

*Arinc Research*

MONOGRAPH 11

DESCRIPTION OF THE  
COMPUTERIZED RELIABILITY  
ANALYSIS METHOD (CRAM)

David E. Van Tijn

November 13, 1964

GPO PRICE \$ \_\_\_\_\_

CFSTI PRICE(S) \$ \_\_\_\_\_

Hard copy (HC) \$4.00

Microfiche (MF) .75

FORM 602 65

**ARINC RESEARCH CORPORATION**

A SUBSIDIARY OF AERONAUTICAL RADIO, INC.



Publication 294-02-14-444

FACILITY FORM 602

<u>N66 34791</u> (ACCESSION NUMBER)	<u>1</u> (CODE)
<u>104</u> (PAGES)	<u>08</u> (CATEGORY)
<u>CR-77414</u> (NASA CR OR TMX OR AD NUMBER)	

ARINC Research Monograph 11

DESCRIPTION OF THE  
COMPUTERIZED RELIABILITY ANALYSIS METHOD (CRAM)

David E. Van Tijn

November 13, 1964

ARINC RESEARCH CORPORATION  
a subsidiary of Aeronautical Radio, Inc.  
1700 K Street, N. W.  
Washington, D. C. 20006

Publication 294-02-14-444

© 1964 ARINC Research Corporation

## FOREWORD

This monograph describes the Computerized Reliability Analysis Method (CRAM), developed by ARINC Research Corporation in conjunction with work performed for George C. Marshall Space Flight Center, National Aeronautics and Space Administration, under Contract NAS8-11087. The Corporation acknowledges with appreciation the support given by NASA to this project.

In concept, CRAM can be used to analyze all systems in which a failure of one part can be defined without reference to any other part. The technique, in brief, is as follows: On the basis of a system description, the operating modes are identified and the system is partitioned into subsystems for individual analysis. Reliability diagrams are prepared for the individual subsystems, and data on the subsystem elements and failure probabilities of the element classes are tabulated. The first computer program converts the coded reliability diagrams into a formula representing the condition for system success; this formula in turn is converted by the second program into a reliability model. From this model and the data tables, the third program computes a reliability value for the subsystem. The subsystem numbers are then combined into a single value representing the system.

This publication continues the series of monographs in which ARINC Research staff members have discussed various facets of the systematic study of reliability, maintainability, and other factors influencing system effectiveness. Monograph 11 is concerned mainly with detailed procedures for using computers to perform the tedious calculations required in reliability prediction. Thus the monograph has some of the aspects of a handbook of instructions. It is aimed at personnel who are themselves involved in the work of reliability prediction; and it assumes that the reader has some previous understanding of the concept of reliability and reliability prediction methods.

## CONTENTS

	<u>Page</u>
FOREWORD	iii
1. INTRODUCTION	1
1.1 What Is CRAM?	1
1.2 Basic Benefits	1
2. THE APPLICABILITY OF CRAM	3
2.1 Types of Systems	3
2.2 Advantages and Disadvantages of CRAM	5
2.3 Work Flow in CRAM	7
3. BASIC RULES OF PROCEDURE	11
3.1 Basic Rules for Dividing the System	12
3.2 Scheduling	14
3.3 Checking and Proofreading	17
3.4 Conclusion	18
4. ENGINEERING ANALYSIS	19
4.1 General Discussion	19
4.1.1 Diagrams	20
4.1.2 Rules and Convenience Blocks	20
4.1.3 Operating Modes	23
4.1.4 Dividing the System into Subsystems	25
4.2 Naming of Blocks	26
4.2.1 Element Names	27
4.2.2 Coding Operating Modes	29
4.3 Preparing Listings	33
4.3.1 Element Table	34
4.3.2 Failure Information Table	36
4.4 Constructing Reliability Diagrams	39
4.4.1 General Discussion and Terminology	39
4.4.2 Rules for Constructing Reliability Diagrams	43
4.4.3 Feedback Systems	47

	<u>Page</u>	
4.4.3.1	Introduction	47
4.4.3.2	Definitions	48
4.4.3.3	The "Cutting" Operation	49
4.5	Failure-Mode-and-Effect Analysis	52
4.6	Examples of the Application of CRAM	55
4.6.1	A Series-Parallel System	55
4.6.2	A Feedback Circuit	60
4.7	Reading Printouts	66
4.7.1	Engineering Printouts	66
4.7.2	Intermediate Printouts	69
5.	PREPARING COMPUTER INPUTS	73
5.1	Coding Diagrams	73
5.1.1	Marking the Diagram	74
5.1.1.1	Numbering Blocks	74
5.1.1.2	Coding the Block Types	76
5.1.2	The Element Designator Format	79
5.1.3	Preparing Diagram Code Sheets	79
5.2	Inputs to Part 1 and Part 2 Computer Programs	80
5.3	Coding Element Failure Information	80
5.3.1	Operating Modes	80
5.3.2	Failure Information	82
5.3.3	Element List	82
5.4	Inputs to Part 3 Program	82
APPENDIX:	COMPUTATION OF RELIABILITY EXPRESSIONS AND FUNCTIONS	83
LIST OF DEFINITIONS		93

## FIGURES AND TABLES

Figure		Page
1	WORK FLOW IN CRAM	8
2	SCHEDULING CHART FOR SYSTEM ANALYSIS	15
3	CONVENTIONAL AND CRAM DIAGRAMS FOR A REDUNDANT SYSTEM	22
4	LIST OF OPERATING MODES	30
5	REPRESENTATION OF DIAGRAM BLOCK IN COMPUTER	31
6	ELEMENT TABLE	35
7	PART CLASS FAILURE INFORMATION TABLE	37
8	BRANCHING FOR OPERATING MODES OF AN ELEMENT	40
9	LOGICAL CONNECTIONS ON DIAGRAMS	42
10	CONSTRUCTION OF A RELIABILITY DIAGRAM	46
11	A FEEDBACK SYSTEM	49
12	TREE DIAGRAM OF FIGURE 11	50
13	A CYCLE WITH TWO INPUTS	51
14	ILLUSTRATIONS OF RULE 4, STEPS 1 AND 2	53
15	SCHEMATIC DIAGRAM OF AN ENGINE CUTOFF CIRCUIT	56
16	RELIABILITY DIAGRAM OF AN ENGINE CUTOFF CIRCUIT	58
17	TABLES FOR ENGINE CUTOFF CIRCUIT	61
18	FEEDBACK CIRCUIT (FLIP-FLOP)	62
19	TABLES FOR FEEDBACK CIRCUIT (FLIP-FLOP)	65
20	FORMAT OF PRINTOUTS OF PART 3 PROGRAM	67
21	PARTIAL PRINTOUT FOR FEEDBACK CIRCUIT OF PART 3 PROGRAM (FLIP-FLOP)	68
22	PART 1 PRINTOUTS	71
23	TEST RELIABILITY DIAGRAM FOR CRAM PART 1	75

Figure		Page
24	CODE SHEETS FOR DIAGRAMS	78
25	SHEETS FOR FAILURE INFORMATION	81
A.1	RELIABILITY DIAGRAM	89

Table		Page
1	COMPUTER SYMBOLS FOR LOGICAL SIGNS	69
2	TYPES OF BLOCKS ON A RELIABILITY DIAGRAM	76
3	COMBINATIONS OF BLOCK TYPES	77



## 1. INTRODUCTION

### 1.1 What Is CRAM?

The term CRAM is an acronym for Computerized Reliability Analysis Method; CRAM, the concept, is a method for analyzing reliability by the use of computers.

What distinguishes CRAM from other computer-based analysis methods is that the computer programs used in CRAM are strictly utility routines; no further programming is needed to make them usable. The reliability diagrams produced in the preparatory analysis are converted to inputs to the first computer program, which, in conjunction with the second program, turns out a conventional reliability model. That is, if  $R$  is a name for the system reliability, and if  $R_1, \dots, R_n$ , are names for the reliabilities of subsystems or parts, then the first two computer programs produce an equation of the form

$$R = \sum \pm (\text{products of } R_j) \quad (1)$$

A third program takes Equation 1, together with information on the failure probabilities of the subsystems or parts, and produces a numerical estimate of the system reliability; for example,

$$R = 0.999954575 \quad (2)$$

The information about failure probabilities may be provided either as a probability or as a rate, complete with time data and  $K$  factors.

### 1.2 Basic Benefits

The use of computer programs for constructing reliability models and computing reliability relieves the analyst of much of the drudgery involved in conducting analyses. As a result, he can concentrate on technical tasks such as producing reliability diagrams and obtaining information on failure probabilities. Furthermore, the machine does the shuffling of models and the computation so rapidly that the complexity of computation ceases to be a factor. If a model

is to be worked out by hand, unusual modes of possible failure often must be neglected, to keep the work within bounds. Experienced reliability engineers know how to estimate such possibilities without impairing the integrity of the answers, but making such estimates requires thought and time, as well as a background of experience.

It is noteworthy, also, that any possibilities neglected in a reliability model must be accounted for in a full failure-mode-and-effect analysis (FMEA)<sup>†</sup>. Therefore, the "manual" method tends to create differences between the FMEA and the reliability model. When computer programs are used, the balance of economy shifts the other way: It becomes less trouble to include a possibility in the diagram than to find reasons for omitting it. If, in fact, no possibilities are omitted from the diagram, then the reliability model can be used to prepare an FMEA, again on a computer.

---

<sup>†</sup> For a definition of a failure-mode-and-effect analysis, see page 52. A list of definitions follows the appendix of this monograph.

## 2. THE APPLICABILITY OF CRAM

There are limitations on the applicability of CRAM, in that not all systems can be converted into the kinds of diagrams which the computer programs accept. Section 2.1 describes the types of systems to which CRAM can be applied.

Even for a system to which CRAM can be applied, one may not wish to use all the machinery of a formal method. The decision whether or not to use CRAM depends on circumstances, but, for the guidance of prospective users, Section 2.2 will discuss the advantages and disadvantages of the method and the manner in which they depend upon the size of the system.

A particularly noteworthy point is that, once the decision is made to use CRAM, then one is committed to a particular sequence of tasks. A brief description of the resulting work flow, and what is entailed in the performance of each task, will be presented in Section 2.3.

### 2.1 Types of Systems

The feature of CRAM which establishes the basic limitation on the applicability of this analysis method is the mathematical reliability model produced by the second computer program. CRAM is applicable only to systems which can be represented by such a model.

Reliability models are formulas in the propositional calculus; they are equivalent to Boolean formulas. (See the discussion in the Appendix.) For instance, if s is a system consisting of subsystems a and b in series, then the model is

$$S^* = A \text{ and } B \quad (3)$$

where the capital letters denote that the corresponding subsystems work for the required length of time and the \* denotes that the entire system works. Equation 3 is a propositional function. If s

consists of a series path through a, and parallel paths through b and c, then the model is

$$S^* = A \text{ and } (B \text{ or } C) \quad (4)$$

In the foregoing interpretation, the concept "a works" needs explanation. The subsystem a performs a function in the system s. If s is a hardware system, then some physical entity (a current or gas) enters a at one end, is operated upon by a, and emerges in changed form at the other end. If the system is to work, the form of the physical entity as it leaves a must be correct. Sometimes it is possible to know whether or not the output of a is correct without reference to the other parts of the system -- for instance, if the function of a is to produce an electrical pulse of a certain shape and size, or if a is a valve which should close against a given pressure. At other times it may be impossible to decide whether the output of a is correct without knowing how other elements are operating -- for instance, if three batteries, each with a rated capacity of 12 amp-hr, are together required to deliver 30 amp-hr. The system can function with one of these batteries well below its rated output, provided the others are up to their rating. If the elements are made to specifications, however, and we arbitrarily say that an element works in the system if it meets its specification, then we are in the situation of not needing to refer to other parts of the system.

The two cases -- systems in which the correctness of element outputs can be ascertained without reference to the other elements, and those in which this is not possible -- lead to very different models. In the first case, CRAM can be used and the system can be modeled within the propositional calculus. In the second case, sophisticated analytical models are necessary; CRAM cannot be used.

From the above discussion it is evident that CRAM can always be used if the element operating mode is determined with regard only to the specifications. This feature makes the method particularly appropriate for systems analysis where the elements are subsystems rather than piece parts.

CRAM imposes no limitation on the number of different operating modes of an element that can be considered. For instance, a diode may be in any one of several "failure" modes -- such as "open," "shorted," or "noisy" -- some of which may not prevent satisfactory circuit performance. All these failure modes can be accommodated separately in the analysis. Similarly, if a subsystem can operate in a number of different modes, these modes can be distinguished and their different effects on the system taken into account.

Further restrictions on the use of CRAM which do now exist are not inherent to the method but reflect the degree of completion of the computer programs. The programs as now constituted can handle only series systems or systems with active redundancy, which do not change their configuration with time. The restrictions are in process of being removed, and, when they are, CRAM will be applicable to all systems which satisfy the requirement of independence of operating modes among the elements, including

- (a) systems with standby redundancy
- (b) systems with repair capability
- (c) systems whose configuration changes with time

## 2.2 Advantages and Disadvantages of CRAM

The diagrams used in the CRAM method are different in appearance from the usual type of reliability diagrams. Because each block is permitted only one successor, the diagram will have more blocks and give the impression of being more complex. In fact, CRAM diagrams are structurally simpler than conventional reliability diagrams. It takes some practice, however, to learn to apply the rules easily. In ARINC Research experience the learning time is about 2 weeks; after that period, an analyst will produce and read CRAM diagrams as easily as other types.

Both the advantages and the disadvantages of CRAM follow from its formality. Tables of elements must be completed, with the name of each element accompanied by the name of the element class to which it belongs. For each element class, failure information must be provided in another table. Both the tables and the diagram must be copied on code sheets and punched, and finally the computer programs must be run. A substantial amount of clerical labor is involved in

this process, and it is this clerical labor that is traded off against the engineering time. Consequently, for simple systems with little redundancy, it is probably easier for the analyst to write the reliability model by hand and compute the reliability on a desk calculator than to incur the administrative time needed to use the computer.

On the other hand, if the system becomes complex or incorporates much redundancy, or if the elements can operate in many different modes which have different effects on system operation, then the formal procedures of CRAM begin to pay off. Again, if the system considered is part of a larger system and subsystems are used in several contexts, the capability of CRAM to account for different uses of the same subsystem becomes very useful. Finally, CRAM provides a permanent system model, together with a permanent record of the assumptions about element failure probabilities. This model can be readily updated. It can also be used for reassessing the system success probability when more information becomes available. If these considerations are important, one would tend to use CRAM.

This discussion of when to use CRAM can be summed up as follows:

(1) One would tend to use CRAM for complex systems incorporating a substantial amount of redundancy, or for multimode systems, or for parts of such systems, or for systems which would be subject to re-evaluation or reassessment.

(2) One would tend not to use CRAM for relatively simple systems, or for systems which are primarily series systems and which are being evaluated on a one-time-only basis.

(3) The dividing line between these classes must be determined in each case.

Section 2.3 outlines in sequence the work that is required in a CRAM analysis and that must be considered in the decision whether or not to use CRAM.

### 2.3 Work Flow in CRAM

Figure 1 displays the tasks that must be performed in a CRAM reliability analysis. Most of these tasks are standard ingredients of any analysis, and detailed descriptions may be found in textbooks on reliability.†

First, a good system description must be obtained. This task usually requires considerable effort, including much searching of literature, conferences with designers, and collating of scattered material and information. The resulting system description may be a collection of schematics, or function diagrams, or block diagrams, or a combination of all three.

Next, the system must be analyzed to identify (1) the different purposes it may serve as it operates in different modes, and (2) the different combinations of subsystems through which each purpose may be accomplished. Each system purpose corresponds to a separate system-success mode, and for each of these modes all combinations of subsystem operating modes which can accomplish the purpose must be collected on a separate diagram.

During this work, it will become clear how the system can be divided into subsystems so that the subsystems operate or fail independently of one another. This partitioning concludes the initial system analysis.

Once the subsystems have been defined, it is necessary to isolate the different modes in which they operate and to collect into sets those which lead to the same system-success modes. For each set of subsystem operating modes which can lead to one system-success mode, one diagram is needed.

All the tasks discussed above are really concerned with defining "success" for the systems. Once this has been done, the reliability analysis of the subsystems can be initiated. The analysis will show what combinations of element operating states are necessary to assure subsystem success.

---

† See, for example, ARINC Research Corporation's Reliability Engineering, William H. von Alven, ed., Englewood Cliffs, N. J., Prentice-Hall, Inc., 1964.

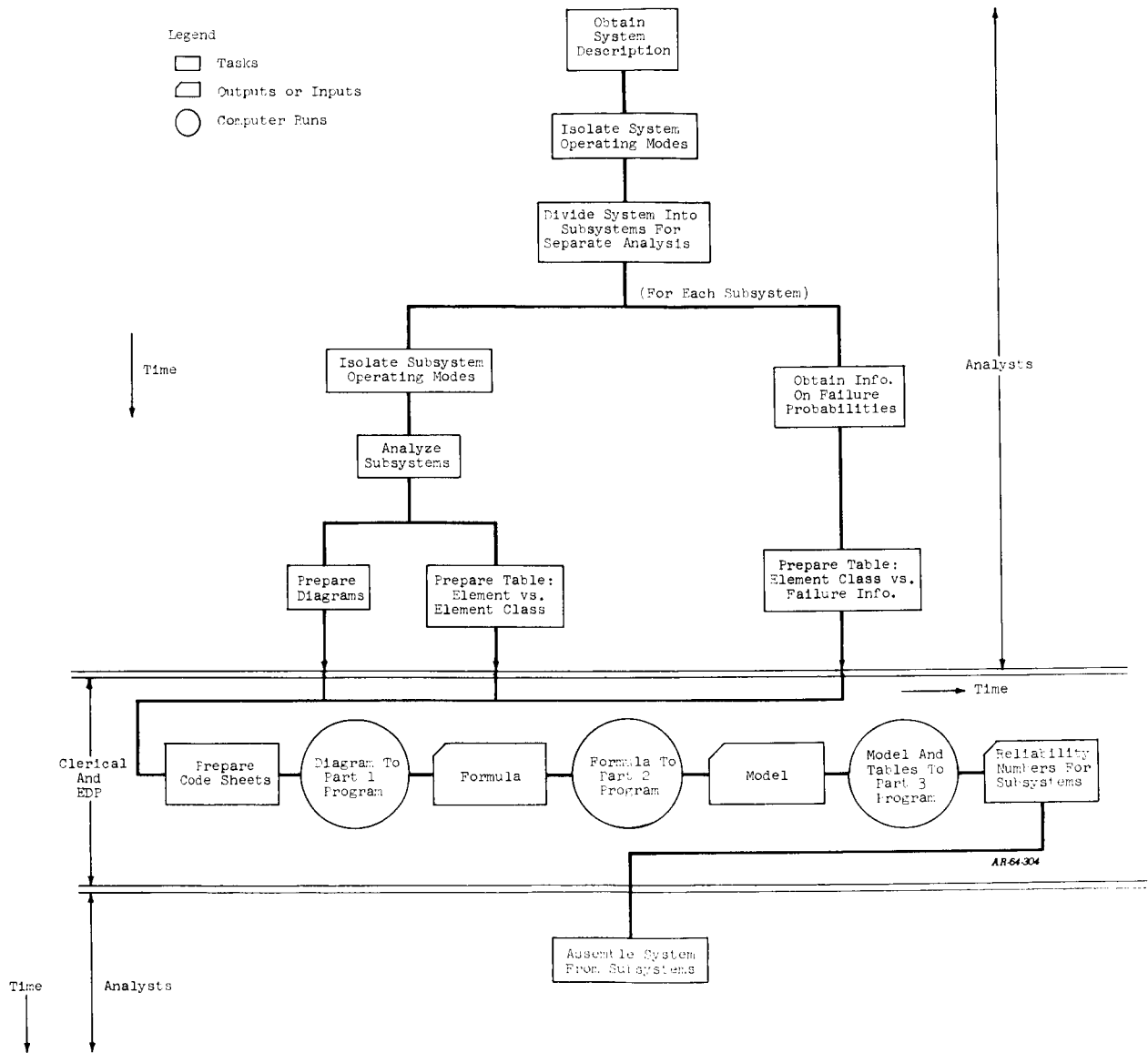


FIGURE 1  
WORK FLOW IN CRAM



The rules for constructing CRAM diagrams envisage a step-by-step analysis of the subsystem, starting with the outputs and working back toward the inputs. Experience has shown that this procedure -- working step by step from outputs to inputs -- is not only a convenient way of drawing diagrams but also helps to ensure that no possibilities are overlooked in the analysis.

