204

40-33
MASTER

NUCLEAR DIVISION

RN-DR-0173

To AEC-NASA Space Nuclear Propulsion Office

TREE ANALYSIS CODE (TRACE)
PROGRAM E64106

NERVA Program

Contract SNP-1

October 1969

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

NUCLEAR ROCKET OPERATIONS

AEROJET-GENERAL CORPORATION

SACRAMENTO, CALIFORNIA

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

RN-DR-0173

TREE ANALYSIS CODE (TRACE)
PROGRAM E64106

NERVA Program

Contract SNP-1

NUCLEAR ROCKET OPERATIONS

October 1969

T. N. Throckmorton

H. R. Tayama

CLASSIFICATION CATEGORY**UNCLASSIFIED**

A handwritten signature in dark ink, appearing to read "J. H. Rasmussen", written over a horizontal line.

CLASSIFYING OFFICER

A handwritten date "10/16/69" in dark ink, written over a horizontal line.

DATE

AEROJET - GENERAL CORPORATION
A SUBSIDIARY OF THE GENERAL TIRE & RUBBER COMPANY

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

CONTENTS

	<u>Page</u>
I. Introduction	1
II. Statement of Problem	1
III. Mathematical Formulation	3
A. Basic Formulation	3
B. Direct Simulation	4
C. Importance Sampling	5
D. The Power Rule	6
E. The Importance Distribution	9
F. Estimation of System Failure	10
Probability Under the Power Rule	
G. Calculated Probability of System Failure	13
IV. Method of Solution	14
A. Program Logic	14
B. Generation of Random Failure	16
Times From Importance Distribution	
C. Random Inhibit Gates	17
D. Priority Gates	18
References	18
Appendix A Listing of "Program Trace"	
Appendix B User's Manual	

I. INTRODUCTION

The TRACE Program is used to determine the probability, and the most likely sources, of a specific failure of a nonrepairable complex system prior to a specified time. The systems considered are binary-response structures that can be represented by a fault tree. A fault tree is a logic diagram, in which an output event (undesired system state or fault) is related by logic, operations symbols to input events that may be lower-level system states, component states, command-function faults, or basic component failures. For this program, the component failures provide the basic input events, which are assumed to have independent time-to-failure probability distributions. The gates describe the logical relationships between events and the manner in which they may combine to produce other events, including the top event or system failure.

Because large systems with logically complex structures preclude analytic solution, computer simulation is used; but, when system failures are rare, direct simulation becomes impractical. In this situation, the Monte Carlo technique of importance sampling is used to reduce the required sample size and improve efficiency by large multiples.

II. STATEMENT OF PROBLEM

Although the probability of a possible failure mode occurring may be extremely small, the consequences for a particular system can be catastrophic, such as the explosion of a missile on its launching pad, the inadvertent firing of a warhead missile, or loss of power of a nuclear engine during certain critical states of launch. Therefore, in addition to estimating the probability of such rare events, the particular component or components must be identified whose malfunction would be the most probable cause of the events. Such an analysis of a large and logically complex system is a formidable analytic task.

Solution of the above problem requires (1) a means of logically representing the structure of a system; i.e., how certain malfunctions or failures may combine to produce the undesired event, and (2) a mathematical technique to analyze the structure once it is defined. The concept of fault trees, originated at Bell Telephone Laboratories (Reference 1), provides a solution to the first requirement by giving an orderly description of the various combinations of possible occurrences within a system that can result in a particular undesirable event. Preparation of a fault tree is a detailed, time-consuming task and is not discussed here. Instead, a technique is discussed for analyzing a fault tree, estimating the probability of tree failure, and identifying the most probable causes. A more detailed discussion of fault trees may be found in Reference 2.

Simulation affords a good mathematical technique for solving such problems, but, when the events are rare, the sample sizes required for direct simulation make this technique impractical. Importance sampling affords a method for reducing sample size and expediting the simulation. The concept of using Monte Carlo importance sampling as a variance-reducing technique has been known for some time (References 3 and 4) but has not been applied to fault-tree simulation until fairly recently. The technique has been developed by the Aerospace Division of The Boeing Company (References 5 and 6). This use of importance sampling involves changing of the given component time-to-failure distributions so that failures occur before the prescribed end-of-mission time with a higher probability than for the given distributions, and occur after the prescribed end of mission time with a correspondingly lower probability. Weighting factors are computed from the relative distributions and are used to adjust for this distortion when the probability of fault-tree failure is estimated.

III. MATHEMATICAL FORMULATION

Most of this section makes free use of information from References 5 and 6, which discuss the more complex problem of simulating repairable systems. This section pertains only to nonrepairable systems, but the ideas are very similar.

A. BASIC FORMULATION

The system is assumed to be composed of independent binary-response (go, no-go) component faults and is itself a binary-response system. Each component fails randomly in time according to individual probability distributions, and all components are "go" (not failed) at time 0. Now let

$$t_i = \text{time of failure of the } i^{\text{th}} \text{ component}$$

and

$$\underline{t} = (t_1, t_2, t_3, \dots).$$

Then $\underline{t}$ is a random vector representing one possible history of all components in the system. If the time-to-failure distribution of the i^{th} component is $f_i(t)$, the distribution of $\underline{t}$, $f(\underline{t})$, is given by

$$f(\underline{t}) = \prod_i f_i(t) \quad (1)$$

since input failures are independent.

Let $s(\underline{t})$ be the time to failure of the system corresponding to a particular $\underline{t}$. Then the probability, p , of a system failure prior to the end of the mission at time T , with all components functioning at time zero, is given by

$$p = P[s(\underline{t}) < T],$$

which is given by

$$p = \int_A f(\underline{t}) d\underline{t}, \quad (2)$$

where A is the set of all vectors $\underline{t}$ such that $s(\underline{t}) < T$.

B. DIRECT SIMULATION

Define the random variable $\delta(\underline{t})$ by

$$\delta(\underline{t}) = \begin{cases} 1 & \text{if } \underline{t} \in A \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

Then, if S is the space of all vectors $\underline{t}$, Equation 2 may be rewritten as

$$p = \int_S \delta(\underline{t}) f(\underline{t}) d\underline{t}. \quad (4)$$

In general consider any integral

$$\theta = \int_S g(x) f(x) dx, \quad (5)$$

where $f(x)$ is a probability density function defined on S, i.e.,

$$\int_X f(x) dx = 1.$$

Then θ is known as the expected value or mean value of the function $g(x)$ over $f(x)$ and may be estimated unbiasedly by

$$\hat{\theta} = \frac{1}{n} \sum_{i=1}^n g(x_i),$$

where $x_1, x_2, \dots, x_n$ is a random sample of size n from $f(x)$.

Hence, if $\underline{t}_1, \underline{t}_2, \dots, \underline{t}_n$ are sampled at random from $f(\underline{t})$, Equation 4 shows that an unbiased estimate of p is given by

$$\hat{p} = \frac{1}{n} \sum_{i=1}^n \delta(\underline{t}_i).$$

This process is called direct simulation and is time-consuming and inefficient for rare events. A rough rule of thumb is that, if it is desired to estimate p within a standard error of 10% of $\hat{p}$, about 100 occurrences of the event are required. When p is 10^{-3} , this rule calls for a sample size of 10^5 .

C. IMPORTANCE SAMPLING

Importance sampling is a slight but powerful variation of direct sampling. Again consider the general integral of Equation 5 and suppose $h(x)$ is another probability density function defined on S . Then, θ may be rewritten as

$$\theta = \int_S g(x) f(x) dx = \int_S g(x) \frac{f(x)}{h(x)} h(x) dx. \quad (6)$$

Now, if n random variables $x'_1, x'_2, \dots, x'_n$ are sampled from $h(x)$, the parameter θ may be estimated by

$$\hat{\theta} = \frac{1}{n} \sum_{i=1}^n g(x'_i) w(x'_i), \quad (7)$$

where

$$w(x) = \frac{f(x)}{h(x)} \quad (8)$$

and is called the weighting function. This weighting function compensates for the distortion caused by sampling from $h(x)$ instead of $f(x)$ and still provides an unbiased estimate. By choosing $h(x)$ so as to concentrate the sample points in the more "important" regions, the standard deviation of $\hat{p}$ can be appreciably reduced from that obtained by direct, or crude, simulation.

Suppose that a new probability density function $f^*(\underline{t})$ is defined on S , and consider Equation 4 rewritten as

$$p = \int_S \delta(\underline{t}) \frac{f(\underline{t})}{f^*(\underline{t})} f^*(\underline{t}) d\underline{t}. \quad (9)$$

Whenever $f^*(\underline{t})$ is identical to $f(\underline{t})$ except for specific components, the ratio $f(\underline{t})/f^*(\underline{t})$ is easily computed. For example, suppose the failure distributions for input components 1 and 2 are changed from $f_1(t)$ and $f_2(t)$ to $f_1^*(t)$ and $f_2^*(t)$. Then

$$f^*(\underline{t}) = f_1^*(t)f_2^*(t) \frac{f(\underline{t})}{f_1(t)f_2(t)},$$

and

$$w(\underline{t}) = \frac{f(\underline{t})}{f^*(\underline{t})} = \frac{f_1(t)f_2(t)}{f_1^*(t)f_2^*(t)}. \quad (10)$$

Consequently, if $\underline{t}_1^*, \underline{t}_2^*, \dots, \underline{t}_n^*$ are a random sample of vectors from $f^*(\underline{t})$, the probability of system failure is estimated by

$$\hat{p} = \frac{1}{n} \sum_{i=1}^n \delta(\underline{t}_i^*) w(\underline{t}_i^*) / n. \quad (11)$$

Thus, the problem becomes one of choosing importance functions for the component failures suitable to emphasize those input failure patterns that cause system failure prior to the end of the mission at time T .

D. THE POWER RULE

In performing a fault-tree simulation with importance sampling, it must be kept in mind that one of the prime purposes is to identify those sets of components most likely to cause system failure. Any set of components whose failure results in a state that induces system failure is called a cutset. A minimal cutset is one for which no proper subset is also a cutset. For direct Monte Carlo simulations, those cutsets that have the

highest probability are most likely to occur, because the probability that such a set will occur in a given trial is equal to its true probability. While importance sampling increases the number of occurrences of rare events compared with the number expected with direct simulation, care must be exercised not to sacrifice the balance of well-ordered probabilities. In other words, those cutsets having the highest true probabilities should still have the highest probabilities of occurring in the simulation.

More precisely, let P_i and P_j be true probabilities for two specific cutsets, and let P_i^* and P_j^* be the corresponding distorted probabilities under importance sampling. Then the following four conditions are necessary:

- (1) $P_i^* > P_i$
- (2) $P_i^* > P_j^*$ whenever $P_i > P_j$
- (3) $P_i^* = P_j^*$ whenever $P_i = P_j$
- (4) The sampling scheme is independent of the structure function.

For any series-parallel structure composed of independent non-repairable components, the probability that system failure will occur as a result of the i^{th} minimal cutset is given by

$$P_i = \prod_j P_{ij},$$

where P_{ij} is the failure probability for the j^{th} component of the cutset. Similarly, if independence between component pairs is maintained under the distorted sampling scheme,

$$P_i^* = \prod_j P_{ij}^*.$$

It follows from Condition 4 that, since the sampling scheme is to be valid for all cases, it must satisfy the requirements for the following special case.

Consider two sets such that

$$P_1 = P_2$$

and

$$P_1 = \prod_{j=1}^{n_1} P_{1j}, \quad P_2 = \prod_{j=1}^{n_2} P_{2j}.$$

Furthermore, let

$$\begin{aligned} P_{1j} &= a \\ P_{2j} &= b \end{aligned} \quad \text{for all } j.$$

Then

$$P_1 = a^{n_1} = P_2 = b^{n_2}.$$

For a distorted sampling scheme that changes a to a^* and b to b^* ,

$$P_1^* = (a^*)^{n_1} = P_2^* = (b^*)^{n_2}$$

under Condition 3. Thus,

$$\frac{\log a^*}{\log a} = \frac{\log b^*}{\log b},$$

which is true only if

$$P_{ij}^* = P_{ij}^r.$$

This implies that

$$P_i^* = P_i^r.$$

Imposing Conditions 1 and 2 leads to the requirement $0 < r < 1$.

Hereafter, any importance density function with the property that

$$P^* = P^r \quad 0 < r < 1 \quad (12)$$

will be called the power rule. For any other sets i and j where $P_i > P_j$, the power rule implies $P_i^* > P_j$ and $P_i^* > P_i$. Direct simulation corresponds to the special case $r = 1$.

E. THE IMPORTANCE DISTRIBUTION

This program assumes exponential time-to-failure distributions for the components. Consider a particular component and let λ denote the mean failure rate (failures per unit of time). The probability density function is given by

$$f(t) = \lambda e^{-\lambda t} \quad t > 0, \quad (13)$$

and the probability of component failure prior to time T is

$$P = \int_0^T \lambda e^{-\lambda t} dt = 1 - e^{-\lambda T}. \quad (14)$$

Using the power rule, let $P^* = P^r$ (for $0 < r < 1$) and define the importance density function by

$$f^*(t) = \begin{cases} \frac{P^*}{P} f(t) & \text{if } 0 < t \leq T \\ \frac{1-P^*}{1-P} f(t) & \text{if } t > T. \end{cases} \quad (15)$$

The fact that this is a probability density function may be verified by

$$\begin{aligned} \int_0^\infty f^*(t) dt &= \int_0^T f^*(t) dt + \int_T^\infty f^*(t) dt \\ &= \frac{P^*}{P} \int_0^T f(t) dt + \frac{1-P^*}{1-P} \int_T^\infty f(t) dt \\ &= P^* + (1-P^*) = 1. \end{aligned}$$

Since the component failures are independent,

$$f^*(\underline{t}) = \prod_i f_i^*(\underline{t}). \quad (16)$$

The weighting function $w(\underline{t})$ is therefore given by

$$w(\underline{t}) = \frac{f(\underline{t})}{f^*(\underline{t})} \prod_i w_i(t), \quad (17)$$

where

$$w_i(t) = \frac{f_i(t)}{f_i^*(t)} = \begin{cases} \frac{P_i}{P_i^*} & \text{if } 0 \leq t \leq T \\ \frac{1-P_i}{1-P_i^*} & \text{if } t > T. \end{cases} \quad (18)$$

F. ESTIMATION OF SYSTEM FAILURE PROBABILITY UNDER THE POWER RULE

If $t_1^*, t_2^*, \dots, t_n^*$ are a random sample of vectors from $f^*(\underline{t})$, Equations 11 and 17 show that the system failure probability may be estimated by

$$\hat{p} = \sum_{i=1}^n \delta(\underline{t}_i^*) w(\underline{t}_i^*) / n,$$

where

$$w(\underline{t}_i^*) = \prod_j w_j(t_{i^*}^*).$$

The following paragraphs, however, show that weights for only those components in the minimal cutset need be used.

From Equation 4 we have the desired probability expressed as

$$p = \int_S \delta(\underline{t}) f(\underline{t}) d\underline{t}, \quad (19)$$

where $\underline{t}$ is a vector of component failure times and S is the space of all such vectors. Now,

$$\int_S f^*(\underline{t}^*) d\underline{t}^* = 1; \quad (20)$$

so we may multiply Equation 19 by this integral without changing its value; i.e.,

$$\begin{aligned} p &= \int_S \delta(\underline{t}) f(\underline{t}) d\underline{t} \int_S f^*(\underline{t}^*) d\underline{t}^* \\ &= \int_{S \times S} \delta(\underline{t}) f(\underline{t}) f^*(\underline{t}^*) d\underline{t} d\underline{t}^*. \end{aligned} \quad (21)$$

Now, let $g(\underline{t}|\underline{t}^*)$ be an importance density for $\underline{t}$ that is conditional on the outcome $\underline{t}^*$. Then Equation 21 may be written

$$p = \int \delta(\underline{t}) \frac{f(\underline{t})}{g(\underline{t}|\underline{t}^*)} g(\underline{t}|\underline{t}^*) f^*(\underline{t}^*) d\underline{t} d\underline{t}^*; \quad (22)$$

and the probability estimate is given by

$$\hat{p} = \sum_j \delta(\underline{t}_j) w(\underline{t}_j, \underline{t}_j^*) / n, \quad (23)$$

where

$$w(\underline{t}_j, \underline{t}_j^*) = \frac{f(\underline{t}_j)}{g(\underline{t}_j|\underline{t}_j^*)} \quad (24)$$

The conditional distribution $g(\underline{t}|\underline{t}^*)$ is defined as follows. Select random t_i^* 's from $f_i^*(t)$ and check the system for failure assuming the t_i^* 's are in fact the life lengths of the components. If the system has not failed, generate new random variables t_i for all the inputs by using the original $f(t)$. If the system has failed, let C be the minimal cutset of components causing the failure. Then define t_i so that $t_i = t_i^*$ for those components in C , and generate new random variables t_i from $f_i(t)$ for those not in C . The importance density for t_i conditional on $\underline{t}^*$ under this definition is

$$g(t_i | t_i^*) = \begin{cases} f_i^*(t_i) & \text{if } i \in C \\ f_i(t_i) & \text{if } i \notin C, \end{cases} \quad (25)$$

and the weight function becomes

$$w(\underline{t}, \underline{t}^*) = \frac{f(\underline{t})}{g(\underline{t}|\underline{t}^*)} = \prod_i \frac{f_i(t_i)}{g_i(t_i|t_i^*)},$$

where

$$\frac{f_i(t_i)}{g_i(t_i|t_i^*)} = \begin{cases} \prod_i \frac{f_i(t_i^*)}{f_i^*(t_i^*)} & \text{for } i \in C \\ 1 & i \notin C. \end{cases} \quad (26)$$

The weight function thus equals one for all components not in the minimal cutset, and, for those in the cutset, the ratio is $f_i(t)/f_i^*(t)$. For the i^{th} component to be in the minimal cutset, t_i^* must be less than T ; so, from Equation 18, this ratio equals P_i/P_i^* . Thus

$$w(\underline{t}, \underline{t}^*) = \prod_{i \in C} \frac{P_i}{P_i^*}, \quad (27)$$

and, from Equation 23,

$$\begin{aligned} \hat{p} &= \sum_{j=1}^n \delta(\underline{t}_j) \prod_{i \in C_j} \frac{P_i}{P_i^*} / n \\ &= \sum_i \prod_{i \in C_k} \frac{P_i}{P_i^*} / n, \end{aligned} \quad (28)$$

where k ranges over only those trials resulting in system failure. In other words, a trial weight consists of the product of component weights for only those components in the minimal cutset; and $\hat{p}$ is obtained as the sum, over all failed trials, of the trial weights divided by the total number of trials.

The variance of $\hat{p}$ is estimated by

$$V(\hat{p}) = \frac{\sum_{j=1}^n (\delta_j w_j - \hat{p})^2}{n(n-1)}. \quad (29)$$

G. CALCULATED PROBABILITY OF SYSTEM FAILURE

Once the minimal cutsets for a specific tree have been identified, another approximation of system failure probability is available. Let the k components of the system be denoted by 1, 2, ..., k ; and suppose the following minimal cutsets have been identified: (1,2), (3,4), and (1,3,5). If ϕ represents the tree status, we can define the Boolean statement

$$\phi = 12 + 34 + 135. \quad (30)$$

The probability of tree failure may be approximated by

$$P(\phi) = P(12) + P(34) + P(135), \quad (31)$$

since, when the probabilities are small, second- and higher-order terms such as $P(12 \text{ and } 34)$ are negligible in comparison with the first-order terms. Since the component faults are assumed to be independent and have exponential failure distributions, $P(\phi)$ is easily calculated for a nonrepairable system by

$$P(\phi) = \sum_k \prod_{i \in C_k} (1 - e^{-\lambda_i T}), \quad (32)$$

where k ranges over all minimal cutsets C_k .

