60

RN-DR-173

A Facsimile Report

Reproduced by
UNITED STATES
ATOMIC ENERGY COMMISSION
Division of Technical Information
P.O. Box 62 Oak Ridge, Tennessee 37830

AEROJET GENERAL

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

CLASSIFYING OFFICER — DATE 10/16/69

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

CONTENTS

		Page
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 Probability Under the Power Rule	10
	G. Calculated Probability of System Failure	13
IV.	Method of Solution	14
	A. Program Logic	14
	B. Generation of Random Failure Times From Importance Distribution	16
	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.

BLANK PAGE

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

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

and

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

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

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

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

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

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

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

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

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

In general consider any integral

$$\theta = \int_S g(x)\, f(x)\, dx, \tag{5}$$

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

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

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

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

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

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

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

Now, if n random variables $x_1', x_2', \ldots, x_n'$ are sampled from $h(x)$, the parameter θ may be estimated by

$$\hat{\theta} = \sum_{i=1}^{n} g(x_i')\, w(x_i')/n, \tag{7}$$

where

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

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

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

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

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

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

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

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 nonrepairable 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

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

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

$$P^* = P^r \qquad 0 < r < 1 \tag{12}$$

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

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

and the probability of component failure prior to time T is

$$P = \int_0^T \lambda e^{-\lambda t}\, dt = 1 - e^{-\lambda T}. \tag{14}$$

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

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

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

$$\int_0^\infty f^*(t)dt = \int_0^T f^*(t)dt + \int_T^\infty f^*(t)dt$$

$$= \frac{P^*}{P} \int_0^T f(t)dt + \frac{1-P^*}{1-P} \int_T^\infty f(t)\, dt$$

$$= P^* + (1-P^*) = 1.$$

Since the component failures are independent,

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

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

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

where

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

F. ESTIMATION OF SYSTEM FAILURE PROBABILITY UNDER THE POWER RULE

If $t_1^*, t_2^*, \ldots, t_n^*$ 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

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

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

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

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

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

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

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

and the probability estimate is given by

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

where

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

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 $f(\underline{t})$. 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

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

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

$$\frac{f_i(t_i)}{g_i(\underline{t}_i|\underline{t}_i^*)} = \begin{cases} \prod_i \dfrac{f_i(t_i^*)}{f_i^*(t_i^*)} & \text{for } i\epsilon C \\ 1 & i\not\epsilon C. \end{cases} \tag{26}$$

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

$$w(\underline{t},t_-^*) = \prod_{i\epsilon C} \frac{P_i}{P_i^*} , \tag{27}$$

and, from Equation 23,

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

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

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

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

$$\phi = 12 + 34 + 135. \tag{30}$$

The probability of tree failure may be approximated by

$$P(\phi) = P(12) + P(34) + P(135), \tag{31}$$

since, when the probabilities are small, second- and higher-order terms such as P(12 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

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

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.

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

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

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 $k = \sum_j k_i$ be the total number of failures out of n trials. The following calculations are then performed:

(1) $\hat{p} = \sum_j k_j wtr_j/n.$

(2) $V(\hat{p}) = \dfrac{(n-k)\hat{p}^2 + \sum_j k_j(wtr_j-\hat{p})^2}{n(n-1)}.$

(3) $Psf = \sum_j Ps_j.$

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

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

which is shown graphically in Figure 1.

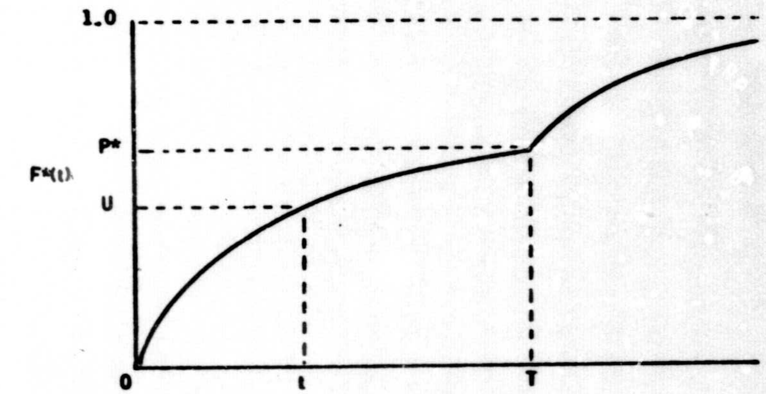


Figure 1 - F*(t) vs t

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

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

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.

BLANK PAGE

APPENDIX A

LISTING OF "PROGRAM TRACE"

BLANK PAGE

APPENDIX A

```
C ***************************************************************************TRAC0001
C                                                                            TRAC0002
C                                 [illegible]                                TRAC0003
C                       TRAC -- TREE ANALYSIS CODE                           TRAC0004
C                                                                            TRAC0005
C                              DEVELOPED BY                                  TRAC0006
C                                                                            TRAC0007
C                       AEROJET-GENERAL CORPORATION                          TRAC0008
C                         SACRAMENTO, CALIFORNIA                             TRAC0009
C                                                                            TRAC0010
C ***************************************************************************TRAC0011
C                                                                            TRAC0012
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
```

