76
9-3-74
ANCR-1156

01-928

UC-32

MOCUS — A COMPUTER PROGRAM TO OBTAIN MINIMAL SETS FROM FAULT TREES

J. B. Fussell

E. B. Henry

N. H. Marshall

Aerojet Nuclear Company

NATIONAL REACTOR TESTING STATION

Idaho Falls, Idaho — 83401

DATE PUBLISHED AUGUST 1974

PREPARED FOR THE

U. S. ATOMIC ENERGY COMMISSION

IDAHO OPERATIONS OFFICE UNDER CONTRACT AT(10-1)-1375

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

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

DISCLAIMER

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

DISCLAIMER

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

DISCLAIMER

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

Printed in the United States of America
Available from
National Technical Information Service
U. S. Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22151
Price: Printed Copy \$4.00; Microfiche \$1.45

LEGAL NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, 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.

ANCR-1156

Mathematics and Computers
TID-4500

**MOCUS -- A COMPUTER PROGRAM TO OBTAIN
MINIMAL SETS FROM FAULT TREES**

J. B. Fussell
E. B. Henry
N. H. Marshall

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, 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.

AEROJET NUCLEAR COMPANY

Published August 1974

**PREPARED FOR THE U. S. ATOMIC ENERGY COMMISSION
IDAHO OPERATIONS OFFICE
UNDER CONTRACT NO. AT(10-1)-1375**

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

GG

ACKNOWLEDGMENT

Automation Industries, Inc./Vitro Laboratories Division provided funding for the development and documentation of the MOCUS computer program.

ABSTRACT

This report is the user's manual for MOCUS, a computer program for obtaining minimal cut sets and minimal path sets from fault trees. The major algorithm used in MOCUS is presented. The input and output of the program are described in detail.

SUMMARY

From a given graphical representation of the Boolean failure logic of a system, called a fault tree, the system failure modes, called the minimal cut sets, and the system success modes, called the minimal path sets, can be determined. The computer program, MOCUS, determines these minimal sets by an algorithm that directly resolves the fault tree into cut and path sets.

Input required by the computer program consists of:

- (1) Control parameters specifying (a) the minimal cut set length to be obtained, (b) the type of minimal sets to be obtained, and (c) the output options
- (2) A description of the fault tree.

Output from MOCUS can include printed and punched minimal cut and path sets for up to 20 gates in the fault tree of interest.

CONTENTS

ACKNOWLEDGMENT	ii
ABSTRACT	iii
SUMMARY	iv
I. INTRODUCTION	1
II. DEFINITIONS AND CONCEPTS	2
1. FAULT TREES	2
2. CUT SETS	2
3. PATH SETS	5
4. HOUSE EVENTS	5
III. PROGRAM DESCRIPTION	6
1. INPUT	6
2. THE MOCUS ALGORITHM	6
IV. INPUT DECK DESCRIPTION	11
1. INPUT GROUP 1	12
2. INPUT GROUP 2	13
3. INPUT GROUP 3	14
4. INPUT GROUP 4	16
V. OUTPUT FROM MOCUS	18
VI. REFERENCES	19
APPENDIX A – SAMPLE INPUT DATA DECK	21
APPENDIX B – SAMPLE PROBLEM JOB CONTROL LANGUAGE FOR THE IBM 360/75 COMPUTER AT THE NRTS	25
APPENDIX C – SAMPLE OUTPUT FOR APPENDIX A DATA DECK	29
APPENDIX D – MOCUS INPUT IN BRIEF	37

FIGURES

1.	Sample fault tree representation	3
2.	Fault tree symbols	4
A-1.	Fault tree for sample problem input	23

TABLES

I.	Input Group 1 Description	12
II.	Input Format for Each Input Card of Group 2	14
III.	Description of House Events in Input Group 3	15
IV.	Input Group 3 Information for Use in KITT-1	15
V.	LAMDA(I) and TAU(I) Interpretations	16
VI.	Description and Format for Input Group 4	17
A-I.	Input Data Deck for Sample Fault Tree	24
D-I.	Input Deck Layout	39
D-II.	Input Group 1 – Control Information	39
D-III.	Input Group 2 – Fault Tree Description	40
D-IV.	Input Group 3 – House Event or House Event and Failure Data	41
D-V.	Input Group 4 – Gates for Which Minimal Sets Are Obtained	42

MOCUS -- A COMPUTER PROGRAM TO OBTAIN MINIMAL SETS FROM FAULT TREES

I. INTRODUCTION

MOCUS is a computer program that locates minimal cut sets or minimal path sets or both from fault trees. Minimal cut sets are modes of system failure and are collections of basic system component failures that are collectively necessary and sufficient to cause system failure by that failure mode. The complete collection of all minimal cut sets is all the unique modes of system failure. Minimal path sets are modes of system success and are collections of basic system component operating states that are collectively necessary and sufficient to guarantee system success by that success mode. The complete collection of all minimal path sets is all the unique system modes of success. Minimal sets are important for qualitative fault tree analysis and are required for quantitative fault tree analysis.