This calculation is provided by the program and should afford reasonable check on $\hat{p}$.

IV. METHOD OF SOLUTION

A. PROGRAM LOGIC

The program uses the following basic logic.

1. A forcing parameter, r , is read in.

The following calculations are performed for each component:

$$P_i = 1 - e^{-\lambda_i T}$$

$$P_i^* = P_i^r$$

$$w_i = P_i / P_i^*$$

2. A trial is initialized.

All components are set to "go;" i.e., not failed (FALSE in the program). All houses* are checked and the appropriate ones are failed. The tree status is checked, and, if failed status is indicated, the program terminates with an error message. If not, the time to failure is randomly generated from importance distributions for each component; and those times that are less than T are stored.

*A house identifies a system state normally expected to be in existence throughout the mission phase being analyzed. Houses always input to an AND gate and are assigned a probability of occurrence of 1.0 when the branch they are in is to be included in the simulated mission, and are assigned 0.0 when it is desired to delete the particular branch from the analysis.

3. Component failures are simulated.

The smallest t_i for all components not already failed is found, and the corresponding component is set to fail. The tree status is checked; if failed, step 4 is performed; if not failed step 3 is repeated. If all checks for $t_i < T$ are completed, a new trial is started at step 2.

4. Minimal cutset is identified.

The procedure used here does not necessarily yield the smallest minimal cutset, but it does provide a minimal cutset in that no proper subset of it is a cutset. By starting with the last component failed and working in the reverse order from that in which the failures occurred, a component is set to "go" and the tree is checked. If the tree is still failed, the last component checked is left at "go" and the next component is set to "go" and so on. If when checked, the tree is not failed, the last component set to "go" is reset to failed; and the next component in the cutset is set to "go." After all failed components have been reset and the tree status checked in this way, those remaining in a failed status form the minimal cutset; i.e., if any one of them is now set to "go," the tree is not failed. If this cutset has already been identified, the number of times it has occurred is increased by one; and a new trial is started at step 2. If this is a new cutset, step 5 is performed.

5. Minimal cutset calculations.

For each new minimal cutset, the following statistics are calculated.

$$\text{Trial Weight} = wtr_i = \prod_{j \in C_i} w_j,$$

$$\text{Cutset probability} = Ps_i = \prod_{j \in C_i} P_j.$$

The status of all gates is checked, and a list of those in a failed status is printed. If the desired number of trials has been completed, step 6 is performed; if not, a new trial is started at step 2.

6. Final Calculations

Let k_j be the number of times the j^{th} minimal cutset occurred and $k = \sum_j k_j$ be the total number of failures out of n trials. The following calculations are then performed:

$$(1) \quad \hat{p} = \sum_j k_j \text{wtr}_j / n.$$

$$(2) \quad V(\hat{p}) = \frac{(n-k)\hat{p}^2 + \sum_j k_j (\text{wtr}_j - \hat{p})^2}{n(n-1)}.$$

$$(3) \quad Psf = \sum_j Ps_j.$$

B. GENERATION OF RANDOM-FAILURE TIMES FROM IMPORTANCE DISTRIBUTION

For a given component, the importance density function $f^*(t)$ is defined by Equation 15. From $f^*(t)$, the cumulative distribution, $F^*(t)$, is obtained as

$$F^*(t) = \begin{cases} \frac{P^*}{P} (1 - e^{-\lambda t}) & t \leq T \\ P^* + \frac{1-P^*}{1-P} (e^{-\lambda T} - e^{-\lambda t}) & t > T, \end{cases} \quad (33)$$

which is shown graphically in Figure 1.

Figure 1 - $F^*(t)$ vs t

A uniform random number U is obtained from subroutine RANDU, which uses the power residue method, and is tested for being less than P^* . If U is $> P^*$, the failure time will be greater than T and is not calculated. If $U \leq P^*$, it is set equal to $F^*(t)$; and the equation is solved for t to give

$$t = -\log_e \left(1 - \frac{P}{P^*} U\right) / \lambda.$$

C. RANDOM INHIBIT GATES

Let q be the probability of failure associated with a random inhibit gate. To maintain the proper order of probabilities, importance sampling is also used on these. Let $q^* = q^R$; then, each time the tree is

checked, a uniform random number U is generated for each gate of this type. If $U \leq q^*$, the gate will fail if its input is failed. If $U > q^*$, the gate cannot fail. Weights q/q^* are also calculated. For each random inhibit gate that is failed for a minimal cutset, these weights are included in the trial weight calculations of step 5.

D. PRIORITY GATES

A priority gate fails only if the inputs to it occur in a particular order. However, when a cutset has been found and a minimal cutset is being identified in step 4, the priority is ignored because the inputs originally failed in the proper order whereas, in selecting a minimal cutset, components are checked in the reverse order. Unless priority is ignored, these gates could not be reset to fail; and erroneous cutsets and critical paths would be obtained. Because the inputs to the gate originally failed in the proper order and it acts like an AND gate, the reverse-order process yields proper results by ignoring priority during this tracing step.

REFERENCES

1. Mearns, A. B., Fault Tree Analysis: The Study of Unlikely Events in Complex Systems, presented at the System Safety Symposium, 8-9 June 1965, Seattle, Washington.
2. Haasl, David, F., Advanced Concepts in Fault Tree Analysis, presented at the System Safety Symposium, 8-9 June 1965, Seattle, Washington.
3. Meyer, Herbert A., ed., Symposium on Monte Carlo Methods, John Wiley & Sons, Inc., New York, 1956.
4. Hammersley, J. M., and Handscomb, D. C., Monte Carlo Methods, John Wiley & Sons, Inc., New York, 1964.
5. Nagel, Phyllis M., Importance Sampling in Systems Simulation, Boeing Document No. D2-84249, 1966.
6. Schroder, R. J., and Nagel, P. M., The Efficient Simulation of Rare Events in Complex Systems, Boeing Document No. D2-114072-1, 1967.

APPENDIX A

LISTING OF "PROGRAM TRACE"

```

C *****TRAC0001
C                                     TRAC0002
C                                     TRAC0003
C                                     TRAC0004
C                                     TRAC0005
C                                     TRAC0006
C                                     TRAC0007
C                                     TRAC0008
C                                     TRAC0009
C                                     TRAC0010
C *****TRAC0011
C                                     TRAC0012
C                                     TRAC0013
C THIS PROGRAM PERFORMS A MONTE CARLO SIMULATION OF FAULT TREESTRAC0013
C TO ASSIST IN ANALYZING LARGE COMPLEX SYSTEMS. IT PROVIDES TRAC0014
C ESTIMATES OF THE PROBABILITY OF SYSTEM FAILURE PRIOR TO A TRAC0015
C SPECIFIED TIME. IT IS USED PRIMARILY TO DETECT MINIMAL CUTSETS, TRAC0016
C THAT IS, THOSE MINIMAL SETS OF COMPONENTS WHOSE FAILURE CAUSES TRAC0017
C SYSTEM FAILURE. THE TIME TO FAILURE FOR A COMPONENT IS ASSUMED TRAC0018
C TO BE A RANDOM VARIABLE FOLLOWING AN EXPONENTIAL DISTRIBUTION. TRAC0019
C BECAUSE SYSTEM FAILURES ARE NORMALLY RARE EVENTS THE TECHNIQUE TRAC0020
C OF IMPORTANCE SAMPLING IS USED TO REDUCE THE NUMBER OF TRIALS TRAC0021
C REQUIRED. TRAC0022
C                                     TRAC0023
C *****TRAC0024
C                                     TRAC0025
C ***** IDENTIFICATION OF VARIABLES *****TRAC0026
C                                     TRAC0027
C COMPNM(1000) - COMPONENT NAMES AND HOUSE NAMES TRAC0028
C COMPST(1000) - COMPONENT STATUS (TRUE = FAILED) TRAC0029
C ENDPRB - SIGNIFIES END OF PROBLEM TRAC0030
C FKTR - FLOATING POINT VALUE OF KTR TRAC0031
C FLGATE(500) - INDICES OF FAILED GATES FOR A MINIMAL CUTSET TRAC0032
C FNSFL = FLOATING POINT VALUE OF NSFL TRAC0033
C GATENM(2000) - GATE NAMES TRAC0034
C GATEST(2000) - GATE STATUS (TRUE = FAILED) TRAC0035
C GTYPE = GATE TYPE - OR, AND, R.I. TRAC0036
C HFLAG - FLAG TO DETECT HOUSES TRAC0037
C IFLAG - INDICATES SYSTEM FAILURE ON HOUSES ALONE TRAC0038
C INH - FLAG INDICATES TOP DUE TO RI GATE RATHER THAN INPUT COMP TRAC0039
C IPRINT - OPTIONAL PRINT FLAG SET BY 'PRINT' CARD. TRAC0040
C IRI(100) - INDICES OF RANDOM INHIBIT GATES TRAC0041
C ITEMP - TEMPORARY INTEGER STORAGE TRAC0042
C IX - CONSTANT INTEGER TO INITIALIZE RANDU (IX MUST BE ODD) TRAC0043
C IY - CONSTANT USED IN RANDOM NUMBER GENERATION TRAC0044
C K - NO. OF COMPONENTS IN MINIMAL CUTSET TRAC0045
C KASE - CASE COUNT TRAC0046
C KOUNT(500) - COUNT OF NUMBER OF OCCURRENCES OF ITH MINIMAL CUT SETTRAC0047
C KTEN - COUNT OF MINIMAL CUTSETS WITH MORE THAN 10 COMPONENTS TRAC0048
C KTR - COUNT OF NUMBER OF TRIALS TRAC0049
C LAMDA(1000) - COMPONENT FAILURE RATES AND HOUSE PROB. OF FAILURE TRAC0050
C LAST - SIGNIFIES END OF COMPONENT (OR GATE) CARDS TRAC0051

```

```

C      LINE - LINE COUNTER FOR OUTPUT                                TRAC0052
C      MINSET(500,10) - STORAGE AREA FOR MINIMAL CUTSETS          TRAC0053
C      MSET(10) - INDICES OF MINIMAL CUTSET COMPONENTS             TRAC0054
C      NC - NUMBER OF COMPONENTS                                    TRAC0055
C      NC1 - NO. COMPONENTS + 1 (INDEX OF FIRST HOUSE)             TRAC0056
C      NCFL - NUMBER OF COMPONENTS FAILED IN A TRIAL PRIOR TO T    TRAC0057
C      NCOMP - TOTAL NUMBER OF COMPONENTS AND HOUSES (NC+NH)       TRAC0058
C      NCUT - NUMBER OF COMPONENTS + HOUSES IN CUTSET              TRAC0059
C      NCUT1 - NCUT-1                                               TRAC0060
C      NCUT2 - NCUT + 1                                             TRAC0061
C      NEW - FLAG FOR STARTING NEW PAGE                             TRAC0062
C      NEXT(1000) - INDICES OF FAILED COMPONENTS                   TRAC0063
C      NG - NUMBER OF GATES                                         TRAC0064
C      NH - NUMBER OF HOUSES                                         TRAC0065
C      NHFL - NUMBER OF HOUSES FAILED                               TRAC0066
C      NHFL1 - NHFLD + 1                                            TRAC0067
C      NMSET(500) - NUMBER COMPONENTS IN ITH MINIMAL CUTSET        TRAC0068
C      NRI - NUMBER OF RANDOM INHIBIT GATES                         TRAC0069
C      NSETS - NUMBER OF DIFFERENT MINIMAL CUTSETS FOUND           TRAC0070
C      NSFL - NUMBER OF SYSTEM FAILURES                             TRAC0071
C      NTR - NUMBER OF TRIALS PER CASE                              TRAC0072
C      P(1000) - PROB OF FAILURE OF ITH COMPONENT PRIOR TO TIME T  TRAC0073
C      P1 - TEMPORARY CALCULATION OF PSF(I)                         TRAC0074
C      PERCNT - PERCENT SYSTEM FAILURES WITH 10 OR LESS COMP. IN MIN. SETTRAC0075
C      PI(100) - PROBABILITY OF FAILURE OF INHIBIT GATES            TRAC0076
C      PROB - PROBABILITY OF SYSTEM FAILURE ESTIMATED FROM WEIGHTING FUNCTRAC0077
C      PS - PROBABILITY OF SYSTEM FAILURE, DIRECT CAL. = SUM OF PSF(I) TRAC0078
C      PSF(500) - PROB FAILURE OF MINIMAL CUTSET BY DIRECT COMPUTATION TRAC0079
C      PSTAR(1000) - PROB OF FAILURE OF ITH COMP FOR IMPORTANCE DIST. TRAC0080
C      R - FORCING PARAMETER FOR IMPORTANCE SAMPLING                TRAC0081
C      RICH = PROBABILITY OF FAILURE FLAGS FOR R.I. GATES.          TRAC0082
C      IF RANDON VARIABLE IS THAN PROBABILITY,FLAG = .TRUE.        TRAC0083
C      OTHERWISE,FLAG = .FALSE.                                     TRAC0084
C      STDERR - STANDARD ERROR OF PROB                              TRAC0085
C      T - MISSION LENGTH                                           TRAC0086
C      TEMP1 - TEMPORARY STORAGE                                     TRAC0087
C      TEMPNM - TEMPORARY NAME STORAGE                              TRAC0088
C      TITLE - TITLE IDENTIFICATION OF CASE                         TRAC0089
C      TNEXT(1000) - TIME OF FAILURE OF COMPONENTS                  TRAC0090
C      U - UNIFORM RANDOM NUMBER BETWEEN 0 AND 1                    TRAC0091
C      VAR - VARIANCE OF PROB                                        TRAC0092
C      WT(1000) - WEIGHTING FUNCTION FOR ITH COMPONENT              TRAC0093
C      WT1 - TEMPORARY CALCULATION OF WTR(I)                       TRAC0094
C      WTI(100) - WEIGHTS FOR RANDOM INHIBIT GATES                  TRAC0095
C      WTR(500) - WEIGHTING FUNCTION FOR TRIAL                      TRAC0096
C      *****TRAC0097
C      *****TRAC0098
C      *****TRAC0099
C      COMMON /MTREE/ TITLE(9), COMPNM(1000), GATENM(2000), PI(100), TRAC0100
1      WTI(10), R, IRI(100), IX, K2, GTYPE(2000), COMPST(1000), TRAC0101
2      GATEST(2000), RICH(100)                                     TRAC0102

```

```

REAL*8 TITLE,  TEMPNM,COMPNM,GATENM, LAST/'LAST'/,ENDPRR/'END '/ TRAC0103
REAL*8 CHANGE/'CHANGE'/, PRINT/'PRINT'/ TRAC0104
REAL*4 LAMDA(1000) TRAC0105
INTEGER*4 FLGATE(500) TRAC0106
INTEGER*2 NMSET(500),HFLAG,H/'H'/ TRAC0107
INTEGER * 2 GTYPE, IAST(500), PG/'P'/ TRAC0108
LOGICAL*1 COMPST,GATEST, TOP,RICH TRAC0109
DIMENSION P(1000),PSTAR(1000),WT(1000),TNEXT(1000),NEXT(1000), TRAC0110
1 WTR(500),KOUNT(500),MINSET(500,10),PSF(500),MSET(10) TRAC0111
DIMENSION RSAVE(20) TRAC0112
C TRAC0113
C *****TRAC0114
C TRAC0115
C DIMENSIONS IMPLY FOLLOWING RESTRICTIONS TRAC0116
C TRAC0117
C MAX OF 1000 COMPONENTS AND HOUSES IN TOTAL TRAC0118
C MAX OF 2000 GATES IN TOTAL TRAC0119
C MAX OF 100 RANDOM INHIBIT GATES TRAC0120
C MAX OF 500 DISTINCT MINIMAL CUTSETS WILL BE FOUND TRAC0121
C TRAC0122
C *****TRAC0123
C TRAC0124
9000 FORMAT('1TRACE -- TREE ANALYSIS CODE, AEROJET-GENERAL CORPORATION,TRAC0125
1 SACRAMENTO, CALIFORNIA'//) TRAC0126
9010 FORMAT( 9A8) TRAC0127
9020 FORMAT( 11, 29X, 9A8 ) TRAC0128
9025 FORMAT(I6, 1X, I9, 1X, E13.6 ) TRAC0129
9030 FORMAT(A1, T1, A6, 1X, F1.0, 1X, E13.6 ) TRAC0130
9040 FORMAT( //25H NUMBER OF TRIALS = ,I9/25H RANDOM NO. CONSTANT TRAC0131
1= ,I9/21H MISSION LENGTH = , F13.1//) TRAC0132
9045 FORMAT(25H NUMBER OF COMPONENTS = ,I9/25H NUMBER OF HOUSES = TRAC0133
1 ,I9/30H TOTAL NO. COMP. AND HOUSES = ,I4//) TRAC0134
9050 FORMAT(// 28X,'COMPONENT NAMES AND PROBABILITIES TIMES 10 TO THE TRAC0135
16TH POWER' //) TRAC0136
9060 FORMAT( 4(1X,I4,2X,A6,2X, E13.6,5X)) TRAC0137
9070 FORMAT(//47X,'HOUSE NAMES AND PROBABILITY ASSIGNED' //) TRAC0138
9080 FORMAT( 5(1X,I4,2X,A6,2X,F3.1,5X)) TRAC0139
9090 FORMAT(' INPUT RESTRICTION VIOLATED. TOTAL NUMBER OF COMPONENTS ATRAC0140
1ND HOUSES MUST NOT EXCEED 1000.'/ 1X,I7,' HAVE BEEN INPUT.'/ TRAC0141
2' PROGRAM WILL LIST FIRST 1000 THEN CHECK FOR VIOLATION OF GATE RETRAC0142
3STRICIONS'//) TRAC0143
9100 FORMAT(A6,F7.6) TRAC0144
9110 FORMAT(///' CASE NUMBER ',I5/' FORCING PARAMETER IS ',F8.6) TRAC0145
9115 FORMAT(//35X,'***** FAILED GATES FOR JTH MINIMAL CUTSET *****'// 'TRAC0146
1 CUTSET NO. OF' / ' NO. GATES', 50X,'GATE NAMES' // ) TRAC0147
9120 FORMAT (/I6,2X,I6,2X,I4(2X,A6) /(16X,I4(2X,A6) ) ) TRAC0148
9125 FORMAT (/I6,'*',1X,I6,2X,I4(2X,A6) / (16X,I4(2X,A6) ) ) TRAC0149
9130 FORMAT(//33X,'***** MINIMAL CUTSETS WITH 10 OR LFSS COMPONENTS **TRAC0150
1***'// TRAC0151
2' CUTSET CUTSET PROBABILITY NO. OF'// TRAC0152
3' RANK NO. OF FAILURE OCCURRENCES',35X,'COMPONENTS'//)TRAC0153

