RAYTHEON COMPANY
EQUIPMENT DIVISION
RAYTHEON

NASA CR-132441

RELIABILITY MODEL DERIVATION OF A FAULT-TOLERANT, DUAL, SPARE-SWITCHING, DIGITAL COMPUTER SYSTEM

FINAL REPORT

ER74-4108

25 MARCH 1974

PREPARED UNDER

NASA CONTRACT NAS1-12668

FOR

NASA LANGLEY RESEARCH CENTER
HAMPTON, VIRGINIA

RAYTHEON COMPANY
EQUIPMENT DEVELOPMENT LABORATORY
SUDBURY, MASSACHUSETTS 01776

RAYTHEON COMPANY
EQUIPMENT DIVISION

ER74-4108

RELIABILITY MODEL DERIVATION OF A FAULT-TOLERANT, DUAL, SPARE-SWITCHING, DIGITAL COMPUTER SYSTEM

ABSTRACT

A computer based reliability projection aid, tailored specifically for application in the design of fault-tolerant computer systems, is described. Its more pronounced characteristics include the facility for modeling systems with two distinct operational modes, measuring the effect of both permanent and transient faults, and calculating conditional system coverage factors. The underlying conceptual principles, mathematical models and computer program implementation, are presented in considerable detail.

March 1974

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont.)

LIST OF ILLUSTRATIONS

FIGURES

LIST OF TABLES

Page Intentionally Left Blank

PART I

The task of predicting computer reliability, and in particular that of fault tolerant configurations, is unfortunately a rather complex one that, of necessity, is performed with ever increasing frequency. The complexity therein arises primarily as a consequence of the rather large set of interactive factors that must be contended with if the prediction is to be meaningful. The quickening pace is the result of both increased application as, for example, in long life satellites and all digital flight control systems, as well as the general recognition that a multitude of reliability enhancement techniques are available and that the proper subset for a given application is best selected on a scientific rather than intuitive basis.

In recent years, a variety of computerized reliability models have been developed in order to minimize both the difficulty and the attendant inaccuracy long associated with such prediction. One or another of these gives proper recognition to such individual factors as coverage, multiple system operating modes, transient versus permanent failure, and so on. What is required, however, and not so readily available, is a unified model set incorporating the sum total of these, as well as a means of quantifying, rather than simply measuring the impact of, the most elusive of these, coverage. (This latter term represents the fact that the measure of reliability transcends hardware availability, and must in addition account for such issues as fault recognition, reconfiguration delay and application imposed time and accuracy constraints.)

The purpose of this report, then, is to describe the development and software implementation of a new series of mathematical models which are specifically designed to contend with these items. It thus portrays a reliability assessment system that provides a unique capacity both for calculating coverage and for measuring reliability as it relates to the totality of these factors. As such, hopefully, the system will serve as a moderate step forward in the evolution of a scientific approach to fault tolerant computer analysis.

.1 OVERVIEW.

The computer program itself, designated CARE2, is designed to run on a 60 bit CDC 6000 series computer system, using the RUN Fortran compiler, under either the KRONOS 2.1 or SCOPE 3.0 operating system. It is available in two versions, the first of which requires a field length of approximately 130K octal to load and execute and the second, approximately 100 K. The versions differ only with regard to their ability to provide output data in the form of reliability plots.

The program is based, in part, on an earlier version entitled CARE*. The latter is somewhat characteristic of its generation in that, although it calculates computer system reliability for a variety of possible configurations, it does not provide an accounting for the effects of transient faults, multiple system operating modes and most aspects of coverage.

The current program, with regard to its modelling capability, is comprised of three major and relatively independent portions. The first of these, designated "equations 1-6", is the collection of specialized reliability models included in original CARE. The second, referred to as either "equation 7" or the "dual mode model", is unique to CARE2, and provides the capability for assessing system reliability as it relates to the set of aforementioned factors. The third, known as the "coverage model", enables the calculation of this commodity under a variety of conditions, and thus serves as input for the dual mode model and, where applicable, equations 1-6.

---

*F.P. Mathur, "CARE Program (Computer-Aided Reliability Estimation)", Technical Memo No. 361-10, Jet Propulsion Laboratory, Pasadena, California, February 24, 1971

CARE2, like its predecessor, portrays a computer system as a series of one or more independent subsystems or stages, wherein system reliability, by definition, is equal to the product of stage reliabilities. In turn, a stage is comprised of a quantity of identical units, termed LRU's, a portion of which serve as active or on-line devices and the remainder, as standby spares. Each stage is then represented by a single equation (i.e., model), selected on the basis of its ability to adequately represent the required physical configuration.

By way of illustration, a typical computer system might be viewed as consisting of four stages representing, respectively, the input, central processing, memory and output functions. In turn, the input stage, for example, might itself consist of three units, whose outputs are majority voted, and a single standby spare. Given then certain simplifying assumptions, the stage can be characterized by equation 4, which represents a "hybrid/simplex" configuration as defined by Mathur*.

In addition, however, the current program also provides the means for analyzing a computer system (or portion thereof) in which the component stages are interdependent such that total reliability is not a simple product of stage reliabilities. This situation is typified by a multi-stage computer wherein, given insufficient LRU's to constitute a fully operational system, a portion of those that remain are switched off and a backup or degenerate mode of operation is entered (cf. Figure 2-1). In this instance, the failure of an LRU in one stage has a conditional effect on the quantity of on-line versus spare LRU's in a second and thus, given unequal failure rates in the two states, a reliability impact on the second stage.

This capability, as well as certain others, is unique to the dual mode model (equation 7). As a consequence of this, and the additional fact that the majority of equation 1-6 configurations

* op. cit.

are but specialized subsets of equation 7, the majority of systems can, in practice, be modelled without reliance on the original six. The following summary discussion thus centers solely on the dual mode and coverage models which, together, constitute both a complete evaluation system and the new modelling portion of CARE2.

.2 DUAL MODE MODEL

This model, which is appended to the original CARE program as the seventh of seven equations, to a large degree supplants the preceding six, and thus serves as the focal point for the majority of reliability evaluations performed within CARE2. The mathematical model which underlies it, though beyond the scope of this summary, is presented in detail in Section 3.1. The capabilities which the model provides, from the perspective of the user, are highlighted in the following paragraphs.

As noted previously, CARE2 represents a computer system as a series of stages, each corresponding to a single class of functional LRU's. The dual mode model, in terms of its software implementation, has a capacity for up to eight such stages, wherein each represents a grouping of like LRU's consisting of zero or more each of on-line and spare units.

The phrase dual mode is, itself, a reference to an ability to model a system comprised of two distinct quantities of on-line LRU's per stage, which correspond, respectively, to the number required for operation in each of two system modes. Thus, the object computer system is represented by two separately specified configurations which, in turn, characterize its hardware complement in both an initial and backup state.

This dual mode provision enables the representation of computer systems which, confronted with critical equipment failures, are designed to reconfigure into a second, and less demanding state. For example, in Figure 2-1, mode 1 is the preferred operating state as a consequence of its provision for rapidly discerning faults via the comparator. Given, however, a permanent failure in either the comparator or all but one of the LRU's of a particular stage, operation can no longer continue in that state. At this point, the object computer system is designed to reconfigure into its mode 2 state wherein sufficient hardware, barring further critical failure, is available to maintain a "degenerate" form of operation.

In addition to stages and their component LRU's, the model also provides for the representation of two object system "single point failure mechanisms." The first of these corresponds to the class of switch and comparator/voter failures which, on occurence, unconditionally cause degeneration from mode 1 to mode 2. The second represents the class of failures, termed catastrophic, which cause immediate and total loss of the system. Each is expressed, within the model, solely in terms of its corresponding permanent failure rate.

LRU's, on the other hand, are considered subject to both transient and permanent faults, wherein the rate of the latter is, in addition, conditioned by LRU status. As a consequence, the model provides for a separate LRU failure rate specification for each of three failure classifications i.e., transient, on-line permanent and standby permanent.

Given the occurence of mode degeneration in a computer system, it is typically accompanied by a partial equipment shutdown in which one or more fully functional LRU's are released from active service. These units then, as a function of computer design specifics, either re-enter the spares pool or, conversely, are not re-assignable and thus, from a computer reliability viewpoint, are purged from the system. The model provides for either possibility, in accordance with a user controlled software switch setting.

In total then, the dual mode model has the capacity for representing a computer system with either one or two distinct hardware configurations, and consisting of one to eight functional stages. In practice, the system is defined in terms of these stages, wherein the user specifies, for each stage, the mode 1, mode 2 and spare LRU count, as well as the applicable transient, on-line and standby failure rates. Additionally, the two single point failure rates and the spares reassignment method (on degeneration) are user specified at the system level.

.3 COVERAGE MODEL

For purposes of this study, coverage is defined as "the conditional probability, given a fault and sufficient hardware replacements to remedy it, that the system is returned to operational status in a form consistent with external time and accuracy needs." In other words, it is a measure of the reduction in system reliability due to issues other than those of hardware availability, and thus serves as an accounting for such factors as the inability to either detect, isolate and/or recover from, certain classes of faults. For example, given both a permanent fault in an on-line memory and a fully operational spare, the conditions of hardware success are, for the moment, met. Those of coverage, however, further dictate that the fault be correctly attributed to the memory, and that the replacement unit then be "reloaded" within certain time constraints.

It follows that coverage is not single valued for a given computer system. For instance, the ability to detect a fault is clearly a function of its location within that system. Further, the ability to recover, as from a memory fault, may depend on the presence of a second unit with identical storage contents, and this may be available in only one of the mode configurations.

From a system perspective then, coverage is readily seen to depend on fault subclass (location, as within an LRU), fault type (permanent or transient) and operating configuration (mode). In addition, it is related to at least two other factors; i.e., whether the fault results in a mode change (which may entail certain risk), and the status of spare LRU's (which are subject to prior failure, thus presenting a possible requirement for trial selection and testing in order to locate an operational unit).

The model provides for the calculation of a single conditional coverage value for each valid combination of these conditions. In turn, the set of computed values are then supplied to the appropriate reliability equation for inclusion in the system reliability assessment. Since, in practice, the original six equations are, at best, ill equipped to contend with coverage, the normal recipient is the dual mode model.

Since the same methodology is utilized to compute each of the conditional coverage values, the remainder of this discussion is addressed to the rationale underlying the modelling and calculation of a single value (i.e., for a fixed set of conditions). It should be specifically noted, however, that each of the parameters referenced therein has a potentially unique value for each set of conditions.

Given the occurence of a fault, the object system is presumed to be equipped with more or less suitable defense mechanisms to contend with it. The coverage model is based on the assumption that, in all but the unprotected case, these consist of one or more fault detectors (e.g., memory parity and software selftest) as well as associated fault isolation and recovery techniques. In this sense, conditional coverage is but an overall performance measure of these, as a group, and as they pertain solely to the fixed conditions.

In addition, it is assumed that, given multiple fault detectors (within the fixed conditions set), they are best represented as statistically independent processes. (Conversely, members of separate sets are treated as mutually exclusive). As such, on occurence of a fault, they can be viewed as competing with one another, though not necessarily with equal vigor. The "winner", if in fact there is one, then calls into play, in sequence, its associated isolation and recovery processes.

Given the detection of fault in the object computer system, the ability to identify the responsible detector can prove instrumental in constraining the isolation and recovery tasks which follow (i.e., individual detectors have more or less unique properties with respect to the type of faults they detect and when they detect them). As a consequence, the coverage model is designed to provide for the calculation of the conditional detection probability associated with each of the competing detectors, given its presence in a competitive situation.

In turn, these values are highly dependent on the following factors:

- The stand-alone success probability of each fault detector (i.e., the percentage of faults it would discern if it were the only detector present).
- The corresponding stand-alone detection rate associated with each.
- The complete time base or schedular interrelationship between each (i.e., individual repetition rates, where applicable, zero-offset delays, etc.).

Given this data as input, the model performs the calculations necessary (cf. Section 3.2) in order to establish the interrelationship that exists between the detectors, and thus provide the conditional detection probability and rate of each, when in competition with the others.

This information is then utilized, in conjunction with corresponding fault isolation and fault recovery time profiles, in order to produce the contribution to coverage (for the fixed condition set) associated with each of the individual detectors. Thus, their sum is the conditional coverage value associated with the set of conditions.

In this regard, the isolation profile is a rate (i.e., probability density) function which represents the performance characteristics of the isolation process corresponding to the particular detector. The referenced recovery profile is, in fact, two separate characterizations, both measured in terms probability (rather than probability density) versus time. The first of these, referred to as the error propagation function, is indicative of the reduction in recovery probability due to the potentially unconstrained propagation of erroneous data, throughout the computer system, during the interval between fault occurence and fault detection. The second, known as the time delay function, corresponds to the potential diminishment of recovery probability as a function of effective computer down time, and as measured from fault occurence on through detection, isolation and, where required, trial selection and testing of spares, until such time as the true recovery process is initiated. (The time delay impact associated with the recovery process itself is handled implicitly in the recovery probability function.)

Again, it should be emphasized that the above discussion describes the coverage calculation process associated with a single subclass location, while operating in a particular mode, and when confronted by a specific type of fault. Also, it should be noted that each of the fault detectors has a unique description, both within and between condition sets, and further that the isolation and two recovery processes also have unique descriptions for each possible combination of detectors and condition sets.

In perspective, then, the repetition of this process, once for each of the possible condition sets, constitutes the process referred to as coverage calculation.

Page Intentionally Left Blank

PART II

SECTION I

INTRODUCTION

The purpose of this combined study and software development effort was to provide an advanced analytical tool for use in estimating the reliability of various fault-tolerant, dual, spare-switching digital computer systems. As such, it was to include a means of accounting not only for permanent (i.e., hard) faults, but also transient faults (i.e., faults which do not persist, but whose effects can) and coverage factor (i.e.,the conditional probability of system recovery given that sufficient operational hardware is available).

As originally conceived, the phrase "dual system" was in reference to a configuration comprised of two identical computers, operating in parallel, and joined following their digital output stage by a comparator. Further, each computer was partitioned into four switchably replaceable units, and these in turn were backed up, at the system level, by a variable quantity of spares.

In the course of developing the accompanying mathematical models, an opportunity for increasing the problem solving capability rather significantly became apparent, but with it, a practical necessity that it be incorporated prior to any further software development. The resulting change, subsequently agreed to by both parties, amounted to substitution of the phrase "dual mode" for "dual system" in the original work statement. With it, however, came a revision of the mathematical models and software so as to include a capability for evaluating virtually any configuration involving sparing at the unit level, and accompanied by any kind of fault detection, isolation and recovery procedures, provided that only two system operating modes are required and that software based numeric limits are not exceeded. The resultant modeling capability thereby extends, for example, to the majority of both "N-modular-redundant" and "hybrid" systems.

The software phase of the effort was specified to be implemented as an addition to an existing statistical reliability prediction program entitled CARE*. This program was developed, by the Jet Propulsion Laboratory, to "serve as a computer aided reliability design tool to designers of ultra-reliable fault-tolerant systems, by facilitating reliability computation, data generation, and comparative evaluation".

CARE's rather singular emphasis on permanent faults and one mode system operation was seen, however, as a rather significant shortcoming with respect to the defined programmatic requirements, and thus indicative of the need either for a serious revision or a nearly stand-alone addition to the existing base.

As a practical matter the latter course was selected, and with it, the corollary development of a dichotomy between old and new halves of the program. This division, though most pronounced in the inner workings of the software, also evidences itself in both the input/output structure and the frequent overlapping of capabilities wherein a given system can be evaluated with either half, though with rather marked differences in terms of modeling depth.

The new portion of this program, at times supplemented by the old, can be used as a reasonably sophisticated analytical tool to assess the reliability of a wide variety of fault-tolerant computer configurations. In addition, it is particularly attuned to the needs of performing sensitivity analyses on variations of a single configuration including, for example, measuring the effect of changes in schedule on the usefulness of a software self-test routine.

* F.P. Mathur, "CARE Program (Computer-Aided Reliability Estimation)", Technical Memo No. 361-10, Jet Propulsion Laboratory, Pasadena, California, February 24, 1971

It should be specifically noted, however, that the necessary inputs to the program include a statistical specification of all significant fault detection, isolation, and recovery mechanisms within the system, i.e., it must be furnished data describing the properties of the various coverage features included in the object computer. The generation of this data is, in some cases, quite difficult, and the values obtained, often subjective. This is seen, however, not as a shortcoming of the study, but rather as an indication that reliability prediction is, itself, a difficult undertaking, and that much additional work remains before it can truly become a science.

The following sections describe the results of the subject effort, including development of the conceptual base, its implementation in terms of mathematical models and subsequently software, and instructions on its utilization in the form of a "users guide" for operation of the program.

Page Intentionally Left Blank

SECTION 2

CONCEPTUAL DEVELOPMENT

The achievement of fault tolerance in a reconfigurable computer system transcends the issue of hardware survivability; a second and equally essential prerequisite is that of fault awareness and the consequent ability to re-establish control successfully in a properly reconfigured system. This necessitates that the system include a methodology for detecting, isolating, and ultimately recovering from any of a necessarily large percentage of faults. Collectively, these methods are referred to herein as "fault tolerant features" (FTF's), and their integrated function is to provide "coverage".

At issue then, in a properly directed system design process, is the selection of an effective FTF set for a particular configuration, with the limiting constraint being that each provides a degree of coverage only at the expense of additional hardware and/or reduction in effective computational throughput. Up to a point therefore, the addition of FTF's enhances the probability of successful operation; beyond that, the ever decreasing gain in coverage (via overlap of faults covered) is quickly overshadowed by the failure mechanisms inherent in the FTF implementation hardware.

This study, which is itself the outgrowth of a current necessity for quantifying fault tolerant computer performance has, therefore, as its fundamental objective, the development of a computer based method for assessing system reliability as it relates to both hardware configuration and FTF's employed therein.

Of the three main tasks in this effort, deriving the mathematical reliability model for a dual mode system, providing for calculation of the coverage factor, and extending the existing CARE program accordingly, the second is undoubtedly the most complex, and necessarily the most open to question. Consequently, much of the following discussion centers on this particular aspect of the study.

2.1 GROUNDRULES AND ASSUMPTIONS

In order to place bounds on both the magnitude of the study and the execution time required for the resulting computer program, certain simplifying assumptions must necessarily be made. At the same time, however, it is of even greater importance that all governing performance factors be properly accounted for. What is called for, then, is a somewhat delicate balance between too much and too little. Unfortunately, as is often the case when developing models, the quantitative information most desired for the decision making process is available only as an output from, rather than input to, the modeling task. As a consequence, therefore, it is necessary to rely rather heavily on intuitive processes rather than known phenomena.

In this study, the following considerations were judged to be both necessary and sufficient in order to provide for a representative reliability assessment:

2.1.1 Fault Simultaneity

Independent and simultaneous, or near simultaneous, fault occurrences are assumed to be sufficiently improbable as to justify ignoring them, both for the purpose of system evaluation and for that of system design as well.

2.1.2 Failure Rate

Constant values are assumed throughout. In particular, three separately specified quantities are provided for each of a quantity of LRU (line replaceable unit) types:

- λ - the permanent-failure rate of on-line (i.e., active) LRU's
- μ - the permanent-failure rate of standby (i.e., spare) LRU's
- γ' - the transient-failure rate of on-line LRU's.

In addition, two single-valued system level rates are included:

- $\hat{\lambda}_2$ - the permanent-failure rate of system components whose loss forces an immediate degeneration in operating mode (e.g., certain non-catastrophic failures in an output voter or comparator)
- $\hat{\lambda}_3$ - the permanent-failure rate of system components whose loss is catastrophic (e.g., loss of all input power to the system).

2.1.3 Coverage

For purposes of this study, coverage is defined as "the conditional probability, given a fault and sufficient hardware replacements to remedy it, that the system is returned to operational status in a form consistent with external time and accuracy needs". In turn, its value is viewed as functionally dependent on three conditional probabilities:

- The probability, given a fault, that it is detected
- The probability, given a detected fault, that it is properly isolated

- The probability, given a properly isolated fault and sufficient spares to remedy it, that a satisfactory cure is effected.

2.1.4 Coverage Conditions

Though often discussed as though it were a single valued quantity for a given system operating in a given environment, coverage is more accurately a set of values, each of which corresponds to a specific situation. In particular, the degree of coverage offered is, at a minimum, dependent on the following conditions:

- operating mode - whether the system is fully operational (e.g., 2 out of 2 channels functioning) or degenerate (e.g., 1 out of 2 channels functioning)
- fault location - whether the fault lies, for example, in the system's input, CPU, memory bit plane, memory word electronics or output area
- fault type - whether the fault is permanent or transient
- spares status - whether initial replacement units, as selected from the spares pool to remedy a fault, are themselves operational or faulty
- degeneration requirements - whether or not it is necessary, as a function of spares status, to change operating mode (i.e., to undergo transition) in the event of a fault.

As a consequence, a unique coverage value exists for each combination of conditions, and must therefore be accounted for in the reliability assessment.

2.1.5 System Failure

It is assumed throughout this study that a coverage failure inevitably results in total system failure. Specifically, this includes:

- failure to detect a fault
- failure to correctly isolate a detected fault
- failure to recover from a correctly isolated fault.

(In addition, of course, lack of sufficient hardware to constitute an operational, degenerate system also leads to the same consequence.)

While this is in general a reasonable assumption, it is possible to conceive of reconfiguration strategies that recover from certain transients, which would otherwise be disabling, by changing to a degenerate mode of operation (e.g., by entering a simplex mode). Similarly, in certain situations, other forms of coverage failure could conceivably also lead to a degenerate mode rather than to system failure.

2.1.6 Spares Reassignment

Given a degenerative failure, a quantity of fully operational on-line LRU's are typically released from active service. Depending on system architecture specifics, those so released may or may not be available for use as spares in the event of subsequent failure. Since either situation may prevail, it is necessary that both be accounted for in the model, with the determination of which is most appropriate for representing a given configuration left to the discretion of the user.

2.1.7 Spares Failure

Since spare LRU's are themselves subject to failure (at rate μ), it is necessary to provide for situations in which the spare selected to replace a failed on-line LRU has itself failed, and must consequently be replaced by yet a second (or, in the general case, nth) spare. In the model, this is provided for in the form of a single delta coverage value per fault classification per operating mode, wherein the as calculated value is a measure of the reduction in recovery probability resulting from incurred downtime during the conditional trial and error replacement period. In calculating this, it is assumed that failed spares are purged from the system in bulk on the occurence of mode degeneration, and as individuals following their selection as a replacement unit.

2.1.8 Fault Classification

The ability of a given detector to recognize a specific fault is, in most instances, highly dependent on the precise nature of the fault. For purposes of this study, it is assumed that two levels of classification are adequate for the purpose of so categorizing faults. The first of these, referred to as fault class, is the LRU type (e.g., CPU or memory) in which the fault occurs. The second, or fault subclass, is an arbitrarily selected portion of an LRU (e.g., arithmetic unit in a CPU).

Thus, by proper partitioning of the object system representation, first into LRU's or classes, and then into subclasses, the various faults can be categorized according to their specific detection characteristics.

2.1.9 Fault Detection

The representation of an object computer system as a series of fault subclasses underlies the approach to coverage measurement utilized in this study. Specifically, each subclass is by design a grouping of possible faults, so categorized solely on the basis of their common susceptibility to one or more fault detectors. In other words, the set of fault detectors associated with a particular subclass is, be definition of the subclass itself, comprised wholly of statistically independent members. In turn, such independence implies that each member detector has a finite probability of detecting any of the possible faults within the subclass, and that it therefore competes with all other members toward the objective of being first to detect a local fault.

This concept of competing fault detectors should not be construed to imply that a degree of equality need exist between them; on the contrary, the contest is of necessity as imbalanced as reality dictates.

For a moment perhaps, it is useful to look at the performance characteristics of a stand-alone detector i.e., one which has no competitors within the subclass. Given both singular occurences of random faults and infinite time, the detector has an associated non-zero probability of detection. This value, as well as the corresponding detection rate profile, forms a substantial portion of the detectors' characterization.

In addition however, it is necessary to consider the occurence rate of the detector itself i.e., the timing basis or schedule on which it executes. Two basic categories are apparent: periodic and aperiodic. The former is populated, in the main, by software based detectors (e.g., software selftest), whereas the latter group consists primarily of hardware detectors (e.g., memory parity). (In both of these cases, it is assumed that the initial execution of the test, if unsuccessful in detecting a specific previous fault, precludes any possibility of later detection, i.e., the fault is transparent to the test.)

Returning then to a competitive situation, and given each detector characterized in terms of its stand-alone detection probability, detection rate and schedule, it would be a relatively simple task to compute an overall subclass detection probability and expected time, based on the interaction (i.e., competition) between detectors. It would not be sufficient, however, for the following reasons:

- knowledge of the specific test by which a fault is detected is frequently utilized, in the object computer, to constrain the isolation process
- individual detectors vary in their ability to either prevent or minimize fault propagation (i.e., contamination of correct information via improper processing). Since this can have major influence on the probability of recovery, knowledge of the specific detector involved must be made available.

It is necessary, therefore, that a conditional detection probability and rate be computed for each subclass member, as the values relate to the totality of competitive detectors.

2.1.10 Fault Isolation

Unlike fault detection, the isolation of faults is assumed to be a single-threaded process in which a lone isolator seeks out the source of difficulty. Since the object system, in all probability however, incorporates multiple isolators (each tailored to a specific task), it is necessary to provide accordingly in the model.

In particular, it is assumed that each detection process is uniquely linked to an isolator such that, given detection of a fault by it, the corresponding isolator is unconditionally brought into play. In turn the isolator, by virtue of its tailored design, inherently capitalizes on whatever unique characteristics the detector may possess in order to constrain the bounds of the search.

By way of example, the isolation process associated with detection by memory parity is clearly minimal (if in fact it exists at all), whereas detection by an external voter implies nothing in the way of fault location, thus mandating a far-ranging and potentially time-consuming search.

As in the case of fault detectors, fault isolators are seldom, if ever, perfect, and thus have both a non-unity success probability and a finite process rate. For analogous reasons then, this must be accounted for in the impending fault recovery process.

2.1.11 Fault Recovery

The hardware aspects of fault recovery are reasonably well disciplined. Either a spare hardware unit and the needed switching apparatus are available, in which case that portion of the cure can be effected, or they are not, in which case the system either degenerates or fails completely (barring the use of techniques such as those of "graceful degradation" in the event of partial memory loss, etc.).

The coverage portion of fault recovery, on the other hand, must contend with all issues save those of hardware availability, and is thus confronted with a rather unwieldly grouping of considerations which, in turn, are considerably more difficult to deal with, both in the object system itself and, consequently, in the model as well.

In particular, it is felt that each of the following are of sufficient importance as to necessitate inclusion in the model:

- the subclass in which the fault is located (for reason of its potential influence on recovery difficulty)
- the time delay between fault occurrence and fault recovery, as it relates to the issue of system downtime. (Of necessity, this incorporates any time devoted to spares checkout including repetitions, where required, to locate a functional spare)
- the degree and location(s) to which fault propagation was contained, and the corresponding time interval during which it was unconstrained. (The model assumes here, as a first order approximation, that this interval starts on occurrence of the fault and ends on its detection. Compensatory adjustments, where called for, can be achieved by modifying the degree of sensitivity)
- the detector responsible for finding the fault, as it relates both to the degree of fault propagation and the time required for detection
- the system operating mode at the time of fault occurrence
- the type of fault involved (permanent or transient)
- the need, or lack of it, for mode degeneration.

In addition, successful recovery is assumed to be conditioned by the following:

- successful detection of the fault
- successful isolation of the fault
- in the case of a permanent fault, availability of a spare LRU or, barring that, sufficient on-line LRU's to maintain operation in a degenerate mode
- in the case of a permanent fault, successful detection of any failure present in a selected spare LRU.

2.1.12 D/I/R Mechanisms

Since each of the object system fault detection (D), isolation (I) and recovery (R) processes has the potential for reacting differently to a fault as a function of its location (subclass), the type of fault (permanent or transient), the system operating mode (fully operational or degenerate) and any requirement for degeneration (given certain hardware losses in a fully operational system), it is necessary to provide a means for modeling each possible reaction seperately. The total set of system reaction descriptions corresponding to detection by a particular detector is then referred to as a D/I/R mechanism.

By way of illustration, take the case of a CPU selftest fault detector. As its name implies, it is most proficient at detecting fault which occur in the CPU; nonetheless, it exhibits a finite capacity to detect those in other subclasses as well. For instance, given that the test executes out of memory, it follows that certain faults located therein will cause it to misexecute, thus resulting in an eventual fault indication (albeit for the wrong reason). Similiarly, though the test is reasonably adept at detecting permanent CPU faults, its capacity in the presence of transients is, at best, marginal.

It follows then, in the general case, that a detectors' performance may be expected to vary as a function of all the aforementioned environmental conditions. In analogous fashion, the isolation and recovery processes associated with that detector may likewise be expected to exhibit similar dependencies.

In context, then, a D/I/R mechanism is the total set of performance descriptors pertaining to a particular fault detector. As such, it consists of separate detection, isolation and recovery performance characterizations, one each for every meaningful combination of conditions.

2.1.13 Performance Characterization

As noted, each of the object system fault detectors, together with its associated fault isolation and fault recovery techniques, is represented in the model as a D/I/R mechanism. Each mechanism is then represented by a set of performance characterizations, one each for all combinations of detection, isolation and recovery processes across a spectrum of conditions.

In turn, each of the individual performance characterizations is expressed in terms of a function specification and a referenced function model. The latter are parametrically expressed probability and probability density functions which serve as a representative form or pattern for the characterization, and the former provide specific numeric details to round out the characterization. The end object of these (i.e., a fully described process representation) is then referred to as a function.

In particular, the following set of function models is viewed as sufficient to enable adequate characterization of the majority of known processes:

- Impulse - a burst of zero duration with finite integral, representing an instantaneous process
- Constant - a pulse with finite duration, representing a constant process
- Finite Pulse String - a series of equally spaced pulses, representing a discontinuous process
- Exponential - a decaying exponential, representing a continuous, but ever decreasing, process.

In practice (i.e., in the software implementation of the model), each function specification is identified, for reason of user convenience, by means of a unique function number. As a consequence, like process characterizations, once defined, can be elicited repeatedly via their identifying number.

Viewed as a whole, D/I/R mechanisms are utilized, on a one for one basis, to describe the totality of performance characteristics associated with each fault detector. In turn, mechanisms are defined in terms of function numbers, one each for all combinations of conditions and processes. Each function number corresponds directly to a function specification which, in turn, contains both quantitative data and a reference to a particular function model. The latter then is a parametric expression which describes the general form of the performance characterization.

2.1.14 DIET vs. D/I/R Process Representation

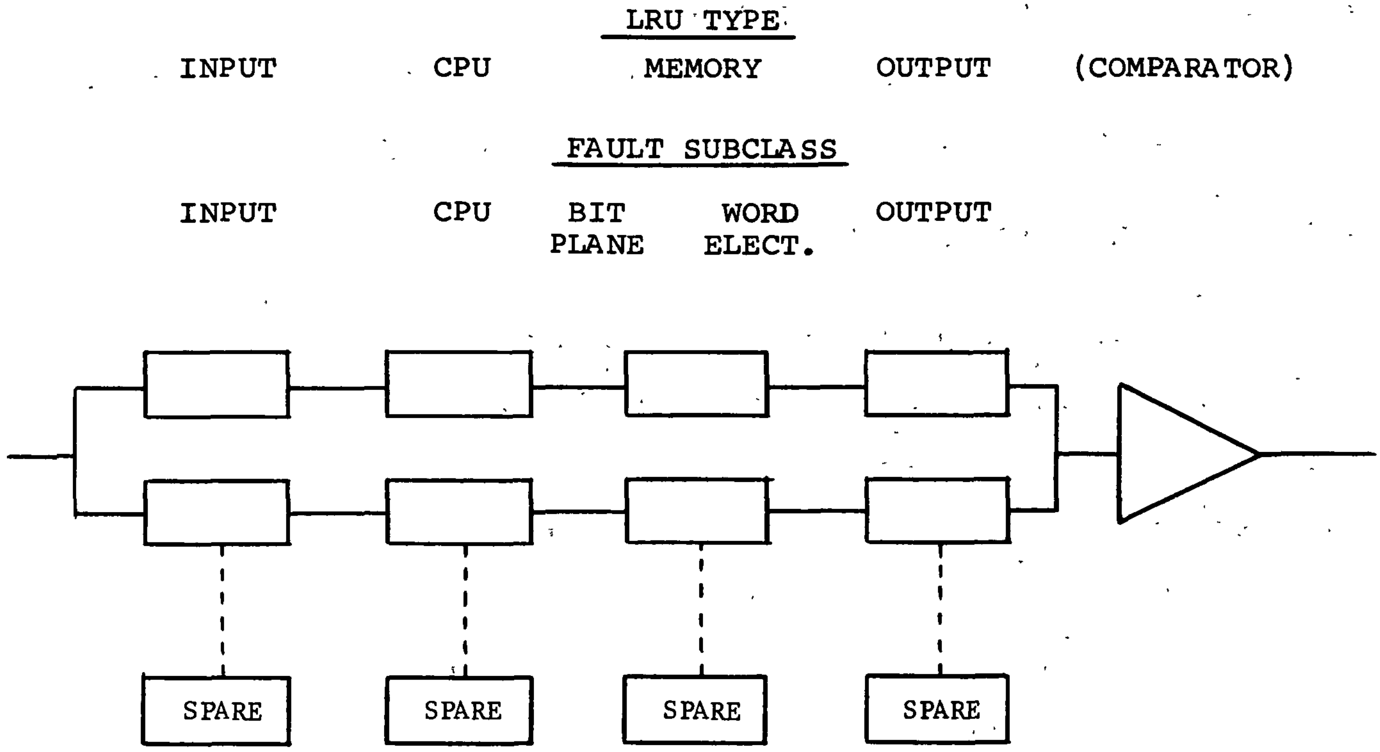
In previous sections, frequent reference has been made to the detection, isolation and recovery process representations which constitute a D/I/R mechanism. In particular, the requirements for modeling each were developed in sections 2.1.9, 10 and 11 respectively, their relation to a D/I/R mechanism presented in section 2.1.12, and the general method utilized to specify them described in section 2.1.13.

As a final note, however, it should be recognized first, that given both a particular detector and set of environmental conditions, there exists a singular set of process representations which describe the corresponding detection/isolation/recovery sequence and second, that the set consists of four representations, one each describing the detection (D), isolation (I), error-propagation-recovery (E) and time-lost-recovery (T) facets of the corresponding mechanism. The latter subdivision (of recovery into components I and E) is a direct consequence of the rationale presented on page 2-10, second and third items respectively.

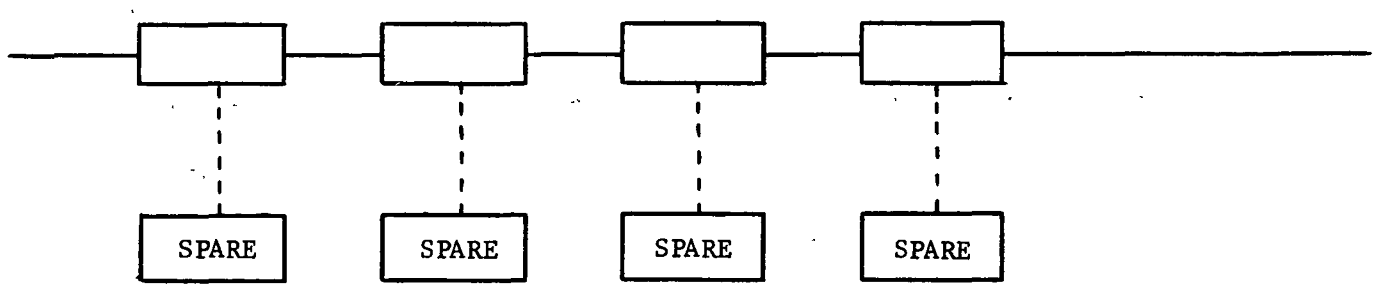
2.2 SAMPLE FUNCTION SELECTIONS

Although the objective of this study is to develop a relatively general-purpose system reliability prediction tool, an underlying goal is its application to a specific fault tolerant computer configuration; a dual channel system consisting of input, CPU, memory and output LRU's, plus an external comparator (cf. figure 2-1).

To this end, D/I/R mechanism definitions and performance characterizations have been made for a set of fault detectors, each member of which has potential application in the subject system. Table 2-1 summarizes these, wherein the reader's leftmost column is a list of the detectors (and equivalently, the D/I/R mechanisms). The double row to the right of each detector defines the mechanism as it relates to varying environmental conditions, the latter consisting of combinations of fault subclass (Input vs. CPU etc., across the top heading row), fault type (Permanent vs. Transient, across the second heading row), and system operating mode (2 channel vs. 1 channel, down the second column from the left). In turn, each rectangle (representing a single mechanisms performance under specific conditions) contains either a zero (representing a null



MODE 1 CONFIGURATION (FULLY OPERATIONAL)



MODE 2 CONFIGURATION (DEGENERATE)

FIGURE 2-1
DUAL CHANNEL SYSTEM
RELIABILITY DIAGRAM

response) or a series of four numbers. In the latter case, the four correspond to detection (D), isolation (I), error-propagation-recovery (E) and time-lost-recovery (T) function numbers, respectively, as indicated by the third heading row.

The function numbers referred in Table 2-1 are defined, in Table 5-4, in terms not yet presented. The rationale leading to their specification and selection, however, is discussed at length in the remainder of this section. Prior to initiating this, though, the following points should be noted:

- individual detectors have been purposely interpreted, in terms of their scope, in a somewhat restrictive fashion. For example, software selftest can be viewed as including some form of memory addressing scheme check; in terms of this study, however, each has been interpreted to be a separate and distinct test, and categorized accordingly.

		INPUT PERM. DIET	INPUT TRANS. DIET	CPU PERM. DIET	CPU TRANS. DIET	MEMORY BIT PLANE PERM. DIET	MEMORY BIT PLANE TRANS. DIET	MEMORY WORD SELECT PERM. DIET	MEMORY WORD SELECT TRANS. DIET	OUTPUT PERM. DIET	OUTPUT TRANS. DIET
MEMORY CODE	2 ch.	0	0	0	0	1 1 1 1	1 1 1 1	0	0	0	0
	1 ch.	0	0	0	0	1 1 1 2	1 1 1 2	0	0	0	0
OUTPUT COMPARE/VOTE	2 ch.	9 2 1 2	9 3 1 2	9 2 1 2	9 3 1 2	9 2 1 2	9 3 1 2	9 2 1 2	9 3 1 2	0	0
	1 ch.	0	0	0	0	0	0	0	0	0	0
I/O WRAP-AROUND	2 ch.	11 1 1 1	0	12 2 1 1	0	12 2 1 1	0	13 2 1 1	0	11 1 1 1	0
	1 ch.	11 1 2 2	0	12 2 2 2	0	12 2 2 2	0	13 2 2 2	0	11 1 2 2	0
CPU SOFTWARE SELFTEST	2 ch.	0	0	10 1 1 1	0	14 2 1 1	0	15 2 1 1	0	0	0
	1 ch.	0	0	10 1 2 2	0	14 2 2 2	0	15 2 2 2	0	0	0
REASONABLENESS TESTS	2 ch.	2 1 1 1	2 1 1 1	16 2 1 1	16 3 1 1	16 2 1 1	16 3 1 1	16 2 1 1	16 3 1 1	0	0
	1 ch.	2 1 1 2	2 1 1 2	16 2 1 2	16 3 1 2	16 2 1 2	16 3 1 2	16 2 1 2	16 3 1 2	0	0
TIMEOUT COUNTERS	2 ch.	3 1 1 1	3 1 1 1	17 2 1 1	17 3 1 1	0	0	3 1 1 1	3 1 1 1	3 1 1 1	3 1 1 1
	1 ch.	3 1 1 2	3 1 1 2	17 2 2 2	17 3 2 2	0	0	3 1 1 2	3 1 1 2	3 1 1 2	3 1 1 2
CPU CODE	2 ch.	0	0	2 1 1 1	2 1 1 1	0	0	0	0	0	0
	1 ch.	0	0	2 1 1 2	2 1 1 2	0	0	0	0	0	0
MEMORY SUMCHECK	2 ch.	0	0	4 2 1 1	0	18 1 1 1	18 3 1 1	19 1 1 1	0	0	0
	1 ch.	0	0	4 2 2 2	0	18 1 2 2	18 3 2 2	19 1 2 2	0	0	0
INVALID INST	2 ch.	0	0	5 1 1 1	5 3 1 1	5 1 1 1	5 3 1 1	5 1 1 1	5 3 1 1	0	0
	1 ch.	0	0	5 1 1 2	5 3 1 2	5 1 1 2	5 3 1 2	5 1 1 2	5 3 1 2	0	0
READ/WRITE PROTECT	2 ch.	0	0	5 1 1 1	5 3 1 1	5 1 1 1	5 3 1 1	5 1 1 1	5 3 1 1	0	0
	1 ch.	0	0	5 1 1 2	5 3 1 2	5 1 1 2	5 3 1 2	5 1 1 2	5 3 1 2	0	0
ADDRESS FEEDBACK	2 ch.	0	0	0	0	0	0	6 1 1 1	6 1 1 1	0	0
	1 ch.	0	0	0	0	0	0	6 1 1 2	6 1 1 2	0	0
ADDRESSING SCHEME CHECK	2 ch.	0	0	7 2 1 1	0	7 2 1 1	0	8 1 1 1	0	0	0
	1 ch.	0	0	7 2 2 2	0	7 2 2 2	0	8 1 2 2	0	0	0

TABLE 2-1
SAMPLE COVERAGE MODEL SELECTIONS

- certain detectors, in the process of searching out faults in a particular unit, have an inherent capacity to detect faults in others as well. This capacity, though perhaps beyond the intended scope of the test is, in many cases, significant enough to warrant inclusion in the model set.
- in many instances, the ability of a detector to find faults in a particular unit, though possibly non-zero, is of insufficient magnitude to warrant consideration in the study.
- the time at which a fault actually occurs is viewed as immaterial and, in its place, emphasis is placed on the time at which it can first affect other processes.

2.2.1 Detection Function Rationale

The following subsections describe the reasoning behind the detection function selections shown in Table 2-1:

2.2.1.1 Memory Code

Any such code, for example, Hamming or simple parity, offers the same form albeit not the same degree of protection. This can be characterized by noting that:

- in a typical implementation, only memory bit plane (i.e., data) faults are protected
- the degree of protection offered is independent of the quantity of channels operating and independent of whether the fault is permanent or transient
- to the extent that a given faulty bit pattern is covered, the fault will be detected on reading the cell and, thus,

given a reasonable computer and software design, mission computation will cease prior to any consequent incorrect processing.

The latter gives rise to a belief that any such code can be modeled via an impulse. Further, since error propagation is precluded, the effective delay time between fault occurrence and fault detection is zero.

2.2.1.2 Output Compare/Vote

The technique requires the presence of at least two operating and fully synchronized channels and is, therefore, inoperative in all one-channel situations. Given two or more, it is capable of detecting all faults save those in the output unit, on their passage from channel to comparator (cf. Figure 2-1).

On the assumption that the comparator/voter receives all digital output commands, but no others, its detection rate can be characterized as a decaying exponential training off eventually to zero. (The presumption here is that all meaningful faults ultimately result in the output of erroneous values, and that the probability as to when is highest immediately following fault occurrence and ever diminishing thereafter.)

Since detection is presumed to occur on output, external, but not internal, faults are precluded from propagating.

2.2.1.3 I/O Wraparound

Test performance is independent of the quantity of channels operating. In effect, only permanent faults are detected in that transients can be observed only in the situation where they occur during performance of the test case. Such faults affect only test data.

The test, by design, checks out portions of the input and output units; indirectly, however, it affords some detection of faults in the remaining units by virtue of its dependence on them for operation of its software-based implementation.

The detection rate can be characterized as a pulse (i.e., a constant detection rate over its operating interval) in that the test consists primarily of the sequential checkout of multiple channels, each with equal likelihood of having failed. Because it is a periodic (i.e., scheduled) test, it affords little protection against fault propagation. Similarly, the delay time between fault occurrence and fault detection is dependent on when the fault occurs with respect to the scheduled test time.

2.2.1.4 CPU Software Selftest

The expression software selftest often implies a medley of assorted tests. For purposes of clarity, all but one of these, CPU test, have been broken out and categorized separately.

The residual test is seen as offering no protection against meaningful transient faults and no protection against input or output faults. By design, it detects the majority of CPU faults, with the rate function seen as a truncated decaying exponential. In this sense, the cutoff portion represents those faults which are not detected, and the non-zero duration of the truncated exponential corresponds to the run time of the test. The decaying exponential form represents a belief that the majority of faults (for reason both of test design and hardware requirements to run the test) are detected near the beginning of the test and that, from there on, it is a fight against ever-diminishing returns up through test end.

The test, by nature of its being a scheduled event, precludes little in the way of fault propagation and, for the same reason, varies in its detect time as a function of the relationship between fault occurrence time and test run time.

Since the test operates out of memory, it offers a degree of fault detection with respect to memory-based faults.

2.2.1.5 Reasonableness Tests

The tests are interpreted to be a series of short duration checks, primarily on the validity of recently calculated data, which are performed throughout the epoch cycle. As such, their performance is unaffected by the quantity of channels operating, as well as by whether the fault is permanent or transient. By reason of inaccessability to internal processes, output faults cannot be detected.

The use of a zero delay impulse model to characterize its detection rate function for input faults, is based on the inherent capacity of the test to preclude certain faults from entering the system. This in turn is seen as equivalent to a zero detection time.

On the other hand, CPU and memory are provided with unscheduled pulse rate functions, as warranted by their aperiodic characteristics (with respect to fault occurrence) and the consequent variant time interval prior to detection.

2.2.1.6 Timeout Counters

Such counters can be incorporated in any or all channel subunits, though incorporation in memory is viewed as offering word select protection only. In all cases, other than CPU, the tests can be adequately characterized by means of an unscheduled zero-delay impulse, due to their essentially instantaneous service. The CPU counter, on the other hand, is presumably a watchdog timer and, thus, has a delay time measured in milliseconds. Since the time of detection is independent of when the fault occurs (within the timeout interval), the test appears as an unscheduled pulse. Consequently, little protection against fault propagation is offered.

All forms of the test are relatively insensitive to whether the fault is permanent or transient, and totally insensitive to the quantity of operating channels.

2.2.1.7 CPU Codes

Though other forms are possible, this class of test is typically implemented as a residue or product code applied to the ALU, data path and register portions of a CPU. As such, it offers singular protection against a large subclass of CPU faults, and does so independent of whether the fault is permanent or transient. The quantity of computers operating has no impact.

Since faults are detected on occurrence, all forms of fault propagation are precluded, and, thus, the proper model is an unscheduled impulse of zero delay.

(Note that the existence of fault partitions within the CPU (from the view of this particular test) suggests rather strongly that the CPU might better have been treated as two subclasses for coverage computation purposes (i.e., one containing the ALU and the other, the control unit.))

2.2.1.8 Memory Sumcheck

The technique, which consists of occasional exclusive-or sumchecks on the contents of invariant subportions of memory, offers a significant degree of memory bit plane and word select fault detection. In the case of bit plane faults, the protection afforded is unaffected by whether the perturbation was transient or permanent, whereas in the word select case, only permanent faults (in any meaningful sense) are detected. In both cases, the quantity of computers operating has no impact.

Since the test, by virtue of its rather excessive run time, is typically partitioned into a series of N subtests, each operating on a portion of memory, it can be characterized by a string of N impulses, typically extending beyond a single epoch cycle. Due to the scheduled nature of the test, detection time is aperiodic with respect to fault occurrence, and little protection is offered against either form of fault propagation.

As a consequence of the test being implemented in software, a small degree of protection is offered against CPU faults.

2.2.1.9 Invalid Instruction

This test, in many respects, serves more as an indidicator of software errors than hardware faults. For completeness it is, however, included.

To the degree that faults are in fact detected, the protection is independent of both the quantity of operating computers, and whether the fault is permanent or transient. No protection against input and output faults is offered. The detection of CPU and memory faults, though severely limited in scope, is instantaneous and independent of the time of fault occurrence.

2.2.1.10 Read/Write Protect

The test, assumed to be implemented in hardware, serves primarily as a software error detector. It does, however, have limited hardware fault detection capabilities, and is, therefore, included.

No protection is offered in the case of input and output faults. CPU and memory faults are in select cases detectable and, to this extent, the protection is independent of the quantity of computers operating and also independent of whether the fault is permanent or transient.

Detection, when it occurs, is instantaneous and, thus, the appropriate model is the zero delay impulse.

The following, severely limited, classes of faults are detectable by this technique:

- CPU - limited instances of incorrect memory address computation
- Memory bit plane - situations where the address portion of an instruction stored in memory is, on access, incorrect to the extent of crossing a protection boundary
- Memory word select - instances where an instruction fetch is performed on an incorrect memory cell, which itself contains an out-of-range address.

In these situations, fault propagation is precluded.

2.2.1.11 Address Feedback

The technique, though seldom discussed, offers a significant degree of singular protection against memory word select faults. Implementation techniques vary, with one of the more straightforward methods consisting of the encoding of address and data, rather than merely data, on storage of a word into memory. Given then a reasonably strong code, for example, six-bit Hamming, a high percentage of addressing faults can be detected on readout via the decoding process.

Protection is instantaneous, and independent of fault occurrence time, quantity of operating computers, and whether the fault is permanent or transient. Fault propagation is precluded.

2.2.1.12 Addressing Scheme Check

Given either a 2½D or 3D memory addressing scheme (preferably the latter), this technique offers a reasonable form of memory word select verification. Its methodology, which is based on the use of coincidence for word selection, enables the bulk of the memory addressing circuitry to be verified (the exception being that fraction which is unique to the variable portion of storage).

The technique, which relies on the exclusive or sumcheck testing of a specific small subset of memory locations, detects addressing faults by virtue of the implication that correct sumcheck signifies correct addressing. The test is implemented in software and, thus, also offers a small degree of CPU and memory bit plane checkout. The latter is supplanted, to a minor extent, via the sumcheck process itself which verifies the contents of the memory subset.

Because of its software basis, the test is necessarily periodic and, therefore, incapable of precluding fault propagation. Similarly, the time required to detect is dependent on the relationship between fault occurrence time and test run time. The test is not dependent on the quantity of computers running. Little or no practical protection is offered against transient faults.

2.2.2 Isolation Function Rationale

Three functions have been selected to represent the isolation process for this particular implementation of the four LRU, dual-channel system (cf. Table 5-4).

2.2.2.1 Primary Isolation Processes

This isolator, represented as a slightly delayed impulse with unity coefficient, typifies the situation in which the fault location implied by the initiating detector was, in fact, the actual location. For example, in the case of CPU selftest, the implied location is the CPU itself and, as a consequence, the first area searched in the isolation process.

Given that the implication is correct, the corresponding determination of this fact is made after a short delay (as measured from the end of the detection process) which itself represents the duration of the isolation function.

2.2.2.2 Secondary Isolation Processes

In the situation where the fault location lies other than in the detector implied LRU, the search is both misguided (initially) and, of necessity, quite extensive in domain. Correspondingly, it is best modeled as a pulse of long duration which, in turn, represents a process with constant isolation probability throughout the total interval.

2.2.2.3 Tertiary Isolation Processes

The isolation process is most complex, and least likely prone to success, in situations where either a transient fault has occurred, or where the fault is totally unlocalized (as with detection by a comparator external to the channel). Consequently, an impulse model, with long initial delay, has been selected as reasonably representative.

2.2.3 Fault Propagation Recovery Function Rationale

Two functions have been selected to represent the fault propagation aspect of the recovery process. The first of these is a pulse of long duration, and represents a situation in which the recovery probability is undiminished, from this effect, up through the end point of the pulse. It corresponds to either of two situations:

- the associated fault detector precludes any appreciable form of fault propagation
- fault propagation is of little consequence, due to the presence of an operational second channel which can be utilized for update purposes

The second function corresponds to the opposite situation, in which fault propagation is of considerable potential effect, and the function is, thus, provided in the form of a decaying exponential, whose diminishment with time represents a like decrease in recovery probability. It is utilized primarily in the combined situation of degenerate system operation and the presence of fault propagation.

2.2.4 Time Lost Recovery Function Rationale

The time lost aspect of fault recovery contends with the effect, on external-to-the-system operational requirements, of channel loss during a "fault-present" interval. In the situation where both channels were operative prior to the fault, one channel normally continues servicing external needs, and, thus, the overall effect is minimal. A long duration pulse is applied in this case.

In the situation either of previous degenerate operation or fault detection by the comparator, however, the total system is temporarily out of service, and, thus, no external processing requirements are met. (Detection by the comparator is assumed to require that both channels cease normal operation and enter into the isolation phase concurrently.) The function selected to represent this situation is a decaying exponential with cutoff in the region of mission critical time.

2.3 TERMINOLOGY DEFINITIONS

Batch Run - The total data processing task performed at one time. It may consist of one or more run-sets.

Category 1 Switch Failure - A failure, typically of spares-switching-hardware, such as to prevent proper operation of a single unit. This possibility must be accounted for in estimating the failure rate of the unit.

Category 2 Switch Failure - A failure, typically of spares-switching-hardware, such as to force degeneration of the system from mode M to mode M+1.

Category 3 Switch Failure - A failure, typically of comparator or voter hardware, such as to disable the entire system.

Channel - The minimum complement of operational hardware required to perform the system computational task.

Coverage - The conditional probability, given both an on-line unit failure and sufficient spare hardware to maintain computational processes, that the failure is detected and operation is successfully re-established, consistent with application imposed time constraints.

Coverage Model - A set of equations for evaluating the ability of a system to recover from permanent or transient hardware faults in a spares-switching environment. Also, the implementation of this model in CARE2.

Degenerative Failure - Any condition which necessitates reconfiguration of the system, from Mode M to Mode M+1, in order to regain full processing capabilities.

D/I/R Mechanism - (Detection/Isolation/Recovery Mechanism) - A coverage model representation of the conditional performance characteristics of all detection, isolation and recovery processes associated with a particular fault detector. In turn, the conditions are comprised of valid combinations of fault subclass, mode, fault type and potential requirement for mode degeneration.

Dual Mode Reliability Model - A set of equations for evaluating the reliability of a system, given that the system has no more than two distinct and relevant operating modes. Also, the implementation of this model in CARE2.

Fault - An event which perturbs computational processes within the computer system and, as a consequence, requires remedial action, in the form of detection, isolation and recovery, in order to preclude an operational anomaly.

Fault Class - The total set of possible faults associated with a single stage of a computer system.

Fault Detector - A hardware or software process whose function is to recognize the occurrence of certain classes or subclasses of faults within a system.

Fault Isolator - A hardware or software process whose function is to isolate a detected fault to the responsible LRU.

Fault Recovery Process - A hardware or software process whose function is to minimize or eliminate the potentially harmful effects of a fault, given that it is properly isolated, by means of appropriate remedial actions.

Fault Subclass - A fractional portion of a fault class, distinguished from other subclasses within the stage by either its unique susceptibility, or lack thereof, to recognition and treatment by one or more fault detection, isolation and/or recovery processes.

<u>FTF</u> - (Fault Tolerant Feature) - A hardware or software based process included in the system for the purpose of detecting, isolating or recovering from certain classes or subclasses of faults.

<u>Hybrid Channel</u> - A fault tolerant computer system in which the component stages contain unequal quantities of on-line LRU's, thereby precluding division of the system into discrete channels.

<u>LRU</u> - (Least Replaceable Unit) - The smallest system component which may be switchably replaced by an equivalent spare, in the event of its failure.

<u>Major Cycle</u> - The time interval between repetitions of the total set of scheduled detectors (i.e., the time after which the total set is repeated).
Note: In CARE2, this value is calculated internally.

<u>Minor Cycle</u> - The time interval between repetitions of the most frequent scheduled detector.
Note: In CARE2, all scheduled detector periods must be integer multiples of the minor cycle.

<u>Mode</u> - One of the possible hardware configurations in which the system can operate. In general, mode M requires more on-line units than mode M+1, and is the preferred operating configuration.

<u>Operational Anomaly</u> - The consequence of a fault which for reason of non-detection, incorrect isolation or unsuccessful recovery, is not remedied and therefore manifests itself in the issuance of improper control signals from the computer system.

<u>Parameter</u> - A variable, either defaulted, input, or precalculated, which is meaningful in either the reliability or coverage model description.
Note: In CARE2, certain parameters, such as the inverse dormancy factor K, cannot be directly input by the user.

Parameter Vector - A one dimensional Fortran array which contains the values of a single parameter for all stages in the system.

Permanent Fault - A fault which, because of its permanent nature, requires either the use of a spare LRU or degeneration in the process of recovering from its effects.

Run - The process of computing system reliability, versus time, for a given list of equation numbers (models) and a fixed set of model parameters.

Run-set - One or more runs, with possible variation of one reliability parameter type (for all stages), through the use of array PARAM. (Coverage parameters may not be varied in this way, for reason of the data base format structure.) Data base management operations take place between run-sets.

Scheduled Detector - A fault detector, usually software, which is initiated at regular periodic intervals.

Spare LRU - A single LRU contained in the spares pool. Equivalent to the term 'standby spare'.

Spares Checkout - The process of testing a standby spare, for proper operation, during the interval between occurrence of a permanent fault and commitment of the particular spare as the replacement.

Spares Pool - The total set of standby LRU's available for substitution in the event of permanent failure in one of the on-line LRU's. Note that substitution is conditioned by a requirement that both the failed LRU and its replacement be contained in the same stage.

Spares Reassignment - The process of releasing the operational LRU's in a defunct channel to the spares pool. Given that the specific computer system design allows for reassignment, the process takes place at the time of degeneration from Mode M to Mode M+1.

Stage - The total set of identical LRU's, including both on-line and standby devices, contained in the system (cf. page 4-2).

Standby Spare - A single LRU contained in the spares pool. Equivalent to the term 'spare LRU'.

System - The full complement of hardware and software comprising a fault tolerant computer configuration. In CARE2, the complex of coverage and hardware reliability data associated with a single execution run defines the current system.

TMR - (Triple Modular Redundancy)- A fault tolerant computer system consisting, initially, of three parallel and active channels in series with a majority voting element. In the event of a singular channel failure, the voter maintains correct system outputs.

Transient Fault - A fault which, because of its temporary nature, requires neither the use of a spare LRU nor degeneration in the process of recovering from its effects.

Transition - The process of switching the computer system operating mode (from Mode M to Mode M+1).

Unit Time - An arbitrary time interval, as for example seconds or hours, used in conjunction with failure rates, detection rates, reliability, etc.. Although consistent units must be utilized within a model (i.e., Reliability or Coverage), there is no need to use common units in separate models.

Unscheduled Detector - A fault detector, usually hardware, in which the detection process is triggered either directly, or effectively in outward appearance, by the occurrence of a fault.

Page Intentionally Left Blank

SECTION 3

MATHEMATICAL MODELS

The total system reliability of a redundant computer configuration can be expressed mathematically in a number of ways. These expressions all tend to be rather cumbersome, however, and therefore one of the objectives of this study was to devise formulations that can be evaluated efficiently by computer. A second, and equally important goal, was that the formulations be of a sufficiently general nature as to provide considerable design flexibility to the user. The following sections present both the end result of these efforts and, as an interpretation aid, a listing and definition of all parameters.

3.1 DUAL MODE RELIABILITY MODEL

The Dual Mode Reliability Model subprogram, with respect to initial plans, has been extended in scope significantly such that it now includes not only dual channel capability, but a variety of others as well. These include both N channel (where N is a user specified integer) and Hybrid configuration (where the quantity of active units differs between stages).

The reliability equations utilized in encoding the model, as well as a functional description of each, are presented in this section. Subscripts and other simple variables used therein are as follows:

- x - Refers to the set of LRU's which constitute a particular stage of the reliability model
- y - Refers to the operating mode
- t - Either the independent time variable or a dummy time variable (made clear by context)
- τ - A dummy time variable for integration or function referencing
- i, j, ℓ - Dummy integer variables for counting or function referencing.

3.1.1 G: Fortran DCG(IUN,MD,I,T) - "The probability of using exactly i spares by time t."

$$G(x,y,i,t) = \begin{cases} e^{-(\gamma_{x,y})t} * (\delta_{x,y})^{i} * \binom{(K_x)(Q_{x,y})(C_{x,y})(\delta_{x,y})^{-1} + i - 1}{i} * \\ \left(1-e^{-(\mu_x)t}\right)^{i} * e^{-(Q_{x,y})(\lambda_x)t} \\ \text{for } K < \infty \\ e^{-(\gamma_{x,y})t} * \dfrac{\left((Q_{x,y})(C_{x,y})(\lambda_x)t\right)^{i}}{i!} * e^{-(Q_{x,y})(\lambda_x)t}, \\ \text{for } K = \infty \end{cases}$$

This function expresses the probability of using exactly i of the available spares by time t under the following assumptions:

- $Q_{x,y}$ LRU's are required, with standby and active LRU's having the constant failure rates μ_x and $\lambda_x = K_x \mu_x$, respectively.
- The coverage factor, given that the ℓth spare tested following a detected failure is the first operational spare, is $C_{x,y}\,\delta_{x,y}^{\ell-1}$. (Note: coverage is defined here as the conditional probability of successful recovery given that an error occurs and sufficient hardware is available. If recovery is possible, the system operational mode is determined solely by the remaining available hardware.)
- Non-recoverable transients occur at the rate $\gamma_{x,y} = \gamma'_{x,y} * (1-P_{r_{x,y}})$ per second. In the case of recoverable transients, the system resumes operation in the same mode as was in effect prior to the transient.

The first term in the above expression, then, is the probability that no non-recoverable transients occur by time t. The product of the remaining terms is the probability that exactly i LRU's sustain failures by time t, either while operating or prior to the time they are needed. (Notice that a failure in a spare LRU is not counted here until that spare would have been used following a failure in an operating LRU.) To verify this, we observe that the reliability of an r-on-m configuration (r spares on m operating elements) having active and standby failure rates λ and μ, respectively, is equivalent to that of an r on $\lambda m/\mu$ configuration having a constant failure rate μ for both active and standby elements.* Further, since with probability $C_{x,y}/\delta_{x,y}$, the system can recover from a failure on the condition that it can also find and successfully activate a standby spare, the rate of such recovered failures is $\lambda_x C_{x,y}/\delta_{x,y}$, and hence the effective number of active elements is $K_x Q_{x,y} C_{x,y}/\delta_{x,y} \triangleq M$. The probability that exactly i spares are used by time t is thus the product of the probability

$$\binom{M+i-1}{i}(1-e^{-\mu_x t})^i e^{-(M-1)\mu_x t}$$

that exactly i recoverable failures occur in the first M+i-1 LRU's, the probability $e^{-\mu_x t}$ that the i^{th} spare is operational, the probability $\delta^i_{x,y}$ that i spares are successfully tested, and the probability

$$\exp\left[-Q_{x,y}\left(1-C_{x,y}/\delta_{x,y}\right)\lambda_{x,y}t\right]$$ that no non-recoverable failures occur during this period.

* C.F. J. J. Stiffler, "On the Efficacy of R-on-M Redundancy", IEEE Trans. on Reliability, (to appear).

3.1.2 R_u: Fortran DCRU(IUN,MD,L,TAU) - "The probability of using at most ℓ spares by time τ."

$$R_u(x,y,\ell,\tau) = \sum_{i=0}^{\ell} G(x,y,i,\tau)$$

The probability of using at most ℓ spares is clearly the sum of the mutually exclusive probabilities that exactly i spares are used, for i = 0,1,..., ℓ.

3.1.3 H: Fortran DCH(IUN,TAU) - "The probability density of a degenerative failure in stage x at time τ."

$$H(x,\tau) = \begin{cases} Q_{x,1}\,\lambda_x \sum_{i=0}^{(S_x)} \left\{ G(x, y{=}1, \left[(S_x)-i\right], \tau) * (C'_x)(\delta'_x)^i * (1-e^{-(\mu_x)\tau})^i \right\}, \\ \quad \text{for } K<\infty \\ \lambda_x Q_{x,1} (C'_x) * G(x, y{=}1, (S_x), \tau), \\ \quad \text{for } K=\infty \end{cases}$$

The conditional probability density of a degenerative failure at time τ, given exactly S_x-i prior failures, is equal to the product of the probability density $Q_{x,1}\lambda_x$ of a failure in an operational unit and the probability $(1-e^{-\mu_x\tau})^i$ that the i remaining spares have failed by time τ, times the probability $C'_x(\delta'_x)^i$ that it is possible to recover from this event. The product of this conditional probability density with the probability that exactly S_x-i failures have occurred by time τ, summed over all i = 0,1,..., S_x, is thus the desired probability density.

3.1.4 S: Fortran DCS(IUN,T,TAU) - "The conditional probability that the set of units in stage x can remain operational in mode y = 2, from time τ to time t, given that the mode changed from 1 to 2 at exactly time τ."

$$S(x,t,\tau) = \begin{cases} \sum_{\ell=0}^{(S_x)} \left[\sum_{j=0}^{(S_x-\ell)} \left\{ \binom{(S_x)-\ell}{j} * (1-e^{-(\mu_x)\tau})^{((S_x)-\ell-j)} * e^{-j(\mu_x)\tau} * G(x,y=1,\ell,\tau) * R_u(x,y=2,j',(t-\tau)) \right\} \right] & \text{for } K<\infty \\ \sum_{\ell=0}^{(S_x)} \left[G(x,y=1,\ell,\tau) * R_u\left(x,y=2,\left[(S_x)-\ell\right]',(t-\tau)\right) \right], & \text{for } K=\infty \end{cases}$$

$$\text{where } j' \equiv \begin{cases} j, & \text{no spares reassignment} \\ j+Q_{x,y=1}-Q_{x,y=2}, & \text{reassignment allowed} \end{cases}$$

$$\left[(S_x)-\ell\right]' \equiv \begin{cases} \left[(S_x)-\ell\right], & \text{no spares reassignment} \\ \left[(S_x)-\ell\right]+Q_{x,1}-Q_{x,2}, & \text{reassignment allowed} \end{cases}$$

The product of the first three terms in the summand is the probability that exactly j of the $S_x - \ell$ unused spares are still operational at time τ. The fourth term is the probability that exactly ℓ spares have been used in mode 1 by time τ, and the fifth term the probability that the remaining j' spares are sufficient to keep the unit operating in mode 2 for the next $(t-\tau)$ seconds. The sum over all $\ell = 0, 1, \ldots, S_x$ and $j = 0, 1, \ldots S_x - \ell$ is thus the probability of the event described.

3.1.5 T: Fortran DCT(IUN,J) - "The probability that the system will survive until time t, and that a failure in the set of units in stage x will have forced degeneration from mode y = 1 to mode y = 2."

$$T(x,t) = \begin{cases} \int_0^t \left[\prod_{\bar{x} \neq x} \left\{ S(\bar{x}, t, \tau) \right\} * R_u(x, y=2, r, (t-\tau)) * H(x,\tau) * e^{-(\hat{\lambda}_2)\tau} \right] d\tau, & \text{for } Q(x,1) > Q(x,2) \\ 0, & \text{for } Q(x,1) \leq Q(x,2) \end{cases}$$

The integrand is the product of the probability density of a failure in unit x at time τ (3rd term), the probability that the unit functions successfully for the next $t-\tau$ seconds in mode 2 (2nd term), the probability that all other units remain in operation until time τ in mode 1 and from time τ to time t in mode 2 (1st term), and the probability that no category-two failures occur during this time (4th term). Since a degenerative failure can occur any time during the interval $0<\tau<t$, T(x,t) is the integral of this density function over that interval.

3.1.6 T_2: Fortran DCT2(J) - "The probability that the system will survive until time t, and that a category 2 switch failure will have forced degeneration from mode y = 1 to mode y = 2."

$$T_2(t) = (\hat{\lambda}_2)(C_2)\int_0^t \left[\left\{ S(x,t,\tau) \right\} * e^{-(\hat{\lambda}_2)\tau} \right] d\tau$$

Here the 1st term in the integral is the probability that all units function successfully in mode 1 until time τ and in mode 2 from time τ to time t. The product $\hat{\lambda}_2 C_2 e^{-\hat{\lambda}_2 \tau}$ is the probability density of a category-two failure at time τ times the conditional probability C_2 that recovery from such a failure is successful. The integral of this quantity over all τ, $0<\tau<t$, is therefore the result sought.

3.1.7 R: Fortran DCR(J,UNITR, RELM1, RSYS) - "The system reliability at time t."

$$R(t) = \left[\prod_x \left\{ R_u(x, y=1, S_x, t) \right\} * e^{-(\hat{\lambda}_2)t} + T_2(t) + \sum_x \left\{ T(x,t) \right\} \right] e^{-(\hat{\lambda}_3)t}$$

The first term here is the product of the probability that each of the units function until time t in mode 1 and that no category-two failures occur during that time. The remaining two terms represent the probabilities that the system successfully survives until time t, with a degeneration occuring sometime during that interval because of a category-two switch failure, and because of a degenerative failure in one of the units, respectively. The sum of these three terms multiplied by the probability that no category-three failures occur during the period in question is thus the probability that the system operates successfully over the entire t-second interval.

The function also computes system and stage reliabilities, assuming mode 1 operation alone, and returns the values as RELM1 and array UNITR, respectively. The expressions for these values are subsets of the above equation:

$$R(t,y=1) = \left[\prod_{x} R_u(x,y=1,S_x,t) \right] e^{-(\hat{\lambda}_2+\hat{\lambda}_3)t}$$

$$R_x(t,y=1) = R_u(x,y=1,S_x,t)$$

3.2 COVERAGE MODEL

The function subprogram COVAGE, in concert with certain other routines, is utilized to calculate coverage for application in both the dual mode reliability model and original CARE equations 2 and 3. This section presents the equations utilized therein, as well as corresponding functional descriptions.

The basic coverage calculation described herein returns a single value C(s) corresponding to a specific system operating mode, fault type (permanent or transient), quantity of spares tested, and fault subclass. The calculative process is then iterated, via separate calls to COVAGE, for each combinatorial set of conditions.

Since the ensuing reliability model accepts coverage values at the stage rather than subclass level, it is necessary to preprocess the above according to the relation

$$C(I) = \sum_{\sigma, L_\sigma = I} (d_\sigma C_\sigma)$$

where C(I) is a coverage factor

of stage I*, and is computed as an average of the C_σ factors returned, as C(s), by COVAGE (cf. section 3.2.2), weighted by the fractional fault occurrence rate d_σ for each fault subclass σ associated with the stage. The association of stage and fault subclass is specified by the user when selecting input values for the linkage variable L_σ (IFSC), and the relative fault rate d_σ (FRAC), for each subclass used.

In turn, the required delta coverage inputs (representing the diminished recovery probability arising from trial repair with a previously failed spare), are computed in accordance with the expression

$$\delta(I) = C(I,s=2)/C(I,s=1)$$

where s is the quantity of spare LRU's which must be trial tested during the recovery process (i.e., s-1 having failed the test).

In perspective then, it should be noted that the following coverage and delta coverage values are computed, and delivered to the reliability model, for each stage of the object computer system:

- Coverage - C(I)
  - $C_{x,y=1}$ - permanent fault, continued Mode 1 operation
  - $C_{x,y=2}$ - permanent fault, continued Mode 2 operation
  - C'_x - permanent fault, transitional from Mode 1 to 2
  - $P_{r_{x,y=1}}$ - transient fault, continued Mode 1 operation
  - $P_{r_{x,y=2}}$ - transient fault, continued Mode 2 operation
- Delta Coverage - δ(I)
  - $\delta_{x,y=1}$ - permanent fault, continued Mode 1 operation
  - $\delta_{x,y=2}$ - permanent fault, continued Mode 2 operation
  - δ'_x - permanent fault, transitional from Mode 1 to 2

* Note that, in the reliability model, x was used to denote a stage.

3.2.1 Subscripts and Simple Variables

$t, \tau, \alpha, \nu, \eta$ - dummy variables for integration, summation and substitution.

i, j, k, ℓ - detector number. The choice of variable is determined as:

- i - general-purpose detector symbol (used on either side of an equation)
- j - scheduled detector symbol (right side of equation only)
- k - non-scheduled detector symbol (right side of equation only)
- ℓ - non-scheduled impulse detector symbol (right side of equation only). Specifically, $\ell_1, \ell_2, \ldots, \ell_{m_i}$ comprises the set of non-scheduled impulse detectors, excluding the i^{th}, which have the same delay (t_{d_i}) as de-detector i, given m_i such detectors.

3.2.2 C: Fortran COVAGE - "The sum of coverage contributions of all competing D/I/R mechanisms given that s spares must be checked prior to recovery."

$$C(s) = \sum_i C(i,s)$$

The coverage value C(s) associated with a single fault sub-class under given spares status and computer system operating conditions is clearly the summation of individual D/I/R mechanism contributions (C(i,s)) over all i.

3.2.3 C(i,s): Fortran COVAGE - "The coverage contribution of the D/I/R mechanism associated with detector i, when all competing detectors are accounted for and when s spares must be checked during the recovery process."

$$C(i,s) = p_i p'_i p_s^s \int_0^\infty g_i(\tau')r'_i(\tau') \int_0^\infty h_i(\tau''-s\tau_s)r''(\tau'+\tau'')d\tau''d\tau'$$

The detection probability density function for the i^{th} detector is $p_i g_i(\tau')$ and the associated isolation density function is, by definition, $p'_i h_i(\tau'')$. If s spares must be checked in order to recover successfully from a fault, the overall recovery probability is decreased by the factor p_s^s (with p_s the probability of successfully checking out a spare) and the isolation delay is effectively increased by the factor $s\tau_s$. The term C(i,s) is thus equal to the conditional probability that the system can still recover given a τ'-second detection delay times the detection probability density function, multiplied by the conditional probability that the system can recover given that it has survived a τ'-second detection delay and must in addition undergo a total of $(\tau'+\tau'')$ seconds down-time, times the corresponding isolation density function, the whole thing integrated over all τ' and τ'', and multiplied by p_s^s.

3.2.4 $g_i(\tau)$: Fortran CVGS, CVG2 and CVG1 - "The conditional detection rate of the i^{th} detector when in competition with all other detectors."

The function $g_i(\tau)$ represents the i^{th} detectors' conditional detection rate when competitive processes are taken into account. In turn, the corresponding detection probability density function is then expressed as $p_i g_i(\tau)$. The rate itself is determined by using one of three equations depending on the nature of the i^{th} detection process, i.e., whether it is scheduled or unscheduled and, in the latter case, whether it is a finite or impulse detector.

- Scheduled $g_i(\tau)$: Fortran CVGS

$$g_i(\tau) = \begin{cases} \prod_k \left[1-p_k F_k(\tau - t_{d_k})\right] g'_i(\tau), & \text{for } 0 \leq \tau \leq n_i T_{mr} \\ 0, & \text{otherwise} \end{cases}$$

The product over k represents the probability that none of the non-scheduled detectors has detected the fault by time τ (note that $F_k(t) = 0$ for all $t < 0$). The function $g_i'(\tau)$ is the conditional detection rate of the i^{th} scheduled detector when competing with the other operative scheduled detectors only (see Section 3.3.4).

- Finite Non-Scheduled $g_i(\tau)$: Fortran CVG2

$$g_i(\tau) = \prod_{k \neq i} \left[1-p_k F_k(\tau - t_{d_k})\right] * \left[1- \sum_j p_j \int_0^\tau g_j'(\eta)d\eta\right] * f_i(\tau - t_{d_i})$$

The product over k is the probability that none of the other non-scheduled detectors has sensed the fault by time τ. Similarly, the second term is the probability that none of the scheduled detectors has succeeded by time τ. The last term is, of course, the detection rate of the detector in question in the absence of any competition.