MOCUS is written in FORTRAN IV for the IBM 360/75 computer. The algorithm used in MOCUS to locate the minimal sets is unique in that it starts at the main failure of interest, called the TOP event, and proceeds to basic, independent component failures, called primary events, to resolve the fault tree into cut or path sets. The algorithm, thereby, does not use deterministic testing, Boolean manipulation, or Monte Carlo techniques. The output from MOCUS is in a format compatible with the input requirements of the KITT computer program^[1] for quantitative reliability evaluation. MOCUS may be used in place of the existing PREP computer code^[1]. The input format used in MOCUS is similar, often identical, to the PREP input format. A main feature of MOCUS is the small execution times to locate even large minimal sets. MOCUS guarantees that all the minimal sets requested have been found. MOCUS input is engineered for easy use by analysts familiar with standard fault tree terminology. Extensive error checking of the input by MOCUS reduces the chance of obtaining erroneous minimal sets.

Background information about fault trees and about minimal set determination is given in Section II. The MOCUS basic input requirements and algorithm description are discussed in Section III. A detailed input data deck description is presented in Section IV. The basic output from MOCUS is described in Section V. The input data deck and the job control language for the IBM 360/75 computer are given in Appendix A and Appendix B. An actual printout is supplied in Appendix C. Appendix D gives a quick reference table for MOCUS input descriptions.

II. DEFINITIONS AND CONCEPTS

The following definitions and concepts are particularly pertinent to MOCUS. A more detailed composite of fault tree definitions is given in Reference 2.

1. FAULT TREES

A fault tree is a graphical representation of the Boolean failure logic associated with the development of a particular TOP event for a particular system. The fault tree is a model that bridges the gap between the real world and mathematics. Construction of fault trees is dealt with in References 2 and 3.

The TOP event of a fault tree occurs when the system passes from the unfailed to the failed state. Fault trees represent monotonic structures in the respect that the occurrence of any primary event never decreases the probability of the TOP event. A given primary event can appear in many places in a fault tree; however, all primary events must represent component failures that are independent from all other primary events.

A fault tree representation is shown in Figure 1. An explanation of the symbols used in fault trees is given in Figure 2. For the purposes of MOCUS, a gate is comprised of both the event description, appearing in a rectangle, and the logic gate, appearing under the rectangle.

2. CUT SETS

Cut sets are collections of primary events, and if all these primary events occur the system failure is guaranteed to occur. For example, for the fault tree in Figure 1, if Events 1, 2, and 3 occur simultaneously the TOP event will occur. Therefore [1,2,3] is a cut set. Also [1,2,3,4], [1,3,4], [1,2], and [1,4] are cut sets.

Minimal cut sets are cut sets such that if any primary event is removed from the set the remaining events collectively are no longer a cut set. A minimal cut set indicates failure; that is, it represents a failed state, when all the primary events in that minimal cut set are simultaneously in the failed state. Therefore, as a minimal cut set indicates failure, the TOP event occurs. If only one minimal cut set indicates failure, repairing any primary event in this minimal cut set repairs the system. Minimal cut sets are the modes of system failure and are the sets of components that are critical with regard to system failure. Minimal cut sets containing n primary events are called n -event minimal cut sets. The minimal cut sets for the fault tree shown in Figure 1 are all two-event minimal cut sets and are [1,4], [1,2], and [2,3]. Such a complete collection of all the minimal cut sets for a fault tree are all the unique modes of failure for the system within the constraint of the TOP event being

Fig. 1 Sample fault tree representation.

System component or basic fault event.

Inputs

OR gate.

This gate is in the failed state if at least one of its inputs is in the failed state.

Inputs

AND gate.

This gate is in the failed state only if all of its inputs are simultaneously in their failed states.

Event symbol.

The rectangle describes the event represented by a gate.

The diamond represents a fault event which is not developed further due to lack of information.

Transfer symbols.

These symbols transfer an entire part of the tree to other locations on the tree.

INHIBIT gate.

The INHIBIT gate represents an event which occurs with some fixed probability of occurrence. The INHIBIT gate is in the failed state only if its inputs are in the failed state and the inhibit condition has occurred.

The house represents an event which is normally expected to occur. It is treated as a switch on the tree, and is set on or off by the MOCUS user.