```



```

9140 FORMAT ('0',I4, 4X,I4,2X,F16.6,5X,I4,3X,10(2X,A6) ) TRAC0154
9150 FORMAT(/// ' ***** SUMMARY DATA FOR CASE NUMBER',I6,' *****'/// TRAC0155
1' NUMBER OF TRIALS COMPLETED      =',I7// TRAC0156
2' NUMBER OF SYSTEM FAILURES        =',I7// TRAC0157
3' PERCENT OF SYSTEM FAILURE        =',F7.2// TRAC0158
4' NO. ADDITIONAL SYSTEM FAILURES   =',I7,' (OVER 10 COMP. IN MINIMATRAC0159
5L CUTSET - NOT INCLUDED IN ABOVE FAILURE COUNT)'/ TRAC0160
6' NO. OF DISTINCT MINIMAL CUTSETS =',I7//) TRAC0161
9160 FORMAT(4X,' ESTIMATES BASED ON IMPORTANCE SAMPLING'/ TRAC0162
1' PROBABILITY OF SYSTEM FAILURE    =',E13.6/ TRAC0163
2' VARIANCE OF ABOVE ESTIMATE      =',E13.6/ TRAC0164
3' STANDARD ERROR OF ESTIMATE      =',F13.6//) TRAC0165
9170 FORMAT ('0',4X,'ESTIMATE BASED ON MINIMAL CUTSETS' / TRAC0166
1 ' PROBABILITY OF SYSTEM FAILURE =',E13.6 //) TRAC0167
9180 FORMAT ( ///' THE OBSERVED PERCENT OF SYSTEM FAILURES,' F7.2, TRAC0168
1' IS WITHIN ACCEPTABLE LIMITS OF 7.5 TO 18.0' ) TRAC0169
9190 FORMAT ( /// ' THE OBSERVED PERCENT OF SYSTEM FAILURES,' F7.2, TRAC0170
1', IS OUTSIDE ACCEPTABLE LIMITS OF 7.5 TO 18.0.'/ TRAC0171
2' A NEW FORCING PARAMETER SHOULD BE TRIED.'/ TRAC0172
3' A LARGER FORCING PARAMETER WILL TEND TO REDUCE THE PERCENTAGE,'/TRAC0173
4' A SMALLER ONE WILL TEND TO INCREASE IT.' ) TRAC0174
9200 FORMAT(///' TREE FAILS WITH HOUSES ALONE'/' PROGRAM WILL GO TO NEXTTRAC0175
1T PROBLEM'/// 49X,' TREE FAILED WITH FOLLOWING HOUSES'//) TRAC0176
9210 FORMAT(10(3X,A6) ) TRAC0177
9220 FORMAT(/// 60X,'FAILED GATES'/ 15(2X,A6) ) TRAC0178
9230 FORMAT( ///' THE MAXIMUM OF 500 DISTINCT MINIMAL CUTSETS HAS BEEN FTRAC0179
1OUND'/' PROGRAM WILL GO TO NEXT CASE FOLLOWING SUMMARY' ) TRAC0180
9240 FORMAT(///' INPUT RESTRICTION VIOLATED -- PROGRAM WILL GO TO NEXT PTRAC0181
1ROBLEM') TRAC0182
9300 FORMAT ('0NOTE.....ASTERISKS AFTER CUTSETS NO. IN TABLE BELOW INDITRAC0183
1CATE ONE OR MORE PRIORITY GATES IS INCLUDED IN FAILURE PATH.'// ) TRAC0184
9310 FORMAT ('+',I2X,'*' ) TRAC0185
C TRAC0186
C *****TRAC0187
10 READ(5,9010, END= 550 ) TITLE TRAC0188
WRITE(6,9000) TRAC0189
NEW = 0 TRAC0190
WRITE(6,9020) NEW, TITLE TRAC0191
NEW = 1 TRAC0192
C *****TRAC0193
C TRAC0194
C ***** INITIALIZE PROBLEM VARIABLES ***** TRAC0195
C TRAC0196
C *****TRAC0197
KASE = 0 TRAC0198
NC = 0 TRAC0199
NH = 0 TRAC0200
NCOMP = 0 TRAC0201
NG = 0 TRAC0202
NHFL = 0 TRAC0203
IFLAG = 0 TRAC0204

```

```

      READ(5,9025) NTR, IX, T
      IF(IX .EQ. 0) IX = 13579
C *****
C      INSURE THAT IX IS ODD
C *****
      IXX = IX/2
      IF ( (IX - 2*IXX) .EQ. 0 ) IX = IX + 1
C *****
C ***** READ COMPONENT CARDS AND CALCULATE PROB OF FAILURE FOR EACH *****
C *****
C      30 READ(5,9030) HFLAG,TEMPNM,TEMP1,TEMP2
      IF ( TEMPNM .EQ. LAST) GO TO 50
      NCOMP = NCOMP + 1
      IF ( NCOMP .GT. 1000 ) GO TO 30
      IF ( HFLAG .EQ. H ) GO TO 40
      NC = NC+1
      COMPNM(NC) = TEMPNM
      LAMDA(NC) = TEMP2
      P(NC) = 1.0 - EXP(-LAMDA(NC) * T /10.0**6)
      GO TO 30
C      40 NH = NH+1
      COMPNM(1001-NH) = TEMPNM
      LAMDA(1001-NH) = TEMP1
      GO TO 30
C *****
C      PLACE HOUSES IN COMPONENT ARRAYS IMMEDIATELY
C      FOLLOWING THE LAST COMPONENT
C *****
C      50 IF(NH .EQ. 0) GO TO 65
      DO 60 I = 1,NH
      TEMPNM = COMPNM(NC+I)
      COMPNM(NC+I) = COMPNM(1001-I)
      COMPNM(1001-I) = TEMPNM
      TEMP2 = LAMDA(NC+I)
      LAMDA(NC+I) = LAMDA(1001-I)
      60 LAMDA(1001-I) = TEMP2
C *****
C ***** WRITE OUT THE COMPONENT INPUT DATA *****
C *****
C      65 WRITE(6,9040) NTR,IX,T
      IF ( NCOMP .LE. 1000 ) GO TO 68
      WRITE(6,9090) NCOMP
      GO TO 70
      68 WRITE(6,9045) NC, NH, NCOMP
      70 WRITE (6,9020) NEW,TITLE

```

```

WRITE(6,9050)                                TRAC0256
N2 = 0                                         TRAC0257
71 N1 = N2 + 1                                TRAC0258
    IF ( (NC-N2) .GT. 260 ) GO TO 74          TRAC0259
72 N2 = NC                                     TRAC0260
73 WRITE(6,9060) (I, COMPNM(I), LAMDA(I), I = N1, N2) TRAC0261
    IF( NC .EQ. N2) GO TO 75                  TRAC0262
    WRITE(6,9020) NEW, TITLE                  TRAC0263
    WRITE(6,9050)                              TRAC0264
    GO TO 71                                   TRAC0265
74 N2 = N2 + 260                              TRAC0266
    GO TO 73                                   TRAC0267
C ----- TRAC0268
C ----- WRITE OUT HOUSE INPUT DATA ----- TRAC0269
C ----- TRAC0270
75 IF ( NH .EQ. 0 ) GO TO 80                  TRAC0271
    N2 = 0                                     TRAC0272
76 N1 = N2 + 1                                TRAC0273
    WRITE(6,9020) NEW, TITLE                  TRAC0274
    WRITE(6,9070)                              TRAC0275
    IF( (NH-N2) .GT. 370 ) GO TO 79           TRAC0276
77 N2 = NH                                     TRAC0277
78 WRITE(6,9080) ( I, COMPNM(NC+I), LAMDA(NC+I), I = N1, N2) TRAC0278
    IF (NH .EQ. N2) GO TO 80                  TRAC0279
    GO TO 76                                   TRAC0280
79 N2 = N2 + 370                              TRAC0281
    GO TO 78                                   TRAC0282
C ***** TRAC0283
C ***** CHECK FOR VIOLATION OF DIMENSION RESTRICTIONS ***** TRAC0284
C ***** READ GATE INPUT DATA ***** TRAC0285
C ***** TRAC0286
80 CALL GATES (NCOMP, NG, NRI)                TRAC0287
    IF(NCOMP .GT. 1000 .OR. NG .GT. 2000 .OR. NRI .GT. 100) GO TO 500 TRAC0288
    IF ( K2 .NE. 0 ) GO TO 500                TRAC0289
C ***** TRAC0290
C ***** TRAC0291
C READ FORCING PARAMETER CARDS AND INITIALIZE TRAC0292
C VARIABLES TO START A NEW CASE. IF IT IS AN TRAC0293
C END CARD GO TO START A NEW PROBLEM          TRAC0294
C                                               TRAC0295
C ***** TRAC0296
C ----- TRAC0297
C THE FOLLOWING FORTRAN STATEMENTS UP TO STATEMENT 95 WERE ADDED OR TRAC0298
C MODIFIED BY F.YEE                           TRAC0299
C                                               TRAC0300
    IPRINT = 0                                TRAC0301
    IP = 0                                     TRAC0302
89 KFLAG = 0                                  TRAC0303
    IPP = 0                                    TRAC0304
90 CONTINUE                                   TRAC0305
    IF(KFLAG .GT. 0) GO TO 92                 TRAC0306

```

```

      READ (5,9100) TEMPNM, R
      IF(TEMPNM .EQ. PRINT) IPRINT = 1
C-----
C   TEST FOR CHANGE OF CONTROL CARD.
C
C   IF(TEMPNM .NE. CHANGE) GO TO 94
C
C   NEW CONTROL CARD OPTION FOLLOWS.
C
      KFLAG = 1
      WRITE (6,9020) NEW, TITLE
      WRITE (6,6000)
6000  FORMAT ('OCHANGE OF CONTROL CARD.' )
      READ (5,9025) NTR, IX, T
      IF(IX .EQ. 0) IX = 13579
      IXX = IX/2
      IF((IX-2*IXX) .EQ. 0) IX = IX+1
      WRITE (6,9040) NTR, IX, T
      DO 91 II=1, NC
91     P(II) = 1.0-EXP(-LAMDA(II)*T/10.0**6)
92     CONTINUE
      IPP = IPP+1
      IF(IPP .GT. IP) GO TO 89
93     R = RSAVE(IPP)
      GO TO 95
94     CONTINUE
      IF ( TEMPNM .EQ. ENDPRB ) GO TO 10
      IP = IP+1
C-----
C   SAVE FORCING PARAMETERS FOR CHANGE OF CONTROL CARD OPTION.
C
C
      IF(IP .LE. 20) RSAVE(IP) = R
95     IF(R .EQ. 0.) GO TO 90
      KASE = KASE + 1
      IF(IPRINT .EQ. 0) GO TO 96
      WRITE(6,9020) NEW, TITLE
      WRITE(6,9110) KASE, R
      WRITE (6,9300)
      WRITE(6,9115)
96     CONTINUE
      LINE = 15
      KTEN = 0
      KTR = 0
      VAR = 0.0
      PRUH = 0.0
      NSFL = 0
      NSETS = 0
      DO 100 I = 1, 500
          NMSFT(I) = 0
          PSF(I) = 0.0
          IAST(I) = 0

```

```

TRAC0307
TRAC0308
TRAC0309
TRAC0310
TRAC0311
TRAC0312
TRAC0313
TRAC0314
TRAC0315
TRAC0316
TRAC0317
TRAC0318
TRAC0319
TRAC0320
TRAC0321
TRAC0322
TRAC0323
TRAC0324
TRAC0325
TRAC0326
TRAC0327
TRAC0328
TRAC0329
TRAC0330
TRAC0331
TRAC0332
TRAC0333
TRAC0334
TRAC0335
TRAC0336
TRAC0337
TRAC0338
TRAC0339
TRAC0340
TRAC0341
TRAC0342
TRAC0343
TRAC0344
TRAC0345
TRAC0346
TRAC0347
TRAC0348
TRAC0349
TRAC0350
TRAC0351
TRAC0352
TRAC0353
TRAC0354
TRAC0355
TRAC0356
TRAC0357

```

```

100 KOUNT(I) = 0                                TRAC0358
DO 110 I = 1, NC                                TRAC0359
    PSTAR(I) = P(I) ** R                        TRAC0360
110 WT(I) = P(I) / PSTAR(I)                    TRAC0361
C ***** TRAC0362
C ***** INITIALIZE TRIAL VARIABLES - START NEW TRIAL ***** TRAC0363
C ***** TRAC0364
C ***** TRAC0365
C ***** TRAC0366
120 IF (KTR .GE. NTR) GO TO 320                 TRAC0367
    KTR = KTR + 1                               TRAC0368
    NHFL = 0                                    TRAC0369
    NCFL = 0                                    TRAC0370
    PS = 1.0                                    TRAC0371
    TOP = .FALSE.                              TRAC0372
    DO 128 I=1, NRI                             TRAC0373
128 RICH(I) = .FALSE.                          TRAC0374
    DO 130 I = 1, NCOMP                         TRAC0375
130 COMPST(I) = .FALSE.                       TRAC0376
    DO 135 I = 1, NG                            TRAC0377
135 GATEST(I) = .FALSE.                       TRAC0378
    IF (NH .EQ. 0) GO TO 150                    TRAC0379
C ----- TRAC0380
C          FAIL HOUSES AS REQUIRED                TRAC0381
C ----- TRAC0382
    NC1 = NC + 1                               TRAC0383
    DO 140 I = NC1, NCOMP                       TRAC0384
    IF (LAMDA(I) .EQ. 0) GO TO 140              TRAC0385
    COMPST(I) = .TRUE.                         TRAC0386
    NHFL = NHFL + 1                             TRAC0387
    NEXT(NHFL) = I                             TRAC0388
    CALL TREE( TOP, I, INH )                    TRAC0389
    IF ( .NOT. TOP ) GO TO 140                  TRAC0390
    IF( INH .EQ. 0 ) GO TO 137                  TRAC0391
    COMPST(I) = .FALSE.                       TRAC0392
    NHFL = NHFL - 1                             TRAC0393
137 IFLAG = 1                                  TRAC0394
    GO TO 297                                   TRAC0395
140 CONTINUE                                   TRAC0396
C ***** TRAC0397
C ***** TRAC0398
C ***** IMPORTANCE SAMPLING OF COMPONENT FAILURE TIMES ***** TRAC0399
C ***** TRAC0400
C ***** TRAC0401
    NCFL = NHFL                                TRAC0402
150 DO 160 I = 1, NC                           TRAC0403
    CALL RANDU(IX,IY,U)                        TRAC0404
    IX = IY                                    TRAC0405
    IF ( U .GT. PSTAR(I) ) GO TO 160           TRAC0406
    NCFL = NCFL + 1                            TRAC0407
    NEXT(NCFL) = I                            TRAC0408

```

```

      TNEXT(NCFL) = -ALOG( 1.0 - WT(I) * U ) / LAMDA(I)*10.0**6      TRAC0409
160  CONTINUE                                                         TRAC0410
      IF ( NCFL .EQ. NHFL ) GO TO 120                                  TRAC0411
C *****TRAC0412
C                                                                 TRAC0413
C ***** FAIL COMPONENTS IN ORDER OF FAILURE TIMES *****      TRAC0414
C                                                                 TRAC0415
C *****TRAC0416
      NHFL1 = NHFL + 1                                                TRAC0417
      DO 180 NCUT = NHFL1, NCFL                                       TRAC0418
        DO 170 I = NCUT,NCFL                                          TRAC0419
          IF ( TNEXT(NCUT) .LE. TNEXT(I) ) GO TO 170                TRAC0420
          TEMP1 = TNEXT(I)                                            TRAC0421
          TNEXT(I) = TNEXT(NCUT)                                       TRAC0422
          TNEXT(NCUT) = TEMP1                                          TRAC0423
          ITEMP = NEXT(I)                                             TRAC0424
          NEXT(I) = NEXT(NCUT)                                         TRAC0425
          NEXT(NCUT) = ITEMP                                          TRAC0426
170    CONTINUE                                                      TRAC0427
C      COMPST(NEXT(NCUT)) = .TRUE.                                     TRAC0428
      NT = NEXT(NCUT)                                                 TRAC0429
      COMPST(NT) = .TRUE.                                            TRAC0430
      CALL TREE( TOP, NEXT(NCUT), INH )                              TRAC0431
      IF ( .NOT. TOP ) GO TO 180                                       TRAC0432
      IF ( INH .EQ. 0 ) GO TO 200                                       TRAC0433
      COMPST(NT) = .FALSE.                                           TRAC0434
      NCUT = NCUT - 1                                                TRAC0435
      GO TO 200                                                       TRAC0436
180  CONTINUE                                                         TRAC0437
      GO TO 120                                                       TRAC0438
C *****TRAC0439
C                                                                 TRAC0440
C ***** TREE IS FAILED - LOCATE MINIMAL CUTSET                  TRAC0441
C                                                                 TRAC0442
C *****TRAC0443
200  K = 0                                                            TRAC0444
      IF ( NCUT .NE. 1 ) GO TO 202                                     TRAC0445
      K = 1                                                            TRAC0446
      MSET(K) = NEXT(NCUT)                                           TRAC0447
      GO TO 255                                                       TRAC0448
202  NCUT2 = NCUT + 1                                                 TRAC0449
      NCUT1 = NCUT - 1                                                TRAC0450
      L = 0                                                            TRAC0451
      DO 220 I = 1, NCUT                                             TRAC0452
        J = NCUT2 - I                                                TRAC0453
C      COMPST( NEXT(J) ) = .FALSE.                                    TRAC0454
        NT = NEXT(J)                                                 TRAC0455
        COMPST(NT) = .FALSE.                                         TRAC0456
        CALL TRACE(TOP, NEXT(J), L)                                  TRAC0457
        IF ( TOP ) GO TO 210                                          TRAC0458
C      COMPST( NEXT(J) ) = .TRUE.                                    TRAC0459

```

```

      NT = NEXT(J)
      COMPST(NT) = .TRUE.
      K = K + 1
      IF ( K .LE. 10 ) GO TO 205
      KTEN = KTEN + 1
      GO TO 120
205  MSET(K) = NEXT(J)
      L = NEXT(J)
      GO TO 220
210  L = 0
220  CONTINUE
      IF ( .NOT. TOP ) CALL TRACE( TOP, 0, L )
      IF ( K .EQ. 1 ) GO TO 255
C *****
C      SORT MINIMAL CUTSET INTO COMP. NO. ORDER
C *****
230  K1 = K - 1
      DO 250 I = 1, K1
          L = I + 1
          DO 250 J = L, K
              IF( MSET(I) .LT. MSET(J)) GO TO 250
              ITEMP = MSET(I)
              MSET(I) = MSET(J)
              MSET(J) = ITEMP
250  CONTINUE
255  NSFL = NSFL + 1
      IF(NSETS .EQ. 0 ) GO TO 280
C *****
C      CHECK IF THIS MINIMAL CUTSET HAS BEEN FOUND PREVIOUSLY
C *****
      DO 270 I = 1, NSETS
          IF ( NMSET(I) .NE. K ) GO TO 270
          DO 260 J = 1, K
              IF(MSET(J).NE. MINSET(I,J)) GO TO 270
260  CONTINUE
          KOUNT(I) = KOUNT(I) + 1
          PROB = PROB + WTR(I)
          GO TO 120
270  CONTINUE
C *****
C      NEW MINIMAL CUTSET -- STORE IN MINSET
C      PERFORM PROBABILITY CALCULATIONS
C *****
280  NSETS = NSETS + 1
      KOUNT(NSETS) = 1
      WT1 = 1.0
      P1 = 1.0
      NMSFT(NSETS) = K
      DO 290 J = 1, K

```

```

TRAC0460
TRAC0461
TRAC0462
TRAC0463
TRAC0464
TRAC0465
TRAC0466
TRAC0467
TRAC0468
TRAC0469
TRAC0470
TRAC0471
TRAC0472
TRAC0473
TRAC0474
TRAC0475
TRAC0476
TRAC0477
TRAC0478
TRAC0479
TRAC0480
TRAC0481
TRAC0482
TRAC0483
TRAC0484
TRAC0485
TRAC0486
TRAC0487
TRAC0488
TRAC0489
TRAC0490
TRAC0491
TRAC0492
TRAC0493
TRAC0494
TRAC0495
TRAC0496
TRAC0497
TRAC0498
TRAC0499
TRAC0500
TRAC0501
TRAC0502
TRAC0503
TRAC0504
TRAC0505
TRAC0506
TRAC0507
TRAC0508
TRAC0509
TRAC0510

