Statistical failure analysis of system timing

by D. R. Tryon
F. M. Armstrong
M. R. Reiter

Techniques are developed that quantify the probability that large computer systems will meet their cycle time objectives. Both approximation techniques and rigorous multivariate statistical techniques are described. A method is developed that enumerates the cycle-limiting paths so that these approaches can be utilized. The results of these techniques enable system designers to ensure that performance and reliability objectives are met.

Introduction

One of the most crucial elements in customer acceptance of large-scale computer systems is processor performance. The essential contributor to this performance is clearly the latch-to-latch cycle time of the processor. Verification of this cycle-time objective prior to hardware implementation is essential in order to ensure that availability and quality requirements are satisfied. Since 1973, various programs and techniques [1–4] have been developed to trace paths and consequently ensure that path delays between storage elements meet cycle-time objectives. Since it is virtually impossible for the designers of large-scale computer systems to predict all paths which are potentially cycle-limiting prior to building hardware, delay calculator programs which rely upon the designer to specify the path characteristics are of limited value. These limitations were the motivation for the development of McWilliams' SCALD timing verifier [5] and Hitchcock's Timing Analysis (TA) [6, 7]. McWilliams' timing verifier utilizes a simulation approach to perform a pessimistic delay analysis of the logic. A pessimistic analyzer can, however, cause the designer to implement unneeded changes. Hitchcock's Timing Analysis is a block-oriented algorithm that provides slack information at each circuit block as a measure of the problem severity. It can utilize both nominal and standard deviation delay for all components of the design. A statistical rather than a worst-case analysis can therefore be accomplished. Using Hitchcock's Timing Analysis, statistical timing verification for all paths in a large-scale computer system can be performed to a user-specified confidence level.

Verification of each system path at a specified confidence level, however, provides no insight into the probability of the system as a whole working. As the designers of large systems continue to stress the performance limitations of their VLSI technologies, the number of potentially cycle-limiting paths can grow into the thousands. The probability of all paths in the system meeting their cycle-time objectives is therefore significantly less than the probability of a single path working. A designer timing all his path delays to a three-sigma confidence level, for example, cannot assume that he has anything close to a three-sigma system design.

This paper describes techniques to quantify system timing failure risks. The paper first reviews the statistical slack computation utilized by Hitchcock's Timing Analysis to ensure that all paths are timed to a specified confidence level. Results using the binomial distribution are then described to approximate the probability of system timing failure as originally described by Shelly and Tryon [8]. Since this approximation technique ignores correlation between paths caused by variations in temperature, power supply

output, and semiconductor manufacturing processes, a statistical technique is developed to address path-to-path correlation. This technique is not feasible, however, for systems with more than five hundred or so cycle-limiting paths. Another approximation approach is then developed that closely models the more rigorous correlation approach and yet permits the evaluation of system failure for any size system.

In order to use these techniques, a determination of the number of paths in the system that are potentially cycle-limiting is required. The Timing Analysis program provides block-count information but not path enumeration. An approach to process data from both Timing Analysis and a Boolean Equivalence analyzer [9] is presented. This approach permits the approximation of the number of cycle-limiting system paths to sufficient accuracy. These data and the techniques described in this paper will enable the large-scale system designer to predict the probability of system timing failure prior to hardware implementation of the design. These results will therefore guide the system designer in properly selecting confidence levels for his path timing verification and should therefore ensure the likelihood of large-scale VLSI computer hardware performing at desired cycle times.

Slack computation

Hitchcock's Timing Analysis program permits the identification of all paths in the logic that are potentially cycle-limiting by using an efficient, block-oriented approach described in [7]. A statistically worst-case arrival time is reported at the input to each storage element in the design. A "slack" value is reported at each circuit in the design which represents the difference between the actual arrival time and the required arrival time to meet cycle time objectives. Slack is computed by the formula

$$SLACK = NC - ND - (beta \times sigma),$$

where NC = nominal arrival time of latching clock signal, ND = nominal arrival time of data, *beta* = confidence level, and *sigma* = standard deviation computed by statistically combining circuit tolerances along the clock and data paths.

A positive slack ensures that the data will arrive before the latching clock, as shown in **Figure 1**. Negative slack is undesirable because there is a significant probability that the data will arrive after the latch clock and consequently will not be latched in the correct cycle. The effective *sigma* that is computed for the slack computation represents a statistical difference of the clock path overall delay tolerance with the data path delay tolerance. A difference calculation rather than a statistical summation is employed, since temperature, power supply, and semiconductor process conditions affect both the data path circuit delays and clock path delays.

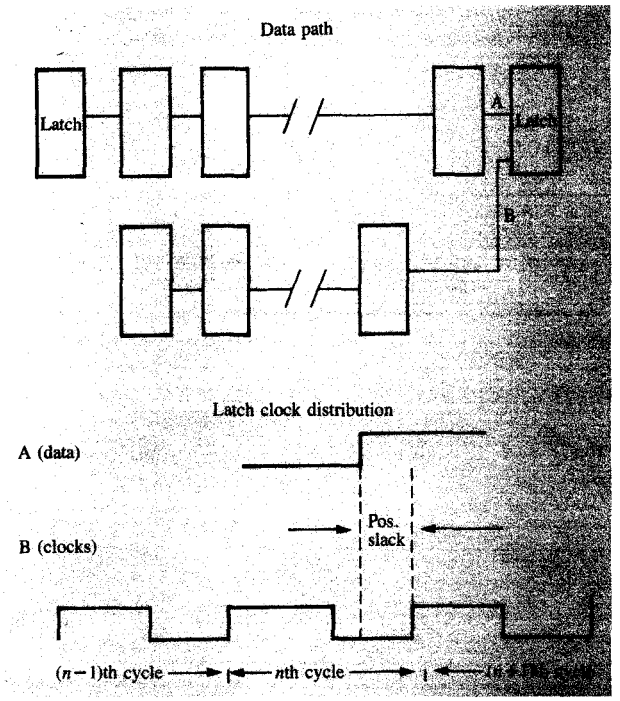

Figure 1

Timing slack generation.

Thus, a data path signal that is slow due to these effects may still get latched if the clock is also slow due to the same effects.

The designer using Timing Analysis can select the *beta* multiplier for the standard deviation. If, for example, the designer wanted to verify all his paths to a three-sigma confidence level, he would select *beta* = 3. The remainder of this paper develops techniques to provide the designer with an overall system confidence level once he has selected this path confidence level.

Methodology

The system failure problem is addressed by initially making a series of simplifying assumptions. After determining solutions based upon these assumptions, the assumptions are removed one by one and models presented. An objective is to demonstrate that the simplified models of the problem are adequate predictors of system timing failure even when the simplifying assumptions are removed.

The system timing problem is solved by initially making two key assumptions:

- All paths are independent.
- The number of cycle-limiting paths is known.

Table 1 Timing problems vs number of paths.

$n = 1000$		$n = 5000$		$n = 10\,000$	
x'	$P(x \leq x')$	x'	$P(x \leq x')$	x'	$P(x \leq x')$
0	0.259	0	0.001	0	0.0000013
1	0.609	5	0.334	5	0.008
2	0.846	10	0.918	10	0.211
3	0.952	15	0.998	15	0.718
4	0.988			20	0.965
5	0.997				

Table 2 Timing problems vs path *beta* ($n = 1000$ paths). $P(x \leq x') = 0.999$.

Beta	x'
4.0	1
3.9	2
3.8	2
3.7	2
3.6	2
3.5	3
3.4	3
3.3	4
3.2	4
3.1	5
3.0	6
2.5	15
2.0	39

These assumptions lead to the application of the binomial theorem to solve the problem. The binomial theorem can be used to predict 1) the probability of all paths meeting cycle objectives, and 2) the probability that there will be no more than a specified number of failures.

The first assumption, that of independence, is then attacked. A model is required that allows specification of path-to-path correlation. The multivariate normal density function is chosen as the model. Since numerical techniques are impractical because of the large number of system paths, a Monte Carlo simulation approach is described. Results from simulations are given under the following assumptions:

- The number of cycle-limiting paths is known,
- This number is less than 500, and
- All off-diagonal correlations are assumed equal.

A conclusion is reached that leads us to believe that we can use the binomial to predict the number of timing failures even if correlation is present. In order to ensure that this conclusion is not a function of our last assumption of equal off-diagonal correlation coefficients, this assumption is attacked. Simulations were performed and are described here that do not rely on this assumption of correlation symmetry. The conclusion is confirmed based upon these additional simulations.

The authors then attack the second assumption of five hundred system paths as the practical maximum due to simulation time constraints. The authors noted a phenomenon known as "dishing" in the behavior of the simulations and hypothesize a modified binomial model to predict system failure. A variety of simulations were run to confirm the validity of this model. This model will predict system failure for any number of cycle-limiting paths.

We are then left with the final assumption that the number of cycle-limiting paths is known. This assumption is then addressed by interpreting data from both Timing Analysis and a Boolean equivalence program.

The binomial approach

An initial approximation of system timing failure probability that features simplicity of calculation is to use the binomial distribution

$$P(y) = \binom{n}{y} P^y (1 - P)^{n-y},$$

where n = total number of cycle-limiting paths, y = number of paths meeting cycle objectives, P = probability of a single path meeting its cycle objective, and $P(y)$ = probability of y paths meeting cycle objectives.