The diagrams are one output of the analysis. The other outputs are:

- (1) A list of element operating modes
- (2) A list of elements, and, for each element, the class to which it belongs
- (3) A list of information from which, for each element class, the failure probability can be computed.

An element class is any group of elements which have the same failure probability. A failure probability may be given directly as a probability or it may be given in the form of a failure rate, a time, and K factors. Because different K factors or times will lead to different failure probabilities, separate element classes must be made up for elements which have different K factors, even if they otherwise belong to the same class.

The three outputs of the analysis -- diagrams, failure-information tables, and element lists -- are the inputs to the three computer programs (identified as the Part 1, Part 2, and Part 3 programs). Accordingly, these outputs must be transferred to code sheets, from which cards will be punched.

The cards representing the diagrams are the inputs to the Part 1 program. The output of this program is a formula representing the condition for system success. This formula, however, is not in a form suitable for reliability computations. For instance, if successful system operation requires a and either b or c, the formula would be†

$$A \text{ and } (\underline{B} \text{ or } \underline{C}) \quad (5)$$

---

† Capital letters represent the supposition that the element denoted by the corresponding small letter operates in the required mode.

The probability of success of this system would be

$$+ P_A P_B + P_A P_C - P_A P_B P_C \quad (6)$$

where  $P_A$ ,  $P_B$ ,  $P_C$  represent the success probabilities of a, b, and c, respectively. The Part 2 program converts formulas such as Equation 5 into series of products like that in Equation 6, and at the same time removes any duplications which may arise; also, it deletes terms which, in effect, would require one element to operate in two different modes at the same time.

The Part 3 program accepts equations such as Equation 6 and the element-list and failure-information tables, substitutes the numbers, and calculates the system reliability.

The final task, reassembling the system from its subsystems, requires little comment. This can either be done by hand, or, if diagrams have been made for the different system modes in terms of the subsystem modes, it can be done on the computer. In the latter case the elements will be the subsystems, and their probabilities will be the outputs of the subsystem analyses.

### 3. BASIC RULES OF PROCEDURE

Any reliability analysis requires the painstaking performance of all the tasks above the first double line in Figure 1. If the organization and monitoring of these tasks is different when CRAM is used, this difference is a consequence only of the fixed format of CRAM outputs and the greater ease with which CRAM separates and recombines subsystems.

Because of the fixed format of the diagrams and other outputs of the CRAM analysis, it is much easier to divide a system into subsystems -- each of which may be assigned to a different analyst -- and then to recombine the results. When this is done, however, a careful control must be exercised to ensure that the subsystems recombine correctly. Also, the scheduling must be carefully managed, since one subsystem is usually an input to another and the reliability of the first must be known before that of the second can be computed.

The ease with which different element operating modes can be accounted for on a CRAM diagram tempts the analyst to record a complete analysis of his subsystem. This feature is desirable, since it tends to provide a model that is more faithful to the original; however, it also necessitates a careful monitoring system to ensure that no mistakes, and especially no typographical errors, are allowed to enter the computer inputs. In general, one of the chief preoccupations in the management of a CRAM analysis will be to maintain accuracy throughout, and procedures for achieving this aim must be established.

The actual job of dividing a system into subsystems is an engineering function; it will be discussed in Section 4.1.4 of this monograph. Management's function is to organize the work so that it can be conducted without confusion. The remainder of Section 3 seeks to ease the management task by presenting several procedural rules and a suggested work schedule. This material states certain

fundamentals of the engineering-analysis work described in Section 4, shows the elements of CRAM in perspective, and emphasizes precautionary measures that are of major importance.

### 3.1 Basic Rules for Dividing the System

The computer programs operate on the names of elements. If two elements appearing on the same diagram are given the same name, the programs will treat them as different occurrences of the same element, and derive the model accordingly. Conversely, if the same element appears twice on a diagram, but under different names, the program will act as if there were two distinct elements. The same principle applies to subsystems. Thus, if two subsystems have a group of elements in common, but the subsystems are given different names, the computer will act as if the common group of elements is in fact two separate groups.

From these considerations we obtain the first two procedural rules (PR):

- PR 1. No two elements in the system may have the same name.
- PR 2. No two subsystems may have elements in common; i.e., elements in common between two subsystems should be broken out as a separate subsystem.

To ensure an orderly analysis and, also, to determine the facts relating to PR 2, above, it is necessary to know how all the subsystems fit together, before they are analyzed. The best way to ensure that this is done is to construct CRAM-type diagrams for the system, with the subsystems appearing as elements. These diagrams provide the basis for the next procedural rule:

- PR 3. Before the subsystem analysis is initiated, construct diagrams of the system modes of interest, in which the subsystem modes of interest are elements.

This procedure also serves as a mechanism for ensuring that PR 1 and PR 2 have been complied with. These system diagrams should be as explicit as possible about the requirements on the subsystem. For instance, if the function of a subsystem is to produce a signal, the block should be labeled with a "correct signal" indication, rather than as "OK". (The precise method of labeling operating modes is explained in Section 4.2.2.) If there are nonsystem inputs,

these should be shown on the diagrams, so that the system diagrams on which the subsystems are blocks will be in all respects like the subsystem diagrams on which the elements are blocks.

The analyst must have a clear understanding of the symbolism used in the diagrams; otherwise much confusion can result. Throughout the analysis if a, for instance, is an element, then A/OK represents the proposition that a has not failed in any way, and A\*/OK represents the proposition that a has not failed and that all the necessary inputs for a are available. The same applies to a subsystem. If, for example, "301/cs", representing the correct signal output of a subsystem, appears as a block on the system diagram and as a terminal block on the subsystem diagram for subsystem 301, then "301\*/cs" can be interpreted in two ways:

On the system diagram it would mean that subsystem 301 and all subsystems which provide 301 with inputs are working correctly. ("Working correctly" in this context means "providing the right output"; i.e., a system may work correctly and yet have failed elements.)

On the subsystem diagram it would mean that all the predecessors shown on the subsystem diagrams are working correctly.

These two interpretations lead to the same formulas if all the subsystems which provide inputs to the subsystem 301 are so shown on the diagram for 301. However, if this is done, the reliability evaluation of 301 must await completion of the reliability evaluations of the inputs. Furthermore, no evaluation of the subsystem 301 by itself will ever be produced. It is better, therefore, to produce a model for a subsystem in which the inputs obtained from other subsystems are not mentioned, and to rely on the system diagrams to perform the integration among subsystems. The next procedural rule is, therefore:

PR 4. On subsystem diagrams the subsystems which provide inputs should be shown, but the latter subsystems should not appear in the model. The subsystem name without a star (e.g., 301/cs) shall designate the conditional event that the subsystem operates correctly, if all the subsystems providing inputs operate correctly.

### 3.2 Scheduling

Figure 2 represents a scheduling chart that has proved a valuable device for management of the analysis of a large system. The chart may apply either to the entire system in one of its modes of interest, or to one of the subsystems, since subsystems too may be broken up into sub-subsystems before being analyzed into elements.

In the first column are listed the subsystems with their required operating modes if the chart is for a system; the sub-subsystems if the chart is for a subsystem. This column therefore contains the full description of the blocks on the corresponding diagrams.

The second column lists the names that have been given to the blocks on the diagram. The first two columns together serve also as a "dictionary" for subsystem names. For example, they would show what subsystem is represented by "301".

The third column contains entries only for those subsystems which are further broken down into sub-subsystems. Each subsystem of this type will then have its own scheduling chart, and the number of the chart will appear in this column.

The remaining columns refer to the tasks detailed on Figure 1. They are, in order:

Column 4	Description complete	This heading refers to the system or description from which the diagram will be made. Note that it is possible, and indeed common, to have sufficient information to construct the system diagram in terms of subsystems before all the details on each subsystem are known.
Column 5	Diagram complete	This heading is self-explanatory. Note that, by PR 3, the system diagrams must be complete before the subsystems can be entered in columns 1 and 2.
Column 6	Diagram checked	The need for checking is discussed in Section 3.3.
Column 7	Diagram to drafting	If a report is required, its production should be scheduled concurrently with the analysis.



Column 8	Diagram coded and punched	
Column 9	Diagram cards checked	See Section 3.3.
Column 10	Element table complete	For a system diagram, this heading refers to a table of subsystems; for a subsystem analyzed into sub-subsystems, it refers to a table of the latter. The items in the table will consist of names of blocks and the classes to which these blocks belong.
Column 11	Element table checked	
Column 12	Element table coded and punched	
Column 13	Element table cards checked	
Column 14	Element table to typing	
Column 15	Class failure table number	It is quite common for several subsystems to consist of the same element classes. In this case only one table needs to be made for the group.
Columns 16-20		These columns are for the tables of failure information for the element classes. The steps are the same; preparation, checking, punching and checking, table to typing.
Column 21	Part 1 program run	The Part 1 program uses only the diagram cards.
Column 22	Part 2 program run	The Part 2 program uses only the output of the Part 1 program.
Column 23	Part 3 program run	The Part 3 program uses the output of Part 2 and the two tables referred to in Columns 10 and 15.

The procedure for use of the chart is to enter the due date and the initials of the responsible staff member in the block and, as the work is completed, to crosshatch those blocks which are complete. The chart thus will show at a glance which subsystems are lagging behind.

In scheduling the computer runs, the analyst should bear in mind that they are usually brief. (An exception is the Part 2 program for very large diagrams, which may take several hours.)



Therefore, it may be more economical to schedule a number of runs of Part 1 followed by a number of runs of Parts 2 and 3, rather than running each subsystem as the inputs become available. It is in general more efficient, and therefore cheaper, to make a number of runs with one program than to take one set of inputs through a series of programs, and then another, and so on. Experience must show, and pressure of due dates determine, to what extent such economies of computer time can be effected.

### 3.3 Checking and Proofreading

The need for accuracy in reliability analysis is not peculiar to CRAM. However, the greater formality and the greater division of labor characteristic of CRAM make it both possible and more necessary to establish procedures for guarding against analytical and typographical errors. The recommended procedures, comprising five parts of PR 5, are as follows:

- PR 5.1. Diagrams should be checked against the system description by a second analyst.
- PR 5.2. Lists of element names vs. element classes should be checked
  - (a) against the diagram for completeness of the list and accuracy of the names
  - (b) against the original system description, as a second check on names, and for the element class to which each element belongs
- PR 5.3. Lists of failure information for element classes should be proofread against the original source.
- PR 5.4. Listings of diagram cards should be proofread against the diagram, as should listings of cards of the element vs. class tables. Listings of failure information should be proofread against the original source.
- PR 5.5 The printout of the Part 3 program should be carefully examined by the original analyst.

If these steps are followed, inaccuracies should be kept to a minimum. It must be stressed that the computer program works only on what it sees. Many mistakes in the inputs will cause it to halt, but if a wrong part-class or operating mode has been coded, the

program will continue the computation with this mistake and supply wrong answers. For this reason the checking and proofreading procedures of PR 5 are of the utmost importance.

### 3.4 Conclusion

The procedural rules and scheduling chart given in Sections 3.1-3.3 are intended as guides to the management of a reliability analysis. The accompanying discussion points out some of the characteristics of CRAM which have led to difficulties in the orderly progression of tasks in the past. The guidelines given will not substitute for attentive management, nor will they solve all difficulties which may arise. They have, however, proved helpful in the past, and it is hoped they can be further developed to provide more help in the future.

## 4. ENGINEERING ANALYSIS

### 4.1 General Discussion

This introduction to the engineering-analysis portion of the CRAM method will consider some of the more general characteristics of CRAM, in comparison with those of the classical reliability analysis method. The differences between the two methods are actually rather small, and this is as it should be, since the classical method produces good results. Perhaps the best way to look at CRAM is as a sharpening of the classical method; the same tasks must be performed by the analysts and the same information must be available. The only difference is that in CRAM, the record of the analysis (the diagram) and of the auxiliary information (kinds of elements and their failure rates) is produced by explicitly stated rules, and hence will be available in a standard form.

The price paid for these advantages of CRAM is that experienced analysts must take the trouble to become familiar with the rules and with the standard forms. The payoff to the analyst is that it relieves him of not only the administrative burden of recording the information but also the computational burden of constructing reliability models from the diagram and computing reliabilities from these models and the failure information.

The payoff to management is that, with the standardization of format and the easing of the administrative and computational load, it becomes much easier to divide systems into subsystems for analysis and to recombine the results. Hence, with the same management effort, larger systems can be subjected to a more detailed analysis than is possible with a less formalized approach.

A point discussed in Section 2.2 is worth restating here: By no means will all systems repay the effort required by the increased formality of the CRAM technique.

#### 4.1.1 Diagrams

"A reliability block diagram may be considered a logic chart which, by means of the arrangement of blocks and lines, depicts the effect of failure of items of equipment on the system's functional capability." †

Any reliability block diagram must satisfy this definition. It must be remembered that such diagrams form the basis of reliability analysis; in fact, the theory of reliability analysis is a theory about reliability diagrams.†† Therefore the method of constructing diagrams is at the root of any method for reliability analysis.

Reliability Engineering, the source of the above-quoted definition, continues with a description of what a reliability diagram should look like:

"Items whose failure causes system failure are shown in series with other items. Items whose failure causes system failure only when some other item has also failed are drawn in parallel with the other items." †

The instructions given in that text are in the nature of conditions which the completed diagram must satisfy. They allow wide latitude in the type of diagram used, and CRAM diagrams as well as the more conventional types satisfy these requirements. In a sense, the rules of CRAM take up one of the options allowed under the instructions.

#### 4.1.2 Rules and Convenience Blocks

CRAM diagrams differ from conventional ones in two respects:

(1) The blocks refer to operating modes of elements, rather than to the elements themselves. This is a natural extension of the concept of a "failed" element; the element has not failed if

---

† ARINC Research Corporation, William H. von Alven, ed., Reliability Engineering, Englewood Cliffs, N. J., Prentice-Hall, Inc., 1964, pg. 286.

†† David E. Van Tijn, "On the Systematic Construction of Reliability Expressions and Functions", paper submitted to Operations Research, September 1964.

it is in one of the operating modes specified in the block. What this means is that it is still performing its function in the system, though it may well have failed in the conventional sense.

(2) As far as the connections are concerned, CRAM diagrams do not have split outputs. Every block except the last has exactly one successor block. This feature is necessary if the Part 1 computer program is to be used; however, it is a feature that is easily attained. Any conventional diagram of the series-parallel type can be converted to a CRAM diagram by a simple repetition of branches. Figure 3 is a comparison of the two types of diagrams.

The real difference in CRAM as a method is in the "rules for constructing diagrams". These rules provide a procedure for converting a system description into a reliability diagram. They impose a step-by-step analysis. The analyst asks himself, first:

"What element outputs are needed to enable the system to operate?" and then:

"What combination of inputs does this element need to be able to function?"

and, finally, since the input to one element is the output of a preceding one:

"How must this (preceding) element operate to provide the outputs required as inputs to the element just examined?"

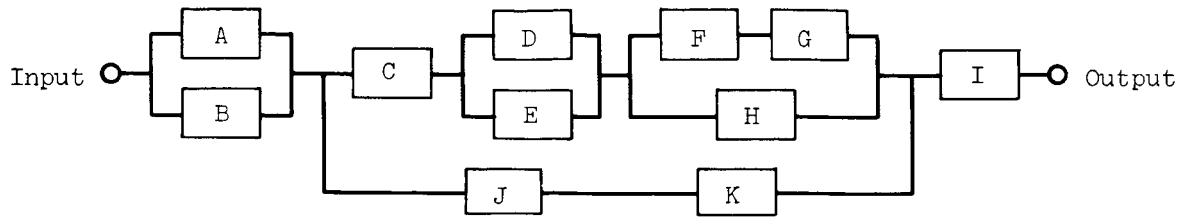
By asking this sequence of questions for each element, the analyst is automatically guided into the construction of reliability diagrams in which the blocks mention explicitly the permissible operating modes of elements and in which all series-parallel systems produce tree diagrams. Furthermore, analysis by this method is somewhat easier than to look at an element and ask the question:

"If this fails, what happens to the system?"

For those systems to which CRAM applies,<sup>†</sup> the two approaches yield the same result.

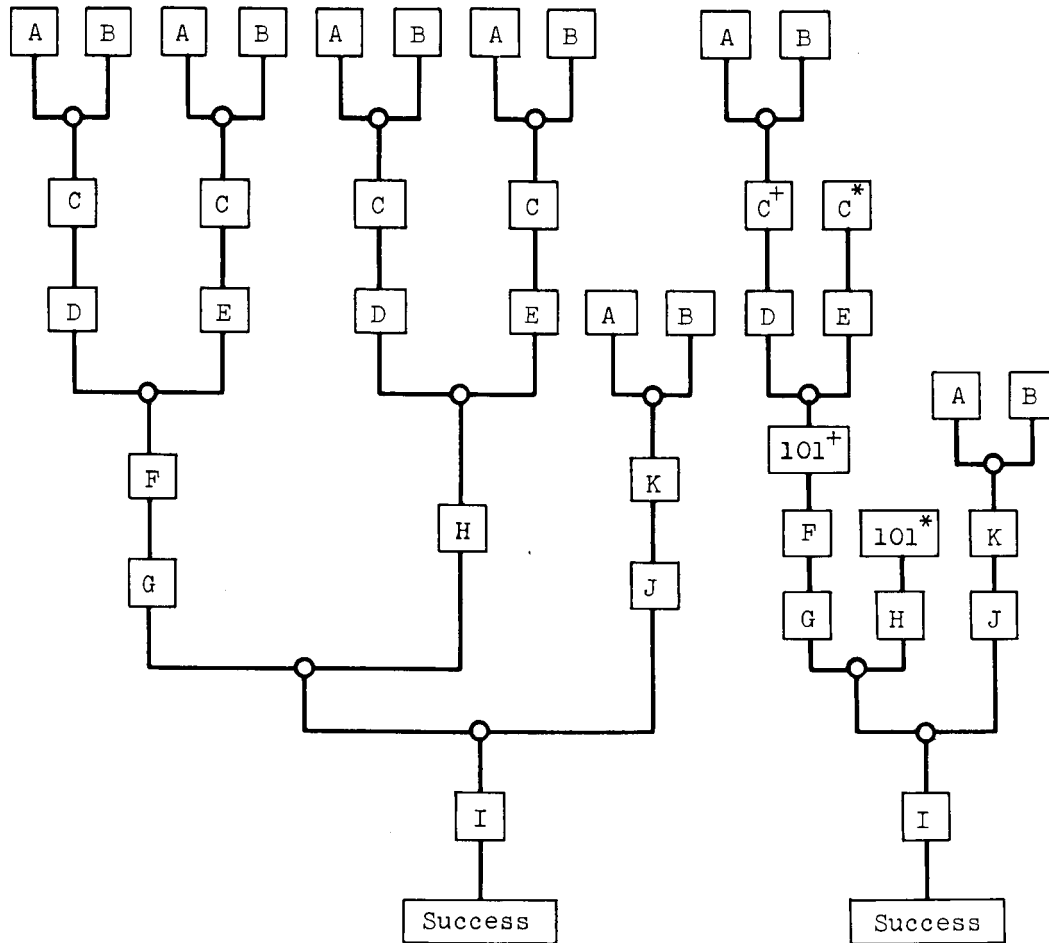
---

† See Section 2.1.