```

```

      MINSET(NSETS,J) = MSET(J)
      IF ( MSET(J) .GT. NC ) GO TO 290
      P1 = P1 * P(MSET(J))
      WT1 = WT1 * WT(MSET(J))
290  CONTINUE
      IF( NRI .EQ. 0 ) GO TO 296
C -----
C   WEIGHTS FOR RANDOM INHIBIT GATES THAT ARE SET TO TRUE
C -----
      DO 295 I = 1,NRI
C   IF ( .NOT. GATEST( IRI(I) ) ) GO TO 295
      NT = IRI(I)
      IF ( .NOT. GATEST(NT) ) GO TO 295
      P1 = P1 * PI(I)
      WT1 = WT1 * WT(I)
295  CONTINUE
296  PSF(NSETS) = P1
      PROB = PROB + WT1
      WTR(NSETS) = WT1
C *****
C ***** DETERMINE WHICH GATES ARE FAILED AND WRITE THEM OUT- NEW CUTSET
C *****
297  L = 0
      DO 300 I = 1, NG
      IF ( .NOT. GATEST(I) ) GO TO 300
      L = L+1
      FLGATE(L) = I
300  CONTINUE
      IF ( IFLAG .EQ. 1 ) GO TO 400
      LINE = LINE + 2 + L/15
      IF ( LINE .LE. 80 ) GO TO 305
      IF(IPRINT .EQ. 0) GO TO 305
      WRITE(6,9020) NEW, TITLE
      WRITE(6,9110) KASE, R
      WRITE (6,9300)
      WRITE(6,9115)
      LINE = 17 + L/15
305  DO 306 J=1,L
      NJT = FLGATE(J)
      IF(GTYPE(NJT) .NE. PG) GO TO 306
      IAST(NSETS) = 1
      IF(IPRINT .EQ. 0) GO TO 307
      WRITE (6,9125)NSETS,L,(GATENM(FLGATE(I)),I =1, L)
      GO TO 307
306  CONTINUE
      IF(IPRINT .EQ. 0) GO TO 307
      WRITE (6,9120)NSETS,L,(GATENM(FLGATE(I)),I =1, L)
307  CONTINUE
      IF ( NSETS .LT. 500 ) GO TO 120
      WRITE(6,9230)
C *****

```

```

TRAC0511
TRAC0512
TRAC0513
TRAC0514
TRAC0515
TRAC0516
TRAC0517
TRAC0518
TRAC0519
TRAC0520
TRAC0521
TRAC0522
TRAC0523
TRAC0524
TRAC0525
TRAC0526
TRAC0527
TRAC0528
TRAC0529
TRAC0530
TRAC0531
TRAC0532
TRAC0533
TRAC0534
TRAC0535
TRAC0536
TRAC0537
TRAC0538
TRAC0539
TRAC0540
TRAC0541
TRAC0542
TRAC0543
TRAC0544
TRAC0545
TRAC0546
TRAC0547
TRAC0548
TRAC0549
TRAC0550
TRAC0551
TRAC0552
TRAC0553
TRAC0554
TRAC0555
TRAC0556
TRAC0557
TRAC0558
TRAC0559
TRAC0560
TRAC0561

```



```

C
C ***** SUMMARY CALCULATIONS FOR END OF CASE
C
C *****
C
320 FKTR = KTR
    PS = 0.0
    IF ( NSFL .NE. 0 ) GO TO 325
    PROR = 0.0
    VAR = 0.0
    STDERR = 0.0
    PERCNT = 0.0
    GO TO 345
325 FNSFL = NSFL
    PROB = PROR / FKTR
    VAR = ( FKTR - FNSFL ) * PROR**2
    DO 330 I = 1, NSETS
        PS = PS + PSF(I)
        TEMP1 = KOUNT(I)
330  VAR = VAR + TEMP1 * (WTR(I) - PROB)**2
    VAR = VAR / (FKTR * (FKTR - 1.0) )
    STDERR = SQRT(VAR)
    PERCNT = 100.0 * FNSFL / FKTR
C.....
C REMOVE HOUSES (IF ANY),FROM THE MINIMAL CUTSETS FOR PRINTING.
C.....
    DO 332 I=1,NSETS
        K = NMSET(I)
        DO 331 J=1,K
            IF(MINSET(I,J) .LE. NC) GO TO 331
            NMSET(I) = J-1
        GO TO 332
331  CONTINUE
332  CONTINUE
C *****
C ***** WRITE OUT MINIMAL CUTSETS AND CASE SUMMARY
C *****
C *****
    LINE = 16
    DO 340 I = 1, NSETS
        LINE = LINE + 2
        IF( LINE .GT. 18 .AND. LINE .LE. 80 ) GO TO 333
        WRITE(6,9020) NEW, TITLE
        WRITE(6,9110) KASE, R
        WRITE (6,9300)
        WRITE(6,9130)
        LINE = 18
333  CONTINUE
C.....
C PRINT IN ORDER OF LARGEST PROBABILITIES.
C.....

```

```

TRAC0562
TRAC0563
TRAC0564
TRAC0565
TRAC0566
TRAC0567
TRAC0568
TRAC0569
TRAC0570
TRAC0571
TRAC0572
TRAC0573
TRAC0574
TRAC0575
TRAC0576
TRAC0577
TRAC0578
TRAC0579
TRAC0580
TRAC0581
TRAC0582
TRAC0583
TRAC0584
TRAC0585
TRAC0586
TRAC0587
TRAC0588
TRAC0589
TRAC0590
TRAC0591
TRAC0592
TRAC0593
TRAC0594
TRAC0595
TRAC0596
TRAC0597
TRAC0598
TRAC0599
TRAC0600
TRAC0601
TRAC0602
TRAC0603
TRAC0604
TRAC0605
TRAC0606
TRAC0607
TRAC0608
TRAC0609
TRAC0610
TRAC0611
TRAC0612

```

```

333 KK = 1                                     TRAC0613
    DO 338 JJ=1,NSETS                         TRAC0614
334 IF (PSF(KK) .GT. 0.) GO TO 335             TRAC0615
    KK = KK+1                                 TRAC0616
    IF (KK - NSETS) 334,334,340               TRAC0617
335 IF (PSF(KK)-PSF(JJ)) 336,337,338          TRAC0618
336 KK = JJ                                   TRAC0619
    GO TO 338                                 TRAC0620
337 IF (KOUNT(KK)-KOUNT(JJ)) 336,338,338     TRAC0621
338 CONTINUE                                 TRAC0622
    K = NMSET(KK)                             TRAC0623
    WRITE (6,9140) I, KK, PSF(KK), KOUNT(KK), TRAC0624
    1 (COMPNM( MINSET(KK,J) ), J=1,K)         TRAC0625
    PSF(KK) = 0.                             TRAC0626
    IF (IAST(KK) .NE. 0) WRITE (6,9310)       TRAC0627
340 CONTINUE                                 TRAC0628
    345 WRITE(6,9020) NEW, TITLE              TRAC0629
    WRITE(6,9110) KASE, R                     TRAC0630
    WRITE(6,9150) KASE, KTR, NSFL, PERCNT, KTR, NSETS TRAC0631
    WRITE(6,9170) PS                           TRAC0632
    WRITE(6,9160) PROR, VAR, STDERR           TRAC0633
    IF ( PERCNT .GT. 18.0 .OR. PERCNT .LT. 7.5 ) GO TO 350 TRAC0634
C WRITE(6,9180) PERCNT                       TRAC0635
C GO TO 90                                    TRAC0636
C 350 WRITE(6,9190) PERCNT                   TRAC0637
C GO TO 90                                    TRAC0638
C *****TRAC0639
C ***** TREE FAILS WITH HOUSES ALONE - WRITE DATA, GO TO NEXT PROBLEM **TRAC0640
C *****TRAC0641
400 WRITE(6,9200)                             TRAC0642
    WRITE(6,9210) (COMPNM(NEXT(I)), I = 1, NHFL ) TRAC0643
    WRITE(6,9220) (GATENM(FLGATE(I)), I = 1, L)   TRAC0644
C *****TRAC0645
C *****TRAC0646
C ***** INPUT RESTRICTION VIOLATED, READ OUT R CARDS TO NEXT PROBLEM TRAC0647
C *****TRAC0648
C *****TRAC0649
500 WRITE(6,9240)                             TRAC0650
510 READ(5,9100) TEMPNM                       TRAC0651
    IF ( TEMPNM .EQ. ENDPRR ) GO TO 10          TRAC0652
    GO TO 510                                   TRAC0653
550 STOP                                       TRAC0654
END                                             TRAC0655
C *****TRAC0657
C *****TRAC0658
C SUBROUTINE RANDU                             TRAC0659
C *****TRAC0660
C PURPOSE                                     TRAC0661
C COMPUTES UNIFORMLY DISTRIBUTED RANDOM REAL NUMBERS BETWEEN TRAC0662
C 0 AND 1.0 AND RANDOM INTEGERS BETWEEN ZERO AND TRAC0663

```

```

C      2**31. EACH ENTRY USES AS INPUT AN INTEGER RANDOM NUMBER      TRAC0664
C      AND PRODUCES A NEW INTEGER AND REAL RANDOM NUMBER.          TRAC0665
C                                                                    TRAC0666
C      USAGE                                                         TRAC0667
C      CALL RANDU(IX,IY,YFL)                                         TRAC0668
C                                                                    TRAC0669
C      DESCRIPTION OF PARAMETERS                                     TRAC0670
C      IX - FOR THE FIRST ENTRY THIS MUST CONTAIN ANY ODD INTEGER   TRAC0671
C            NUMBER WITH NINE OR LESS DIGITS. AFTER THE FIRST ENTRY, TRAC0672
C            IX SHOULD BE THE PREVIOUS VALUE OF IY COMPUTED BY THIS   TRAC0673
C            SUBROUTINE.                                             TRAC0674
C      IY - A RESULTANT INTEGER RANDOM NUMBER REQUIRED FOR THE NEXT   TRAC0675
C            ENTRY TO THIS SUBROUTINE. THE RANGE OF THIS NUMBER IS   TRAC0676
C            BETWEEN ZERO AND 2**31                                  TRAC0677
C      YFL- THE RESULTANT UNIFORMLY DISTRIBUTED, FLOATING POINT,    TRAC0678
C            RANDOM NUMBER IN THE RANGE 0 TO 1.0                     TRAC0679
C                                                                    TRAC0680
C      REMARKS                                                         TRAC0681
C      THIS SUBROUTINE IS SPECIFIC TO SYSTEM/360                     TRAC0682
C      THIS SUBROUTINE WILL PRODUCE 2**29 TERMS                      TRAC0683
C      BEFORE REPEATING                                             TRAC0684
C                                                                    TRAC0685
C      SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                 TRAC0686
C      NONE                                                         TRAC0687
C                                                                    TRAC0688
C      METHOD                                                         TRAC0689
C      POWER RESIDUE METHOD DISCUSSED IN IBM MANUAL C20-8011,        TRAC0690
C      RANDOM NUMBER GENERATION AND TESTING                         TRAC0691
C                                                                    TRAC0692
C      .....TRAC0693
C      SUBROUTINE RANDU(IX,IY,YFL)                                   TRAC0694
C      IY = IX * 65539                                               TRAC0695
C      IF(IY) 5, 6, 6                                               TRAC0696
C      5 IY = IY + 2147483647 + 1                                     TRAC0697
C      6 YFL = IY                                                    TRAC0698
C      YFL = YFL * .4656613E-9                                       TRAC0699
C      RETURN                                                         TRAC0700
C      END                                                           TRAC0701
C*****TRAC0702
C                                                                    TRAC0703
C      THIS MULTI-ENTRY SUBROUTINE WILL                             TRAC0704
C      (GATES) 1. READ GATE CARDS (EACH GROUP FOLLOWED BY          TRAC0705
C      'LAST' CARD), SET UP DICTIONARIES                            TRAC0706
C      (TRFE) 2. CALC. P* AND WGT. FOR R.I. GATES WHEN 'R'         TRAC0707
C      CHANGES, CHECK R.I. GATES AND THEIR PROBABILITY,          TRAC0708
C      TRACE A FAILED COMPONENT THRU ALL POSSIBLE                  TRAC0709
C      PATHS TO THE TOP                                             TRAC0710
C      (TRACE) 3. CHANGE STATUS OF 1 OR 2 COMPONENTS AND TRACE     TRAC0711
C      SAME THRU ALL POSSIBLE PATHS TO THE TOP                     TRAC0712
C                                                                    TRAC0713
C                                                                    TRAC0714

```

```

C*****TRAC0715
C      COMPNM = COMPONENT NAMES                                TRAC0716
C      COMPST = COMPONENT STATUS                                TRAC0717
C      COUT   = SUBSCRIPTS OF GATES FED BY A COMPONENT        TRAC0718
C      DATA  = WORKING AREA                                    TRAC0719
C      FALSE  = '0' - SUCCESS FLAG                             TRAC0720
C      GATENM = GATE NAMES                                       TRAC0721
C      GATEST = GATE STATUS                                       TRAC0722
C      GIN(.,1)=SUBSCRIPTS OF GATES FED TO A GATE              TRAC0723
C      (.,2)=SUBSCRIPTS OF COMPONENTS FED TO A GATE            TRAC0724
C      GOUT   = SUBSCRIPTS OF GATES FED BY A GATE              TRAC0725
C      GTYPE  = GATE TYPE - OR, AND, R.I.                      TRAC0726
C      ICOMP  = SUBSCRIPT OF COMPONENT THAT DID NOT FAIL        TRAC0727
C      ICOMPT = SUBSCRIPT OF COMPONENT THAT FAILED              TRAC0728
C      INH    = FLAG, ON MEANS TREE WAS TOPPED BECAUSE OF RI GATES TRAC0729
C      IPRINT = PRINT FLAG, ON=PRINT GATE CARD TABLE          TRAC0730
C      IRI    = SUBSCRIPTS OF R.I. GATES (POSITION IN GATENM ARRAY) TRAC0731
C      IRIF   = FLAG, ON MEANS TREE IS BEING TRACED BECAUSE OF TRAC0732
C              CHANGE OF STATUS OF RI GATE                     TRAC0733
C      ITRACE = 0, ENTRY TREE...NOT 0, ENTRY TRACE              TRAC0734
C      IX     = ODD INTEGER, 9 OR LESS DIGITS, INPUT TO OBTAIN YFL TRAC0735
C      IY     = ODD INTEGER RETURNED FROM RANDU                 TRAC0736
C      K      = FLAG, NO. OF GATE NAMES READ IN SO FAR          TRAC0737
C      K1     = FLAG, NO. OF R.I. GATES READ IN SO FAR          TRAC0738
C      K2     = FLAG, NO. OF COMP. NAME INPUT IN ERROR          TRAC0739
C      K3     = FLAG, NO. OF GATES WITH INCORRECT TYPE CODE     TRAC0740
C      K4     = COUNTER OF RI GATES BEING CHECKED               TRAC0741
C      M      = MAXIMUM NUMBER OF COMPONENTS                    TRAC0742
C      N      = MAXIMUM NUMBER OF GATES                         TRAC0743
C      NCOMP  = NUMBER OF COMPONENTS INPUT                      TRAC0744
C      NG     = NUMBER OF GATES INPUT                           TRAC0745
C      NJ(ODD)= SUBSCRIPT OF GATE OR COMP.                      TRAC0746
C      (EVEN)= RELATIVE LATERAL POSITION OF GATE/COMP.          TRAC0747
C              (NJ IS USED TO KEEP TRACK OF TRACING)           TRAC0748
C      NRI    = NUMBER OF R.I. GATES INPUT                      TRAC0749
C      N1     = MAXIMUM NUMBER OF GATES FED BY A GATE OR COMP. TRAC0750
C      N2     = MAXIMUM NUMBER OF GATES OR COMP. FED TO A GATE TRAC0751
C      N3     = MAXIMUM NUMBER OF R.I. GATES                    TRAC0752
C      PI     = INPUT PROBABILITIES ASSOC. WITH R.I. GATES     TRAC0753
C      PIN    = PROBABILITY OF R.I. GATE INPUT                  TRAC0754
C      PSTAR  = PROBABILITY USED TO SET FAIL CHANCE OF R.I. GATE TRAC0755
C      R      = EXPONENT 'R'                                    TRAC0756
C      RICH   = PROBABILITY OF FAILURE FLAGS FOR R.I. GATES.    TRAC0757
C              IF RANDOM VARIABLE IS THAN PROBABILITY,FLAG = .TRUE. TRAC0758
C              OTHERWISE,FLAG = .FALSE.                         TRAC0759
C      RK     = PREVIOUS 'R' VALUE                               TRAC0760
C      TOP    = STATUS OF TREE                                   TRAC0761
C      TRUE   = '1' - FAIL FLAG                                  TRAC0762
C      TYPE   = GATE TYPE OF GATE JUST READ IN                 TRAC0763
C      WTI    = WEIGHTS                                          TRAC0764
C      YFL    = UNIFORM RANDOM NUMBER                           TRAC0765

```

```

C*****TRAC0766
C
  SUBROUTINE TREE (TOP, ICOMPT, INH )
  IMPLICIT REAL*8 (A-H,O-Z)
  COMMON /MTREE/ TITLE(9), COMPNM(1000), GATENM(2000), PI(100),
1    WTI(10), R, IRI(100), IX, K2, GTYPE(2000), COMPST(1000),
2    GATEST(2000), RICH(100)
  REAL*4 PI, WTI, R
  LOGICAL*1 COMPST, GATEST, TOP, RICH
  REAL * 8 LAST, TITLE
  REAL * 4 PSTAR(100), PIN, YFL, RK
  INTEGER * 4 GOUT(2000,4), GIN(2000,8,2), COUT(1000,4)
  INTEGER * 2 GTYPE, TYPE, OR, AND, RI, PG
  DIMENSION NJ(200), DATA(16)
  DATA LAST, PRINT, OR, AND, RI / 'LAST', 'PRINT', 'R', 'A', 'I' /
  DATA PG/ 'P' /, STAR/ '*****'/
  DATA M, N, N1, N2, N3 /1000, 2000, 4, 8, 100/

C
  ITRACE = 0
  INH = 0
  IRIF = 0
  IF( NRI .EQ. 0 ) GO TO 40
  IF( R .EQ. RK ) GO TO 30

C-----
C    NEW R VALUE, CALC. P* AND WEIGHT FOR ALL R.I. GATES
C-----
  DO 20 I=1,NRI
    PSTAR(I) = PI(I) ** R
  20 WTI(I) = PI(I)/ PSTAR(I)
  RK = R

C-----
C    SET PROBABILITY OF FAILURE AT EACH R.I. GATE
C-----
  30 K4 = 1
  31 CALL RANDU ( IX, IY, YFL )
  IX = IY
  NJT = IRI(K4)
  IF( YFL .LE. PSTAR(K4) ) GO TO 32
  IF( .NOT. RICH(K4) ) GO TO 35
  RICH(K4) = .FALSE.
  IF( .NOT. GATEST(NJT) ) GO TO 35

C-----
C    CHANGE RI GATE STATUS FROM TRUE TO FALSE AND TRACE TREE
C-----

