MFAULT: A Computer Program for Analyzing Fault Trees

by
**P. J. Pelto
W. L. Purcell**

November 1977

**Prepared for the
U.S. Department of Energy
under Contract EY-76-C-06-1830**

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the Department of Energy, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

The views, opinions and conclusions contained in this report are those of the contractor and do not necessarily represent those of the United States Government or the United States Department of Energy

PACIFIC NORTHWEST LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
Under Contract EY-76-C-06-1830

Printed in the United States of America
Available from
National Technical Information Service
United States Department of Commerce
5255 Port Royal Road
Springfield, Virginia 22151

Price: Printed Copy \$____*; Microfiche \$3.00

*Pages	NTIS Selling Price
001-025	\$4.50
026-050	\$5.00
051-075	\$5.50
076-100	\$6.00
101-125	\$6.50
126-150	\$7.00
151-175	\$7.75
176-200	\$8.50
201-225	\$8.75
226-250	\$9.00
251-275	\$10.00
276-300	\$10.25

3 3679 00062 7036

MFAULT: A COMPUTER PROGRAM FOR ANALYZING
FAULT TREES

by
P. J. Pelto
W. L. Purcell

November 1977

BATTELLE
Pacific Northwest Laboratories
Richland, Washington 99352

CONTENTS

ACKNOWLEDGMENTS.	vii
SUMMARY	ix
1.0 INTRODUCTION	1
2.0 INFORMATION FOR THE USER	3
2.1 GENERAL DESCRIPTION OF PROGRAM	3
2.2 INPUT DESCRIPTION FOR MFAULT	5
2.3 OUTPUT FROM MFAULT	9
2.4 SAMPLE CASE.	9
3.0 PROCEDURES FOR SOLUTION	11
3.1 FAULT TREE TRANSFORMATION	11
3.2 CUT SET IDENTIFICATION	11
3.3 TREATMENT OF ABSORPTION.	12
3.3.1 Duplicate Cut Sets	13
3.3.2 Minimal Cut Sets.	13
3.4 PROBABILITY CALCULATION.	14
3.5 SCREENING PROCEDURES.	14
4.0 ADDITIONAL PROGRAMMING INFORMATION.	17
4.1 GENERAL INFORMATION	17
4.1.1 External Files	17
4.1.2 Nonstandard Functions	17
4.2 SUBROUTINE DESCRIPTIONS.	18
4.3 USE OF SCRATCH STORAGE IN MFAULT.	28
4.4 EXAMPLE OF CUT SET IDENTIFICATION PROCEDURE	29
REFERENCES	35
APPENDIX A	A-1
APPENDIX B	B-1

•

•

•

•

•

•

•

•

FIGURES

1	Risk Assessment Computer Package	2
2	Example Fault Tree	10
3	Subroutines Used in MFAULT	19
4	Flow Chart for MFAULT	20
5	Flow Chart for MIXEM	22
6	Flow Chart for ORGT.	23
7	Flow Chart for ANDGT	24
8	Flow Chart for ARRANG	25
9	Flow Chart for MINIMAL.	26
10	Flow Chart for WEED.	27
11	Example Fault Tree Branch.	30

ACKNOWLEDGMENTS

The authors acknowledge the many helpful discussions with T. H. Smith during the development of the program. We thank R. E. Barlow and P. K. Pande for providing an early version of the computer program TREEL.

SUMMARY

A description and user instructions are presented for MFAULT, a FORTRAN computer program for fault tree analysis. MFAULT identifies the cut sets of a fault tree, calculates their probabilities, and screens the cut sets on the basis of specified cut-offs on probability and/or cut set length.

MFAULT is based on an efficient upward-working algorithm for cut set identification. The probability calculations are based on the assumption of small probabilities and constant hazard rates (i.e., exponential failure distributions). Cut sets consisting of repairable components (basic events) only, non-repairable components only, or mixtures of both types can be evaluated. Components can be on-line or standby. Unavailability contributions from pre-existing failures, failures on demand, and testing and maintenance down-time can be handled.

MFAULT can analyze fault trees with AND gates, OR gates, inhibit gates, on switches (houses) and off switches. The code is presently capable of finding up to ten event cut sets from a fault tree with up to 512 basic events and 400 gates. It is operational on the CONTROL DATA CYBER 74 computer.

1.0 INTRODUCTION

A risk-based fault tree analysis method has been developed at Pacific Northwest Laboratories (PNL) for analysis of nuclear fuel cycle operations.⁽¹⁾ This method was developed for the Energy Research and Development Administration (ERDA)-sponsored risk analysis of systems for managing high-level waste. A computer package consisting of three programs has been written to assist in the performance of a risk assessment: ACORN⁽²⁾ (draws fault trees), MFAULT (analyzes fault trees), and RAFT⁽³⁾ (calculates risk). Figure 1 gives a summary of the input, output and interrelationships of these programs.

This report gives a description and user instructions for MFAULT, a computer program for analyzing fault trees. Section 2 gives a basic description of the program along with input/output information and a sample case. Section 3 describes the calculation procedures used in more detail. Section 4 provides additional programming information.

Reference 1 gives a background discussion of the application of MFAULT to assist in performing a risk assessment. The identification and preliminary evaluation of the cut sets of large fault trees in reasonable amounts of computer time are required for this approach. Several analytical fault tree codes were available in mid-1974 when this work was being performed. However, it was concluded that none were capable of meeting all the necessary requirements. An early version of TREEL⁽⁴⁾/MOCUS was extensively modified to increase its capacity and perform probability calculations. In performing this modification, a more efficient (for large fault trees) upward working cut set algorithm was developed and incorporated into a code, FAULTG.⁽¹⁾

MFAULT is a refined version of FAULTG and its characteristics and operation are detailed in this report. Some familiarity with fault tree methodology is assumed. References 1, 5, and 6 will give the reader a general background in systems safety analysis and fault tree analysis. MFAULT can be used independently or as part of the complete risk analysis code package developed at PNL.

FIGURE 1. Risk Assessment Computer Package

2.0 INFORMATION FOR THE USER

2.1 GENERAL DESCRIPTION OF PROGRAM

MFAULT identifies the cut sets in a fault tree, calculates their probabilities, and screens them on the basis of specified cut-offs on probability and/or cut set length. The code consists of three main sections. The first transforms the input tree logic into an ordered form for further analysis and performs error checking. The second section is the algorithm to identify the individual cut sets in this ordered tree and to perform the specified degree of Boolean reduction. In the third section, the probability calculations are made.

MFAULT is based on an efficient upward-working algorithm for cut set identification. The probability calculations are based on the assumption of small probabilities and constant hazard rates (i.e., exponential failure distributions). Cut sets consisting of repairable components (basic events) only, non-repairable components only, or mixtures of both types can be evaluated. Components can be on-line or standby. Unavailability contributions from pre-existing failures, failures on demand, and testing and maintenance down-time can be handled.

MFAULT first processes those gates having only basic event inputs. As the code processes each gate on its way to the top event, checks are made at the user's option (1) for all duplicate or nonminimal cut sets, (2) of cut set length against the specified cut-off, and (3) of the calculated cut set probability against the specified cut-off. The gate name and each surviving cut set are stored on disk for use in upper levels of the tree. These steps are repeated until the top of the tree is reached.

The code eliminates a gate and its cut sets from mass storage when the gate is no longer needed, thus, resulting in efficient mass storage usage. Once the specified top of the tree is reached, the cut sets are printed and/or a file is created for further processing into RAFT.⁽³⁾ This upward algorithm results in reduced running time because duplicate cut sets and cut sets not surviving the cut-offs are discarded low in the tree and undergo no unneeded processing. A gate needs to be evaluated only once.

MFAULT has the following features and options: (1) an algorithm to estimate the number of binary indicated cut sets versus the number of events per cut set; (2) the capability of reading in the complete fault tree and then specifying any gate as the top gate to analyze; (3) the option of using a probability cut-off; (4) the option of using a cut-off on the number of events per cut set; (5) the option of recording the disposition of cut sets at each gate; (6) the option of printing cut sets and/or creating a file; (7) the option of creating a file of cut sets discarded at the top gate; (8) the option of checking for redundancy of cut sets; (9) the option of finding minimal cut sets; (10) the option of performing cut set probability calculations.

Fault trees with AND gates, OR gates, inhibit gates, on switches (houses) and off switches can be analyzed. The code is presently capable of finding up to ten event cut sets from a fault tree with 512 basic events and 400 gates. It is operational on the CONTROL DATA CYBER 74 computer. Running time is a function of tree size, structure, and the values of the cut-offs on probability and/or cut set length. Complex fault trees containing millions of cut sets have been evaluated in terms of the top few hundred or thousand cut sets. Computer requirements were on the order of 5 minutes on the CYBER 74 system.

2.2 INPUT DESCRIPTION FOR MFAULT

MFAULT input consists of the following groups:

Title Card	
Control Information	(*DATA)
Tree Description	(*TREE)
Component Failure Data	(*RATES).

The first card of the case is the title card and can contain 80 columns of case identification. The beginning of each group of information following the title is indicated by a special control card of the form *KEYWORD beginning in Column 1. The end of each group of information is indicated by a special control card END starting in Column 1. An example of the input deck is shown in Appendix A.

MFAULT requires that each logic gate and component (basic event) be given an alphanumeric name. To simplify cut set identification and handling, component (basic event) names must be one character and an integer presently not to exceed 512. Gate names are not restricted and can consist of up to six alphanumeric characters. One card is read for each logic gate in the tree. The card contains the gate name, type of gate (AND, OR), number of gates input to it, number of components input to it, and names of the gates and components. Each component and gate is internally indexed and placed in Fortran logic arrays.