BLANK PAGE

```
C       LINE - LINE COUNTER FOR OUTPUT
C       MINSET(500,10) - STORAGE AREA FOR MINIMAL CUTSETS
C       MSET(10) - INDICES OF MINIMAL CUTSET COMPONENTS
C       NC - NUMBER OF COMPONENTS
C       NC1 - NO. COMPONENTS + 1 (INDEX OF FIRST HOUSE)
C       NCFL - NUMBER OF COMPONENTS FAILED IN A TRIAL PRIOR TO T
C       NCOMP - TOTAL NUMBER OF COMPONENTS AND HOUSES (NC+NH)
C       NCUT - NUMBER OF COMPONENTS + HOUSES IN CUTSET
C       NCUT1 - NCUT-1
C       NCUT2 - NCUT + 1
C       NEW - FLAG FOR STARTING NEW PAGE
C       NEXT(1000) - INDICES OF FAILED COMPONENTS
C       NG - NUMBER OF GATES
C       NH - NUMBER OF HOUSES
C       NHFL  - NUMBER OF HOUSES FAILED
C       NHFL1  - NHFLD + 1
C       NMSET(500) - NUMBER COMPONENTS IN ITH MINIMAL CUTSET
C       NRI - NUMBER OF RANDOM INHIBIT GATES
C       NSETS - NUMBER OF DIFFERENT MINIMAL CUTSETS FOUND
C       NSFL - NUMBER OF SYSTEM FAILURES
C       NTR - NUMBER OF TRIALS PER CASE
C       P(1000) - PROB OF FAILURE OF ITH COMPONENT PRIOR TO TIME T
C       P1 - TEMPORARY CALCULATION OF PSF(I)
C       PERCNT - PERCENT SYSTEM FAILURES WITH 10 OR LESS COMP. IN MIN. SET
C       PI(100) - PROBABILITY OF FAILURE OF INHIBIT GATES
C       PROB - PROBABILITY OF SYSTEM FAILURE ESTIMATED FROM WEIGHTING FUNC
C       PS - PROBABILITY OF SYSTEM FAILURE, DIRECT CAL. = SUM OF PSF(I)
C       PSF(500) - PROB FAILURE OF MINIMAL CUTSET BY DIRECT COMPUTATION
C       PSTAR(1000) -  PROB OF FAILURE OF ITH COMP FOR IMPORTANCE DIST.
C       R - FORCING PARAMETER FOR IMPORTANCE SAMPLING
C           RICH   = PROBABILITY OF FAILURE FLAGS FOR R.I. GATES.
C                  IF RANDON VARIABLE IS THAN PROBABILITY,FLAG = .TRUE.
C                  OTHERWISE,FLAG = .FALSE.
C       STDERR - STANDARD ERROR OF PROB
C       T - MISSION LENGTH
C       TEMP1 - TEMPORARY STORAGE
C       TEMPNM - TEMPORARY NAME STORAGE
C       TITLE - TITLE IDENTIFICATION OF CASE
C       TNEXT(1000) - TIME OF FAILURE OF COMPONENTS
C       U - UNIFORM RANDOM NUMBER BETWEEN 0 AND 1
C       VAR - VARIANCE OF PROB
C       WT(1000) - WEIGHTING FUNCTION FOR ITH COMPONENT
C       WT1 - TEMPORARY CALCULATION OF WTR(I)
C       WTI(100) - WEIGHTS FOR RANDOM INHIBIT GATES
C       WTR(500) - WEIGHTING FUNCTION FOR TRIAL
C
C ***********************************************************************
C
      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*8 TITLE,   TEMPNM,COMPNM,GATENM,LAST/'LAST'/,ENDPRR/'END '/
      REAL*8 CHANGE/'CHANGE'/, PRINT/'PRINT'/
      REAL*4 LAMDA(1000)
      INTEGER*4 FLGATE(500)
      INTEGER*2  NMSET(500),HFLAG,H/'H'/
      INTEGER * 2 GTYPE, IAST(500), PG/'P'/
      LOGICAL*1 COMPST,GATEST,TOP,RICH
      DIMENSION P(1000),PSTAR(1000),WT(1000),TNEXT(1000),NEXT(1000),
     1          WTR(500),KOUNT(500),MINSET(500,10),PSF(500),MSET(10)
      DIMENSION RSAVE(20)
C
C ***********************************************************************
C
C                 DIMENSIONS IMPLY FOLLOWING RESTRICTIONS
C
C      MAX OF 1000 COMPONENTS AND HOUSES IN TOTAL
C      MAX OF 2000 GATES IN TOTAL
C      MAX OF  100 RANDOM INHIBIT GATES
C      MAX OF 500 DISTINCT MINIMAL CUTSETS WILL BE FOUND
C
C ***********************************************************************
C
 9000 FORMAT('1TRACE -- TREE ANALYSIS CODE, AEROJET-GENERAL CORPORATION,
     1 SACRAMENTO, CALIFORNIA'//)
 9010 FORMAT( 9A8)
 9020 FORMAT( I1, 29X, 9A8 )
 9025 FORMAT(I6, 1X, I9, 1X, E13.6 )
 9030 FORMAT(A1, T1, A6, 1X, F1.0, 1X, E13.6 )
 9040 FORMAT( //25H NUMBER OF TRIALS =       ,I9/25H RANDOM NO. CONSTANT
     1=    ,I9/21H MISSION LENGTH =     , F13.1//)
 9045 FORMAT(25H NUMBER OF COMPONENTS =  ,I9/25H NUMBER OF HOUSES =
     1 ,I9/30H TOTAL NO. COMP. AND HOUSES = ,I4///)
 9050 FORMAT(//  28X,'COMPONENT NAMES AND PROBABILITIES TIMES 10 TO THE
     16TH POWER' //)
 9060 FORMAT( 4(1X,I4,2X,A6,2X,  E13.6,5X))
 9070 FORMAT(//47X,'HOUSE NAMES AND PROBABILITY ASSIGNED'  //)
 9080 FORMAT(  5(1X,I4,2X,A6,2X,F3.1,5X))
 9090 FORMAT(' INPUT RESTRICTION VIOLATED.  TOTAL NUMBER OF COMPONENTS A
     1ND HOUSES MUST NOT EXCEED 1000.'/ 1X,I7,' HAVE BEEN INPUT.'/
     2' PROGRAM WILL LIST FIRST 1000 THEN CHECK FOR VIOLATION OF GATE RE
     3STRICTIONS'//)
 9100 FORMAT(A6,F7.6)
 9110 FORMAT(///' CASE NUMBER ',I5/' FORCING PARAMETER IS ',F8.6)
 9115 FORMAT(//35X,'***** FAILED GATES FOR ITH MINIMAL CUTSET *****'// '
     1  CUTSET  NO. OF' / '     NO.    GATES', 50X,'GATE NAMES' // )
 9120 FORMAT (/I6,2X,I6,2X,14(2X,A6) /(16X,14(2X,A6) ) )
 9125 FORMAT (/I6,'*',1X,I6,2X,14(2X,A6) / (16X,14(2X,A6) ) )
 9130 FORMAT(///33X,'***** MINIMAL CUTSETS WITH 10 OR LESS COMPONENTS **
     1***'//
     2' CUTSET   CUTSET    PROBABILITY      NO. OF'/
     3'  RANK      NO.     OF FAILURE    OCCURRENCES',35X,'COMPONENTS'/)
```

```
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  UTSET - 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   =',F13.6/                          TRAC0163
    2' VARIANCE OF ABOVE ESTIMATE      =',F13.6/                          TRAC0164
    3' STANDARD ERROR OF ESTIMATE      =',F13.6//)                        TRAC0165
9170 FORMAT ('0',4X,'ESTIMATE BASED ON MINIMAL CUTSETS' /                 TRAC0166
    1 ' PROBABILITY OF SYSTEM FAILURE  =',F13.6 //)                       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 NEXTRAC0175
    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 ('+',12X,'*' )                                                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                                               TRAC0205
     IF(IX .EQ. 0) IX = 13579                                             TRAC0206
C ***********************************************************************TRAC0207
C          INSURE THAT IX IS ODD                                          TRAC0208
C ***********************************************************************TRAC0209
     IXX = IX/2                                                           TRAC0210
     IF ( (IX - 2*IXX) .EQ. 0 )  IX = IX + 1                              TRAC0211
C ***********************************************************************TRAC0212
C                                                                         TRAC0213
C ***** READ COMPONENT CARDS AND CALCULATE PROB OF FAILURE FOR EACH ****TRAC0214
C                                                                         TRAC0215
C ***********************************************************************TRAC0216
  30 READ(5,9030) HFLAG,TEMPNM,TEMP1,TEMP2                                TRAC0217
     IF ( TEMPNM .EQ. LAST) GO TO 50                                      TRAC0218
     NCOMP = NCOMP + 1                                                    TRAC0219
     IF ( NCOMP .GT. 1000 )  GO TO 30                                     TRAC0220
     IF ( HFLAG .EQ. H ) GO TO 40                                         TRAC0221
     NC = NC+1                                                            TRAC0222
     COMPNM(NC) = TEMPNM                                                  TRAC0223
     LAMDA(NC) = TEMP2                                                    TRAC0224
     P(NC) = 1.0 - EXP(-LAMDA(NC) * T /10.0**6)                           TRAC0225
     GO TO 30                                                             TRAC0226
  40 NH = NH+1                                                            TRAC0227
     COMPNM(1001-NH) = TEMPNM                                             TRAC0228
     LAMDA(1001-NH) = TEMP1                                               TRAC0229
     GO TO 30                                                             TRAC0230
C ***********************************************************************TRAC0231
C                                                                         TRAC0232
C        PLACE HOUSES IN COMPONENT ARRAYS IMMEDIATELY                     TRAC0233
C        FOLLOWING THE LAST COMPONENT                                     TRAC0234
C                                                                         TRAC0235
C ***********************************************************************TRAC0236
  50 IF(NH .EQ. 0) GO TO 65                                               TRAC0237
     DO 60 I = 1,NH                                                       TRAC0238
        TEMPNM = COMPNM(NC+I)                                             TRAC0239
        COMPNM(NC+I) = COMPNM(1001-I)                                     TRAC0240
        COMPNM(1001-I) = TEMPNM                                           TRAC0241
        TEMP2 = LAMDA(NC+I)                                               TRAC0242
        LAMDA(NC+I) = LAMDA(1001-I)                                       TRAC0243
  60    LAMDA(1001-I) = TEMP2                                             TRAC0244
C ***********************************************************************TRAC0245
C                                                                         TRAC0246
C *****  WRITE OUT THE COMPONENT INPUT DATA  *****                        TRAC0247
C                                                                         TRAC0248
C ***********************************************************************TRAC0249
  65 WRITE(6,9040) NTR,IX,T                                               TRAC0250
     IF ( NCOMP .LE. 1000 )  GO TO 68                                     TRAC0251
     WRITE(6,9090)  NCOMP                                                 TRAC0252
     GO TO 70                                                             TRAC0253
  68 WRITE(6,9045) NC, NH, NCOMP                                          TRAC0254
  70   WRITE (6,9020) NEW,TITLE                                           TRAC0255
```

```
      WRITE(6,9050)
      N2 = 0
   71 N1 = N2 + 1
      IF ( (NC-N2) .GT. 260 )  GO TO 74
   72 N2 = NC
   73 WRITE(6,9060) (I, COMPNM(I), LAMDA(I), I = N1, N2)
      IF( NC .EQ. N2)  GO TO 75
      WRITE(6,9020) NEW, TITLE
      WRITE(6,9050)
      GO TO 71
   74 N2 = N2 + 260
      GO TO 73
C ---------------------------------------------------------------------
C -----  WRITE OUT HOUSE INPUT DATA  -----
C ---------------------------------------------------------------------
   75 IF ( NH .EQ. 0 )  GO TO 80
      N2 = 0
   76 N1 = N2 + 1
      WRITE(6,9020)  NEW, TITLE
      WRITE(6,9070)
      IF( (NH-N2) .GT. 370 )  GO TO 79
   77 N2 = NH
   78 WRITE(6,9080) ( I, COMPNM(NC+I), LAMDA(NC+I), I = N1, N2)
      IF (NH .EQ. N2)  GO TO 80
      GO TO 76
   79 N2 = N2 + 370
      GO TO 78
C *********************************************************************
C *****  CHECK FOR VIOLATION OF DIMENSION RESTRICTIONS  *****
C *****  READ GATE INPUT DATA                           *****
C *********************************************************************
   80 CALL GATES (NCOMP, NG, NRI)
      IF(NCOMP .GT. 1000 .OR. NG .GT. 2000 .OR. NRI .GT. 100)  GO TO 500
      IF ( K2 .NE. 0 )  GO TO 500
C *********************************************************************
C
C         READ FORCING PARAMETER CARDS AND INITIALIZE
C         VARIABLES TO START A NEW CASE.  IF IT IS AN
C         END CARD GO TO START A NEW PROBLEM
C
C *********************************************************************
C---------------------------------------------------------------------
C     THE FOLLOWING FORTRAN STATEMENTS UP TO STATEMENT 95 WERE ADDED OR
C     MODIFIED BY F.YEE
C
      IPRINT = 0
      IP    = 0
 89   KFLAG  = 0
      IPP   = 0
 90   CONTINUE