For example, let us assume that a system designer has estimated that he has a thousand independent, potentially cycle-limiting paths in his design and has used Timing Analysis to verify each to a three-sigma confidence level. The designer could simply substitute 0.99865 for P, 1000 for n, and 1000 for y to predict the probability of his entire system meeting its performance objective. (0.99865 is the value of the normal cumulative density function at $z = 3.0$.) Thus,

$$P(1000) = (0.99865)^{1000} = 0.259.$$

The system designer may, however, be willing to tolerate a few path failures on his test floor. In this case, he is more interested in the probability of having a specified number of failures than the probability of all paths working. The binomial distribution can be used for this problem by summing the appropriate terms for different values of y. **Table 1** summarizes the results from applying the binomial. A designer who has estimated that he has 1000 cycle-limiting paths in his design can be 99.7 percent sure that he will have no more than five timing problems. The designer may consider this an acceptable risk if he has sufficient hardware debugging time to identify and fix these problems.

An upper limit to the number of timing failures is also of considerable interest. **Table 2** illustrates the effect of changing the path confidence level on the number of expected system timing failures if the designer estimated that

he had 1000 cycle-limiting paths in his design. A 99.9 percent assurance level was chosen for this table. In other words, there is a 99.9 percent probability that there will be six or fewer timing failures in the design if each has been verified to three-sigma confidence. This is derived by summing terms of the binomial until 0.999 is reached. He could have the same 99.9 percent assurance that there will be no more than one timing failure if he were to verify each path at four-sigma level. Using the higher path sigma level does not come free, however. The designer might introduce many design changes that would not have been necessary to meet machine cycle time if he specified a sigma level that was too high. By studying data such as those in Table 2, the designer can make the necessary trade-offs in deciding whether the higher path sigma confidence level is justified. These trade-offs include schedule and hardware costs to make the design change versus the time and cost of finding and fixing any problems in hardware.

The primary advantage of the binomial approach is the simplicity of computation. This approach has a significant disadvantage, however. The binomial distribution can only be exercised if all paths are assumed to be independent of one another. Although this may be a reasonable assumption in some cases, it is certainly not always the case. Power supply output, temperature, semiconductor manufacturing processes, and logical commonalities are factors which create correlation between path delays. An approach is therefore required that will compute system failure probability with path-to-path correlation present.

The correlation solution

• *A mathematical model*

By invoking the central limit theorem, we can claim that each path slack

$$x_i \sim N(u_{x_i}, \sigma^2_{x_i}),$$

where $u_{x_i} = NC - ND$ as defined previously, and σ_{x_i} = effective sigma from slack computation.

The probability that any path has positive slack is given by

$$P(x_i > 0) = \int_0^\infty \phi(x_i; u_{x_i}, \sigma^2_{x_i}) dx_i.$$

However, the probability that all paths in a system are operating successfully is not the product of the individual probabilities,

$$P(x_1 x_2 \cdots x_k > 0) \neq P(x_1 > 0) P(x_2 > 0) \cdots P(x_k > 0),$$

due to the path-to-path correlation.

While this lack of independence between paths implies that the probability of system success cannot be determined as a simple product, the multivariate normal density

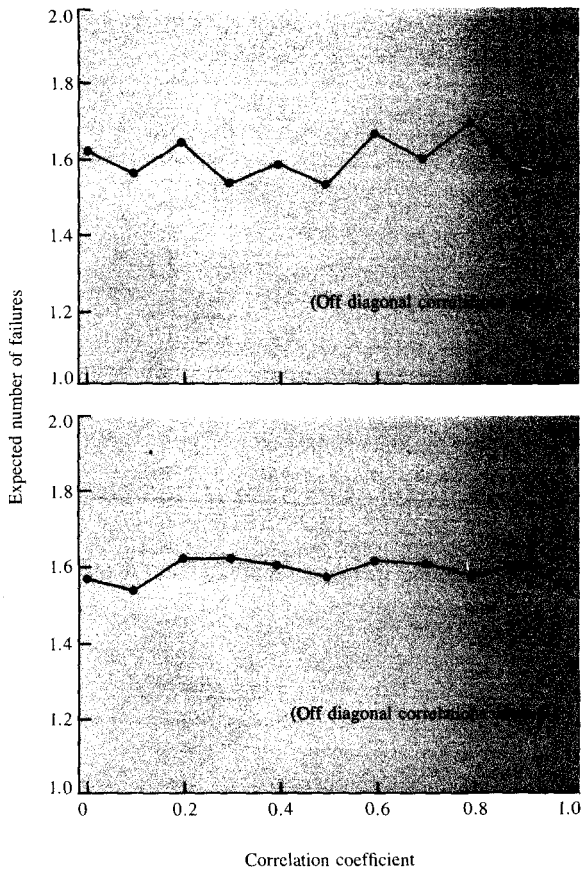

Figure 2

Expected number of failures versus correlation coefficient. Number of paths = 10; number of simulations = 2000; mean = 1; and sigma = 1.

function can be used. This function can be written in the form

$$\phi(\underline{x}) = \frac{1}{(2\pi)^{k/2} |\Sigma|^{1/2}} \exp\left[-1/2(\underline{x} - \underline{u})' \Sigma^{-1} (\underline{x} - \underline{u})\right],$$

where

$$\underline{x} = [x_1 x_2 \cdots x_k],$$

$$\underline{u} = [u_1 u_2 \cdots u_k],$$

$$\Sigma = \begin{bmatrix} \sigma_k^2 & \rho_{k,k-1}\sigma_k\sigma_{k-1} & \cdots & \rho_{k,1}\sigma_k\sigma_1 \\ & \sigma_{k-1}^2 & & \\ & & \ddots & \\ \rho_{1,k}\sigma_1\sigma_k & & \cdots & \sigma_1^2 \end{bmatrix}.$$

This function considers the correlation structure and thus will provide an adequate statistical model. Naively stated,

Table 3 25-path example. The number of simulations was 500, the mean 2.00, the sigma 1.00, and the correlation coefficient 0.40. The expected number of bad paths is 0.6420.

No. failing paths	*Freq.*	*Rel. freq.*	*Cum. freq.*	*Rel. cum. freq.*
0	361	0.722	361	0.722
1	70	0.140	431	0.862
2	33	0.066	464	0.928
3	15	0.030	479	0.958
4	3	0.006	482	0.964
5	6	0.012	488	0.976
6	3	0.006	491	0.982
7	1	0.002	492	0.984
8	4	0.008	496	0.992
9	1	0.002	497	0.994
10	2	0.004	499	0.998
11	0	0.000	499	0.998
12	1	0.002	500	1.000

the answer to our problem lies in the solution of

$$P(x_1x_2 \cdots x_k > 0) = \int_{x_1=0}^{\infty}\int_{x_2=0}^{\infty}\cdots\int_{x_k=0}^{\infty}\phi(\underline{x})dx_1dx_2\cdots dx_k.$$

However, the above equation cannot be solved in closed form. Numerical techniques such as Gaussian integration are impractical due to the fact that there can be thousands of cycle-limiting paths. A Monte Carlo simulation approach was therefore tried. This approach takes advantage of certain properties of conditional forms of the distribution by partitioning as follows:

$$\underline{x} = \begin{bmatrix} x_j \\ \hline x_{j-1} \\ \vdots \\ x_2 \\ x_1 \end{bmatrix} \begin{matrix} \underline{x}_1 \\ \hline \\ \\ \\ \\ \underline{x}_2 \end{matrix}, \quad \underline{u} = \begin{bmatrix} u_j \\ \hline u_{j-1} \\ \vdots \\ u_2 \\ u_1 \end{bmatrix} \begin{matrix} \underline{u}_1 \\ \hline \\ \\ \\ \\ \underline{u}_2 \end{matrix},$$

$$\Sigma = \left[\begin{array}{c|c} \Sigma_{11} & \Sigma_{12} \\ \hline \Sigma_{21} & \Sigma_{22} \end{array}\right].$$

Note that this partition is done in a manner such that $\underline{x}_1$ contains the single random variable x_j, $\underline{u}_1$ contains the single mean u_j, and

$$\Sigma_{11} = \sigma_j^2.$$

Given this partition, it can be shown that [10]

$$E(\underline{x}_1|\underline{x}_2) = \underline{u}_1 + \Sigma_{12}\Sigma_{22}^{-1}(\underline{x}_2 - \underline{u}_2),$$

$$V(\underline{x}_1|\underline{x}_2) = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{12}'.$$

The manner in which the above derivations are utilized to solve the system failure problem via simulation can best be described through the following three-path example.

1. Consider the three random variables

   $x_1 \sim N(u_1, \sigma_1^2)$,

   $x_2 \sim N(u_2, \sigma_2^2)$,

   $x_3 \sim N(u_3, \sigma_3^2)$,

   with variance-covariance structure

   $$\Sigma = \begin{bmatrix} \sigma_3^2 & \rho_{32}\sigma_3\sigma_2 & \rho_{31}\sigma_3\sigma_1 \\ \rho_{23}\sigma_2\sigma_3 & \sigma_2^2 & \rho_{21}\sigma_2\sigma_1 \\ \rho_{13}\sigma_1\sigma_3 & \rho_{12}\sigma_1\sigma_2 & \sigma_1^2 \end{bmatrix}.$$

2. Using standard techniques to generate normally distributed random numbers, pick a value for x_1 from the distribution $N(u_1, \sigma_1^2)$.
3. Then, define

   $$\underline{x} = \begin{bmatrix} x_2 \\ \hline x_1 \end{bmatrix} \begin{matrix} \underline{x}_1 \\ \hline \underline{x}_2 \end{matrix}, \quad \underline{u} = \begin{bmatrix} u_2 \\ \hline u_1 \end{bmatrix} \begin{matrix} \underline{u}_1 \\ \hline \underline{u}_2 \end{matrix},$$

   $\Sigma_{11} = \sigma_2^2$, $\Sigma_{22} = \sigma_1^2$, $\Sigma_{21} = \rho_{12}\sigma_1\sigma_2$, $\Sigma_{12} = \rho_{21}\sigma_2\sigma_1$,

   and we have

   $x_2 \sim N[E(x_2|x_1); V(x_2|x_1)]$,

   where

   $E(x_2|x_1) = \underline{u}_1 + \Sigma_{12}\Sigma_{22}^{-1}(\underline{x}_2 - \underline{u}_2)$

   and

   $V(x_2|x_1) = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{12}'$.

