Automatic Circuit Characterization through Computer Experiments

Sharad Mehrotra, Paul Franzon, Slobodan Simovich, Michael Steer

NCSU-VLSI 94-14[1]

January 1994

Electronics Research Laboratory

Department of Electrical and Computer Engineering

North Carolina State University

Raleigh NC 27695-7911

Abstract

In order to design high performance circuits, the relationship between circuit performance and design parameters must be precisely established. Previously, experimental design techniques have been employed for performance modeling of MOS VLSI devices and circuits. In this paper we describe a new computer-aided methodology for characterizing a family of electrical circuits. This methodology is based on multi-stage experimental design and prediction through data interpolation. The technique presented here is fully automated and hence helps the designer in efficiently characterizing any circuit response based on full circuit simulations. Through examples, we show the power of this technique in characterizing highly non-linear relationships between circuit performance and the design parameters, in a variety of applications.

[1]This work was supported by the National Science Foundation under grant MIP-901704 and by Cadence Design Systems

1 Introduction

There are a number of design applications where it is necessary to precisely establish the relationship between a circuit's electrical responses and certain *designable parameters*. The process of observing the behavior of a circuit block under different conditions and building a simplified model that closely mimics this behavior is referred to as *characterization* [2]. Several authors have addressed this issue, albeit with different end goals in mind. Milor et. al. [11] use such characterizations for computing parametric yield of analog ICs. Yu et.al. [18] characterize VLSI circuit performances in the presence of manufacturing fluctuations. Low et.al. [10] use characterizations for building macromodels of the IC fabrication process. Circuit characterizations are used for design optimization of VLSI devices [1] and circuits [3]. Our particular application has been that of using characterizations for formulating interconnect design rules [7].

The two conflicting goals in obtaining this characterization are accuracy and efficiency. For example, analytical modeling is a very efficient technique for response estimation, but is not satisfactory for high performance circuits. Circuit simulation using SPICE, for example, is accurate, but running multiple parametric simulations is quite expensive. It is clear that this trade-off can be best achieved if simple models are devised based on a minimal, but sufficient, set of accurate simulations. Such models make use of responses obtained at a few points in the *design space* to predict responses accurately over the entire design space. Traditionally, this has been done by using experimental design techniques for selecting points for simulation in the design space and then fitting polynomial models to the obtained responses through least square regression [1], [10], [15].

The rationale of these methods has been brought to question in [13], for cases where the responses are obtained from circuit simulation. Regression techniques introduce systematic bias in the models. Also, classical experimental design techniques [4] are biased by the assumed form of the model. They [13] propose methods for optimizing the experimental design and postulate novel prediction techniques, suitable for deterministic experiments. The main drawback of their method is the excessive time required to optimize the experimental design, though large increases in accuracy over classical techniques are observed.

Our approach of computer experimental design and response prediction draws from the ideas presented in [13], [16]. Specifically, we use Moving Least Square Interpolation [9] for modeling the responses. We use Latin Hypercube Sampling in a sequential experimental design for determining the points for simulation. A novel method of error characterization is used to determine the region in the design space in which to perform the next experiment.

The organization of the rest of this paper is as follows: In section 2. the experimental design problem is formulated. Sections 3 and 4 respectively give our methodology and some specifics of the implementation. Section 5 illustrates the use of this method for characterizing very fast data latches and high speed interconnect. Section 6 is devoted to a brief discussion of some open issues and the conclusions.