      IF(KFLAG .GT. 0)        GO TO 92
```



```
      READ (5,9100) TEMPNM, R
      IF(TEMPNM .EQ. PRINT)  IPRINT =1
C---------------------------------------------------------------------
C     TEST FOR CHANGE OF CONTROL CARD.
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
      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
      PHUH = 0.0
      NSFL = 0
      NSETS = 0
      DO 100 I = 1,500
        NMSFT(I) = 0
        PSF(I) = 0.0
      IAST(I)  = 0
```

```
  100   KOUNT(I) = 0
      DO 110 I = 1,NC
        PSTAR(I) = P(I) ** R
  110   WT(I) = P(I) / PSTAR(I)
C ***********************************************************************
C
C *****  INITIALIZE TRIAL VARIABLES - START NEW TRIAL  *****
C
C ***********************************************************************
  120 IF (KTR .GE. NTR)  GO TO 320
      KTR = KTR + 1
      NHFL = 0
      NCFL =0
      PS = 1.0
      TOP = .FALSE.
      DO 128 I=1,NRI
 128  RICH(I) = .FALSE.
      DO 130 I = 1, NCOMP
  130   COMPST(I) = .FALSE.
      DO 135 I = 1, NG
  135   GATEST(I) = .FALSE.
      IF (NH .EQ. 0)  GO TO 150
C -----------------------------------------------------------------------
C               FAIL HOUSES AS REQUIRED
C -----------------------------------------------------------------------
      NC1 = NC + 1
      DO 140 I = NC1, NCOMP
        IF (LAMDA(I) .EQ. 0)  GO TO 140
        COMPST(I) = .TRUE.
        NHFL = NHFL  + 1
        NEXT(NHFL) = I
        CALL TREE( TOP,I,INH )
        IF ( .NOT. TOP )  GO TO 140
        IF( INH .EQ. 0 )  GO TO 137
        COMPST(I) = .FALSE.
        NHFL = NHFL - 1
  137   IFLAG = 1
        GO TO 297
  140   CONTINUE
C ***********************************************************************
C
C *****  IMPORTANCE SAMPLING OF COMPONENT FAILURE TIMES  *****
C
C ***********************************************************************
      NCFL = NHFL
  150 DO 160 I = 1, NC
        CALL RANDU(IX,IY,U)
        IX = IY
        IF ( U .GT. PSTAR(I) )  GO TO 160
        NCFL = NCFL + 1
        NFXT(NCFL) = I
```



```
        TNEXT(NCFL) =-ALOG( 1.0 - WT(I) * U ) / LAMDA(I)*10.0**6
  160   CONTINUE
      IF ( NCFL .EQ. NHFL )  GO TO 120
C ***********************************************************************
C
C *****  FAIL COMPONENTS IN ORDER OF FAILURE TIMES *****
C
C ***********************************************************************
      NHFL1  = NHFL  + 1
      DO 180 NCUT =  NHFL1, NCFL
        DO 170 I = NCUT,NCFL
          IF ( TNEXT(NCUT) .LE. TNEXT(I) )  GO TO 170
          TEMP1 = TNEXT(I)
          TNEXT(I) = TNEXT(NCUT)
          TNFXT(NCUT) = TEMP1
          ITEMP = NEXT(I)
          NEXT(I) = NEXT(NCUT)
          NEXT(NCUT) = ITEMP
  170     CONTINUE
C       COMPST(NFXT(NCUT)) = .TRUF.
        NT = NFXT(NCUT)
        COMPST(NT) = .TRUE.
        CALL TREE( TOP, NFXT(NCUT), INH )
        IF ( .NOT. TOP )  GO TO 180
        IF ( INH .EQ. 0 )  GO TO 200
        COMPST(NT) = .FALSE.
        NCUT = NCUT - 1
        GO TO 200
  180   CONTINUF
      GO TO 120
C ***********************************************************************
C
C *****  TREE IS FAILED - LOCATE MINIMAL CUTSET
C
C ***********************************************************************
  200 K = 0
      IF ( NCUT .NE. 1 )  GO TO 202
      K = 1
      MSET(K) = NFXT(NCUT)
      GO TO 255
  202 NCUT2 = NCUT + 1
      NCUT1 = NCUT - 1
      L = 0
      DO 220 I = 1, NCUT
        J = NCUT2 - I
C       COMPST( NEXT(J) ) = .FALSE.
        NT = NFXT(J)
        COMPST(NT) = .FALSE.
        CALL TRACE(TOP, NEXT(J), L)
        IF ( TOP )  GO TO 210
C       COMPST( NEXT(J) ) = .TRUE.
```

```
      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 266
************************************************************************
C       SORT MINIMAL CUTSET INTO COMP. NO. ORDER
C ***********************************************************************
230 K1 = K - 1
    DO 250 I = 1, K1
      J1 = I + 1
      DO 250 J = J1, K
        IF( MSET(I) .LT. MSET(J) ) GO TO 250
        ITEMP = MSET(I)
        MSET(I) = MSET(J)
        MSET(J) = ITEMP
250 CONTINUE
266 NSEL = NSEL + 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
C        NEW MINIMAL CUTSET -- STORE IN MINSET
C        PERFORM PROBABILITY CALCULATIONS
C
C ***********************************************************************
280 NSETS = NSETS + 1
    KOUNT(NSETS) = 1
    WT1 = 1.0
    P1 = 1.0
    NMSET(NSETS) = K
    DO 290 J = 1,K
```



```
      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 * WTI(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 ***********************************************************************
```

```
C                                                                       TRAC0562
C ***** SUMMARY CALCULATIONS FOR END OF CASE                             TRAC0563
C                                                                       TRAC0564
C ***********************************************************************TRAC0565
C                                                                       TRAC0566
  320 FKTR = KTR                                                        TRAC0567
      PS = 0.0                                                          TRAC0568
      IF ( NSEL .NE. 0 )  GO TO 325                                     TRAC0569
      PROB = 0.0                                                        TRAC0570
      VAR = 0.0                                                         TRAC0571
      STDERR = 0.0                                                      TRAC0572
      PERCNT = 0.0                                                      TRAC0573
      GO TO 345                                                         TRAC0574
  325 FNSEL = NSEL                                                      TRAC0575
      PROB = PROB / FKTR                                                TRAC0576
      VAR = ( FKTR - FNSEL ) * PROB**2                                  TRAC0577
      DO 330 I = 1, NSETS                                               TRAC0578
        PS = PS + PSF(I)                                                TRAC0579
        TEMP1 = KOUNT(I)                                                TRAC0580
  330   VAR = VAR + TEMP1 * (XIR(I) - PROB)**2                          TRAC0581
      VAR = VAR /(FKTR *(FKTR - 1.0) )                                  TRAC0582
      STDERR = SQRT(VAR)                                                TRAC0583
      PERCNT = 100.0 * FNSEL / FKTR                                     TRAC0584
C.....                                                                  TRAC0585
C     REMOVE HOUSES (IF ANY) FROM THE MINIMAL CUTSETS FOR PRINTING.     TRAC0586
C.....                                                                  TRAC0587
      DO 332 I=1,NSETS                                                  TRAC0588
      K  = NMSET(I)                                                     TRAC0589
      DO 331 J=1,K                                                      TRAC0590
      IF(MINSET(I,J) .LE. NC)  GO TO 331                                TRAC0591
      NMSET(I) = J-1                                                    TRAC0592
      GO TO 332                                                         TRAC0593
  331 CONTINUE                                                          TRAC0594
  332 CONTINUE                                                          TRAC0595
C ***********************************************************************TRAC0596
C                                                                       TRAC0597
C ***** WRITE OUT MINIMAL CUTSETS AND CASE SUMMARY                      TRAC0598
C                                                                       TRAC0599
C ***********************************************************************TRAC0600
      LINE = 16                                                         TRAC0601
      DO 340 I =1, NSETS                                                TRAC0602
        LINE = LINE + 2                                                 TRAC0603
        IF( LINE .GT. 18 .AND. LINE .LE. 50 )  GO TO 333                TRAC0604
        WRITE(6,9020) NEW, TITLE                                        TRAC0605
        WRITE(6,9110) KASE, R                                           TRAC0606
         WRITE (6,9300)                                                 TRAC0607
        WRITE(6,9130)                                                   TRAC0608
        LINE = 18                                                       TRAC0609
C.....                                                                  TRAC0610
C     PRINT IN ORDER OF LARGEST PROBABILITIES.                          TRAC0611
C.....                                                                  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, KTEN, NSETS                TRAC0631
      WRITE(6,9170) PS                                                  TRAC0632
      WRITE(6,9160) PROB, VAR, STDERR                                   TRAC0633
C     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
      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. ENDPRB ) GO TO 10                                 TRAC0652
      GO TO 510                                                         TRAC0653
  550  STOP                                                             TRAC0654
      END                                                               TRAC0655
C                                                                       TRAC0656
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 NEXTTRAC0675
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                                                                            TRAC0694
      SUBROUTINE RANDU(IX,IY,YFL)                                            TRAC0695
      IY = IX * 65539                                                        TRAC0696
      IF(IY) 5, 6, 6                                                         TRAC0697
    5 IY = IY + 2147483647 + 1                                               TRAC0698
    6 YFL = IY                                                               TRAC0699
      YFL = YFL * .4656613E-9                                                TRAC0700
      RETURN                                                                 TRAC0701
      END                                                                    TRAC0702
C************************************************************************TRAC0703
C                                                                            TRAC0704
C     THIS MULTI-ENTRY SUBROUTINE WILL                                       TRAC0705
C          (GATES) 1.  READ GATE CARDS (EACH GROUP FOLLOWED BY               TRAC0706
C                      'LAST' CARD), SET UP DICTIONARIES                     TRAC0707
C          (TREE)  2.  CALC. RF AND WGT. FOR R.I. GATES WHEN 'R'             TRAC0708
C                      CHANGES. CHECK R.I. GATES AND THEIR PROBABILITY,      TRAC0709
C                      TRACE A FAILED COMPONENT THRU ALL POSSIBLE            TRAC0710
C                      PATHS TO THE TOP                                      TRAC0711
C          (TRACE) 3.  CHANGE STATUS OF 1 OR 2 COMPONENTS AND TRACE          TRAC0712
C                      SAME THRU ALL POSSIBLE PATHS TO THE TOP               TRAC0713
C                                                                            TRAC0714
```

