suitable vector x_0^d within A which optimizes the response y^p in some sense. When $p = 1$, we are interested in x_0^d which maximizes or minimizes y . Otherwise, we are interested in either a point that maximizes or minimizes a linear function or y^p , or maximizes or minimizes only one element of y^p subject to constraints on the values of the other elements.

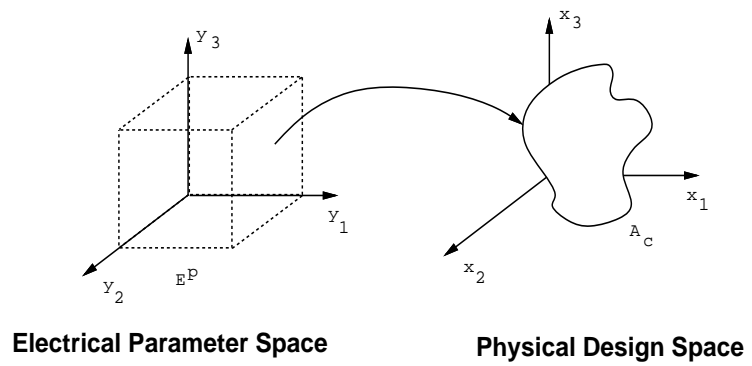


Figure 1.6: Feasible Physical Design Space

1.5.1 Problem Complexity

The computational complexity of a problem is usually measured by the minimal computational resources, such as time or memory, required for its solution. It is also the minimal cost among all algorithms that solve the problem. The problems that we seek efficient solutions to in this thesis relate to function approximation and function optimization. The solution domain for these problems has, in general, infinitely many elements. Thus only partial information will be available about the problem. Another feature of our problem is that information is expensive. Given

these facts, the solutions that we seek for these problems will be approximate. Hence there is a notion of error in the solution. We generally require that the problem be solved within an error threshold ϵ . The ϵ -complexity of a problem is thus defined as the minimal cost among all algorithms which solve the problem with error at most ϵ .

The cost and errors can be variously defined, e.g. *worst case* cost and error, or *average case* cost and error. The error might be absolute or relative. In Traub et. al. [68], these terms are rigorously defined. The computational complexity of problems where only limited information about the solution domain is available is called *information-based complexity*. This is a relatively new research field. In this section, some definitions and relevant results from this subject are stated. The idea is to give the reader a flavor of the methods for evaluating optimality of the algorithms and the difficulty in solving some of the problems that will be dealt with in this thesis.

Let F be a set and G be a normed linear space over the field of real or complex numbers. The problems are defined in terms of an operator $S : F \rightarrow G$ called the solution operator. Elements f from F are called problem elements and elements $S(f)$ are called solution elements. For each f in F we wish to compute an approximation to $S(f)$. Let $U(f)$ be the computed approximation. The distance between $U(f)$ and $S(f)$ will be measured by the error criterion, the simplest of which is absolute error $\|S(f) - U(f)\|$. Let $\epsilon \geq 0$. $U(f)$ is an ϵ -approximation of f iff $\|S(f) - U(f)\| \leq \epsilon$. So the goal is to compute elements $U(f)$ such that they are ϵ -approximations for all elements in f from F or on the average. For example, assume $f \in C^r[0, 1]$ and that the derivatives of f are uniformly bounded by 1. Thus:

$$F = \{f \in C^r[0, 1] : |f^{(k)}(t)| \leq 1, k = 0, 1, \dots, r\} \quad (1.2)$$

Let $S(f)$ be the integration operator $S(f) = \int_0^1 f(t)dt$. Hence $G = R$. Then $U(f)$ is an ϵ -approximation iff $U(f)$ is a real number and $\|S(f) - U(f)\| \leq \epsilon$.

To compute an ϵ -approximation, we need some information about f . We can gather partial information about f by computations of the form $L(f)$. For example, for the integration problem, we can compute the function or its derivatives at a certain point, i.e., $L(f) = f^{(i)}(x)$ where $0 \leq i \leq r$ and $x \in [0, 1]$. So a number of different information operators can be computed for f . There is *adaptive* and *non-adaptive* information about f . So information about f is given as the set

$$N(f) = [L_1(f), L_2(f), \dots, L_n(f)] \quad (1.3)$$

A basic assumption is that computing information about f is expensive. Hence there is a cost c associated with each $L(f)$. The total cost of $N(f)$

$$\text{cost}(N, f) \geq cn \quad (1.4)$$

depending on whether the information is non-adaptive or adaptive. Knowing $N(f)$ the approximation $U(f)$ is computed by combining the information to produce an element of G which approximates $S(f)$, by a mapping $\phi : N(F) \rightarrow G$. That is $U(f) = \phi(N(f))$. ϕ is our algorithm for computing the approximation. The total cost of computing the ϵ -approximation is then given as

$$\text{cost}(U, f) = \text{cost}(N, f) + \text{cost}(\phi, N(f)) \quad (1.5)$$

The computational complexity of an approximation U can be considered in the

1. **Worst Case Setting:** The worst case cost of U is defined as

$$\text{cost}(U) = \sup_{f \in F} \text{cost}(U, f) \quad (1.6)$$

2. **Average case setting:** Let μ be a probability measure defined on F . The average case cost of U is defined as

$$\text{cost}(U) = \int_F \text{cost}(U, f) \mu(df) \quad (1.7)$$

The ϵ -complexity of a problem is defined as the minimal cost over all U with error less than ϵ ,

$$\text{comp}(\epsilon) = \inf\{\text{cost}(U) : U \text{ is an } \epsilon - \text{approximation}\} \quad (1.8)$$

With these definitions, many complexity results are given in [68]. Some of the important ones relate to function approximation and optimization of smooth functions. The ϵ -complexity of the approximation problem for smooth nonperiodic functions is given by:

$$\text{comp}(\epsilon; q) = O(c(\frac{q}{\epsilon})^{\eta+1/r_{min}}) \quad \forall \eta > 0$$

Here $f : D = [0, 1]^d \rightarrow R$ is r_j times continuously differentiable in direction j , $j = 1, \dots, d$, and the partial derivatives of $f(x)$ are all zero, when one or more of the components of x is zero, and are bounded above by q for all $x \in D$. $r_{min} = \min_j r_j$. This complexity is almost independent of the dimension d of the function domain. This is because if d increases, so does the smoothness of the function, thus leaving the problem complexity almost invariant.

The following constrained non-linear optimization problem is considered. Let $f = [f_0, f_1, \dots, f_m]$ where $f_j : D \subset R^d \rightarrow R$ is a continuous scalar function. The solution operator is defined by :

$$S(f) = \min\{f_0(x) \mid x \in D, f_j(x) \leq 0, j = 1, \dots, m\}$$

where the permissible information operations consist of evaluations of f and f' at points from D . If F is the class of r -times continuously differentiable functions such that the r th derivative of f_j is uniformly bounded, then the ϵ -complexity is given as

$$\text{comp}(\epsilon) = \Theta(c(\epsilon^{-d/r})) \quad (1.9)$$

which is exponential in d . Hence for even moderate ϵ and large d , the problem is intractable in the worst-case. If however, F is the class of *convex* functions which satisfy a Lipschitz condition with a uniform constant on a bounded convex set D , then

$$\text{comp}(\epsilon) = \Theta(\text{cln}(1/\epsilon)) \quad (1.10)$$

Hence this problem is dimensionally independent except for a constant in the Θ notation which depends polynomially on d . This quantifies the value of convexity as opposed to smoothness for optimization.

These results are important for this research. We know that the optimization problem is nearly intractable in the worst-case in the absence of convexity. Hence we can only hope to get good average behavior from any algorithm. The complexity in the average case setting has yet to be established. Wasilkowski [69] gives the upper bound on the 1-dimensional problem as $\Theta((\sqrt{\ln \epsilon^{-1}}/\epsilon)^{2/(2r+1)})$. However, no results are known for higher dimensions though it is conjectured that the problem might be mildly dependent on d .

1.6 Thesis Outline

The rest of this thesis is organized as follows:

In chapter 2, existing work in high speed circuit characterization and optimization is reviewed, and the background for developing a new methodology is provided.

In chapter 3, the main research goals are identified and the achievements of this work are stated.

In chapter 4, the general methodology for the simulation based characterization and optimization of electrical circuits is presented. New techniques for experimental design, and for employing stochastic models for circuit optimization are presented.

In chapter 5 the methodology for performing global routing and generating wiring rules for interconnect with significant transmission line effects is presented. This methodology helps in identifying global signal paths and bounds on wirelengths to achieve constraints on the electrical performance of the interconnect.

In chapter 6 the software tools developed to support the characterization and optimization methodology, and for global routing and wiring rule generation are presented.

The performance of the characterization technique is established through several case studies in chapter 7. Also, the optimization technique is employed for parametric optimization of several difficult circuit designs. In addition, the global wiring using the proposed methodology is conducted for two MCM designs and a PCB design. Detailed wiring rules are generated for the PCB design.

In chapter 8 the contributions of this thesis work are summarized and future work is listed. The appendices give details of the MCM and PCB design examples.

Chapter 2

Background and Existing Approaches

In the previous chapter, the simulation based design problem was formally defined, and the problem complexity examined. There exists some body of literature which serves as background work for this thesis. In this chapter, the existing approaches, mainly for circuit characterization and optimization are reviewed. In the first section, techniques employed for circuit macromodeling are reviewed. Circuit macromodeling has been extensively employed for statistical circuit design. The important differences between statistical design, and the design problem studied in this thesis are pointed out. The need for investigating different techniques for circuit characterization is emphasized.

2.1 Macromodeling

As stated earlier, circuit macromodeling has been extensively employed for statistical circuit design. The aim of statistical circuit design is to find circuit implementations that are robust to process variations [14]. In this case, the parameters affecting circuit performance are split into *designable parameters*, and *non-designable parameters*. The designable parameters are have a fixed value for a particular design, while the

non-designable parameters are random in nature, with a certain distribution. The aim of statistical circuit design usually is parametric yield maximization, i.e. to maximize the likelihood of a circuit meeting the performance constraints, given the uncertainty in circuit behavior due to the randomness of the non-designable parameters. Inasmuch, the circuit yield, which is the number of circuits that would meet the desired specifications has to be maximized. The circuit yield is an average quantity and has to be evaluated by examining multiple circuits chosen from the given distribution of the non-designable parameters. Each such evaluation has to be done by circuit simulation. Hence it is necessary to build a simplified model of the circuit performance in order to reduce the time taken for yield evaluation and optimization.

One technique considered by several researchers for model building is *response surface modeling*. Response surface models are constructed by fitting a linear or quadratic polynomial to a set of simulated points through linear regression. The number of simulated points is larger than the number of terms in the polynomial. The extra degree of freedom is used to compute the modeling error, or variability of the response surface [13]. In Alvarez et. al.[1], a response surface methodology is employed for VLSI device design. Low [42] employs a similar methodology is used for building macromodels of the IC fabrication process. *Variable screening* is employed to reduce the number of terms in the regression model. Biernacki and Bandler [5] describe an efficient method for quadratic response approximation is suggested.

Another technique employed for generating macromodels is interpolation. The interpolation model passes exactly through all simulation points. Such a technique has been recently proposed by Styblikowski and Aftab[64] using maximally flat quadratic interpolation. In all these interpolation schemes, the number of simulation points is exactly equal to the number of terms in the interpolating polynomial.

There are a number of deficiencies with these techniques that make them

unsuitable for the purpose of designing high speed digital circuits. The response surface methods do not make full use of information obtained from an accurate circuit simulator. This is not a drawback for circuit yield calculation, where only the statistical average is of interest. Moreover, it is possible to directly compute the variability of the response surface, which is useful for yield enhancement algorithms. For digital circuit design, where the variability in circuit performance is not a major concern, the loss of information in fitting models by regression is a serious drawback. Moreover, the circuit responses tend to be highly non-linear and hence response surface methods are likely to be very erroneous. The interpolation techniques suffer from this same drawback. The modeling error is being artificially avoided here by keeping the number of terms in the polynomial and the number of data points the same, under the assumption that the departure of the model from linearity is small [64]. A new philosophy to analysis of simulation results has to be adopted.

2.2 Experimental Design

In addition to finding a suitable approximation for the circuit responses, it is necessary to determine which circuits to simulate to gather sufficient information to approximate the response over the entire design space. The task of choosing points for simulation is termed *experimental design*. Experimental design is usually tied closely to model building, especially for response surface methods. The most commonly used experimental design techniques are:

1. **Random Designs:** This is the simplest design scheme, and is useful for generating large samples from a known distribution. Many variations of random sampling exist, e.g. importance sampling. Random sampling is useful for functions which are cheap to evaluate, and gives high confidence in the generated

aggregate statistics.

2. **Factorial Designs:** In factorial sampling, each of the variables, or *factors* in the experiment are discretized into *levels*. All levels of each factor are combined in a *full factorial design*. For example, if there are 3 levels for n factors, the total number of experimental points is 3^n . Factorial sampling is extensively employed with regression analysis. The number of experiments is of course exponential in the number of factors. To alleviate this problem, *fractional factorial sampling* is employed by systematically removing some of the experimental points in a full design. This sampling plan is very useful when the model is assumed to be linear and there are a large number of factors in the design.
3. **Central Composite Designs:** Central composite designs are an extension of factorial sampling. Each factor is divided into five levels, $-1, -\alpha, 0, \alpha, 1$. There are two subplans, a fractional factorial plan on the $-1, +1$ levels of each factor, and a *star design*, which consists of the center point and n pairs of axial points, one for each factor whose level is set at $+\alpha$ and $-\alpha$ respectively, while keeping all other factor levels at 0. This is illustrated in figure 2.1. This plan help in determining the quadratic terms in a regression polynomial and can be used for variable screening [42].
4. **Latin Hypercube Sampling:** Latin Hypercube Sampling (LHS) [45] is a special form of random sampling. In this method, the range of each variable is stratified into N strata, where N is the total number of samples to be drawn. One sample is drawn from each stratum of each variable. Let one such sample be $x_{ij}, i = 1, \dots, N$, where j is the variable. The samples for each $j, j = 1, \dots, d$, where d is the total number of variables, are randomly combined to form the Latin Hypercube Sample. Figure 2.2 shows a 4 and 9

sample LHS for two variables. A unique advantage of LHS is that the sample projects uniformly on each variable space. Hence there is no need to perform variable screening. Several extensions to LHS have been proposed. Iman et. al. [26] have proposed designs that preserve any correlations that might be known among input parameters. Tang et. al. [65] have proposed designs that satisfy a maximin distance criterion. Fortran code for generating LHS designs from several distributions is available [27].

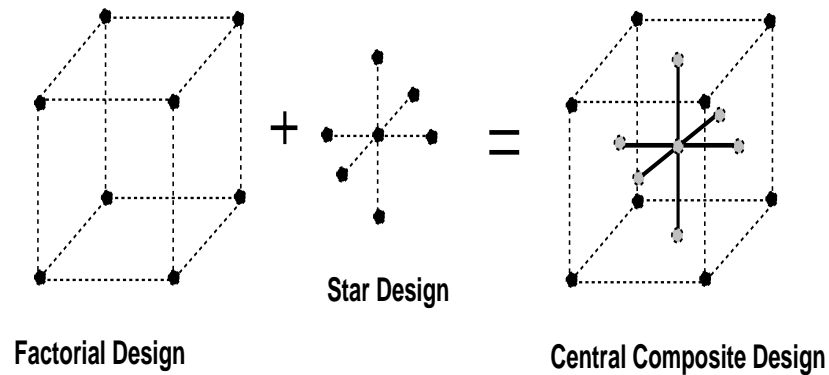


Figure 2.1: Central Composite Design

Of the above sampling schemes, LHS is the most attractive for high speed digital designs. Firstly, the form of the response function is not supposed. Also, the design points are uniformly distributed, hence each variable is represented as best as it can be in the sample. The design is very easy to generate, and any size designs can be generated.

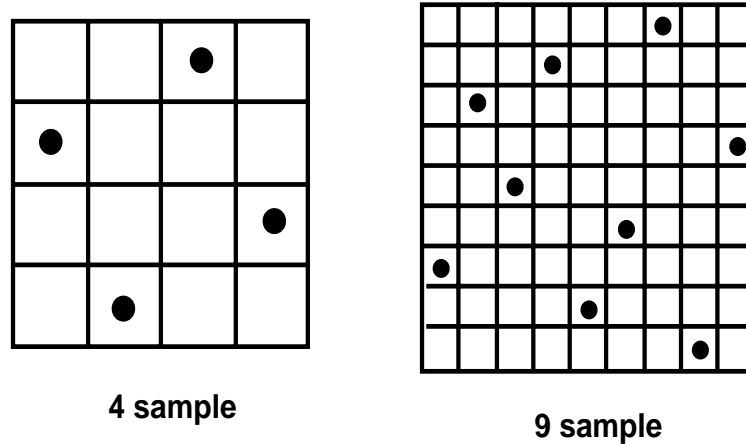


Figure 2.2: Latin Hypercube Designs

2.3 Circuit Optimization

The design of electronic circuits for optimal performance has been the subject of extensive research. Several review papers [2], [7] and [41] have been published on this subject. The basic assumption in all this work is that the topology of the circuit is fixed, and only the component values are variable. The optimization problem may have multiple objectives and several constraints. Delight.Spice [48] is a general purpose optimization package coupled to the SPICE circuit simulator. As mentioned in chapter 1, there are several difficulties in coupling a circuit simulator to a non-linear optimization tool. The greatest single difficulty is caused by the cost of objective evaluation. Most optimization techniques do not directly address the complexity of objective evaluation. Also, only locally optimal solutions are generated. As is obvious from the discussion in section 1.5.1, it is impossible to find globally optimal solutions in the general case. However, there is a tremendous research effort directed to inventing global optimization procedures. In this section, some of these procedures

are reviewed, with special emphasis on those applied towards circuit optimization.

2.3.1 Global Optimization Approaches

Several global optimization approaches have been published. For good reviews see Zilinskas[67], Kan[31] and Horst[25]. The most successful methods for global optimization seem to be those that incorporate random search in some form. These methods are termed *stochastic methods*. Some interesting and well analyzed stochastic algorithms are presented by Kan in [31], [32]. The algorithms assumed that the function being optimized, though it may be non-convex and have several local minima is twice continuously differentiable. Furthermore, there is a strictly descent algorithm P , which can be started at any point in the input-variable space S , to lead to a stationary point. The global framework of these algorithms is as follows:

1. N points are drawn from a uniform distribution over S and the function evaluated at these points. These points are added to the (initially empty) sample.
2. A procedure selects a (possibly empty) subset of the enlarged sample and P is applied to each of its elements. The stationary points thus found are added to a (initially empty) set X^* .
3. A stopping rule decides whether to return to Step 1 or to stop. If the method is stopped then the element of X^* with smallest function value is the candidate solution.

The efficiency of these algorithms is measured by the number of *local searches conducted* and not by the *number of function evaluations*. The method is said to fail if the local search is started, although the resulting minimum is already known, or if no local search is started in a component of the level set of function corresponding to

the largest function value in the reduced set, even though this component contains sample points.

Another particularly interesting class of algorithms are those based on some *statistical model of the objective function*. These algorithms model the objective function stochastically. Then, given some measurements, it is possible to calculate the conditional distribution of the stochastic function for untried values. In the general case, a stochastic function $\zeta(x)$ is defined by a family of multidimensional probability distributions $F_{x_1, \dots, x_k}(y_1, \dots, y_k) = P(\zeta(x_i) < y_i, i = 1, \dots, k)$. For example, if the multidimensional distribution is Gaussian, then the stochastic function is defined by the a priori average function $\mu(x)$ and covariance $\sigma(x_i, x_j)$. If some values of the stochastic function are known, e.g. $\zeta(x_i) = y_i, i = 1, \dots, k$ then the conditional distribution of $\zeta(x)$ is again Gaussian with the mean value

$$m_k(x \mid \zeta(x_i) = y_i, i = 1, \dots, k) = \mu(x) + (\sigma(x, x_1), \dots, \sigma(x, x_k)) \Sigma_k^{-1} (y_1 - \mu(x_1), \dots, y_k - \mu(x_k))^T$$

and the variance

$$s_k^2(x \mid \zeta(x_i) = y_i, i = 1, \dots, k) = \sigma(x, x) - (\sigma(x, x_1), \dots, \sigma(x, x_k)) \Sigma_k^{-1} (\sigma(x, x_1), \dots, \sigma(x, x_k))^T$$

Based on this conditional distribution, the choice of the next point can be made. It seems more likely to find a point with small function value where $m_k(x \mid \cdot)$ is small. However, large values of $s_k^2(x \mid \cdot)$ indicate regions of great uncertainty, i.e. regions where function values can differ greatly from the conditional mean. Hence a rational choice has to be made. Several optimality criteria can be proposed here. For example:

$$P_k(x) = P(\zeta(x) < z_{ok} \mid \zeta(x_i) = y_i, i = 1, \dots, k)$$

where z_{ok} is some number smaller than the smallest observed value so far. This is the conditional probability of achieving a value smaller than the level z_{ok} . So stochastic optimization algorithms can be characterized as follows:

1. Choose the stochastic function to be used as a model.
2. Define the criterion of rationality for the current step.
3. Construct an algorithm for optimizing this criterion.

In the next section some optimization algorithms that fit the above description are reviewed.

2.3.2 Algorithms based on Statistical Modeling

Several algorithms for global optimization using a stochastic model function have been investigated. In Groch et. al. [22], the model in the multidimensional case is not really a stochastic function. Instead, the posterior mean and variance of the one dimensional Weiner process are generalized. The choice of the next point of evaluation is made by minimizing:

$$w_i^{k+1}(x) = m_i^k(x) - c\sigma_i^k(x)$$

where the experimental region is divided by disjoint simplicial subregions. The addition of each new point causes the experimental region to be further sub-divided. The attractive feature of this method is that it is an intuitive generalization of the scalar method and the auxiliary computations required to find the next best point are quite simple. Only an approximation to the global optimum can be located in this manner.

In Adachi[43], as in Schagen[56], the model function is a stationary stochastic process. Then, an interpolating function is fitted to the available data. This

interpolating function is exact at the data points and its derivatives are easy to compute. The variance of approximating the true response by this function is also easy to compute. The next point is chosen by minimizing the interpolating function starting from the smallest data point found thus far, subject to a constraint on the coefficient of variation. The optimization procedure is iterative, each iteration giving different local optima of the objective function. The auxiliary computations are extensive, namely the use of a local optimizer to find the minimum of the interpolating function.

In Schagen[57] the model is a stationary stochastic function. In addition to the interpolating function, a repulsive function is defined, which repels the new data point from the existing points and the boundary of the acceptable region. A composite function is defined as a weighted sum of these two functions. This composite function is optimized to determine the position of a new data point. A procedure is given to find the starting point of this local optimization.

In review, the main goal of the optimization process is to minimize the number of computations of the real objective function, which usually involves a full circuit simulation. Hence the chosen method should be such that each objective calculation improves the estimate of the global optimum. Auxiliary computations can be intensive, as long as they improve the efficiency of the algorithm. So the thrust of the investigation should be in capturing the objective function as closely as possible with the given data. Additional data should only be generated after full use has been made of the existing one. The approach of Adachi [43] obviously does not do so in the general case, since only one local optimization is carried out on the approximate function, before additional data is generated. Our goal should be to find absolutely the most promising points each time, either as candidates for the global optimum, or as information bearers for it. Thus the algorithm should employ

a model and a choice criterion to maximize this likelihood. Exact convergence of the overall algorithm is not a major concern. Maximizing the likelihood of improvement is most important. In section 4.4 a new methodology based on the P-Algorithm [74] is described. This a powerful interactive optimization methodology, which requires very few actual objective function evaluations.

2.4 Summary

It is clear that high speed circuit characterization requires the suitable choice of a response macromodel, and a methodology for experimental design, to capture the circuit performance accurately and efficiently. In chapter 4 a new technique employing data interpolants for capturing response, and LHS for experimental design is presented, for the characterization of circuit performance. Performance optimization would require the development of a global optimization strategy which requires few objective function evaluations. Stochastic modeling of circuit responses provides a promising method for capturing circuit performance for optimization. In chapter 4 a powerful interactive optimization methodology that employs stochastic models is presented.

Chapter 3

Thesis Statement

3.1 Research Goals

The main goal of this research was to devise efficient methods for simulation based characterization and optimization of high speed circuits. Efficiency is measured by the number of simulations required to achieve a certain level of confidence, or required accuracy, in the characterization.

The following were the general goals of the presented research work:

1. Establishing a general methodology for characterizing arbitrary electrical responses of a system based on full circuit simulation.
2. Establishing an efficient methodology for optimizing a system to achieve bounds on electrical responses evaluated through an expensive circuit simulation.
3. Establishing a general methodology for generating design rules, that are supplied to automatic layout synthesis programs as constraints in order to achieve performance goals.

The following were the achievements of this research, consistent with the

objectives outlined above:

1. Development and implementation of a new heuristic methodology for sequentially sampling a design space to reduce predictive error. This methodology was employed for generating accurate characterizations of high speed interconnects on Multi-chip modules in reasonably few simulations. The characterizations were employed for fast evaluation of global routing trees in MCM and PCB layouts.
2. Implementation of a sequential approach for optimization of arbitrary smooth functions. This procedure was employed for solving several transistor sizing problems in high performance CMOS VLSI circuits, and determining optimal termination scheme for a high speed data bus.
3. Development of optimal and heuristic methods for generating performance constrained routing trees.
4. Development of a strategy for feasibility estimation of a layout through Global Routing.
5. Development of a strategy for employing global routing results to generate bounds on net-lengths suitable for use in a performance driven layout synthesis tool.

Chapter 4

Characterization and Optimization Methodology

In this chapter, the methodology for characterization and optimization of high speed circuits will be described. To this end, the relationship between certain *physical design variables* and certain *electrical performances* have to be investigated. Characterization is based on a response surface methodology. The attempt is to find a suitable approximate the true response using a simpler *predictor function*, such the response can be predicted over a certain *design space* within specified accuracy. The number of true response evaluations necessary to formulate the predictor function has to be kept as small as possible. The form of the predictor function should be such that it is much faster to evaluate than the true response. Optimization is based on a similar premise. The objective here, though, is to be able to locate the best value of the true response over a certain design space. Again, the number of true response evaluations has to be kept as small as possible.

4.1 Characterization Methodology

Formally, the objective is as follows:

Consider a general electrical network which obeys a set of nonlinear differential-algebraic equations of the form:

$$G(\zeta, x, t) = 0, \quad (4.1)$$

where ζ is a vector of instantaneous node voltages and currents, x is a set of *design parameters*, and t is time. The parameters specified by x depends on the level of abstraction used in the problem specification (e.g. various inductances, capacitances etc. in the circuit model, in a circuit level representation)

Let ϕ represent the set of performance parameters for the network. The exact x to ϕ mapping can be obtained only by running a computer simulation that solves the system of equations G numerically. The objective is to obtain a predictor function $\phi^*(x)$, which is relatively much cheaper to evaluate than a full circuit simulation, and is a good approximation of $\phi(x)$ over a range of x which is referred to as the *design space*. $\phi^*(x)$ is obtained by conducting a computer experiment in which $\phi(x)$ is evaluated at n sample sites $\{x_1, \dots, x_n\}$ using the computer simulation. $\phi^*(x)$ must satisfy the following restrictions:

1. Predictable accuracy:

$$|\phi^*(x) - \phi(x)| \leq \epsilon, \quad (4.2)$$

for each component of ϕ , where ϵ is some scalar error measure, over the design space.

2. Unbiasedness: If the value of ϕ is known at a certain point x^* , then ϕ^* should have the same value at x^* , i.e., $\phi(x^*) = \phi^*(x^*) \forall x^* \in \{x_1, \dots, x_n\}$.

Hence the objective of the experimental design is to choose a suitable predictor function $\phi^*(x)$ and n sample sites $\{x_1, \dots, x_n\}$ such that the unbiasedness conditions is

satisfied and the error of prediction is minimized. On first glance, the unbiasedness condition might appear overtly restrictive. However, there are several predictor functions, e.g. BLUP in [53], Moving Least Square Interpolant [34] etc. that easily achieve this condition. The unbiasedness condition helps us formulate the cross-validation error-measure [70]. It also accounts for "outliers" in the data, and helps in designing experiments for fully conservative designs where the "outliers" are of great concern because they represent strong non-linearities in the response, and not "noisy" observations, as is the case for physical experiments.

Design for computer experiments has been the subject of some work recently by Sacks et. al.[53], [54], [70]. The main issue addressed in these papers is that of designing a computer experiment to investigate a response function Y by running a simulation code at various choices of input factors x . The goal of the experiment is to form a good approximation of the true response Y , i.e. approximate Y by a polynomial function so that a reasonable value of Y at an untried input can be predicted by interpolation. Since this response function is only approximate, the true response is modeled by Response = Simple model + Departure. The systematic departure of the true response from the simple model is described by a stochastic process, that is

$$Y(x) = \sum_{j=1}^k \beta_j f_j(x) + Z(x) \quad (4.3)$$

where β_j are scalars, $f_j(x)$ are polynomial terms and $Z(x)$ is a stochastic model of the departure of the true response from the polynomial. with zero mean and covariance $V(w, y)$ between any pair of processes $Z(w)$ and $Z(y)$. The covariance is given as

$$V(w, y) = \sigma^2 R(w, y), \quad (4.4)$$

where σ^2 is the variance and $R(w, y)$ is the correlation function. $Z(x)$ represents the departure of the response from the polynomial model given by the first term of equation 4.3. The form of this function is, of course, unknown. For smooth responses, values of Z at points close to each other will be highly correlated. The authors, thus assume a covariance structure for Z that will reflect this property, namely:

$$V(w, y) = \sigma_Z^2 \exp(-\theta \sum_{j=1}^d (w_j - y_j)^2)$$

between two points w and y in the d dimensional input space. The parameter θ is the most critical factor in this correlation structure. Prediction by interpolation is hard for large and small values of θ . Once it is specified, predictions of Y for unknown x can be made from data $Y(s_1), \dots, Y(s_n)$ obtained from a set of design points $S = (s_1, \dots, s_n)$. The predictor $Y^*(x)$ (also called the Best Linear Unbiased Predictor) is the sum of a generalized least squares estimate of the first term in Equation 4.3, using the sampled responses, and a smoothing term, expressed as an interpolant of the residuals at the sampled points (see appendix B. This smoothing term can also be seen as the posterior mean of the random process $Z(x)$. The uncertainty in prediction is described by the Mean Squared Error (MSE) which is essentially the posterior variance of $Z(x)$.

The papers then deal with the issue of choosing the best design, i.e. values for s_1, \dots, s_n . This is done by minimizing the *Integrated mean square criterion* which is defined as

$$J_\theta(S, Y') = \frac{1}{\sigma_Z^2} \int E_\theta(Y'(x) - Y(x))^2 dx \quad (4.5)$$

A crucial assumption in this process is that of assuming a value for θ . Then by formulating the integral, an optimization routine is executed to minimize the equation above as a function of the $n \times d$ design-point coordinates. Other design

criterion are possible such as that of minimizing the maximum value of the MSE over the design space. The main point here is that the optimization of the design strategy is computationally quite expensive and this cost must be weighed against that of performing some extra simulations to get more information.

The paper also discusses methods for choosing a robust estimate of θ when its true value is not known. This is critical since the choice of θ will greatly affect the experimental design. For this, the authors compare several assumed values of θ in estimating $J_\theta(S, Y')$ at several choices of true θ . The chosen value of θ is that which gives best approximation for the whole range of θ values (in a minimax sense).

Sacks[54] discusses some sequential design methods also. These designs adapt to the information already gathered, both about the regression model and the correlation structure. However, the theoretical treatment of such designs is considered to be quite difficult. One sequential design scheme used in this paper is as follows : The experimental region is divided into a number of subregions. A first experiment is performed using the design method stated above. After that, a new point is added to the box which has the maximum contribution to the IMSE. This point is chosen such that it most reduces the contribution to the IMSE in that box.

The issues discussed in these papers are obviously very relevant. However, there are several reasons why these approaches are not directly applicable to our problem domain:

1. Firstly, the task of generating optimal designs, though amenable to automation, will prove to be computationally too expensive for our purposes. Since each simulation run is relatively (compared to the minutes of CRAY time required by the author's experiments) cheap for us, we should resort to cheaper, approximate designs, with more number of simulation runs.

2. The correlation structure will not be known to us apriori. So the experiment should be designed to infer the structure from the results. Again, we cannot afford the process of comparing the efficiency of several different values of θ . Hence a sequential strategy is more suitable for us.

Latin Hypercube based designs are discussed briefly by the authors, though theoretical treatment is not given. These designs have a tendency to fill out the design space. Hence they seem to be quite suitable as the first design strategy. The follow-on designs must be based on the predictive ability of the earlier samples. The "Box-based" method is fairly easy to implement, and a similar method is described here. The stochastic model will not be directly used for design or prediction. The reasons for this are detailed below. Prediction is done by a data interpolant, which is easier to evaluate than the BLUP. No measure of uncertainty in prediction (like the MSE) is available with this predictor. Hence the accuracy of the predictor is measured by *cross validation*.

4.2 Sequential Experimental Design

For the responses of interest for this work, there is very little prior information available about the nature of the responses. In this scenario, sequential sampling is the most suitable. With sequential sampling, the sampling can be repeated to reduce predictive error by further sampling in the regions where the error seems to be concentrated. The approach here is to keep the same sampling strategy during each step of the experimentation. Only the extent of the design variables x , (subsequently called *experimental region*) change from one step to the next. Since at each step, we try to characterize the entire experimental region, an experiment design with *space filling* property, i.e. one which distributes sites uniformly over the experimental re-

gion, is required. Latin Hypercube Sampling (LHS) is very suitable for this purpose. Another advantage of using LHS is that it is well adapted for situations where some of the parameters have a statistical distribution, or a certain correlation structure.

4.2.1 Characterization of Error in Prediction

After each step in the sequential experiment, the data is analyzed to determine the error in prediction at untried input values. The next experiment is defined in subregions where the error is largest. This is a crucial step in the characterization process. Usually, this is done by computing some global error statistics. This however, indicates *when* to resample, but with no indication of *where* to sample more points. One way of performing error characterization is to generate a small random sample of responses and compare them with the values computed by the predictor function. This method only characterizes error at the newly evaluated sites and takes a long time to perform.

Our method of obtaining global error measures is to characterize the error at each of the points simulated thus far. For this, the response value at each point is computed by the predictor function, assuming that the true response value at this point is not known. i.e.,

$$\begin{array}{l} \forall x_i, \quad i = 1 \dots n, \\ \text{Compute } | \phi^*(x_i) - \phi(x_i) |, \end{array} \quad (4.6)$$

where $\phi^*(x_i)$ is computed based on $\phi(x_j)$'s, $j = 1 \dots n, j \neq i$.

This error measure is termed *cross-validation* [70]. The merit of this strategy is that it gives desired error of prediction at each simulated point, without being biased by the value of the response at that point. This method is all the more attractive since our predictor function is *local* in nature, as described below. Also, since

the simulated points are scattered uniformly over the experimental region, this gives a good error characterization over the entire experimental region.

4.2.2 Predictor Function

According to the unbiasedness condition stated above, the predictor function should be exact at the sampled points. The usual least square error predictor, in general, fails to do this. The predictor function derived from the stochastic model of Sacks[54] is a data interpolant. Several other forms of multi-dimensional data interpolation are possible, one of which is Moving Least Square Interpolation [34].

With this method, the response model is given by

$$\phi^*(x) = \sum_{j=1}^n a_j b_j(x), \quad (4.7)$$

where $b_1(x), \dots, b_n(x)$ are n linearly independent polynomials in x . These functions are supplied by the user. The unbiasedness condition is satisfied through proper choice of the interpolants. Whenever $\phi^*(x)$ is evaluated, moving least squares are used to calculate the a_j s. resulting in an unbiased estimate of the response at the point x . The a_j 's are calculated so that a weighted sum of the error of prediction at all sample points is minimized. This is achieved by solving the system of equations:

$$E_z(\phi^*) = \sum_{i=1}^N w_i(x)(\phi^*(x) - \phi(x_i))^2, \quad (4.8)$$

where $w_i(x)$ is the weight assigned to the error at x_i , and x_1, \dots, x_N are the N distinct sample points. The error $E_z(\phi^*)$ is minimized by solving the system of equations:

$$BW(z)B^T a(x) = BW(x)\phi \quad (4.9)$$

where B is an $n \times N$ matrix whose j th row is $[b_j(x_1), \dots, b_j(x_n)]$, ϕ is the $N \times 1$ vector of responses at the sample points, and $W(x)$ is a diagonal weighting matrix,

with elements $w_{ii}(x)$ being the weights assigned to the error at x_i 's.

$$W(x) = \text{diag}[w_1(x), \dots, w_N(x)] \quad (4.10)$$

Note that when $W(x)$ is the identity matrix, the normal equations are the same as those for the usual least square minimization. The weighting function chosen was of the form

$$w_i(x) = w(d(x, x_i)). \quad (4.11)$$

Here d is the Euclidean distance between two points. In order to achieve exact interpolation at the sampled points, the function w should go to infinity at the sampled points x_i 's. Functions of the form

$$x_i(z) = e^{-\alpha \|x - x_i\|^2} / (\|x - x_i\|^2) \quad (4.12)$$

have this behavior. These functions also attenuate rapidly and hence minimize the influence of remote data values (i.e. $\phi^*(x)$ is *local* in nature), while smoothing the response. In practice, the singularity at x_i s can be removed by replacing the $\|x - x_i\|^2$ in the denominator by $\|x - x_i\|^2 + \epsilon$ where $\epsilon > 0$ is a very small constant.