2 Problem Statement

Formally, the objective is as follows:

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

$$G(\zeta, z, t) = 0, \tag{1}$$

where ζ is a vector of instantaneous node voltages and currents, z is a set of *design parameters*, and t is time. The parameters specified by z 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 z 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 $\phi^*(z)$, which is relatively much cheaper to evaluate than a full circuit simulation, and is a good approximation of $\phi(z)$ over a range of z which is referred to as the *design space*. $\phi^*(z)$ is obtained by conducting a computer experiment in which $\phi(z)$ is evaluated at n sample sites $\{z_1, \ldots, z_n\}$ using the computer simulation. $\phi^*(z)$ must satisfy the following restrictions:

1. Predictable accuracy:

$$| \phi^*(z) - \phi(z) | \leq \epsilon, \tag{2}$$

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 $z^\star$, then ϕ^* should have the same value at $z^\star$, i.e., $\phi(z^\star) = \phi^*(z^\star) \forall z^\star \in \{z_1, \ldots, z_n\}$.

Hence the objective of the experimental design is to choose a suitable predictor function $\phi^*(z)$ and n sample sites $\{z_1, \ldots, z_n\}$ 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 [12], Moving Least Square Interpolant [9] etc. that easily achieve this condition. The unbiasedness condition helps us formulate the cross-validation error-measure [17]. 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.

In the next section, we first describe how the sample sites are selected using sequential experimental design and then discuss the predictor function used.

3 Sequential Experimental Design

The main goal of *experimental design* is to choose sites in the design space to be characterized such that the error made at all untried inputs by the predictor function is minimized. This is quite a formidable task, especially if no prior information is available about the nature of the responses, as is the case in the applications described here. 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. Our approach is to keep the same sampling strategy during each step of the experimentation. Only the extent of the design variables z, (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 region, is required. Latin Hypercube Sampling (LHS) is very suitable for this purpose.

3.1 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.

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.,

$$\begin{aligned} &\forall z_i, && i = 1 \ldots n, \\ &Compute && \mid \phi^*(z_i) - \phi(z_i) \mid, \end{aligned} \tag{3}$$

where $\phi^*(z_i)$ is computed based on $\phi(z_j)$'s, $j = 1 \ldots n$, $j \neq i$.

This error measure is termed *cross-validation* [17]. 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.

3.2 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. In this section we give a brief justification of using data interpolation for prediction and describe

its exact form. First, we summarize the approach of [13][16] towards the same problem.

In [13] a stochastic response model $\hat{\phi}(z)$ is postulated for designing the experiment and a predictor function $\phi^*(z)$ is formulated based on the experimental results. The response model is chosen to be

$$\hat{\phi}(z) = \Sigma_{j=1}^{n} a_j b_j(z) + X(z), \tag{4}$$

where a_j are scalars, $b_j(z)$ are polynomial terms and $X(z)$ is a stochastic model of the departure of the true response from the polynomial. with zero mean and covariance $V(y,z)$ between any pair of processes $X(y)$ and $X(z)$. The covariance is given as

$$V(y,z) = \sigma^2 R(y,z), \tag{5}$$

where σ^2 is the variance and $R(y,z)$ is the correlation. $X(z)$ represents the departure of the response from the polynomial model given by the first term of equation 6. Suppose that the response is known at a certain set of *sample points*. In [13], the predictor function, $\phi^*(z)$, is the expected value of the stochastic process, $\hat{\phi}(z)$, used to model the response of the simulator:

$$\hat{\phi}(z) = \Sigma_{j=1}^{n} a_j b_j(z) + X(z), \tag{6}$$

$\phi^*(z)$ is the sum of a generalized least squares estimate of the first term in Equation 6, using the sampled responses, and a smoothing term, expressed as an interpolant of the residuals at the sampled points. This smoothing term can also be seen as the posterior mean of the random process $X(z)$.

In [16] stochastic functions are used to model objective functions for the purpose of finding a global optimum of an unknown smooth function. When the stochastic function has a multidimensional normal distribution (as is the response model in [13], with a zero order polynomial), the value of the objective function at any point in the input space is a Gaussian random variable. Under certain simple axioms, the posterior mean of the random variables can be approximated by interpolation of the sampled (or known) objective function values.