14

APPENDIX A


```
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         ICOMPF = SUBSCRIPT OF COMPONENT THAT DID NOT FAIL                  TRAC0727
C         ICOMPT = SUBSCRIPT OF COMPONENT THAT FAILED                        TRAC0728
C         INN    = 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 RANDON 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
```

15

APPENDIX A

```
C**********************************************************************TRAC0766
C                                                                       TRAC0767
      SUBROUTINE TREE (TOP, ICOMPT, INH )                               TRAC0768
      IMPLICIT REAL*8 (A-H,O-Z)                                         TRAC0769
      COMMON /MTREE/ TITLE(9), COMPNM(1000), GATENM(2000), PI(100),     TRAC0770
     1     WTI(10), R, IRI(100), IX, K2, GTYPE(2000), COMPST(1000),     TRAC0771
     2     GATEST(2000), RICH(100)                                      TRAC0772
      REAL*4 PI, WTI, R                                                 TRAC0773
      LOGICAL*1 COMPST, GATEST, TOP, RICH                               TRAC0774
      REAL * 8 LAST, TITLE                                              TRAC0775
      REAL * 4 PSTAR(100), PIN, YFL, RK                                 TRAC0776
      INTEGER * 4 GOUT(2000,4), GIN(2000,6,2), COUT(1000,4)             TRAC0777
      INTEGER * 2 GTYPE,        TYPE, OR, AND, RI, PG                   TRAC0778
      DIMENSION NJ(200), DATA(16)                                       TRAC0779
      DATA LAST, PRINT, OR, AND, RI / 'LAST', 'PRINT', 'R', 'A', 'I' /  TRAC0780
      DATA PG/ 'P' /, STAR/ '********'/                                 TRAC0781
      DATA M, N, N1, N2, N3 /1000, 2000, 6, 8, 100/                     TRAC0782
C                                                                       TRAC0783
      ITRACE = 0                                                        TRAC0784
      INH = 0                                                           TRAC0785
      IRIF = 0                                                          TRAC0786
      IF( NRI .EQ. 0) GO TO 40                                          TRAC0787
      IF( R .EQ. RK ) GO TO 30                                          TRAC0788
C                                                                       TRAC0789
C-----------------------------------------------------------------------TRAC0790
C                                                                       TRAC0791
C        NEW R VALUE, CALC. PI AND WEIGHT FOR ALL R.I. GATES            TRAC0792
C                                                                       TRAC0793
C-----------------------------------------------------------------------TRAC0794
      DO 20 I=1,NRI                                                     TRAC0795
      PSTAR(I) = PI(I) ** R                                             TRAC0796
   20 WTI(I) = PI(I)/ PSTAR(I)                                          TRAC0797
      RK = R                                                            TRAC0798
C-----------------------------------------------------------------------TRAC0799
C                                                                       TRAC0800
C        SET PROBABILITY OF FAILURE AT EACH R.I. GATE                   TRAC0801
C                                                                       TRAC0802
C-----------------------------------------------------------------------TRAC0803
   30 K4 = 1                                                            TRAC0804
   31 CALL RANDU ( IX, IY, YFL )                                        TRAC0805
      IX = IY                                                           TRAC0806
      NJT = IRI(K4)                                                     TRAC0807
      IF( YFL .LE. PSTAR(K4) ) GO TO 32                                 TRAC0808
      IF( .NOT. RICH(K4) ) GO TO 35                                     TRAC0809
      RICH(K4) = .FALSE.                                                TRAC0810
       IF( .NOT. GATEST(NJT) ) GO TO 35                                 TRAC0811
C-----------------------------------------------------------------------TRAC0812
C                                                                       TRAC0813
C        CHANGE RI GATE STATUS FROM TRUE TO FALSE AND TRACE TREE        TRAC0814
C                                                                       TRAC0815
C-----------------------------------------------------------------------TRAC0816
```


APPENDIX A


```
      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(COMPST(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) = COUT (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
```


APPENDIX A

```
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 O 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. 1 ) 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. ICOMPT ) 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. ICOMPT ) 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
```



```
                                                                        TRAC0919
   70 J = NJ(J1)                                                        TRAC0920
C-----------------------------------------------------------------------TRAC0921
C                                                                       TRAC0922
C                .....CHECK GATES.....                                  TRAC0923
C                                                                       TRAC0924
C-----------------------------------------------------------------------TRAC0925
      DO 80 I=1,N2                                                      TRAC0926
      IF(GIN(J,I,1) .EQ. 0) GO TO 90                                    TRAC0927
C     IF(GATEST(GIN(J,I,1))    ) GO TO 130                              TRAC0928
      NJT = GIN(J,I,1)                                                  TRAC0929
      IF( .NOT. GATEST(NJT) ) GO TO 130                                 TRAC0930
   80 CONTINUE                                                          TRAC0931
C-----------------------------------------------------------------------TRAC0932
C                                                                       TRAC0933
C                .....CHECK COMPONENTS.....                             TRAC0934
C                                                                       TRAC0935
C-----------------------------------------------------------------------TRAC0936
   90 DO 100 I=1,N2                                                     TRAC0937
      IF(GIN(J,I,2) .EQ. 0) GO TO 105                                   TRAC0938
C     IF( .NOT. COMPST(GIN(J,I,2))    )GO TO 130                        TRAC0939
      NJT = GIN(J,I,2)                                                  TRAC0940
      IF( .NOT. COMPST(NJT) ) GO TO 130                                 TRAC0941
  100 CONTINUE                                                          TRAC0942
  105 GATEST(J) =.TRUE.                                                 TRAC0943
C-----------------------------------------------------------------------TRAC0944
C                                                                       TRAC0945
C          GATE FED TO, FAILED                                          TRAC0946
C                                                                       TRAC0947
C-----------------------------------------------------------------------TRAC0948
  110 NJ(J1+1) = J2                                                     TRAC0949
      IF(GOUT(NJ(J1),1) .NE. 0 ) GO TO 120                              TRAC0950
C-----------------------------------------------------------------------TRAC0951
C                                                                       TRAC0952
C          TREE TOPPED.....SET 'TOP' TO TRUE (FAILED)                   TRAC0953
C                                                                       TRAC0954
C-----------------------------------------------------------------------TRAC0955
      TOP =.TRUE.                                                       TRAC0956
      GO TO 130                                                         TRAC0957
C                                                                       TRAC0958
  120 J1 = J1 + 2                                                       TRAC0959
      NJ(J1) = GOUT(NJ(J1-2),1)                                         TRAC0960
      J2 = 1                                                            TRAC0961
      GO TO 50                                                          TRAC0962
C-----------------------------------------------------------------------TRAC0963
C                                                                       TRAC0964
C          GATE FED DID NOT FAIL.........AT LEAST 1 GATE OR COMPONENT   TRAC0965
C          DID NOT FAIL                                                 TRAC0966
C                                                                       TRAC0967
C-----------------------------------------------------------------------TRAC0968
  130 J2 = J2 + 1                                                       TRAC0969
      IF(J1 .EQ. 1)    GO TO 145
```

```
  135 IF(J2 .GT. N1) GO TO 140
      IF (J1 .EQ. 1 ) GO TO 145
      IF(GOUT(NJ(J1-2),J2) .EQ. 0) GO TO 140
      NJ(J1) = GOUT(NJ(J1-2),J2)
      GO TO 50
  140 J2 = NJ(J1-1) + 1
      J1 = J1 - 2
      IF(J1 .GT. 1)    GO TO 135
      J1 = 1
  145 IF(J2 .GT. N1)   GO TO 150
      IF( TRIP .EQ. 0 ) GO TO 146
      NJ(1) = GOUT(TRI(K4),J2)
      GO TO 147
  146 NJ( 1) = COUT(ICOMPT,J2)
  147 IF( NJ(J1) .EQ. 0 ) GO TO 150
      GO TO 50
C
  150 IF( ITRACE .NE. 0 ) GO TO 300
      IF( TRIP .NE. 0 ) GO TO 35
      RETURN
C
C**************************************************************************
C**************************************************************************
C
      ENTRY TRACE ( TOP, ICOMPF, ICOMPT )
C
C**************************************************************************
C**************************************************************************
      ITRACE = 1
      TRIP = 0
      IF( ICOMPT .NE. 0 ) GO TO 40
  300 IF( ICOMPF .EQ. 0 ) GO TO 450
C-------------------------------------------------------------------------
C
C          COMPONENT SET TO FALSE, THAT IS SUCCESS
C                TRACE TREE UPWARDS.  GATES MAY BE CHANGED TO SUCCESS.
C                IF TOPPED, SET TOP TO FALSE.
C
C-------------------------------------------------------------------------
      NJ( 1) = COUT (ICOMPF,1)
  305 J1 = 1
      J2 = 1
  310 IF( GTYPE(NJ(J1)) .EQ. OR ) GO TO 320
C-------------------------------------------------------------------------
C
C          'AND' OR 'RI' OR 'PRIORITY' GATE.....NOT FAILED
C
C-------------------------------------------------------------------------
C     GATEST(NJ(J1)) =.FALSE.
      NJT = NJ(J1)
      GATEST(NJT) = .FALSE.
```


APPENDIX A


```
      GO TO 370
C-------------------------------------------------------------------------
C
C          'OR' GATE.....CHECK ALL GATES AND COMPONENTS FED TO GATE
C
C-------------------------------------------------------------------------
  320 J = NJ(J1)
C-------------------------------------------------------------------------
C
C                .....CHECK GATES.....
C
C-------------------------------------------------------------------------
      DO 330 I=1,N2
      IF(GIN(J,I,1) .EQ. 0 ) GO TO 340
C     IF(GATEST(GIN(J,I,1)) )GO TO 400
      NJT = GIN(J,I,1)
      IF( GATEST(NJT) ) GO TO 400
  330 CONTINUE
C-------------------------------------------------------------------------
C
C                .....CHECK COMPONENTS.....
C
C-------------------------------------------------------------------------
  340 DO 350 I=1,N2
      IF(GIN(J,I,2) .EQ. 0 ) GO TO 360
C     IF(COMPST(GIN(J,I,2))  ) GO TO 400
      NJT = GIN(J,I,2)
      IF( COMPST(NJT) ) GO TO 400
  350 CONTINUE
  360 GATEST(J) =.FALSE.
C-------------------------------------------------------------------------
C
C          GATE FED TO, NOT FAILED
C
C-------------------------------------------------------------------------
  370 NJ(J1+1) = J2
      IF(GOUT(NJ(J1),1) .NE. 0 ) GO TO 380
C-------------------------------------------------------------------------
C
C          TREE TOPPED.....SET TOP TO FALSE (NOT FAILED)
C
C-------------------------------------------------------------------------
      TOP =.FALSE.
      GO TO 400
C
  380 J1 = J1 + 2
      NJ(J1) = GOUT(NJ(J1-2),1)
      J2 = 1
      GO TO 310
C-------------------------------------------------------------------------
C
```