- Non-Scheduled Impulse $g_i(\tau)$: Fortran CVG1

$$g_i(\tau) = \left[1 - \sum_{\nu=1}^{m_i} \frac{(-1)^{\nu+1}}{\nu+1} \sum_{\ell_1<\ell_2<\ldots<\ell_\nu} p_{\ell_1} p_{\ell_2} \ldots p_{\ell_\nu}\right] *$$

$$\prod_{\substack{k \neq i \\ k \neq \ell_1, \ell_2, \ldots \ell_{m_i}}} \left[1 - p_k F_k (t_{d_i} - t_{d_k})\right] * \left[1 - \sum_j p_j \int_0^{t_{d_i}} g_j'(\eta)\, d\eta\right] * \mu_0(\tau - \tau_{d_i}),$$

for $f_i(\tau) = \mu_0(\tau - \tau_{d_i})$

where $\mu_0(x)$ is a unit impulse at $x = 0$

This expression has the same interpretation as the previous one except that the $f_i(\tau)$ is an impulse function. The first term assumes a different form here, reflecting the fact that if ν impulse detectors having simultaneous delays τ_d all detect the same fault, only one of them is declared the winner. In this event, it is assumed that each of the successful detectors has the probability $1/\nu$ of being declared the victor. The first term in this expression, then, is the conditional probability that the impulse detector in question is declared the winner over its simultaneously occuring impulse competitors, given that it is, in fact, successful in detecting the error. The second term is the probability that none of the other non-scheduled detectors finds the error prior to time τ_{d_i}. The remaining terms are as previously defined.

3.2.5 $g_i'(\tau)$: Fortran CVGP4 and CVGP3 - "The conditional detection rate of the i^{th} scheduled detector when in competition with all other scheduled detectors."

The function $g_i'(\tau)$ represents the i^{th} scheduled detectors' conditional detection rate when competitive scheduled processes only are taken into account. It is determined using either of two equations depending on the nature of the i^{th} detection process, i.e., whether it is a finite or impulse detector.

- Finite $g_i'(\tau)$: Fortran CVGP4

$$g'_i(\tau) = \begin{cases} \dfrac{1-F_i(\tau)}{n_i T_{mr}} + \dfrac{n_i}{n_c} \displaystyle\sum_{\ell=1}^{n_c/n_i} \dfrac{1}{n_i T_{mr}} \int_0^{\tau} \prod_{j\neq i} \left[1-p_j \bar{F}_j^{\ell}(\eta-\tau+n_i T_{mr}+t_{d_i}-t_j^{\ell})\right] * f_i(\eta)d\eta \\ \text{for } 0 \leq \tau \leq \Delta t_i \\ \dfrac{n_i}{n_c} \displaystyle\sum_{\ell=1}^{n_c/n_i} \dfrac{1}{n_i T_{mr}} \int_0^{\Delta t_i} \prod_{j\neq i} \left[1-p_j \bar{F}_j^{\ell}(\eta-\tau+n_i T_{mr}+t_{d_i}-t_j^{\ell})\right] * f_i(\eta)d\eta \\ \text{for } \Delta t_i < \tau \leq n_i T_{mr} \\ 0, \text{ otherwise} \end{cases}$$

The first term in the expression for $g_i'(\tau)$ when $0 \leq \tau \leq \Delta t_i$ represents the conditional detection rate of the i^{th} scheduled detector given that the error occurs sometime during the period during which that detector is active (e.g., while the associated diagnostic program is running).

Note: It is assumed here, and throughout this discussion, that the probability that an error is detected by a given detector during a given interval is equal to the integral of its probability density function over that interval. Furthermore, each portion of a scheduled test is assumed to have a probability of detecting a particular fault only during its first exposure to that fault. Thus, for example, if a fault occurs when a given scheduled program of T_0 seconds duration is in progress and has t_0 seconds left to run, and if the fault is not detected during those t_0 seconds, the fault has a chance of being detected only during the first T_0-t_0 seconds of that program when it is next run, and if it is not detected then, it will not be detected by that program during any subsequent runs.

The ℓ^{th} term in the summation in both expressions for $g_i'(\tau)$ (i.e., for $0 \leq \tau \leq \Delta t_i$ and for $\Delta t_i < \tau \leq n_i T_{mr}$) is the conditional detection rate of the i^{th} scheduled detector, given that the fault occurs either during or following the ℓ^{th} scheduling of that detector, but is not detected during that scheduled run. Accordingly, the product denotes the probability that none of the other scheduled detectors exposed to the fault during the τ seconds prior to its detection is successful in detecting it. This product, multiplied by the detection rate function $f_i(\eta)$ associated with the i^{th} detector and averaged over all η gives the desired result. Summing these conditional detection rates over a major cycle then yields the desired detection rate for the i^{th} scheduled detector when competing only with other scheduled detectors. Note that there are exactly n_c/n_i repetitions of the i^{th} detector during one major cycle. (A major cycle is defined as the overall period of the combined scheduled tests.)

- Impulse $g_i'(\tau)$: Fortran CVGP3

$$g'_i(\tau) = \begin{cases} \dfrac{n_i}{n_c} \displaystyle\sum_{\ell=1}^{n_c/n_i} \dfrac{1}{n_i T_{mr}} \prod_{j \neq i} \left[1 - p_j \bar{F}_j^{\ell} \left(-\tau + n_i T_{mr} + t_{d_i} - t_j^{\ell} \right) \right] \\ \text{for } 0 \leq \tau \leq n_i T_{mr} \text{ and } f_i(\tau) = \mu_0 t_{d_i} \\ 0, \text{ otherwise} \end{cases}$$

This expression is simply a specialization of the previous expression for the case in which $f_i(\tau)$ represents a scheduled impulse detector (i.e., $f_i(\tau)$ is an impulse function).

3.3 PARAMETER DEFINITIONS AND APPLICABILITY

The quantity of parameters contained in CARE2, with reference to the original CARE program, has of necessity increased rather significantly. In particular, the total set now consists of:

- CARE/CARE2 compatible parameters - Variables retained from the original CARE program for use in CARE2
- Dual Mode Reliability Model parameters - Variables which were introduced to allow implementation of Equation #7
- Coverage Model parameters- Variables which were introduced to allow calculation of coverage.

Table 3-1 lists the major parameters currently utilized in both the models and their Fortran implementation, and denotes their applicability with respect to both coverage and each of the seven basic reliability forms (i.e., equations 1-6 of original CARE and equation 7 as added in CARE2). In addition, it provides a categorization of each with respect either to its source or its use as a model output.

These same parameters are then defined, in Table 3-2, in the form a, A(N), alpha, where:

- a - is the symbol used in analytic expressions
- A(N) - is the Fortran mnemonic, with subscripts where applicable
- alpha - is the parameter name.

TABLE 3-1

PARAMETER/MODEL APPLICABILITY

Parameter	Coverage Model	Reliability Model/ Equation Number							
		1	2	3	4	5	6	7	
C	R	-	I*	I*	-	-	-	I*	
C'	R	-	-	-	-	-	-	I*	
C_2	-	-	-	-	-	-	-	I	
C_σ	C	-	-	-	-	-	-	-	Note 1
d_σ	I	-	I	I	-	-	-	I	Note 1
δ	R	-	-	-	-	-	-	I*	
δ'	R	-	-	-	-	-	-	I*	
$F_i(\eta)$	C	-	-	-	-	-	-	-	
$\bar{F}_i^\ell(\eta)$	C	-	-	-	-	-	-	-	
$F_j^\ell(\eta,i)$	C	-	-	-	-	-	-	-	
$f_i(\tau)$	I	-	-	-	-	-	-	-	
$g_i(\tau)$	C	-	-	-	-	-	-	-	
$g_i'(\tau)$	C	-	-	-	-	-	-	-	
$h_i(\tau)$	I	-	-	-	-	-	-	-	
γ	-	-	-	-	-	-	-	C	
γ'	-	-	-	-	-	-	-	I	
IET	S	-	-	-	-	-	-	-	
IGENC	I	-	-	-	-	-	-	-	
IGENP	I	-	-	-	-	-	-	-	
K	-	C	C	C	C	-	-	C	
L_σ	I	-	I	I	-	-	-	I	Note 1
γ	-	I	I	I	I	I	I	I	
$\Sigma\lambda$	-	-	-	-	-	-	-	C	

cf. continuation sheet 2 of this table for explanatory symbols and notes.

TABLE 3-1 (Cont.)

Parameter	Coverage Model	Reliability Model/ Equation Number 1	2	3	4	5	6	7
$\hat{\lambda}_2$	-	-	-	-	-	-	-	I
$\hat{\lambda}_3$	-	-	-	-	-	-	-	I
MD	S	-	-	-	-	-	-	S
μ	-	I	I	I	I	-	-	I
N	-	I	-	-	-	-	-	-
n_c	C	-	-	-	-	-	-	-
n_i	I	-	-	-	-	-	-	-
P	-	-	-	-	-	I	-	-
p_i	I	-	-	-	-	-	-	-
p_i'	I	-	-	-	-	-	-	-
$p_i f_i(\tau)$	C	-	-	-	-	-	-	-
$p_i' h_i(\tau)$	C	-	-	-	-	-	-	-
P_r	R	-	-	-	-	-	-	I*
P_s	I	-	-	-	-	-	-	-
Q	-	-	I	I	-	-	-	I
R	-	R	R	R	R	R	R	R
$r_i(\tau', \tau'')$	C	-	-	-	-	-	-	-
$r_i'(\tau')$	I	-	-	-	-	-	-	-
$r_i''(\tau'+\tau'')$	I	-	-	-	-	-	-	-
RSGN	-	-	-	-	-	-	-	I
RV	-	I	I	I	I	I	-	-
S or r	-	I	I	I	I	-	-	I

TABLE 3-1 (Cont.-2)

Parameter	Coverage Model	Reliability Model/ Equation Number 1	2	3	4	5	6	7	
s	S	-	-	-	-	-	-	-	
σ	S	-	-	-	-	-	-	-	Note 1
t_{d_i}	I	-	-	-	-	-	-	-	
Δt_i	I	-	-	-	-	-	-	-	
$t_{d_i}+\Delta t_i$	C	-	-	-	-	-	-	-	
T_i	C	-	-	-	-	-	-	-	
$t_j^{\ell}(i)$	C	-	-	-	-	-	-	-	
T_{mr}	I	-	-	-	-	-	-	-	
τ_s	I	-	-	-	-	-	-	-	
W	-	I	I	I	I	I	I	-	
Z	-	I	I	I	I	I	I	-	

C - computed by program

I - input by user or default table

I*- input by user, default table, or coverage model

R - result (i.e., output) of model

S - subscript used for summation, selection, etc.

\- - not applicable

Note 1: These parameters are used to specify the relationship between coverage subclasses and reliability stages for equations 2, 3 and 7. They are included in the table for reason of competeness although, as a consequence, the relationship to column headings is somewhat strained.

TABLE 3-2

PARAMETER DEFINITIONS

Symbol	Fortran Mnemonic	Name	Definition
C	C1(I), C2(I)	Coverage factor	The conditional probability, for mode equals 1, 2 respectively, that the system can recover from a permanent hardware failure in an LRU of stage I, given that sufficient spare hardware is available.
C'	CTR(I)	Transitional coverage factor	The conditional probability that the system can recover from a permanent failure in an LRU of Stage I, given that no spares are available and hence that the system must degenerate from Mode M to Mode M+1.
C_2	CCSF	Switch failure coverage factor	The coverage factor associated with recovery via degeneration, due to a category 2 switch failure.
C_σ	COVAGE	COVAGE returned valve	The coverage factor for fault subclass σ under a given set of conditions i.e., given values of MD, IET and JS in the COVAGE argument list.
d_σ	FRAC	-	The fraction of class (i.e., stage) faults which are attributed to member subclass σ.
δ	CD1(I), CD2(I)	Delta coverage factor	A term defined by the equation $C_i = C\delta^i$, with C_i the conditional probability, for Mode equals 1, 2 respectively, that the system can recover given that the first i of i+1 or more spares has failed in the idle state, that the next has not, and that recovery is initially attempted utilizing the i failed spares.

Symbol	Fortran Mnemonic	Name	Definition
δ'	CDTR(I)	Delta transitional coverage factor	A term, defined by the equation $C_i = C'(\delta')^i$, with C'_i the conditional probability of recovery given that all i of the remaining spares have failed and that all i are tested to ascertain this prior to degeneration.
$F_i(\eta)$	--	Cumulative detection probability function	The integral of the function $f_i(\tau)$, i.e., the probability that an ideal detector will detect an error within $\eta + \tau_{d_i}$ time units after its occurrence. Note: It is recommended that the user formulate the integrals for any *general function models* created, and provide these as associated integral models.
$\overline{F}_i^{\ell}(\eta) = (1-F_i^{\ell}(\eta))$	--	--	Probability of non-detection (with $F_i^1(\eta) \overset{\Delta}{=} F_i(\eta)$).
$F_j^{\ell}(\eta,i)$	--	--	The probability that an ideal scheduled detector (with a detection rate identical to that of detector j) will detect a fault within η time units after the initial delay, when detector j is scheduled to begin $t_j^{\ell}(i)$ time units after the occurrence of a fault.
$f_i(\tau)$	--	--	The user specified conditional detection rate of detector i in the absence of any competitive detection processes.
$g_i(\tau)$	CVGS, CVG1, CVG2	--	The conditional detection rate of the i^{th} detector when in competition with all other detectors.

Symbol	Fortran Mnemonic	Name	Definition
$g'_i(\tau)$	CVGP3, CVGP4	--	The conditional detection rate of the i^{th} scheduled detector when in competition with all other scheduled detectors.
$h_i(\tau)$	-	Isolation rate function	The user specified isolation rate of isolator i.
γ	GM1(I), GM2(I)	System transient failure rate	The product $\gamma' \cdot (1-P_r)$ for a unit of stage I, and for mode equals 1,2 respectively.
γ'	GMP(I)	Transient failure rate	The rate at which transient hardware errors take place in on-line units of stage I. Expressed in failures per unit time.
--	IET	Fault type code	A code which specifies the "duration" of a fault. Equal to: - 1 for permanent faults - 2 for transient faults.
--	IGENC(I)	Coverage calculation flag	An integer variable which conditionally specifies the source of permanent fault coverage factors for stage I: - for IGENC < 0, coverage is calculated prior to each run-set - for IGENC=0, coverage is not calculated (i.e., either input or defaulted) - for IGENC > 0, coverage is calculated prior to the first run-set only. Note: A negative coverage value input by the user serves as a higher precedence command, and forces calculation of a "replacement" value, given only that coverage itself is applicable to the corresponding reliability model.

Symbol	Fortran Mnemonic	Name	Definition
--	IGENP(I)	Transient recovery calculation flag	An integer variable similar to IGENC, except that transient fault recovery probabilities (P_r) are the candidates for calculation.
K	K(I)	Inverse dormancy factor	The ratio of λ/μ for LRU's of stage I.
L_σ	IFSC	Fault class indicator	A pointer or linkage to fault subclass σ, wherein the subclass accounts for a fractional portion (d_σ) of the faults which occur in stage L_σ of the system.
λ	LAM(I)	On-line failure rate	The rate at which permanent hardware failures take place in on-line units of stage I. Expressed in failures per unit time.
$\Sigma\lambda$	SUMLAM	Simplex on-line failure rate	The sum of on-line failure rates over units of all stages, for the purpose of calculating simplex reliability.
$\hat{\lambda}_2$	SLH2	Category 2 switch failure rate	The occurrence rate of permanent hardware failures which cause degeneration from mode M to mode M+1. Expressed in failures per unit time.
$\hat{\lambda}_3$	SLH3	Category 3 switch failure rate	The occurrence rate of permanent hardware failures which cause total system failure. Expressed in failures per unit time.
--	MD	Mode	The computer system operating mode. In the Dual Mode model, modes are distinguished by the quota of on-line units for each stage. In the Coverage model, D/I/R

Symbol	Fortran Mnemonic	Name	Definition
			processes are defined separately for modes 1 and 2, as well as dummy mode 0. MD is equal to: - 1 for full up - 2 for degenerate - 0 for transitional (coverage only, during transition from MD=1 to MD=2).
μ	MU(I)	Stand-by failure rate	The rate at which permanent hardware failures take place in stand-by LRU's (spares) of stage I. Expressed in failures per unit time.
N	N(I)	Modular redundancy factor	The number of identical active units in a fully operational NMR system.
n_c	LCM	--	The least common multiple of the ni's where ni is the repetition factor (where applicable) of the i^{th} detector, in terms of minor cycles.
n_i	IREP(I)	Repetition factor	The integer ratio of the repetition period of the i^{th} scheduled detector to the minor cycle duration, i.e., $T_i=n_iT_{mr}$.
P	P(I)	--	The probability of stage I failing with a logical zero output.
p_i	--	--	The conditional probability that the i^{th} detector will detect an arbitrary fault, given infinite time and no competitive detection processes.
p_i'	---	--	The probability that the i^{th} isolator is able to isolate a fault.

Symbol	Fortran Mnemonic	Name	Definition
$p_i f_i(\tau)$	--	--	The detection probability density function of detector i in a non-competitive environment (i.e., when no other detectors are operative). The value of p_i and the function $f_i(\tau)$ are defined explicitly, as is the initial detection delay τ_{d_i}. The "rate function" $f_i(\tau)$, then, represents the rate of detection for an ideal detector, i.e., one which guarantees detection, given an adequate amount of time. Since τ_{d_i} is explicit, $f_i(\tau)$ must be defined so that $f_i(0^+) \neq 0$. The duration of the function is defined as the value of τ after which detection cannot occur.
$p_i' h_i(\tau)$	--	--	The isolation probability density function of the process associated with detector i. Note also $h_i(\tau)$ is a rate function defined similarly to $f_i(\tau)$.
P_r	PRC1(I), PRC2(I)	Transient recovery probability	The probability, for mode equals 1, 2 respectively, that the system can recover from a transient fault occurring in an LRU of stage I.

Symbol	Fortran Mnemonic	Name	Definition
p_s	PFDS(ISU)	--	Probability of detecting a failure in a failed spare during checkout. Note that this has the effect of directly reducing the final coverage result, since no provision is made for subsequent recovery from undetected failures in spares. The value selected must therefore account for all secondary recovery capabilities of the computer system. Note also that the value is, for reason of its inclusion in the coverage model, assigned to a fault subclass rather than an LRU.
Q	Q1(I),Q2(I)	Quota	The number of on-line LRU's required in stage I for system operation in mode equals 1, 2 respectively.
R (t)	R	Reliability	The probability that the system is operational at time t given that it was operational at time 0.
$r_i(\tau',\tau'')$	--	Recovery probability function	The conditional probability of system recovery, given detection and isolation delays of τ' and τ'', respectively. $r_i(\tau',\tau'') \equiv r_i'(\tau') \cdot r_i''(\tau'+\tau'')$
$r_i'(\tau')$	--	Fault propagation recovery function	The conditional probability of system recovery, given detection time τ' for the i^{th} detector at the end of which, fault propagation ceases.

Symbol	Fortran Mnemonic	Name	Definition
$r_i''(\tau'+\tau'')$	--	Time lost recovery function	The conditional probability of system recovery, given total detection and isolation delay of $(\tau'+\tau'')$ for the i^{th} detector, isolator pair at the end of which, fault recovery is initiated.
--	RSGN	Reassignment flag	A logical variable which for RSGN =.TRUE., enables operational LRU's in a failed channel to be reassigned i.e., released to the spares pool =.FALSE., precludes reassignment (Note: RSGN must be set true if the quota in mode M+1 is greater than in mode M).
RV	RV(I)	Restoring organ reliability	An overall limiting probability of success applied to stage I.
S or r	S(I)	Spares	The number of spare LRU's available in stage I at time t = 0.
s	JS	--	The quantity of spare LRU's which must be checked out before recovery can proceed. (normally 0 or 1).
σ	ISU	Fault subclass	An integer subscript (0<ISU≤8) which identifies a set of competitive D/I/R processes (i.e., a fault subclass). One or more such sets may be assigned to a single stage of the reliability model.

Symbol	Fortran Mnemonic	Name	Definition
t_{d_i}	TDEL	Delay time	The delay time associated with initiation of detection process i. For scheduled detectors, it is measured relative to the beginning of a major cycle; for non-scheduled detectors, it is the interval between occurrence of a fault and the instant it can first be detected.
Δt_i	TDUR	Duration time	The time interval during which a detection function is non-zero.
$t_{d_i}+\Delta t_i$	--	--	Finishing time for detector i, i.e., the time relative to the start of a major cycle (scheduled detectors) or the occurrence of a fault (non-scheduled detectors) after which detector i is not effective. Note: $\int_0^\infty f_i(\tau)d\tau = \int_0^{\Delta t_i} f_i(\tau)d\tau \equiv 1.0$ (delay is external to function)
T_i	--	--	The repetition period of the i^{th} scheduled detector.
$t_j^{\ell}(i)$	CYTJL(J,L,I)	--	Largest solution t to the equation $t = t_j+\nu T_j-(\ell-1)T_i$, $\nu=0,1,2,\ldots$, in the range (t_i, t_i+T_i). (If there is no

Symbol	Fortran Mnemonic	Name	Definition
			solution in this range, $F_j^{\ell}(\eta) = 1$.) Thus, t_j^{ℓ} represents the starting time of the last occurrence of the j^{th} scheduled test in the interval $(t_i+(\ell-1)T_i,\ t_i+\ell T_i)$.
T_{mr}	TMINOR	Minor Cycle	The minor cycle duration, i.e., the greatest common divisor of the repetition period T_i.
τ_s	TFDS(ISU)	Checkout Time	The average on-line time required to test a single spare, given an accompanying test success probability of p_s. Note that the value is, for reason of its inclusion in the coverage model, assigned to a fault subclass rather than an LRU.
W	W(I)	Division factor	The number of identical sub-units which comprise one unit in stage I. The sub-unit failure rate λ_s thus relates to the unit failure rate λ as $\lambda_s = \lambda/W$.
Z	Z(I)	Iteration factor	The number of identical units operating in series which comprise stage I. The reliability of the stage is thus the product of the reliabilities of the Z sub-stages.

SECTION 4

SOFTWARE

CARE2 was developed using the CDC RUN76 compiler, under the KRONOS 2.1 Operating System, on the CDC 6700 Time-sharing Computer at Raytheon MSD, Bedford, Massachusetts. The program is essentially UNSI Fortran IV, with the exception that, like original CARE, it is designed for execution on a 60 bit CDC computer. The field length required for the complete program version is less than 130K (octal), and for the reduced version (without plotting options) less than 100K.

Program modules added were designed to be of comparable scope with existing modules as far as programming efficiency and run time considerations would permit. The statistical, mathematical or implementation significance of each revised or added subprogram is described in this section.

4.1 PROGRAMMING CONSIDERATIONS

The task of including both a dual mode reliability model and a model for the calculation of coverage factors, was originally expected to entail additions to CARE only. Ideally, the former would replace dummy subprogram NEQ7, and the latter would require one additional CALL statement in the main program, plus the implementing code. However, the relatively complex method of passing statistical parameters to the reliability models, combined with the need for communication between stages of the dual mode model, disallowed this ideal. Simply stated, the CARE program requires 1) that an equation use at most one parameter of each type, and 2) that individual systems be evaluated serially. Thus major revisions of the data base structure were undertaken. The possibility of expanding the existing structure to accommodate a large number of additional parameters was rejected in favor of a simpler format.

In CARE2, a computer system is represented as a series of one or more stages, each of which is comprised of identical subunits including one or more optional spares. Each invocation of equations 1-6 corresponds to the definition of a single such stage, whereas equation 7 has internal provision for representing up to eight stages.

For the group of up to 10 stages which can be modelled at one time, there exists a base run vector (i.e., an array of dimension 10) for each parameter type. In addition, any one of 19 parameter types may be selected for variation (cf. sheet 2 of Table 4-2), which is accomplished internally by alternately refreshing the selected run vector from a single 10 x 16 iteration array and evaluating the system. (The latter process, consisting of up to 16 evaluations or runs, is termed a run-set.) Optionally, run-sets may also be repeated, following user changes to run vectors, iteration data and iteration parameter.

Coverage factors are treated the same as other parameters when input directly by the user. Conversely, however, they can be computed, for equations to which they apply, provided that seperate coverage input data is supplied. These inputs, because of their relative diversity, cannot be varied as described above. Most, with the exception of the flags which request calculation and the linkage table which unites the two halves of the program data base, can be changed at the run-set level.

In structuring the input algorithms for CARE2, it was apparent that both the size of the data base and the assumption that the program would be used for sensitivity analyses, indicated an "inputs only for changes" rule should apply throughout. Thus program defaults, where provided, are initiated prior to user inputs (except in unaltered portions of READIN). The default values themselves, in the case of basic reliability model parameters, may optionally be input early in the batch run. In addition, by setting LSTCH to true (cf. Table 5-1), the inputs required for parameter variation can often be minimized. In this case, non-default inputs to the iteration array PARAM will be extended upward along the iteration dimension, replacing only default values.

The development of CARE2 was performed using MODIFY, a system level program on the Bedford 6700. MODIFY facilitates the manipulation of source code by representing the original and subsequent versions as card deck images stored in disk files. Cards and decks are inserted, moved or deleted, by introducing directives in groups which are also represented as card decks, and are saved with the source for future reference.

The identifying names seen in columns 73-80 of the CARE2 source code refer to the origin of the card. Thus cards bearing the subprogram name, in the case of original CARE subprograms, were in fact in the original program. The modification set name, where it appears, in general refers to the reason for the card's replacement or addition.

For example, CAREFIX cards arose from early modifications which were required to enable the CARE program to operate properly on the Bedford computer. MAINLOG cards indicate basic logic modifications, and PARMOD cards refer to the restructured parameter data base.

4.2 SUBPROGRAM DESCRIPTIONS

This section describes, in paragraph form, each of the new and altered subprograms, as they exist in the "complete" version of the CARE2 program. Each is also described in flow diagram form (cf. Appendix A) as well as in the source listing itself (cf. Appendix B). Table 4-1 is provided in order to aid in their location.

TABLE 4-1

CROSS-REFERENCE LISTING OF CARE2 SUBPROGRAMS

Subprogram Name	Source Listing S/R Order #	Paragraph #	Flowchart #	Comments
BISECT	2	-	-	Original
*CARE2	1	4.2.1.1	A-1	Modified CARE main program
COVAGE	47	4.2.4.1	A-18	Added
COVGEN	37	4.2.2.3	A-8	Added
CVGPI	54	4.2.4.8	A-25	Added
CVGP3	51	4.2.4.5	A-22	Added
CVGP4	52	4.2.4.6	A-23	Added
CVGS	50	4.2.4.4	A-21	Added
CVG1	48	4.2.4.2	A-19	Added
CVG2	49	4.2.4.3	A-20	Added
CVTJL	53	4.2.4.7	A-24	Added
DCG	45	4.2.3.7	A-16	Added
DCH	43	4.2.3.5	A-14	Added
DCOMB	46	4.2.3.8	A-17	Added
DCR	39	4.2.3.1	A-10	Added
DCRU	44	4.2.3.6	A-15	Added
DCS	42	4.2.3.4	A-13	Added
DCT	41	4.2.3.3	A-12	Added
DCT2	40	4.2.3.2	A-11	Added
*EQUAL	3	-	-	Original
FEVAL	56	4.2.4.10	A-27	Added
FFAC	4	-	-	Original
FINTEG	57	4.2.4.11	A-28	Added
FNCK	5	-	-	Original

*Affected by array dimension modifications, as required for optional reduced field length execution.

TABLE 4-1 (cont.)

Subprogram Name	Source Listing S/R Order #	Paragraph #	Flowchart #	Comments
FN1	61	4.2.4.15	A-32	Added
FN1I	62	4.2.4.16	A-33	Added
FN2	63	4.2.4.17	A-34	Added
FN2I	64	4.2.4.18	A-35	Added
FN3	65	4.2.4.19	A-36	Added
FN3I	66	4.2.4.20	A-37	Added
FN4	67	4.2.4.21	-	Added (dummy routine)
FN4I	68	4.2.4.22	-	Added (dummy routine)
FN5	69	4.2.4.23	-	Added (dummy routine)
FN5I	70	4.2.4.24	-	Added (dummy routine)
FN6	71	4.2.4.25	-	Added (dummy routine)
FN6I	72	4.2.4.26	-	Added (dummy routine)
IGET	58	4.2.4.12	A-29	Added
INTEGR	6	-	-	Original
IPUT	59	4.2.4.13	A-30	Added
ISHIFT	60	4.2.4.14	A-31	Added
MOVC	26	4.2.1.4	A-4	Modified to use ISHIFT
NEQ1A	9	4.2.1.2	A-2	Corrected logic error
NEQ1B	10	-	-	Original
NEQ2A	7	-	-	Original
NEQ2B	8	-	-	Original
NEQ3	11	-	-	Original
NEQ4A	12	-	-	Original
NEQ4B	13	-	-	Original
NEQ5	14	-	-	Original
NEQ6	15	-	-	Original

TABLE 4-1 (cont.-2)

Subprogram Name	Source Listing S/R Order #	Paragraph #	Flowchart #	Comments
*NEQ7	16	4.2.1.3	A-3	Replaced dummy
PARAR1	17	-	-	Original
PGET	38	4.2.2.4	A-9	Added
PLOTN	24	-	-	Original
*PLOTR	22	-	-	Original
*PLOTRV	18	-	-	Original
*PLOTT	23	-	-	Original
PROD	20	-	-	Original
PROD1	21	-	-	Original
RCOMB	19	-	-	Original
*READIN	34	4.2.1.5	A-5	Modified
READIN2	35	4.2.2.1	A-6	Added
REDUC	27	-	-	Original
RELATE	28.5	-	-	Deleted
RELEQS	30	-	-	Original
*RIFDIF	29	-	-	Original
RITE	30.5	-	-	Deleted
ROMBD	28	-	-	Original
ROWPLT	31	-	-	Original
SCAN	33.5	-	-	Deleted
SEARCH	31.5	-	-	Deleted
SIMPLE	33	-	-	Original
SIMPR1	32	-	-	Original
SPECIT	55	4.2.4.9	A-26	Added
TRANSFR	36	4.2.2.2	A-7	Added
WRNR	25	-	-	Original

*Affected by array dimension modifications, as required for optional reduced field length execution.

4.2.1 Altered Subprograms

4.2.1.1 CARE2 (Main subprogram)

The basic order of processing is the same as that of CARE. All output options which were allowed in CARE exist unchanged in CARE2, with the exception of simultaneously varying more than one parameter type and automatically obtaining all possible combinations. The current ability to repetitively input changes to both parameter and coverage data provides an equivalent capability and, in addition, allows for direct user selection of the combinations to be evaluated. The restrictions on plotting options which applied to evaluations of a product of reliabilities now apply, analogously, to evaluation of a dual mode system (Eq. 7) with more than one stage.

A dual mode system of up to 8 stages can be modelled in series with other computer models, by declaring array PROD as a like quantity of 7's, followed by the numbers of the other desired equations.

As in CARE, the invariant data for a batch run is specified via READIN. The additional data now contained in this category, as included in Figure 5-1 and Table 3-1, consists of:

- flags which enable the calculation of coverage and the display of intermediate coverage results
- a flag enabling the display of mode 1 reliability of the dual mode system model and its individual stages
- a flag enabling display of the reliability model parametric inputs, iteration array and dual mode system special data (defaulted to .TRUE.).

- linkage data to allow calculated coverage values to be applied to reliability models
- a flag to specify the method used to fill the iteration array
- a debug flag (see below).

Following READIN, the variant data is input via READIN2. (The data following READIN, is termed variant because the call to this subprogram is the beginning of a main subprogram loop which may be continued indefinitely.) All inputs to READIN2 are made via control cards, which determine the data to be input as well as the format to be used. This data includes both parameters for the reliability models and inputs to the coverage model.

After all data has been processed, the coverage driver, COVGEN, is called to effect all requested coverage calculations and transfer their result to the run vectors, using the linkage algorithm. The iteration array, PARAM, is then adjusted as required, and the prepared inputs to the reliability models are (optionally) displayed.

If the DEBUG flag has been set, control passes directly back to READIN2, and no reliability model evaluation occurs. This is useful 1) for checking data cards for validity, and 2) when coverage analyses only are desired.

If DEBUG is not set, reliability equations are evaluated sequentially, and reliability versus the independent variable, followed by the selected computational and plot options, are printed. In the case of dual mode system, the independent variable must be time and must begin at zero, due to the use of structured numerical integration.

An additional, more specific, overview of CARE2 is provided in Appendix A (cf. flow diagram A-1).

4.2.1.2 NEQ1A

A simple correction was made to this routine during the initial investigation of CARE. Specifically, a card sequence error was found in the algorithms' source listing which in turn caused the summation on L (or LX) to be set equal to its last term. (cf. flow diagram A-2)

4.2.1.3 NEQ7

As mentioned in Section 4.1, the data available to this sub-routine via its argument list is inadequate for evaluation of the dual mode model. The model is thus referenced separately by the main program, with parameters held in common blocks.

NEQ7 is capable of interpolating the results stored in array R(121,25), when referenced via RELEQS by subprograms which perform secondary calculations (e.g., MTF). It uses a 2nd order (parabolic) integration technique. (cf. flow diagram A-3)

4.2.1.4 MOVC

This routine, which transfers six-bit characters between words of memory, originally required a Langley installation routine called SETBIT. It now uses ISHIFT, which is written in COMPASS and included in the source code. (cf. flow diagram A-4)

4.2.1.5 READIN

The basic input package provided by READIN remains unaltered with the exception that the reliability model parameters (Q, N, LAM, etc.) have been removed from namelist $VAR. As in CARE, READIN is referenced once by CARE2, and thus data processed by it cannot be changed during later execution.

Since CARE inputs were originally based on a time-sharing question and answer format, they are rather awkward for use in batch mode, and for clarity require the use of either a standardized procedure or analysis of subprogram READIN. The substantial number of new inputs now required to accommodate the dual mode and coverage models complicate this matter further. In an attempt to clarify the issue somewhat, the current input algorithm is shown in Figure 5-1, in flowchart form.

The essential differences between CARE and CARE2 in this context are:

- A number of logical variables have been added to Namelist $OPTSON, including DEFCHNG and COVPRC.
- Two additional Namelists, $DEFAULT and $COVCAL, are read following $VAR, conditional on their respective flags, DEFCHNG and COVPRC, being set .TRUE.
- As in CARE, the inputs to READIN are invariant for the entire batch run. The addition of subprogram READIN2 allows most parametric data to be altered between run-sets
- The basic reliability model parameters (Q, S, LAM, etc.) have been moved from Namelist $VAR in READIN to Namelist $PARVEC in READIN2.

More detailed information on the current complement of variables and flags, as well as an overview of the revised READIN flow and the context of how it is used in CARE2, is provided in Sections 2.3, 4.2.1.1 and 5.1.2 and flow diagram A-5.

4.2.2 New Utility Subprograms

4.2.2.1 READIN2 (NRS,DMFLG,IRES,LVARY,DVAL,DEBUG)

This subroutine is called by CARE2 directly after READIN, and again after completing evaluation of the reliability models and secondary calculations. It is thus possible to continue altering the parametric and/or coverage data base, and in turn re-evaluating the models, indefinitely.

Due to the "inputs only for changes" rule, extensive use is made of the Namelist feature of Fortran. Also, for this reason, all input processing is carried out in direct response to control cards. The routine continuously reads these cards, using a standard format, and processes them along with other indicated data, such as namelist inputs, until such time as a return to model evaluation is specifically requested.

The control cards contain an identifying code in columns 1-3 which indicates one or more of the following:

- a reference to a particular Namelist follows in the input stream
- the control card contains specific information in a fixed format
- the subroutine is to perform a specific processing task.

A code which begins with "I" requests the reading of a Namelist, and those beginning with "P" cause the immediate printing of a portion of the data base. A general description of each of the 12 control codes is contained in table 5-2. (cf. flow diagram A-6)

4.2.2.2 TRANSFR(X,I)

This subroutine conditionally prints a line of parametric input data starting at the address of X. I is the element of array PCW(22), which contains applicability information concerning the parameter vector to be printed. Only those values pertaining to the applicable reliability model are printed. (cf. flow diagram A-7)

4.2.2.3 COVGEN

If one or more of the coverage factors are to be calculated, this routine is called by CARE2 directly after return from READIN2. COVGEN responds to requests for coverage calculation from both the IGENC(10) and IGENP(10) flags (which correspond to stages of the reliability models) and to individual requests initiated by setting coverage parameters to negative values. For each request, the routine determines applicability by testing the proper element of PCW(22).

Since delta coverage is computed using corresponding basic coverage, the order of calculation is by parameter type, with subclass linkage tests secondary. A fault subclass in linked to a stage if the corresponding element in IFSC(8) is the stage number. The corresponding fractional fault rate in FRAC(8) is then applied. (cf. flow diagram A-8)

4.2.2.4 PGET(IR,I,NPROD)

PGET refreshes the run vector of the parameter to be varied with NPROD values from row I of the iteration array PARAM (16,10). IR indicates the parameter of interest. (cf. flow diagram A-9)

4.2.3 Dual Mode Model Implementation Subprograms

4.2.3.1 DCR(J,UNITR,RELM1,RSYS)

This subroutine, driven by the time step counter J, computes the dual mode system (eq. 7) reliability RSYS at the point in time t = STEP*(J-1). It also evaluates the system and individual stage reliabilities as they apply to mode 1 operation alone (RELM1,UNITR(8)). (cf. section 3.1.7 and flow diagram A-10)

4.2.3.2 DCT2(J)

This function computes the probability that the system will have survived to time t=STEP*(J-1), having degraded due to a category 2 switch failure. (cf. section 3.1.6 and flow diagram A-11)

4.2.3.3 DCT(IUN,J)

This function computes the probability that the system will have survived to time t=STEP*(J-1), having degraded due to a failure in stage IUN. (cf. section 3.1.5 and flow diagram A-12)

4.2.3.4 DCS(IUN,T,TAU)

This function computes the conditional probability that stage IUN can survive in mode 2 from time TAU to time T, given that a degenerative failure occurred in the stage at time TAU. (cf. section 3.1.4 and flow diagram A-13)

4.2.3.5 DCH(IUN,TAU)

This function computes the probability density of a degenerative failure in stage IUN of the system at time TAU. (cf. section 3.1.3 and flow diagram A-14)

4.2.3.6 DCRU(IUN,MD,L,TAU)

This function computes the probability of using at most L spare units in stage IUN by time TAU, given the system is operating in mode MD. (cf. section 3.1.2 and flow diagram A-15)

4.2.3.7 DCG(IUN,MD,I,T)

This function computes the probability of using exactly I spare units in stage IUN by time T, given the system is operating in mode MD. (cf. section 3.1.1 and flow diagram A-16)

4.2.3.8 DCOMB(TOP,K)

This function computes the binomial coefficient of the expression:

$$\binom{TOP}{K}$$

It is equivalent to the RCOMB function except for the case when K=0. (cf. flow diagram A-17)

4.2.4 <u>Coverage Model Implementation Subprograms</u>

4.2.4.1 COVAGE (ISU, MD, IET, JS)

This function returns a single value, either a coverage factor or transient recovery probability, for each reference. (Since little sharing of intermediate variables is possible in coverage calculations, efficiency is not lost by the single value format.) The routine is driven by COVGEN, which provides the proper arguments for the type of coverage value required for the reliability models.

The Fortran statement for referencing COVAGE is:

CVAL = COVAGE(ISU,MD,IET,JS)

where

- ISU is the fault class or subclass
- MD is the computer system operational mode
- IET is the major fault type, either a permanent or transient failure
- JS is the number of spare units which must be checked out during the recovery process.

With reference to the definition of coverage (cf. section 2.3) as a conditional probability, the arguments MD and IET can be taken as the "given" conditions, i.e., coverage input data may be entirely different for differing values of these variables. In addition, a coverage factor is computed independently for each class or subclass of faults, where a class is defined as those faults occurring in a specific stage of the computer model. Within the fault class, faults may arbitrarily be divided into subclasses, provided that a decimal fraction, specifying the relative rate of fault occurrence, is assigned to each subclass. A subclass may be those faults occurring in a specific part, or subunit of an LRU, or those having a certain characteristic, such as "easy to find", etc.. The coverage factor applied to a computer stage is the average of the factors calculated for the applicable subclasses, weighted by the relative rates of occurrence assigned by the user.

Each call to the COVAGE sub-program results in a calculation of the systems' conditional ability to detect, isolate (to a specific LRU) and recover from the specified subclass of faults (ISU), under given conditions of mode, fault type and quantity of spares checked (as defined by MD, IET and JS respectively). To accomplish this, it is necessary that the components of the recovery system (i.e., the FTF's) be described, as input to the model, in terms of their stand-alone capacity to contribute. The model evaluation then combines the effects of these by statistically accounting for the competitive nature of the detectors, and the conditional isolation and recovery success probability associated with each.

The quantity of the data required by the model, due to its flexibility, is necessarily large. The user can, however, do much to simplify the corresponding data specification task by approaching it systematically. A convenient (although not necessary) starting point for this is to first list the names of all hardware and software fault detection devices available in the computer system. If there are 20 or less, a unique number is then assigned to each name which, in turn, represents the D/I/R mechanism corresponding to that detector. (If there are more than 20, sharing of numbers is required.)

Figure 4-1, by way of example, represents the processes (for all valid combinations of mode and fault type) which deal with faults of subclass 3. Each row therein represents the portion of a D/I/R mechanism corresponding to a single detector and each column, the set of coexistent processes which compete for detection of (and subsequent recovery from) faults of a particular type and occurring during a particular mode of operation. The four numbers

Coverage Mechanisms for Fault Subclass

#3 - MEMORY BIT PLANE

Mechanism Number	Detector Name	Fault type/mode PERM/1	PERM/2	TRANS/1	TRANS/2	PERM/0
		D I E T	D I E T	D I E T	D I E T	D I E T
1	6 BIT MEMORY CODE (Hamming)	1, 1, 1, 1	1, 1, 1, 2	1, 1, 1, 1	1, 1, 1, 2	1, 1, 1, 1
2	OUTPUT COMPARATOR	9, 2, 1, 2	Not Applicable	9, 3, 1, 2	N/A	9, 2, 1, 2
3	I/O WRAPAROUND	12, 2, 1, 1	12, 2, 2, 2	N/A	N/A	12, 2, 1, 1
4	CPU SELFTEST	14, 2, 1, 1	14, 2, 2, 2	N/A	N/A	14, 2, 1, 1
20	unused					

FIGURE 4-1

SAMPLE FUNCTION SELECTION CHART

contained in each intersect, designated function numbers, specify the detection, isolation, and 2 recovery characteristic functions selected to represent the components of the corresponding mechanism. Each such number designates a fully specified time function whose parameters are contained in one of four specification arrays, corresponding respectively to detection (D), isolation (I), error-propagation-recovery (E) and time-delay-recovery (T) functions (cf. Table 5-4). Each array has sufficient room for a number of specification lists which, in turn, are common to the entire model.

Looking again at figure 4-1, it should be noted that the only restrictions placed on the selection of function numbers is that a non-zero (i.e., enabled) detector be accompanied by non-zero function numbers for isolation and recovery, and that all 4 functions be properly defined.

In order to generate a characteristic curve versus time (or its integral), the coverage model uses an evaluation sub-program which applies the specifications of the selected function number to one of several common function models. Each model, in turn, is a Fortran sub-program, in the form of a generalized equation with one independent variable and up to 3 fixed parameters, as for example:

$$f(t) = a(e^{-bt}+c).$$

The user generates specifications at run time, in part, by linking each function number to a particular function model, and in part, by selecting appropriate values for a, b and c. He may also alter, or add to, the set of function models, recompiling only the corresponding model sub-programs. Once specification lists have been input, the D/I/R sequences (i.e., mechanism portions corresponding to specific conditions) can be defined, one at a time or in groups, by selecting 4 function numbers for each.

It should be noted that the functions selected are characteristic curves which describe detection, isolation and recovery capabilities under a given set of conditions. A software detection device which is good at finding permanent failures in a CPU register may perform poorly when the register simply "drops a bit." The characteristic curves, and thus the functions (numbers) selected, will be different even though the same detector is used.

The 4 processes comprising a D/I/R sequence must be clearly understood, from a statistical point of view, before a realistic recovery system can be modelled. Up to this point, these functions have not been given a physical significance in order to avoid confusion. Although the 4 functions are implemented in the same manner, the detection and isolation functions have units of probability density, whereas the 2 recovery functions have units of probability. More specifically, the detection and isolation functions are rate functions, which for the present purposes means that the shape and coefficient of each detection and isolation function is provided and utilized separately. These rate functions (exclusive of the coefficient) are required by the model to have an integral of 1.0, with the coefficient then lying between 0.0 and 1.0. A normalization routine is provided in READIN2 to assure that the rate functions are correct.

The error-propagation-recovery and time-delay-recovery functions give the probability of successful recovery versus the time 1) that errors are allowed to propagate through the system and 2) that is required for detection plus isolation of a fault. The model treats these probabilities as conditional upon each other, such that the success probability for a given detection time and a given isolation time is the product of the two recovery functions.

Other data items which affect coverage are the probability of recognizing a failure in a spare LRU, as selected to replace a failed active unit, and the time required to test such a spare, whether or not it has failed. (cf. section 3 and flow diagram A-18)

4.2.4.2 CVG1(I,T)

This function computes the coefficient of the function for unscheduled impulse detector I, at time T units after the occurrence of a fault. The g function is an impulse in this case, and its coefficient indicates the competitive effectiveness of the detector. (cf. section 3.2.4 and flow diagram A-19)

4.2.4.3 CVG2(I,T)

This function computes the value of the g function for unscheduled finite detector I, at time T units after the occurrence of a fault. The g function is a measure of the competitive effectiveness of the detector. (cf. section 3.2.4 and flow diagram A-20)

4.2.4.4 CVGS(I,T)

This function computes the value of the g function for scheduled detector I, at time T units into the period of the test (detector). The detector may be either an impulse or finite function, and the calculation is based in part on the sample g' values saved in array GPAR(20,101). The g function is a measure of the competitive effectiveness of the detector. (cf. section 3.2.4 and flow diagram A-21)

4.2.4.5 CVGP3(I)

This subroutine computes and stores N+1 samples of the g' function for scheduled impulse detector I, over its period (i.e., the time between successive runnings of the test). The assumption is made that no detection can be made after one such period, measured from the occurrence of a fault. The g' function is a measure of the effectiveness of the test, in competition with other scheduled detectors. (cf. section 3.2.5 and flow diagram A-22)

4.2.4.6 CVGP4(I)

This subroutine computes and stores N+1 samples of the g' function for scheduled finite detector I, over its period (i.e., the time between successive runnings of the test). The assumption is made that no detection can be made after one such period, measured from the occurrence of a fault. The g' function is a measure of the effectiveness of the test, in competition with other scheduled detectors. (cf. section 3.2.5 and flow diagram A-23)

4.2.4.7 CVTJL(J,L,I)

This function returns the time difference value $t_j^{\ell}(i)$ for use by CVGP3 and CVGP4 in evaluating the competition between scheduled detectors, i.e., the effect of detector J on detector of interest I. (cf. flow diagram A-24)

4.2.4.8 CVGPI(I,T)

This function interpolates the sample g' values (for detector I) which have been saved in array GPAR(20,101) by CVGP3 and CVGP4. The result is used by CVGS in determining the effectiveness of scheduled detectors in competition. (cf. flow diagram A-25)

4.2.4.9 SPECIT(FLIST,NUM,IGFT,ISCH,IREP,INTF,COEF,TDEL,P1,P2,P3,TDUR)

This subroutine sets up a specification list (in one of four arrays) which defines a characteristic curve (function) versus time for either a detection, isolation, error-propagation-recovery or time-lost-recovery process. The respective specification arrays which receive this data are FDET(7,200), FISO(7,50), FEPR(7,25), or FTLR(7,25). FLIST is the address of one of these arrays, and NUM the column position within it where the data will be stored. The remaining arguments are the function specifications:

IGFT	The identification number of the function model to be employed in evaluating the curve
ISCH	The scheduling indicator. Equal to 0 for unscheduled, or 1 for scheduled detectors. Does not apply to other processes
IREP	The repetition factor of scheduled detectors only, i.e., the number of minor cycles in the detector period
INTF	The integral-defined indicator. Equal to 0 if the integral model corresponding to IGFT above does not exist, and 1 if it does
COEF	The explicit coefficient of the function. For detection and isolation, it is also the infinite-time success probability
TDEL	The delay time associated with a detector or isolator. For detectors, it is measured either from the occurrence of a fault (unscheduled), or from the beginning of the test period (scheduled)
P1,P2,P3	Arbitrary parameters which are passed to function model number IGFT via COMMON block CVB4, and used for the evaluation of the process characteristic. If normalization is to be performed using the routine in READIN2, P1 must be an internal coefficient in the function model selected to represent either a detection or isolation rate function

- TDUR The duration of any finite process, including recovery. This is used to determine upper limits for numerical integrations, and for reasonableness sampling when requested.

(cf. flow diagram A-26)

4.2.4.10 FEVAL(FLIST,NUM,T)

This function evaluates the characteristic function representing a detection, isolation, error-propagation-recovery, or time-lost-recovery process. The function specifications are retrieved from the array position indicated by FLIST (which may stand for FDET, FISO, FEPR, or FTLR) and NUM (the column position within the array). The independent variable T is passed to the function model. (cf. flow diagram A-27)

4.2.4.11 FINTEG(FLIST,NUM,T)

This function evaluates the time integral of the characteristic function representing detection (although it could be used for another process). The function specifications are retrieved as in FEVAL, and the integral-defined indicator is tested. If it is greater than 0, the independent variable is passed to the proper integral model. Otherwise, a numerical integration (Simpson 3 point) is performed using N+1 samples of the function, which are returned by FEVAL. N is a local variable which may be altered by the user, via a recompilation of FINTEG alone. (cf. flow diagram A-28)

4.2.4.12 IGET(IWORD,INDEX)

This function returns the INDEX'ed 12 bit field of IWORD (one of 5 in the 60 bit word), as a right justified integer. (cf. flow diagram A-29)

4.2.4.13 INPUT(IWORD,INDEX,ICODE)

This subroutine packs the rightmost 12 bits of ICODE into the INDEX'ed 12 bit field of IWORD, after first clearing the field. (cf. flow diagram A-30)

4.2.4.14 ISHIFT(IWORD,NBITS)

This COMPASS function returns, as an integer, the result of performing a left circular shift of NBITS on IWORD. (cf. flow diagram A-31)

4.2.4.15-16 FN1(T) and FN1I(T)

This function model and corresponding integral model represent the characteristics of a "single pulse" or constant amplitude function. General parameter P1 is the amplitude, and the only parameter used in this case. It is important to note that the model does not incorporate a cutoff of its own, but is defined simply as FN1=P1. The coverage model applies the TDEL and TDUR values in its evaluation, in such a way as to properly service the numerical integral routines. (cf. flow diagrams A-32 and A-33)

4.2.4.17-18 FN2(T) and FN2I(T)

This function model and corresponding integral model represent the characteristics of a "pulse train" function, with amplitude P1, pulse width P2, and pulse repetition period P3.

Due to the discontinuities inherent in this function, it is not recommended for general use, except on an experimental basis. (cf. flow diagrams A-34 and A-35)

4.2.4.19-20 FN3(T) and FN3I(T)

This function model and corresponding integral model represent the characteristics of an exponential function. The equation is:

$$FN3(T)=P1(e^{-P2*T}+P3)$$

(cf. flow diagrams A-36 and A-37)

4.2.4.21-26 FN4(T), FN4I(T), FN5(T), FN5I(T), FN6(T) and FN6I(T)

These dummy functions are provided to allow the user to expand the inventory of models, as more complex systems for coverage modelling are evolved.

4.3 MAJOR ARRAYS AND COVERAGE DATA BASE

The following sections describe both the new format of the reliability model parameter arrays, and the changes in dimension needed to run the reduced field length version of CARE2.

4.3.1 Reliability Parameters

As mentioned previously, the original 16X10 parameter arrays proved too cumbersome for use with the dual mode model, due to the complex method employed for extracting individual elements. The new version of this data base consists of a set of 19 independent and 3 dependent parameter vectors, of dimension 10. Each vector holds the inputs, for up to 10 computer stages, for one parameter (e.g., Q, LAM etc.).

A parameter control array, PCW(22), contains applicability and other information in packed format, with one element of the array corresponding to each parameter. The first part of table 4-2 shows the fields within a word, and the second part of the table gives the parameter order and association of parameters with default variables.

4.3.2 Reduced Field Length CARE2

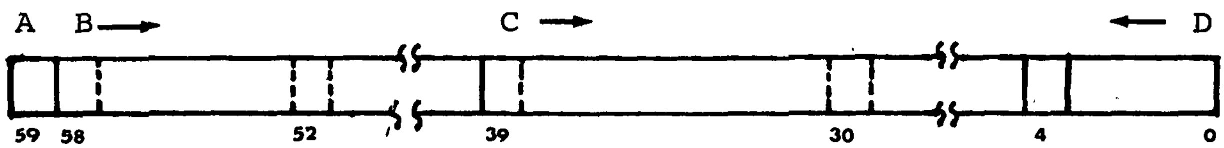
In the course of debugging the program with the new models included, the dimensions of certain of the plotting arrays not required for reliability computation, were temporarily reduced in size. The result was a 30K (octal) savings in required field length for execution. The user who wishes to obtain only tables of reliability can convert the complete program version by making the changes shown in table 4-3.

TABLE 4-2

PCW AND DEFAULTS

-ARRAY PCW (22)-

PCW contains packed information pertaining to the type and usage of each parameter in the reliability model. The fields within each word (one per parameter) are as follows:



<u>Field</u>	<u>Meaning if bit position is equal to one</u>
A	Parameter is represented in real format (as is default), otherwise it is in integer format
B_i	Parameter is used in the i^{th} reliability equation, and B_i is located at (59-i)
C_i	Parameter is used in the reliability equation associated with the i^{th} stage, and C_i is located at (40-i). (These bits are set by READIN for use by COVGEN & RITE2)
D	5 bit right justified integer giving the subscript of the default value in either INTDFS(4) or RLDFS(10), depending on field A (bit 59).

TABLE 4-2 (Cont.)

-Parameter Names, Defaults and Usage-

PCW SUBSCRIPT	NAME	DEFAULT	ARRAY POSITION	DEFAULT VALUE	EQUATION 1	2	3	4	5	6	7
1	Q=Q1	Q1DEFr	RLDFS(1)	2.0		X	X				X
2	Q2	Q2DEFr	RLDFS(2)	1.0							X
3	N	NDEFi	INTDFS(1)	3	X						
4	S	SDEFi	INTDFS(2)	0	X	X	X	X			X
5	W	WDEFi	INTDFS(3)	1	X	X	X	X	X	X	
6	Z	ZDEFi	INTDFS(4)	1	X	X	X	X	X	X	
7	LAM	LAMDEFr	RLDFS(3)	10^{-6}	X	X	X	X	X	X	X
8	MU	MUDEFr	RLDFS(4)	10^{-6}	X	X	X	X			X
9	GMP	GMPDEFr	RLDFS(5)	10^{-6}							X
10	C=CI	CDEFr	FLDFS(6)	1.0		X	X				X
11	CD1	CDEFr	RLDFS(6)	1.0							X
12	C2	"	"	"							X
13	CD2	"	"	"							X
14	CTR	CTRDEFr	RLDFS(7)	"							X
15	CDTR	"	"	"							X
16	PRC1	PRCDEFr	RLDFS(8)	"							X
17	PRC2	"	"	"							X
18	RV	RVDEFr	RLDFS(9)	"	X	X	X	X	X		
19	P	PDEFr	RLDFS(10)	0.5					X		
20	K		"	N/A	X	X	X	X			X
21	GM1		"	N/A							X
22	GM2		"	N/A							X

Note:

- other parameters (scalars) are printed when DCFLG = .TRUE.
- internal representation is denoted by subscript r for real and i for integer.

TABLE 4-3

DIMENSION CHANGES FOR REDUCED FIELD LENGTH CARE2

Subroutines affected:

* CARE2 (main subprogram)
* EQUAL
* NEQ7
* PLOTR
* PLOTRV
* PLOTT
* READIN
* RIFDIF

Arrays affected:

Complete version	Reduced version
R(121,25)	R(121,10)
DIFF(121,15)	DIFF(1,1)
RIF(121,15)	RIF(1,1)
GAIN(121,15)	GAIN(1,1)
ABSC(121,3)	ABSC(1,1)
G(121,16)	G(1,1)
XY(121,19)	XY(1,4)
RLRV(210,17	RLRV(1,2)
FDUM(3233)	FDUM(1)

SECTION 5

PROGRAM OPERATION

CARE2 is designed to run on a 60 bit CDC 6000 Series computer system, using the RUN Fortran compiler in batch mode, under either the KRONOS 2.1 or SCOPE 3.0 operating system. Source and data input are nominally provided on punched cards, and output is nominally produced on a line printer. However, both operating systems allow reassignment of input/output devices with little effort.

In modifying and adding to the Fortran source code, care was taken to avoid using features of extended Fortran versions which were not used in the original CARE program. Reliance on library routines was also held to a minimum. The SETBIT routine is no longer required, and the ISHIFT function is included in the source as a Fortran callable COMPASS program. Other library references are to long-time standards such as AMIN1, OR, etc..

In comparison with CARE, which requires about 110K octal field length to load and execute, CARE2 runs in slightly under 130K (the precise figures depend somewhat on the compiler and operating system used). The reduced version of CARE2 runs in about 100K.

5.1 USER'S GUIDE

This section, although intended for those familiar with the operation of CARE, provides the information required for compiling all the input data to model computer systems and coverage systems with CARE2. An input algorithm, in flowchart form, is included to simplify usage. The use of the Fortran Namelist feature and control cards on input facilitates sensitivity analyses, which is an expected mode of operation.

5.1.1 Processing Order

The general processing order of the program as a whole is as follows:

1) Input the computer configuration to be modelled, using one or more equation numbers to represent stages in series or, in the case of the dual mode model, a number of 7's to represent the stages within that model.
2) Input the upper limit of the independent variable and the step size to be used to generate reliability tables.
3) Input data to specify the desired computational and plotting options, changes to defaults (optional), and linkage data for the subsequent transfer of calculated coverage values to the reliability model date base.
4) Input non-default values for parameter base run vectors, coverage function selection and specification data, and special dual mode and coverage model variable data, as required.
5) Input iteration data, if any. (Enables the re-running of computer model evaluations with variations in a single parameter).
6) Print out selected parts of the coverage data base, if requested.

7) Compute any requested (and applicable) coverage factors, optionally displaying intermediate results (conditional D/I/R mechanism contributions).
8) Optionally print out parametric data to be used in reliability model evaluations, including calculated coverage and special variables.
9) Compute and print tables of reliability versus the selected independent variable, sequentially for each equation in the configuration list. If the dual mode model (equation 7) is evaluated, optionally print the mode 1 system and stage reliabilities versus time (ahead of the standard reliability table).
10) Compute product, MTF and other selected options, including plots.
11) Repeate steps 4 through 10, changing only the desired variant data. (Note that the iteration parameter and data will remain the same if not actively altered or defeated.)

5.1.2 Use of the Input Algorithm

Figure 5-1 is a working guide for the creation of an input deck. The requirements are unusually complex, partly due to the volume and variety of data, and partly since the input routine was originally written in a question-and-answer format for use in a time-sharing environment.

Each input record must of course contain the proper information in the proper format. Thus the algorithm, in flow chart form, covers all possible inputs in both READIN (invariant data) and READIN2 (variant data). It is not intended as a complete flow diagram for either of these subroutines.

In following the diagram, note that a decision block contains either the name of a logical variable or shows some test of a variable. Input blocks which refer to Namelist names (designated by $) have comments showing the names and dimensions of its contained variables.

Table 5-1 describes many of these variables in operational terms. The 12 possible control cards which initiate processing tasks in READIN2 are described in Table 5-2. Input blocks which show fixed format cards refer to format entries in Table 5-3.

FIGURE 5-1

INPUT ALGORITHM

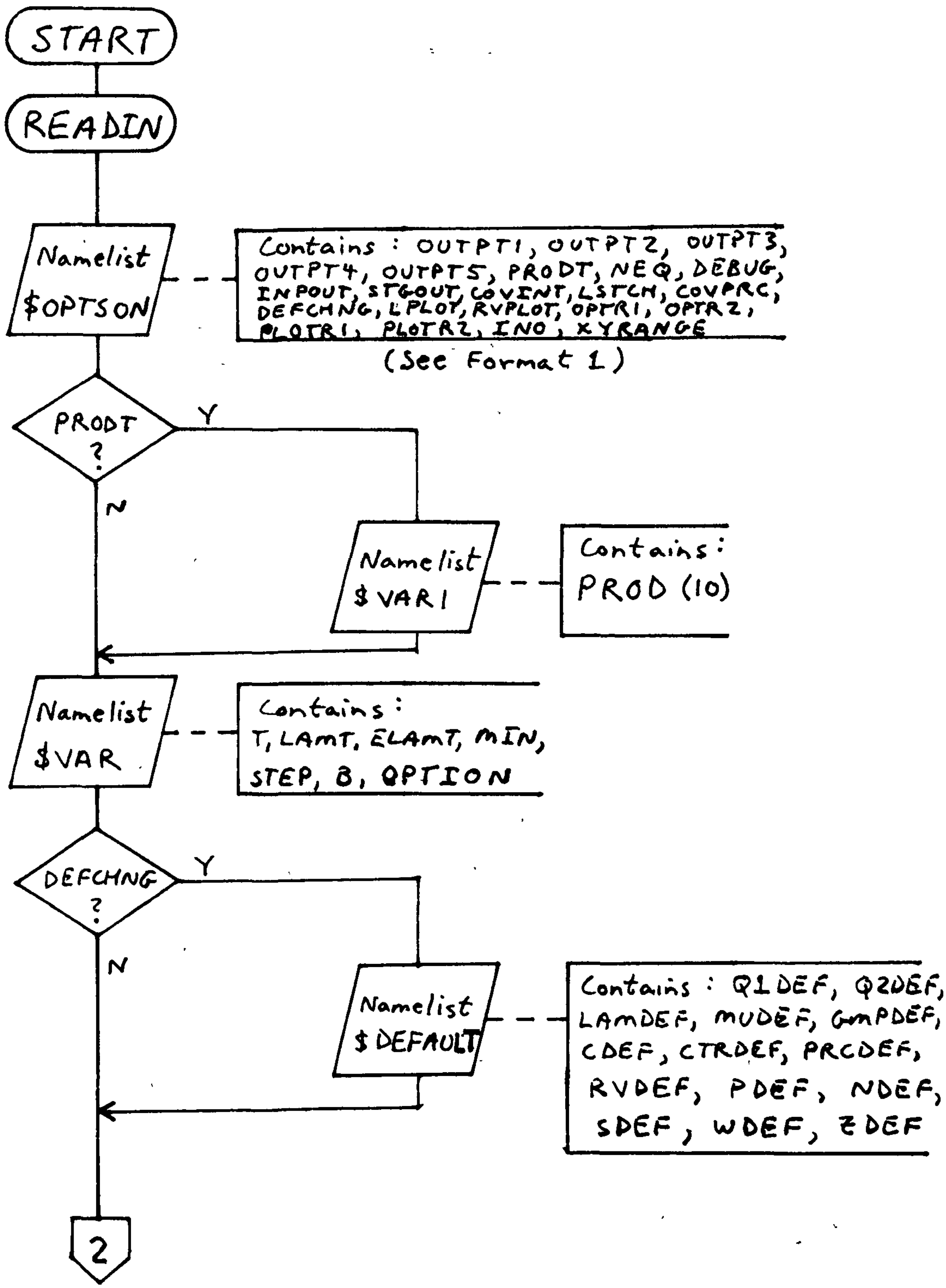


FIGURE 5-1 (Cont.)

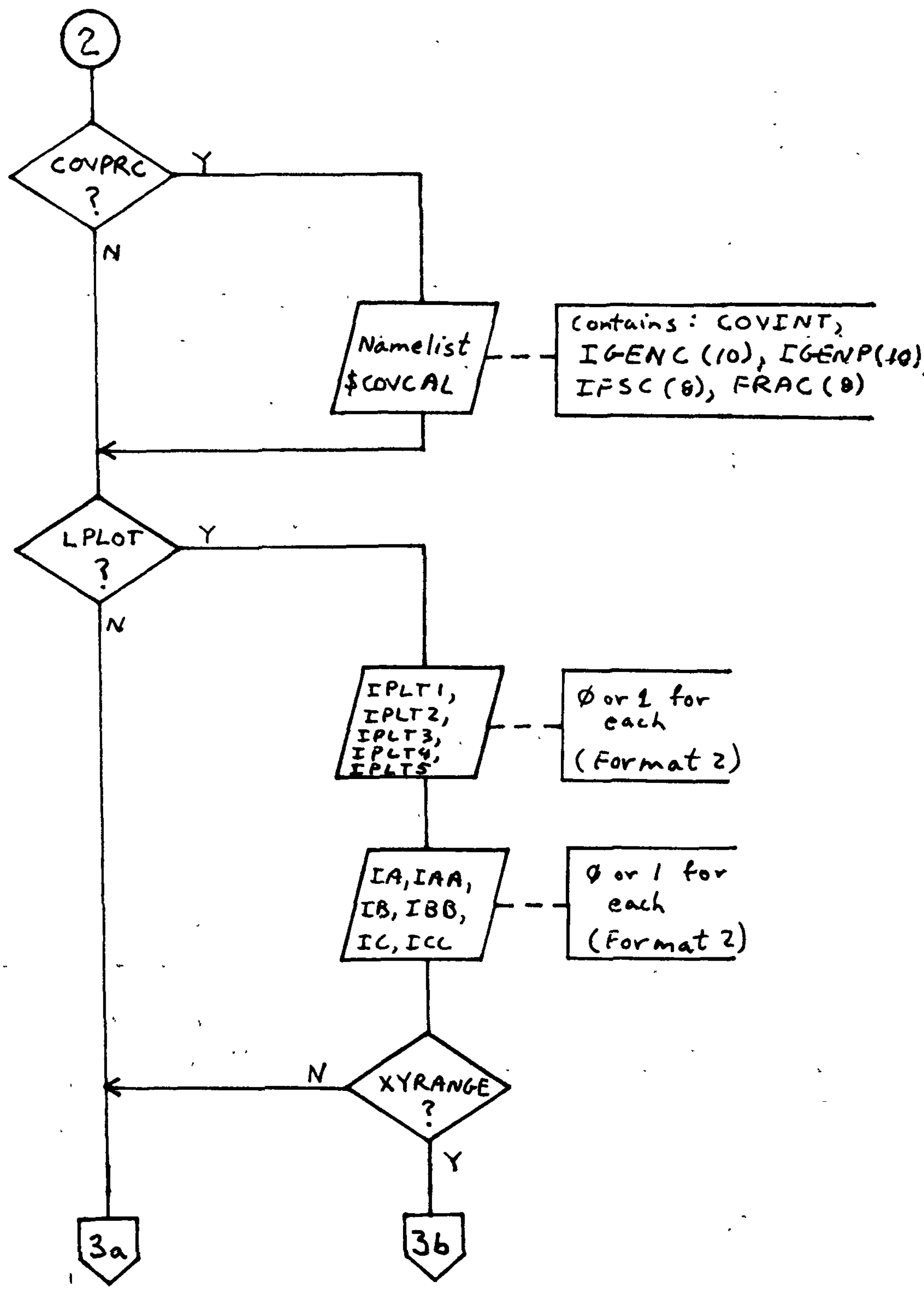


FIGURE 5-1 (Cont.-2)

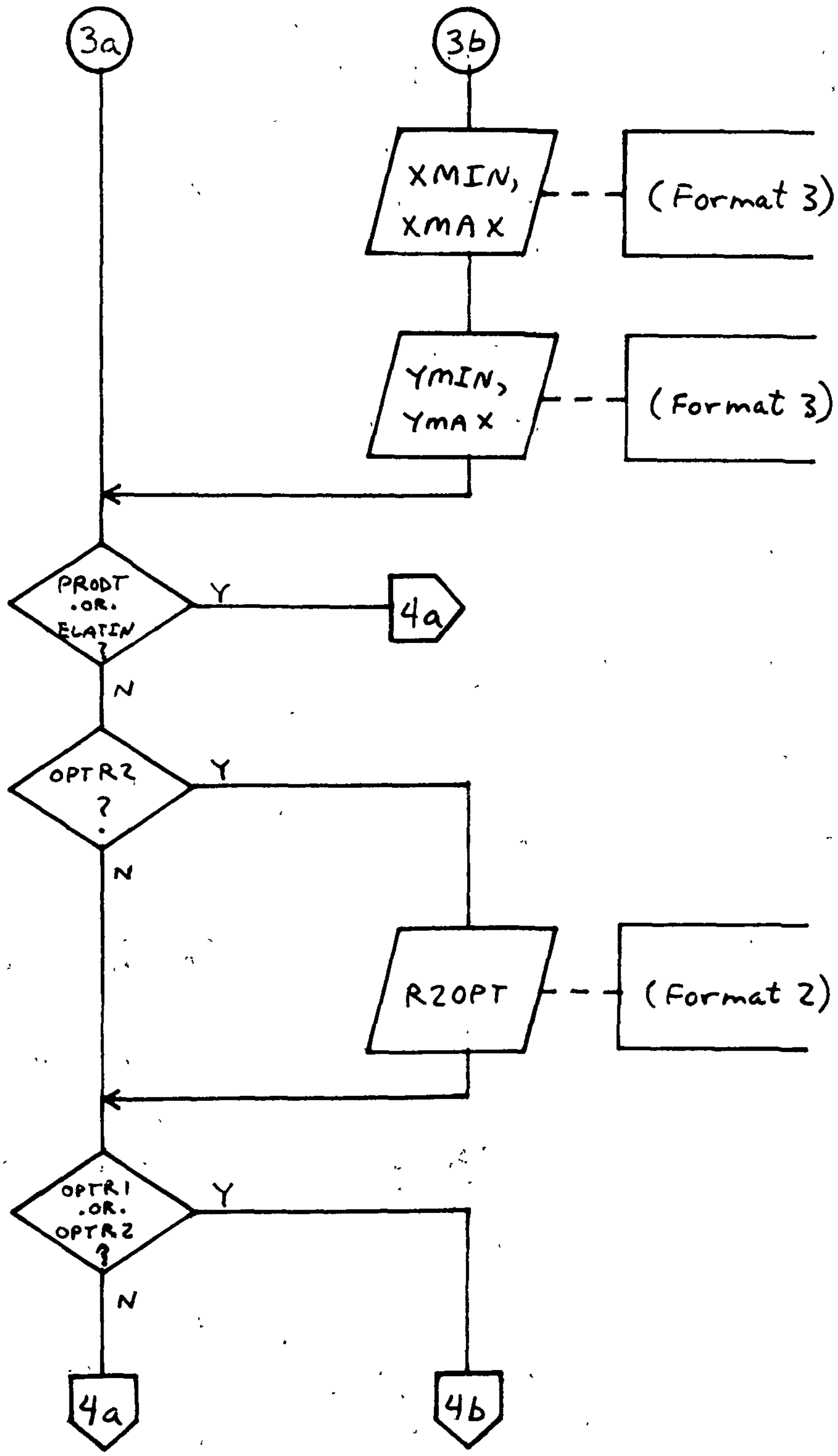


FIGURE 5-1 (Cont.-3)

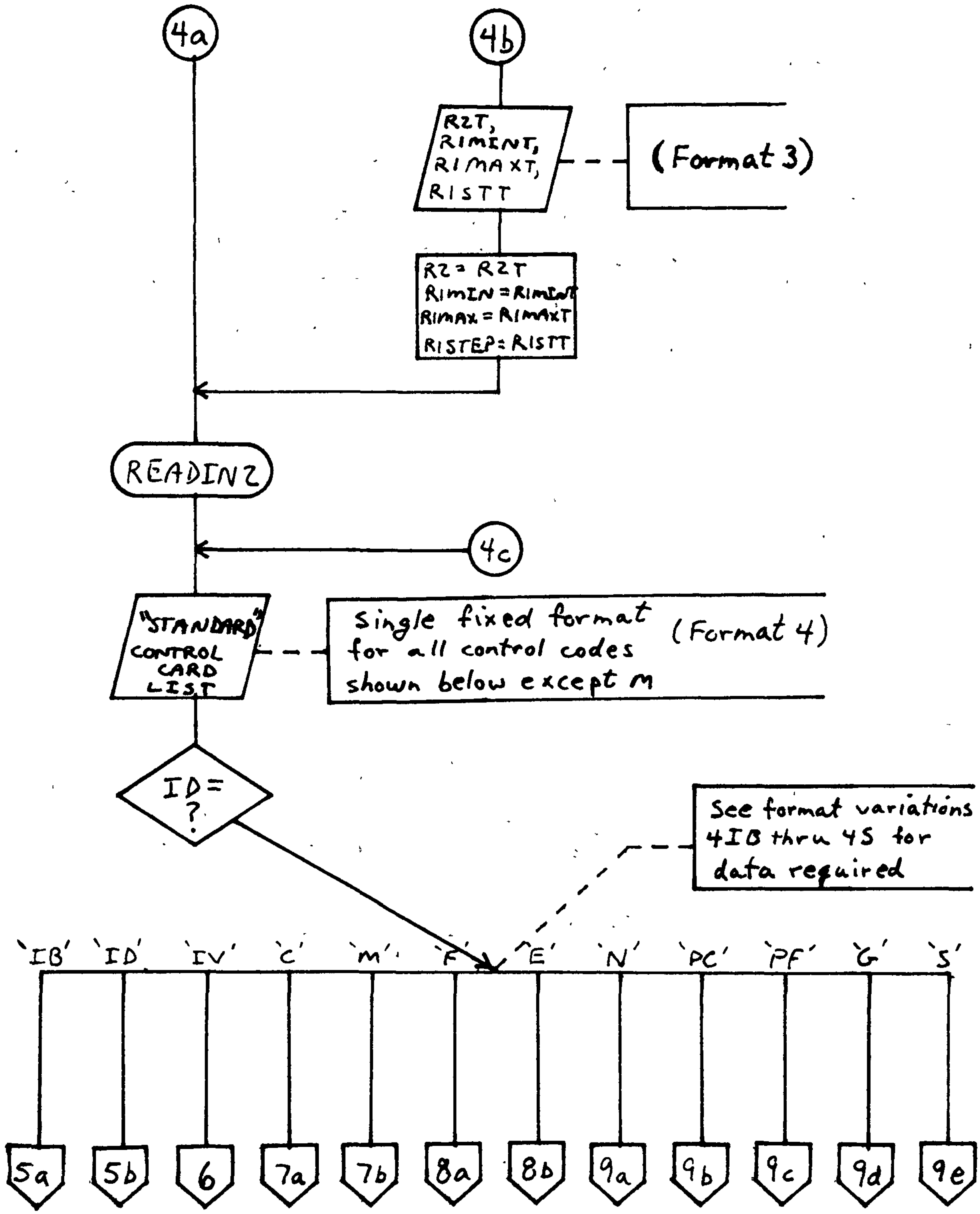


FIGURE 5-1 (Cont.-4)