(a) Conventional Diagram

AR-64-362



(b) CRAM Diagram  
without Convenience Blocks

(c) CRAM Diagram  
with Convenience Blocks

FIGURE 3

CONVENTIONAL AND CRAM DIAGRAMS FOR A REDUNDANT SYSTEM

The rules now do two things: They provide a notation for the logical combination of the inputs, which each element needs, and, in case the system is a feedback system, they provide a method for converting the reliability diagram into a series-parallel diagram which

will lead to the correct model. A proof exists<sup>†</sup> that this can always be done if the requirement of independence of element operating modes is satisfied.

A notational device is provided by the optional rule, in the "convenience blocks". Convenience blocks are used if branches of the diagram must be repeated to avoid the split outputs of conventional diagrams. They are a name for the branch. A convenience block may represent either an element and all its predecessors, or it may be inserted in any line of the diagram. In the latter case it represents the elements which feed into this line, and all their predecessors, but there is no element corresponding to the convenience block. In Figure 3(c), C\* is a convenience block representing element c and its predecessors; 101 is a convenience block inserted into the diagram, without a corresponding element.

Both types of convenience blocks are used in the same way. The first time they are used, when they are shown with all their predecessors a "+" is added to the name of the block. This symbol will cause the computer to put the whole expression for the block into a special storage. Then the next time the branch is needed as an input to an element, the same name is repeated, but with a "\*" added to it. This sign will cause the computer to call the expression for the branch from the special storage.

The convenience blocks serve the valuable purpose of restricting the number of extra blocks to one for each place where the outputs of a block on a conventional diagram would be split. If they were not used, the number of blocks on a CRAM diagram would tend to become very large. The effect of convenience blocks may be seen by comparing Figure 3(b) with Figure 3(c).

#### 4.1.3 Operating Modes

In a reliability analysis, one takes the different possible system performances of interest and assigns to each a probability of occurrence, which is based on the probabilities of the different failure and success modes of the elements during the time of interest.

---

<sup>†</sup> Van Tijn, op. cit. (footnote page 20).

The different system performances must be carefully extracted from the function of the system in prospective missions. For example, the performances of interest for the following systems are:

- (1) Manned space vehicle
  - (a) Success, as planned, or
  - (b) Enough thrust for alternate mission, or
  - (c) Enough time for successful abort, or
  - (d) Catastrophic malfunction
- (2) Warning system
  - (a) Correct warning of an actual malfunction, or
  - (b) Incorrect warning of a malfunction when there is none
- (3) Bombing-navigation system
  - (a) Two methods of visual bombing, or
  - (b) Two methods of instrument bombing

(Each of these four methods places different demands on the equipment.)

Because the value desired is the probability of each of these performances, a diagram must be constructed for each one. Each performance is defined as a kind of success, and the system is traced for each success step by step, with a recording at each step of the modes of element operation which will permit successful operation of the system up to that point. The diagrams themselves are a pictorial representation of a failure-mode-and-effect analysis. Each failure mode and each success mode of every element will appear in at least one of the diagrams, neatly filed under the "effect" (i.e., the system performance in which it participates) and accompanied by all the other element success and failure modes that are necessary for this effect to occur.

The same process works for subsystems. For each system mode, a diagram is constructed showing the operating modes of the subsystems relevant to that particular system mode. Those subsystem modes that appear on all the diagrams are the subsystem modes of interest in the analysis. Each of these subsystem modes will then be the subject of an individual subsystem diagram.



This process may seem cumbersome, but in practice it is usually quite simple. Many subsystems are restricted to very few modes -- often only two: there either is or is not an output. The whole process, furthermore, is one that must be performed in any analysis; and again CRAM, regarded as a bookkeeping procedure, ensures an orderly progression in this operating-mode analysis.

#### 4.1.4 Dividing the System into Subsystems

The procedure of dividing the system into subsystems has been mentioned in Section 3.1. It is a standard technique that must be performed in any system analysis, and a detailed discussion has no place in this monograph.

It must be realized, however, that to a very large extent the division of a system into subsystems is a matter of convenience. The reason is that it is easier to think about, and to analyze, perhaps a dozen small parts of a system than one large system. Again the subsystem-system relationship is one that can be reproduced on any level. In fact, very large systems are built in this way. They are divided into parts, which are separately contracted for; each part is again divided into parts; and this process continues through perhaps as many as 14 levels, until one finally reaches a contractor who has the responsibility of building a particular equipment out of piece parts. The analytic breakdown quite often parallels the breakdown used to manage the original hardware procurement, and, even when it does not, a hierarchical structure of models will probably prove convenient.

Restrictions that must be observed in the construction of the system model are briefly discussed below.

##### (1) Completeness of Model

At any structural level, a model for the equipment must contain all available information on how the equipment works and what is needed to energize it. This means that the model for an operating mode of an equipment must show explicitly:

- (a) The inputs needed by the equipment in this mode
- (b) The elements needed for the equipment to operate in this mode
- (c) The outputs produced when the equipment operates in this mode

## (2) Preservation of Independence

If two subsystems or elements are identified by different designators† and treated on separate diagrams, the CRAM analysis will assume that the failures or successes of the corresponding equipments are independent. If the two subsystems have a group of elements in common, however, this independence is violated, and hence we may not assign designators to the two subsystems as such. Instead, we must assign a designator to the group of common elements, and separate and distinct designators to the remainders of the two subsystems. The subsystems may still have names, but the diagram for each subsystem will now show at least two designators: one for the common elements, and one for the elements peculiar to the subsystems. On any subsystem diagrams containing both subsystems, we must again use the designators for the separate groups, rather than designators for the two subsystems, since otherwise we shall implicitly assume that the common group of elements in the actual equipment is duplicated, rather than shared.

These two points are the ones to which special attention must be paid in a CRAM analysis. For further discussion on the breakdown of systems, the reader is referred to the standard literature on systems analysis.

### 4.2 Naming of Blocks

A block on a CRAM diagram signifies the assumption that the element is operating in a mode enabling success for the path in which the block appears. To perform its function, the block must contain the following information:

- (a) A name for the element
- (b) An indication of the operating modes of the element which allow success for the path.

---

† The word "designator" in this section refers to the term used to represent an element on a block diagram; for example, if the element is a power supply, the designator might be "PWSUP".

This information must be presented in a code that is readily understandable to the analyst who prepares the diagram, to the analyst who checks the diagram (see PR 5, page 17), and to the EDP representative who must convert the diagram to a computer code. (see Sec. 5.1.3, page 79). Instructions for performing this coding operation are given in Sections 4.2.1 and 4.2.2.

#### 4.2.1 Element Names

Elements may be subsystems, sub-subsystems, black boxes, piece parts, or even parts of piece parts. The names of the piece parts, and perhaps of the black boxes, will be found on the original system description. It will be found advantageous to use, wherever possible, the same names on the diagram as on the original system description. When these are too long, a mnemonic contraction can be used.

As far as other elements are concerned, any suitable names can be employed. One technique which has been used in a large-scale analysis is to assign a series of numbers to each subsystem and to use these numbers to distinguish between different operating modes of the subsystem. These numbers are also used within each diagram as names for convenience blocks. (For a discussion of convenience blocks, see page 23; also, following paragraphs of this section.) Another method might be to assign two or three letters as a mnemonic code to each subsystem and use numbers thereafter to distinguish convenience blocks on different diagrams. The important requirements are that duplication of names be avoided and that the code assigned to subsystems and sub-subsystems be carefully recorded on the diagram on which these subsystems or sub-subsystems appear as elements. (See PR 3, page 12.)

In any computer run, all element names will be assumed to be of the same length. It is not necessary that the names appearing in the blocks be of the same length, but the length of the longest name should be noted by the analyst, since this will be the length to which all names will be expanded. Because a long name tends to slow down the computer program, and hence increases the cost, unnecessarily long names should be avoided.

The computer program recognizes two kinds of convenience blocks:

- (a) Convenience blocks which correspond to elements
- (b) Convenience blocks to which no elements correspond.

These will represent branches of the diagram.

If a block which represents an element is also to be used as a convenience block, a "+" sign is added to the element name the first time it appears, and a "\*" is added on all subsequent uses. The last block of a diagram is usually not coded.

Convenience blocks may be used as names for branches of the diagram. If no element terminates in a particular branch, then a convenience block of the second kind -- as described in (b) above -- must be inserted. The diagram must show that this is a block of the second kind, so that the computer program will omit this element name in the construction of the model. A convention -- for example, the letter C -- may be used for this purpose. The EDP group, in translating the diagram into a computer code, will make the appropriate interpretation. (See Table 2, page 76.)

The above discussion can be summed up in the following coding instructions:

- CI 1. For parts, the same name as appears on the original description (or a mnemonic therefor) should be used on the diagram wherever possible.
- CI 2. To avoid duplication of names between elements on different diagrams, a code may be used as a prefix to the name of a part.
- CI 3. Any rational system may be used to assign names to elements which are subsystems, sub-subsystems, or convenience blocks. These names should be carefully recorded on the diagrams on which such elements appear. Any continuing policy concerning such nomenclature should be documented.

CI 4. Convenience blocks are coded as follows:

- (a) For those blocks which contain element names, by adding a "+" to the element name on its first appearance and a "\*" on subsequent uses of the block.
- (b) For those convenience blocks which represent branches not terminating in elements, any convenient name may be used. The letter "C" must appear in the upper left-hand corner of the block.
- (c) The terminal block of each diagram will contain a name for the system mode diagramed. No special code need be added to this name.

#### 4.2.2 Coding Operating Modes

Element operating modes, too, must be entered into the blocks of a reliability diagram. For elements which are parts, this is best done by a mnemonic code, such as:

op for "open"  
sh for "short"  
le for "leak"  
un for "unstable"  
OK for "OK"  
 $\overline{\text{OK}}$  for "not OK"

Blocks of the type considered in CI 4(a), above, carry the operating code of the element named in the block.

The operating modes of systems or subsystems will be represented by the terminal blocks of different diagrams, and when they appear as elements in other diagrams, the indication "OK" or " $\overline{\text{OK}}$ " will suffice. Convenience blocks described in CI 4(c), above, can be given their operating mode designators in this manner.

Convenience blocks which do not correspond to elements or subsystems -- the type referred to in CI 4(b) -- should not carry operating modes, since their names will not appear in the model constructed from the diagram.

For reasons explained in the following paragraphs, it is important that all operating modes of all elements on a diagram be recorded, and that a list of these modes be attached to the diagram. A form for such a list, which must be completed for each diagram, appears as Figure 4. The last two modes must always be  $\overline{OK}$  and OK, in that order.

1. Diagram for \_\_\_\_\_
2. Diagram name<sup>†</sup> \_\_\_\_\_
3. Operating modes:

Mode Number	1	2	3	4
Mode Name				
Mode Designator <sup>††</sup>				
	5	6	7	8
	9	10	11	12

<sup>†</sup> This is the name in the last block of the diagram.

<sup>††</sup> These are the mnemonic codes used to indicate modes on the blocks.

FIGURE 4  
LIST OF OPERATING MODES

In the computer programs, each diagram block is identified, first, by a series of characters representing the element name, and then by a succession of zeros or 1's which indicate the operating modes required. This technique, illustrated in Figure 5,

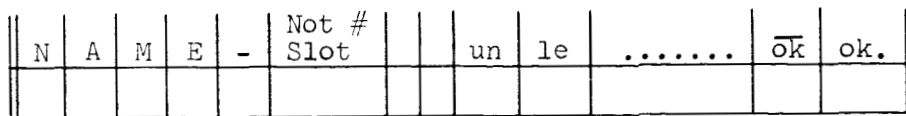


FIGURE 5

REPRESENTATION OF DIAGRAM BLOCK IN COMPUTER

reduces the requirements for computer storage memory but, because of the use of the "Not Slot" (as explained on page 32), results in a minor restriction on the ease of designating element operating modes. The remaining slots are called the "tag". Each operating mode corresponds to a position in this tag, and in the Part 3 program the appearance of a 1 in one of these positions will cause the program to look up the probability that the element is in this mode in a probability table. Only the position is used in this search; therefore, all the probabilities -- for example, that the different elements are "short" -- must be found in the same column of the table. EDP will convert the mode descriptions in the blocks into a tag; however, to do this properly, EDP must be told how many modes there are and what position to give them in the tags. This is the function of the form shown in Figure 4.

Operating modes can be indicated on the blocks in two ways: by inclusion or by exclusion. Inclusion is used when all the modes which allow success of the path are shown. For instance, if A is a diode which performs its function in a path provided it conducts current, this situation can be shown as

A/OK,sh

indicating that success can be obtained in that path with A either shorted or working perfectly.

If "open" and "short" are the only failure modes of interest for this diode, an equivalent statement would have been:

"A must not open."

When this verbalization is used, it can be shown on the block as an indication by exclusion. To indicate an excluded mode, one draws a bar over the name of the mode, thus;

$$A/\overline{op}$$

Either way can be used, entirely at the option of the analyst, but in one block only one method can be used. As an example,

$$A/\overline{op},sh$$

would not be acceptable. The "Not Slot" in Figure 5 is used to indicate the method used, the number 1 representing exclusion and a zero representing inclusion.

Another point of difference between exclusion and inclusion is the interpretation to be drawn when several mode designators follow the element name in the block. If two or more modes appear without bars (inclusion), this means that any one of these modes will allow success. If several mode names appear, each with a bar over it (exclusion), this is interpreted to mean that all the mentioned modes are excluded. These interpretations are consistent with the operations of the computer program.

The above discussion can be summed up in the following coding instructions.

- CI 5. For each diagram, a list of operating modes must be prepared. A duplicate should be made for use in preparation of the failure information table to be described in Section 4.3.2.
- CI 6. The operating modes which allow success to the path in which the block appears must be indicated in the block. Either inclusion or exclusion may be used. The mode indication may either appear under the element name or to the right of it with a slash mark (/) separating the name from the mode indication. The indicators for different modes will be separated by commas.

If inclusion is used, the indicators of all allowed modes will appear (without bars over them).



If exclusion is used, the indicators of all prohibited modes will appear, each with a bar over it.

- CI 7. Convenience blocks corresponding to elements will carry the mode indication of the element.

Convenience blocks not corresponding to elements will carry no mode indication; however, they will carry "C" in the upper left-hand corner.

#### 4.3 Preparing Listings

The block diagram indicates which combinations of element operating modes are required for system success. From it, the Part 1 and Part 2 computer programs construct a model which expresses system success as a function of success of different combinations of elements in different modes. This function can be interpreted as an event, the occurrences of the element modes as elementary events. Under this interpretation the probability of system success will be obtained by substituting in the output of the Part 2 program the probabilities that the elements are in the indicated modes. This is done in the Part 3 program.

A two-stage process is used to substitute probabilities for the element modes in the model. First, the elements are assigned to element classes; then, failure-probability information is supplied to each class. If two elements are assigned to the same class, they will be given the same failure probability; therefore, even if the elements are actually the same, they must be assigned to different classes if they differ either in application factors or in operating time.

For each of the two stages, a separate form must be completed by the analyst. These forms are described below.

#### 4.3.1 Element Table

The element table is shown in Figure 6. It is a simple listing of elements which are mentioned on the diagrams, and of the part class to which each element belongs.

As many as five symbols may be used to identify the part class. The assignment of names to the part classes is essentially arbitrary; however, the following four-part instruction will help to keep the process orderly.

- CI 9. (a) If identical parts differ in application factors for the same mode, use one symbol to code the application factors.
- (b) If identical parts are used for different periods of time, use one symbol to code the time periods.
- (c) Use the remaining symbols (3 or 4) for a mnemonic code of the part classes.
- (d) For elements which themselves have diagrams, use diagram names wherever possible.

Again, to avoid duplication of effort it is important that everyone concerned in the analysis use the same code for part classes. The next instruction is, therefore:

- CI 10. Use the same part-class code throughout the system analysis.

The completion of the element table now proceeds as follows:

- (1) The block
  - (a) "System/subsystem Description" is completed with the full English name of the system or subsystem.
  - (b) "Diagram Name" refers to the name in the last block of the diagram.
  - (c) Page number and number of pages in the table are entered in the upper right corner.
  - (d) The number of symbols used for the element names is entered in the "Length of Element Name" block.

System/Subsystem Description				
Diagram Name		No. of Elements on Table	Checked	(initials)
Length of Element Name		Part Classes Checked		(initials)

p— of —

Element	Class	Element	Class	Element	Class

FIGURE 6  
ELEMENT TABLE

AR-64-360

- (2) The first, third, and fifth columns are completed from the diagrams. The third and fifth columns are used as needed, and if additional space is needed a second sheet is used. The information described under (1) above, is first recorded on the second sheet. Convenience blocks which do not correspond to elements are not entered in the table.
- (3) The number of elements (other than convenience blocks not corresponding to elements) is then counted, both on the diagram and in the table. The numbers are compared and, if they agree, this number is entered into the block "No. of Elements on Table".
- (4) The part-class information is obtained from the system description, and the prearranged part-class code is entered into columns 2, 4, and 6, as needed.
- (5) On completion of the table the work is checked, preferably with the help of the second analyst who checked the diagram, and the check blocks are initialled.

#### 4.3.2 Failure Information Table

The failure information table is illustrated in Figure 7. It contains information to be used in computing, for each element class, the probability that it will operate in a particular mode at the end of a particular period of time.

This information can be obtained in two ways: (1) directly, perhaps from previous computations; or (2) in the form of failure rate, time, and application factors. The computer program reads a code symbol, and, in the first case, merely enters the probability in a table; in the second case, it enters in the table the value of

$$P = 1 - e^{-\lambda K_1 \dots K_n t} \quad (7)$$

where  $\lambda$  is the failure rate,  $K_1, \dots, K_n$  the application factors, and  $t$  the time.

Frequently, two diagrams will use the same element classes, except perhaps for subsystems supplying inputs to the systems. In this case it is advisable to use the same table for all the element

p. \_\_\_ of \_\_\_

System/Subsystem Description																
Use With Table No.	Diagram Name	No. of Operating Modes (Less OK)														
Checked for Completeness		Numbers Checked (initials)														
Operating Modes	Designation No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
Element Class	Mode No.	P/ Blank	Probability/ Rate	K <sub>1</sub>				K <sub>2</sub>				K <sub>3</sub> <sup>†</sup>				t

† Any number of K factors may be shown.

AR-64-375

FIGURE 7  
PART CLASS FAILURE INFORMATION TABLE

classes the diagrams have in common, and to make up a second sheet for the element classes peculiar to the particular diagram. The block in Figure 7 designated "Use with Table Number" is for identification of the table. The number of the diagram for which the original table was constructed is inserted there, if the sheet contains only those classes not found in that table.

The form in which the numbers are to be entered in the table is a so-called floating point form. Any number, P, can be expressed as a fraction 0.a times a power of ten

$$P = \pm 0. a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 \times 10^{\pm n_1 n_2}$$

where  $a_1 \neq 0$ . The number is then entered as

$$a_1 a_2 a_3 a_4 a_5 a_6 a_7 \overset{\pm}{a_8} n_1 \overset{\pm}{n_2}$$

As an example, 0.00397 would appear as 3970000002<sup>+-</sup>, and -397 would appear as 3970000003<sup>-+</sup>.

This form is used in tables which form an input to the computer program, because it uses exactly ten digits for each number. For ease of reading, the computer outputs use a slightly different form of the floating point notation. It is as follows:

$$(-) a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 E(-) n_1 n_2$$

Only minus signs are printed; therefore, the two numbers in the above example, if they were machine outputs, would appear as

$$.39700000 E-02 \quad \text{and} \quad -.39700000 E 03, \text{ respectively.}$$

At most, eight significant figures can be carried. The final zeros need not be entered in the table, but they will be punched.