Prediction by MLSI has several advantages. It is cheaper to evaluate than the BLUP in [54], since it involves an $n \times n$ matrix inversion, instead of the $N \times N$ covariance matrix for BLUP computation. Also, there is no need to formulate a correlation structure, although there are some alternatives in choosing the weighting function. In Sacks et. al.[70], the parameters in the correlation structure are estimated to best "fit" the data through likelihood. This can be a very expensive operation, sometimes giving marginal increase in the quality of prediction. The approach here, however, *is to generate more samples in regions where the predictor function has poor fit to the data*. No measure of the uncertainty (like the MSE) is available for this interpolant. Hence cross-validation is used to characterize the error in prediction. Another advantage of MLSI over the BLUP is that it gives a local

polynomial approximation of the response. This property turns out to be extremely useful for generating *design rules*, i.e. determining which part of the design space has response values lesser or greater than a certain threshold.

4.3 Implementation

In this section, we describe the implementation of the ideas outlined in the previous section. A software module, called the Study Generator, has been developed with these algorithms. Figure 4.1 shows a block description of the Study Generator. From the user input the variables that form the dimensions of the design space are specified along with the constraints that define the design space to be characterized. LHS is performed within this space and the error is evaluated as describe in the previous section. The error criterion is used to determine the sub-regions that need further sampling. The Study Generator uses MetaSim [19] for automatic specification of the simulations and extraction of electrical responses[58]. Details of the software are described in chapter 6.

4.3.1 Identifying the Design Variables and Initial Experimental Region

The design variables and their ranges are user specified. In general, the ranges of design variables are interrelated. For example, several interconnect lengths in a layout, though independent variables, are constrained together by the size of the chip or board. Hence the required ranges of the variables are specified by linear inequalities. These inequalities represent closed half spaces, in the Euclidean space of these variables. The design space, is the closure of the polytope which represents the intersection of these half-spaces. The initial experimental region is specified as the smallest

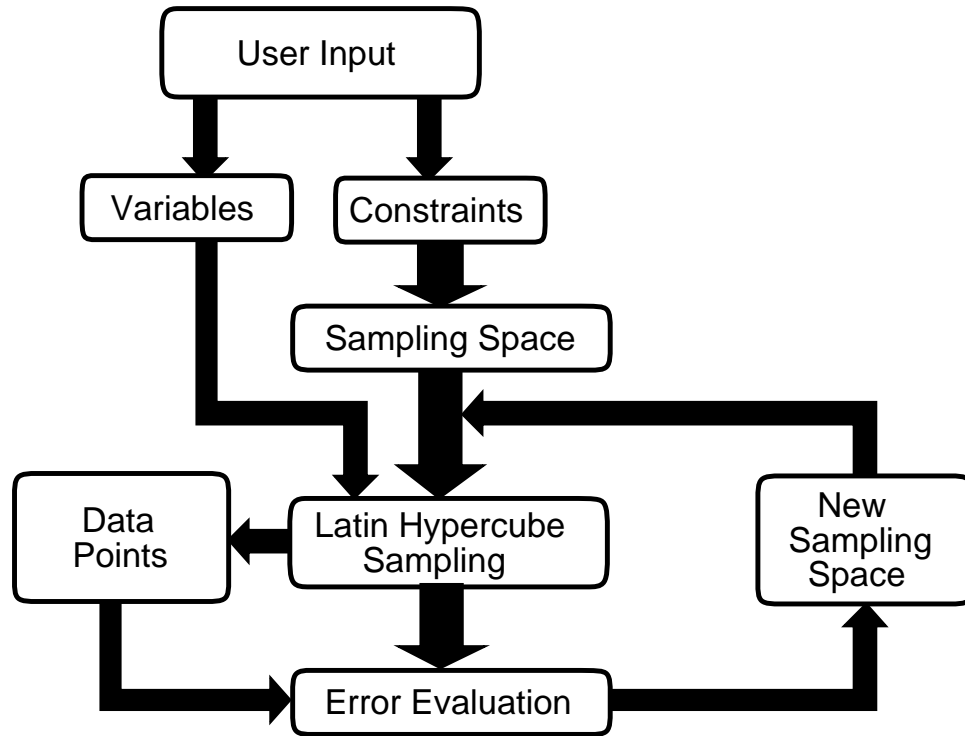


Figure 4.1: Study Generator

hyperrectangular region containing this polytope. To determine this hyperrectangle, the extreme vertices of the polytope along each independent axis have to be found. This can be done by linear programming. Suppose that the design space is defined by k inequalities:

$$\sum_{i=1}^d a_i x_i \leq b_i \quad i = 1, \dots, k \quad (4.13)$$

The initial design space is specified by the upper and lower bounds x_{iu} and x_{il} respectively, on each design variable x_i . The bounds can be calculated as follows.

$$\begin{aligned} & \text{Maximize } x_i \\ & \text{subject to } \sum_{j=1}^d a_j x_j \leq b_j \quad j = 1, \dots, k \end{aligned}$$

gives x_{iu} , and

$$\begin{aligned} & \text{Minimize } x_i \\ & \text{subject to } \sum_{j=1}^d a_j x_j \leq b_j \quad j = 1, \dots, k \end{aligned}$$

gives x_{il} .

LHS is used to determine sample sites in this region. However, before actually simulating the circuit at a sample point, it is verified to see if it also lies in the interior of the polytope. In order to avoid a low sample count as a result of rejecting too many points, a Monte-Carlo evaluation of the volume of the polytope is made. Extra samples are drawn in the LHS to reflect the volumetric ratio of the polytope and the experimental region. This strategy helps in giving a well distributed sample over the polytope with a very tractable sampling scheme.

4.3.2 Identifying the next Experimental Region

In section 7.6, *cross-validation* is suggested as the method for estimating the predictive error. Error is evaluated at every point simulated thus far, as given by equation 7.12. If this error is greater than a certain threshold, it implies that the neighboring points of x_i , do not interpolate well, either because of a local large non-linearity, or sparsity of points in its vicinity. In either case, it is desirable to sample more points in the neighborhood of x_i . The neighborhood of x_i is defined as a ball of radius which is half the minimum scaled distance between x_i and all the other design points, i.e.,

$$r(i) = 0.5 \star \min_{j=1, \dots, N, j \neq i} \|x_i - x_j\|_2. \quad (4.14)$$

The half minimum distance criterion is used to eliminate overlap between the neighborhoods of adjacent points. Each component of x_i is scaled by the length of the

original experimental region along that direction. The intersection of the largest hypercube that fits inside the intersection of this ball and the polytope representing the design space, is the experimental region. Again a sample is drawn from the hypercubic region using LHS. As before, each sampled point is checked to ensure that it lies in the design space. This process is repeated for all the sample points where the error measure exceeds the user specified threshold.

To obtain more uniform designs, a slightly different approach is adopted. If the initial design space is hyperrectangular, then it can be efficiently partitioned to evaluate the global accuracy of prediction. the dual goal to be satisfied here is that the distribution of sampled points should be sufficiently uniform to ensure global accuracy, and also concentrated in the region where prediction is inaccurate due to strong non-linearities. To accomplish this, the initial design space is divided into smaller hyperrectangles after the first stage of sampling. The number of hyperrectangles is such that each hyperrectangle should have at least k points if the design was perfectly uniform, where k is a user specified constant. Then error estimation is performed as before and the average error in each hyperrectangle is found. If certain hyperrectangles contain less than k samples, extra sampling is done in these to ensure uniformity. Further sampling capability is divided between all hyperrectangles based on the average error in each hyperrectangle. In effect, this strategy compensates for the non-uniformity of LHS in addition to performing error based sampling. This is similar to Sacks' [54] approach where sample points are added to regions with largest IMSE. Note that the sampling approach is simple, easy to implement, and adds very little overhead. This is in contrast to Sacks' approach where considerable time is spent in optimizing the experimental design. This is because our applications are such that sample evaluations are not prohibitively expensive.

4.3.3 Multiple Response Evaluation

When multiple responses are considered, the error measure is difficult to define, especially since one response might follow the model fairly accurately, and another might not. Young et.al. [73] have suggested using desirability functions for multiple response optimization. A similar function can be defined for error characterization.

Suppose that ϵ_r is the acceptable error for performance ϕ_r . Then for each x_i :

$$e_r(i) \equiv \begin{cases} \left(\frac{|\phi_r(x_i) - \phi_r^*(x_i)|}{\epsilon_r} \right)^{s_r} & |\phi_r(x_i) - \phi_r^*(x_i)| \leq \epsilon_r \\ 1 & \text{otherwise} \end{cases}$$

where $0 < s_r \leq 1$. The cumulative error at x_i is given by:

$$e(i) = [e_1(i), \dots, e_q(i)]^{\frac{1}{q}}$$

Where q is the number of responses considered. s_r can be chosen appropriately for giving greater importance to error in a particular response.

4.3.4 Stopping Rules

In order to stop the iterations in the sequential experimental design strategy, some appropriate stopping criterion must be devised. Ideally, a stop criterion would be one based on the cross validation error measure. However, the sequential sampling technique is such that it concentrates more points in the region where the unpredictability of the response is the most. Thus, if all sampled points are used for error characterization, the cumulative error, might improve very slowly over successive iterations. One possibility is to use the first sample only for error estimation. Since this sample is scattered fairly uniformly, the error estimate should reflect the global prediction accuracy of the total sample. From the results shown in section 1.5.1 it is obvious that we cannot hope to achieve a desired accuracy in prediction using any

predictor function with a reasonable number of samples. However, our requirements for the characterization should not be so stringent. There are several reasons for this:

1. Even though computer experiments are largely accurate, the model is not. The fabricated circuit's performance can differ significantly enough from the simulation results, due to incompleteness of the model and processing uncertainties.
2. Usually the designers goal is to achieve a certain bound on the performance. Hence an inaccurate prediction, if it does not lead the designer to falsely assume that the bound is met, when in fact it is not, is harmless. Hence the prediction accuracy is most required only close to the performance bound, which forms a small part of the design space. The reason such a bound was not considered in the problem formulation above, though, is that the same characterization is employed for several designs, each of them having different bounds. Hence good prediction on average is quite acceptable.
3. At design time, the values of the physical design variables are not precisely known. This is especially true for one application considered in this work, namely, global routing. In global routing, net lengths are very approximately determined. Hence as long as the predictor helps the designer in discriminating between good and bad designs, the absolute accuracy may be compromised.
4. After physical design is completed, the performance of the layout is always verified through a process of circuit extraction and simulation. As long as the design has a few errors, they can be easily fixed in a redesign. Here is the strongest case for a characterization that has good average behavior, but is much cheaper to evaluate compared to circuit simulation. Using a characterization at design time, drastically reduces the time for the first design. The

few modifications that may be needed because of inaccuracy in the characterizations are a very small price to pay compared to the time saved in the first design iteration.

In chapter 7 many example characterizations are shown. These examples help establish the properties of the sampling scheme and the predictor function. Several other characterizations are employed for global routing for boards. In essence, this is the real test for the characterization methodology. The global routing results show that the characterizations really are accurate enough for design purposes.

4.4 Optimization Methodology

Consider the following unconstrained optimization problem:

$$\min_x \zeta(x) \quad x \in A \subset R^d \tag{4.15}$$

where x is a d dimensional vector, A is a finite subset of R^d . $\zeta(x)$ is the objective function whose value at any $x \in A$ can be determined only through an expensive simulation. Besides this, there is very little information about the objective function. Suppose the objective function is perceived to be continuous and “smooth”, but not unimodal. Convexity of the function cannot be assumed. Hence from the results in sections 1.5.1 it is apparent that the problem is going to be quite intractable in the worst case. However, we would be satisfied with good solutions to the problem. In most designs, it is only necessary to find designs that meet certain performance requirements. Absolute optimality of the performance is not necessary. Given the fact that the function is “smooth”, and the information about the function is expensive, it seems reasonable to approximate the $\zeta(x)$ with one that is simpler to evaluate. The optimization approach should be adaptive, i.e., the optimization methodology

should be such that as more information becomes available about the function $\zeta(x)$, the approximation of the optimal point should become more accurate. Also the optimization method should narrow the domain that has to be searched for an optimal solution in each iteration. The information operator for this problem is going to be only the value of $\zeta(x)$ for a given x . No explicit gradient information would be available. From the methods presented in the previous section for characterization, it is quite clear that an approach based on stochastic modeling would be quite suitable for optimization also. A sequential strategy has to be employed. The goal here is quite different though. We are not interested in reducing the global uncertainty in prediction, but only in regions where the function has a small value. Now $\zeta(x)$ may have multiple local optima. Hence a purely descent based algorithm would not be suitable, as suggested in Adachi[43].

We know that with a stochastic model, and a set of “measurements” on the objective function $\zeta(x)$, prediction of the value of $\zeta(x)$ at untried points can be made using the conditional distribution of $\phi(x)$. This prediction is only probabilistic, i.e., at each x , there is a probability distribution associated with the possible values for $\zeta(x)$, specified by the mean and variance given in equations B.4 and B.6. Hence a search strategy for points of small $\zeta(x)$ should be formulated, based on the conditional distribution. It seems more likely to find a point with small function value where $m_k(x | \cdot)$ is small. However, large values of $s_k^2(x | \cdot)$ indicate regions of great uncertainty, i.e. regions where function values can differ greatly from the conditional mean. Hence a rational choice has to be discriminate between points of small mean but large variance or points of small variance but somewhat larger mean. One such algorithm is given by Zilinskas [67], called the *P-Algorithm*. This algorithm, and its properties are described in detail in the next section.

4.4.1 P-Algorithm

The P-algorithm was developed and characterized by Zilinskas in [74], [75]. It is an iterative procedure. At each iteration, a new observation point x_{k+1} is chosen that has the highest probability of $\phi(x_{k+1})$ being smaller than y_{ok} which is some chosen value smaller than the mean value of ϕ at each point in A . i.e.,

$$x_{k+1} = \text{Arg max}_{x \in A} P_x(y_{ok}) \quad (4.16)$$

is chosen as the next observation point where, y_{ok} is some value less than $\min_{x \in A} m_k(x | x_i, \zeta(x_i), i = 1, \dots, k)$, and

$$P_x(y_{ok}) = \text{Probability}(\phi(x) \leq y_{ok}) . \quad (4.17)$$

Based on rather intuitive axioms, it is shown in [67] that $\phi(x)$ can be assumed to be a Gaussian random variable whose conditional mean $m_k(x)$ and variance $s_k(x)$ is given as:

$$m_k(x | x_i, \zeta(x_i), i = 1, \dots, k) = \sum_{i=1}^k w_i^k \zeta(x_i) \quad (4.18)$$

$$s_k^2(x | x_i, \zeta(x_i), i = 1, \dots, k) = \gamma_k \sum_{i=1}^k (\sigma(x, x) - \sigma(x, x_i)) w_i^k \quad (4.19)$$

where w_i^k are weights chosen such that $\sum_{i=1}^k w_i^k = 1$ and $m_k(x | x_i, \zeta(x_i), i = 1, \dots, k) = \zeta(x_i)$ at the k observed points, i.e. the mean value interpolates the known responses. Zilinskas [75] has proved that a sequence of points thus generated, converges to the global optimum of $\zeta(x)$.

The P-algorithm is a general formulation of a strategy to maximize the information gained by each function evaluation, and is quite easy to implement. However, in the implementation of the P-algorithm [67], several decisions have to be made that affect the speed and the accuracy of the method, namely the following:

- The appropriate form of the weighting functions w_i has to be chosen.
- The appropriate form of the covariance $\sigma(x, y)$ has to be chosen. It has been suggested that σ should be such that $(\sigma(x, x) - \sigma(x, z)) = \|x - z\|$ where $\|x - z\|$ is the Euclidean norm.
- An appropriate search method for finding x_{k+1} must be devised. This could be another multimodal optimization problem.
- An appropriate value of y_{ok} has to be chosen. It has been shown in [75] that too small a value of y_{ok} leads to the points of greatest uncertainty. If y_{ok} greater than or equal to $\min_{x \in A} m_k(x \mid x_i, \zeta(x_i), i = 1, \dots, k)$, then the next point chosen will be those x values that attain this minimum.

In the next section, the implementation of the P-algorithm is described. The process is iterative, hence the user can stop the iterations any time the best existing solution is satisfactory. The number of simulations to be run are directly controlled by the user. The algorithm only identifies the most promising points for simulation.

4.4.2 Implementation

The P-Algorithm is a formalization of an intuitive search strategy, and in fact, provides a framework for devising global optimization algorithms. The particular implementation of the P-Algorithm for this thesis is now described:

1. Choose k points x_i , $i = 1, \dots, k$ randomly from A using Latin Hypercube Sampling [45] and compute $\zeta(x_i)$ by simulation. Start iteration $l = 1$.
2. Using the BLUP and MSE expression in [53] (See appendix B), find the mean $m(x_j)$ and variance $s(x_j)$ at $N \gg k$ uniformly distributed points in A .

3. Find the smallest value of $m(x_j)$, i.e.

$$m_l(x) = \min_{j \in 1 \dots N} m(x_j). \quad (4.20)$$

Let $y_{ok}^l = m_l(x) - \epsilon_l$. At each x_j find the probability $P_{x_j}(y_{ok}^l)$.

4. Choose m_l points with largest probability $P_x(y_{ok}^l)$ from the N points.
5. Compute $\zeta(x)$ at the n_l points found above. If $\min_{j \in 1 \dots n_l} \zeta(x_j)$ is satisfactory, then stop, else continue
6. $k = k + n_l$. If $k > K_{max}$, then stop, else $l = l + 1$, go to step 2.

This algorithm is parameterized by the constants k , n_l , ϵ_l and K_{max} . These parameters have to be adapted to the specific problem or left to the designer's judgement. For example $k = 10 * d$, $n_l = 2d$, where d is the dimensionality of the design space A , were found to be good values for problems considered in this thesis. In this way the designer directly controls the number of simulations to be run. The choice of N search points can be suitably biased by the designer's judgement, and can account for constraints on the design space. Remember that only the mean and variance has to be estimated at the N points, which is quite inexpensive. In most applications considered here, the design space is finite, i.e., it is a finite subset of R^d . Hence the design space can be explored exhaustively with the predictor function.

4.4.3 Constrained Optimization

Constraints on the design further complicates the optimization problem. Fortunately, when the design domain is finite, handling constraints will be considerably easier. The optimization strategy is already search based. Hence the constraints can be evaluated whenever the performance to be optimized is being evaluated at a search point, either through simulation, or by prediction.

If calculating the constraints is relatively much cheaper than evaluating the objective function (e.g. analytical constraints on the design variables) they can be handled very naturally by screening each of the N points chosen in step 2 of the algorithm for constraint violation. Those points that violate the constraints are rejected. When the constraints are implicit and can be checked for only after simulation, e.g. a maximum delay or power restriction when doing a skew optimization, a different procedure has to be adopted. If the constraint can be evaluated through the same simulation, then another stochastic model can be used to model the constraint.

Consider the following non-linear constrained optimization problem:

$$\begin{aligned} & \text{minimize} && f_0(x) \\ & \text{subject to the constraints} && f_i(x) \leq c_i \quad i = 1, \dots, p \end{aligned}$$

We assume that $f_i(x)$ can be computed using the same simulation as $f_0(x)$. Hence we approximate $f_0(x), \dots, f_p(x)$ through separate model functions $\phi_0(x), \dots, \phi_p(x)$. Now the function $\phi_0(x)$ can be minimized probabilistically only. Similarly the constraints can be satisfied only with a certain probability depending upon the conditional distribution of $\phi_1(x), \dots, \phi_p(x)$. So the optimization problem is reformulated as follows:

$$\begin{aligned} & \text{maximize} && P(\phi_0(x) \leq y_{ok}) \\ & \text{subject to the constraint} && P(\phi_1(x) \leq c_1, \dots, \phi_p(x) \leq c_p) \geq C \end{aligned}$$

If the constraining functions are independent, then the constraint satisfaction probability

$$P(\phi_1(x) \leq c_1, \dots, \phi_p(x) \leq c_p) = \prod_{i=1}^p P(\phi_i(x) \leq c_i) \quad (4.21)$$

Alternatively, a lower bound can be put on the satisfaction of each constraint:

$$\begin{aligned} & \text{maximize} && P(\phi_0(x) \leq y_{ok}) \\ & \text{subject to the constraint} && P(\phi_i(x) \leq c_i) \geq C \quad i = 1, \dots, p \end{aligned}$$

This ensures that each constraint has a high probability of being satisfied. The choice of the constant C is tricky. If C is too high, then the candidate points for further simulation will all be far away from the boundary of the feasible region. If the constant C is too low, however, the points chosen for further simulation might not satisfy the constraints at all. This will imply a slower convergence to the optimal value which has to lie in the feasible region. Again, the efficacy of this method depends on the design problem. If the optimal solution lies close to the constraint boundary, then it is difficult to detect. If, however, the optimal solution lies well within the constraints, then the method will work quite effectively. Thus this method is not suitable for a very tightly constrained problem.

Chapter 5

Global Routing and Rule Generation

5.1 High performance layout design

High performance layout synthesis requires the careful management of resources, such that the design can be completed and the performance bounds can be satisfied. The layout design task is broken down into several subtasks, from floorplanning and placement, to global and detailed wire routing. In the floorplanning and placement phase, location of the components on the wiring substrate are determined. The main concern here is that the total substrate area is minimized and there is sufficient room to route the wires as well as meet the performance requirements. In the global wiring phase, the approximate course for each net in the design is determined such that there is a high likelihood of meeting the performance constraints and finding non-intersecting paths for each net. Hence the global wiring phase is the primary design task where electrical performance can be given proper consideration for all the nets in the design. In this chapter, a new global routing methodology is proposed that helps in generating approximate paths for each net such that the delay and noise requirements are met, and the routing resources are properly utilized. Also, a methodology is given for generating precise *wiring rules* for each net in the design. The combination of

global wiring paths and wiring rules, when specified to a detailed router, will enable the layout to be successfully completed.

5.2 Global Routing

Routing traces on a high speed board or MCM is a complex task. The objective of the routing is to find non intersecting paths for each net in the design. A more difficult objective in high speed designs is to meet a number of electrical constraints that the signals propagating on the wires have to satisfy. Current commercial routers are ill-equipped to meet these demands. The reason are two-fold. Firstly, the constraints are specified in the electrical domain, while the routers handle purely geometric constraints. Hence a suitable method has to be devised to translate the electrical constraints into geometric constraints. Secondly, most PCB routers are inherently sequential. They route one net at a time. Hence the nets that are routed later in the design have to contend with obstacles created by wires routed earlier. This is termed the net ordering problem. The myopic view of the router thus imposes artificial constraints on some of the nets, making it nearly impossible to satisfy the electrical constraints. Thus there is a need to decompose the routing problem into two phases: a *Global Routing* phase, where approximate paths for the wires are determined, so that the electrical constraints are likely to be satisfied and non-intersecting routs can be determined for each net; and a *detailed routing* phase where exact courses for the wires, along the paths specified by the global routing phase.

There are several advantages to performing a global routing, prior to detailed routing. If a good estimation of routing congestion can be made during global routing, then the following detailed routing phase is very likely to successfully route all the nets. Secondly, it is possible to manage all the nets at the same time in global

routing, and hence the net ordering problem is largely eliminated. Also, it is possible to determine whether the electrical constraints are likely to be met at all. This provides very good feedback to the placement program in terms of the problems to be corrected for producing a feasible design. Hence the global routing phase performs a feasibility estimation from the electrical perspective, in addition to identifying the routs for all nets that are most likely to lead to a successful layout design.

A necessary condition for the global routing phase to be formulated is that the output of the placement program must allow the definition of a routing graph [40]. Hence the routing region on the board or MCM must be decomposable into subregions that can be assigned to edges in a routing graph. This depends on the fabrication technology. If there are few routing layers available, then the topological freedom of wires is limited, which tends to increase the decomposability of the routing region, hence permitting a more accurate determination of the routing graph. For this reason, global routing is entirely feasible for MCM-D technology. Moreover, the small size of vias, and the similar electrical characteristics of the wiring layers make the performance modeling problem considerably easier. For MCM-L and multilayer PCBs, the large number of wiring layers provide great topological flexibility. However, for MCM-L and multilayer PCBs, the detailed routing problem can be considered an *area routing* problem, and a lot of the concepts presented here can be extended to these technologies. In this thesis, the design problem for MCM-D and PCBs with a small number of wiring layers only, are considered. The reader is referred to chapter 9 in Lengauer[40] for a detailed exposition on the general global routing problem and the extension to area routing on multilayer PCBs.

5.2.1 Formulation

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. To formalize this problem, the notion of a routing graph has to be established. The routing graph is established from the floorplan. In Lengauer[40], two design styles are distinguished. The *channel-free* design style, where mostly variable cells are used in the floorplan, and routing through the cells is possible. In such a case, the routing graph is the dual of the floorplan as shown in figure 5.1. The other style uses fixed cells with pin locations known beforehand. Here wiring through the cells is undesirable and has to be performed in *channels* along the cell, or chip, boundaries. The routing is termed the *channel intersection graph*. An example of such a graph for the floorplan shown in figure 5.1 is shown in figure 5.2. Since in PCBs and MCMs, the chip sizes and pin locations are fixed, the latter design style is more applicable.

The edges in the routing graph have a wiring capacity associated with them. This capacity determines the number of wires that can be routed through a wiring channel in the given technology. For fixed wire pitch, an upper bound on the channel capacity is easily determined. 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: 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 *nets* N , where each net is a subset of V . Each edge is labeled with a capacity $c : E \rightarrow R^+$ and edge lengths $l : E \rightarrow R^+$. Each net has a multiplicity $k_n \geq 1$. In addition, for each net $i \in N$, there is a set of admissible routes, or *trees* $T_i^1, \dots, T_i^{k_i}$. A solution to the global routing problem is a set of admissible routes, one or more for each net, such that the

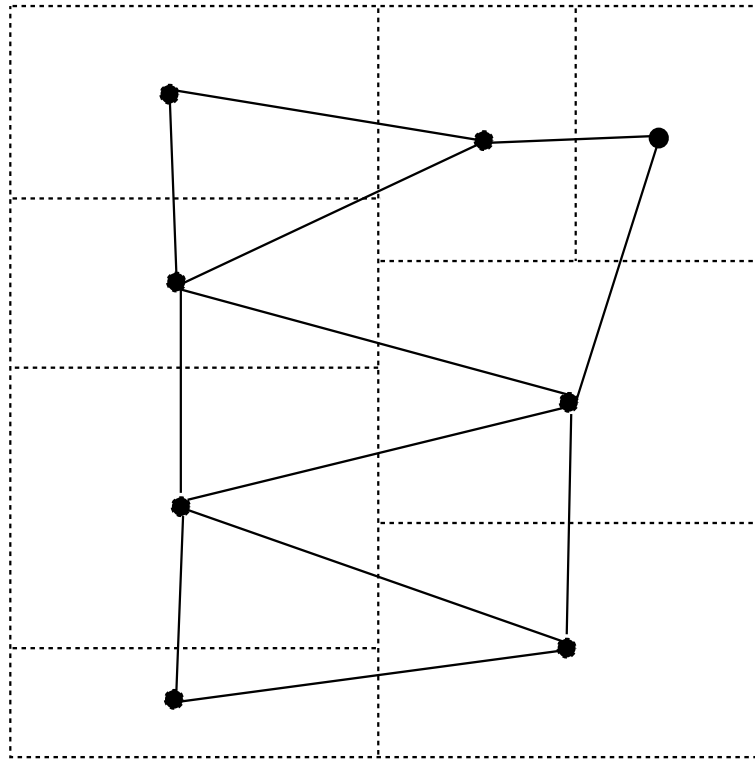


Figure 5.1: Channel free design style

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 :

$$U(e) = \sum_{i \in N, t \in T_i, e \in T_i^t} w(i, t) \quad (5.1)$$

The weights $w(i, t)$ denote the number of wires in super-net i that are routed using tree t . The objective function to be minimized over all such feasible solutions varies. Some formulations try to minimize wirelength. For our purpose, it is most important that the routing alternative chosen for a net, satisfies the electrical constraints for the net. Hence a benefit function $b : T \rightarrow R$ should be associated with each tree. $b(i, j)$ reflects the likelihood of satisfying the electrical constraints

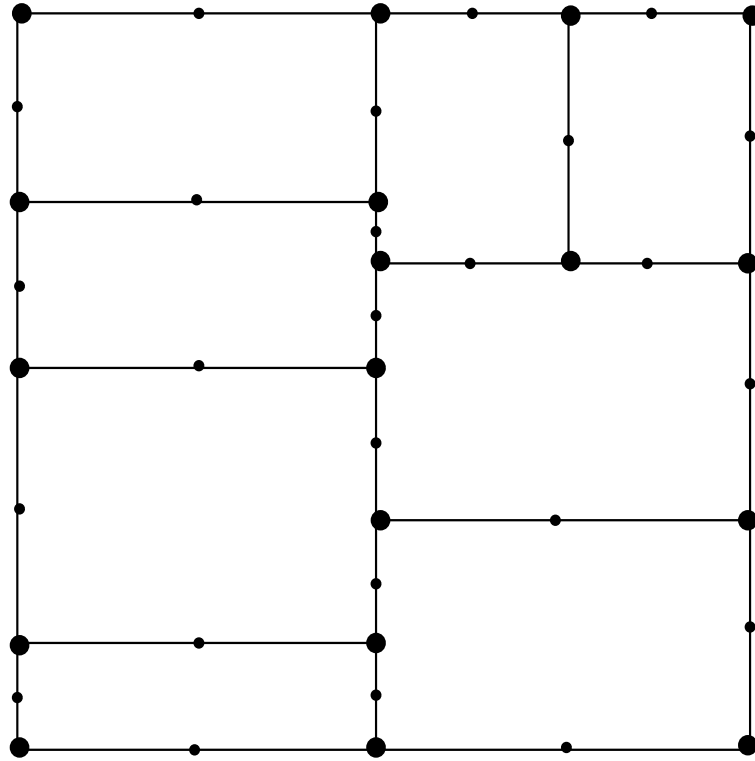


Figure 5.2: Channel Graph

associated with net i when routed using tree T_i^j . Hence the objective of the global routing is to maximize $B(T)$:

$$B(T) = \sum_{i \in N, j \in i_i} b(i, j) \quad (5.2)$$

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.

5.2.2 Constrained 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. This research has primarily focussed on the *Steiner tree* generation problem, with the objective of minimizing total wirelength. From the performance standpoint, if a simple RC delay model is assumed, and a lumped approximation is used for the net capacitance, then the wirelength minimization objective is equivalent to the delay minimization objective. The minimal Steiner tree construction problem is NP-hard [20]. Several optimal and approximation algorithms have been developed for the Steiner tree problem. Korte [33] has a comprehensive discussion of the work on this subject.

Recently, timing driven Steiner tree generation algorithms have started to emerge [8], [9], [6],[24],[63]. These algorithms model the delay of the tree using a distributed RC model or the Elmore delay model [16]. Again, these models are inadequate for predicting the delay for MCM and PCB interconnect, which are dominated by transmission line effects. The problem of modeling the delay and other signal integrity requirements has already been explored in this thesis. It is quite clear that it is near impossible to have analytical expressions relating delay and noise to the tree topology and wirelength. 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 technology 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. Also, the routing resources do not have to be absolutely minimized, unlike chip design, where chip area is at a real premium. Hence the wirelength can be longer than optimal without resulting in a dramatic increase in resource requirement. 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 the 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.

The topologies used for generating the routing trees are the same as those described in the characterization section. For point to point nets, there is only one topological way of constructing routing trees. For multi-point nets, several topologies have been shown to have good delay and noise characteristics [35], for example daisy chains, daisy chains with stubs, far end clusters and near end clusters. These are shown in figure 5.3. The next two sections present optimal and heuristic algorithms for generating short wirelength trees for point-to-point and multi-point nets respectively.

Point-to-point nets

For point to point nets, the routing trees are constructed by following the shortest path from the driver pin to the receiver pin.

Def. The *shortest* path between two vertices 'u' and 'v', in a weighted graph G is defined as the path $u v_1 v_2 \dots v$ such that $\sum_i w(v_i, v_{i+1})$ is minimum.

There is extensive literature on solving the shortest path problem. See [36], [4] and chapter 3 in [40] for a good review of existing literature. Lengauer [39] and Rote [52] describe a generalized extension of the shortest path problems. In the definition given above, the path cost is composed by performing a real addition of

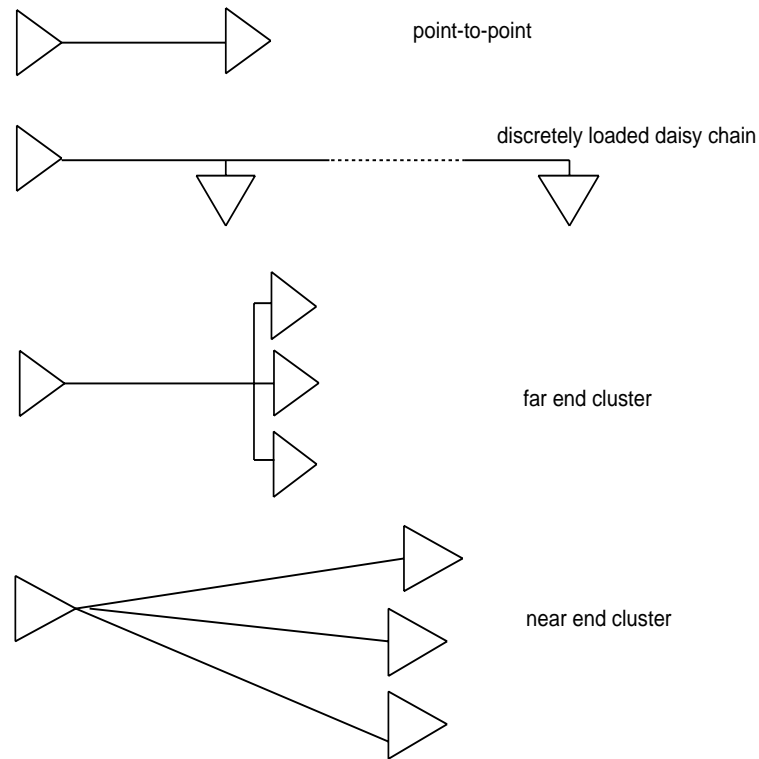


Figure 5.3: Net Topologies

the edge weight along the graph. In general, addition can be replaced by any binary operation \odot to yield costs of paths from cost of edges. The path costs are aggregated using the “minimum” operation, which is generalized by the operation $\oplus M(R) \rightarrow R$, where $M(R)$ is the set of all countable multisets that are composed of elements in R . R is the set from where the edge labels take their values, which we have assumed to be R^+ so far. With this generalization, the shortest path problem asks for computing the values:

$$d_{ij} = \oplus \{w(p) \mid p \text{ is a path from } i \text{ to } j\} \quad (5.3)$$

where, for a path $p = (v_0, \dots, v_k)$,

$$w(p) = (\dots(w(v_0, v_1) \odot w(v_1, v_2)) \dots) \odot w(v_{k-1}, v_k) \quad (5.4)$$

The combination $C = (R, \oplus, \odot, 0, 1)$ where 0 and 1 are the neutral elements for the \oplus and \odot operations respectively, is called an *algebraic cost structure*. Lengauer [39] lists a set of properties that C must satisfy for it to be a *closed semiring*. With such a cost structure, the algorithms of Floyd[17], Tarjan[66] may be employed to solve the path problem. Lengauer[39] shows how to eliminate the associativity (on \odot) and distributivity (on \oplus) restrictions, and to solve path problems for such cost structures. This approach holds a lot of promise for optimally solving path problems relating to signal integrity requirements, especially in lossy interconnect, where the delay is non-linearly related to the length of the interconnect. In this thesis, the problem of finding short paths that meet the signal integrity constraints is solved by finding several paths with short wirelength, and screening them using the characterizations. This procedure helps keep the problem quite tractable, and, as the results demonstrate, does not compromise performance constraints.

The problem to be solved, then, is to enumerate the k shortest paths for a given pair of source and destination vertices. A restriction to be imposed on these paths is that there should be no repeated vertices along any of the paths, the reason being that a path length can be extended by adding a small cycle in the path. The cycle acts only as a shorted loop in the interconnect and hence only adds a discontinuity on the net. Also, the extra loop does nothing to reduce the congestion. Hence all the short paths should be such that they have no repeated nodes on them. An algorithm to solve the k shortest path problem, with no repeated nodes, is given in Lawler[36]. This algorithm is described next with an illustration.

The data structure required to is a list of shortest paths called P . The

source vertex is called v_1 and the destination is v_n . Initially, P is empty. The shortest path computation proceeds as follows:

1. Compute the shortest path from v_1 to v_n using Dijkstra's shortest path algorithm [12]. Place this path on P and set $m = 1$.
2. If P is empty, then stop. There are no more paths between v_1 and v_n . Otherwise, remove the shortest path from P and output it as the m th shortest path, P_m .
3. Computing k paths, asks for forcing the short path procedure to avoid using certain edges in the graph. *Suppose* that P_m contains the vertices v_1, \dots, v_{n-1}, v_n , and that P_m is the shortest path from v_1 to v_n subject to the condition that it is forced to go through vertices v_1, \dots, v_p where $p \leq n - 1$ and that certain edges from v_p were excluded in doing this shortest path computation (This information is stored with P_m as part of the same entry in P).

If $p = n - 1$, find the shortest path from v_1 to v_n subject to the condition that edges $(v_1, v_2), (v_2, v_3), \dots, (v_{n-2}, v_{n-1})$ are included and that (v_{n-1}, v_n) is excluded, in addition to the other edges excluded in calculating P_m . If such a path exists, then place it in P along with a record of the conditions under which it was obtained.

If $p < n - 1$, then find the shortest path from v_1 to v_n subject to the following sets of conditions:

- (a) Edges $(v_1, v_2), (v_2, v_3), \dots, (v_{p-1}, v_p)$ are included and edges (v_p, v_{p+1}) is excluded, in addition to the edges excluded in calculating P_m .
- (b) Edges $(v_1, v_2), (v_2, v_3), \dots, (v_p, v_{p+1})$ are included and edges (v_{p+1}, v_{p+2}) is excluded, in addition to the edges excluded in calculating P_m .

⋮

(c)

(d) Edges $(v_1, v_2), (v_2, v_3), \dots, (v_{n-2}, v_{n-1})$ are included and edges (v_{n-1}, v_n) is excluded, in addition to the edges excluded in calculating P_m .

If such paths exist, then place them in P along with a record of the conditions under which they were obtained. Increment m and return to step 2.