Fig. 2 Fault tree symbols.

analyzed. MOCUS determines all the minimal cut sets for the TOP event or other specified gates in the fault tree.

3. PATH SETS

Path sets are collections of primary events and if none of these primary events occur the system failure is guaranteed to not occur. For example, for the fault tree in Figure 1, if Events 1, 2, and 4 are not failed the system cannot be failed. Therefore [1,2,4] is a path set. Also [1,2,3,4], [1,2,3], [2,4], and [2,3,4] are path sets.

Minimal path sets are path sets such that if any primary event is removed from the set the remaining events collectively are no longer a path set. For the TOP event to occur, at least one event in every path set must occur. A path set is failed if one event in the path set is failed. Minimal path sets containing n primary events are called n-event minimal path sets. The minimal path sets for the fault tree in Figure 1 are all two-event minimal path sets and are [1,2], [1,3], and [2,4]. Such a complete collection of all the minimal path sets for a fault tree includes all the unique success modes of the system within the constraint of the TOP event being analyzed.

The minimal path sets can be determined directly from the minimal cut sets and vice versa and are in this respect equivalent sources of information. Also either can be used to determine quantitative system reliability characteristics. MOCUS determines the minimal path sets for the TOP event or other specified gates in the fault tree.

4. HOUSE EVENTS

A house event is an event with a probability of zero or unity of occurrence. Houses are useful in a particular analysis to "turn on" or, equivalently, to "mask out" portions of a fault tree. Also, house events can be used to study the effect of a particular primary event being failed. Many dynamic systems can be analyzed with one fault tree containing house events. Without house events, dynamic systems often require a fault tree for each phase of operation. Skillful use of house events allows much additional information to be extracted from a fault tree that would not be possible otherwise.

MOCUS treats house events explicitly by pruning the fault tree as required by the house events before the cut sets are determined.

III. PROGRAM DESCRIPTION

From a description of the fault tree, MOCUS locates the minimal cut sets or the minimal path sets or both for specified fault events in the fault tree. MOCUS will treat house events. The house events appear in the minimal set output if their state is not described.

1. INPUT

For the purposes of describing the fault tree, MOCUS accepts AND and OR logic gates. Other types of logic gates must be described in terms of AND and OR logic gates. Inhibit gates are input as AND logic gates with the inhibit condition as an event input to this AND logic gate.

Each fault event and primary event must be labeled with an alphanumeric name. This name can be any combination of from one to eight characters. If a primary event appears more than once in the fault tree, it must be given the same alphanumeric name every time it appears.

Other input to MOCUS specifies

- (1) The job title to be printed out with the output
- (2) Whether minimal cut sets, minimal path sets, or both are to be determined
- (3) The maximum length of the minimal cut sets to be obtained
- (4) Whether a house event is turned on or off
- (5) The fault events for which minimal cut sets or minimal path sets are to be obtained
- (6) Whether the output is to be printed or punched or printed and punched

The fault tree description input to MOCUS is identical to the fault tree description input to TREBIL, a program in the PREP computer program package. Existing input data for TREBIL can be used for partial fulfillment of the input requirements of MOCUS.

2. THE MOCUS ALGORITHM

The algorithm used in MOCUS starts with the TOP event and resolves the fault tree to obtain the minimal sets. A key point of this algorithm is that an AND gate alone always

increases the size of a cut set and increases the number of path sets. An OR gate alone always increases the number of cut sets and increases the size of path sets. To describe the algorithm, the term Boolean Indicated Sets (BIS) is introduced. The BIS are defined such that, if all the primary events are different, the BIS are precisely the minimal sets. This definition of the BIS does not mean that the method is limited to fault trees with primary events appearing only once in the fault tree.

First the algorithm will be given to obtain the Boolean Indicated Cut Sets (BICS). To obtain the BICS, each gate in the fault tree is randomly named with a value ω and each primary event with a value ϕ . The following definitions apply to this approach:

- $\rho_{\omega,i} \equiv i^{\text{th}}$ input to gate ω
- $\lambda_{\omega} \equiv$ number of inputs to gate ω
- $x \equiv$ the x^{th} BICS
- $y \equiv$ the y^{th} entry in a BICS
- $\Delta_{x,y} \equiv$ variable representing the y^{th} entry in the x^{th} BICS
- $x_{\text{max}} \equiv$ largest value of x yet used
- $y_{\text{max}} \equiv$ largest value of y yet used in the x^{th} BICS.