Input Cards		
Variable	Columns	Description
(10A8)		
TITLE	1-80	Case Title. All 80 columns may be used.
(A1, A7, 9A8)		
CARD (1)	1	The character, *
CARD (2)	2-8	DATAbbb. Section of input which provides control information.
(A6, 4X, 6I5, E10.0, 2I5)		
ITOPTR	1-6	Name of gate which is the desired top of the tree for calculations. Default: logical top of the tree.

Variable	Columns	Description
JTMAX	11-15	Maximum cut set size to be identified (JTMAX presently limited to 10 or less)
IFP1	16-20	Cut set disposition table 0 no disposition information printed 1 cut set disposition table printed for each gate
IFP2	21-25	Maximum number of cut sets printed. If number of cut sets greater than, IFP2, no printing of cut sets.
IFP3	26-30	Minimal cut set calculation 0 minimal cut sets not determined 1 minimal cut sets determined
IFP4	31-35	Binary indicated cut set calculation -X print predicted cut sets for each gate by size up to size X for the entire tree 0 no print X print predicted cut sets for the specified top gate and/or the actual top gate by size up to size X (X presently limited to 20)
IYCODE	36-40	0 omit probability calculation 1 perform probability calculation
THRESH	41-50	Value of probability cut-off
IWEED	51-55	0 omit check for duplicate cut sets 1 perform check for duplicate cut sets
IPROBD	56-60	0 no file created 1 file of the top gate cut sets not surviving the cut-offs is created
(2A5, 5X, A8)		
NOGO	1-5	NOGO: edit data only DEFAULT: perform calculation
XPUNCH	6-10	PUNCH: punch cards for *TREE block in FORTRAN logical order DEFAULT: no cards punched

Variable	Columns	Description
NORATE	16-23	NOBRATES read only *RATE and END card for block *RATE DEFAULT: read block *RATE
(10A8)		
CARD (1)	1-8	ENDbbbb. End of control information.
(A1, A7, 9A8)		
CARD (1)	1	The character, *
CARD (2)	2-8	TREEbbb. Section of input which provides tree logic description
[A6, 2X, A3, 10X, 211, 2X, 6(A6, 2X), A6]		One card for each gate in the tree.
GATE (1, J)	1-6	Gate name
GATE (2, J)	9-11	Gate type (AND, ORb)
NGG (J)	22	Number of input gates
NCC (J)	23	Number of input components (basic events)
GATE (3, J)	26-31	Input gate or component name ^(a)
GATE (4, J)	34-39	Input gate or component name
GATE (5, J)	42-47	Input gate or component name
GATE (6, J)	50-55	Input gate or component name
GATE (7, J)	58-63	Input gate or component name
GATE (8, J)	66-71	Input gate or component name
GATE (9, J)	74-79	Input gate or component name
[A6, 2X, A3, 10X, 211, 2X, 6(A6, 2X), A6]		
GATE (1, J)	1-6	ENDbbbb. End of tree logic
(A1, A7, 9A8)		
CARD (1)	1	The character, *
CARD (2)	2-8	RATESbb. Section of input which provides com- ponent failure rate data
(A6, 21X, I3, 3E12.4)		One card for each component (basic event) in the tree

^(a) Note that component names must be one character and an integer presently not to exceed 512. Gate names are not restricted and can consist of up to six alphanumeric characters.

Variable	Columns	Description
NAM (J)	1-6	Component name
MODE (J)	28-30	0 repairable component 1 nonrepairable component
PROB (J)	31-42	Probability of component
UNAV (J)	43-54	Unavailability of component
FACTX	55-66	Multiplier for PROB (J) and UNAV (J) Default: 1
(A6, 21X, I3, 3E12.4)		
NAM(J)	1-6	ENDbbb. End of component failure rate data

2.3 OUTPUT FROM MFAULT

Output includes 1) the input control information and user options; 2) the fault tree logic description; 3) the predicted number of cut sets and their size for each gate of the fault tree (option); 4) the input probability data; 5) the ordered fault tree; 6) the ordered tree for the specified top; 7) the cut set disposition table for each gate (option); 8) the cut sets and their probabilities ordered by cut set length (option); and 9) a cross reference table of components and gates.

In addition, a binary tape or disk file (Tape 9) can be created which contains the gate name, the cut sets, and their probabilities in blocks of one hundred. This tape can be input into RAFT⁽³⁾ or used for further processing of the cut sets. A tape or disk file (Tape 11) of the top gate cut sets not surviving the cut-offs can also be created.

2.4 SAMPLE CASE

Figure 2 shows an example fault tree plotted by the ACORN⁽²⁾ program. Appendix A gives the MFAULT input information and the resulting output after processing the sample tree.

FIGURE 2. Example Fault Tree

3.0 PROCEDURES FOR SOLUTION

Brief descriptions of the procedures used by MFAULT as it processes a fault tree are given in the following five sections. Reference to the detailed example in Section 4.4 may be helpful to the reader.

3.1 FAULT TREE TRANSFORMATION

The fault tree transformation operations of MFAULT are revised versions of those in TREEL.⁽⁴⁾ As described in Section 2.2, the fault tree logic input to MFAULT consists of a card for each gate of the fault tree. Each card contains the gate name, the gate type (AND, OR), the number of gates input, the number of components (basic events) input, and the names of these gates and components. This information is converted into a logical numbering sequence from the bottom to the top of the tree. The input components are ordered and renumbered consecutively from one. The tree logic statements are then searched to identify gates with only components input. Then gates with previously defined gates and possibly additional components input are identified. These gates are numbered consecutively in the above order from 2001. This ordered tree is used in the cut set identification sections of MFAULT.

During this fault tree transformation process, error checking and preliminary fault tree calculations are performed. The predicted number of cut sets for each gate and their length are output along with a cross-reference table of components and gates. Both are helpful in fault tree simplification efforts and provide qualitative information with minimal computer time.

3.2 CUT SET IDENTIFICATION

Although developed independently, MFAULT is based on the same type of upward-working algorithm for cut set identification used in the MICSUP code.⁽⁴⁾ The MICSUP algorithm has been described by Chatterjee.⁽⁷⁾

The cut set identification algorithm starts with the lowest level gates identified in the fault tree transformation operations discussed in Section 3.1. For an AND gate the cut set is the combination of the input components and for an OR gate each input component forms an individual cut set. The resulting cut sets for each gate are stored and used as required as the process continues up through the fault tree. The cut sets for higher level AND gates are all the combinations of the input cut sets. The cut sets for higher level OR gates are simply the union of the input cut sets.

As each gate is processed, several operations may be performed. These can consist of (1) various degrees of Boolean absorption; (2) probability calculations; (3) screening procedures. Each of these are discussed in the following sections.

3.3 TREATMENT OF ABSORPTION

A fault tree is a binary model which is converted to a probabilistic model by use of Boolean algebra. Boolean algebra has the following three laws which affect the identity of fault tree cut sets:

$$X \cap X = X \quad (1)$$

$$X \cup X = X \quad (2)$$

$$X \cup (X \cap X) = X \quad (3)$$

Application of these laws to the fault tree cut sets results in the minimal cut sets for the fault tree. Analysis of a fault tree may result in cut sets with repeated events or in repeated cut sets. Application of equation (1) and equation (2) results in the removal of such events or cut sets. As discussed in detail in reference 1, the choice of using minimal or non-minimal cut sets, as represented by use of equation (3), depends on the particular application for the fault tree. The use of fault trees in risk analysis where both the probability and consequences of an event are considered may require that the type of absorption indicated by equation (3) not be performed.

As an example, consider the following cut sets: X_1 , X_1X_2 , X_1X_2 , $X_2X_3X_3$. Use of equation (2) results in the cut sets X_1 , X_1X_2 , X_2X_3 . Application of equation (3) gives the cut sets X_1 , X_2X_3 which are the minimal cut sets for this example. As explained above, certain fault tree applications may require the cut set X_1X_2 to be retained.

MFAULT always removes redundant events within a cut set and has the option of removing duplicate cut sets and determining minimal cut sets. When specifying these options, the user should recognize that absorption checking is typically responsible for a large fraction of the running time of fault tree computer codes. The methods for removing duplicate cut sets and determining minimal cut sets are given in the following sections.

3.3.1 Duplicate Cut Sets

When cut sets are identified at each gate, they are ordered by level (number of components in a cut set) and the components within a cut set are ordered by increasing index. MFAULT uses a hash addressing scheme to check for and remove duplicate cut sets. This operation is presently performed at each gate of the fault tree. (It may be advantageous to check for duplicates only at the top gate and a simple code modification will accomplish this.) The cut set component indices are added to get a hash table address. For each level a cut set must be compared with every cut set whose address is equal to see if it is a duplicate (typically 1 to 5 checks per 1000 cut sets). When cut sets are found with duplicate addresses, the cut sets are compared and either discarded as a duplicate cut set or stored in the hash table or its overflow table (bump table).

3.3.2 Minimal Cut Sets

Minimal cut sets are presently determined at the top gate of the fault tree. (It may be advantageous to check for minimal cut sets at each gate and a simple code modification will accomplish this.) The minimal cut set operation starts after the cut sets have been ordered (by level and component index) and duplicate cut sets have been removed (if desired). Each cut set is compared against all other cut sets of its level or greater and with initial and final indices in the same range. Cut sets which are found to be non-minimal are deleted. Use of the minimal cut set algorithm may require a large amount of computer time.

3.4 PROBABILITY CALCULATION

Fault trees are typically quantified in terms of the top event probability or unavailability or the individual cut set's probability or unavailability. The cut set probability, P_i , is the probability of the system failure (top event) occurring in time t and by means of cut set i . The cut set unavailability, Q_i , is the probability the system is down at any time due to its failing by cut set i . The probability or unavailability of the top event is conservatively approximated by the sum of the probabilities or unavailabilities of all the cut sets.