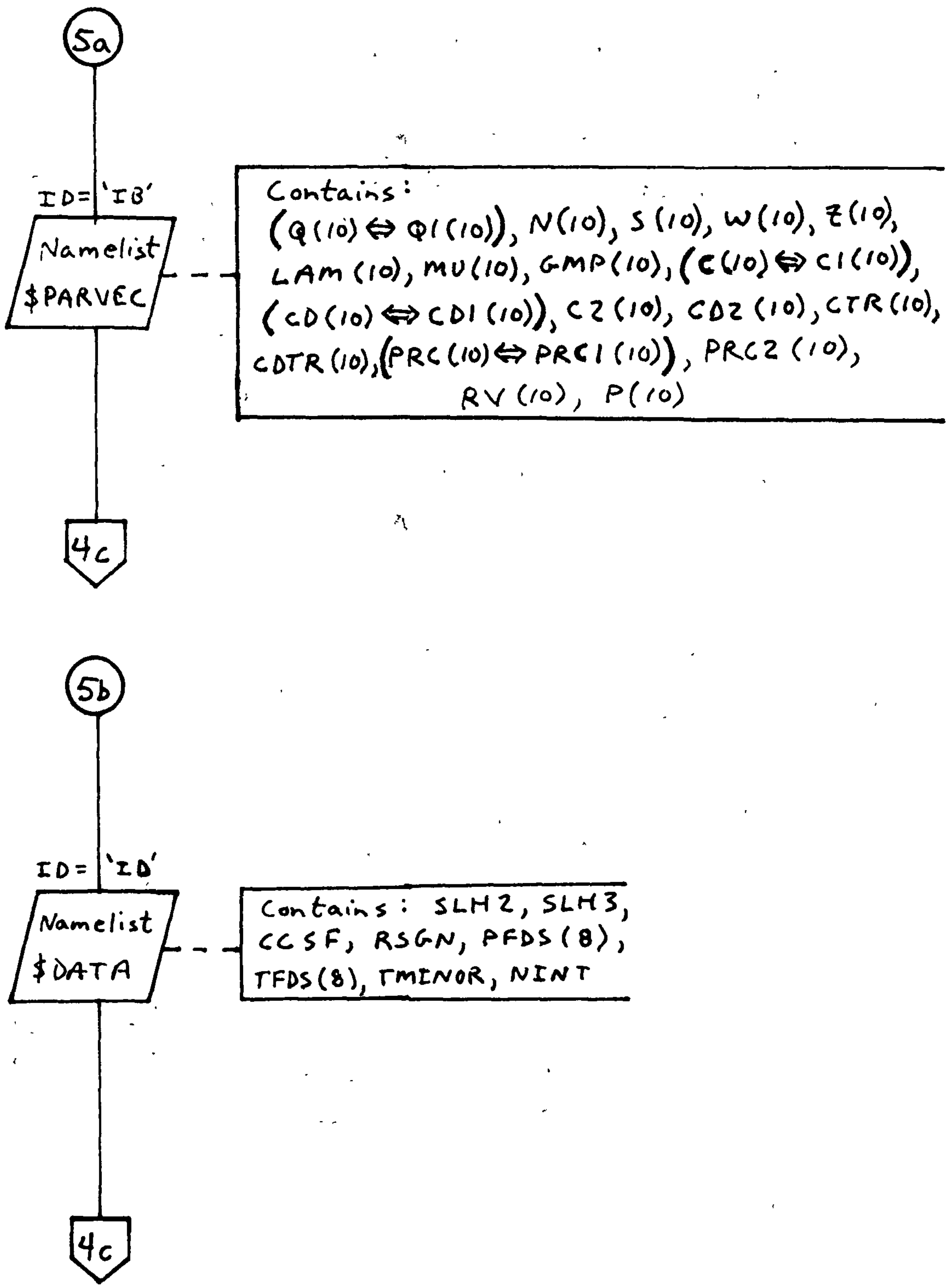


FIGURE 5-1 (Cont.-5)

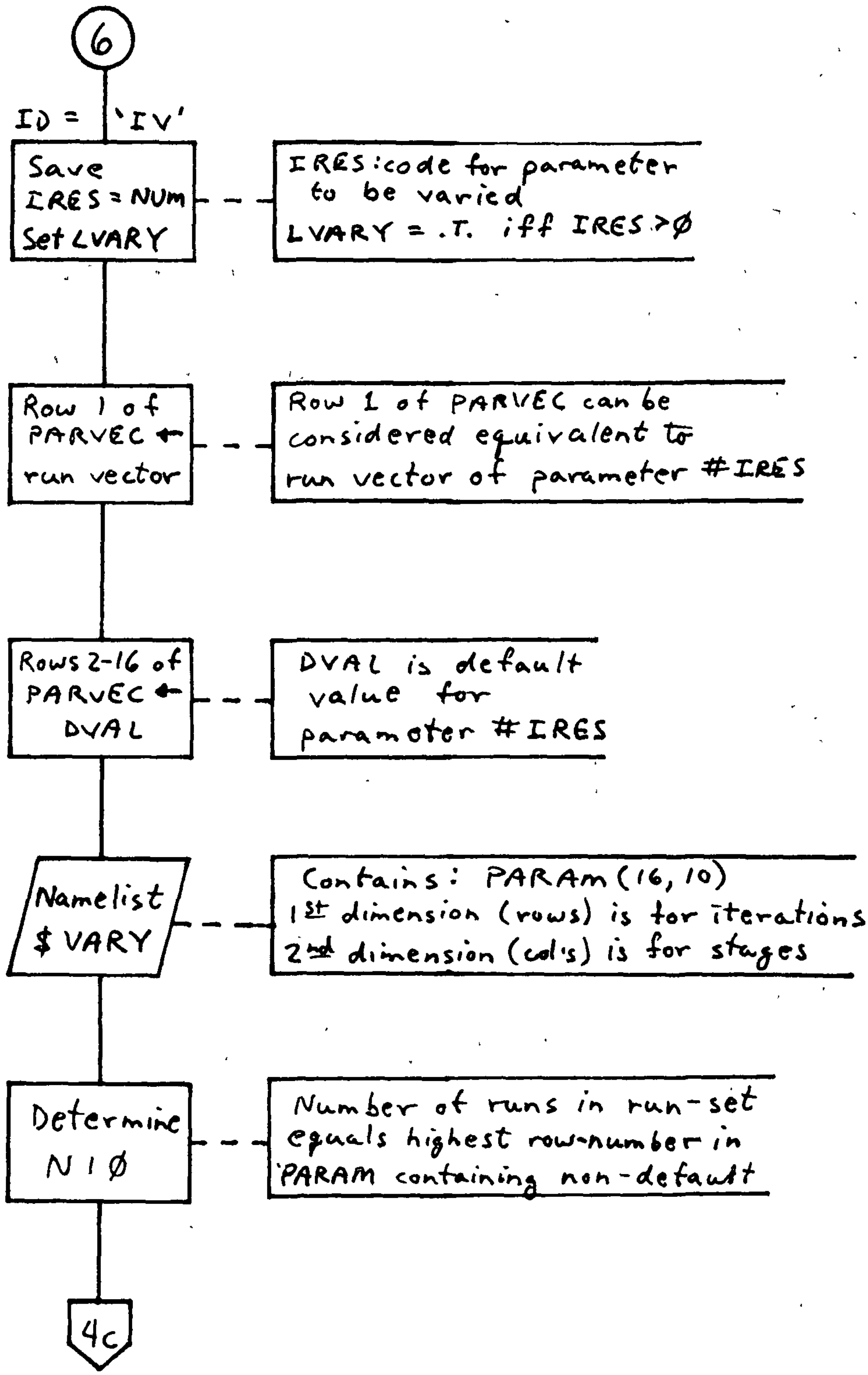


FIGURE 5-1 (Cont.-6)

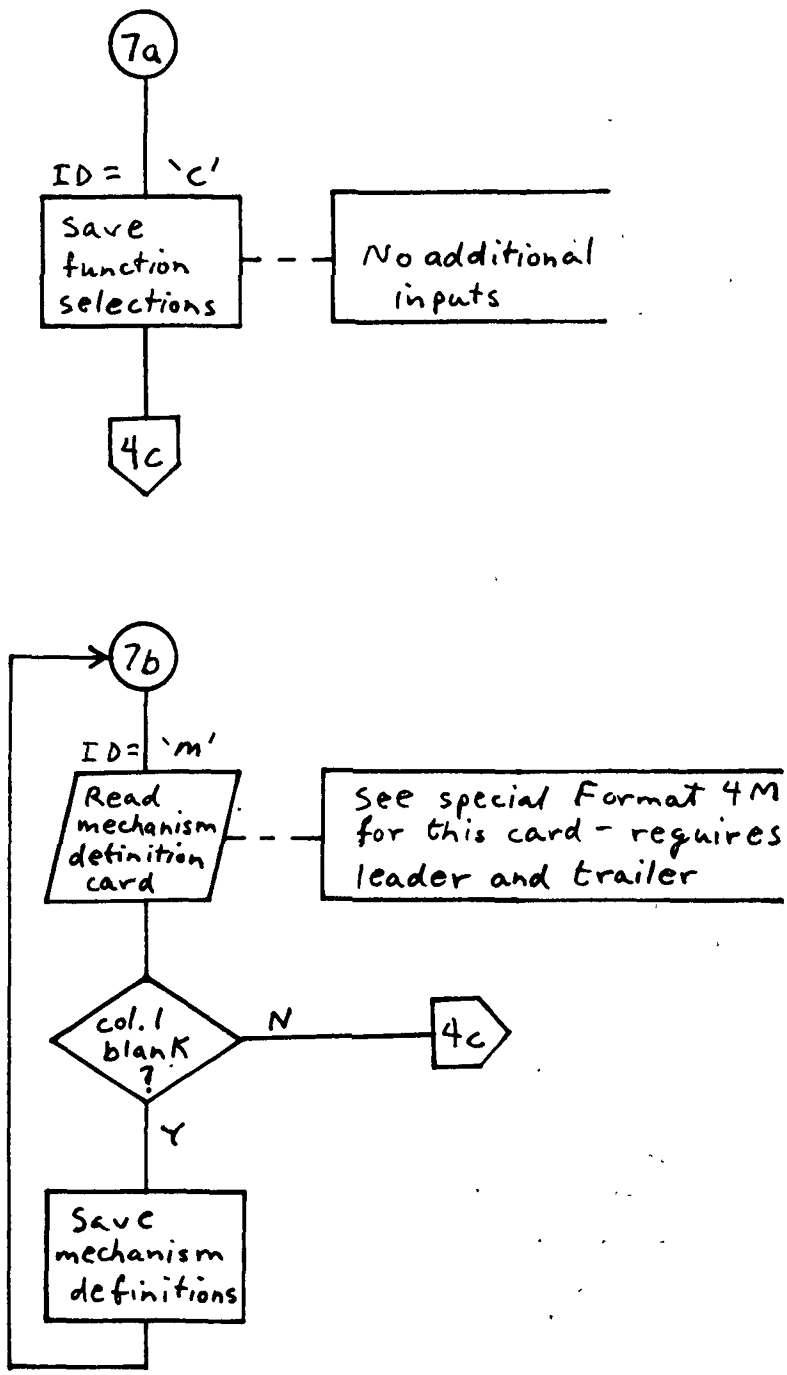


FIGURE 5-1 (Cont.-7)

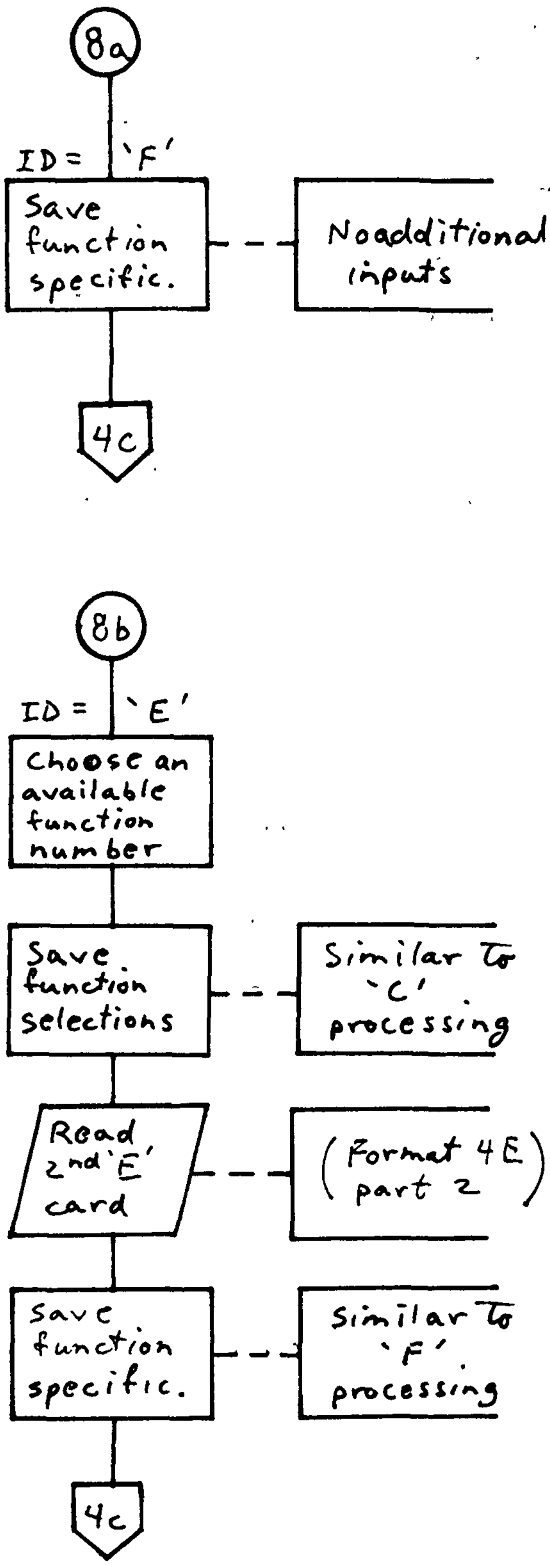


FIGURE 5-1 (Cont.-8)

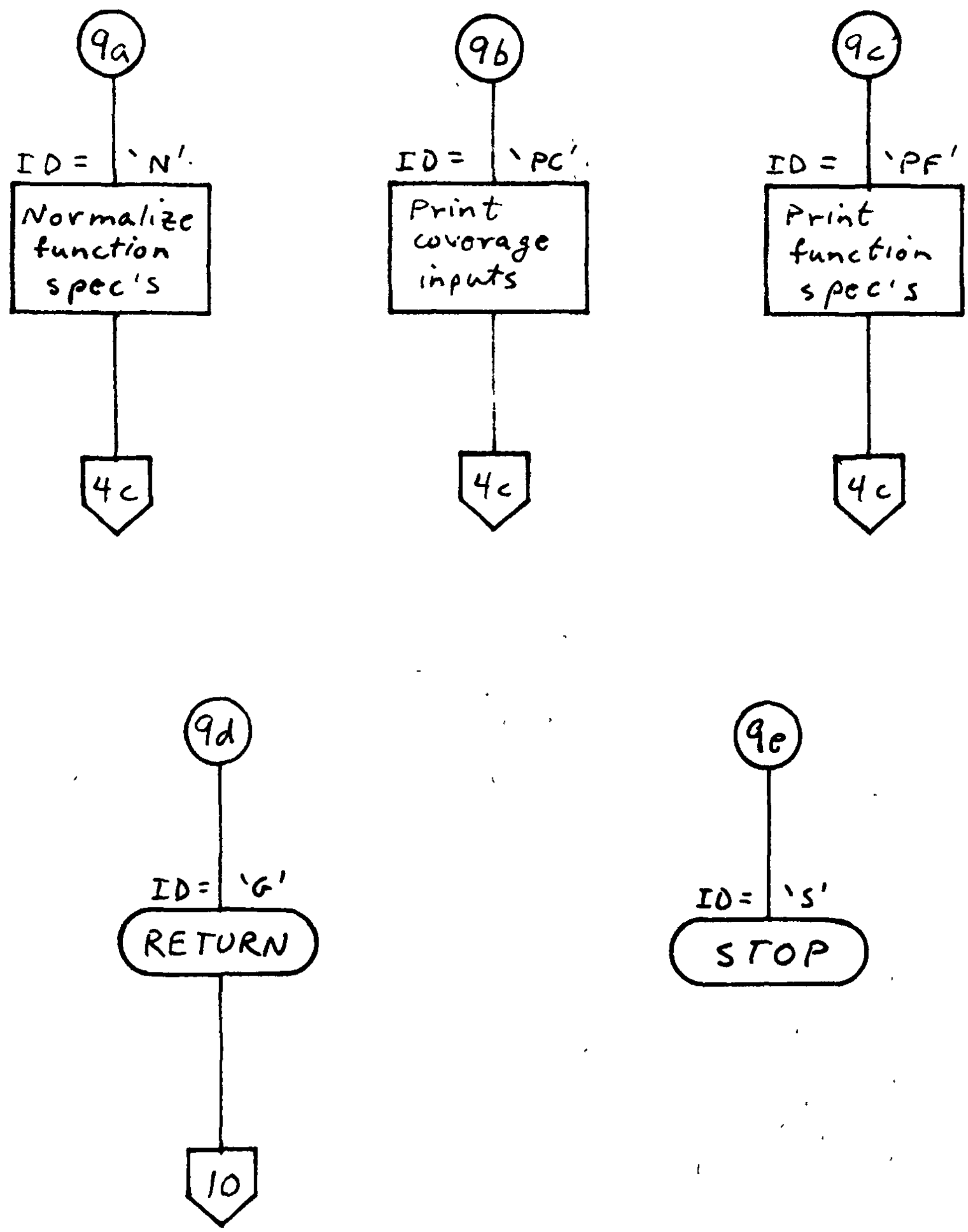


FIGURE 5-1 (Cont.-9)

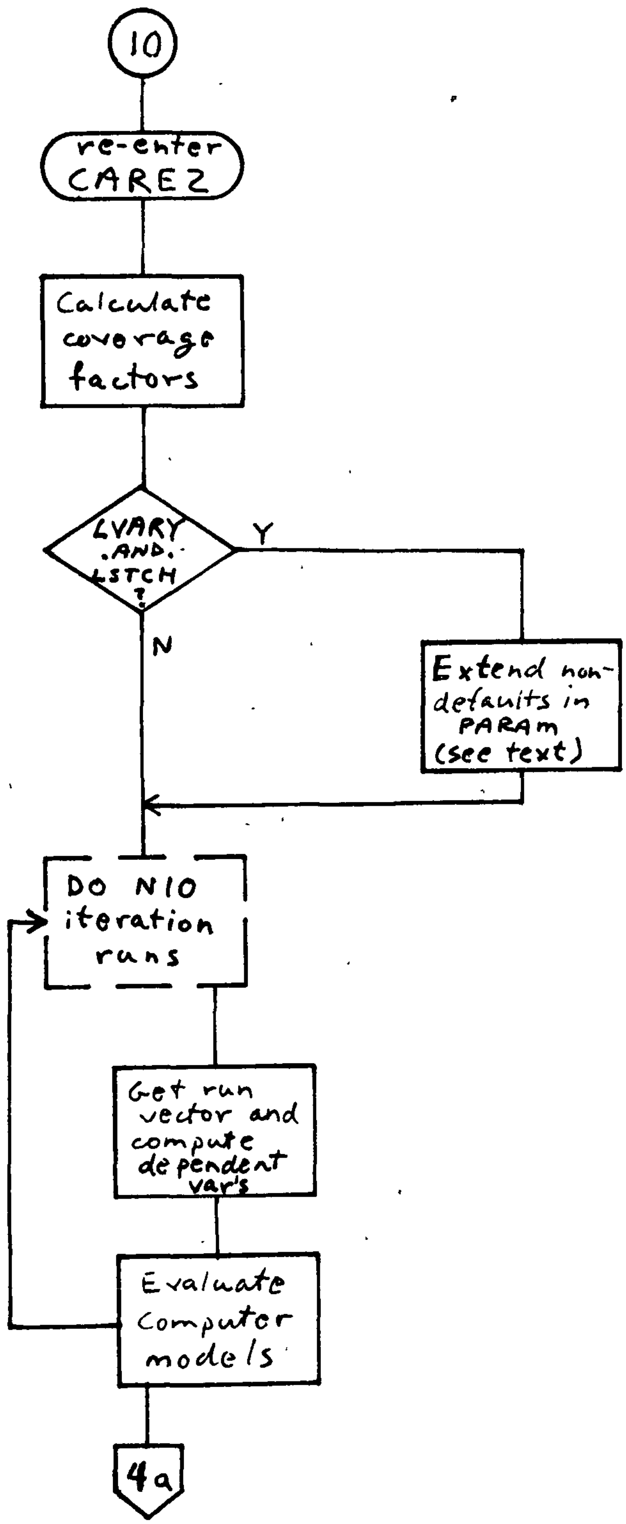


TABLE 5-1

DEFINITIONS OF LOGICAL AND CONTROL VARIABLES

PRODT	Logical: Specifies whether a list of equation numbers is desired rather than a single equation.
NEQ	Control: Specifies the single equation to be evaluated.
DEBUG	Logical: Specifies (if TRUE) that input format errors detected in READIN2 will not cause the job to abort, and that the reliability models will not be exercised.
INPOUT	Logical: Specifies (if TRUE) that a table of parameter values to be used for the subsequent run-set will be printed (regardless of the setting of DEBUG)
STGOUT	Logical: Specifies (if TRUE) that mode 1 reliability results for the dual mode model are to be printed.
COVINT	Logical: Specifies (if TRUE) that individual mechanism contributions are to be printed during the calculation of coverage.
LSTCH	Logical: Determines the convention used in the pre-processing of parameter iteration data in array PARAM. If TRUE, non-default elements replace default elements along the iteration dimension (those of increasing row subscript are replaced). For example, if the parameter default is 0, and a 1 is placed in PARAM (1,3), 0's in positions (2,3), (3,3), etc. will be replaced by 1's, up to the next non-default.
COVPRC	Logical: Specifies (if TRUE) that some coverage factors will be calculated (under the condition that subsequent requirements are met).
DEFCHNG	Logical: Specifies (if TRUE) that some default values for parameter vector initialization are to be changed.

TABLE 5-1 (cont.)

LPLOT	Logical: Specifies (if TRUE) that the reliability product table is to be plotted on the line printer.
PROD(10)	Control: The list of equation numbers for a product of reliabilities, to specify the number of stages in the dual mode model, or both.
T	Control: The upper limit for reliability tables when the independent variable is time. Time must be used when the dual mode model is evaluated.
LAMT	Control: The upper limit for reliability tables when the independent variable is failure rate * time.
ELAMT	Control: The upper limit for reliability tables when the independent variable is the exponential of (failure rate * time).
MIN	Control: The lower limit of the independent variable. This must be 0.0 when the dual mode model is evaluated.
STEP	Control: The increment of the independent variable.
ID	Control: The 1 or 2 letter code beginning in column 1 of a READIN2 control card, which identifies an input or processing request.
XYRANGE,OPTR2	See original CARE documentation.

TABLE 5-2

CONTROL CARDS

Identifier		Control Function
IB	-	Input the base-run parameter vectors (for reliability models) via namelist PARVEC
ID	-	Input system data via namelist DATA, specifically: SLH2, SLH3, CCSF, RSGN, PFDS, TFDS, TMINOR, and NINT.
IV	-	Input the variation parameter type code, followed by namelist VARY with changed values for array PARAM.
C	-	Read a subset of the D/I/R mechanism data base (function number arrays). One function number is specified for one process (detection, isolation, recovery (1 of 2)), but the number may be distributed over any or all of 4 dimensions: • subclass of faults (1 thru 8) • mode (0, 1 or 2) • error type (1 = permanent, 2 = transient) • mechanism (1 thru 20)
M	-	Begin reading D/I/R mechanism definitions (function number selections) at the rate of one mechanism (for one fault subclass) per card. The selections must completely define the characteristics of the 4 processes comprising the mechanism, for all conditions of mode and error type.
F	-	Read one function specification for one process (detection, isolation, recovery) of the recovery system. Function number and specifications use a common fixed format (based on the scheduled detection function), but fields which do not apply may be left blank.

TABLE 5-2 (Cont.)

Identifier	Control Function
E	- Read an explicitly defined subset of the D/I/R mechanism data base. This is a combination of the C and F control cards, in which the function number selection is automatic. The formats for the auxiliary information (on 2 cards) are identical to those of the C and F cards.
N	- Normalize the function specifications. The normalization involves all defined functions for all 4 processes. A trial integration is performed for detection and isolation rate functions over their specified non-zero ranges. The respective P1 values are then adjusted to obtain an integral value of 1.0 (P1 is assumed to be an internal coefficient of each function). The recovery probability functions are sampled over their specified ranges, and must be ≤ 1.0 at each sample point. Finally, the explicit coefficients for all functions (each of 4 specification arrays) are tested, and must be ≤ 1.0. Appropriate messages are printed during or after normalization.
PC	- Display the contents of the D/I/R mechanism data base. The function numbers selected to define recovery subsystems are printed, with conditions which may cause errors flagged with the letter X.
PF	- Display the specifications of all defined functions,
G	- Exit READIN2 and execute the run-set. Coverage and reliability models will be exercised if the DEBUG flag is not set. Otherwise, the requested coverage calculations are performed and the selected runset input data is printed (including coverage and iteration values), prior to re-entering READIN2.
S	- Stop the program (optional, equivalent to an end-of-file on input).

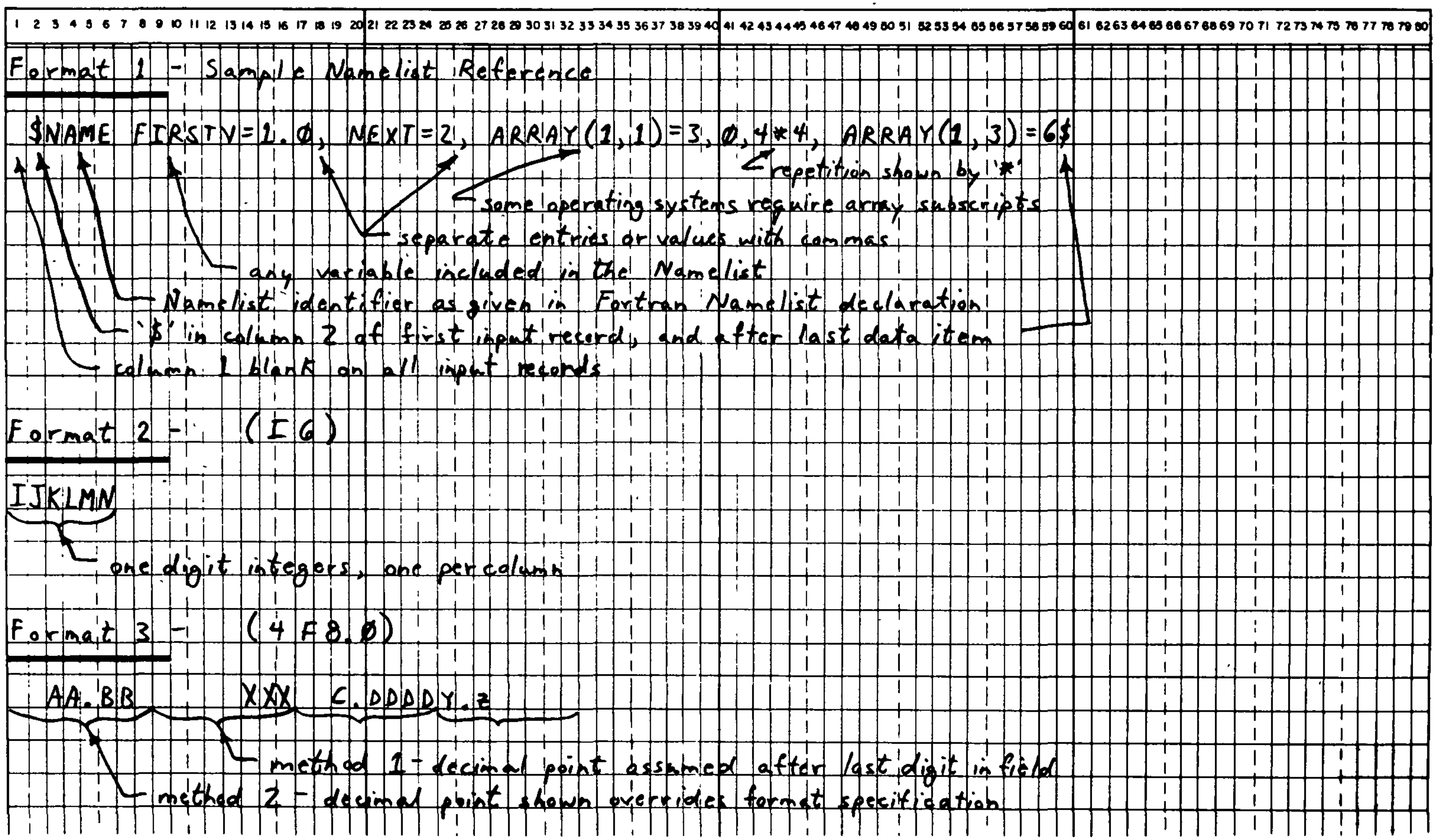


TABLE 5-3
INPUT FORMATS AND EXAMPLES

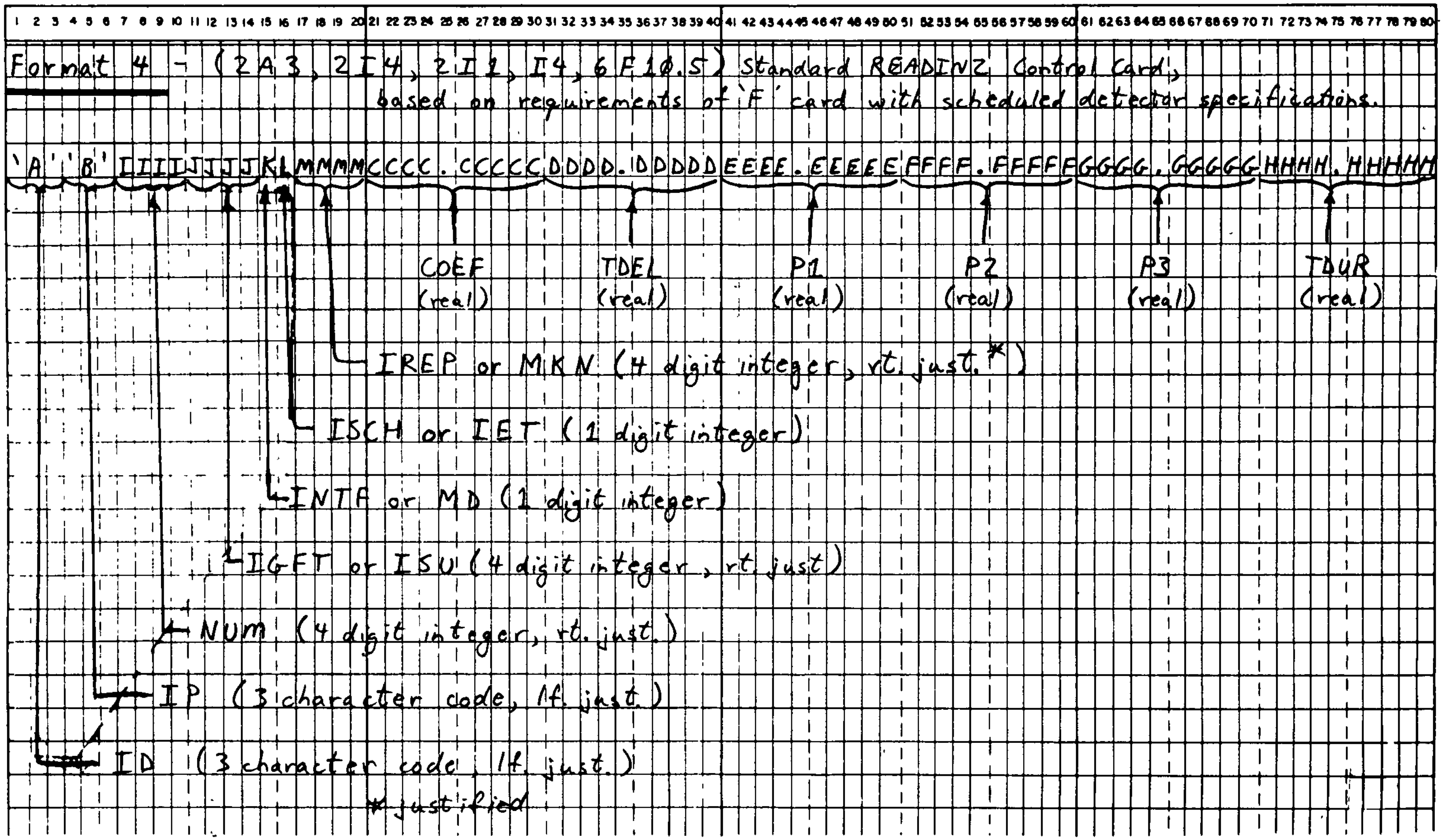


TABLE 5-3 (Cont.)

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

Format 4IB - Input base run vectors

'A'
└ 'IB' in col's 1 + 2

Format 4ID - Input system data

'A'
└ 'ID' in col's 1 + 2

Format 4IV - Input variation data

'A' IIII
└ Parameter vector to be replaced prior to each run. Specifically:

1 ⇒ Q=Q1 (10)	6 ⇒ Z (10)	11 ⇒ CD1(10)	16 ⇒ PRC1(10)
2 ⇒ Q2 (10)	7 ⇒ LAM (10)	12 ⇒ C2 (10)	17 ⇒ ARC2(10)
3 ⇒ N (10)	8 ⇒ MU(10)	13 ⇒ CD2(10)	18 ⇒ RV (10)
4 ⇒ S (10)	9 ⇒ GMP(10)	14 ⇒ CTR (10)	19 ⇒ P(10)
5 ⇒ W (10)	10 ⇒ C=C1(10)	15 ⇒ CDTR(10)	

└ 'IV' in col's 1 + 2

TABLE 5-3 (Cont. -2)

Format 4C – Input Function Selections (1 process)
+ 4E part 1

`'A' 'B' I I I J J J K L M M M M`

- M M M M — mechanism 1 thru 20, or > 20 for all mechanisms
- L — error type – 1 = permanent; 2 = transient, > 2 for both
- K — mode 1, 2, or Ø, or > 2 for all
- J J J — fault subclass 1 thru 8, or > 8 for all
- I I I — function number (for 'E' card, generated by program)
- 'B' — process code – 'D' for detection, 'I' for isolation, (col. 4) 'E' for error propagation recovery, 'T' for time lost recovery
- 'A' — 'C' in col. 1 (or 'E' in col. 1)

Format 4M – Input Function Selections (1 mechanism for one subclass per card)
– leader and trailer cards containing 'M' in col. 1 required. special format for data cards shown below

`ISU MKN D11 I11 E11 T11 D12 I12 E12 T12 D21 I21 E21 T21 D22 I22 E22 T22 D1Ø I1Ø E1Ø T1Ø`

- col. 1 blank
- not used
- ISU — fault subclass (rt. just.)
- MKN — mechanism (rt. just)
- D11 I11 E11 T11 — perm/mode 1
- D12 I12 E12 T12 — perm/mode 2
- D21 I21 E21 T21 — transient/mode 1
- D22 I22 E22 T22 — transient/mode 2
- D1Ø I1Ø E1Ø T1Ø — perm/mode Ø

function numbers are input above as 3 digit right justified integers
20 function selections define characteristics of mechanism MKN operating on faults of subclass ISU.

TABLE 5-3 (Cont. -3)

Format 4F – Input Function Specifications (1 process)
4E part 2

`A' `B' IIIIJJJJKLMMMMCCCC.CCCCCDDDD.DDDDDEEEE.EEEEEFFFF.FFFFFGGGG.GGGGGHHHH.HHHHH

- explicit coefficient
- explicit delay (0 for rec. fn's)
- first arb. parameter
- second arb. parameter
- third arb. parameter
- function duration
- repetition factor — scheduled detectors only
- scheduling code, =0 for unsched., =1 for sched detectors — detectors only
- integral model defined code, =0 for undef., =1 for defined — finite detectors only
- function model number (and integral model number), =0 for impulse — all functions*
- function number assignment (not required for E card) — all functions
- process code – `D' for detection, `I' for isolation, (col. 4) `E' for error propagation recovery, `T' for time lost recovery (not required for E card).
- `F' in col. 1 (not required for E card)

* detectors and isolators only may be impulse functions

Formats 4N, 4PC, 4PF, 4G, and 4S require only the identifying code in column 1 or col's 1 and 2:

`A'
- control card identifier `N', `PC', `PF', `G' or `S'

TABLE 5-3 (Cont. -4)

5.1.3 Specific Input Information

Inputs for the dual mode model are described in Sections 2.3 and 3.3, and are relatively easy to comprehend. For reason of its greater flexibility, coverage model input requirements are considerably more complex, and thus additional learning aids might well prove useful.

In particular, it is felt that preliminary experimentation, by the user, may be required in order to become sufficiently familiar with the tool to use it effectively. For this reason, a set of input values is provided (cf. Tables 5-4 and 5-5) which reflect the coverage system described in Section 2.2.

Table 5-4 gives specifications for the above referenced detection, isolation and recovery characteristics, which, in turn, may then be applied to measure the effects of these processes on different subclasses of faults, under various conditions, e.g., permanent versus transient faults, mode 1 versus mode 2 operation, etc.. Table 5-6 defines the individual variables which comprise the specification of a single function, (cf. Format 4F in Table 5-3 for individual specification requirements of the 4 processes) as well as the other important variables in the coverage model.

Table 5-5 gives the function selection data for a complete system of fault coverage. For example, the 4 numbers in the upper right hand corner of the table (under the heading "Permanent Fault/ Mode O") indicate the non-competitive (i.e., stand-alone) effectiveness of detector #2 and its associated isolation and recovery processes, under a particular set of specified conditions (i.e., that a permanent failure has occurred in the stage linked with fault subclass 1, and that the recovery is predicted on degenerating from mode 1 to mode 2 operation (shown as mode O)).

TABLE 5-4

FUNCTION SPECIFICATIONS

FUNCTION	#	IGFT	ISCH	IREP	INTF	COEF	TDEL	A≡P1	B≡P2	C≡P3	TDUR
DETECTION (D)	1	0	0	-	1	1	0	-	-	-	-
"	2	0	0	-	1	0.25	0	-	-	-	-
"	3	0	0	-	1	0.1	0	-	-	-	-
"	4	0	1	1	1	0.05	19.3	-	-	-	-
"	5	0	0	-	1	0.025	0	-	-	-	-
"	6	0	0	-	1	.9844	0	-	-	-	-
"	7	0	1	1	1	0.05	19.7	-	-	-	-
"	8	0	1	1	1	1	19.7	-	-	-	-
"	9	3	0	-	1	1	0	.0910	0.09	0	50
"	10	3	1	1	1	0.95	0	.7218	0.7	0	5
"	11	1	1	1	1	0.8	5	0.5	-	-	2
"	12	1	1	1	1	0.05	5	0.5	-	-	2
"	13	1	1	1	1	0.2	5	0.5	-	-	2
"	14	1	1	1	1	0.05	0	0.2	-	-	5
"	15	1	1	1	1	0.2	0	0.2	-	-	5
"	16	1	0	-	1	0.1	0	0.1	-	-	10
"	17	1	0	-	1	0.25	0	0.1	-	-	10
"	18	0	1	4	1	1	19.3	-	-	-	0
"	19	0	1	4	1	0.8	19.3	-	-	-	0
ISOLATION (I)	1	0	-	-	1	1	0.5	-	-	-	-
"	2	1	-	-	1	0.99	0	0.04	-	-	25
"	3	0	-	-	1	0.8	20	-	-	-	-
RECOVERY (E)	1	1	-	-	1	1	-	1	-	-	10,000
"	2	3	-	-	1	.1	-	1	.0063	0	200
RECOVERY (T)	1	1	-	-	1	.9999	-	1	-	-	10,000
"	2	3	-	-	1	.9999	-	1	0.023	-	200

Note: column heading mnemonics are defined on the second sheet of Table 5-6.

TABLE 5-5
FUNCTION SELECTIONS - BY FAULT SUBCLASS

Mechanism Number	Permanent Fault/Mode 1	Permanent Fault/Mode 2	Transient Fault/Mode 1	Transient Fault/Mode 2	Permanent Fault/Mode 0 (Transitional)
	D I E T	D I E T	D I E T	D I E T	D I E T
Subclass 1					
2	9 2 1 2	0 ---*	9 3 1 2	0 ---	9 2 1 2
3	11 1 1 1	11 1 2 2	0 ---	0 ---	11 1 1 1
5	2 1 1 1	2 1 1 2	2 1 1 1	2 1 1 2	2 1 1 1
6	3 1 1 1	3 1 1 2	3 1 1 1	3 1 1 2	3 1 1 1
Subclass 2					
2	9 2 1 2	0 ---	9 3 1 2	0 ---	9 2 1 2
3	12 2 1 1	12 2 2 2	0 ---	0 ---	12 2 1 1
4	10 1 1 1	10 1 2 2	0 ---	0 ---	10 1 1 1
5	16 2 1 1	16 2 1 2	16 3 1 1	16 3 1 2	16 2 1 1
6	17 2 1 1	17 2 2 2	17 3 1 1	17 3 2 2	17 2 1 1
7	2 1 1 1	2 1 1 2	2 1 1 1	2 1 1 2	2 1 1 1
8	4 2 1 1	4 2 2 2	0 ---	0 ---	4 2 1 1
9	5 1 1 1	5 1 1 2	5 3 1 1	5 3 1 2	5 1 1 1
10	5 1 1 1	5 1 1 2	5 3 1 1	5 3 1 2	5 1 1 1
12	7 2 1 1	7 2 2 2	0 ---	0 ---	7 2 1 1
Subclass 3					
1	1 1 1 1	1 1 1 2	1 1 1 1	1 1 1 2	1 1 1 1
2	9 2 1 2	0 ---	9 3 1 2	0 ---	9 2 1 2
3	12 2 1 1	12 2 2 2	0 ---	0 ---	12 2 1 1
4	14 2 1 1	14 2 2 2	0 ---	0 ---	14 2 1 1
5	16 2 1 1	16 2 1 2	16 3 1 1	16 3 1 2	16 2 1 1
8	18 1 1 1	18 1 2 2	18 3 1 1	18 3 2 2	18 1 1 1
9	5 1 1 1	5 1 1 2	5 3 1 1	5 3 1 2	5 1 1 1
10	5 1 1 1	5 1 1 2	5 3 1 1	5 3 1 2	5 1 1 1
12	7 2 1 1	7 2 2 2	0 ---	0 ---	7 2 1 1
Subclass 4					
2	9 2 1 2	0 ---	9 3 1 2	0 ---	9 2 1 2
3	13 2 1 1	13 2 2 2	0 ---	0 ---	13 2 1 1
4	15 2 1 1	15 2 2 2	0 ---	0 ---	15 2 1 1
5	16 2 1 1	16 2 1 2	16 3 1 1	16 3 1 2	16 2 1 1
6	3 1 1 1	3 1 1 2	3 1 1 1	3 1 1 2	3 1 1 1
8	19 1 1 1	19 1 2 2	0 ---	0 ---	19 1 1 1
9	5 1 1 1	5 1 1 2	5 3 1 1	5 3 1 2	5 1 1 1
10	5 1 1 1	5 1 1 2	5 3 1 1	5 3 1 2	5 1 1 1
11	6 1 1 1	6 1 1 2	6 1 1 1	6 1 1 2	6 1 1 1
12	8 1 1 1	8 1 2 2	0 ---	0 ---	8 1 1 1
Subclass 5					
3	11 1 1 1	11 1 2 2	0 ---	0 ---	11 1 1 1
6	3 1 1 1	3 1 1 2	3 1 1 1	3 1 1 2	3 1 1 1

*0 --- represents a null contribution to coverage. Accordingly, when setting up program input data, no entry need be provided.

TABLE 5-6

COVERAGE MODEL PARAMETERS

COVAGE Arguments

- ISU - Fault subclass number (1-8)
- MD - Mode of operation:
  - 1 for full up
  - 2 for degenerate
  - 0 for transitional (from MD = 1 to MD = 2)
- IET - Fault type code:
  - 1 for permanent
  - 2 for transient
- JS - Number of spare LRU's which must be checked out before recovery can proceed. (normally 0 or 1).

Basic Working Variables

- Function Pointer Arrays (Common Block CVB1).
  - NDET(20,8) - Detection rate function numbers
  - NISO(20,8) - Isolation rate function numbers
  - NEPR(20,8) - Error propagation recovery function numbers
  - NTLR(20,8) - Time lost recovery function numbers

  where:
    - o 1st subscript is D/I/R mechanism
    - o 2nd subscript is fault subclass number
- Function Specification Arrays (Common Block CVB2)
  - FDET(7,200) - Detection function specifications
  - FISO(7,50) - Isolation function specifications
  - FEPR(7,25) - Error propagation recovery function specifications
  - FTLR(7,25) - Time lost recovery function specifications

  where:
    - o 1st subscript is word within a specification
    - o 2nd subscript is function number within the specification array.

TABLE 5-6 (Cont.)

Function Specification Element Descriptors

- - CODES - (Word #1) - contains 5 packed integer codes:
  - o IGFT - Function Model Identifier (specifies a characteristic curve), (0 = impulse, 1 = constant, 2 = pulse train, 3 = decaying exponential), (0 not applicable for recovery)
  - o ISCH - Scheduling code (0 = no, 1 = yes), (detectors only)
  - o IREP - Repetition factor in minor cycles, (scheduled detectors only)
  - o IDEF - Function defined flag (0 = no, 1 = yes), (set automatically when specifications are input)
  - o INTF - Integral defined flag (0 = no, 1 = yes).
  - COEF - (Word #2) - Explicit function multiplier
  - TDEL - (Word #3) - Delay before the function becomes non-zero (must be zero for recovery functions)
  - P1, P2, P3 - (Word #4, 5, 6) - Arbitrary parameters passed to the function model for evaluation. Note that if automatic normalization is requested by the user, all function models used by detection or isolation functions must use P1 as the normalizing coefficient
  - TDUR - (Word #7) - Time during which the function is non-zero (must be zero for impulse functions).

- Computer System Dependent Variables (Common Blocks CVBO, CVB5 and CVB6)
  - PFDS(8) - Probability of detecting a failure in a spare LRU during initial checkout, for each of 8 fault subclasses
  - TFDS(8) - Time delay incurred in checking out a spare LRU, for each of 8 fault subclasses

TABLE 5-6 (Cont. -2)

- LCM - Least common multiple of the ni's where ni is the repetition factor (where applicable) of the ith detector, expressed in terms of minor cycles, and automatically generated by CARE2
- TMINOR - Duration of a minor cycle, in units compatible with all rate functions
- IFSC(8) - Reliability model stage in which the a fault subclass is located, for each of 8 subclasses
- FRAC(8) - Fraction of total stage (class) faults which occur in a fault subclass, for each of 8 subclasses. (Used in conjunction with IFSC to specify the relationship between subclasses and stages, thus enabling calculation of stage coverage values on the basis of weighted average subclass coverages)

COVAGE Call Level Intermediate Variables

- NDSAV(20) - Detector rate function numbers
- NISAV(20) - Isolation rate function numbers
- NR1SAV(20) - Error Propogation recovery function numbers
- NR2SAV(20) - Time lost recovery function numbers
- ICHAR(20) - Characteristic of detection/recovery mechanism:
  - 0 if mechanism is inoperative
  - 1 if detector is unscheduled - impulse
  - 2 if detector is unscheduled - finite
  - 3 if detector is scheduled-impulse
  - 4 if detector is scheduled-finite
- IREPAR(20) - Repetition rate of detector (if applicable)

TABLE 5-6 (Cont. -3)

- PERIAR(20) - Period of repetition (if applicable) of detection functions
- TDELAR(20) - Delay times for detection functions
- PDIRAR(20) - Non-competitive probability of detection, isolation, and recovery, given that JS spares must be checked before recovery can proceed
- INDEX - Function of arguments IET and MD as follows:

IET	MD	INDEX
-	0	5
1	1	1
1	2	2
2	1	3
2	2	4

- GPFLG - Flag indicating the meaning of array CVGP:
  when .FALSE., GPAR = g' (i,τ)
  when .TRUE., GPAR = G' (i,τ)

<u>D/I/R Evaluation Level Intermediate Variables</u>

- IMPG (logical) - = .TRUE. when detector is an impulse
- IMPH (logical) - = .TRUE. when isolator is an impulse
- GPAR(N) - one row of the GPAR(20,N) array, which is alternately used to hold values of $g'(i,\tau)$ and $G'(i,\tau)$
- INDEX - (constant for COVAGE call) - function of MD and IET
- TGFIN - Last time at which g(t) is non-zero
- THFIN - Last time at which h(t') is non-zero
- TR1FIN - Last time at which $r'(\tau')$ is non-zero
- TR2FIN - Last time at which $r'(\tau + \tau')$ is non-zero

5.2 OPERATIONAL LIMITS

The numerical limits of CARE2 are shown in Table 5-7. Most of these are self explanatory, however particular attention is directed to the last item, entitled numerical integration steps. As noted, there are three independent step-size controls, applicable to the various integration algorithms, all of which use the Simpson 3-point parabolic integration scheme. (This standard numerical integration method was chosen because of its affinity for exponential curves).

For the dual mode model, the integrations performed in subprograms DCT and DCT2 divide the integrand into 3-point Simpson's Rule segments, with a base width of STEP time units. Optimal condition accuracy tests performed here show minor variation in the 8th decimal digit, given a T/STEP ratio of 10 or more.

The integrations performed in COVAGE, and its secondary subprograms, use a fixed number of 3-point segments controlled by NINT, and pre-determine the non-zero portion of the integrands for optimum efficiency. Since new function models, of any form, may be added to the current set (cf. section 4.2.4.21), it necessarily falls to the user to determine their accuracy. This may be accomplished by experimenting with NINT, which may be set at run time via control card ID in READIN2.

The last integration control is variable N in FINTEG (the coverage characteristic curve (function) integrator). N may be altered by recompiling FINTEG.

The user is encouraged to provide a corresponding integral model for each function model that is added, thereby precluding the need for numerical integration (cf. Section 4.2.4.15). Otherwise, the corresponding function specifications must be tagged as "integral not defined" (INTF=0), and the attendant inaccuracies of an additional integration contended with (cf. INTF discussion in section 4.2.4.9).

TABLE 5-7

OPERATIONAL LIMITATIONS

Item	Upper Limit	Comment
Stages per equation type		
- equations 1 thru 6	1	
- equation 7	8	
Stages per system	10	
Total Equations of type		
- 1 thru 6 per system	10	
- 7 per system	1	
Fault classes/subclasses		
- per stage (equations 1,4,5,6)	0	
- per stage (equations 2,3,7)	8	
- per system	8	
Fault types		
- equations 1 thru 6	1	(permanent)
- equation 7	2	(permanent and transient)
Fault rates per stage		
- equations 1,2,3,4	2	(on-line and standby)
- equations 5,6	1	(on-line)
- equation 7	3	(on-line,standby and transient)
Modes		
- equations 1,2,3,5,6	1	(full-up)
- equation 4	2	(triplex, simplex)
- equation 7	2	(full-up, degenerate)
Calculated Coverage types		
- equations 1,4,5,6	0	
- equations 2,3	1	(permanent)
- equation 7	5	(permanent modes 1 and 2, transient modes 1 and 2, transitional)

TABLE 5-7 (Cont.)

Item	Upper Limit	Comment
Detectors per fault subclass		
- equations 1,4,5,6	0	
- equations 2,3,7	20	
D/I/R mechanisms per system	20	
Function specifications per system		
- Detection	200	
- Isolation	50	
- Recovery type E	25	
- Recovery type T	25	
Function models	4096	(4 currently defined)
Variable parameters per function model	3	
Time steps	121	
Parameter variations in run-set	16	
Run-sets per batch run	No limit	
Finite K	$<10^5$	(higher value, specified as λ/μ, calls $K=\infty$ model)
Numerical integration steps		
- Equation 7	2*[T/STEP]	
- Coverage model	100	(may be changed between run-sets)
- Function models with undefined integrals	20	(may be increased w/o limit by changing data statement)

SECTION 6

EXAMPLES AND TEST CASE DATA

The test case data contained in this section was submitted to Langley Research Center, under separate cover, in compliance with Work Statement Paragraph 9.0. The sample problems so outlined were executed, first at the Raytheon computer facility and second at the Langley computer facility, in order to demonstrate both proper program execution and transferability between sites.

Test cases and data were selected on the combined basis of reflecting realistic problem input situations, and exercising both new interface code and new subroutines as supplied by Raytheon. In addition, the cases were chosen to serve as a tutorial base both for new users of CARE/CARE2 and also for those with previous experience in original CARE, but unfamiliar with the changes incorporated in the revised program.

Each of the listings is annotated in order to facilitate in its interpretation. In the interest of brevity, all but the first reliability plot has been deleted.

6.1 JOB #1 - PRODUCT OF EQUATIONS 4 AND 2

Specifics:

- The first stage is a hybrid/simplex configuration consisting of 3 active LRU's and 1 spare LRU.
- The second stage is a standby replacement configuration consisting of 2 series LRU's, each with a single dedicated spare LRU.
- Lambda, for both stages, equals $1.0*10^{-4}$ failures per hour per LRU.
- Mu, for the first stage, equals $1.0*10^{-4}$ failures per hour per LRU initially and then, in sequence, takes on the values $5.0*10^{-5}$ and 0.0 respectively.
- Mu, for the second stage, is held constant at 0.0.
- Mission time equals $3*10^{4}$ hours.
- Time step (for print and plot purposes) equals 10^{3} hours.
- Coverage and Voter Reliability are defaulted.
- Output request is for both tabular reliability printout and plot.

6.2 JOB #2 - PRODUCT OF EQUATION 4 AND 2, WITH CALCULATED COVERAGE

Specifics:

- Same as Job #1 except for Coverage values as related to permanent faults.
- Coverage for the second stage is to be calculated.

- Two stage 2 detectors are employed, each with the following characteristics:
  - Scheduled
  - Truncated exponential distribution
  - 95% detection probability
  - Run time of 0.01 seconds
  - One repetition per second
  - First delayed 0.0 seconds, second delayed 0.5 seconds
- Given detection of the fault, isolation and recovery probabilities are unity.

6.3 JOB #3 - DUAL MODE MODEL (DUAL CHANNEL) WITH PRESET COVERAGE

Specifics:

- Dual Channel configuration with two stages.
- Each stage has a single spare LRU.
- Spares reassignment, in the event of a single channel loss, is not allowed.
- Operation in degenerate mode (mode 2) is allowed, and requires one fully operational channel.
- Lambda, for both stages, is equal to $1*10^{-4}$ failures per hour.
- Mu, for both stages, is equal to $5.0*10^{-5}$ failures per hour.
- The transient error rate, for both stages, equals $1.1*10^{-5}$ per hour.
- Category 2 and 3 switch failure rates are $1.0*10^{-7}$ and $2.0*10^{-8}$ per hour respectively.

- Coverage values for the first stage are unity.
- Coverage values for the second stage, for <u>full-up/degenerate</u> operation respectively, are:
  - o Permanent - 0.999/0.98, with deltas of 0.999/0.998
  - o Transitional - 0.999, with delta of 0.999
  - o Transient - 0.99/0.95
- Mission time equals 10^4 hours.
- Time step equals 10^3 hours.
- Output request is for both tabular reliability printout and plot.

6.4 JOB #4 - DUAL MODE MODEL (DUAL CHANNEL) WITH PRESET COVERAGE, MULTIPLE RUNS WITHIN RUNSET, AND SPARES REASSIGNMENT

Specifics:

- Same as Job #3 except for run time variation in spares quantity, and allowance of spares reassignment.
- Parameter selected for variation is the quantity of spare LRU's associated with Stage 1.
- This quantity takes on the values 0, 1 and 2 for runs 1, 2 and 3 respectively.
- Reassignment of spares, in the event of singular channel loss, is allowed.

6.5 JOB #5 - DUAL MODE MODEL (DUAL CHANNEL) WITH CALCULATED COVERAGE AND MULTIPLE RUNSETS

Specifics:

- Same as Job #3 except for Stage 2 coverage and multiple Stage 1 parameter variations.
- Lambda/Mu values, for the first stage, are as follows:
  - o Run 1, Runset 1 - $1.0*10^{-4}/5.0*10^{-5}$ failures per hour.
  - o Run 2, Runset 1 - $1.0*10^{-4}/1.0*10^{-4}$ failures per hour.
  - o Run 1, Runset 2 - $1.0*10^{-3}/5.0*10^{-4}$ failures per hour.
  - o Run 2, Runset 2 - $1.0*10^{-3}/1.0*10^{-3}$ failures per hour.
- Coverage, for Stage 2, is to be calculated based on the presence of Output Compare and CPU Selftest D/I/R mechanisms, using default data values.

6.6 JOB #6 - DUAL MODE MODEL (HYBRID CHANNEL) WITH CALCULATED COVERAGE AND MULTIPLE RUNSETS

Specifics:

- The first stage requires 3 operating LRU's in mode 1, 2 in mode 2, and has 1 spare LRU.
- The second stage requires 1 operating LRU in both modes, and has 2 spare LRU's.
- The third stage requires 2 operating LRU's in mode 1, 1 in mode 2, and has 1 spare LRU.

- Lambda, for stages 1, 2 and 3, equals $1.0*10^{-5}$, $2.5*10^{-4}$ and $0.5*10^{-4}$ failures per hour per LRU, respectively.
- K, for all stages, equals 2.0.
- Spares reassignment is allowed.
- Coverage values for the first stage, for mode 1/mode 2 operation respectively, are
  - o Permanent - 0.998/0.99, with deltas of 0.995/0.99
  - o Transitional - 0.996, with delta of 0.995
  - o Transient - 0.99/0.97
- Coverage for the second stage is to be calculated based on the presence of CPU Selftest, Invalid Instruction and CPU Code, D/I/R mechanisms, using default data values.
- Coverage for the third stage is to be calculated assuming 2 fault subclasses, as described in succeeding line items.
- The first fault subclass accounts for 63% of LRU faults, and employs memory code as the sole D/I/R mechanism. Default data values are to be used.
- The second fault subclass accounts for 37% of LRU faults, and employs address feedback as the sole D/I/R mechanism. Default data values are to be used only where not over-ridden by the following:

- o Checkout time, for a spare, equals 0.1 seconds
- o Detection probability equals 0.9996
- o Isolation probability equals 0.99985 and has a pulse distribution of length 0.2 seconds
- o Error propagation recovery coefficient equals 0.96 for mode 2 operation

- A second run to be performed, in which CPU Selftest for Stage 2 is to be executed every 0.4 seconds.
- Mission time equals $16*10^3$ hours.
- Time step equals $2*10^3$ hours.
- Output request is for tabular reliability data only.

6.7 JOB #7 - PRODUCT OF EQUATIONS 7, 4 AND 2

Specifics:

- Run consists of producing a product reliability, assumming the configurations of Job #3 and Job #1 to be in series in a reliability block diagram.
- All data values are as defined in the respective jobs with the exception of Mission time, which is set equal to 10^4 hours for the combined configuration.

FIGURE 6-1

JOB #1 INPUT/OUTPUT LISTINGS

```
 $OPTSON OUTPT1=T,OUTPT3=T,LPLOT=T,PRODT=T$

 $VAR1 PROD(1)=4,2$

 $VAR T=3.0E4,STEP=1.0E3,OPTION=1$

10000

100000

IB

 $PARVEC Q(2)=1,Z(2)=2,S(1)=1,1,LAM(1)=2*1.0E-4,MU(1)=1.0E-4,0.0$

IV       8

 $VARY PARAM(1,1)=1.0E-4,5.0E-5,0.0, PARAM(1,2)=3*0.0$

G

S
```

FIGURE 6-1 (Cont.-2)

CARE2 (COMPUTER-AIDED RELIABILITY ESTIMATION)

```
DO YOU WISH TO HAVE YOUR RESPONSES TO MY
QUESTIONS PRINTED BACK FOR VERIFICATION.                OUTPT3= T
DO YOU WISH TO FORM A COMPLEX EQUATION WHICH IS
THE PRODUCT OF THE PRIMARY EQUATIONS.                   PRODT=  T

$VAR1
PROD    =  4,  2,  0,  0,  0,  0,  0,  0,  0,  0,
$END
DO YOU WISH TO HAVE 2-D RELIABILITY PLOTS-              LPLOT=  T
INPUT A 1 IN THE COLUMN SPECIFIED BELOW IF YOU WISH
THE CORRESPONDING PLOT OPTION, OTHERWISE INPUT 0.
NOTE* WHEN PERFORMING PRODUCT OF RELIABILITIES, NO OTHER
PLOT OPTION BESIDES PRODUCT OF RELIABILITIES MAY BE SPECIFIED.
COLUMN 1 - PLOTS PRODUCT OF RELIABILITIES,
COLUMN 2 - PLOTS RELIABILITY
COLUMN 3 - PLOTS DIFF, RIF, AND GAIN
COLUMN 4 - PLOTS MTF AND RELIABILITY AT MTF
COLUMN 5 - PLOTS UNRELIABILITY
10000
FOR ABSCISSA, INPUT 1 IN COLUMN 1 IF ABSCISSA IS T,
1 IN COLUMN 2 IF ABSCISSA IS LOG(T) - BASE 10,
1 IN COLUMN 3 IF ABSCISSA IS LAMT,
1 IN COLUMN 4 IF ABSCISSA IS LOG(LAMT) - BASE 10,
1 IN COLUMN    5 IF ABSCISSA IS EXP(-LAMBDA(T)),
1 IN COLUMN 6 IF ABSCISSA IS LOG(EXP(-LAMT)) - BASE 10,
100000
DO YOU WISH TO HAVE PRINTED TABLE OF RELIABILITY RESULTS        OUTPT1=  T
DO YOU WISH TO HAVE PRINTED TABLE OF DIFF, RIF,
AND GAIN RESULTS -                                              OUTPT2=  F
DO YOU WISH MTF AND RELIABILITY AT MTF RESULTS PRINTED          OUTPT4=  F
DO YOU WANT PRINTED RESULTS OF THE MAXIMUM MISSION
TIME CALCULATIONS -                                             OUTPT5=  F
```

FIGURE 6-1 (Cont.-3)

DATA BASE FOR RUN-SET
1
****

STAGE/EQUATION NUMBER

PARAMETER	1/4	2/2	
Q=Q1		1.00000000	(Equation #; Stage #)
S	1	1	← # of spares
W	1	1	← # of subunits/unit
Z	1	2	← # of identical systems in series
LAM	.00010000	.00010000	← on-line (failure rates)
MU	.00010000	0.00000000	← standby (failure rates)
C=C1		1.00000000	← permanent coverage value (n/a to Eq. 4)
RV	1.00000000	1.00000000	

PARAMETER TO BE VARIED = MU

ITERATION		
1	.00010000	0.00000000
2	5.0000E-05	0.
3	0.00000000	0.00000000

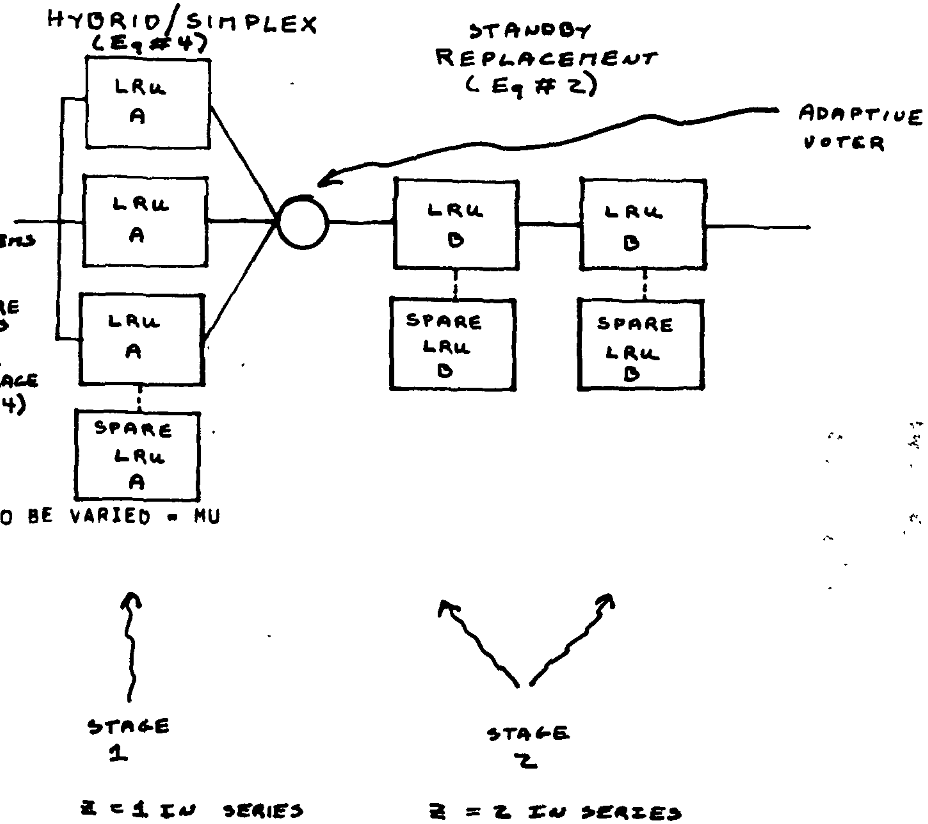


FIGURE 6-1 (Cont.-4)

RUN 1 RELIABILITY RESULTS

VALUES OF MU USED ARE AS SPECIFIED IN ITERATION 1 ABOVE

STAGE 1 EQUATION NUMBER 4

T	REL	UNREL	SIMREL	SIMGAIN	SIMRIF
0.000	1.0000000	0.0000000	1.0000000	1.0000000E+00	1.0000000E+35
1000.000	.9983584	.0016416	.9048374	1.1033567E+00	5.7970848E+01
2000.000	.9891672	.0108328	.8187308	1.2081715E+00	1.6733367E+01
3000.000	.9696913	.0303087	.7408182	1.3089464E+00	8.5514083E+00
4000.000	.9401482	.0598518	.6703200	1.4025363E+00	5.5082701E+00
5000.000	.9021363	.0978637	.6065307	1.4873713E+00	4.0205844E+00
6000.000	.8577434	.1422566	.5488116	1.5629105E+00	3.1716526E+00
7000.000	.8090678	.1909322	.4965853	1.6292625E+00	2.6366152E+00
8000.000	.7579842	.2420158	.4493290	1.6869249E+00	2.2753518E+00
9000.000	.7060520	.2939480	.4065697	1.7366077E+00	2.0188277E+00
10000.000	.6545004	.3454996	.3678794	1.7791165E+00	1.8295840E+00
11000.000	.6042532	.3957468	.3328711	1.8152769E+00	1.6857467E+00
12000.000	.5559707	.4440293	.3011942	1.8458878E+00	1.5737831E+00
13000.000	.5100963	.4899037	.2725318	1.8716948E+00	1.4849209E+00
14000.000	.4669006	.5330994	.2465976	1.8933755E+00	1.4132507E+00
15000.000	.4265211	.5734789	.2231302	1.9115349E+00	1.3546615E+00
16000.000	.3889951	.6110049	.2018965	1.9267053E+00	1.3062145E+00
17000.000	.3542873	.6457127	.1826835	1.9393502E+00	1.2657588E+00
18000.000	.3223112	.6776888	.1652989	1.9498691E+00	1.2316879E+00
19000.000	.2929458	.7070542	.1495686	1.9586044E+00	1.2027810E+00
20000.000	.2660485	.7339515	.1353353	1.9658475E+00	1.1780952E+00
21000.000	.2414651	.7585349	.1224564	1.9718452E+00	1.1568928E+00
22000.000	.2190363	.7809637	.1108032	1.9768057E+00	1.1385892E+00
23000.000	.1986032	.8013968	.1002588	1.9809041E+00	1.1227161E+00
24000.000	.1800105	.8199895	.0907180	1.9842871E+00	1.1088947E+00
25000.000	.1631092	.8368908	.0820850	1.9870772E+00	1.0968158E+00
26000.000	.1477581	.8522419	.0742736	1.9893766E+00	1.0862250E+00
27000.000	.1338243	.8661757	.0672055	1.9912704E+00	1.0769115E+00
28000.000	.1211841	.8788159	.0608101	1.9928291E+00	1.0686993E+00
29000.000	.1097224	.8902776	.0550232	1.9941115E+00	1.0614406E+00
30000.000	.0993335	.9006665	.0497871	1.9951659E+00	1.0550108E+00

RELIABILITY, ETC. OF FIRST STAGE AS F(T)

STAGE 2 EQUATION NUMBER 2

T	REL	UNREL	SIMREL	SIMGAIN	SIMRIF
0.000	1.0000000	0.0000000	1.0000000	1.0000000E+00	1.0000000E+35
1000.000	.9906642	.0093358	.9048374	1.0948533E+00	1.0193309E+01
2000.000	.9652609	.0347391	.8187308	1.1789723E+00	5.2180129E+00
3000.000	.9274917	.0725083	.7408182	1.2519828E+00	3.5745101E+00
4000.000	.8806848	.1193152	.6703200	1.3138273E+00	2.7631003E+00
5000.000	.8277287	.1722713	.6065307	1.3646940E+00	2.2840104E+00
6000.000	.7710572	.2289428	.5488116	1.4049578E+00	1.9707470E+00
7000.000	.7126652	.2873348	.4965853	1.4351319E+00	1.7520145E+00
8000.000	.6541447	.3458553	.4493290	1.4558258E+00	1.5922007E+00
9000.000	.5967290	.4032710	.4065697	1.4677165E+00	1.4715423E+00
10000.000	.5413411	.4586589	.3678794	1.4715178E+00	1.3781933E+00
11000.000	.4886419	.5113581	.3328711	1.4679615E+00	1.3046219E+00
12000.000	.4390749	.5609251	.3011942	1.4577800E+00	1.2458094E+00
13000.000	.3929072	.6070928	.2725318	1.4416932E+00	1.1982818E+00
14000.000	.3502660	.6497340	.2465970	1.4203985E+00	1.1595560E+00
15000.000	.3111692	.6888308	.2231302	1.3945635E+00	1.1278093E+00
16000.000	.2755525	.7244475	.2018965	1.3648205E+00	1.1016719E+00
17000.000	.2432911	.7567089	.1826835	1.3317629E+00	1.0800937E+00

RELIABILITY, ETC. OF SECOND STAGE AS F(T)

FIGURE 6-1 (Cont.-5)

18000.000	.2142180	.7857820	.1652989	1.2959433E+00	1.0622553E+00
19000.000	.1881382	.8118618	.1495686	1.2578721E+00	1.0475076E+00
20000.000	.1648407	.8351593	.1353353	1.2180175E+00	1.0353292E+00
21000.000	.1441075	.8558925	.1224564	1.1768063E+00	1.0252965E+00
22000.000	.1257200	.8742800	.1108032	1.1346243E+00	1.0170618E+00
23000.000	.1094645	.8905355	.1002588	1.0918188E+00	1.0103372E+00
24000.000	.0951359	.9048641	.0907180	1.0486995E+00	1.0048824E+00
25000.000	.0825399	.9174601	.0820850	1.0055412E+00	1.0004958E+00
26000.000	.0714947	.9285053	.0742736	9.6258557E-01	9.9700712E-01
27000.000	.0618320	.9381680	.0672055	9.2004547E-01	9.9427833E-01
28000.000	.0533972	.9466028	.0608101	8.7809730E-01	9.9216893E-01
29000.000	.0460491	.9539509	.0550232	8.3690318E-01	9.9059269E-01
30000.000	.0396600	.9603400	.0497871	7.9659309E-01	9.8945474E-01

T	REL PRODUCT
0.000	1.0000000
1000.000	.9890380
2000.000	.9548044
3000.000	.8993806
4000.000	.8279742
5000.000	.7467241
6000.000	.6613692
7000.000	.5765945
8000.000	.4958314
9000.000	.4213217
10000.000	.3543080
11000.000	.2952634
12000.000	.2441128
13000.000	.2004205
14000.000	.1635394
15000.000	.1327202
16000.000	.1071886
17000.000	.0861950
18000.000	.0690449
19000.000	.0551143
20000.000	.0438556
21000.000	.0347969
22000.000	.0275372
23000.000	.0217400
24000.000	.0171255
25000.000	.0134630
26000.000	.0105639
27000.000	.0082746
28000.000	.0064709
29000.000	.0050526
30000.000	.0039396

SYSTEM RELIABILITY (≡ PRODUCT OF STAGE RELIABILITIES) AS F(T)

FIGURE 6-1 (Cont.-6)

PLOT OF RELIABILITY
VS. TIME

```
1.000   AA
 .980
 .960     A
 .940
 .920
 .900      A
 .880
 .860
 .840       A
 .820
 .800
 .780
 .760        A
 .740
 .720
 .700
 .680         A
 .660
 .640
 .620
 .600
 .580          A
 .560
 .540
 .520
 .500           A
 .480
 .460
 .440            A
 .420
 .400
 .380
 .360             A
 .340
 .320
 .300              A
 .280
 .260               A
 .240
 .220                A
 .200
 .180                 A
 .160
 .140                  A
 .120                   A
 .100                    A
 .080                     A
 .060                      AA
 .040                        AAA
 .020                           AAAAAAA
-.000   ----------------------------------------

 CHAN   0000000000111111111122222222223---------
 NUMB   0123456789012345678901234567890---------
CHANNEL VALUES FOR X AXIS

 CONT   12
  OF    00
 CHAN   00
```

FIGURE 6-1 (Cont.-7)

```
            .0001000000000000000000000000000
             0000999876554332211100000000000
             0000950356802504063197643322111

CONTENTS OF CHANNELS FOR HISTOGRAM   A   FOLLOW

            .1000000000000000000000000000000
     CONT    0998876544322211100000000000000
      UF     0859246792594063086543221110000
     CHAN    0949761651454032769534717308653
             0083973583321457110188571452409
             0408727932161242994160443667754

                    PRODUCT OFRELYBILITY  VS  MISSION    TIME T
                                                PLOT FOR EQUATIONS  4   2
             (NI MEANS NOT INPUTED OR DEFAULT VALUE USED)
LAMBDA        MU        S    N    K        Q        C        RV        Z    W    P        MUT

                    1 PLOT(S) COMPLETED
```

FIGURE 6-1 (Cont.-8)

RUN 2 RELIABILITY RESULTS

WITH SECOND VALUE OF MU

STAGE 1 EQUATION NUMBER 4

T	REL	UNREL	SIMREL	SIMGAIN	SIMRIF
0.000	1.0000000	0.0000000	1.0000000	1.0000000E+00	1.0000000E+35
1000.000	.9985462	.0014538	.9048374	1.1035642E+00	6.5458759E+01
2000.000	.9902986	.0097014	.8187308	1.2095534E+00	1.8684824E+01
3000.000	.9725751	.0274249	.7408182	1.3128390E+00	9.4505842E+00
4000.000	.9453251	.0546749	.6703200	1.4102593E+00	6.0298192E+00
5000.000	.9098163	.0901837	.6065307	1.5000335E+00	4.3629757E+00
6000.000	.8678538	.1321462	.5488116	1.5813326E+00	3.4143109E+00
7000.000	.8213362	.1786638	.4965853	1.6539681E+00	2.8176655E+00
8000.000	.7720221	.2279779	.4493290	1.7181669E+00	2.4154584E+00
9000.000	.7214221	.2785779	.4065697	1.7744120E+00	2.1302132E+00
10000.000	.6707658	.3292342	.3678794	1.8233305E+00	1.9199724E+00
11000.000	.6210096	.3789904	.3328711	1.8656158E+00	1.7602790E+00
12000.000	.5728642	.4271358	.3011942	1.9019761E+00	1.6360272E+00
13000.000	.5268314	.4731686	.2725318	1.9331006E+00	1.5374396E+00
14000.000	.4832409	.5167591	.2465970	1.9596385E+00	1.4579386E+00
15000.000	.4422859	.5577141	.2231302	1.9821879E+00	1.3929535E+00
16000.000	.4040534	.5959466	.2018965	2.0012898E+00	1.3392199E+00
17000.000	.3685508	.6314492	.1826835	2.0174277E+00	1.2943503E+00
18000.000	.3357268	.6642732	.1652989	2.0310286E+00	1.2565629E+00
19000.000	.3054889	.6945111	.1495686	2.0424664E+00	1.2245036E+00
20000.000	.2777170	.7222830	.1353353	2.0520663E+00	1.1971273E+00
21000.000	.2522737	.7477263	.1224564	2.0601095E+00	1.1736160E+00
22000.000	.2290121	.7709879	.1108032	2.0668376E+00	1.1533214E+00
23000.000	.2077822	.7922178	.1002588	2.0724572E+00	1.1357244E+00
24000.000	.1884343	.8115657	.0907180	2.0771449E+00	1.1204048E+00
25000.000	.1708230	.8291770	.0820850	2.0810503E+00	1.1070194E+00
26000.000	.1548084	.8451916	.0742736	2.0843003E+00	1.0952859E+00
27000.000	.1402580	.8597420	.0672055	2.0870021E+00	1.0849703E+00
28000.000	.1270472	.8729528	.0608101	2.0892460E+00	1.0758771E+00
29000.000	.1150595	.8849405	.0550232	2.0911080E+00	1.0678422E+00
30000.000	.1041870	.8958130	.0497871	2.0926518E+00	1.0607269E+00