   Using the same random number generating techniques referred to above, select a value for x_2.
4. Now redefine

   $$\underline{x} = \begin{bmatrix} x_3 \\ \hline x_2 \\ x_1 \end{bmatrix} \begin{matrix} \underline{x}_1 \\ \hline \\ \underline{x}_2 \end{matrix}, \quad \underline{u} = \begin{bmatrix} u_3 \\ \hline u_2 \\ u_1 \end{bmatrix} \begin{matrix} \underline{u}_1 \\ \hline \\ \underline{u}_2 \end{matrix},$$

   $$\Sigma_{11} = \sigma_3^2, \quad \Sigma_{22} = \begin{bmatrix} \sigma_2^2 & \rho_{21}\sigma_2\sigma_1 \\ \rho_{12}\sigma_1\sigma_2 & \sigma_1^2 \end{bmatrix},$$

   $$\Sigma_{21} = \begin{bmatrix} \rho_{23}\sigma_2\sigma_3 \\ \rho_{13}\sigma_1\sigma_3 \end{bmatrix}, \quad \Sigma_{12} = [\rho_{32}\sigma_3\sigma_2 \quad \rho_{31}\sigma_3\sigma_1],$$

   and we have

   $x_3 \sim N[E(x_3|x_1; x_2); V(x_3|x_1; x_2)]$,

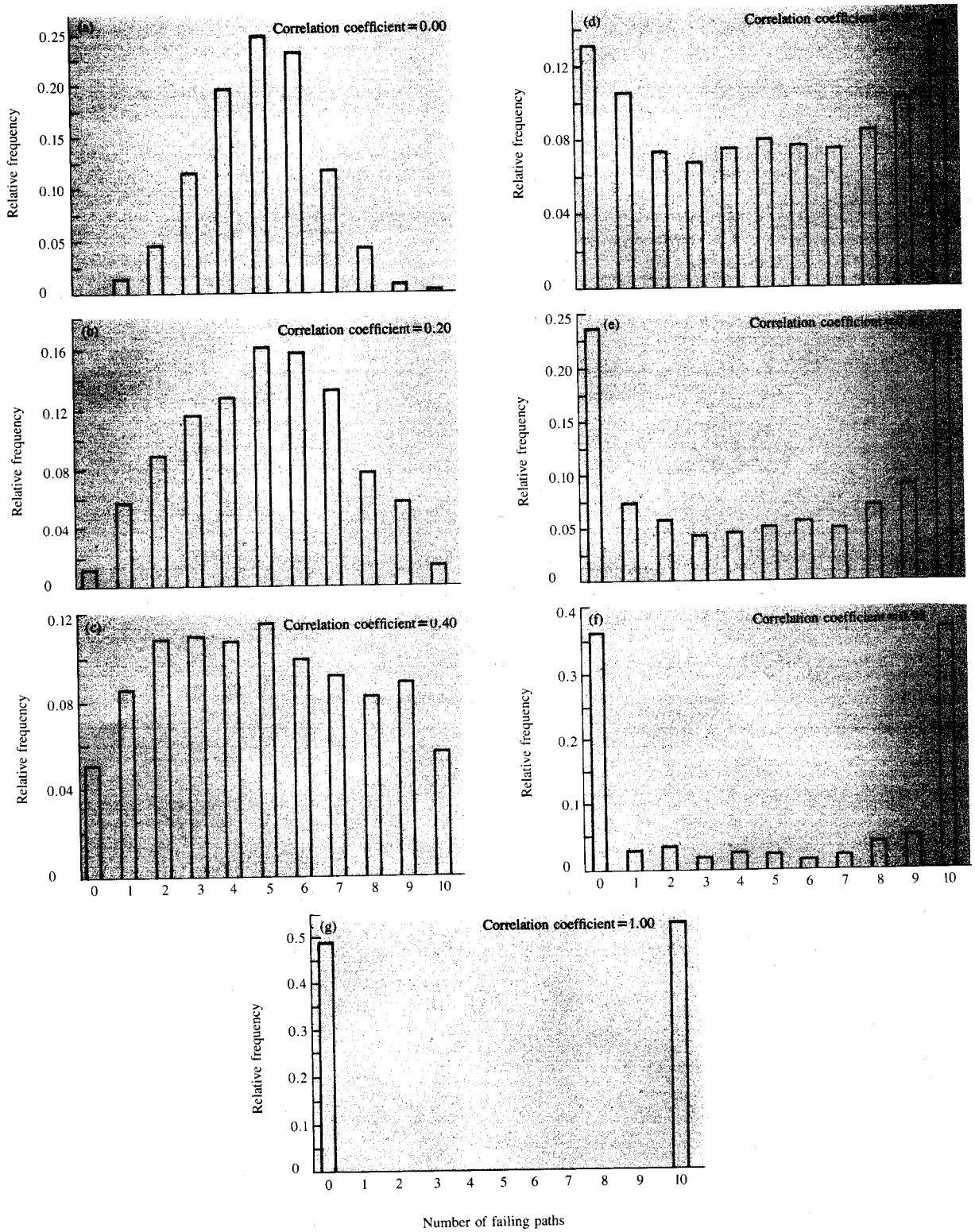

Figure 3

Dishing effect with $n = 10$ and $P = 0.50$.

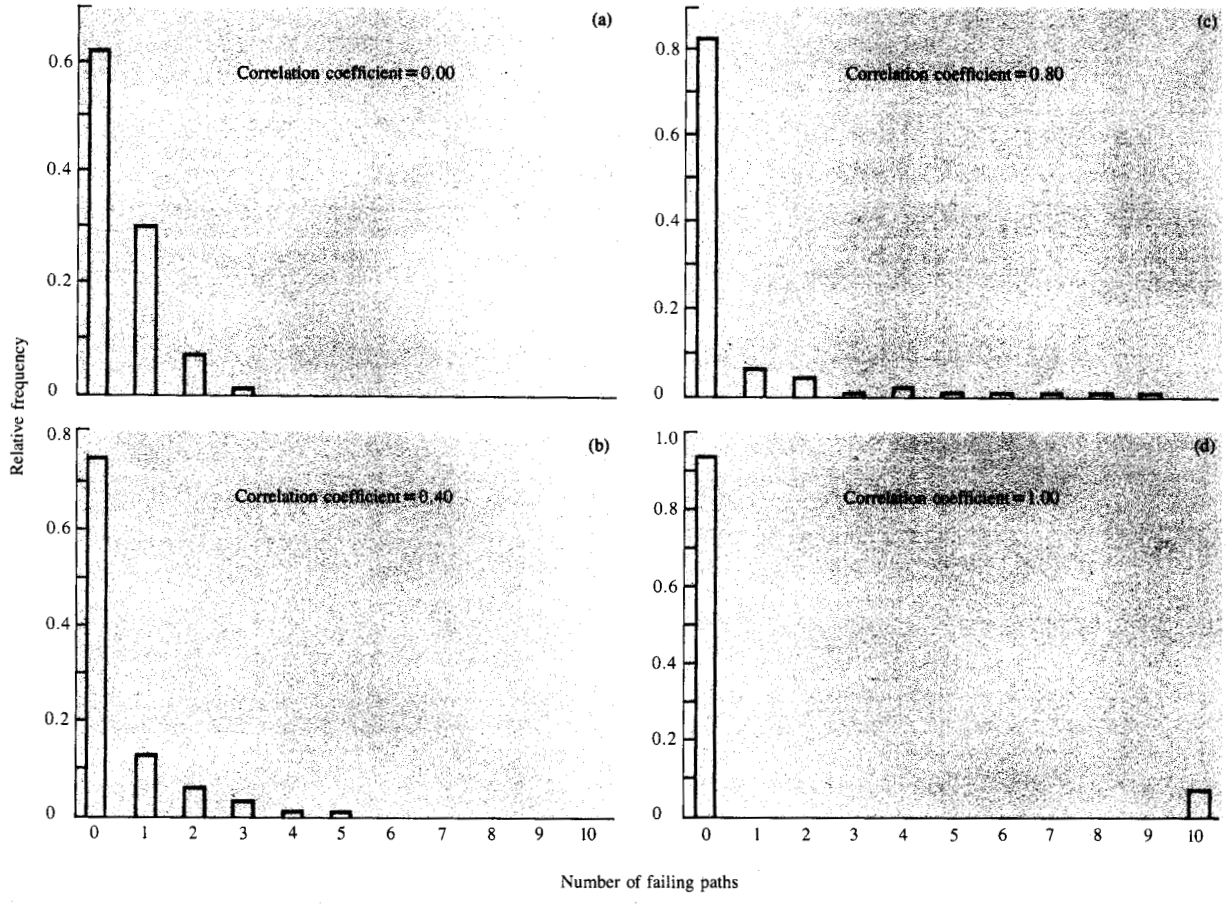

Figure 4

Dishing effect with $n = 10$ and $P = 0.95$.

where

$$E(x_3 | x_1; x_2) = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2)$$

$$= u_3 + [\rho_{32}\sigma_3\sigma_2 \quad \rho_{31}\sigma_3\sigma_1]\begin{bmatrix} \sigma_2^2 & \rho_{21}\sigma_2\sigma_1 \\ \rho_{12}\sigma_1\sigma_2 & \sigma_1^2 \end{bmatrix}^{-1}\begin{bmatrix} x_2 - u_2 \\ x_1 - u_1 \end{bmatrix}$$

and

$$V(x_3 \mid x_1; x_2) = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{12}'$$

$$= \sigma_3^2 - [\rho_{32}\sigma_3\sigma_2 \quad \rho_{31}\sigma_3\sigma_1]\begin{bmatrix} \sigma_2^2 & \rho_{21}\sigma_2\sigma_1 \\ \rho_{12}\sigma_1\sigma_2 & \sigma_1^2 \end{bmatrix}^{-1}\begin{bmatrix} \rho_{32}\sigma_3\sigma_2 \\ \rho_{31}\sigma_3\sigma_1 \end{bmatrix}.$$

Again using random number techniques, select a value for x_3.