In the spirit of the above discussion, we chose Moving Least Square Interpolation for prediction[9]. Following is a brief description of this method:

The response model is given by

$$\phi^*(z) = \Sigma_{j=1}^{n} a_j b_j(z), \tag{7}$$

where $b_1(z), \ldots, b_n(z)$ are n linearly independent polynomials in z. These functions are supplied by the user. 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:

$$BW(z)B^T a(z) = BW(z)\phi \tag{8}$$

where B is an $n \times N$ matrix whose jth row is $[b_j(z_1), \ldots, b_j(z_n)]$, ϕ is the $N \times 1$ vector of responses at the sample points, and $W(z)$ is a diagonal weighting matrix, with elements $w_{ii}(z)$ being the weights assigned to the error at z_i's. In order to achieve exact interpolation at the sampled points, the function w should go to infinity at the sampled points z_i's. Functions of the form

$$w_i(z) = e^{-\alpha||z-z_i||^2}/(||z - z_i||^2) \tag{9}$$

have this behavior. These functions also attenuate rapidly and hence minimize the influence of remote data values (i.e. $\phi^*(z)$ is *local* in nature), while smoothing the response.

Prediction by MLSI has several advantages. It is cheaper to evaluate than the BLUP in [13], since it involves an $n \times n$ matrix inversion, instead of the $N \times N$ covariance matrix for BLUP computation. In our experience, there is no appreciable difference in accuracy between the two. Also, there is no need to formulate a correlation structure, although there are some alternatives in choosing the weighting function. In [17], 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. Our approach, however, *is to generate more samples in regions where the predictor function has poor fit to the data.* The *local* nature of the predictor allows us to use *cross-validation* for error estimation. The importance of this fact cannot be overemphasized. In order to characterize error locally, the predictor function must also have local behavior.

4 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 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 [8] for automatic specification of the simulations and extraction of electrical responses[14].

4.1 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 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. 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.