Appendix B gives a detailed discussion of the probability and unavailability calculational methods used in MFAULT. The equations used are based on the assumption of small probabilities and constant hazard rates (i.e., exponential failure distributions). Cut sets consisting of repairable components only, non-repairable components only, or mixtures of both types can be evaluated. Components can be on-line or standby. Unavailability contributions from pre-existing failures, failures on demand, and testing and maintenance down-time can be handled.

3.5 SCREENING PROCEDURES

MFAULT has the options of applying cut offs on the cut set length and the cut set probability. Judicious use of these cut offs reduces the calculational effort required to evaluate large fault trees with little loss of overall accuracy. Reference 1 discusses methods for deriving these cut offs for the application of fault trees to a risk assessment.

As the cut set is identified at each gate and after the elimination of duplicate events, a check is made of the cut set length with the input cut off value. Cut sets exceeding this value are dropped from the analysis. Next the probability calculation is performed and the cut set probability is compared with the input probability cut off. Cut sets lower than this value are dropped from the analysis. This process continues as each gate is processed and the top of the tree is reached. A minor complication in applying the probability cut off must be noted. When treating certain standby and online component configurations, it is possible for a partial

cut set to have a zero probability but the complete cut set to have a greater than zero probability. For this reason no cut set with zero probability is deleted until the top of the tree is reached.

4.0 ADDITIONAL PROGRAMMING INFORMATION

4.1 GENERAL INFORMATION

MFAULT is written in FORTRAN-IV. The program has been compiled and executed under the CONTROL DATA CYBER 74 SCOPE 3.4.2 operating system. Current program dimensions allow the code to load with under 130,000 octal words of memory.

4.1.1 External Files

The following logical units are used by MFAULT:

<u>Logical Unit</u>	<u>Use</u>
2	Punched card output of ordered tree
5	Input data cards
6	Output line printer
7	Scratch Disk file for storing cut sets (IDISK*)
8	Scratch Disk file for storing cut sets (IDISC*)
9	Scratch Disk file for storing cut sets; final output file (NEWGT*)
10	Scratch Disk file used for storing cut sets (IANDD*)
11	Disk file of cut sets dropped at the top gate
12	Scratch Disk file used for storing cut sets (JANDD*)
25	Scratch Disk file used for storing cut set records when hashing

4.1.2 Nonstandard Functions

Most of the nonstandard functions used by MFAULT are a result of the use of Record Manager. Record Manager is a series of CDC software routines which are used to facilitate input and output record handling. Record Manager calls used by MFAULT are given below:

* These are the initial names of the logical units. Logical units are swapped through the program. See Section 4.3.

FILEWA	-	call to set up a word addressable file
OPENM	-	opens word addressable file
GET	-	reads variable record from word addressable file
PUT	-	writes variable record from word addressable file
CLOSEM	-	closes word addressable file

Other nonstandard functions used by MFAULT include:

EOF	-	function to check end of file condition
SECOND	-	function which returns elapsed time
SHIFT	-	function which accesses a single character per word
LOCF	-	function which assesses relative locations of program variable

Each of the nonstandard functions will have to be replaced by the equivalent function on the user's system.

4.2 SUBROUTINE DESCRIPTIONS

Figure 3 shows the fourteen subroutines used in MFAULT and the order in which they are used. MFAULT can be conveniently divided into two major branches. AOGATE and its related subroutines convert the input tree into an ordered form for computer analysis and provide preliminary information on the size and number of cut sets. This section of MFAULT is a revised version of TREEL⁽⁴⁾. MIXEM and its selected subroutines identify the cut sets or minimal cut sets of the fault tree, calculate the probabilities, and screen the cut sets on the basis of specified cutoffs on cut set length and/or probability.

The functions of the subroutines used in MFAULT are described below. Flow charts are given for the more important subroutines.

MFAULT is the executive program which reads input, performs input checking, and controls the sequence of fault tree calculations. A flow chart of MFAULT is given in Figure 4.

FIGURE 3. Subroutines used in MFAULT

FIGURE 4. Flow Chart for MFAULT

BOOB prints out error messages.

AOGATE builds the ordered FORTRAN tree logic.

DNA calculates the number and size of Binary Indicated Cut Sets (BICS) for AND gates.

RO calculates the number and size of BICS for OR gates.

FIXUP constructs the cross reference table between the input tree and the ordered FORTRAN tree.

MIXEM controls the flow of cut set calculations and prints out the cut sets. A flow chart for MIXEM is given in Figure 5.

ORGT determines cut sets for OR gates. A flow chart for ORGT is given in Figure 6.

ANDGT determines cut sets for AND gates, orders events in ascending order, and applies the cut set length cutoff. It builds the file of cut sets dropped at the top gate. A flow chart for ANDGT is given in Figure 7.

ARRANG orders cut sets by increasing length. It constructs the cut set disposition table. A flow chart for ARRANG is given in Figure 8.

MINIMAL determines minimal cut sets. A flow chart for MINIMAL is given in Figure 9.

WEED checks for duplicate cut sets. A flow chart for WEED is given in Figure 10.

REPAIR performs probability calculations and applies the probability cutoff.

XREF constructs the component-gate cross reference table.

FIGURE 5. Flow Chart for MIXEM

FIGURE 6. Flow Chart for ORGT

FIGURE 7. Flow Chart for ANDGT

FIGURE 8. Flow Chart for ARRANG

FIGURE 9. Flow Chart for MINIMAL

FIGURE 10. Flow Chart for WEED

4.3 USE OF SCRATCH STORAGE IN MFAULT

MFAULT is designed to analyze large fault trees. Extensive use of scratch storage (disk) is used by MFAULT to facilitate the identification and storage of large numbers of cut sets. As the cut sets from each gate are identified, they are written in blocks of 100 to await further processing. All of the disk used for scratch storage (IDISK, IDISC, IANDD, JANDD, NEWGT) have the same format. They are blocked in records of 100. Each record contains the following:

- WORD 1 - Alphanumeric name of the gate
- WORD 2 - Integer number of records in the block
- WORD 3 - Integer level (number of components) for the cut set
- WORD 4 - Probability value for the cut set
- WORD 5-14 - Integer index of the components that make up the cut set
- WORD 15-END - WORDS 3-14 are repeated WORD 2 times to complete the block

To save computer storage and time, a gate is dropped from scratch storage when the highest reference (i.e., last use) of the gate in the tree structure is reached. When the top of the tree is reached only one gate resides on scratch storage, the top gate.

Disk units are swapped throughout the execution of MFAULT. All input to the cut set calculation steps are read from disk IDISK. The new unordered cut sets are written to disk NEWGT. All existing and new ordered cut sets are written to IDISC. IDISK and IDISC are swapped at the beginning of each gate calculation (subroutine MIXEM). Other unit swapping occurs in subroutine ANDGT (NEWGT and IANDD) and subroutine MINIMAL (IANDD and JANDD; IDISK and IDISC). When the cut set calculation steps are completed, disk IDISK is copied to disk NEWGT, which is set to unit 9. The information on unit 9 can be saved as permanent storage for use in a computer program such as RAFT⁽³⁾.

4.4 EXAMPLE OF CUT SET IDENTIFICATION PROCEDURE

To illustrate the calculational methods used by MFAULT, an example is given using a branch of the sample fault tree (Section 2.4) shown in Figure 11.

By inspection the following cut sets are obtained:

X29 X30 X29 X29 X31
X29 X30 X31 X29 X31
X29 X30 X32 X29 X31
X29 X30 X29 X30
X29 X30 X31 X30
X29 X30 X32 X30
X29 X30 X29 X32
X29 X30 X31 X32
X29 X30 X32 X32

Further reduction of duplicate events within a cut set results in:

X29 X30 X31
X29 X30 X31
X29 X30 X31 X32
X29 X30
X29 X30 X31
X29 X30 X32
X29 X30 X32
X29 X30 X31 X32
X29 X30 X32

Elimination of duplicate cut sets gives:

X29 X30 X31
X29 X30 X31 X32
X29 X30
X29 X30 X32

Further reduction to minimal cut sets results in:

X29 X30

FIGURE 11. Example Fault Tree Branch

The procedures used by MFAULT to solve the above problem are outlined in this section. First the tree input is ordered and each gate and component (basic event) are given a cross reference index. The last reference for each gate is determined and the following ordered tree is obtained:

(2001)	A22	AND	(1) X29	(3) X31	
(2002)	A20	OR	(1) X29	(3) X31	(4) X32
(2003)	A19	AND	(1) X29	(2) X30	
(2004)	A21	OR	(2001) A22	(2) X30	(4) X32
(2005)	A18	AND	(2003) A19	(2002) A20	(2004) A21

TOP = (2005) A18

In this example, all gates below index 2003 are component only gates (only basic events are input) and are not processed until needed. Gate processing begins at gate 2004 (A21) and continues to the top of the tree.

Since gate A21 is an OR gate and has two components among its inputs, two cut sets of level one are produced:

X30

X32

The other input to gate A21 is gate A22 which is a component only AND gate. One additional cut set is produced:

X29 X31

If probability calculations are desired, they are done at this point and those cut sets less than the specified cutoff will be dropped. Also if additional previously defined gates had been input to gate A21, disk IDISK would have been searched for those gates and their corresponding cut sets and probabilities. These cut sets would be simply appended to the list of cut sets just calculated to complete the total list of cut sets for this gate.