```

GATEST(NJT) = .FALSE.	TRAC0817
IRIF = 1	TRAC0818
NJ(1) = GOUT(NJT,1)	TRAC0819
GO TO 305.	TRAC0820
C	TRAC0821
32 IF(RICH(K4)) GO TO 35	TRAC0822
RICH(K4) = .TRUE.	TRAC0823
IF(GATEST(NJT)) GO TO 35	TRAC0824
NJT1 = GIN(NJT,1,1)	TRAC0825
C IF(NJT1 .EQ. 0) NJT1 = GIN(NJT,1,2)	TRAC0826
IF(NJT1 .EQ. 0) GO TO 34	TRAC0827
IF(.NOT. GATEST(NJT1)) GO TO 35	TRAC0828
C-----	TRAC0829
C	TRAC0830
C CHANGE RI GATE FROM FALSE TO TRUE AND TRACE TREE	TRAC0831
C	TRAC0832
C-----	TRAC0833
33 CONTINUE	TRAC0834
GATEST(NJT) = .TRUE.	TRAC0835
IRIF = 1	TRAC0836
NJ(1) = GOUT(NJT,1)	TRAC0837
GO TO 45	TRAC0838
34 NJT1 = GIN(NJT,1,2)	TRAC0839
IF(NJT1 .EQ. ICOMPT) GO TO 35	TRAC0840
IF(COMPT(NJT1)) GO TO 33	TRAC0841
C	TRAC0842
35 IF(.NOT. TOP) GO TO 37	TRAC0843
INH = 1	TRAC0844
RETURN	TRAC0845
37 IF(K4 .GE. NRI) GO TO 40	TRAC0846
K4 = K4 + 1	TRAC0847
GO TO 31	TRAC0848
C	TRAC0849
40 NJ(1) = CRUT (ICOMPT,1)	TRAC0850
IRIF = 0	TRAC0851
45 J1 = 1	TRAC0852
J2 = 1	TRAC0853
50 NJT = NJ(J1)	TRAC0854
IF(GATEST(NJT)) GO TO 110	TRAC0855
IF(GTYPE(NJT) .EQ. AND) GO TO 70	TRAC0856
IF(GTYPE(NJT) .EQ. RI) GO TO 55	TRAC0857
IF(GTYPE(NJT) .EQ. PG) GO TO 66	TRAC0858
C-----	TRAC0859
C	TRAC0860
C 'OR' GATE.....FAILED	TRAC0861
C	TRAC0862
C-----	TRAC0863
C GATEST(NJ(J1)) = .TRUE.	TRAC0864
NJT = NJ(J1)	TRAC0865
GATEST(NJT) = .TRUE.	TRAC0866
GO TO 110	TRAC0867

```

C-----TRAC0868
C                                     TRAC0869
C      'RI' GATE                      TRAC0870
C                                     TRAC0871
C-----TRAC0872
55 DO 60 I=1,NRI                      TRAC0873
   IF( NJ(J1) .EQ. IRI(I)) GO TO 65   TRAC0874
60 CONTINUE                          TRAC0875
C 65 GATEST(NJ(J1)) = RICH(I)         TRAC0876
65 NJT = NJ(J1)                      TRAC0877
   GATEST(NJT) = RICH(I)              TRAC0878
C   IF(RICH(I) ) GOT 0 120            TRAC0879
   IF(RICH(I) ) GO TO 110             TRAC0880
   GO TO 130                          TRAC0881
C-----TRAC0882
C                                     TRAC0883
C      'PRIORITY' GATE               TRAC0884
C                                     TRAC0885
C-----TRAC0886
66 IF( ITRACE .NE. 0 ) GO TO 70       TRAC0887
   IF( GIN(NJ(J1),1,1) .EQ. 0 ) GO TO 69 TRAC0888
C-----TRAC0889
C                                     TRAC0890
C      PRIORITY ORDER = GATE, GATE OR GATE, COMPONENT TRAC0891
C                                     TRAC0892
C-----TRAC0893
   NJT = GIN(NJ(J1),1,1)              TRAC0894
   IF( J1 .EQ. 1 ) GO TO 67           TRAC0895
   IF( NJT .EQ. NJ(J1-2) ) GO TO 130  TRAC0896
   GO TO 68                           TRAC0897
67 IF( NJT .EQ. ICOMP ) GO TO 130     TRAC0898
68 IF( .NOT. GATEST(NJT) ) GO TO 130  TRAC0899
   NJT = NJ(J1)                      TRAC0900
   GATEST(NJT) = .TRUE.              TRAC0901
   GO TO 110                          TRAC0902
C-----TRAC0903
C                                     TRAC0904
C      PRIORITY ORDER = COMPONENT, GATE OR COMPONENT, COMPONENT TRAC0905
C                                     TRAC0906
C-----TRAC0907
69 NJT = GIN(NJ(J1),1,2)              TRAC0908
   IF( NJT .EQ. ICOMP ) GO TO 130     TRAC0909
   IF( .NOT. COMPST(NJT) ) GO TO 130  TRAC0910
   NJT = NJ(J1)                      TRAC0911
   GATEST(NJT) = .TRUE.              TRAC0912
   GO TO 110                          TRAC0913
C-----TRAC0914
C                                     TRAC0915
C      'AND' GATE.....CHECK ALL GATES AND COMPONENTS FED TO GATE TRAC0916
C                                     TRAC0917
C-----TRAC0918

```

```

70 J = NJ(J1)
-----
C
C
C      .....CHECK GATES.....
C
C
C      DO 80 I=1,N2
C      IF(GIN(J,I,1) .EQ. 0) GO TO 90
C      IF(GATEST(GIN(J,I,1)) ) GO TO 130
C      NJT = GIN(J,I,1)
C      IF( .NOT. GATEST(NJT) ) GO TO 130
C      80 CONTINUE
-----
C
C
C      .....CHECK COMPONENTS.....
C
C
C      90 DO 100 I=1,N2
C      IF(GIN(J,I,2) .EQ. 0) GO TO 105
C      IF( .NOT. COMPST(GIN(J,I,2)) ) GO TO 130
C      NJT = GIN(J,I,2)
C      IF( .NOT. COMPST(NJT) ) GO TO 130
C      100 CONTINUE
C      105 GATEST(J) = .TRUE.
-----
C
C
C      GATE FED TO, FAILED
C
C
C
C
C      110 NJ(J1+1) = J2
C      IF(GOUT(NJ(J1),1) .NE. 0 ) GO TO 120
-----
C
C
C      TREE TOPPED.....SET 'TOP' TO TRUE (FAILED)
C
C
C
C      TOP = .TRUE.
C      GO TO 130
-----
C
C
C      120 J1 = J1 + 2
C      NJ(J1) = GOUT(NJ(J1-2),1)
C      J2 = 1
C      GO TO 50
-----
C
C
C      GATE FED DID NOT FAIL.....AT LEAST 1 GATE OR COMPONENT
C      DID NOT FAIL
C
C
C
C
C      130 J2 = J2 + 1
C      IF(J1 .EQ. 1) GO TO 145

```

```

TRAC0919
TRAC0920
TRAC0921
TRAC0922
TRAC0923
TRAC0924
TRAC0925
TRAC0926
TRAC0927
TRAC0928
TRAC0929
TRAC0930
TRAC0931
TRAC0932
TRAC0933
TRAC0934
TRAC0935
TRAC0936
TRAC0937
TRAC0938
TRAC0939
TRAC0940
TRAC0941
TRAC0942
TRAC0943
TRAC0944
TRAC0945
TRAC0946
TRAC0947
TRAC0948
TRAC0949
TRAC0950
TRAC0951
TRAC0952
TRAC0953
TRAC0954
TRAC0955
TRAC0956
TRAC0957
TRAC0958
TRAC0959
TRAC0960
TRAC0961
TRAC0962
TRAC0963
TRAC0964
TRAC0965
TRAC0966
TRAC0967
TRAC0968
TRAC0969

```



```

135 IF(J2 .GT. N1) GO TO 140                                TRAC0970
    IF (J1 .EQ. 1 ) GO TO 145                                TRAC0971
    IF(GOUT(NJ(J1-2),J2) .EQ. 0) GO TO 140                  TRAC0972
    NJ(J1) = GOUT(NJ(J1-2),J2)                               TRAC0973
    GO TO 50                                                  TRAC0974
140 J2 = NJ(J1-1) + 1                                         TRAC0975
    J1 = J1 - 2                                               TRAC0976
    IF(J1 .GT. 1)      GO TO 135                             TRAC0977
    J1 = 1                                                     TRAC0978
145 IF(J2 .GT. N1)      GO TO 150                             TRAC0979
    IF( IRIF .EQ. 0 ) GO TO 146                             TRAC0980
    NJ(J1) = GOUT(IRI(K4),J2)                                TRAC0981
    GO TO 147                                                  TRAC0982
146 NJ(J1) = COUT(ICOMPT,J2)                                  TRAC0983
147 IF( NJ(J1) .EQ. 0 ) GO TO 150                             TRAC0984
    GO TO 50                                                  TRAC0985
C                                                                    TRAC0986
150 IF( ITRACE .NE. 0 ) GO TO 300                             TRAC0987
    IF( IRIF .NE. 0 ) GO TO 35                                TRAC0988
    RETURN                                                    TRAC0989
C                                                                    TRAC0990
C*****TRAC0991
C*****TRAC0992
C                                                                    TRAC0993
    ENTRY TRACE ( TOP, ICOMP, ICOMPT )                        TRAC0994
C                                                                    TRAC0995
C*****TRAC0996
C*****TRAC0997
    ITRACE = 1                                                TRAC0998
    IRIF = 0                                                  TRAC0999
    IF( ICOMPT .NE. 0 ) GO TO 40                               TRAC1000
    300 IF( ICOMP .EQ. 0 ) GO TO 450                           TRAC1001
C-----TRAC1002
C                                                                    TRAC1003
C                                                                    TRAC1004
    COMPONENT SET TO FALSE, THAT IS SUCCESS                  TRAC1005
C                                                                    TRAC1006
    TRACE TREE UPWARDS. GATES MAY BE CHANGED TO SUCCESS.    TRAC1007
C                                                                    TRAC1008
C-----TRAC1008
    NJ( 1) = COUT (ICOMP,1)                                   TRAC1009
    305 J1 = 1                                                 TRAC1010
    J2 = 1                                                     TRAC1011
    310 IF( GTYPE(NJ(J1)) .EQ. OR ) GO TO 320                 TRAC1012
C-----TRAC1013
C                                                                    TRAC1014
C                                                                    TRAC1015
    'AND' OR 'R1' OR 'PRIORITY' GATE.....NOT FAILED        TRAC1016
C-----TRAC1017
C                                                                    TRAC1018
    GATEST(NJ(J1)) = .FALSE.                                  TRAC1019
    NJT = NJ(J1)                                              TRAC1020
    GATEST(NJT) = .FALSE.

```

```

      GO TO 370
C-----TRAC1021
C-----TRAC1022
C-----TRAC1023
C      'OR' GATE.....CHECK ALL GATES AND COMPONENTS FED TO GATE
C-----TRAC1024
C-----TRAC1025
C-----TRAC1026
      320 J = NJ(J1)
C-----TRAC1027
C-----TRAC1028
C-----TRAC1029
C      .....CHECK GATES.....
C-----TRAC1030
C-----TRAC1031
C-----TRAC1032
      DO 330 I=1,N2
C-----TRAC1033
      IF(GIN(J,I,1) .EQ. 0 ) GO TO 340
C-----TRAC1034
      IF(GATEST(GIN(J,I,1)) )GO TO 400
C-----TRAC1035
      NJT = GIN(J,I,1)
C-----TRAC1036
      IF( GATEST(NJT) ) GO TO 400
C-----TRAC1037
      330 CONTINUE
C-----TRAC1038
C-----TRAC1039
C-----TRAC1040
C      .....CHECK COMPONENTS.....
C-----TRAC1041
C-----TRAC1042
C-----TRAC1043
      340 DO 350 I=1,N2
C-----TRAC1044
      IF(GIN(J,I,2) .EQ. 0 ) GO TO 360
C-----TRAC1045
      IF(COMPST(GIN(J,I,2)) ) GO TO 400
C-----TRAC1046
      NJT = GIN(J,I,2)
C-----TRAC1047
      IF( COMPST(NJT) ) GO TO 400
C-----TRAC1048
      350 CONTINUE
C-----TRAC1049
      360 GATEST(J) =.FALSE.
C-----TRAC1050
C-----TRAC1051
C-----TRAC1052
C      GATE FED TO, NOT FAILED
C-----TRAC1053
C-----TRAC1054
C-----TRAC1055
      370 NJ(J1+1) = J2
C-----TRAC1056
      IF(GOUT(NJ(J1),1) .NE. 0 ) GO TO 380
C-----TRAC1057
C-----TRAC1058
C-----TRAC1059
C      TREE TOPPED.....SET TOP TO FALSE (NOT FAILED)
C-----TRAC1060
C-----TRAC1061
C-----TRAC1062
      TOP =.FALSE.
C-----TRAC1063
      GO TO 400
C-----TRAC1064
C-----TRAC1065
C-----TRAC1066
      380 J1 = J1 + 2
C-----TRAC1067
      NJ(J1) = GOUT(NJ(J1-2),1)
C-----TRAC1068
      J2 = 1
C-----TRAC1069
      GO TO 310
C-----TRAC1070
C-----TRAC1071

```

```

C      GATE FED TO REMAINED FAILED.....AT LEAST 1 GATE OR COMPONENT TRAC1072
C      FAILED TRAC1073
C      TRAC1074
C-----TRAC1075
400 J2 = J2 + 1 TRAC1076
      IF(J1 .EQ. 1) GO TO 425 TRAC1077
410 IF(J2 .GT. N1) GO TO 420 TRAC1078
      IF( J1 .EQ. 1 ) GO TO 425 TRAC1079
      IF(GOUT(NJ(J1-2),J2) .EQ. 0 ) GO TO 420 TRAC1080
      NJ(J1) = GOUT(NJ(J1-2),J2) TRAC1081
      GO TO 310 TRAC1082
420 J2 = NJ(J1-1) + 1 TRAC1083
      J1 = J1 - 2 TRAC1084
      IF(J1 .GT. 1) GO TO 410 TRAC1085
      J1 = 1 TRAC1086
425 IF(J2 .GT. N1) GO TO 450 TRAC1087
      IF(IRIF .EQ. 0) GO TO 426 TRAC1088
      NJ(J1) = GOUT(IRI(K4),J2) TRAC1089
      GO TO 427 TRAC1090
426 NJ(J1) = COUT(ICOMPJ,J2) TRAC1091
427 IF(NJ(J1) .EQ. 0) GO TO 450 TRAC1092
      GO TO 310 TRAC1093
450 IF( ITRACE .EQ. 0 ) GO TO 35 TRAC1094
      RETURN TRAC1095
C      TRAC1096
C*****TRAC1097
C*****TRAC1098
C      TRAC1099
      ENTRY GATES (NCOMP, NG, NRI) TRAC1100
C      TRAC1101
C*****TRAC1102
C*****TRAC1103
C      TRAC1104
C-----TRAC1105
C      TRAC1106
C      INITIALIZE VARIABLES TRAC1107
C      TRAC1108
C-----TRAC1109
      IPRINT = 0 TRAC1110
      K = 0 TRAC1111
      K1 = 0 TRAC1112
      K2 = 0 TRAC1113
      K3 = 0 TRAC1114
      RK = 0.0 TRAC1115
      DO 605 I=1,N1 TRAC1116
      DO 600 J=1,M TRAC1117
600 COUT (J,I) = 0 TRAC1118
      DO 604 J=1,N TRAC1119
604 GOUT (J,I) = 0 TRAC1120
605 CONTINUE TRAC1121
      DO 610 I=1,N2 TRAC1122

```

```

DO 610 J=1,N                                TRAC1123
GIN (J,I,1) = 0                             TRAC1124
610 GIN (J,I,2) = 0                         TRAC1125
C                                           TRAC1126
C-----TRAC1127
C                                           TRAC1128
C      READ GATE CARDS                     TRAC1129
C                                           TRAC1130
C-----TRAC1131
650 READ(5, 900, END=895) TYPE, GATE,IP,NG,NC,PIN,(DATA(I),I=1,8) TRAC1132
900 FORMAT( A1, T1,A6,      3I1, F7.6, 8(1X A6)) TRAC1133
      IF( GATE .NE. PRINT ) GO TO 655        TRAC1134
      IPRINT = 1                            TRAC1135
      GO TO 650                             TRAC1136
655 IF( GATE .EQ. LAST ) GO TO 750          TRAC1137
      IF( NG + NC .GT. 8 ) READ(5, 901, END= 895) (DATA(I),I=9,16) TRAC1138
901 FORMAT(16X 8(1X A6))                   TRAC1139
      IF( K .EQ. 0 ) GO TO 665              TRAC1140
C-----TRAC1141
C                                           TRAC1142
C      CHECK IF GATE NAME HAS BEEN REGISTERED IN GATE TRAC1143
C      NAME ARRAY. IF NO, STORE NAME AND TYPE. TRAC1144
C                                           TRAC1145
C-----TRAC1146
      DO 660 I=1,K                          TRAC1147
      IF( GATE .EQ. GATENM(I)) GO TO 670    TRAC1148
660 CONTINUE                               TRAC1149
665 K = K + 1                              TRAC1150
      IF( K .LE. N ) GO TO 668              TRAC1151
C                                           TRAC1152
C=====TRAC1153
C.....ERROR - MAXIMUM GATE INPUT EXCEEDED.....TRAC1154
666 WRITE(6, 904) DATA(J1), GATE         TRAC1155
904 FORMAT(/ ' MAXIMUM GATE INPUT EXCEEDED, GATE 'A6, ' CANNOT BE ADDETRAC1156
      1D TO TABL.....GATE CARD NAME IS ' A6) TRAC1157
      NG = K                               TRAC1158
      NRI = K1                             TRAC1159
667 READ( 5,900, END=895 ) TYPE, GATE     TRAC1160
      IF( GATE .EQ. LAST ) GO TO 890       TRAC1161
      GO TO 667                            TRAC1162
C=====TRAC1163
C                                           TRAC1164
C      668 GATENM(K) = GATE                TRAC1165
C      I = K                              TRAC1166
C      670 GTYPE(I) = TYPE                 TRAC1167
C-----TRAC1168
C                                           TRAC1169
C      CHECK AND STORE R.I. GATE INFORMATION. TRAC1170
C                                           TRAC1171
C-----TRAC1172
      IF( TYPE .NE. RI ) GO TO 675        TRAC1173

```

```

      K1 = K1 + 1
      IF( K1 .LE. N3 ) GO TO 672
C=====
C.....ERROR - MAXIMUM R.I. GATE INPUT EXCEEDED.....
      WRITE(6, 903) GATENM(I)
      903 FORMAT( / ' MAXIMUM R.I. GATE INPUT EXCEEDED, GATE ' A6, ' CANNOT
      1BE ADDED TO TABLE.....' )
      NRI = K1
      NG = K
      GO TO 667
C=====
C
      672 IRI(K1) = I
      PI(K1) = PIN
C
      675 IF( TYPE .EQ. AND .OR. TYPE .EQ. OR .OR. TYPE .EQ. RI .OR.
      1 TYPE .EQ. PG ) GO TO 678
      K3 = K3 + 1
C=====
C.....ERROR - INCORRECT GATE TYPE CODE
      WRITE( 6,918 ) GATENM(I)
      918 FORMAT( / ' GATE ' A6, ' CONTAINS INCORRECT TYPE CODE.....' )
C=====
C
      678 IF( NG .EQ. 0 ) GO TO 710
      J2 = 0
C-----
C
C.....STORE FEEDER GATE NAME IF NOT PREVIOUSLY REGISTERED.....
C-----
C
      NG1 = K
      DO 700 J1=1,NG
      DO 680 J=1,NG1
      IF(DATA(J1) .EQ. GATENM(J)) GO TO 685
      680 CONTINUE
      K = K + 1
      IF( K .GT. N ) GO TO 666
      682 GATENM(K) = DATA(J1)
      J = K
C-----
C
C.....STORE SUBSCRIPT OF FEEDER GATE.....
C-----
C
      685 J2 = J2 + 1
      IF( TYPE .NE. PG .OR. IP .LE. 1 ) GO TO 687
      IF( J2 .EQ. 1 ) GO TO 686
      GIN(I,1,1) = J