Instructions for completing the table now follow:

- (1) (a) "System/Subsystem Description" is completed with the full English name of the system or subsystem.
- (b) "Use with Table No." is left blank unless this is a second or third sheet of a table previously made. In the latter case, the diagram name see (c), below of the previous table is entered here.
- (c) "Diagram Name" refers to the name in the last block of the diagram.

- (d) "p. of " is completed as required.
- (2) The operating-mode dictionary is completed from the duplicate copy of the list of operating modes. The order of the modes must be the same as on that list.
- (3) The columns are filled out as follows:
- (a) Element Class - the name of the element class as it appears on the element list (Figure 6)
  - (b) Mode No. - for each element class the numbers of the modes in which it can operate are listed in order, in columnar form.
  - (c) P/Blank - if the information is a failure probability, a "P" is entered in this column; if a rate, the column is left blank.
  - (d) Probability/Rate - the probability or rate is entered here, in floating point notation.
  - (e)  $K_1, \dots, K_3$  - application factors in floating point notation [needed if (d) is a rate]
  - (f) t - the time during which the part-class is required, in floating point notation [needed if (d) is a rate]
- (4) The completed form is checked against the element list for completeness, and against the original information for accuracy of the numbers entered. If at all possible, a second analyst is employed to help with the checking. The check blocks are initialed as the checks are made.

#### 4.4 Constructing Reliability Diagrams

##### 4.4.1 General Discussion and Terminology

As indicated in Section 4.1.3, the rationale underlying the CRAM rules for constructing diagrams is that, when CRAM applies, a step-by-step system analysis becomes possible, and that such a step-by-step procedure is an efficient format for the systems analysis.

The analysis starts with the final outputs of the system, which are required for "system success" as defined for the system operating mode being diagramed. For uniformity, these outputs are connected by lines to a terminal convenience block, which contains the name

of the system being diagrammed. How these connecting lines are drawn will be discussed presently.

As the analyst works from the final system outputs back toward the input, the same two questions recur concerning each element (except input blocks, for which only the first question applies):

- (a) What operating modes of this element allow it to perform its function?
- (b) What combination of inputs is required to allow this element, in these modes, to perform its function?

Two or more operating modes of an element which require the same inputs may be combined in one block, in the manner discussed in Section 4.2.2. If different element operating modes require different combinations of inputs, they must be represented by separate blocks on the diagram. An example of these two possibilities is shown in Figure 8.

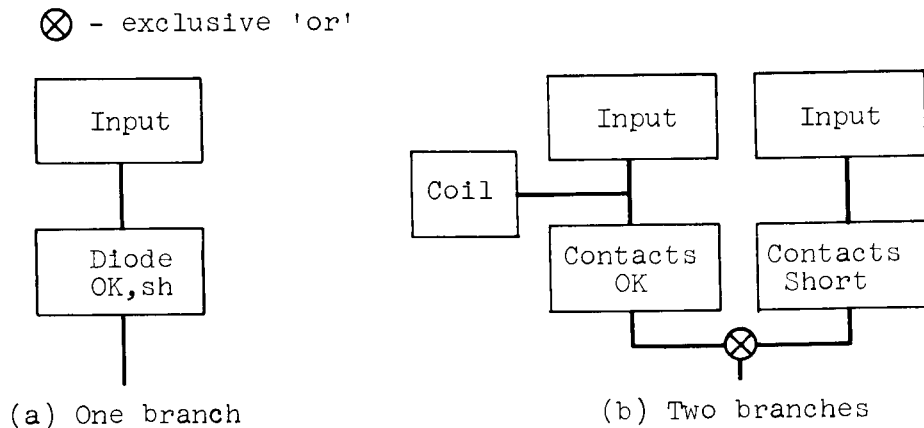


FIGURE 8

BRANCHING FOR OPERATING MODES OF AN ELEMENT

Part (a) of Figure 8 represents a diode whose function is to pass a current. It performs that function both when it works perfectly and when it is shorted. Part (b) of Figure 8 represents a pair of normally open relay contacts which also have the function of transmitting a current. If the contacts have failed short, this



function will be performed without further conditions; if the contacts have not failed, they can be activated only if the relay coil functions. The diagram therefore must show two branches -- one for each operating mode of the contacts which allows success.

The symbol  $\otimes$  in Figure 8(b) is one of the symbols used to represent types of connections between blocks. A hypothetical example will be used to explain this symbolism.

Suppose three elements, a, b, and c, provide inputs to a fourth element, d, and suppose, for simplicity, that each of the elements has only one correct operating mode. On the reliability diagram, A indicates that a operates correctly. D\* indicates that d has not failed and that enough inputs are available to allow d to function. Various input requirements are possible:

(a) In a series system, a, b, and c are all required for d to operate. This situation can be expressed in a formula as

$$D^* = D \text{ and } (\underline{A} \text{ and } \underline{B} \text{ and } \underline{C}) \quad (8)$$

(b) Another possibility is that a and b are in parallel, and c is always required. This may be expressed as

$$D^* = D \text{ and } [(\underline{A} \text{ or } \underline{B}) \text{ and } \underline{C}] \quad (9)$$

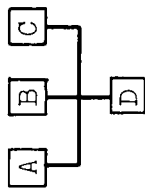
(c) A third possibility is that c is used as an input, but if c fails the system is manually switched to use both a and b. To express this we use  $\bar{C}$  to indicate that c has failed, and the formula becomes:

$$D^* = D \text{ and } [\underline{C} \text{ or } \bar{C} \text{ and } (\underline{A} \text{ and } \underline{B})] \quad (10)$$

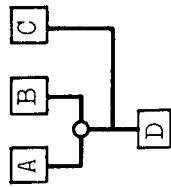
(d) A fourth possibility is that any two of the three inputs suffice. The mathematical expression is:

$$D^* = D \text{ and } [(\underline{A} \text{ and } \underline{B}) \text{ or } (\underline{A} \text{ and } \underline{C}) \text{ or } (\underline{B} \text{ and } \underline{C})] \quad (11)$$

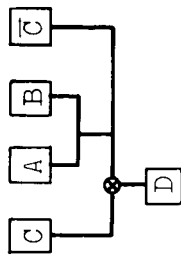
Symbols are used on the diagram to indicate which logical combination is needed. An illustration is given in Figure 9, where parts (a), (b), (c), (d), and (e) refer, respectively, to the cases expressed by Equations 8 through 12. The symbols used are:



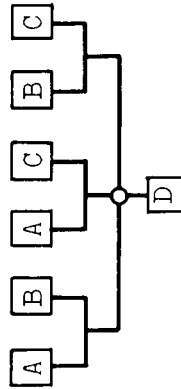
(a)  $D^* = D$  and [A and B and C]



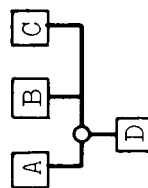
(b)  $D^* = D$  and [(A or B) and C]






(c)  $D^* = D$  and [C or  $\bar{C}$  and (A and B)]



(d)  $D^* = D$  and [(A and B) or (A and C) or (B and C)]



(e)  $D^* = D$  and [A or (B and C)]

Legend:  
 and  
 or (inclusive)  
 or (exclusive)

AR-64-368

FIGURE 9  
 LOGICAL CONNECTIONS ON DIAGRAMS

- (a) a straight-line connection to represent "and"
- (b) a circle to represent the inclusive "or", i.e., either branch leading into the circle, or both of the branches together, will meet the requirement
- (c) a circle enclosing an x to represent the exclusive "or", i.e., either branch will allow success, but the two branches cannot occur together

Care must be used in the placement of the symbols. For instance, in Figure 9(b) the branch from C joins the branch from (A or B) below the circle. This relationship is represented in Equation 9. If the branch from C were brought in above the junction of the branches, through A and through B, the result would be Figure 9(e), the formula for which is:

$$D^* = D \text{ and } [A \text{ or } (B \text{ and } C)] \quad (12)$$

This equation is Equation 9 with the roles of A and C interchanged.

The form of notation described above may at first seem unfamiliar but its utility will soon become apparent. Its main advantage is that the diagram now graphically displays the conditions on inputs to the elements; no further explanation or footnotes are needed.

For diagrams of series-parallel systems, notation is all that is needed. The rules for constructing these diagrams follow in Section 4.4.2. However, to convert diagrams of feedback systems into tree diagrams, some additional definitions and terminology are needed. These will be given, together with the rules for making the conversion, in Section 4.4.3.

#### 4.4.2 Rules for Constructing Reliability Diagrams

The rules in this section will produce a reliability diagram in standard form for any series-parallel system. For a feedback system, they will produce a diagram that displays this feedback, which can then be reduced to standard form by Rule 4 in Section 4.4.3.

It is assumed that a system description is available that will allow the analyst to answer the following questions:

- (1) Which system outputs are needed for success of the system, and which inputs to the system are not part of the system?
- (2) For each element, which operating modes of the element allow it to perform its function; and, for each operating mode, which logical combination of inputs is needed to allow this element to perform its function while operating in the particular mode indicated? (The inputs to an element are either outputs of its immediate predecessors or system inputs.)

The rules are as follows:

- Rule 1. Draw a single terminal block to represent system success. Its predecessors are the elements producing the system outputs needed, as specified in question 1, above.
- Rule 2. Starting with the terminal block, and continuing until all the system inputs are reached, perform the following steps:
- Step 1. Above each block, draw blocks for the immediate predecessor elements. If different operating modes of these elements lead to different success paths, draw separate blocks for these modes. If these modes do not lead to different paths, combine them in one block.
- Step 2. Indicate graphically what logical combination of inputs is required for the functioning of each set of element modes represented by a block. Use the following symbols
- ⊗ for exclusive 'or'
  - for inclusive 'or'
  - straight lines for 'and'
- For a series-parallel system, this process will terminate when all the system inputs are reached. For a feedback system<sup>†</sup>, the following procedure is needed.

---

<sup>†</sup> For definition, see Section 4.4.3.

Step 3. Draw the return paths of feedback loops in such a manner that each loop is shown only once. Use the logical notation of Step 2 to show the connections.

The use of convenience blocks is covered in Rule 3:

Rule 3. If a branch is to be repeated in the diagram, one can

- (a) insert a "+" in the last block of the branch, or, if there is no block,
- (b) insert a block into the branch and attach a "+" to the name of this block, if the branch does not terminate in a block;

then, instead of repeating the whole branch, repeat only its terminal block with a "\*" attached to the name of that block.

Caution

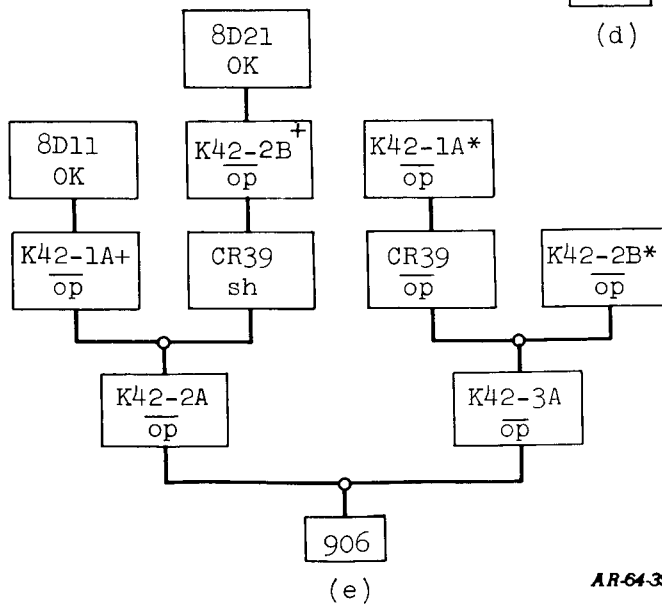
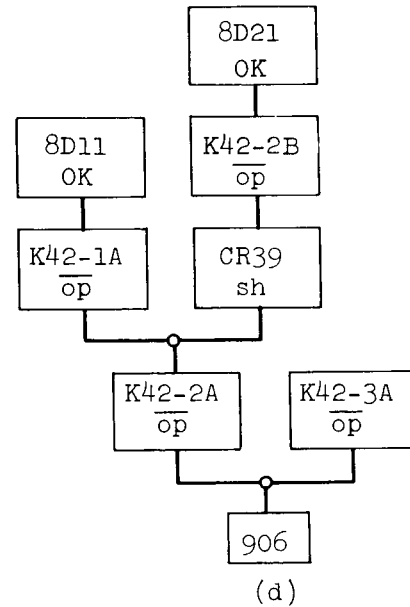
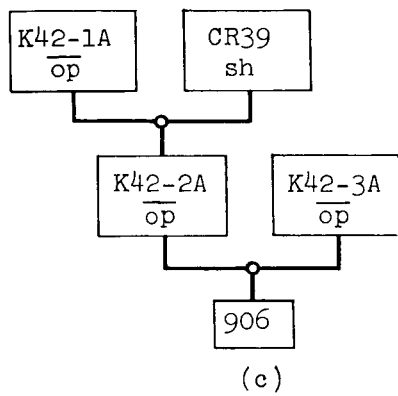
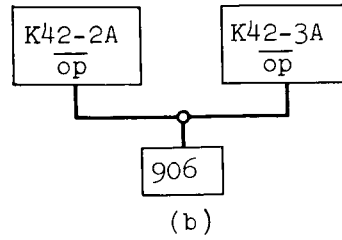
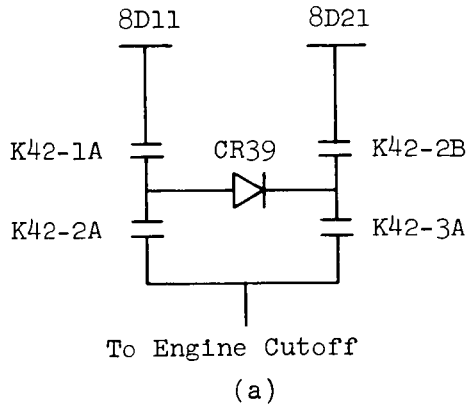
Starred blocks must be drawn as input-blocks.

The simple example given below illustrates the use of Rules 1, 2, and 3. Further examples are given in Section 4.6.

EXAMPLE

For this illustration, part of the circuit of Figure 15 is employed as Figure 10(a). The relay coils of contacts K42-1A, K42-2A, K42-2B, and K42-3A are assumed perfect, and are not shown. A signal must pass from one of the positive buses 8D11 or 8D21 to the line marked "To Engine Cutoff". The procedure is as follows:

- (1) By Rule 1 a block, marked "906", is drawn to represent system success.
- (2) A signal can reach 906 either from contact K42-2A or from contact K42-3A. Draw these blocks. To permit the signal to pass, it suffices that the contacts not be failed in the open position. Thus the mode indication is " $\overline{op}$ " in both blocks. This is Step 1 of Rule 2.
- (3) Draw the connections between the blocks. (Step 2, Rule 2) Either contact can pass the signal; therefore, the  $\bigcirc$  must be used. At this point we have Figure 10(b). Next, we examine how the signal can reach contact K42-2A. One path



AR-64-357

FIGURE 10  
CONSTRUCTION OF A RELIABILITY DIAGRAM

is from contact K42-1A. If the diode has shorted, there is a path through the diode. K42-1A must not have failed open.

- (4) Draw the predecessor elements to K42-1A (Rule 2, Step 1).
- (5) Draw the connections (Rule 2, Step 2). Since either path will pass a signal, the  $\bigcirc$  must again be used. This process results in Figure 10(c).

We observe that the signal passes to contact K42-1A from bus 8D11, which must be OK, and to diode CR39 from contact K42-2B, which must not have failed open. Contact K42-2B gets its current from bus 8D21.

- (6) All the connections mentioned above are in series. The result of drawing these blocks and their connections is Figure 10(d). Examination of the other branch shows that contact K42-3A gets its current either from diode CR39 or from contact K42-2B. The signal passes through either path, provided the element has not failed open.
- (7) Draw the predecessors of K42-3A and the connections. Again, since either path suffices, the  $\bigcirc$  must be used.

Finally, we observe that CR39 receives current from K42-1A, which was previously drawn. We place a "+" in this original block and use the block  $K42-1A/\overline{op}$ , with a "\*", as a predecessor for CR39 in the right-hand branch. Similarly, the branch ending in  $K42-2B/\overline{op}$  drawn before; hence we insert a "+" in the block on the left, and a "\*" in the similarly named block on the right. This is the manner of using Rule 3. The final diagram is shown in Figure 10(e).

#### 4.4.3 Feedback Systems

##### 4.4.3.1 Introduction

If the system that is being analyzed is a true feedback system, the resulting diagram will display the feedback. It is quite possible to write the reliability expression for feedback loops directly; however, an easier way is to work from a sequential display, i.e., one in which each element has a single successor and a unique set of immediate predecessors.

To obtain this simplification in the course of preparing the reliability expression, and also to display in sequential form the consequences of particular malfunctions, the feedback-type diagram must be converted to a sequential one. The conversion is achieved by Rule 4. Before the rule is stated, however, it is necessary to develop and define terminology that will give further insight into the reliability structure of feedback systems.

#### 4.4.3.2 Definitions

As indicated previously, the analysis requires that, for each element, the immediate predecessor elements be known. One element is a predecessor of a second if it is either (1) an immediate predecessor or (2) an immediate predecessor of an element that is a predecessor of the second element.

##### Definition 1

Feedback occurs when, in a chain of elements, say  $A_0, A_1, \dots, A_n, A_0$ , each element is its own predecessor.

##### Definition 2

A cycle is any group of elements, each of which is its own predecessor. A maximal cycle is a cycle to which no other element can be added without destroying the cyclic property. Since we shall remove maximal cycles only, we omit the word "maximal" when no confusion can arise.

##### Definition 3

A circuit is a cycle (not necessarily maximal) in which only one element is repeated. A circuit is the simplest kind of cycle. It consists of a group of elements in series, plus a lead from the last element back to the first. Two circuits are distinct if at least one of them contains an element that is not in the other.



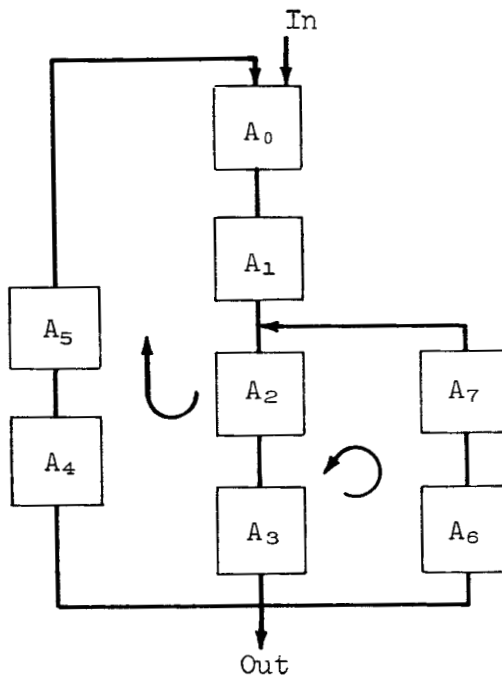
Definition 4

An entrance to a cycle is an element of the cycle that has an immediate predecessor outside the cycle. An exit is an element that has an immediate successor outside the cycle.

Definition 5

A path is a chain of elements, each of which is an immediate successor of the preceding one, and in which no element is repeated.

These definitions are illustrated in Figure 11.



$(A_0, A_1, A_2, A_3, A_4, A_5, A_0)$   
and  $(A_2, A_3, A_6, A_7, A_2)$  are  
distinct circuits. The whole  
group is a (maximal) cycle.  
 $A_0$  is an entrance.  
 $A_3$  is an exit.  
 $(A_0, A_1, A_2, A_3)$  is a path.