All cut sets for A21 have now been identified and are written to disk NEWGT. NEWGT is read and the cut sets are ordered by increasing level. At this point, at the user's option, duplicate cut sets are eliminated from the NEWGT disk (subroutine WEED). Gate A21 and its ordered cut sets are appended to disk IDISC.

The next gate to be processed is gate A18. Inputs to gate A18 consist of gates A19, A20, and A21. Since gate A21 is a previously processed input gate, disk IDISK is searched to find gate A21 and its corresponding cut sets. The list of cut sets from gate A21 is written to disk IANDD. Since this is its last reference, A21 cut sets are not written to disk IDISC.* The component only input gates to gate A18 are processed next. The IANDAL table is constructed for gate A20.

partial cut set number			
level	1	2	3
1	X29	X31	X32

Since gate A20 is an OR gate three partial cut sets of level one are produced. The final component only input gate to A18 is gate A19. Gate A19 is a component only AND gate. One cut set is produced:

X29 X30

This cut set is logically combined with the existing partial cut sets in the IANDAL table and results in:

partial cut set number			
level	1	2	3
1	X29	X31	X32
2	X29	X29	X29
3	X30	X30	X30

At this point all partial cut sets from disk IANDD and the IANDAL table are logically combined to obtain the cut sets from gate A18.

*For the more general case for a particular gate, disk IDISK is searched to find the previously processed input gates of interest. If not the last reference, the input gate and its cut sets are written to disk IDISC. If it is the first previously processed input gate, the input gate and its cut sets are written to disk IANDD. Additional previously processed input gates are written to disk IANDD.

The ordering of components and checking for duplicate components within a cut set are the next operations performed. As each cut set is generated, its indices are placed into an array (ICINDX) which initially contains all zero values. Nonzero positions now correspond to the component indices. Since all duplicate components are placed in the same position, the duplicate component reduction is automatic. The nonzero locations of the array are then checked to generate the ordered cut set. The cut set level is determined by simply counting the nonzero values. The cut sets are written to disk NEWGT. The cut sets for gate A18 are shown below:

```

X29 X30
X29 X30 X31
X29 X30 X32
X29 X30 X32
X29 X30 X31 X32
X29 X30 X32
X29 X30 X31
X29 X30 X31
X29 X30 X31 X32

```

If the probability calculations are desired, they are done at this point and those cut sets less than the specified cutoff will be dropped.

The next step is to arrange the cut sets in ascending order and to eliminate duplicate cut sets, at the user's option. Disk NEWGT is read and the cut sets are ordered by increasing level. At this point, duplicate cut sets are eliminated from the NEWGT disk (subroutine WEED). This results in the following cut sets which are written to disk IDISC.

```

X29 X30
X29 X30 X31
X29 X30 X32
X29 X30 X31 X32

```

The final step in the cut set calculation is the reduction to minimal cut sets (user's option). These operations are performed by the subroutine MINIMAL. For gate A18 one cut set is output:

X29 X30

REFERENCES

1. T. H. Smith, P. J. Pelto, D. L. Stevens, G. D. Seybold, W. L. Purcell, and L. V. Kimmel, A Risk-Based Fault Tree Analysis Method for Identification, Preliminary Evaluation, and Screening of Potential Accidental Release Sequences in Nuclear Fuel Cycle Operations, BNWL-1959, Battelle, Pacific Northwest Laboratories, Richland, WA, January 1976.
2. J. L. Carter, ACORN: A Computer Program for Plotting Fault Trees, BNWL-2144, Battelle, Pacific Northwest Laboratories, Richland, WA, November 1977.
3. G. D. Seybold, RAFT: A Computer Program for Fault Tree Risk Calculations, BNWL-2146, Battelle, Pacific Northwest Laboratories, Richland, WA, November 1977.
4. P. K. Pande, M. E. Spector and P. Chatterjee, Computerized Fault Tree Analysis: TREEL and MICSUP, Publication ORC75-3, Operations Research Center, University of California, Berkeley, CA, April 1975.
5. H. E. Lambert, Systems Safety Analysis and Fault Tree Analysis, UCID-16238, Lawrence Livermore Laboratory, University of California, May 1973.
6. USAEC, Reactor Safety Study, an Assessment of Accident Risks in Commercial Nuclear Power Plants, WASH-1400, October 1975.
7. P. Chatterjee, Fault Tree Analysis: Min Cut Set Algorithms, AD 774 100, University of California, Berkeley, CA, January 1974.

APPENDIX A

SAMPLE CASE INPUT AND OUTPUT INFORMATION

MFAULT Input

TESTS, T100, CM130000,
ACCOUNT (PW=RCC039, UN=RCC039)
ATTACH (FAULT, MFAULT, ID=RNWLP)
RFL (130000)
FAULT.

PELTO ROR PH 2752

SMALL TEST TRFE

*DATA

A23 10 1 500 1 -15 1 0. 1 0

FND

*TRFF.

	OR	61	A1	A2	A3	A4	A5	A6	X10
A23	AND	11	A7	X11					
A2	OR	40	A8	A17					
A3	AND	03	X1	X2	A12	A18			
A4	AND	03	X4	X5	X3				
A5	AND	03	X7	X8	X6				
A6	AND	11	A13	X14	X9				
A7	AND	02	X12	X13					
A8	AND	02	X10	X26					
A9	AND	11	A14	X17					
A10	AND	11	A15	X18					
A11	AND	12	A16	X22	X23				
A12	AND	02	X11	X25					
A13	AND	02	X15	X16					
A14	AND	02	X21	X27					
A15	AND	02	X19	X20					
A16	AND	02	X24	X28					
A17	AND	30	A9	A10	A11				
A18	AND	30	A19	A20	A21				
A19	AND	02	X29	X30					
A20	OR	03	X29	X31	X32				
A21	OR	12	A22	X30	X32				
A22	AND	02	X29	X31					

FND
*RATES

X1	1	.02	.02
X2	1	.03	.03
X3	1	.05	.05
X4	0	.07	.009
X5	0	.03	.008
X6	0	.09	.02
X7	1	.07	.07
X8	0	.01	.002
X9	1	.007	.007
X10	1	.0003	.0003
X11	1	.009	.009
X12	0	.06	.008

X13	0	.0	.009
X14	0	.02	.007
X15	0	.009	.003
X16	0	.0	.003
X17	0	.04	.007
X18	1	.001	.001
X19	1	.02	.02
X20	1	.0	.008
X21	0	.01	.002
X22	1	.008	.008
X23	0	.0	1.
X24	1	.009	.009
X25	0	.03	.008
X26	0	.03	.008
X27	0	.0	.2
X28	0	.0	.07
X29	1	.02	.02
X30	0	.03	.008
X31	1	.007	.007
X32	1	.07	.07

FND

```

*****
*
*  A PROGRAM FOR ANALYZING FAULT TREES
*
*****

```

SMALL TEST TREE

DATA INPUT LIST

```

TOP      = A23
JIMAX    = 10
IFP1     = 1
IFP2     = 500
IFP3     = 1
IFP4     = -15
IYCODE   = 1
THRESH   = 0.
IWEED    = 1
IPROB    = 0

```

THE FOLLOWING GATE TYPES ARE ALLOWED
AND
OR

 *
 FAULT A PROGRAM FOR ANALYZING FAULT TREES
 *

SMALL TEST TREE

LISTING OF INPUT TREE LOGIC

NAME	TYPE	6	1	A1	INPUTS----	A3	A4	A5	A6	X10
A23	OR			A2						
A1	AND	1	1	A7	X11					
A2	OR	4	0	A8	A17					
A3	AND	0	3	X1	X2	A12				
A4	AND	0	3	X4	X5	X3				
A5	AND	0	3	X7	X3	X6				
A6	AND	1	1	A13	X14	X9				
A7	AND	0	2	X12	X13					
A8	AND	0	2	X10	X26					
A9	AND	1	1	A14	X17					
A10	AND	1	1	A15	X18					
A11	AND	1	2	A16	X22					
A12	AND	0	2	X11	X25	X23				
A13	AND	0	2	X15	X16					
A14	AND	0	2	X21	X27					
A15	AND	0	2	X19	X20					
A16	AND	0	2	X24	X28					
A17	AND	3	0	A9	A10					
A18	AND	3	0	A19	A20	A11				
A19	AND	0	2	X29	X30	A21				
A20	OR	0	3	X29	X31					
A21	OR	1	2	A22	X30	X32				
A22	AND	0	2	X29	X31					
END		-0-0								

RETURN

END

THERE ARE 32 COMPONENTS INDEXED IN THIS TREE

NUMBER OF BINARY INDICATED CUT SETS IN FAULT TREE 18
 MAX SIZE OF BINARY INDICATED CUT SET IN FAULT TREE 10

NUMBER AND LENGTH OF BINARY INDICATED CUTSETS

LENGTH	2001 A22	2002 A20	2003 A19	2004 A16	2005 A15	2006 A14	2007 A13	2008 A12	2009 A8	2010 A7
1	0	3	0	0	0	0	0	0	0	0
2	1	0	1	1	1	1	1	1	1	1
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0
10	1	3	1	1	1	1	1	1	1	1

LENGTH	2011 A5	2012 A4	2013 A3	2014 A21	2015 A18	2016 A11	2017 A10	2018 A9	2019 A6	2020 A1
1	0	0	0	2	0	0	0	0	0	0
2	0	0	0	1	0	0	0	0	0	0
3	1	1	1	0	0	0	1	1	1	1
4	0	0	0	0	6	1	0	0	0	0
5	0	0	0	0	3	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0
10	1	1	1	3	9	1	1	1	1	1