APPENDIX A

```
C        GATE FED TO REMAINS FAILED.....AT LEAST 1 GATE IN COMPONENT
C        FAILED
C
C--------------------------------------------------------------------------
  400 J2 = J2 + 1
      IF( J1 .EQ. 1) [illegible]
  410 IF( J2 .GT. N1) GO TO [illegible]
      IF( J1 .EQ. 1 ) GO TO [illegible]
      IF( GOUT(NJ(I1-2), J2) [illegible]
      NJ(J1) = GOUT(NJ(I1-2), [illegible])
      GO TO 410
  420 J2 = NJ(J1-1) + 1
      J1 = J1 - [illegible]
      IF( J1 .GT. 1) [illegible]
      J1 = 1
  425 IF( J2 .GT. N1) [illegible]
      IF( [illegible] .EQ. [illegible] ) GO TO 42[illegible]
      NJ(J1) = GOUT( [illegible] )
      GO TO 42[illegible]
  426 NJ(J1) = [illegible]
  427 IF( NJ(J1) [illegible] ) GO TO [illegible]
      GO TO 410
  450 IF( ITRACE [illegible] ) GO TO [illegible]
      RETURN
C
C*************************************************************************
C*************************************************************************
C
C     ENTRY GATES (NCOMP, NG, NRI)
C
C*************************************************************************
C*************************************************************************
C--------------------------------------------------------------------------
C
C                 INITIALIZE VARIABLES
C
C--------------------------------------------------------------------------
      IPRINT = 0
      K = 0
      K1 = [illegible]
      K2 = 0
      K3  =0
      RK = 0.0
      DO 605 I=1,N1
      DO 600 J=1,M
  600 COUT (J,I) = 0
      DO 604 J=1,N
  604 GOUT (J,I) = 0
  605 CONTINUE
      DO 610 I=1,N[illegible]
```



```
      DO 610 J=1,N
      GIN (J,I,1) = 0
  610 GIN (J,I,2) = 0
C
C--------------------------------------------------------------------------
C
C          READ GATE CARDS
C
C--------------------------------------------------------------------------
  650 READ(5, 900, END=895) TYPE, GATE,IP,NG,NC,PIN,(DATA(I),I=1,8)
  900 FORMAT( A1, T1,A6,    3I1, F7.6, 8(1X A6))
      IF( GATE .NE. PRINT ) GO TO 655
      IPRINT = 1
      GO TO 650
  655 IF( GATE .EQ. LAST ) GO TO 750
      IF( NG + NC .GT. 8 ) READ(5, 901, END= 895) (DATA(I),I=9,16)
  901 FORMAT(16X 8(1X A6))
      IF( K .EQ. 0 ) GO TO 665
C--------------------------------------------------------------------------
C
C                 CHECK IF GATE NAME HAS BEEN REGISTERED IN GATE
C                 NAME ARRAY.  IF NO, STORE NAME AND TYPE.
C
C--------------------------------------------------------------------------
      DO 660 I=1,K
      IF( GATE .EQ. GATENM(I)) GO TO 670
  660 CONTINUE
  665 K = K + 1
      IF( K .LE. N ) GO TO 668
C
C==========================================================================
C.............ERROR - MAXIMUM GATE INPUT EXCEEDED.........................
  666 WRITE(6, 904) DATA(J1), GATE
  904 FORMAT(/ ' MAXIMUM GATE INPUT EXCEEDED, GATE 'A6, ' CANNOT BE ADDE
     1D TO TABLE.....GATE CARD NAME IS ' A6)
      NG = K
      NRI = K1
  667 READ( 5,900, END=895 ) TYPE, GATE
      IF( GATE .EQ. LAST ) GO TO 890
      GO TO 667
C==========================================================================
C
  668 GATENM(K) = GATE
      I = K
  670 GTYPE(I) = TYPE
C--------------------------------------------------------------------------
C
C                 CHECK AND STORE R.I. GATE INFORMATION.
C
C--------------------------------------------------------------------------
      IF( TYPE .NE. RI ) GO TO 675
```

```
      K1 = K1 + 1
      IF( K1 .LE. N3 ) GO TO 672
C
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=========================================================================
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
```



```
      GO TO 689
  686 GIN(I,2,1) = J
      GO TO 689
  687 GIN(I,J2,1) = J
C-------------------------------------------------------------------------
C
C..............STORE SUBSCRIPT OF GATE...................................
C
C-------------------------------------------------------------------------
  689 DO 690 J3=1,N1
      IF(GOUT(J,J3) .NE. 0) GO TO 690
      GOUT(J,J3) = I
      GO TO 700
  690 CONTINUE
  700 CONTINUE
C-------------------------------------------------------------------------
C
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.
C              STORE SUBSCRIPT OF GATE FOR EACH FEEDER GATE AND
C              COMPONENT AS A GATE BEING 'FED TO'.
C
C-------------------------------------------------------------------------
  710 IF(NC .EQ. 0) GO TO 650
      NCO = NCOMP
      IF( NCOMP .GT. M ) NCO = M
      J2 = 0
      J4 = NG+1
      J5 = NG + NC
      DO 740 J1=J4,J5
      DO 715  J=1,NCO
      IF(DATA(J1) .EQ. COMPNM(J)) GO TO 720
  715 CONTINUE
      K2 = K2 + 1
C
C=========================================================================
C..............ERROR - COMPONENT NAME REQUESTED NOT FOUND................
      WRITE(6,905) DATA(J1), GATE
  905 FORMAT(/ ' COMPONENT NAME -'A6, '- DESIGNATED FOR GATE ' A6,
     1   ' NOT FOUND IN TABLE.....')
      GO TO 740
C=========================================================================
C
C-------------------------------------------------------------------------
C
C-------------------------------------------------------------------------
C
C..............STORE SUBSCRIPT OF FEEDER COMPONENT.......................
C
C-------------------------------------------------------------------------
  720 J2 = J2 + 1
      IF( TYPE .NE. PG ) GO TO 726
```

```
      IF( IP .GT. 1 ) GO TO 722
      IF( NG .EQ. 0 ) GO TO 726
      GIN(I,2,2) = J
      GO TO 727
  722 IF( J2 .NE. 1 ) GO TO 723
      IF( NG .NE. 0 ) GO TO 726
      GIN(I,2,2) = J
      GO TO 727
  723 GIN(I,1,2) = J
      GO TO 727
  726 GIN(I,J2,2) = J
C-------------------------------------------------------------------------
C
C..............STORE SUBSCRIPT OF GATE FED COMPONENT......................
C
C-------------------------------------------------------------------------
  727 DO 730 J3=1,N1
      IF( COUT(J,J3) .NE. 0 ) GO TO 730
      COUT(J,J3) = I
      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     FORTRAN STATEMENTS FROM HERE TO STATEMENT 752 WERE ADDED BY F.YEE
C-------------------------------------------------------------------------
      IF(NRI.EQ.0)              GO TO 752
      WRITE (6,9075)
      WRITE (6,9080) (I,GATENM(IRI(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
```



```
C                 PRINT TABLE OF GATES AND THE GATES AND COMPONENTS
C                 THAT FEED THEM
C
C-------------------------------------------------------------------------
      WRITE (6,6000)   TITLE
 6000 FORMAT ('1',29X,9A8 )
      WRITE( 6,922 )
  922 FORMAT(////  ' NOTE.....ASTERISKS AFTER NAMES IN TABLES BELOW INDI
     1CATE GATES THAT ARE NOT FED BY ANY GATES OR COMPONENTS, OR'/ 10X
     1     'GATES AND COMPONENTS THAT DO NOT FEED ANY GATES.')
      WRITE(6, 908)
  908 FORMAT(// 36X 'FED BY THESE GATES', T94,'FED BY THESE COMPONENTS'/
     1   ' NO.     GATE        1      2      3      4      5      6
     2 7      8        1      2      3      4      5      6      7
     3 8'/ )
      LINE = 13
      DO 780 I=1,NG
      DO 765 J=1,N2
      IF(GIN(I,J,1) .NE. 0) GO TO 765
      J1 = J-1
      GO TO 766
  765 CONTINUE
      J1 = N2
  766 IF( LINE .LT. 79 ) GO TO 767
      WRITE( 6, 902 )
      WRITE( 6, 908 )
      LINE = 5
  767 WRITE(6,910) I, GATENM(I)
  910 FORMAT(/I5, 3X A6)
      LINE = LINE + 2
      IF( GTYPE(I) .NE. PG ) GO TO 768
      IF( GIN(I,1,1) .NE. 0 ) WRITE( 6,911 ) GATENM(GIN(I,1,1))
      IF( GIN(I,2,1) .NE. 0 ) WRITE( 6,919 ) GATENM(GIN(I,2,1))
  919 FORMAT( '+' 24X A6)
      GO TO 770
  768 IF( J1 .EQ. 0 ) GO TO 770
      WRITE(6,911) (GATENM(GIN(I,J,1)),J=1,J1)
  911 FORMAT( '+' 16X 8(1X A6))
  770 J2 = J1
      DO 772 J=1,N2
      IF(GIN(I,J,2) .NE. 0 ) GO TO 772
      J1 = J-1
      GO TO 775
  772 CONTINUE
      J1 = N2
  775 IF( GTYPE(I) .NE. PG ) GO TO 778
      IF( GIN(I,1,2) .NE. 0 ) WRITE( 6,912 ) COMPNM(GIN(I,1,2))
      IF( GIN(I,2,2) .NE. 0 ) WRITE( 6,920 ) COMPNM(GIN(I,2,2))
  920 FORMAT( '+' 83X A6)
      GO TO 780
  778 IF( J1 .EQ. 0 ) GO TO 779
```

```
      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 .O 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(/I5, 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
```

28

APPENDIX A


```
      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( '+' 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
```

1.462*

APPENDIX A

APPENDIX B

USER'S MANUAL FOR PROGRAM E64106, TREE ANALYSIS CODE

BLANK PAGE

APPENDIX B

PURPOSE OF PROGRAM

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

A cutset is a set of basic inputs (components) whose combined failure ill result in occurrence of the top undesired event in the tree (the specific ystem failure represented by the fault tree). A minimal cutset is a cutset or which the top event in the tree will not occur if any one of the components as not failed. A fault path is a set of gates through which component ailures 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.