5. At this point we have generated a single three-dimensional point $(x_1\ x_2\ x_3)$. This point is now examined to determine how many of the three paths were successful (i.e., $x_i > 0$) and how many failed. The number of failing paths is noted.
6. The procedure in steps 1–5 is then repeated over and over in a typical Monte Carlo simulation manner with the final result being the distribution of the number of failing paths.

Finding the conditional mean and variance as previously described requires that the matrix Σ grow by one row and one column at each step of the process and that the inverse be determined at each step. The procedure for continuously finding these inverses is greatly simplified by taking advantage of the following relationship. Given

$$\Sigma = \left[\begin{array}{c|c} \Sigma_{11} & \Sigma_{12} \\ \hline \Sigma_{21} & \Sigma_{22} \end{array}\right],$$

such that Σ_{11} and Σ_{22} are both square, then it can be shown that [10]

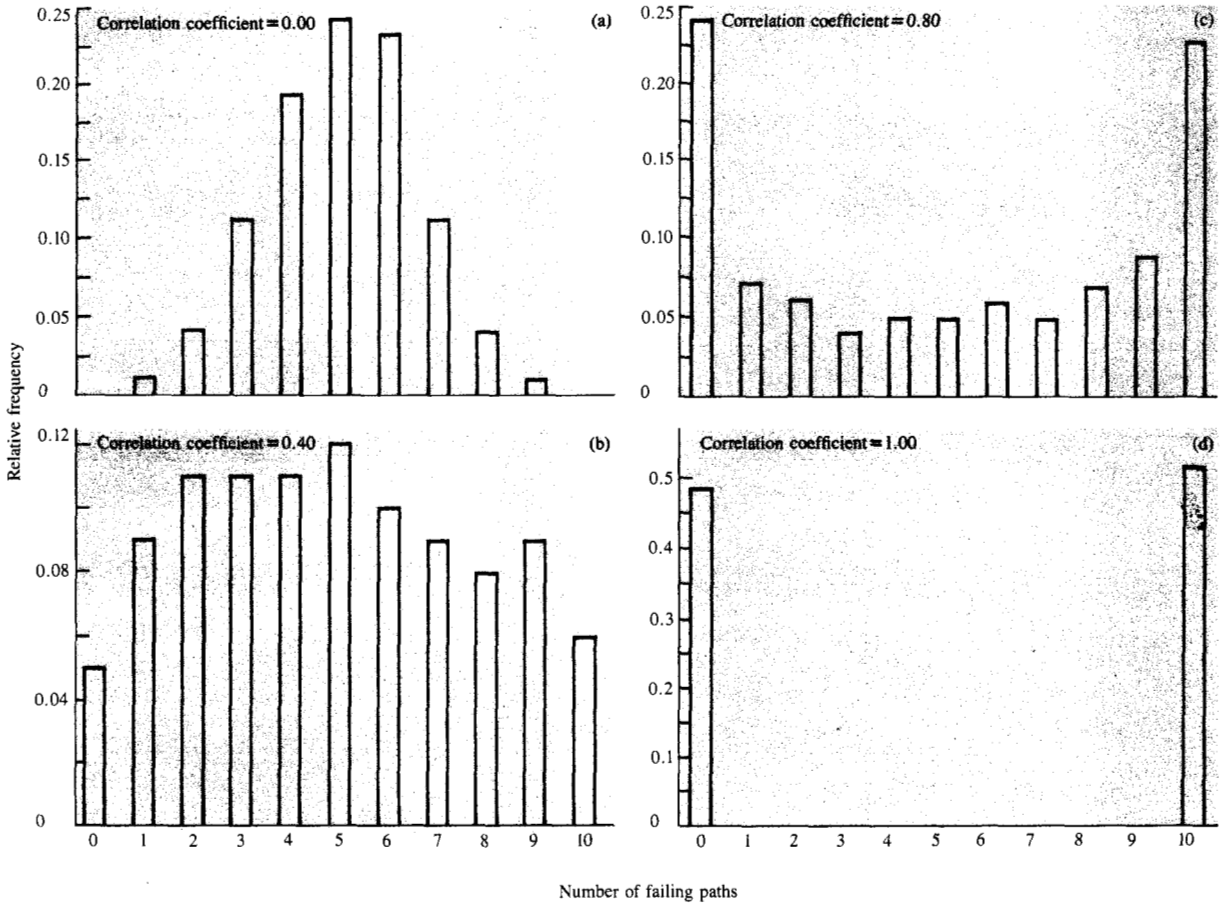

Figure 5

Dishing effect with $n = 10$ and $P = 0.50$.

$$\Sigma^{-1} = \begin{bmatrix} (\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21})^{-1} & -(\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21})^{-1}\Sigma_{12}\Sigma_{22}^{-1} \\ -\Sigma_{22}^{-1}\Sigma_{21}(\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21})^{-1} & \Sigma_{22}^{-1} + \Sigma_{22}^{-1}\Sigma_{21}(\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21})^{-1}\Sigma_{12}\Sigma_{22}^{-1} \end{bmatrix},$$

which reduces the inversion process to a much simpler sequence of vector multiplications.

• *Simulation results*

An APL routine was written that incorporates the mathematical model described in the previous section. **Table 3** of this paper shows the results of running 500 simulations on a 25-path system where each path has a $u = 2.0$ and a *sigma* = 1. All correlation coefficients are assumed to be 0.40.

Exercising this simulation program showed that results could be obtained for a system of up to approximately 500 paths. After that, computer size and time constraints made analysis of larger systems prohibitive. While 500 paths represent a formidable set of matrices to solve, this number of paths would not begin to model the requirements of large systems. Therefore, it was decided to exercise the simulation model at path sizes that could be reasonably handled. The objective was to establish a relationship between the probability of success of a path, the number of paths, and the correlation coefficient between paths.

The simulation routine will accept individual means and sigmas for individual paths and will accept specific correlation coefficients between specific paths. However, in order to facilitate data entry, the results documented here are subject to the following constraints:

1. All paths have the same mean delay and standard deviation (sigma).
2. All off-diagonal correlation coefficients are assumed to be equal.