LENGTH	2021 A17	2022 A2	2023 A23
1	0	0	1
2	0	2	2
3	0	0	5
4	0	6	6
5	0	3	3
6	0	0	0
7	0	0	0
8	0	0	0
9	0	0	0
10	1	1	1
10	1	12	18

```

*****
*
* INFALTY A PROGRAM FOR ANALYZING FAULT TREES
*
*****

```

SMALL TEST TREE

LISTING OF INPUT PROBABILITY DATA

TREE INDEX	COMPONENT NAME	MODE	PROBABILITY	UNAVAILITY
1	x1	1	.020000000	.020000000
2	x2	1	.030000000	.030000000
3	x3	1	.050000000	.050000000
4	x4	0	.025000000	.003330000
5	x5	0	.030000000	.008000000
6	x6	0	.090000000	.020000000
7	x7	1	.070000000	.070000000
8	x8	0	.010000000	.002000000
9	x9	1	.007000000	.007000000
10	x10	1	.000300000	.000300000
11	x11	1	.000000000	.009000000
12	x12	0	.060000000	.008000000
13	x13	0	.000000000	.009000000
14	x14	0	.020000000	.007000000
15	x15	0	.000000000	.003000000
16	x16	0	.000000000	.003000000
17	x17	0	.040000000	.007000000
18	x18	1	.001000000	.001000000
19	x19	1	.020000000	.020000000
20	x20	1	.000000000	.008000000
21	x21	0	.010000000	.002000000
22	x22	1	.000000000	.008000000
23	x23	0	.000000000	1.
24	x24	1	.000000000	.009000000
25	x25	0	.030000000	.008000000
26	x26	0	.030000000	.006000000
27	x27	0	.000000000	.200000000
28	x28	0	.000000000	.070000000
29	x29	1	.020000000	.020000000
30	x30	0	.030000000	.008000000
31	x31	1	.007000000	.007000000
32	x32	1	.070000000	.070000000

 *
 MFAULT A PROGRAM FOR ANALYZING FAULT TREES
 *

SMALL TEST TREE

ORDERED TREE LOGIC STRUCTURE

(2001) A22	= AND C (29) X29	(31) X31	(32) X32	
(2002) A20	= OR C (29) X29	(31) X31		
(2003) A19	= AND C (29) X29	(30) X30		
(2004) A16	= AND C (24) X24	(28) X28		
(2005) A15	= AND C (19) X19	(20) X20		
(2006) A14	= AND C (21) X21	(27) X27		
(2007) A13	= AND C (15) X15	(16) X16		
(2008) A12	= AND C (11) X11	(25) X25		
(2009) A8	= AND C (10) X10	(26) X26		
(2010) A7	= AND C (12) X12	(13) X13		
(2011) A5	= AND C (7) X7	(8) X8	(9) X9	
(2012) A4	= AND C (4) X4	(5) X5	(6) X6	
(2013) A3	= AND C (1) X1	(2) X2	(3) X3	
(2014) A21	= OR C (2001) A22	(30) X30	(32) X32	
(2015) A18	= AND C (2003) A19	(2002) A20	(2014) A21	
(2016) A11	= AND C (2004) A16	(22) X22	(23) X23	
(2017) A10	= AND C (2005) A15	(14) X14		
(2018) A9	= AND C (2006) A14	(17) X17		
(2019) A6	= AND C (2007) A13	(14) X14		
(2020) A1	= AND C (2010) A7	(11) X11		
(2021) A17	= AND C (2018) A9	(2017) A10	(2015) A18	
(2022) A2	= OR C (2009) A8	(2021) A17	(2012) A4	
(2023) A23	= OR C (2020) A1	(2022) A2	(2013) A3	
TOP = (2023) A23			(2011) A5	
			(2019) A6	
			(10) X10	

.....
 FAULT A PROGRAM FOR ANALYZING FAULT TREES

SMALL TEST TREE

CUT SET DISPOSITION TABLE

OR GATE A21	INPUTS A22	X30	X32		.021 SECONDS USED	1.752 TIME
LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4	LEVEL 5	LEVEL 6	LEVEL 7
LEVEL 8	LEVEL 9	LEVEL 10				
MAD	0	0	0	0	0	0
PROB	0	0	0	0	0	0
DUP	0	0	0	0	0	0
MIN	0	0	0	0	0	0
MAX	2	1	0	0	0	0
TOTAL CUTSETS RETAINED	3	CUTSETS RELEASED--LARGER THAN JIMAX(10)	0	PROBABILITY	0	DUPLICATION
					0	NON MINIMAL
						0
AND GATE A18	INPUTS A19	A20	A21		.044 SECONDS USED	1.810 TIME
LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4	LEVEL 5	LEVEL 6	LEVEL 7
LEVEL 8	LEVEL 9	LEVEL 10				
MAD	0	0	0	0	0	0
PROB	0	0	0	0	0	0
DUP	0	0	0	0	0	0
MIN	0	0	0	0	0	0
MAX	0	1	2	1	0	0
TOTAL CUTSETS RETAINED	4	CUTSETS RELEASED--LARGER THAN JIMAX(10)	0	PROBABILITY	0	DUPLICATION
						5
						NON MINIMAL
						0
AND GATE A11	INPUTS A14	X22	X23		.019 SECONDS USED	1.844 TIME
LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4	LEVEL 5	LEVEL 6	LEVEL 7
LEVEL 8	LEVEL 9	LEVEL 10				
MAD	0	0	0	0	0	0
PROB	0	0	0	0	0	0
DUP	0	0	0	0	0	0
MIN	0	0	0	0	0	0
MAX	0	0	0	1	0	0
TOTAL CUTSETS RETAINED	1	CUTSETS RELEASED--LARGER THAN JIMAX(10)	0	PROBABILITY	0	DUPLICATION
						0
						NON MINIMAL
						0
AND GATE A10	INPUTS A15	X18			.021 SECONDS USED	1.879 TIME
LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4	LEVEL 5	LEVEL 6	LEVEL 7
LEVEL 8	LEVEL 9	LEVEL 10				
MAD	0	0	0	0	0	0
PROB	0	0	0	0	0	0
DUP	0	0	0	0	0	0
MIN	0	0	0	0	0	0
MAX	0	0	0	0	0	0
TOTAL CUTSETS RETAINED	1	CUTSETS RELEASED--LARGER THAN JIMAX(10)	0	PROBABILITY	0	DUPLICATION
						0
						NON MINIMAL
						0
AND GATE A9	INPUTS A14	X17			.024 SECONDS USED	1.917 TIME
LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4	LEVEL 5	LEVEL 6	LEVEL 7
LEVEL 8	LEVEL 9	LEVEL 10				
MAD	0	0	0	0	0	0
PROB	0	0	0	0	0	0
DUP	0	0	0	0	0	0
MIN	0	0	0	0	0	0
MAX	0	0	0	0	0	0
TOTAL CUTSETS RETAINED	1	CUTSETS RELEASED--LARGER THAN JIMAX(10)	0	PROBABILITY	0	DUPLICATION
						0
						NON MINIMAL
						0
AND GATE A6	INPUTS A13	X14			.027 SECONDS USED	1.958 TIME
LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4	LEVEL 5	LEVEL 6	LEVEL 7
LEVEL 8	LEVEL 9	LEVEL 10				
MAD	0	0	0	0	0	0
PROB	0	0	0	0	0	0
DUP	0	0	0	0	0	0
MIN	0	0	0	0	0	0
MAX	0	0	0	0	0	0
TOTAL CUTSETS RETAINED	1	CUTSETS RELEASED--LARGER THAN JIMAX(10)	0	PROBABILITY	0	DUPLICATION
						0
						NON MINIMAL
						0
AND GATE A1	INPUTS A7	X11			.020 SECONDS USED	1.991 TIME
LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4	LEVEL 5	LEVEL 6	LEVEL 7
LEVEL 8	LEVEL 9	LEVEL 10				
MAD	0	0	0	0	0	0
PROB	0	0	0	0	0	0
DUP	0	0	0	0	0	0
MIN	0	0	0	0	0	0
MAX	0	0	0	0	0	0
TOTAL CUTSETS RETAINED	1	CUTSETS RELEASED--LARGER THAN JIMAX(10)	0	PROBABILITY	0	DUPLICATION
						0
						NON MINIMAL
						0
AND GATE A17	INPUTS A9	A10	A11		.051 SECONDS USED	2.055 TIME
LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4	LEVEL 5	LEVEL 6	LEVEL 7
LEVEL 8	LEVEL 9	LEVEL 10				
MAD	0	0	0	0	0	0
PROB	0	0	0	0	0	0
DUP	0	0	0	0	0	0
MIN	0	0	0	0	0	0
MAX	0	0	0	0	0	0
TOTAL CUTSETS RETAINED	1	CUTSETS RELEASED--LARGER THAN JIMAX(10)	0	PROBABILITY	0	DUPLICATION
						0
						NON MINIMAL
						0
OR GATE A2	INPUTS A4	A17	A12	A18	.034 SECONDS USED	2.103 TIME
LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4	LEVEL 5	LEVEL 6	LEVEL 7
LEVEL 8	LEVEL 9	LEVEL 10				
MAD	0	0	0	0	0	0
PROB	0	0	0	0	0	0
DUP	0	0	0	0	0	0
MIN	0	0	0	0	0	0
MAX	0	1	2	1	0	0
TOTAL CUTSETS RETAINED	7	CUTSETS RELEASED--LARGER THAN JIMAX(10)	0	PROBABILITY	0	DUPLICATION
						0
						NON MINIMAL
						0
OR GATE A23	INPUTS A1	A2	A3	A4	A5	A6
LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4	LEVEL 5	LEVEL 6	LEVEL 7
LEVEL 8	LEVEL 9	LEVEL 10				
MAD	0	0	0	0	0	0
PROB	0	0	0	0	0	0
DUP	0	0	0	0	0	0
MIN	0	0	0	0	0	0
MAX	0	1	2	1	0	0
TOTAL CUTSETS RETAINED	9	CUTSETS RELEASED--LARGER THAN JIMAX(10)	0	PROBABILITY	0	DUPLICATION
						0
						NON MINIMAL
						4