FIGURE 11  
A FEEDBACK SYSTEM

4.4.3.3 The "Cutting" Operation

The purpose of the rules is to convert a diagram like Figure 11 into a tree diagram, in which there is no feedback but which will have the same reliability expression for the exits. To do this, we

perform an operation on diagrams called "cutting". The result of cutting the diagram of Figure 11 is displayed in Figure 12.

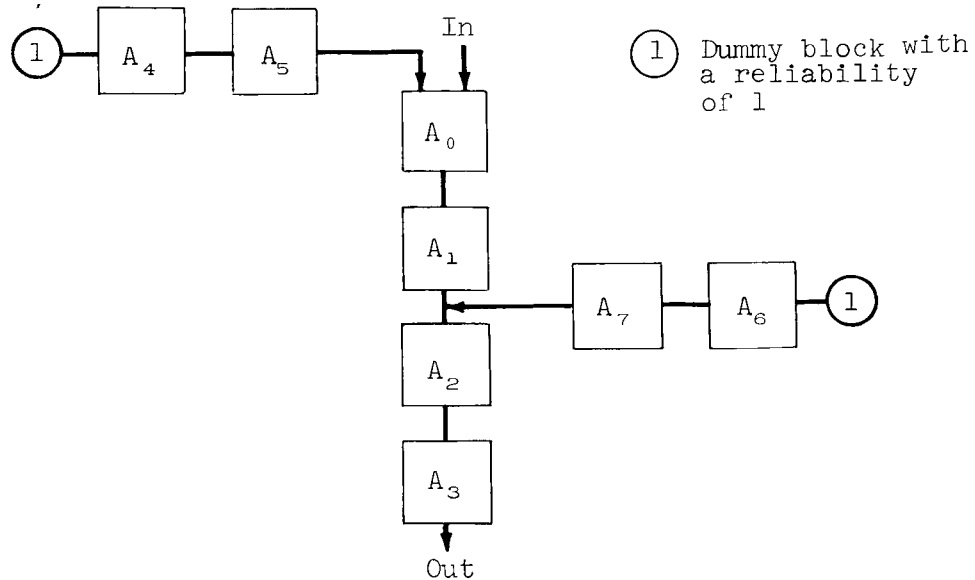


FIGURE 12

TREE DIAGRAM OF FIGURE 11

Only circuits are "cut". The procedure is evident from a comparison of Figures 11 and 12. It is stated in Definition 6.

Definition 6

Cutting a circuit just before a circuit element A consists of: writing the circuit elements in sequence, starting with A, and replacing the circuit predecessor of A (there is exactly one of these) by the symbol (1), which represents a dummy element with a reliability of 1. All the logical symbols are, of course, preserved in this operation.

The need for the (1) arises because of the need to preserve the logic symbols. (The alternative to use of the (1) would be to keep track of the logic symbols; thus the method described here is easier to explain, easier to program, and easier to prove correct.) If the circuit ( $A_0, A_1, A_2, A_3, A_4, A_5, A_0$ ) of Figure 11 had another input at  $A_4$ , the circuit would appear as in part (a) of Figure 13.

The result of cutting the one circuit is shown in part (b) of Figure 13, where the "and" symbol preceding  $A_4$  has been preserved.

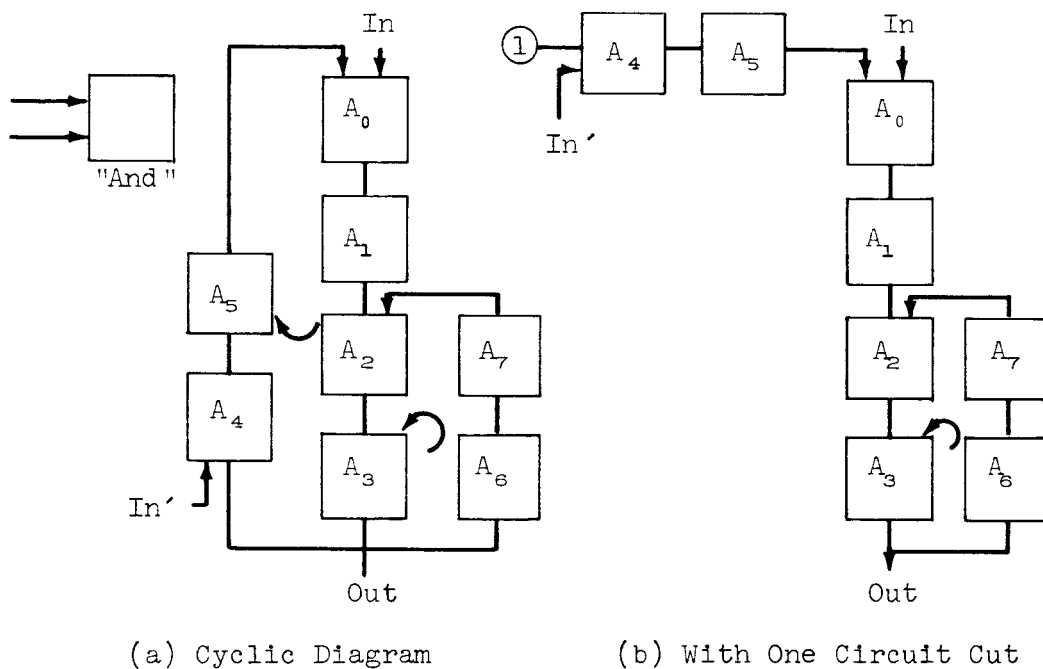


FIGURE 13

A CYCLE WITH TWO INPUTS

A significant point is that the diagram obtained by cutting preserves the reliability expression of the last element preceding the cut, but not necessarily of any other. The tree diagram obtained by repeated cuts will yield the proper reliability expressions for the exits, but not necessarily for any other element in the diagram. For this reason, we shall always cut just after an exit, either to

the cycle or to the circuit we are cutting. With this preparation, we can now state the rules for removing maximal cycles from cyclic diagrams.

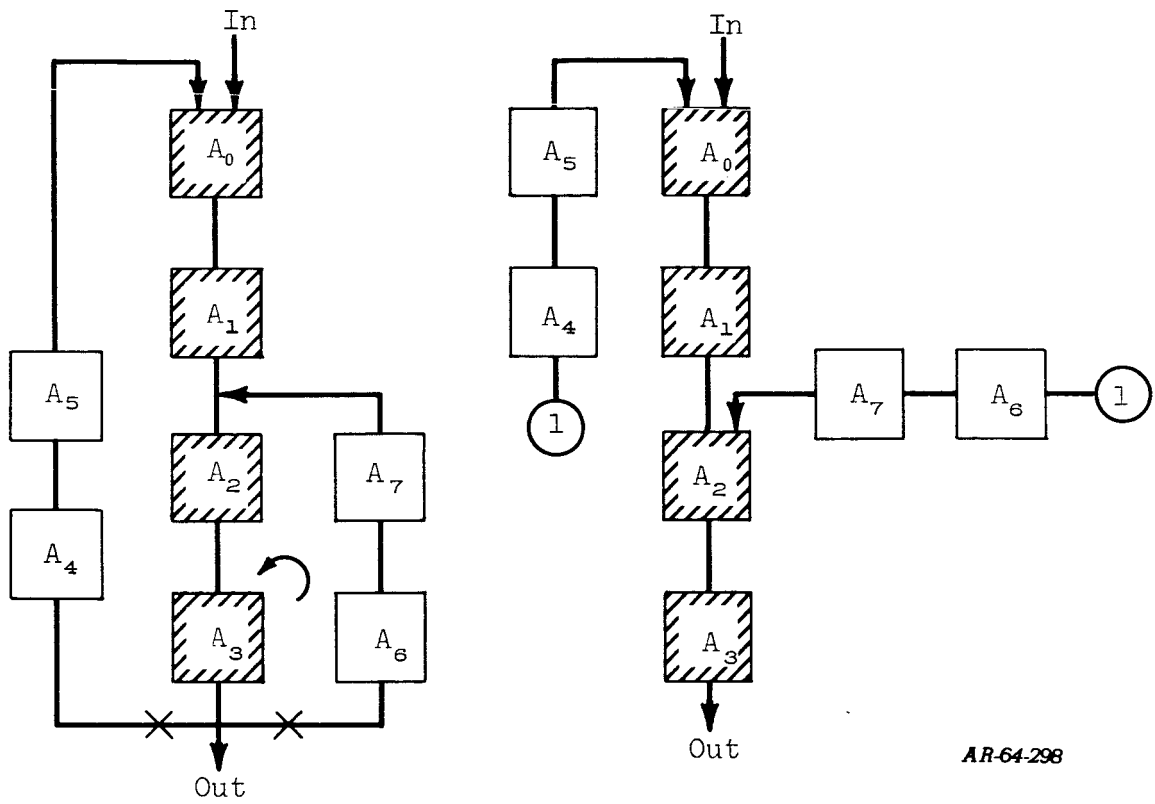
#### Rule 4.

- Step 1. In a maximal cycle, choose an entrance, an exit, and a path from the entrance to the exit.
- Step 2. Cut all circuits that intersect (have elements in common with) the chosen path just after the last element they share with the path.
- Step 3. If any circuits now appear as predecessors to different branches of the diagram obtained by Steps 1 and 2 (which is now acyclic), draw them in separately as predecessors each time they appear.
- Step 4. Repeat Steps 1, 2, and 3 for the remaining (maximal) cycles.

Rule 4 is illustrated in Figure 14. Part (a) of this figure shows a fairly simple feedback diagram for which only the first two steps of Rule 4 are needed.  $A_0$  is the entrance to the cycle, and  $A_3$  is the exit. The circuits are  $(A_0, A_1, A_2, A_3, A_4, A_5, A_0)$  and  $(A_2, A_3, A_6, A_7, A_2)$ . First a path is chosen from the entrance ( $A_0$ ) to the exit ( $A_3$ );  $A_0, A_1, A_2, A_3$  is a rather obvious choice. In Figure 14(a) this path is shaded, and the cuts are indicated by x's. The cuts are made just after the place where the cycles leave the chosen path; in this case, between  $A_3$  and  $A_4$  for the first cycle and between  $A_3$  and  $A_6$  for the second cycle. Figure 14(b) shows the results of making the cuts at the required places, and, since this diagram is free of cycles, it is the required tree diagram in standard form.

#### 4.5 Failure-Mode-and-Effect Analysis

A failure-mode-and-effect analysis (FMEA) is a listing which displays for each element failure the effect on the system if the element fails in the indicated mode. Usually the probability of this occurrence is given as well.



AR-64-298

(a) Cyclic Diagram  
 ▨ = chosen path  
 X = prospective cut

(b) After Step 2  
 (Transformation Completed)

FIGURE 14  
 ILLUSTRATIONS OF RULE 4, STEPS 1 AND 2

The conventional FMEA (called "first order") is used to draw attention to the single-element failures which can cause system failure. For a highly redundant system, however, it should provide information, for example, on each pair of failures which can jointly fail the system. This type is called a second-order FMEA, and similar distinctions are made depending on the extent of the information provided.

A secondary emphasis in an FMEA is on part characteristics. One question that it can answer is: "How many system failures can occur due to diode shorts?" The FMEA can then be used to obtain an estimate of the increase in system reliability which would result from a possible increase in part reliability in a particular mode. In addition, the FMEA supplies information about the relative seriousness of different failure modes of one part class.

CRAM diagrams contain all the information needed to construct an FMEA. As an illustration assume that all the mode information in each block is given by exclusion, e.g., as " $\overline{op}$ ", or " $\overline{le}$ " (for "leaks"). Then each failure mode mentioned in each block will disable the branch in which this block appears. If the identical element reappears as different blocks in several branches, all these branches will be simultaneously disabled. The failure mode will be a system failure if no success branch remains.

A first-order FMEA can be obtained from a CRAM diagram in two ways:

- (1) By a search of the diagram for all the series elements. (The analyst must remember that an element may appear as a block in several different branches.)
- (2) By a small computer program<sup>†</sup> that operates on the output of the Part 1 computer program (the formula). This program essentially "fails" each element in turn in each excluded mode, observes whether or not the system fails, records the findings, and, if the system fails, prints the probability that the failure will occur.

---

<sup>†</sup> This program is not at present being developed, since the n<sup>th</sup>-order program referred to below will perform the same function.

For construction of an FMEA which provides more than a first-order analysis, a computer is a virtual necessity because of the large number of pairs or triplets of failures that can occur in a system of any complexity. A computer program that will produce an nth-order FMEA from the output of the Part 1 CRAM computer program is currently being developed. However, because of the great number of possible combinations of failures, a second-order FMEA will probably be the limit of the practicable. The computer program will serve primarily to relieve the engineering analyst of the task of rearranging manually information which is already implicit in the diagram.

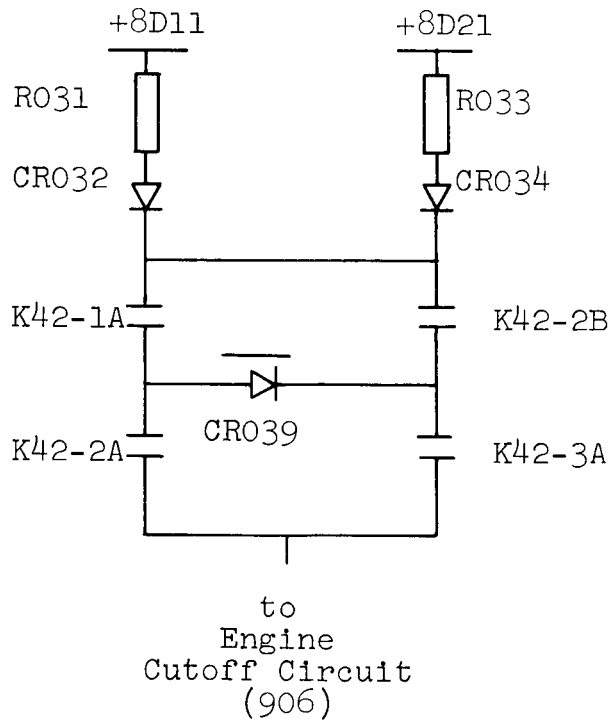
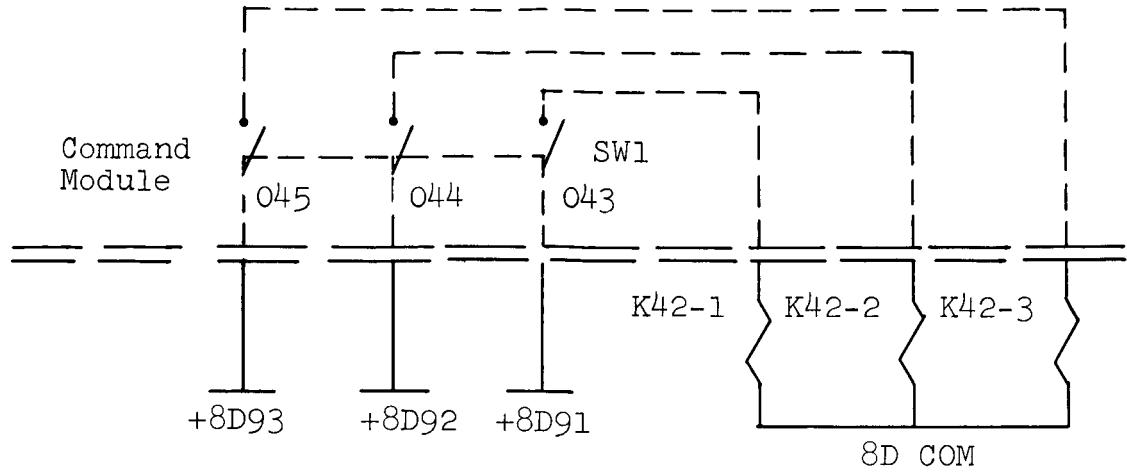
#### 4.6 Examples of the Application of CRAM

This section will give detailed examples of the application of the CRAM method to the point where the analyst has completed the "package" of information needed for the computer.

##### 4.6.1 A Series-Parallel System

A typical redundant circuit, which has the function of manually generating an engine cutoff signal, is used for this example. The circuit, which is triggered by a redundant switch, has as components: relays K42-1, K42-2, and K42-3, several power buses, two resistors, and two diode rectifiers. Figure 15 is a schematic diagram of the circuit.

When the switch is closed, as shown in the schematic, the coils of relays K42-1, K42-2, and K42-3 are energized from the +8D91, +8D92, and +8D93 buses, respectively. The normally open contacts of these relays are closed, and a voltage from the +8D11 and +8D21 buses can flow to the vehicle engine-cutoff circuit. When the switch is opened, the coils are de-energized, and the contacts open so that no current can flow to the vehicle engine-cutoff circuit.



AR-64-356

FIGURE 15  
SCHEMATIC DIAGRAM OF AN ENGINE CUTOFF CIRCUIT



The contacts of the relays are arranged in a two-out-of-three voting circuit. If one pole of the switch is open, one coil will not be energized. If relay K42-1 is not energized, current will flow through the contacts of K42-2B and K42-3A on the right side and complete the circuit. If K42-2 is not energized, current will flow through the K42-1A contacts, the diode, and the K42-3A contacts, completing the circuit. If K42-3 is not energized, current will flow through the K42-1A and K42-2A contacts on the left side, completing the circuit. Similarly, if any one coil is energized falsely -- by one pole of the switch closing, or by a short in the circuit that energizes the coils -- current will not flow through the circuit. Therefore, only two of the three relays must operate correctly to permit proper operation of the circuit, assuming that diode CR039 has not failed. If any one contact fails, with no other failure occurring, the circuit will not fail.

Two buses, +8D11 and +8D21, supply a positive voltage to the contact circuit through isolating resistors and diodes. If one bus fails, the other bus will supply the required voltage.

The reliability diagram for this circuit is shown in Figure 16, which represents the conduction of current to the vehicle engine-cutoff circuit when the switch is closed. This condition may be designated the "correct signal" case -- i.e., a signal is produced in the engine cutoff circuit when necessary (when the switch is closed). Hence, Figure 16 represents the reliability diagram for the "correct signal" case, i.e., when switches 043, 044, and 045 are intentionally closed to energize relay coils K42-1, K42-2, and K42-3. The output of this case is a signal (designated 906), which is generated at the bottom. As outlined below, this type of diagram can be easily constructed from the output backward.

(1) The output has been given the name 906, which appears at the bottom of the diagram.