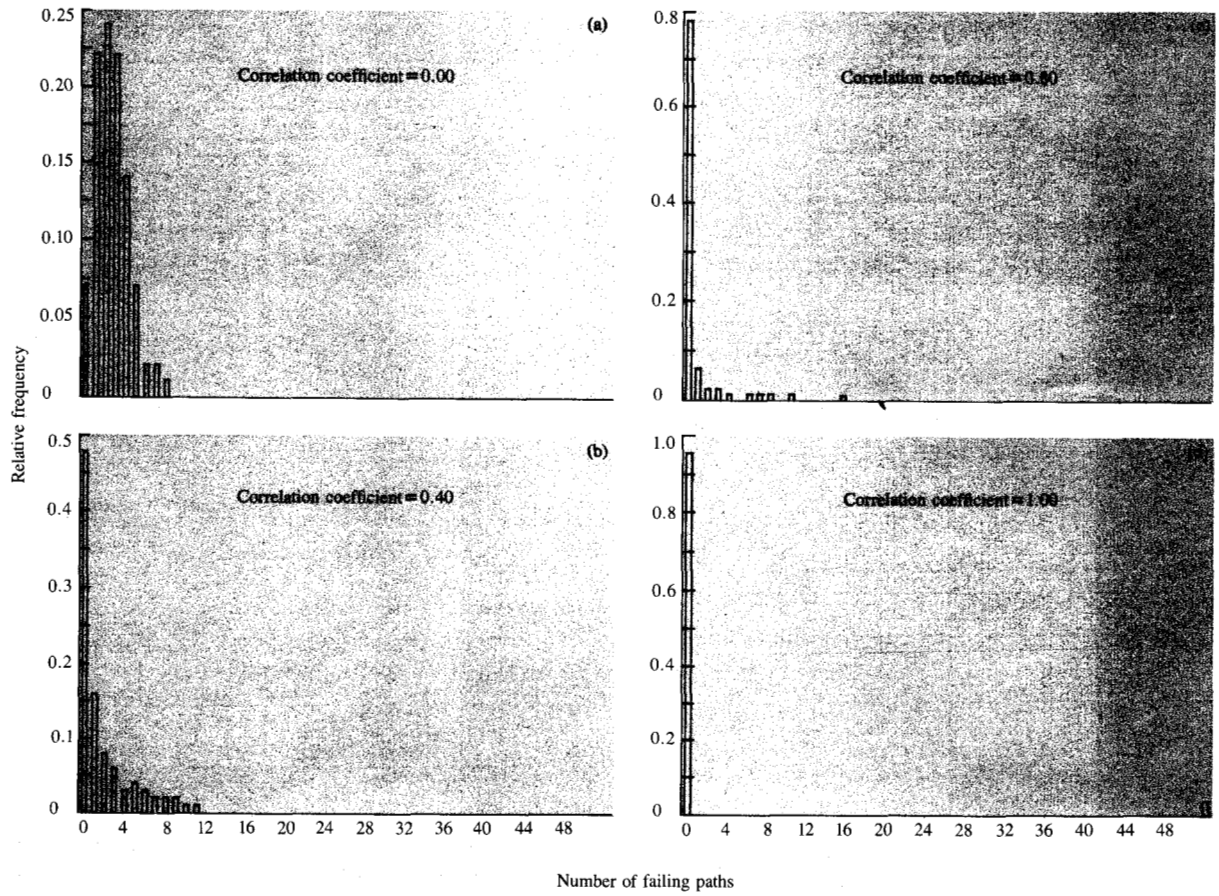

Figure 6

Dishing effect with $n = 50$ and $P = 0.95$.

Table 4 Ten-path example.

Correlation	*Expected failures*
0.0	1.57
0.1	1.54
0.2	1.62
0.3	1.62
0.4	1.60
0.5	1.57
0.6	1.61
0.7	1.60
0.8	1.57
0.9	1.60
0.95	1.54
0.99	1.68
1.0	1.53

Additional simulations were then performed to ensure that these assumptions did not affect the conclusions.

The first relationship that was discovered is that the expected number of failing paths is independent of the correlation coefficient. This is shown graphically in **Figure 2.** Each data point in Fig. 2(a) was determined from 2000 simulations of a ten-path system where each path had $\mu = 1.0$ and *sigma* = 1.0. This is a highly useful result since it implies that the binomial approximation can be used to predict the expected number of timing failures.

In order to assure ourselves that this conclusion is not a function of our simplifying assumption of equal off-diagonal correlation, the following experiment was run. Consider a ten-path system which is made up of two subgroups such that the correlation within each subgroup is given by $\rho = \rho_i$ and the correlation between subgroups is $\rho = 0$. The correlation matrix then looks like

$$\begin{bmatrix} \rho_i & 0 \\ 0 & \rho_i \end{bmatrix} \rho_i = 1 \text{ on the diagonal.}$$

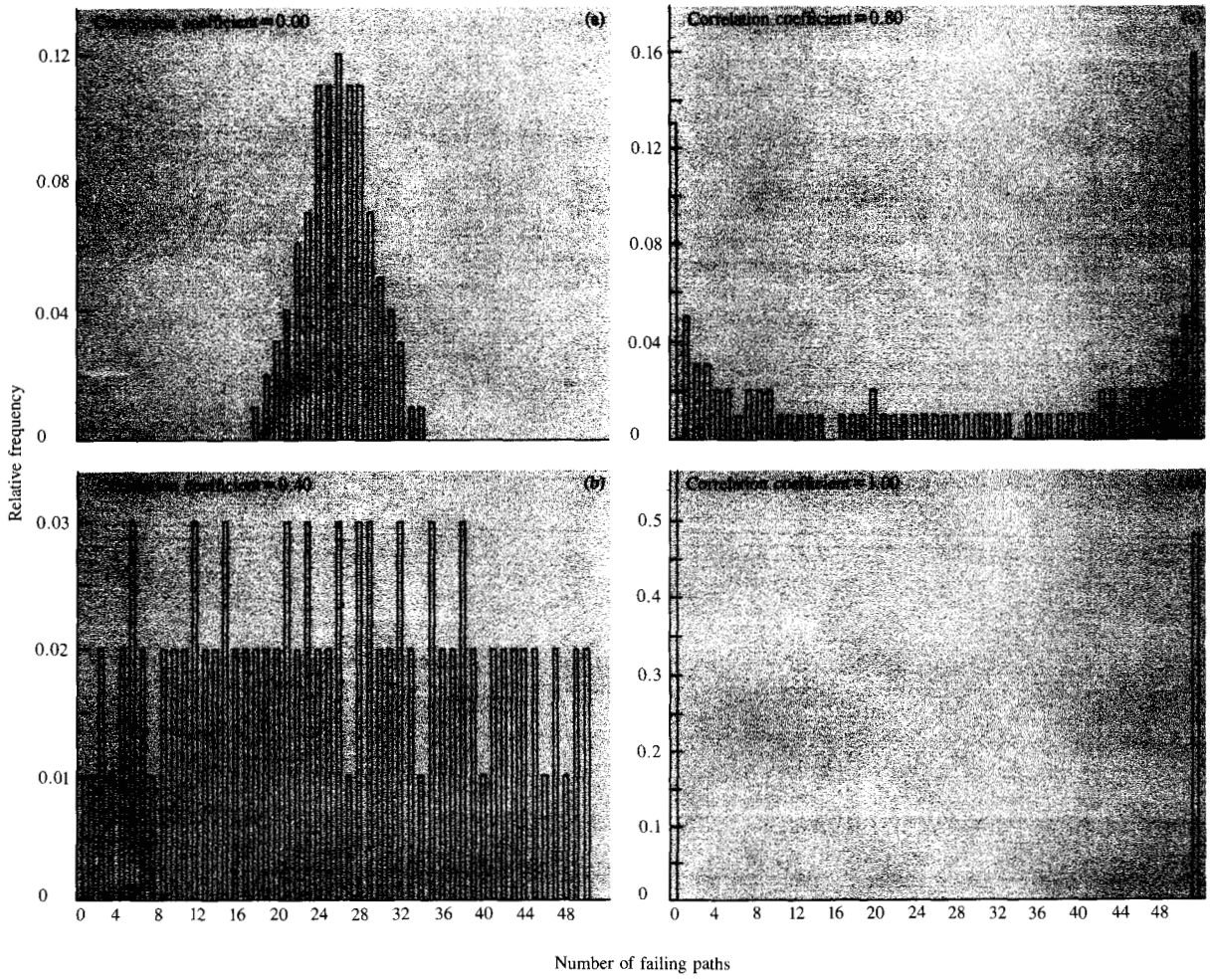

Figure 7

Dishing effect with $n = 50$ and $P = 0.50$.

This was run for values of ρ_i with the results shown in **Table 4.** A graphical representation of these results is shown in Fig. 2(b). Further, a three-path example with arbitrary correlation as follows was run:

$$\begin{bmatrix} 1 & 0.7 & 0.4 \\ 0.7 & 1 & 0.2 \\ 0.4 & 0.2 & 1 \end{bmatrix}.$$

The results are shown in **Table 5.** The data from Table 4 and Table 5 substantiate the conclusion that the expected number of timing failures is independent of correlation.

Table 5 Three-path example.

Correlation	*Expected failures*
0.0	0.4720
0.2	0.4810
0.4	0.4950
0.6	0.4835
0.8	0.4570
0.99	0.5360

The next relationship discovered is what the authors choose to call the "dishing" effect. This effect is shown in **Figure 3.** Figure 3(a) shows the distribution of the number of failing paths for a ten-path system where the probability of success for each path is $P(s) = 0.50$ and the paths are independent. Figures 3(b–g) show what happens to this distribution as the correlation coefficient is changed from 0.0