TOTAL PSEUDO = 0

CUTSETS AT LEVEL 1 A23

PROBABILITY CUT SFT

1 3.0000E-04 10

CUTSETS AT LEVEL 2 A23

PROBABILITY CUT SFT

1 2.0777E-04 11 25
2 4.6600E-04 29 36

CUTSETS AT LEVEL 3 A23

PROBABILITY CUT SFT

1 2.6100E-04 7 8 9
2 4.5344E-04 4 5 6
3 3.0000E-05 1 2 3
4 3.8277E-07 14 15 16
5 3.0740E-04 11 12 13

CUTSETS AT LEVEL 10 A23

PROBABILITY CUT SFT

1 5.8363E-18 17 18 19 20 21 22 23 24 27 28

THE TOTAL PROBABILITY EQUALS 1.0116E-03

```

*****
*
* MFAULT A PROGRAM FOR ANALYZING FAULT TREES *
*
*****

```

SMALL TEST TREE

CROSS REFERENCE TABLE OF COMPONENTS AND GATES

TREE INDEX	COMPONENT NAME	NUMBER OF GATES INPUT	GATES INPUT BY THIS COMPONENT
1	X1	1	A3
2	X2	1	A3
3	X3	1	A3
4	X4	1	A4
5	X5	1	A4
6	X6	1	A4
7	X7	1	A5
8	X8	1	A5
9	X9	1	A5
10	X10	2	A8
11	X11	2	A12
12	X12	1	A7
13	X13	1	A7
14	X14	1	A6
15	X15	1	A13
16	X16	1	A13
17	X17	1	A9
18	X18	1	A10
19	X19	1	A15
20	X20	1	A15
21	X21	1	A14
22	X22	1	A11
23	X23	1	A11
24	X24	1	A16
25	X25	1	A12
26	X26	1	A8
27	X27	1	A14
28	X28	1	A16
29	X29	3	A22
30	X30	2	A21
31	X31	2	A22
32	X32	2	A21
			A20
			A19
			A20
			A20
			A23
			A1

APPENDIX B

EQUATIONS FOR EVALUATING CUT SET AND TOP EVENT PROBABILITIES

APPENDIX B⁽¹⁾

EQUATIONS FOR EVALUATING CUT SET AND TOP EVENT PROBABILITIES

The material in this appendix is from Reference 1. The probability equations used in the fault tree analysis code MFAULT are described. The equations assume independence of basic events within a cut set. Modifications for the case of dependent events are given in Reference 1. Other assumptions are discussed below.

B.1 INTRODUCTION

Two quantities are of prime interest in quantitative probability evaluation of fault trees. The first is the probabilities of certain individual cut sets. This information indicates the relative importance of cut sets in their contributions to the occurrence of the top event. The second quantity is the probability of the top event of the tree or of the major branch. The probability of the top event is conservatively approximated by the sum of the probabilities of all the cut sets.^(2,3,4) For rare events, the approximation is quite accurate.

The task of finding the second quantity of interest encompasses that of the first. The failure probability of each cut set is required. This is the probability that all basic events of a given cut set coexist at some time during the mission period.

The subroutine MIXEM and its supporting subroutines identify all the cut sets of a fault tree or major branch which survive the chain length cutoff. These cut sets may be minimal or nonminimal. The choice depends on the user's application and is discussed in Section 3.3. The cut sets will have the following appearance: X_1 , X_3X_5 , $X_9X_{11}X_{42}$, etc. Each X_i is a basic event (circle or diamond), inhibit condition, house (on-switch), or off-switch. It is also assumed that suitable data or estimates are available for the basic events, inhibit conditions, etc.

Equations to evaluate the failure probabilities of cut sets are discussed in B.2. Much of the material is drawn from Reference 2. Discussions

there regarding basic assumptions are generally not repeated. B.3 applies the equations in a form suitable for computer use.

B.2 EQUATIONS FOR EVALUATION OF CUT SET PROBABILITY

B.2.1 All Nonrepairable Components

If all of the n components in a cut set are nonrepairable, P (defined as the probability of failure for that cut set in the time interval from zero to t) is given by the exponential distribution

$$P = (1 - e^{-\lambda_1 t})(1 - e^{-\lambda_2 t}) \dots (1 - e^{-\lambda_n t})$$

where λ_i = failure rate of component i , assumed constant in the time interval from zero to t (failures are also considered random and independent).

t = operating time or mission time.

If $\lambda t \ll 1$, then $(1 - e^{-\lambda t})$ can be replaced by λt with very little error. The small probability approximation, which will be used throughout the remainder of this Appendix, is always conservative. The above equation becomes

$$P = \lambda_1 t \lambda_2 t \dots \lambda_n t. \quad (1)$$

For computing P , all one-component cut sets are considered to be non-repairable. The computing scheme of Section B.3.2 automatically accomplishes this.

B.2.2 All Repairable Components

If all (or even several) of the n components in a cut set are repairable, a general solution is not practical because of complexities involving maintenance schedules, backup configurations, etc. However, a simplified treatment for cut set failure probability, assuming all $\lambda_{o_i} t \ll 1$ and yielding steady-state, asymptotic values, uses

$$\begin{aligned} P = & Q_2 Q_3 \dots Q_n \lambda_{o_1} t + Q_1 Q_3 \dots Q_n \lambda_{o_2} t \\ & + \dots + Q_1 Q_2 \dots Q_{n-1} \lambda_{o_n} t \end{aligned} \quad (2)$$

where

λ_{oi} = operating failure rate of component i

Q_i = unavailability of component i, defined as the probability of being in a failed state when required, or as the expected fraction of the time that the component is down. Values of Q_i for Equation (2) are given in Equations (3) and (4).

It is assumed that the Q_i are independent and that not all components in a cut set can be simultaneously disabled for maintenance and/or testing.

For an on-line component, one whose fault is typically the event

$$Q_i = \lambda_{oi} \tau_i + \frac{t_{Ti}}{t_{TIi}} + \frac{t_{Mi}}{t_{MIi}} \text{ for } \lambda_{oi} \tau_i \ll 1 \quad (3)$$

where

λ_{oi} = operating failure rate

$\tau_i = \tau_{Di} + \tau_{Ri}$ = average duration time of the fault

τ_{Di} = average time to detect failure*

*If the component is not monitored, but is periodically tested, $\tau_D = t_{TI}/2$.
If the component is monitored (annunciated), as well as periodically tested, the following expression⁽⁵⁾ is recommended:

$$\tau_D = \tau_{DG} Q_G + \tau_{DNG} (1 - Q_G)$$

where

τ_{DG} = average time to detect component failure if the monitor functions

Q_G = availability of the monitor

τ_{DNG} = average time to detect component failure if the monitor does not function.

In many cases, $\tau_{DNG} = t_{TI}/2$.

τ_{R_i} = average time to repair a failure, once detected

t_{T_i} = average test downtime (some components can be tested on-line, in which case $t_{T_i} = 0$)

$t_{T I_i}$ = average interval between tests

t_{M_i} = average maintenance downtime

$t_{M I_i}$ = average maintenance interval.

Here and in Equation (3) the timing of maintenance and of testing among components in the same cut set is considered random, so that the Q_i are independent.

A standby component is one which is not on-line (and generally not operating) until demanded. All repairable components can be considered either on-line or standby, but not both. Thus the two categories are all-encompassing as well as mutually exclusive. For a standby component, one whose fault is typically the event

Q_i comprises contributions originating prior to the demand (uncorrected failures; test; and maintenance) and those occurring at the demand (startup transients, etc.).

$$Q_i = \lambda_{S_i} \tau_i + Q_{d_i} + \frac{t_{T_i}}{t_{T I_i}} + \frac{t_{M_i}}{t_{M I_i}} \text{ for } \lambda_{S_i} \tau_i \ll 1 \quad (4)$$

where τ_i , t_{T_i} , $t_{T I_i}$, t_{M_i} , and $t_{M I_i}$ are as used previously,

λ_{S_i} = standby failure rate (related to a standby component failing before it is demanded). λ_{S_i} may or may not equal λ_{O_i} . If the standby component is kept operating (but not on-line), λ_{S_i} often equals λ_{O_i} .

Q_{d_i} = the probability of component i failing to function on demand,
given that it was capable of functioning immediately prior
to the demand.*

B.2.3 Additional Considerations

Equations (1) - (4) are the basic results for evaluation of cut sets. However, several considerations, discussed below, necessitate adjustments to these equations.

B.2.3.1 Special Considerations for Standby Components

Consider a two-member cut set X_1X_2 , where component 1 is on-line and component 2 is its standby. The failure probability of cut set X_1X_2 consists of two terms: The first term represents the probability of (on-line) component 1 failing during the mission multiplied by the probability that (standby) component 2 is unavailable when demanded. The second term consists of three factors representing the situation in which component 1 fails, then component 2 comes on-line but fails before component 1 is repaired. The failure probability of cut set X_1X_2 can be shown to be

$$P = Q_2 \lambda_{o_1} t + (\lambda_{o_1} t)(1-Q_2)(\lambda_{o_2} \tau_1)$$

if $\lambda_{o_1} t$ and $\lambda_{o_1} \tau_1$ are both $\ll 1$. Assuming $(1-Q_2) \approx 1$ and rearranging,

$$P = Q_2 \lambda_{o_1} t + Q_1 \lambda_{o_2} t$$

where

Q_2 is given by Equation (4)

and $Q_1 = \lambda_{o_1} \tau_1$.