(2) A signal can be produced at 906 through either of the two contacts, K42-2A or K42-3A. This dual possibility is indicated by the small circle joining the two branches leading into 906. To obtain the signal, the contacts must not have failed open. However, there are two possibilities for contact K42-2A: If the contact has failed short, current can flow from the bus 8D11, provided

Operating Modes	No.	1	2
	Desig.	op	sh

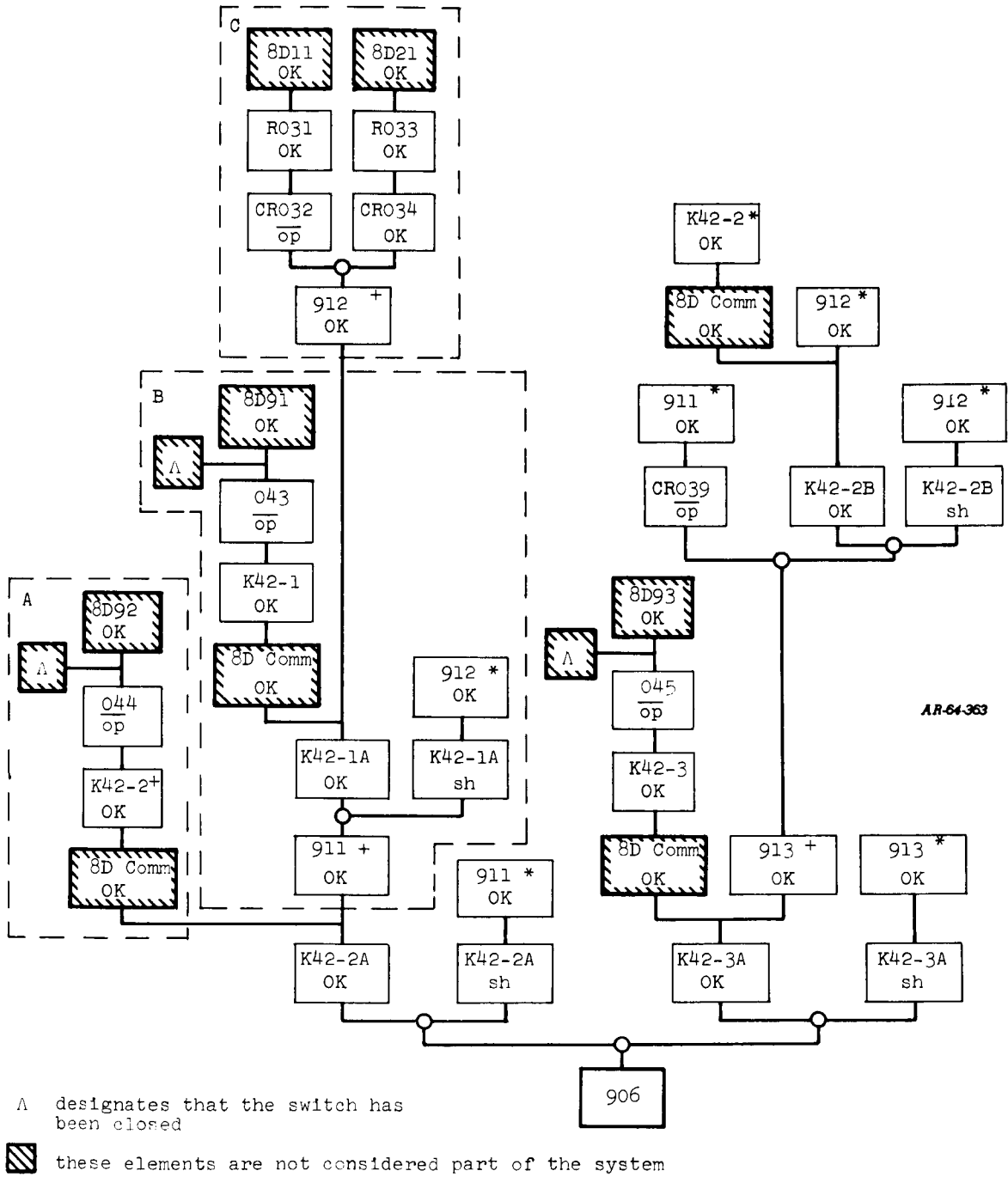


FIGURE 16  
RELIABILITY DIAGRAM OF AN ENGINE CUTOFF CIRCUIT

the other elements work; if the contact has not failed short, coil K42-2 must operate to close the contact, and the branch from 8D92 to K42-2 is still required.

(3) Coil K42-2 operates if the coil is neither open nor short, if the switch does not fail open, and if the bus 8D92 provides the necessary voltage. This branch is drawn on the left of the diagram. The coil K42-2 will be required again -- to work the other contact; hence, we put a "+" in the block and later will merely repeat the block name, adding a "\*" (Rule R3). This branch is indicated by a dashed box marked "A" (Figure 16).

(4) The current through contact K42-2A arrives from contact K42-1A. The situation described in (2) and (3), above, applies to this contact, also, producing the part of Figure 16 in the dashed box marked "B". (Note: Since this part of the diagram represents only the current flowing into K42-2A, it applies equally to the block K42-2A/sh. We therefore give it a name as a convenience block -- 911 -- and repeat this convenience block above the K42-2A/sh block.)

(5) The current flows into the contact K42-1A from diode CRO32 or diode CRO34, and we draw this part of the diagram, with the buses above the block K42-1A/OK. The K42-1A/sh branch receives current the same way; therefore, if we represent this part of the diagram by a convenience block (912), we need only repeat the one block above the K42-1A/sh box. This part is marked "C" in Figure 16.

Thus far we have drawn the ways in which a signal can arrive at 906 through contact K42-2A.

Current can reach contact K42-3A in two ways: one is analogous to the path already drawn, through contact K42-2B; the other is through the diode CRO39 and the contact K42-1A. We shall draw this latter branch first.

(6) Current flows through contact K42-3A either if the contact has failed short or if it is OK, and the coil K42-3 operates the contact. Since current flows into these two branches in the same way, we shall represent the flow by a convenience block (913).

(7) Current arrives at 913 in one of two ways: from contact K42-2B or from diode CR039. The latter branch obtains current from contact K42-1A, which is represented by convenience block 911 on the left side of the diagram.

(8) Current can flow through contact K42-2B in two ways: if it is shorted, from either of the diodes CR032 or CR034 (already designated 912); or from the same diodes if K42-2B is OK and if the coil operates the contacts.

Step 8, above, concludes the derivation of the diagram. The buses and the action of throwing the switch are not considered part of the system, and are therefore crosshatched on the diagram. These elements will not appear in the formula, nor in the element table. The element table and part class failure information table appear as Figures 17(a) and 17(b). The list of operating modes is shown in abbreviated form on the inset in Figure 16.

The resistor is not expected to fail short; it is therefore given a failure probability of zero in the short mode. For the switches no failure-rate information was available, but a failure probability was estimated from tests, and this appears in the table. These three lines have a P in the third column, to indicate that the numbers in the next column are probabilities and not rates.

With the completion and checking of these two tables, the package which the analyst is responsible for is now complete. Its further history appears in Section 5, where the punching of data from this package is discussed.

#### 4.6.2 A Feedback Circuit

Figure 18(a) represents part of a National Bureau of Standards design for a flip-flop circuit which has two stable states. In one of these states transistor Q001 is conducting, causing its collector voltage (at the point marked "O") to be very close to 0 (ground), since the voltage drop across the transistor is very small. The collector voltage of Q001 is applied through resistor R006 to the base of transistor Q002, preventing it from conducting. Since Q002 is turned off, its collector voltage will approach the -12 v from

(a) Element Table

System/Subsystem Description		Engine Cut-off Circuit		
Diagram Name	906	No. of Elements on Table	15	Checked (initials)
Length of Element Name	5	Part Classes Checked		(initials)

p 1 of 1

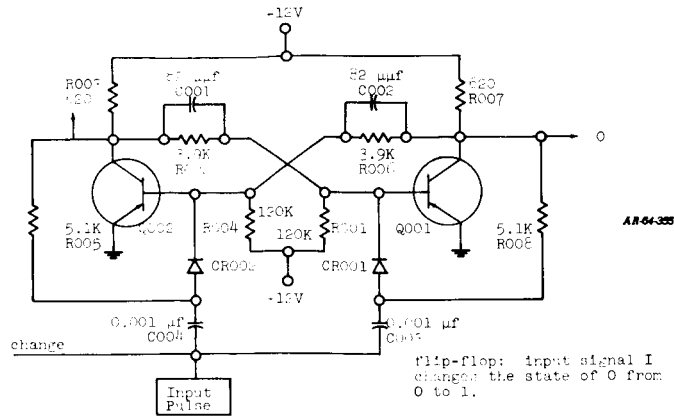
Element	Class	System/Subsystem Description Engine Cut-off Circuit		No. of Operating Modes 2 (Less OK)	Numbers Checked	t		
		Use With Table No.	Diagram Name 906					
Checked for Completeness		Operating Modes		P/ Blank	Probability/ Rate	K <sub>1</sub>	K <sub>2</sub>	t
Designation No.	op	sh	2					
K42-2A	CONG					2000000001	3210000003	4700000001
K42-3A	CONG					2000000001	1930000003	4700000001
K42-1A	CONG					2000000001	3450000003	4700000001
K42-2	COIG					2000000001	3450000003	4700000001
044	SWI							4700000001
K42-1	COIG					2000000001		4700000001
043	SWI							4700000001
CR032	DIOGP					1500000000	5350000003	4700000001
RO31	RFFHS					1500000000	1930000003	4700000001
CR034	DIOGP					1900000001	6420000003	4700000001
RO33	RFFHS							4700000001
K42-3	COIG					3200000007		4700000001
045	SWI					1470000007		4700000001
CR039	DIOGP					2280000008		4700000001
K42-2B	CONG					1120000007		4700000001
						9500000008		4700000001
						3600000007		4700000001
						3200000007		4700000001
						3156000006		4700000001
						1890000006		4700000001
						0000000000		4700000001

(b) Part Class Failure Information Table

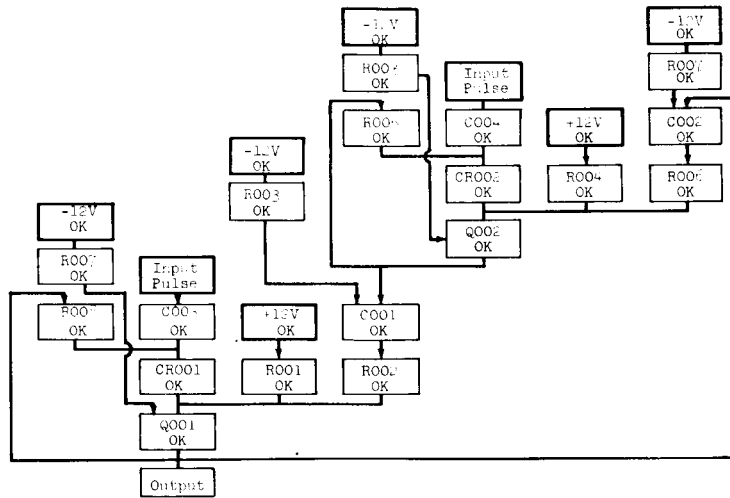
AR-64-365

FIGURE 17

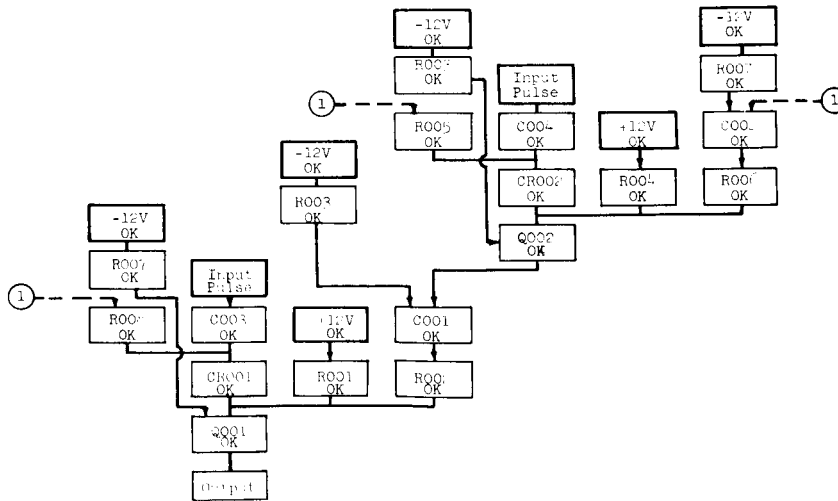
TABLES FOR ENGINE CUTOFF CIRCUIT



(a) Schematic



(b) Cyclic Reliability Diagram



(c) Cyclic Reliability Diagram

FIGURE 1  
FEEDBACK CIRCUIT (FLIP-FLOP)

the power supply which is applied through resistor R003. This collector voltage is applied through R002 to the base of Q002 to keep it in conduction.

The other stable state occurs when Q002 is conducting and Q001 is turned off.

Changing the flip-flop from one stable state to the other is accomplished by turning off the transistor which is conducting and allowing the other transistor to conduct.

The input pulse changes the state of the flip-flop in the following way:

If Q001 is conducting, the voltage at its collector will be almost 0 v, while the voltage at the collector of Q002 will be almost -12 v. There will be a small negative potential at the base of Q001, and a small positive potential at the base of Q002.

The resistors R003 and R005 apply the transistor collector voltages to capacitors C003 and C004, and the normal level of the input signal is -12 v, so the voltage across C003 is about 12 v, and the voltage across C004 is about 0 v. Capacitor C003 will therefore be charged and capacitor C004 will not.

As the input signal is changed to 0 v, C003 is discharged. The new voltage across C003 is about 0 v, which will not allow it to retain its charge. The voltage at the junction between C003 and C001 becomes positive, since it is about 12 v higher than the input signal. Diode C001 conducts, causing the base of Q001 to become momentarily positive, cutting it off. The collector voltage of Q001 will now turn Q002 on, allowing the collector voltage of Q002 to hold Q001 off. Diodes C001 and C002 prevent any reaction of the flip-flop when the input signal returns to -12 v.

Capacitors C001 and C002 act to decrease the switching time of the transistors and allow higher-frequency operation of the flip-flop.

Resistors R001 and R004 supply a positive voltage from the +12 v power supply to make sure that the transistors will turn off at the proper times.

After the flip-flop has switched, a second input pulse (from -12 v to 0 v) will change the flip flop back to its original state.

To draw the reliability diagram of the circuit, it is first necessary to specify the output we want to consider. This is chosen to be the output of transistor Q001, which is marked "0" on the schematic.

In accordance with the rules given in Section 4.4.2, the predecessor of the output, Q001, is added to the reliability diagram. This transistor must function to enable the output to be successful. Resistor R007 and its predecessor, the -12 v power supply, are essential to the operation of Q001 and are therefore shown on the diagram as its predecessors. Other predecessors to Q001 are C001, R001, R002, and C001, all of which must work.

One of the predecessors of C001 is R008, whose predecessor is the output "0". This relationship is shown on the diagram as a feedback loop from the output to C001. The other predecessor of C001 is C003, which is shown on the diagram with the input pulse as its predecessor.

The predecessor of R001 is the +12 v power supply. The predecessors of R002 and C001 are R003, the -12 v power supply, and Q002. R003 and the -12 v power supply are also predecessors of Q002, as are C002, R004, R006, and C002.

R005, a predecessor of C002, has Q002 as its predecessor, and is shown as a feedback loop from Q002. The other predecessors of C002 are C004 and the input pulse. The predecessor of R004 is the +12 v power supply. For R006 and C002 the predecessors are R007, the -12 v power supply, and Q001. The path from Q001 to R006 and C002 forms another feedback path.

Figure 18(b) is the completed reliability diagram.

The final reliability diagram, Figure 18(c), is obtained from Figure 18(b) by "cutting" the tree circuits. This procedure is a straightforward application of Steps 1 and 2 of Rule 4 (page 52).

The element table and part class failure information table are shown in Figures 19(a) and 19(b), respectively.



(a) Element Table

System/Subsystem Description		Flip-flop		
Diagram Name	Output	No. of Elements on Table	16	Checked (initials)
Length of Element Name	5	Part Classes Checked	(initials)	

p 1 of 1

Element	Class	System/Subsystem Description Flip-flop		No. of Operating Modes 2	Numbers Checked	t	
		Use With Table No.	Diagram Name Output	(Less OK)			
Checked for Completeness							
Element Class	Mode No.	Designation No.	P/Blank	Probability/Rate	K <sub>1</sub>	K <sub>2</sub>	
							op 1
Diode	1			770000007	321000003	180000000	100000001
Diode	2			230000007	193000003	180000000	100000001
Resistor	1			138750007	385000003	100000001	100000001
Resistor	2			750000009	640000002	100000001	100000001
Capacitor	1			648750007	345000003	100000001	100000001
Capacitor	2			637500008	207000003	100000001	100000001
Transistor	1			389200006	321000003	900000000	100000001
Transistor	2			233100006	193000003	900000000	100000001

(b) Part Class Failure Information Table

FIGURE 19

AR-64-366

TABLES FOR FEEDBACK CIRCUIT (FLIP-FLOP)

## 4.7 Reading Printouts

### 4.7.1 Engineering Printouts

As a general practice, the Part 3 program will return one set of printouts for each reliability diagram with its accompanying set of three lists. Similar printouts can be obtained on parts of a diagram. If the latter are desired, the analyst must tell the EDP representative which parts should be separately processed. Such a separate printout must be complete in that it must show the basic inputs; i.e., it is not possible to take a part out of the middle of a diagram without making a separate diagram.

The format of a printout is shown in Figure 20, and a partial reproduction appears in Figure 21. A printout contains the following information:

- (a) The list of elements, with the element class to which each element belongs
- (b) A list of element classes, which lists for each class and each failure mode the probability of failure in that mode
- (c) Listed in order, the terms which make up the expansion of the formula.

These terms have the form

$$+ A \cdot B \cdot C \cdot D$$

where the letters represent conditions that elements operate in certain modes. The printout then lists the element names of A, B, C, and D in columnar form and shows for each:

- (1) the requirements on its modes
- (2) the probability that each requirement is met

For each term, the probability that the conditions are all met is printed with the sign of the term. That is, if  $p$  is the probability that all conditions were met, then  $+p$  is printed if the term had a  $+$  sign,  $-p$  if the term was, for example,  $-ABCD$ . This printout -- as described in (c) -- is fairly time-consuming, and it can be suppressed.

(a) Element List

Column	1	2
Content	Element name	Part class name

(b) Part Class Probabilities

Column	1	2	3	4
Content	Part class (1) name	Mode number	Mode designator	Probability (2) that the part class is in designated mode

(c) Term List, Probabilities  
(This printout-out can be suppressed)

Column	1	2	3	4	5	6
Content	Term number (3)	Element name	Must be/must not be in	Numbers of included/excluded modes	Probability that element does not satisfy condition <sub>4</sub> (2) in columns 3 and 4(2)	Probability that the term satisfies(4) all its conditions

- {1} A space is left between part classes.
- {2} In floating point notation.
- {3} Three spaces are left between terms.
- {4} The sign on the probability is the sign of the term.

Note: The system probability of success (part d) is printed on a separate sheet.

FIGURE 20  
FORMAT OF PRINTOUTS OF PART 3 PROGRAM

DEFINITIONS OF ELEMENTS TO PART CLASSES

PART CLASS ID.	ELEMENT SYMBOL
RESIS	R0010
RESIS	R0020
RESIS	R0030
RESIS	R0040
RESIS	R0050
RESIS	R0060
RESIS	R0070
RESIS	R0080
CAPAC	C0010
CAPAC	C0020
CAPAC	C0030
CAPAC	C0040
DIODE	D0001
DIODE	D0002
TRANS	Q0010
TRANS	Q0020

LISTING OF PROBABILITIES PART CLASS BEING IN SPECIFIED STATES

PART CLASS ID.	MODE NO.	MODE DESIG.	PROR. OF MODE
RESIS	001	OP	.53418607 E-05
RESIS	002	SH	.47999998 E-07
RESIS	003	NG	.00000000 E 00
RESIS	0 K		.99999461 E 00
CAPAC	001	OP	.22381624 E-04
CAPAC	002	SH	.13196241 E-05
CAPAC	003	NG	.00000000 E 00
CAPAC	0 K		.99997629 E 00
DIODE	001	OP	.44490501 E-05
DIODE	002	SH	.79901968 E-06
DIODE	003	NG	.00000000 E 00
DIODE	0 K		.99999475 E 00
TRANS	001	OP	.11243355 E-03
TRANS	002	SH	.40488650 E-04
TRANS	003	NG	.00000000 E 00
TRANS	0 K		.99984707 E 00

LISTING OF CONJUNCTIONS AND PROBABILITIES OF THEIR OCCURRENCE

MODE DICTIONARY 1 2 3 4  
OP SH NG OK

TERM NO.	ELEMENT	REQUIREMENT	PROB. REQ. IS NOT MET	PROB. CONJUNCTION OF REQ. MET
1	Q0010	MUST BE IN MODE 4	.15293000 E-03	
1	R0070	MUST BE IN MODE 4	.53900000 E-05	
1	R0080	MUST BE IN MODE 4	.53900000 E-05	
1	C0030	MUST BE IN MODE 4	.23710000 E-04	
1	D0001	MUST BE IN MODE 4	.52500000 E-05	
1	R0010	MUST BE IN MODE 4	.53900000 E-05	
1	C0010	MUST BE IN MODE 4	.23710000 E-04	
1	R0030	MUST BE IN MODE 4	.53900000 E-05	
1	R0050	MUST BE IN MODE 4	.53900000 E-05	
1	C0040	MUST BE IN MODE 4	.23710000 E-04	
1	D0002	MUST BE IN MODE 4	.52500000 E-05	
1	R0040	MUST BE IN MODE 4	.53900000 E-05	
1	C0020	MUST BE IN MODE 4	.23710000 E-04	
1	R0060	MUST BE IN MODE 4	.53900000 E-05	
1	Q0020	MUST BE IN MODE 4	.15293000 E-03	
1	R0020	MUST BE IN MODE 4	.53900000 E-05	

.99954575 E 00

PROB. OF SUCCESS IS .99954575 E 00

AR-64364

FIGURE 21

PARTIAL PRINTOUT OF PART 3 PROGRAM FOR FEEDBACK CIRCUIT (FLIP-FLOP)

- (d) The system probability of success is printed separately. All numbers are printed in floating point form, which is explained on page 38.