The values ω , ϕ , $\rho_{\omega,i}$, λ_{ω} and the gate type (AND or OR) are assumed as input, where values of $\rho_{\omega,i}$ are discernible values of ω or ϕ . $\Delta_{1,1}$ is first set equal to the ω value representing the gate immediately under the TOP event. From this point on, the goal is to eliminate all ω values from the $\Delta_{x,y}$ matrix. When this elimination is complete, only ϕ values remain and the BICS are determined. To accomplish this elimination, an ω value is located in the $\Delta_{x,y}$ matrix, the values of x , y , and ω are noted, and

$$\Delta_{x,y} = \rho_{\omega,1}. \quad (1)$$

For ω being an AND gate:

$$\Delta_{x,y_{\text{max}}+1} = \rho_{\omega,\pi} \quad \pi = 2, 3, \dots, \lambda_{\omega} \quad (2)$$

where y_{max} is incremented when π is incremented.

For ω being an OR gate:

$$\begin{aligned}
\Delta_{x_{\max}+1,n} &= \Delta_{x,n} \quad n = 1, 2, \dots, y_{\max} \quad n \neq y \\
&= \rho_{\omega,\pi} \quad n = y \\
\pi &= 2, 3, \dots, \lambda_{\omega}
\end{aligned} \tag{3}$$

where $x_{\max}$ is incremented when π is incremented.

Processes represented by Equations (1) and (2) or (3) are repeated until all the entries in the $\Delta_{x,y}$ matrix become values of ϕ . The BICS are then determined.

The duplicate events appearing in any one BICS, if any, are eliminated. Also all BICS that are supersets of any other BICS are discarded. A superset is a BICS that contains every primary event that some other BICS contains plus additional primary events. After this winnowing, the minimal cut sets are determined.

To determine the minimal path sets, the algorithm is modified only by using Equation (2) for OR gates and Equation (3) for AND gates.

An example best illustrates this method. The fault tree in Figure 1 is used in this example. Each gate in that figure is labeled with a letter and each primary event with a number. This input to the methodology is then

ω	Gate Type	λ_{ω}	$\rho_{\omega,i}$	
A	AND	2	B	C
B	OR	2	1	D
C	OR	2	2	4
D	AND	2	2	3

The solution is begun by setting a $\Delta_{x,y}$ matrix, as:

$$\Delta_{x,y}$$

		y →		
x ↓	A			

To determine the minimal cut sets, since A is an AND gate, Equations (1) and (2) are used to obtain

$$\Delta_{x,y}$$

$y \rightarrow$	B.	C	
$x \downarrow$			

To resolve Gate B, Equations (1) and (3) are used to obtain

$$\Delta_{x,y}$$

$y \rightarrow$	1	C	
$x \downarrow$	D	C	

Both C gates are resolved by Equations (1) and (3) to obtain

$$\Delta_{x,y}$$

$y \rightarrow$	1	2	
$x \downarrow$	D	2	
	1	4	
	D	4	

Finally both D gates are resolved by Equations (1) and (2) to obtain

$$\Delta_{x,y}$$

$y \rightarrow$	1	2	
$x \downarrow$	2	2	3
	1	4	
	2	4	3

(4)

Matrix (4) contains the BICS. To obtain the minimal cut sets, duplication of events is removed for each row of Matrix (4), that is, from each BICS, and supersets are then discarded. Each row of the $\Delta_{x,y}$ matrix is then a minimal cut set. For this problem the minimal cut sets are [1,2], [2,3], and [1,4].

To determine the minimal path sets, the procedure is modified by using Equation (2) everywhere Equation (3) was used and using Equation (3) everywhere Equation (2) was used. The minimal path sets for the fault tree given in Figure 1 are [1,2], [2,4], and [1,3].

IV. INPUT DECK DESCRIPTION

The input to MOCUS consists of the four input groups:

- (1) Control information
- (2) Fault tree description
- (3) Failure and repair data for the components
- (4) Gates for which minimal sets are to be found.

The beginning of each group of data is indicated by a special control card of the form:

Column 1	Column 2	Column 3
*	blank	KEYWORD

with KEYWORD replaced by DATA for Input Group 1, TREE for Input Group 2, RATES for Input Group 3, and TESTS for Input Group 4. Input Group 3 and Input Group 4 are optional. The end of each input group is indicated by a card with the word END beginning in the first card column. Each of these input groups may be preceded by any number of comments cards. The first such card appearing in the input deck is used as a title card and its comments are printed on each page of output.