The result is identical to that using Equation (2) with Q_1 input as $\lambda_{o_1} \tau_1$.

We neglect maintenance and testing contributions to Q_1 in the above equation for P because it is assumed that the online component will not be maintained until its standby is brought on line. It is also assumed that

*In many data sources, $\lambda_{s_i} \tau_i$ and Q_{d_i} are combined.

an on-line component and its standby are always connected by an AND gate and therefore always appear together in a cut set. This avoids the possibility of Q_1 being called upon to assume different values in different cut sets.

B.2.3.2 Mixed Cut Sets

If m of the n components in a cut set are nonrepairable, Equation (2) is modified so that the factor $\frac{1}{m}$ ($m \neq 0$) multiplies each additive term whose $\lambda_{o_i} t$ represents a nonrepairable component.* Thus, in general, m of the n additive terms in Equation (2) would be multiplied by this factor. It can be shown that the remaining $n-m$ additive terms are each multiplied by the factor $\frac{1}{m+1}$. These $n-m$ terms are those whose $\lambda_{o_i} t$ represents a repairable component.

It is noted that the unavailability Q_i for a nonrepairable component i is $\lambda_i t$. Also, it is assumed that all $\tau_i \ll t$.

Example - given the cut set $X_1 X_2 X_3 X_4$
where X_1, X_3, X_4 are nonrepairable.

Thus $n = 4, m = 3$.

Equation (2) gives

$$P = Q_2 Q_3 Q_4 \lambda_1 t + Q_1 Q_3 Q_4 \lambda_2 t + Q_1 Q_2 Q_4 \lambda_3 t + Q_1 Q_2 Q_3 \lambda_4 t.$$

This result is adjusted by $\frac{1}{m} = \frac{1}{3}$ for terms involving λt of a non-repairable component. The term involving $\lambda_2 t$ is adjusted by $\frac{1}{m+1} = \frac{1}{4}$. The result is

$$P = \frac{1}{3} Q_2 Q_3 Q_4 \lambda_1 t + \frac{1}{4} Q_1 Q_3 Q_4 \lambda_2 t + \frac{1}{3} Q_1 Q_2 Q_4 \lambda_3 t + \frac{1}{3} Q_1 Q_2 Q_3 \lambda_4 t.$$

B.2.3.3 Allowed Outage Times

If $\tau_{\max}$ is the maximum allowed outage time without propagating a failure sequence, then for each component i with this condition, the factor

*Precise analysis shows that, for unequal λ_i , these factors differ from $\frac{1}{m}$. However, the sum of the m factors is still unity, and the overall probability of the cut set is unchanged.

$$f_i = P(\tau_{D_i} + \tau_{R_i} > \tau_{\max_i}) = e^{-\tau_{\max_i} / (\tau_{D_i} + \tau_{R_i})} \quad (5)$$

is applied to every such Q_i or $\lambda_{O_i}t$ appearing in Equation (2). In some cases, more than one component within a cut set may have such a factor. If m of the n components in the cut set have such factors, then each of the n additive terms in Equation (2) will have m multiplicative f -factors. Extreme values of the f -factor exponent can cause misleading results. In such cases, more basic treatment of event probability is preferred.

B.2.3.4 Inhibit Conditions, Houses, and Switches

Term X_i of a cut set may represent an inhibit condition to term X_j with an inhibit fraction k (where $0 < k < 1$). If all components in the cut set are nonrepairable, then the factor k multiplies $\lambda_j t$ in Equation (1). If there are ℓ such inhibit conditions in the cut set, there will be ℓ multiplicative k -values in Equation (1). For cut sets of repairable components, the factor k (where k is an inhibit fraction for X_j) multiplies each $\lambda_j t$ and Q_j , for each additive term in Equation (2). If there are ℓ inhibit conditions in the cut set, there will be ℓ multiplicative k -values in each additive term of Equation (2). From a more mechanistic viewpoint, an inhibit condition can be accommodated in Equation (2) by setting $\lambda_{O_i} = 0$ and $Q_i = k$.

If term X_i of a cut set is a house (condition which is always present, an "on switch" with a probability of one), an analogous approach is used. Then $\lambda_{O_i} = 0$ and $Q_i = 1$ in Equation (2). The opposite "off switch" condition of zero probability is handled by setting $\lambda_{O_i} = 0$ and $Q_i = 0$.

B.3 USE OF THE EQUATIONS

In this section, a simplified method for evaluating the failure probability of a cut set is derived from the foregoing equations.

B.3.1 Input Requirements

For each basic event X_i in a cut set, the following quantities are input:

$P_i = \lambda_{o_i} t$. Do not confuse P_i with P . The latter quantity is the probability for the entire cut set.

Q_i from the Equation (3) for on-line repairable components or from Equation (4) for standby repairable components. For non-repairable components, $Q_i = P_i$.

$m_i = \begin{cases} 0 & \text{if } X_i \text{ is repairable} \\ 1 & \text{if } X_i \text{ is nonrepairable.} \end{cases}$

Recall that for an on-line component with a standby, $Q_i = \lambda_{o_i} \tau_i$. For an inhibit condition, $P_i = 0$, $Q_i = k$, $m_i = 0$. For a house or switch, $P_i = 0$, $Q_i = 0$ or 1 , $m_i = 0$.

If a propagation delay factor f_i from Equation (5) is desired, both the P_i and Q_i values should be multiplied by $f_i (< 1)$ before being input. If there is no propagation delay, $f_i = 1$.

B.3.2 Calculational Procedure

For an n-term cut set $X_1 X_2 \dots X_n$, with

$$m = m_1 + m_2 + \dots + m_n = \sum_{i=1}^n m_i,$$

the following n products, each of n terms, are formed:

$$P(1) = Q_2 Q_3 \dots Q_n P_1$$

$$P(2) = Q_1 Q_3 Q_4 \dots Q_n P_2$$

$$P(3) = Q_1 Q_2 Q_4 \dots Q_n P_3$$

.

$$P(n) = Q_1 Q_2 \dots Q_{n-1} P_n.$$

The notation used is that of Roberts.⁽⁶⁾

If $m = 0$, the cut set probability is

$$P = P(1) + P(2) + \dots + P(n).$$

For the case of $m \neq 0$, a multiplicative factor must be applied:

If $m_1 = 1$, then

$$P(1) = \frac{1}{m} P(1)$$

If $m_1 = 0$, then

$$P(1) = \frac{1}{m+1} P(1)$$

If $m_2 = 1$, then

$$P(2) = \frac{1}{m} P(2)$$

If $m_2 = 0$, then

$$P(2) = \frac{1}{m+1} P(2)$$

·
·
·

If $m_n = 1$, then

$$P(n) = \frac{1}{m} P(n)$$

If $m_n = 0$, then

$$P(n) = \frac{1}{m+1} P(n)$$

The cut set probability is again

$$P = P(1) + P(2) + \dots + P(n).$$

B.3.3 Sample Calculations

Given the cut set $X_1X_2X_3X_4$.

a. All input events represent failures of repairable components.

Input:	<u>Event</u>	<u>P_i</u>	<u>Q_i</u>	<u>m_i</u>
	X ₁	P ₁	Q ₁	0
	X ₂	P ₂	Q ₂	0
	X ₃	P ₃	Q ₃	0
	X ₄	P ₄	Q ₄	0

Output:

$$m = 0 + 0 + 0 + 0 = 0$$

$$P(1) = Q_2 Q_3 Q_4 P_1$$

$$P(2) = Q_1 Q_3 Q_4 P_2$$

$$P(3) = Q_1 Q_2 Q_4 P_3$$

$$P(4) = Q_1 Q_2 Q_3 P_4$$

$$P = P(1) + P(2) + P(3) + P(4)$$

b. All input events nonrepairable

Input:	<u>Event</u>	<u>P_i</u>	<u>Q_i</u>	<u>m_i</u>
	X ₁	P ₁	Q ₁	1
	X ₂	P ₂	Q ₂	1
	X ₃	P ₃	Q ₃	1
	X ₄	P ₄	Q ₄	1

Note: For nonrepairable components $P_i = Q_i$

Output:

$$m = 1 + 1 + 1 + 1 = 4$$

$$P(1) = \frac{1}{4} Q_2 Q_3 Q_4 P_1$$

$$P(2) = \frac{1}{4} Q_1 Q_3 Q_4 P_2$$

$$P(3) = \frac{1}{4} Q_1 Q_2 Q_4 P_3$$

$$P(4) = \frac{1}{4} Q_1 Q_2 Q_3 P_4$$

Since

$$Q_1 = P_1$$

$$Q_2 = P_2$$

$$Q_3 = P_3$$

$$Q_4 = P_4,$$

$$P = P_1 P_2 P_3 P_4$$

c. Events X_2 and X_3 nonrepairable, X_1 and X_4 repairable

Input:	<u>Event</u>	<u>P_i</u>	<u>Q_i</u>	<u>m_i</u>
	X_1	P_1	Q_1	0
	X_2	P_2	Q_2	1
	X_3	P_3	Q_3	1
	X_4	P_4	Q_4	0

Output:

$$m = 2$$

$$P(1) = \frac{1}{3} Q_2 Q_3 Q_4 P_1$$

$$P(2) = \frac{1}{2} Q_1 Q_3 Q_4 P_2$$

$$P(3) = \frac{1}{2} Q_1 Q_2 Q_4 P_3$$

$$P(4) = \frac{1}{3} Q_1 Q_2 Q_3 P_4$$

The above equations could be simplified, since $P_2 = Q_2$ and $P_3 = Q_3$.

$$P = P(1) + P(2) + P(3) + P(4)$$

d. Events X_1 , X_2 , X_4 repairable. X_3 is an inhibit condition to X_4 with probability k , where $0 < k < 1$.