In Figure 21, failure mode 003, designated "NG", is a "dummy" mode used to show the probability that the element is not in any of the other listed modes. In this printout, modes 001, 002, and OK represent the only possible states (insofar as this particular analysis is concerned). Therefore the probability for failure mode 003 is zero.

#### 4.7.2 Intermediate Printouts

An intermediate printout, available in the output of the Part 1 computer program, provides a formula representing the condition for system success as derived from the diagram. The computer from which the sample printouts were obtained uses unconventional symbols for logical connectives and parentheses. These symbols are listed in Table 1.

TABLE 1 COMPUTER SYMBOLS FOR LOGICAL SIGNS		
Logical Sign	Meaning	Computer Symbol
(	Left paren	⌈
)	Right paren	⌋
&	"and"	∩
v	Inclusive "or"	,
∨	Exclusive "or"	;

The analyst may find it useful to rewrite the formula printed by the machine, using conventional symbols.

The formula is interpreted in the usual manner, as explained in the appendix. An analyst who is accustomed to reading such formulas is well advised to compare them with the diagrams from which they were obtained, as an additional check on the accuracy of the diagram and of the transposition to computer inputs.

As an option, the Part 1 program can print out partial formulas, and these, too, can serve as checks on part of the diagram. The principal value of this feature, however, is to provide two formulas from one diagram. This capability is useful if a part of the system shown on the diagram is duplicated elsewhere, and a subsystem success probability is therefore needed.

Figure 22 shows, as an illustration, Part 1 printouts for the series-parallel system (906 signal) and the feedback (flip-flop) circuit used as examples in Sections 4.6.1 and 4.6.2. In each instance, the formula as printed by the machine is also shown converted into conventional symbols. As an additional illustration, the printout for the flip-flop includes the output to the Part 2 program.

Formula printed by machine

0001000001R007000001R006000001C0030C0001CR00100001R001000001C001000001
0001 0001000001R005000001C004000001CR002000001R004000001
0001 0002000001R006000001C002000001R002000001

Formula converted to conventional symbols

(((((044/op & K422/OK & (((043/op & K421/OK & ((R031/OK & CR32/op v R33/OK
& CR34/OK)))) & K421A/OK v (((R31/OK & CR32/op v R33/OK & CR32/op v R33/OK
& CR34/OK)))) & K42-1A/OK v (((R31/OK & CR32/op & ((((((43/op & K42-1/OK
& ((R31/OK & CR32/op v R33/OK & CR34/OK)))) & K421A/OK v (((R431/OK & CR32/op
v R33/OK & CR34/OK))) & K421A/sh))) & CR39/op v ((044/op & K4-23/OK &
((R31/OK & CR32/op v R33/OK & CR34/OK)))) & K422B/OK v (((R31/OK & CR32/op
v R33/OK & CR34/OK)) & K422B/sh))) & K423A/OK v ((((((43/op & K421/OK
& ((R31/OK & CR32/op v R33/OK & CR34/OK)))) & K421A/OK v (((R31/OK & CR32/op
v R33/OK & CR34/OK)) & K421A/sh))) & CR39/op v (((44/op & K423/OK & ((R31/OK
& CR32/op v R33/OK & CR34/OK)))) & K422B/OK v (((R31/OK & CR32/op v R33/OK
& CR34/OK))) & K422B/sh))) & K423A/sh)))

(a) Series-Parallel System (906 Signal)

Formula for System Success

Formula printed by machine

0001 00001000001R007000001R006000001C0030C0001CR00100001R001000001C001000001
0001 0001000001R005000001C004000001CR002000001R004000001
0001 0002000001R006000001C002000001R002000001

Formula converted to conventional symbols

Q1/OK & ((R7/OK & ((R8/OK & C3/OK)) & CR1/OK & R1/OK & C1/OK & ((R3/OK
& ((R3/OK & ((R5/OK & C4/OK)) & CR2/OK & R4/OK & R7/OK & C2/OK & R6/OK))
& Q2/OK)) & R2/OK))

Output to Part II

0001 0001000001R007000001R006000001C0030C0001CR00100001R001000001C001000001
0001 0001000001R005000001C004000001CR002000001R004000001
0001 0002000001R006000001C002000001R002000001

(b) Feedback Circuit (Flip-Flop)

FIGURE 22

PART 1 PRINTOUTS

## 5. PREPARING COMPUTER INPUTS

### 5.1 Coding Diagrams

The discussion of the work flow in CRAM (page 7) pointed out that the first computer program accepts the diagram and produces a formula.

The computer program works serially. It constructs a formula step by step, adding to it on the basis of the last card it has read; it then reads the next card, representing the next block, and continues making the formula. This process is possible only if the cards are so ordered that, when a block is reached, all the formulas for all the predecessors of this block are already available in the machine. One set of coding instructions will deal with this numbering of blocks.

Again, if a block is one of several predecessors to another, a formula for it must be constructed and stored while the formulas for the other predecessors are made. This formula must be coded. If a block has several predecessors, the formulas for these predecessors must be retrieved from storage, for use in making up the formula; this step must be coded into the card, since it is a special operation. Convenience blocks must be stored in a special place, and, if there is no element corresponding to them, care must be taken to exclude the name of the convenience block from the formula. This exclusion must be coded in the card. Finally, input blocks are peculiar in that they have no predecessor, and this fact must be coded. The second set of instructions deals with the coding of information concerning blocks, such as that described above.

A third set of instructions describes how to code the connections, i.e., which are the predecessors, and what logical function is required of them. In the same set, the coding of the element name and mode is described in 1-2-3 fashion (as explained in Section 4.2.2).



Finally, there are several options concerning the printout of the program; these are described and coded in the fourth and last set of instructions.

The preparation of input cards is a three-stage process:

- (1) The diagram is marked with the necessary information, which consists, for each block, of the block number and block type.
- (2) The auxiliary information (length of element name, number of operating modes) is used to set the format for the element code.
- (3) Code sheets are prepared from the marked diagram, in accordance with the format described in Section 5.1.3.

These three stages are now described in detail.

#### 5.1.1 Marking the Diagram

##### 5.1.1.1 Numbering Blocks

To conserve space in the computer, and also to enable the program to work serially, all blocks are given numbers, as shown in Figure 23. These are assigned by the following rule.

#### Rule for Numbering Blocks

- (1) Block numbers consist of four places, XXXX, starting with 0001, and continuing in order.
- (2) Convenience blocks carry, in addition, a "+" punch over the high-order position, e.g., 0<sup>+</sup>013, if block number 13 is a convenience block.
- (3) The number 0001 is given to the upper left-hand input block.
- (4) If 0001 is the only input to its successor, the latter is given the number 0002; if it is not the only input, the left-most input block heading the next branch to the right is given the number 0002.

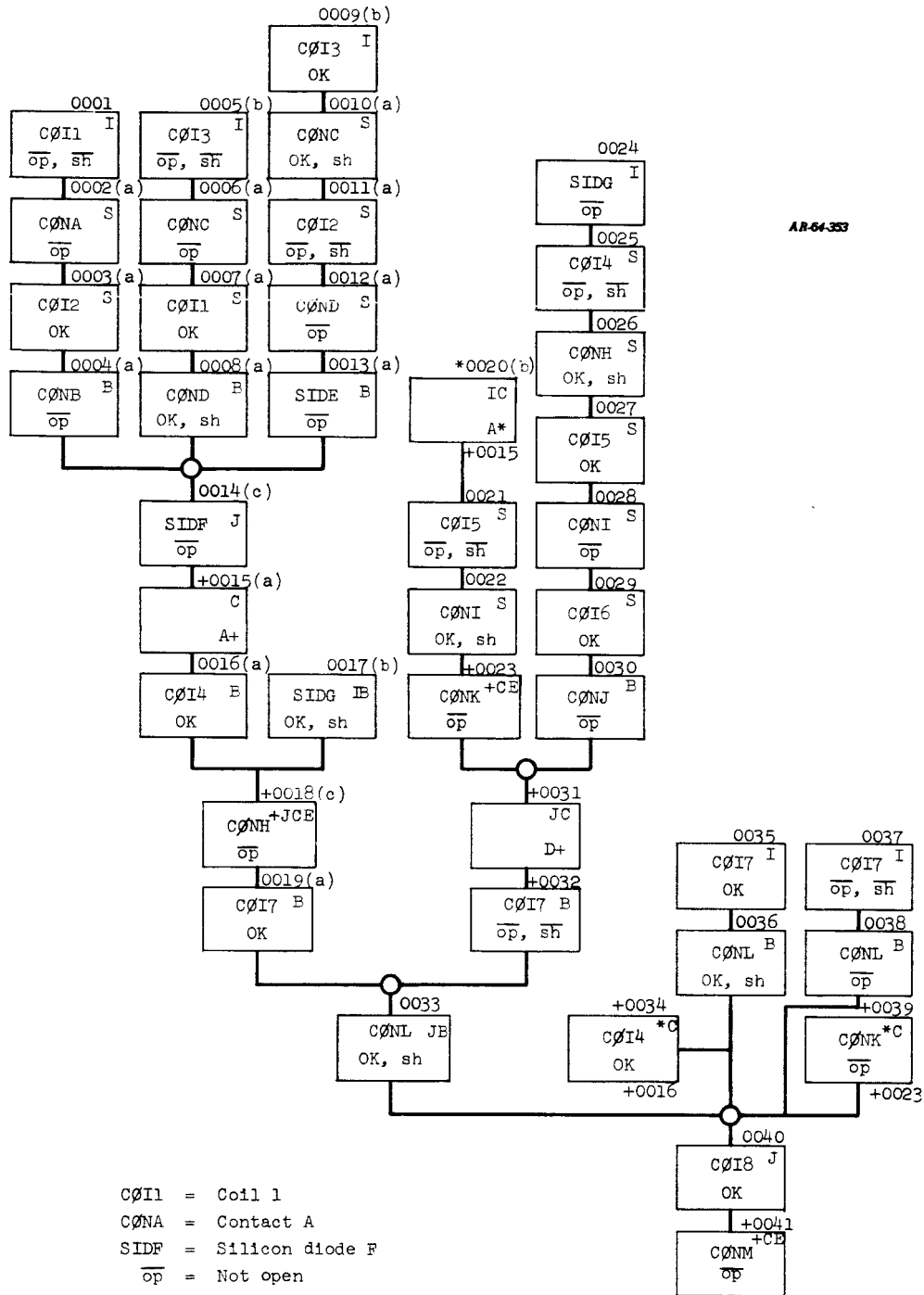


FIGURE 23  
 TEST RELIABILITY DIAGRAM FOR CRAM PART 1

- (5) Continue in the same fashion, always giving the next number to:
- (a) a series successor, if there is one, or
  - (b) to the left-hand top element of the next branch to the right, if there is one not yet numbered, or
  - (c) to the successor, if all inputs now have numbers.

The first 20 blocks on Figure 23 have been marked (a), (b), or (c), depending on which of the above clauses applied.

#### 5.1.1.2 Coding the Block Types

The different block-types which can occur on a diagram, with their definitions, are given in Table 2.

TABLE 2		
TYPES OF BLOCKS ON A RELIABILITY DIAGRAM		
Name	Definition	Symbol
Input block	No predecessor	I
Convenience block with element	See page	CE
Convenience block without element	See page	C
Junction block	A block with more than one predecessor	J
Branch block	One of several predecessors to its successor	B
Series block	None of these	S

A block may be of several types; for instance, it may be a convenience block and a junction block, or it may be an input block and a branch block. Figure 23 was constructed to test the Part 1 computer program, and it contains all possible combinations of block types several times. These combinations are listed in Table 3. The block type must be written on the diagram before code sheets can be prepared as specified below.

TABLE 3 COMBINATIONS OF BLOCK TYPES		
Symbol	Types in Combination	Example
I	Input block (single element)	0001,0024
IC	Input and convenience block	+ 0020
IB	Input and branch block	0017
J	Junction block	0040
JB	Junction and branch block (element)	0033
JCE	Junction and convenience block with element	+ 0018
JC	Junction and convenience block without element	+ 0031
CE	Convenience block with element	+ 0023
C	Convenience block without element	+ 0015
B	Branch block (single element)	0036
S	Series element (none of the others)	0028

#### Rule for Coding Block Types

- (1) Each block must be given the symbol specified in Figure 24, part (b).
- (2) When a convenience block appears with an "\*", i.e., when a previously stored convenience block is called for, the number of the block where it was first named must be written under the starred block. The block where it was first named has the name with a "+" in the corner.
- (3) Convenience blocks with stars are coded as "C", or "JC", i.e., as blocks without elements. These are the same blocks mentioned above. On first mention, i.e., when they carry a "+", convenience blocks are coded C or CE, depending on whether or not they represent an element.

(a) Information on Code Sheet

Columns	1-4	5,6	7,8,9	10	11-10+m(1)	10+m+1	10+m+2 -10+n+1	10-n+2 - end, and additional cards as required report 1-10+n+1
Entry	Block number	Card number	Block types (see part b)	Print-out instruction (see part c)	Element name, if any; otherwise blank	Not (2) slot	Operating mode indicator: 1 if mode is mentioned; 0 if mode is not mentioned (3)	(a) For convenience blocks, the second (lower) number on diagram (b) For series blocks, the number of the predecessor block (c) For junction blocks, the formula expressed in block numbers enclosed in parens.

- (1) m is the number of symbols in element names. If the name has < m symbols, leave the last spaces blank.  
 (2) Code a 1, if the elements are marked by exclusion (op, OK), a zero if marked by inclusion (op, OK).  
 (3) n is the number of operating modes allowed, including OK and OK. Consult mode list to find the position corresponding to each mode.

(b) Code for Block Type

Block Type	I	IC	IB	J	JB	JCE	JC	CE	C	B	S
Code C-7	I	I	I	J	J	J	J	blank	blank	blank	blank
Code C-8	blank	blank	+	blank	+	2	2	1	2	+	blank
Code C-9	blank	1	blank	blank	blank	blank	1	blank	blank	blank	blank

(c) Code for Printout

Type of Printout	Code in Column 10
Print and punch (1) formula in block numbers, and element table	1
Print and punch (1) formula in terms of element designators	2
Print and punch last convenience entry in block numbers and element table	3
Print and punch last convenience entry in terms of element designator	4

- (1) These are punched from the work area; they are used for intermediate printouts.

(d) Machine Logical Symbols

Conventional Symbol	Machine Symbol
(	%
)	□
&	£
v	,
∇	:

AR-64-377

FIGURE 24  
CODE SHEETS FOR DIAGRAMS

### 5.1.2 The Element Designator Format

Each block containing an element name represents an assertion about the operating mode of the element. This assertion must be coded for insertion into the machine. The procedure follows:

#### Rule for Determining Format

- (1) Determine the maximum number of symbols in the element names on the diagram. This number is  $m$ .
- (2) Let  $n$  be the number of operating modes on the "list of operating modes" supplied with the diagram. The last two modes should be  $\overline{OK}$  and  $OK$ , in that order. This point should be checked.
- (3) The number of columns needed for an element designator is now

$$m + n + 1$$

### 5.1.3 Preparing Diagram Code Sheets

The diagram code sheets are made up of the following information:

- (1) The block number
- (2) The card number for this block (it may happen that two cards are needed to represent a block)
- (3) A block-type code, as illustrated in Figure 24, part (b)
- (4) A print-out instruction, if wanted. This code can be used to obtain intermediate printouts.
- (5) Element name. (If the element name is less than  $m$  symbols in length, leave blanks for the last spaces.)
- (6) Element tag, describing the required operating modes
- (7)
  - (a) For convenience blocks called in, the block number at which they were originally defined
  - (b) For series blocks, the block number of the predecessor

(c) For junction blocks, the required formula

The formula appearing as item 7c, above, is the propositional function of the predecessor blocks needed as input to the junction block. It uses special symbols for "and", "or", and the "exclusive or" which are displayed in Figure 24, part (d). Instructions for making these formulas may be found in the appendix. Figure 24 shows the detailed coding instructions.

## 5.2 Inputs to Part 1 and Part 2 Computer Programs

The inputs to the Part 1 computer program now consist of:

- (1) The program deck
- (2) A header card having:  
m in columns 1-3 (m is the maximum length of element names)  
n+1 in columns 4-6 (n is the number of operating modes,  
including  $\overline{OK}$  and OK)  
H in column 80, to indicate that it is a header card.
- (3) The cards representing the diagram, as punched from the sheets described in Section 5.1.3. These cards must be in order.

The input to the Part 2 computer program consists of:

- (1) The program deck
- (2) A header card having:  
m-1 in columns 1-3 (m is the maximum length of element names)  
n+1 in columns 4-6 (n is the number of operating modes,  
including  $\overline{OK}$  and OK)  
H in column 80, to indicate that it is a header card.
- (3) The output of the Part 1 program

## 5.3 Coding Element Failure Information

### 5.3.1 Operating Modes

Operating modes will be recorded on a single card, which also carries the additional information indicated in Figure 25, part (a).

(a) Operating Mode List (one card)

Card Column	1-3	4-6	7-12	13-50	51-80
Content	n-1 <sup>1,2</sup>	Number of elements <sup>1</sup>	m <sup>1,3</sup>	Any heading to be printed on output page	Mode descriptions. Maximum of 14 different modes (2 letters each), of which "OK" is always last. <sup>4</sup>

(b) Probability/Rate List

Card Column	1-5	6-9	10	11-20	21-30	..(use additional cards)	71-80
Content	Part class name	Blank	P, if probabilities; blank, if rates	Probability, <sup>5</sup> or rate for first mode	Probability or rate for second mode <sup>5</sup>	etc.	Operating time <sup>5</sup>

(c) Application Factor/Time for Above Cards<sup>6</sup>

Card Column	1-5	6-10	11,12	21-30	31-70	71-80
Content	Part class name	Blank	Mode no. <sup>1</sup>	Application factor (or 1) <sup>5</sup>	etc.	Operating time <sup>5</sup>

(d) Element List

Card Column	1-5	11-m
Content	Part class name	Element name

- {1} Use leading zeros for 1- and 2-digit numbers, e.g., 001, 021.
- {2} n-1 is the number of operating modes excluding OK.
- {3} m is the maximum number of symbols in an element name.
- {4} See Section 5.3.1 for further explanation.
- {5} Floating point notation: ± ±

- {6} ± 0.abcdefg10±ij = abcdefghij  
If column 10 in the (b) card is blank, then one (c) card must be punched for each operating mode for which the (b) card has a rate. If no application factor has been given, a floating point 1 is entered in columns 21-30.

FIGURE 25

SHEETS FOR FAILURE INFORMATION



The operating modes are identified by their location numbers, followed by a two-digit mnemonic code. The modes  $\overline{OK}$  and OK must be included. For instance, if open and short are the only failure modes, the identification would be

o1 op o2 sh o3  $\overline{OK}$  o4 OK

with the spaces above not represented on the card.