STAGE 2 EQUATION NUMBER 2

T	REL	UNREL	SIMREL	SIMGAIN	SIMRIF
0.000	1.0000000	0.0000000	1.0000000	1.0000000E+00	1.0000000E+35
1000.000	.9906642	.0093358	.9048374	1.0948533E+00	1.0193309E+01
2000.000	.9652609	.0347391	.8187308	1.1789723E+00	5.2180129E+00
3000.000	.9274917	.0725083	.7408182	1.2519828E+00	3.5745101E+00
4000.000	.8806848	.1193152	.6703200	1.3138273E+00	2.7631003E+00
5000.000	.8277287	.1722713	.6065307	1.3646940E+00	2.2840104E+00
6000.000	.7710572	.2289428	.5488116	1.4049578E+00	1.9707470E+00
7000.000	.7126652	.2873348	.4965853	1.4351315E+00	1.7520145E+00
8000.000	.6541447	.3458553	.4493290	1.4558258E+00	1.5922007E+00
9000.000	.5967290	.4032710	.4065697	1.4677165E+00	1.4715423E+00
10000.000	.5413411	.4586589	.3678794	1.4715178E+00	1.3781933E+00
11000.000	.4886419	.5113581	.3328711	1.4679615E+00	1.3046219E+00
12000.000	.4390749	.5609251	.3011942	1.4577800E+00	1.2458094E+00
13000.000	.3929072	.6070928	.2725318	1.4416932E+00	1.1982818E+00
14000.000	.3502660	.6497340	.2465970	1.4203985E+00	1.1595560E+00
15000.000	.3111692	.6888308	.2231302	1.3945635E+00	1.1278093E+00
16000.000	.2755525	.7244475	.2018965	1.3648205E+00	1.1016719E+00
17000.000	.2432911	.7567089	.1826835	1.3317629E+00	1.0800937E+00

FIGURE 6-1 (Cont.-9)

18000.000	.2142180	.7857820	.1652989	1.2959433E+00	1.0622553E+00
19000.000	.1881382	.8118618	.1495686	1.2578721E+00	1.0475076E+00
20000.000	.1648407	.8351593	.1353353	1.2180175E+00	1.0353292E+00
21000.000	.1441075	.8558925	.1224564	1.1768063E+00	1.0252965E+00
22000.000	.1257200	.8742800	.1108032	1.1346243E+00	1.0170618E+00
23000.000	.1094645	.8905355	.1002588	1.0918188E+00	1.0103372E+00
24000.000	.0951359	.9048641	.0907180	1.0486995E+00	1.0048824E+00
25000.000	.0825399	.9174601	.0820850	1.0055412E+00	1.0004958E+00
26000.000	.0714947	.9285053	.0742736	9.6258557E-01	9.9700712E-01
27000.000	.0618320	.9381680	.0672055	9.2004347E-01	9.9427233E-01
28000.000	.0533972	.9466028	.0608101	8.7809730E-01	9.9216893E-01
29000.000	.0460491	.9539509	.0550232	8.3690318E-01	9.9059269E-01
30000.000	.0396600	.9603400	.0497871	7.9659309E-01	9.8945474E-01

T	REL PRODUCT
0.000	1.0000000
1000.000	.9892240
2000.000	.9558965
3000.000	.9020553
4000.000	.8325334
5000.000	.7530811
6000.000	.6691649
7000.000	.5853378
8000.000	.5050142
9000.000	.4304935
10000.000	.3631131
11000.000	.3034513
12000.000	.2515303
13000.000	.2069959
14000.000	.1692628
15000.000	.1376257
16000.000	.1113379
17000.000	.0896651
18000.000	.0719187
19000.000	.0574741
20000.000	.0457791
21000.000	.0363545
22000.000	.0287914
23000.000	.0227448
24000.000	.0179269
25000.000	.0140997
26000.000	.0110680
27000.000	.0086724
28000.000	.0067840
29000.000	.0052984
30000.000	.0041321

FIGURE 6-1 (Cont.-10)

RUN 3 RELIABILITY RESULTS

WITH THIRD VALUE OF MU

STAGE 1 EQUATION NUMBER 4

T	REL	UNREL	SIMREL	SIMGAIN	SIMRIF
0.000	1.0000000	0.0000000	1.0000000	1.0000000E+00	1.0000000E+35
1000.000	.9987387	.0012613	.9048374	1.1037769E+00	7.5446916E+01
2000.000	.9914862	.0085138	.8187308	1.2110039E+00	2.1291123E+01
3000.000	.9756726	.0243274	.7408182	1.3170202E+00	1.0653893E+01
4000.000	.9510108	.0489892	.6703200	1.4187414E+00	6.7296472E+00
5000.000	.9184337	.0815663	.6065307	1.5142411E+00	4.8239183E+00
6000.000	.8794336	.1205664	.5488116	1.6024324E+00	3.7422388E+00
7000.000	.8356671	.1643329	.4965853	1.6828270E+00	3.0633844E+00
8000.000	.7887312	.2112688	.4493290	1.7553535E+00	2.6064946E+00
9000.000	.7400474	.2599526	.4065697	1.8202229E+00	2.2828406E+00
10000.000	.6908143	.3091857	.3678794	1.8778280E+00	2.0444690E+00
11000.000	.6419988	.3580012	.3328711	1.9286708E+00	1.8634821E+00
12000.000	.5943496	.4056504	.3011942	1.9733102E+00	1.7226800E+00
13000.000	.5484224	.4515776	.2725318	2.0123245E+00	1.6109485E+00
14000.000	.5046080	.4953920	.2465970	2.0462863E+00	1.5208219E+00
15000.000	.4631614	.5368386	.2231302	2.0757453E+00	1.4471199E+00
16000.000	.4242286	.5757714	.2018965	2.1012180E+00	1.3861464E+00
17000.000	.3878703	.6121297	.1826835	2.1231816E+00	1.3352015E+00
18000.000	.3540820	.6459180	.1652989	2.1420713E+00	1.2922710E+00
19000.000	.3228109	.6771891	.1495686	2.1582798E+00	1.2558256E+00
20000.000	.2939697	.7060303	.1353353	2.1721585E+00	1.2246850E+00
21000.000	.2674472	.7325528	.1224564	2.1840195E+00	1.1979254E+00
22000.000	.2431174	.7568826	.1108032	2.1941381E+00	1.1748148E+00
23000.000	.2208458	.7791542	.1002588	2.2027564E+00	1.1547665E+00
24000.000	.2004945	.7995055	.0907180	2.2100857E+00	1.1373055E+00
25000.000	.1819258	.8180742	.0820850	2.2163103E+00	1.1220437E+00
26000.000	.1650054	.8349946	.0742736	2.2215897E+00	1.1086616E+00
27000.000	.1496036	.8503964	.0672055	2.2260621E+00	1.0968938E+00
28000.000	.1355971	.8644029	.0608101	2.2298466E+00	1.0865187E+00
29000.000	.1228694	.8771306	.0550232	2.2330457E+00	1.0773501E+00
30000.000	.1113113	.8886887	.0497871	2.2357472E+00	1.0692303E+00

STAGE 2 EQUATION NUMBER 2

T	REL	UNREL	SIMREL	SIMGAIN	SIMRIF
0.000	1.0000000	0.0000000	1.0000000	1.0000000E+00	1.0000000E+35
1000.000	.9906642	.0093358	.9048374	1.0948533E+00	1.0193309E+01
2000.000	.9652609	.0347391	.8187308	1.1789723E+00	5.2180129E+00
3000.000	.9274917	.0725083	.7408182	1.2519828E+00	3.5745101E+00
4000.000	.8806848	.1193152	.6703200	1.3138273E+00	2.7631003E+00
5000.000	.8277287	.1722713	.6065307	1.3646940E+00	2.2840104E+00
6000.000	.7710572	.2289428	.5488116	1.4049578E+00	1.9707470E+00
7000.000	.7126652	.2873348	.4965853	1.4351315E+00	1.7520145E+00
8000.000	.6541447	.3458553	.4493290	1.4558258E+00	1.5922007E+00
9000.000	.5967290	.4032710	.4065697	1.4677165E+00	1.4715423E+00
10000.000	.5413411	.4586589	.3678794	1.4715178E+00	1.3781933E+00
11000.000	.4886419	.5113581	.3328711	1.4679615E+00	1.3046219E+00
12000.000	.4390749	.5609251	.3011942	1.4577800E+00	1.2458094E+00
13000.000	.3929072	.6070928	.2725318	1.4416932E+00	1.1982818E+00
14000.000	.3502660	.6497340	.2465970	1.4203985E+00	1.1595560E+00
15000.000	.3111692	.6888308	.2231302	1.3945635E+00	1.1278093E+00
16000.000	.2755525	.7244475	.2018965	1.3648205E+00	1.1016719E+00
17000.000	.2432911	.7567089	.1826835	1.3317629E+00	1.0800937E+00

FIGURE 6-1 (Cont.-11)

18000,000	.2142180	.7857820	.1652989	1,2959433E+00	1,0622553E+00
19000,000	.1881382	.8118618	.1495686	1,2578721E+00	1,0475076E+00
20000,000	.1648407	.8351593	.1353353	1,2180175E+00	1,0353292E+00
21000,000	.1441075	.8558925	.1224564	1,1768063E+00	1,0252965E+00
22000,000	.1257200	.8742800	.1108032	1,1346243E+00	1,0170618E+00
23000,000	.1094645	.8905355	.1002588	1,0918188E+00	1,0103372E+00
24000,000	.0951359	.9048641	.0907180	1,0486995E+00	1,0048824E+00
25000,000	.0825399	.9174601	.0820850	1,0055412E+00	1,0004958E+00
26000,000	.0714947	.9285053	.0742736	9,6258557E-01	9,9700712E-01
27000,000	.0618320	.9381680	.0672055	9,2004347E-01	9,9427233E-01
28000,000	.0533972	.9466028	.0608101	8,7809730E-01	9,9216893E-01
29000,000	.0460491	.9539509	.0550232	8,3690318E-01	9,9059269E-01
30000,000	.0396600	.9603400	.0497871	7,9659309E-01	9,8945474E-01

T	REL PRODUCT
0,000	1,0000000
1000,000	.9894147
2000,000	.9570428
3000,000	.9049282
4000,000	.8375407
5000,000	.7602139
6000,000	.6780936
7000,000	.5955509
8000,000	.5159443
9000,000	.4416077
10000,000	.3739662
11000,000	.3137075
12000,000	.2609640
13000,000	.2154791
14000,000	.1767470
15000,000	.1441215
16000,000	.1168972
17000,000	.0943654
18000,000	.0758507
19000,000	.0607331
20000,000	.0484582
21000,000	.0385411
22000,000	.0305647
23000,000	.0241748
24000,000	.0190742
25000,000	.0150161
26000,000	.0117970
27000,000	.0092503
28000,000	.0072405
29000,000	.0056580
30000,000	.0044146

FIGURE 6-2

JOB #2 INPUT/OUTPUT LISTINGS

```
 $OPTSON OUTPT1=T,OUTPT3=T,COVINT=T,PRODT=T,COVPRC=T$

 $VAR1 PROD(1)=4,2$

 $VAR T=3.0E4,STEP=1.0E3,OPTION=1$

 $COVCAL IGENC(2)=1,IFSC(1)=2,FRAC(1)=1.0$

ID

 $DATA PFDS(1)=1.0, TFDS(1)=0.0, TMINOR=1.0$

IB

 $PARVEC Q(2)=1,Z(2)=2,S(1)=1,1,LAM(1)=2*1.0E-4,MU(1)=1.0E-4,0.0$

IV        8

 $VARY PARAM(1,1)=1.0E-4,5.0E-5,0.0, PARAM(1,2)=3*0.0$

F  D    1   311   1  0.95     0.0      1.0      100.0    0.0      0.01

F  D    2   311   1  0.95     0.5      1.0      100.0    0.0      0.01

F  I    1   010      1.0      0.0      1.0      0.0      0.0      0.0
```

FIGURE 6-2 (Cont.-2)

F	E	1	110	1.0	0.0	1.0	0.0	0.0	100.0
F	T	1	110	1.0	0.0	1.0	0.0	0.0	100.0
C	D	1	111	1					
C	D	2	111	2					
C	I	1	111	50					
C	E	1	111	50					
C	T	1	111	50					
PC									
N									
PF									
G									
S									

FIGURE 6-2 (Cont.-3)

CARE2 (COMPUTER-AIDED RELIABILITY ESTIMATION)

[SAME SYSTEM CONFIGURATION AS JOB # 1, BUT WITH PERMANENT COVERAGE CALCULATED FOR SECOND STAGE]

```
DO YOU WISH TO HAVE YOUR RESPONSES TO MY
QUESTIONS PRINTED BACK FOR VERIFICATION.                    OUTPT3= T
DO YOU WISH TO FORM A COMPLEX EQUATION WHICH IS
THE PRODUCT OF THE PRIMARY EQUATIONS.                       PRODT=  T

$VAR1
PROD    =  4,  2,  0,  0,  0,  0,  0,  0,  0,  0,
$END
DO YOU WISH TO HAVE 2-D RELIABILITY PLOTS-                 LPLOT=  F
DO YOU WISH TO HAVE PRINTED TABLE OF RELIABILITY RESULTS    OUTPT1=  T
DO YOU WISH TO HAVE PRINTED TABLE OF DIFF, RIF,
AND GAIN RESULTS -                                          OUTPT2=  F
DO YOU WISH MTF AND RELIABILITY AT MTF RESULTS PRINTED      OUTPT4=  F
DO YOU WANT PRINTED RESULTS OF THE MAXIMUM MISSION
TIME CALCULATIONS =                                         OUTPT5=  F
                                          COVERAGE DATA BASE
                                          ********************
```

[A CHECKOUT ROUTINE IS USED TO VERIFY OPERATION OF SPARE PRIOR TO SWITCHING IT IN TO REPLACE A FAULTY UNIT. THE CHECKOUT IS 100% ACCURATE, AND REQUIRES ZERO TIME.]

- FAULT SUBCLASS LINKAGE AND PARAMETERS -

FAULT SUBCLASS	LINKED TO STAGE	FRACTION OF FAULT RATE	PROB. OF FAULT DET. IN SPARES	TIME FOR FAULT DET. IN SPARES
1	2	1.000000	1.000000	0.000000
2	0	0.000000	1.000000	0.000000
3	0	0.000000	1.000000	0.000000
4	0	0.000000	1.000000	0.000000
5	0	0.000000	1.000000	0.000000
6	0	0.000000	1.000000	0.000000
7	0	0.000000	1.000000	0.000000
8	0	0.000000	1.000000	0.000000

[FIRST (AND ONLY) STAGE WITH COVERAGE CALCULATED IS #2.]

[FAULTS WITHIN THE STAGE ARE CATEGORIZED INTO ONE GROUP, WHICH ACCOUNTS FOR 100% OF STAGE FAULTS]

[NOT USED IN THIS RUN] (subclasses 2–8)

MINOR CYCLE DURATION (FOR SCHEDULED DETECTORS) = 1.000000

[PERIODIC FAULT DETECTION SOFTWARE ROUTINES ARE REPEATED NO MORE FREQUENTLY THAN EVERY 1.0 ARBITRARY TIME UNITS]

- D/I/R MECHANISM DEFINITIONS - (X MEANS FUNCTION UNDEFINED).

FAULT TYPE/MODE

SUBCLASS	MECHANISM	PERM/1 D	I	E	T	PERM/2 D	I	E	T	TRANS/1 D	I	E	T	TRANS/2 D	I	E	T	PERM/0 D	I	E	T
1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	2	2	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[TWO "WAYS" TO DETECT, ISOLATE & RECOVER FROM A FAULT]

[D FOR FAULT DETECTOR] [I FOR FAULT ISOLATOR] [E FOR ERROR PROPAGATION FAULT RECOVERY] [T FOR TIME LOST FAULT RECOVERY]

[NON-ZERO VALUES MEANINGFUL ONLY FOR DUAL MODE SYSTEM]

[PERMANENT FAULT, MODE 1 SYSTEM OPERATION COVERAGE ELEMENT SPECIFICATIONS]

```
NORMALIZATION COMPLETE FOR DETECTION  FUNCTIONS
NORMALIZATION COMPLETE FOR ISOLATION  FUNCTIONS
NORMALIZATION COMPLETE FOR E. P. REC. FUNCTIONS
NORMALIZATION COMPLETE FOR T. L. REC. FUNCTIONS
```

[FUNCTION SPECIFICATION IDENTIFIERS (DEFINED ON NEXT PAGE)]

FIGURE 6-2 (Cont.-4)

COVERAGE FUNCTION SPECIFICATIONS

****

1 DENOTES INTEGRAL OF FUNCTION IS DEFINED, AND NEED NOT BE FORMED DURING EVALUATION

PROBABILITY OF DETECTING FAULT IF THIS WERE THE ONLY DETECTOR

FUNCTION PARAMETERS - FOR THIS FUNCTION (TYPE #3) $F = P1(e^{-P2T} + P3)$

= DETECTION =

FUNCTION NUMBER	IGFT	INTF	ISCH	IREP	COEF	TDEL	P1	P2	P3	TDUR
1	3	1	1	1	.950000	0.	1.5820E+02	1.0000E+02	0.	1.0000E-02
2	3	1	1	1	.950000	5.0000E-01	1.5820E+02	1.0000E+02	0.	1.0000E-02

FUNCTION TYPE (3 = DECAYING EXPONENTIAL)

ZERO OFFSET OF FUNCTION ON TIME SCALE

= ISOLATION =

FUNCTION NUMBER	IGFT	INTF	COEF	TDEL	P1	P2	P3	TDUR
1	0	1	1.000000	0.	1.0000E+00	0.	0.	0.

DURATION OF IMPULSE ≡ 0.0

FUNCTION TYPE (0 = IMPULSE)

= FAULT PROPAGATION RECOVERY =

FUNCTION NUMBER	IGFT	INTF	COEF	P1	P2	P3	TDUR
1	1	1	1.000000	1.0000E+00	0.	0.	1.0000E+02

DURATION OF CONSTANT = 100 TIME UNITS (I.E. A PULSE)

FUNCTION TYPE (1 = CONSTANT)

= TIME LOST RECOVERY =

FUNCTION NUMBER	IGFT	INTF	COEF	P1	P2	P3	TDUR
1	1	1	1.000000	1.0000E+00	0.	0.	1.0000E+02

COVERAGE CONTRIBUTIONS = ISU = 1 MD = 1 IET = 1 JS = 1

MECHANISM	CONTRIBUTION
1	.49875
2	.49875

FAULT SUBCLASS # 1

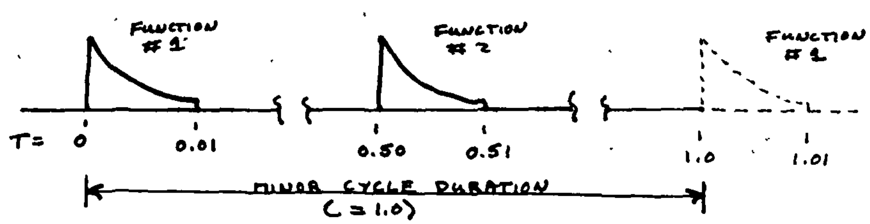
1 for PERMANENT, 2 FOR TRANSIENT FAULT

OF SPARES CHECKED BETWEEN FAULT TIME AND RECOVERY TIME (A CONDITION)

OPERATING MODE OF SYSTEM:
1 FOR FULLY OPERATIONAL
2 FOR DEGENERATE
0 FOR TRANSITIONAL

IN THIS SITUATION, FAULT ISOLATION AND RECOVERY PROCESSES WERE DEFINED TO BE PERFECT. THUS COVERAGE = DETECTION PROBABILITY. GIVEN TWO DETECTORS, IN COMPETITION, EACH WITH DETECTION PROBABILITY OF 0.95 (COEF VALUE) AND GIVEN A SYMMETRICAL SCHEDULE AS BELOW, THEN TOTAL DETECTION CAPABILITY = COVERAGE $= 1-(1-0.95)^2 = 0.9975$, WITH EACH MECHANISM CONTRIBUTING HALF OR 0.49875

TWO EQUAL (EXCEPT FOR DELAY TIME) DETECTORS ARE DEFINED. WITH RESPECT TO THE MINOR CYCLE, THEY APPEAR AS FOLLOWS:



VIEW IS OF DETECTION RATE FUNCTION (AREA ≡ 1.0 FOR ALL DETECTORS AND ISOLATORS) (DETECTION OCCURS ONLY WHEN NON-ZERO)

FIGURE 6-2 (Cont.-5)

DATA BASE FOR RUN-SET
1
****

STAGE/EQUATION NUMBER

PARA-METER	1/4	2/2
Q=Q1		1.00000000
S	1	1
W	1	1
Z	1	2
LAM	.00010000	.00010000
MU	.00010000	0.00000000
C=C1		.99750000
RV	1.00000000	1.00000000

NOTE THAT, GIVEN A STAGE COMPRISED OF Z IDENTICAL SUBSTAGES IN SERIES, COVERAGE FOR EACH OF THE SUBSTAGES, AS WELL AS FOR THE STAGE AS A WHOLE, IS A SINGLE VALUE. THUS, THE COMPUTED VALUE OF C APPLIES EQUALLY, IN THIS EXAMPLE, TO EACH OF THE TWO SUBSTAGES AND TO THE ENTIRE STAGE. NOTE ALSO THAT THE SAME REASONING APPLIES TO NON-UNITY VALUES OF W.

PARAMETER TO BE VARIED = MU

ITER-ATION		
1	.00010000	0.00000000
2	5.0000E-05	0.
3	0.00000000	0.00000000

THESE VALUES ARE COMPUTED VIA ITERATIONS OF THE EQUATION SHOWN IN SECTION 3.2.3. THEY ARE THEN SUMMED, AS PER THE EQUATION IN 3.2.2, TO FORM SUBCLASS COVERAGE FOR THE STATED CONDITIONS (ISU = 1, MD = 1, ETC.). STAGE COVERAGE, IN TURN, IS COMPUTED VIA THE FIRST EQUATION IN SECTION 3.2, BY MEANS OF WEIGHTING AND TOTALING APPLICABLE SUBCLASS COVERAGES, AND IS THEN PRINTED OUT HERE.

NOTE THAT IN THIS PARTICULAR EXAMPLE, THE USER HAS SPECIFIED (AS INDICATED BY THE LINKAGE TABLE ON PAGE 6-21) THAT THE STAGE CONTAINS ONLY ONE FAULT SUBCLASS. AS A CONSEQUENCE, SUBCLASS AND STAGE COVERAGES ARE EQUAL. IN A LESS TRIVIAL SITUATION, THIS IS FREQUENTLY NOT THE CASE.

RUN 1 RELIABILITY RESULTS FIGURE 6-2 (Cont.-6)

STAGE 1 EQUATION NUMBER 4

T	REL	UNREL	SIMREL	SIMGAIN	SIMRIF
0.000	1.0000000	0.0000000	1.0000000	1.0000000E+00	1.0000000E+35
1000.000	.9983584	.0016416	.9048374	1.1033567E+00	5.7970848E+01
2000.000	.9891672	.0108328	.8187308	1.2081715E+00	1.6733367E+01
3000.000	.9696913	.0303087	.7408182	1.3089464E+00	8.5514083E+00
4000.000	.9401482	.0598518	.6703200	1.4025363E+00	5.5082701E+00
5000.000	.9021363	.0978637	.6065307	1.4873713E+00	4.0205844E+00
6000.000	.8577434	.1422566	.5488116	1.5629105E+00	3.1716526E+00
7000.000	.8090678	.1909322	.4965853	1.6292625E+00	2.6366152E+00
8000.000	.7579842	.2420158	.4493290	1.6869249E+00	2.2755518E+00
9000.000	.7060520	.2939480	.4065697	1.7366077E+00	2.0188277E+00
10000.000	.6545004	.3454996	.3678794	1.7791165E+00	1.8295840E+00
11000.000	.6042532	.3957468	.3328711	1.8152769E+00	1.6857467E+00
12000.000	.5559707	.4440293	.3011942	1.8458878E+00	1.5737831E+00
13000.000	.5100963	.4899037	.2725318	1.8716948E+00	1.4849209E+00
14000.000	.4669006	.5330994	.2465970	1.8933755E+00	1.4132507E+00
15000.000	.4265211	.5734789	.2231302	1.9115349E+00	1.3546615E+00
16000.000	.3889951	.6110049	.2018965	1.9267053E+00	1.3062145E+00
17000.000	.3542873	.6457127	.1826835	1.9393502E+00	1.2657588E+00
18000.000	.3223112	.6776888	.1652989	1.9498691E+00	1.2316879E+00
19000.000	.2929458	.7070542	.1495686	1.9586044E+00	1.2027810E+00
20000.000	.2660485	.7339515	.1353353	1.9658475E+00	1.1780952E+00
21000.000	.2414651	.7585349	.1224564	1.9718452E+00	1.1568928E+00
22000.000	.2190363	.7809637	.1108032	1.9768057E+00	1.1385892E+00
23000.000	.1986032	.8013968	.1002588	1.9809041E+00	1.1227161E+00
24000.000	.1800105	.8199895	.0907180	1.9842871E+00	1.1088947E+00
25000.000	.1631092	.8368908	.0820850	1.9870772E+00	1.0968158E+00
26000.000	.1477581	.8522419	.0742736	1.9893766E+00	1.0862250E+00
27000.000	.1338243	.8661757	.0672055	1.9912704E+00	1.0769115E+00
28000.000	.1211841	.8788159	.0608101	1.9928291E+00	1.0686993E+00
29000.000	.1097224	.8902776	.0550232	1.9941115E+00	1.0614406E+00
30000.000	.0993335	.9006665	.0497871	1.9951659E+00	1.0550108E+00

STAGE 2 EQUATION NUMBER 2

T	REL	UNREL	SIMREL	SIMGAIN	SIMRIF
0.000	1.0000000	0.0000000	1.0000000	1.0000000E+00	1.0000000E+35
1000.000	.9902140	.0097860	.9048374	1.0943557E+00	9.7243202E+00
2000.000	.9644566	.0355434	.8187308	1.1779900E+00	5.0999482E+00
3000.000	.9264218	.0735782	.7408182	1.2505386E+00	3.5225345E+00
4000.000	.8794271	.1205729	.6703200	1.3119511E+00	2.7342790E+00
5000.000	.8263498	.1736502	.6065307	1.3624204E+00	2.2658728E+00
6000.000	.7696121	.2303879	.5488116	1.4023247E+00	1.9583859E+00
7000.000	.7111987	.2888013	.4965853	1.4321784E+00	1.7431180E+00
8000.000	.6526919	.3473081	.4493290	1.4525925E+00	1.5855403E+00
9000.000	.5953165	.4046835	.4065697	1.4642424E+00	1.4664061E+00
10000.000	.5399886	.4600114	.3678794	1.4678413E+00	1.3741412E+00
11000.000	.4873630	.5126370	.3328711	1.4641193E+00	1.3013671E+00
12000.000	.4378782	.5621218	.3011942	1.4538069E+00	1.2431573E+00
13000.000	.3917976	.6082024	.2725318	1.4376217E+00	1.1960956E+00
14000.000	.3492451	.6507549	.2465970	1.4162587E+00	1.1577370E+00
15000.000	.3102364	.6897636	.2231302	1.3903829E+00	1.1262841E+00
16000.000	.2747053	.7252947	.2018965	1.3606242E+00	1.1003851E+00
17000.000	.2425258	.7574742	.1826835	1.3275736E+00	1.0790024E+00

FIGURE 6-2 (Cont.-7)

18000.000	.2135300	.7864700	.1652989	1.2917811E+00	1.0613260E+00
19000.000	.1875224	.8124776	.1495686	1.2537548E+00	1.0467136E+00
20000.000	.1642917	.8357083	.1353353	1.2139609E+00	1.0346490E+00
21000.000	.1436198	.8563802	.1224564	1.1728237E+00	1.0247126E+00
22000.000	.1252882	.8747118	.1108032	1.1307274E+00	1.0165598E+00
23000.000	.1090834	.8909166	.1002588	1.0880173E+00	1.0099050E+00
24000.000	.0948004	.9051996	.0907180	1.0450015E+00	1.0045100E+00
25000.000	.0822453	.9177547	.0820850	1.0019532E+00	1.0001747E+00
26000.000	.0712367	.9287633	.0742736	9.5911271E-01	9.9673023E-01
27000.000	.0616066	.9383934	.0672055	9.1668962E-01	9.9403351E-01
28000.000	.0532006	.9467994	.0608101	8.7486519E-01	9.9196297E-01
29000.000	.0458781	.9541219	.0550232	8.3379451E-01	9.9041510E-01
30000.000	.0395114	.9604886	.0497871	7.9360867E-01	9.8930167E-01

T	REL PRODUCT
0.000	1.0000000
1000.000	.9885885
2000.000	.9540089
3000.000	.8983432
4000.000	.8267918
5000.000	.7454801
6000.000	.6601298
7000.000	.5754080
8000.000	.4947301
9000.000	.4203244
10000.000	.3534228
11000.000	.2944906
12000.000	.2434475
13000.000	.1998545
14000.000	.1630628
15000.000	.1323224
16000.000	.1068590
17000.000	.0859238
18000.000	.0688231
19000.000	.0549339
20000.000	.0437096
21000.000	.0346792
22000.000	.0274427
23000.000	.0216643
24000.000	.0170651
25000.000	.0134150
26000.000	.0105258
27000.000	.0082445
28000.000	.0064471
29000.000	.0050339
30000.000	.0039248

RUN 2 RELIABILITY RESULTS

STAGE 1 EQUATION NUMBER 4

T	REL	UNREL	SIMREL	SIMGAIN	SIMRIF
0.000	1.0000000	0.0000000	1.0000000	1.0000000E+00	1.0000000E+35
1000.000	.9985462	.0014538	.9048374	1.1035642E+00	6.5458759E+01
2000.000	.9902986	.0097014	.8187308	1.2095534E+00	1.8684824E+01
3000.000	.9725751	.0274249	.7408182	1.3128390E+00	9.4505842E+00
4000.000	.9453251	.0546749	.6703200	1.4102593E+00	6.0298192E+00
5000.000	.9098163	.0901837	.6065307	1.5000335E+00	4.3629757E+00
6000.000	.8678538	.1321462	.5488116	1.5813326E+00	3.4143109E+00
7000.000	.8213362	.1786638	.4965853	1.6539681E+00	2.8176655E+00
8000.000	.7720221	.2279779	.4493290	1.7181669E+00	2.4154584E+00
9000.000	.7214221	.2785779	.4065697	1.7744120E+00	2.1302132E+00
10000.000	.6707658	.3292342	.3678794	1.8233305E+00	1.9199724E+00
11000.000	.6210096	.3789904	.3328711	1.8656158E+00	1.7602790E+00
12000.000	.5728642	.4271358	.3011942	1.9019761E+00	1.6360272E+00
13000.000	.5268314	.4731686	.2725318	1.9331006E+00	1.5374396E+00
14000.000	.4832409	.5167591	.2465970	1.9596385E+00	1.4579386E+00
15000.000	.4422859	.5577141	.2231302	1.9821879E+00	1.3929535E+00
16000.000	.4040534	.5959466	.2018965	2.0012898E+00	1.3392199E+00
17000.000	.3685508	.6314492	.1826835	2.0174277E+00	1.2943503E+00
18000.000	.3357268	.6642732	.1652989	2.0310286E+00	1.2565629E+00
19000.000	.3054889	.6945111	.1495686	2.0424664E+00	1.2245036E+00
20000.000	.2777170	.7222830	.1353353	2.0520663E+00	1.1971273E+00
21000.000	.2522737	.7477263	.1224564	2.0601095E+00	1.1736160E+00
22000.000	.2290121	.7709879	.1108032	2.0668376E+00	1.1533214E+00
23000.000	.2077822	.7922178	.1002588	2.0724572E+00	1.1357244E+00
24000.000	.1884343	.8115657	.0907180	2.0771449E+00	1.1204048E+00
25000.000	.1708230	.8291770	.0820850	2.0810503E+00	1.1070194E+00
26000.000	.1548084	.8451916	.0742736	2.0843003E+00	1.0952859E+00
27000.000	.1402580	.8597420	.0672055	2.0870021E+00	1.0849703E+00
28000.000	.1270472	.8729528	.0608101	2.0892460E+00	1.0758771E+00
29000.000	.1150595	.8849405	.0550232	2.0911080E+00	1.0678422E+00
30000.000	.1041870	.8958130	.0497871	2.0926518E+00	1.0607269E+00

STAGE 2 EQUATION NUMBER 2

T	REL	UNREL	SIMREL	SIMGAIN	SIMRIF
0.000	1.0000000	0.0000000	1.0000000	1.0000000E+00	1.0000000E+35
1000.000	.9902140	.0097860	.9048374	1.0943557E+00	9.7243202E+00
2000.000	.9644566	.0355434	.8187308	1.1779900E+00	5.0999482E+00
3000.000	.9264218	.0735782	.7408182	1.2505586E+00	3.9225345E+00
4000.000	.8794271	.1205729	.6703200	1.3119511E+00	2.7342790E+00
5000.000	.8263498	.1736502	.6065307	1.3624204E+00	2.2658728E+00
6000.000	.7696121	.2303879	.5488116	1.4023247E+00	1.9583859E+00
7000.000	.7111987	.2888013	.4965853	1.4321784E+00	1.7431180E+00
8000.000	.6526919	.3473081	.4493290	1.4525925E+00	1.5855403E+00
9000.000	.5953165	.4046835	.4065697	1.4642424E+00	1.4664061E+00
10000.000	.5399886	.4600114	.3678794	1.4678413E+00	1.3741412E+00
11000.000	.4873630	.5126370	.3328711	1.4641193E+00	1.3013671E+00
12000.000	.4378782	.5621218	.3011942	1.4538069E+00	1.2431573E+00
13000.000	.3917976	.6082024	.2725318	1.4376217E+00	1.1960956E+00
14000.000	.3492451	.6507549	.2465970	1.4162587E+00	1.1577370E+00
15000.000	.3102364	.6897636	.2231302	1.3903829E+00	1.1262841E+00
16000.000	.2747053	.7252947	.2018965	1.3606242E+00	1.1003851E+00
17000.000	.2425258	.7574742	.1826835	1.3275736E+00	1.0790024E+00

FIGURE 6-2 (Cont.-9)

18000.000	.2135300	.7864700	.1652989	1.2917811E+00	1.0613260E+00
19000.000	.1875224	.8124776	.1495686	1.2537548E+00	1.0467136E+00
20000.000	.1642917	.8357083	.1353353	1.2139609E+00	1.0346490E+00
21000.000	.1436198	.8563802	.1224564	1.1728237E+00	1.0247126E+00
22000.000	.1252882	.8747118	.1108032	1.1307274E+00	1.0165598E+00
23000.000	.1090834	.8909166	.1002588	1.0880173E+00	1.0099050E+00
24000.000	.0948004	.9051996	.0907180	1.0450015E+00	1.0045100E+00
25000.000	.0822453	.9177547	.0820850	1.0019532E+00	1.0001747E+00
26000.000	.0712367	.9287633	.0742736	9.5911271E-01	9.9673023E-01
27000.000	.0616066	.9383934	.0672055	9.1668962E-01	9.9403351E-01
28000.000	.0532006	.9467994	.0608101	8.7486519E-01	9.9196297E-01
29000.000	.0458781	.9541219	.0550232	8.3379451E-01	9.9041510E-01
30000.000	.0395114	.9604886	.0497871	7.9360867E-01	9.8930167E-01

T	REL PRODUCT
0.000	1.0000000
1000.000	.9887744
2000.000	.9551001
3000.000	.9010147
4000.000	.8313445
5000.000	.7518265
6000.000	.6679108
7000.000	.5841333
8000.000	.5038926
9000.000	.4294745
10000.000	.3622059
11000.000	.3026571
12000.000	.2508448
13000.000	.2064113
14000.000	.1687695
15000.000	.1372132
16000.000	.1109956
17000.000	.0893831
18000.000	.0716877
19000.000	.0572860
20000.000	.0456266
21000.000	.0362315
22000.000	.0286925
23000.000	.0226656
24000.000	.0178636
25000.000	.0140494
26000.000	.0110280
27000.000	.0086408
28000.000	.0067590
29000.000	.0052787
30000.000	.0041166

RUN 3 RELIABILITY RESULTS

STAGE 1 EQUATION NUMBER 4

T	REL	UNREL	SIMREL	SIMGAIN	SIMRIF
0.000	1.0000000	0.0000000	1.0000000	1.0000000E+00	1.0000000E+35
1000.000	.9987387	.0012613	.9048374	1.1037769E+00	7.5446916E+01
2000.000	.9914862	.0085138	.8187308	1.2110039E+00	2.1291123E+01
3000.000	.9756726	.0243274	.7408182	1.3170202E+00	1.0653893E+01
4000.000	.9510108	.0489892	.6703200	1.4187414E+00	6.7296472E+00
5000.000	.9184337	.0815663	.6065307	1.5142411E+00	4.8239183E+00
6000.000	.8794336	.1205664	.5488116	1.6024324E+00	3.7422388E+00
7000.000	.8356671	.1643329	.4965853	1.6828270E+00	3.0633844E+00
8000.000	.7887312	.2112688	.4493290	1.7553535E+00	2.6064946E+00
9000.000	.7400474	.2599526	.4065697	1.8202229E+00	2.2828406E+00
10000.000	.6908143	.3091857	.3678794	1.8778280E+00	2.0444690E+00
11000.000	.6419988	.3580012	.3328711	1.9286708E+00	1.8634821E+00
12000.000	.5943496	.4056504	.3011942	1.9733102E+00	1.7226800E+00
13000.000	.5484224	.4515776	.2725318	2.0123245E+00	1.6109485E+00
14000.000	.5046080	.4953920	.2465970	2.0462863E+00	1.5208219E+00
15000.000	.4631614	.5368386	.2231302	2.0757453E+00	1.4471199E+00
16000.000	.4242286	.5757714	.2018965	2.1012180E+00	1.3861464E+00
17000.000	.3878703	.6121297	.1826835	2.1231816E+00	1.3352015E+00
18000.000	.3540820	.6459180	.1652989	2.1420713E+00	1.2922710E+00
19000.000	.3228109	.6771891	.1495686	2.1582798E+00	1.2558256E+00
20000.000	.2939697	.7060303	.1353353	2.1721585E+00	1.2246850E+00
21000.000	.2674472	.7325528	.1224564	2.1840195E+00	1.1979254E+00
22000.000	.2431174	.7568826	.1108032	2.1941381E+00	1.1748148E+00
23000.000	.2208458	.7791542	.1002588	2.2027564E+00	1.1547665E+00
24000.000	.2004945	.7995055	.0907180	2.2100857E+00	1.1373055E+00
25000.000	.1819258	.8180742	.0820850	2.2163103E+00	1.1220437E+00
26000.000	.1650054	.8349946	.0742736	2.2215897E+00	1.1086616E+00
27000.000	.1496036	.8503964	.0672055	2.2260621E+00	1.0968938E+00
28000.000	.1355971	.8644029	.0608101	2.2298466E+00	1.0865187E+00
29000.000	.1228694	.8771306	.0550232	2.2330457E+00	1.0773501E+00
30000.000	.1113113	.8886887	.0497871	2.2357472E+00	1.0692303E+00

STAGE 2 EQUATION NUMBER 2

T	REL	UNREL	SIMREL	SIMGAIN	SIMRIF
0.000	1.0000000	0.0000000	1.0000000	1.0000000E+00	1.0000000E+35
1000.000	.9902140	.0097860	.9048374	1.0943557E+00	9.7243202E+00
2000.000	.9644566	.0355434	.8187308	1.1779900E+00	5.0999482E+00
3000.000	.9264218	.0735782	.7408182	1.2505386E+00	3.5225345E+00
4000.000	.8794271	.1205729	.6703200	1.3119511E+00	2.7342790E+00
5000.000	.8263498	.1736502	.6065307	1.3624204E+00	2.2658728E+00
6000.000	.7696121	.2303879	.5488116	1.4023247E+00	1.9583859E+00
7000.000	.7111987	.2888013	.4965853	1.4321784E+00	1.7431180E+00
8000.000	.6526919	.3473081	.4493290	1.4525925E+00	1.5855403E+00
9000.000	.5953165	.4046835	.4065697	1.4642424E+00	1.4664061E+00
10000.000	.5399886	.4600114	.3678794	1.4678413E+00	1.3741412E+00
11000.000	.4873630	.5126370	.3328711	1.4641193E+00	1.3013671E+00
12000.000	.4378782	.5621218	.3011942	1.4538069E+00	1.2431573E+00
13000.000	.3917976	.6082024	.2725318	1.4376217E+00	1.1960956E+00
14000.000	.3492451	.6507549	.2465970	1.4162587E+00	1.1577370E+00
15000.000	.3102364	.6897636	.2231302	1.3903829E+00	1.1262841E+00
16000.000	.2747053	.7252947	.2018965	1.3606242E+00	1.1003851E+00
17000.000	.2425258	.7574742	.1826835	1.3275736E+00	1.0790024E+00

FIGURE 6-2 (Cont.-11)

18000.000	.2135300	.7864700	.1652989	1.2917811E+00	1.0613260E+00
19000.000	.1875224	.8124776	.1495686	1.2537548E+00	1.0467136E+00
20000.000	.1642917	.8357083	.1353353	1.2139609E+00	1.0346490E+00
21000.000	.1436198	.8563802	.1224564	1.1728237E+00	1.0247126E+00
22000.000	.1252882	.8747118	.1108032	1.1307274E+00	1.0165598E+00
23000.000	.1090834	.8909166	.1002588	1.0880173E+00	1.0099050E+00
24000.000	.0948004	.9051996	.0907180	1.0450015E+00	1.0045100E+00
25000.000	.0822453	.9177547	.0820850	1.0019532E+00	1.0001747E+00
26000.000	.0712367	.9287633	.0742736	9.5911271E-01	9.9673023E-01
27000.000	.0616066	.9383934	.0672055	9.1668962E-01	9.9403351E-01
28000.000	.0532006	.9467994	.0608101	8.7486519E-01	9.9196297E-01
29000.000	.0458781	.9541219	.0550232	8.3379451E-01	9.9041510E-01
30000.000	.0395114	.9604886	.0497871	7.9360867E-01	9.8930167E-01

T	REL PRODUCT
0.000	1.0000000
1000.000	.9889650
2000.000	.9562454
3000.000	.9038843
4000.000	.8363447
5000.000	.7589474
6000.000	.6768227
7000.000	.5943254
8000.000	.5147984
9000.000	.4405624
10000.000	.3730319
11000.000	.3128864
12000.000	.2602528
13000.000	.2148706
14000.000	.1762319
15000.000	.1436895
16000.000	.1165378
17000.000	.0940686
18000.000	.0756071
19000.000	.0605343
20000.000	.0482968
21000.000	.0384107
22000.000	.0304597
23000.000	.0240906
24000.000	.0190070
25000.000	.0149625
26000.000	.0117544
27000.000	.0092166
28000.000	.0072138
29000.000	.0056370
30000.000	.0043981

FIGURE 6-3

JOB #3 INPUT/OUTPUT LISTINGS

```
 $OPTSON OUTPT1=T,OUTPT3=T,LPLOT=T,PRODT=T,STGOUT=T$

 $VAR1 PROD(1)=7,7$

 $VAR T=1.0E4,STEP=1.0E3,OPTION=1$

10000

100000

IB

 $PARVEC Q1(1)=2,2, Q2(1)=1,1, S(1)=1,1,

     LAM(1)=2*1.0E-4, MU(1)=2*5.0E-5, GMP(1)=2*1.1E-5,

     C1(1)=1.0,0.999,CD1(1)=1.0,0.999,C2(1)=1.0,0.98,CD2(1)=1.0,0.998,

     CTR(1)=1.0,0.999,CDTR(1)=1.0,0.999,PRC1(1)=1.0,0.99,PRC2(1)=1.0,0.95$

ID

 $DATA SLH2=1.0E-7, SLH3=2.0E-8, RSGN=F$

G

S
```

FIGURE 6-3 (Cont.-2)

CARE2 (COMPUTER-AIDED RELIABILITY ESTIMATION)

```
DO YOU WISH TO HAVE YOUR RESPONSES TO MY
QUESTIONS PRINTED BACK FOR VERIFICATION.                  OUTPT3= T
DO YOU WISH TO FORM A COMPLEX EQUATION WHICH IS
THE PRODUCT OF THE PRIMARY EQUATIONS.                     PRODT=  T

$VAR1
PROD    =  7,  7,  0,  0,  0,  0,  0,  0,  0,  0,
$END
DO YOU WISH TO HAVE 2-D RELIABILITY PLOTS=               LPLOT=  T
INPUT A 1 IN THE COLUMN SPECIFIED BELOW IF YOU WISH
THE CORRESPONDING PLOT OPTION, OTHERWISE INPUT 0.
NOTE= WHEN PERFORMING PRODUCT OF RELIABILITIES, NO OTHER
PLOT OPTION BESIDES PRODUCT OF RELIABILITIES MAY BE SPECIFIED.
COLUMN 1 - PLOTS PRODUCT OF RELIABILITIES
COLUMN 2 - PLOTS RELIABILITY
COLUMN 3 - PLOTS DIFF, RIF, AND GAIN
COLUMN 4 - PLOTS MTF AND RELIABILITY AT MTF
COLUMN 5 - PLOTS UNRELIABILITY
10000
FOR ABSCISSA, INPUT 1 IN COLUMN 1 IF ABSCISSA IS T,
1 IN COLUMN 2 IF ABSCISSA IS LOG(T) - BASE 10,
1 IN COLUMN 3 IF ABSCISSA IS LAMT,
1 IN COLUMN 4 IF ABSCISSA IS LOG(LAMT) - BASE 10,
1 IN COLUMN    5 IF ABSCISSA IS EXP(-LAMBDA(T)),
1 IN COLUMN 6 IF ABSCISSA IS LOG(EXP(-LAMT)) - BASE 10.
100000
DO YOU WISH TO HAVE PRINTED TABLE OF RELIABILITY RESULTS  OUTPT1=  T
DO YOU WISH TO HAVE PRINTED TABLE OF DIFF, RIF,
AND GAIN RESULTS =                                        OUTPT2=  F
DO YOU WISH MTF AND RELIABILITY AT MTF RESULTS PRINTED    OUTPT4=  F
DO YOU WANT PRINTED RESULTS OF THE MAXIMUM MISSION
TIME CALCULATIONS =                                       OUTPT5=  F
```

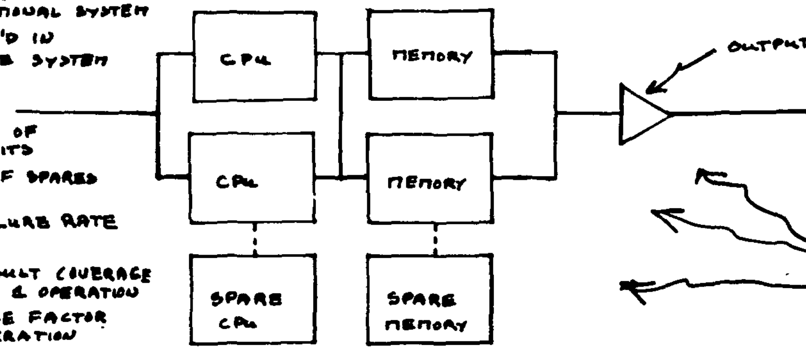
FIGURE 6-3 (Cont.-3)

DATA BASE FOR RUN-SET
1
****

STAGE/EQUATION NUMBER

PARAMETER	1/7	2/7	
Q=Q1	2.00000000	2.00000000	# OF UNITS REQ'D IN FULLY OPERATIONAL SYSTEM
Q2	1.00000000	1.00000000	# OF UNITS REQ'D IN DEGENERATE SYSTEM
S	1	1	# OF SPARES
LAM	.00010000	.00010000	FAILURE RATE OF ON-LINE UNITS
MU	5.0000E-05	5.0000E-05	FAILURE RATE OF SPARES
GMP	1.1000E-05	1.1000E-05	TRANSIENT FAILURE RATE
C=C1	1.00000000	.99900000	PERMANENT FAULT COVERAGE FACTOR - MODE 1 OPERATION
CD1	1.00000000	.99900000	DELTA COVERAGE FACTOR - MODE 1 OPERATION
C2	1.00000000	.98000000	PERMANENT COVERAGE FACTOR - MODE 2 OPERATION
CD2	1.00000000	.99800000	DELTA COVERAGE FACTOR - MODE 2 OPERATION
CTR	1.00000000	.99900000	TRANSITIONAL COVERAGE FACTOR (APPLIES WHEN DEGENERATING FROM MODE 1 TO MODE 2
CDTR	1.00000000	.99900000	DELTA TRANSITIONAL COVERAGE FACTOR (")
PRC1	1.00000000	.99000000	TRANSIENT COVERAGE FACTOR - MODE 1 OPERATION
PRC2	1.00000000	.95000000	" " " " 2 "

CPU AND MEMORY SELECTIONS ARE ARBITRARY



DUAL CHANNEL SYSTEM (USING DUAL MODE MODEL - EQUATION # 7)

ALL COVERAGE VALUES ARE PRE-ASSIGNED, NOT COMPUTED

DUAL MODE SYSTEM PARAMETERS

NUMBER OF STAGES	2	
CHANNEL FAILURE RATE	.00000010	FAILURE RATE OF HARDWARE (E.G. SWITCHES) WHICH WOULD TAKE OUT ENTIRE CHANNEL)
SYSTEM FAILURE RATE	.00000002	"SINGLE POINT" FAILURE RATE (E.G. COMPARATOR)
CHANNEL FAILURE COV.	1.00000000	COVERAGE VALUE TO BE APPLIED IN EVENT OF CHANNEL LOSS DUE TO "SWITCH" FAILURE
REASSIGNMENT SWITCH	F	F (FALSE) SPECIFIES THAT, GIVEN THE LOSS OF A SINGLE CHANNEL, ITS UNITS WHICH ARE STILL FUNCTIONAL CANNOT BE USED AS SPARES FOR THE REMAINING CHANNEL

T (TRUE) = $\overline{F}$

RUN 1 RELIABILITY RESULTS

FIGURE 6-3 (Cont.-4)

COMPUTED BY 2nd EQUATION IN SECTION 3.1.7

COMPUTED BY 3rd EQUATION IN SECTION 3.1.7

DUAL MODE SYSTEM RELIABILITY IN MODE 1

TIME	SYSTEM REL.	STAGE 1	STAGE 2
0.000	1.0000000	1.0000000	1.0000000
1000.000	.9569892	.9784506	.9781833
2000.000	.8558788	.9254776	.9250189
3000.000	.7295625	.8545920	.8540043
4000.000	.6000171	.7751271	.7744604
5000.000	.4799951	.6933780	.6926714
6000.000	.3756112	.6134504	.6127336
7000.000	.2887028	.5378890	.5371840
8000.000	.2186291	.4681415	.4674637
9000.000	.1635062	.4048975	.4042575
10000.000	.1209855	.3483364	.3477407

SYSTEM AND COMPONENT STAGE RELIABILITIES, AT TIME T, IF DEGENERATE SYSTEM OPERATION WERE NOT ALLOWED. USEFUL INFORMATION ONLY FOR SYSTEM ANALYSIS WORK.

COMPUTED BY 1st EQUATION IN SECTION 3.1.7

STAGE 2 EQUATION NUMBER 7

← IDENTIFIES LAST STAGE IN THE SYSTEM, IN THE CASE OF DUAL CHANNEL — WOULD BE CLEARER IF IT READ STAGES 1-2 IN THIS CASE.

← DUAL CHANNEL SYSTEM

NOTE: DUAL CHANNEL SYSTEMS MUST START IN STAGE POSITION 1 ALWAYS. THEY MAY "EXTEND" FROM THERE THRU STAGE POSITION 8, WHERE DESIRED

T	REL	UNREL	SIMREL	SIMGAIN	SIMRIF
0.000	1.0000000	0.0000000	1.0000000	1.0000000E+00	1.0000000E+35
1000.000	.9980114	.0019886	.8187308	1.2189739E+00	9.1153913E+01
2000.000	.9870321	.0129679	.6703200	1.4724788E+00	2.5422673E+01
3000.000	.9616920	.0383080	.5488116	1.7523171E+00	1.1777924E+01
4000.000	.9208261	.0791739	.4493290	2.0493361E+00	6.9552030E+00
5000.000	.8662972	.1337028	.3678794	2.3548400E+00	4.7278046E+00
6000.000	.8015662	.1984338	.3011942	2.6612936E+00	3.5216072E+00
7000.000	.7305658	.2694342	.2465970	2.9625905E+00	2.7962415E+00
8000.000	.6569852	.3430148	.2018965	3.2540691E+00	2.3267320E+00
9000.000	.5839004	.4160996	.1652989	3.5323914E+00	2.0060127E+00
10000.000	.5136460	.4863540	.1353353	3.7953591E+00	1.7778505E+00

THIS IS THE DUAL CHANNEL SYSTEM RELIABILITY WITH ALL COVERAGES, PERMANENT & TRANSIENT FAILURE RATES, OPERATING MODES, SWITCH FAILURES, ET. AL. TAKEN INTO ACCOUNT

T	REL PRODUCT
0.000	1.0000000
1000.000	.9980114
2000.000	.9870321
3000.000	.9616920
4000.000	.9208261
5000.000	.8662972
6000.000	.8015662
7000.000	.7305658
8000.000	.6569852
9000.000	.5839004
10000.000	.5136460

SAME AS REL COLUMN DIRECTLY ABOVE UNLESS ADDITIONAL "SYSTEMS" ARE IN SERIES

FIGURE 6-4

JOB #4 INPUT/OUTPUT LISTINGS

```
 $OPTSON OUTPT1=T, OUTPT3=T, PRODT=T, LSTCH=T,STGOUT=T$

 $VAR1 PROD(1)=7,7$

 $VAR T=1.0E4, STEP=1.0E3, OPTION=1$

IB

 $PARVEC Q1(1)=2,2, Q2(1)=1,1, S(1)=0,1,

      LAM(1)=2*1.0E-4, MU(1)=2*5.0E-5, GMP(1)=2*1.1E-5,

      C1(1)=1.0,0.999,CD1(1)=1.0,0.999,C2(1)=1.0,0.98,CD2(1)=1.0,0.998,

      CTR(1)=1.0,0.999,CDTR(1)=1.0,0.999,PRC1(1)=1.0,0.99,PRC2(1)=1.0,0.95$

ID

 $DATA SLH2=1.0E-7, SLH3=2.0E-8, RSGN=T$

IV        4

 $VARY PARAM(1,1)=0,1,2$

G

S
```

FIGURE 6-4 (Cont.-2)

CARE2 (COMPUTER-AIDED RELIABILITY ESTIMATION)

```
DO YOU WISH TO HAVE YOUR RESPONSES TO MY
QUESTIONS PRINTED BACK FOR VERIFICATION,                  OUTPT3= T
DO YOU WISH TO FORM A COMPLEX EQUATION WHICH IS
THE PRODUCT OF THE PRIMARY EQUATIONS,                     PRODT=  T

$VAR1
PROD    =  7,  7,  0,  0,  0,  0,  0,  0,  0,  0,
$END
DO YOU WISH TO HAVE 2-D RELIABILITY PLOTS=               LPLOT=  F
DO YOU WISH TO HAVE PRINTED TABLE OF RELIABILITY RESULTS  OUTPT1=  T
DO YOU WISH TO HAVE PRINTED TABLE OF DIFF, RIF,
AND GAIN RESULTS =                                        OUTPT2=  F
DO YOU WISH MTF AND RELIABILITY AT MTF RESULTS PRINTED    OUTPT4=  F
DO YOU WANT PRINTED RESULTS OF THE MAXIMUM MISSION
TIME CALCULATIONS =                                       OUTPT5=  F
```

DATA BASE FOR RUN-SET
1
****

STAGE/EQUATION NUMBER

PARAMETER	1/7	2/7
Q=Q1	2.00000000	2.00000000
Q2	1.00000000	1.00000000
S	0	1
LAM	.00010000	.00010000
MU	5.0000E-05	5.0000E-05
GMP	1.1000E-05	1.1000E-05
C=C1	1.00000000	.99900000
CD1	1.00000000	.99900000
C2	1.00000000	.98000000
CD2	1.00000000	.99800000
CTR	1.00000000	.99900000
CDTH	1.00000000	.99900000
PRC1	1.00000000	.99000000
PRC2	1.00000000	.95000000

ITERATION		
1	0.00000000	1.00000000
2	1.00000000	1.00000000
3	2.00000000	1.00000000

IDENTICAL TO JOB 3 EXCEPT:

- REASSIGNMENT SWITCH IS SET TRUE, THEREBY ENABLING THE "CANNIBALIZING" OF OPERATIONAL LRU'S IN A FAILED CHANNEL FOR USE, WHEN AND IF REQUIRED, IN THE REMAINING CHANNEL
- THE QUANTITY OF SPARE LRU'S ASSIGNED TO STAGE 1 IS TO BE VARIED (FROM 0 TO 1 TO 2), WITH RELIABILITY ASSESSMENTS MADE FOR EACH VALUE (I.E., THREE RUNS IN THE RUNSET)

PARAMETER TO BE VARIED = S (S FOR SPARES)

DUAL MODE SYSTEM PARAMETERS

NUMBER OF STAGES	2
CHANNEL FAILURE RATE	.00000010
SYSTEM FAILURE RATE	.00000002
CHANNEL FAILURE COV.	1.00000000
REASSIGNMENT SWITCH	T

FIGURE 6-4 (Cont.-4)

RUN 1 RELIABILITY RESULTS

0 SPARES IN STAGE 1

DUAL MODE SYSTEM RELIABILITY IN MODE 1

TIME	SYSTEM REL.	STAGE 1	STAGE 2
0.000	1.0000000	1.0000000	1.0000000
1000.000	.8007727	.8187308	.9781833
2000.000	.6199099	.6703200	.9250189
3000.000	.4685188	.5488116	.8540043
4000.000	.3478205	.4493290	.7744604
5000.000	.2546667	.3678794	.6926714
6000.000	.1844190	.3011942	.6127336
7000.000	.1323567	.2465970	.5371840
8000.000	.0942887	.2018965	.4674637
9000.000	.0667512	.1652989	.4042575
10000.000	.0470051	.1353353	.3477407

STAGE 2 EQUATION NUMBER 7

T	REL	UNREL	SIMREL	SIMGAIN	SIMRIF
0.000	1.0000000	0.0000000	1.0000000	1.0000000E+00	1.0000000E+35
1000.000	.9897248	.0102752	.8187308	1.2088526E+00	1.7641424E+01
2000.000	.9612376	.0387624	.6703200	1.4339981E+00	8.5051571E+00
3000.000	.9178250	.0821750	.5488116	1.6723862E+00	5.4905791E+00
4000.000	.8632186	.1367814	.4493290	1.9211283E+00	4.0259207E+00
5000.000	.8010786	.1989214	.3678794	2.1775575E+00	3.1777408E+00
6000.000	.7346907	.2653093	.3011942	2.4392590E+00	2.6339286E+00
7000.000	.6668183	.3331817	.2465970	2.7040817E+00	2.2612380E+00
8000.000	.5996600	.4003400	.2018965	2.9701352E+00	1.9935640E+00
9000.000	.5348707	.4651293	.1652989	3.2357790E+00	1.7945570E+00
10000.000	.4736201	.5263799	.1353353	3.4996058E+00	1.6426630E+00

T	REL PRODUCT
0.000	1.0000000
1000.000	.9897248
2000.000	.9612376
3000.000	.9178250
4000.000	.8632186
5000.000	.8010786
6000.000	.7346907
7000.000	.6668183
8000.000	.5996600
9000.000	.5348707
10000.000	.4736201

RUN 2 RELIABILITY RESULTS

DUAL MODE SYSTEM RELIABILITY IN MODE 1

1 SPARE IN STAGE 1

TIME	SYSTEM REL.	STAGE 1	STAGE 2
0.000	1.0000000	1.0000000	1.0000000
1000.000	.9569892	.9784506	.9781833
2000.000	.8558788	.9254776	.9250189
3000.000	.7295625	.8545920	.8540043
4000.000	.6000171	.7751271	.7744604
5000.000	.4799951	.6933780	.6926714
6000.000	.3756112	.6134504	.6127336
7000.000	.2887028	.5378890	.5371840
8000.000	.2186291	.4681415	.4674637
9000.000	.1635062	.4048975	.4042575
10000.000	.1209855	.3483364	.3477407

STAGE 2 EQUATION NUMBER 7

T	REL	UNREL	SIMREL	SIMGAIN	SIMRIF
0.000	1.0000000	0.0000000	1.0000000	1.0000000E+00	1.0000000E+35
1000.000	.9982021	.0017979	.8187308	1.2192068E+00	1.0082331E+02
2000.000	.9891514	.0108486	.6703200	1.4756405E+00	3.0389128E+01
3000.000	.9692395	.0307605	.5488116	1.7660696E+00	1.4667793E+01
4000.000	.9376641	.0623359	.4493290	2.0868098E+00	8.8339292E+00
5000.000	.8953913	.1046087	.3678794	2.4339259E+00	6.0427147E+00
6000.000	.8443679	.1556321	.3011942	2.8034000E+00	4.4901123E+00
7000.000	.7869564	.2130436	.2465970	3.1912657E+00	3.5363800E+00
8000.000	.7255572	.2744428	.2018965	3.5937081E+00	2.9080864E+00
9000.000	.6623751	.3376249	.1652989	4.0071360E+00	2.4722737E+00
10000.000	.5992955	.4007045	.1353353	4.4282277E+00	2.1578610E+00

T	REL PRODUCT
0.000	1.0000000
1000.000	.9982021
2000.000	.9891514
3000.000	.9692395
4000.000	.9376641
5000.000	.8953913
6000.000	.8443679
7000.000	.7869564
8000.000	.7255572
9000.000	.6623751
10000.000	.5992955

NOTE THE IMPROVEMENT IN RELIABILITY DUE TO THE ENABLING OF SPARES CANNIBALIZING (OR RE-ASSIGNMENT). COMPARE THIS COLUMN WITH THAT OF JOB # 3.

FOR EXAMPLE, AT T = 10,000, SYSTEM RELIABILITY HAS GONE FROM 0.514 TO 0.599

NOTE ALSO THAT THE EFFECT IS MUCH LESS PRONOUNCED, FOR THIS SYSTEM, AT A MORE REASONABLE VALUE OF T. FOR EXAMPLE, AT T = 1000, SYSTEM RELIABILITY HAS GONE FROM 0.99801 TO 0.99820

FIGURE 6-4 (Cont.-6)

RUN 3 RELIABILITY RESULTS

2 SPARES IN STAGE 1

DUAL MODE SYSTEM RELIABILITY IN MODE 1

TIME	SYSTEM REL.	STAGE 1	STAGE 2
0.000	1.0000000	1.0000000	1.0000000
1000.000	.9760362	.9979247	.9781833
2000.000	.9120174	.9861812	.9250189
3000.000	.8204658	.9610739	.8540043
4000.000	.7143059	.9227700	.7744604
5000.000	.6046012	.8733781	.6926714
6000.000	.4994950	.8157782	.6127336
7000.000	.4041299	.7529441	.5371840
8000.000	.3211105	.6875805	.4674637
9000.000	.2511594	.6219571	.4042575
10000.000	.1937580	.5578600	.3477407

STAGE 2 EQUATION NUMBER 7

T	REL	UNREL	SIMREL	SIMGAIN	SIMRIF
0.000	1.0000000	0.0000000	1.0000000	1.0000000E+00	1.0000000E+38
1000.000	.9988844	.0011156	.8187308	1.2200401E+00	1.6248047E+02
2000.000	.9935286	.0064714	.6703200	1.4821705E+00	5.0944188E+01
3000.000	.9809743	.0190257	.5488116	1.7874516E+00	2.3714626E+01
4000.000	.9596347	.0403653	.4493290	2.1357064E+00	1.3642205E+01
5000.000	.9291321	.0708679	.3678794	2.5256429E+00	8.9197040E+00
6000.000	.8900379	.1099621	.3011942	2.9550299E+00	6.3549688E+00
7000.000	.8435820	.1564180	.2465970	3.4208936E+00	4.8165995E+00
8000.000	.7913770	.2086230	.2018965	3.9197160E+00	3.8255779E+00
9000.000	.7351865	.2648135	.1652989	4.4476193E+00	3.1520341E+00
10000.000	.6767479	.3232521	.1353353	5.0005282E+00	2.6748928E+00

T	REL PRODUCT
0.000	1.0000000
1000.000	.9988844
2000.000	.9935286
3000.000	.9809743
4000.000	.9596347
5000.000	.9291321
6000.000	.8900379
7000.000	.8435820
8000.000	.7913770
9000.000	.7351865
10000.000	.6767479

FIGURE 6-5

JOB #5 INPUT/OUTPUT LISTINGS

```
 $OPTSON OUTPT1=T, OUTPT3=T, PRODT=T, LSTCH=T, COVPRC=T,STGOUT=T$

 $VAR1 PROD(1)=7,7$

 $VAR T=1.0E4, STEP=1.0F3, OPTION=1$

 $COVCAL COVINT=T, IGENC(2)=1, IGENP(2)=1, IFSC(2)=2, FRAC(2)=1.0$

IB

 $PARVEC Q1(1)=2,2, Q2(1)=1,1, S(1)=1,1,

     LAM(1)=2*1.0E-4, MU(1)=2*5.0E-5, GMP(1)=2*1.1E-5$

ID

 $DATA SLH2=1.0E-7, SLH3=2.0E-8, RSGN=F, TMINOR=1000.0$

IV       8

 $VARY PARAM(1,1)=5.0E-5,1.0E-4$

F  D     9   310        1.0       0.0       .091011   .09      0.0      50.0

F  D    10   311    1   .95       0.0       .721796   .7       0.0       5.0

F  I     1   01         1.0       0.5

F  I     2   11         .99       0.0       .04                          25.0

F  I     3   01         0.8       0.0
```

FIGURE 6-5 (Cont.-2)

```
F  E    1  11      1.0                1.0                          10000.0

F  E    2  31      1.0                1.0     .00625     0.0       200.0

F  T    1  11      .9999              1.0                          10000.0

F  T    2  31      .9999              1.0     .023       0.0       200.0

M

             2   2  9  2  1  2         9  3  1  2          9  2  1  2

             2   4 10  1  1  1 10  1  2  2                10  1  1  1

W

PC

N

PF

G

IB

 $PARVEC LAM(1)=1.0E-3$

IV      8

 $VARY PARAM(1,1)=5.0E-4,1.0E-3$

G

S
```

```
DO YOU WISH TO HAVE YOUR RESPONSES TO MY
QUESTIONS PRINTED BACK FOR VERIFICATION,                     OUTPT3= T
DO YOU WISH TO FORM A COMPLEX EQUATION WHICH IS
THE PRODUCT OF THE PRIMARY EQUATIONS,                        PRODT= T

SVAR1
PROD    = 7, 7, 0, 0, 0, 0, 0, 0, 0, 0,
SEND
DO YOU WISH TO HAVE 2-D RELIABILITY PLOTS-                  LPLOT= F
DO YOU WISH TO HAVE PRINTED TABLE OF RELIABILITY RESULTS     OUTPT1= T
DO YOU WISH TO HAVE PRINTED TABLE OF DIFF, RIF,
AND GAIN RESULTS -                                           OUTPT2= F
DO YOU WISH MTF AND RELIABILITY AT MTF RESULTS PRINTED       OUTPT4= F
DO YOU WANT PRINTED RESULTS OF THE MAXIMUM MISSION
TIME CALCULATIONS -                                          OUTPT5= F
```

COVERAGE DATA BASE
****

- FAULT SUBCLASS LINKAGE AND PARAMETERS -

SAME SYSTEM CONFIGURATION AS JOB # 3, BUT WITH CALCULATED, RATHER THAN PRESET, COVERAGE FOR STAGE 2

FAULT SUBCLASS	LINKED TO STAGE	FRACTION OF FAULT RATE	PROB. OF FAULT DET. IN SPARES	TIME FOR FAULT DET. IN SPARES
1	0	0.000000	1.000000	0.000000
2	2	1.000000	1.000000	0.000000
3	0	0.000000	1.000000	0.000000
4	0	0.000000	1.000000	0.000000
5	0	0.000000	1.000000	0.000000
6	0	0.000000	1.000000	0.000000
7	0	0.000000	1.000000	0.000000
8	0	0.000000	1.000000	0.000000

MINOR CYCLE DURATION (FOR SCHEDULED DETECTORS) = 1000.000000

- D/I/R MECHANISM DEFINITIONS -   (X MEANS FUNCTION UNDEFINED)

FAULT TYPE/MODE

SUBCLASS	MECHANISM	PERM/1 D	I	E	T	PERM/2 D	I	E	T	TRANS/1 D	I	E	T	TRANS/2 D	I	E	T	PERM/0 D	I	E	T
2	2	9	2	1	2	0	0	0	0	9	3	1	2	0	0	0	0	9	2	1	2
2	4	10	1	1	1	10	1	2	2	0	0	0	0	0	0	0	0	10	1	1	1

SIMILIAR TO JOB # 2 EXCEPT THAT THE DUAL MODE MODEL ACCEPTS ADDITIONAL FORMS OF COVERAGE.

```
NORMALIZATION COMPLETE FOR DETECTION  FUNCTIONS
NORMALIZATION COMPLETE FOR ISOLATION  FUNCTIONS
NORMALIZATION COMPLETE FOR E. P. REC. FUNCTIONS
NORMALIZATION COMPLETE FOR T. L. REC. FUNCTIONS
```

COVERAGE FUNCTION SPECIFICATIONS

FIGURE 6-5 (Cont.-4)

- DETECTION -

FUNCTION NUMBER	IGFT	INTF	ISCH	IREP	COEF	TDEL	P1	P2	P3	TDUR
9	3	1	0	0	1.000000	0.	9.1011E-02	9.0000E-02	0.	5.0000E+01
10	3	1	1	1	.950000	0.	7.2180E-01	7.0000E-01	0.	5.0000E+00

- ISOLATION -

FUNCTION NUMBER	IGFT	INTF	COEF	TDEL	P1	P2	P3	TDUR
1	0	1	1.000000	5.0000E-01	-0.	-0.	-0.	-0.
2	1	1	.990000	0.	4.0000E-02	-0.	-0.	2.5000E+01
3	0	1	.800000	0.	-0.	-0.	-0.	-0.