BLANK PAGE

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

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.

B. GATE TYPES

The logic of a te 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:

Gate Type	Logic
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.

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.

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.

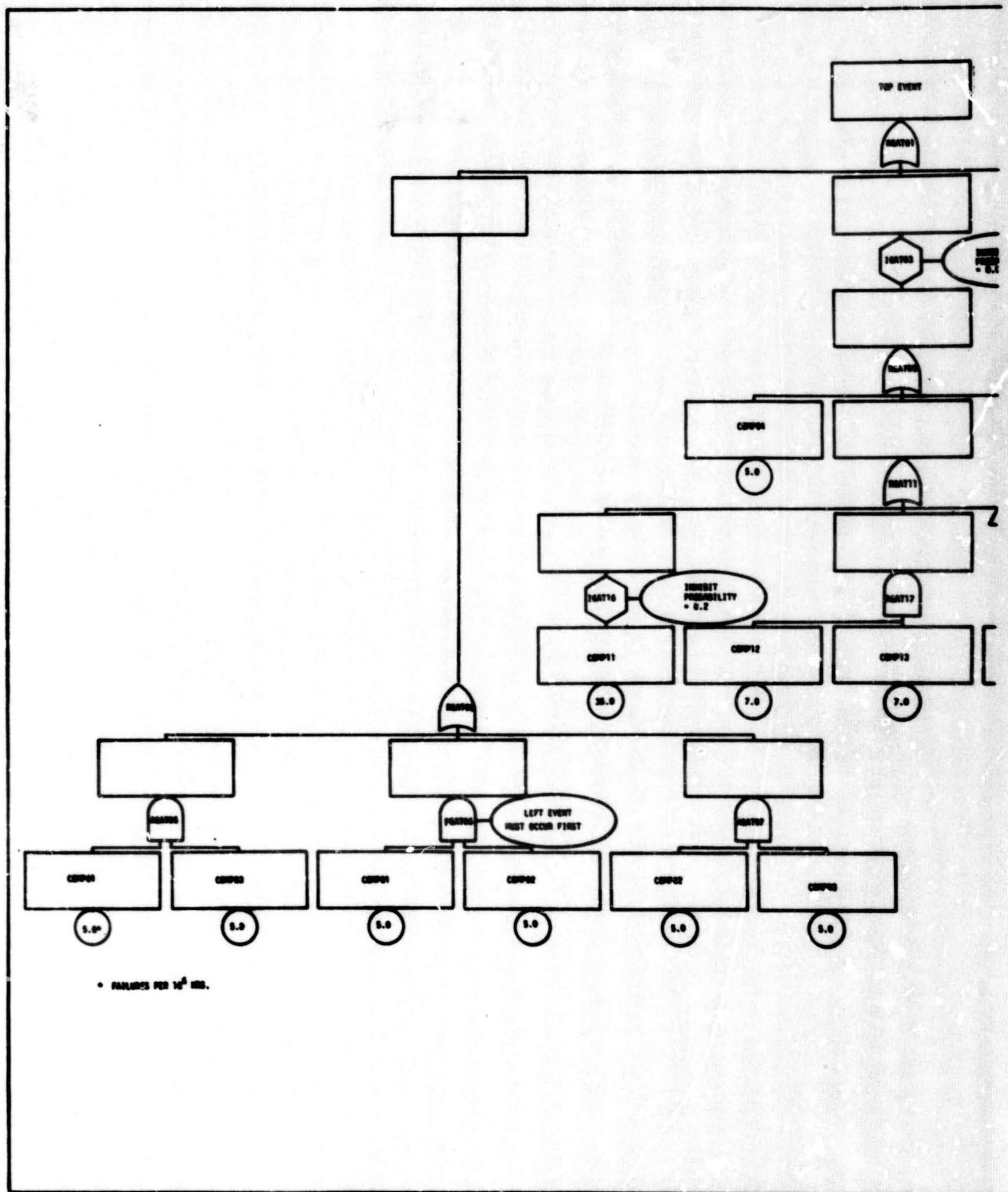
3. (Optional) Failed gates for each minimal cutset.

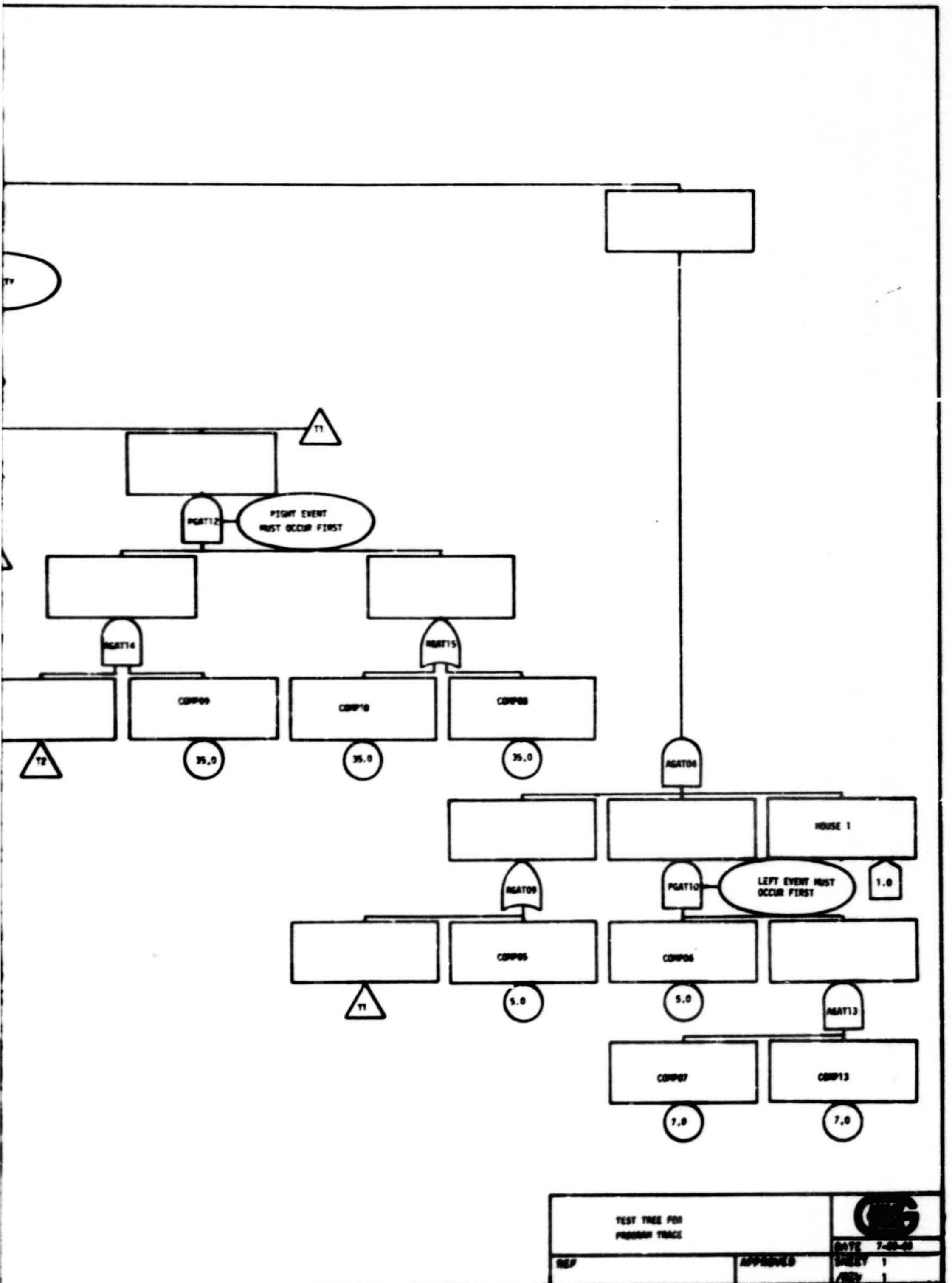
4. Cutset rank, cutset number, calculated probability of failure, number of occurrences and components contained in each minimal cutset.

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.