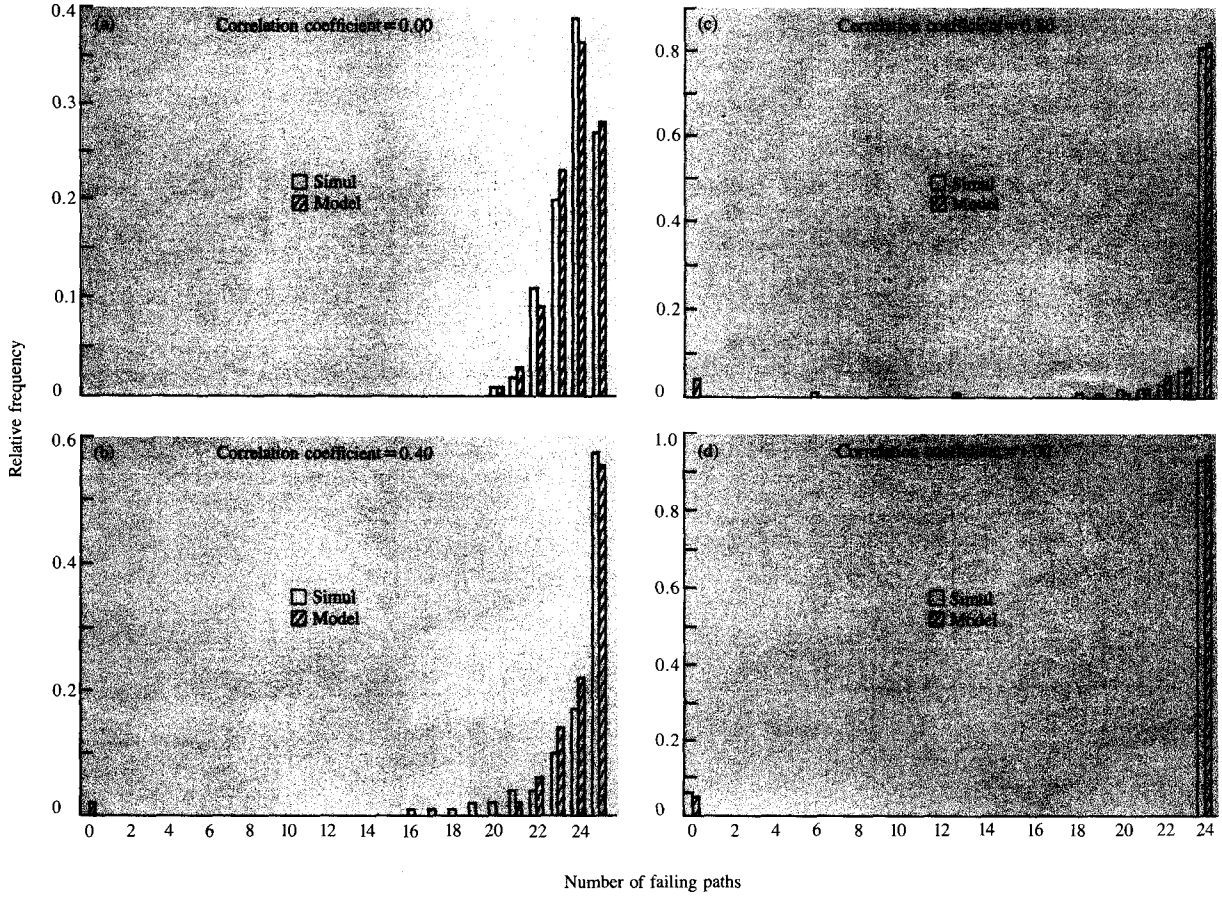

Figure 8

Dish binomial versus simulation with $n = 25$ and $P = 0.05$.

to 1.0 in several increments. Note that, as the correlation increases, the relative frequency decreases in the middle and increases in the tails. This phenomenon becomes more pronounced as the correlation increases until it takes on absolute proportions at $\rho = 1.0$. This makes intuitive sense since a correlation of 1.0 implies that if one path is "good," all paths are good. Similarly, if one path is "bad" (i.e., slow), all paths must be bad.

Further examination of this phenomenon is shown in **Figures 4–7,** which detail the "dishing" effect for the following cases: $N = 10, 50$, $P(s) = 0.95, 0.50$, and correlation $(\rho) = 0.00, 0.40, 0.80, 1.00$.

The adjusted binomial model

The problem now becomes one of defining a probability density function that emulates this dishing phenomenon. If such a distribution could be determined in terms of n, P, and ρ, it could be extrapolated in terms of n for systems with large numbers of paths.

As a starting point, let us again consider the binomial distribution written in terms of the number of failing paths x:

$$P(x) = \binom{n}{x} P^{n-x}(1 - P)^x,$$

where n = total number of cycle-limiting paths, x = number of paths failing cycle objectives, P = probability of a single path meeting its cycle objective, and $P(x)$ = probability of x paths failing cycle objectives.

Let us now make the following adjustments to arrive at what the authors choose to call the "adjusted" or "dish" binomial:

$$P(x) = P^n + \rho(P - P^n) \qquad \text{for } x = 0,$$

$$P(x) = \binom{n}{x} P^{n-x}(1-P)^x(1-\rho) \qquad \text{for } x = 1, 2, \cdots, (n-1),$$

$$P(x) = (1 - P)^n + \rho[(1 - P) - (1 - P)^n] \qquad \text{for } x = n.$$

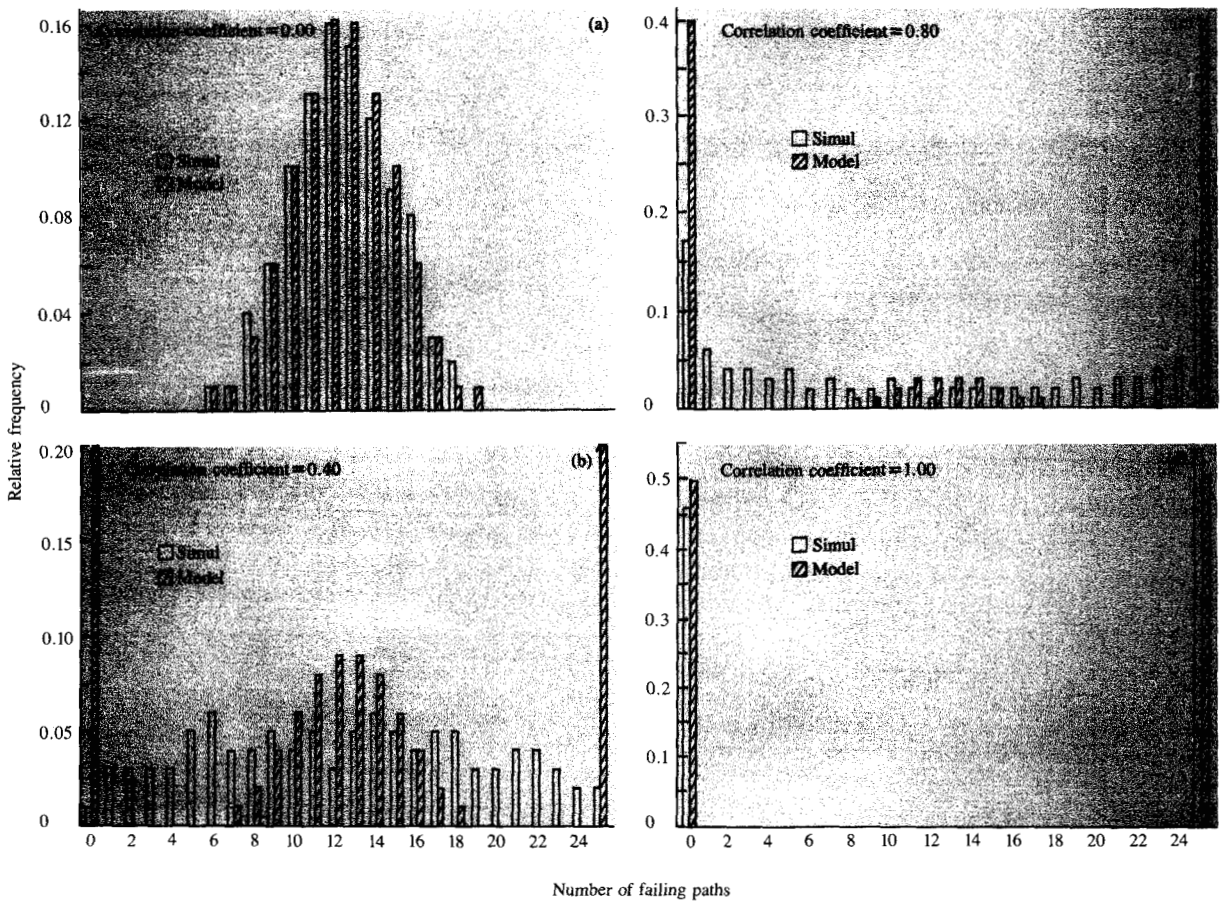

Figure 9

Dish binomial versus simulation with $n = 25$ and $P = 0.50$.

This distribution was tabulated and compared to simulation results for the range of values $N = 25$, $P = 0.05$, 0.50, 0.95, $\rho = 0.00$, 0.40, 0.80, 1.00.

The results of this comparison are shown in **Figures 8–10.** Inspection of this comparison shows excellent correlation in some areas and significant deviations in others. One area of close comparison is at the point where the probability of path success is very high. Since system designers typically verify each path in their system to high confidence levels, this is precisely the area of interest! The authors also simulated systems of 10 and 50 paths over the same set of probabilities and correlations to substantiate these conclusions. Therefore, the first term of the dish binomial appears to be a reasonable model. This term is

$$P^n + \rho(P - P^n) \qquad \text{for } x = 0.$$

The results of using this model for systems of size $n = 1000$, 10 000, 50 000, and 100 000 cycle-limiting paths are shown in **Figures 11–14.** We now have a system failure model which combines treatment of correlation with ease of computation.