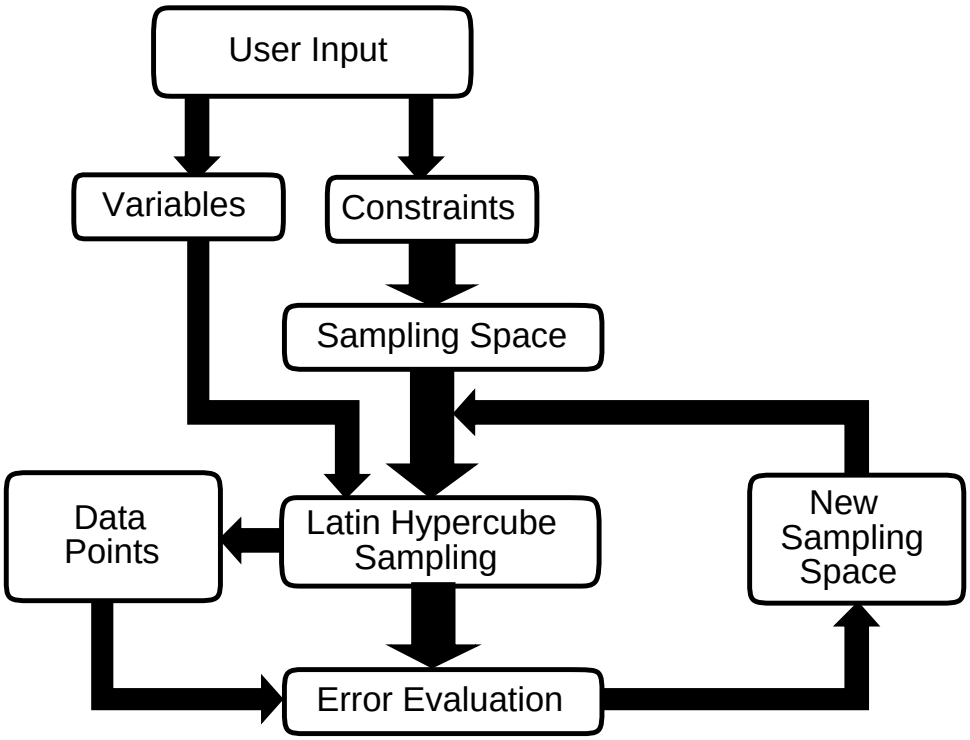


Figure 1: Study Generator

4.2 Identifying the next Experimental Region

In section 3.1, *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 3. If this error is greater than a certain threshold, it implies that the neighboring points of z_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 z_i. The neighborhood of z_i is defined as a ball of radius which is half the minimum scaled distance between z_i and all the other design points, i.e.,

$$r(i) = 0.5 \star \min_{j=1,\ldots,N, j \neq i} ||z_i - z_j||_2. \tag{10}$$

The half minimum distance criterion is used to eliminate overlap between the neighborhoods of adjacent points. Each component of z_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. 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.

5 IV. Examples

The feasibility of this approach is tested by using the Study Generator to characterize two very different circuits. The first application is to determine clock circuit timing design rules consistent with a high speed latch design used in the DEC Alpha[5] microprocessor. The second is to determine wiring rules for a high speed net on a Multi-Chip Module. The common denominator here is the highly non-linear nature of the relationship between the designable circuit parameters and the circuit responses of interest.

5.1 Clock Timing Design for Correct Latch Operation

In this example, we characterize a latch structure similar to one used in the DEC Alpha chip[5] (figure 2). 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, we studied the effect of clock rise time, data rise time, and clock skew on race-through in this latch. Race-through is detected by studying the apparent delay of a signal passing through two cascaded latches. 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.

The following inequalities describe the design space to be characterized:

$$0.1ns \leq \text{clock rise time} \leq 1.5ns$$
$$0.1ns \leq \text{data rise time} \leq 1.5ns$$
$$0.1ns \leq \text{clock skew} \leq 1.1ns$$

In the experiment design, two sampling stages were used, with 75 points taken in the first stage and 50 in the second. A first order polynomial in all three variables was chosen for $b_1(z)$ in the interpolation. Another separate characterization was carried out, using MetaSim[8] with a total of 384 points placed on a regular grid in the design space. The predictor function described earlier was used to estimate the response at the same grid points, based on the observations from the experiment. Figure 3 shows the plot of signal delay as a function of data and clock rise times, for data rise time of 0.3ns and clock period of 5ns with a 50% duty cycle. Figure 4 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.

The error statistics, comparing the predicted to actual response are shown in Table 1. Error 1 is the error in estimating the responses at the 384 grid 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 validational error on the first 75 sample points, and Cross error 2, is

the error reported on all the points after resampling.

The error statistics show that the average error in prediction reduces significantly after resampling. The cross validational error does not, however, improve with resampling. The reasons for, and implications of, this are discussed in the next section.

This example illustrates how our sequential sampling strategy can be employed in a real circuit design situation. Using the response function, clock design rules can be generated to ensure race-free operation quickly and accurately. This process is discussed elsewhere [7].

5.2 High Speed Interconnect Design for Signal Integrity

In this characterization study, the relationship between interconnect length and signal settling time in a high speed net was studied. Figure 5 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 6. 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 [6]. Hence the settling time has a highly non-linear relationship to the interconnect length.

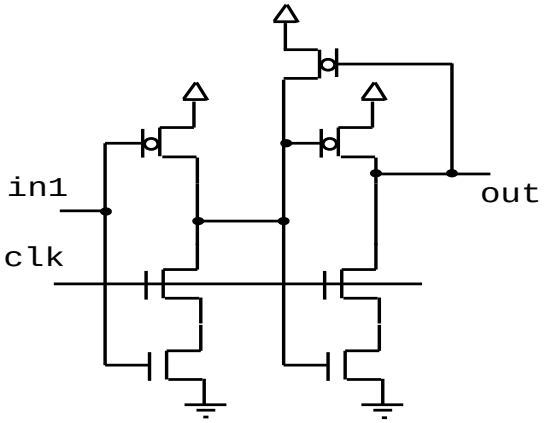
The following inequalities describe the design space to be characterized:

$$1mm \leq branch1 \leq 10cm$$
$$1mm \leq branch2 \leq 10cm$$
$$1mm \leq stub \leq 10cm$$