The total input deck has the following format:

- (1) TITLE CARD
- (2) [COMMENT CARDS]
- (3) *bDATA (b = blank column)
- (4) [CONTROL PARAMETERS]
- (5) END
- (6) [COMMENTS CARDS]
- (7) *bTREE
- (8) [FAULT TREE DESCRIPTION]
- (9) END

- (10) [COMMENTS CARDS]
- (11) *bRATES
- (12) [FAILURE AND REPAIR DATA]
- (13) END
- (14) *bTESTS
- (15) [GATES FOR WHICH MINIMAL SETS ARE FOUND]
- (16) END

1. INPUT GROUP 1

Input Group 1 contains the maximum length minimal cut sets found and switches to control the output. Table I contains a brief description of the contents of Input Group 1.

TABLE I

INPUT GROUP 1 DESCRIPTION
Format [I5, 5X, 2(A10)]

FORTRAN Variable	Description
ITMAX	Maximum size of the minimal cut sets to be found.
IDEX1	Switch that determines whether minimal cut sets, minimal path sets, or both are to be found.
IDEX2	Switch that determines whether the minimal sets are to be printed or punched, or both.

ITMAX is used only when minimal cut sets are being determined. Neither the length nor the number of minimal path sets can be limited because quantitative and qualitative analyses require total path set information. Specifying ITMAX to be only as large as required for cut set determination can significantly reduce the execution time of MOCUS. If ITMAX is set to zero or left blank all the minimal sets are found. ITMAX must be right justified in the field.

IDEX1 specifies which minimal sets are located. If the word CUT is input for IDEX1, the minimal cut sets are determined. If the word PATH is input for IDEX1, the minimal path sets are located. If the word BOTH is input for IDEX1, both the minimal cut sets and minimal path sets are found. IDEX1 must be left justified in the field.

IDEX2 specifies the form of the output. If the word PRINT is input for IDEX2, the minimal sets are printed. If the word PUNCH is input for IDEX2, the minimal sets are punched in a format compatible for use in the KITT codes^[1] for quantitative system reliability evaluation. If the word BOTH is input for IDEX2, the minimal sets are printed and punched. The punch option allows MOCUS to punch cards or write to external storage, or both. IDEX2 must be left justified in the field.

2. INPUT GROUP 2

Information required for Input Group 2 is the name of each gate in the fault tree, its type, the number of inputs to it, and the names of its inputs. The information is supplied one gate per card. A description of the format for Input Group 2 is given in Table II.

All of the inputs described in Table II must begin in the left most of the columns indicated for each of them except for Inputs 5 and 6 which must be right justified in the field. All gates which are inputs must be listed before any components. All inhibit conditions and house events are input in Input Group 2 in exactly the same manner as components are input. The cards in Input Group 2 can be supplied in any order. Sometimes placing the card for the TOP event first can be advantageous as is described for Input Group 4 (Section IV-4).

TABLE II

INPUT FORMAT FOR EACH INPUT CARD OF GROUP 2

Inputs	Card Columns
1. Gate name (any eight alphanumeric characters)	1 to 8
2. Blank	9
3. Gate type ("AND" OR "OR")	10 to 12
4. Blank	13
5. Number of gate inputs	14 to 15
6. Number of component inputs	16 to 17
7. First input name	19 to 26
8. Blank	27
9. Second input name	28 to 35
10. Blank	36
.	.
.	.
.	.

3. INPUT GROUP 3

Input Group 3 is used to specify whether house events have a failure probability of zero (are "off") or unity (are "on"). The name of the house event, NAM(I), must be the same alphanumeric name as used in Input Group 2. Table III provides a brief description of the two variables included in Input Group 3 when house events are described. Data for two house events can be included on one card. If the house event is off, TAU(I) is set to 0.0 or left blank. If the house event is always failed, TAU(I) is set to 1.0.

TABLE III

DESCRIPTION OF HOUSE EVENTS IN INPUT GROUP 3

Format [2(A8, 12X, F10.3, 1X)]

FORTRAN Variable	Description
NAM(I)	I^{th} house event name
TAU(I)	Switch to turn house on or off

Input Group 3 can also be used to supply failure data to be punched by MOCUS for direct use in KITT-1 if IDEX of Input Group 1 is set equal to PUNCH or BOTH. This versatility is identical to that provided by PREP. In this case the information required is a failure rate for each component, repair data for each component, and inhibit failure probabilities. The description of this option is described in Tables IV and V. Input Group 3 is optional. The default values are 1×10^{-6} for failure rates and 0.0 for repair times. Only the house event information affects the minimal set determination.