Figure 5.4 shows a graph on which the 3 shortest paths from vertex A to vertex F have to be computed. Table 5.1 shows the paths in list P at the beginning of step 2 in the algorithm for several iterations.

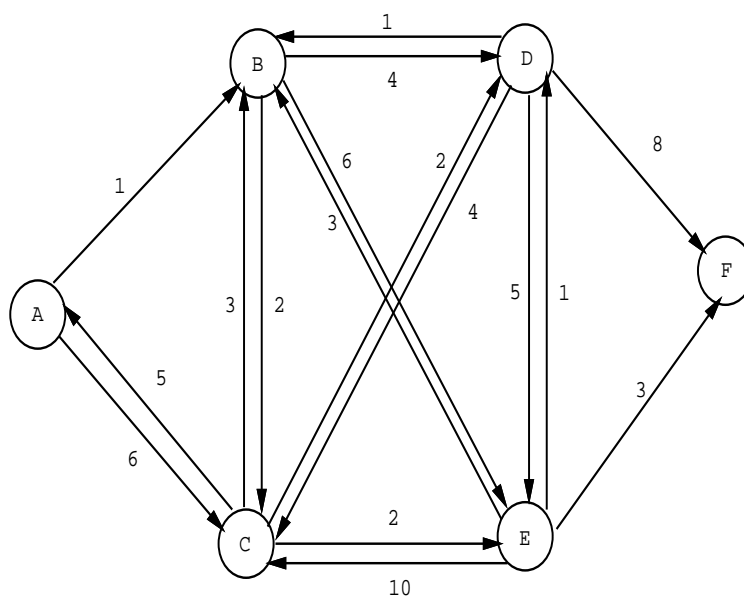


Figure 5.4: Example Graph

Table 5.1: Short Path list for graph in figure 5.4

m	P	length	deleted edges	fixed	
1.	A B C E F	8	None	A	P1
2.	A B E F	10	[BC]	B	P2
	A B C D E F	13	[CE]	C	
	A B C E D F	14	[EF]	D	
3.	A B C D E F	13	[CE]	C	P3
	A B C E D F	14	[EF]	D	
	A B D E F	13	[BC][BE]	B	
	A B E D F	16	[BC][EF]	E	

Multi-point nets

For multi-point nets, there are several possible topologies for connecting a net. The ones with good electrical properties are the *daisy chain*, *far-end cluster* and *near-end cluster* topologies. These topologies are shown in figure 5.3. Generating short wire-length trees in these topologies make extensive use of an algorithm to find the shortest path between two points. Hence the discussion in the previous section on extensions of the shortest path methods to incorporate more complicated cost structures extends to these tree generation methods also. The next few sections outline the algorithms for generating short wirelength trees in various topologies.

Daisy Chain

A short daisy chain is essentially a sequence of short paths from one net terminal to the next. So, if the terminal sequence is specified, generating the shortest daisy chain is straightforward. The algorithm for generating the k shortest daisy chains makes use of the k shortest paths procedure presented in the previous section. The

data structure needed here is 3 lists of shortest paths P^0 , P^1 and P^2 . Initially all 3 lists are empty. The algorithm adds terminals one at a time. Suppose the net has m terminals (1 driver pin and $m - 1$ receiver pins), v_1, \dots, v_m . The following is an outline of the algorithm for generating the k shortest paths:

```

for(i=0; i < m; i++)
{
find_paths(v_i, v_{i+1}, k, P^0);
append_paths(P^0, P^1, P^2, m, i);
path_copy(P^1, P^2);
}

```

The procedure *find_paths* computes the k shortest paths between it's first two arguments and puts them on P^0 . The procedure *append_paths* takes the k shortest paths between vertices v_i and v_{i+1} listed in P^0 and the k shortest daisy chains from vertex v_1 to v_i listed in P^1 , and uses them to find the k shortest daisy chains from vertex v_1 to v_{i+1} and lists them in P^2 . Finally procedure *path_copy* copies the list P^2 to P^1 to start the next iteration. So, at the start of each iteration, the shortest daisy chain upto vertex i is stored in P^1 . At the end of m iterations, P^1 contains the k shortest daisy chains connecting v_1, \dots, v_m . Procedure *find_paths* is essentially the k shortest paths procedure outlined in the previous section. Procedure *append_paths* is outlined next.

The *append_path* procedure takes as arguments two path lists (P^0 and P^1) with k paths, and produces another path list (P^2) with k paths, which are a concatenation of a path in P^1 followed by P^0 . A total of k^2 such paths exist, of which the shortest k must be put on list P^2 . Let indicies i, j and l index paths in P^0, P^1

and P^2 respectively. Each of i , j and l range from $1, \dots, k$. The following lemma proves shows gives a restriction on i and j for a given value of k :

Lemma 1. If the l th shortest path in P^2 is the concatenation of the i th shortest path in P_0 and the j th shortest path in P^1 then:

1. $i \leq l$
2. $j \leq l$
3. $i + j \leq l + 1$

Proof: The paths in P^0 and P^1 are in increasing order of path length. It suffices to prove that the third condition, since if $i + j \leq l + 1$ and $j \geq 1$ and $i \geq 1$ implies the first two conditions. For any i and j , there are $i \star j - 1$ path concatenations that are shorter than the concatenation of i and j . So, we need to show that if $i + j > l + 1$, then $i \star j - 1 \geq l$. Suppose $i + j = l + 2$. Then, the smallest value of $i \star j$ is achieved when either $i = l + 1$ or $j = l + 1$. Then $i \star j - 1 = l + 1 - 1 = l$. Hence if $i + j \geq l + 1$, there are at least l paths shorter than the concatenation of P_i^0 and P_j^1 . Thus the l th shortest path must have $i + j \leq l + 1$.

Hence the k shortest paths are given by as the k shortest paths in the concatenations of paths in P^0 and P^1 whose indicies satisfy Lemma 1.

Stubs

Daisy chains are perhaps the best way of connecting several loads to a driver from the signal integrity viewpoint. In some cases, however, pure daisy chaining is prohibitive.

One example of this is connecting Pin Grid Arrays. The routing resources underneath the PGA are extremely limited. Using a daisy-chain configuration requires two wires to be escaped from under the PGA (see figure 5.5). If a short stub could be used instead, then only one wire needs to be escaped per pin. Similar considerations apply to Ball Grid Arrays and flip-chip solder bumped die. Using stubs generally leads to shorter wirelengths. The drawback is that the stub adds extra capacitive discontinuity. If the stub gets electrically long, then it can give rise to discrete reflections leading to very poor electrical behavior. Nevertheless, short stubs usually are very useful to reduce congestion, and do not lead to a serious performance degradation.

For global routing, short stubs are introduced by considering only nodes that are a certain short distance from a terminal node as candidates for stub points. The maximum allowed stub length is called l_{stub} . For a three terminal net (v_1, v_2, v_3) , all the nodes that are less than a distance l_{stub} from the v_2 node are considered candidate nodes for the stub point. Whichever gives a shorter total tree length is chosen as the stub point:

$$v_{stub} = \text{Arg} \min_{v \in V, v \neq (v_1, v_2, v_3), d(v_2, v) \leq l_{stub}} (d(v_1, v) + d(v_2, v) + d(v_3, v))$$

For a k terminal net (v_1, \dots, v_k) , $k > 3$, $k - 2$ stub points, $(v_{stub}^1, \dots, v_{stub}^{k-2})$ are to be located. A greedy approach is adopted. The problem is solved 3 terminals at a time. First a stub point is found for (v_1, v_2, v_3) . Then, a stub point is found for (v_{stub}^1, v_3, v_4) and so on, using the above definition, until all the terminals are connected:

$$v_{stub}^l = \text{Arg} \min_{v \in V, v \neq (v_{stub}^{l-1}, v_{l+1}, v_{l+2}), d(v_{l+1}, v) \leq l_{stub}} (d(v_{stub}^{l-1}, v) + d(v_{l+1}, v) + d(v_{l+2}, v))$$

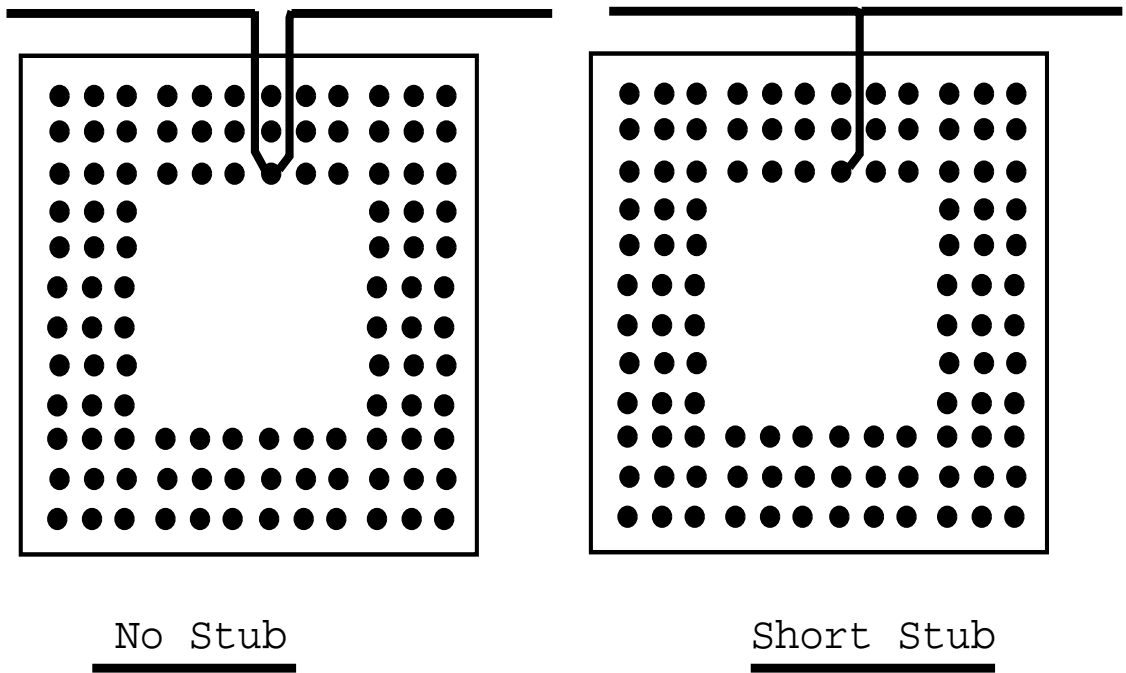


Figure 5.5: Adding a stub saves an escape path on a PGA

Far-end cluster

The far-end cluster connects a net with a single Steiner point. All the terminals are connected directly to this point. A good far-end cluster tree has short paths between the Steiner points and the loads. This suggests the following definition of the Steiner point:

$$v_{st} = \text{Arg} \min_{v \in V} \max_{i \in \{2, \dots, m\}} d(v, v_i)$$

Finding the Steiner point, then, entails knowing the shortest paths between all pairs of vertices in the graph. If a shortest path matrix is precomputed, then there is very little effort required to identify the Steiner point from the above

definition. Building the shortest path matrix takes $O(n^3)$ time. Its useful to build this matrix before any routing trees are generated, as all the algorithms make extensive use of shortest paths between pairs of vertices.

Near-end cluster

The near-end cluster connects a net without a Steiner point. All that is required is determining the shortest paths from the driver pin to each of the receiver pins. With a pre-computed shortest path matrix, this takes $O(m)$ time, where m is the number of receivers.

After enumerating a set of routing trees, the benefit function which ascertains the likelihood of the signal integrity constraints being met, if a net is routed using a certain tree, has to be determined. The characterization results have to be used for this purpose. The details of this are described in the next section.

5.2.3 Benefit Function

The benefit function describes the likelihood of the signal integrity constraints being met by a certain routing tree of a given net. The benefit function is hard to compute exactly for two reasons:

1. The electrical properties of the tree are not precisely known, since the values predicted by the characterizations are uncertain.
2. The physical variables related to the tree are not precisely known. The routing trees are generated from the global routing graph. This channel lengths in this graph are only approximately known. Moreover, the pin locations are

abstracted to a node in the graph. This gives rise to considerable uncertainty in the length estimates from the routing graph.

The characterization can be captured using either the Moving Least Square Interpolant or the Stochastic Model. If the stochastic model is used, then the uncertainty in prediction is captured in the MSE estimate. Suppose that there are n electrical properties of interest, p_1, \dots, p_n , and there are upper and lower bounds l_1, \dots, l_n and u_1, \dots, u_n respectively. Then the benefit function is given as:

$$B(x) = \prod_{i=1}^n P(p_i(x) \leq u_i, p_i(x) \geq l_i) \quad (5.5)$$

where x is the vector of physical design variables for the routing tree. The probability $P(p_i(x) \leq u_i, p_i(x) \geq l_i)$ is easily calculated given the BLUP and MSE values for each of the electrical properties.

With the Moving Least Square Interpolant, 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 remains the same as in equation 5.5. The $P()$ function can be redefined as shown in figures 5.6 and 5.7. The function in figure 5.6 is:

$$\begin{aligned} P_i(x, u, l) &= \exp\left(\frac{l_i - p_i(x)}{e_i}\right) & p_i(x) \leq l_i \\ &= \exp\left(\frac{u_i - p_i(x)}{e_i}\right) & p_i(x) \geq u_i \\ &= 1 & l_i < p_i(x) < u_i(x) \end{aligned}$$

where e_i is the error threshold in the characterization for p_i .

The function in 5.7 is:

$$P_i(x, u, l) = \frac{1.0}{1.0 + \exp\left(l_i - \frac{p_i(x)}{e_i}\right)} \times \frac{1.0}{1.0 + \exp\left(\frac{p_i(x) - u_i}{e_i}\right)} \quad (5.6)$$

The uncertainty in the wirelength is very hard to model in the benefit function, given that the uncertainty in the edge lengths is not known. The uncertain-

ity created by abstracting the pin position, can be managed by adding or subtracting, the distance of the pin location from the graph node which it maps to, from the global routed length. The uncertainty in the edge lengths is not known until after detailed routing. One possibility is to multiply the global routed length by a factor greater than 1 to indicate the inefficiency of the local router. Another possibility is to perform a global routing with no correction for edge length uncertainty. Then, the congestion on the edges can be examined, and a router inefficiency factor can be associated with each edge depending on the congestion.

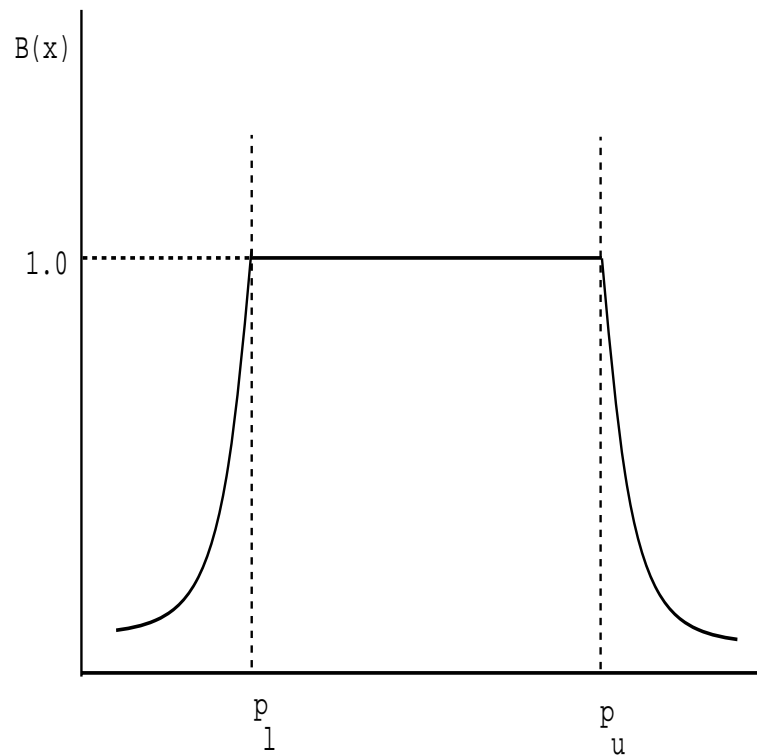


Figure 5.6: Benefit function

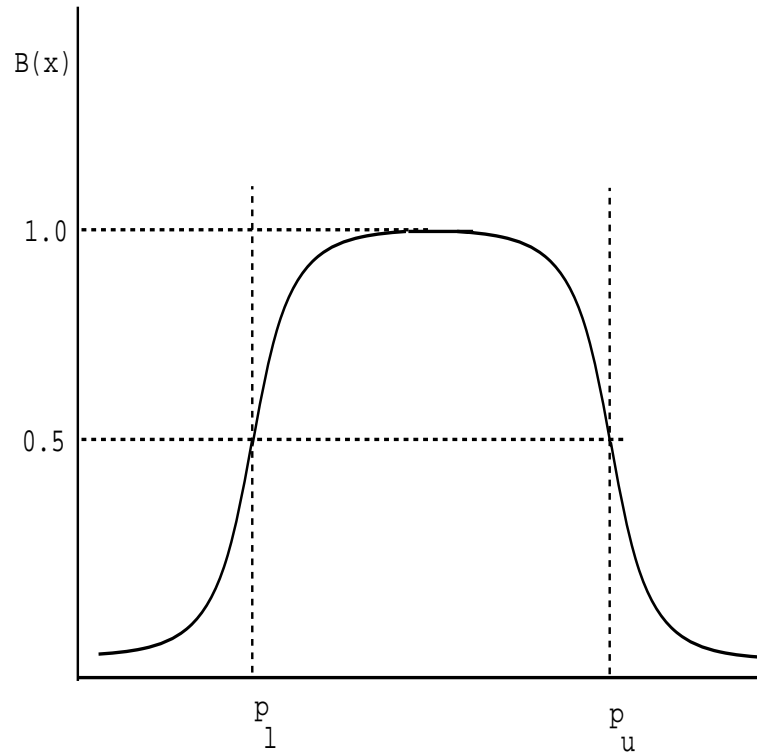


Figure 5.7: Benefit function

5.2.4 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 i of net j . Then the global routing problem is given by the following integer program:

$$\begin{array}{ll}
 \text{Maximize} & \sum_{i=1}^N \sum_{j=1}^{n_i} b_{ij} y_{ij} \\
 \text{subject to} & \sum_{j=1}^{n_i} y_{ij} = d_i \quad i = 1, \dots, N \\
 & \sum_{i=1}^N \sum_{j=1}^{n_i} a_{ij}^k y_{ij} \leq c_k \quad k = 1, \dots, m \\
 & 0 \leq y_{ij} \\
 \text{integer} & y_{ij}
 \end{array}$$

Here b_{ij} is the benefit for the tree i of net j , N is to total number of super-nets, d_i is the cardinality of super-net i , n_i is the number of routing trees for net i , c_k is the capacity of edge k . a_{ij}^k is a $(0, 1)$ matrix that specifies whether or not tree i of net j uses the edge k .

5.2.5 Solution Methods

Integer programming is, in general, NP-Hard. There are numerous ways of solving integer programs, e.g. cutting plane algorithms, branch and bound and Lagrangian relaxation [40]. One method that has been shown to be very effective for solving the global routing problem is a randomized rounding technique to the linear relaxation of the integer program [51] [50]. The basic idea is to relax the integer constraint in the formulation, which make 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. Carden[29][30] has shown how to correct the solution if some capacity constraints are violated after the rounding. In this thesis work, 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 no need to perform a rounding. In those cases where non-integer solutions, a simple rounding led to very few constraint violations, although the objective function might be sub-optimal. However, simply by rounding up the variables with higher benefit and rounding down variables with smaller benefit, a good solution is easily found.

5.3 Wiring Rule Generation

The solution of the integer program gives us a global routing solution where each net has a high likelihood of meeting the signal integrity constraints, and the following detailed routing is likely to be successful if a good estimate of channel capacities was made in the routing graph. There still exists the problem of driving a detailed router from the global routing solution, since the nets cannot, in general, be routed to exactly the wirelength estimated by the global router. For a detailed router, explicit constraints on the geometry of the routing trees should be specified, which, when adhered to, provide a high likelihood of the signal integrity requirements to be met.

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 [59]. However, such a global rule tends to be fairly conservative [60]. The reason is that wiring rules are not uniquely defined. Though a wiring rule can be specified with minimum length information obtained from placement, the global router gives a better starting point for wiring rule generation, as the global wiring length correspond to trees which are likely to keep congestion manageable. If the length estimates are kept at their minimum possible in the routing graph, then we are certain that the routed length can only be longer than that estimated from global routing. 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 [37] [38], 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 [49] is used to cal-

culate 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. This is illustrated in figure 5.8.

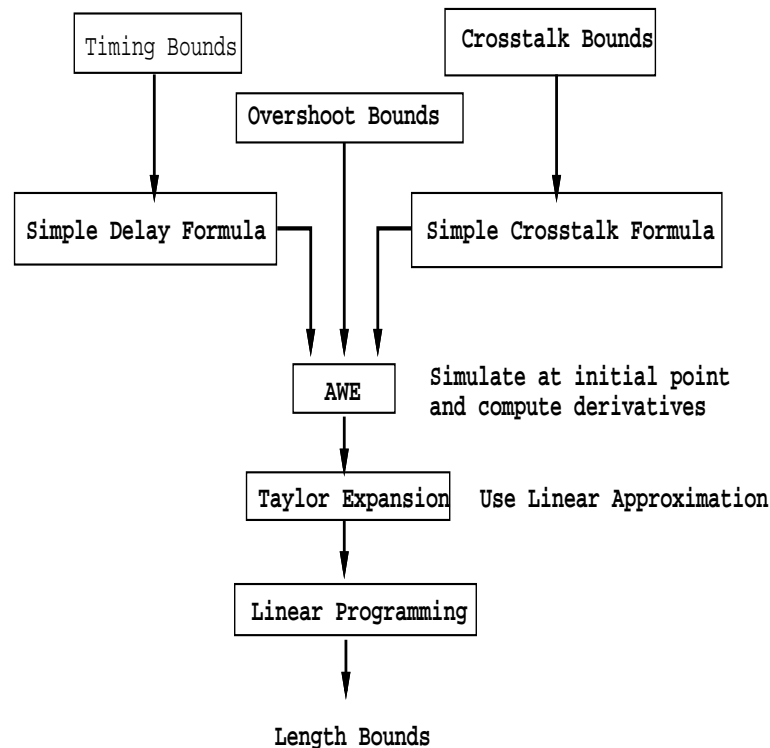


Figure 5.8: Lee's Approach to Rule Generation

The main drawback of this technique is that the simple equations cannot account for complex driver models. Also, the formulation is limited to daisy chained nets. Settling delay cannot be captured by this method either. Also, the net length bounds are generated totally disregarding the constraints induced by the board place-

ment. Nevertheless, the approach is quite attractive and can be extended to overcome these shortcomings.

The global routed solution gives us a minimum estimate of the net length. We need to find out how much these lengths can be relaxed by without violating the electrical constraints. Moving Least Square Interpolation gives us a local estimate of the electrical performance. Recall that the form of the Interpolant is:

$$\phi(x) = \sum_{i=1}^n a_i(x)b_i(x) \quad (5.7)$$

where $\phi(x)$ is the predicted value for some electrical performance for net length vector x and b_i 's are polynomial basis functions. If the basis functions are chosen to be linear in x , then the form of $\phi(x)$ is that of a local linear approximation. If we choose x as the global routed net length, then $\sum_{i=1}^n a_i(x)b_i(x)$ is a linear approximation of the electrical performance. This will serve the same purpose as the Taylor Series expansion in Lee's formulation. Note that any electrical performance can be approximated in this manner. So the minimum length constraints obtained from the global routed length and the electrical constraints specified by equating the linear approximations of the electrical performance to the bounds, describe a polytope over the space of net lengths. To obtain absolute bounds on the net lengths a largest hypercube has to be fitted in this polytope. This can be achieved by linear programming as presented in the next section. The overall procedure is illustrated in figure 5.9

5.3.1 Formulation

Suppose that the net is described by a vector of *physical design variables* (\underline{x}) = (x_1, \dots, x_d) . There are m electrical performances of interest p_1, \dots, p_m , with upper and lower bounds, given by u_1, \dots, u_m and l_1, \dots, l_m . The global routed solution

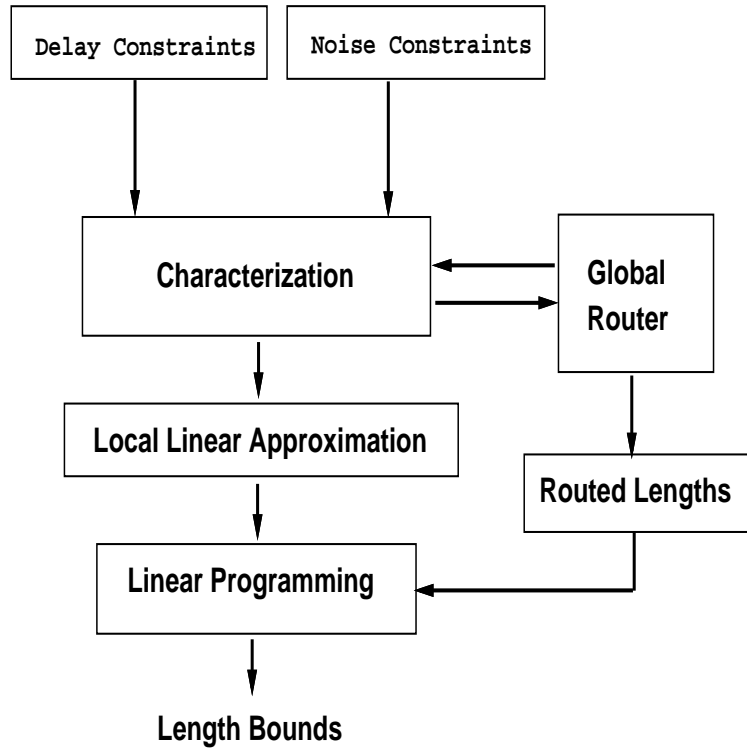


Figure 5.9: Rule Generation Approach

is the point $x_c = (x_{c1}, \dots, x_{cd})$. The electrical performances are approximated as $p_j = \sum_{i=0}^d a_{ij}x_i$ where $x_0 = 1$. The wiring rule generation problem is to fit a maximal hypercube in the polytope defined by the linear inequalities:

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

The problem of fitting the largest hypercube is a special case of fitting a largest *norm body* in a polytope [14]. A norm body is defined as:

$$\{x \mid n(x - x_0) \leq r\} \quad (5.8)$$

where x_0 is the center of the norm body and r defines its extent. For example if $n(\cdot)$ is the 2-norm:

$$n_2(x) = \|x\|_2 = \left(\sum_{i=1}^d x_i^2 \right)^{\frac{1}{2}} \quad (5.9)$$

the norm body is a hypersphere of radius r centered about x_0 . Similarly if $n(\cdot)$ is the max or infinity norm:

$$n_\infty(x) = \|x\|_\infty = \max_i \{ |x_i| \} \quad (5.10)$$

the norm body is a hypercube centered about x_0 with side $2r$.

Associated with each norm $n(x)$ is a dual norm $n^*(x)$ defined by

$$n^*(x) = \max_y \{ y^T x \mid n(y) \leq 1 \} \quad (5.11)$$

If $n(\cdot)$ is the p th norm:

$$n_p(x) = \|x\|_p = \left(\sum_{i=1}^d x_i^p \right)^{\frac{1}{p}} \quad (5.12)$$

the dual norm is

$$n_p^*(x) = \|x\|_q \quad (5.13)$$

where

$$\frac{1}{p} + \frac{1}{q} = 1 \quad (5.14)$$

So the dual norm of the 2-norm is the 2-norm itself, while the dual norm of the max norm is the 1-norm.

The constraints bounding the polytope define hyperplanes in the physical design space. The hyperplane corresponding to the j th constraint is denoted π_j . For example the hyperplane corresponding to the constraint $\sum_{i=0}^d a_{ij}x_i \leq u_j$ is $\pi_j \equiv \sum_{i=1}^d a_{ij}x_i = u_j - a_{0j}$. The distance in norm $n(x)$ from a point x to a hyperplane π is given as:

$$d_n(x, \pi) = \min_y \{n(y - x) \mid y \in \pi\} \quad (5.15)$$

The following theorem relating the distance from a point to a hyperplane, to the dual norm is given in [14]:

Theorem 1. Let $n(x)$ be a norm of x and $n^*(x)$ be the corresponding dual norm of x . Then the distance in n -norm from point x^0 to hyperplane $\pi \equiv \{x \mid \eta^T x = b\}$ is

$$d_n(x^0, \pi) = \frac{|b - \eta^T x^0|}{n^*(\gamma\eta)} \quad (5.16)$$

where

$$\gamma = \text{sgn}(b - \eta^T x^0) \quad (5.17)$$

Now the problem of fitting a maximal hypercube in the polytope defined by the constraints is given as:

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

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.

Using theorem 1, this can be written as a linear program:

$$\begin{aligned} & \text{maximize} && r \\ & x^0, r \\ & \text{subject to the constraints} \\ & \eta_j^T x^0 + r n^*(\eta_j) \leq b_j, && j = 1, \dots, d + 2m \end{aligned}$$

In our problem the norm $n(x)$ is the max norm and hence the dual norm $n^*(x)$ is the 1-norm. Hence the linear program is stated as:

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

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

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

Of course, several extensions to this formulation are possible. The most important one is *scaling*. All design variables do not have the same scale. Hence a scaling vector T can be defined to scale down each variable to the same magnitude. Another advantage of scaling is to allow one variable a greater degree of freedom over another. For example, when trying to meet noise constraints, the stubs cannot be

lengthened as much as the branches on the main line. Hence, the stubs length can be scaled up and the branch length scaled down in the formulation, to allow greater freedom in branch lengths.

5.4 Summary

In this chapter, a methodology for performing global routing in PCBs and MCMs is described. The purpose of the global routing is to identify promising paths for all the nets in the design, so that timing and signal integrity constraints are met, and the routing congestion is manageable. These approximate routes for each net can then be passed to a detailed router, to get a correctly constructed layout. Most current PCB and MCM routers are not, however, capable of following such guidance. In this case, the electrical constraints must be translated to constraints on the wirelength. A methodology is proposed for generating such constraints, or *Wiring Rules* that employs the global routing results and the characterizations of delay and noise. In chapter 6, the software tools developed to support the global routing and rule generation procedure are described. Chapter 7 reports the results of global routing and rule generation as performed on two MCM examples and the Intel Pentium board design.

Chapter 6

Tools

6.1 Introduction

In this chapter, the software tools developed to support the methodology for characterization and optimization are described. Also, the global router is described along with its interfaces to the characterization tool.

The characterization and optimization tool set are components in the Signal Integrity Advisor being developed at NCSU. The interfaces are to a user or user program and to the MetaSim [60] software. The architecture of the entire tool set is shown in figure 6.1. First, a brief description of MetaSim is given, followed by details of the characterization tool and the global router.

6.2 MetaSim

MetaSim is a tool for the automated management of simulation studies and waveform analysis. The MetaSim software modules are shown in figure 6.2. MetaSim interfaces to a user or user program through two files, the command file, and the study file. The command file contains a description of the order or simulations to be performed along with a description of the design variables, the waveform analysis to be performed and

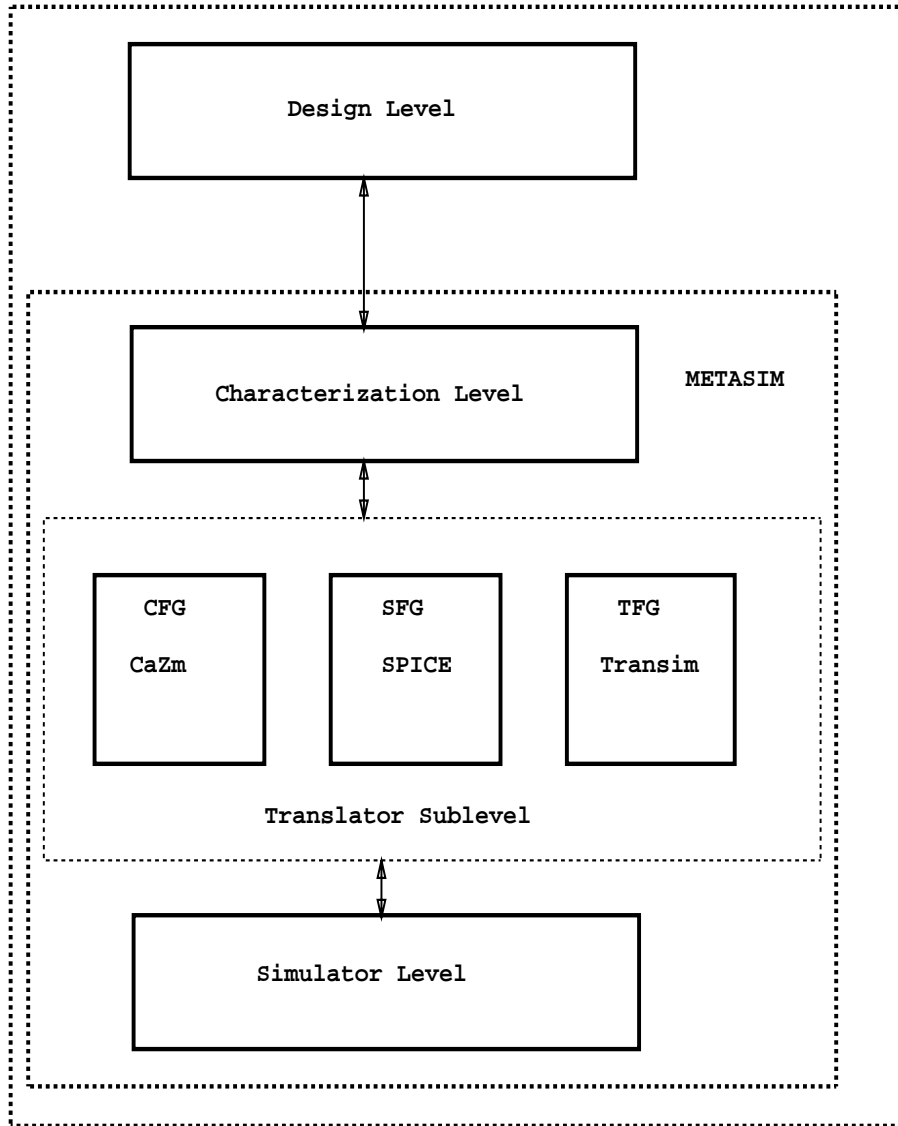


Figure 6.1: Tools Hierarchy

the manner in which results are to be reported. It also has capabilities of performing regression and statistical error analysis on the generated data. The study file can be either a generic description of an interconnect structure, or a simulator-specific circuit file, with certain values replaced by variable templates. These templates are the same as the variable names given in the command file.

When the study file describes a generic interconnect structure, MetaSim interfaces to a program called the CaZm File Generator (CFG) [60]. CFG contains routines for generating CaZm netlist by converting a generic description of the interconnect to a distributed RLC tree.

6.3 Characterization Tool

The characterization tool, called the Study Generator, contains a set of routines for parsing user information about the experiment to be run, generating user specified number of samples using Latin Hypercube Sampling, computing cross-validation error for an internally generated, or externally specified sample and performing error-based resampling. In addition, there are routines for Moving Least Square Interpolation, and calculating the BLUP and MSE, at any data point using either an internal or an external sample set. These set of routines support the full optimization and characterization methodology from a user input. The user input is a file giving the number of samples, a description of the variables and their ranges, a description of the responses and the basis functions to be used for the interpolant or the stochastic model. The full description of the user-interface file is given in Appendix D. Data can be generated in the Study Generator either internally, or it can use external data. Also, a query file can be specified whence the Study Generator will return the interpolated value of the response at the points specified in the query file.

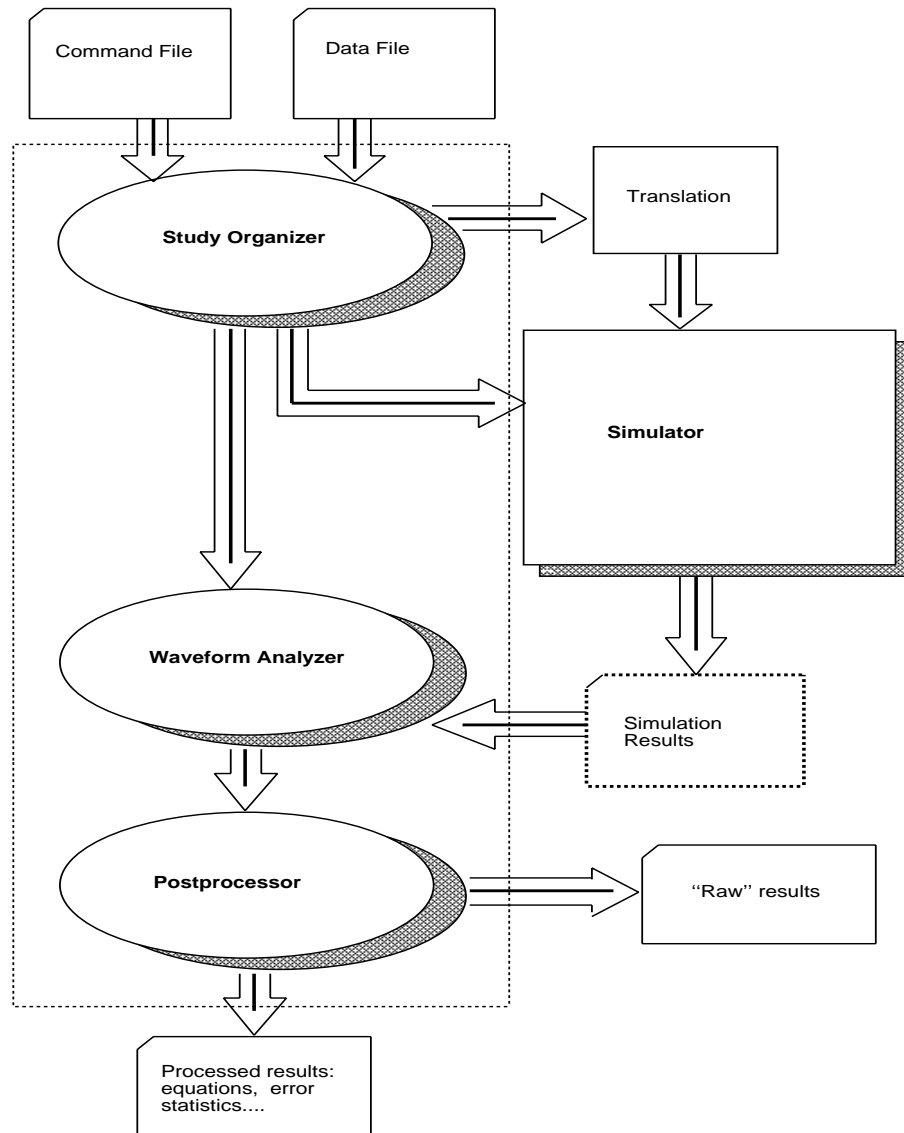


Figure 6.2: MetaSim

The modes of running the study generator are as follows:

1. When all data has to be generated internally, through a two stage study. In this case the number of initial points have to specified. The number of points in the second stage are specified by changing the parameter `MAX_RESAMPLES` in `sg.h` . No query file should be specified.
2. When external data has to be refined. In this case, the data flag must be `e!` and the number of existing data points must be specified. The number of resamples is specified as in 1 above. Again no query file.
3. When only querying should be done. In this case, external data must be specified, and a query file must be specified.