```

```

      TRAC1174
      TRAC1175
      TRAC1176
      TRAC1177
      TRAC1178
      TRAC1179
      TRAC1180
      TRAC1181
      TRAC1182
      TRAC1183
      TRAC1184
      TRAC1185
      TRAC1186
      TRAC1187
      TRAC1188
      TRAC1189
      TRAC1190
      TRAC1191
      TRAC1192
      TRAC1193
      TRAC1194
      TRAC1195
      TRAC1196
      TRAC1197
      TRAC1198
      TRAC1199
      TRAC1200
      TRAC1201
      TRAC1202
      TRAC1203
      TRAC1204
      TRAC1205
      TRAC1206
      TRAC1207
      TRAC1208
      TRAC1209
      TRAC1210
      TRAC1211
      TRAC1212
      TRAC1213
      TRAC1214
      TRAC1215
      TRAC1216
      TRAC1217
      TRAC1218
      TRAC1219
      TRAC1220
      TRAC1221
      TRAC1222
      TRAC1223
      TRAC1224

```

```

GO TO 689                                TRAC1225
686 GIN(I,2,1) = J                        TRAC1226
GO TO 689                                TRAC1227
687 GIN(I,J2,1) = J                      TRAC1228
-----
C                                         TRAC1229
C                                         TRAC1230
C.....STORE SUBSCRIPT OF GATE.....    TRAC1231
C                                         TRAC1232
-----
C                                         TRAC1233
689 DO 690 J3=1,N1                       TRAC1234
      IF(GOUT(J,J3) .NE. 0) GO TO 690    TRAC1235
      GOUT(J,J3) = I                     TRAC1236
      GO TO 700                           TRAC1237
690 CONTINUE                             TRAC1238
700 CONTINUE                             TRAC1239
-----
C                                         TRAC1240
C                                         TRAC1241
C                                         TRAC1242
C          PICK UP ALL 'FEEDER' GATES AND COMPONENTS AND
C          STORE SUBSCRIPTS FOR GATE JUST READ IN. 'FEEDER'
C          REFERS TO GATES/COMPONENTS FED INTO A GATE.    TRAC1244
C          STORE SUBSCRIPT OF GATE FOR EACH FEEDER GATE AND
C          COMPONENT AS A GATE BEING 'FED TO'.            TRAC1246
C                                         TRAC1247
-----
C                                         TRAC1248
710 IF(NC .EQ. 0) GO TO 650              TRAC1249
      NCO = NCOMP                         TRAC1250
      IF( NCOMP .GT. M ) NCO = M          TRAC1251
      J2 = 0                             TRAC1252
      J4 = NG+1                           TRAC1253
      J5 = NG + NC                        TRAC1254
      DO 740 J1=J4,J5                     TRAC1255
      DO 715 J=1,NCO                       TRAC1256
      IF(DATA(J1) .EQ. COMPNM(J)) GO TO 720 TRAC1257
715 CONTINUE                             TRAC1258
      K2 = K2 + 1                         TRAC1259
C                                         TRAC1260
C=====TRAC1261
C.....ERROR - COMPONENT NAME REQUESTED NOT FOUND.....TRAC1262
      WRITE(6,905) DATA(J1), GATE       TRAC1263
905 FORMAT(/ ' COMPONENT NAME -'A6, '- DESIGNATED FOR GATE ' A6, TRAC1264
1      ' NOT FOUND IN TABLE.....')      TRAC1265
      GO TO 740                           TRAC1266
C=====TRAC1267
C                                         TRAC1268
C-----TRAC1269
C                                         TRAC1270
C.....STORE SUBSCRIPT OF FEEDER COMPONENT.....TRAC1271
C                                         TRAC1272
C-----TRAC1273
720 J2 = J2 + 1                           TRAC1274
      IF( TYPE .NE. PG ) GO TO 726        TRAC1275

```

```

      IF( IP .GT. 1 ) GO TO 722
      IF( NG .EQ. 0 ) GO TO 726
      GIN(1,2,2) = J
      GO TO 727
722  IF( J2 .NE. 1 ) GO TO 723
      IF( NG .NE. 0 ) GO TO 726
      GIN(1,2,2) = J
      GO TO 727
723  GIN(1,1,2) = J
      GO TO 727
726  GIN(1,J2,2) = J
-----
C-----
C-----
C.....STORE SUBSCRIPT OF GATE FOR COMPONENT.....
C-----
C-----
727  DO 730 J3=1,N1
      IF( COUT(J,J3) .NE. 0 ) GO TO 730
      COUT(J,J3) = 1
      GO TO 740
730  CONTINUE
740  CONTINUE
      GO TO 650
-----
C-----
C-----
C.....ALL GATE CARDS READ, CHECK PRINT FLAG.....
C-----
C-----
750  NG = K
      NRI = K1
-----
C-----
C-----
C-----
      IF(NRI.EQ.0)          GO TO 752
      WRITE (6,9075)
      WRITE (6,9080) (I,GATENM(1,I),PI(I),I=1,NRI)
9075  FORMAT (//45X,'INHIBIT GATES AND PROBABILITY ASSIGNED'//)
9080  FORMAT ( 5(1X,I4,2X,A6,2X,F4.2,5X)/ )
752  CONTINUE
-----
C-----
      IF( K2 .EQ. 0 ) GO TO 760
      WRITE(6, 907) K2
907  FORMAT( / ' NUMBER OF COMPONENT NAMES IN ERROR IS' I5 )
760  CONTINUE
902  FORMAT( '1')
      WRITE(6, 906) NG, NRI
906  FORMAT(///,10X,'TOTAL NUMBER OF GATES IS',I5/ ' NUMBER OF RANDOM I
      INHIBIT GATES IS' I5 )
      IF( IPRINT .EQ. 0 ) GO TO 890
-----
C-----
C-----

```

```

TRAC1276
TRAC1277
TRAC1278
TRAC1279
TRAC1280
TRAC1281
TRAC1282
TRAC1283
TRAC1284
TRAC1285
TRAC1286
TRAC1287
TRAC1288
TRAC1289
TRAC1290
TRAC1291
TRAC1292
TRAC1293
TRAC1294
TRAC1295
TRAC1296
TRAC1297
TRAC1298
TRAC1299
TRAC1300
TRAC1301
TRAC1302
TRAC1303
TRAC1304
TRAC1305
TRAC1306
TRAC1307
TRAC1308
TRAC1309
TRAC1310
TRAC1311
TRAC1312
TRAC1313
TRAC1314
TRAC1315
TRAC1316
TRAC1317
TRAC1318
TRAC1319
TRAC1320
TRAC1321
TRAC1322
TRAC1323
TRAC1324
TRAC1325
TRAC1326

```

```

C          PRINT TABLE OF GATES AND THE GATES AND COMPONENTS      TRAC1327
C          THAT FEED THEM                                           TRAC1328
C                                                                    TRAC1329
C-----
C          WRITE (6,6000) TITLE                                     TRAC1330
6000 FORMAT ('1',29X,9A8 )                                         TRAC1331
C          WRITE( 6,922 )                                           TRAC1332
922 FORMAT(///// ' NOTE.....ASTERISKS AFTER NAMES IN TABLES BELOW INDITRAC1333
1CATE GATES THAT ARE NOT FED BY ANY GATES OR COMPONENTS, OR'/ 10X TRAC1334
1 'GATES AND COMPDNENTS THAT DO NOT FEED ANY GATES.')

```



```

WRITE(6,912) (COMPNM(GIN(I,J,2)), J=1,J1)
912 FORMAT( '+' 75X 8(1X A6))
GO TO 780
779 IF( J2 .EQ. 0 ) WRITE( 6,921 ) STAR
921 FORMAT( '+' 13X A2)
780 CONTINUE
C-----
C
C          PRINT TABLE OF GATES AND COMPONENTS AND THE GATES
C          THEY FEED
C-----
C
LINE = 80
IF( LINE .LT. 65 ) GO TO 785
WRITE (6,6000) TITLE
LINE = 0
785 WRITE( 6, 913 )
913 FORMAT(//// )
LINE = LINE + 5
790 WRITE( 6,914 )
914 FORMAT( T24, 'FEEDS THESE GATES', T83, 'FEEDS THESE GATES' /
1 ' NO. GATE 1 2 3 4', T59,
2 'NO. COMPONENT 1 2 3 4' / )
LINE = LINE + 3
K = NG
IF( NCOMP .GT. NG ) K = NCOMP
DO 840 I = 1,K
IF( I .GT. NG ) GO TO 810
DO 795 J= 1,N1
IF( GOUT(I,J) .NE. 0 ) GO TO 795
J1 = J - 1
GO TO 800
795 CONTINUE
J1 = N1
800 IF( LINE .LT. 79 ) GO TO 805
WRITE( 6, 902 )
WRITE( 6,914 )
LINE = 3
805 IF( J1 .EQ. 0 ) GO TO 806
WRITE( 6,915 ) I, GATENM(I), (GATENM(GOUT(I,J)),J=1,J1)
915 FORMAT(/15, 3X A6, 3X 4(1X A6))
GO TO 807
806 WRITE (6,915) I, GATENM(I)
WRITE( 6,921 ) STAR
807 IFF = 0
IF( I .GT. NCOMP ) GO TO 835
GO TO 815
C
810 IFF = 1
815 DO 820 J = 1,N1
IF( COUT(I,J) .NE. 0 ) GO TO 820

```

```

TRAC1378
TRAC1379
TRAC1380
TRAC1381
TRAC1382
TRAC1383
TRAC1384
TRAC1385
TRAC1386
TRAC1387
TRAC1388
TRAC1389
TRAC1390
TRAC1391
TRAC1392
TRAC1393
TRAC1394
TRAC1395
TRAC1396
TRAC1397
TRAC1398
TRAC1399
TRAC1400
TRAC1401
TRAC1402
TRAC1403
TRAC1404
TRAC1405
TRAC1406
TRAC1407
TRAC1408
TRAC1409
TRAC1410
TRAC1411
TRAC1412
TRAC1413
TRAC1414
TRAC1415
TRAC1416
TRAC1417
TRAC1418
TRAC1419
TRAC1420
TRAC1421
TRAC1422
TRAC1423
TRAC1424
TRAC1425
TRAC1426
TRAC1427
TRAC1428

```

```

      J1 = J-1
      GO TO 825
820 CONTINUE
      J1 = N1
825 IF( IFF .NE. 0 ) GO TO 830
      IF( J1 .EQ. 0 ) GO TO 828
      WRITE( 6,916 ) I, COMPNM(I), (GATENM(COUT(I,J)),J=1,J1)
916 FORMAT( 'I' 55X I5, 4X A6, 5X 4(1X A6))
      GO TO 835
828 WRITE( 6,916 ) I, COMPNM(I)
      WRITE( 6,921 ) STAR
      GO TO 835
830 IF ( LINE .LT. 79 ) GO TO 832
      WRITE( 6,902 )
      WRITE( 6,914 )
      LINE = 3
832 IF( J1 .EQ. 0 ) GO TO 834
      WRITE( 6,917 ) I, COMPNM(I), (GATENM(COUT(I,J)),J=1,J1)
917 FORMAT(/56X I5, 4X A6, 5X 4(1X A6))
      GO TO 835
834 WRITE( 6,917 ) I, COMPNM(I)
      WRITE( 6,921 ) STAR
835 LINE = LINE + 2
840 CONTINUE
890 RETURN
C
C=====
C.....ERROR - PREMATURE EOF, NO 'LAST' CARD OR OTHER DATA.....
895 WRITE(6, 909)
909 FORMAT(/ ' NO ''LAST'' CARD OR OTHER DATA INPUT.....')
      STOP
C=====
C
      END

```

```

TRAC1429
TRAC1430
TRAC1431
TRAC1432
TRAC1433
TRAC1434
TRAC1435
TRAC1436
TRAC1437
TRAC1438
TRAC1439
TRAC1440
TRAC1441
TRAC1442
TRAC1443
TRAC1444
TRAC1445
TRAC1446
TRAC1447
TRAC1448
TRAC1449
TRAC1450
TRAC1451
TRAC1452
TRAC1453
TRAC1454
TRAC1455
TRAC1456
TRAC1457
TRAC1458
TRAC1459
TRAC1460
TRAC1461
TRAC1462

```

1,462*

APPENDIX B

USER'S MANUAL FOR
PROGRAM E64106,
TREE ANALYSIS CODE

APPENDIX B

I. PURPOSE OF PROGRAM

This program performs a Monte Carlo simulation to identify the minimal cutsets and critical paths of a fault tree and to estimate the probabilities of fault-tree failure. A fault tree is a diagram that provides an orderly description of those relationships between possible events within a system that can result in a specific failure of the system. The basic input events to a tree are primary and secondary component failures. Such failures are usually time-dependent and independent of each other. In this program, the time to failure for a component is assumed to be a random variable with an exponential distribution. The relationships between events and the manner in which they may combine to produce other events are described by logic gates. Various types of gates are allowed and described in the input section.

A cutset is a set of basic inputs (components) whose combined failure will result in occurrence of the top undesired event in the tree (the specific system failure represented by the fault tree). A minimal cutset is a cutset for which the top event in the tree will not occur if any one of the components has not failed. A fault path is a set of gates through which component failures cause the top of the tree to fail. Critical paths are those fault paths that contribute significant to the probability of tree failure.

The program applies the technique of importance sampling to reduce computer time requirements. This technique consists of modifying the component time-to-failure distributions so failures occur prior to the prescribed end-of-mission time, with a higher probability than for the original distribution, and, after that time, with a correspondingly lower probability. Because this procedure increases the percentage of system failures obtained, minimal cutsets are detected with fewer trials. The procedure also acts as a variance reducing technique and, through the use of weighting functions, provides an unbiased estimate of the probability of fault-tree failure. The power rule is used for the importance sampling.

APPENDIX B

II. RESTRICTIONS

Any number of problems (trees) may be stacked, and any number of cases per tree (different forcing parameters) may be run. Number of trials, random number constant, and mission time may be changed. For each problem, the following restrictions hold:

1. The total number of components must not exceed 1000.
2. The total number of gates must not exceed 2000.
3. No more than 100 random inhibit gates may be used.
4. No gate may have more than eight input components and eight input gates.
5. Priority gates must have two inputs. These inputs may be two gates, two components, or one of each.
6. Random inhibit gates may have only one input other than the associated probability.
7. Any gate or component may provide input to a maximum of four gates.
8. The number of gates along the path from any component to the top of the tree must not exceed 100.

The program will summarize results and terminate prior to completing the requested number of trials if the tree fails because of houses alone or if the number of distinct minimal cutsets identified exceeds 500. If a

APPENDIX B

minimal cutset contains over ten components and houses, it will be ignored; and the trial will be considered successful. The number of such events, however, will be counted and printed out.

III. OPERATING INSTRUCTIONS

The program is written in FORTRAN IV for the IBM 360/65. No disks or special tapes are used. Approximately 310,000 bytes of core are required.

IV. INPUT

A. COMPONENT TYPES

Two types of basic inputs or components are handled by the program.

1. Standard Component

A standard component fails during a trial (a simulated mission length) according to a failure time randomly selected from an exponential distribution. The user furnishes the failure rate (mean number of failures per million time units, normally hours) for each standard component.

2. House

A house is either always on (i.e., the system state described by the house exists for the duration of the mission) or always off for the entire set of trials. Houses are useful in determining subsystem effects on tree failure by permitting simulation of the remainder of the system with that subsystem in a prescribed on or off status. The user furnishes the status for each house. A house input goes only to an AND gate.

APPENDIX B

B. GATE TYPES

The logic of a gate determines when it fails on the basis of the failure status of its input. An input to a gate may be either a component or another gate. Four types of gates are handled by the program:

<u>Gate Type</u>	<u>Logic</u>
OR	An OR gate is failed if any one or more of its inputs is failed.
AND	An AND gate is failed if all its inputs are failed; otherwise, it is not failed.
PRIORITY	A PRIORITY gate is limited to two inputs. It is failed only if both inputs are failed and the failures occurred in a prescribed order.
RANDOM INHIBIT	A RANDOM INHIBIT gate has only one input in addition to an assigned probability for the inhibit conditions. At any particular time, it is failed with that probability only if its input is also failed.

C. INPUT CARDS

The following input cards are required for each problem (unless noted as optional) in the order indicated. Time units are noted as hours, but any time unit is acceptable if used consistently.

1. Title Card

Col. 1-72	Any alphanumeric identification desired to identify the problem.
-----------	--

APPENDIX B

2. Control Card

All values on this card must be right-adjusted.

Col. 1-6	Number of trials to be run per case.
Col. 7	Blank
Col. 8-16	Random Number Constant - any odd integer. If blank, program assumes 13579. Different constants will produce different sequences of random numbers.
Col. 17	Blank
Col. 18-30	Mission length in hours (E format).

3. Component Cards

Col. 1-6	Component Name - left-adjusted. First character must be H if a house.
Col. 7	Blank
Col. 8	House status. (Blank if not a house) 0 - house off 1 - house on
Col. 9	Blank
Col. 10-22	Failure rate - number of failures per million hours (E format, right-adjusted)

4. LAST Card

Col. 1-4	LAST - Indicates end of component cards.
----------	--

APPENDIX B

5. Gate Cards

Col. 1-6	Gate Name - left-adjusted. First character of name determines gate type. R - OR gate A - AND gate P - PRIORITY gate I - RANDOM INHIBIT gate
Col. 7	Blank if not priority gate. 1 - first input given must occur first. 2 - second input given must occur first.
Col. 8	Number of input gates to this gate.
Col. 9	Number of input components to this gate.
Col. 10-16	Probability of failure if RANDOM INHIBIT gate; otherwise blank.
Col. 17	Blank
Col. 18-72	Names of input gates and components. Gates listed first, components second. Names left-adjusted with one blank column after each six-column field. If more than eight inputs, continue on second card in columns 18-72.

6. PRINT Card (Optional)

Col. 1-5	PRINT - If included program will print out all gates and their inputs along with gates to which each component and gate provides input.
----------	---

7. LAST Card

Col. 1-4	LAST - Indicates end of gate cards.
----------	-------------------------------------

8. PRINT Card (Optional)

Col. 1-5	PRINT - If included program will print all the failed gates for the minimal cutsets for each of the following forcing parameter cards in 9.
----------	---

APPENDIX B

9. Forcing Parameter Cards (As many as desired; one required.
 The program will save a maximum of
 20 forcing parameters for a possible
 change of Control Card.)

 Col. 1-9 Blank
 10-16 Forcing parameter R ($0 < R \leq 1$)
10. CHANGE Card (Optional card; can be used any number of times.)

 Col. 1-6 CHANGE - Indicate that the next card is a
 new control card. (See Input Cards,
 Section b.)
11. Control Card (Optional) (Additional to initial Control Card)
 Format for additional ---- CHANGE Card.
12. END Card

 Col. 1-3 END - Indicates end of problem.