T1
PGAT12
RIGHT EVENT MUST OCCUR FIRST
AGAT14
AGAT15
COMP09
COMP10
COMP08
T2
35.0
35.0
35.0
AGAT04
HOUSE 1
AGAT09
PGAT10
LEFT EVENT MUST OCCUR FIRST
1.0
COMP05
COMP06
T1
5.0
5.0
AGAT13
COMP07
COMP13
7.0
7.0
TEST TREE FOR PROGRAM TRACE
REF
APPROVED
SHEET 1
REV 1

2

BLANK PAGE

CUSTOMER INSTRUCTIONS	KEYPUNCH INSTRUCTIONS	
1. ENTER DATA LEGIBLY WITHIN SPACES PROVIDED	[X] PUNCH 1 CARD PER HAND POSTED LINE ITEM	CUSTOMER:
2. DISTINGUISH BETWEEN: I ≠ 1, O ≠ 0, Z ≠ 2, U ≠ V, Ī ≠ 1	[ ] PUNCH ALL * LINES WHETHER POSTED OR NOT. IF NECESSARY PROVIDE BLANK CARDS	DATE: 7-29-69
3. Component Names must be left adjusted.	[ ] PUNCH ALL * LINES THAT ARE HAND POSTED	JOB NO.: B64106
	[ ] PAS INCLUDING SPACES	PROGRAMMER:
	[ ] ALL SPACES MAY BE IGNORED	TITLE:
	[ ] ALL SPACES MAY BE IGNORED EXCEPT ON T CARD	FORM APPROVED (KEY PUNCH):
	[ ] ALL SPACES MAY BE IGNORED EXCEPT (Specify cols.)	DATE:
	[ ] ALL SIGNS AND KP LINES MUST BE PUNCHED	
	[ ] DO NOT PUNCH PRE-PRINTED SIGNS SHOWN AFTER LAST HANDWRITTEN VALUE ENTRY	
	[X] USE 240 SYMBOLS	

COMPONENT INPUT SHEET

TEST TREE FOR PROGRAM TRACE

200	12487	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		

ABC3 0000 (REV 10/00)

PLEASE PRINT CLEARLY - USE BLACK PENCIL

CUSTOMER INSTRUCTIONS	KEYPUNCH INSTRUCTIONS	
1. ENTER DATA LEGIBLY WITHIN SPACES PROVIDED	[X] PUNCH 1 CARD PER HAND POSTED LINE ITEM	CUSTOMER:
2. DISTINGUISH BETWEEN: I ≠ 1, O ≠ 0, Z ≠ 2, U ≠ V, Ī ≠ 1	[ ] PUNCH ALL * LINES WHETHER POSTED OR NOT. IF NECESSARY PROVIDE BLANK CARDS	DATE: 7-29-69
3/ All Gate Names must be left adjusted	[ ] PUNCH ALL * LINES THAT ARE HAND POSTED	JOB NO.: B64106
	[ ] PAS INCLUDING SPACES	PROGRAMMER:
	[ ] ALL SPACES MAY BE IGNORED	TITLE:
	[ ] ALL SPACES MAY BE IGNORED EXCEPT ON T CARD	FORM APPROVED (KEY PUNCH):
	[ ] ALL SPACES MAY BE IGNORED EXCEPT (Specify cols.)	DATE:
	[ ] ALL SIGNS AND KP LINES MUST BE PUNCHED	
	[ ] DO NOT PUNCH PRE-PRINTED SIGNS SHOWN AFTER LAST HANDWRITTEN VALUE ENTRY	
	[X] USE 240 SYMBOLS	

EXAMPLE OF THE CORRESPONDING INPUT REQUIRED.

GATE INPUT SHEET

RGAT01	30		RGAT02	IGAT03	AGAT04
RGAT02	30		AGAT05	PGAT06	AGAT07
IGAT03	10	.08	AGAT08		
AGAT04	21		RGAT09	PGAT10	HOUSE1
AGAT05	02		COMP01	COMP03	
PGAT06	102		COMP01	COMP02	
AGAT07	02		COMP02	COMP03	
AGAT08	21		PGAT12	RGAT11	COMP04
RGAT09	11		PGAT12	COMP05	
PGAT10	211		AGAT13	COMP06	
RGAT11	21		AGAT17	IGAT16	
PGAT12	220		AGAT14	RGAT15	
AGAT13	02		COMP07	COMP13	
AGAT14	11		AGAT17	COMP09	
RGAT15	02		COMP10	COMP08	
IGAT16	01	.2	COMP11		
AGAT17	02		COMP12	COMP13	
PRINT					
LAST					
		0.40			
		0.36			
		0.30			
		0.45			

ABC3 0000 (REV 10/00)

PLEASE PRINT CLEARLY - USE BLACK PENCIL

Keypunch Instructions
PUNCH 1 CARD PER HAND POSTED LINE ITEM
PUNCH ALL LINES WHETHER POSTED OR NOT. IF NECESSARY PROVIDE BLANK CARDS
PUNCH ALL LINES THAT ARE HAND POSTED
PAS INCLUDING SPACES
ALL SPACES MAY BE IGNORED
ALL SPACES MAY BE IGNORED EXCEPT ON 1 CARD
ALL SPACES MAY BE IGNORED EXCEPT (Specify cols.)
ALL SIGNS AND XP LINES MUST BE PUNCHED
DO NOT PUNCH PRE-PRINTED SIGNS SHOWN AFTER LAST HANDWRITTEN VALUE ENTRY
USE 360 SYMBOLS

JOB NO. 864106

TITLE: TRACE

```
CHANGE
   300      12489      0.72E+03
CHANGE
   300      36903      1.00E+05
END
```

PLEASE PRINT CLEARLY - USE BLACK PENCIL


```
TRACE -- TREE ANALYSIS CODE, AEROJET-GENERAL CORPORATION, SACRAMENTO, CALIFORNIA

                                  TEST TREE FOR PROGRAM TRACE

NUMBER OF TRIALS =              300
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	0.800000E 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 0.90   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 1	2	3	4	5	6	7	8	FED BY THESE COMPONENTS 1	2	3	4	5	6	7	8
1	RGAT01	RGAT02	IGAT03	AGAT04													
2	RGAT02	AGAT05	RGAT06	AGAT07													
3	IGAT03	RGAT08															
4	AGAT04	RGAT09	RGAT10							HOUSE1							
5	AGAT05									COMP01	COMP03						
6	RGAT06									COMP01	COMP02						
7	AGAT07									COMP02	COMP03						
8	RGAT08	RGAT12	RGAT11							COMP04							
9	RGAT09	RGAT12								COMP05							
10	RGAT10		AGAT13							COMP06							
11	RGAT12	RGAT15	AGAT14														
12	RGAT11	AGAT17	IGAT16														
13	AGAT13									COMP07	COMP13						
14	AGAT17									COMP12	COMP13						
15	IGAT16									COMP11							
16	AGAT14	AGAT17								COMP09							
17	RGAT15									COMP10	COMP08						

APPENDIX B

16

TEST TREE FOR PROGRAM TRACE

NO.	GATE	FEEDS THESE GATES 1	2	3	4
1	RGAT01**				
2	RGAT02	RGAT01			
3	IGAT03	RGAT01			
4	AGAT04	RGAT01			
5	AGAT05	RGAT02			
6	RGAT06	RGAT02			
7	AGAT07	RGAT02			
8	RGAT08	IGAT03			
9	RGAT09	AGAT04			
10	RGAT10	AGAT04			
11	RGAT12	RGAT08	RGAT09		
12	RGAT11	RGAT08			
13	AGAT13	RGAT10			
14	AGAT17	RGAT11	AGAT14		
15	IGAT16	RGAT11			
16	AGAT14	RGAT12			
17	RGAT15	RGAT12			

NO.	COMPONENT	FEEDS THESE GATES 1	2	3	4
1	COMP01	AGAT05	RGAT06		
2	COMP02	RGAT06	AGAT07		
3	COMP03	AGAT05	AGAT07		
4	COMP04	RGAT08			
5	COMP05	RGAT09			
6	COMP06	RGAT10			
7	COMP07	AGAT13			
8	COMP08	RGAT15			
9	COMP09	AGAT14			
10	COMP10	RGAT15			
11	COMP11	IGAT16			
12	COMP12	AGAT17			
13	COMP13	AGAT13	AGAT17		
14	HOUSE1	AGAT04			

APPENDIX B

17

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.

***** 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

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	10	CCMP11
2	1	0.287480E-03	9	CCMP06
3	3	0.129113E-04	2	CCMP01 CCMP03
4	4*	0.129113E-04	1	CCMP01 CCMP02
5	5	0.202109E-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   = 0.713450E-03

    ESTIMATES BASED ON IMPORTANCE SAMPLING
PROBABILITY OF SYSTEM FAILURE   = 0.955440E-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 0.380000
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. OF                                       GATE NAMES
 NO.    GATES

  1       5     RGAT01  IGAT03  RGAT08  RGAT11  IGAT16
  2       3     RGAT01  IGAT03  RGAT08
  3       3     RGAT01  RGAT02  AGAT07
  4*      3     RGAT01  RGAT02  RGAT06
  5       3     RGAT01  RGAT02  AGAT05
```

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 2
FORCING PARAMETER IS [illegible]

NOTE.....ASTERISK[illegible] [illegible] IN TABLE WHICH 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	[illegible]	[illegible]	22	COMP11
[illegible]	[illegible]	[illegible]	17	COMP05
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
4	[illegible]	[illegible]	1	[illegible]
[illegible]	[illegible]	[illegible]	1	[illegible]

APPENDIX B

22

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 2
FORCING PARAMETER IS 0.250000

***** 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 = 0.159098E-03

APPENDIX B

23

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.

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

CUTSET NO.	NO. OF GATES	GATE NAMES
1	3	RGATE1 IGATE1 RGATE8
2	5	RGATE1 IGATE1 RGATE8 RGATE11 IGATE1[illegible]
3	3	RGATE1 IGATE1 AGATE8
4	3	RGATE1 IGATE2 AGATE7
5*	3	RGATE1 IGATE2 RGATE8
6	5	RGATE1 IGATE1 RGATE8 RGATE11 AGATE1[illegible]

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	COMP11
2	1	0.287480E-03	17	COMP06
3	3	0.125123E-04	8	COMP01 COMP03
4	4	0.125123E-04	3	COMP02 COMP03
5	5*	0.125123E-04	1	COMP01 COMP02
6	6	0.282180E-05	1	COMP12 COMP13

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   = 0.726463E-03

    ESTIMATES BASED ON IMPORTANCE SAMPLING
PROBABILITY OF SYSTEM FAILURE   = 0.965849E-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 0.400000
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	RGATE1	IGATE3	RGATE8		
2	5	RGATE1	IGATE3	RGATE8	RGATE11	AGATE17
3	5	RGATE1	IGATE3	RGATE8	RGATE11	IGATE16
4**	3	RGATE1	PGATE2	PGATE8		
5	3	RGATE1	RGATE2	AGATE7		

APPENDIX B

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 4
FORCING PARAMETER IS 0.450000

NOTE.....ASTERISKS AFTER CUTSET 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	[illegible]	0.[illegible]581672E-03	5	COMP11
2	[illegible]	0.[illegible]574[illegible]E-03	4	COMP04
3	4*	0.1[illegible]E-04	1	COMP01 COMP02
4	[illegible]	0.1[illegible]E-04	1	COMP02 COMP03
5	[illegible]	0.[illegible]E-0[illegible]	1	COMP12 COMP13

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 4
FORCING PARAMETER IS 0.450000

***** 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 = 0.713469E-03

ESTIMATES BASED ON IMPORTANCE SAMPLING
PROBABILITY OF SYSTEM FAILURE = 0.656797E-03
VARIANCE OF ABOVE ESTIMATE = 0.204736E-07
STANDARD ERROR OF ESTIMATE = 0.224667E-03

TEST TREE FCS PROGRAM TRACE

CHANGE OF CONTROL CARD.