The study generator has 4 modules. The first is the external interface which basically parses the user information. The second is the Sampler, which performs Latin Hypercube sampling over the currently specified design space. The third is the error evaluator and interpolator, which performs Moving Least Square interpolation, both for error checking and for querying. The fourth is the interface to MetaSim, which generates the command file for MetaSim for each study.

When a typical study is performed, first the user interface is invoked for parsing the input file. Then the feasible space generator is called with the user specified constraints. This returns the smallest hyperrectangle containing the feasible region. Then, the Sampler is invoked which generates an LHS sample over this hyperrectangle. The MetaSim interface then writes a command file for these sample points and invokes MetaSim. After MetaSim finishes running, and if resampling is needed, the error evaluator is invoked. This then reads in the MetaSim generated, or external, as the case may be, data and performs the error checking. For each point

in error, its neighborhood is identified and the Sampler is called again, to generate samples in this neighborhood.

6.4 Global Router

The global router tool set is shown in figure 6.3. The global router contains routines for parsing the user description of the routing graph and the description of the nets and their types, for generating routing trees in several topologies as described in section 5.2.2, for calculating the benefit function for each tree, and writing the integer (or relaxed linear) program for solving the global routing problem. For calculating the benefit function, the global router interfaces to the characterization tool. With each net there is a description of the net constraints. These constraints are either specified as physical constraints or electrical constraints. If electrical constraints are given, then the characterization is invoked to predict the electrical performance of each tree. The value returned by the characterization is used to determine the benefit function. Also, if wiring rules are to be generated, a linear expansion of the electrical performance about the global tree length is returned. The global router then writes the linear programs for solving for the maximum and minimum bounds on the branch lengths, as given in section 5.3.

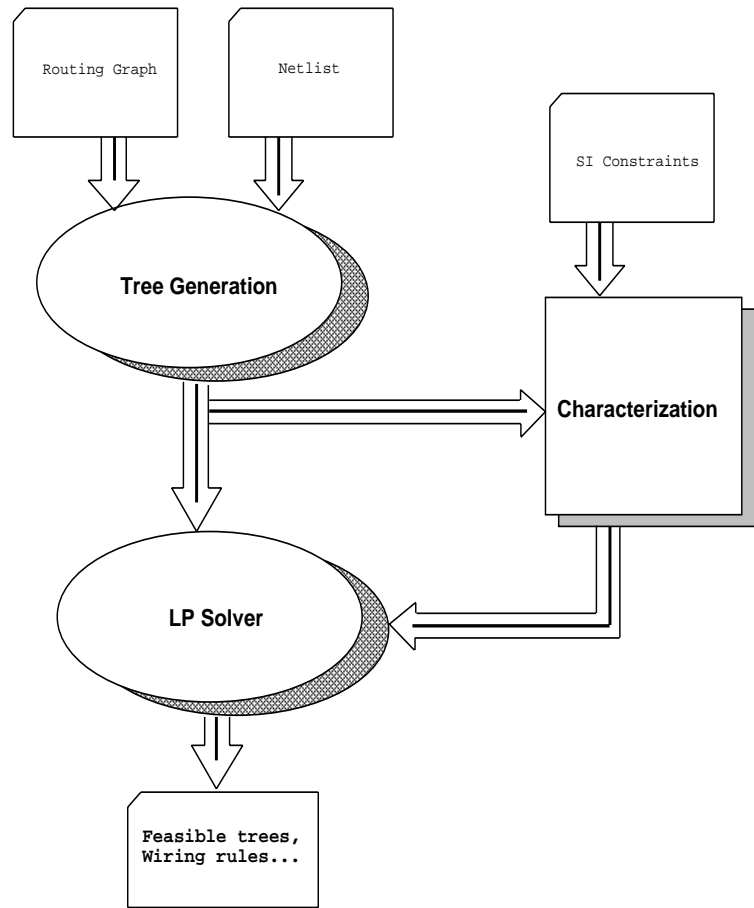


Figure 6.3: Global Router

Chapter 7

Experimental Results

The characterization and optimization methodology for high speed circuits presented in chapter 4 is heuristic. In this chapter, several circuit characterization and optimization examples are presented to establish the efficacy of this methodology, and to investigate several heuristic methods for improving the accuracy of the characterizations. The global routing and rules generation methodology presented in section 5.3 is executed on several routing examples. The aim of this investigation is to establish the utility of employing predictor function for tree screening and the effectiveness of the rule generation methodology.

7.1 Characterization Experiments

This section presents characterization results for several circuit design examples. The attempt here is to establish some of the properties of the characterization methodology.

7.1.1 Multi-Chip Module Interconnect

3 terminal net

In this characterization study, the relationship between interconnect length and signal settling time in a high speed net was studied. Figure 7.1 shows the topology of a two receiver net on a thin film MCM. The driver is a 32 mA CMOS buffer designed in the MCNC 0.8μ process. The designable parameters are the lengths of the interconnect segments in this configuration. The circuit performance was measured by the signal settling time, shown in Figure 7.2. A noise budget of 0.3V for reflection noise was chosen. Due to the lossy nature of this interconnect, the reflections from the loads and the stubs are absorbed in the line losses when the lengths get sufficiently long [18]. Hence the settling time has a highly non-linear relationship to the interconnect length.

The following inequalities describe the design space to be characterized:

$$\begin{aligned} 1 \text{ mm} &\leq l_1 \leq 10 \text{ cm} \\ 1 \text{ mm} &\leq l_2 \leq 10 \text{ cm} \\ 1 \text{ mm} &\leq l_3 \leq 10 \text{ cm} \end{aligned}$$

First, a large characterization using 1,000 sample points over a full grid in the design space was carried out, for benchmarking the results obtained from experimental characterizations. A set of several different experimental characterizations of this same net were performed using the Study Generator. The intent of this set of characterizations was to establish some properties of our sequential-experimental design, the predictor function and the error measure.

The first characterization was performed using 100 samples in the first stage and then a total of 50 samples in the second stage. Another characterization

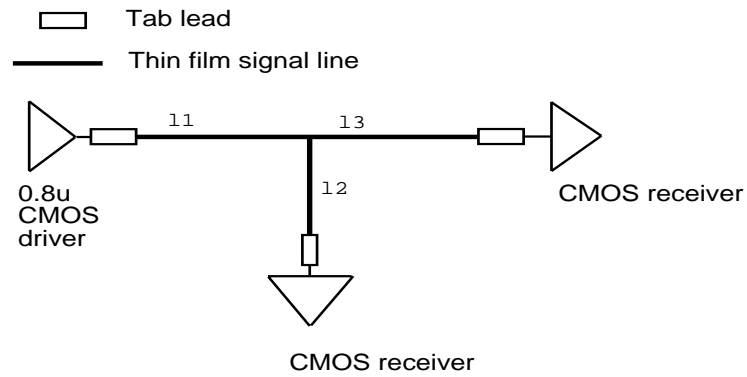


Figure 7.1: Net Topology

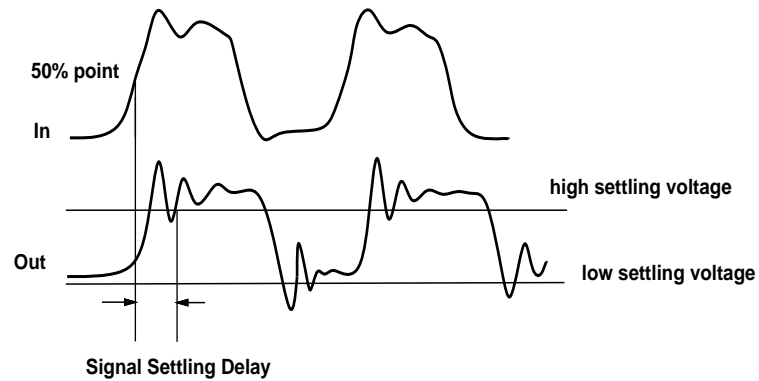


Figure 7.2: Signal Settling time

was performed using 50 samples in the first stage and 100 samples in the second stage. A third characterization was performed using 150 points in the first stage and 50 points in the second stage. A linear model was used for interpolation. Figure 7.4 shows the same response as in Figure 7.3 using the predictor function from the first characterization. Also a full quadratic model was used for interpolation in the first characterization. In each case, the responses at the 1,000 full grid points were gener-

ated using the predictor function. The error statistics when comparing the predicted to the actual response at these 1,000 points for all 3 cases are reported in Table 7.1.

Table 7.1: Error Statistics for the MCM interconnect characterization

	maximum error (ns)	mean error (ns)	error variance (ns)
100 initial points linear model	1.8	0.3	.28
50 initial linear model	2.07	.32	.31
100 initial points quadratic model	1.86	.219	.27
150 initial points linear model	1.82	.23	.27

From these results, it is apparent that the prediction error is sensitive to the division of points between the sampling stages. Very few points in the initial stage result in a poor coverage of the design space. The subsequent samples are concentrated around the points in the first sampling stage and hence it is not possible to improve the distribution of sampling sites by the resampling scheme. One way to rectify this problem would be to draw another random sample, and add it to the existing sample.

Two further characterization were carried out to assess the effect of the number of sampling stages, and the number of samples drawn in each stage, on the prediction error, and the relation of the cross validation error and the prediction error. The first characterization had 50 samples in the first stage, and 30 samples in each subsequent stage. The second had 50 samples in the first stage, and 20 samples in each subsequent stage. The rectangle based resampling technique was used. The minimum

Figure 7.3: Partial characterization of MCM interconnect. $l_1 = 3\text{cm}$

number of points in each hyperrectangle is 3. The number of hyperrectangles N_{rect} is then given as:

$$N_{rect} = d^{\lfloor \frac{\log(samples/3)}{\log(d)} \rfloor} \quad (7.1)$$

A total of 4 sampling stages were performed. Table 7.2 shows the results of error characteristics for the first characterization, and table 7.3 shows the results of the second characterization.

In both the experiments the prediction error is not improved after the first resampling. This is because the available samples for the second stage were entirely used up in distributing the sample uniformly over all the hyperrectangular partitions, and hence none were available for improving the prediction error. In

Figure 7.4: Sampled characterization MCM interconnect. $l_1 = 3\text{cm}$

Table 7.2: Error Statistics for the 3 Terminal Net Characterization

Error	Statistic	Stage 1	Stage 2	Stage 3	Stage 4
Cross Validation	max (ns)	1.15	1.22	1.35	1.37
first 50	mean (ns)	.257	.249	.239	.24
Cross Validation	max (ns)	1.15	1.22	1.31	1.73
full sample	mean (ns)	.257	.318	.317	.354
Prediction	max (ns)	1.92	1.93	1.90	1.82
	mean (ns)	.293	.308	.318	.329

Table 7.3: Error Statistics for the 3 Terminal Net Characterization

Error	Statistic	Stage 1	Stage 2	Stage 3	Stage 4
Cross Validation	max (ns)	1.15	.849	.869	.882
first 50	mean (ns)	.257	.224	.214	.215
Cross Validation	max (ns)	1.15	1.62	1.42	1.30
full sample	mean (ns)	.257	.314	.329	.294
Prediction	max (ns)	1.92	2.00	1.74	1.72
	mean (ns)	.292	.303	.291	.278

the second sampling scheme, the prediction error is significantly improved after 4 sampling stages. Also, the cross-validation error follows the trend for the prediction error. This observation is quite significant. The prediction error is **not** available for determining the efficacy of the sampling. Hence as the cross validation error is a suitable determinant of the overall prediction error, then stopping rules for resampling can be based on the cross validation error alone.

4 terminal net

This characterization example is again of a high speed net on an MCM. The topology of the net is shown in figure 7.5. The interconnect cross-section is shown in figure 7.6. The electrical parameters of interest are the 50% delay, the settling delay to 8% of the supply rails, and peak undershoot (as shown in figure 7.7) at each receiver on this net. The physical design variables are the branch lengths in this interconnect. The design space is given as:

$$\begin{aligned}
 1 \text{ mm} &\leq l_1 \leq 10 \text{ cm} \\
 1 \text{ mm} &\leq l_2 \leq 10 \text{ cm} \\
 1 \text{ mm} &\leq l_3 \leq 10 \text{ cm} \\
 1 \text{ mm} &\leq l_4 \leq 5 \text{ cm} \\
 1 \text{ mm} &\leq l_5 \leq 5 \text{ cm}
 \end{aligned}$$

For design purposes, linear basis functions were chosen for interpolation. Three designs were performed. The first had 200 samples in the first sampling stage, and another 200 in the second sampling stage using the cross-validation error measure, and defining next sampling regions in the neighborhood of points in error. The second design also had the same two sampling stages. However, the new sampling regions were defined using the rectangular partitioning described in section 4.3.2. The third design had 400 samples drawn randomly using LHS. To compare the performance of the sampling schemes, a set of 2000 samples was generated for verifying the accuracy of prediction. The response values at these 2000 points were predicted from the characterizations, and compared against the true response.

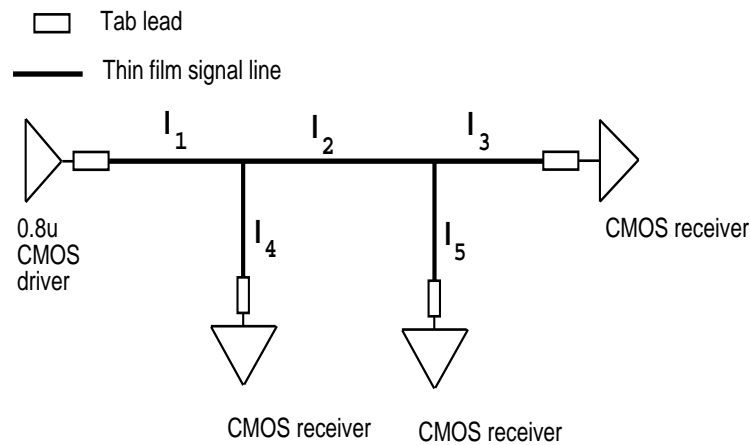


Figure 7.5: MCM Net Topology

The characterization here has multiple objective functions. Hence it is

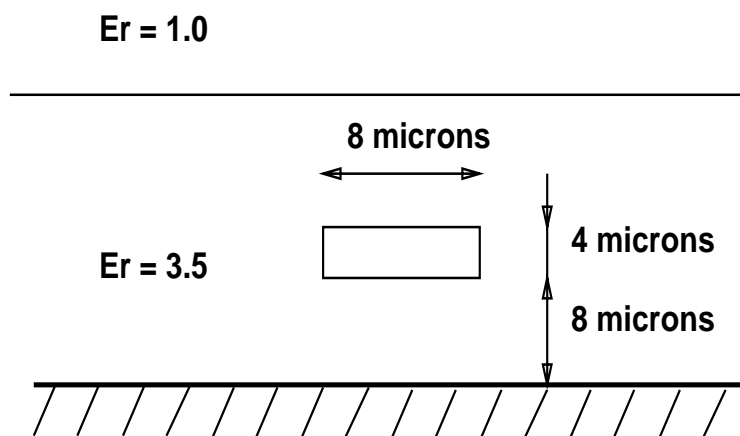


Figure 7.6: Interconnect Cross-section

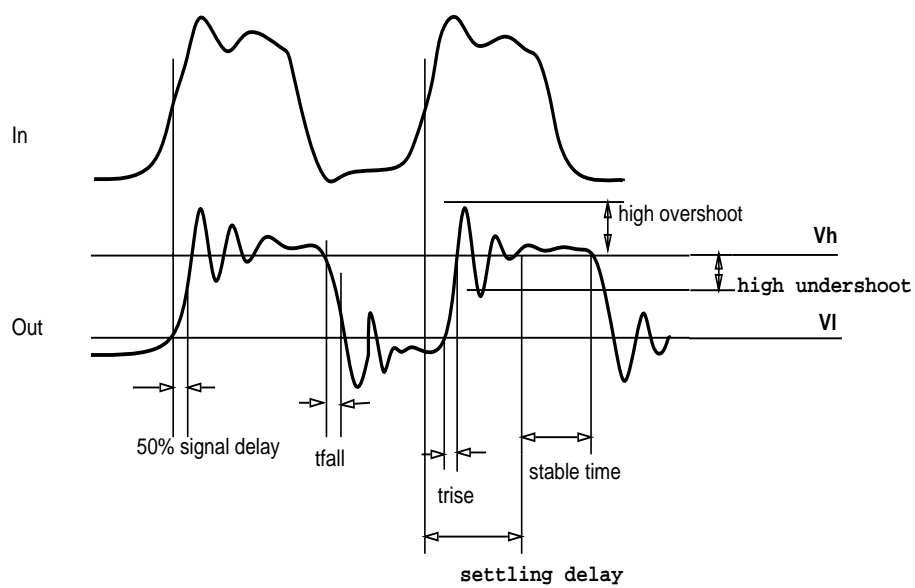


Figure 7.7: Waveform Parameters

Table 7.4: Error Statistics for the 4 Terminal Net Characterization: Delay and Settling Delay

Study	Error	Del12 (ns)	Del13 (ns)	Del14 (ns)	Stb12 (ns)	Stb13 (ns)	Stb14 (ns)
200 samples	max	1.21	1.27	1.18	2.28	2.08	2.19
	mean	.075	.114	.079	.28	.195	.25
2 stage	max	1.20	1.26	1.16	2.18	2.13	2.09
	mean	.076	.12	.085	.28	.22	.27
Rectangle	max	1.21	1.28	1.17	2.27	1.95	2.19
	mean	.076	.11	.081	.31	.24	.31
Rectangle	max	1.21	1.28	1.16	2.06	1.83	2.18
	mean	.077	.114	.081	.334	.246	.311
400 samples	max	1.22	1.31	1.17	2.2	2.08	2.50
	mean	.77	.107	.075	.235	.185	.234

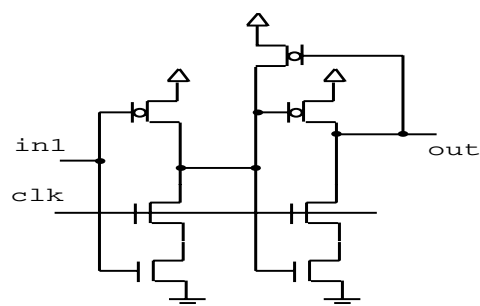
Table 7.5: Error Statistics for the 4 Terminal Net Characterization: High and Low Undershoot

Study	Error	undh2 (V)	undh3 (V)	undh4 (V)	undl2 (V)	undl3 (V)	undl4 (V)
200 samples	max	1.11	1.8	.98	1.16	1.57	1.26
	mean	.102	.114	.103	.091	.124	.126
2 stage	max	1.10	1.77	1.15	1.12	1.06	1.26
	mean	.105	.115	.105	.092	.117	.103
Rectangle	max	1.34	1.81	1.55	1.46	1.65	1.52
	mean	.105	.110	.108	.105	.125	.116
Rectangle	max	1.48	1.80	1.62	1.39	1.65	1.56
	3 stage mean	.103	.106	.103	.106	.117	.114
400 samples	max	1.28	1.77	1.76	1.32	1.14	1.21
	mean	.101	.112	.100	.087	.124	.937

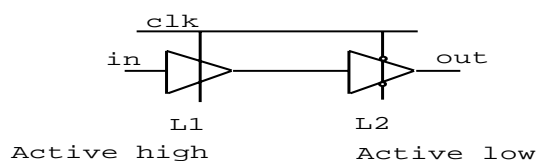
likely that errors in all the performance parameters will not be improved. Nevertheless, the two stage neighborhood sampling shows an improvement in almost all the peak error values compared to the fully random sample. The two stage rectangle based resampling is does not reduce the prediction error. Again, the rectangle based resampling is forced to distribute the points uniformly in the second sampling stage. Hence the prediction error statistics do not improve after the second stage. The third sampling stage does improve the peak prediction error statistics a little. One conclusion that can be reached here is that the two stage neighborhood sampling performs better than the two stage rectangular sampling, and the fully random sampling.

7.1.2 High Speed Data Latch

In this example, a latch structure similar to one used in the DEC Alpha chip[15] is characterized (figure 7.8). Data race through was a major concern in these latches as logic design used a single phase clock. The latch was designed using the MCNC 0.8μ process parameters with minimum size transistors, except for the weak feedback transistors which were chosen to have ten times the channel length of the other devices. The fast process corner was used to emphasize race-through. In this setting, the effect of clock rise time, data rise time, and clock skew on race-through in this latch was studied. Race-through is detected by studying the apparent delay of a signal passing through two cascaded latches. The first latch is transparent to data when the clock is high while the second is active when the clock is low. Hence data should propagate through the two latches, after the high clock period and one latch delay. With a 50% clock duty cycle, if this propagation delay is less than one half clock cycle, a race-through has occurred. Otherwise the signal is latched correctly. In general, the relationship of the signal delay to the study parameters quite difficult to model analytically.



Active High Latch



Test Circuit

Figure 7.8: Schematic of Latch Circuit

The following inequalities describe the design space to be characterized:

$$0.1 \text{ ns} \leq \text{clock rise time} \leq 1.5 \text{ ns}$$

$$0.1 \text{ ns} \leq \text{data rise time} \leq 1.5 \text{ ns}$$

$$0.1 \text{ ns} \leq \text{clock skew} \leq 1.1 \text{ ns}$$

In the experiment design, two sampling stages were used, with 50 points taken in the first stage and 35 in the second. A first order polynomial in all three variables was chosen for the interpolation. The resampling regions were defined as the neighborhood of the points in the first sample. Another separate characterization was carried out using MetaSim with a total of 500 points placed picked randomly from this design space. The predictor function was used to estimate the response at these same, based on the observations from the experiment. Figure 7.9 shows the plot of signal delay as a function of data and clock rise times, for data rise time of

0.3 ns and clock period of 5 ns with a 50% duty cycle. Figure 7.10 shows a plot of the same response, but using the predictor function. The piece-wise linear nature of the response is clearly captured by the predictor function.

Figure 7.9: Signal Delay plot for data rise time of 0.3 ns: Actual response

The error statistics, comparing the predicted to actual response are shown in Table 7.6. Error 1 is the error in estimating the responses at the 500 points with a predictor based only on the results of the first experiment. Error 2 gives the same statistics when prediction is performed using all the sample points after the 2nd experiment. Cross error 1 reports the statistics of the cross validation error on the first 50 sample points, and Cross error 2, is the error reported on all the points after resampling. First error is the cross validation error at the first 50 points using all 85 points for prediction.

Figure 7.10: Signal Delay plot for data rise time of 0.3 ns: Predicted response

Table 7.6: Error Statistics for the Latch Characterization

	maximum error (ns)	mean error (ns)	error variance (ns)
Cross error 1	4.87	.762	-
Cross error 2	4.46	.850	-
First error	4.21	.625	-
Error 1	4.71	.46	.70
Error 2	3.75	.65	.65

Figure 7.11: Scatter plot of sample points. (a) Initial Sample (b) 2nd Sample

Figure 7.11 (a) shows scatter plots of the first 50 data points in the clock skew, clock rise-time space and Figure 7.11 (b) shows the 35 data points in the second. Comparing figure Figure 7.11 (b) with Figure 7.9 clearly shows that the resampled points lie in the region where the response is highly non-linear.

From the error statistics, it is obvious that the peak error improves considerably after resampling. The value of the peak error is large. This is because of the large discontinuity in the signal delay characterization. Again, the cross validation error statistics follow the trends of the prediction error statistics fairly well. An interesting observation was made by employing the characterization for generating a simple design rule. Suppose that a delay threshold of 2.5 ns was set for determining data race-through, i.e., if the signal delay was smaller than 2.5 ns, then a race-through

was assumed to have occurred. Out of the 500 simulated points, 42 were erroneously predicted to have data race-through. After resampling this number dropped to 21. Hence accurate prediction was achieved over 96% of the design space. This suggests that the combination of the sampling scheme and predictor function is very accurate for generating design rules. An extension of this concept is the rule generation methodology presented in section 5.3. In the routing examples, the results of rule generation confirm the efficacy of this approach.

The conclusions reached from the characterization studies are as follows:

1. The two stage neighborhood sampling provides a good method for achieving accurate circuit characterizations.
2. The cross-validation error is a good measure of the predictive error. The magnitude of the cross-validation error is always smaller than that for prediction. Hence some adjustment must be made to the cross-validation error statistics in formulating a stopping criterion.
3. The total sampling capability has to be judiciously employed. If the number of samples in the first stage is small, then subsequent sampling will not improve the predictive error substantially. It is best to use a larger share of the sampling capability in the first sample to get a good coverage of the design space.

The real test of the characterization methodology is in using the results for design. Several characterizations were employed for global routing and rule generation. The high degree of success in the global routing and the accuracy of the wiring rules, suggests that the characterization methodology holds a lot of promise for high speed designs.

7.2 Optimization Experiments

In this section, several examples are presented demonstrating the power of the stochastic optimization technique presented in chapter 4 in determining good designs for difficult circuit optimization problems. The first problem is that of optimizing the transistor sizes in a combinational gate structure suitable for wave-pipelined circuits for minimum delay skew subject to a constraint on the maximum delay through the circuit. The second example is that of optimizing the transistor sizes in a clock driver circuit so as to minimize clock skew. The third example is also of transistor sizing for a bistable latch circuit to reduce metastability. The last example is of determining suitable termination values for a high speed data bus for meeting signal integrity requirements. In all these examples, the design domain is finite and the optimization objective can be computed only through expensive circuit simulations. The stochastic optimization technique helps in determining good designs through objective evaluations for only a small fraction of the feasible designs.

7.2.1 Combinational Logic Element for Wave-pipelining

The design of wave-pipelined circuits involves very careful control of the delay of each path in the combinational blocks. Techniques have been proposed by De Micheli et al. [72] for balancing the path delays by inserting active delay elements. For ECL technology, they have shown how the delay of each gate can be accurately controlled through the tail current. For CMOS gates, however, the delay is data dependent. For the CMOS NAND gate, for example, the rising delay is substantially smaller when both inputs switch from 1 to 0, as opposed to one input being fixed at 1 and the other switching from 1 to 0. One way of avoiding this data dependence is to use the cross coupled biased-CMOS NAND gate shown in figure 7.12. This gate,

however, consumes considerable static power. Another A gate structure suitable for wave pipelining is shown in figure 7.13. Here the transistor M3 is used to add extra resistance to the pull-up chain to reduce the effect of the simultaneous switching of both inputs. It also has the deleterious effect of slowing down the circuit. Hence a proper balance has to be struck between the maximum delay through the gate, as well as the data-dependent spread [47]. The delay spread has to be minimized over process variations also. Of course, the easiest parameters to control in this optimization are the transistor sizes. Hence the goal of the optimization is to obtain a suitable sizing scheme such that the delay spread through each circuit block is minimized, subject to a constraint on the maximum delay through the circuit.

The optimization problem is formalized as follows:

$$\text{Find } x^* = \text{Arg min}_{x \in A} \max_V \delta^*(x) \quad (7.2)$$

$$\text{subject to } \max_V \text{delay}(x) \leq D_{max} . \quad (7.3)$$

Here V denotes the nominal and the four process corner MOSFET models. A is the hypercube formed by restricting the widths of M1-M3 between $3.6\mu\text{m}$ and $10\mu\text{m}$, and V_{bias} between 0.0 and 2.0 V. The widths of N1 and N2 are constrained to be one-half the width of M1 and M2 respectively. x is an arbitrary vector of feasible transistor widths and bias voltage. Note that the minimum allowed feature size is $0.6\mu\text{m}$ and hence the widths of N1 and N2 were restricted to vary in quanta of $0.6\mu\text{m}$ only. The transistor lengths were kept at their minimum permitted value. The skew $\delta^*(x)$ is defined as the variation in delay through the circuit shown in figure 7.13 over the six possible input transitions (see figure 7.14), and the $\text{delay}(x)$ is the largest delay over these input transitions. D_{max} is the maximum delay constraint, which was 1 ns for this example.

The models for delay and skew were initially established by simulating $k = 100$ different sizing schemes, selected randomly using Latin Hypercube Sampling [45]. Each sizing scheme was simulated for the four process corners and the nominal process. The worst delay over the six input transitions, and the skew over the six transitions was extracted from the simulation results. Separate models were built for the worst values of skew and worst data dependent delay over the design space of transistor sizes. The first row of table 1 shows the sizing scheme with the best skew value, satisfying the delay constraint, among these 100 points. This sizing scheme is not feasible since these sizes are not permitted by the design technology. The second row shows the nearest feasible point to this sizing and the delay and skew value for that circuit.

Since the number of possible sizings is small, all the feasible alternatives (216 distinct sizing schemes) were evaluated with five values of bias voltage ranging from 0 to 2.0V using equations B.4 and B.6. This constitutes an exhaustive search of the design space using the models. Since the smallest possible value for the skew is 0, y_{ok} was chosen to be zero. The delay constraint was checked for using the mean of the predictor function only. This is equivalent to setting the threshold C to 0.5 in equation 4.4.3 in section 4.4.3. 10 feasible sizes and bias voltages with the largest probability that satisfied the delay constraint $D_{max} \leq 1$ ns were chosen for resimulation. Table 1 shows the results for these sets of simulations. The best sizing in the second set was considerably better than the results of the first 100 samples and was considered quite suitable for design and hence no further simulations were performed. The total time taken for simulation was 540 cpu seconds on a DECstation 5000 while the overhead of model building and searching was less than 1 cpu second.

This example illustrates how the optimization procedure is employed. The search space is pruned by the designer's judgement and a good solution is found with

very few simulations. In the next example, the methodology is further expanded to include continuous variables with a very different objective formulation.

Table 7.7: Results for Delay Controlled Element

	M1 (μm)	M2 (μm)	M3 (μm)	vbias (V)	delay (ns)	skew (ns)
Best Random Sizing	6.15	8.79	8.55	.98	.99	.77
Closest Feasible Sizing	6.0	8.4	8.4	.98	1.0	.77
Best Sizing after Optimization	7.2	8.4	9.6	0.0	.88	.54

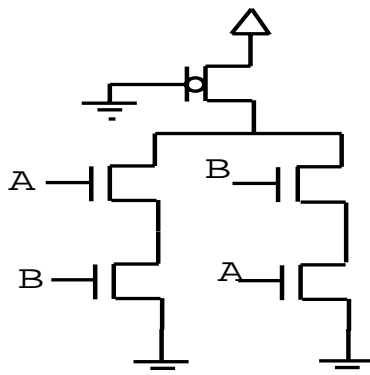
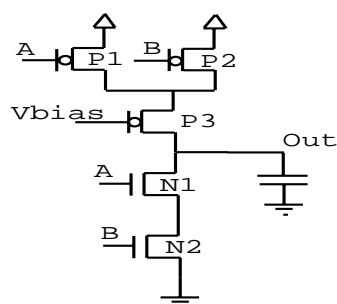


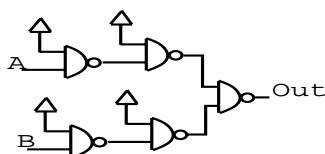
Figure 7.12: Cross coupled NAND gate

7.2.2 Clock Driver Circuit

The second example is that of skew optimization of a single to differential input clock driver circuit shown in figure 7.16. Such a clock driver is used for a two phased latching scheme shown in figure 7.15. The latches are transparent, during the high or the low period of the clock. It is desirable then to have no skew between the clock signals



Delay Controlled Circuit Element



Test Circuit

Figure 7.13: Circuit Block for Wave-pipelining

as skew effectively reduces the clock period. Hence it is desired to obtain a signal and its complement from this circuit's outputs such that there is minimum skew between the two signals, i.e. to minimize δ^* (see figure 7.16) which is the maximum of the high and low skew between the clock signal and its complement. This skew has to be minimized over the process variations. This optimization has to be done using a suitable transistor sizing scheme. The absolute delay through this circuit is not a concern, hence the optimization is essentially unconstrained. Temperature and power supply variations were also considered. The model was built over the space of transistor sizes, process, temperature and power supply variations. As in the previous example, process variations were considered by simulating each sizing scheme over the 4 process corners and the nominal process.

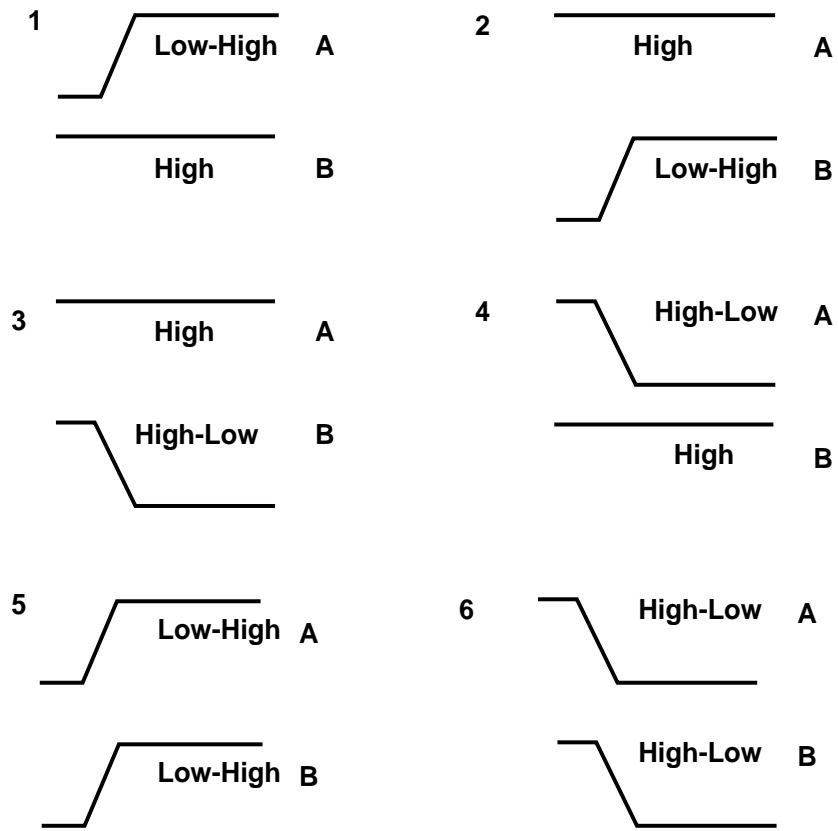


Figure 7.14: Possible input transitions

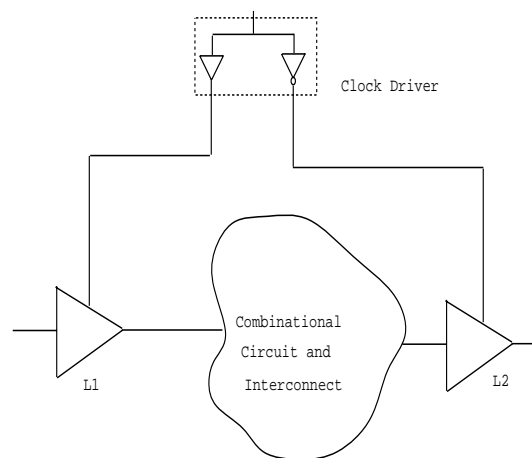


Figure 7.15: 2 phase latching scheme

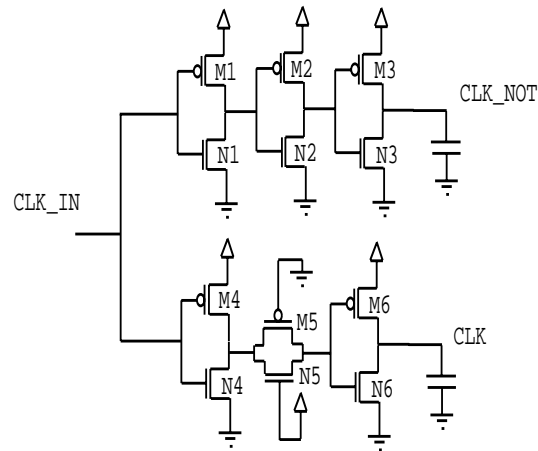


Figure 7.16: Clock Driver Circuit

Table 7.8: Results for Clock Driver Circuit

	M1 (μm)	M2 (μm)	M3 (μm)	M4 (μm)	M5 (μm)	M6 (μm)	skew (ns)
Designer's choice	9.6	4.8	4.8	9.6	4.8	4.8	.29
Optimal Point	7.2	3.6	8.4	9.6	9.6	4.8	.11

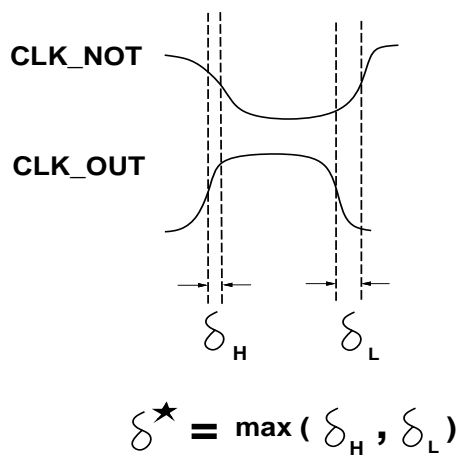


Figure 7.17: Skew Definition

The problem is formalized as follows:

$$\text{Find } x^* = \text{Arg min}_{x \in A_w} \max_{E; \max_V} \delta^*(x) \quad (7.4)$$

Here, A_w is the hypercube formed by restricting the widths P1-P6 between 3.6 μm to 12 μm , and E represents the temperature variation between 25-75° C and V_{dd} between 4.75-5.25 V. As before the widths of N1-N6 are constrained to be one-half the widths of P1-P6 respectively. V represents the process variations considered. For this problem, the sizing provided by a circuit design expert had a worst case skew of 290 ps (row 1 of table 7.8). For optimization purposes, the worst process dependent skew model was built using $k = 100$ sizing scheme, selected randomly. The model was of the worst data skew over the process variations. Power supply and temperatures were considered model variables. For optimization purposes, the effect of temperature and supply variation was first factored out. To do this, 1000 random points were sampled in (A_w) . At each of these 1000 points, the *model* was evaluated for 9 different combinations of the supply voltage and temperature variations. The smallest value of the probability $P(y_{ok})$ (equation 4.17) over these 9 combinations was found for each of the 1000 points. This value was used to estimate the likelihood of a sizing being the best. i.e. :

$$x^* = \text{Arg max}_{x \in A_w} \min_E P_{x_w}(y_{ok}) \quad (7.5)$$

was the target for further simulation. Again y_{ok} was chosen to be 0.0 which is the minimum possible value of the skew. From the 1000 sizing schemes evaluated, the 10 sizing schemes (instead of only one as suggested by equation 7.5) with the largest probability were chosen for further simulation. These schemes were verified using the 5 process parameters and the 4 corners of power supply and temperature fluctuations. The smallest worst-case skew among these sizings was only 110 ps, a significant improvement over the expert's design (row 2 of table 7.8). The total simulation time

was 640 cpu seconds on a DECstation 5000 and the overhead of model building and optimization was less than 10 cpu seconds.