V. OUTPUT

A. Each Problem or Tree

1. Title at top of each page.
2. Number of trials, random number constant, mission length.
3. Number of components and number of houses input.
4. Component names and corresponding failure probabilities.
5. House names and probability (1.0 or 0.0) assigned.
6. Inhibit gates and the assigned probabilities.
7. Number of gates input.
8. (Optional) Gate names and their corresponding inputs along with gates to which each component and gate provides input.

APPENDIX B

B. For Each Case

1. Case number.
2. Forcing parameter for this case.
3. (Optional) Failed gates for each minimal cutset.
4. Cutset rank, cutset number, calculated probability of failure, number of occurrences and components contained in each minimal cutset.
5. Number of trials completed, number of system failures, percent of system failures, number of minimal cutsets found, and number of system failures caused by minimal cutsets of over ten components.
6. A calculated probability of system failure based on the identified minimal cutsets and their original time to failure distributions.
7. Based on importance sampling; estimated probability of system failure, its variance and its standard error.

CUSTOMER INSTRUCTIONS	KEYPUNCH INSTRUCTIONS	CUSTOMER	DATE
1 ENTER DATA LEGIBLY WITHIN SPACES PROVIDED	☒ PUNCH 1 CARD PER HAND POSTED LINE ITEM	JOB NO E64106	7-29-69
2 DISTINGUISH BETWEEN 1 vs I, 8 vs 0, 2 vs 2, 4 vs V, 5 vs S	PUNCH ALL LINES WHETHER POSTED OR NOT IF NECESSARY PROVIDE BLANK CARDS		
3. Component Names must be left adjusted.	PUNCH ALL LINES THAT ARE HAND POSTED	PROGRAMMER	
	PAS INCLUDING SPACES	TITLE	
	ALL SPACES MAY BE IGNORED	FORM APPROVED (KEY PUNCH)	DATE
	ALL SPACES MAY BE IGNORED EXCEPT ON T CARD		
	ALL SPACES MAY BE IGNORED EXCEPT (Specify cols)		
	ALL SIGNS AND KP LINES MUST BE PUNCHED		
	DO NOT PUNCH PRE PRINTED SIGNS SHOWN AFTER LAST HANDWRITTEN VALUE ENTRY		
	☒ USE 360 SYMBOLS		

COMPONENT INPUT SHEET

TEST TREE FOR PROGRAM TRACE																																																																															
200	12489	0.72E+03																																																																													
COMP01	0.5E+01																																																																														
COMP02	0.5E+01																																																																														
COMP03	0.5E+01																																																																														
COMP04	0.5E+01																																																																														
COMP05	0.5E+01																																																																														
COMP06	0.5E+01																																																																														
HOUSE1	1																																																																														
COMP07	0.7E+01																																																																														
COMP08	3.5E+01																																																																														
COMP09	3.5E+01																																																																														
COMP10	3.5E+01																																																																														
COMP11	3.5E+01																																																																														
COMP12	0.7E+01																																																																														
COMP13	0.7E+01																																																																														
LAST																																																																															

CUSTOMER INSTRUCTIONS	KEYPUNCH INSTRUCTIONS	CUSTOMER		DATE
1 ENTER DATA LEGIBLY WITHIN SPACES PROVIDED	PUNCH 1 CARD PER HAND POSTED LINE ITEM	JOB NO.	PROGRAMMER	
2 DISTINGUISH BETWEEN I vs l Ø vs o Z vs 2, U vs V S vs 5	PUNCH ALL LINES WHETHER POSTED OR NOT IF NECESSARY PROVIDE BLANK CARDS	E64106		
	PUNCH ALL LINES THAT ARE HAND POSTED	TITLE		
	PAS INCLUDING SPACES	TRACE		
	ALL SPACES MAY BE IGNORED	FORM APPROVED (KEY PUNCH)		
	ALL SPACES MAY BE IGNORED EXCEPT ON T CARD	DATE		
	ALL SPACES MAY BE IGNORED EXCEPT (Spec fy cols)			
	ALL SIGNS AND KP LINES MUST BE PUNCHED			
	DO NOT PUNCH PRE PRINTED SIGNS SHOWN AFTER LAST HANDWRITTEN VALUE ENTRY			
	USE 360 SYMBOLS			

[illegible]

TRACE -- TREE ANALYSIS CODE, AEROJET-GENERAL CORP., SACRAMENTO, CALIFORNIA

TEST TREE FOR PROGRAM TRACE

NUMBER OF TRIALS = 200
RANDOM NO. CONSTANT = 12489
MISSION LENGTH = 720.0

NUMBER OF COMPONENTS = 13
NUMBER OF HOUSES = 1
TOTAL NO. COMP. AND HOUSES = 14

TEST TREE FOR PROGRAM TRACE

COMPONENT NAMES AND PROBABILITIES TIMES 10 TO THE 6TH POWER

1	COMP01	0.500000E 01	2	COMP02	0.500000E 01	3	COMP03	0.500000E 01	4	COMP04	0.500000E 01
5	COMP05	0.500000E 01	6	COMP06	0.500000E 01	7	COMP07	0.700000E 01	8	COMP08	0.350000E 02
9	COMP09	0.350000E 02	10	COMP10	0.350000E 02	11	COMP11	0.350000E 02	12	COMP12	0.700000E 01
13	COMP13	0.700000E 01									

TEST TREE FOR PROGRAM TRACE

HOUSE NAMES AND PROBABILITY ASSIGNED

1 HOUSE1 1.0

INHIBIT GATES AND PROBABILITY ASSIGNED

1 IGAT03 0.08

2 IGAT16 0.20

TOTAL NUMBER OF GATES IS 17
NUMBER OF RANDOM INHIBIT GATES IS 2

TEST TREE FOR PROGRAM TRACE

NOTE.....ASTERISKS AFTER NAMES IN TABLES BELOW INDICATE GATES THAT ARE NOT FED BY ANY GATES OR COMPONENTS, OR GATES AND COMPONENTS THAT DO NOT FEED ANY GATES.

NO.	GATE	FED BY THESE GATES								FED BY THESE COMPONENTS							
		1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	RGAT01	RGAT02	IGAT03	AGAT04													
2	RGAT02	AGAT05	PGAT06	AGAT07													
3	IGAT03	RGAT08															
4	AGAT04	RGAT09	PGAT10							HOUSE1							
5	AGAT05									COMP01	COMP03						
6	PGAT06									COMP01	COMP02						
7	AGAT07									COMP02	COMP03						
8	RGAT08	PGAT12	PGAT11							COMP04							
9	PGAT09	PGAT12								COMP05							
10	PGAT10		AGAT13							COMP06							
11	PGAT12	RGAT15	AGAT14														
12	RGAT11	AGAT17	IGAT16														
13	AGAT13									COMP07	COMP13						
14	AGAT17									COMP12	COMP13						
15	IGAT16									COMP11							
16	AGAT14	AGAT17								COMP09							
17	PGAT15									COMP10	COMP08						

APPENDIX B

TEST TREE FOR PROGRAM TRACE

NO.	GATE	FEEDS THESE GATES			
		1	2	3	4
1	RGAT01**				
2	PGAT02	RGAT01			
3	IGAT03	PGAT01			
4	AGAT04	RGAT01			
5	AGAT05	PGAT02			
6	PGAT06	RGAT02			
7	AGAT07	RGAT02			
8	RGAT08	IGAT03			
9	RGAT09	AGAT04			
10	PGAT10	AGAT04			
11	PGAT12	RGAT08	PGAT09		
12	RGAT11	RGAT08			
13	AGAT13	PGAT10			
14	AGAT17	RGAT11	AGAT14		
15	IGAT16	RGAT11			
16	AGAT14	PGAT12			
17	PGAT15	PGAT12			

NO.	COMPONENT	FEEDS THESE GATES			
		1	2	3	4
1	CCMP01	AGAT05	PGAT06		
2	CCMP02	PGAT06	AGAT07		
3	CCMP03	AGAT05	AGAT07		
4	CCMP04	RGAT08			
5	CCMP05	RGAT09			
6	CCMP06	PGAT10			
7	CCMP07	AGAT13			
8	CCMP08	RGAT15			
9	CCMP09	AGAT14			
10	CCMP10	RGAT15			
11	CCMP11	IGAT16			
12	CCMP12	AGAT17			
13	CCMP13	AGAT13	AGAT17		
14	PCUSE1	AGAT04			

18

APPENDIX B

CUTSET NO.	NO. OF GATES
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1
11	1
12	1
13	1
14	1
15	1
16	1
17	1
18	1
19	1
20	1
21	1
22	1
23	1
24	1
25	1
26	1
27	1
28	1
29	1
30	1
31	1
32	1
33	1
34	1
35	1
36	1
37	1
38	1
39	1
40	1
41	1
42	1
43	1
44	1
45	1
46	1
47	1
48	1
49	1
50	1
51	1
52	1
53	1
54	1
55	1
56	1
57	1
58	1
59	1
60	1
61	1
62	1
63	1
64	1
65	1
66	1
67	1
68	1
69	1
70	1
71	1
72	1
73	1
74	1
75	1
76	1
77	1
78	1
79	1
80	1
81	1
82	1
83	1
84	1
85	1
86	1
87	1
88	1
89	1
90	1
91	1
92	1
93	1
94	1
95	1
96	1
97	1
98	1
99	1
100	1

GATE NAMES

1	3	RGATC1	IGATC3	RGATC8		
2	5	RGATC1	IGATC3	RGATC2	RGAT11	IGAT16
3	3	RGATC1	RGAT02	AGATC5		
4*	3	RGATC1	RGAT02	PGATC6		
5	5	RGATC1	IGATC3	RGATC8	RGAT11	AGAT17

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 1
FORCING PARAMETER IS 0.400000

NOTE.....ASTERISKS AFTER CUTSETS NO. IN TABLE BELOW INDICATE ONE OR MORE PRIORITY GATES IS INCLUDED IN FAILURE PATH.

***** MINIMAL CUTSETS WITH 10 OR LESS COMPONENTS *****				
CUTSET RANK	CUTSET NO.	PROBABILITY OF FAILURE	NO. OF OCCURRENCES	COMPONENTS
1	2	0.398162E-03	14	CCMP11
2	1	0.287480E-03	9	CCMP04
3	3	0.129133E-04	2	CCMP01 CCMP03
4	4*	0.129133E-04	1	CCMP01 CCMP02
5	5	0.202189E-05	1	CCMP12 CCMP13

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 1
FORCING PARAMETER IS 0.400000

***** SUMMARY DATA FOR CASE NUMBER 1 *****

NUMBER OF TRIALS COMPLETED = 200
NUMBER OF SYSTEM FAILURES = 27
PERCENT OF SYSTEM FAILURE = 13.50
NO. ADDITIONAL SYSTEM FAILURES = 0 (OVER 10 COMP. IN MINIMAL CUTSET - NOT INCLUDED IN ABOVE FAILURE COUNT)
NO. OF DISTINCT MINIMAL CUTSETS = 5

ESTIMATE BASED ON MINIMAL CUTSETS
PROBABILITY OF SYSTEM FAILURE = 0.713490E-03

ESTIMATES BASED ON IMPORTANCE SAMPLING
PROBABILITY OF SYSTEM FAILURE = 0.995440E-03
VARIANCE OF ABOVE ESTIMATE = 0.371153E-07
STANDARD ERROR OF ESTIMATE = 0.192653E-03

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 2
FORCING PARAMETER IS 0.350000

NOTE.....ASTERISKS AFTER CUTSETS NO. IN TABLE BELOW INDICATE ONE OR MORE PRIORITY GATES IS INCLUDED IN FAILURE PATH.

***** FAILED GATES FOR ITH MINIMAL CUTSET *****

CUTSET NO.	NO. OF GATES	GATE NAMES				
1	5	RGAT01	IGAT03	RGAT08	PGAT11	IGAT16
2	3	RGATC1	IGAT03	RGAT08		
3	3	RGATC1	PGAT02	AGAT07		
4*	3	RGATC1	RGAT02	PGAT06		
5	3	RGATC1	PGAT02	AGAT05		

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 2
FORCING PARAMETER IS 0.350000

NOTE.....ASTERISKS AFTER CUTSETS NO. IN TABLE BELOW INDICATE ONE OR MORE PRIORITY GATES IS INCLUDED IN FAILURE PATH.

***** MINIMAL CUTSETS WITH 10 OR LESS COMPONENTS *****

CUTSET RANK	CUTSET NO.	PROBABILITY OF FAILURE	NO. OF OCCURRENCES	COMPONENTS
1	1	0.358162E-03	22	CCMP11
2	2	0.287480E-03	17	CCMP04
3	3	0.129133E-04	7	CCMP02 CCMP03
4	4*	0.129133E-04	1	CCMP01 CCMP02
5	5	0.129133E-04	1	CCMP01 CCMP03

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 2
FORCING PARAMETER IS 0.350000

***** SUMMARY DATA FOR CASE NUMBER 2 *****

NUMBER OF TRIALS COMPLETED = 200
NUMBER OF SYSTEM FAILURES = 48
PERCENT OF SYSTEM FAILURE = 24.00
NO. ADDITIONAL SYSTEM FAILURES = 0 (OVER 10 COMP. IN MINIMAL CUTSET - NOT INCLUDED IN ABOVE FAILURE COUNT)
NO. OF DISTINCT MINIMAL CUTSETS = 5

ESTIMATE BASED ON MINIMAL CUTSETS
PROBABILITY OF SYSTEM FAILURE = 0.724382E-03

ESTIMATES BASED ON IMPORTANCE SAMPLING
PROBABILITY OF SYSTEM FAILURE = 0.113234E-02
VARIANCE OF ABOVE ESTIMATE = 0.253111E-07
STANDARD ERROR OF ESTIMATE = 0.159055E-03

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 3
FORCING PARAMETER IS C.300000

NOTE.....ASTERISKS AFTER CUTSETS NO. IN TABLE BELOW INDICATE ONE OF MORE PRIORITY GATES IS INCLUDED IN FAILURE PATH.

***** FAILED GATES FOR ITH MINIMAL CUTSET *****

CUTSET NO.	NO. OF GATES	GATE NAMES				
1	3	RGATC1	IGATC3	RGATC8		
2	5	RGATC1	IGATC3	RGATC8	RGAT11	IGAT16
3	3	RGATC1	RGATC2	AGATC5		
4	3	RGATC1	RGATC2	AGATC7		
5*	3	RGATC1	RGATC2	PGATC6		
6	5	RGATC1	IGATC3	RGATC8	RGAT11	AGAT17

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 3
FORCING PARAMETER IS 0.300000

NOTE.....ASTERISKS AFTER CUTSETS NO. IN TABLE BELOW INDICATE ONE OR MORE PRIORITY GATES IS INCLUDED IN FAILURE PATH.

***** MINIMAL CUTSETS WITH 10 OR LESS COMPONENTS *****

CUTSET RANK	CUTSET NO.	PROBABILITY OF FAILURE	NO. OF OCCURRENCES	COMPONENTS
1	2	0.358162E-03	29	CCMP11
2	1	0.287480E-03	17	CCMP04
3	3	0.129133E-04	8	CCMP01 CCMP03
4	4	0.129133E-04	3	CCMP02 CCMP03
5	5*	0.129133E-04	1	CCMP01 CCMP02
6	6	0.202185E-05	1	CCMP12 CCMP13

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 3
FORCING PARAMETER IS 0.300000

***** SUMMARY DATA FOR CASE NUMBER 3 *****

NUMBER OF TRIALS COMPLETED = 200
NUMBER OF SYSTEM FAILURES = 59
PERCENT OF SYSTEM FAILURE = 29.50
NO. ADDITIONAL SYSTEM FAILURES = 0 (OVER 10 COMP. IN MINIMAL CUTSET - NOT INCLUDED IN ABOVE FAILURE COUNT)
NO. OF DISTINCT MINIMAL CUTSETS = 6

ESTIMATE BASED ON MINIMAL CUTSETS
PROBABILITY OF SYSTEM FAILURE = 0.726403E-03

ESTIMATES BASED ON IMPORTANCE SAMPLING
PROBABILITY OF SYSTEM FAILURE = 0.905845E-03
VARIANCE OF ABOVE ESTIMATE = 0.132537E-07
STANDARD ERROR OF ESTIMATE = 0.115125E-03

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 4
FORCING PARAMETER IS C.450000

NOTE.....ASTERISKS AFTER CUTSETS NO. IN TABLE BELOW INDICATE ONE OR MORE PRIORITY GATES IS INCLUDED IN FAILURE PATH.

***** FAILED GATES FOR ITH MINIMAL CUTSET *****

CUTSET NO.	NO. OF GATES	GATE NAMES
1	3	RGATC1 IGATC3 RGATC8
2	5	RGATC1 IGATC3 RGATC8 RGAT11 AGAT17
3	5	RGATC1 IGATC3 RGATC8 RGAT11 IGAT16
4*	3	RGATC1 RGATC2 PGATC6
5	3	RGATC1 RGATC2 AGATC7

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 4
FORCING PARAMETER IS C.450000

NOTE.....ASTERISKS AFTER CUTSETS NO. IN TABLE BELOW INDICATE ONE OR MORE PRIORITY GATES IS INCLUDED IN FAILURE PATH.

***** MINIMAL CUTSETS WITH 10 OR LESS COMPONENTS *****

CUTSET RANK	CUTSET NO.	PROBABILITY OF FAILURE	NO. OF OCCURRENCES	COMPONENTS
1	3	0.358162E-03	9	CCMP11
2	1	0.267480E-03	4	CCMP04
3	4*	0.129133E-04	1	CCMP01 CCMP02
4	5	0.129133E-04	1	CCMP02 CCMP03
5	2	0.202189E-05	1	CCMP12 CCMP13

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 4
FORCING PARAMETER IS C.450000

***** SUMMARY DATA FOR CASE NUMBER 4 *****

NUMBER OF TRIALS COMPLETED = 200
NUMBER OF SYSTEM FAILURES = 16
PERCENT OF SYSTEM FAILURE = 8.00
NO. ADDITIONAL SYSTEM FAILURES = 0 (OVER 10 COMP. IN MINIMAL CUTSET - NOT INCLUDED IN ABOVE FAILURE COUNT)
NO. OF DISTINCT MINIMAL CUTSETS = 5

ESTIMATE BASED ON MINIMAL CUTSETS
PROBABILITY OF SYSTEM FAILURE = 0.713490E-03

ESTIMATES BASED ON IMPORTANCE SAMPLING
PROBABILITY OF SYSTEM FAILURE = 0.856797E-03
VARIANCE OF ABOVE ESTIMATE = 0.504736E-07
STANDARD ERROR OF ESTIMATE = 0.224663E-03

TEST TREE FOR PROGRAM TRACE

CHANGE OF CONTROL CARD.

NUMBER OF TRIALS = 300
RANDOM NO. CONSTANT = 12489
MISSION LENGTH = 720.0

30-000 Business Bureau, Inc. 30

30

APPENDIX B

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 5
FORCING PARAMETER IS 0.40000

NOTE.....ASTERISKS AFTER CUTSETS NO. IN TABLE BELOW INDICATE ONE OR MORE PRIORITY GATES IS INCLUDED IN FAILURE PATH.