TABLE IV

INPUT GROUP 3 INFORMATION FOR USE IN KITT-1

Format [2(A8, 2X, F10.6, F10.3, 1X)]

FORTRAN Variable	Description
NAM(I)	I^{th} component name (must be same as for Input Group 2)
LAMDA(I)	I^{th} component failure rate (per 10^6 hours) or inhibit switch
TAU(I)	I^{th} component repair time or inhibit failure probability

TABLE V

LAMDA(I) AND TAU(I) INTERPRETATIONS

LAMDA(I)	TAU(I)	Interpretation
Positive	Positive	Component failure with failure rate LAMDA(I) and constant repair time TAU(I)
	0.0 or blank	Nonrepairable component failure with failure rate LAMDA(I)
Negative 0.0 or blank	between 0.0 and 1.0	Inhibit failure with constant probability of failure TAU(I)

4. INPUT GROUP 4

Input Group 4 specifies the gates for which minimal sets are to be found. One card is required for each gate. The gate alphanumeric name must be identical to that used in Input Group 2 and must begin in the left most column of the field. If no Input Group 4 data cards are supplied then the minimal sets specified in Input Group 1 are determined for the first gate described in Input Group 2 only. If Input Group 4 data cards are supplied then the minimal sets are not determined for the first gate described in Input Group 2 unless these minimal sets are specifically requested. The directions supplied in Input Group 1 and Input Group 3 apply to all the gates in Input Group 4.

If information is punched by MOCUS for use in KITT-1 or KITT-2, the minimal sets are punched for all the gates in the Input Group 4 data unless otherwise specified or, if no Input Group 4 cards are supplied, for the first gate in Input Group 2. To specify that the minimal sets for certain gates in Input Group 4 are not to be punched, any alphanumeric characters are entered in Columns 11 to 18 of the appropriate Input Group 4 card.

Table VI gives the description and format for Input Group 4. Up to 20 Input Group 4 cards can be supplied.

TABLE VI

DESCRIPTION AND FORMAT FOR INPUT GROUP 4
Format (A8, 2X, A8)

FORTRAN Variable	Description
NGTE	Alphanumeric name of gate for which minimal sets are to be found.
DUM1	Minimal set punch suppression flag. If DUM1 is any alphanumeric character the minimal sets for NGTE are not punched.

V. OUTPUT FROM MOCUS

The output from MOCUS always contains (a) an input data edit, (b) the number of minimal sets obtained, and (c) punched or printed minimal sets for at least one gate. MOCUS output can also contain the following:

- (1) Printed minimal cut sets and minimal path sets for up to 20 specified gates
- (2) Punched failure data and minimal cut sets or minimal path sets for up to 20 specified gates
- (3) A cross reference table that correlates a component failure integer name as supplied to KITT to each component failure alphanumeric name as supplied by the user.

The punched information can appear on cards, on a data record, or on both for subsequent quantitative evaluation and includes the complete set of failure rates and repair times for all primary events, the specified minimal sets, and the number of minimal sets as well. A sample output is given in Appendix C.

VI. REFERENCES

1. W. E. Vesely and R. E. Narum, *PREP and KITT: Computer Codes for the Automatic Evaluation of a Fault Tree*, IN-1349 (August 1970).
2. J. B. Fussell, *Synthetic Tree Model -- A Formal Methodology for Fault Tree Construction*, ANCR-1098 (March 1973).
3. D. F. Haasl, "Advanced Concepts in Fault Tree Analysis", *System Safety Symposium*, June 8-9, 1965, Seattle: The Boeing Company .

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

APPENDIX A
SAMPLE INPUT DATA DECK

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

APPENDIX A

SAMPLE INPUT DATA DECK

An example of MOCUS input data is given for the fault tree shown in Figure A-1, the data deck is given in Table A-I.

The minimal cut sets that contain seven or less events and minimal path sets will be printed for gates TOP, GATE2, and GATE3. The house event is "turned on"; that is, failure is indicated. The minimal sets for gates TOP and GATE2 will be punched for use in KITT.

ANC - A - 2364

Fig. A-1 Fault tree for sample problem input.