- FAULT PROPAGATION RECOVERY -

FUNCTION NUMBER	IGFT	INTF	COEF	P1	P2	P3	TDUR
1	1	1	1.000000	1.0000E+00	-0.	-0.	1.0000E+04
2	3	1	1.000000	1.0000E+00	6.2500E-03	0.	2.0000E+02

- TIME LOST RECOVERY -

FUNCTION NUMBER	IGPT	INTF	COEF	P1	P2	P3	TDUR
1	1	1	.999900	1.0000E+00	-0.	-0.	1.0000E+04
2	3	1	.999900	1.0000E+00	2.3000E-02	0.	2.0000E+02

UNLIKE JOB # 2, ISOLATION AND RECOVERY FUNCTIONS ARE NOW DEFINED TO BE LESS THAN PERFECT

COVERAGE CONTRIBUTIONS - ISU = 2 MD = 1 IET = 1 JS = 1

MECHANISM	CONTRIBUTION
4	.01583
2	.59925

COVERAGE CONTRIBUTIONS - ISU = 2 MD = 1 IET = 1 JS = 2

MECHANISM	CONTRIBUTION
4	.01583
2	.59925

COVERAGE CONTRIBUTIONS - ISU = 2 MD = 2 IET = 1 JS = 1

MECHANISM	CONTRIBUTION
4	.03200

COVERAGE CONTRIBUTIONS - ISU = 2 MD = 2 IET = 1 JS = 2

MECHANISM	CONTRIBUTION
4	.03200

COVERAGE CONTRIBUTIONS - ISU = 2 MD = 0 IET = 1 JS = 0

MECHANISM	CONTRIBUTION
4	.01593
2	.59925

"CONTRIBUTION" TELLS NET WORTH OF EACH DETECTION/ISOLATION/RECOVERY MECHANISM (WITH RESPECT TO ITS CONTRIBUTION TO THE WHOLE OF COVERAGE). LOW VALUES INDICATE (OBVIOUSLY ENOUGH) THAT THE PARTICULAR MECHANISM CONTRIBUTES LITTLE TO SYSTEM RELIABILITY, AND IS PERHAPS A CANDIDATE FOR REMOVAL FROM THE SYSTEM

FIGURE 6-5 (Cont.-5)

```
COVERAGE CONTRIBUTIONS -  ISU = 2 MD = 0 IET = 1 JS = 1
     MECHANISM  CONTRIBUTION
        4          .01583
        2          .59925

COVERAGE CONTRIBUTIONS =  ISU = 2 MD = 1 IET = 2 JS = 0
     MECHANISM  CONTRIBUTION
        2          .64202

COVERAGE CONTRIBUTIONS -  ISU = 2 MD = 2 IET = 2 JS = 0
     MECHANISM  CONTRIBUTION
```

FIGURE 6-5 (Cont.-6)

DATA BASE FOR RUN-SET
1
****

STAGE/EQUATION NUMBER

PARA-METER	1/7	2/7
Q=Q1	2.00000000	2.00000000
Q2	1.00000000	1.00000000
S	1	1
LAM	.00010000	.00010000
MU	5.0000E-05	5.0000E-05
GMP	1.1000E-05	1.1000E-05
C=C1	1.00000000	.61507954
CD1	1.00000000	1.00000000
C2	1.00000000	.03200396
CD2	1.00000000	1.00000000
CTR	1.00000000	.61507954
CDTR	1.00000000	1.00000000
PRC1	1.00000000	.64201783
PRC2	1.00000000	0.00000000

PARAMETER TO BE VARIED = MU

ITER-ATION		
1	5.0000E-05	5.0000E-05
2	1.0000E-04	5.0000E-05

DUAL MODE SYSTEM PARAMETERS

NUMBER OF STAGES	2
CHANNEL FAILURE RATE	.00000010
SYSTEM FAILURE RATE	.00000002
CHANNEL FAILURE COV.	1.00000000
REASSIGNMENT SWITCH	F

RUN 1 RELIABILITY RESULTS

FIGURE 6-5 (Cont.-7)

DUAL MODE SYSTEM RELIABILITY IN MODE 1

TIME	SYSTEM REL.	STAGE 1	STAGE 2
0.000	1.0000000	1.0000000	1.0000000
1000.000	.8935777	.9784506	.9133674
2000.000	.7594244	.9254776	.8207726
3000.000	.6221214	.8545920	.7282369
4000.000	.4955078	.7751271	.6395670
5000.000	.3859877	.6933780	.5570113
6000.000	.2953104	.6134504	.4817391
7000.000	.2226007	.5378890	.4141890
8000.000	.1657121	.4681415	.3543188
9000.000	.1220597	.4048975	.3017840
10000.000	.0890893	.3483364	.2560635

STAGE 2 EQUATION NUMBER 7

T	REL	UNREL	SIMREL	SIMGAIN	SIMRIF
0.000	1.0000000	0.0000000	1.0000000	1.0000000E+00	1.0000000E+35
1000.000	.9212702	.0787298	.8187308	1.1252419E+00	2.3024215E+00
2000.000	.8440753	.1559247	.6703200	1.2592124E+00	2.1143541E+00
3000.000	.7665343	.2334657	.5488116	1.3967165E+00	1.9325680E+00
4000.000	.6889366	.3110634	.4493290	1.5332565E+00	1.7702853E+00
5000.000	.6126042	.3873958	.3678794	1.6652309E+00	1.6317177E+00
6000.000	.5391237	.4608763	.3011942	1.7899537E+00	1.5162546E+00
7000.000	.4699096	.5300904	.2465970	1.9055772E+00	1.4212726E+00
8000.000	.4060078	.5939922	.2018965	2.0109698E+00	1.3436262E+00
9000.000	.3480503	.6519497	.1652989	2.1055818E+00	1.2803153E+00
10000.000	.2962916	.7037084	.1353353	2.1893149E+00	1.2287258E+00

T	REL PRODUCT
0.000	1.0000000
1000.000	.9212702
2000.000	.8440753
3000.000	.7665343
4000.000	.6889366
5000.000	.6126042
6000.000	.5391237
7000.000	.4699096
8000.000	.4060078
9000.000	.3480503
10000.000	.2962916

RUN 2 RELIABILITY RESULTS

DUAL MODE SYSTEM RELIABILITY IN MODE 1

TIME	SYSTEM REL.	STAGE 1	STAGE 2
0.000	1.0000000	1.0000000	1.0000000
1000.000	.8900207	.9745558	.9133674
2000.000	.7494620	.9133369	.8207726
3000.000	.6068182	.8332956	.7282369
4000.000	.4766313	.7455985	.6395670
5000.000	.3659473	.6573780	.5570113
6000.000	.2758306	.5729849	.4817391
7000.000	.2048609	.4948780	.4141890
8000.000	.1501768	.4242536	.3543188
9000.000	.1089728	.3614856	.3017840
10000.000	.0783719	.3064317	.2560635

STAGE 2 EQUATION NUMBER 7

T	REL	UNREL	SIMREL	SIMGAIN	SIMRIF
0.000	1.0000000	0.0000000	1.0000000	1.0000000E+00	1.0000000E+35
1000.000	.9211050	.0788950	.8187308	1.1250402E+00	2.2976025E+00
2000.000	.8430630	.1569370	.6703200	1.2577022E+00	2.1007149E+00
3000.000	.7639538	.2360462	.5488116	1.3920146E+00	1.9114410E+00
4000.000	.6843640	.3156360	.4493290	1.5230800E+00	1.7446393E+00
5000.000	.6059776	.3940224	.3678794	1.6472178E+00	1.6042756E+00
6000.000	.5306737	.4693263	.3011942	1.7618986E+00	1.4889550E+00
7000.000	.4600469	.5399531	.2465970	1.8655821E-00	1.3953120E-00
8000.000	.3952172	.6047828	.2018965	1.9575237E+00	1.3196531E+00
9000.000	.3368106	.6631894	.1652989	2.0375856E+00	1.2586166E+00
10000.000	.2850254	.7149746	.1353353	2.1060686E+00	1.2093642E+00

T	REL PRODUCT
0.000	1.0000000
1000.000	.9211050
2000.000	.8430630
3000.000	.7639538
4000.000	.6843640
5000.000	.6059776
6000.000	.5306737
7000.000	.4600469
8000.000	.3952172
9000.000	.3368106
10000.000	.2850254

DATA BASE FOR RUN-SET 2 FIGURE 6-5 (Cont.-9)

****

STAGE/EQUATION NUMBER

PARA-METER	1/7	2/7
Q=Q1	2.00000000	2.00000000
Q2	1.00000000	1.00000000
S	1	1
LAM	.00100000	.00010000
MU	5.0000E-05	5.0000E-05
GMP	1.1000E-05	1.1000E-05
C=C1	1.00000000	.61507954
CD1	1.00000000	1.00000000
C2	1.00000000	.03200396
CD2	1.00000000	1.00000000
CTR	1.00000000	.61507954
CDTR	1.00000000	1.00000000
PRC1	1.00000000	.64201783
PRC2	1.00000000	0.00000000

PARAMETER TO BE VARIED = MU

ITER-ATION		
1	5.0000E-04	5.0000E-05
2	1.0000E-03	5.0000E-05

DUAL MODE SYSTEM PARAMETERS

NUMBER OF STAGES	2
CHANNEL FAILURE RATE	.00000010
SYSTEM FAILURE RATE	.00000002
CHANNEL FAILURE COV.	1.00000000
REASSIGNMENT SWITCH	F

FIGURE 6-5 (Cont.-10)

RUN 1 RELIABILITY RESULTS

DUAL MODE SYSTEM RELIABILITY IN MODE 1

TIME	SYSTEM REL.	STAGE 1	STAGE 2
0.000	1.0000000	1.0000000	1.0000000
1000.000	.3181210	.3483364	.9133674
2000.000	.0530309	.0646264	.8207726
3000.000	.0074118	.0101814	.7282369
4000.000	.0009561	.0014957	.6395670
5000.000	.0001181	.0002121	.5570113
6000.000	.0000142	.0000295	.4817391
7000.000	.0000017	.0000041	.4141890
8000.000	.0000002	.0000006	.3543188
9000.000	.0000000	.0000001	.3017840
10000.000	.0000000	.0000000	.2560635

STAGE 2 EQUATION NUMBER 7

T	REL	UNREL	SIMREL	SIMGAIN	SIMRIF
0.000	1.0000000	0.0000000	1.0000000	1.0000000E+00	1.0000000E+35
1000.000	.7017676	.2982324	.3328711	2.1082263E+00	2.2369430E+00
2000.000	.3119092	.6880908	.1108032	2.8149850E+00	1.2922668E+00
3000.000	.1152445	.8847555	.0368832	3.1245830E+00	1.0885684E+00
4000.000	.0396967	.9603033	.0122773	3.2333315E+00	1.0285528E+00
5000.000	.0133108	.9866892	.0040868	3.2570340E+00	1.0093484E+00
6000.000	.0044177	.9955823	.0013604	3.2474126E+00	1.0030709E+00
7000.000	.0014605	.9985395	.0004528	3.2252197E+00	1.0010091E+00
8000.000	.0004821	.9995179	.0001507	3.1983624E+00	1.0003315E+00
9000.000	.0001590	.9998410	.0000502	3.1697815E+00	1.0001089E+00
10000.000	.0000525	.9999475	.0000167	3.1405665E+00	1.0000358E+00

T	REL PRODUCT
0.000	1.0000000
1000.000	.7017676
2000.000	.3119092
3000.000	.1152445
4000.000	.0396967
5000.000	.0133108
6000.000	.0044177
7000.000	.0014605
8000.000	.0004821
9000.000	.0001590
10000.000	.0000525

RUN 2 RELIABILITY RESULTS

FIGURE 6-5 (Cont.-11)

DUAL MODE SYSTEM RELIABILITY IN MODE 1

TIME	SYSTEM REL.	STAGE 1	STAGE 2
0.000	1.0000000	1.0000000	1.0000000
1000.000	.2798512	.3064317	.9133674
2000.000	.0410201	.0499894	.8207726
3000.000	.0052337	.0071894	.7282369
4000.000	.0006355	.0009941	.6395670
5000.000	.0000755	.0001356	.5570113
6000.000	.0000089	.0000184	.4817391
7000.000	.0000010	.0000025	.4141890
8000.000	.0000001	.0000003	.3543188
9000.000	.0000000	.0000000	.3017840
10000.000	.0000000	.0000000	.2560635

STAGE 2 EQUATION NUMBER 7

T	REL	UNREL	SIMREL	SIMGAIN	SIMRIF
0.000	1.0000000	0.0000000	1.0000000	1.0000000E+00	1.0000000E+35
1000.000	.6802679	.3197321	.3328711	2.0436378E+00	2.0865249E+00
2000.000	.2908902	.7091098	.1108032	2.6252877E+00	1.2539621E+00
3000.000	.1050750	.8949250	.0368832	2.8488615E+00	1.0761984E+00
4000.000	.0358036	.9641964	.0122773	2.9162318E+00	1.0243998E+00
5000.000	.0119488	.9880512	.0040868	2.9237834E+00	1.0079571E+00
6000.000	.0039579	.9960421	.0013604	2.9094675E+00	1.0026079E+00
7000.000	.0013075	.9986925	.0004528	2.8873117E+00	1.0008557E+00
8000.000	.0004315	.9995685	.0001507	2.8623702E+00	1.0002808E+00
9000.000	.0001423	.9998577	.0000502	2.8364462E+00	1.0000922E+00
10000.000	.0000469	.9999531	.0000167	2.8101848E+00	1.0000302E+00

T	REL PRODUCT
0.000	1.0000000
1000.000	.6802679
2000.000	.2908902
3000.000	.1050750
4000.000	.0358036
5000.000	.0119488
6000.000	.0039579
7000.000	.0013075
8000.000	.0004315
9000.000	.0001423
10000.000	.0000469

FIGURE 6-6

JOB #6 INPUT/OUTPUT LISTINGS

```
 $OPTSON OUTPT1=T, OUTPT3=T, PRODT=T, COVPRC=T,STGOUT=T$

 $VAR1 PROD(1)=7,7,7$

 $VAR T=1.6E4, STEP=2.0E3, OPTION=1$

 $COVCAL COVINT=T, IGENC(2)=-1,1, IGENP(2)=1,1,

     IFSC(2)=2,3,3, FRAC(2)=1.0,0.63,0.37$

IB

 $PARVEC Q1(1)=3,1,2, Q2(1)=2,1,1, S(1)=1,2,1, LAM(1)=1.0E-5,2.0E-4,5.0E-5,

     MU(1)=5.0E-6,1.25E-4,2.5E-5, C1(1)=.998, C2(1)=.99, CD1(1)=.995,

     CD2(1)=.99, CTR(1)=.996, CDTR(1)=.995, PRC1(1)=.99, PRC2(1)=.97$

ID

 $DATA SLH2=0.0, SLH3=0.0, RSGN=T, TFDS(4)=100., TMINOR=1000.0$

F  D    1   010       1.0       0.0

F  D    2   010       .25       0.0

F  D    5   010       .025      0.0

F  D   10   311   1   .95       0.0      .721796    0.7      0.0      5.0

F  D   20   01        .9996     0.0

F  I    1   01        1.0       0.5

F  I    3   01        0.8       20.0

F  I    4   11       .99985     0.0      .005                        200.0
```

FIGURE 6-6 (Cont.-2)

```
F  E    1  11      1.0                 1.0                            10000.0

F  E    2  31      1.0                 1.0      .00625     0.0          200.0

F  E    3  11       .96                1.0                            10000.0

F  T    1  11      .9999               1.0                            10000.0

F  T    2  31       .9999              1.0       .023      0.0          200.0

M

         2   4 10  1  1  1 10  1  2  2                             10  1  1  1

         2   7  2  1  1  1  2  1  1  2  2  1  1  1  2  1  1  2  2  1  1  1

         2   9  5  1  1  1  5  1  1  2  5  3  1  1  5  3  1  2  5  1  1  1

         3   1  1  1  1  1  1  1  1  2  1  1  1  1  1  1  1  2  1  1  1  1

         4  11 20  4  1  1 20  4  3  2 20  4  1  1 20  4  3  2 20  4  1  1

W

PC

N

PF

G

ID

 $DATA TMINOR=400.0$

G

S
```

CARE2 (COMPUTER-AIDED RELIABILITY ESTIMATION)   FIGURE 6-6 (Cont. -3)

```
DO YOU WISH TO HAVE YOUR RESPONSES TO MY
QUESTIONS PRINTED BACK FOR VERIFICATION,                  OUTPT3= T
DO YOU WISH TO FORM A COMPLEX EQUATION WHICH IS
THE PRODUCT OF THE PRIMARY EQUATIONS,                     PRODT=  T

$VAM1
PROD    =  7,  7,  7,  0,  0,  0,  0,  0,  0,  0,
$END
DO YOU WISH TO HAVE 2-D RELIABILITY PLOTS=               LPLOT=  F
DO YOU WISH TO HAVE PRINTED TABLE OF RELIABILITY RESULTS  OUTPT1=  T
DO YOU WISH TO HAVE PRINTED TABLE OF DIFF, RIF,
AND GAIN RESULTS =                                        OUTPT2=  F
DO YOU WISH MTF AND RELIABILITY AT MTF RESULTS PRINTED    OUTPT4=  F
DO YOU WANT PRINTED RESULTS OF THE MAXIMUM MISSION
TIME CALCULATIONS =                                       OUTPT5=  F
```

COVERAGE DATA BASE FIGURE 6-6 (Cont. -4)
****

= FAULT SUBCLASS LINKAGE AND PARAMETERS =

STAGE 2 HAS A SINGLE FAULT SUBCLASS

STAGE 3 HAS 2 FAULT SUBCLASSES. IT MIGHT REPRESENT, FOR EXAMPLE, A MEMORY UNIT CONSISTING OF:

- FIRST SUBCLASS - BIT PLANES - ACCOUNTS FOR 63% OF MEMORY FAILURES
- SECOND SUBCLASS - WORD ELECTRONICS - ACCOUNTS FOR 37% OF MEMORY FAILURES

FAULT SUBCLASS	LINKED TO STAGE	FRACTION OF FAULT RATE	PROB. OF FAULT DET. IN SPARES	TIME FOR FAULT DET. IN SPARES
1	0	0.000000	1.000000	0.000000
2	2	1.000000	1.000000	0.000000
3	3	.630000	1.000000	0.000000
4	3	.370000	1.000000	100.000000
5	0	0.000000	1.000000	0.000000
6	0	0.000000	1.000000	0.000000
7	0	0.000000	1.000000	0.000000
8	0	0.000000	1.000000	0.000000

FINITE SPARES CHECKOUT TEST

MINOR CYCLE DURATION (FOR SCHEDULED DETECTORS) = 1000.000000

= D/I/R MECHANISM DEFINITIONS = (X MEANS FUNCTION UNDEFINED)

PARTITIONING INTO SUBCLASSES ENABLES THE APPLICATION OF DIFFERENT D/I/R MECHANISMS TO THE COMPONENTS OF A SINGLE STAGE

		FAULT TYPE/MODE																			
		PERM/1				PERM/2				TRANS/1				TRANS/2				PERM/0			
SUBCLASS	MECHANISM	D	I	E	T	D	I	E	T	D	I	E	T	D	I	E	T	D	I	E	T
2	4	10	1	1	1	10	1	2	2	0	0	0	0	0	0	0	0	10	1	1	1
2	7	2	1	1	1	2	1	1	2	2	1	1	1	2	1	1	2	2	1	1	1
2	9	5	1	1	1	5	1	1	2	5	3	1	1	5	3	1	2	5	1	1	1
3	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	2	1	1	1	1
4	11	20	4	1	1	20	4	3	2	20	4	1	1	20	4	3	2	20	4	1	1

NORMALIZATION COMPLETE FOR DETECTION FUNCTIONS

NORMALIZATION COMPLETE FOR ISOLATION FUNCTIONS

NORMALIZATION COMPLETE FOR E. P. REC. FUNCTIONS

NORMALIZATION COMPLETE FOR T. L. REC. FUNCTIONS

COVERAGE FUNCTION SPECIFICATIONS   FIGURE 6-6 (Cont. -5)

****

- DETECTION -

FUNCTION NUMBER	IGFT	INTF	ISCH	IREP	COEF	TDEL	P1	P2	P3	TDUR
1	0	1	0	0	1.000000	0.	-0.	-0.	-0.	-0.
2	0	1	0	0	.250000	0.	-0.	-0.	-0.	-0.
5	0	1	0	0	.025000	0.	-0.	-0.	-0.	-0.
10	3	1	1	1	.950000	0.	7.2180E-01	7.0000E-01	0.	5.0000E+00
20	0	1	0	0	.999600	0.	-0.	-0.	-0.	-0.

- ISOLATION -

FUNCTION NUMBER	IGFT	INTF	COEF	TDEL	P1	P2	P3	TDUR
1	0	1	1.000000	5.0000E-01	-0.	-0.	-0.	-0.
3	0	1	.800000	2.0000E+01	-0.	-0.	-0.	-0.
4	1	1	.999850	0.	5.0000E-03	-0.	-0.	2.0000E+02

- FAULT PROPAGATION RECOVERY -

FUNCTION NUMBER	IGFT	INTF	COEF	P1	P2	P3	TDUR
1	1	1	1.000000	1.0000E+00	-0.	-0.	1.0000E+04
2	3	1	1.000000	1.0000E+00	6.2500E-03	0.	2.0000E+02
3	1	1	.960000	1.0000E+00	-0.	-0.	1.0000E+04

- TIME LOST RECOVERY -

FUNCTION NUMBER	IGFT	INTF	COEF	P1	P2	P3	TDUR
1	1	1	.999900	1.0000E+00	-0.	-0.	1.0000E+04
2	3	1	.999900	1.0000E+00	2.3000E-02	0.	2.0000E+02

COVERAGE CONTRIBUTIONS - ISU = 2 MD = 1 IET = 1 JS = 1

MECHANISM	CONTRIBUTION
4	.69462
7	.24685
9	.02187

COVERAGE CONTRIBUTIONS - ISU = 2 MD = 1 IET = 1 JS = 2

MECHANISM	CONTRIBUTION
4	.69462
7	.24685
9	.02187

COVERAGE CONTRIBUTIONS - ISU = 2 MD = 2 IET = 1 JS = 1

MECHANISM	CONTRIBUTION
4	.02340
7	.24403
9	.02162

COVERAGE CONTRIBUTIONS - ISU = 2 MD = 2 IET = 1 JS = 2

FIGURE 6-6 (Cont. -6)

```
     MECHANISM  CONTRIBUTION
         4         .02340
         7         .24403
         9         .02162

COVERAGE CONTRIBUTIONS =  ISU = 2 MD = 0 IET = 1 JS = 0
     MECHANISM  CONTRIBUTION
         4         .69462
         7         .24685
         9         .02187

COVERAGE CONTRIBUTIONS =  ISU = 2 MD = 0 IET = 1 JS = 1
     MECHANISM  CONTRIBUTION
         4         .69462
         7         .24685
         9         .02187

COVERAGE CONTRIBUTIONS =  ISU = 2 MD = 1 IET = 2 JS = 0
     MECHANISM  CONTRIBUTION
         7         .24685
         9         .01750

COVERAGE CONTRIBUTIONS =  ISU = 2 MD = 2 IET = 2 JS = 0
     MECHANISM  CONTRIBUTION
         7         .24403
         9         .01105

COVERAGE CONTRIBUTIONS =  ISU = 3 MD = 1 IET = 1 JS = 1
     MECHANISM  CONTRIBUTION
         1         .99990

COVERAGE CONTRIBUTIONS =  ISU = 4 MD = 1 IET = 1 JS = 1
     MECHANISM  CONTRIBUTION
        11         .99935

COVERAGE CONTRIBUTIONS =  ISU = 3 MD = 1 IET = 1 JS = 2
     MECHANISM  CONTRIBUTION
         1         .99990

COVERAGE CONTRIBUTIONS =  ISU = 4 MD = 1 IET = 1 JS = 2
     MECHANISM  CONTRIBUTION
        11         .99935

COVERAGE CONTRIBUTIONS =  ISU = 3 MD = 2 IET = 1 JS = 1
     MECHANISM  CONTRIBUTION
         1         .98847

COVERAGE CONTRIBUTIONS =  ISU = 4 MD = 2 IET = 1 JS = 1
     MECHANISM  CONTRIBUTION
        11         .01881

COVERAGE CONTRIBUTIONS =  ISU = 3 MD = 2 IET = 1 JS = 2
     MECHANISM  CONTRIBUTION
         1         .98847

COVERAGE CONTRIBUTIONS =  ISU = 4 MD = 2 IET = 1 JS = 2
     MECHANISM  CONTRIBUTION
        11        0.00000

COVERAGE CONTRIBUTIONS =  ISU = 3 MD = 0 IET = 1 JS = 0
     MECHANISM  CONTRIBUTION
         1         .99990

COVERAGE CONTRIBUTIONS =  ISU = 4 MD = 0 IET = 1 JS = 0
     MECHANISM  CONTRIBUTION
        11         .99935
```

FIGURE 6-6 (Cont. -7)

```
COVERAGE CONTRIBUTIONS =  ISU = 3 MD = 0 IET = 1 JS = 1
     MECHANISM  CONTRIBUTION
         1         .99990

COVERAGE CONTRIBUTIONS =  ISU = 4 MD = 0 IET = 1 JS = 1
     MECHANISM  CONTRIBUTION
        11         .99935

COVERAGE CONTRIBUTIONS =  ISU = 3 MD = 1 IET = 2 JS = 0
     MECHANISM  CONTRIBUTION
         1         .99990

COVERAGE CONTRIBUTIONS =  ISU = 4 MD = 1 IET = 2 JS = 0
     MECHANISM  CONTRIBUTION
        11         .99935

COVERAGE CONTRIBUTIONS =  ISU = 3 MD = 2 IET = 2 JS = 0
     MECHANISM  CONTRIBUTION
         1         .98847

COVERAGE CONTRIBUTIONS =  ISU = 4 MD = 2 IET = 2 JS = 0
     MECHANISM  CONTRIBUTION
        11         .20647
```

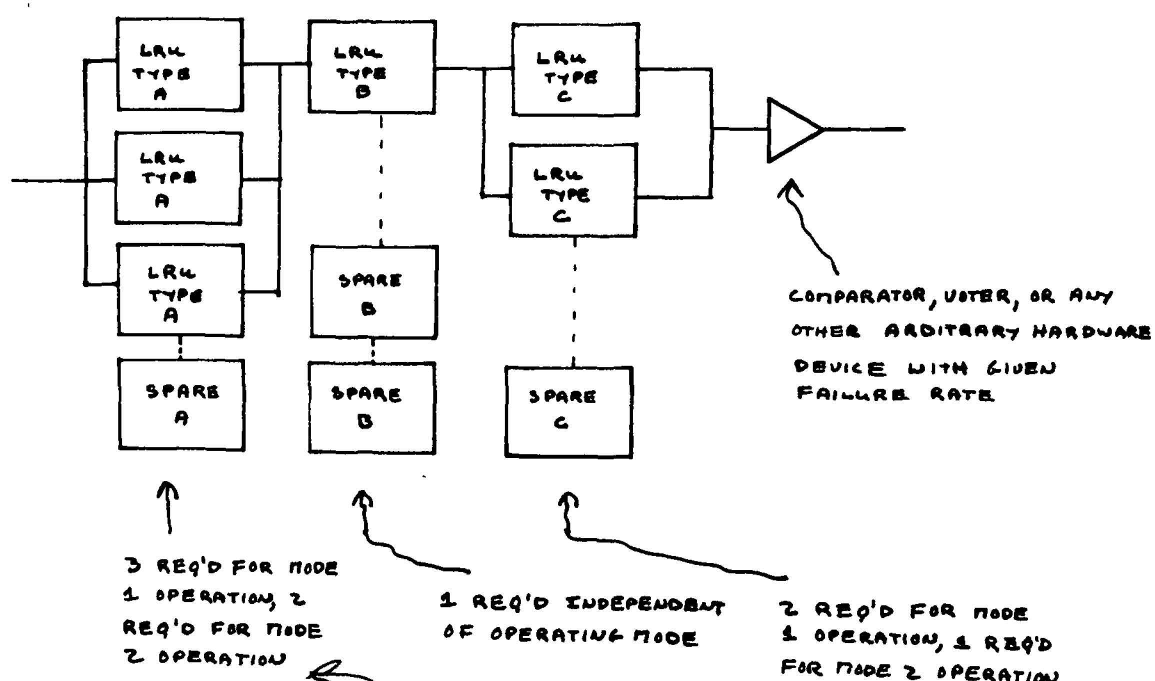
DATA BASE FOR RUN-SET 1 FIGURE 6-6 (Cont. -8)

****

STAGE/EQUATION NUMBER

PARAMETER	1/7	2/7	3/7
Q=Q1	3.00000000	1.00000000	2.00000000
Q2	2.00000000	1.00000000	1.00000000
S	1	2	1
LAM	1.0000E-05	2.0000E-04	5.0000E-05
MU	5.0000E-06	1.2500E-04	2.5000E-05
QMP	1.0000E-06	1.0000E-06	1.0000E-06
C=C1	.99800000	.96334470	.99969654
CD1	.99500000	1.00000000	1.00000000
C2	.99000000	.28905341	.62969525
CD2	.99000000	1.00000000	.98894540
CTR	.99600000	.96334470	.99969654
CDTR	.99500000	1.00000000	1.00000000
PRC1	.99000000	.26434856	.99969654
PRC2	.97000000	.25507415	.69912694

HYDRID CONFIGURATION



DUAL MODE SYSTEM PARAMETERS

NUMBER OF STAGES	3
CHANNEL FAILURE RATE	0.00000000
SYSTEM FAILURE RATE	0.00000000
CHANNEL FAILURE COV.	1.00000000
REASSIGNMENT SWITCH	T

VALUES FROM q=q1 and q2

RUN 1 RELIABILITY RESULTS

FIGURE 6-6 (Cont. -9)

DUAL MODE SYSTEM RELIABILITY IN MODE 1

TIME	SYSTEM REL.	STAGE 1	STAGE 2	STAGE 3
0.000	1.0000000	1.0000000	1.0000000	1.0000000
2000.000	.9388961	.9978564	.9616839	.9784016
4000.000	.7854347	.9920432	.8555599	.9253990
6000.000	.5940845	.9830308	.7072456	.8544976
8000.000	.4165251	.9712447	.5533451	.7750263
10000.000	.2760450	.9570697	.4160345	.6932771
12000.000	.1753641	.9408529	.3038842	.6133534
14000.000	.1078572	.9229071	.2173059	.5377984
16000.000	.0646886	.9035139	.1529654	.4680584

STAGE 3 EQUATION NUMBER 7

T	REL	UNREL	SIMREL	SIMGAIN	SIMRIF
0.000	1.0000000	0.0000000	1.0000000	1.0000000E+00	1.0000000E+35
2000.000	.9590514	.0409486	.5945205	1.6131510E+00	9.9021593E+00
4000.000	.8425223	.1574777	.3534547	2.3836785E+00	4.1056319E+00
6000.000	.6803874	.3196126	.2101361	3.2378422E+00	2.4713170E+00
8000.000	.5153285	.4846715	.1249302	4.1249306E+00	1.8054903E+00
10000.000	.3722851	.6277149	.0742736	5.0123490E+00	1.4747562E+00
12000.000	.2596605	.7403395	.0441572	5.8803708E+00	1.2910872E+00
14000.000	.1763588	.8236412	.0262523	6.7178292E+00	1.1822473E+00
16000.000	.1173528	.8826472	.0156076	7.5189753E+00	1.1152729E+00

T	REL PRODUCT
0.000	1.0000000
2000.000	.9590514
4000.000	.8425223
6000.000	.6803874
8000.000	.5153285
10000.000	.3722851
12000.000	.2596605
14000.000	.1763588
16000.000	.1173528

COVERAGE CONTRIBUTIONS = ISU = 2 MD = 1 IET = 1 JS = 1

MECHANISM	CONTRIBUTION
4	.69462
7	.24685
9	.02187

COVERAGE CONTRIBUTIONS = ISU = 2 MD = 1 IET = 1 JS = 2

MECHANISM	CONTRIBUTION
4	.69462
7	.24685
9	.02187

COVERAGE CONTRIBUTIONS = ISU = 2 MD = 2 IET = 1 JS = 1

MECHANISM	CONTRIBUTION
4	.05851
7	.24403
9	.02162

COVERAGE CONTRIBUTIONS = ISU = 2 MD = 2 IET = 1 JS = 2

```
     MECHANISM  CONTRIBUTION
         4          .05851
         7          .24403
         9          .02162

COVERAGE CONTRIBUTIONS =  ISU = 2 MD = 0 IET = 1 JS = 0
     MECHANISM  CONTRIBUTION
         4          .69462
         7          .24685
         9          .02187

COVERAGE CONTRIBUTIONS =  ISU = 2 MD = 0 IET = 1 JS = 1
     MECHANISM  CONTRIBUTION
         4          .69462
         7          .24685
         9          .02187
```

FIGURE 6-6 (Cont. -10)

DATA BASE FOR RUN=SET 2

FIGURE 6-6 (Cont. -11)

****

STAGE/EQUATION NUMBER

PARA-METER	1/7	2/7	3/7
Q=Q1	3.00000000	1.00000000	2.00000000
Q2	2.00000000	1.00000000	1.00000000
S	1	2	1
LAM	1.0000E-05	2.0000E-04	5.0000E-05
MU	5.0000E-06	1.2500E-04	2.5000E-05
GMP	1.0000E-06	1.0000E-06	1.0000E-06
C=C1	.99800000	.96334470	.99969654
CD1	.99500000	1.00000000	1.00000000
C2	.99000000	.32415782	.62969525
CD2	.99000000	1.00000000	.98894540
CTR	.99600000	.96334470	.99969654
CDTR	.99500000	1.00000000	1.00000000
PRC1	.99000000	.26434856	.99969654
PRC2	.97000000	.25507415	.69912694

DUAL MODE SYSTEM PARAMETERS

NUMBER OF STAGES	3
CHANNEL FAILURE RATE	0.00000000
SYSTEM FAILURE RATE	0.00000000
CHANNEL FAILURE COV.	1.00000000
REASSIGNMENT SWITCH	T

DUAL MODE SYSTEM RELIABILITY IN MODE 1

TIME	SYSTEM REL.	STAGE 1	STAGE 2	STAGE 3
0.000	1.0000000	1.0000000	1.0000000	1.0000000
2000.000	.9388961	.9978564	.9616839	.9784016
4000.000	.7854347	.9920432	.8555599	.9253990
6000.000	.5940845	.9830308	.7072456	.8544976
8000.000	.4165251	.9712447	.5533451	.7750263
10000.000	.2760450	.9570697	.4160345	.6932771
12000.000	.1753641	.9408529	.3038842	.6133534
14000.000	.1078972	.9229071	.2173059	.5377984
16000.000	.0646886	.9035139	.1529654	.4680584

STAGE 3 EQUATION NUMBER 7

T	REL	UNREL	SIMREL	SIMGAIN	SIMRIF
0.000	1.0000000	0.0000000	1.0000000	1.0000000E+00	1.0000000E+35
2000.000	.9591357	.0408643	.5945205	1.6132928E+00	9.9225865E+00
4000.000	.8429157	.1570843	.3534947	2.3847913E+00	4.1159118E+00
6000.000	.6811161	.3188839	.2101361	3.2413099E+00	2.4769642E+00
8000.000	.5162436	.4837564	.1249302	4.1322558E+00	1.8089058E+00
10000.000	.3732119	.6267881	.0742736	5.0248278E+00	1.4769369E+00
12000.000	.2604801	.7395199	.0441572	5.8989316E+00	1.2925181E+00
14000.000	.1770195	.8229805	.0262523	6.7429980E+00	1.1831965E+00
16000.000	.1178514	.8821486	.0156076	7.5509169E+00	1.1159032E+00

T	REL PRODUCT
0.000	1.0000000
2000.000	.9591357
4000.000	.8429157
6000.000	.6811161
8000.000	.5162436
10000.000	.3732119
12000.000	.2604801
14000.000	.1770195
16000.000	.1178514

FIGURE 6-7

JOB #7 INPUT/OUTPUT LISTINGS

```
 $OPTSON OUTPT1=T,OUTPT3=T,LPLOT=T,PRODT=T,LSTCH=T,DEFCHNG=T,STGDUT=T$

 $VAR1 PROD(1)=7,7,4,2$

 $VAR T=1.0E4, STEP=1.0E3, OPTION=1$

 $DEFAULT LAMDEF=1.0E-4$

10000

100000

IB

 $PARVEC Q1(1)=2,2,0,1, Q2(1)=1,1, S(1)=4*1, Z(2)=2,

     MU(1)=2*5.0E-5,1.0E-4,0.0, GMP(1)=2*1.1E-5,

     C1(2)=.999, CD1(2)=.999, C2(2)=.98, CD2(2)=.998, CTR(2)=.999, CDTR(2)=.999,

     PRC1(2)=.99, PRC2(2)=.95$

ID

 $DATA SLH2=1.0E-7,SLH3=2.0E-8, RSGN=F$

IV      8

 $VARY PARAM(1,3)=1.0E-4,5.0E-5,0.0, PARAM(1,4)=0.0$

G

S
```

```
DO YOU WISH TO HAVE YOUR RESPONSES TO MY
QUESTIONS PRINTED BACK FOR VERIFICATION,                OUTPT3= T
DO YOU WISH TO FORM A COMPLEX EQUATION WHICH IS
THE PRODUCT OF THE PRIMARY EQUATIONS,                   PRODT=  T

$VAR1
PROD    =  7,  7,  4,  2,  0,  0,  0,  0,  0,  0,
$END
DO YOU WISH TO HAVE 2-D RELIABILITY PLOTS-             LPLOT=  T
INPUT A 1 IN THE COLUMN SPECIFIED BELOW IF YOU WISH
THE CORRESPONDING PLOT OPTION, OTHERWISE INPUT 0,
NOTE- WHEN PERFORMING PRODUCT OF RELIABILITIES, NO OTHER
PLOT OPTION BESIDES PRODUCT OF RELIABILITIES MAY BE SPECIFIED,
COLUMN 1 - PLOTS PRODUCT OF RELIABILITIES
COLUMN 2 - PLOTS RELIABILITY
COLUMN 3 - PLOTS DIFF, RIF, AND GAIN
COLUMN 4 - PLOTS MTF AND RELIABILITY AT MTF
COLUMN 5 - PLOTS UNRELIABILITY
10000
FOR ABSCISSA, INPUT 1 IN COLUMN 1 IF ABSCISSA IS T,
1 IN COLUMN 2 IF ABSCISSA IS LOG(T) - BASE 10,
1 IN COLUMN 3 IF ABSCISSA IS LAMT,
1 IN COLUMN 4 IF ABSCISSA IS LOG(LAMT) - BASE 10,
1 IN COLUMN    5 IF ABSCISSA IS EXP(-LAMBDA(T)),
1 IN COLUMN 6 IF ABSCISSA IS LOG(EXP(-LAMT)) - BASE 10,
100000
DO YOU WISH TO HAVE PRINTED TABLE OF RELIABILITY RESULTS    OUTPT1=  T
DO YOU WISH TO HAVE PRINTED TABLE OF DIFF, RIF,
AND GAIN RESULTS -                                      OUTPT2=  F
DO YOU WISH MTF AND RELIABILITY AT MTF RESULTS PRINTED  OUTPT4=  F
DO YOU WANT PRINTED RESULTS OF THE MAXIMUM MISSION
TIME CALCULATIONS -                                     OUTPT5=  F
```

FIGURE 6-7 (Cont.-3)

DATA BASE FOR RUN-SET 1

****

STAGE/EQUATION NUMBER

PARAMETER	1/7	2/7	3/4	4/2
Q=Q1	2.00000000	2.00000000		1.00000000
Q2	1.00000000	1.00000000		
S	1	1	1	1
W			1	1
Z			1	1
LAM	.00010000	.00010000	.00010000	.00010000
MU	5.0000E-05	5.0000E-05	1.0000E-04	0.
GMP	1.1000E-05	1.1000E-05		
C=C1	1.00000000	.99900000		1.00000000
CD1	1.00000000	.99900000		
C2	1.00000000	.98000000		
CD2	1.00000000	.99800000		
CTR	1.00000000	.99900000		
CDTR	1.00000000	.99900000		
PRC1	1.00000000	.99000000		
PRC2	1.00000000	.95000000		
RV			1.00000000	1.00000000

CONFIGURATION IS THAT OF JOD'S # 3 and # 1 WIRED IN SERIES (I.E. BOTH MUST SUCCEED) WITH EXCEPTION THAT THE STANDBY REPLACEMENT SYSTEM (EQ. # 2) HAS ONLY 1 REPLICATION (3) INSTEAD OF 2

PARAMETER TO BE VARIED = MU

ITERATION				
1	5.0000E-05	5.0000E-05	1.0000E-04	0.
2	5.0000E-05	5.0000E-05	5.0000E-05	0.
3	5.0000E-05	5.0000E-05	0.	0.

DUAL MODE SYSTEM PARAMETERS

NUMBER OF STAGES	2
CHANNEL FAILURE RATE	.00000010
SYSTEM FAILURE RATE	.00000002
CHANNEL FAILURE COV.	1.00000000
REASSIGNMENT SWITCH	F

RUN 1 RELIABILITY RESULTS

FIGURE 6-7 (Cont.-4)

DUAL MODE SYSTEM RELIABILITY IN MODE 1

TIME	SYSTEM REL.	STAGE 1	STAGE 2
0.000	1.0000000	1.0000000	1.0000000
1000.000	.9569892	.9784506	.9781833
2000.000	.8558788	.9254776	.9250189
3000.000	.7295625	.8545920	.8540043
4000.000	.6000171	.7751271	.7744604
5000.000	.4799951	.6933780	.6926714
6000.000	.3756112	.6134504	.6127336
7000.000	.2887028	.5378890	.5371840
8000.000	.2186291	.4681415	.4674637
9000.000	.1635062	.4048975	.4042575
10000.000	.1209855	.3483364	.3477407

STAGE 2 EQUATION NUMBER 7

T	REL	UNREL	SIMREL	SIMGAIN	SIMRIF
0.000	1.0000000	0.0000000	1.0000000	1.0000000E+00	1.0000000E+35
1000.000	.9980114	.0019886	.8187308	1.2189739E+00	9.1153913E+01
2000.000	.9870321	.0129679	.6703200	1.4724788E+00	2.5422673E+01
3000.000	.9616920	.0383080	.5488116	1.7523171E+00	1.1777924E+01
4000.000	.9208261	.0791739	.4493290	2.0493361E+00	6.9552050E+00
5000.000	.8662972	.1337028	.3678794	2.3548400E+00	4.7278046E+00
6000.000	.8015662	.1984338	.3011942	2.6612936E+00	3.5216072E+00
7000.000	.7305658	.2694342	.2465970	2.9625905E+00	2.7962415E+00
8000.000	.6569852	.3430148	.2018965	3.2540691E+00	2.3267320E+00
9000.000	.5839004	.4160996	.1652989	3.5323914E+00	2.0060127E+00
10000.000	.5136460	.4863540	.1353353	3.7953591E+00	1.7778505E+00

SAME VALUES AS JOB #3

STAGE 3 EQUATION NUMBER 4

T	REL	UNREL	SIMREL	SIMGAIN	SIMRIF
0.000	1.0000000	0.0000000	1.0000000	1.0000000E+00	1.0000000E+35
1000.000	.9983584	.0016416	.9048374	1.1033567E+00	5.7970848E+01
2000.000	.9891672	.0108328	.8187308	1.2081715E+00	1.6733367E+01
3000.000	.9696913	.0303087	.7408182	1.3089464E+00	8.5514083E+00
4000.000	.9401482	.0598518	.6703200	1.4025363E+00	5.5082701E+00
5000.000	.9021363	.0978637	.6065307	1.4873713E+00	4.0205844E+00
6000.000	.8577434	.1422566	.5488116	1.5629105E+00	3.1716526E+00
7000.000	.8090678	.1909322	.4965853	1.6292625E+00	2.6366152E+00
8000.000	.7579842	.2420158	.4493290	1.6869249E+00	2.2753518E+00
9000.000	.7060520	.2939480	.4065697	1.7366077E+00	2.0188277E+00
10000.000	.6545004	.3454996	.3678794	1.7791165E+00	1.8295840E+00

SAME VALUES AS JOB #1

STAGE 4 EQUATION NUMBER 2

T	REL	UNREL	SIMREL	SIMGAIN	SIMRIF
0.000	1.0000000	0.0000000	1.0000000	1.0000000E+00	1.0000000E+35
1000.000	.9953212	.0046788	.9048374	1.1000000E+00	2.0338926E+01
2000.000	.9824769	.0175231	.8187308	1.2000000E+00	1.0344590E+01
3000.000	.9630637	.0369363	.7408182	1.3000000E+00	7.0169911E+00
4000.000	.9384481	.0615519	.6703200	1.4000000E+00	5.3561263E+00
5000.000	.9097960	.0902040	.6065307	1.5000000E+00	4.3619939E+00
6000.000	.8780986	.1219014	.5488116	1.6000000E+00	3.7012572E+00

DIFFERENT VALUES THAN JOB #1 DUE TO A CHANGE IN λ

7000.000	.8441950	.1558050	.4965853	1.7000000E+00	3.2310564E+00
8000.000	.8087921	.1912079	.4493290	1.8000000E+00	2.8799602E+00
9000.000	.7724824	.2275176	.4065697	1.9000000E+00	2.6082827E+00
10000.000	.7357589	.2642411	.3678794	2.0000000E+00	2.3922112E+00

FIGURE 6-7 (Cont.-5)

T	REL PRODUCT
0.000	1.0000000
1000.000	.9917112
2000.000	.9592312
3000.000	.8980997
4000.000	.8124266
5000.000	.7110221
6000.000	.6037263
7000.000	.4989845
8000.000	.4027659
9000.000	.3184667
10000.000	.2473485

RELIABILITY OF THE TOTAL SYSTEM

DUAL MODE SYSTEM RELIABILITY IN MODE 1

TIME	SYSTEM REL.	STAGE 1	STAGE 2
0.000	1.0000000	1.0000000	1.0000000
1000.000	.9569892	.9784506	.9781833
2000.000	.8558788	.9254776	.9250189
3000.000	.7295625	.8545920	.8540043
4000.000	.6000171	.7751271	.7744604
5000.000	.4799951	.6933780	.6926714
6000.000	.3756112	.6134504	.6127336
7000.000	.2887028	.5378890	.5371840
8000.000	.2186291	.4681415	.4674637
9000.000	.1635062	.4048975	.4042575
10000.000	.1209855	.3483364	.3477407

STAGE 2 EQUATION NUMBER 7

T	REL	UNREL	SIMREL	SIMGAIN	SIMRIF
0.000	1.0000000	0.0000000	1.0000000	1.0000000E+00	1.0000000E+35
1000.000	.9980114	.0019886	.8187308	1.2189739E+00	9.1153915E+01
2000.000	.9870321	.0129679	.6703200	1.4724788E+00	2.5422673E+01
3000.000	.9616920	.0383080	.5488116	1.7523171E+00	1.1777924E+01
4000.000	.9208261	.0791739	.4493290	2.0493361E+00	6.9552050E+00
5000.000	.8662972	.1337028	.3678794	2.3548400E+00	4.7278046E+00
6000.000	.8015662	.1984338	.3011942	2.6612936E+00	3.5216072E+00
7000.000	.7305658	.2694342	.2465970	2.9625905E+00	2.7962415E+00
8000.000	.6569852	.3430148	.2018965	3.2540691E+00	2.3267320E+00
9000.000	.5839004	.4160996	.1652989	3.5323914E+00	2.0060127E+00
10000.000	.5136460	.4863540	.1353353	3.7953591E+00	1.7778505E+00

STAGE 3 EQUATION NUMBER 4

T	REL	UNREL	SIMREL	SIMGAIN	SIMRIF
0.000	1.0000000	0.0000000	1.0000000	1.0000000E+00	1.0000000E+35
1000.000	.9985462	.0014538	.9048374	1.1035642E+00	6.5458759E+01
2000.000	.9902986	.0097014	.8187308	1.2095534E+00	1.8684824E+01
3000.000	.9725751	.0274249	.7408182	1.3128390E+00	9.4505842E+00
4000.000	.9453251	.0546749	.6703200	1.4102593E+00	6.0298192E+00
5000.000	.9098163	.0901837	.6065307	1.5000335E+00	4.3629757E+00
6000.000	.8678538	.1321462	.5488116	1.5813326E+00	3.4143109E+00
7000.000	.8213362	.1786638	.4965853	1.6539681E+00	2.8176655E+00
8000.000	.7720221	.2279779	.4493290	1.7181669E+00	2.4154584E+00
9000.000	.7214221	.2785779	.4065697	1.7744120E+00	2.1302132E+00
10000.000	.6707658	.3292342	.3678794	1.8233305E+00	1.9199724E+00

STAGE 4 EQUATION NUMBER 2

T	REL	UNREL	SIMREL	SIMGAIN	SIMRIF
0.000	1.0000000	0.0000000	1.0000000	1.0000000E+00	1.0000000E+35
1000.000	.9953212	.0046788	.9048374	1.1000000E+00	2.0338926E+01
2000.000	.9824769	.0175231	.8187308	1.2000000E+00	1.0344590E+01
3000.000	.9630637	.0369363	.7408182	1.3000000E+00	7.0169911E+00
4000.000	.9384481	.0615519	.6703200	1.4000000E+00	5.3561265E+00
5000.000	.9097960	.0902040	.6065307	1.5000000E+00	4.3619939E+00

6000.000	.8780986	.1219014	.5488116	1.6000000E+00	3.7012572E+00
7000.000	.8441950	.1558050	.4965853	1.7000000E+00	3.2310564E+00
8000.000	.8087921	.1912079	.4493290	1.8000000E+00	2.8799602E+00
9000.000	.7724824	.2275176	.4065697	1.9000000E+00	2.6082827E+00
10000.000	.7357589	.2642411	.3678794	2.0000000E+00	2.3922112E+00

FIGURE 6-7 (Cont.-7)

T	REL PRODUCT
0.000	1.0000000
1000.000	.9918978
2000.000	.9603284
3000.000	.9007705
4000.000	.8169002
5000.000	.7170751
6000.000	.6108425
7000.000	.5065509
8000.000	.4102251
9000.000	.3253994
10000.000	.2534956

RUN 3 RELIABILITY RESULTS

FIGURE 6-7 (Cont.-8)

DUAL MODE SYSTEM RELIABILITY IN MODE 1

TIME	SYSTEM REL.	STAGE 1	STAGE 2
0.000	1.0000000	1.0000000	1.0000000
1000.000	.9569892	.9784506	.9781833
2000.000	.8558788	.9254776	.9250189
3000.000	.7295625	.8545920	.8540043
4000.000	.6000171	.7751271	.7744604
5000.000	.4799951	.6933780	.6926714
6000.000	.3756112	.6134504	.6127336
7000.000	.2887028	.5378890	.5371840
8000.000	.2186291	.4681415	.4674637
9000.000	.1635062	.4048975	.4042575
10000.000	.1209855	.3483364	.3477407

STAGE 2 EQUATION NUMBER 7

T	REL	UNREL	SIMREL	SIMGAIN	SIMRIF
0.000	1.0000000	0.0000000	1.0000000	1.0000000E+00	1.0000000E+35
1000.000	.9980114	.0019886	.8187308	1.2189739E+00	9.1153913E+01
2000.000	.9870321	.0129679	.6703200	1.4724788E+00	2.5422673E+01
3000.000	.9616920	.0383080	.5488116	1.7523171E+00	1.1777924E+01
4000.000	.9208261	.0791739	.4493290	2.0493361E+00	6.9552050E+00
5000.000	.8662972	.1337028	.3678794	2.3548400E+00	4.7278046E+00
6000.000	.8015662	.1984338	.3011942	2.6612936E+00	3.5216072E+00
7000.000	.7305658	.2694342	.2465970	2.9625905E+00	2.7962415E+00
8000.000	.6569852	.3430148	.2018965	3.2540691E+00	2.3267320E+00
9000.000	.5839004	.4160996	.1652989	3.5323914E+00	2.0060127E+00
10000.000	.5136460	.4863540	.1353353	3.7953591E+00	1.7778505E+00

STAGE 3 EQUATION NUMBER 4

T	REL	UNREL	SIMREL	SIMGAIN	SIMRIF
0.000	1.0000000	0.0000000	1.0000000	1.0000000E+00	1.0000000E+35
1000.000	.9987387	.0012613	.9048374	1.1037769E+00	7.5446916E+01
2000.000	.9914862	.0085138	.8187308	1.2110039E+00	2.1791123E+01
3000.000	.9756726	.0243274	.7408182	1.3170202E+00	1.0653893E+01
4000.000	.9510108	.0489892	.6703200	1.4187414E+00	6.7296472E+00
5000.000	.9184337	.0815663	.6065307	1.5142411E+00	4.8239183E+00
6000.000	.8794336	.1205664	.5488116	1.6024324E+00	3.7422388E+00
7000.000	.8356671	.1643329	.4965853	1.6828270E+00	3.0633844E+00
8000.000	.7887312	.2112688	.4493290	1.7553535E+00	2.6064946E+00
9000.000	.7400474	.2599526	.4065697	1.8202229E+00	2.2828406E+00
10000.000	.6908143	.3091857	.3678794	1.8778280E+00	2.0444690E+00

STAGE 4 EQUATION NUMBER 2

T	REL	UNREL	SIMREL	SIMGAIN	SIMRIF
0.000	1.0000000	0.0000000	1.0000000	1.0000000E+00	1.0000000E+35
1000.000	.9953212	.0046788	.9048374	1.1000000E+00	2.0338926E+01
2000.000	.9824769	.0175231	.8187308	1.2000000E+00	1.0344590E+01
3000.000	.9630637	.0369363	.7408182	1.3000000E+00	7.0169911E+00
4000.000	.9384481	.0615519	.6703200	1.4000000E+00	5.3561265E+00
5000.000	.9097960	.0902040	.6065307	1.5000000E+00	4.3619939E+00

6000.000	.8780986	.1219014	.5488116	1.6000000E+00	3.7012572E+00
7000.000	.8441950	.1558050	.4965853	1.7000000E+00	3.2310364E+00
8000.000	.8087921	.1912079	.4493290	1.8000000E+00	2.8799602E+00
9000.000	.7724824	.2275176	.4065697	1.9000000E+00	2.6082827E+00
10000.000	.7357589	.2642411	.3678794	2.0000000E+00	2.3922112E+00

FIGURE 6-7 (Cont.-9)

T	REL PRODUCT
0.000	1.0000000
1000.000	.9920889
2000.000	.9614800
3000.000	.9036393
4000.000	.8218135
5000.000	.7238669
6000.000	.6189930
7000.000	.5153894
8000.000	.4191037
9000.000	.3338004
10000.000	.2610723

SECTION 7

SUMMARY AND CONCLUSIONS

The mathematical models and computer program described in this report are viewed by the authors as but an early step in the development of adequate, fault tolerant computer, design evaluation aids.

The tool provided is by no means a panacea. For example, the requirement for user determination of individual fault detection, isolation and recovery process characteristics, is viewed as a rather significant drawback. There are others as well.

Nevertheless, we also view it as a major improvement over other available reliability evaluation aids. For example, the provision for two distinct operating modes contrasts favorably with the single mode approximations typically encountered. Similiarly, the inclusion of a means for calculating multiple coverage factors, conditioned on failure type and location, operating mode, and spares status, goes far beyond the more usual mathematical reliability model.

In addition, the generality of the model enables its use on a wide variety of system configurations, and variations therein. These include TMR, TMR with spares switching, hybrid, and a majority of others as well.

There remains much to be done, however, prior to the transformation of fault tolerant computer design from that of a rather mystical art, to that of a science. Most noteworthy, perhaps, is the task of gathering adequate failure rate statistics within integrated circuit chips, and also the task of establishing and implementing measurement criteria for individual fault detection, isolation and recovery processes. Hopefully, these will prove to be forthcoming from other endeavors.

Page Intentionally Left Blank

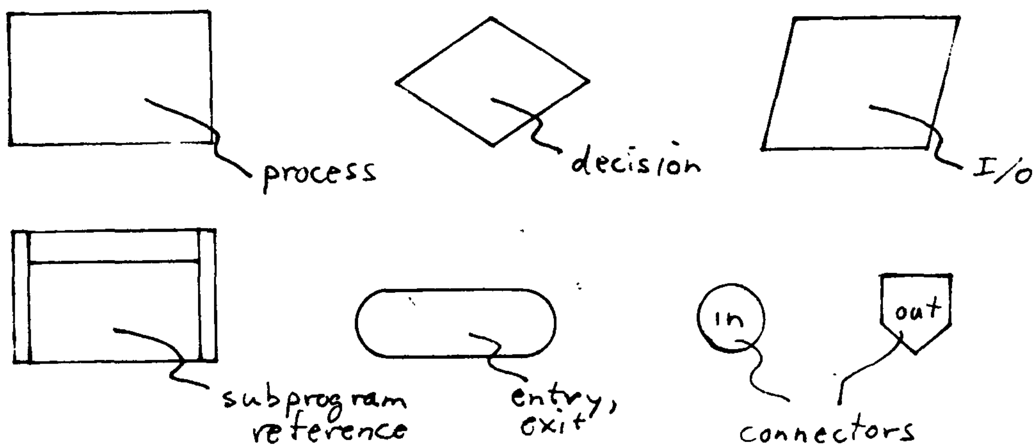
APPENDIX A

FLOW DIAGRAMS

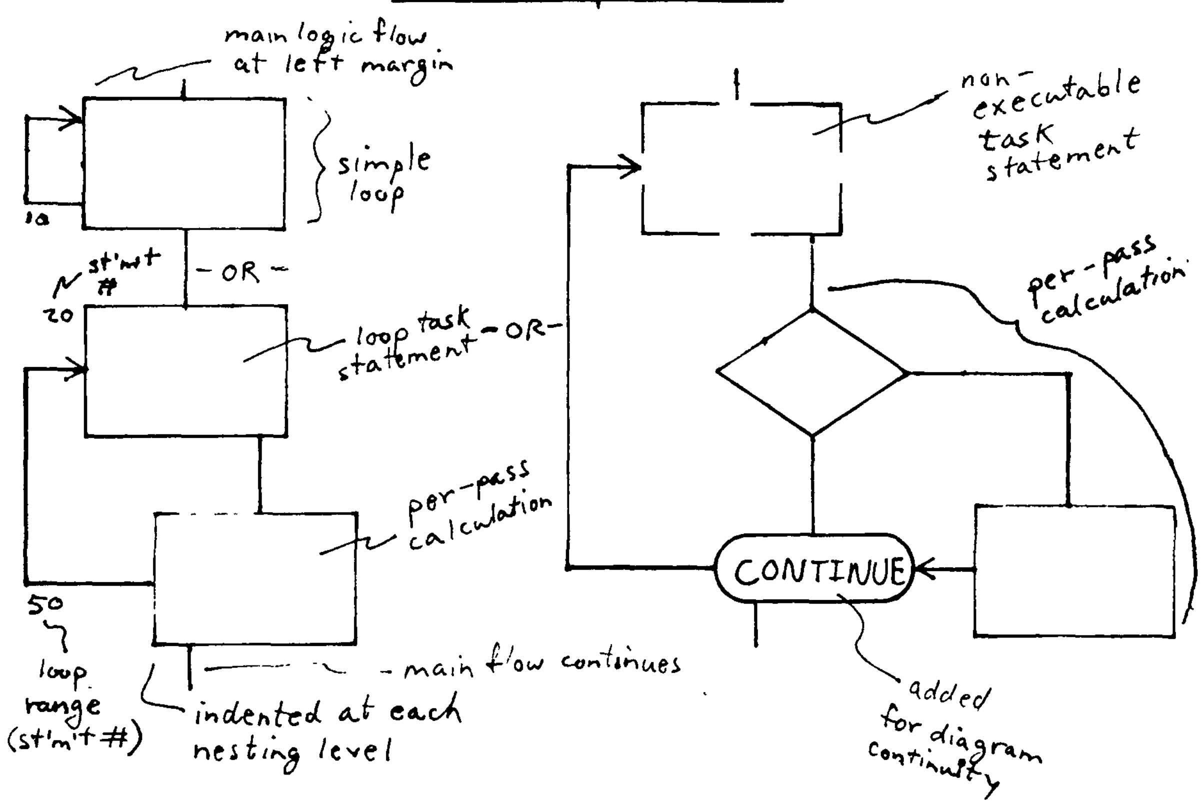
FLOW DIAGRAM

A-0

FLOW DIAGRAM CONVENTIONS

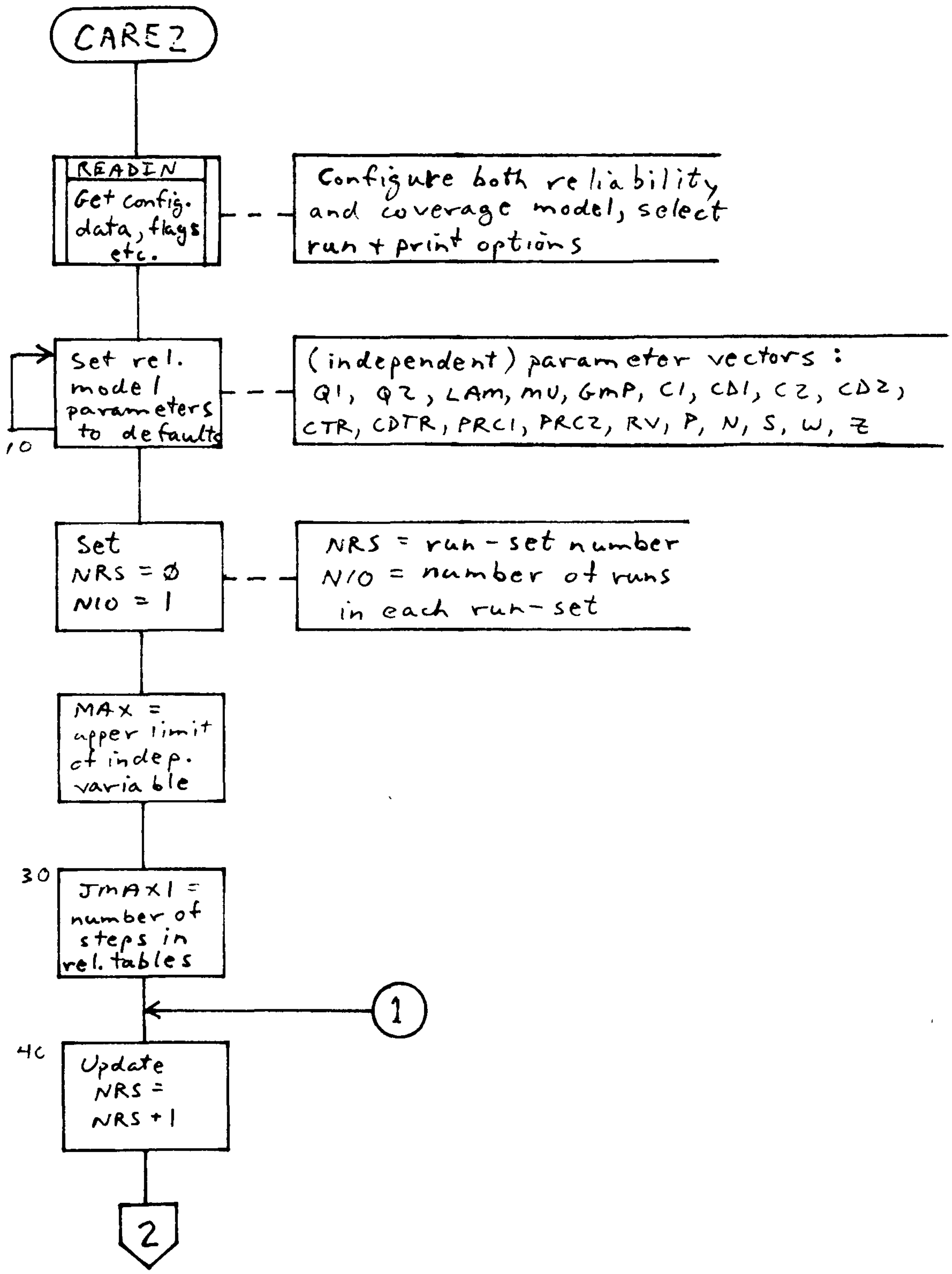


iterative procedures



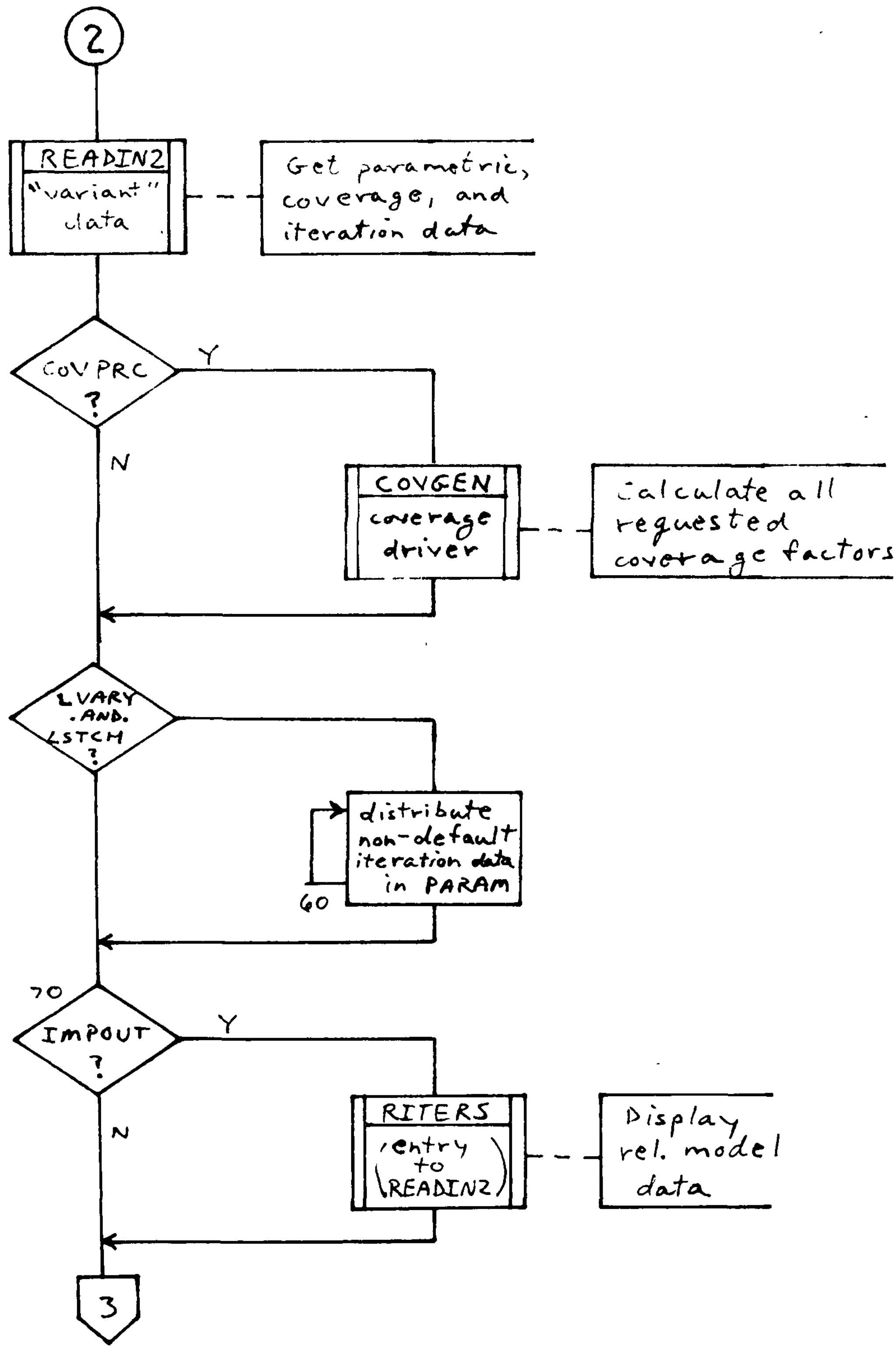
FLOW DIAGRAM

A-1



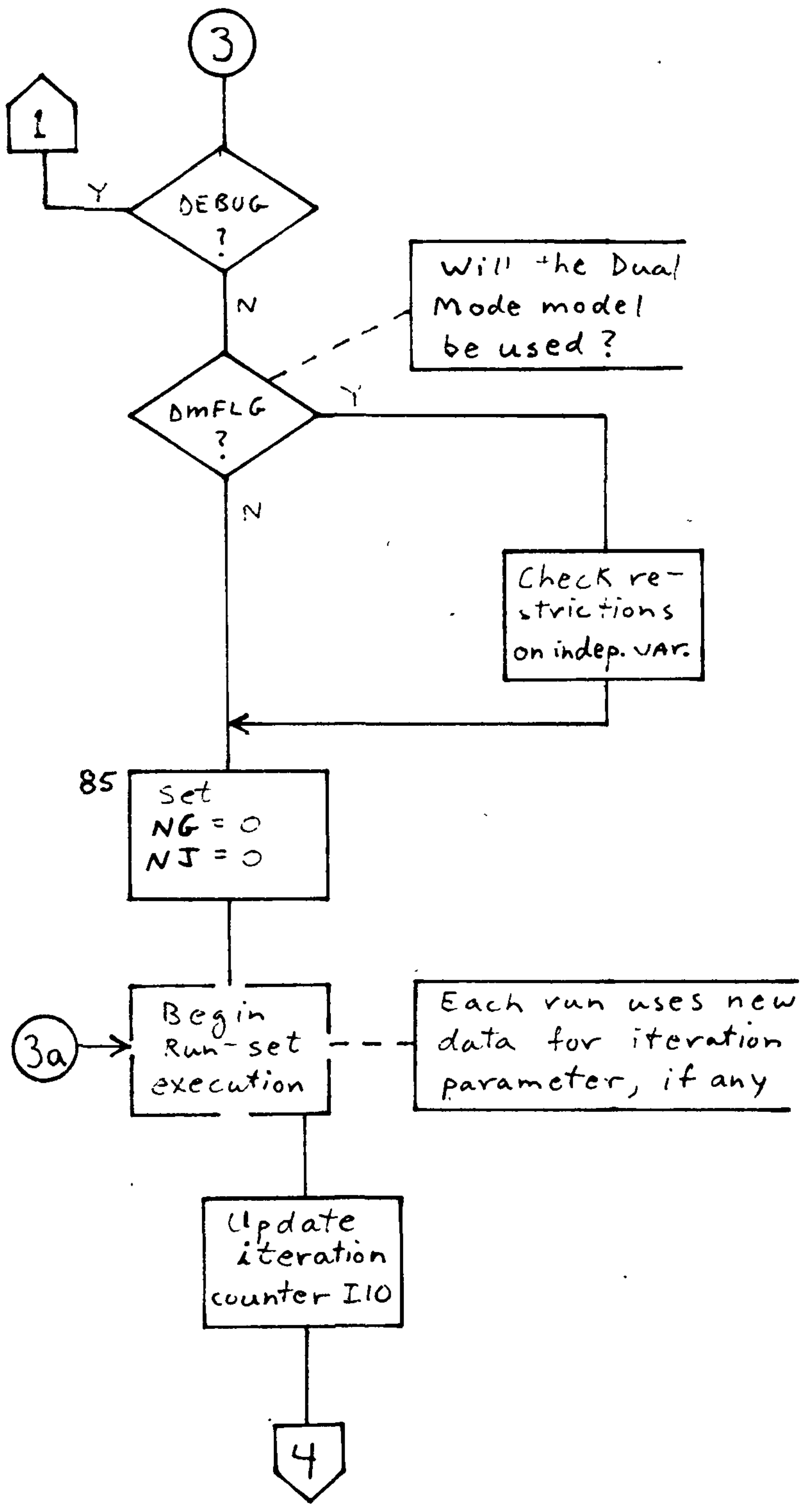
FLOW DIAGRAM

A-1



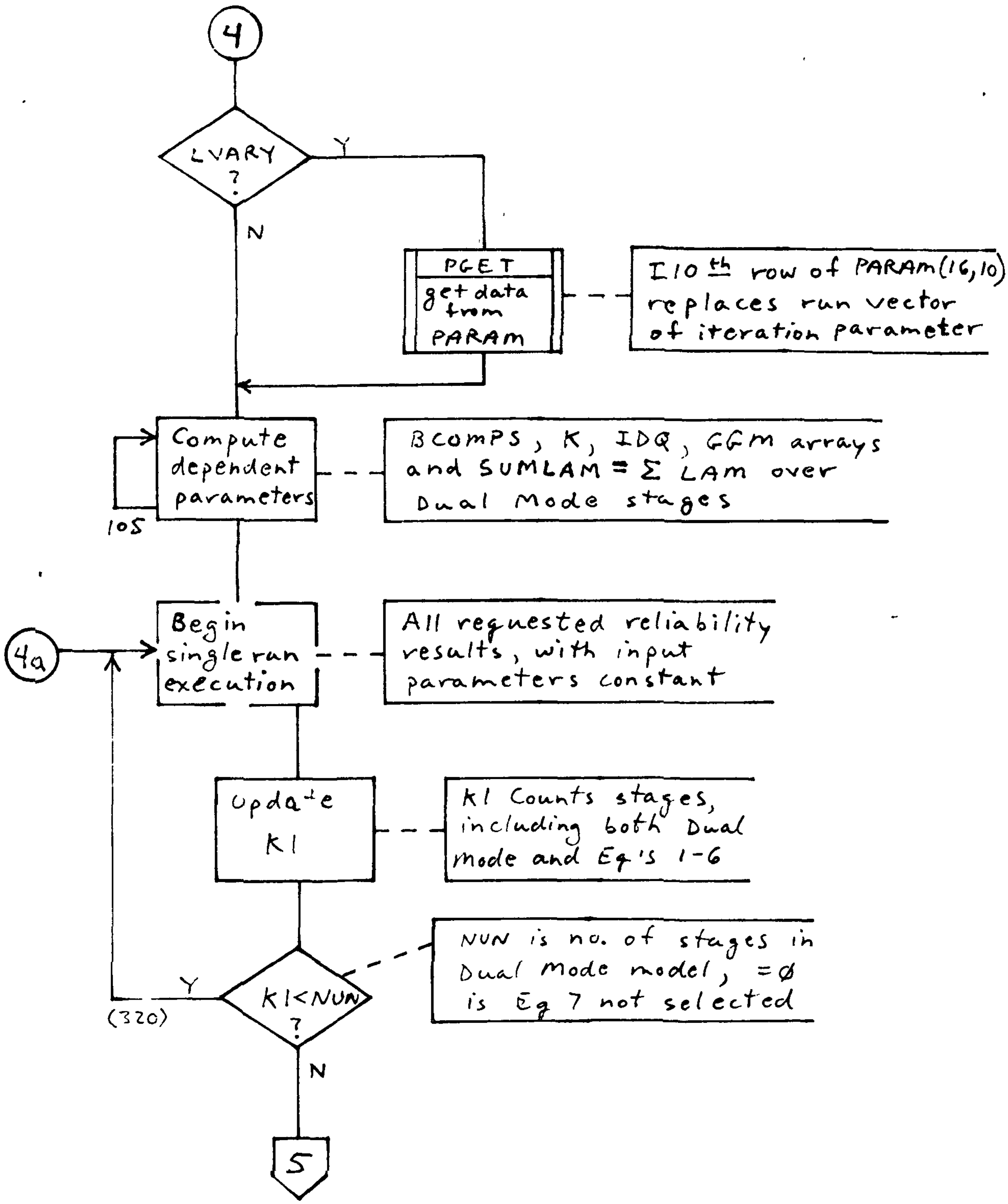
FLOW DIAGRAM

A-1



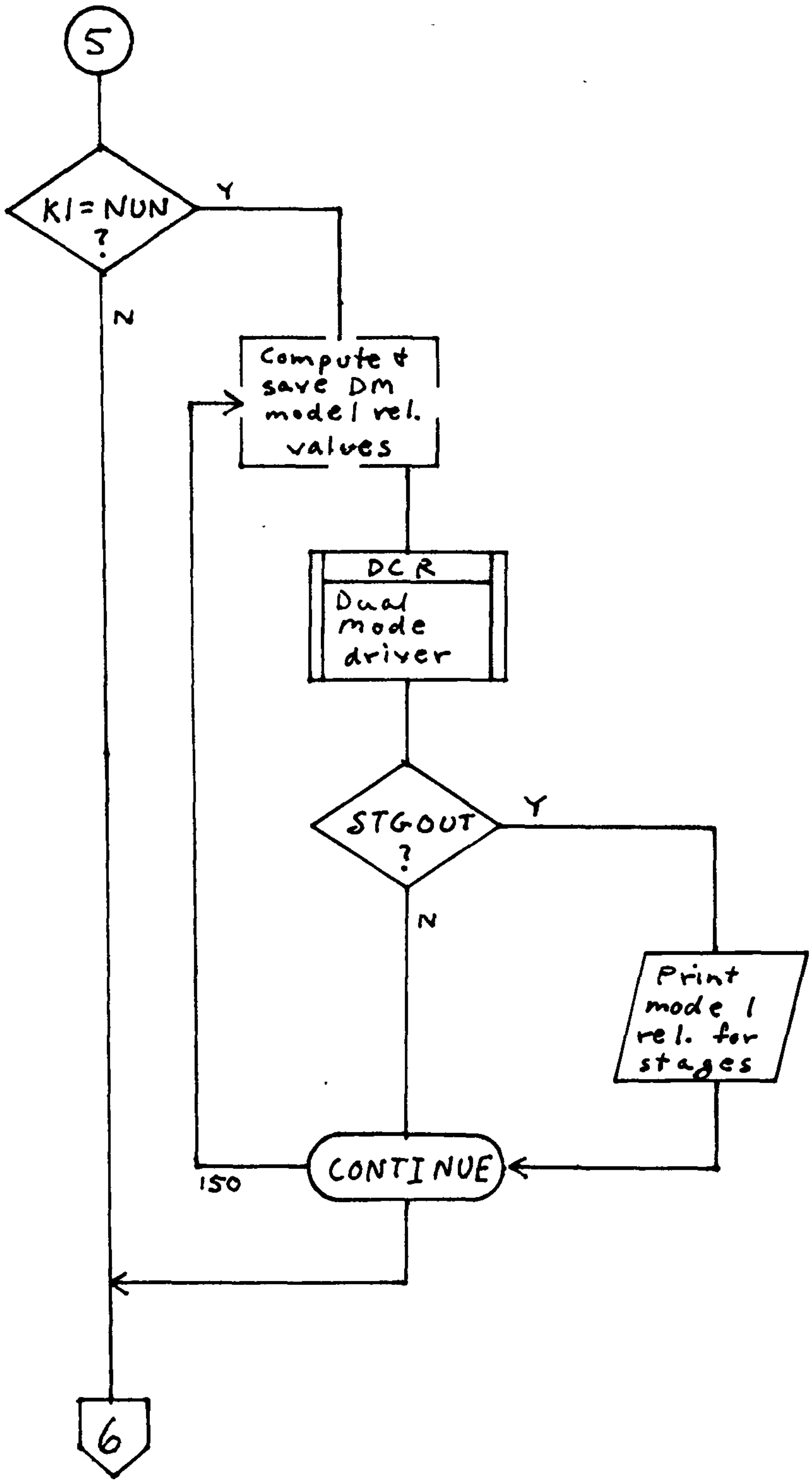
FLOW DIAGRAM

A-1



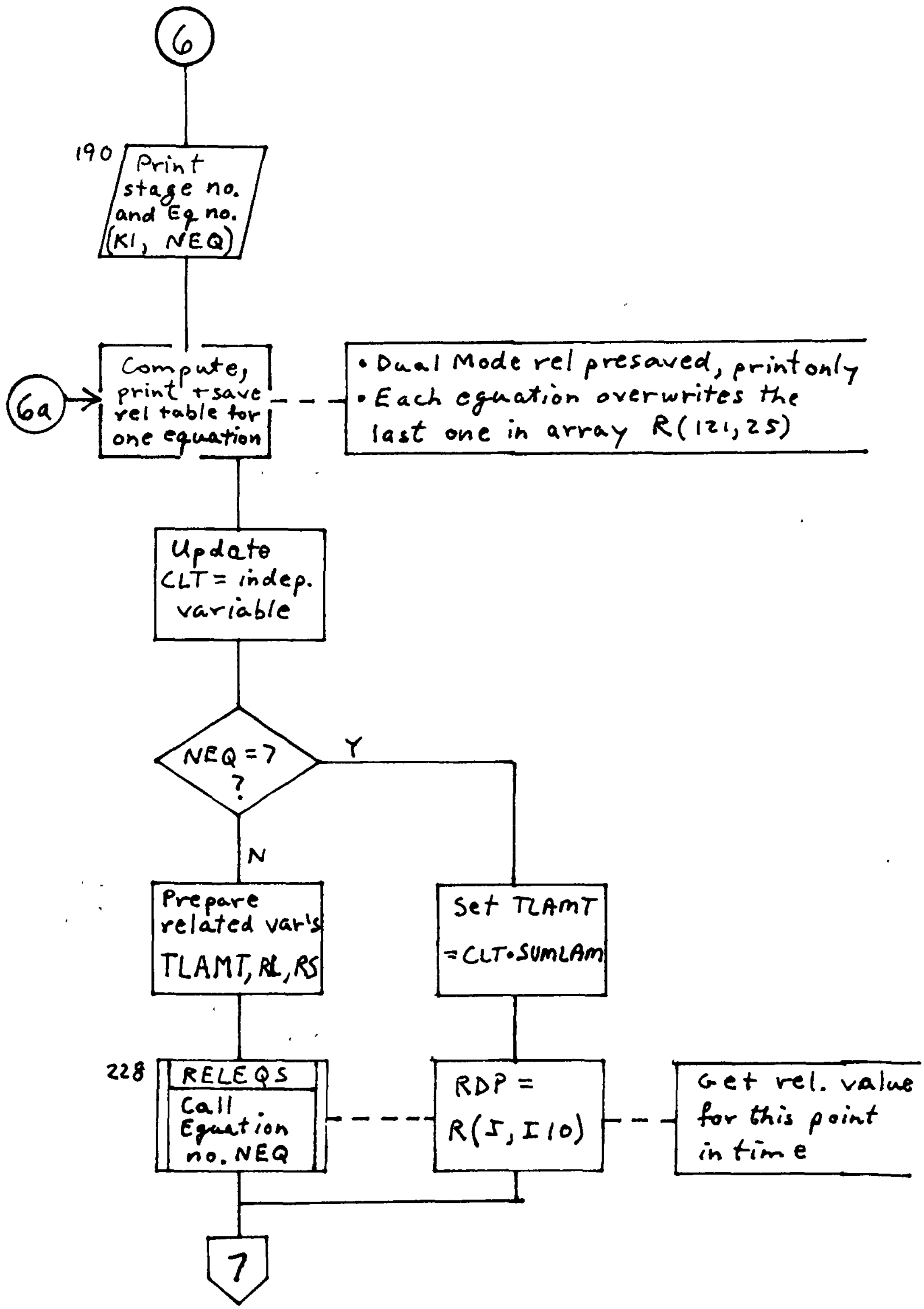
FLOW DIAGRAM

A-1



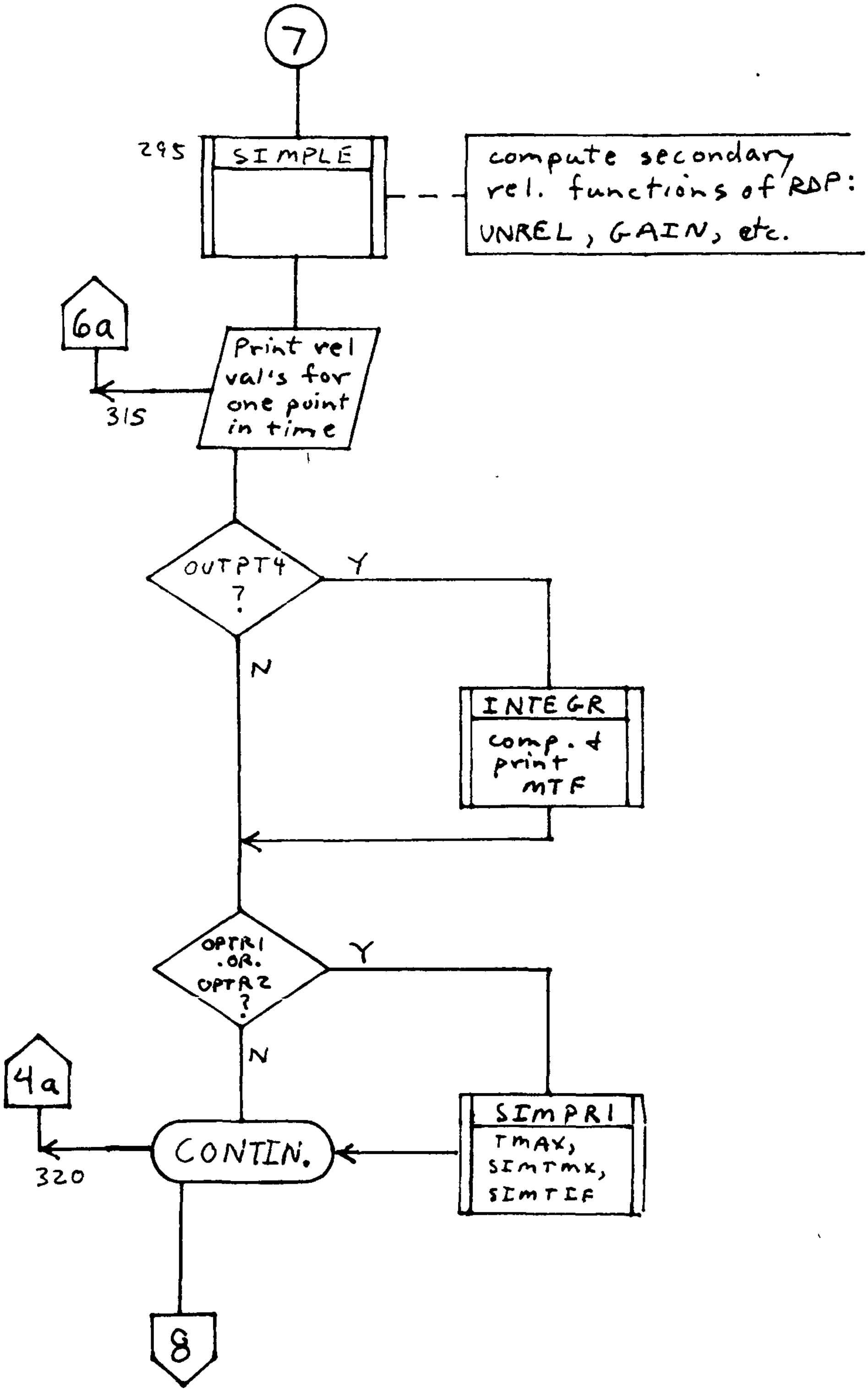
FLOW DIAGRAM

A-1



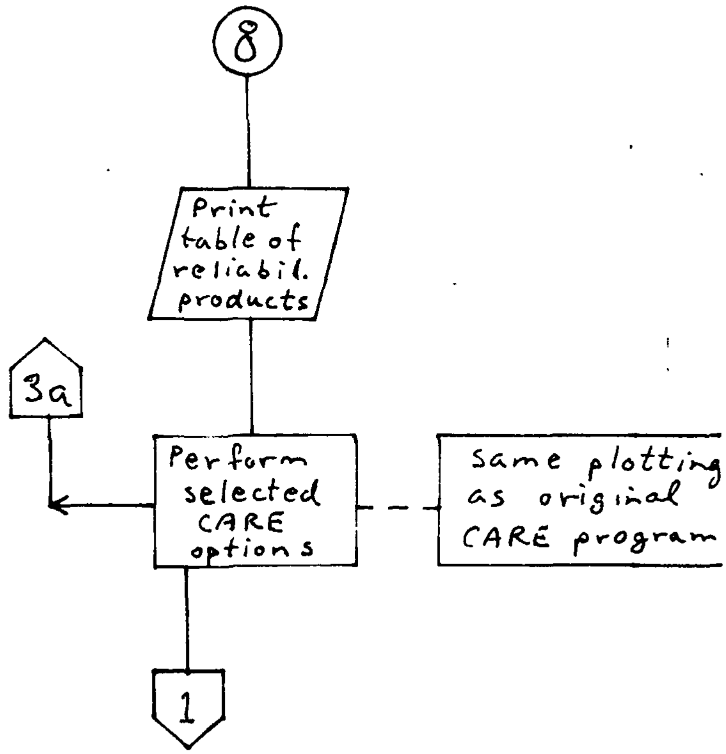
FLOW DIAGRAM

A-1



FLOW DIAGRAM

A-1



FLOW DIAGRAM

A-2

NEQ1A

The orginal subroutine was corrected as follows:

original lines

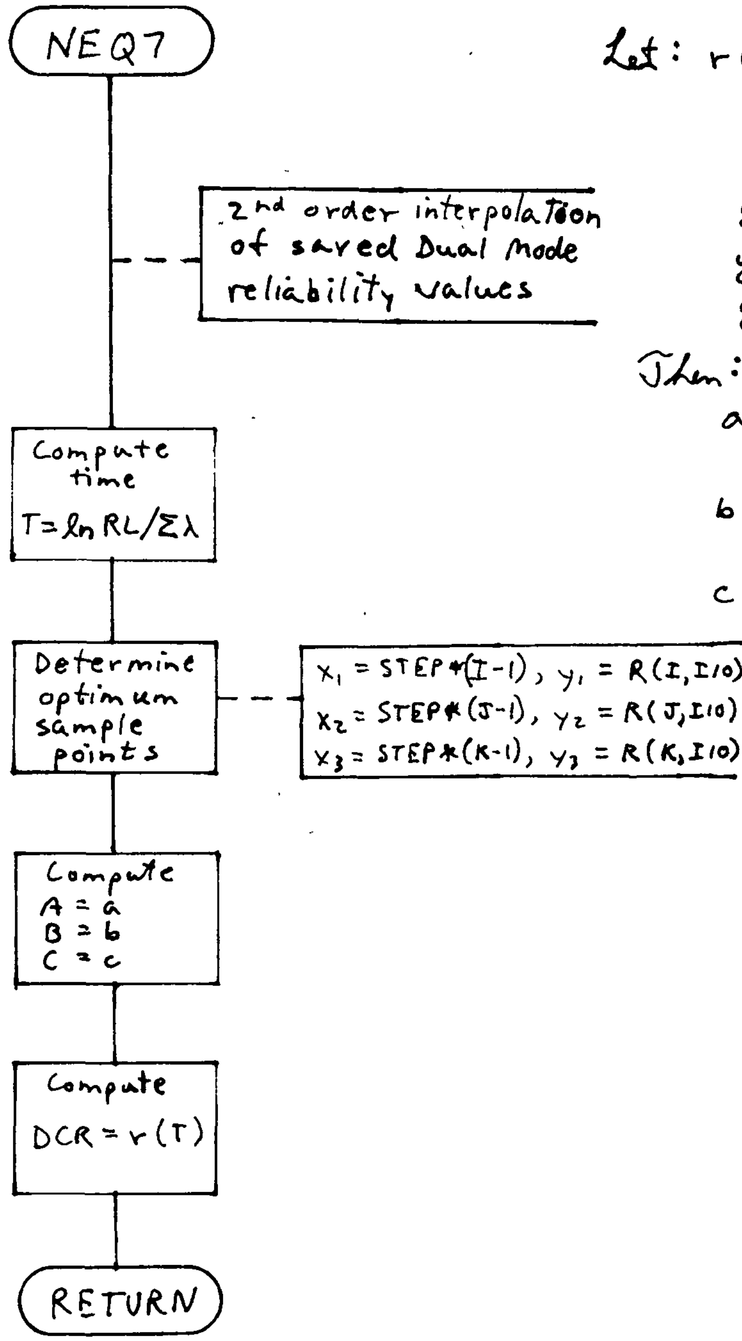
```
   DO 70 LX=1,II
   L = LX-1
   SUM1 = 0.0
   SUM2 = 0.0
     |
     |
     |
60 SUM2 = ~
70 SUM1 = ~
```

modified lines

```
   SUM1 = 0.0
   DO 70 LX=1,II
   L=LX-1
   SUM2=0.0
     |
     |
     |
60 SUM2 = ~
70 SUM1 = ~
```

FLOW DIAGRAM

A-3



Let: $r(x) \equiv ax^2 + bx + c$,

$x_2 = x_1 + \Delta x$,

$x_3 = x_1 + 2\Delta x$,

$y_1 = r(x_1)$,

$y_2 = r(x_2)$, and

$y_3 = r(x_3)$

Then:

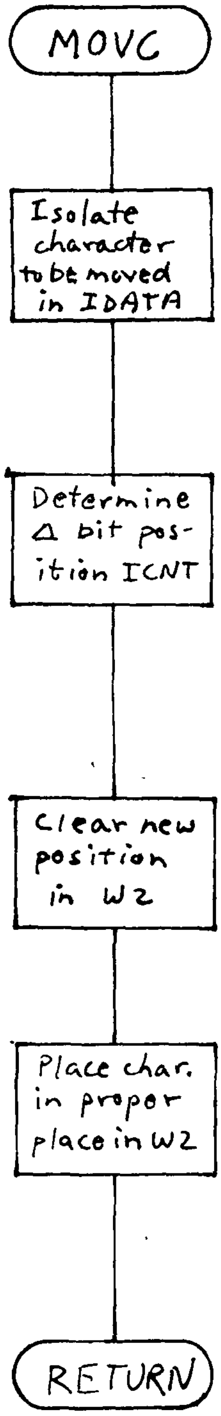
$$a = \frac{(y_1 - 2y_2 + y_3)}{2(\Delta x)^2}$$

$$b = \frac{y_3 - y_1}{2\Delta x} - 2a x_2$$

$$c = y_2 - a x_2^2 - b x_2$$

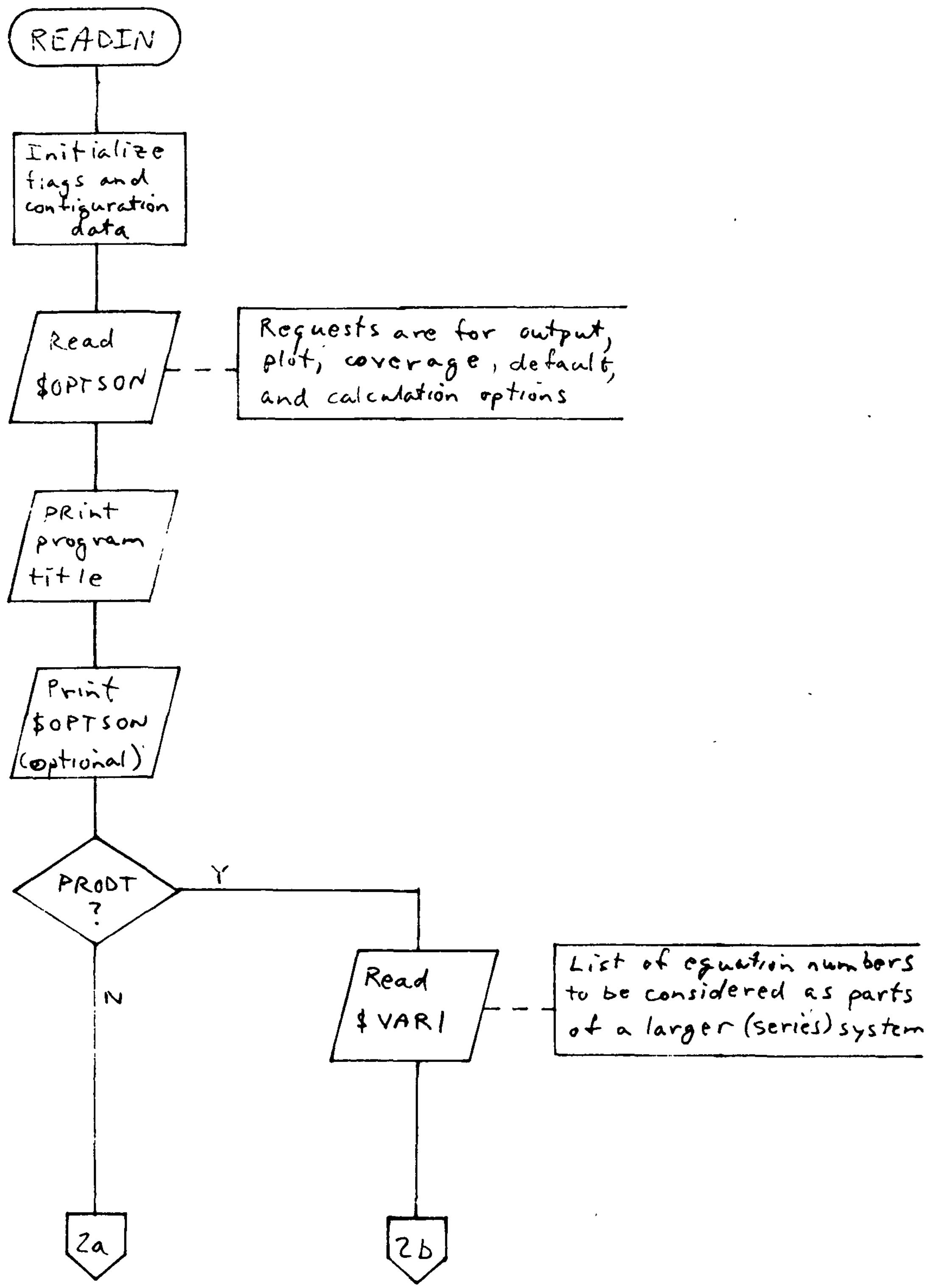
FLOW DIAGRAM

A-4



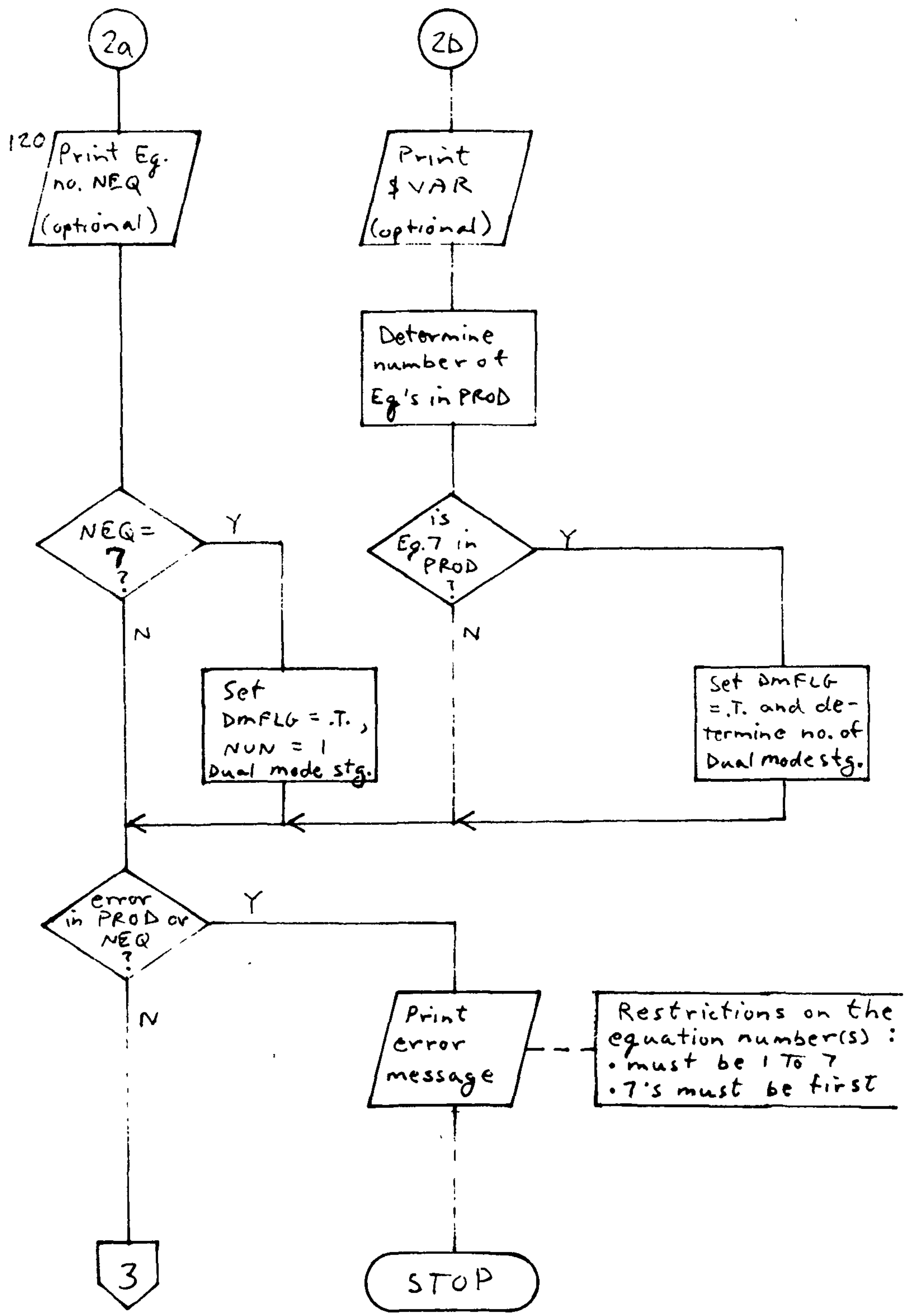
FLOW DIAGRAM

A-5



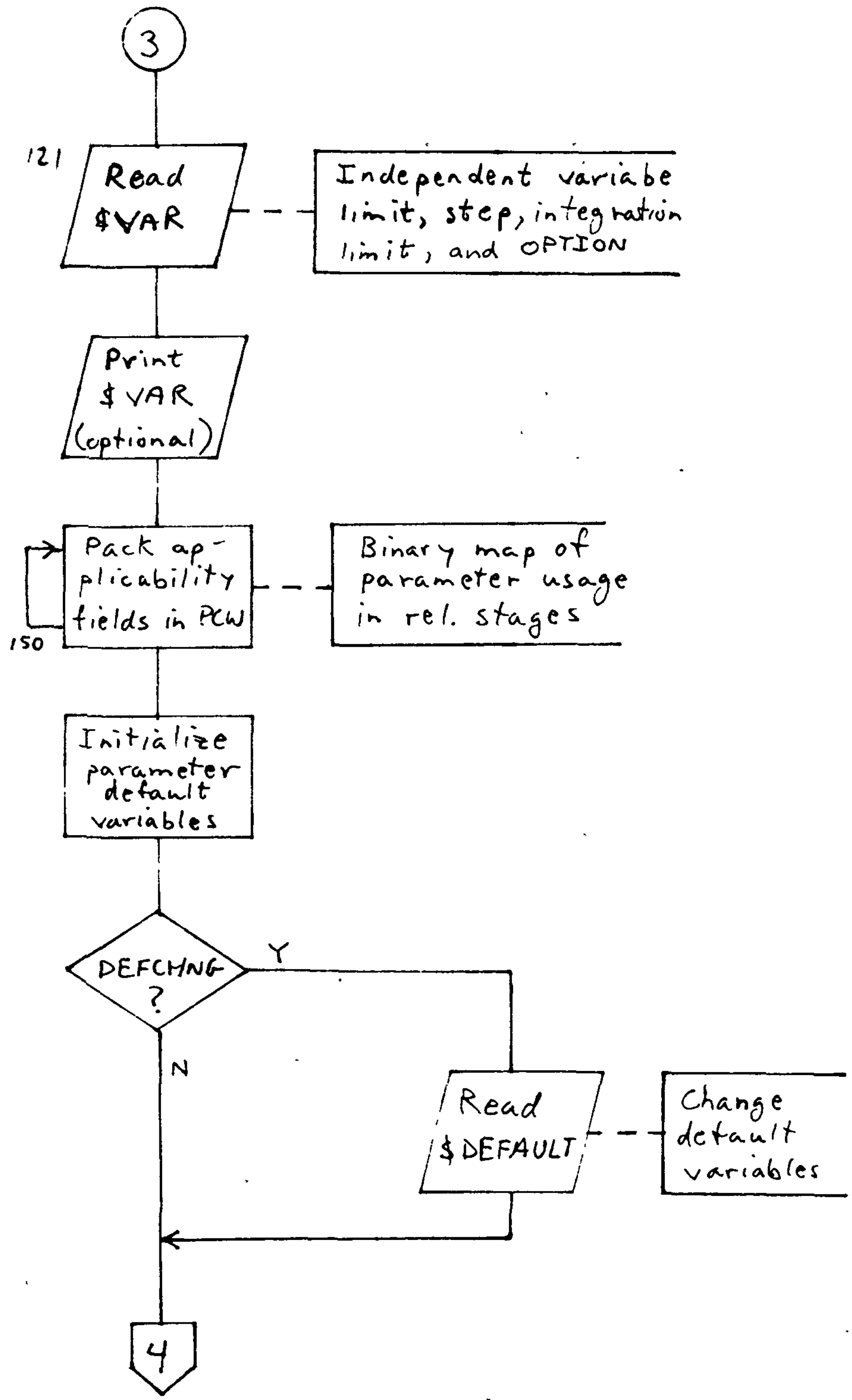
FLOW DIAGRAM

A-5



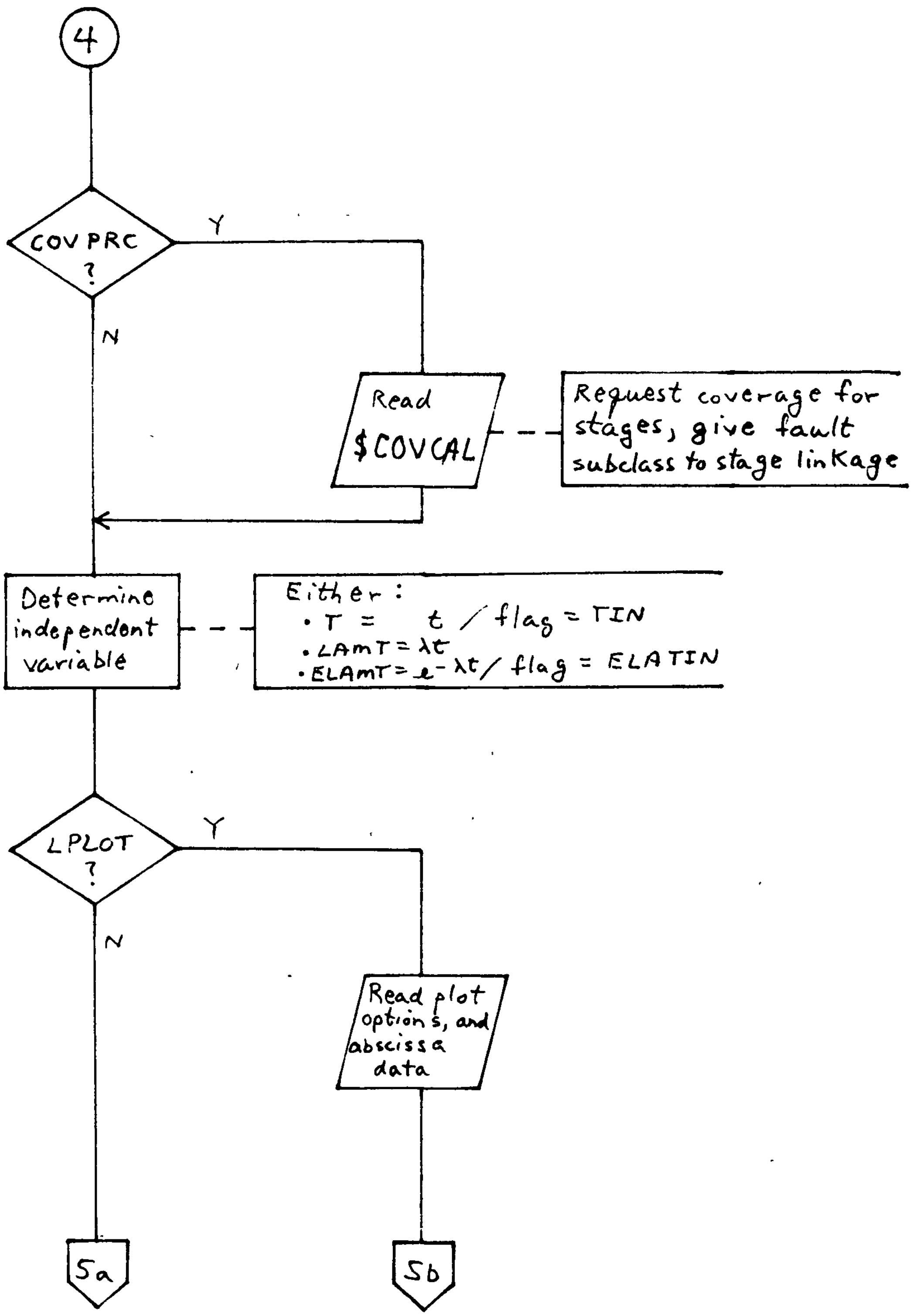
FLOW DIAGRAM

A-5



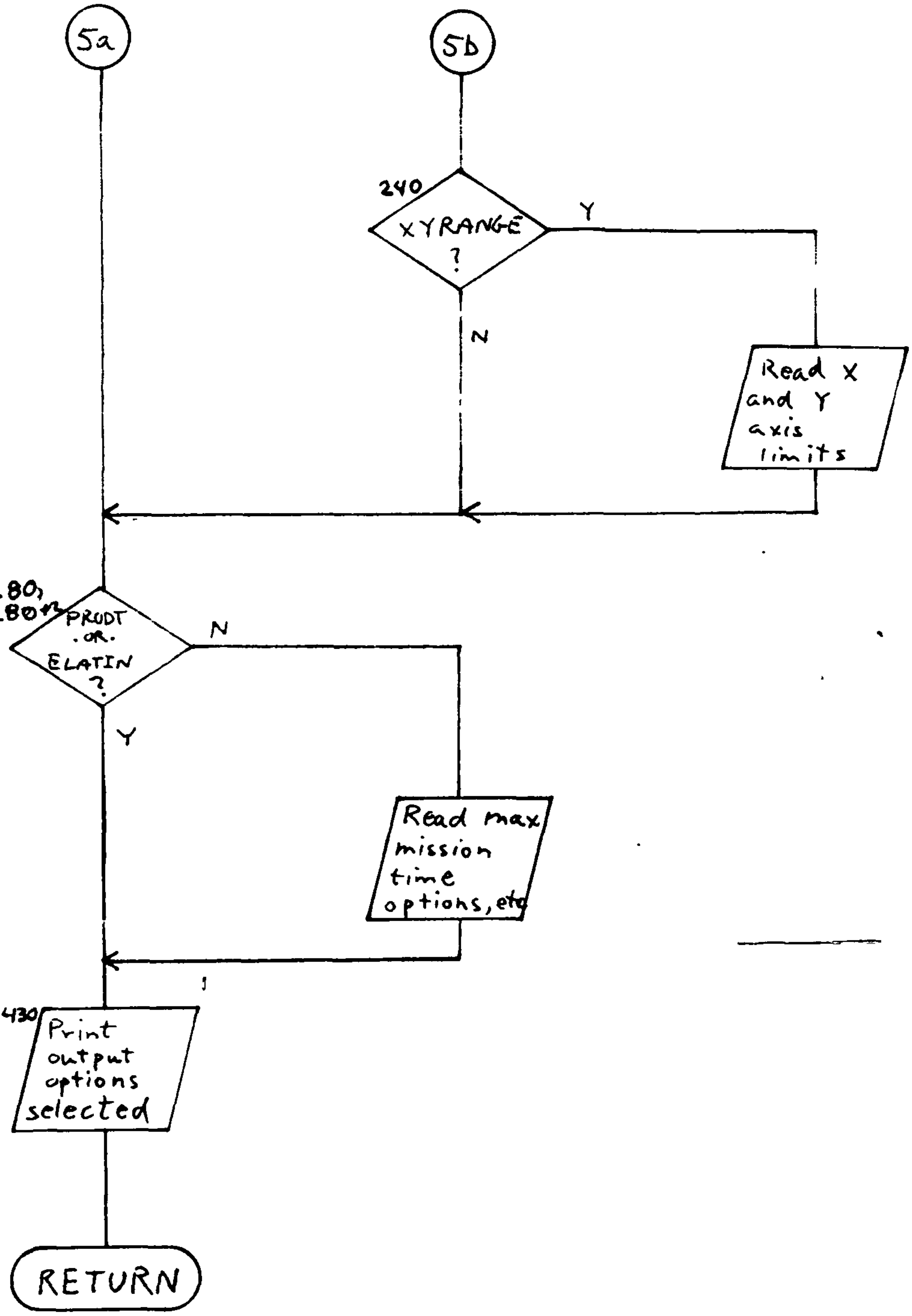
FLOW DIAGRAM

A-5



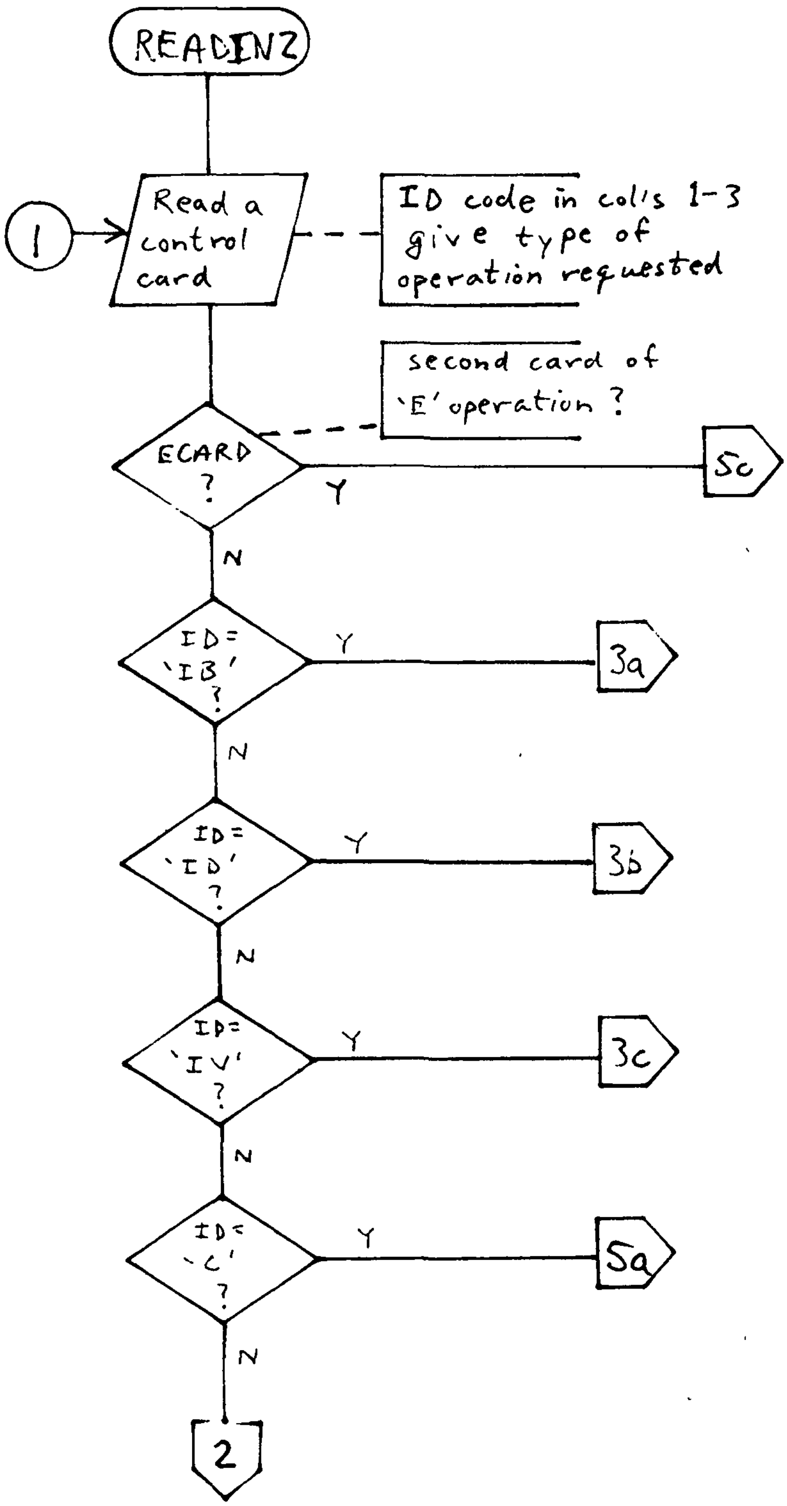
FLOW DIAGRAM

A-5



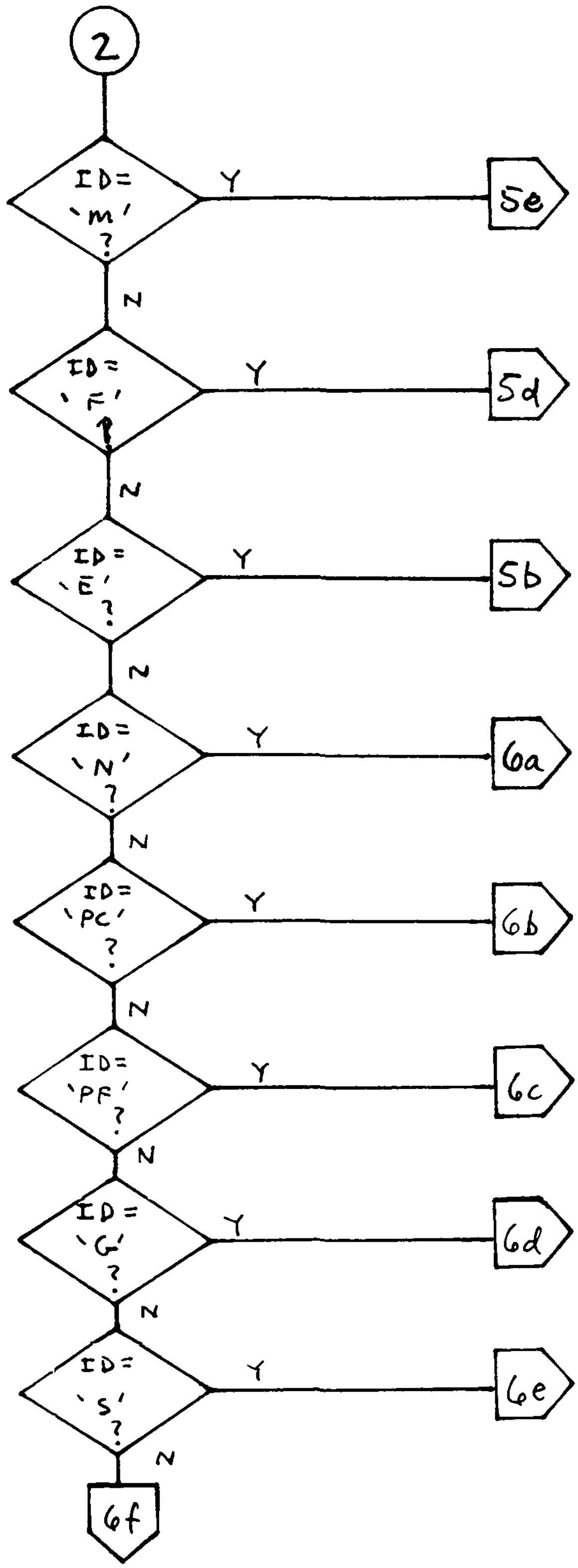
FLOW DIAGRAM

A-6

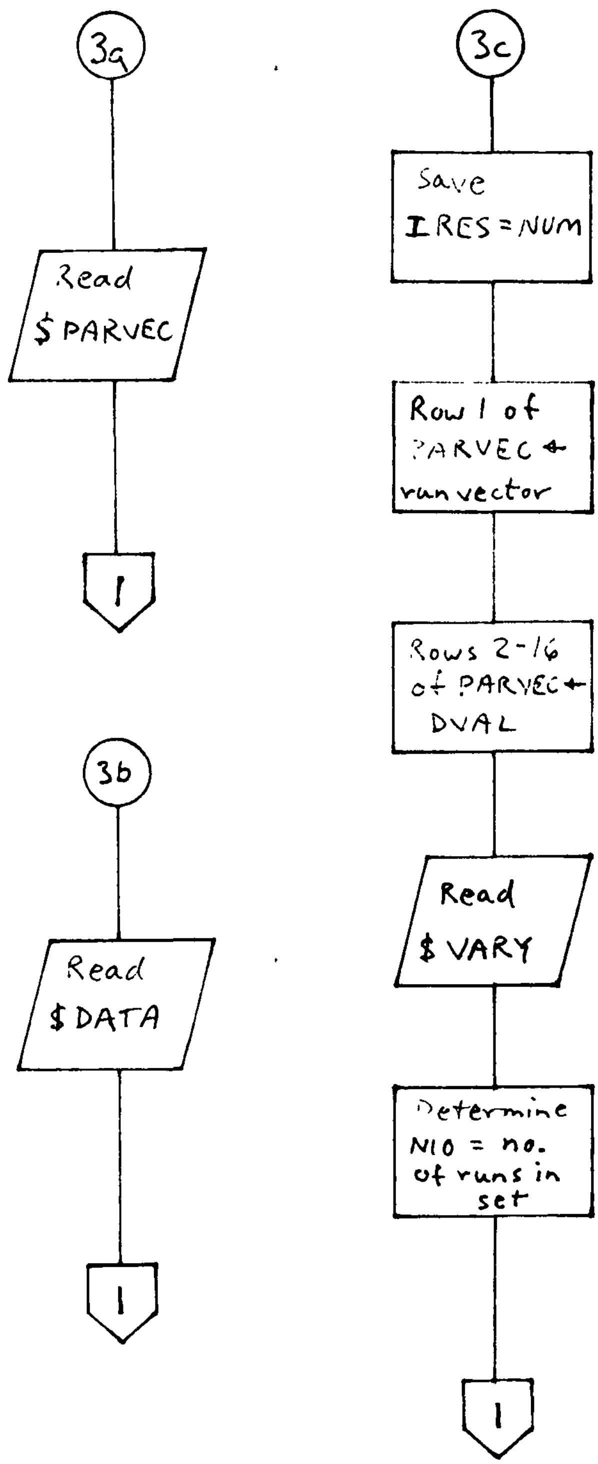


FLOW DIAGRAM

A-6

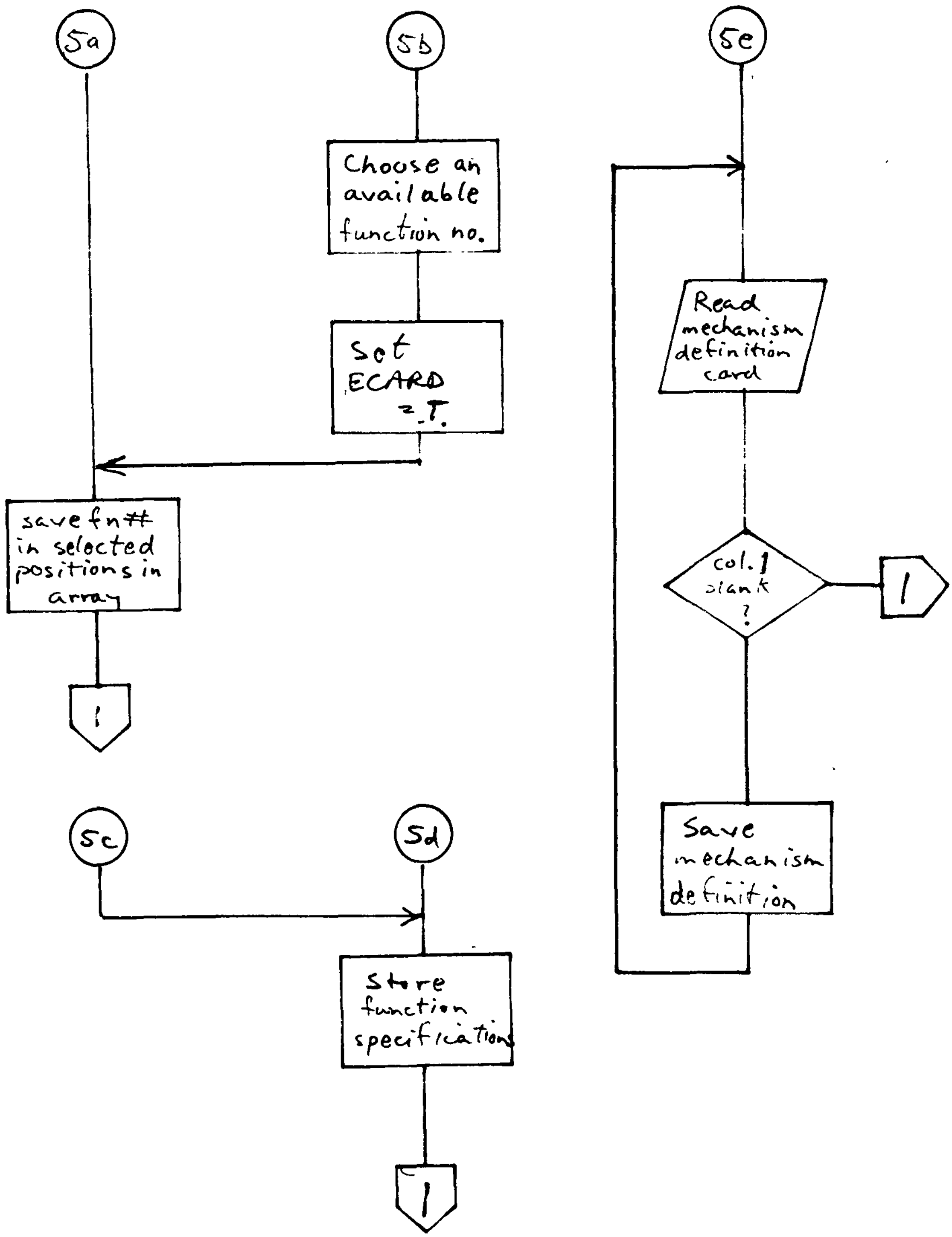


FLOW DIAGRAM
A-6



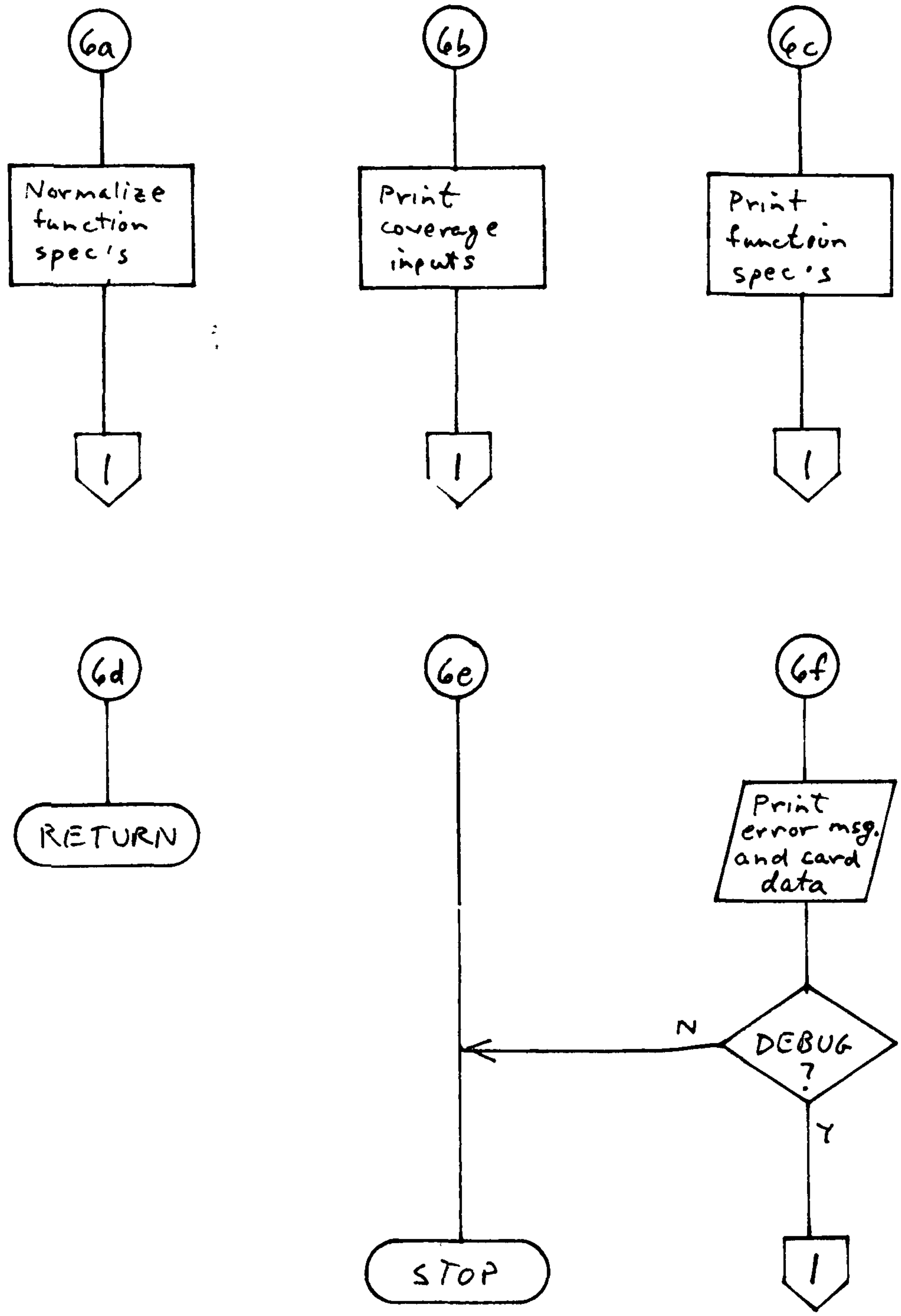
FLOW DIAGRAM

A-6



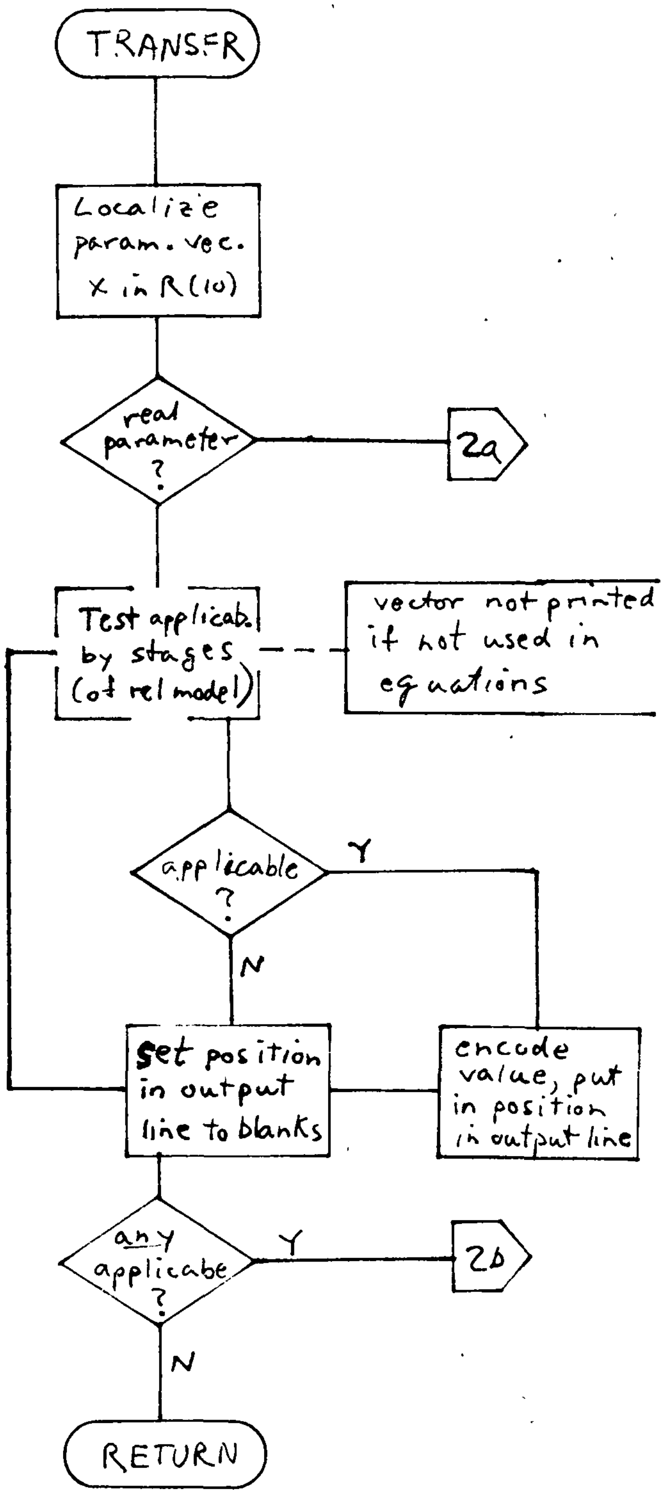
FLOW DIAGRAM

A-6



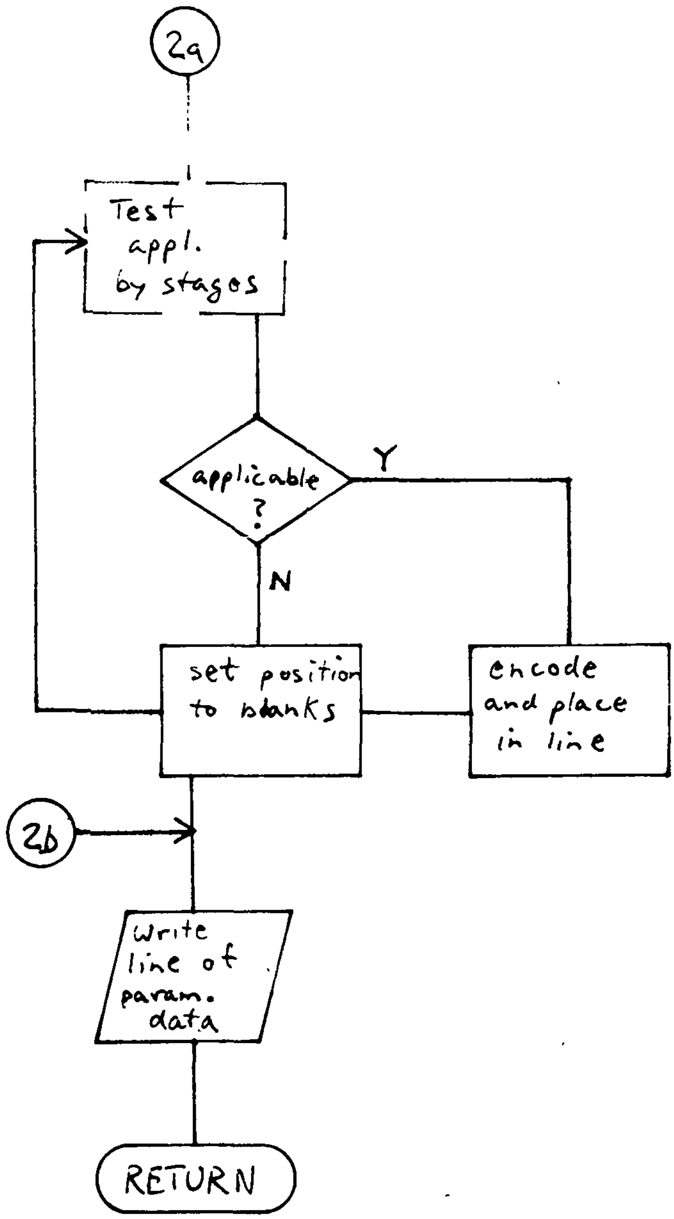
FLOW DIAGRAM

A-7



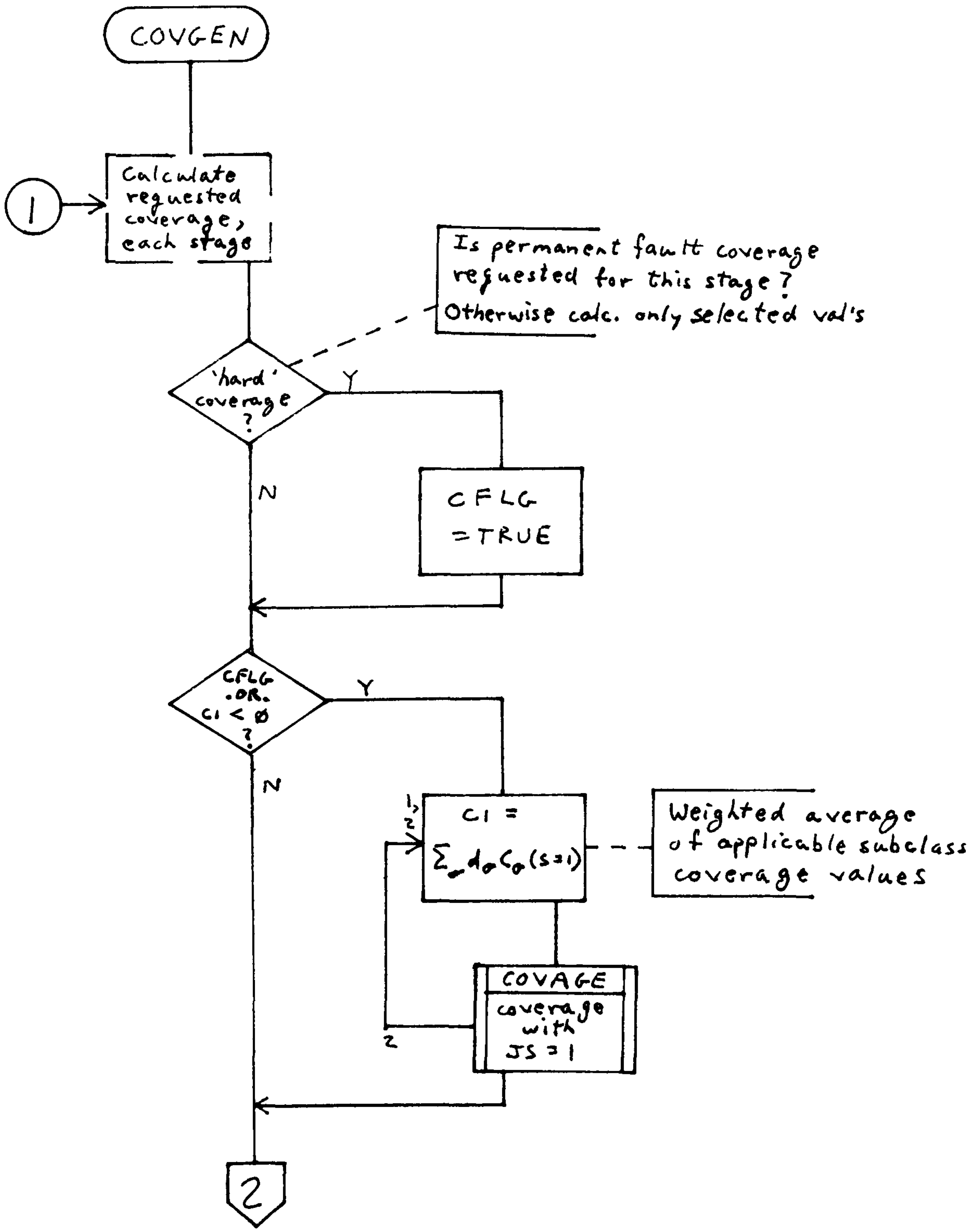
FLOW DIAGRAM

A-7



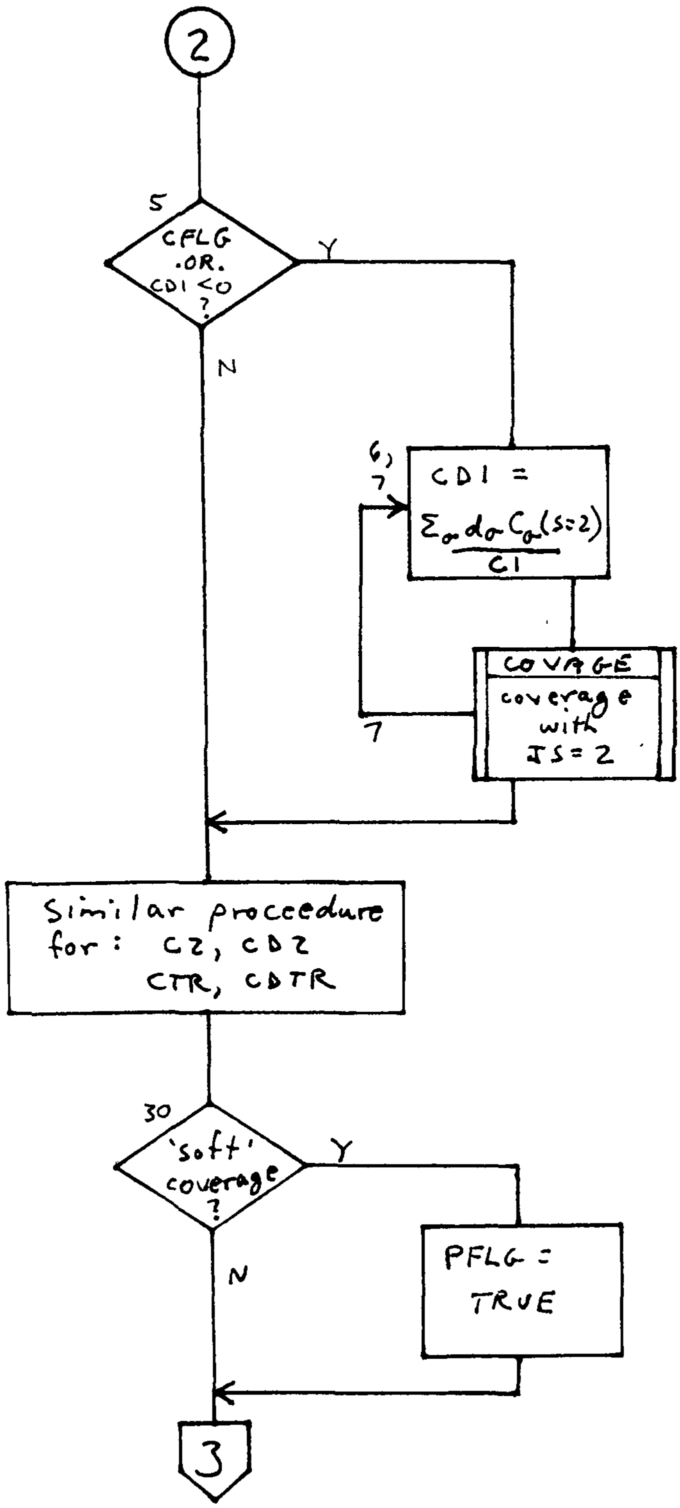
FLOW DIAGRAM

A-8



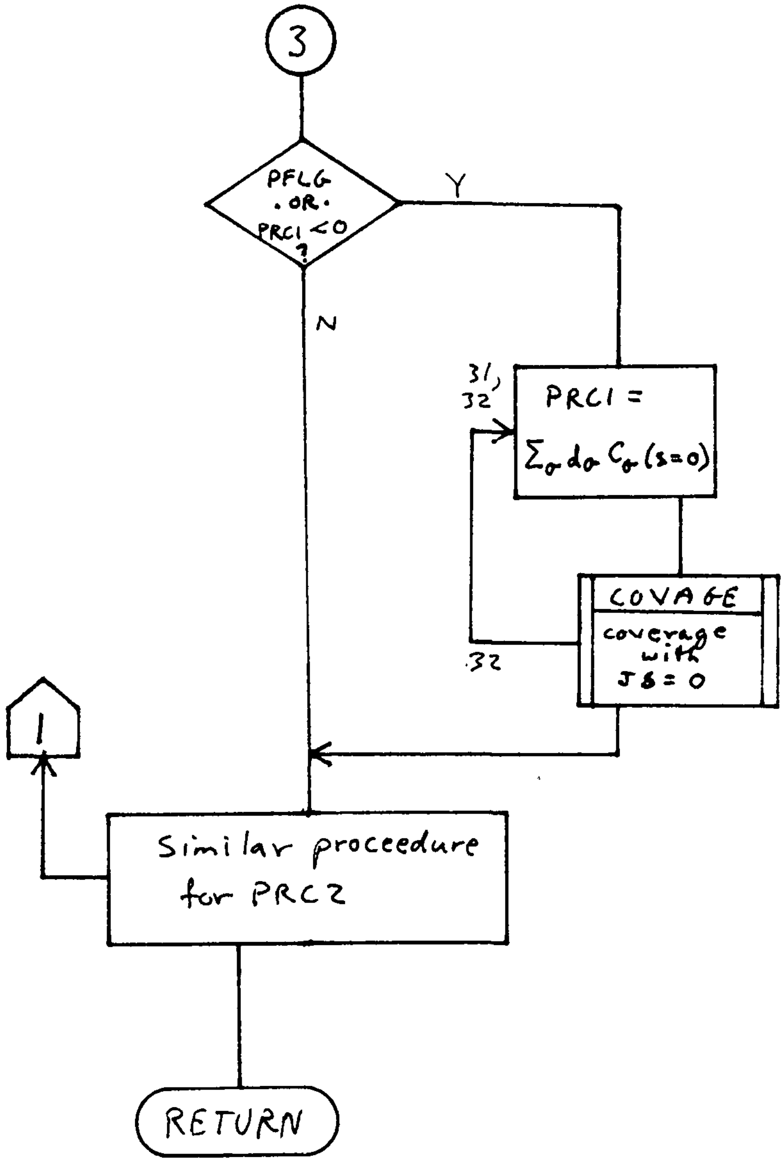
FLOW DIAGRAM

A-8



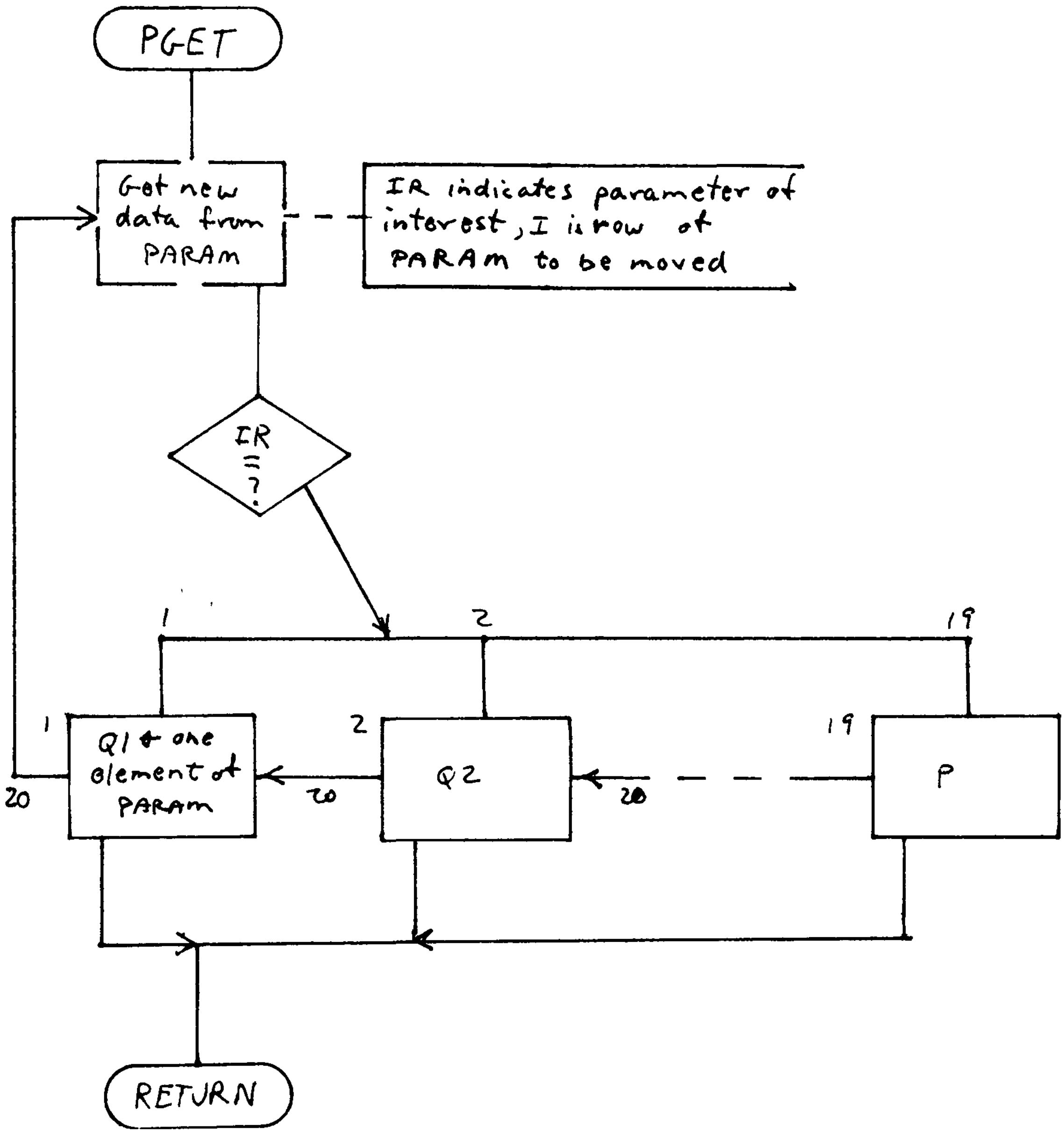
FLOW DIAGRAM

A-8



FLOW DIAGRAM

A-9



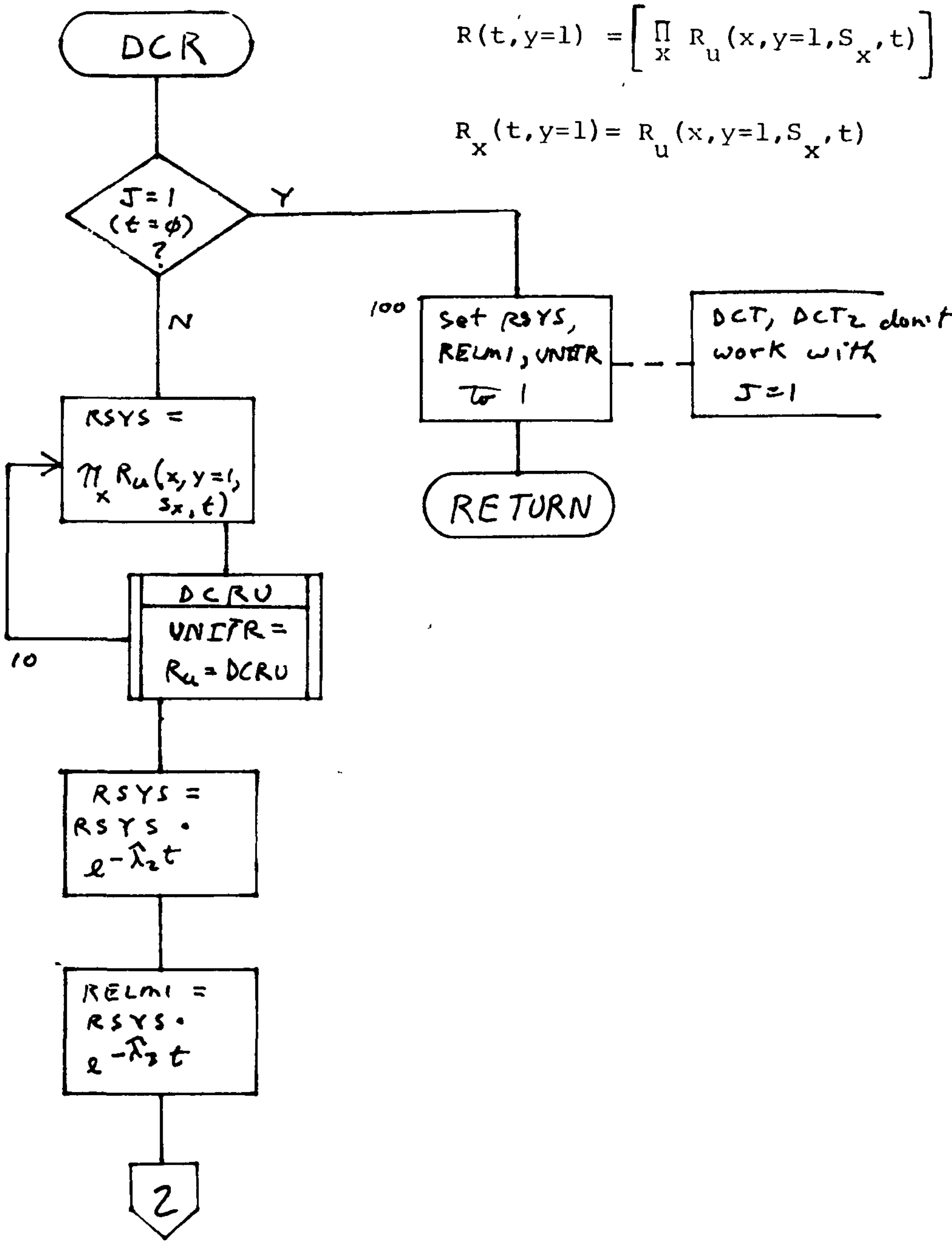
FLOW DIAGRAM

A-10

$$R(t) = \left[\prod_x \left\{ R_u(x, y=1, S_x, t) \right\} * e^{-(\hat{\lambda}_2)t} + r_2(t) + \sum_x \left\{ r(x,t) \right\} \right] e^{-(\hat{\lambda}_3)t}$$