### 5.3.2 Failure Information

The failure information is contained on two types of cards, which must be merged in the right order. If rates are given as failure information for the operating modes of a part class, then a card (or cards) containing the failure rates must be followed by cards containing, for each operating mode, the application factors and the time.

If the failure information is provided in the form of probabilities, the second type of card is not needed.

Detailed descriptions of these cards are given in Figure 25, parts (b) and (c), respectively.

### 5.3.3 Element List

The element list must also be converted into cards. The format is shown in Figure 25, part (d). These cards contain only the name of the element and the name of the part class to which it belongs.

## 5.4 Inputs to Part 3 Program

The inputs to the Part 3 program can now be summarized as follows:

- (1) The header card, with the system name
- (2) The card containing the operating mode list [Figure 25(a)]
- (3) The failure information cards [parts (b) and (c) of Figure 25], suitably merged
- (4) The element list [Figure 25(d)]
- (5) The output of the Part 2 program

## APPENDIX

### COMPUTATION OF RELIABILITY EXPRESSIONS AND FUNCTIONS

#### 1. General Discussion<sup>†</sup>

Once the reliability diagram in standard form has been constructed for a function diagram, there remains the task of constructing the reliability expression for the system, and finding the reliability function of the system in terms of the reliability function of the elements. Although this task is now being assigned to computers, a discussion of how it is performed is still necessary.

Given a reliability diagram in standard form, one starts the computation of the reliability expression from the top, i.e., with occurrences<sup>††</sup> that have no predecessors. For these occurrences, we have

$$A_i^* = A_i, \quad i = 1, \dots, n_A \quad (\text{A.1})$$

Immediate successors of these occurrences have propositional functions of the  $A_i$  only;

$$B_j^* = B_j \cdot \phi_{B_j}(A_{i_1}^*, \dots, A_{i_{n_B}}^*) \quad j = 1, \dots, n_B \quad (\text{A.2})$$

This equation expresses the definition of reliability. From (A.1) we can then substitute into (A.2) for the  $A_i^*$  and obtain

$$B_j^* = B_j \cdot \phi_{B_j}(A_{i_1}, \dots, A_{i_{n_B}}) \quad (\text{A.3})$$

which is an expression for the  $B_j^*$  in terms of individual occurrences. If C is an occurrence that has only B's for predecessors, and if, for example, the propositional function of C is

---

<sup>†</sup> In this discussion the standard logical notations "." and "v" are used for "and" and "or", respectively.

<sup>††</sup> "Occurrences" is the word used for blocks in reliability block diagrams.

$$\phi_C(B_1^*, B_2^*, B_3^*)$$

then, again, we have

$$C^* = C \cdot \phi_C(B_1^*, B_2^*, B_3^*) \quad (A.4)$$

By substitution from (A.3) we can express  $C^*$  as a function of individual occurrences, namely  $C$ ,  $B_j$ 's, and  $A_i$ 's.

Since the reliability diagram is a tree diagram, this substitution process will terminate with an expression for the final occurrence (which represents system success) in terms of the individual occurrences of the diagram.

In carrying out this process, it must be recalled that several occurrences may represent the same operating mode of the same element, and several occurrences may represent different modes of the same element. When this occurs, we have to use an arithmetic of sets to obtain the proper reliability expression. For instance, suppose that the diagram contains two occurrences of the element "a" in mode zero, denoted by  $A_0$ , and that the resulting reliability expression contains either

$$A_0 \cdot A_0 \text{ or} \quad (A.5)$$

$$A_0 \vee A_0 \quad (A.6)$$

The first expression is true if the element "a" is in state zero, and if element "a" is in state zero. However, since repeating the phrase does not impose any further restriction,

$$A_0 \cdot A_0 = A_0 \quad (A.7)$$

Similarly, in (A.6) the possibility that either "a" is in state zero or "a" is in state zero still amounts to "a" being in state zero, so that

$$A_0 \vee A_0 = A_0 \quad (A.8)$$

On the other hand, if we produce terms of the form

$$A/0 \cdot A/1 \quad (A.9)$$

requiring "a" to be operating in mode zero and mode 1, then this event will never happen. Thus,

$$A_0 \cdot A_1 = \textcircled{0} \quad (\text{A.10})$$

where  $\textcircled{0}$  designates an event that will never take place.

When one finally arrives at a reliability expression for a system, and applies the simplifications previously indicated, then the reliabilities of the different elements stated at different times must still be used to find the reliability of the system. This is still a fairly complicated procedure.

Suppose the expression is

$$S^* = (AB \vee AC \vee BC) \quad (\text{A.11})$$

with  $P_A$ ,  $P_B$ ,  $P_C$  representing the probabilities that "a", "b", and "c" are in the required states.

Then, by the rules for finding the probabilities of compound events, we have

$$\begin{aligned} \Pr(S^*) &= \Pr (AB \vee AC \vee BC) \\ &= \Pr (AB) + \Pr (AC) + \Pr (BC) \\ &\quad - [\Pr (AB \cdot AC) + \Pr (AB \cdot BC) + \Pr (AC \cdot BC)] \\ &\quad + \Pr (AB \cdot AC \cdot BC) \end{aligned} \quad (\text{A.12})$$

The parentheses on the third and fourth lines of Equation A.12 contain events like (A.7), which must be simplified. For example,

$$A \cdot B \cdot AC = ABC \quad (\text{A.13})$$

This simplification must be made before the numbers  $P_A$ ,  $P_B$ ,  $P_C$  are substituted; otherwise, we would get the wrong answers. For instance, if we substitute on the left-hand side of (A.13) we get

$$P_A^2 P_B P_C$$

while the right-hand side gives us

$$P_A P_B P_C$$

which is the correct term.

If all these contractions are made, then (A.12) becomes

$$\Pr(S^*) = \Pr(AB) + \Pr(AC) + \Pr(BC) - 2\Pr(ABC) \quad (\text{A.14})$$

in which a substitution of numbers can be made.

Section 2 gives precise instructions for obtaining reliability expressions and functions from reliability diagrams, and Section 3 presents further examples.

## 2. Rules and Procedures for Computing Reliability Expressions and Functions

The discussion in Section 1 may be formally summarized by the following instructions and rules. It is assumed that a reliability diagram in standard form is available.

### 2.1 Rules for Constructing Reliability Expressions

Rule RE1. Start computing reliability expressions with occurrences that have no predecessors. For one such occurrence, the reliability expression  $A^*$  is just the name of the occurrence itself. Thus,

$$A^* = A$$

Rule RE2. If  $A, \dots, B$  are the predecessors of an occurrence  $C$ ;  $\phi_C$  is the propositional function of  $C$ ;  $A^*, \dots, B^*$  are the reliability expressions for the occurrences  $A, \dots, B$ ; and, similarly,  $C^*$  is the reliability expression for the occurrence of  $C$ , then

$$C^* = C \cdot \phi_C(A^*, \dots, B^*)$$

Rule RE3. Throughout the computation of reliability expressions, apply the rules of Boolean algebra to simplify the expressions as far as possible. Examples of such rules are:

$$A \cdot A = A$$

$$A \vee A = A$$

$$A \cdot \textcircled{1} = A$$

$$A \cdot \textcircled{0} = \textcircled{0}$$

$$A(B \vee C) = AB \vee AC$$

and hence

$$A(B \vee AC) = AB \vee AC$$

etc.

Rule RE4. If  $A_1$  and  $A_2$  represent different states of the same element, then any term  $A_1A_2$  must be replaced by  $\textcircled{0}$ . Thus,

$$A_1A_2 = \textcircled{0}$$

where  $A_1$  and  $A_2$  represent different states of the same element. By combining this expression with the Boolean algebra rules in RE3 we get

$$A_1A_2BCD = \textcircled{0}$$

These rules suffice to obtain reliability expressions from reliability diagrams.

In order to see all the different parallel paths that can lead to system success, however, it is recommended that the final expression be transformed into a standard form. This transformation can always be made by use of the rules in RE3.

Rule RE5. Transform the final reliability expression until it has the standard form

$$\pi_0(\pi_1 \vee \dots \vee \pi_n)$$

where

$$\pi_0, \pi_1, \dots, \pi_n$$

are conjunctions, i.e., terms of the form

$$A_1 \cdot A_2 \cdot \dots \cdot A_n$$

and no element of  $\pi_0$  occurs in any of the **other**  $\pi_i$ .

In this presentation  $\pi_0$  has all the series elements, and the  $\pi_i (i > 0)$  contain parallel success paths among the remaining elements.

Rules RE1 to RE5 will produce reliability expressions in standard form.

## 2.2 Rules for Computing Reliability Functions from Expressions

Rule RF1. It will be found advantageous to compute the reliability function from the expression obtained by RE1 to RE4, i.e., before the expression has been put in standard form.

Rule RF2. In order to compute the reliability of a complicated expression, it is necessary to express this probability as sums and differences of probabilities of conjunctive terms. This is done by repeated applications of the rule

$$\Pr(A \vee B) = \Pr(A) + \Pr(B) - \Pr(AB)$$

and its generalization.

Rule RF3. Given a complex, bracketed expression, application of the rule RF2 should start with the outside bracket.

For example,

$$\Pr[A \vee B(C \vee D)] = \Pr(A) + \Pr[B(C \vee D)] \\ - \Pr[AB(C \vee D)]$$

Rule RF4. After every application of RF2 the simplification rules, RE3 and RE4, must be applied to the expressions produced. That is, for any term "Pr( - )" on the right, the expression in parentheses following "Pr" must be simplified if possible. Application of rules RF1 to RF4 will produce an equation of the form

$$\Pr(S^*) = \sum \pm \Pr(\pi_i)$$

where the  $\pi_i$  are conjunctive terms in simplest forms. To obtain reliability functions, we must substitute for the terms  $\Pr(\pi_i)$ .





The interpretation of this diagram is that

- (1) either A or F suffices to work C;
- (2) either B or G suffices to work D; and
- (3) both C and D must work for E to work.

Expressed in symbols, this information is:

- (1')  $\phi_C = (A^* \vee F^*)$
- (2')  $\phi_D = (B^* \vee G^*)$ , and
- (3')  $\phi_E = (C^* \cdot D^*)$ .

Now, if we apply the rules for the formation of reliability expressions, we get:

$$\left. \begin{array}{l} A^* = A \\ F^* = F \\ B^* = B \\ G^* = G \end{array} \right\} \text{ by RE1, since the occurrences are} \\ \text{without predecessors.}$$

Then:

$$C^* = C \cdot (A^* \vee F^*), \text{ from (1') and RE2;}$$

$$D^* = D \cdot (B^* \vee G^*), \text{ from (2') and RE2;}$$

and

$$\begin{aligned} E^* &= E \cdot (C^* \cdot D^*), \text{ from (3') and RE2,} \\ &= E \cdot [C \cdot (A \vee F) \cdot D \cdot (B \vee G)], \text{ by substitution.} \end{aligned}$$

The expression

$$E^* = E \cdot C \cdot D (A \vee F) (B \vee G) \tag{A.15}$$

is a reliability expression.

Since no element occurs twice, and only one state of each element occurs, the rules for simplifying expressions, RE3 and RE4, do not apply.

Finally, we transform (A.15) into standard form by using the rule of Boolean algebra:

$$(A \vee F)(B \vee G) = AB \vee AG \vee FB \vee FG$$

which yields the reliability expression, in standard form,

$$E^* = ECD(AB \vee AG \vee FB \vee FG) \tag{A.16}$$

The interpretation of (A.16) is that, for the system to be able to work, it is necessary that

- (1) E, C, and D all work; and that
- (2) at least one of the pairs A and B, A and G, F and B or F and G works.

To find the reliability function, let  $P_A, P_B, P_C, P_D, P_F, P_G, P_E$  stand for the reliability functions of A, B, C, D, F, G, and E, respectively, and suppose they are all independent.

Then, RF1 says that we must use expression (A.15) for our computation:

$$\Pr(E^*) = \Pr [ECD(A \vee F)(B \vee G)] \quad (A.17)$$

By Rule RF3, we expand this as

$$\Pr(E^*) = \Pr[ECDA(B \vee G) \vee ECDF(B \vee G)] \quad (A.18)$$

By Rule RF2 the right-hand side of (A.18) can be expanded as follows:

$$\begin{aligned} \Pr(E^*) &= \Pr [ECDA(B \vee G)] + \Pr [ECDF(B \vee G)] \\ &\quad - \Pr [ECDA(B \vee G) \cdot ECDF(B \vee G)] \end{aligned} \quad (A.19)$$

Now the last parentheses contains many sets twice; hence, by RF4 this last expression must be simplified first by RE3.

$$ECDA(B \vee G) \cdot ECDF(B \vee G) = ECDAF(B \vee G) \quad (A.20)$$

Substituting (A.20) back into (A.19) yields

$$\begin{aligned} \Pr(E^*) &= \Pr[ECDA(B \vee G)] + \Pr[ECDF(B \vee G)] \\ &\quad - \Pr[ECDAF(B \vee G)] \end{aligned} \quad (A.21)$$

Now, by RF2, each of the parentheses can be again expanded:

$$\Pr[ECDA(B \vee G)] = \Pr(ECDAB) + \Pr(ECDAG) - \Pr(ECDABG)$$

$$\Pr[ECDF(B \vee G)] = \Pr(ECDFB) + \Pr(ECDFG) - \Pr(ECDFGB)$$

$$\Pr[ECDAF(B \vee G)] = \Pr(ECDAFB) + \Pr(ECDAFG) - \Pr(ECDAFBG) \quad (A.22)$$

where the rules for simplification (RE3) have again been applied to the last term on each line, i.e.,

$$ECDAB \cdot ECDAG = ECDABG$$

If we substitute (A.22) into (A.21), we will have expressed  $\Pr(E^*)$  as a sum and difference of probabilities of products; i.e.,

$$\begin{aligned} \Pr(E^*) = & \Pr(ECDAB) + \Pr(ECDAG) - \Pr(ECDABG) \\ & + \Pr(ECDFB) + \Pr(ECDFG) - \Pr(ECDFBG) \\ & - \Pr(ECDAFB) - \Pr(ECDAFG) + \Pr(ECDAFBG) \end{aligned} \quad (A.23)$$

Now if we substitute the P's for the probabilities of products, so that, for instance,

$$\Pr(ECDAB) = P_E P_C P_D P_A P_B$$

and factor out  $P_E P_C P_D$ , then we get the reliability function

$$\begin{aligned} \Pr(E^*) = & P_E P_C P_D \left[ P_A P_B + P_A P_G - P_A P_B P_G + P_F P_B + P_F P_G - P_F P_B P_G \right. \\ & \left. - P_A P_F P_B - P_A P_F P_G + P_A P_F P_B P_G \right] \end{aligned} \quad (A.24)$$

which is the final answer.

## LIST OF DEFINITIONS

active redundancy. A system has active redundancy if it can perform its function in two or more separate ways and all of the equipment operates at the same time.

acyclic. Without cycles (series-parallel).

and. An operation that makes a proposition from two other propositions, e.g., "Socrates is Greek, and two is greater than one." The resultant proposition is true only if both the original propositions are true.

application factor. Same as K factor.

circuit. A cycle, not necessarily maximal, in which only one element is repeated. See page 48.

conjunctive term. A term consisting of propositions connected only by and.

convenience blocks. A device used to represent a complete branch of a reliability diagram. See page 23.

cutting, cut. An operation performed to convert a diagram of a feedback system into a tree diagram. See pages 49-50.

cycle. Any group of elements, each of which is its own predecessor.

designator. A symbol or number used as a name for an element represented by a block on a diagram.

distinct circuits. Two circuits are distinct if at least one of them contains an element that is not in the other.

EDP. Electronic data processing.

element. The word used to denote the pieces into which a system is analyzed. They may be subsystems, sub-subsystems, parts, or even parts of parts.

element operating modes. The possible ways in which an element can operate or fail.

entrance. An element of a cycle that has an immediate predecessor outside the cycle.

exclusion. Permissible element modes are mentioned by exclusion if the modes which cause the system to fail are mentioned, each with an overbar; for example,  $\overline{op}$ ,  $\overline{sh}$ . See page 33.

exclusive or. A connective which makes a proposition out of two other propositions. The resultant proposition is true if either of the original propositions is true, but the original two propositions cannot both be true; e.g., "Two is greater than one or (exclusive) one is greater than two".

exit. An element that has an immediate successor outside the cycle.

failure-mode-and-effect analysis (FMEA). A listing of all element failure modes and the effects on system operation if they occur. See page 52.

feedback. Feedback occurs when, in a chain of elements, say  $A_0, A_1, \dots, A_n, A_0$ , each element is its own predecessor.

floating point. A way of writing numbers which is used in computers. The numbers are written as a decimal fraction followed by a power of ten. For instance, 345.5 is written  $3455000003$ , where the last two digits represent the exponent of the power of ten (3). The sign of the number is written over the eighth digit, the sign of the exponent over the tenth digit. See page 38.

header card. A card used as part of the input to a computer program which defines the format of the inputs, and also contains the name of the system.

inclusion. Permissible element operating modes are mentioned by inclusion if the modes which allow the system to function are mentioned; for example, op,ok. See page 32.

inclusive or. See or.

junction block. A block on a diagram with more than one immediate predecessor, i.e., a block where several branches join.

K factors. Application factors used to adjust the failure rates of parts to account for stresses and environmental conditions.

maximal cycle. A cycle to which no other element can be added without destroying the cyclic property. See page 48.

multimode system. A system that can perform different functions or the same function at different levels of effectiveness. The functions or levels are called modes, and, in general, a slightly different set of elements is required for operation in each mode. Hence, each mode requires a separate diagram.

Not Slot. Part of the complete element name. It indicates whether the mentioned element operating modes are excluded or permitted. See pages 31-32.

optional rule. The rule for constructing diagrams which covers the use of convenience blocks. See page 45.

or. An operation that makes a proposition out of two other propositions, e.g., "Socrates is Italian or two is greater than one." The resultant proposition is true if either or both of the original propositions are true.

Part 1 program. This computer program accepts inputs representing the blocks of a diagram, and produces a propositional function representing the condition for system success.

Part 2 program. This computer program accepts the output of the Part 1 program, and produces the expression for the system reliability in terms of the element reliabilities.

Part 3 program. This computer program accepts the output of the Part 2 program, the element list, and the failure information table, and produces the system reliability.

path. A chain of elements, each of which is an immediate successor of the preceding one, and in which no element is repeated. See page 49.

piece parts. The smallest piece of equipment that can be replaced, e.g., a tube, a diode, a gear.

predecessors. A block B on a diagram is a predecessor of another block C, if the path from B to the terminal block of the diagram passes through C.

program deck. The cards containing the computer program.

propositional calculus. The part of symbolic logic which deals with statements. A statement is a declarative sentence, e.g., "Socrates is Greek", "Two is larger than one."

propositional function. A proposition made up of simpler propositions and connectives, e.g., "Socrates is Greek and two is greater than one." And is a connective. The other connectives used are "or", and "either, or".

reliability block diagram. A graphical representation of the conditions for successful operation of the system.

series system. A system which operates correctly only if all the elements operate correctly.

standby redundancy. A system has standby redundancy if it can perform its function in two or more separate ways and the equipment for one of these modes is not switched on until the equipment(s) for the other mode(s) has (have) failed.

system modes. See multimode system.

tag. The part of the complete element name that mentions the element operating modes which are excluded (the number 1 entered in the Not Slot) or permitted (a zero entered in the Not Slot). See pages 31-32.

terminal block. The last block on a reliability block diagram. It represents system success.

tree diagram. A diagram in which each block has only a single successor (except the terminal block, which has none).

utility routines. Computer programs which can be used to solve a wide variety of problems, without requiring alteration.