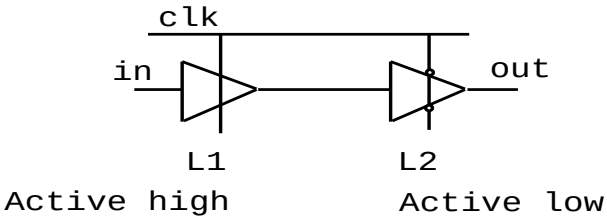
First, a large characterization using 1,000 sample points over a full grid in the design space was carried out, for benchmarking the results obtained from experimental characterizations. A set of several different experimental characterizations of this same net were performed using the Study Generator.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

	maximum error	mean error	error variance
Cross error 1	3.9ns	.69ns	-
Cross error 2	3.95ns	.74ns	-
Error 1	2.4ns	.55ns	.68ns
Error 2	3.1ns	.36ns	.70ns

Table 1: Error Statistics for the Latch Characterization



Active High Latch



Test Circuit

Figure 2: Schematic of Latch Circuit

Figure 3: Signal Delay plot for data rise time of 0.3ns: True response

Figure 4: Signal Delay plot for data rise time of 0.3ns: Predicted response

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 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 8 shows the same response as in Figure 7 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 generated 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 2.

The error statistics bring out some important points about our methodology. Firstly, the way the total sampling capability (i.e. the total number of simulations allowed) is partitioned between the first stage and the next does, to a certain extent, affect the accuracy of the characterization. It appears that having more points in the first stage guarantees a better coverage of the sample space, and hence makes it possible to better locate the irregularities in the response. Also, use of a higher order polynomial for interpolation improves the predictive error. However, this also makes the interpolation slightly more expensive.

6 Conclusions and Discussion

In this paper, a highly automated methodology for characterizing electrical circuits is presented. The examples demonstrate the power of this technique in capturing highly irregular relationships with good accuracy over a large design space, in reasonable time. However, there are some aspects, as evident from the examples, that need further discussion.

Somehow the initial number of samples, the number of sampling stages and the number of samples in each stage need to be decided. We have left this to the discretion of the user, especially since the form of the response is not known a priori. The number of samples needed for an accurate characterization is related to the nature of the response only, not to the complexity of the circuit models. An experiment which is perceived to have a fairly linear response should be conducted with a small number of simulations, from where the