INPUT DATA DECK FOR SAMPLE FAULT TREE

24

APPENDIX B

SAMPLE PROBLEM JOB CONTROL LANGUAGE FOR THE

IBM 360/75 COMPUTER AT THE NRTS

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

APPENDIX B

SAMPLE PROBLEM JOB CONTROL LANGUAGE FOR THE

IBM 360/75 COMPUTER AT THE NRTS

```
//GO EXEC PGM=*.F.L.SYSLMOD,REGION=120K
//FT05F001 DD DDNAME=SYSIN
//FT06F001 DD SYSOUT=A,SPACE=(TRK,(20,5)),
// DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1596)
//FT07F001 DD SYSOUT=B
//FT10F001 DD DSN=88FL1,UNIT=DKSCRA,DISP=(NEW,PASS),SPACE=(TRK,5),
// DCB=(RECFM=VB,LRECL=84,BLKSIZE=1684)
//FT10F002 DD DSN=88FL2,UNIT=DKSCRA,DISP=(NEW,PASS),SPACE=(TRK,5),
// DCB=(RECFM=VB,LRECL=84,BLKSIZE=1684)
//FT10F003 DD DSN=88FL3,UNIT=DKSCRA,DISP=(NEW,PASS),SPACE=(TRK,5),
// DCB=(RECFM=VB,LRECL=84,BLKSIZE=1684)
//FT10F004 DD DSN=88FL4,UNIT=DKSCRA,DISP=(NEW,PASS),SPACE=(TRK,5),
// DCB=(RECFM=VB,LRECL=84,BLKSIZE=1684)
//FT14F001 DD UNIT=DKSCRA,DISP=(NEW,DELETE),SPACE=(TRK,20),
// DCB=(RECFM=VBS,BLKSIZE=1600)
//FT15F001 DD UNIT=DKSCRA,DISP=(NEW,DELETE),SPACE=(TRK,20),
// DCB=(RECFM=VBS,BLKSIZE=1600)
//FT16F001 DD UNIT=DKSCRA,DISP=(NEW,DELETE),SPACE=(TRK,20),
// DCB=(RECFM=VBS,BLKSIZE=1600)
//SYSUDUMP DD SYSOUT=A
//SYSIN DD *
```

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

APPENDIX C

SAMPLE OUTPUT FOR APPENDIX A DATA DECK

APPENDIX C

SAMPLE OUTPUT FOR APPENDIX A DATA DECK

MOCUS - - VERSION 3774

```

*****
-----
*
- MOCUS - - A COMPUTER PROGRAM TO OBTAIN MINIMAL SETS FROM FAULT TREES -
*
-----
*****

```

MOCUS INPUT EDIT

SAMPLE PROBLEM INPUT

* DATA

7 BOTH BOTH

END

SAMPLE FAULT TREE DESCRIPTION

* TREE

TOP OR 2 0 GATE1 GATE2

GATE2 OR 0 2 PE3 PE4

GATE1 AND 1 1 GATE3 PE1

GATE3 OR 0 2 PE2 HE1

END

EVENT HE1 IS A HOUSE

* RATES

HE1 1.0

END

* TESTS

TOP

GATE2

GATE3 NO PUNCH

END

SAMPLE PROBLEM INPUT

NUMBER OF GATES, NG----- 4

NUMBER OF COMPONENTS, NI----- 5

MAXIMUM SIZE OF SETS, ITMAX----- 7

CUT SET - PATH SET SWITCH, IDEX1-----BOTH

PRINT - PUNCH SWITCH, IDEX2-----BOTH

MOCUS - - VERSION 3774

SAMPLE PROBLEM INPUT

MOCUS - - VERSION 3/74

NAME	TYPE	INPUTS----
TOP	OR	2 0 GATE1 GATE2
GATE2	OR	0 2 PE3 PE4
GATE1	AND	1 1 GATE3 PE1
GATE3	OR	0 2 PE2 HE1

SAMPLE PROBLEM INPUT

MOCUS - - VERSION 3/74

COMPONENT NAMES AND FAILURE RATES (PER HOUR)

	COMPONENT NAME	LAMDA (FAILURE*INTENSITY/HR)	TAU
1	PE2	1.000000-06	0.0
2	HE1	0.0	1.000000-00
3	PE3	1.000000-06	0.0
4	PE4	1.000000-06	0.0
5	PE1	1.000000-06	0.0

SAMPLE PROBLEM INPUT

MOCUS - - VERSION 3/74

	COMPONENT NAME	NUMBER OF GATES INPUT	GATES INPUT BY COMPONENT
1	PE2	1	GATE3
2	HE1	1	GATE3
3	PE3	1	GATE2
4	PE4	1	GATE2
5	PE1	1	GATE1

MOCUS - - VERSION 3/74

HOUSES DETECTED IN INPUT TREE

	HOUSE	STATE
1	HE1	ON

IDENTIFICATION KEY FOR COMPONENT DATA WRITTEN TO UNIT 10

1 PE2

2 HE1

3 PE3

4 PE4

5 PE1

MOCUS - - VERSION 3774

MINIMAL CUT SETS FOR GATE TOP

CUT SETS WITH 1 COMPONENTS

- 1) PE1
- 2) PE3
- 3) PE4

TOTAL NUMBER OF CUT SETS FOUND WAS 3

ALL CUT SETS HAVE BEEN DETERMINED.