7.2.3 Bistable Latch

The next design example is a bistable latch circuit shown in figure 7.18. This latch structure is used for a wave-pipelined sampler circuit [47]. This latch samples the differential data, in and inb with a single ended clock clk . When clk is high, the latch is transparent with x_{out} and x_{outb} following in and inb . The primary consideration for this latch design is metastability. To reduce metastability problems a reasonably fast transition time is needed to lessen the impact of noise on the circuit performance. The circuit performance can be improved by transistor sizing. Again, the design has to be made robust to process variations. From a design standpoint, the sizes of transistor m1, m2, m6, m7 are kept the same for symmetry, and so are transistor m4 and m5, and transistor m10 and m11. The size of transistor m0 is also variable. So there are a total of four independent transistor sizes to be varied. The sizes of the inverters are fixed.

The speed of the latch has to be optimized. The latch transitions are shown in figure 7.19. The objective function is defined as the maximum of the delays t_{lh} , t_{hl} , t_{inv_hl} and t_{inv_lh} . This delay has to be optimized over the process variations using a suitable sizing for the transistor sizes, w_1 to w_4 . The problem is formalized as follows:

$$\text{Find } x^* = \text{Arg min}_{x \in A_w} \max_V t^*(x)$$

Where t^* is the maximum delay defined above. A_w is $[1.2\mu, 1.8\mu, \dots, 9.0\mu]^4$. All simulations were restricted to be from this grid. V is again the four process corners and the nominal process model for the transistors. The waveforms for the design used in Gray et. al. [47] with the nominal process are shown in figure 7.20. To build the model, 100 random samples were drawn from A_w and simulated to determine the worst t^* over the process conditions. The best t^* and transistor widths from this set

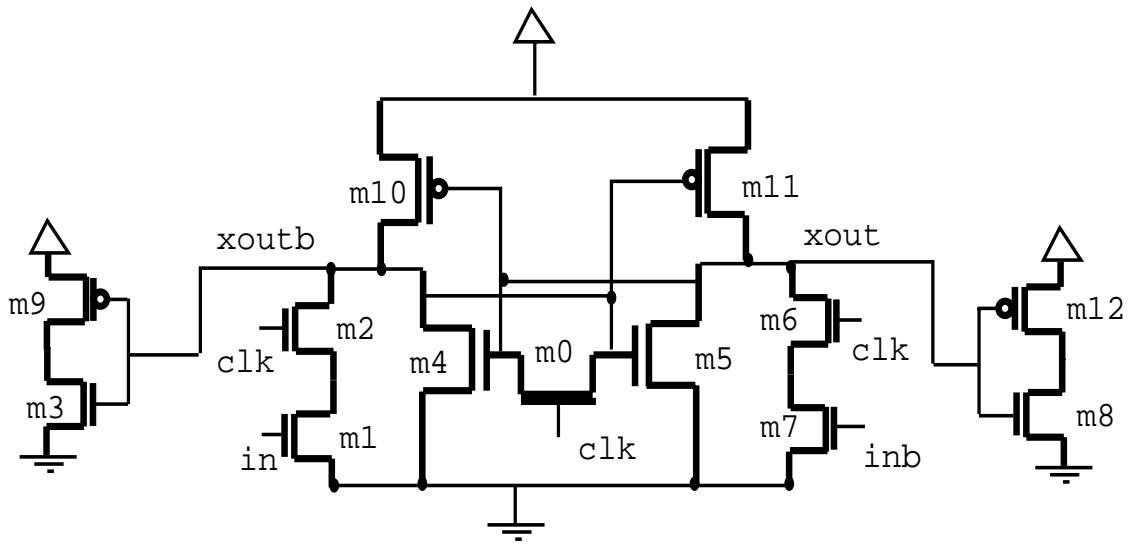
are shown in table 7.9. 2000 random samples were drawn from A_w and the t^* value was predicted from the model. The threshold y_{ok} was set to 0.8 ns. The 10 designs with the largest probability of having skew lesser than this threshold were simulated. The performance of the best design among these is shown in 7.9. The waveforms from simulations with the nominal process parameters are shown in figure 7.21. There is a considerable improvement in design performance with the stochastic optimization methodology.

Table 7.9: Results for Bistable Latch

	W1 (μm)	W2 (μm)	W3 (μm)	W4 (μm)	t^* (ns)
Best Random Sizing	1.8	8.4	4.8	4.2	1.69
Optimal Point	1.2	9.0	9.0	2.4	1.51

7.2.4 High Speed Memory Bus

The design examples discussed above were all transistor sizing problems. The optimization technique is not limited to transistor sizing only. It is applicable in general to any design problem which has a finite design space, and the objective is evaluated only through circuit simulation. One such design problem is the optimization of a high speed bus on a backplane. This design example was provided by IBM [62]. The network topology is shown in figure 7.22. Due to the structure of the backplane, the trace lengths are fixed. There are fixed positions available for inserting damping resistors. The large distributed discontinuities and the bidirectional nature of the driver-receiver pairs, makes this an extremely hard optimization problem. The optimization problem was of minimizing the settling delay on the bus at all the receivers. The optimization variables are the values of the terminating resistors. The symmetry inherent in the bus structure allows the collapsing of the resistor values into 3



$$\begin{aligned}
 m0 \blacksquare w1 \quad [m10, m11] \blacksquare w2 \\
 [m1, m2, m6, m7] \blacksquare w3 \quad [m4, m5] \blacksquare w4
 \end{aligned}$$

Figure 7.18: Bistable Latch Circuit

variables as shown in figure 7.22. The resistor values have to be chosen from a set of standard values.

The optimization problem is formulated as follows:

$$\min_x \zeta(x) \quad x \in A \subset R^d$$

where

$$x = (R_1, R_2, R_3)$$

and

$$A = [5, 10, 22, 33, 39, 47, 56]^3$$

$$\zeta(x) = \max_d \max_r t_{settle}^{dr}(x)$$

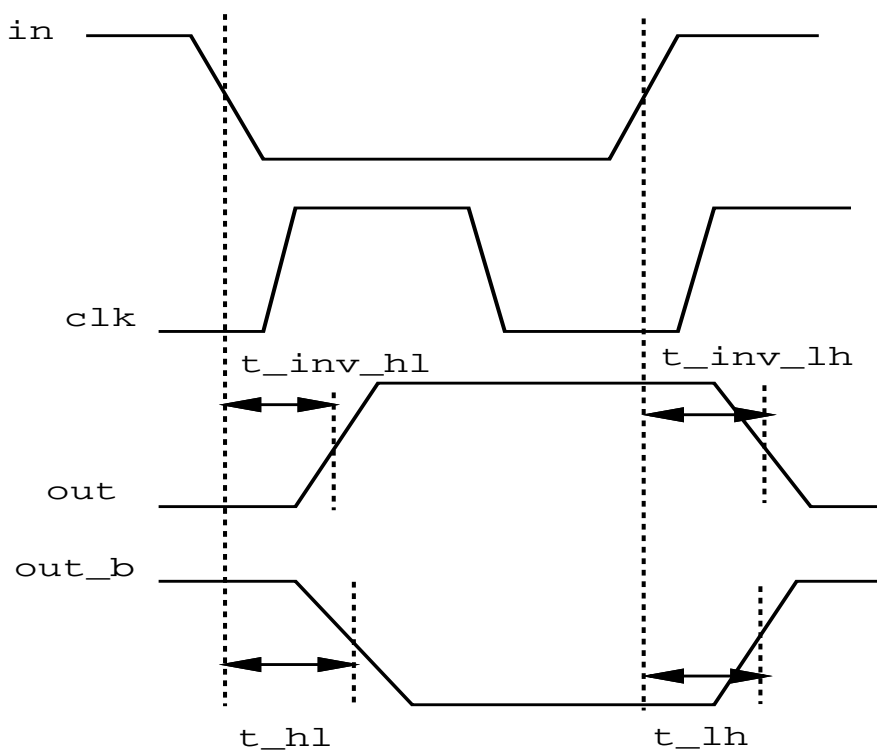


Figure 7.19: Latch Waveforms

Where d and r are the driver and receiver locations. It is impossible to treat driver and receiver values as variables in the optimization. Hence each objective value computation requires a separate simulation for 4 driver locations. (The other locations are symmetric to these). Even though the design space is very small (343 distinct resistance combinations), the objective function is very expensive to evaluate. Hence it is imperative to keep objective evaluations as few as possible.

Initially the model was built using 25 objective function evaluations, for design points chosen randomly. The smallest objective function value from this set was 17.74 ns. All the simulated design points and objective function values are shown in table 7.10. The design space was searched exhaustively using the predictor functions. The smallest predicted mean value was 8.23 ns. Hence y_{ok} was set to 7 ns.

Figure 7.20: Latch Waveforms for design in Sampler

Seven designs with the largest probability of function value below this threshold were chosen for simulation. There was no improvement in the best function value after this step. However, the predicted values were drastically different, indicating that there was considerable uncertainty near the newly generated designs. The smallest predicted value over the whole design space was 1.27 ns. Hence the threshold y_{ok} was reduced to zero. 5 new designs were located with a large probability of function value below this threshold. The best function value improved to 17.03 ns. The smallest

Figure 7.21: Latch Waveforms for Optimal Point

predicted value from this data set went up to 10.1 ns indicating that the design space was quite well explored. A threshold of 5 ns was chosen. 8 points were identified with large probabilities of exceeding this threshold. Of these 4 points were chosen randomly, so that the frequency of a given value for a certain variable was the same among in both sets. For example, $R1$ had a value of 33σ in four of the designs, hence it was given this value in two of the evaluated designs. These four designs were simulated, but there was no improvement in the best data value. When function

values were predicted with the using the 41 designs simulated thus far, only 3 other designs has a smaller predicted value than the best data value of 17.03ns. The one with the smallest predicted value was simulated, but the true value was 18.31 ns. No predicted value was smaller that the best data after this. Hence the optimal value was perceived to be attained.

The total number of designs simulated was 42 out of a possible 343. Hence the optimal value was achieved by exploring on 12.2% of the total designs, for a problem where very little knowledge about the objective function was available. This demonstrates the power of this technique and sets a standard for the type of design problems where ist is most applicable. The objective function here is very expensive to evaluate, the design domain is finite but cannot be explored exhaustively. The stochastic prediction provides a very suitable guidance in this scenario for identifying good designs, and exploring the design space thoroughly, to ensure local minima are captured.

7.3 Global Routing and Wiring Rule Generation

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 [28]. 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

Table 7.10: Simulated Points for PCB Bus

R1 (Ω)	R2 (Ω)	R3 (Ω)	settling delay (ns)
39	39	10	18.98
5	33	47	44.85
39	22	39	19.45
56	22	22	22.70
10	39	47	21.98
47	22	10	44.49
10	22	39	44.40
39	33	56	44.26
10	39	39	20.64
33	39	22	18.34
22	5	47	45.75
22	39	10	45.71
22	10	33	45.63
10	39	22	20.62
5	5	33	39.40
47	10	39	43.92
33	33	39	19.37
39	33	33	17.74
56	10	56	43.94
5	33	39	31.88
10	22	33	44.75
33	56	56	44.87
5	39	22	45.70
47	22	22	19.00
56	5	5	46.26
39	33	22	17.74
39	39	22	44.27
39	47	22	22.44
47	39	22	20.14
47	47	22	23.04
56	39	22	22.74
56	47	22	23.92
33	33	22	17.03
33	56	10	25.95
47	33	22	19.03
33	56	5	26.01
33	47	22	22.00
39	47	10	22.51
39	33	10	17.75
47	39	5	20.11
33	39	5	18.50
33	39	10	18.31

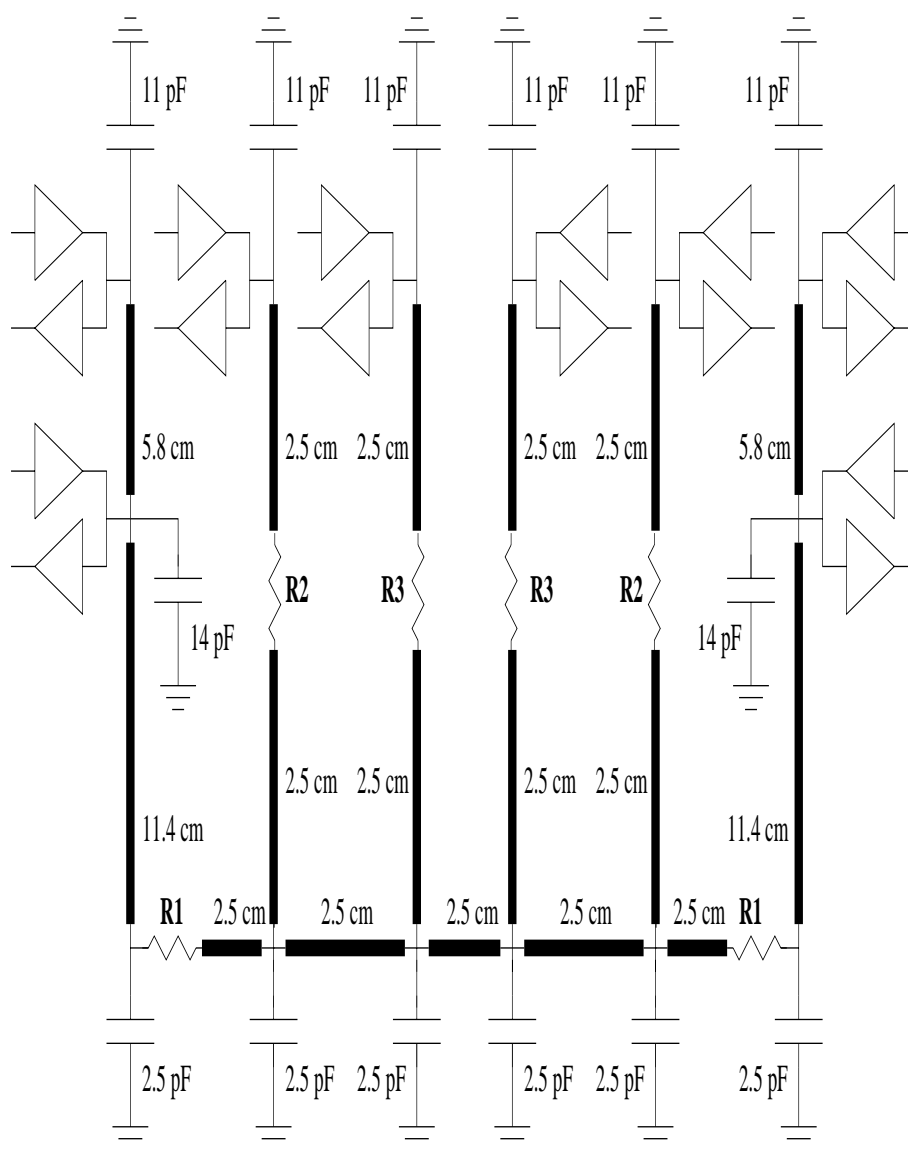


Figure 7.22: Circuit model for PCB Bus

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 of evaluation as to how well the global routing procedure is able handle performance constraints.

7.3.1 Generation of Timing Constraints

For the two MCC designs no explicit pin to pin timing information is given. In fact, no information about driver and receiver circuits, package parasitics, and interconnect parameters is available. Hence somehow, a reasonable set of timing constraints need to be generated in order to demonstrate the utility of the global routing and characterization methodology.

It is known from industry sources, that timing constraints for CMOS circuits usually have a lognormal distribution [10]. Figure 7.23 shows two curves. One is the distribution of the actual timing slacks for pin to pin connections. The other is that of the minimum required timing slacks based on the placement. Note that the peak of the available timing slack curve should always be to the right of the required slack curve for the design to be feasible. The closer these peaks are to each other, the tighter the design is.

Figure 7.24 shows the distribution of the shortest path lengths connecting the nets for the first design example given here, and it looks very much like a lognormal distribution. Based on this observation, the following methodology for generation timing constraints for the two MCC design examples was adopted:

1. The distribution of the minimum required timing constraints was generated by assuming certain driver and receiver models, package parasitics and interconnect parameters, and determining the shortest paths in the global routing graph connecting each of the nets in the design. This distribution is assumed to be lognormal.
2. The first (*a*) and the ninety-ninth (*b*) percentile of the distribution are determined.

3. These percentiles are multiplied by derating factors to give the distribution of timing slacks.
4. N random samples are drawn from this new distribution, where N is the total number of nets in the design.
5. For each net in the design, it's percentile in the distribution of minimum required timing constraints is determined. Then a value from the same percentile in the new distribution is generated, and assigned as the allowed timing slack for that net.

In this way, timing constraints consistent with the placement are generated for each design. The tightness of these constraints is controlled by the derating factors for a and b .

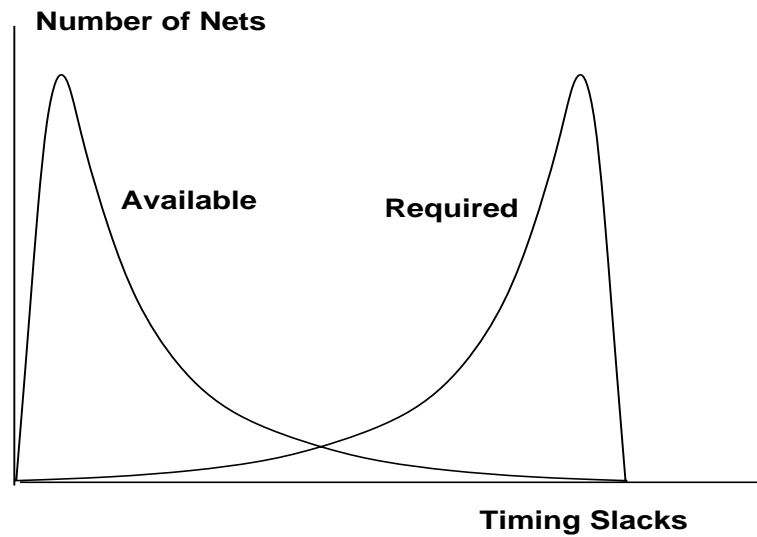


Figure 7.23: Distribution of timing slacks for CMOS systems

Figure 7.24: Distribution of shortest path lengths for MCC1

7.3.2 MCC Design Example 1.

This routing example models a next generation supercomputer on a 6 x 6 inch substrate with 37 gate arrays chips and 18 high density connectors. The chips are 1.5 x 1.5 cm. with 35 mil TAB leads on a 4 mil pitch. The connectors are placed around the perimeter of the 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 7.25.

Two graph models were generated for this design. The first mapped each chip and edge connector to one vertex in the channel graph. This graph has a total of 64 vertices, 112 edges and 307 supernets. The netlist for this model is given in Appendix A.1.1. This channel graph is shown in figure 7.26.

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. The netlist for this model is given in Appendix A.1.2. This channel graph is shown in figure 7.27.

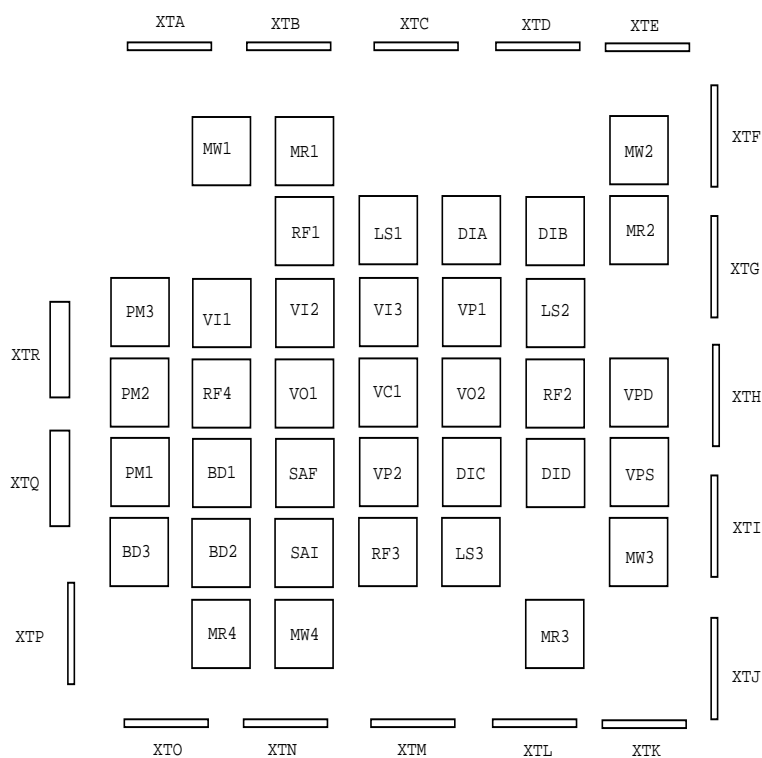


Figure 7.25: Placement for MCC1

Almost all nets in this examples have only two or three terminals. No information on 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 as shown in figure 7.6, the parameters for which are from the AT&T process. The package parasitic values, and equivalent circuits for the drivers and receivers are given in the CaZm File Generator [60] format circuit description of a point to point net in Appendix A.3.

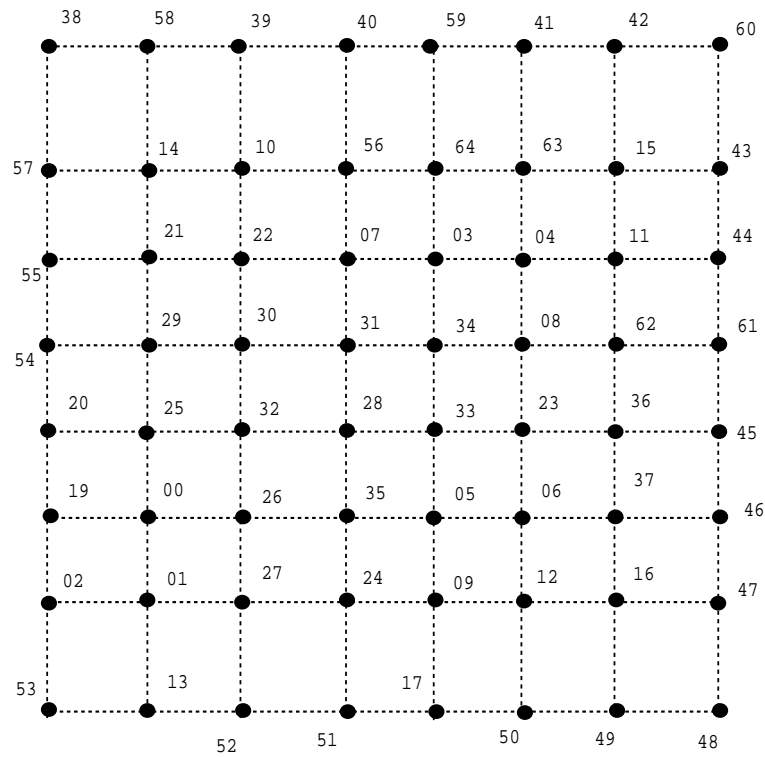


Figure 7.26: Coarse Channel Graph for MCC1

The delay constraints were assumed to be for settling delay to within 8% of the supply voltage. The timing constraints were generated as described in section 7.3.1. Settling delay for the two and three terminal nets were characterized using 75 and 150 samples respectively. The design space for the two terminal nets was:

$$0 \text{ cm} \leq l \leq 20 \text{ cm} \quad (7.6)$$

The design space for the 3 terminal nets was:

$$\begin{aligned} 0 \text{ cm} &\leq l_1 \leq 20 \text{ cm} \\ 0 \text{ cm} &\leq l_2 \leq 5 \text{ cm} \\ 0 \text{ cm} &\leq l_3 \leq 20 \text{ cm} \end{aligned}$$

Several routing experiments were run by varying the following parameters:

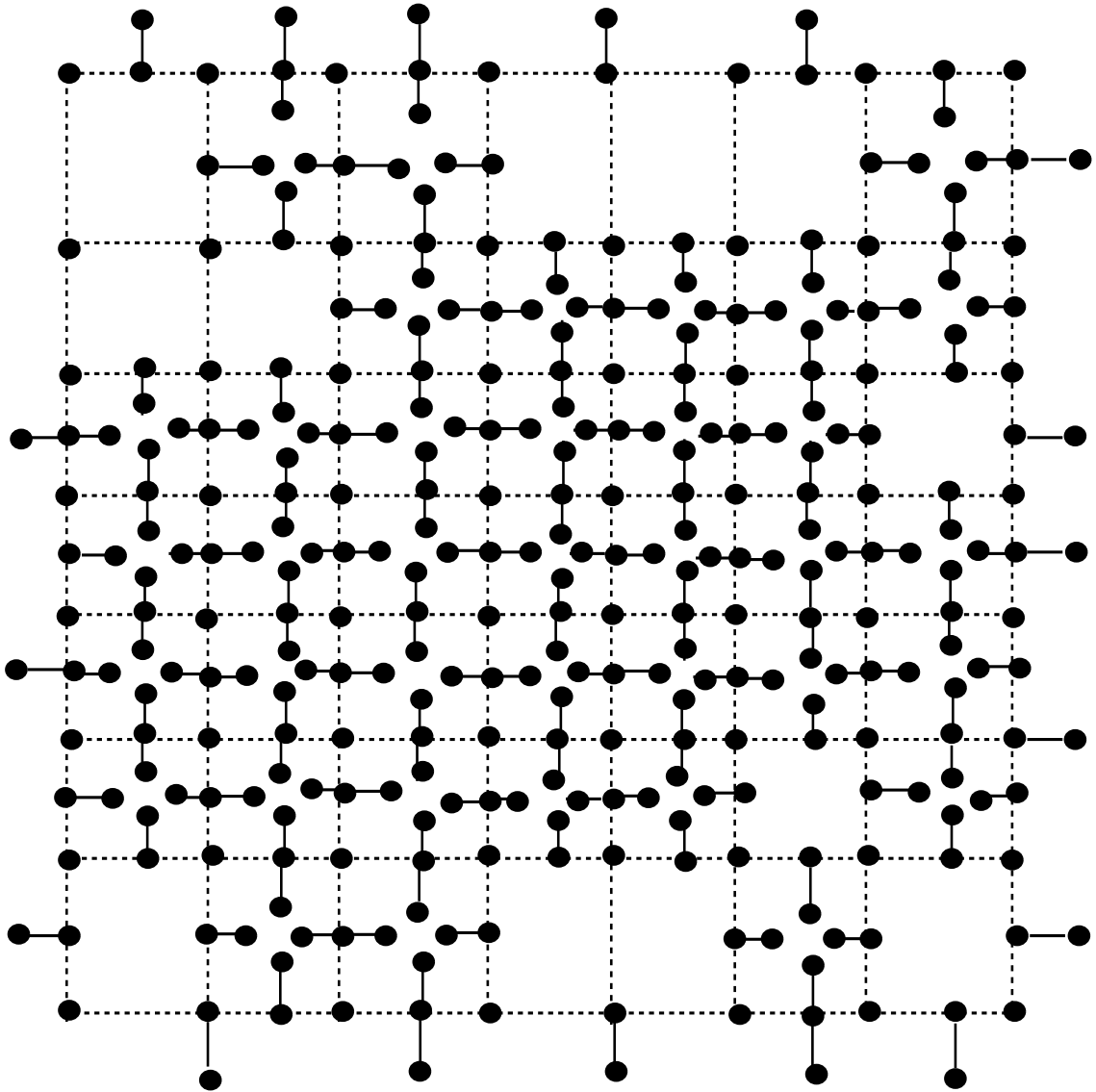


Figure 7.27: Fine Channel Graph for MCC1

1. Graph Model:

- (a) Coarse: The Coarse Graph model shown in figure 7.26
- (b) Fine: the Fine Graph model shown in figure 7.27

2. Routing Trees

- (a) SP: Shortest paths for point to point nets

- (b) C: Daisy chains
- (c) P: permutations of receivers on the daisy chain.
- (d) B: Daisy chain with stubs
- (e) ST: Steiner tree

3. Benefit Function

- (a) ABS: Benefit function shown in figure 5.6
- (b) PROB: Benefit function shown in figure 5.7

4. Channel Capacities

5. Timing Slacks

- (a) a: relaxation factor for the 1st percentile in the lognormal distribution
- (b) b : relaxation factor for the 99th percentile in the lognormal distribution

Table 7.11 gives a description of 10 different studies performed for this design. Table 7.12 shows the results for these routing studies. The second 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 third column shows the number of nets for which the simulated delay met the constraints. The total number of nets is 7114.

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. Comparing experiments 6 and 7, it seems that using steiner trees and stubs on the daisy chains did not improve the routing quality. This is because the timing constraints were very tight, and any attempt to trade-off wirelength for delay

Table 7.11: Routing Experiments for MCC1

Expt. No.	Model	Slacks		Penalty	Capacity	Trees	
		a	b			PTP	3T
1	Coarse	2	2	ABS	1000	3SP	2C 2P
2	Coarse	1	2	ABS	1000	3SP	2C 2P
3	Coarse	2	2	ABS	600	3SP	2C 2P
4	Coarse	1	2	ABS	600	3SP	2C 2P
5	Coarse	1	2	PROB	600	3SP	2C 2P
6	Coarse	1	2	PROB	500	3SP	2C 2P
7	Coarse	1	2	PROB	500	6SP	ST, 2B
8	Coarse	1	2	PROB	450	6SP	ST, 2B
9	Fine	1	2	PROB	500,140,300	2SP	2C
10	Fine	1	2	ABS	500,140,300	2SP	2C

Table 7.12: Routing Results for MCC1

Experiment	Routing Objective	Successful Nets
1	7108	7102
2	7104.1	7102
3	7109.2	7102
4	7104.1	7102
5	4808.3	7104
6	4803.9	7104
7	4810.3	6977
8	4810.3	6977
9	4665.3	7085
10	7093.5	7086

and noise fails. Comparing experiments 9 and 10, the two different benefit functions performed about the same. Hence the extra pessimism of the benefit function PROB was not necessary.

7.3.3 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.

There are two chip footprints: 550 x 550 mils with 448 pins and 330 x 330 mils with 272 pins. The SBSTRAT footprint distributes the I/O pins around the perimeter of the substrate. This example has them fixed on a 1.77 x 1.51 inch substrate.

Figure 7.28 shows the placement for this example. The channel graph for this placement is shown in figure 7.29. This graph has a total of 69 vertices and 78 edges. The number of super nets is 119. Note that the edges of the substrate have multiple nodes assigned to them. This is done to get a better estimate of net lengths for global routing. The routing problem is particularly difficult because of the multi-pin nets in the design. There are six net classes:

1. two pin nets: 61 supernets
2. three pin nets: 23 supernets
3. four pin nets: 14 supernets
4. five pin nets: 3 supernets
5. six pin nets: 4 supernets
6. seven pin nets: 13 supernets

Again, no information regarding interconnect parameters, package para-

itics or driver/receiver models is available for this design. The models used were the same as those for the first example.

The channel capacities in this layout are hard to estimate. A first routing was performed with large capacities assigned to each channel, and a routing with the maximum benefit was found. The maximum channel utilization for this routing was used as capacities in the follow on designs.

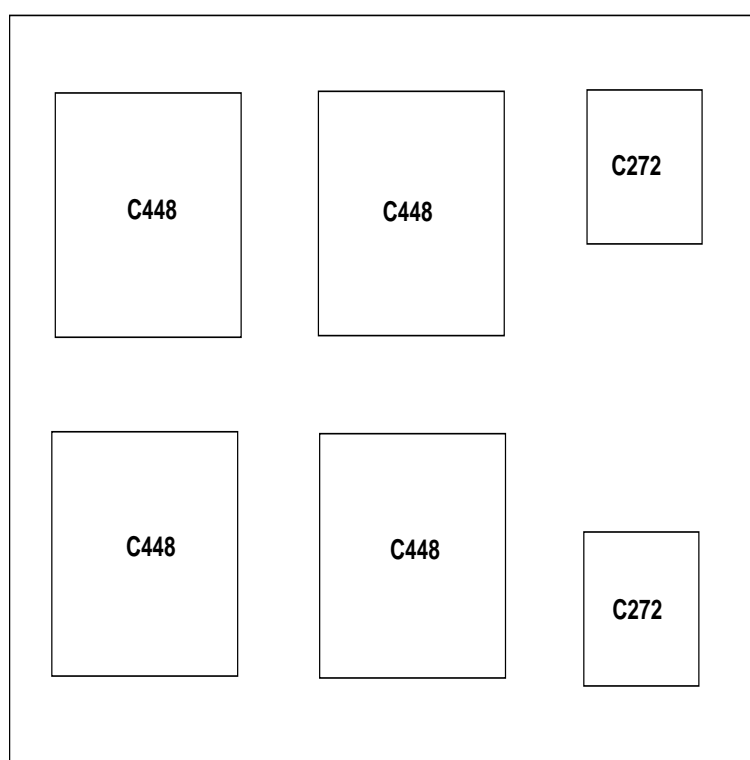


Figure 7.28: Placement for MCC2

Several routing experiments were performed using this design also, as described in table 7.13.

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

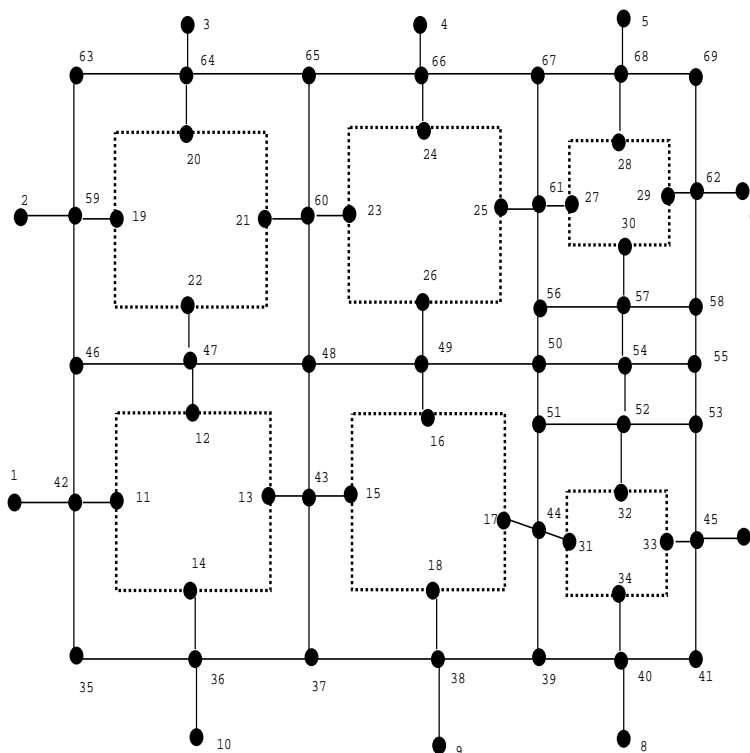


Figure 7.29: Channel Graph for MCC2

Table 7.13: Routing Experiments for MCC2

No.	Slacks		Capacity	Penalty	Trees		
	a	b			PTP	3,4 T	5,6,7 T
1	2.0	2.0	300	ABS	2SP	C,2P	C,4P
2	3.0	3.0	300	ABS	2SP	C,2P	C,4P
3	2.0	2.0	300	ABS	2SP	B,2P	C,4P
4	3.0	3.0	300	ABS	2SP	B,2P	C,4P
5	3.0	3.0	240	ABS	2SP	B,2P	C,4P
6	2.0	2.0	240	PROB	2SP	B,2P	C,4P

Table 7.14: Routing Results for MCC2

Experiment	Routing Objective	Successful Nets
1	767.25	734
2	786.32	760
3	765.69	753
4	796.0	793
5	796.0	793
6	759.9	753

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, as is evidenced in comparing the results of experiment 2 and 4. However, the time taken for characterization increases considerably, as each stub introduces an extra characterization variable. Hence the stubs were introduced only for the 3 and 4 terminal nets, where the large number of nets justified the extra simulations required to characterize the effect of stubs on the routing. Note that the routing results and the objective function track fairly well, again indicating the usefulness of the characterization in measuring the performance of the design. Also, note how the trade-offs between the tightness of timing constraints, and the required routing resources can be explored easily and efficiently through the global routing procedure.