NUMBER OF TRIALS = 300
RANDOM NO. CONSTANT = 12485
MISSION LENGTH = 720.0

TEST TREE FCS PROGRAM TRACE

CASE NUMBER [illegible]
FORCING PARAMETER IS 0.400000

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	RGATE1	IGATE3	RGATE8		
2	5	RGATE1	IGATE3	RGATE8	RGAT11	IGAT16
3	3	RGATE1	RGATE2	AGATE8		
4*	3	RGATE1	RGATE2	RGATE6		
5	5	RGATE1	IGATE3	RGATE8	RGAT11	AGAT17
6	3	RGATE1	RGATE2	AGATE7		

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	0.358168E-03	25	COMP11	
2	1	0.287480E-03	16	COMP06	
3	3	0.189133E-04	3	COMP01	COMP03
4	4	0.189133E-04	2	COMP01	COMP02
5	6	0.189133E-04	1	COMP02	COMP03
6	5	0.202199E-05	1	COMP12	COMP13

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 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   = 0.112470E-02
VARIANCE OF ABOVE ESTIMATE      = 0.286218E-07
STANDARD ERROR OF ESTIMATE      = 0.169180E-03
```


APPENDIX B

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 6
FORCING PARAMETER IS 0.250000

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	PGATE1 IGATE2 PGATE8
2	5	PGATE1 IGATE2 PGATE8 PGATE11 IGATE16
3*	3	PGATE1 PGATE2 PGATE8
4	3	PGATE1 PGATE2 AGATE7
5	3	PGATE1 PGATE2 AGATE9

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 6
FORCING PARAMETER IS 0.250000

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	25	CCMP11
2	1	0.287480E-03	22	CCMP04
3	5	0.129123E-04	4	CCMP01 CCMP03
4	4	0.129123E-04	4	CCMP02 CCMP03
5	3*	0.129123E-04	2	CCMP01 CCMP02

APPENDIX B

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 6
FORCING PARAMETER IS 0.100000

***** 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 = 0.724308E-03

ESTIMATES BASED ON IMPORTANCE SAMPLING
PROBABILITY OF SYSTEM FAILURE = 0.900399E-03
VARIANCE OF ABOVE ESTIMATE = 0.140160E-07
STANDARD ERROR OF ESTIMATE = 0.118391E-03

APPENDIX B

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.

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

CUTSET NO.	NO. OF GATES	GATE NAMES				
1	5	RGATE1	IGATE3	RGATE8	RGAT11	IGAT16
2	3	RGATE1	RGATE2	AGATE7		
3*	3	RGATE1	RGATE2	PGATE6		
4	3	RGATE1	IGATE3	RGATE8		
5	3	RGATE1	RGATE2	AGATE8		
6*	5	RGATE1	AGATE4	RGATE5	PGAT10	AGAT13
7	5	RGATE1	IGATE3	RGATE8	RGAT11	AGAT17

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 7
FORCING PARAMETER IS 0.200000

NOTE.....ASTERISKS AFTER CUTSET 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	0.2591628E-03	53	COMP11
2	4	0.2274805E-03	40	COMP04
3	3*	0.129123E-04	5	COMP01 COMP02
4	5	0.129123E-04	5	COMP01 COMP03
5	2	0.129123E-04	4	COMP02 COMP03
6	7	0.2021258E-05	4	COMP12 COMP13
7	6*	0.2802665E-05	2	COMP05 COMP06 COMP07 COMP13

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 7
FORCING PARAMETER IS 0.200000

***** 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 = 3 (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 = 0.724403E-03

ESTIMATES BASED ON IMPORTANCE SAMPLING
PROBABILITY OF SYSTEM FAILURE = 0.115810E-02
VARIANCE OF ABOVE ESTIMATE = 0.106040E-07
STANDARD ERROR OF ESTIMATE = 0.102980E-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	RGAT01	IGAT03	RGAT08	RGAT11	IGAT16
2	3	RGAT01	IGAT03	RGAT08		
3	5	RGAT01	IGAT03	RGAT08	RGAT11	AGAT17
4*	3	RGAT01	RGAT02	RGAT08		

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.

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

CUTSET RANK	CUTSET NO.	PROBABILITY OF FAILURE	NO. OF OCCURRENCES	COMPONENTS	
1	1	0.358168E-03	6	COMP11	
2	2	0.287080E-03	12	COMP04	
3	4*	0.189133E-04	1	COMP01	COMP02
4	3	0.262169E-05	1	COMP12	COMP13

```
CASE NUMBER      8
FORCING PARAMETER IS 0.450000

***** 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      = 0.165700E-03
```



```
CHANGE OF CONTROL CARD.

NUMBER OF TRIALS =             300
RANDOM NO. CONSTANT =        34503
MISSION LENGTH =            1000.0
```

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.

***** 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	0.5503133E-03	29	COMP11
2	3	0.3990030E-03	15	COMP04
3	1	0.2487555E-04	3	COMP02 COMP03
4	4	0.2487555E-04	2	COMP01 COMP03
5	5	0.3852640E-05	1	COMP12 COMP13

TEST TREE FOR PROGRAM TRACE

```
CASE NUMBER        9
FORCING PARAMETER IS 0.400000

***** 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   = 0.100258E-02

      ESTIMATES BASED ON IMPORTANCE SAMPLING
PROBABILITY OF SYSTEM FAILURE   = 0.159754E-02
VARIANCE OF ABOVE ESTIMATE      = 0.456973E-07
STANDARD ERROR OF ESTIMATE      = 0.213670E-03
```


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.

                    ***** FAILED GATES FOR ITH MINIMAL CUTSET *****
CUTSET  NO. OF
  NO.   GATES                                  GATE NAMES

    1      3    PGATE1  PGATE2  AGATE7
    2      5    PGATE1  IGATE3  PGATE8  PGAT11  AGAT17
    3      3    PGATE1  RGATE2  AGATE5
    4      3    PGATE1  IGATE3  RGATE8
    5      5    PGATE1  IGATE3  RGATE8  PGAT11  IGAT16
    6*     3    PGATE1  PGATE2  PGATE6
```

TEST TAPE FOR PROGRAM TRACE

CASE NUMBER 10
FORCING PARAMETER IS 0.250000

NOTE.....ASTERISKS AFTER CUTSET 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	9	0.[illegible]E-03	30	CCMP11	
2	4	0.2555[illegible]E-03	24	CCMP04	
3	1	0.2481[illegible]E-04	8	CCMP02	CCMP03
4	3	0.2481[illegible]E-04	6	CCMP01	CCMP03
5	[illegible]	0.2481[illegible]E-04	4	CCMP01	CCMP02
6	2	0.[illegible]E-05	[illegible]	CCMP12	CCMP13

TEST TAPE FOR PROGRAM TRACE

CASE NUMBER 10
FORCING PARAMETER IS 0.250000

***** 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-0[illegible]
VARIANCE OF ABOVE ESTIMATE = 0.263[illegible]E-07
STANDARD ERROR OF ESTIMATE = 0.1624[illegible]E-03

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 11
FORCING PARAMETER IS 0.200000
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	3	PGATE1 PGATE2 AGATE7
2	3	PGATE1 IGATE3 PGATE8
3	5	PGATE1 IGATE3 PGATE8 PGAT11 IGAT16
4	3	PGATE1 PGATE2 PGATE6
5	5	PGATE1 IGATE3 PGATE8 PGAT11 AGAT17
6	3	PGATE1 PGATE2 AGATE9

TEST TREE FOR PROGRAM TRACE

CASE NUMBER 11
FORCING PARAMETER IS 0.200000
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	0.880213E-03	46	COMP11
2	2	0.799003E-03	36	COMP04
3	1	0.240799E-04	18	COMP02 COMP03
4	6	0.240799E-04	6	COMP01 COMP03
5	4	0.240799E-04	3	COMP01 COMP02
6	5	0.399164E-05	7	COMP12 COMP13

TEST CASE 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 = 0.134292E-02
VARIANCE OF ABOVE ESTIMATE = 0.150229E-07
STANDARD ERROR OF ESTIMATE = 0.122566E-03

TEST CASE 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.

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

CUTSET NO.	NO. OF GATES	GATE NAMES				
1	5	PGATE1	IGATE3	PGATE8	PGATE11	IGATE16
2	3	PGATE1	IGATE3	PGATE8		
3	3	PGATE1	PGATE2	AGATE7		
4**	3	PGATE1	PGATE2	PGATE8		
5	5	PGATE1	IGATE3	PGATE8	PGATE11	AGATE17

TEST CASE FOR PROGRAM TRACE

CASE NUMBER 12
FORCING PARAMETER IS 0.450000

NOTE.....ASTERISKS AFTER CUTSETS NO. IN TABLE WHICH 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	0.550213E-03	22	COMP11
2	2	0.299003E-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 CASE FOR PROGRAM TRACE

CASE NUMBER 12
FORCING PARAMETER IS 0.450000

***** 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 = 0.100296E-02

ESTIMATES BASED ON IMPORTANCE SAMPLING
PROBABILITY OF SYSTEM FAILURE = 0.140798E-02
VARIANCE OF ABOVE ESTIMATE = 0.719409E-07
STANDARD ERROR OF ESTIMATE = 0.268228E-03