MINIMAL CUT SETS FOR GATE GATE2

CUT SETS WITH 1 COMPONENTS

- 1) PE3
- 2) PE4

TOTAL NUMBER OF CUT SETS FOUND WAS 2

ALL CUT SETS HAVE BEEN DETERMINED.

34

MINIMAL CUT SETS FOR GATE GATE3

THIS GATE IS FAILED (BECAUSE OF THE HOUSE EVENTS).

MINIMAL PATH SETS FOR GATE TOP

PATH SETS WITH 3 COMPONENTS

1) PE1 PF3 PE4

TOTAL NUMBER OF PATH SETS FOUND WAS 1

ALL PATH SETS HAVE BEEN DETERMINED.

35

MINIMAL PATH SETS FOR GATE GATE2

PATH SETS WITH 2 COMPONENTS

1) PE3 PE4

TOTAL NUMBER OF PATH SETS FOUND WAS 1

ALL PATH SETS HAVE BEEN DETERMINED.

MINIMAL PATH SETS FOR GATE GATE3

THIS GATE IS FAILED (BECAUSE OF THE HOUSE EVENTS).

THE W ARRAY IS DIMENSIONED FOR 7000 REAL*8 WORDS.

111 WORDS WERE USED AND THERE WERE 6889 EXCESS.

MOCUS USED 4 ROWS AND 4 COLUMNS IN THE ICS MATRIX.

EXTERNAL SCRATCH STORAGE WAS NOT REQUIRED.

APPENDIX D

MOCUS INPUT IN BRIEF

This appendix is a quick reference input guide to MOCUS input.

APPENDIX D

MOCUS INPUT IN BRIEF

TABLE D-I

INPUT DECK LAYOUT

1. TITLE CARD
2. {COMMENTS CARDS}
3. *bDATA (b = blank column)
4. {CONTROL PARAMETERS}
5. END
6. {COMMENTS CARDS}
7. *bTREE
8. {FAULT TREE DESCRIPTION}
9. END
10. {COMMENTS CARDS}
11. *bRATES
12. {FAILURE AND REPAIR DATA}
13. END
14. *bTESTS
15. {GATES FOR WHICH MINIMAL SETS ARE FOUND}
16. END

TABLE D-II

INPUT GROUP 1 -- CONTROL INFORMATION

FORMAT [I5, 5X, 2(A10)]

Number of Cards = 1

FORTTRAN Variable	Description
ITMAX	Maximum size of the minimal cut sets to be found.
IDEX1	Switch that determines whether minimal cut sets, minimal path sets, or both are to be found.
IDEX2	Switch that determines whether the minimal sets are to be printed, punched, or both.

TABLE D-III

INPUT GROUP 2 -- FAULT TREE DESCRIPTION

Number of cards = number of gates

Inputs	Card Columns
1. Gate name (any eight alphanumeric characters)	1 to 8
2. Blank	9
3. Gate type ("AND" OR "OR")	10 to 12
4. Blank	13
5. Number of gate inputs	14 to 15
6. Number of component inputs	16 to 17
7. First input name	19 to 26
8. Blank	27
9. Second input name	28 to 35
10. Blank	36

TABLE D-IV

INPUT GROUP 3 -- HOUSE EVENT OR HOUSE EVENT AND FAILURE DATA

House Event

FORMAT [2(A8, 12X, F10.3, 1X)]

Number of cards -- Variable

FORTRAN Variable	Description
NAM(I)	I^{th} house event name
TAU(I)	Switch to turn house on or off

Failure Data

FORMAT [2(A8, 2X, F10.6, F10.3, 1X)]

Number of cards -- Variable

FORTRAN Variable	Description
NAM(I)	I^{th} component name (must be same as in Input Group 2)
LAMDA(I)	I^{th} component failure rate (per 10^6 hours) or inhibit switch
TAU(I)	I^{th} component repair time or inhibit failure probability

LAMDA(I)	TAU(I)	Interpretation
Positive	Positive	Component failure with failure rate LAMDA(I) and constant repair time TAU(I)
	0.0 or blank	Nonrepairable component failure with failure rate LAMDA(I)
Negative 0.0 or blank	between 0.0 and 1.0	Inhibit failure with constant probability of failure TAU(I)

TABLE D-V

INPUT GROUP 4 -- GATES FOR WHICH MINIMAL SETS ARE OBTAINED

FORMAT (A8, 2X, A8)

Number of cards = number of gates for which minimal sets are obtained
(must be less than 21).

FORTRAN Variable	Description
NGTE	Alphanumeric name of gate for which minimal sets are to be found.
DUM1	Minimal set punch suppression flag. If DUM1 is any alphanumeric character the minimal sets for NGTE are not punched.