7.3.4 Intel Pentium Board Design

The last design example is of the Intel Pentium Board Design. Figure 7.30 shows the component placement for the PCIset ISA Reference Design PCB Layout. 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. Design guidelines for this board have been published [28]. The board does not have any congestion problem. Hence this design will be used solely to illustrate the efficiency of the tree generation and the rule generation procedure. The channel graph for this board placement is shown in figure 7.31.

The Intel EZ-Route Layout Guidelines are published in [28]. These layout guidelines list a graphical description of the trees to be used for routing the board, bounds on the lengths on the branches in these trees, and allowed flight times for the signals. The driver-receiver models are specified using the IBIS standard. A range of assumed interconnect models is given. Though the design information is quite complete, it is unclear as to which set of interconnect, driver/receiver and parasitic models were used to derive the wiring constraints.

The aim of this design study is to compare the effectiveness of the routing procedure and the wiring rule generation procedure. Physical design guidelines are given for this board, though the information about the driver and receiver models is uncertain. Hence the following approach was taken for this design:

1. A set of driver/receiver models, interconnect models and parasitic models was assumed.
2. The given physical design constraints were treated as the experimental design region.
3. The given electrical constraints were ignored, since precise information about the interconnect parameters, package parasitics and driver/receiver models used to generate these was not available.

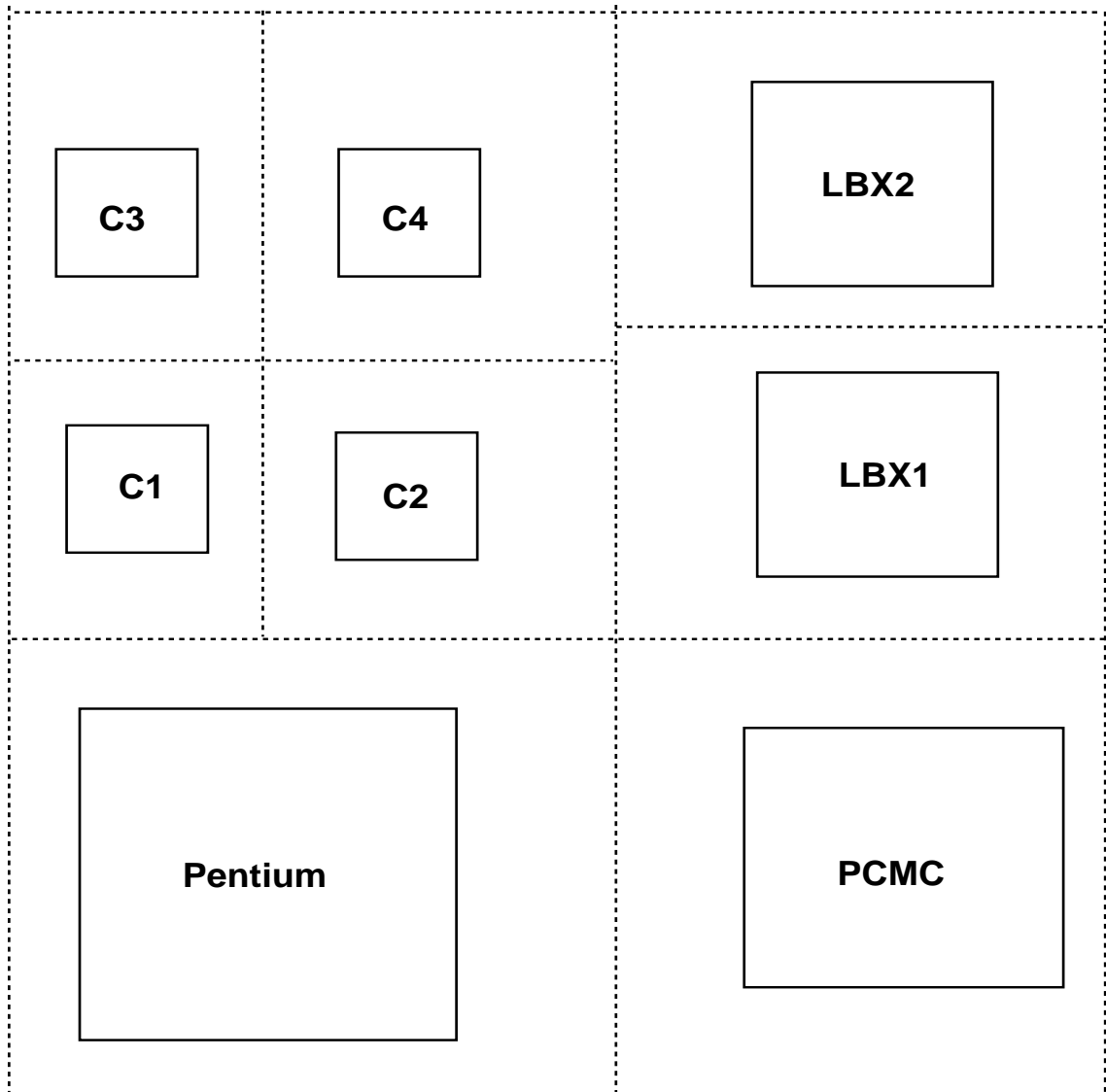


Figure 7.30: Placement for Intel Pentium Board

4. The interconnect structures were simulated over the experimental design region.
5. The maximum values for the electrical parameters over the experimental region were then treated as constraints on the electrical performance.
6. The global routing/wiring rule generation procedure was then executed on this

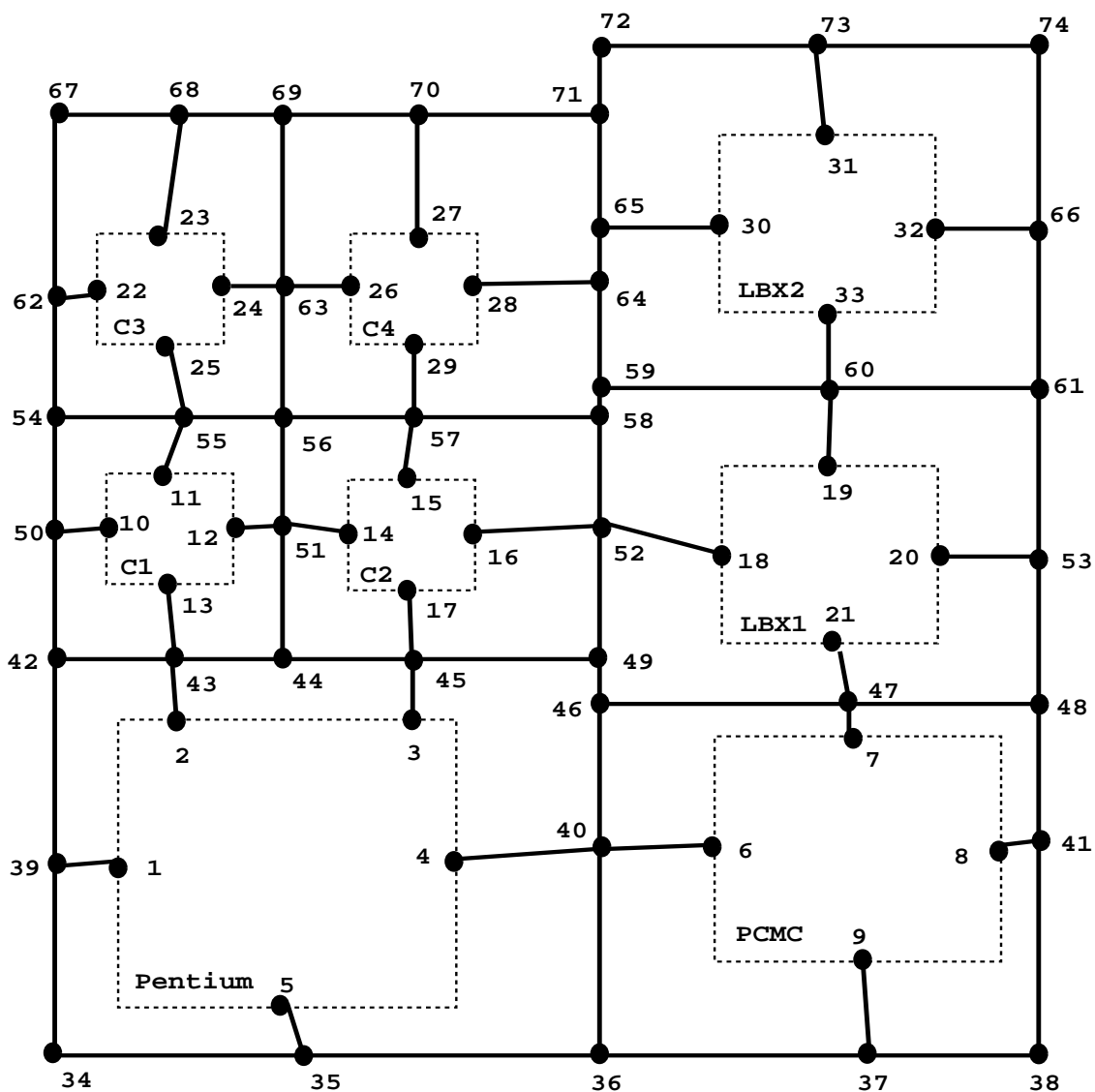


Figure 7.31: Channel Graph for Pentium Board

design to come up with a new set of design constraints.

7. This new set of constraints was evaluated for *safeness* and *flexibility* [60].

There are five net classes in the design, namely:

1. Address Bus: These nets connect address pins on the Pentium processor to

the PCMC and LBXs.

2. Data Bus: These nets connect data pins on the Pentium processor to the L2 Cache and the LBXs.
3. Control Signals from Pentium to PCMC.
4. Control Signals from PCMC to Pentium.
5. Control Signal from PCMC to LBXs.

The cache control signals and the DRAM connections are not modeled in this design. The DRAM nets are heavily loaded and require careful termination. The regular layout structure should allow for the optimization of terminating resistor values as presented in section 7.2.4. All clocks and clock like signals are excluded from this design also for the same reasons.

Table 7.15 shows the design space for each of the net classes, and the maximum settling delay in this design space from simulation. Tables 7.16, 7.17, 7.18 and 7.19 show the design spaces generated using the rule generation methodology. Some of 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 design are such that they may overestimate the actual routed lengths in some cases. Notice that the safeness coefficient for the all the rules for Address Bus and Data Bus nets is 100%. This suggests that the rule generation methodology does indeed produce very safe design rules.

Table 7.15: Constraints for Intel Pentium Design

Net Class	Design Space (m)	Settling delay (ns)
Pentium-PCMC Control	$0.0 \leq l_0 \leq .165$	7.49
PCMC-Pentium Control	$.058 \leq l_0 \leq .12$	7.82
PCMC-LBX Control	$0.0 \leq l_0 \leq .20$	10.61
	$0.0 \leq l_1 \leq .14$	11.60
Data Bus	$0.0 \leq l_0 \leq .12$	1.94
	$0.0 \leq l_1 \leq .11$	1.42
Address Bus	$0.0 \leq l_0 \leq .11$	6.58
	$0.0 \leq l_1 \leq .12$	10.15

7.4 Summary

The characterization, optimization, global routing and rule generation methodology were illustrated through several design examples in this chapter. The properties of the characterization methodology were established using MCM interconnect and a latch design. The optimization methodology was employed for several difficult transistor sizing problems. In each case, very good sizing schemes were identified with few simulations. A backplane bus design problem was treated as a termination resistor sizing problem, and a good termination scheme was found. This optimization illustrates was done very interactively by an experienced designer. The proposed optimization methodology helped the designer in drastically reducing the number of simulations required to generate the optimal solution. The global routing methodology was employed on two benchmark MCM layouts and the Intel Pentium Board. The combination of global routing and a-priori characterizations was proved to be very effective in determining the feasibility of the layout and electrical constraints. The rule generation methodology was illustrated using the Pentium design. Most of the generated wiring rules were 100% safe. This provides a complete solution path for the performance driven routing problem. The wiring rules and the global wiring

Table 7.16: Generated Rules for Intel Pentium Design

Number	Net Class	Design Rule (m)	Safeness
1	Pentium-PCMC Control	$8.812600e-02 \leq l_0 \leq 2.165140e-01$	88.2%
2	PCMC-Pentium Control	$1.646970e-01 \leq l_0 \leq 2.202630e-01$	85.7%
3	PCMC-Pentium Control	$1.445000e-02 \leq l_0 \leq 2.429300e-01$	78.6%
4	Address Bus	$8.740680e-02 \leq l_0 \leq 9.561120e-02$ $9.823780e-02 \leq l_1 \leq 1.064422e-01$	100%
5	Address Bus	$1.444700e-02 \leq l_0 \leq 5.908500e-02$ $9.824100e-02 \leq l_1 \leq 1.428790e-01$	100%
6	Address Bus	$8.740680e-02 \leq l_0 \leq 9.561120e-02$ $9.823780e-02 \leq l_1 \leq 1.064422e-01$	100%
7	Address Bus	$8.740900e-02 \leq l_0 \leq 1.130310e-01$ $5.923400e-02 \leq l_1 \leq 8.485600e-02$	100%
8	Address Bus	$5.490000e-02 \leq l_0 \leq 1.190900e-01$ $1.444700e-02 \leq l_1 \leq 7.863700e-02$	100%
9	Address Bus	$1.444700e-02 \leq l_0 \leq 8.497700e-02$ $5.923400e-02 \leq l_1 \leq 1.297640e-01$	100%
10	Address Bus	$1.278631e-01 \leq l_0 \leq 1.358369e-01$ $5.923410e-02 \leq l_1 \leq 6.720790e-02$	100%
11	Data Bus	$6.934800e-02 \leq l_0 \leq 1.058340e-01$ $6.934800e-02 \leq l_1 \leq 1.058340e-01$	100%
12	Data Bus	$1.358028e-01 \leq l_0 \leq 1.387972e-01$ $6.934780e-02 \leq l_1 \leq 7.234220e-02$	100%
13	Data Bus	$1.358023e-01 \leq l_0 \leq 1.378977e-01$ $8.812930e-02 \leq l_1 \leq 9.022470e-02$	100%
14	Data Bus	$1.358028e-01 \leq l_0 \leq 1.387972e-01$ $6.934780e-02 \leq l_1 \leq 7.234220e-02$	100%
15	Data Bus	$1.358023e-01 \leq l_0 \leq 1.378977e-01$ $8.812930e-02 \leq l_1 \leq 9.022470e-02$	100%

paths can be passed to a detailed routing tool to achieve routing that meets signal integrity and layout constraints.

Table 7.17: Generated Rules for Intel Pentium Design...

Number	Net Class	Design Rule (m)	Safeness
16	Data Bus	No rules	
17	Data Bus	No rules	
18	Data Bus	No rules	
19	Data Bus	$6.934800e-02 \leq l_0 \leq 9.303400e-02$ $8.957700e-02 \leq l_1 \leq 1.132630e-01$	100%
20	Data Bus	No rules	
21	Data Bus	$6.934780e-02 \leq l_0 \leq 7.478420e-02$ $1.170218e-01 \leq l_1 \leq 1.224582e-01$	100%
22	Data Bus	$4.045300e-02 \leq l_0 \leq 8.280900e-02$ $7.657100e-02 \leq l_1 \leq 1.189270e-01$	100%
23	Data Bus	$4.045300e-02 \leq l_0 \leq 7.429100e-02$ $8.957100e-02 \leq l_1 \leq 1.234090e-01$	100%
24	Data Bus	$4.045300e-02 \leq l_0 \leq 7.429100e-02$ $8.957100e-02 \leq l_1 \leq 1.234090e-01$	100%
25	Data Bus	$6.934700e-02 \leq l_0 \leq 9.931100e-02$ $7.946000e-02 \leq l_1 \leq 1.094240e-01$	100%
26	Data Bus	$6.934800e-02 \leq l_0 \leq 9.303400e-02$ $8.957700e-02 \leq l_1 \leq 1.132630e-01$	100%
27	Data Bus	$6.934800e-02 \leq l_0 \leq 9.303400e-02$ $8.957700e-02 \leq l_1 \leq 1.132630e-01$	100%
28	Data Bus	No rules	
29	Data Bus	No rules	
30	Data Bus	$6.067891e-02 \leq l_0 \leq 6.259109e-02$ $1.256939e-01 \leq l_1 \leq 1.276061e-01$	100%

Table 7.18: Generated Rules for Intel Pentium Design...

Number	Net Class	Design Rule (m)	Safeness
31	Data Bus	$4.045300e-02 \leq l_0 \leq 8.280900e-02$ $7.657100e-02 \leq l_1 \leq 1.189270e-01$	100%
32	Data Bus	$1.155800e-02 \leq l_0 \leq 3.871800e-02$ $1.155800e-01 \leq l_1 \leq 1.427400e-01$	100%
33	Data Bus	$1.155740e-01 \leq l_0 \leq 1.463860e-01$ $3.178400e-02 \leq l_1 \leq 6.259600e-02$	100%
34	Data Bus	$8.668000e-02 \leq l_0 \leq 1.318200e-01$ $3.178500e-02 \leq l_1 \leq 7.692500e-02$	100%
35	Data Bus	$9.824200e-02 \leq l_0 \leq 1.337780e-01$ $4.045300e-02 \leq l_1 \leq 7.598900e-02$	100%
36	Data Bus	$9.823970e-02 \leq l_0 \leq 1.128203e-01$ $8.379470e-02 \leq l_1 \leq 9.837530e-02$	100%
37	Data Bus	$6.934800e-02 \leq l_0 \leq 9.303400e-02$ $8.957700e-02 \leq l_1 \leq 1.132630e-01$	100%
38	Data Bus	$6.934800e-02 \leq l_0 \leq 1.058340e-01$ $6.934800e-02 \leq l_1 \leq 1.058340e-01$	100%
39	Data Bus	$6.934700e-02 \leq l_0 \leq 1.101810e-01$ $6.067900e-02 \leq l_1 \leq 1.015130e-01$	100%
40	Data Bus	$5.779000e-02 \leq l_0 \leq 1.227100e-01$ $2.167100e-02 \leq l_1 \leq 8.659100e-02$	100%
41	Data Bus	$6.934700e-02 \leq l_0 \leq 1.279690e-01$ $2.167100e-02 \leq l_1 \leq 8.029300e-02$	100%
42	Data Bus	$9.823840e-02 \leq l_0 \leq 1.056216e-01$ $9.823840e-02 \leq l_1 \leq 1.056216e-01$	100%
43	Data Bus	$9.824700e-02 \leq l_0 \leq 1.291130e-01$ $5.056600e-02 \leq l_1 \leq 8.143200e-02$	100%
44	Data Bus	$9.823840e-02 \leq l_0 \leq 1.056216e-01$ $9.823840e-02 \leq l_1 \leq 1.056216e-01$	100%

Table 7.19: Generated Rules for Intel Pentium Design...

Number	Net Class	Design Rule (m)	Safeness
45	Data Bus	$9.824670e-02 \leq l_0 \leq 1.164933e-01$ $7.657070e-02 \leq l_1 \leq 9.481730e-02$	100%
46	Data Bus	$6.934700e-02 \leq l_0 \leq 1.058330e-01$ $6.934700e-02 \leq l_1 \leq 1.058330e-01$	100%
47	Data Bus	No rules	
48	Data Bus	No rules	
49	Data Bus	No rules	
50	Data Bus	$1.155770e-02 \leq l_0 \leq 2.159230e-02$ $1.415827e-01 \leq l_1 \leq 1.516173e-01$	100%
51	Data Bus	$9.824080e-02 \leq l_0 \leq 1.034192e-01$ $1.025808e-01 \leq l_1 \leq 1.077592e-01$	100%
52	Data Bus	$6.934700e-02 \leq l_0 \leq 8.356100e-02$ $1.040230e-01 \leq l_1 \leq 1.182370e-01$	100%
53	Data Bus	$9.824660e-02 \leq l_0 \leq 1.026334e-01$ $1.040166e-01 \leq l_1 \leq 1.084034e-01$	100%
54	Data Bus	$6.934700e-02 \leq l_0 \leq 7.772100e-02$ $1.126930e-01 \leq l_1 \leq 1.210670e-01$	100%
55	PCMC-LBX Control	$1.444700e-02 \leq l_0 \leq 1.149890e-01$ $9.823900e-02 \leq l_1 \leq 1.987810e-01$	100%
56	PCMC-LBX Control	$9.824000e-02 \leq l_0 \leq 1.538000e-01$ $8.235000e-02 \leq l_1 \leq 1.379100e-01$	100%
57	Pentium-PCMC Control	$1.683110e-01 \leq l_0 \leq 2.178690e-01$	72%

Chapter 8

Conclusion

The work presented in this thesis accomplished some important goals:

- A comprehensive simulation based circuit characterization methodology was put in place. It provides the capability of characterizing arbitrary circuit topologies by varying almost any design parameters and investigating their effect on all measurable performance parameters. This methodology is crucially important in designing high speed circuit layouts. The accuracy of the characterization helps in breaking the layout design-extraction- simulation cycle, and would provide a considerable speed-up of the design process.
- An new methodology was developed for optimizing circuit responses by manipulating component values in the circuit design, based on stochastic modeling of circuit responses. This formulation is extremely interactive, and helps the designer achieve a tight control over the number of circuit simulations conducted , which constitutes the single most time consuming step in the optimization process. Several difficult transistor sizing problems, where analytical modeling of the responses is very difficult, and a backplane bus termination problem were optimally solved with few simulations. Of course, the number of simu-

lations to be run is dependent on the number of independent variables. The main application area for this technique is circuits with responses that are very hard to model, but with few independent components to be manipulated.

- The global routing and rule generation methodology helps in solving challenging layout problems. The global routing procedure provides a fast and accurate method for estimating the feasibility of conducting the layout for a given set of electrical constraints and module placement. In addition, the rule generation methodology provides flexible, yet accurate bounds on interconnect length, which can be passed to a detailed router to achieve successful routing under tight electrical constraints. The whole characterization, global routing and rule generation process provides a turnkey solution for the timing and signal integrity driven routing problem, as was demonstrated using the MCM and Intel Pentium design examples.

8.1 Future Work

There are several new avenues to be explored, some of which have been only partially addressed in this thesis:

- Though the characterization methodology is well established, there are several new problem areas which need to be explored. One very important issue is the modeling of inaccuracy in the simulator, and the proper modeling of process variations in the sampling methodology. If process statistics are available, then LHS can still be employed for generating the first sample. However, the resampling is complicated. A new measure of error-characterization has to be defined, to account for the fact that values of statistical variables occur with different likelihoods.

- The optimization methodology is quite powerful for small circuits. However, for addressing large circuit designs, the simplistic search techniques incorporated for determining the points for further simulation has to be enhanced. Another issue to explore is hierarchical model construction and problem decomposition, to allow problem instances of large dimensionality to be effectively explored. Also, several analog design problems, and many high speed layout problems, e.g. clock tree optimization should be broached using this optimization strategy.
- The global routing technique needs a fair bit of extension. The tree generation algorithms are quite simple, and new methods to find optimal routing trees have to be devised. Finding the optimal daisy chain, when no net sequencing is given, is an open problem. It is suspected that this problem is exponentially dependent on the number of nodes in the chain. Consider the following transformation. Let $G = (V, E)$ be the routing graph and $c \in V$ be the nodes to be connected in a daisy chain. Let $|c| = n$. In the optimal daisy chain in G , the path between any two consecutive nodes must be the shortest such path between those nodes in G . Construct a complete graph G' with vertex set c . The weight of each edge in G' is the length of the shortest path between its terminals in G . The length of the optimal daisy chain in G' would be the same as the length of the optimal daisy chain in G . Assume that the sequencing in the optimal daisy chain is (v_1, v_2, \dots, v_n) . Then the weight of the optimal daisy chain is $w(c) - w(v_1, v_n)$ where $w(c)$ is the weight of some cycle in G . One good way of generating a daisy chain is to find the shortest Hamiltonian cycle in G' and then delete the longer of the two edges adjacent to v_1 . However, the shortest Hamiltonian cycle is equivalent to the traveling salesman problem which is known to be NP-hard. But edge deletion from the smallest Hamiltonian cycle is *not* a characterization of the optimal daisy chain, so there

might exist a polynomial time solution to this problem.

- An extension to this problem is that of finding the shortest daisy chain with stubs with some limit on the stub length. Again, this is an open problem.
- Another problem not considered here is the iterative solution of the global routing problem. It is assumed that a sufficiently large number of trees has been generated so that the edge capacities are guaranteed to be satisfied. This will not be the case in general. Then the global routing problem has to be solved iteratively. First, the routing problem has to be solved while meeting as many capacity constraints as possible with the given set of routing trees. Also, if the routing trees are satisfactory from the electrical standpoint, then limits can be generated for each branch in the routing trees with the rule generation procedure. Now the global routing graph is updated. The congestion on the heavily utilized edges has to be reduced. Hence another weight is associated with each edge, which is inversely related to the routing congestion on that edge. Now routing trees are to be generated that are optimal according to this new set of weights. However, the routing trees must obey the length limits generated according to the original weights. Hence all shortest path problems encountered in the tree generation process are replaced by *constrained shortest path problems*. The constrained problem is to find shortest paths according to the new edge weights subject to bounds on the path lengths calculated using the original weights. Some solutions to this problems are proposed in [61]. The global routing procedure has to be extended to be able to solve highly constrained layouts by iterating on this process.
- The global routing procedure allows us to consider technology alternatives also. Net terminations, and driver resizing can be considered by treating them as routing trees with different benefit functions. Since terminations are usually

expensive, the cost of termination can be incorporated in the benefit function, so as to avoid using terminations as much as possible. Noise budgeting can be treated as well, by treating different reflection noise budgets for a given net as separate routing alternatives. These possibilities need to be further explored.

- The only noise source treated in this work is reflection noise. Simultaneous switching noise and cross-talk noise have not been considered, primarily due to modeling difficulties associated with these problems. Minimizing cross-talk through global routing needs to be studied.

Bibliography

- [1] A. R. Alvarez, B. L. Abdi, D. L. Young, H. D. Weed, and J. Teplik. Application of statistical design and response surface methods to computer-aided VLSI device design. *IEEE Transactions on Computer Aided Design*, CAD-7:pp. 272–288, Feb. 1988.
- [2] J. W. Bandler and S. H. Chen. Circuit optimization : The state of the art. *IEEE Transactions on Microwave Theory and Techniques*, MTT-36:pp. 424–443, Feb. 1988.
- [3] J. Benkoski and A. J. Strojwas. A new approach to hierarchical and statistical timing simulations. *IEEE Transactions on Computer Aided Design*, CAD-6:pp. 1039–1052, Nov 1987.
- [4] D. P. Bertsekas. *Linear network optimization: Algorithms and codes*. MIT Press, 1991.
- [5] R. M. Biernacki and J. W. Bandler. Efficient quadratic approximation for statistical design. *IEEE Transactions on Circuits and Systems*, 36:pp. 1449–1554, 1989.
- [6] K. D. Boese, A. B. Kahng, and G. Robins. High-performance routing trees with identified critical paths. In *Proceedings of the 30th Design Automation Conference*, pages pp. 182–187, 1993.
- [7] R. K. Brayton, G. D. Hatchel, and A. L. Sangiovanni-Vincentelli. A survey of optimization techniques for integrated-circuit design. *Proceedings of IEEE*, 69:pp. 1334–1362, Oct. 1981.
- [8] J. Cong, A. Kahng, G. Robins, M. Sarafzadeh, and C. K. Wong. Provably good performance-driven global routing. *IEEE Transactions on Computer-Aided Design*, 11(6):pp. 739–752, 1992.
- [9] J. Cong, K.-S. Leung, and D. Zhou. Performance-driven interconnect design based on distributed RC delay model. In *Proceedings of the 30th Design Automation Conference*, pages pp. 606–611, 1993.

- [10] E. Davidson. Personal communications, 1994.
- [11] E.E. Davidson and G.A. Katopis. Package Electrical Design. In R.R. Tummala and E.J. Rymaszewski, editors, *Microelectronics Semiconductor Handbook*, chapter 3. Van Nostrand Reinhold, 1989.
- [12] E. W. Dijkstra. A note on two problems in connection with graphs. *Numerische Math*, 1:pp. 269–271, 1959.
- [13] S. W. Director, P. Feldmann, and K. Krishna. Statistical integrated circuit design. *IEEE Journal of Solid State Circuits*, 28(3):pp. 193–202, 1993.
- [14] S. W. Director, W. Maly, and A. J. Storjwas. *VLSI Design for Manufacturing : Yield Enhancement*. Kluwer Academic Publishers, 1990.
- [15] D. W. Dobberpuhl and et al. A 200-MHz 64-b dual-issue CMOS microprocessor. *IEEE Journal of Solid-State Circuits*, 27(11):pp. 1555–1565, Nov. 1992.
- [16] W.C. Elmore. The transient response of damped linear networks with particular regard to wideband amplifiers. *Journal of Applied Physics*, 19(1):55–63, 1948.
- [17] R. W. Floyd. Algorithm 97 : Shortest Path. *Communications of ACM*, 5, 1962.
- [18] P. D. Franzon. Chapter 11. In D. A. Doane and P. D. Franzon, editors, *Multi-chip Module Technologies and Alternatives: The Basics*. Van Nostrand Reinhold, 1992.
- [19] P.D. Franzon, S. Simovich, M. Steer, M. Basel, S. Mehrotra, and T. Mills. Tools to aid in wiring rule generation for high speed interconnects. In *Proceedings of the 29th Design Automation Conference*, 1992.
- [20] M. R. Garey and D. S. Johnson. *Computers and Intractability: A guide to the theory of NP-Completeness*. W. H. Freeman, 1979.
- [21] C. T. Gray, W. Liu, and R. Cavin. *Wave-pipelining: theory and CMOS implementation*. Kluwer Academic Press, 1994.
- [22] A. Groch, S. W. Director, and L. M. Vidigal. A new global optimization method for electronic circuit design. *IEEE Transactions on Circuits and Systems*, CAS-32:160–169, 1985.
- [23] W. Hobbs, R. Rosenbaum A. Muranyi, and D. Telian. IBIS: I/O buffer information specification overview. Technical Report Version 1.0, Intel Corporation, 1994.
- [24] X.-L. Hong, T. Xue, E. S. Kuh, C.-K. Cheng, and J. Huang. Performance-driven steiner tree algorithms for global routing. In *Proceedings of the 30th Design Automation Conference*, pages pp. 177–181, 1993.

- [25] Reiner Horst. Recent advances in global optimization: A tutorial study. In Santosh Kumar, editor, *Recent developments in mathematical programming*. Gordon and Breach Science Publishers, 1991.
- [26] R. L. Iman and W. J. Conover. A distribution-free approach to inducing rank correlations among input variables. *Comm. Stat.*, B11(3):311–334, 1982.
- [27] R. L. Iman and M. J. Shortencarier. A FORTRAN 77 program and user's guide for the generation of Latin Hypercube and random samples for use with computer models. Technical Report SAND83-2365, Sandia National Laboratories, 1984.
- [28] Intel Corporation. *Pentium Processor/82430 PCIset Open Design Guide*, 1993.
- [29] R. C. Carden IV and C.-K. Cheng. Feasibility estimation and cost optimization for multichip module technologies. In *Proc. of IEEE ASIC Conference and Exhibit*, pages P9-1.1 – P9-1.4, 1991.
- [30] R. C. Carden IV and C.-K. Cheng. A global router using an efficient approximate multicommodity multiterminal flow algorithm. In *Proc. of 28th Design Automation Conference*, pages pp. 322–327, 1991.
- [31] A. H. G. Rinnooy Kan and G. T. Timmer. Stochastic global optimization methods part I: Clustering methods. *Mathematical Programming*, 39:pp. 27–56, 1987.
- [32] A. H. G. Rinnooy Kan and G. T. Timmer. Stochastic global optimization methods part II: Multi level methods. *Mathematical Programming*, 39:pp. 57–78, 1987.
- [33] B. Korte, H. J. Promel, and A. Steger. Steiner trees in VLSI-Layout. In B. Korte, L. Lovasz, H. J. Promel, and A. Schrijver, editors, *Paths, Flows and VLSI-Layout*, pages 185–214. Springer-Verlag, 1990.
- [34] Peter Lancaster and Kestutis Salkauskas. *Curve and surface fitting : An introduction*. Academic Press, 1986.
- [35] D. P. LaPotin and Y.-H. Chen. Early matching of system requirements and package capabilities. In *IEEE International Conf. on Computer-Aided Design*, pages pp. 394–397, 1989.
- [36] E. L. Lawler. *Combinatorial optimization: Networks and Matroids*. Holt, Rinehart and Winston, 1976.
- [37] J. B. Lee, E. B. Shragowitz, and D. Poli. Bounds on net lengths for high speed PCBs. In *Proc. of Int'l. Conf. on Computer-Aided Design*, pages 73–76, 1993.
- [38] Jaebum Lee and Eugene Shragowitz. Synthesis of transmission line interconnects in presence of distortions. Technical Report TR 94-39, Computer Science Department, Univeristy of Minnesota, 1994.

- [39] T. Lengauer and D. Theune. Efficient algorithms for path problems with general cost criteria. In *18th Int. Symposium on Automata, Languages and Programming, Springer LNCS No. 510*, pages pp. 314–326, 1991.
- [40] Thomas Lengauer. *Combinatorial Algorithms for Integrated Circuit Layout*. John Wiley and Sons, 1990.
- [41] M. R. Lightner, T. N. Trick, and R. P. Zug. Circuit optimization and design. In A. E. Ruehli, editor, *Circuit Analysis, Simulation and Design*, pages 333–391. Elsevier Science Publishers, 1987.
- [42] K. K. Low and S. W. Director. A new methodology for the design centering of IC fabrication process. *IEEE Transactions on Computer Aided Design*, CAD-10:pp. 895–903, July 1991.
- [43] Jin-Qin Lu and Takehiko Adachi. A parameter optimization method for electronic circuit design using stochastic model function. *Electronics and Communications in Japan*, 75(4):13–25, 1992.
- [44] Mark D. Matson and Lance G. Glasser. Macromodeling and optimization of digital MOS VLSI circuits. *IEEE Transactions on Computer Aided Design*, CAD-5(4):pp. 659–679, Oct 1986.
- [45] 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.
- [46] L. W. Nagel. SPICE2: A computer program to simulate semiconductor circuits. Technical Report ERL-M520, Electronics Research Lab, University of California, 1975.
- [47] W. Van Noije, C. T. Gray, W. Liu, T. Hughes, and R. Cavin. CMOS sampler with 1GBits/s bandwidth with 25ps resolution. In *Proceedings of the Custom Integrated Circuits Conference*, pages 27.5.1–27.5.4, 1993.
- [48] W. Nye, D. Riley, A. Sangiovanni-Vincentelli, and A. Tits. DELIGHT.SPICE: An Optimization-Based System for the Design of Integrated Circuits. *IEEE Trans. on Computer-Aided Design*, CAD-7(4):pp. 501–519, April 1988.
- [49] L. T. Pillage and R. A. Rohrer. Asymptotic Waveform Evaluation form timing analysis. *IEEE Transactions on Computer Aided Design*, 9:pp. 352–366, April 1990.
- [50] P. Raghavan. Integer programming for VLSI design. *Discrete Applied Mathematics*, 40:29–43, 1992.
- [51] P. Raghavan and C. D. Thompson. Multiterminal global routing: A deterministic approximation. *Algorithmica*, 6:73–82, 1991.

- [52] G. Rote. Path problems in graphs. *Computing Supplement*, 7:155–189, 1990.
- [53] J. Sacks, S. B. Schiller, and W. J. Welch. Designs for computer experiments. *Technometrics*, 31:41–47, 1989.
- [54] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):pp. 409–435, 1989.
- [55] S. Sapatnekar and S. M. Kang. *Design automation for timing-driven layout synthesis*. Kluwer Academic Press, 1993.
- [56] I. P. Schagen. Stochastic interpolating functions-applications in optimization. *Journal of Institute of Mathematical Applications*, 26:pp. 93–101, 1980.
- [57] I. P. Schagen. Sequential exploration of unknown multidimensional functions as an aid to optimization. *IMA Journal of Numerical Analysis*, 4:pp. 337–347, 1984.
- [58] S. Simovich, P. Franzon, and M. Steer. A method for automated waveform analysis of transient responses in digital circuits. *Electronics Letters*, 29(8), April 21 1993.
- [59] S. Simovich, S. Mehrotra, P. Franzon, and M. Steer. Delay and reflection noise macromodeling for signal integrity management of PCBs and MCMs. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 17(1):15–21, 1994.
- [60] Slobodan Simovich. *A methodology for automated simulation-based electrical characterization and design of high-speed systems*. PhD thesis, North Carolina State University, 1994.
- [61] C. C. Skiscim and B. L. Golden. Solving k-shortest and constrained shortest path problems efficiently. *Annals of Operations Research*, 20:249–282, 1989.
- [62] Larry D. Smith. Personal communications, 1994.
- [63] M. Sriram and S. M. Kang. *Physical Design of Multi Chip Modules*. Kluwer Academic Press, 1993.
- [64] M. A. Styblinski and S. A. Aftab. Combination of interpolation and self-organizing approximation techniques-A new approach to circuit performance modeling. *IEEE Transactions on Computer-Aided Design*, 12(11):1775–1784, 1993.
- [65] Boxin Tang. OA-based Latin Hypercubes : IIQP research report RR-91-10. Technical report, Department of Statistics and Actuarial Science, University of Waterloo, 1991.
- [66] R. E. Tarjan. Fast algorithms for solving path problems. *Journal of the Association for Computing Machinery*, 28(3):594–614, 1981.