Let us look at an application of these data. Either by approximation or by using the technique described in the next section, the designer might estimate that he has 1000 cycle-limiting paths in his design. Let us assume that the technologists tell the designer of our hypothetical system that the paths in his design correlate with each other with an average ρ of 0.4. The designer now has the option of selecting a *beta* confidence level for each path for his Timing Analysis. Figure 11 tells the designer that he will have a 55 percent chance that all paths in his design will be fast enough if he verifies each path to a three-sigma level. If he selects *beta* = 3.5 for his Timing Analysis runs, this probability will increase to above 85 percent. Four-sigma analysis will provide close to a 99 percent confidence. The higher confidence levels could, of course, cause a designer to fix a

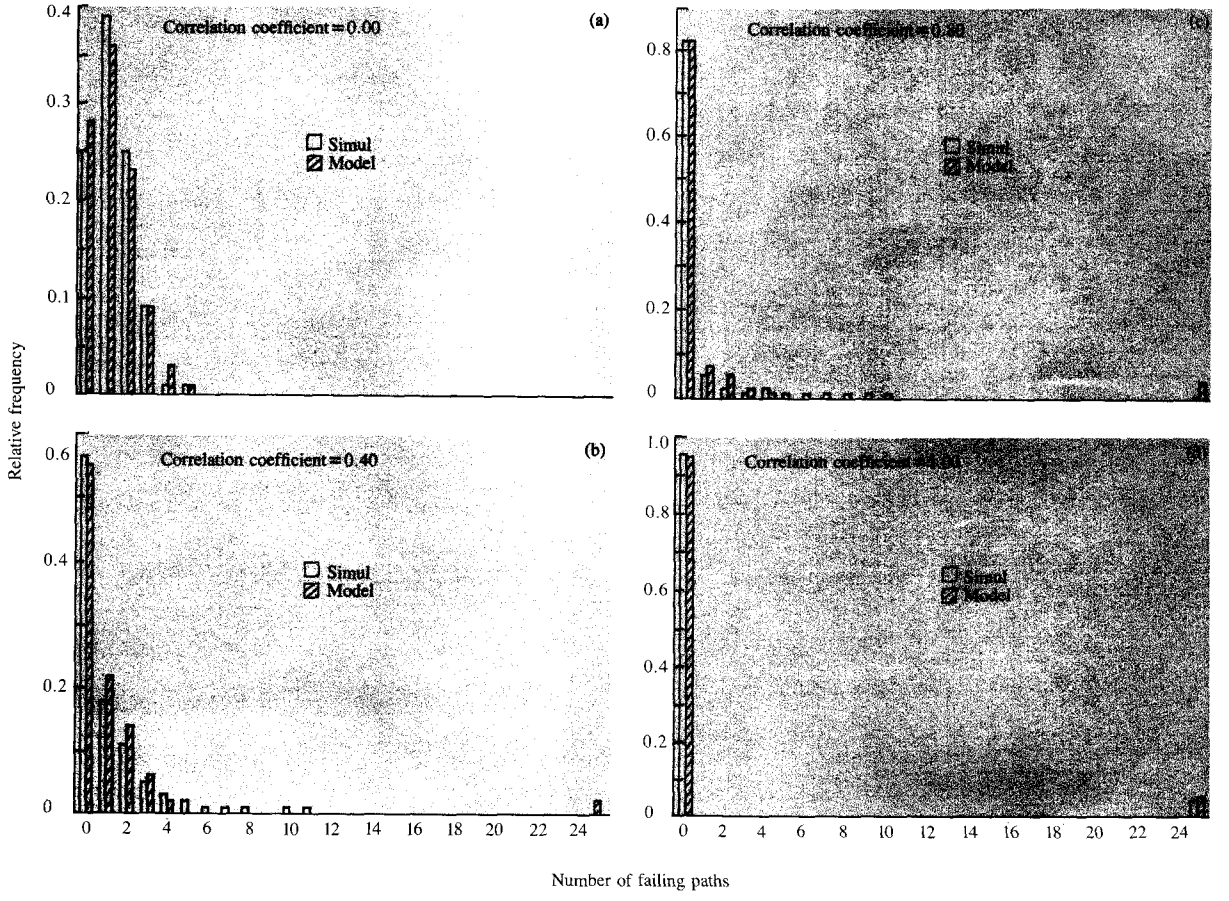

Figure 10

Dish binomial versus simulation with $n = 25$ and $P = 0.95$.

path that might have worked anyway. These data will therefore guide the designer in making the trade-offs between the costs of redesign and the costs of fixing timing problems after they occur in hardware.

The enumeration of cycle-limiting paths

Although convenient to use, our dish binomial model is of little value unless the system designer can accurately estimate the number of potentially cycle-limiting paths in his design. The Timing Analysis program, however, is a block-oriented tool that reports slack results for circuit blocks rather than on a path basis. The number of potentially critical paths is therefore not readily available. A technique [11] was therefore developed by the authors to establish the relationship between the number of circuit blocks with known slacks and the number of cycle-limiting paths.

The first step in this process is to determine the size of the computer system in terms of the number of circuit blocks. This information is readily available from Timing Analysis errata. The following formulation was then developed to relate the number of paths to this number of circuit blocks:

$$p_c = b_T(\%b_c)(t/b_T)(SRL/t)(p_c/SRL), \quad (1)$$

where p_c is the number of critical paths, b_T is the total circuit block count, $\%b_c$ is the percent of circuit blocks in the critical slack range, t is the number of timing tests, and SRL is the number of shift register latches. Each of the terms of this formulation is now described. The percentage of blocks in a specified slack range is readily determined from timing errata. For example, the system designer may only concern himself initially with circuit slacks from 0 to $+x$ time units, where x might represent five percent of his machine cycle.

Interpretation of the timing output is complicated somewhat by the fact that there may be several timing relationships tested at each Shift Register Latch (SRL), the storage element used in IBM's large-system designs. For example, a test to ensure that the data arrive in time to be

latched may be complemented by another test that ensures that the data will be triggered properly in starting the next cycle. A program that lists the originating timing test for each slack was developed and used as a Timing Analysis postprocessor. This program provides the ratio of timing tests to circuit blocks. An examination of the types of timing tests that are performed at any SRL yields the ratio of tests per SRL. Multiplication of the first four terms shown in Eq. (1) provides the number of latching SRLs in the design that participate in potentially cycle-limiting paths. We now face the intriguing problem of determining the ratio of paths to SRLs. Unfortunately, Timing Analysis output does not provide us with this information.

The authors chose to utilize the Boolean equivalence program described in [9] to aid in this determination. This program is typically used to establish an equivalence between a flowchart description of the design which is functionally simulated and a logical description of the design which is ultimately manufactured. The authors, however, saw an opportunity to use this program in solving the path problem. One of the outputs of this program is the average fan-in per segment. A segment is defined as a grouping of logic terminating either in an SRL or a primary output of the logic under analysis, as shown in **Figure 15.** By modifying one of the inputs to the equivalence program, only those segments terminating in SRLs were selected. The average fan-in per SRL can therefore be obtained. Many of these paths, however, may not be close to cycle-limiting and therefore are of little concern in this analysis. Let us then examine a wide cross section of SRLs with this average fan-in in the Timing Analysis errata and determine how many of these inputs are cycle-limiting paths. If we have chosen a representative cross section of these SRLs, we now know the ratio of critical paths to SRLs in our design. The error inherent in determining the average number of critical paths per SRL can be minimized by choosing a sufficiently large sample of SRLs with average fan-in. The authors suggest examining at least fifty SRLs in a large-scale design to minimize the potential sampling error. Multiplication of the terms of Eq. (1) then provides the desired number of paths.

We now have enough information for the number of paths to use in our dish binomial model if it is assumed that all of our critical paths have the same probability of success. This assumption would produce pessimistic results that can be considered as an upper bound for the problem. It would be more accurate to get an assessment of that probability of success for each path. Although this is clearly impossible, it is still more beneficial to use a different probability of success for each slack range.

In order to determine the standard deviation for critical paths, both nominal (*beta* = 0) and three-sigma (*beta* = 3) Timing Analysis runs were performed. For a given path, the difference in slack between the two runs is three times the standard deviation. A path at zero slack with *beta* equal to 3 would have a probability of success of 0.9987; the cumulative normal distribution evaluated from minus infinity to 3 is 0.9987. If the standard deviation were equal to one time unit, the probability of success for a path with +0.5 units slack would be 0.9988, which is comparable to zero slack if *beta* were equal to 3.5. This closer approximation of probability of success can then be used to find the number of failing paths at various confidence levels

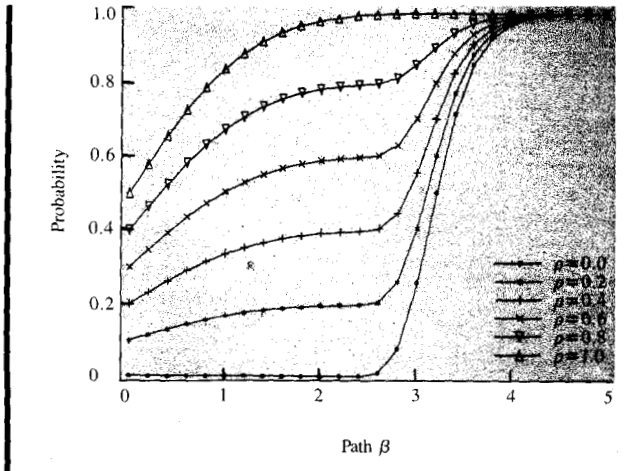

Figure 11

Probability of system success with $n = 1000$ paths.

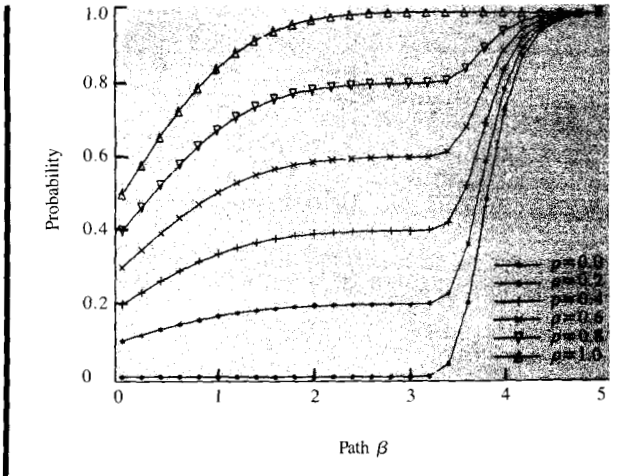

Figure 12

Probability of system success with $n = 10\,000$ paths.