***** FAILED GATES FOR ITH MINIMAL CUTSET *****

CUTSET NO.	NO. OF GATES	GATE NAMES
1	3	RGAT01 IGAT03 RGAT08
2	5	RGAT01 IGAT03 RGAT08 RGAT11 IGAT16
3	3	RGAT01 RGAT02 AGAT05
4*	3	RGAT01 RGAT02 RGAT06
5	5	RGAT01 IGAT03 RGAT08 RGAT11 AGAT17
6	3	RGAT01 RGAT02 AGAT07

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 5
FORCING PARAMETER IS 0.400000

NOTE.....ASTERISKS AFTER CUTSETS NO. IN TABLE BELOW INDICATE ONE OR MORE PRIORITY GATES IS INCLUDED IN FAILURE PATH.

***** MINIMAL CUTSETS WITH 10 OR LESS COMPONENTS *****

CUTSET RANK	CUTSET NO.	PROBABILITY OF FAILURE	NO. OF OCCURRENCES	COMPONENTS
1	2	0.398162E-03	25	CCMP11
2	1	0.287480E-03	16	CCMP04
3	3	0.129133E-04	3	CCMP01 CCMP03
4	4*	0.129133E-04	2	CCMP01 CCMP02
5	6	0.129133E-04	1	CCMP02 CCMP03
6	5	0.202189E-05	1	CCMP12 CCMP13

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 5
FORCING PARAMETER IS 0.400000

***** SUMMARY DATA FOR CASE NUMBER 5 *****

NUMBER OF TRIALS COMPLETED = 300
NUMBER OF SYSTEM FAILURES = 48
PERCENT OF SYSTEM FAILURE = 16.00
NO. ADDITIONAL SYSTEM FAILURES = 0 (OVER 10 COME. IN MINIMAL CUTSET - NOT INCLUDED IN ABOVE FAILURE COUNT)
NO. OF DISTINCT MINIMAL CUTSETS = 6

ESTIMATE BASED ON MINIMAL CUTSETS
PROBABILITY OF SYSTEM FAILURE = 0.726403E-03

ESTIMATES BASED ON IMPORTANCE SAMPLING
PROBABILITY OF SYSTEM FAILURE = 0.118476E-02
VARIANCE OF ABOVE ESTIMATE = 0.286218E-07
STANDARD ERROR OF ESTIMATE = 0.169180E-03

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 6
FORCING PARAMETER IS C.250000

NOTE.....ASTERISKS AFTER CUTSETS NO. IN TABLE BELOW INDICATE ONE OR MORE PRIORITY GATES IS INCLUDED IN FAILURE PATH.

***** FAILED GATES FOR ITH MINIMAL CUTSET *****

CUTSET NO.	NO. OF GATES	GATE NAMES				
1	3	RGATC1	IGAT03	RGATC8		
2	5	RGATC1	IGAT03	RGATC8	RGAT11	IGAT16
3*	3	RGATC1	RGAT02	RGATC6		
4	3	RGATC1	RGATC2	AGATC7		
5	3	RGATC1	RGAT02	AGATC5		

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 6
FORCING PARAMETER IS 0.350000

NOTE.....ASTERISKS AFTER CUTSETS NO. IN TABLE BELOW INDICATE ONE OR MORE PRIORITY GATES IS INCLUDED IN FAILURE PATH.

***** MINIMAL CUTSETS WITH 10 OR LESS COMPONENTS *****

CUTSET RANK	CUTSET NO.	PROBABILITY OF FAILURE	NO. OF OCCURRENCES	COMPONENTS	
1	2	0.398162E-03	25	CCMP11	
2	1	0.287480E-03	22	CCMP04	
3	5	0.129133E-04	6	CCMP01	CCMP03
4	4	0.129133E-04	4	CCMP02	CCMP03
5	3*	0.129133E-04	2	CCMP01	CCMP02

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 6
FORCING PARAMETER IS 0.350000

***** SUMMARY DATA FOR CASE NUMBER 6 *****

NUMBER OF TRIALS COMPLETED = 300
NUMBER OF SYSTEM FAILURES = 59
PERCENT OF SYSTEM FAILURE = 19.67
NO. ADDITIONAL SYSTEM FAILURES = 0 (OVER 10 COMP. IN MINIMAL CUTSET - NOT INCLUDED IN ABOVE FAILURE COUNT)
NO. OF DISTINCT MINIMAL CUTSETS = 5

ESTIMATE BASED ON MINIMAL CUTSETS
PROBABILITY OF SYSTEM FAILURE = 0.724382E-03

ESTIMATES BASED ON IMPORTANCE SAMPLING
PROBABILITY OF SYSTEM FAILURE = 0.906359E-03
VARIANCE OF ABOVE ESTIMATE = 0.140164E-07
STANDARD ERROR OF ESTIMATE = 0.118391E-03

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 7
FORCING PARAMETER IS C.200000

NOTE.....ASTERISKS AFTER CUTSETS NO. IN TABLE BELOW INDICATE ONE OR MORE PRIORITY GATES IS INCLUDED IN FAILURE PATH.

***** FAILED GATES FOR ITH MINIMAL CUTSET *****

CUTSET NO.	NO. OF GATES	GATE NAMES				
1	5	RGATC1	IGATC3	RGATC8	RGAT11	IGAT16
2	3	RGATC1	RGATC2	AGATC7		
3*	3	RGATC1	RGATC2	PGATC6		
4	3	RGATC1	IGATC3	RGATC8		
5	3	RGATC1	RGATC2	AGATC5		
6*	5	RGATC1	AGATC4	RGATC9	PGAT10	AGAT13
7	5	RGATC1	IGATC3	RGATC8	RGAT11	AGAT17

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 7
FORCING PARAMETER IS 0.300000

NOTE.....ASTERISKS AFTER CUTSETS NO. IN TABLE BELOW INDICATE ONE OR MORE PRIORITY GATES IS INCLUDED IN FAILURE PATH.

***** MINIMAL CUTSETS WITH 10 OR LESS COMPONENTS *****

CUTSET RANK	CUTSET NO.	PROBABILITY OF FAILURE	NO. OF OCCURRENCES	COMPONENTS
1	1	0.398162E-03	53	CCMP11
2	4	0.287480E-03	40	CCMP04
3	3*	0.129133E-04	5	CCMP01 CCMP02
4	5	0.129133E-04	5	CCMP01 CCMP03
5	2	0.129133E-04	4	CCMP02 CCMP03
6	7	0.202189E-05	4	CCMP12 CCMP13
7	6*	0.326366E-05	2	CCMP05 CCMP06 CCMP07 CCMP13

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 7
FORCING PARAMETER IS 0.300000

***** SUMMARY DATA FOR CASE NUMBER 7 *****

NUMBER OF TRIALS COMPLETED = 300
NUMBER OF SYSTEM FAILURES = 113
PERCENT OF SYSTEM FAILURE = 37.67
NO. ADDITIONAL SYSTEM FAILURES = 0 (OVER 10 COMP. IN MINIMAL CUTSET - NOT INCLUDED IN ABOVE FAILURE COUNT)
NO. OF DISTINCT MINIMAL CUTSETS = 7

ESTIMATE BASED ON MINIMAL CUTSETS
PROBABILITY OF SYSTEM FAILURE = 0.726403E-03

ESTIMATES BASED ON IMPORTANCE SAMPLING
PROBABILITY OF SYSTEM FAILURE = 0.119810E-02
VARIANCE OF ABOVE ESTIMATE = 0.104043E-07
STANDARD ERROR OF ESTIMATE = 0.102002E-03

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 8
FORCING PARAMETER IS 0.450000

NOTE.....ASTERISKS AFTER CUTSETS NO. IN TABLE BELOW INDICATE ONE OR MORE PRIORITY GATES IS INCLUDED IN FAILURE PATH.

***** FAILED GATES FOR ITH MINIMAL CUTSET *****

CUTSET NO. NO. OF GATES

GATE NAMES

1	5	FGAT01	IGAT03	PGAT08	RGAT11	IGAT16
2	3	RGAT01	IGAT03	PGAT08		
3	5	RGAT01	IGAT03	PGAT08	RGAT11	AGAT17
4*	3	RGAT01	RGAT02	PGAT06		

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 8
FORCING PARAMETER IS C.450000

NOTE.....ASTERISKS AFTER CUTSETS NO. IN TABLE BELOW INDICATE ONE OR MORE PRIORITY GATES IS INCLUDED IN FAILURE PATH.

***** MINIMAL CUTSETS WITH 10 OR LESS COMPONENTS *****

CUTSET RANK	CUTSET NO.	PROBABILITY OF FAILURE	NO. OF OCCURRENCES	COMPONENTS
1	1	0.398162E-03	6	COMP11
2	2	0.2E7480E-03	12	COMP04
3	4*	0.129133E-04	1	COMP01 COMP02
4	3	0.202189E-05	1	COMP12 COMP13

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 8
FORCING PARAMETER IS C.459C00

***** SUMMARY DATA FOR CASE NUMBER 8 *****

NUMBER OF TRIALS COMPLETED = 300
NUMBER OF SYSTEM FAILURES = 20
PERCENT OF SYSTEM FAILURE = 6.67
NO. ADDITIONAL SYSTEM FAILURES = 0 (OVER 10 COMP. IN MINIMAL CUTSET - NOT INCLUDED IN ABOVE FAILURE COUNT)
NO. OF DISTINCT MINIMAL CUTSETS = 4

ESTIMATE BASED ON MINIMAL CUTSETS
PROBABILITY OF SYSTEM FAILURE = 0.700577E-03

ESTIMATES BASED ON IMPORTANCE SAMPLING
PROBABILITY OF SYSTEM FAILURE = 0.730217E-03
VARIANCE OF ABOVE ESTIMATE = 0.274551E-07
STANDARD ERROR OF ESTIMATE = 0.165708E-03

TEST TREE FOR PROGRAM TRACE

CHANGE OF CONTROL CARD.

NUMBER OF TRIALS = 300
RANDOM NO. CONSTANT = 36603
MISSION LENGTH = 1000.0

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 9
FORCING PARAMETER IS C.400000

NOTE.....ASTERISKS AFTER CUTSETS NO. IN TABLE BELOW INDICATE ONE OR MORE PRIORITY GATES IS INCLUDED IN FAILURE PATH.

***** FAILED GATES FOR ITH MINIMAL CUTSET *****

CUTSET NO.	NO. OF GATES	GATE NAMES
1	3	RGAT01 RGAT02 AGAT07
2	5	RGAT01 IGAT03 RGAT08 RGAT11 IGAT16
3	3	RGAT01 IGAT03 RGAT08
4	3	RGAT01 RGAT02 AGAT05
5	5	RGAT01 IGAT03 RGAT08 RGAT11 AGAT17

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 9
FORCING PARAMETER IS 0.400000

NOTE.....ASTERISKS AFTER CUTSETS NO. IN TABLE BELOW INDICATE ONE OR MORE PRIORITY GATES IS INCLUDED IN FAILURE PATH.

***** MINIMAL CUTSETS WITH 10 OR LESS COMPONENTS *****

CUTSET RANK	CUTSET NO.	PROBABILITY OF FAILURE	NO. OF OCCURRENCES	COMPONENTS
1	2	0.550313E-03	29	CCMP11
2	3	0.359003E-03	15	CCMP04
3	1	0.248755E-04	3	CCMP02 CCMP03
4	4	0.248755E-04	2	CCMP01 CCMP03
5	5	0.389264E-05	1	CCMP12 CCMP13

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 9
FORCING PARAMETER IS C.400000

***** SUMMARY DATA FOR CASE NUMBER 9 *****

NUMBER OF TRIALS COMPLETED = 300
NUMBER OF SYSTEM FAILURES = 50
PERCENT OF SYSTEM FAILURE = 16.67
NO. ADDITIONAL SYSTEM FAILURES = 0 (OVER 10 COMP. IN MINIMAL CUTSET - NOT INCLUDED IN ABOVE FAILURE COUNT)
NO. OF DISTINCT MINIMAL CUTSETS = 5

ESTIMATE BASED ON MINIMAL CUTSETS
PROBABILITY OF SYSTEM FAILURE = 0.100296E-02

ESTIMATES BASED ON IMPORTANCE SAMPLING
PROBABILITY OF SYSTEM FAILURE = 0.155794E-02
VARIANCE OF ABOVE ESTIMATE = 0.456573E-07
STANDARD ERROR OF ESTIMATE = 0.213676E-03

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 10
FORCING PARAMETER IS G.350000

NOTE.....ASTERISKS AFTER CUTSETS NO. IN TABLE BELOW INDICATE ONE OR MORE PRIORITY GATES IS INCLUDED IN FAILURE PATH.

***** FAILED GATES FOR ITH MINIMAL CUTSET *****

CUTSET NO.	NO. OF GATES	GATE NAMES
1	3	RGATC1 RGATC2 AGATC7
2	5	RGATC1 IGATC3 RGATC8 RGAT11 AGAT17
3	3	RGATC1 RGATC2 AGATC5
4	3	RGATC1 IGATC3 RGATC8
5	5	RGATC1 IGATC3 RGATC8 RGAT11 IGAT16
6*	3	RGATC1 RGATC2 RGATC6

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 10
FORCING PARAMETER IS 0.350000

NOTE.....ASTERISKS AFTER CUTSETS NO. IN TABLE BELOW INDICATE ONE OR MORE PRIORITY GATES IS INCLUDED IN FAILURE PATH.

***** MINIMAL CUTSETS WITH 10 OR LESS COMPONENTS *****

CUTSET RANK	CUTSET NO.	PROBABILITY OF FAILURE	NO. OF OCCURRENCES	COMPONENTS
1	5	0.550313E-03	36	CCMP11
2	4	0.399003E-03	24	CCMP04
3	1	0.248755E-04	8	CCMP02 CCMP03
4	3	0.248755E-04	6	CCMP01 CCMP03
5	6*	0.248755E-04	4	CCMP01 CCMP02
6	2	0.389264E-05	5	CCMP12 CCMP13

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 10
FORCING PARAMETER IS C.350000

***** SUMMARY DATA FOR CASE NUMBER 10 *****

NUMBER OF TRIALS COMPLETED = 300
NUMBER OF SYSTEM FAILURES = 83
PERCENT OF SYSTEM FAILURE = 27.67
NO. ADDITIONAL SYSTEM FAILURES = 0 (OVER 10 COMP. IN MINIMAL CUTSET - NOT INCLUDED IN ABOVE FAILURE COUNT)
NO. OF DISTINCT MINIMAL CUTSETS = 6

ESTIMATE BASED ON MINIMAL CUTSETS
PROBABILITY OF SYSTEM FAILURE = 0.102783E-02

ESTIMATES BASED ON IMPORTANCE SAMPLING
PROBABILITY OF SYSTEM FAILURE = 0.147330E-02
VARIANCE OF ABOVE ESTIMATE = 0.263585E-07
STANDARD ERROR OF ESTIMATE = 0.162476E-03

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 11
FORCING PARAMETER IS 0.300000

NOTE.....ASTERISKS AFTER CUTSETS NO. IN TABLE BELOW INDICATE ONE OR MORE PRIORITY GATES IS INCLUDED IN FAILURE PATH.

***** FAILED GATES FOR ITH MINIMAL CUTSET *****

CUTSET NO.	NO. OF GATES	GATE NAMES
1	3	RGAT01 RGAT02 AGAT07
2	3	RGAT01 IGAT03 RGAT08
3	5	RGAT01 IGAT03 RGAT08 RGAT11 IGAT16
4*	3	RGAT01 RGAT02 PGAT06
5	5	RGAT01 IGAT03 RGAT08 RGAT11 AGAT17
6	3	RGAT01 RGAT02 AGAT05

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 11
FORCING PARAMETER IS 0.300000

NOTE.....ASTERISKS AFTER CUTSETS NO. IN TABLE BELOW INDICATE ONE OR MORE PRIORITY GATES IS INCLUDED IN FAILURE PATH.

***** MINIMAL CUTSETS WITH 10 OR LESS COMPONENTS *****

CUTSET RANK	CUTSET NO.	PROBABILITY OF FAILURE	NO. OF OCCURRENCES	COMPONENTS
1	3	0.550313E-03	46	CCMP11
2	2	0.359003E-03	36	CCMP04
3	1	0.248755E-04	12	CCMP02 CCMP03
4	6	0.248755E-04	6	CCMP01 CCMP03
5	4*	0.248755E-04	3	CCMP01 CCMP02
6	5	0.269264E-05	7	CCMP12 CCMP13

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 11
FORCING PARAMETER IS 0.300000

***** SUMMARY DATA FOR CASE NUMBER 11 *****

NUMBER OF TRIALS COMPLETED = 300
NUMBER OF SYSTEM FAILURES = 110
PERCENT OF SYSTEM FAILURE = 36.67
NO. ADDITIONAL SYSTEM FAILURES = 0 (OVER 10 COMP. IN MINIMAL CUTSET - NOT INCLUDED IN ABOVE FAILURE COUNT)
NO. OF DISTINCT MINIMAL CUTSETS = 6

ESTIMATE BASED ON MINIMAL CUTSETS
PROBABILITY OF SYSTEM FAILURE = 0.102783E-02

ESTIMATES BASED ON IMPORTANCE SAMPLING
PROBABILITY OF SYSTEM FAILURE = 0.134852E-02
VARIANCE OF ABOVE ESTIMATE = 0.150225E-07
STANDARD ERROR OF ESTIMATE = 0.122566E-03

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 12
FORCING PARAMETER IS C.450000

NOTE.....ASTERISKS AFTER CUTSETS NO. IN TABLE BELOW INDICATE ONE OR MORE PRIORITY GATES IS INCLUDED IN FAILURE PATH.

***** FAILED GATES FOR ITH MINIMAL CUTSET *****

CUTSET NO.	NO. OF GATES	GATE NAMES
1	5	RGAT01 IGAT03 RGAT08 RGAT11 IGAT16
2	3	RGAT01 IGAT03 RGAT08
3	3	RGAT01 RGAT02 AGAT07
4*	3	RGAT01 RGAT02 PGAT06
5	5	RGAT01 IGAT03 RGAT08 RGAT11 AGAT17

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 12
FORCING PARAMETER IS 0.450000

NOTE.....ASTERISKS AFTER CUTSETS NO. IN TABLE BELOW INDICATE ONE OR MORE PRIORITY GATES IS INCLUDED IN FAILURE PATH.

***** MINIMAL CUTSETS WITH 10 OR LESS COMPONENTS *****

CUTSET RANK	CUTSET NO.	PROBABILITY OF FAILURE	NO. OF OCCURRENCES	COMPONENTS
1	1	0.550313E-03	22	COMP11
2	2	0.399003E-03	8	COMP04
3	3	0.248755E-04	4	COMP02 COMP03
4	4*	0.248755E-04	2	COMP01 COMP02
5	5	0.389264E-05	2	COMP12 COMP13

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 12
FORCING PARAMETER IS C.450000

***** SUMMARY DATA FOR CASE NUMBER 12 *****

NUMBER OF TRIALS COMPLETED	=	300
NUMBER OF SYSTEM FAILURES	=	38
PERCENT OF SYSTEM FAILURE	=	12.67
NO. ADDITIONAL SYSTEM FAILURES	=	0 (OVER 10 COMP. IN MINIMAL CUTSET - NOT INCLUDED IN ABOVE FAILURE COUNT)
NO. OF DISTINCT MINIMAL CUTSETS	=	5

ESTIMATE BASED ON MINIMAL CUTSETS
PROBABILITY OF SYSTEM FAILURE = C.100296E-02

ESTIMATES BASED ON IMPORTANCE SAMPLING
PROBABILITY OF SYSTEM FAILURE = C.160798E-02
VARIANCE OF ABOVE ESTIMATE = C.719465E-07
STANDARD ERROR OF ESTIMATE = C.268228E-03