- [67] Aimo Torn and Antanas Zilinskas. *Global Optimization*. Springer-Verlag, 1989.
- [68] J. F. Traub, G. W. Wasilkowski, and H. Wozniakowski. *Information Based Complexity*. Academic Press, 1988.
- [69] G. W. Wasilkowski. On average complexity of global optimization problems. *Mathematical Programming*, 97:pp. 313–324, 1992.
- [70] W. J. Welch, R. J. Buck, J. Sacks, H. P. Wynn, T. J. Mitchell, and M. D. Morris. Screening, predicting, and computer experiments. *Technometrics*, 34(1):15–25, Feb 1992.
- [71] Jr. William R. Blood. *MECL - system design handbook*. Motorola Inc., 1988.
- [72] D. Wong, G. De Micheli, and M. Flynn. Designing high-performance digital circuits using wave pipelining. In *VLSI'89*, pages pp. 241–252, 1989.
- [73] D. L. Young, J. Teplik, H. D. Weed, N. T. Tracht, and A. R. Alvarez. Application of statistical design and response surface methods to computer-aided VLSI device design II: Desirability functions and Taguchi methods. *IEEE Transactions on Computer Aided Design*, CAD-10:pp. 103–115, Jan. 1991.
- [74] A. Zilinskas. Axiomatic approach to statistical models and their use in multi-modal optimization theory. *Mathematical Programming*, 22:pp. 104–116, 1982.
- [75] A. Zilinskas. Axiomatic characterization of a global optimization algorithm and investigation of its search strategy. *Operations Research Letters*, 4(1):pp. 35–39, 1985.

Appendix A

MCC Designs

This appendix lists the netlists for the two MCM design examples and a CaZm File Generator Input file for a point-to-point net, which includes the driver, receiver, interconnect and package parasitic models used for simulating all the nets in the two designs. The format of the netlists is as follows:

- $\text{Pin}(1)1, \text{Pin}(2), \dots, \text{Pin}(n) : N$
- $\text{TC}(2), \dots, \text{TC}(n)$

Here $\text{Pin}(i)$ is the number of the node in the graph that each of the pins in a net are abstracted to. $\text{TC}(i)$ is the maximum settling delay from $\text{Pin}(1)$ to $\text{Pin}(i)$ for that net. N is the multiplicity of the net.

A.1 MCC1 Netlist

A.1.1 Coarse Graph

14	38	:	104	12	48	:	55	7	22	:	19
8.54e-10				9.32e-10				6.59e-10			
14	39	:	104	12	49	:	55	1	22	:	7
1.15e-09				5.19e-10				1.80e-09			
14	40	:	103	20	54	:	1	20	22	:	2
1.90e-09				8.54e-10				1.44e-09			
14	41	:	103	21	54	:	4	8	22	:	6
2.12e-09				1.32e-09				2.12e-09			
15	42	:	97	21	55	:	3	22	28	:	21
5.74e-10				7.28e-10				1.71e-09			
15	43	:	97	13	51	:	54	22	33	:	23
7.52e-10				7.33e-10				1.99e-09			
15	44	:	97	13	52	:	54	2	22	:	15
9.52e-10				1.03e-09				2.30e-09			
15	45	:	97	13	53	:	54	8	23	:	9
1.79e-09				1.06e-09				6.44e-10			
16	46	:	98	13	50	:	54	7	23	:	16
1.08e-09				1.39e-09				2.22e-09			
16	47	:	98	13	30	:	26	23	28	:	27
1.22e-09				1.97e-09				9.41e-10			
16	48	:	98	13	17	:	9	23	26	:	32
1.05e-09				4.92e-10				1.64e-09			
16	49	:	98	13	31	:	12	23	32	:	12
1.44e-09				2.48e-09				1.36e-09			
17	50	:	80	0	13	:	3	0	23	:	30
8.78e-10				1.07e-09				2.07e-09			
17	51	:	80	14	17	:	3	23	31	:	4

9.86e-10	3.03e-09	1.29e-09
17 52 : 80	15 17 : 4	23 27 : 3
1.37e-09	3.00e-09	2.08e-09
17 53 : 80	16 17 : 4	9 23 : 4
1.47e-09	1.69e-09	1.36e-09
19 54 : 188	10 16 : 2	23 24 : 11
6.67e-10	3.87e-09	1.79e-09
19 55 : 188	11 16 : 2	23 25 : 6
1.49e-09	1.87e-09	1.79e-09
10 40 : 55	12 16 : 1	23 33 : 11
5.37e-10	1.08e-09	7.57e-10
10 41 : 55	13 16 : 1	20 23 : 5
8.65e-10	2.05e-09	2.05e-09
10 38 : 55	22 32 : 3	8 24 : 9
1.45e-09	1.40e-09	2.19e-09
10 39 : 55	0 22 : 28	7 24 : 14
6.12e-10	1.47e-09	2.07e-09
11 42 : 55	22 31 : 4	24 28 : 27
3.81e-10	1.29e-09	1.17e-09
11 43 : 55	22 27 : 5	24 26 : 32
4.34e-10	2.05e-09	1.11e-09
11 44 : 55	9 22 : 3	24 32 : 12
1.02e-09	2.78e-09	1.23e-09
11 45 : 55	22 23 : 6	0 24 : 27
1.65e-09	2.53e-09	1.33e-09
12 46 : 55	22 24 : 2	24 31 : 3
1.13e-09	2.50e-09	1.50e-09

12 47 : 55	22 25 : 5	24 27 : 4
1.29e-09	8.24e-10	6.01e-10
9 24 : 4	7 30 : 10	28 30 : 32
6.67e-10	5.94e-10	8.46e-10
24 25 : 10	7 29 : 6	28 33 : 32
1.76e-09	1.13e-09	6.90e-10
20 24 : 5	7 33 : 11	20 28 : 4
2.11e-09	1.92e-09	1.41e-09
24 33 : 10	1 7 : 1	28 31 : 14
1.48e-09	1.99e-09	5.30e-10
7 25 : 17	7 31 : 6	29 34 : 12
1.33e-09	1.20e-09	1.55e-09
8 25 : 11	7 27 : 2	29 30 : 93
2.22e-09	1.87e-09	7.77e-10
25 28 : 27	9 17 : 1	20 29 : 9
1.07e-09	1.51e-09	8.20e-10
25 26 : 26	9 32 : 1	29 35 : 9
1.18e-09	1.74e-09	1.81e-09
25 32 : 12	1 17 : 7	29 31 : 75
5.70e-10	9.36e-10	9.62e-10
0 25 : 29	1 2 : 24	29 33 : 7
7.85e-10	6.70e-10	1.90e-09
25 31 : 2	1 27 : 7	3 29 : 20
1.35e-09	7.71e-10	1.48e-09
25 27 : 2	1 20 : 13	4 29 : 9
1.54e-09	1.26e-09	1.70e-09
9 25 : 4	1 25 : 1	6 29 : 12

2.16e-09	1.07e-09	2.53e-09
20 25 : 5	1 28 : 3	5 29 : 20
6.23e-10	1.90e-09	2.25e-09
25 33 : 10	31 : 15	8 29 : 6
1.32e-09	2.22e-09	1.68e-09
0 1 : 17	2 23 : 15	3 30 : 24
5.05e-10	3.05e-09	9.48e-10
0 2 : 63	2 24 : 13	6 30 : 33
1.11e-09	1.36e-09	2.19e-09
0 20 : 12	2 25 : 14	5 30 : 33
1.10e-09	1.33e-09	1.80e-09
0 8 : 5	2 20 : 4	30 31 : 24
2.66e-09	1.06e-09	5.37e-10
7 9 : 64	2 28 : 1	20 30 : 8
2.39e-09	2.00e-09	1.44e-09
0 7 : 2	26 27 : 280	30 35 : 4
1.67e-09	7.57e-10	1.28e-09
7 8 : 95	0 27 : 2	4 30 : 16
1.79e-09	1.11e-09	1.36e-09
8 9 : 16	14 27 : 2	30 34 : 3
1.86e-09	2.39e-09	1.00e-09
1 8 : 1	7 28 : 4	8 30 : 6
3.05e-09	1.37e-09	1.54e-09
1 9 : 1	28 34 : 11	4 31 : 8
1.56e-09	1.10e-09	9.13e-10
9 31 : 4	28 35 : 8	5 31 : 2
1.88e-09	7.85e-10	1.23e-09

0 9 : 1	28 29 : 48	6 31 : 15
1.79e-09	1.30e-09	1.81e-09
9 27 : 1	27 28 : 1	20 31 : 12
1.14e-09	1.41e-09	1.77e-09
19 31 : 48	33 34 : 2	11 13 : 1
2.31e-09	6.84e-10	4.18e-09
2 31 : 1	6 34 : 4	12 14 : 1
2.68e-09	1.33e-09	4.18e-09
27 31 : 1	5 34 : 4	14 15 : 2
1.73e-09	1.04e-09	3.11e-09
31 33 : 29	6 32 : 24	10 14 : 1
9.98e-10	1.70e-09	1.23e-09
31 35 : 1	6 36 : 96	5 20 : 1
1.06e-09	1.06e-09	1.99e-09
31 34 : 1	5 32 : 24	20 36 : 10
4.25e-10	1.57e-09	2.39e-09
32 33 : 132	5 36 : 96	3 20 : 4
9.57e-10	1.30e-09	2.30e-09
31 32 : 15	33 35 : 2	2 21 : 2
8.96e-10	9.32e-10	1.26e-09
27 32 : 5	35 36 : 17	11 21 : 2
9.17e-10	1.74e-09	2.41e-09
27 33 : 3	32 35 : 10	21 32 : 2
1.82e-09	9.57e-10	1.32e-09
30 33 : 1	6 35 : 4	21 33 : 4
1.51e-09	1.01e-09	2.22e-09
3 32 : 24	14 20 : 6	21 31 : 2

1.39e-09	1.08e-09	1.33e-09
3 37 : 166	16 20 : 1	21 23 : 2
3.05e-09	3.13e-09	2.62e-09
3 36 : 96	20 34 : 2	8 21 : 4
2.23e-09	2.14e-09	2.29e-09
4 32 : 16	20 21 : 23	15 21 : 2
1.77e-09	7.85e-10	3.87e-09
4 37 : 161	14 21 : 5	9 21 : 4
2.42e-09	4.74e-10	3.05e-09
4 36 : 64	10 17 : 21	16 21 : 1
1.70e-09	2.59e-09	3.87e-09
34 37 : 25	10 11 : 1	19 21 : 1
2.25e-09	1.74e-09	1.02e-09
6 37 : 48	2 32 : 1	10 29 : 16
1.18e-09	1.76e-09	1.41e-09
5 37 : 48	8 28 : 1	10 30 : 32
1.39e-09	1.56e-09	1.16e-09
35 37 : 2	1 32 : 1	10 22 : 4
1.83e-09	1.49e-09	1.16e-09
3 6 : 16	3 28 : 1	10 23 : 4
2.31e-09	8.85e-10	2.59e-09
4 6 : 32	4 34 : 1	10 24 : 4
1.71e-09	4.80e-10	2.42e-09
4 5 : 16	36 37 : 2	10 25 : 4
1.26e-09	9.13e-10	1.88e-09
3 34 : 27	5 35 : 1	10 31 : 32
9.27e-10	4.34e-10	1.41e-09

21	34	:	1	5	6	:	3	0	10	:	33
1.70e-09				7.33e-10				2.08e-09			
34	36	:	44	14	16	:	1	10	19	:	1
1.23e-09				4.18e-09				2.47e-09			
32	34	:	10	10	13	:	1	11	29	:	16
1.30e-09				3.05e-09				2.14e-09			
11	30	:	32	11	19	:	1	12	31	:	32
1.83e-09				3.80e-09				2.31e-09			
11	22	:	4	12	29	:	16	0	12	:	33
1.79e-09				2.86e-09				2.09e-09			
11	23	:	4	12	30	:	32	12	19	:	1
1.00e-09				2.44e-09				2.50e-09			
11	24	:	4	12	22	:	4	13	29	:	16
2.44e-09				3.87e-09				1.70e-09			
11	25	:	4	12	23	:	4	13	22	:	4
2.55e-09				1.73e-09				2.25e-09			
11	17	:	27	12	24	:	4	13	23	:	4
3.87e-09				9.57e-10				3.11e-09			
11	31	:	32	12	25	:	4	13	24	:	4
1.27e-09				2.47e-09				1.51e-09			
0	11	:	33	12	17	:	19	13	25	:	4
2.94e-09				1.12e-09				1.45e-09			
10	11	17	:	9	4.57e-09	4.57e-09					
1.00e-08	1.00e-08			7	8	28	:	108			
12	13	17	:	9	3.21e-09	3.18e-09					
1.94e-09	1.95e-09			28	29	30	:	11			

7	8	22	:	7	1.94e-09	1.95e-09
4.18e-09	4.40e-09	28	30	31	:	5
1	2	22	:	3	1.61e-09	1.61e-09
2.77e-09	2.77e-09	6	34	35	:	8
7	8	23	:	16	2.74e-09	2.78e-09
2.37e-09	2.37e-09	5	34	35	:	4
1	2	23	:	3	2.35e-09	2.35e-09
4.05e-09	4.05e-09	5	6	35	:	84
7	8	24	:	14	1.63e-09	1.59e-09
4.18e-09	4.40e-09	15	16	20	:	4
1	2	24	:	3	4.57e-09	4.57e-09
1.95e-09	1.94e-09	20	34	35	:	2
7	8	25	:	10	4.05e-09	4.18e-09
4.18e-09	4.40e-09	4	20	29	:	2
1	2	25	:	3	4.18e-09	4.40e-09
1.95e-09	1.94e-09	10	11	21	:	4
0	1	2	:	64	4.57e-09	4.57e-09
1.21e-09	8.04e-10	1	21	22	:	2
7	8	9	:	15	2.78e-09	2.78e-09
4.05e-09	3.95e-09	14	15	21	:	2
1	24	25	:	13	4.57e-09	4.50e-09
2.53e-09	2.77e-09	15	16	21	:	2
1	7	8	:	2	4.57e-09	4.50e-09
4.57e-09	4.57e-09	0	7	21	:	2
1	8	9	:	1	3.52e-09	3.52e-09
1.00e-08	1.00e-08	0	9	21	:	2
1	22	23	:	1	4.57e-09	4.50e-09

A.1.2 Fine Graph

180 322 : 28	5 217 : 29	182 324 : 19	65 194 : 29
1.15e-09	9.06e-10	1.85e-09	8.59e-10
181 322 : 29	4 217 : 27	183 324 : 18	26 194 : 29
8.32e-10	1.22e-09	2.12e-09	1.28e-09
182 322 : 28	65 219 : 26	180 324 : 19	5 194 : 27
1.85e-09	1.01e-09	1.15e-09	1.20e-09
183 322 : 28	26 219 : 28	181 324 : 19	4 194 : 28
2.12e-09	1.57e-09	8.32e-10	4.93e-10
182 321 : 30	4 219 : 29	66 222 : 56	5 193 : 17
1.85e-09	1.22e-09	4.20e-10	1.20e-09
183 321 : 28	65 217 : 29	114 222 : 57	4 193 : 15
2.12e-09	1.01e-09	1.57e-09	4.93e-10
180 323 : 28	26 217 : 26	66 221 : 58	65 195 : 13
1.15e-09	1.57e-09	4.20e-10	8.59e-10
181 323 : 30	5 219 : 26	114 221 : 52	26 195 : 14
8.32e-10	9.06e-10	1.57e-09	1.28e-09
183 323 : 29	3 190 : 28	66 223 : 52	66 250 : 1
2.12e-09	9.14e-10	4.20e-10	9.64e-10
180 321 : 29	2 190 : 29	114 223 : 55	66 278 : 3
1.15e-09	1.20e-09	1.57e-09	1.42e-09
181 321 : 26	1 190 : 28	66 224 : 22	114 278 : 2
8.32e-10	1.41e-09	4.20e-10	6.51e-10

182 323 : 26	18 190 : 28	114 224 : 24	182 325 : 13
1.85e-09	1.25e-09	1.57e-09	6.03e-10
184 330 : 28	1 189 : 26	182 326 : 29	180 327 : 11
5.05e-10	1.41e-09	6.03e-10	1.27e-09
164 330 : 29	18 189 : 23	183 326 : 29	181 327 : 12
5.24e-10	1.25e-09	1.13e-09	7.38e-10
123 330 : 28	3 191 : 24	180 326 : 27	183 325 : 12
1.00e-09	9.14e-10	1.27e-09	1.13e-09
98 330 : 28	2 191 : 24	181 326 : 28	184 317 : 13
1.77e-09	1.20e-09	7.38e-10	7.50e-10
123 329 : 29	18 191 : 23	180 325 : 17	123 319 : 11
1.00e-09	1.25e-09	1.27e-09	8.41e-10
98 329 : 27	3 189 : 24	181 325 : 15	98 319 : 12
1.77e-09	9.14e-10	7.38e-10	1.75e-09
184 331 : 26	2 189 : 23	182 327 : 13	164 317 : 12
5.05e-10	1.20e-09	6.03e-10	7.10e-10
164 331 : 28	1 191 : 21	183 327 : 14	65 193 : 13
5.24e-10	1.41e-09	1.13e-09	8.59e-10
98 331 : 29	123 332 : 14	184 318 : 29	5 195 : 11
1.77e-09	1.00e-09	7.50e-10	1.20e-09
184 329 : 29	98 332 : 13	164 318 : 29	4 195 : 12
5.05e-10	1.77e-09	7.10e-10	4.93e-10
164 329 : 26	184 332 : 14	123 318 : 27	26 193 : 12
5.24e-10	5.05e-10	8.41e-10	1.28e-09
123 331 : 26	164 332 : 14	98 318 : 28	2 186 : 28
1.00e-09	5.24e-10	1.75e-09	5.82e-10
65 218 : 28	5 220 : 15	123 317 : 17	1 186 : 29

1.01e-09	9.06e-10	8.41e-10	9.06e-10
26 218 : 29	4 220 : 14	98 317 : 15	18 186 : 28
1.57e-09	1.22e-09	1.75e-09	1.03e-09
5 218 : 28	65 220 : 15	184 319 : 13	3 186 : 28
9.06e-10	1.01e-09	7.50e-10	1.38e-09
4 218 : 28	26 220 : 15	164 319 : 14	18 185 : 14
1.22e-09	1.57e-09	7.10e-10	1.03e-09
3 185 : 13	195 219 : 1	206 301 : 1	270 301 : 2
1.38e-09	1.04e-09	2.20e-09	2.46e-09
2 187 : 12	185 217 : 1	198 301 : 1	266 271 : 3
5.82e-10	2.26e-09	2.30e-09	7.10e-10
1 187 : 13	192 217 : 1	270 298 : 8	266 269 : 1
9.06e-10	1.90e-09	4.20e-10	7.10e-10
3 187 : 13	192 220 : 2	270 306 : 6	269 298 : 1
1.38e-09	1.90e-09	2.30e-09	4.20e-10
2 185 : 14	258 302 : 3	262 270 : 17	262 269 : 4
5.82e-10	1.44e-09	1.15e-09	1.15e-09
1 185 : 12	226 302 : 22	230 270 : 21	258 271 : 4
9.06e-10	1.25e-09	1.66e-09	1.25e-09
18 187 : 12	290 302 : 3	258 270 : 6	258 269 : 2
1.03e-09	1.41e-09	1.25e-09	1.25e-09
18 192 : 6	206 302 : 4	226 270 : 13	227 270 : 2
1.25e-09	2.20e-09	2.04e-09	2.04e-09
3 192 : 4	214 302 : 2	225 269 : 1	206 271 : 1
9.14e-10	2.80e-09	2.04e-09	2.20e-09
2 192 : 4	270 302 : 1	269 290 : 1	214 271 : 2
1.20e-09	2.46e-09	1.48e-09	1.31e-09

1 192 : 5	210 302 : 1	206 270 : 2	210 269 : 1
1.41e-09	2.64e-09	2.20e-09	1.71e-09
185 286 : 4	254 302 : 4	214 270 : 1	254 269 : 1
2.04e-09	1.08e-09	1.31e-09	1.68e-09
187 286 : 6	302 306 : 15	225 270 : 3	270 307 : 1
2.04e-09	6.15e-10	2.04e-09	2.30e-09
185 189 : 3	202 302 : 4	210 270 : 2	210 297 : 3
5.50e-10	1.78e-09	1.71e-09	2.11e-09
185 192 : 3	250 302 : 1	254 270 : 1	210 306 : 10
5.50e-10	1.23e-09	1.68e-09	2.13e-09
187 192 : 3	298 302 : 5	266 270 : 1	210 298 : 5
5.50e-10	1.97e-09	7.10e-10	2.11e-09
187 290 : 7	262 302 : 11	250 270 : 2	210 262 : 12
2.51e-09	1.77e-09	2.29e-09	1.05e-09
185 290 : 5	262 301 : 6	269 306 : 6	210 263 : 2
2.51e-09	1.77e-09	2.30e-09	1.05e-09
187 226 : 2	226 303 : 1	270 305 : 1	210 261 : 2
1.16e-09	1.25e-09	2.30e-09	1.05e-09
185 226 : 1	290 303 : 1	262 271 : 4	210 230 : 17
1.16e-09	1.41e-09	1.15e-09	9.79e-10
192 324 : 1	301 306 : 2	230 269 : 3	210 258 : 6
3.09e-09	6.15e-10	1.66e-09	1.38e-09
191 324 : 2	266 303 : 3	226 271 : 3	210 226 : 8
3.09e-09	2.05e-09	2.04e-09	1.49e-09
191 332 : 2	266 301 : 5	226 269 : 6	209 226 : 6
4.00e-09	2.05e-09	2.04e-09	1.49e-09
189 332 : 2	266 302 : 6	271 290 : 2	210 290 : 2

4.00e-09	2.05e-09	1.48e-09	1.34e-09
189 220 : 1	298 303 : 1	271 305 : 1	206 210 : 2
1.90e-09	1.97e-09	2.30e-09	6.61e-10
220 327 : 2	262 303 : 4	271 306 : 1	210 214 : 2
3.90e-09	1.77e-09	2.30e-09	4.25e-10
220 319 : 1	226 301 : 5	261 270 : 2	210 227 : 3
1.75e-09	1.25e-09	1.15e-09	1.49e-09
219 319 : 1	303 306 : 2	270 303 : 3	210 271 : 2
1.75e-09	6.15e-10	2.46e-09	1.71e-09
210 254 : 3	209 227 : 1	253 262 : 3	198 225 : 14
1.80e-09	1.49e-09	1.15e-09	9.41e-10
210 250 : 2	209 270 : 1	230 253 : 7	225 250 : 8
2.11e-09	1.71e-09	1.14e-09	1.13e-09
211 269 : 2	209 254 : 4	230 255 : 5	226 297 : 1
1.71e-09	1.80e-09	1.14e-09	2.47e-09
209 269 : 2	254 307 : 1	226 253 : 2	226 298 : 1
1.71e-09	1.42e-09	4.40e-10	2.47e-09
210 305 : 2	254 298 : 8	227 254 : 3	226 299 : 2
2.13e-09	1.99e-09	4.40e-10	2.47e-09
211 262 : 2	254 305 : 4	254 306 : 2	203 227 : 1
1.05e-09	1.42e-09	1.42e-09	3.93e-10
209 262 : 6	254 262 : 9	255 307 : 3	203 225 : 1
1.05e-09	1.15e-09	1.42e-09	3.93e-10
211 230 : 9	254 263 : 6	253 305 : 1	225 271 : 1
9.79e-10	1.15e-09	1.42e-09	2.04e-09
209 230 : 1	254 261 : 5	225 253 : 3	225 255 : 1
9.79e-10	1.15e-09	4.40e-10	4.40e-10

209 229 : 4	230 254 : 7	253 301 : 1	198 226 : 3
9.79e-10	1.14e-09	1.08e-09	9.41e-10
210 225 : 4	231 254 : 2	254 266 : 4	201 226 : 1
1.49e-09	1.14e-09	1.48e-09	3.93e-10
211 225 : 1	229 254 : 2	255 305 : 2	226 250 : 1
1.49e-09	1.14e-09	1.42e-09	1.13e-09
211 226 : 3	254 258 : 5	253 298 : 3	201 227 : 5
1.49e-09	7.88e-10	1.99e-09	3.93e-10
211 301 : 1	253 258 : 2	255 258 : 4	199 227 : 8
2.64e-09	7.88e-10	7.88e-10	9.41e-10
211 270 : 1	227 253 : 1	253 257 : 1	197 227 : 9
1.71e-09	4.40e-10	7.88e-10	9.41e-10
211 266 : 1	225 254 : 5	206 255 : 1	197 225 : 8
1.25e-09	4.40e-10	1.45e-09	9.41e-10
209 266 : 3	226 254 : 12	214 255 : 2	227 271 : 1
1.25e-09	4.40e-10	2.28e-09	2.04e-09
210 299 : 1	254 290 : 2	226 255 : 1	227 255 : 1
2.11e-09	1.34e-09	4.40e-10	4.40e-10
210 307 : 2	206 254 : 1	255 263 : 1	199 225 : 6
2.13e-09	1.45e-09	1.15e-09	9.41e-10
211 263 : 1	214 254 : 1	253 271 : 2	201 225 : 2
1.05e-09	2.28e-09	1.68e-09	3.93e-10
211 261 : 1	255 262 : 1	209 253 : 1	214 306 : 16
1.05e-09	1.15e-09	1.80e-09	2.64e-09
209 261 : 1	255 270 : 1	210 253 : 1	214 307 : 17
1.05e-09	1.68e-09	1.80e-09	2.64e-09
211 258 : 2	211 253 : 1	250 253 : 3	214 305 : 15

1.38e-09	1.80e-09	3.74e-10	2.64e-09
209 258 : 4	253 270 : 1	202 226 : 2	213 307 : 5
1.38e-09	1.68e-09	3.93e-10	2.64e-09
211 227 : 1	250 254 : 1	202 227 : 2	228 307 : 2
1.49e-09	3.74e-10	3.93e-10	1.67e-09
211 290 : 1	255 306 : 2	198 227 : 15	299 307 : 5
1.34e-09	1.42e-09	9.41e-10	1.80e-09
206 211 : 1	253 306 : 2	227 250 : 2	213 297 : 2
6.61e-10	1.42e-09	1.13e-09	1.87e-09
209 214 : 1	255 261 : 2	202 225 : 2	214 297 : 6
4.25e-10	1.15e-09	3.93e-10	1.87e-09
214 298 : 7	286 307 : 1	203 290 : 2	199 209 : 1
1.87e-09	4.59e-10	2.25e-09	1.52e-09
298 305 : 19	282 307 : 3	201 250 : 6	199 255 : 1
1.80e-09	1.15e-09	1.43e-09	1.23e-09
299 305 : 14	290 307 : 3	197 303 : 1	199 303 : 1
1.80e-09	9.06e-10	2.30e-09	2.30e-09
299 306 : 3	297 307 : 16	197 269 : 2	199 261 : 1
1.80e-09	1.80e-09	2.85e-09	2.01e-09
297 306 : 4	299 308 : 13	197 209 : 3	231 255 : 1
1.80e-09	1.80e-09	1.52e-09	1.14e-09
298 307 : 11	297 308 : 3	197 255 : 4	229 253 : 2
1.80e-09	1.80e-09	1.23e-09	1.14e-09
228 299 : 1	206 305 : 1	198 303 : 2	206 230 : 31
2.47e-09	1.72e-09	2.30e-09	6.30e-10
202 297 : 1	286 308 : 3	198 269 : 1	206 231 : 42
2.88e-09	4.59e-10	2.85e-09	6.30e-10

202 214 : 1	192 214 : 1	198 210 : 1	206 229 : 20
1.31e-09	1.40e-09	1.52e-09	6.30e-10
210 215 : 1	214 299 : 1	198 253 : 2	205 230 : 11
4.25e-10	1.87e-09	1.23e-09	6.30e-10
215 255 : 1	214 257 : 1	199 269 : 3	207 229 : 38
2.28e-09	1.63e-09	2.85e-09	6.30e-10
213 302 : 1	192 201 : 2	199 210 : 2	207 230 : 1
2.80e-09	1.12e-09	1.52e-09	6.30e-10
213 271 : 1	191 201 : 1	199 253 : 2	207 231 : 26
1.31e-09	1.12e-09	1.23e-09	6.30e-10
214 290 : 1	191 202 : 3	199 301 : 2	211 229 : 1
1.71e-09	1.12e-09	2.30e-09	9.79e-10
214 227 : 1	189 202 : 1	197 253 : 2	205 231 : 31
1.64e-09	1.12e-09	1.23e-09	6.30e-10
206 214 : 1	198 202 : 6	197 301 : 2	205 229 : 33
8.15e-10	4.76e-10	2.30e-09	6.30e-10
215 307 : 4	203 302 : 2	197 211 : 3	231 269 : 1
2.64e-09	1.78e-09	1.52e-09	1.66e-09
215 305 : 1	202 206 : 4	198 270 : 1	229 269 : 3
2.64e-09	7.35e-10	2.85e-09	1.66e-09
215 306 : 1	202 250 : 2	198 211 : 1	229 271 : 3
2.64e-09	1.43e-09	1.52e-09	1.66e-09
213 306 : 2	198 203 : 4	199 270 : 2	231 271 : 1
2.64e-09	4.76e-10	2.85e-09	1.66e-09
213 305 : 3	199 203 : 4	199 211 : 2	205 232 : 13
2.64e-09	4.76e-10	1.52e-09	6.30e-10
286 305 : 6	199 201 : 5	199 302 : 3	207 232 : 24

4.59e-10	4.76e-10	2.30e-09	6.30e-10
282 305 : 3	197 201 : 2	197 254 : 3	208 229 : 2
1.15e-09	4.76e-10	1.23e-09	6.30e-10
266 307 : 8	197 202 : 3	197 271 : 3	208 232 : 7
1.87e-09	4.76e-10	2.85e-09	6.30e-10
266 305 : 3	202 255 : 1	197 250 : 1	205 225 : 2
1.87e-09	9.85e-10	1.17e-09	1.05e-09
202 305 : 1	203 206 : 1	199 250 : 3	201 208 : 1
2.18e-09	7.35e-10	1.17e-09	7.35e-10
290 305 : 3	203 263 : 1	199 271 : 3	208 324 : 2
9.06e-10	1.77e-09	2.85e-09	2.55e-09
297 305 : 7	203 261 : 1	197 302 : 3	202 208 : 1
1.80e-09	1.77e-09	2.30e-09	7.35e-10
208 231 : 1	282 294 : 3	261 281 : 1	238 286 : 6
6.30e-10	1.49e-09	1.26e-09	1.64e-09
261 308 : 1	282 286 : 16	267 282 : 2	287 289 : 4
1.24e-09	3.79e-10	1.75e-09	5.87e-10
262 308 : 1	281 286 : 14	250 283 : 2	250 285 : 2
1.24e-09	3.79e-10	8.81e-10	1.40e-09
202 262 : 1	282 287 : 2	281 310 : 9	234 285 : 2
1.77e-09	3.79e-10	1.43e-09	1.48e-09
262 294 : 1	282 285 : 2	282 310 : 3	285 314 : 8
8.50e-10	3.79e-10	1.43e-09	1.38e-09
234 263 : 5	250 282 : 2	282 314 : 3	286 314 : 4
6.76e-10	8.81e-10	1.73e-09	1.38e-09
263 282 : 12	234 282 : 5	283 310 : 8	285 290 : 2
1.26e-09	1.71e-09	1.43e-09	5.87e-10

262 282 : 2	283 294 : 4	283 314 : 3	286 290 : 1
1.26e-09	1.49e-09	1.73e-09	5.87e-10
261 282 : 12	264 283 : 5	281 314 : 2	284 286 : 1
1.26e-09	1.26e-09	1.73e-09	3.79e-10
205 261 : 1	264 281 : 5	242 281 : 6	284 287 : 2
1.32e-09	1.26e-09	2.40e-09	3.79e-10
263 286 : 3	264 282 : 7	238 282 : 4	287 294 : 3
1.06e-09	1.26e-09	2.25e-09	1.06e-09
263 266 : 10	267 303 : 1	242 282 : 1	287 310 : 10
6.67e-10	2.05e-09	2.40e-09	1.04e-09
263 294 : 1	265 303 : 1	242 283 : 5	285 310 : 5
8.50e-10	2.05e-09	2.40e-09	1.04e-09
234 261 : 2	265 301 : 1	238 283 : 7	287 314 : 4
6.76e-10	2.05e-09	2.25e-09	1.38e-09
250 261 : 2	282 290 : 13	238 281 : 9	241 287 : 2
1.24e-09	8.19e-10	2.25e-09	2.30e-09
261 266 : 14	283 290 : 8	281 294 : 4	237 285 : 2
6.67e-10	8.19e-10	1.49e-09	1.64e-09
261 265 : 2	281 290 : 4	283 300 : 3	285 299 : 4
6.67e-10	8.19e-10	1.87e-09	1.43e-09
264 267 : 2	266 282 : 4	281 300 : 3	285 297 : 2
6.67e-10	1.75e-09	1.87e-09	1.43e-09
264 265 : 1	263 283 : 1	234 283 : 2	264 287 : 6
6.67e-10	1.26e-09	1.71e-09	1.06e-09
263 265 : 1	266 283 : 1	234 281 : 2	263 287 : 5
6.67e-10	1.75e-09	1.71e-09	1.06e-09
261 294 : 3	281 291 : 9	283 287 : 14	261 287 : 1

8.50e-10	8.19e-10	3.79e-10	1.06e-09
264 294 : 6	263 281 : 3	283 285 : 7	234 287 : 2
8.50e-10	1.26e-09	3.79e-10	1.48e-09
234 264 : 1	282 291 : 1	286 310 : 7	285 309 : 2
6.76e-10	8.19e-10	1.04e-09	1.04e-09
261 285 : 4	282 289 : 2	242 286 : 9	281 288 : 2
1.06e-09	8.19e-10	2.30e-09	3.79e-10
264 285 : 7	283 286 : 14	242 287 : 11	264 288 : 1
1.06e-09	3.79e-10	2.30e-09	1.06e-09
264 286 : 2	281 287 : 8	238 287 : 9	263 285 : 3
1.06e-09	3.79e-10	1.64e-09	1.06e-09
264 266 : 2	281 285 : 10	238 285 : 15	284 288 : 1
6.67e-10	3.79e-10	1.64e-09	3.79e-10
261 289 : 4	250 281 : 2	242 285 : 10	290 314 : 5
7.65e-10	8.81e-10	2.30e-09	8.32e-10
291 314 : 2	202 291 : 2	259 266 : 15	259 268 : 1
8.32e-10	2.25e-09	9.10e-10	9.10e-10
289 314 : 1	249 291 : 1	257 266 : 8	259 310 : 8
8.32e-10	1.80e-09	9.10e-10	1.30e-09
239 289 : 2	205 291 : 1	258 291 : 4	257 310 : 10
1.29e-09	1.82e-09	1.14e-09	1.30e-09
242 289 : 4	215 289 : 2	258 289 : 5	246 311 : 23
1.71e-09	1.71e-09	1.14e-09	2.91e-09
242 290 : 3	249 289 : 1	205 258 : 3	246 309 : 26
1.71e-09	1.80e-09	1.14e-09	2.91e-09
242 291 : 3	250 291 : 3	259 289 : 1	246 310 : 30
1.71e-09	1.80e-09	1.14e-09	2.91e-09

243 290 : 2	265 291 : 6	257 292 : 2	274 309 : 19
1.71e-09	8.15e-10	1.14e-09	2.07e-09
241 290 : 2	213 291 : 1	258 292 : 3	274 310 : 19
1.71e-09	1.71e-09	1.14e-09	2.07e-09
288 290 : 4	202 289 : 1	259 265 : 10	274 311 : 18
5.87e-10	2.25e-09	9.10e-10	2.07e-09
261 290 : 1	203 289 : 5	257 265 : 10	245 310 : 3
7.65e-10	2.25e-09	9.10e-10	2.91e-09
261 291 : 1	203 291 : 1	257 267 : 18	245 311 : 11
7.65e-10	2.25e-09	9.10e-10	2.91e-09
264 291 : 2	201 291 : 4	208 258 : 2	247 311 : 14
7.65e-10	2.25e-09	1.14e-09	2.91e-09
263 291 : 1	234 289 : 1	267 292 : 1	257 311 : 1
7.65e-10	1.04e-09	8.15e-10	1.30e-09
263 289 : 2	289 294 : 1	265 292 : 2	259 311 : 3
7.65e-10	6.87e-10	8.15e-10	1.30e-09
264 290 : 3	265 289 : 1	208 265 : 3	259 309 : 2
7.65e-10	8.15e-10	1.77e-09	1.30e-09
250 290 : 3	283 289 : 3	259 267 : 12	247 310 : 1
1.80e-09	8.19e-10	9.10e-10	2.91e-09
284 290 : 5	285 291 : 2	267 301 : 1	247 309 : 15
8.19e-10	5.87e-10	2.05e-09	2.91e-09
287 291 : 3	266 289 : 7	267 271 : 2	273 311 : 5
5.87e-10	8.15e-10	7.10e-10	2.07e-09
250 289 : 3	285 289 : 2	209 267 : 2	245 309 : 14
1.80e-09	5.87e-10	1.25e-09	2.91e-09
224 289 : 16	284 291 : 7	255 267 : 3	257 314 : 6

1.99e-09	8.19e-10	1.48e-09	1.91e-09
224 290 : 10	288 289 : 5	267 302 : 3	259 313 : 2
1.99e-09	5.87e-10	2.05e-09	1.91e-09
223 290 : 6	267 289 : 5	209 265 : 4	245 314 : 22
1.99e-09	8.15e-10	1.25e-09	2.55e-09
221 290 : 3	267 291 : 3	255 265 : 2	246 314 : 25
1.99e-09	8.15e-10	1.48e-09	2.55e-09
224 291 : 13	266 291 : 4	265 302 : 2	247 314 : 11
1.99e-09	8.15e-10	2.05e-09	2.55e-09
283 291 : 8	284 292 : 3	265 269 : 3	247 315 : 21
8.19e-10	8.19e-10	7.10e-10	2.55e-09
281 289 : 6	258 266 : 10	267 269 : 1	245 313 : 16
8.19e-10	9.10e-10	7.10e-10	2.55e-09
284 289 : 6	258 267 : 18	253 267 : 1	247 313 : 27
8.19e-10	9.10e-10	1.48e-09	2.55e-09
199 291 : 1	258 265 : 30	265 287 : 1	274 314 : 7
2.64e-09	9.10e-10	1.48e-09	1.69e-09
275 314 : 5	242 311 : 2	247 293 : 1	275 311 : 4
1.69e-09	2.33e-09	2.19e-09	2.07e-09
273 315 : 4	243 311 : 5	245 293 : 3	274 316 : 4
1.69e-09	2.33e-09	2.19e-09	1.69e-09
274 313 : 14	241 309 : 3	257 294 : 3	275 316 : 1
1.69e-09	2.33e-09	1.42e-09	1.69e-09
274 315 : 11	241 313 : 6	259 295 : 3	275 315 : 4
1.69e-09	1.65e-09	1.42e-09	1.69e-09
246 313 : 4	242 315 : 9	259 293 : 3	242 309 : 3
2.55e-09	1.65e-09	1.42e-09	2.33e-09