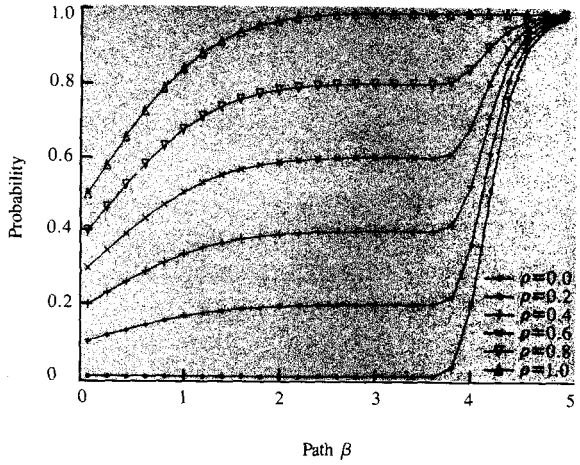

Figure 13

Probability of system success with $n = 50\,000$ paths.

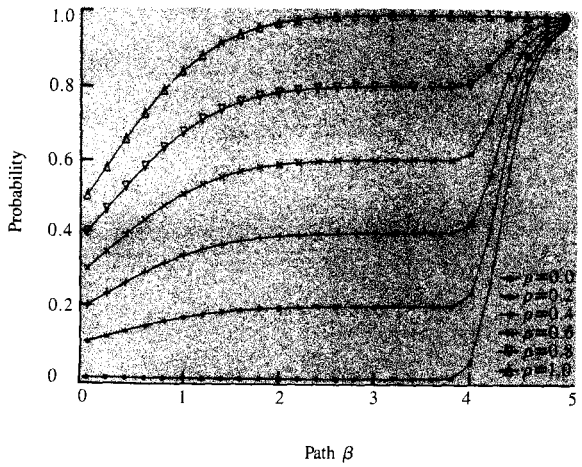

Figure 14

Probability of system success with $n = 100\,000$ paths.

within smaller slack ranges. The number of failures over all the small slack ranges can then be added to produce results for the entire machine. This concept of weighting the paths with more accurate probabilities leads to a better estimate of failing paths in the system.

Many designers of large-scale computer systems will not use the Timing Analysis and Boolean equivalence programs as part of their verification methodology. These system designers, however, should be able to adapt the techniques described here to predict the failure characteristics of their systems.

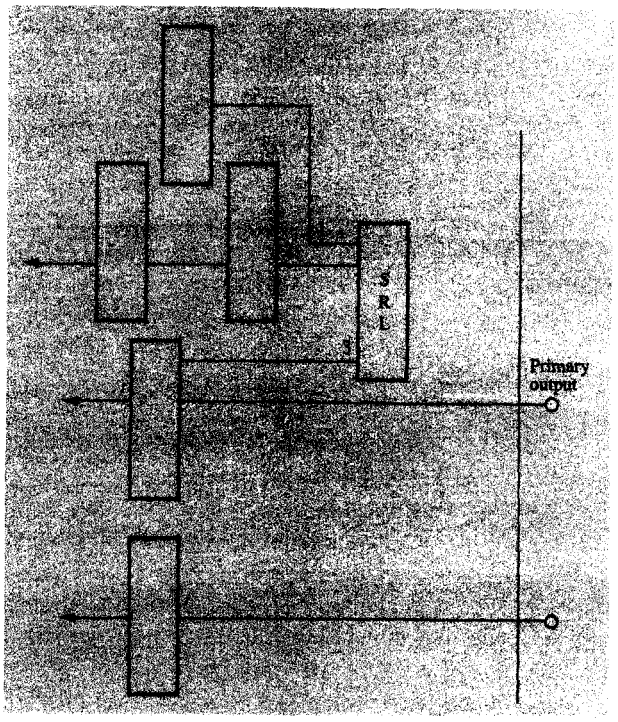

Figure 15

Logic segments; SRL fan-in = 3.

Conclusions

The techniques developed in this paper will enable designers of large VLSI computer systems to predict the timing failure characteristics of their designs prior to hardware implementation. Armed with this knowledge, designers and project managers can more intelligently quantify the various schedule and technical trade-offs required when developing a high-performance processor. These trade-offs include balancing the costs of redesign to fix timing problems identified in a paper phase of the design against the costs of fixing timing problems in hardware. The ultimate reward is the ability to deliver a large-scale computer system on schedule that meets performance and reliability objectives.

Acknowledgments

The authors are indebted to the many who provided assistance in this analysis. In particular, Evan Cohn was instrumental in the development of the dish binomial model. Brock Osborne provided insight on matrix reduction techniques, and both he and Steve Tomas developed many of the APL routines mentioned in this paper. James Shelly and Robert Hitchcock often provided helpful suggestions about various statistical timing issues, and Arthur Nadas

provided the original suggestion to attack the multivariate distribution via a simulation approach.

References

1. D. J. Pilling and H. B. Sun, "Computer-Aided Prediction of Delays in LSI Logic Systems," *Proceedings of the Tenth ACM/IEEE Design Automation Workshop*, Portland, OR, 1973, pp. 182–186.
2. M. A. Wold, "Design Verification and Performance Analysis," *Proceedings of the Fifteenth Design Automation Conference*, Las Vegas, NV, 1978, pp. 264–270.
3. T. Sasaki, A. Yamada, T. Aoyama, K. Hasegawa, S. Kato, and S. Sato, "Hierarchical Design Verification for Large Digital Systems," *Proceedings of the Eighteenth Design Automation Conference*, Nashville, TN, 1981, pp. 105–112.
4. R. Kamikawai, M. Yamada, T. Chiba, K. Furumaya, and Y. Tsuchiya, "A Critical Path Delay Check System," *Proceedings of the Eighteenth Design Automation Conference*, Nashville, TN, 1981, pp. 118–123.
5. T. M. McWilliams, "Verification of Timing Constraints on Large Digital Systems," *Proceedings of the Seventeenth Design Automation Conference*, Minneapolis, MN, 1980, pp. 139–147.
6. R. B. Hitchcock, "Timing Verification and the Timing Analysis Program," *Proceedings of the Nineteenth Design Automation Conference*, Las Vegas, NV, 1982, pp. 594–603.
7. R. B. Hitchcock, G. L. Smith, and D. D. Cheng, "Timing Analysis of Computer Hardware," *IBM J. Res. Develop.* **26**, No. 1, 100–105 (January 1982).
8. J. H. Shelly and D. R. Tryon, "Statistical Techniques of Timing Verification," *Proceedings of the Twentieth Design Automation Conference*, Miami Beach, FL, 1983, pp. 396–402.
9. G. L. Smith, R. J. Bahnsen, and H. Halliwell, "Boolean Comparison of Hardware and Flowcharts," *IBM J. Res. Develop.* **26**, No. 1, 106–116 (January 1982).
10. D. F. Morrison, *Multivariate Statistical Methods*, McGraw-Hill Book Co., Inc., New York, 1967.
11. M. R. Reiter and D. R. Tryon, "Critical Path Determination," *IBM Tech. Disclosure Bull.* **26**, No. P08-82-0331 (December 1983).

Received November 28, 1983; revised February 8, 1984

Fred M. Armstrong *IBM Data Systems Division, P. O. Box 390, Poughkeepsie, New York 12602.* Mr. Armstrong is a senior engineer in the product assurance organization. He received the B.A. degree in mathematics from St. Michael's College, Winooski, Vermont, in 1960 and the M.S. degree in applied statistics from Villanova University, Villanova, Pennsylvania, in 1968. He has been with IBM in the semiconductor area for the past fifteen years, primarily involved in research and development of mathematical models and techniques for projecting yield capabilities for various IBM semiconductor and substrate programs. He has previous experience in the defense and space industry with Raytheon and General Electric in the area of reliability modeling and systems assurance on the Gemini and Polaris systems. He also served for two years as a statistical consultant to the Institute of Advanced Studies at the U.S. Army War College in Carlisle, Pennsylvania, with major responsibilities in the area of war gaming and simulation. In 1983 he won a Division Award for the implementation of a statistical process control system in the area of multi-layer ceramic processing.

Mark R. Reiter *IBM Data Systems Division, P.O. Box 390, Poughkeepsie, New York 12602.* Mr. Reiter is a project engineer and the manager of advanced processor development, design verification. He joined IBM as a junior engineer in 1979 and worked on the development of a verification test package. He has been engaged in the static analysis and timing analysis of the advanced processor design. Mr. Reiter received a B.S. in electrical engineering from the University of Detroit in 1979. At present he is working toward a Master's degree at Syracuse University. Mr. Reiter is a member of the Institute of Electrical and Electronics Engineers, Eta Kappa Nu, and Tau Beta Pi.

David R. Tryon *IBM Data Systems Division, P.O. Box 390, Poughkeepsie, New York 12602.* Mr. Tryon is a development engineer and manager of the Design Verification and Functional Simulation Departments for advanced processor designs. He joined IBM in 1977 at Poughkeepsie and has been involved in a variety of projects in the design automation area, including timing analysis and functional simulation. He was appointed manager of the Design Verification Department in 1981 and assumed his current responsibilities in 1982. Prior to joining IBM, Mr. Tryon made contributions in the fields of digital signal processing, microprocessor development, and microcode design for space and military applications as a member of RCA's Advanced Technology Laboratory. He received a B.S. in electrical engineering from the University of Florida in 1974 and received the school's Outstanding Scholar Award. He is a member of the Institute of Electrical and Electronics Engineers and is licensed as a registered professional engineer in the state of New York. Mr. Tryon received an IBM DSD Achievement Award in 1983 for his work in timing verification.