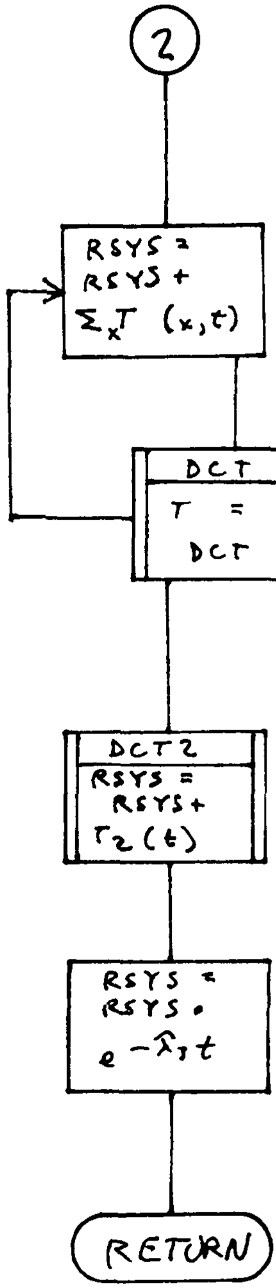
$$R(t, y=1) = \left[\prod_x R_u(x, y=1, S_x, t) \right] e^{-(\hat{\lambda}_2 + \hat{\lambda}_3)t}$$

$$R_x(t, y=1) = R_u(x, y=1, S_x, t)$$



FLOW DIAGRAM

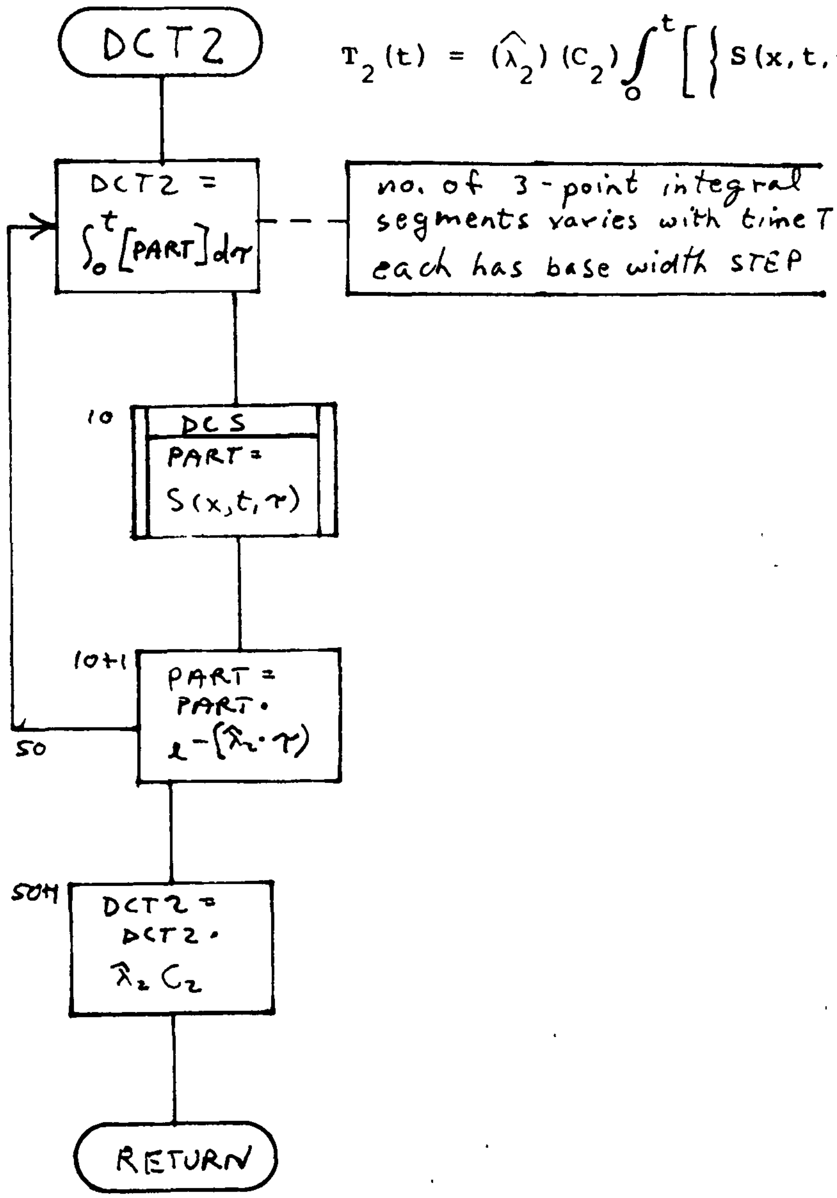
A-10



FLOW DIAGRAM

A-11

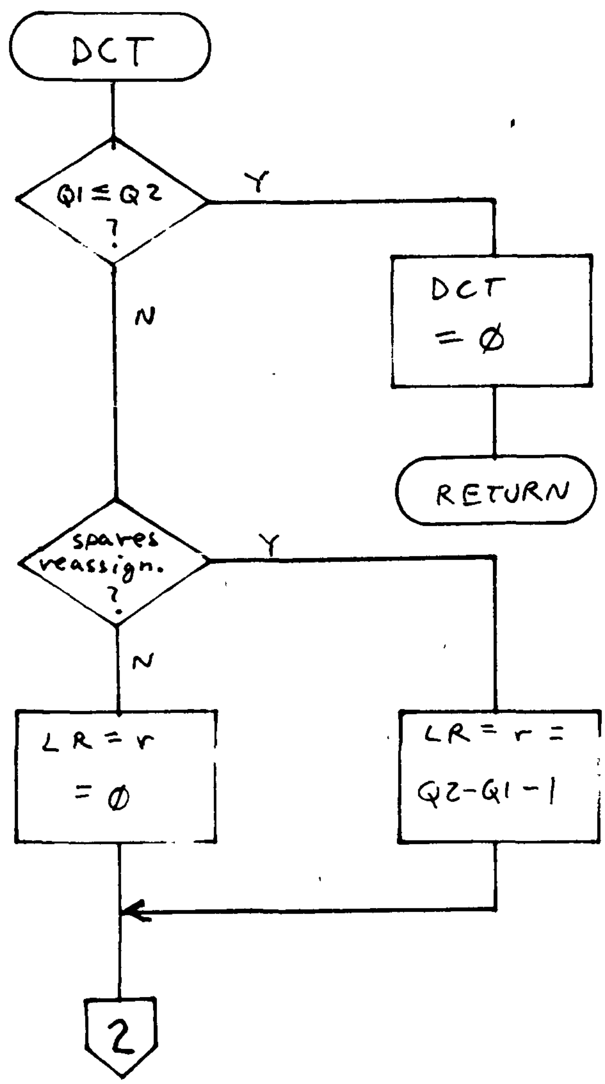
$$T_2(t) = (\hat{\lambda}_2)(C_2)\int_0^t \left[\left\{ S(x,t,\tau) \right\} * e^{-(\hat{\lambda}_2)\tau} \right] d\tau$$



FLOW DIAGRAM

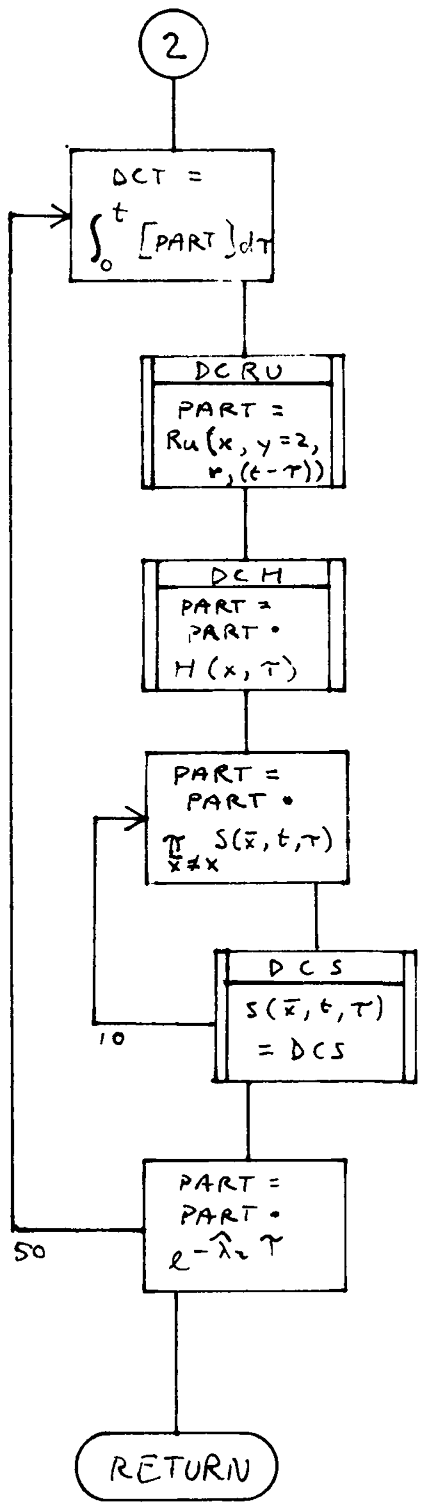
A-12

$$T(x,t) = \begin{cases} \int_0^t \left[\prod_{\bar{x} \neq x} \left\{ S(\bar{x},t,\tau) \right\} * R_u(x, y=2, r, (t-\tau)) * H(x,\tau) * e^{-(\hat{\lambda}_2)\tau} \right] d\tau, \\ \quad \text{for } Q(x,1) > Q(x,2) \\ 0, \\ \quad \text{for } Q(x,1) \leq Q(x,2) \end{cases}$$



FLOW DIAGRAM

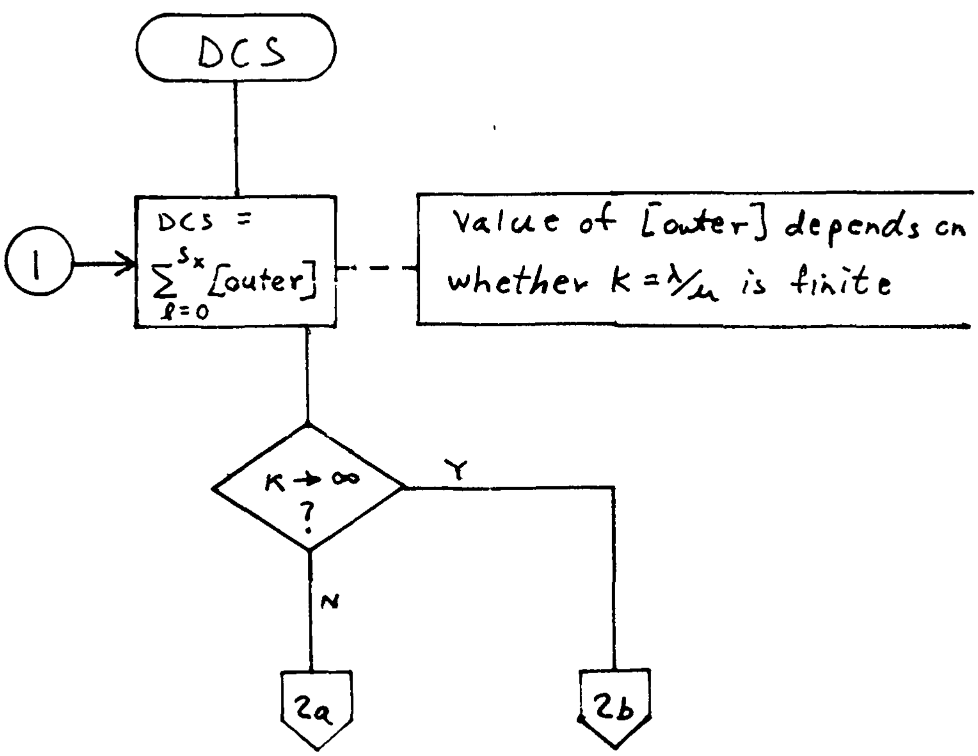
A-12



FLOW DIAGRAM

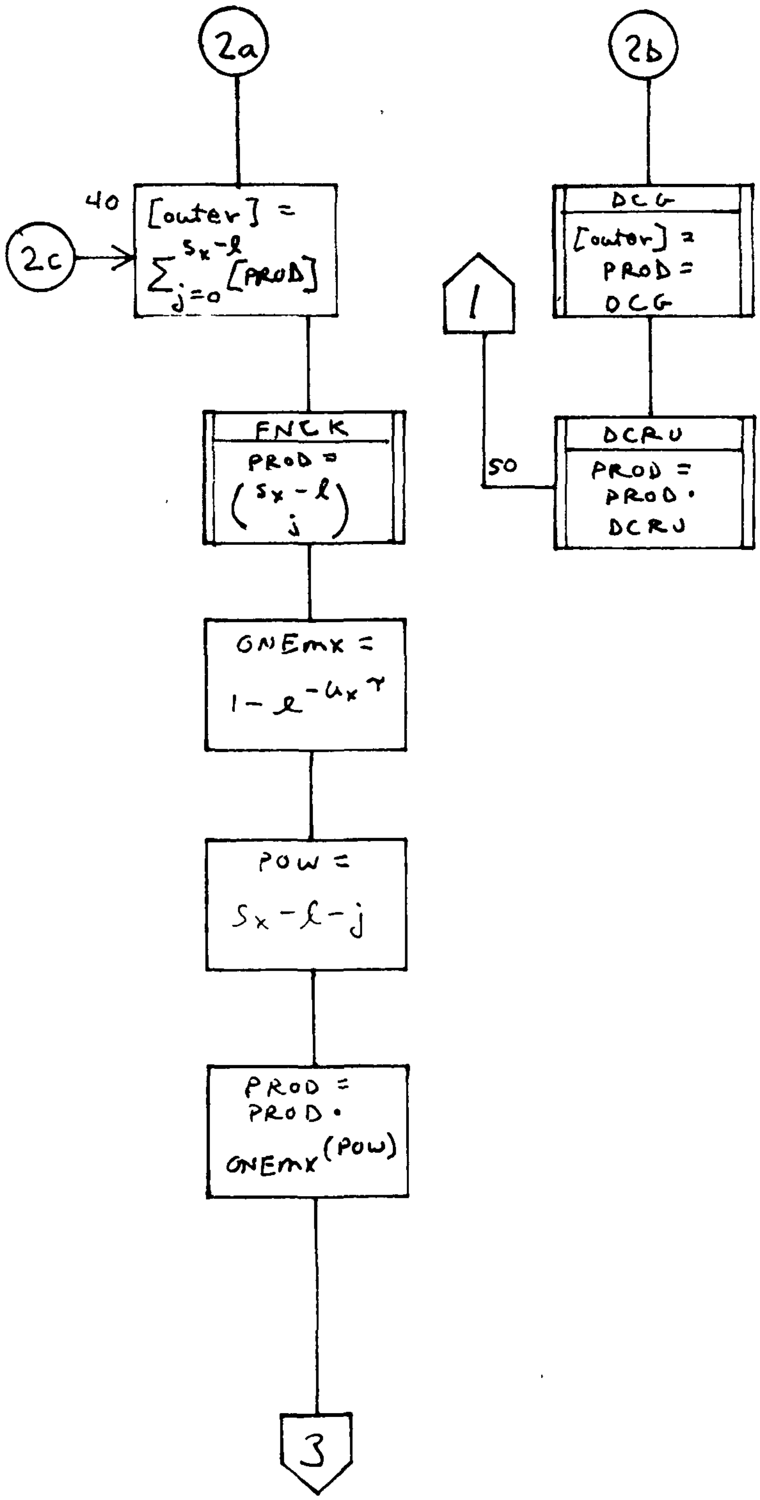
A-13

$$
S(x,t,\tau) = \begin{cases}
\displaystyle\sum_{\ell=0}^{(S_x)} \left[\sum_{j=0}^{(S_x)-\ell} \left\{ \binom{(S_x)-\ell}{j} * \left(1-e^{-(\mu_x)\tau}\right)^{((S_x)-\ell-j)} * e^{-j(\mu_x)\tau} * G(x,y=1,\ell,\tau) * R_u(x,y=2,j',(t-\tau)) \right\} \right] & \text{for } K<\infty \\[2ex]
\displaystyle\sum_{\ell=0}^{(S_x)} \left[G(x,y=1,\ell,\tau) * R_u\left(x,y=2,\left[(S_x)-\ell\right]',(t-\tau)\right) \right], & \text{for } K=\infty
\end{cases}
$$



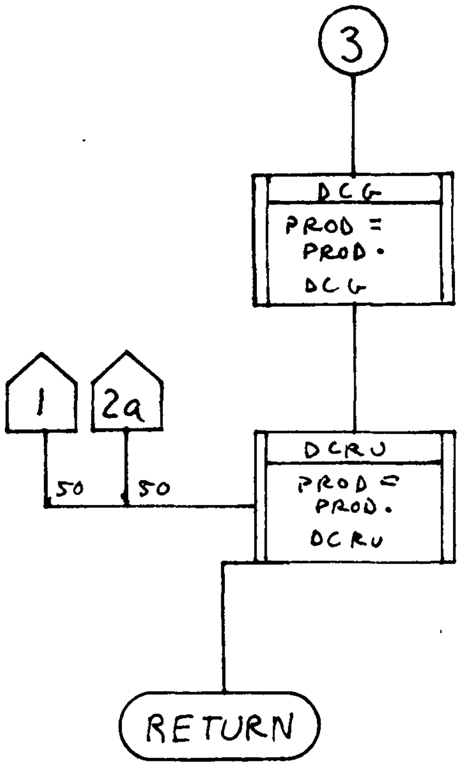
FLOW DIAGRAM

A-13



FLOW DIAGRAM

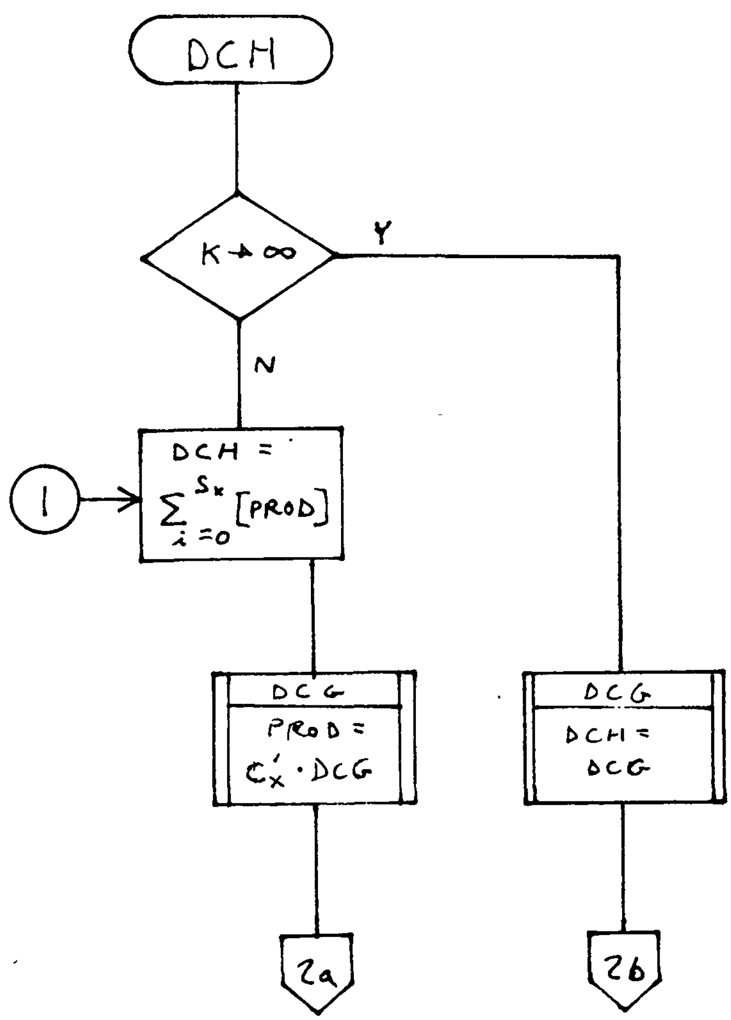
A-13



FLOW DIAGRAM

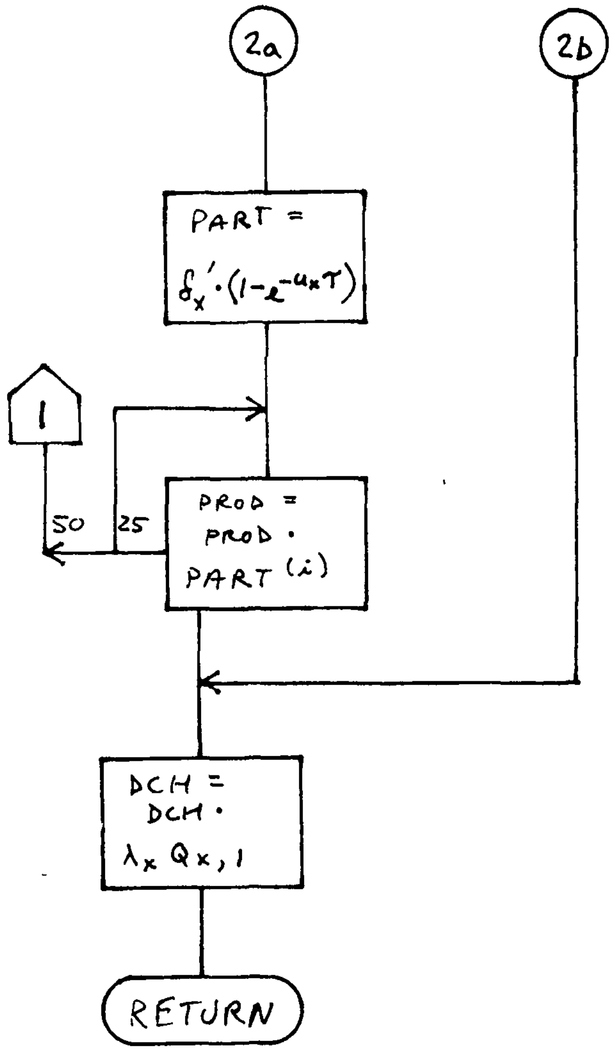
A-14

$$H(x,\tau) = \begin{cases} Q_{x,1}\,\lambda_x \sum_{i=0}^{(S_x)} \left\{ G\left(x, y=1, \left[(S_x)-i\right], \tau\right) * (C'_x)(\delta'_x)^{i} * \left(1-e^{-(\mu_x)\tau}\right)^{i} \right\}, & \text{for } K<\infty \\ \lambda_x Q_{x,1}(C'_x) * G\left(x, y=1, (S_x), \tau\right), & \text{for } K=\infty \end{cases}$$



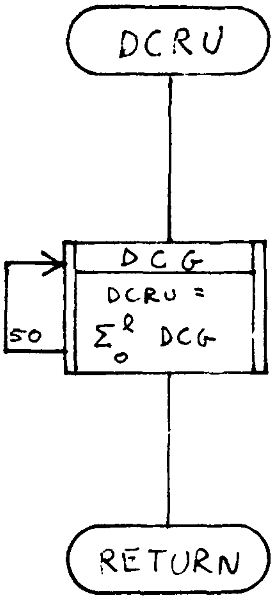
FLOW DIAGRAM

A-14



FLOW DIAGRAM

A-15

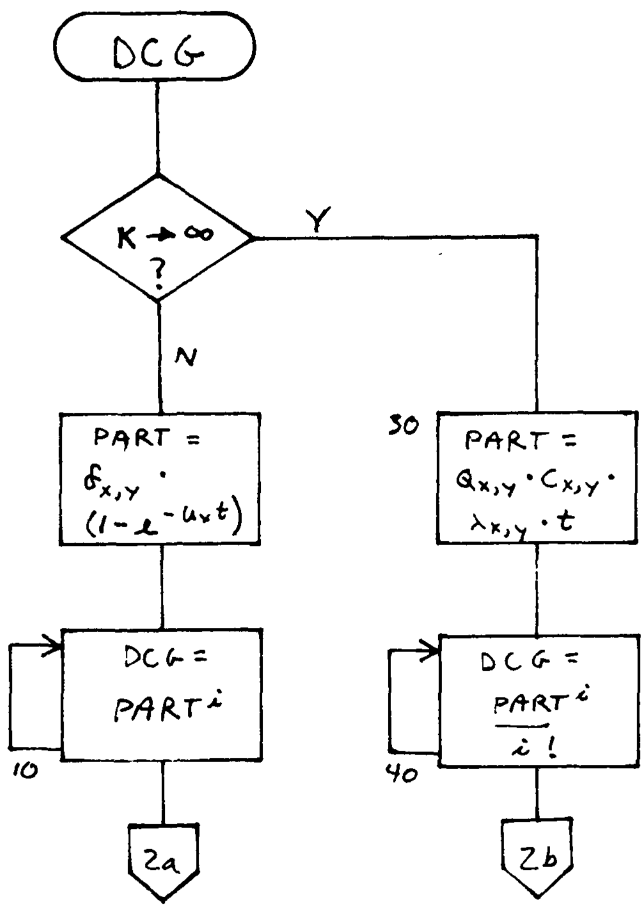


$$R_u(x,y,\ell,r) = \sum_{i=0}^{\ell} G(x,y,i,r)$$

FLOW DIAGRAM

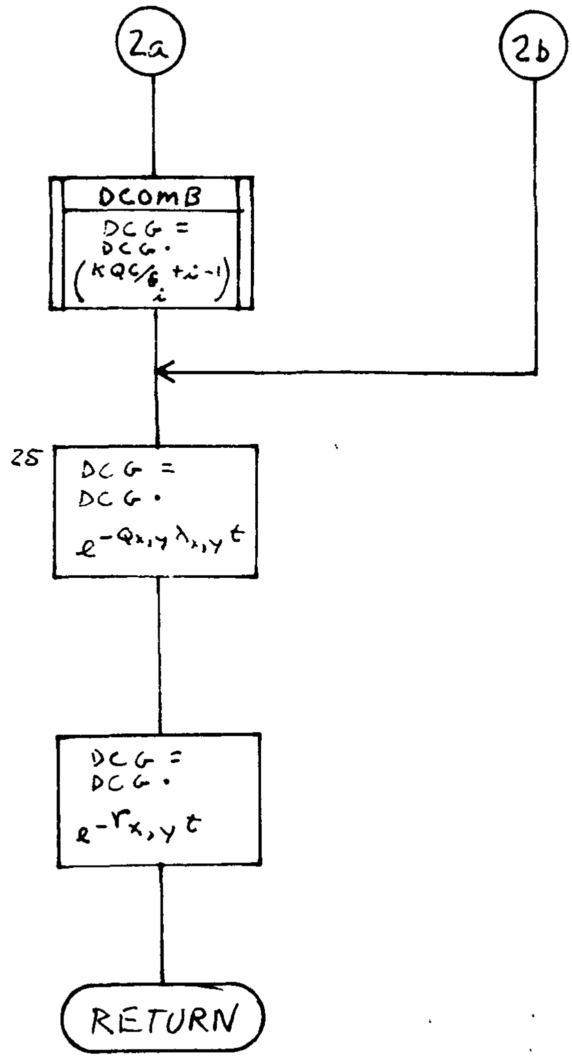
A-16

$$G(x,y,i,t) = \begin{cases} e^{-(\gamma_{x,y})t} * (\delta_{x,y})^{i} * \binom{(K_x)(Q_{x,y})(C_{x,y})(\delta_{x,y})^{-1}+i-1}{i} * \left(1-e^{-(\mu_x)t}\right)^{i} * e^{-(Q_{x,y})(\lambda_x)t} \\ \text{for } K<\infty \\ e^{-(\gamma_{x,y})t} * \dfrac{\left((Q_{x,y})\ (C_{x,y})\ (\lambda_x)t\right)^{i}}{i!} * e^{-(Q_{x,y})\ (\lambda_x)t}, \\ \text{for } K=\infty \end{cases}$$



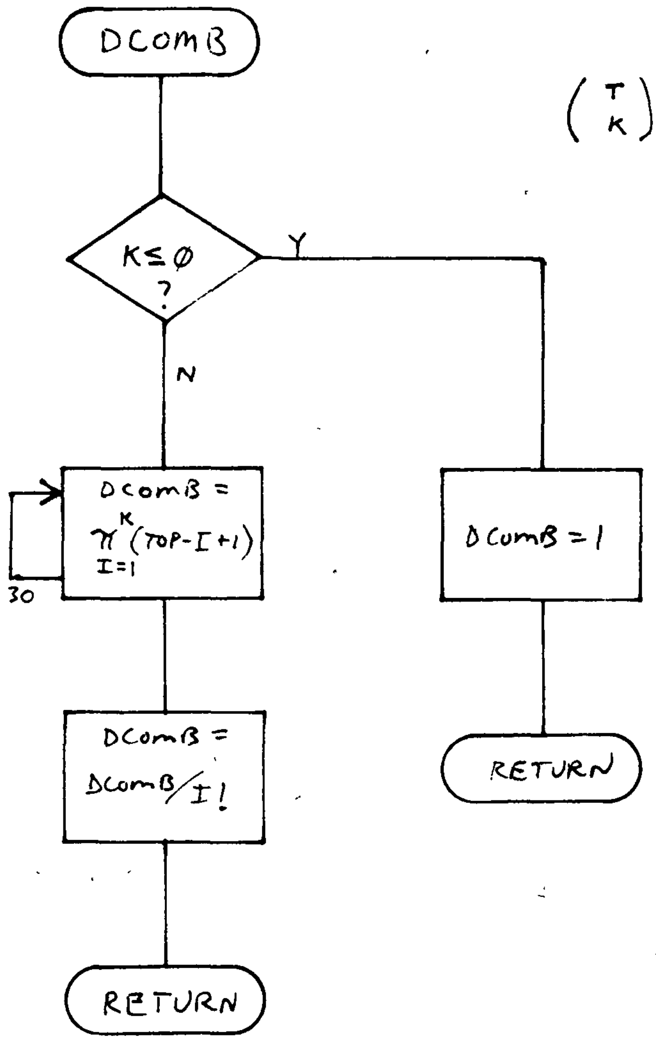
FLOW DIAGRAM

A-16



FLOW DIAGRAM

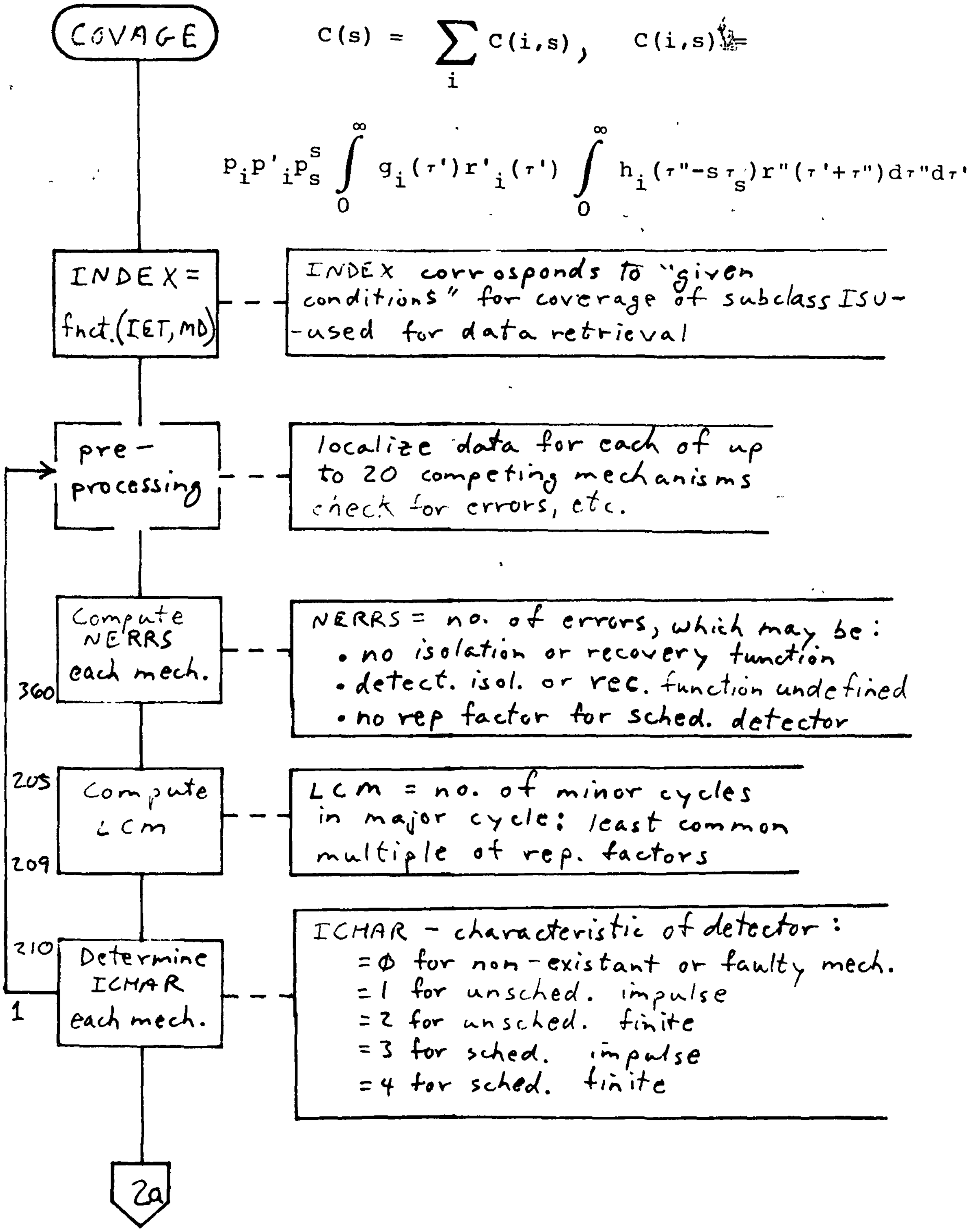
A-17



$$\binom{T}{K} = \begin{cases} \frac{1}{K!} \prod_{I=1}^{K} (T-K+1), & \text{for } K>0 \\ 1, & \text{for } K \le 0 \end{cases}$$

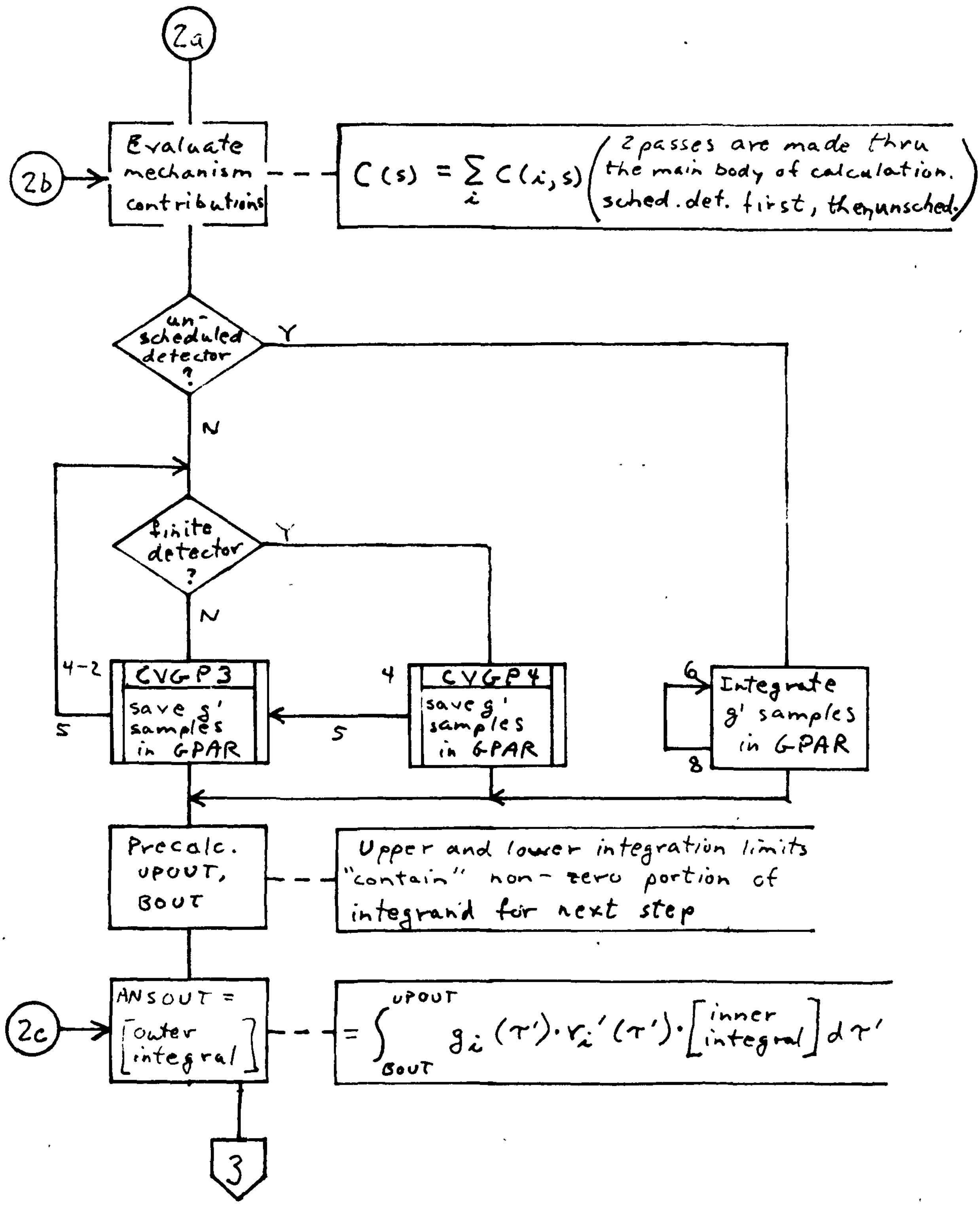
FLOW DIAGRAM

A-18



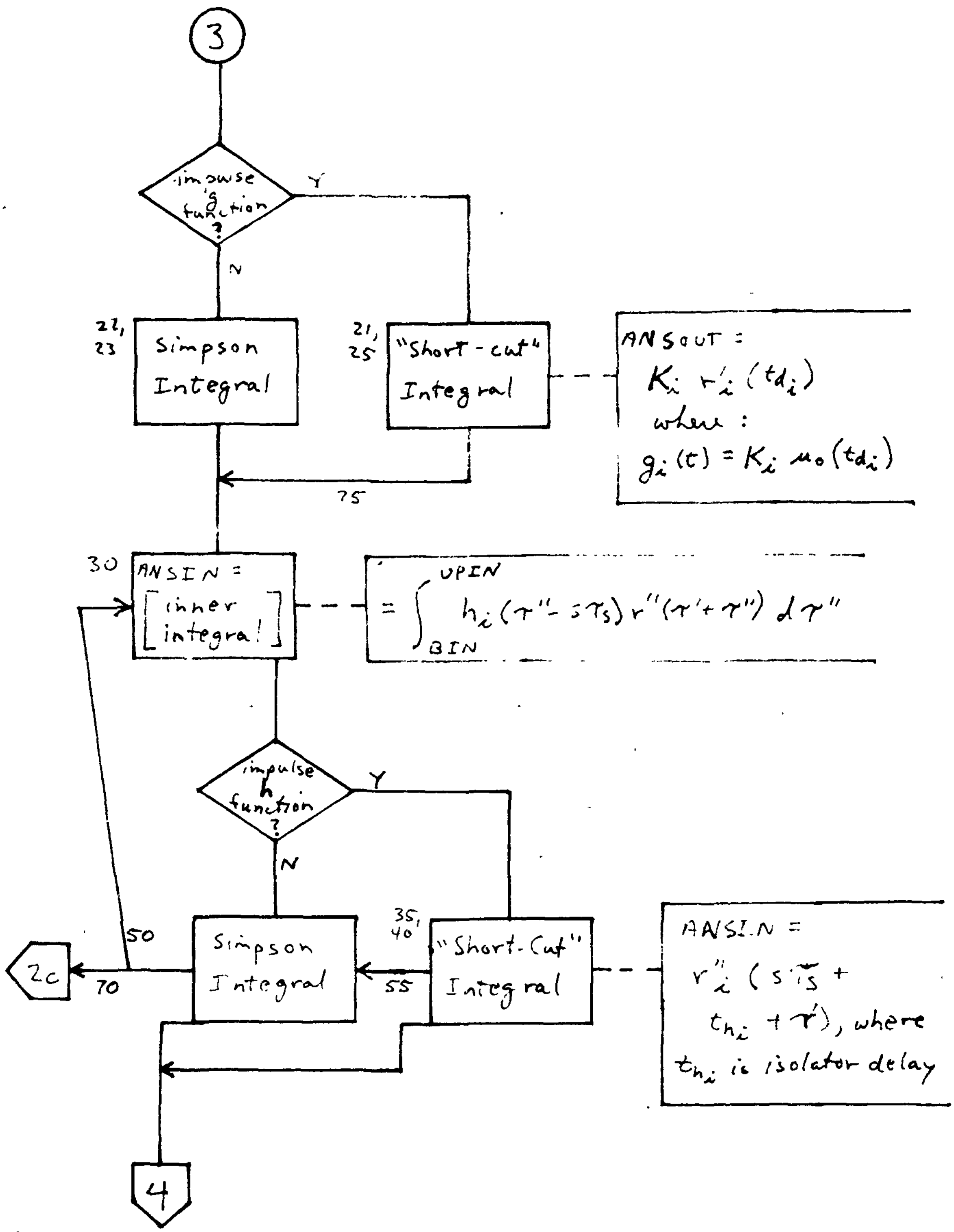
FLOW DIAGRAM

A-18



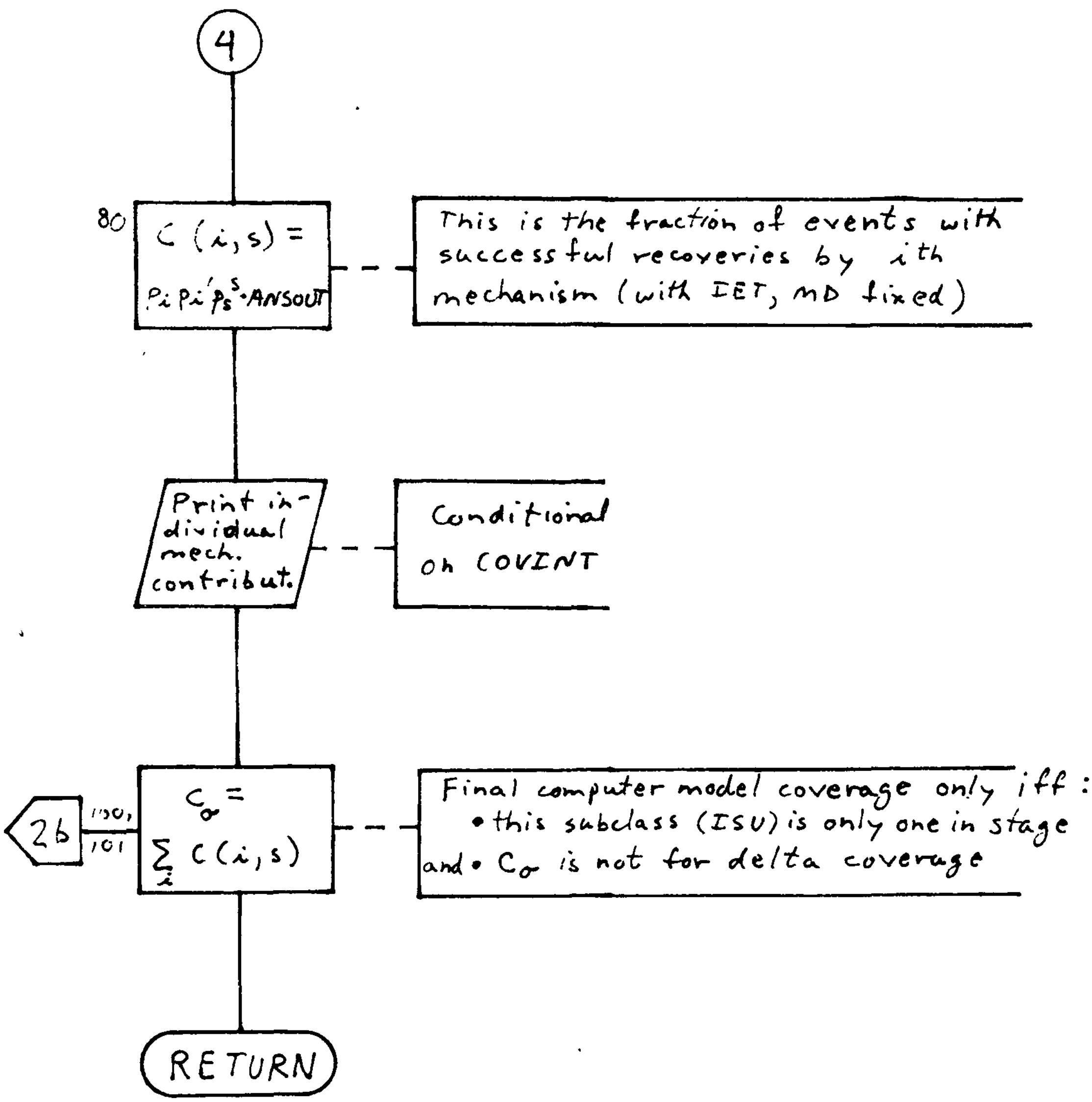
FLOW DIAGRAM

A-18



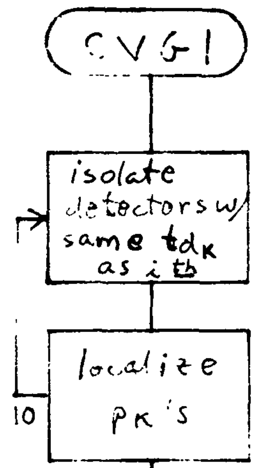
FLOW DIAGRAM

A-18



FLOW DIAGRAM

A-19

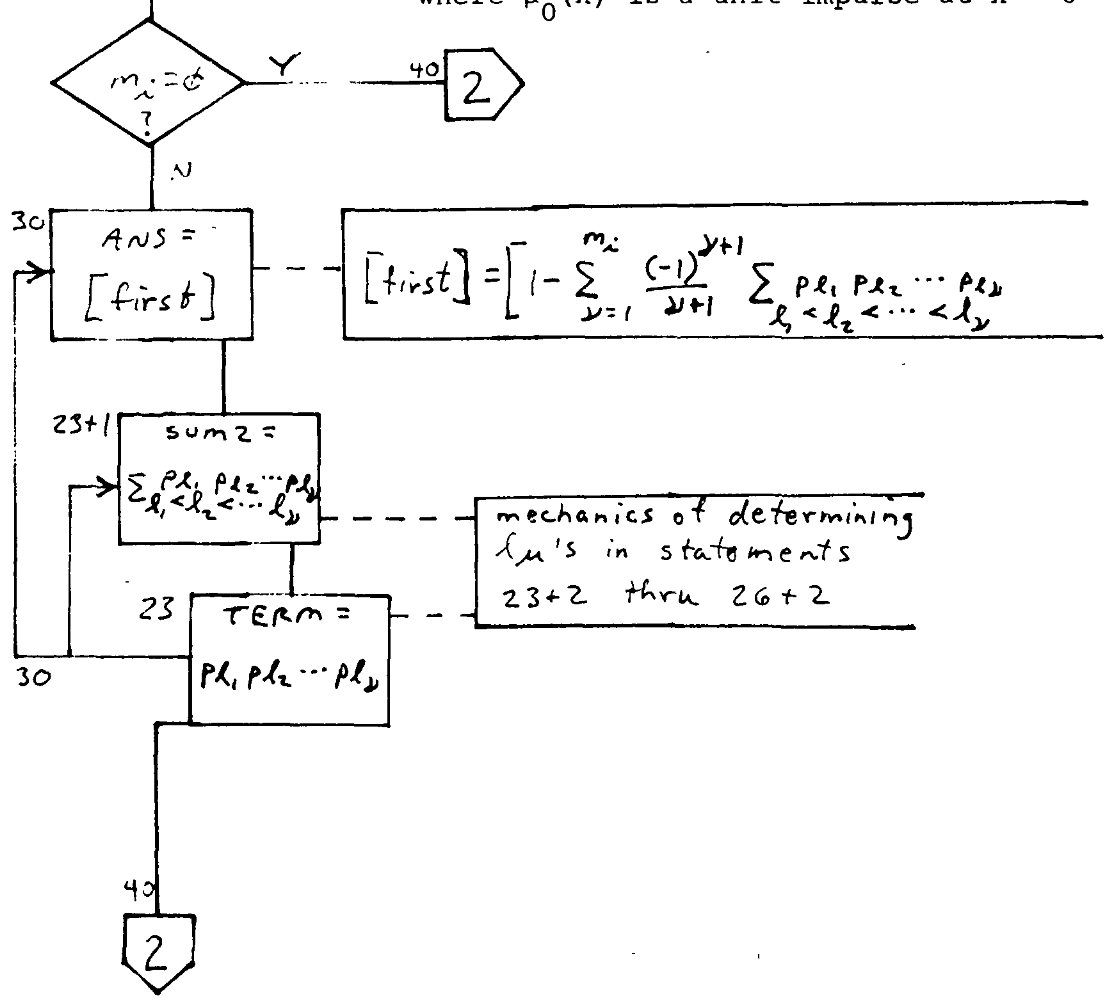


$$g_i(\tau) = \left[1 - \sum_{\nu=1}^{m_i} \frac{(-1)^{\nu+1}}{\nu+1} \sum_{\ell_1<\ell_2<\cdots<\ell_\nu} p\ell_1 p\ell_2 \cdots p\ell_\nu\right] *$$

$$\prod_{\substack{k\neq i\\ k\neq \ell_1,\ell_2,\ldots,\ell_{m_i}}} \left[1 - p_k F_k(t_{d_i} - t_{d_k})\right] * \left[1 - \sum_j p_j \int_0^{t_{d_i}} g_j'(\eta)\, d\eta\right] *$$

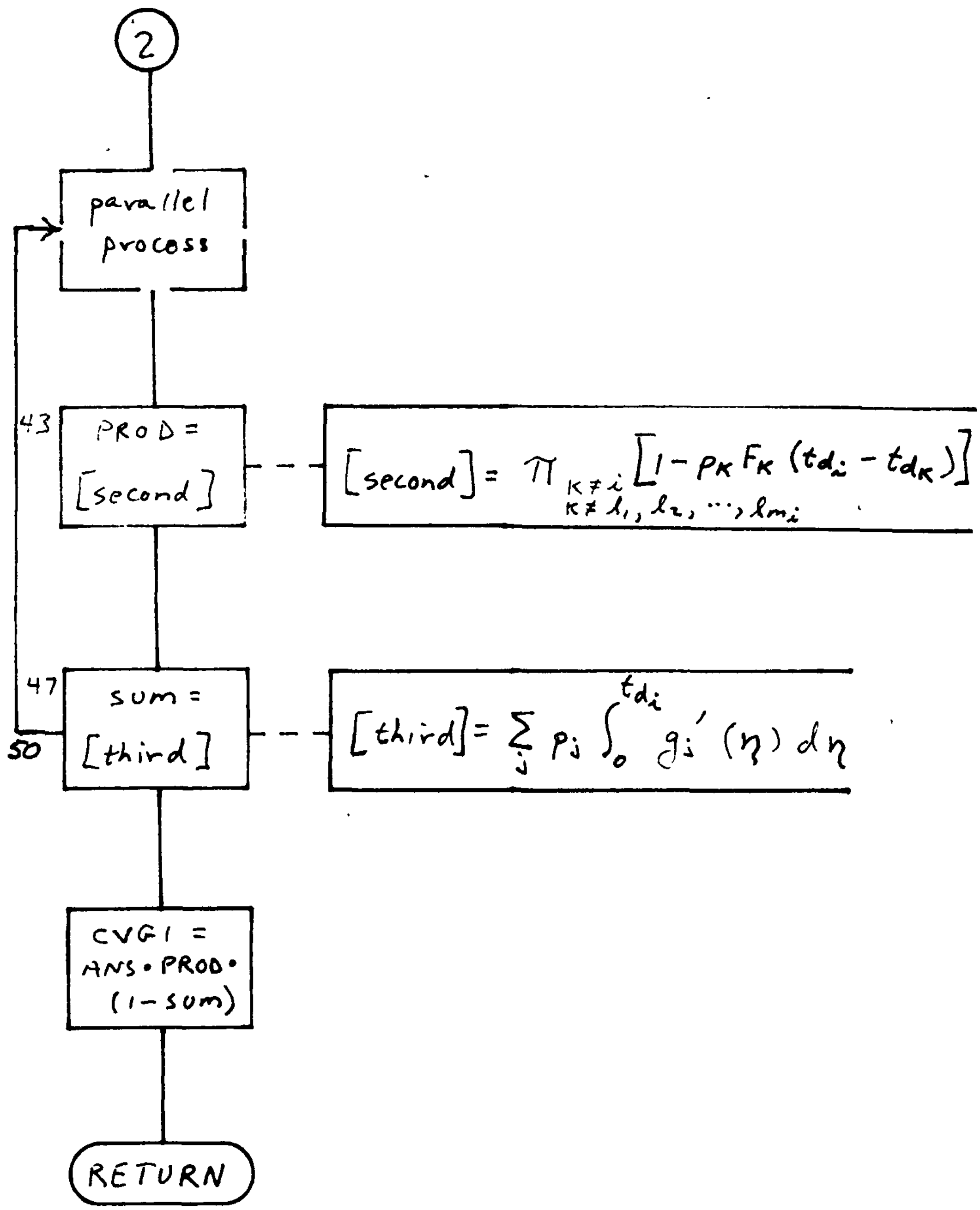
$$\mu_0(\tau - \tau_{d_i}), \quad \text{for } f_i(\tau) = \mu_0(\tau - \tau_{d_i})$$

where $\mu_0(x)$ is a unit impulse at $x = 0$



FLOW DIAGRAM

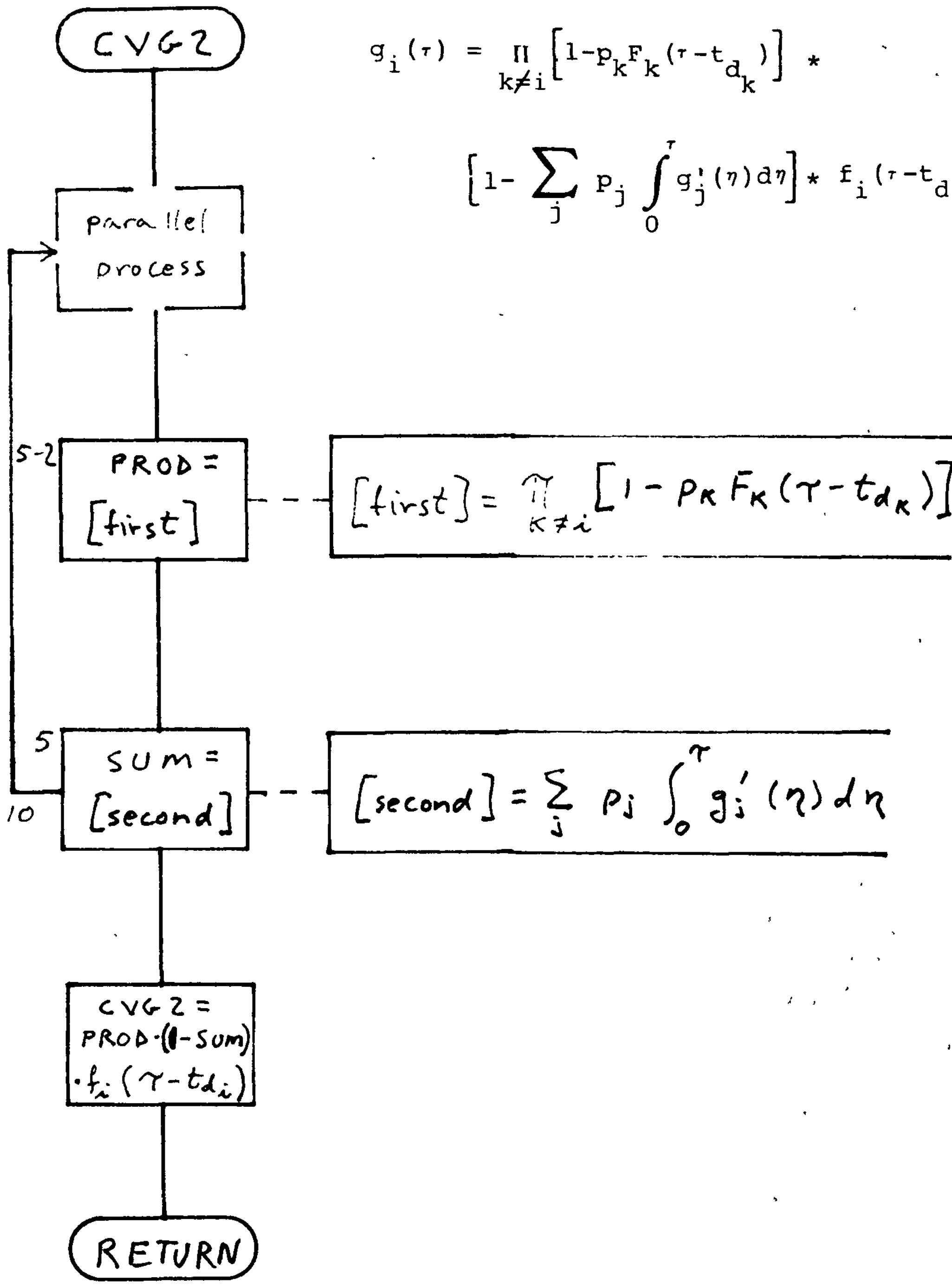
A-19



FLOW DIAGRAM

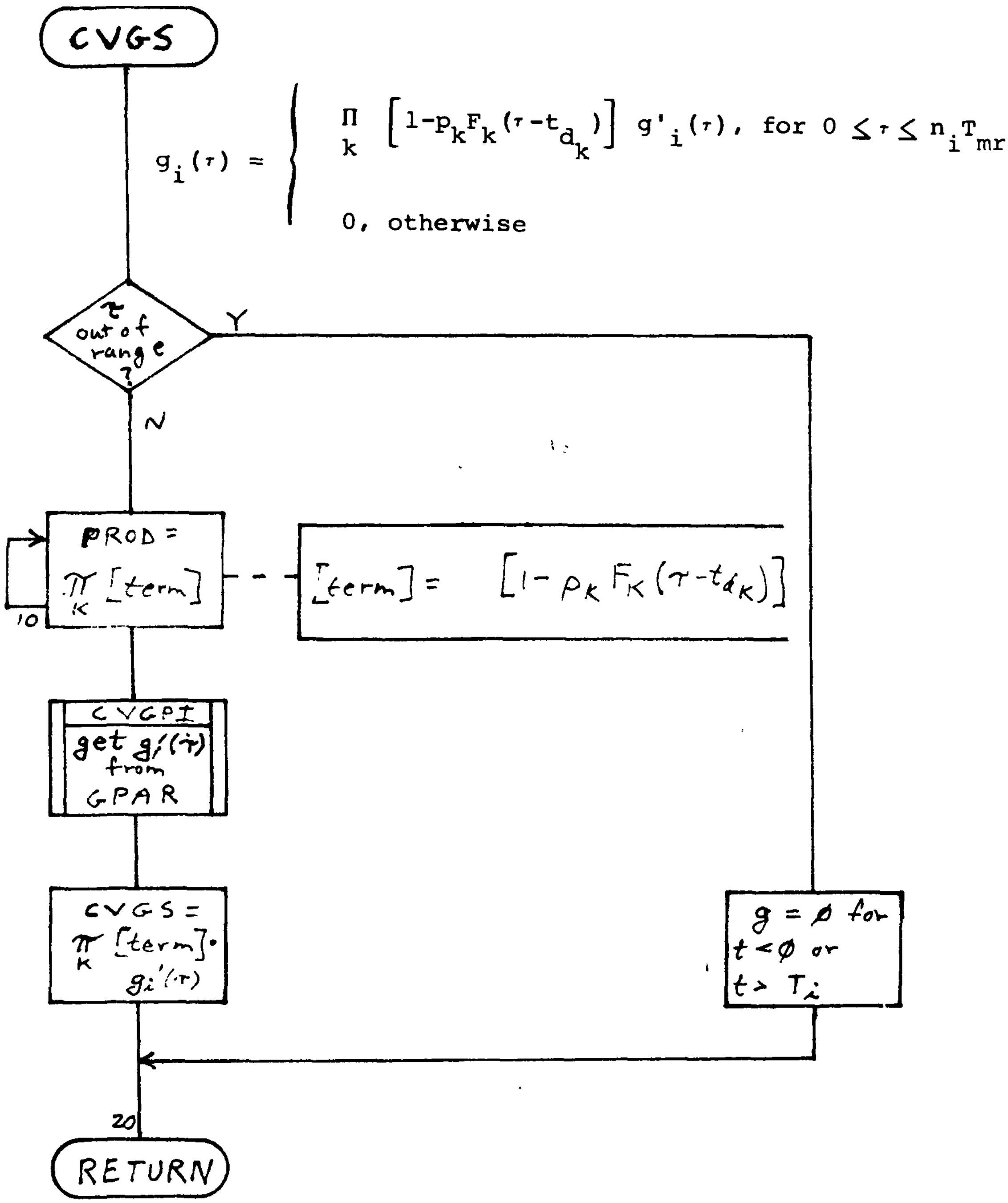
A-20

$$g_i(\tau) = \prod_{k \neq i} \left[1 - p_k F_k(\tau - t_{d_k})\right] * \left[1 - \sum_j p_j \int_0^\tau g_j'(\eta)\, d\eta\right] * f_i(\tau - t_{d_i})$$



FLOW DIAGRAM

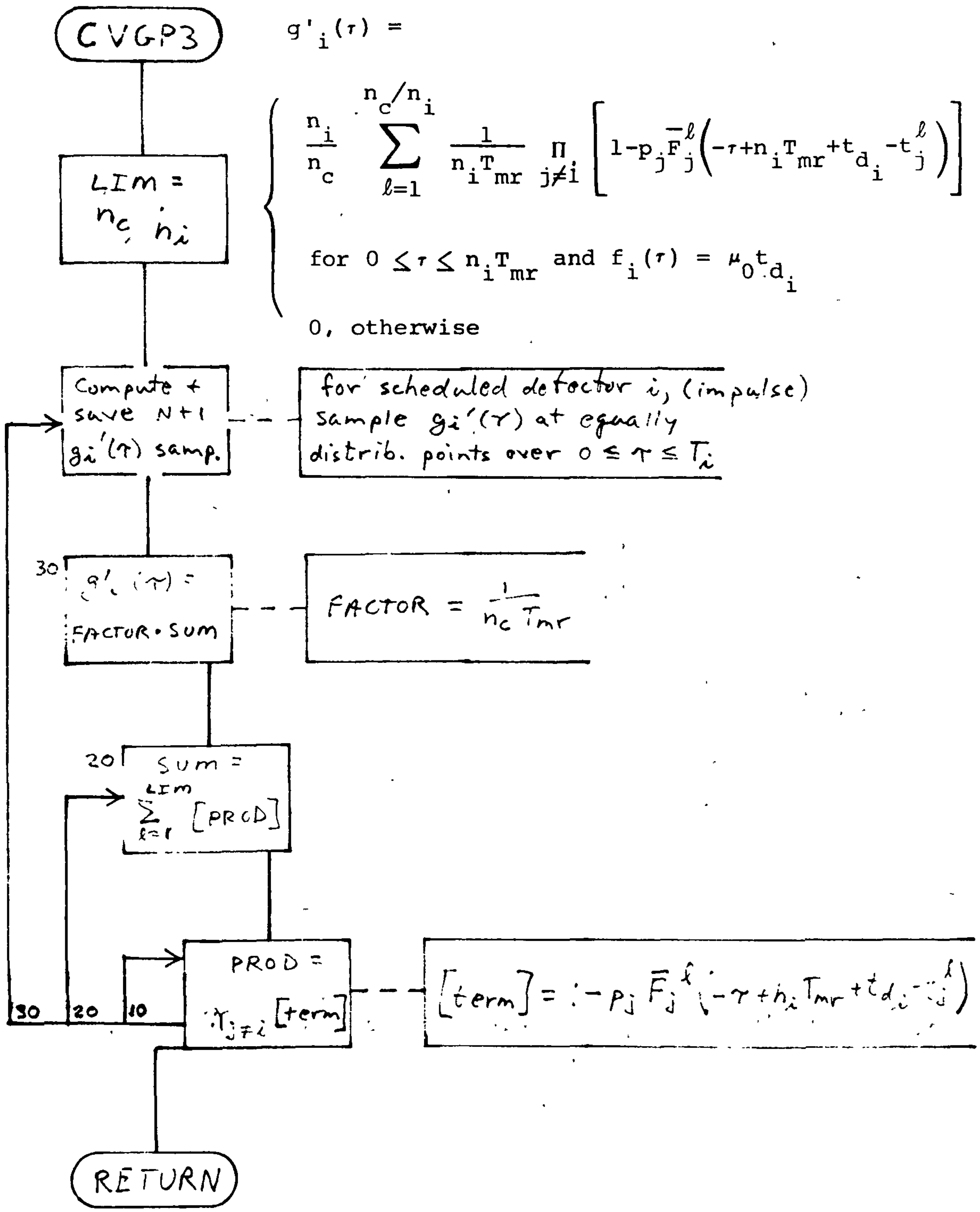
A-21



FLOW DIAGRAM

A-22

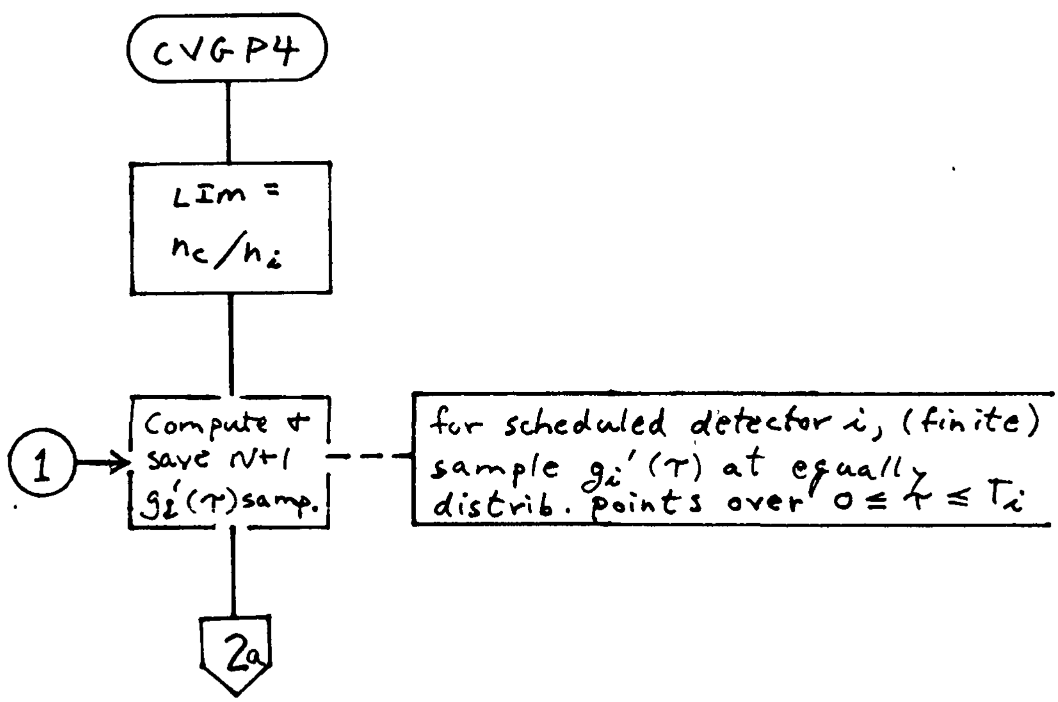
$$g'_i(\tau) = \begin{cases} \dfrac{n_i}{n_c} \displaystyle\sum_{\ell=1}^{n_c/n_i} \dfrac{1}{n_i T_{mr}} \prod_{j \neq i} \left[1 - p_j \bar{F}_j^{\ell}\left(-\tau + n_i T_{mr} + t_{d_i} - t_j^{\ell}\right) \right] \\ \text{for } 0 \leq \tau \leq n_i T_{mr} \text{ and } f_i(\tau) = \mu_0 t_{d_i} \\ 0, \text{ otherwise} \end{cases}$$



FLOW DIAGRAM

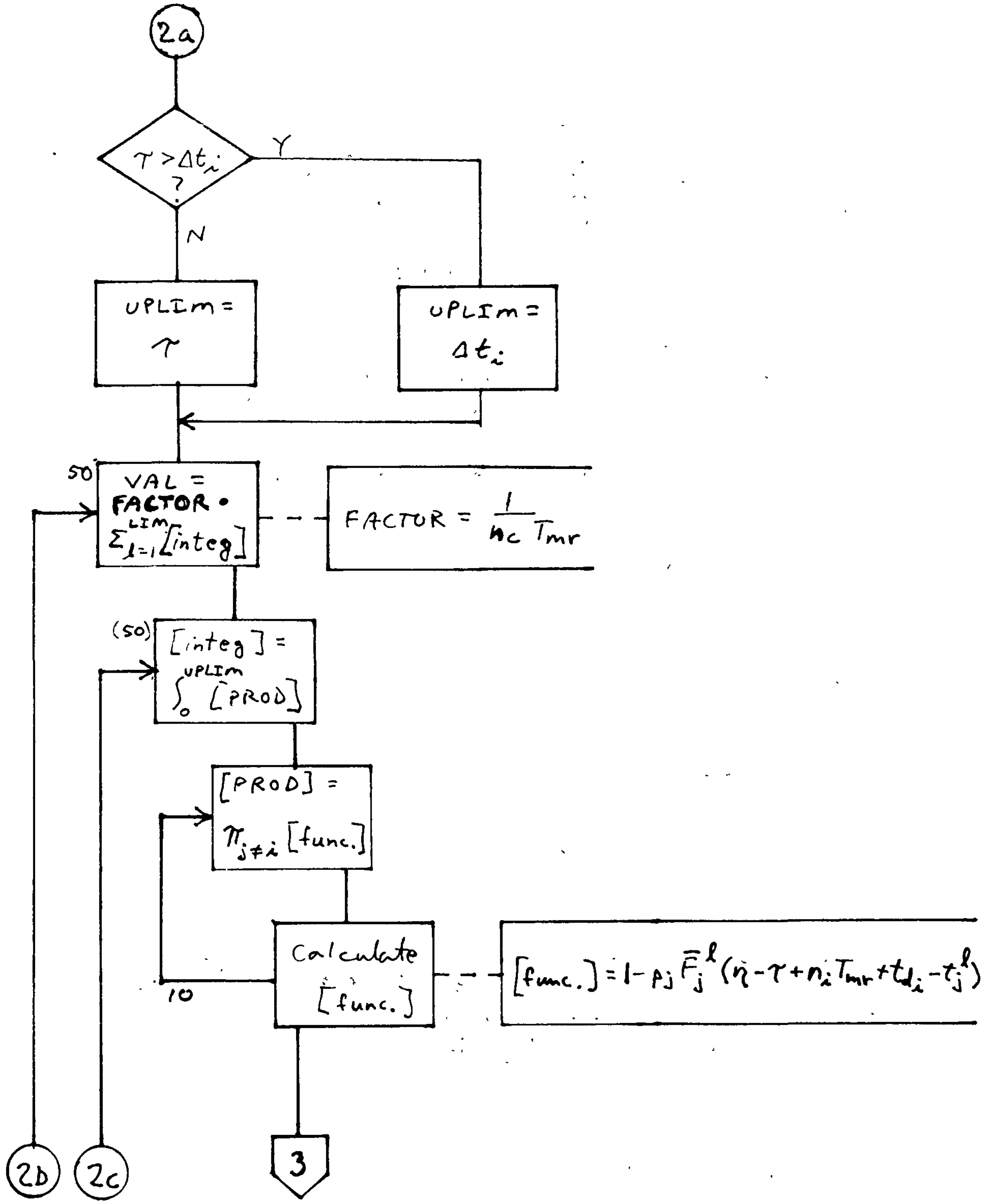
A-23

$$g'_i(\tau) = \begin{cases} \dfrac{1-F_i(\tau)}{n_i T_{mr}} + \dfrac{n_i}{n_c} \displaystyle\sum_{\ell=1}^{n_c/n_i} \dfrac{1}{n_i T_{mr}} \int_0^{\tau} \prod_{j\neq i} \left[1-p_j \bar{F}_j^{\ell}(\eta-\tau+n_i T_{mr}+t_{d_i}-t_j^{\ell})\right] \\ \quad * f_i(\eta)\,d\eta \qquad \text{for } 0 \le \tau \le \Delta t_i \\ \dfrac{n_i}{n_c} \displaystyle\sum_{\ell=1}^{n_c/n_i} \dfrac{1}{n_i T_{mr}} \int_0^{\Delta t_i} \prod_{j\neq i} \left[1-p_j \bar{F}_j^{\ell}(\eta-\tau+n_i T_{mr}+t_{d_i}-t_j^{\ell})\right] \\ \quad * f_i(\eta)\,d\eta \qquad \text{for } \Delta t_i < \tau \le n_i T_{mr} \\ 0, \text{ otherwise} \end{cases}$$