Input:	<u>Event</u>	<u>P_i</u>	<u>Q_i</u>	<u>m_i</u>
	X_1	P_1	Q_1	0
	X_2	P_2	Q_2	0
	X_3	0	k	0
	X_4	P_4	Q_4	0

Output:

$$m = 0$$

$$P(1) = Q_2 k Q_4 P_1$$

$$P(2) = Q_1 k Q_4 P_2$$

$$P(3) = 0$$

$$P(4) = Q_1 Q_2 k P_4$$

$$P = P(1) + P(2) + P(3) + P(4)$$

$$= P(1) + P(2) + P(4)$$

e. Same as 4, except that X_3 is a house (on-switch)

Input: same, except $Q_3 = 1$

Output: same, except k is replaced by 1.

f. Events X_1, X_3 repairable. X_2 represents a repairable standby component to X_1 . X_4 represents a repairable standby component to X_3 .

Input:	<u>Event</u>	<u>P_i</u>	<u>Q_i</u>	<u>m_i</u>
	X_1	P_1	$Q_1 = \lambda_{o1} \tau_1$	0
	X_2	P_2	Q_2	0
	X_3	P_3	$Q_3 = \lambda_{o3} \tau_3$	0
	X_4	P_4	Q_4	0

Output:

$$m = 0$$

$$P(1) = Q_2 Q_3 Q_4 P_2$$

$$P(2) = Q_1 Q_3 Q_4 P_2$$

$$P(3) = Q_1 Q_2 Q_4 P_3$$

$$P(4) = Q_1 Q_2 Q_3 P_4$$

$$P = P(1) + P(2) + P(3) + P(4)$$

g. Events X_1, X_3 nonrepairable. X_2 represents a repairable standby component to X_1 . X_4 represents a repairable standby component to X_3 .

Input:	<u>Event</u>	<u>P_i</u>	<u>Q_i</u>	<u>m_i</u>
	X_1	P_1	Q_1	1
	X_2	P_2	Q_2	0
	X_3	P_3	Q_3	1
	X_4	P_4	Q_4	0

Output:

$$m = 2$$

$$P(1) = \frac{1}{2} Q_2 Q_3 Q_4 P_1$$

$$P(2) = \frac{1}{3} Q_1 Q_3 Q_4 P_2$$

$$P(3) = \frac{1}{2} Q_1 Q_2 Q_4 P_3$$

$$P(4) = \frac{1}{3} Q_1 Q_2 Q_3 P_4$$

The above equations could be simplified since $P_1 = Q_1$, and $P_3 = Q_3$.

$$P = P(1) + P(2) + P(3) + P(4).$$

REFERENCES

1. T. H. Smith, P. J. Pelto, D. L. Stevens, G. D. Seybold, W. L. Purcell, and L. V. Kimmel, A Risk-Based Fault Tree Analysis Method for Identification, Preliminary Evaluation, and Screening of Potential Accidental Release Sequences in Nuclear Fuel Cycle Operations, BNWL-1959, Battelle, Pacific Northwest Laboratories, Richland, WA, January 1976.
2. USAEC, Reactor Safety Study, An Assessment of Accident Risks in Commercial Nuclear Power Plants, WASH-1400, October 1975.
3. W. E. Vesely, Analysis of Fault Trees by Kinetic Tree Theory, IN-1330, Idaho Nuclear Corporation, Idaho Falls, ID, October 1969.
4. H. Lambert, Fault Tree Analysis: An Overview, UCRL-75904, Lawrence Livermore Laboratory, Livermore, CA, August 1974.
5. D. F. Haasl, "Fault Tree Analysis", course notes, Joint Center for Graduate Study, Richland, WA, February 1974.
6. N. H. Roberts, "Systems Safety Analysis", course notes, University of Washington, Seattle, WA, 1975.

DISTRIBUTION

No. of
Copies

No. of
Copies

OFFSITE

UNITED STATES

A. A. Churm U.S. DOE Chicago Patent Group 9800 South Cass Avenue Argonne, IL 60439	R. E. Barlow UCLA, Berkeley Berkeley, CA 94720
27 U.S. DOE Technical Information Center	G. R. Burdick Aerojet Nuclear Company 550 Second Street Idaho Falls, ID 83401
W. Brobst U.S. DOE Division of Environmental Control Technology Transportation Branch U.S. DOE Headquarters Washington, DC 20545	H. C. Claiborne Union Carbide Corporation (HNL) P. O. Box Y Oak Ridge, TN 37830
J. A. Sisler U.S. DOE Division of Environmental Control Technology Transportation Branch U.S. DOE Headquarters Washington, DC 20545	J. Curtis Department of Transportation Materials Transportation Bureau 2100 Second St. S.W. Washington, DC 20590
W. K. Eister U.S. DOE Division of Nuclear Fuel Cycle and Production Washington, DC 20545	W. S. Durant duPont Company, Aiken (U.S. DOE) E. I. duPont DeNemours and Co. Savannah River Laboratory Aiken, SC 29801
R. D. Walton U.S. DOE Division of Nuclear Fuel Cycle and Production Washington, DC 20545	D. Egan Environmental Protection Agency 5207 Longsdale Drive Springfield, VA 22151
G. E. Apostolakis UCLA 5532 Boelter Hall Los Angeles, CA 90024	R. C. Erdmann Science Application, Inc. 2680 Hanover Street Palo Alto, CA 94303
	R. R. Fullwood Science Applications, Inc. 2680 Hanover Street Palo Alto, CA 94303

No. of
Copies

J. B. Fussell
University of Tennessee
Department of Nuclear Engineering
Knoxville, TN 37916

D. F. Haas
Institute of System Sciences
3085 North Bar Drive
North Bend, OR 97459

J. D. Hill
Battelle Memorial Institute
505 King Avenue
Columbus, OH 43201

R. Howard
Stanford University
Stanford, CA 93045

B. R. Judd
Management Analysis Co.
Palo Alto, CA 94303

H. E. Lambert
Lawrence Livermore Laboratory
Livermore, CA 94550

G. S. Lellouche
Electric Power Research Institute
3412 Hillview Avenue
P.O. Box 10412
Palo Alto, CA 94303

S. E. Logan
University of New Mexico
Albuquerque, NM 87131

D. Okrent
UCLA
5532 Boelter Hall
Los Angeles, CA 90024

T. G. Priddy
Sandia Laboratories
P.O. Box 5800
Albuquerque, NM 87115

No. of
Copies

N. Rasmussen
Massachusetts Institute of
Technology
Cambridge, MA 02139

W. Rowe
Environmental Protection Agency
401 M. Street
Washington, DC 20460

P. Rubel
Union Carbide Corporation (HNL)
P.O. Box Y
Oak Ridge, TN 37830

T. H. Smith
EEG Company
Idaho Falls, ID 83401

D. Turner
Union Carbide Corporation
Office of Waste Isolation
P.O. Box Y
Oak Ridge, TN 37830

W. E. Vesely
NRC Office of Nuclear Regulatory
Research
Washington, DC 20545

I. B. Wall
NRC Office of Nuclear Regulatory
Research
Washington, DC 20545

R. Williams
Electric Power Research Institute
3412 Hillview Avenue
P.O. Box 10412
Palo Alto, CA 94304

FOREIGN

G. D. Bell
United Kingdom Atomic Energy
Authority
Safety and Reliability
Directorate
Warrington WA3 4NE
UNITED KINGDOM

No. of
Copies

H. W. M. Braun
C.C.R. EURATOM
21020 Ispra (Va)
C.P. 5
ITALIA

H. Dyroff
NUKEM GmbH
645 Hanau 11
Postfach 110080
WEST GERMANY

Ferruccio Gera
Laboratorio Rifiuti
Radioattivi
C.N.E.N. - C.S.N. Casaccia
Casella Postale 2400
00100 Rome, ITALY

S. Hartwig
Battelle Institute, e.v.
Am Romerhof 35
600 Frankfurt Main 90
GERMANY

International Atomic Energy
Agency
Kärtner Ring 11
P.O. Box 590
A-1011, Vienna, AUSTRIA

H. Krause
Nuclear Research Center
Waste Management Department
D75 Karlsruhe
Weberstr. 5
GERMANY

M. Stammier
Battelle-Institute, e.v.
6000 Frankfurt am Main 90
Am Romerhof 35
Frankfurt, GERMANY

No. of
Copies

H. J. Wingender
NUKEM GmbH
645 Hanau 11
Postfach 110080
WEST GERMANY

H. Witte
NUKEM GmbH
645 Hanau 11
Postfach 110080
WEST GERMANY

ONSITE

3 US DOE Richland Operations
Office

H. E. Ransom
R. B. Goranson
M. J. Zamorski

4 Rockwell Hanford Operations

R. A. Deju
R. E. Isaacson
D. D. Wodrich
File Copy

5 Exxon

J. L. Carter

1 Joint Center for Graduate Study

J. Cooper

1 United Nuclear Industries, Inc.

P. A. Crosetti

3 HEDL

J. E. Ecker
J. W. Hagen
D. E. Simpson

No. of
Copies

65 Battelle-Northwest

N. E. Carter
J. C. Fox
R. J. Hall
S. W. Heaberlin
J. W. Lichfield
T. I. McSweeney
E. S. Murphy
P. J. Pelto (20)
A. M. Platt
W. L. Purcell
R. E. Rhoads
G. D. Seybold
D. L. Stevens
J. W. Voss
L. D. Williams
W. K. Winegardner (25)
Technical Information (5)
Technical Publications