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

Table 2: Error Statistics for the MCM interconnect characterization

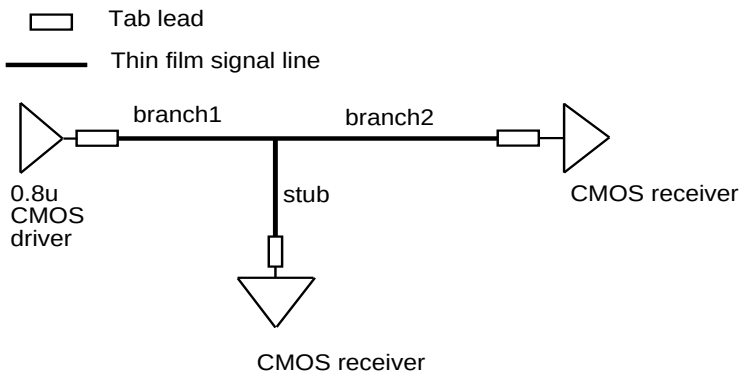


Figure 5: Net Topology

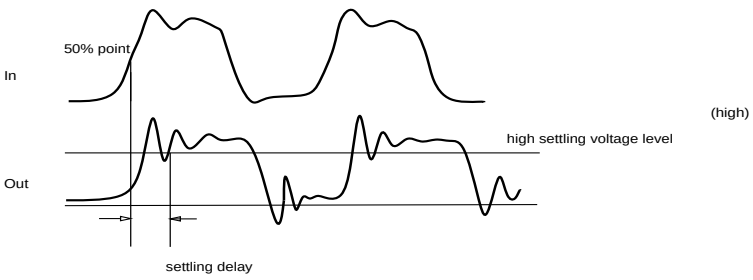


Figure 6: Signal Settling time

Figure 7: Partial characterization of MCM interconnect. Branch 1 = 3cm

Figure 8: Sampled characterization MCM interconnect. Branch 1 = 3cm

software can take over to determine the number of further simulations required. However, for highly irregular responses, a larger number of samples is required in the first stage.

Smoothness of the response is far more important in determining the total number of samples needed than the dimensionality of the input space. In general, the number of samples required grows only linearly with dimension of the design space. Using a little engineering judgement in the first sampling pass can be quite helpful in reducing overall simulation time. Also, since all generated data is reusable for a similar design, oversimulation is not necessarily a big drawback.

In a fully automated experiment, some appropriate criterion must be devised for stopping the iterations in the sequential experimentation. Ideally, a stop criterion would be one based on the error measure described in section 3.2. 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 error characterization at the points obtained in the first sampling stage only. We have kept our implementation deliberately flexible to allow for the formulation of efficient heuristics for managing simulation time.

7 Acknowledgements

The authors would like to thank George Katopis, J.C. Lu and Zaki Rakib for discussions during our work. The CMOS driver models were provided by Wayne Detloff of MCNC.

References

[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 CAD*, 7:pp. 272–288, Feb. 1988.

[2] J. Benkoski and A. J. Strojwas. A new approach to hierarchical and statistical timing simulations. *IEEE Transactions on CAD*, 6:pp. 1039–1052, Nov 1987.

[3] M. C. Bernardo, R. Buck, L. Liu, W. A. Nazaret, J. Sacks, and W. J. Welch. Integrated circuit design optimization using a sequential strategy. *IEEE Transactions on CAD*, 11:pp. 353–360, March 1992.

[4] G. E. P. Box and N. R. Draper. *Emprical Model Building and Response Surfaces.* Wiley, New York, 1987.

[5] 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.

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

[7] P. D. Franzon, S. Simovich, S. Mehrotra, and Michael Steer. Macromodels for generating signal integrity and timing management advice for package design. In *Proc. of IEEE Electronic Comp. and Tech. Conf.*, pages 523–529, 1993.

[8] 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 *29th Proc. Design Automation Conference*, 1992.

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

[10] K. K. Low and S. W. Director. An efficient methodology for building macromodels of IC fabrication process. *IEEE Transactions on CAD*, 8:pp. 1299–1313, Dec. 1989.

[11] L. Milor and A. Sangiovanni-Vincentelli. Computing parametric yield accurately and efficiently. In *Proceedings of ICCAD*, pages pp. 116–119, 1990.

[12] J. Sacks, S. B. Schiller, and W. J. Welch. Designs for computer experiments. *Technometrics*, 31:41–47, 1989.

[13] 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.

[14] 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.

[15] A. J. Strojwas and S. W. Director. An efficient algorithm for parametric fault simulation of monolithic IC's. *IEEE Transactions on CAD*, 8:pp. 1049–1058, Aug. 1991.

[16] Aimo Torn and Antanas Zilinskas. *Global Optimization*. Springer-Verlag, 1989.

[17] 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.

[18] T. K. Yu, S. M. Kang, I. N. Hajj, and T. N. Trick. Statistical modeling of VLSI circuit performances. In *Proceedings of ICCAD*, pages pp. 224–227, 1986.