245 315 : 15	239 315 : 6	257 293 : 1	242 312 : 1
2.55e-09	1.24e-09	1.42e-09	2.33e-09
259 314 : 3	237 313 : 3	284 294 : 1	243 309 : 1
1.91e-09	1.24e-09	1.49e-09	2.33e-09
259 315 : 1	238 313 : 1	268 294 : 2	241 257 : 3
1.91e-09	1.24e-09	4.84e-10	1.69e-09
257 315 : 2	239 313 : 6	241 295 : 1	241 259 : 5
1.91e-09	1.24e-09	1.31e-09	1.69e-09
257 313 : 2	243 313 : 3	242 295 : 1	242 259 : 7
1.91e-09	1.65e-09	1.31e-09	1.69e-09
246 315 : 5	294 309 : 9	242 293 : 2	242 273 : 8
2.55e-09	9.26e-10	1.31e-09	1.17e-09
273 314 : 2	294 311 : 11	239 293 : 1	243 273 : 15
1.69e-09	9.26e-10	1.01e-09	1.17e-09
245 294 : 5	293 309 : 2	237 293 : 1	243 275 : 13
2.19e-09	9.26e-10	1.01e-09	1.17e-09
242 247 : 7	294 312 : 3	237 294 : 2	241 275 : 9
9.55e-10	9.26e-10	1.01e-09	1.17e-09
242 245 : 7	295 309 : 2	247 312 : 2	241 273 : 9
9.55e-10	9.26e-10	2.91e-09	1.17e-09
237 245 : 2	248 295 : 1	245 312 : 4	243 248 : 9
1.39e-09	2.19e-09	2.91e-09	9.55e-10
238 245 : 5	248 293 : 3	248 311 : 9	237 259 : 1
1.39e-09	2.19e-09	2.91e-09	1.34e-09
238 248 : 12	248 294 : 9	248 309 : 4	238 257 : 8
1.39e-09	2.19e-09	2.91e-09	1.34e-09
238 247 : 5	278 294 : 1	248 315 : 9	238 259 : 5

1.39e-09	1.75e-09	2.55e-09	1.34e-09
243 245 : 2	273 294 : 6	248 312 : 10	238 275 : 9
9.55e-10	1.43e-09	2.91e-09	1.34e-09
241 245 : 3	274 294 : 5	248 313 : 6	239 275 : 8
9.55e-10	1.43e-09	2.55e-09	1.34e-09
241 248 : 16	275 294 : 14	273 309 : 4	237 275 : 12
9.55e-10	1.43e-09	2.07e-09	1.34e-09
242 248 : 3	273 295 : 7	274 312 : 10	237 273 : 7
9.55e-10	1.43e-09	2.07e-09	1.34e-09
239 247 : 3	273 293 : 4	275 312 : 5	238 273 : 11
1.39e-09	1.43e-09	2.07e-09	1.34e-09
237 247 : 2	247 294 : 3	275 313 : 7	237 248 : 14
1.39e-09	2.19e-09	1.69e-09	1.39e-09
243 247 : 1	274 295 : 2	273 316 : 5	242 257 : 5
9.55e-10	1.43e-09	1.69e-09	1.69e-09
234 245 : 1	274 293 : 4	275 309 : 3	242 275 : 5
1.69e-09	1.43e-09	2.07e-09	1.17e-09
241 311 : 1	275 295 : 2	273 312 : 9	238 260 : 6
2.33e-09	1.43e-09	2.07e-09	1.34e-09
239 259 : 1	244 275 : 3	263 308 : 2	237 249 : 1
1.34e-09	1.17e-09	1.24e-09	2.00e-09
239 273 : 8	235 276 : 5	203 228 : 1	250 276 : 9
1.34e-09	1.90e-09	3.93e-10	2.42e-09
239 276 : 16	233 276 : 3	203 303 : 1	250 312 : 4
1.34e-09	1.90e-09	1.78e-09	2.13e-09
243 257 : 4	250 292 : 1	197 260 : 1	251 281 : 2
1.69e-09	1.80e-09	1.65e-09	8.81e-10

243 316 : 7	250 324 : 4	207 211 : 1	251 285 : 1
1.65e-09	1.04e-09	6.61e-10	1.40e-09
241 316 : 2	251 324 : 1	207 308 : 1	249 285 : 1
1.65e-09	1.04e-09	1.72e-09	1.40e-09
241 315 : 4	249 324 : 1	261 300 : 1	249 288 : 2
1.65e-09	1.04e-09	1.49e-09	1.40e-09
241 276 : 14	220 249 : 1	239 287 : 1	249 276 : 1
1.17e-09	2.77e-09	1.64e-09	2.42e-09
243 276 : 10	228 250 : 1	285 292 : 1	250 288 : 1
1.17e-09	1.13e-09	5.87e-10	1.40e-09
237 257 : 1	250 294 : 2	244 291 : 1	251 288 : 1
1.34e-09	2.05e-09	1.71e-09	1.40e-09
237 260 : 2	250 303 : 1	271 291 : 1	251 284 : 1
1.34e-09	1.23e-09	1.48e-09	8.81e-10
237 276 : 11	250 269 : 1	203 259 : 1	197 278 : 2
1.34e-09	2.29e-09	1.29e-09	1.38e-09
239 248 : 5	251 269 : 1	283 313 : 1	278 319 : 2
1.39e-09	2.29e-09	1.73e-09	2.66e-09
243 315 : 1	211 249 : 2	264 309 : 1	259 278 : 1
1.65e-09	2.11e-09	8.52e-10	1.29e-09
233 267 : 1	250 264 : 1	295 313 : 1	267 278 : 2
1.01e-09	1.24e-09	6.15e-10	2.06e-09
233 275 : 3	250 278 : 12	248 275 : 1	277 291 : 1
1.90e-09	3.93e-10	1.13e-09	1.57e-09
234 273 : 3	251 278 : 5	248 276 : 1	271 277 : 1
1.90e-09	3.93e-10	1.13e-09	2.55e-09
234 276 : 3	249 278 : 6	244 285 : 1	278 300 : 4

1.90e-09	3.93e-10	2.30e-09	2.33e-09
234 259 : 2	202 251 : 1	233 239 : 1	278 332 : 2
1.13e-09	1.43e-09	7.51e-10	3.90e-09
235 257 : 4	251 271 : 1	239 244 : 2	213 278 : 1
1.13e-09	2.29e-09	5.70e-10	2.96e-09
233 260 : 4	211 251 : 1	237 244 : 1	214 279 : 2
1.13e-09	2.11e-09	5.70e-10	2.96e-09
234 267 : 1	249 253 : 1	234 248 : 1	265 277 : 2
1.01e-09	3.74e-10	1.69e-09	2.06e-09
234 244 : 4	249 264 : 1	220 324 : 1	220 278 : 1
9.62e-10	1.24e-09	4.14e-09	3.90e-09
244 276 : 7	202 249 : 2	185 327 : 1	278 291 : 1
1.17e-09	1.43e-09	2.77e-09	1.57e-09
240 276 : 6	203 250 : 2	185 319 : 1	257 278 : 1
1.34e-09	1.43e-09	4.14e-09	1.29e-09
240 275 : 5	278 324 : 5	193 324 : 1	271 278 : 1
1.34e-09	7.75e-10	4.14e-09	2.55e-09
240 273 : 3	189 325 : 1	324 331 : 2	214 278 : 1
1.34e-09	2.42e-09	2.85e-09	2.96e-09
244 273 : 3	317 327 : 1	324 327 : 1	224 278 : 1
1.17e-09	1.87e-09	1.28e-09	8.94e-10
114 279 : 1	291 327 : 3	285 317 : 2	292 317 : 4
6.51e-10	1.23e-09	1.78e-09	1.31e-09
66 279 : 1	289 327 : 5	285 319 : 4	292 320 : 7
1.42e-09	1.23e-09	1.78e-09	1.31e-09
284 325 : 7	289 328 : 3	288 319 : 8	285 320 : 2
1.36e-09	1.23e-09	1.78e-09	1.78e-09

284 327 : 2	292 328 : 8	281 319 : 1	292 319 : 6
1.36e-09	1.23e-09	2.08e-09	1.31e-09
283 327 : 4	292 327 : 2	281 317 : 2	291 317 : 3
1.36e-09	1.23e-09	2.08e-09	1.31e-09
288 325 : 8	291 325 : 2	283 317 : 2	291 320 : 3
8.52e-10	1.23e-09	2.08e-09	1.31e-09
287 325 : 8	289 325 : 2	288 317 : 1	289 320 : 4
8.52e-10	1.23e-09	1.78e-09	1.31e-09
285 327 : 4	191 328 : 3	288 320 : 4	189 319 : 2
8.52e-10	2.42e-09	1.78e-09	3.90e-09
281 327 : 1	189 328 : 2	302 319 : 1	192 320 : 11
1.36e-09	2.42e-09	1.87e-09	3.90e-09
281 325 : 2	192 328 : 6	269 319 : 1	291 319 : 2
1.36e-09	2.42e-09	8.92e-10	1.31e-09
288 328 : 6	192 327 : 5	209 319 : 1	225 320 : 4
8.52e-10	2.42e-09	2.41e-09	2.96e-09
288 327 : 3	292 325 : 3	255 319 : 1	228 320 : 10
8.52e-10	1.23e-09	2.51e-09	2.96e-09
301 327 : 1	291 328 : 4	302 317 : 2	228 319 : 4
1.07e-09	1.23e-09	1.87e-09	2.96e-09
271 327 : 1	227 327 : 5	269 317 : 2	227 319 : 2
2.55e-09	2.09e-09	8.92e-10	2.96e-09
211 327 : 1	225 325 : 6	209 317 : 1	227 317 : 2
2.67e-09	2.09e-09	2.41e-09	2.96e-09
253 327 : 1	225 328 : 2	255 317 : 1	225 317 : 2
1.71e-09	2.09e-09	2.51e-09	2.96e-09
301 325 : 1	228 328 : 6	209 320 : 1	225 319 : 3

1.07e-09	2.09e-09	2.41e-09	2.96e-09
269 325 : 2	227 328 : 7	255 320 : 1	228 317 : 2
2.55e-09	2.09e-09	2.51e-09	2.96e-09
209 325 : 1	225 327 : 2	303 320 : 1	227 320 : 4
2.67e-09	2.09e-09	1.87e-09	2.96e-09
253 325 : 1	228 327 : 3	271 320 : 1	224 320 : 1
1.71e-09	2.09e-09	8.92e-10	3.81e-09
302 325 : 1	228 325 : 2	211 320 : 1	193 283 : 3
1.07e-09	2.09e-09	2.41e-09	3.11e-09
209 328 : 2	189 327 : 1	253 319 : 1	195 281 : 2
2.67e-09	2.42e-09	2.51e-09	3.11e-09
255 328 : 1	192 325 : 1	192 319 : 3	195 284 : 6
1.71e-09	2.42e-09	3.90e-09	3.11e-09
302 328 : 1	224 328 : 1	192 317 : 7	193 284 : 4
1.07e-09	2.55e-09	3.90e-09	3.11e-09
269 328 : 1	284 317 : 5	191 320 : 4	193 288 : 13
2.55e-09	2.08e-09	3.90e-09	2.41e-09
255 327 : 1	284 319 : 2	287 320 : 4	195 288 : 9
1.71e-09	2.08e-09	1.78e-09	2.41e-09
191 327 : 2	283 319 : 4	287 319 : 1	195 287 : 2
2.42e-09	2.08e-09	1.78e-09	2.41e-09
285 328 : 3	287 317 : 6	289 317 : 3	195 285 : 1
8.52e-10	1.78e-09	1.31e-09	2.41e-09
193 281 : 1	193 287 : 2	196 227 : 6	188 281 : 2
3.11e-09	2.41e-09	2.25e-09	1.75e-09
193 303 : 2	193 285 : 1	196 225 : 4	188 303 : 1
3.90e-09	2.41e-09	2.25e-09	2.21e-09

193 271 : 2	196 285 : 1	195 227 : 3	188 271 : 1
1.75e-09	2.41e-09	2.25e-09	2.99e-09
193 211 : 2	196 288 : 3	195 225 : 1	188 211 : 1
1.04e-09	2.41e-09	2.25e-09	1.28e-09
196 253 : 1	196 289 : 4	193 225 : 1	188 253 : 1
2.55e-09	2.00e-09	2.25e-09	1.30e-09
196 303 : 1	195 292 : 6	196 224 : 1	188 301 : 1
3.90e-09	2.00e-09	2.55e-09	2.21e-09
196 271 : 1	195 291 : 3	185 284 : 4	187 269 : 2
1.75e-09	2.00e-09	1.75e-09	2.99e-09
195 211 : 2	196 292 : 10	188 284 : 4	187 209 : 1
1.04e-09	2.00e-09	1.75e-09	1.28e-09
195 253 : 2	193 291 : 2	187 284 : 4	187 255 : 1
2.55e-09	2.00e-09	1.75e-09	1.30e-09
195 303 : 1	193 289 : 1	187 283 : 1	187 301 : 1
3.90e-09	2.00e-09	1.75e-09	2.21e-09
195 271 : 1	193 292 : 4	187 288 : 5	185 209 : 2
1.75e-09	2.00e-09	2.04e-09	1.28e-09
193 253 : 1	196 291 : 2	185 287 : 2	185 255 : 1
2.55e-09	2.00e-09	2.04e-09	1.30e-09
192 193 : 2	195 228 : 7	185 288 : 8	185 301 : 1
8.52e-10	2.25e-09	2.04e-09	2.21e-09
192 196 : 12	193 228 : 7	188 288 : 1	185 269 : 1
8.52e-10	2.25e-09	2.04e-09	2.99e-09
192 195 : 5	196 228 : 4	188 283 : 1	188 255 : 1
8.52e-10	2.25e-09	1.75e-09	1.30e-09

192 319 327 : 3	198 202 211 : 1	213 297 307 : 2
1.00e-08 1.00e-08	1.95e-09 1.94e-09	4.04e-09 3.98e-09
185 192 195 : 2	254 298 305 : 1	214 297 307 : 1
1.94e-09 1.95e-09	4.14e-09 4.43e-09	4.04e-09 3.98e-09
185 191 195 : 2	254 298 306 : 1	214 299 308 : 1
1.94e-09 1.95e-09	4.14e-09 4.43e-09	4.04e-09 3.98e-09
191 319 327 : 1	254 299 306 : 3	202 210 253 : 2
1.00e-08 1.00e-08	4.14e-09 4.43e-09	2.42e-09 2.77e-09
191 317 325 : 1	254 297 306 : 4	202 211 253 : 2
1.00e-08 1.00e-08	4.14e-09 4.43e-09	2.42e-09 2.77e-09
187 189 195 : 1	198 202 254 : 3	202 211 254 : 2
1.94e-09 1.95e-09	1.95e-09 1.94e-09	2.42e-09 2.77e-09
189 317 325 : 2	255 298 306 : 1	203 211 254 : 2
1.00e-08 1.00e-08	4.14e-09 4.43e-09	2.42e-09 2.77e-09
187 192 193 : 4	198 202 226 : 17	203 209 254 : 1
1.94e-09 1.95e-09	1.22e-09 8.01e-10	2.42e-09 2.77e-09
192 317 325 : 2	198 202 227 : 10	201 209 255 : 1
1.00e-08 1.00e-08	1.22e-09 8.01e-10	2.42e-09 2.77e-09
298 302 306 : 6	198 202 225 : 18	202 209 255 : 2
4.14e-09 4.43e-09	1.22e-09 8.01e-10	2.42e-09 2.77e-09
198 202 302 : 3	198 201 227 : 2	203 209 255 : 1
2.77e-09 2.77e-09	1.22e-09 8.01e-10	2.42e-09 2.77e-09
298 303 306 : 1	197 202 225 : 2	201 299 308 : 1
4.14e-09 4.43e-09	1.22e-09 8.01e-10	4.59e-09 4.59e-09
270 298 306 : 9	197 202 227 : 3	201 215 297 : 1
2.37e-09 2.37e-09	1.22e-09 8.01e-10	1.00e-08 1.00e-08

271 298 306 : 3	198 203 227 : 2	202 297 308 : 1
2.37e-09 2.37e-09	1.22e-09 8.01e-10	4.59e-09 4.59e-09
198 202 271 : 2	199 201 227 : 2	202 269 303 : 1
4.04e-09 4.04e-09	1.22e-09 8.01e-10	4.59e-09 4.59e-09
198 202 269 : 1	197 201 227 : 3	261 297 308 : 14
4.04e-09 4.04e-09	1.22e-09 8.01e-10	3.55e-09 3.18e-09
269 298 306 : 1	199 203 227 : 1	262 297 308 : 4
2.37e-09 2.37e-09	1.22e-09 8.01e-10	3.55e-09 3.18e-09
269 298 307 : 1	198 203 225 : 1	262 297 307 : 3
2.37e-09 2.37e-09	1.22e-09 8.01e-10	3.55e-09 3.18e-09
270 298 305 : 2	199 203 225 : 2	262 299 307 : 9
2.37e-09 2.37e-09	1.22e-09 8.01e-10	3.55e-09 3.18e-09
210 297 306 : 1	199 201 225 : 1	262 299 305 : 4
4.14e-09 4.43e-09	1.22e-09 8.01e-10	3.55e-09 3.18e-09
210 298 306 : 7	214 297 306 : 1	263 299 305 : 3
4.14e-09 4.43e-09	4.04e-09 3.98e-09	3.55e-09 3.18e-09
210 298 307 : 1	214 298 306 : 1	263 299 308 : 5
4.14e-09 4.43e-09	4.04e-09 3.98e-09	3.55e-09 3.18e-09
198 202 210 : 1	214 298 307 : 2	262 299 308 : 7
1.95e-09 1.94e-09	4.04e-09 3.98e-09	3.55e-09 3.18e-09
198 202 209 : 1	214 298 305 : 2	263 297 305 : 2
1.95e-09 1.94e-09	4.04e-09 3.98e-09	3.55e-09 3.18e-09
210 299 305 : 1	215 298 305 : 1	263 297 308 : 16
4.14e-09 4.43e-09	4.04e-09 3.98e-09	3.55e-09 3.18e-09
211 298 305 : 1	215 299 305 : 2	261 299 308 : 9
4.14e-09 4.43e-09	4.04e-09 3.98e-09	3.55e-09 3.18e-09
210 298 305 : 1	213 299 305 : 1	263 297 307 : 3

4.14e-09	4.43e-09	4.04e-09	3.98e-09	3.55e-09	3.18e-09
211 298	307 : 2	213 299	307 : 1	261 297	305 : 2
4.14e-09	4.43e-09	4.04e-09	3.98e-09	3.55e-09	3.18e-09
261 299	305 : 1	2.73e-09	2.77e-09	220 250	332 : 3
3.55e-09	3.18e-09	234 241	294 : 1	4.59e-09	4.59e-09
261 299	307 : 2	2.73e-09	2.77e-09	220 251	332 : 1
3.55e-09	3.18e-09	234 237	241 : 8	4.59e-09	4.59e-09
262 297	305 : 2	1.61e-09	1.58e-09	234 251	295 : 2
3.55e-09	3.18e-09	234 237	243 : 15	4.04e-09	4.14e-09
263 300	308 : 9	1.61e-09	1.58e-09	249 281	316 : 1
3.55e-09	3.18e-09	233 237	243 : 4	4.14e-09	4.43e-09
261 300	308 : 9	1.61e-09	1.58e-09	249 284	316 : 1
3.55e-09	3.18e-09	234 239	241 : 9	4.14e-09	4.43e-09
261 300	307 : 2	1.61e-09	1.58e-09	278 319	327 : 3
3.55e-09	3.18e-09	234 237	244 : 6	4.59e-09	4.59e-09
263 299	307 : 2	1.61e-09	1.58e-09	201 278	303 : 1
3.55e-09	3.18e-09	235 237	243 : 2	2.77e-09	2.77e-09
261 282	286 : 5	1.61e-09	1.58e-09	278 324	329 : 1
1.94e-09	1.95e-09	235 239	243 : 2	4.59e-09	4.50e-09
263 282	286 : 6	1.61e-09	1.58e-09	220 278	332 : 2
1.94e-09	1.95e-09	233 239	243 : 5	4.59e-09	4.00e-09
263 286	290 : 3	1.61e-09	1.58e-09	228 278	308 : 1
1.61e-09	1.61e-09	234 239	244 : 14	3.46e-09	3.30e-09
261 286	290 : 2	1.61e-09	1.58e-09	213 228	278 : 1
1.61e-09	1.61e-09	235 239	244 : 2	4.59e-09	4.50e-09
234 242	295 : 1	1.61e-09	1.58e-09	201 278	301 : 1
2.73e-09	2.77e-09	235 237	241 : 4	2.77e-09	2.77e-09

234 243 295 : 3	1.61e-09 1.58e-09	214 228 278 : 1
2.73e-09 2.77e-09	233 240 241 : 4	4.59e-09 4.50e-09
234 238 293 : 1	1.61e-09 1.58e-09	279 324 332 : 1
2.36e-09 2.36e-09	233 239 244 : 1	4.59e-09 4.50e-09
234 239 293 : 3	1.61e-09 1.58e-09	277 317 325 : 1
2.36e-09 2.36e-09	235 240 241 : 4	4.59e-09 4.59e-09
234 243 293 : 1	1.61e-09 1.58e-09	227 278 308 : 1
2.73e-09 2.77e-09	233 240 244 : 4	3.46e-09 3.30e-09
234 243 294 : 2	1.61e-09 1.58e-09	

A.2 MCC2 Netlist

30 31 : 2
8.558187e-10
3 19 : 2
1.392610e-09
3 31 : 2
2.236338e-09
3 21 : 14
1.286569e-09
3 32 : 23
2.190876e-09
3 33 : 20
2.474589e-09
4 33 : 15

2.110956e-09

4 32 : 14

1.884742e-09

4 15 : 2

1.944457e-09

4 24 : 7

9.587758e-10

4 26 : 1

1.932803e-09

27 34 : 1

1.679307e-09

4 11 : 2

2.303777e-09

4 21 : 10

1.417045e-09

4 20 : 4

1.111004e-09

4 31 : 4

1.907195e-09

5 24 : 4

1.164918e-09

5 25 : 3

1.024050e-09

5 28 : 6

9.663066e-10

5 29 : 36

7.930719e-10

5 30 : 14
1.606588e-09
6 29 : 5
1.024050e-09
6 30 : 4
1.000641e-09
6 28 : 4
1.000641e-09
6 19 : 4
2.371572e-09
16 19 : 19
1.846918e-09
16 22 : 6
8.275937e-10
15 19 : 7
1.780808e-09
7 15 : 3
2.056522e-09
7 16 : 34
1.600827e-09
7 18 : 42
1.536823e-09
7 17 : 11
1.559997e-09
28 31 : 2
1.693314e-09
8 17 : 48

1.191867e-09

8 18 : 2

8.275937e-10

8 28 : 1

2.091589e-09

8 11 : 3

2.170789e-09

8 12 : 1

2.162915e-09

9 14 : 42

1.375603e-09

9 13 : 6

1.249362e-09

9 12 : 16

1.780808e-09

10 12 : 17

1.751243e-09

10 13 : 17

1.143481e-09

10 28 : 3

2.590423e-09

10 29 : 2

2.438143e-09

12 19 : 3

9.587758e-10

11 19 : 6

1.041165e-09

1 13 : 36
1.830380e-09
1 14 : 2
1.170631e-09
1 28 : 3
2.590423e-09
1 25 : 24
2.311725e-09
1 26 : 2
1.854570e-09
27 31 : 3
9.001821e-10
2 28 : 1
2.162915e-09
2 23 : 3
1.907195e-09
2 20 : 2
1.191867e-09
2 26 : 3
1.701484e-09
2 24 : 5
1.884742e-09
2 25 : 22
2.170789e-09
2 18 : 1
2.311725e-09
2 32 : 4

2.110956e-09
28 30 31 : 1
2.183297e-09 2.135618e-09
27 27 30 : 1
3.150290e-09 3.150290e-09
4 14 18 : 1
2.663126e-09 2.663126e-09
6 20 27 : 6
2.727163e-09 2.779389e-09
6 20 23 : 4
2.606339e-09 2.611852e-09
6 21 28 : 2
2.665122e-09 2.665122e-09
6 21 31 : 8
2.727163e-09 2.679813e-09
7 15 22 : 1
2.606339e-09 2.606339e-09
8 15 22 : 1
2.411570e-09 2.224875e-09
8 22 26 : 1
2.631564e-09 2.631564e-09
8 11 22 : 1
2.631564e-09 2.631564e-09
8 21 28 : 1
2.707275e-09 2.793748e-09
10 11 22 : 2
1.995411e-09 1.679307e-09

10 12 19 : 22
2.503296e-09 2.335979e-09
10 11 19 : 1
1.998768e-09 1.693314e-09
1 22 26 : 4
1.998768e-09 1.679307e-09
2 21 26 : 5
2.411570e-09 2.319945e-09
2 22 26 : 7
1.995411e-09 1.536823e-09
2 21 34 : 2
2.727163e-09 2.665122e-09
2 11 22 : 1
1.995411e-09 1.693314e-09
2 14 26 : 1
2.663126e-09 2.639020e-09
2 20 26 : 13
2.522122e-09 2.056522e-09
2 19 24 : 1
1.606588e-09 1.600000e-09
6 20 23 27 : 18
3.047159e-09 2.959762e-09 3.114925e-09
6 20 23 28 : 1
2.987125e-09 2.932265e-09 3.016742e-09
6 11 15 20 : 2
3.277636e-09 3.277636e-09 3.277636e-09
6 21 28 34 : 1

3.170980e-09 3.170980e-09 3.193713e-09
6 14 18 21 : 1
3.203192e-09 3.203192e-09 3.224456e-09
6 17 21 28 : 1
2.883962e-09 2.860964e-09 2.915197e-09
6 11 15 21 : 1
3.203192e-09 3.193713e-09 3.203192e-09
6 11 15 26 : 1
3.203192e-09 3.193713e-09 3.193713e-09
1 11 15 26 : 1
7.426493e-10 1.000000e-09 1.130480e-09
1 11 15 27 : 1
2.021899e-09 1.995411e-09 2.319945e-09
1 13 17 26 : 9
2.844201e-09 2.819338e-09 2.887457e-09
1 12 16 26 : 10
1.000000e-09 6.751943e-10 6.751943e-10
2 14 18 21 : 2
2.915197e-09 2.915197e-09 2.932265e-09
2 11 15 26 : 1
2.639020e-09 2.631564e-09 2.639020e-09
6 11 15 20 27 : 1
3.361708e-09 3.439595e-09 3.492332e-09 3.501728e-09
1 11 15 21 28 : 1
2.679813e-09 2.793748e-09 2.793748e-09 2.793748e-09
2 14 18 21 31 : 1
3.170980e-09 3.193713e-09 3.224456e-09 3.224456e-09

3 14 18 21 23 34 : 1

3.361708e-09 3.389365e-09 3.474842e-09 3.439595e-09 3.474842e-09

6 11 15 20 23 28 : 2

3.626911e-09 3.667658e-09 3.765321e-09 3.870319e-09 3.898948e-09

6 11 15 20 23 27 : 2

3.626911e-09 3.700068e-09 3.870319e-09 3.898948e-09 3.944243e-09

2 14 18 21 23 28 : 1

3.224456e-09 3.224456e-09 3.253358e-09 3.253358e-09 3.253358e-09

4 14 18 21 23 28 34 : 2

3.658836e-09 3.730726e-09 3.973665e-09 3.973665e-09 4.152545e-09

4.281626e-09

6 11 15 22 23 27 34 : 7

3.834603e-09 4.186674e-09 4.323136e-09 4.732713e-09 5.039084e-09

5.540168e-09

6 11 15 22 24 27 34 : 1

3.834603e-09 4.252939e-09 4.474361e-09 4.863346e-09 5.644897e-09

5.964141e-09

6 11 15 22 23 30 34 : 3

3.765321e-09 4.152545e-09 4.323136e-09 4.509933e-09 4.863346e-09

5.358833e-09

6 11 15 22 24 30 34 : 14

4.085753e-09 4.600834e-09 5.358833e-09 5.993835e-09 6.945959e-09

7.604576e-09

6 11 15 22 24 30 33 : 3

3.794072e-09 4.252939e-09 4.474361e-09 4.863346e-09 5.785037e-09

5.785037e-09

7 11 15 22 23 27 34 : 3

3.765321e-09 4.098084e-09 4.355209e-09 4.713373e-09 4.911712e-09
5.540168e-09
7 11 15 22 24 27 34 : 1
3.794072e-09 4.186674e-09 4.474361e-09 4.863346e-09 5.431446e-09
5.964141e-09
8 11 15 22 23 28 34 : 1
3.658836e-09 3.944243e-09 4.098084e-09 4.281626e-09 4.474361e-09
4.863346e-09
1 13 17 21 23 28 31 : 1
3.415505e-09 3.579115e-09 3.658836e-09 3.658836e-09 3.700068e-09
3.730726e-09
1 14 18 21 23 28 31 : 1
3.253358e-09 3.292134e-09 3.320627e-09 3.320627e-09 3.320627e-09
3.361708e-09
2 14 18 21 23 28 31 : 5
3.320627e-09 3.389365e-09 3.492332e-09 3.492332e-09 3.543975e-09
3.588761e-09
2 11 15 21 23 28 34 : 2
3.292134e-09 3.320627e-09 3.361708e-09 3.389365e-09 3.439595e-09
3.474842e-09
2 14 18 21 23 28 34 : 1
3.415505e-09 3.501728e-09 3.658836e-09 3.658836e-09 3.667658e-09
3.700068e-09

A.3 Point to Point Net Description

*Configuration parameters

Cfg point-to-point

brch_count 1

*Driver parameters

Drv_type abstract

Drv_file drv1 Rout 30 Cout 1e-12

*Receiver parameters

Rec_type abstract

Rec_file rec1 Rin 10e5 Cin 1e-12

*Branch parameters

Branch 1:

Brch_type_1 bstrip Brch_length_1 \$branch1

Height1A_1 8e-6 Height1B_1 8e-6 Thick1_1 4e-6 Width1_1 8e-6

Er1_1 3.5

Brch_load_1 loaded

*IOpad parameters

L_io .05e-9 C_io 0.2e-12

*Chip attachment parameters

L_catt_drv 0.1e-9 C_catt_drv 4e-13

L_catt_rec 0.1e-9 C_catt_rec 4e-13

Appendix B

Equations

B.1 Stochastic Model Equations

The model used for the problem above was:

$$Y(x) = \beta_0 + \sum_{j=1}^d \beta_j x_j + \phi(x) \quad (\text{B.1})$$

where x is the d dimensional parameter vector and $x_0 = 1$. β is a $(d + 1) \times 1$ vector of unknown coefficients. $\phi(\cdot)$ is a random process with mean zero and covariance:

$$V(w, x) = \sigma^2 \prod_{i=1}^d \exp(-|w_i - x_i|) \quad (\text{B.2})$$

Suppose that n values of $Y(x)$ are given at sample points s_1, \dots, s_n . These values are in the $n \times 1$ vector ζ_S . F is the $n \times (d + 1)$ matrix of the n parameter vectors at the sample sites augmented by a unit vector, i.e.

$$F = \begin{bmatrix} 1 & s_{11} & \dots & s_{1d} \\ \vdots & & & \\ 1 & s_{n1} & \dots & s_{nd} \end{bmatrix} \quad (\text{B.3})$$

Then the Best Linear Unbiased Predictor \hat{y} of $Y(\cdot)$ is given as:

$$\hat{y} = X\hat{\beta} + r'(x)R^{-1}(\zeta_S - F\hat{\beta}) \quad (\text{B.4})$$

where

$$\hat{\beta} = (F'R^{-1}F)^{-1}F'R^{-1}\zeta_S \quad (\text{B.5})$$

where, as before, R is the $n \times n$ covariance matrix of the stochastic function $\phi(\cdot)$ at the n sample locations, and r is the $n \times 1$ vector of covariances $V(x, s_k)$, $k = 1, \dots, n$, and $X = [1 \ x_1 \ \dots \ x_d]$. The Mean Squared Error (MSE) is given as:

$$\text{MSE}(\hat{y}(x)) = \sigma^2 \left([X \ r'(x)] \begin{bmatrix} 0 & F' \\ F & R \end{bmatrix}^{-1} \begin{bmatrix} X' \\ r(x) \end{bmatrix} \right) \quad (\text{B.6})$$

Appendix C

Intel Design

C.1 Netlist

```
class = "pent_pcmc_cont";  
5 9 : 11  
class = "pcmc_pent_cont";  
6 1 : 6  
class = "pcmc_pent_cont";  
6 4 : 1  
class = "addr_bus";  
5 6 30 : 2  
class = "addr_bus";  
4 6 30 : 1  
class = "addr_bus";  
5 6 33 : 1  
class = "addr_bus";  
5 6 18 : 6
```

```
class = "addr_bus";
4 7 21 : 5
class = "addr_bus";
4 6 18 : 1
class = "addr_bus";
5 7 18 : 1
class = "data_bus";
2 24 30 : 1
class = "data_bus";
1 24 30 : 1
class = "data_bus";
1 24 31 : 2
class = "data_bus";
1 23 30 : 2
class = "data_bus";
1 23 31 : 2
class = "data_bus";
1 22 31 : 1
class = "data_bus";
3 22 31 : 2
class = "data_bus";
1 22 30 : 1
class = "data_bus";
3 25 30 : 2
class = "data_bus";
1 25 31 : 1
class = "data_bus";
```

```
2 25 31 : 1
class = "data_bus";
2 12 18 : 2
class = "data_bus";
2 12 19 : 1
class = "data_bus";
3 12 19 : 1
class = "data_bus";
2 11 18 : 2
class = "data_bus";
2 11 19 : 1
class = "data_bus";
3 11 19 : 1
class = "data_bus";
2 10 19 : 2
class = "data_bus";
3 10 19 : 1
class = "data_bus";
2 10 18 : 1
class = "data_bus";
3 13 18 : 2
class = "data_bus";
2 13 19 : 2
class = "data_bus";
2 28 30 : 1
class = "data_bus";
3 28 30 : 3
```

```
class = "data_bus";
3 27 30 : 3
class = "data_bus";
3 27 33 : 1
class = "data_bus";
3 26 33 : 3
class = "data_bus";
3 26 30 : 1
class = "data_bus";
3 29 33 : 4
class = "data_bus";
3 16 18 : 2
class = "data_bus";
4 16 18 : 2
class = "data_bus";
4 15 21 : 1
class = "data_bus";
4 15 18 : 3
class = "data_bus";
4 14 21 : 3
class = "data_bus";
4 14 18 : 1
class = "data_bus";
4 17 21 : 4
class = "data_bus";
1 23 32 : 1
class = "data_bus";
```



```
2 25 32 : 1
class = "data_bus";
2 11 20 : 1
class = "data_bus";
2 13 20 : 1
class = "data_bus";
3 27 32 : 1
class = "data_bus";
3 29 32 : 1
class = "data_bus";
4 15 20 : 1
class = "data_bus";
4 17 20 : 1
class = "pcmc_lbx_cont";
7 21 33 : 8
class = "pcmc_lbx_cont";
7 19 31 : 5
class = "pent_pcmc_cont";
1 9 : 6
```

Appendix D

User Interface

D.1 The User Interface

The Study Generator will interface to the user through a file, in which the design variables and the constraints will be specified. The bold-faced names are the keywords.

Hierarchical description of the user file :

user-file :

- circuit
- variables
- signal
- constraint

circuit :

- **simulator** = [simulator name]

- **flags** = [simulation flags]
- **file** = [netlist file]
- **data** = [**i!** **e!**] [data file]
- **samples** = [number of samples]
- **query** = [query file]

variables :

- type [d! c! s!]
- variable_name
- **s!** (Statistical Variable)
 - type [**u!** **g!**]
 - range [lower bound] [upper bound]
- **d!** [model_name] [model_name]

signal :

- parameters
- noise

constraint :

- type [physical timing noise]
- variables

- inequality

signal :

- **rise_time** = [value]
- **fall_time** = [value]
- **swing** = [value]
- **sequence** = [value]

noise :

- **Vol** = [value]
- **Voh** = [value]
- **Vil** = [value]
- **Vih** = [value]
- **Vul** = [value]
- **Vuh** = [value]

constraint :

- physical :

- $a_{11}, a_{12}, \dots, a_{1n} : b_1$ [$<$ $>$]
- $a_{21}, a_{22}, \dots, a_{2n} : b_2$ [$<$ $>$]
- .
- .
- $a_{m1}, a_{m2}, \dots, a_{mn} : b_m$ [$<$ $>$]
- $\sum_{i=1}^n a_{ki} x_i$ [$<$ $>$] b_k (Each constraint line represents some linear combination of the physical design variables (x_i), and a constant (b_k) that is greater than, less than, or equal to it).

- timing :

- **delay** =
 - * node_num
 - * delay_level
 - * range
 - * **fit** =
 - variables
 - * **error** = [value]
- **delay_stb** =
 - * node_num
 - * delay_level
 - * stable_level
 - * range
 - * **fit** =
 - variables

- * **error** = [value]

- **rise_time** =

- **fall_time** =

- **noise** :

- **undsht** =

- * **node_num**

- * **noise_level**

- * **range**

- * **fit** =

- **variables**

- * **error** = [value]

- **ovsht** =

- **level** =