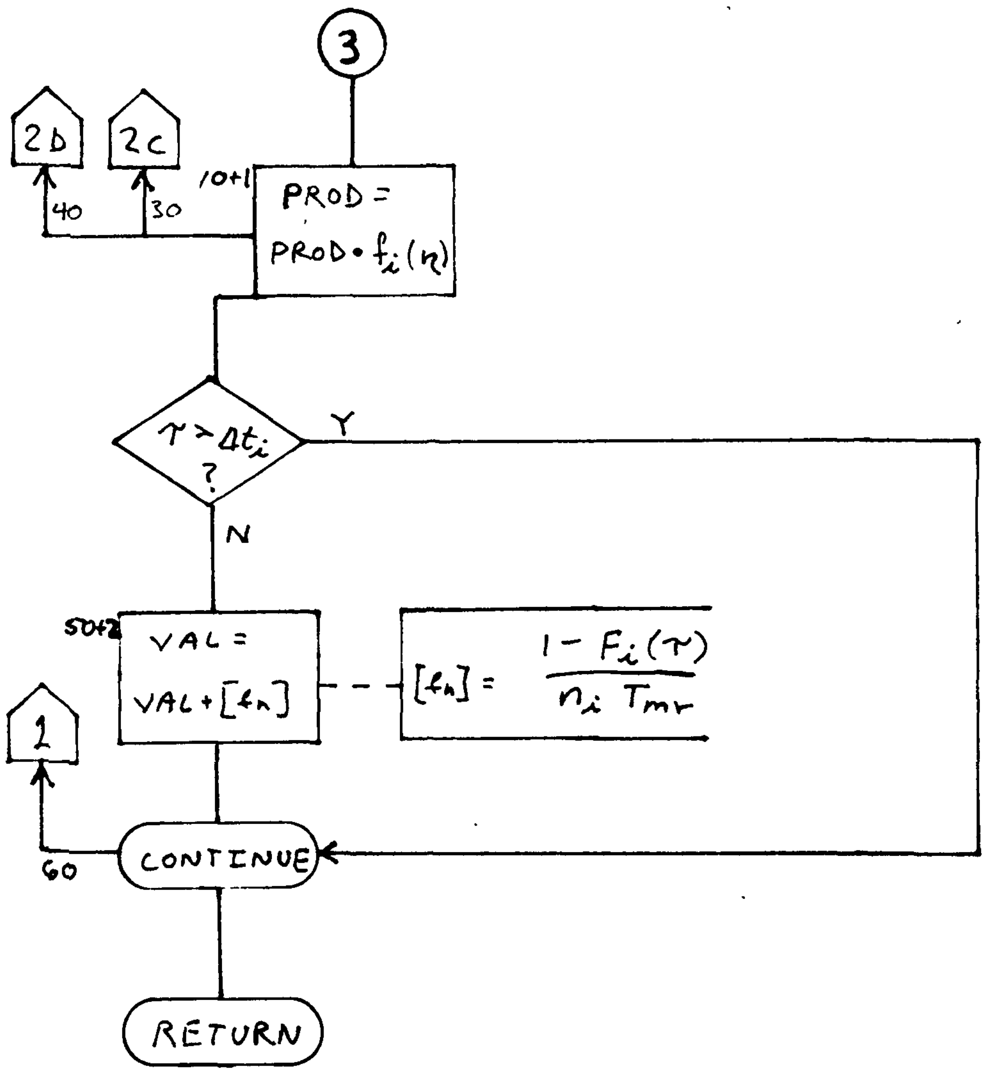
FLOW DIAGRAM

A-23



FLOW DIAGRAM

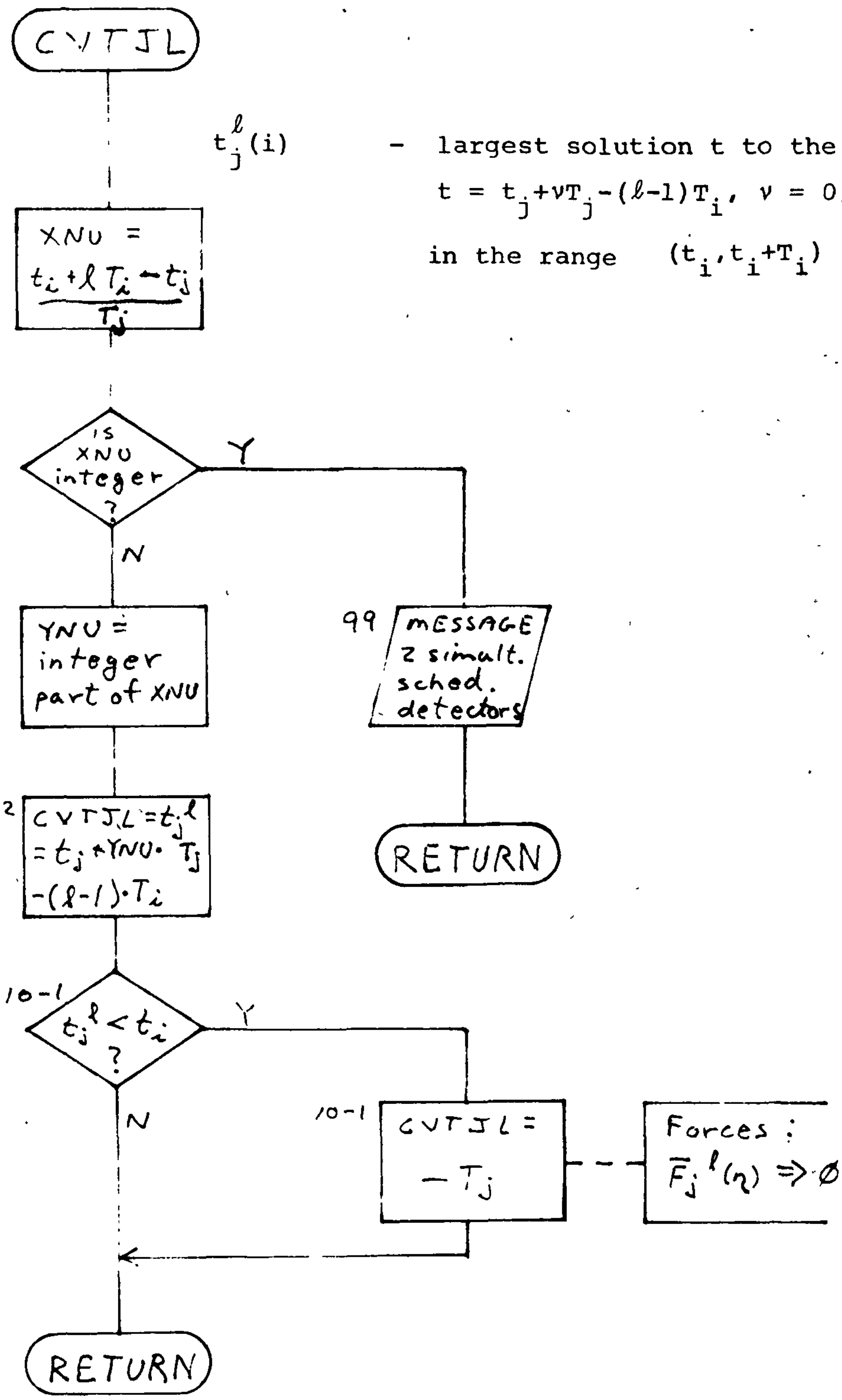
A-23



FLOW DIAGRAM

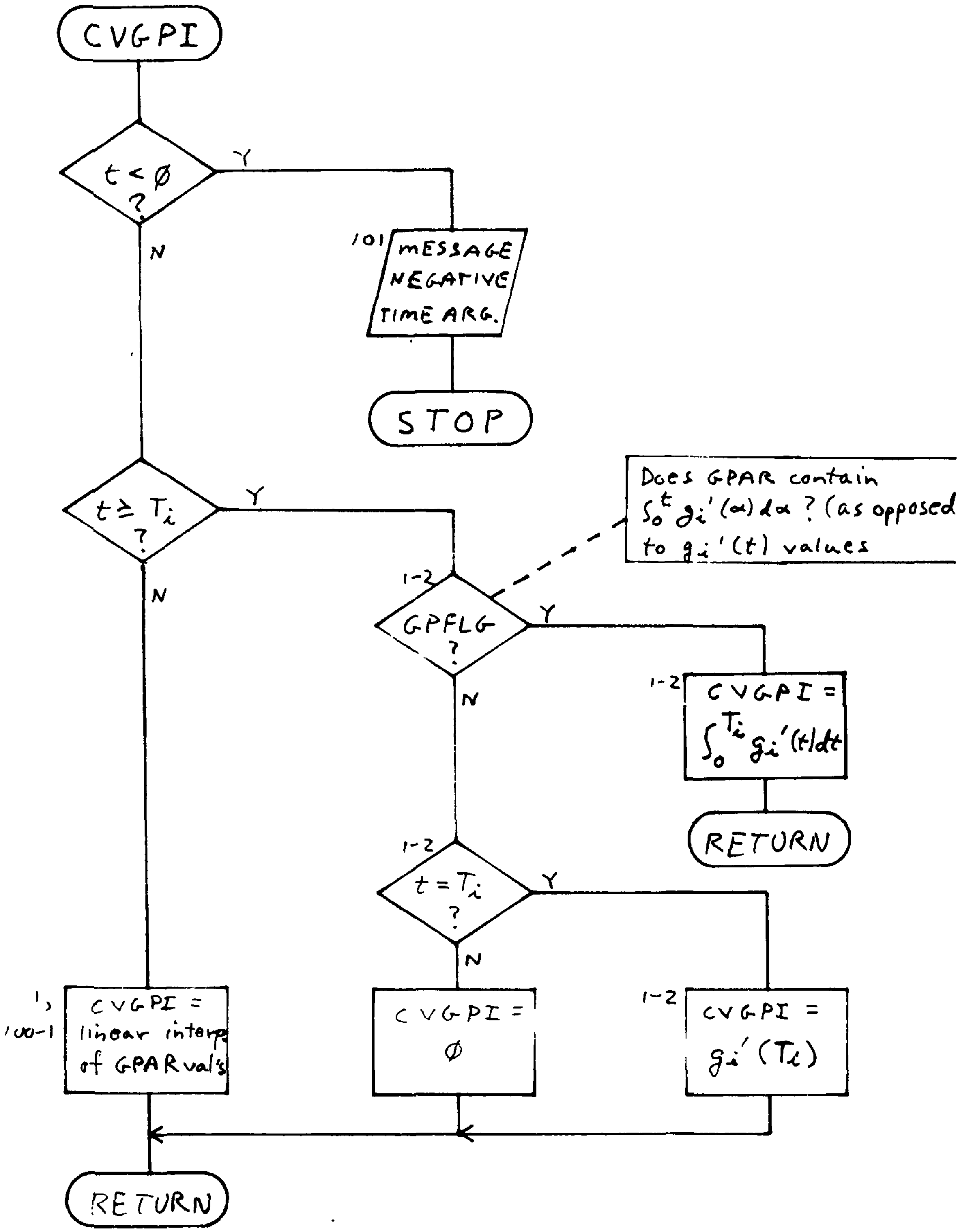
A-24

$t_j^{\ell}(i)$ - largest solution t to the equation $t = t_j + \nu T_j - (\ell-1)T_i$, $\nu = 0,1,2,\ldots,$ in the range (t_i, t_i+T_i)



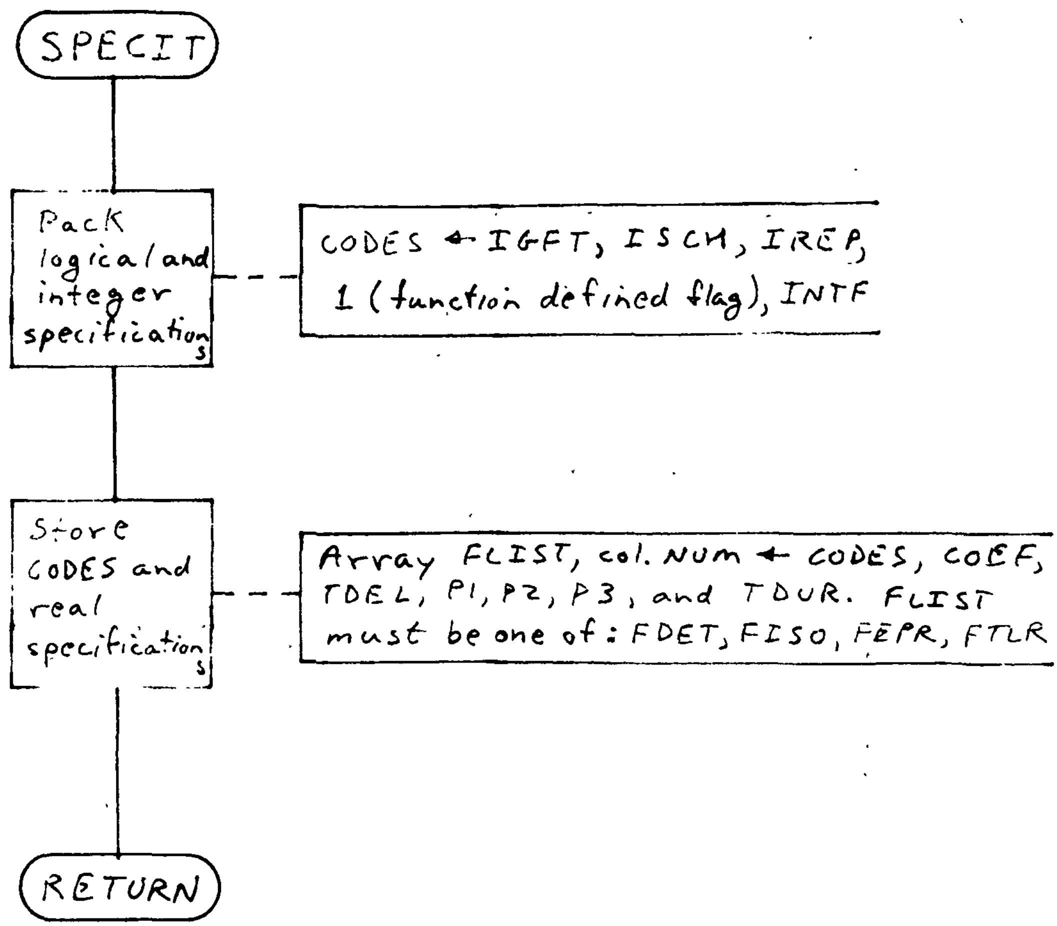
FLOW DIAGRAM

A-25



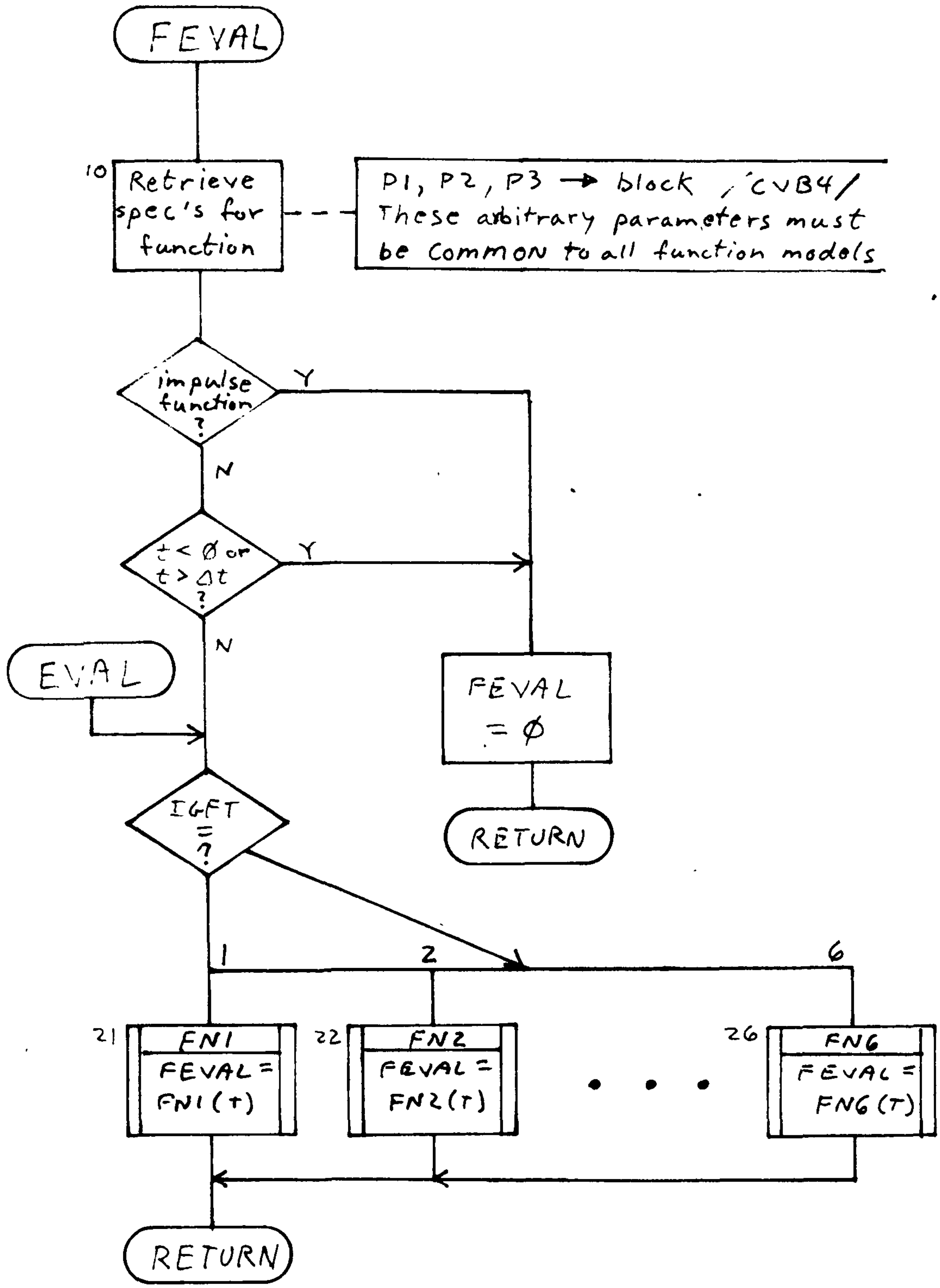
FLOW DIAGRAM

A-26



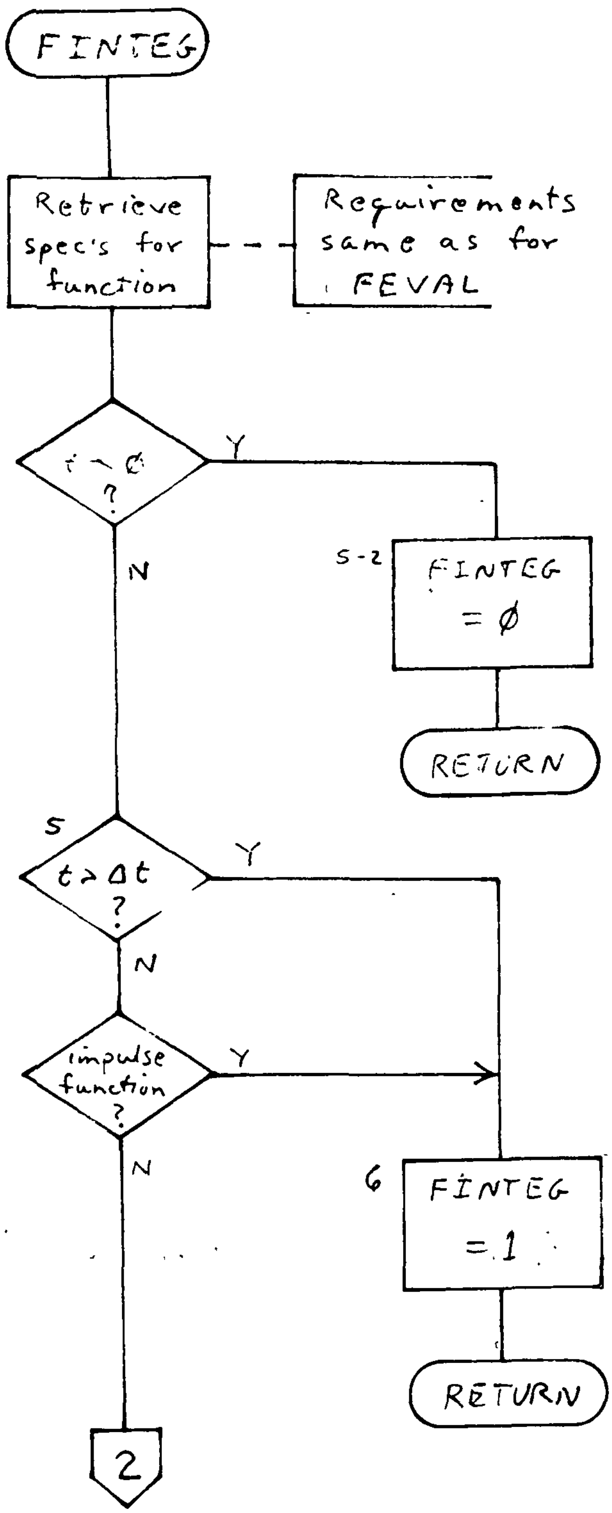
FLOW DIAGRAM

A-27



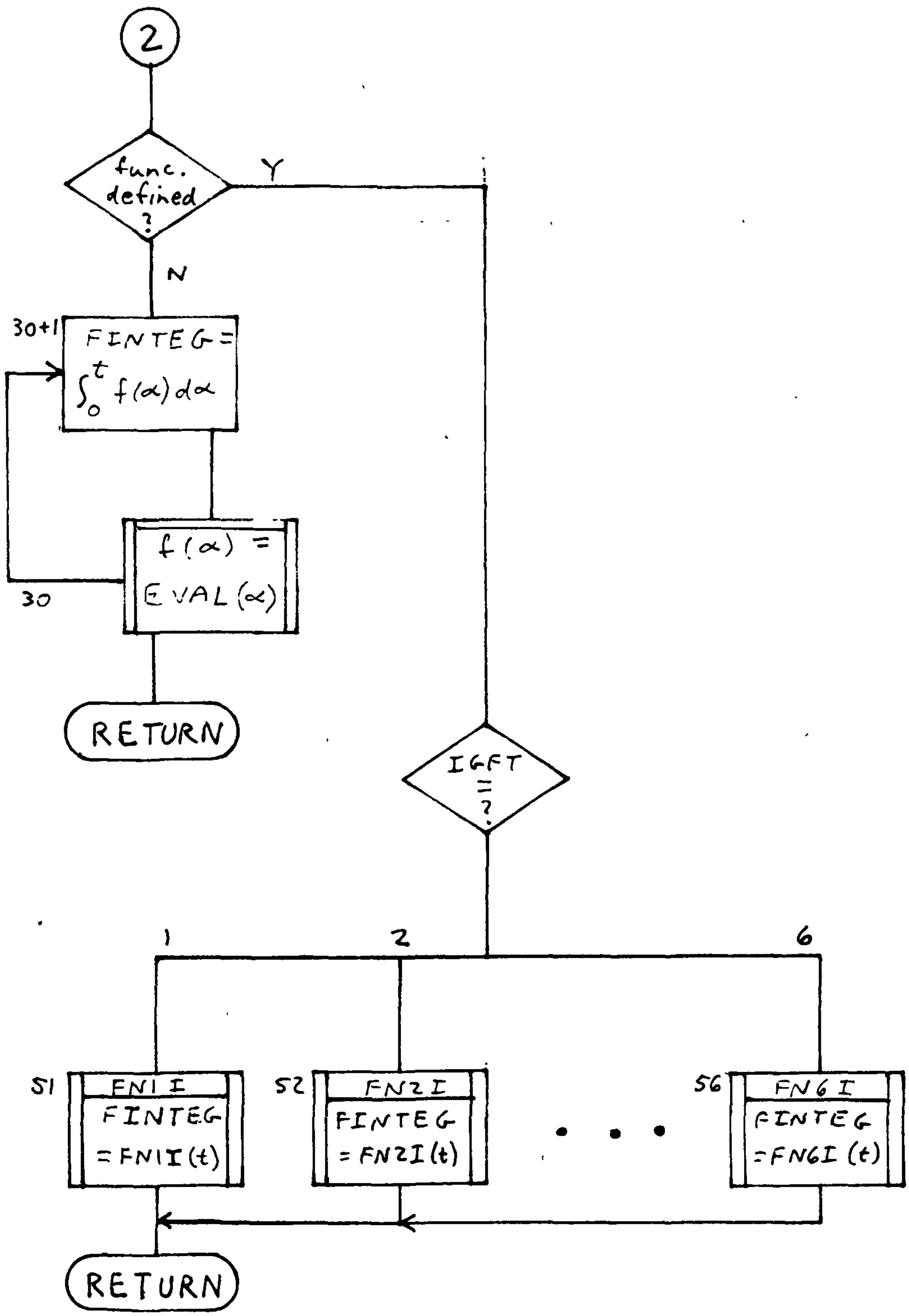
FLOW DIAGRAM

A-28



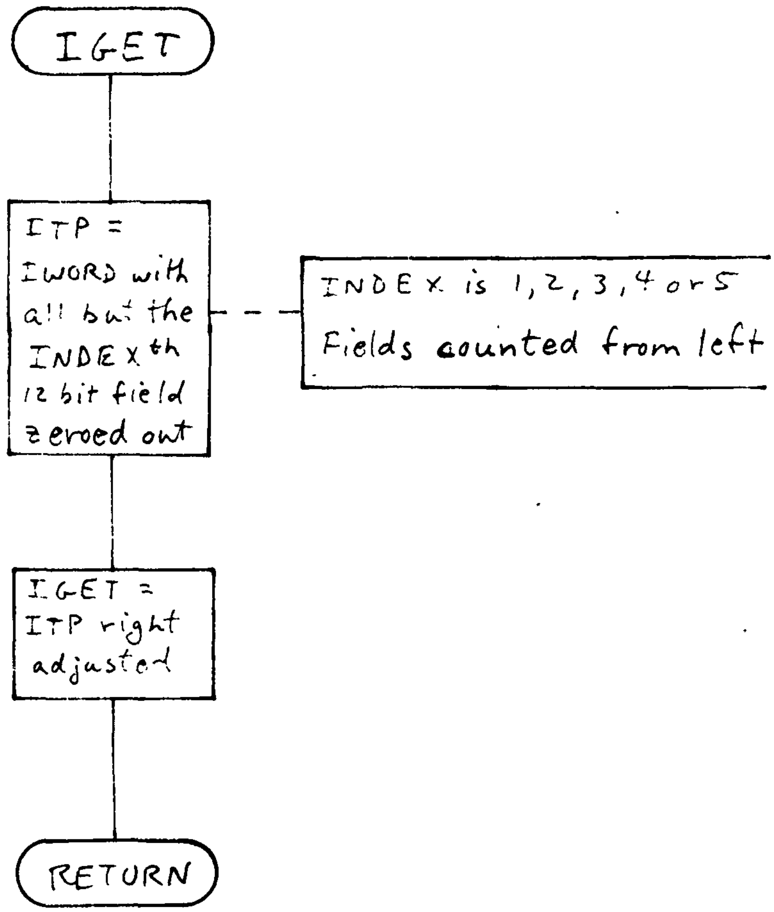
FLOW DIAGRAM

A-28



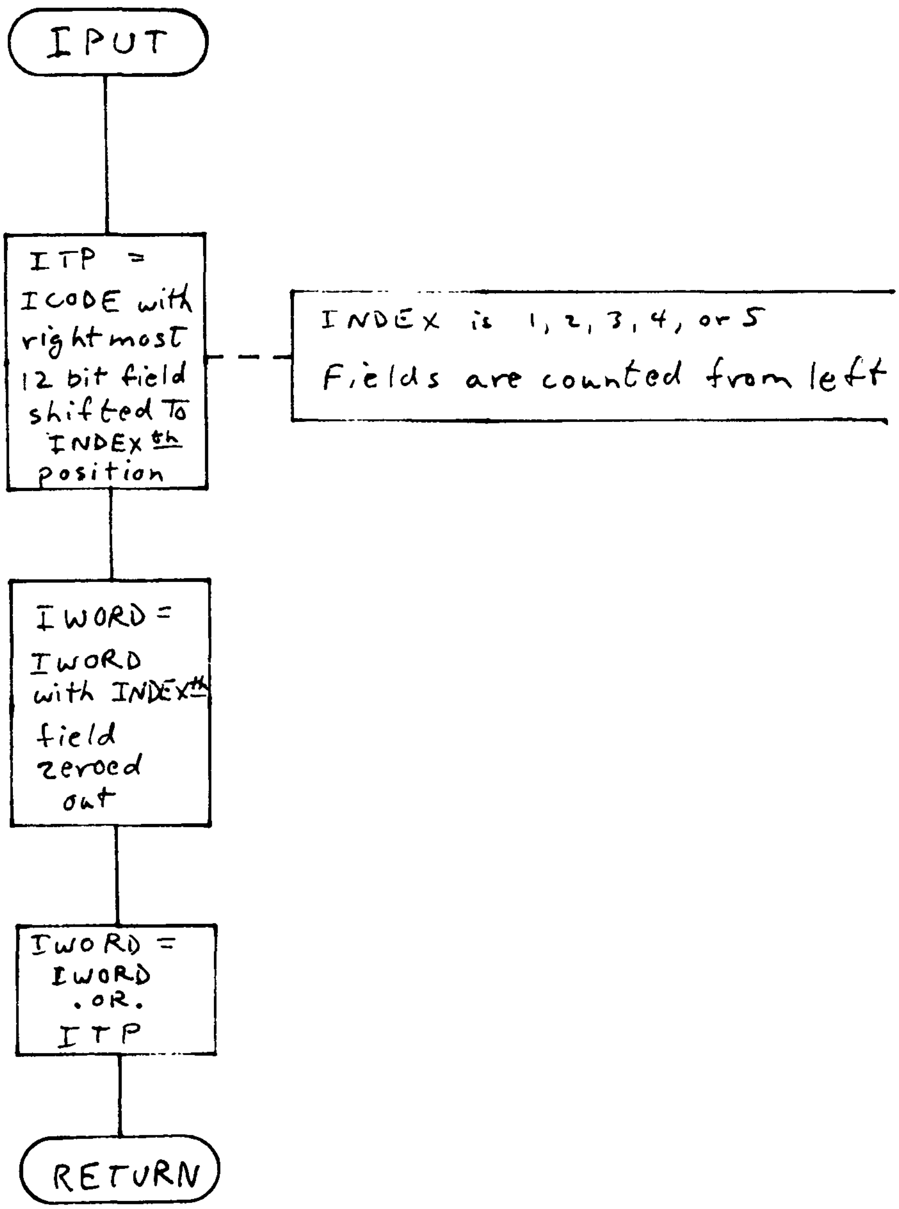
FLOW DIAGRAM

A-29



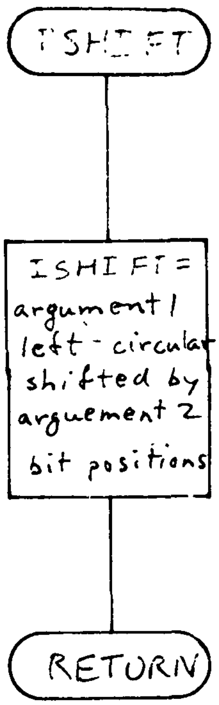
FLOW DIAGRAM

A-30



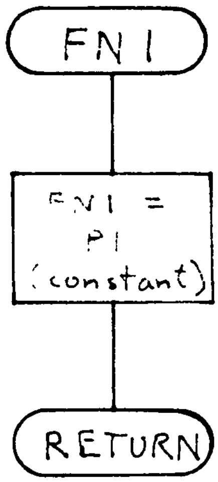
FLOW DIAGRAM

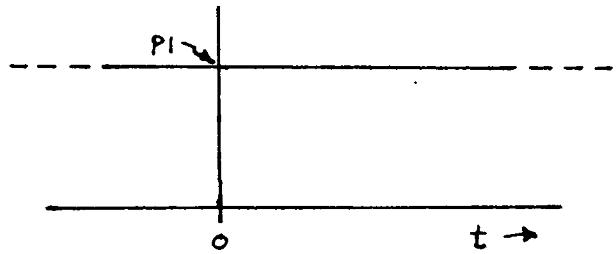
A-31



FLOW DIAGRAM

A-32

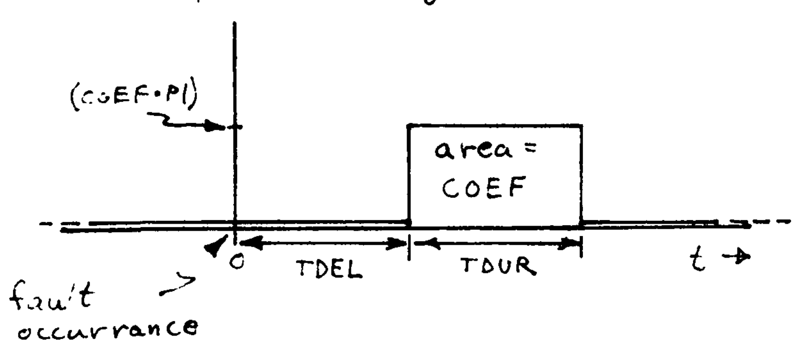




GRAPH OF FN1 (t)
(SINGLE PULSE)

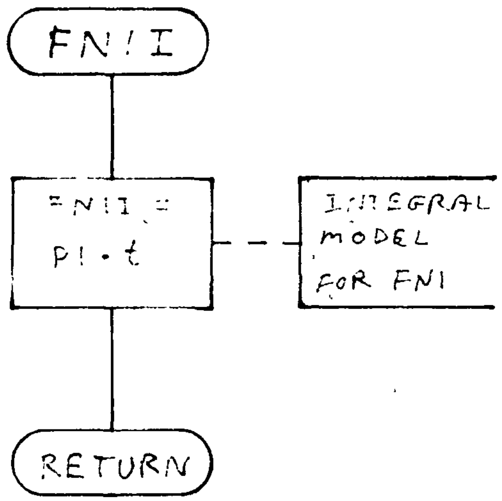
Note:

Function and integral models should be defined for all t as above. Evaluation functions FEVAL and FINTEG and their calling programs provide the additional constaints required by the coverage model. Thus an unscheduled detection function using FN1 is effectively like the figure below –



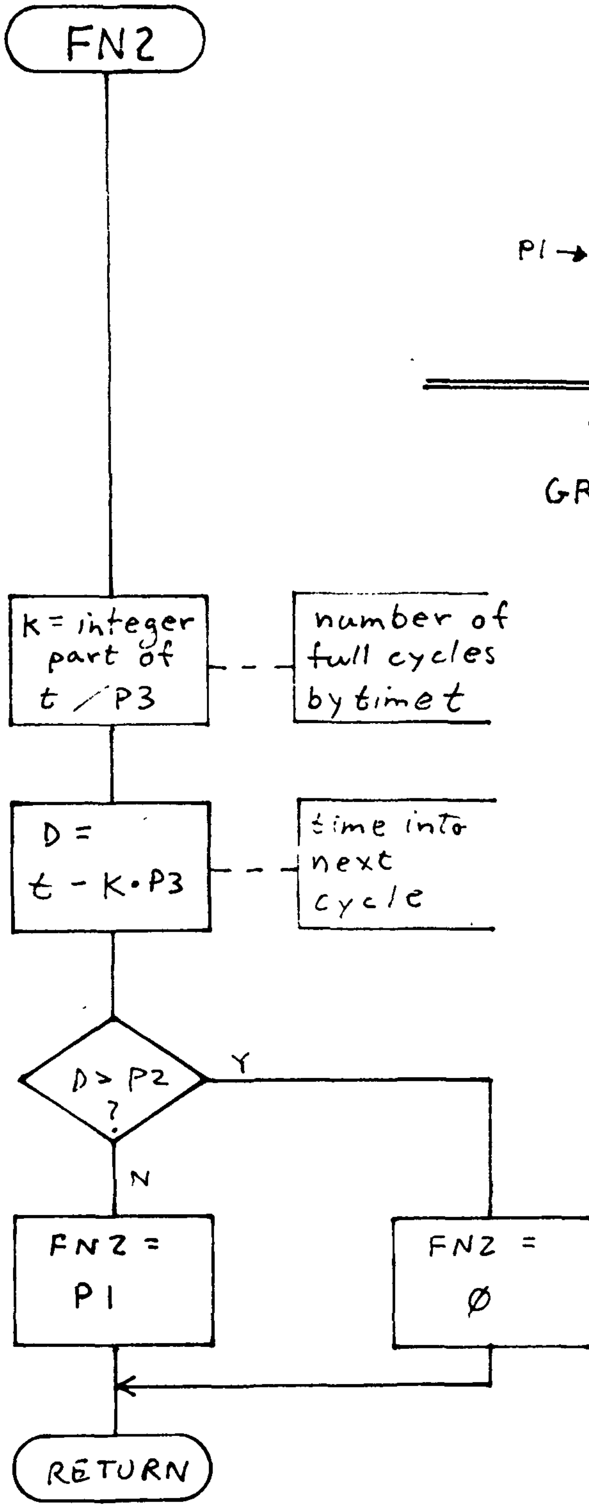
FLOW DIAGRAM

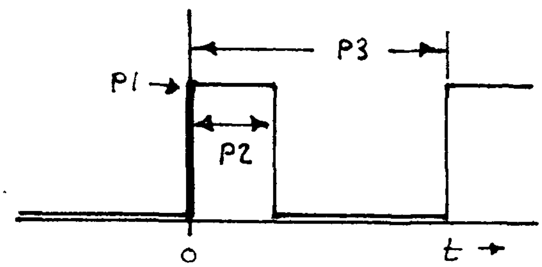
A-33



FLOW DIAGRAM

A-34

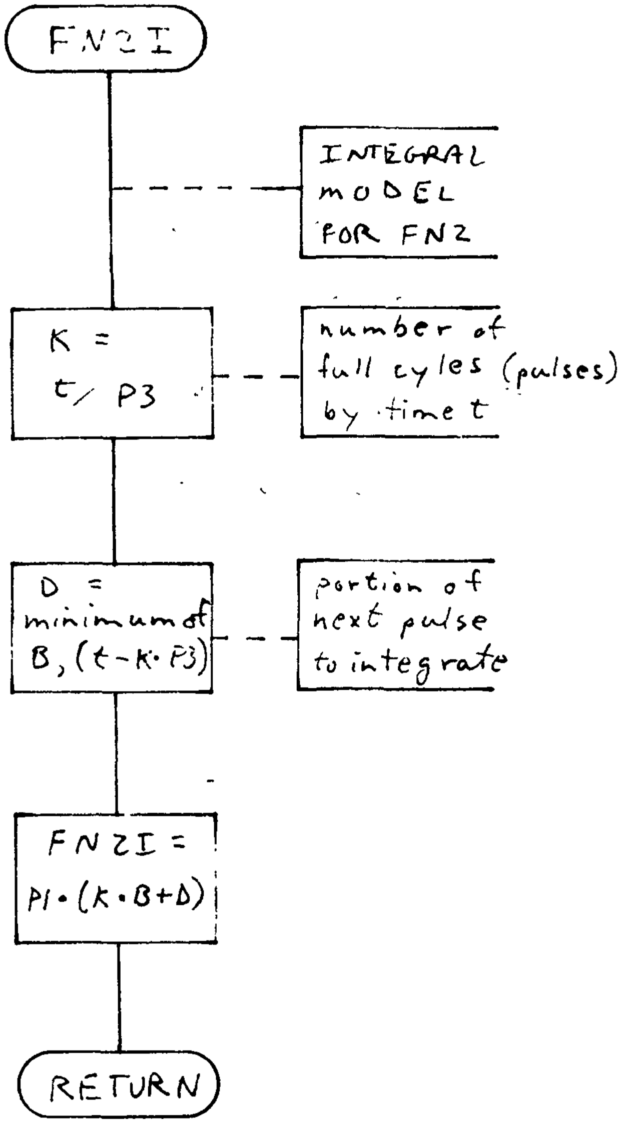




GRAPH OF FN2 (t)
(PULSE TRAIN)

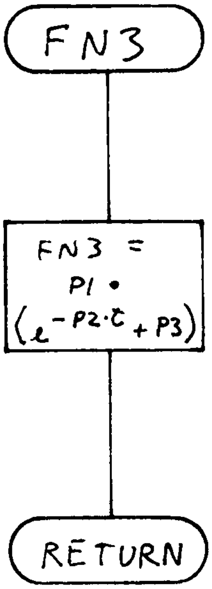
FLOW DIAGRAM

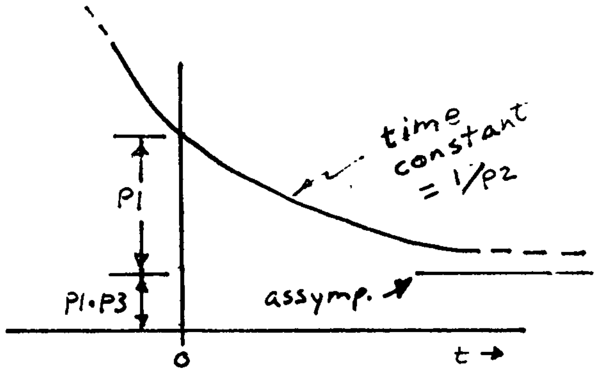
A-35



FLOW DIAGRAM

A-36

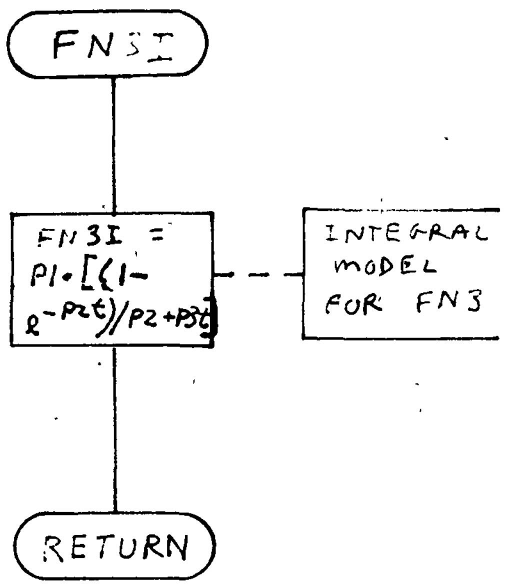




GRAPH OF FN3
(EXPONTIAL)

FLOW DIAGRAM

A-37



APPENDIX B

PROGRAM SOURCE LISTING

```
      PROGRAM CARE2(INPUT=201,OUTPUT=201,TAPE5=INPUT,TAPE6=OUTPUT)       MAINLOG    1
C                                                                         TITLE      1
C     ***********************************************************         TITLE      2
C     *       THIS VERSION OF CARE IS ENTITLED CARE2, AND          *      TITLE      3
C     *  WAS PREPARED FOR THE LANGLEY RESEARCH CENTER BY           *      TITLE      4
C     *  THE EQUIPMENT DIVISION OF RAYTHEON COMPANY.               *      TITLE      5
C     *  S.J. BAVUSO OF THE CENTER SERVED AS CONTRACT MONITOR.     *      TITLE      6
C     *                                                            *      TITLE      7
C     *       CARE WAS ORIGINALLY DEVELOPED BY F.P. MATHUR         *      TITLE      8
C     *  OF THE JET PROPULSION LABORATORY, AND MORE RECENTLY       *      TITLE      9
C     *  WAS UPDATED AND MODIFIED FOR USE AT THE CENTER BY         *      TITLE     10
C     *  S.J. BAVUSO.                                              *      TITLE     11
C     *                                                            *      TITLE     12
C     *       CARE2 EXTENDS THE ORIGINAL PROGRAM#S CAPABILITIES    *      TITLE     13
C     *  IN TWO PRIMARY AREAS                                      *      TITLE     14
C     *                                                            *      TITLE     15
C     *     - ADDITION OF A COVERAGE CALCULATION CAPABILITY        *      TITLE     16
C     *                                                            *      TITLE     17
C     *     - ADDITION OF A DUAL MODE MODEL (EQUATION 7)           *      TITLE     18
C     *                                                            *      TITLE     19
C     *       THE PROGRAM IS IMPLEMENTED IN CDC FORTRAN 2.1        *      TITLE     20
C     *  FOR EXECUTION UNDER SCOPE 3.0                             *      TITLE     21
C     ***********************************************************         TITLE     22
C                                                                         TITLE     23
      REAL LAMT,MIN,MAX                                                   CARE      30
      REAL MUT(10),MTF(16),FOFX(16)                                       CARE      32
      REAL R(121,10),PDT(121),SIMPX(121),DI,SIMRF                         CRUSH      1
      REAL DIFF(1,1),RIF(1,1),GAIN(1,1),ABSC(1,1)                         CRUSH      2
      REAL G(1,1),XY(1,4),RLRV(1,2),FDUM(1),ENC(20)                       CRUSH      3
      REAL TMAX(101,15)                                                   CARE      36
      DIMENSION FRAC(8),PCW(22),NAMES(22),RLDFS(10),INTDFS(4),UNITR(8)    MAINLOG    2
      DIMENSION QQ(10,2),Q(10),Q1(10),Q2(10),N(10)                        PARMOD     1
      REAL LAM(10),MU(10),K(10),RV(10),P(10)                              PARMOD     2
      DIMENSION CC(10,2),C(10),C1(10),C2(10),CTR(10)                      PARMOD     3
      DIMENSION CCD(10,2),CD(10),CD1(10),CD2(10),CDTR(10)                 PARMOD     4
      DIMENSION GGM(10,2),GM(10),GM1(10),GM2(10),GMP(10)                  PARMOD     5
      DIMENSION PPRC(10,2),PRC(10),PRC1(10),PRC2(10),PARAM(16,10)         PARMOD     6
      INTEGER S(10),W(10),Z(10),ANSWER(3)                                 PARMOD     7
      INTEGER IDUM(11),PROD(10 ),PAR(6),OPTION,R2OPT                      CARE      38
      INTEGER IGENC(10),IGENP(10),IFSC(8),IDQ(10)                         MAINLOG    3
      LOGICAL  PRODT,TIN,MUTIN,XYRANGE,LPLOT,GPLOT,NOPDT                  CARE      39
      LOGICAL LVARY,DEBUG,INPOUT,STGOUT,COVINT,LSTCH,COVPRC,DMFLG         MAINLOG    4
      LOGICAL BCOMP,ELATIN,RVPLOT,TRUNC,OPTR1,OPTR2,PLOTR1,PLOTR2         CARE      40
      LOGICAL BCOMPS(10),RSGN                                             PARMOD     8
      LOGICAL OUTPT1,OUTPT2,OUTPT3,OUTPT4,OUTPT5,RVLOCS                   CARE      41
      EQUIVALENCE (XY(1,1),ABSC(1,1)), (XY(1,4),G(1,1))                   CARE      42
      EQUIVALENCE (QQ(1,1),Q(1),Q1(1)),(QQ(1,2),Q2(1))                    PARMOD     9
      EQUIVALENCE (CC(1,1),C(1),C1(1)),(CC(1,2),C2(1))                    PARMOD    10
      EQUIVALENCE (CCD(1,1),CD(1),CD1(1)),(CCD(1,2),CD2(1))               PARMOD    11
      EQUIVALENCE (GGM(1,1),GM(1),GM1(1)),(GGM(1,2),GM2(1))               PARMOD    12
      EQUIVALENCE (PPRC(1,1),PRC(1),PRC1(1)),(PPRC(1,2),PRC2(1))          PARMOD    13
      COMMON /PLUTIN/ TIN,MIN,MAX,STEP,DIFF,RIF,GAIN,MTF,FOFX,            CARE      43
     *  PDT,SIMPX,XMIN,XMAX,YMIN,YMAX,XYRANGE,IA,IAA,IB,IBB,              CARE      44
     *  IC,ICC,IPLT1,IPLT2,IPLT3,IPLT4,IPLT5                              CARE      45
      COMMON/GRAPH/ NEQ,PROD,NPROD,PAR,ENC,AB,PRODT                       CARE      46
      COMMON /PARA1/QQ,IDQ,N,S                                            PARMOD    14
      COMMON /PARA2/LAM,MU,K,BCOMPS,SLH2,SLH3                             FINAL      1
      COMMON /PARA3/RV,Z,W,P                                              PARMOD    16
      COMMON /PARA4/CC,CCD,CTR,CDTR,CCSF                                  PARMOD    17
      COMMON /PARA5/GGM,GMP,PPRC                                          PARMOD    18
      COMMON/GRAFF/RLRV,PSYMBL(16)                                        CARE      48
      COMMON /PARM/ PARAM,N10,K1                                          CARE      49
      COMMON /OUTPT/ OUTPT1,OUTPT2,OUTPT3,OUTPT4,OUTPT5                   CARE      50
```

```
      COMMON /DRIVE/CLT,DMSTEP,HSTEP,NUN,RSGN,JMAX,I10,SUMLAM            FINAL      2
      COMMON /DEFS/PCW,NAMES,RLDFS,INTDFS                                MAINLOG    6
      COMMON /CVBG/COVINT,IGENC,IGENP,IFSC,FRAC                          MAINLOG    7
      COMMON IDUM,FDUM,R                                                 CARE      51
      DATA IRES,LVARY/0,.F./                                             POSTPCH    1
      DATA ITEST1/1/,ITEST2/2/                                           CARE      52
      DATA JDIM/121/                                                     CARE      53
      DATA NOPDT/.FALSE./,NOGP/1/                                        CARE      54
      DATA PSYMBL/1HA,1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,1HJ,1HK,           CARE      55
     *1HL,1HM,1HN,1HO,1HP/                                               CARE      56
      DATA NAMES/4HQ=Q1,2HQ2,1HN,1HS,1HW,1HZ,3HLAM,2HMU,3HGMP,4HC=C1,    MAINLOG    8
     *      3HCD1,2HC2,3HCD2,3HCTR,4HCDTR,4HPRC1,4HPRC2,2HRV,1HP,        MAINLOG    9
     *      1HK,3HGM1,3HGM2/                                             MAINLOG   10
      DATA PCW/54200000000000000001B,40200000000000000002B,              MAINLOG   11
     *20000000000000000001B,36200000000000000002B,37400000000000000003B,MAINLOG   12
     *37400000000000000004B,77600000000000000003B,76200000000000000004B,MAINLOG   13
     *40200000000000000005B,54200000000000000006B,40200000000000000006B,MAINLOG   14
     *40200000000000000006B,40200000000000000006B,40200000000000000007B,MAINLOG   15
     *40200000000000000007B,40200000000000000010B,40200000000000000010B,MAINLOG   16
     *77000000000000000011B,41000000000000000012B,76200000000000000000B,MAINLOG   17
     *40200000000000000000B,40200000000000000000B/                       MAINLOG   18
      DATA SLM2,SLM3,CCSF,RSGN/2*1.0E-8,1.0,.T./                         MAINLOG   19
      DATA STAGE/5HSTAGE/                                                MAINLOG   20
C                                                                        CARE      57
C        * READ IN DATA FOR RELIABILITY EQUATIONS *                      CARE      58
      CALL READIN( T,LAMT,MUT,B,OPTION,NPROD,PROD,PRODT,MUTIN,           CARE      59
     *  NEQ,RVPLOT,PMIN,PMAX,PSTEP ,ELAMT,ELATIN,LPLOT,                  CARE      60
     *  DEBUG,INPOUT,STGOUT,LSTCH,COVPRC,DMFLG,NUN,                      MAINLOG   21
     *  R2,R1MIN,R1MAX,R1STEP,OPTR1,OPTR2,PLOTR1,PLOTR2,R2OPT)           CARE      61
      DO 10  I=1,10                                                      MAINLOG   22
      Q1(I)=RLDFS(1)                                                     MAINLOG   23
      Q2(I)=RLDFS(2)                                                     MAINLOG   24
      LAM(I)=RLDFS(3)                                                    MAINLOG   25
      MU(I)=RLDFS(4)                                                     MAINLOG   26
      GMP(I)=RLDFS(5)                                                    MAINLOG   27
      C1(I)=RLDFS(6)                                                     MAINLOG   28
      CD1(I)=C1(I)                                                       MAINLOG   29
      C2(I)=C1(I)                                                        MAINLOG   30
      CD2(I)=C1(I)                                                       MAINLOG   31
      CTR(I)=RLDFS(7)                                                    MAINLOG   32
      CDTR(I)=CTR(I)                                                     MAINLOG   33
      PRC1(I)=RLDFS(8)                                                   MAINLOG   34
      PRC2(I)=PRC1(I)                                                    MAINLOG   35
      RV(I)=RLDFS(9)                                                     MAINLOG   36
      P(I)=RLDFS(10)                                                     MAINLOG   37
      N(I)=INTDFS(1)                                                     MAINLOG   38
      S(I)=INTDFS(2)                                                     MAINLOG   39
      W(I)=INTDFS(3)                                                     MAINLOG   40
   10 Z(I)=INTDFS(4)                                                     MAINLOG   41
      NRS=0                                                              MAINLOG   42
      N10=1                                                              FINAL      3
      MAX=LAMT                                                           CARE      75
      IF (TIN) MAX=T                                                     CARE      76
      IF (ELATIN) MAX=ELAMT                                              CARE      77
      DMSTEP=STEP                                                        MAINLOG   43
      HSTEP=STEP/2.0                                                     MAINLOG   44
      JMAX=(MAX-MIN)/STEP                                                MAINLOG   45
      IF(JMAX.LE.JDIM)  GO TO 30                                         MAINLOG   46
      JMAX=JDIM                                                          MAINLOG   47
      MAX=JMAX*STEP+MIN                                                  MAINLOG   48
   30 JMAX1=JMAX+1                                                       MAINLOG   49
C                                                                        MAINLOG   50
C        *  READ IN DATA FOR THE CURRENT RUN SET -   *                   MAINLOG   51
C        *  PERFORM GENERAL DATA MAINTENANCE         *                   MAINLOG   52
C                                                                        MAINLOG   53
   40 NRS=NRS+1                                                          MAINLOG   54
```

```
      CALL READIN2(NRS,DMFLG,IRES,LVARY,DVAL,DEBUG)                    MAINLOG    55
C                                                                      FINAL       4
C          CALCULATE ANY REQUESTED COVERAGE FACTORS, TO BE USED        FINAL       5
C          AS INPUT FOR CURRENT RUN-SET                                FINAL       6
C                                                                      FINAL       7
      IF(COVPRC)  CALL COVGEN(NRS,NPROD)                               MAINLOG    56
      IF(.NOT.(LVARY.AND.LSTCH))  GO TO 70                             MAINLOG    57
      DO 60  K1=1,NPROD                                                MAINLOG    58
      XPAR=DVAL                                                        MAINLOG    59
      DO 60  I10=1,N10                                                 MAINLOG    60
      IF(PARAM(I10,K1).EQ.DVAL.AND.XPAR.NE.DVAL)                       MAINLOG    61
     *     PARAM(I10,K1)=XPAR                                          MAINLOG    62
   60 XPAR=PARAM(I10,K1)                                               MAINLOG    63
   70 IF(INPOUT)  CALL RITERS(NRS,DMFLG,IRES,LVARY,DVAL,DEBUG)         MAINLOG    64
      IF(DEBUG)  GO TO 40                                              MAINLOG    65
      IF(.NOT.DMFLG)  GO TO 85                                         MAINLOG    66
C                                                                      FINAL       8
C          CHECK THAT RESTRICTIONS ON INDEPENDENT VARIABLE             FINAL       9
C          ARE MET IF DUAL MODE MODEL WILL BE USED                     FINAL      10
C                                                                      FINAL      11
      IF(TIN)  GO TO 80                                                MAINLOG    67
      WRITE(6,3000)                                                    MAINLOG    68
      STOP                                                             MAINLOG    69
   80 IF(MIN.EQ.0.0)  GO TO 85                                         MAINLOG    70
      WRITE(6,3001)                                                    MAINLOG    71
      STOP                                                             MAINLOG    72
   85 NG=0                                                             MAINLOG    73
      NJ=0                                                             CARE      137
C                                                                      FINAL      12
C          DO LOOP FOR VARIATION OF PARAMETER OF INTEREST              FINAL      13
C          BEGINS.  FOR EACH RUN, THE PARAMETER VECTOR IS              FINAL      14
C          REPLACED BY ONE ROW OF PARAM(16,10)                         FINAL      15
C                                                                      FINAL      16
      DO 380 I10=1,N10                                                 CARE      138
      IF(LVARY)  CALL PGET(IRES,I10,NPROD)                             MAINLOG    74
      SUMLAM=0.0                                                       MAINLOG    75
C                                                                      FINAL      17
C          COMPUTE ANY REQUIRED DEPENDENT PARAMETERS FOR               FINAL      18
C          EQUATIONS 1 THROUGH 7                                       FINAL      19
C                                                                      FINAL      20
      DO 105  K1=1,NPROD                                               MAINLOG    76
      BCOMPS(K1)=MU(K1).LE.0.0                                         MAINLOG    77
      K(K1)=100000.                                                    MAINLOG    78
      IF(BCOMPS(K1))  GO TO 101                                        MAINLOG    79
      K(K1)=LAM(K1)/MU(K1)                                             MAINLOG    80
      IF(K(K1).GE.100000.)  BCOMPS(K1)=.T.                             MAINLOG    81
  101 IF(K1.GT.NUN)  GO TO 105                                         MAINLOG    82
      IDQ(K1)=Q1(K1)-Q2(K1)                                            MAINLOG    83
      IF(IDQ(K1).GE.0.OR.RSGN)  GO TO 102                              MAINLOG    84
      RSGN=.T.                                                         MAINLOG    85
      WRITE(6,3004)K1                                                  MAINLOG    86
  102 SUMLAM=SUMLAM+LAM(K1)                                            MAINLOG    87
      DO 103  M=1,2                                                    MAINLOG    88
  103 GGM(K1,M)=GMP(K1)*(1.0-PPRC(K1,M))                               MAINLOG    89
  105 CONTINUE                                                         MAINLOG    90
      WRITE(6,3005)I10                                                 MAINLOG    91
      DO 120 J=1,JDIM                                                  CARE      139
  120 PDT(J)=1.0                                                       CARE      140
C                                                                      FINAL      21
C          DO LOOP FOR PRODUCT OF SYSTEM RELIABILITIES V.S.            FINAL      22
C          TIME BEGINS.  DUAL MODE RELIABILITY IS OUTPUT AS            FINAL      23
C          THAT OF THE LAST STAGE OF THAT MODEL.                       FINAL      24
      DO 320 K1=1,NPROD                                                CARE      141
      IF (PRODT) NEQ=PROD(K1)                                          CARE      142
      BCOMP=BCOMPS(K1)                                                 MAINLOG    92
      IF(.NOT.DMFLG)  GO TO 190                                        MAINLOG    93
```

```
      IF(K1.LT.NUN)  GO TO 320                                      MAINLOG    94
      IF(K1.GT.NUN)  GO TO 190                                      MAINLOG    95
C                                                                   MAINLOG    96
C          PREPARE TABLE OF RELIABILITY VALUES                      MAINLOG    97
C          FOR THE DUAL MODE MODEL AND PRINT                        MAINLOG    98
C          MODE 1 REL FOR STAGES (OPTION)                           MAINLOG    99
C                                                                   MAINLOG   100
      IF(STGOUT)  WRITE(6,3002)((STAGE,I),I=1,NUN)                  MAINLOG   101
      DO 150  J=1,JMAX1                                             MAINLOG   102
      CLT=(J-1)*DMSTEP                                              MAINLOG   103
      CALL DCR(J,UNITR,RELM1,DCREL)                                 MAINLOG   104
      IF(STGOUT)  WRITE(6,3005)CLT,RELM1,(UNITR(I),I=1,NUN)         MAINLOG   105
  150 R(J,I10)=DCREL                                                MAINLOG   106
  190 RVLOC8=RVPLOT                                                 MAINLOG   107
      WRITE(6,5006)K1,NEQ                                           MAINLOG   108
      IF (.NOT.(BCOMP.AND.ELATIN)) GO TO 194                        CARE      200
      WRITE (6,1003)                                                CARE      201
      CALL EXIT                                                     CARE      202
  194 IF (.NOT.(BCOMP.AND.RVPLOT)) GO TO 198                        CARE      203
      RVLOC8=.FALSE.                                                CARE      204
      WRITE (6,1005) NEQ                                            CARE      205
  198 IF(BCOMP.OR.K(K1).GE.0.0)  GO TO 200                          PARMOD     19
      WRITE(6,1050)K(K1)                                            PARMOD     20
      GO TO 40                                                      MAINLOG   109
C                                                                   CARE      209
C          PREPARE TABLE OF RELIABILITY VALUES                      MAINLOG   110
C          FOR ONE EQUATION (NO,S 1 - 6)                            MAINLOG   111
C                                                                   CARE      214
  200 IF(ELATIN)  GO TO 204                                         MAINLOG   112
      IF(TIN)  GO TO 201                                            MAINLOG   113
      WRITE(6,5007)                                                 MAINLOG   114
      GO TO 210                                                     MAINLOG   115
  201 WRITE(6,5008)                                                 MAINLOG   116
      GO TO 210                                                     MAINLOG   117
  204 WRITE(6,5009)                                                 MAINLOG   118
  210 DO 315  J=1,JMAX1                                             MAINLOG   119
      CLT=(J-1)*STEP+MIN                                            MAINLOG   120
      IF(NEQ.NE.7)  GO TO 211                                       MAINLOG   121
      RDP=R(J,I10)                                                  MAINLOG   122
      TLAMT=CLT*SUMLAM                                              MAINLOG   123
      GO TO 295                                                     MAINLOG   124
  211 IF(.NOT.ELATIN)  GO TO 212                                    MAINLOG   125
      RL=CLT                                                        CARE      217
      IF (RL) 206,206,207                                           CARE      218
  206 RS=0.0                                                        CARE      219
      GO TO 209                                                     CARE      220
  207 IF(RL .GT. .99999999)GO TO 208                                CARE      221
      IF(K(K1).GT.0.0)  RS=RL**(1.0/K(K1))                          PARMOD     21
      GO TO 209                                                     CARE      223
  208 RS=1.0                                                        CARE      224
  209 TLAMT=1.0                                                     CARE      225
      GO TO 228                                                     CARE      226
  212 TLAMT=CLT                                                     CARE      227
      IF (.NOT.TIN) GO TO 215                                       CARE      228
      TLAMT=LAM(K1)*CLT                                             PARMOD     22
  215 RL= EXP(-TLAMT)                                               CARE      230
      IF(K(K1).GT.0.0)  RS=RL**(1.0/K(K1))                          PARMOD     23
C                                                                   CARE      232
C         COMPUTE RELIABILITY OF APPROPRIATE EQUATION               CARE      233
  228 CALL RELEQS(NEQ,BCOMP,RL,RS,S(K1),N(K1),K(K1),Q(K1),          PARMOD     24
     *     C(K1),RV(K1),Z(K1),W(K1),P(K1),TLAMT,RDP)                PARMOD     25
C                                                                   CARE      236
  290 R(J,I10)=RDP                                                  CARE      237
C                                                                   CARE      238
C     *** COMPUTE UNRELIABILITY, SIMPLE RELIABILITY, ***            CARE      239
C     *** SIMPLE RIF, AND SIMPLE GAIN                ***            CARE      240
```

```
  295 CALL SIMPLE(ELATIN,TLAMT,RL,SIMPX(J),SIMRF,SG,QI,RDP)               MAINLOG   126
C                                                                        CARE      242
      IF (.NOT.OUTPT1) GO TO 310                                         CARE      243
C                                                                        CARE      244
C         WRITE RELIABILITY RESULTS                                      CARE      245
      IF (ELATIN) GO TO 304                                              CARE      246
      IF (TIN) GO TO 300                                                 CARE      247
      WRITE(6,1007)TLAMT,R(J,I10),QI,SIMPX(J),SG,SIMRF                   Q1TOQI      2
      GO TO 310                                                          CARE      249
  300 WRITE(6,1007)CLT,R(J,I10),QI,SIMPX(J),SG,SIMRF                     Q1TOQI      3
C                                                                        CARE      251
      GO TO 310                                                          CARE      252
  304 WRITE(6,1008)RL,RS,R(J,I10),GI,SG,SIMRF                            Q1TOQI      4
C                                                                        CARE      254
C     *** COMPUTE ABSCISSA VALUES FOR PLOTS ***                          CARE      255
  310 ABSC(J,1)=CLT                                                      CARE      256
      ABSC(J,2)=TLAMT                                                    CARE      257
      ABSC(J,3)=EXP(-TLAMT)                                              CARE      258
C                                                                        CARE      259
C     *** COMPUTE PRODUCT OF RELIABILITIES ***                           CARE      260
      PDT(J)=PDT(J)*R(J,I10)                                             CARE      261
C                                                                        CARE      262
  315 CONTINUE                                                           MAINLOG   127
      NJ=NJ+1                                                            CARE      265
      IF(ELATIN) GO TO 318                                               CARE      266
      IF(.NOT.OUTPT4)  GO TO 318                                         MAINLOG   128
C                                                                        CARE      267
C       * CALL INTEGRATION ROUTINE TO COMPUTE *                          CARE      268
C       * MTF AND THE RELIABILITY AT MTF      *                          CARE      269
      CALL INTEGR(TIN,BCOMP,LAM(K1),MU(K1),S(K1),N(K1),K(K1),            PARMOD     26
     *     Q(K1),C(K1),RV(K1),Z(K1),W(K1),P(K1),NEQ,B,MTF(NJ),FOFX(NJ))  PARMOD     27
C                                                                        CARE      273
C         COMPUTE TMAX, SIMTMX, AND SIMTIP                               CARE      274
  318 IF(OPTR1.OR.OPTR2)  CALL SIMPR1(TIN,BCOMP,LAM(K1),S(K1),N(K1),     PARMOD     28
     *     K(K1),Q(K1),C(K1),RV(K1),Z(K1),W(K1),P(K1),NEQ,R2,R1MIN,      PARMOD     29
     *     R1MAX,R1STEP,PLOTR1,OPTR1,I10,TMAX)                           PARMOD     30
C                                                                        CARE      278
C                                                                        CARE      279
      CLT=MIN                                                            CARE      280
      ELATIN = .FALSE.                                                   CARE      281
  320 CONTINUE                                                           CARE      282
C                                                                        CARE      283
C         END OF DO LOOP FOR PRODUCT OF RELIABILITY                      CARE      284
C                                                                        CARE      285
      IF (.NOT.(PRODT.AND.OUTPT1)) GO TO 360                             CARE      286
C                                                                        CARE      287
C       * WRITE TABLE OF VALUES FOR THE *                                CARE      288
C       * PRODUCT OF RELIABILITIES      *                                CARE      289
      IF (TIN) GO TO 330                                                 CARE      290
      WRITE (6,2006)                                                     CARE      291
      GO TO 340                                                          CARE      292
  330 WRITE (6,2007)                                                     CARE      293
  340 CT=MIN                                                             CARE      294
      JP=1                                                               CARE      295
  350 WRITE (6,2008) CT,PDT(JP)                                          CARE      296
      JP=JP+1                                                            CARE      297
      CT=CT+STEP                                                         CARE      298
      IF (CT.LE.MAX) GO TO 350                                           CARE      299
C                                                                        CARE      300
  360 IF(RVLOCS)  CALL EQUAL(BCOMP,MU(K1),S(K1),N(K1),K(K1),Q(K1),       PARMOD     31
     *     C(K1),Z(K1),W(K1),P(K1),NEQ,I10,RLRV,DELRL)                   PARMOD     32
      IF (I10.EQ.1) GO TO 370                                            CARE      303
      IF (R2OPT.EQ.1.AND.I10.LT.N10) GO TO 364                           CARE      304
C                                                                        CARE      305
C         COMPUTE TMAX1, TMAX2, AND RATIF                                CARE      306
      IF (OPTR2) CALL PARAR1 (R2,R1MIN,R1MAX,R1STEP,PLOTR2,              CARE      307
```

```
     *  R2UPT,I10,TMAX)                                                CARE     308
C                                                                       CARE     309
  364 IF (OPTION.EQ.1.AND.I10.LT.N10) GO TO 370                        CARE     310
      IF(PRODT) GO TO 370                                               CARE     311
C                                                                       CARE     312
C     *** COMPUTE DIFF, RIF, AND GAIN ***                               CARE     313
      IF (OPTION.NE.2) NG=0                                             CARE     314
      CALL RIFDIF (I10,OPTION,IRES,N10,TIN,MIN,MAX,STEP,JDIM,R,         CARE     315
     *  DIFF,RIF,GAIN,NG,LPLOT,GPLOT)                                   CARE     316
      IF (IPLT3.NE.1.OR.OPTION.EQ.2) GO TO 370                          CARE     317
      IF (GPLOT) CALL PLOTG (NG,ABSC,XY,G,3,PARAM,K1,2)                 CARE     318
  370 IF( IPLT1 .EQ. 1) CALL PLOTPD(1,ABSC,XY,G,1,PARAM,K1,3)           CARE     319
  380 CONTINUE                                                          CARE     320
C                                                                       CARE     321
C          END OF DO LOOP FOR FAMILY OF PARAMETER                       CARE     322
C                                                                       CARE     323
      IF(LVARY)  CALL PGET(IRES,1,NPROD)                                MAINLOG  129
      IF (PRODT) GO TO 390                                              CARE     324
      IF (IPLT2.EQ.1.OR.IPLT5.EQ.1) CALL PLOTR (N10,ABSC,XY,G,3,PARAM,  CARE     325
     *  K1,1)                                                           CARE     326
      IF (IPLT3.EQ.1.AND.OPTION.EQ.2) CALL PLOTG (NG,ABSC,XY,G,3,PARAM, CARE     327
     *  K1,2)                                                           CARE     328
      IF (IPLT4.EQ.1) CALL PLOTPD (N10,ABSC,XY,G,3,PARAM,K1,3)          CARE     329
  390 IF(RVPLOT) CALL PLOTRV(I10,DELRL)                                 CARE     330
      GO TO 40                                                          CARE     342
 9999 STOP                                                              CARE     343
C                                                                       CARE     344
C             FORMAT STATEMENTS                                         CARE     345
C                                                                       CARE     346
 1001 FORMAT (6A1)                                                      CARE     347
 1002 FORMAT (80A1)                                                     CARE     348
 1003 FORMAT(* T OR LAMT MUST BE GIVEN FOR THE EQUATION YOU SPECIFIED*, CARE     349
     * /,* ENTIRE  DATA CASE IS TERMINATED*)                            CARE     350
 1004 FORMAT( * THE VARIABLE THAT IS TO BE USED*,/,                     CARE     351
     ** FOR THE FAMILY OF PARAMETERS IS= *,33X, * PAR= *,6A1)           CARE     352
 1005 FORMAT (* RV VERSUS R PLOT NOT TO BE PERFORMED FOR THIS CASE*,/,  CARE     353
     *  * SINCE CALCULATIONS ARE FOR THE B PART OF EQUATION *,I2)       CARE     354
 1006 FORMAT(* INCORRECT PARAMETER FOR EQUATION *,I2,* TRY ANOTHER*)    CARE     355
 1007 FORMAT(1H ,F10.3,3F12.7,2E18.7)                                   MAINLOG  130
 1008 FORMAT(( 1H ,F7.5,2X,E14.5,2(2X,F10.7),2(2X,E16.7))               CARE     357
 1028 FORMAT(* DO YOU WISH TO SPECIFY ANOTHER PARAMETER*,/,             CARE     358
     ** ANSWER YES OR NO*)                                              CARE     359
 1029 FORMAT(* TYPE IN NEW PARAMETER OR PARAMETERS (FOR PRODUCT OF*,/,  CARE     360
     ** RELIABILITIES) AS BEFORE*)                                      CARE     361
 1030 FORMAT(* IMPROPER VALUE FOR K - K = *,E14.6,/,                    CARE     362
     ** K MUST SATISFY 0.LE.K.LE.INFINITY*,/,                           CARE     363
     ** CALCULATIONS ARE NOT PERFORMED FOR THIS K VALUE*)               CARE     364
 1032 FORMAT (1H ,80A1)                                                 CARE     365
 2006 FORMAT(//6X,*LAMT*,3X,*REL PRODUCT*)                              POSTPCH    2
 2007 FORMAT(//8X,*T*,4X,*REL PRODUCT*)                                 POSTPCH    3
 2008 FORMAT(1X,F10.3,F12.7)                                            POSTPCH    4
 3000 FORMAT(10X,*FOR DUAL MODE MODEL, INDEP. VARIABLE MUST BE TIME*/   MAINLOG  131
     *       25X,*JOB ABORTED*)                                         MAINLOG  132
 3001 FORMAT(10X,*FOR DUAL MODE MODEL, TIME MUST START AT 0.0*/         MAINLOG  133
     *       25X,*JOB ABORTED*)                                         MAINLOG  134
 3002 FORMAT(//43X,*DUAL MODE SYSTEM RELIABILITY IN MODE 1*//4X,        MAINLOG  135
     *       *TIME*,6X,*SYSTEM REL.  *,8(A5,I3,4X))                     MAINLOG  136
 3003 FORMAT(1X,F10.3,9F12.7)                                           MAINLOG  137
 3004 FORMAT(/25X,*= WARNING =*/10X,*Q2 GREATER THAN Q1 *,              MAINLOG  138
     *       *FOR STAGE*,I3,* - RSGN FORCED TO TRUE*)                   MAINLOG  139
 3005 FORMAT(1H1,49X,*RUN*,I5,* RELIABILITY RESULTS*/                   MAINLOG  140
     *       50X,*----------------------------*//)                      MAINLOG  141
 3006 FORMAT(//49X,*STAGE*,I3,* EQUATION NUMBER*,I2/)                   MAINLOG  142
 3007 FORMAT(6X,*LAMT*,7X,*REL*,8X,*UNREL*,7X,*SIMREL*,9X,              MAINLOG  143
     *       *SIMGAIN*,9X,*SIMRIF*)                                     MAINLOG  144
 3008 FORMAT(8X,*T*,8X,*REL*,8X,*UNREL*,7X,*SIMREL*,9X,                 MAINLOG  145
```

```
     *     *SIMGAIN*,9X,*SIMRIF*)                                       MAINLOG   146
 3009 FORMAT(2X,*ELAMT*,6X,*EMUT*,9X,*REL*,8X,*UNREL*,9X,               MAINLOG   147
     *     *SIMGAIN*,10X,*SIMRIF*)                                      MAINLOG   148
C                                                                       CARE      369
C                                                                       CARE      370
      END                                                               CARE      371
      SUBROUTINE BISECT( NEQ,BCOMP,S,N,K,Q,C,RV,Z,W,P,FIRST,LAMT,R)     BISECT      1
      INTEGER NEQ,S,N,Z,W,ITER,ITMAX                                    BISECT      2
      LOGICAL BCOMP,FIRST                                               BISECT      3
C                                                                       BISECT      4
      REAL K,LAMT,INF                                                   BISECT      5
C       ******************************                                  BISECT      6
C       *                            *                                  BISECT      7
C       * THIS SUBROUTINE COMPUTES   *                                  BISECT      8
C       * LAMT FOR GIVEN RELIABILITY *                                  BISECT      9
C       * USING REGULA FALSI METHOD  *                                  BISECT     10
C       *                            *                                  BISECT     11
C       ******************************                                  BISECT     12
C                                                                       BISECT     13
      DATA TWO/2.0  /, INF/1.0E35/, TOL/1.0E-4/, ITMAX/40/              BISECT     14
      ITER=1                                                            BISECT     15
      IF (R) 5,10,30                                                    BISECT     16
    5 WRITE (6,1005)                                                    BISECT     17
      WRITE (6,1004)                                                    BISECT     18
      RETURN                                                            BISECT     19
   10 IF (FIRST) GO TO 20                                               BISECT     20
      LAMT=INF                                                          BISECT     21
      RETURN                                                            BISECT     22
   20 WRITE (6,1001)                                                    BISECT     23
      WRITE (6,1004)                                                    BISECT     24
      RETURN                                                            BISECT     25
   30 IF (R-1.0E0) 60,50,40                                             BISECT     26
   40 WRITE (6,1002)                                                    BISECT     27
      WRITE (6,1004)                                                    BISECT     28
      RETURN                                                            BISECT     29
   50 LAMT=0.                                                           BISECT     30
      RETURN                                                            BISECT     31
   60 LAMT=10.0                                                         BISECT     32
   70 RL= EXP(-LAMT)                                                    BISECT     33
      IF(K .EQ. 0.0 ) K=1.0                                             BISECT     34
      RS=RL**(1.0/K)                                                    BISECT     35
C          OBTAIN RELIABILITY FOR UPPER BOUND FOR LAMT                  BISECT     36
      CALL RELEQS (NEQ,BCOMP,RL,RS,S,N,K,Q,C,RV,Z,W,P,LAMT,REL)         BISECT     37
C                                                                       BISECT     38
      DIF=REL-R                                                         BISECT     39
      IF ( ABS(DIF).LE.TOL) RETURN                                      BISECT     40
      IF (DIF) 80,90,100                                                BISECT     41
C                                                                       BISECT     42
C          UPPER BOUND IS ATTAINED FOR LAMT                             BISECT     43
   80 X2=LAMT                                                           BISECT     44
      Y2=DIF                                                            BISECT     45
      GO TO 105                                                         BISECT     46
C                                                                       BISECT     47
   90 RETURN                                                            BISECT     48
C                                                                       BISECT     49
  100 LAMT=TWO*LAMT                                                     BISECT     50
      GO TO 70                                                          BISECT     51
C                                                                       BISECT     52
C          LOWER BOUND FOR LAMT IS 0.0D0                                BISECT     53
  105 LAMT=0.0                                                          BISECT     54
      X1=LAMT                                                           BISECT     55
C                                                                       BISECT     56
      RL=EXP(-LAMT)                                                     BISECT     57
C          OBTAIN RELIABILITY FOR LOWER BOUND FOR LAMT                  BISECT     58
      CALL RELEQS (NEQ,BCOMP,RL,RS,S,N,K,Q,C,RV,Z,W,P,LAMT,REL)         BISECT     59
C                                                                       BISECT     60
```

```
      Y1=REL-R                                                          BISECT    61
C                                                                       BISECT    62
C           INTERPOLATE FOR NEW VALUE OF LAMT BY REGULA FALSI METHOD    BISECT    63
  110 LAMT=(X1*Y2-X2*Y1)/(Y2-Y1)                                        BISECT    64
      RL= EXP(-LAMT)                                                    BISECT    65
      IF (.NOT.BCOMP) RS=RL**(1.0E0/K)                                  BISECT    66
C                                                                       BISECT    67
C           OBTAIN RELIABILITY FOR NEW LAMT                             BISECT    68
      CALL RELEQS (NEQ,BCOMP,RL,RS,S,N,K,Q,C,RV,Z,W,P,LAMT,REL)         BISECT    69
      DIF=REL-R                                                         BISECT    70
C                                                                       BISECT    71
C           CONVERGENCE TEST                                            BISECT    72
      IF ( ABS(DIF).LE.TOL) RETURN                                      BISECT    73
C                                                                       BISECT    74
      ITER=ITER+1                                                       BISECT    75
C                                                                       BISECT    76
C           TEST FOR MAX NUMBER OF ITERATIONS EXCEEDED                  BISECT    77
      IF (ITER.LE.ITMAX) GO TO 120                                      BISECT    78
      WRITE(6,1003)                                                     BISECT    79
      RETURN                                                            BISECT    80
C                                                                       BISECT    81
  120 IF (DIF*Y2) 130,130,140                                           BISECT    82
  130 X1=X2                                                             BISECT    83
      Y1=Y2                                                             BISECT    84
      X2=LAMT                                                           BISECT    85
      Y2=DIF                                                            BISECT    86
      GO TO 110                                                         BISECT    87
  140 X2=LAMT                                                           BISECT    88
      Y2=DIF                                                            BISECT    89
      GO TO 110                                                         BISECT    90
C                                                                       BISECT    91
 1001 FORMAT(* REFERENCE POINT(R2 MUST BE GREATER THAN 0.0*)            BISECT    92
 1002 FORMAT(* R1 OR R2 MUST BE LESS THAN OR EQUAL TO 1.0*)             BISECT    93
 1003 FORMAT (* CONVERGENCE PROBLEM WHEN CALCULATING MAXIMUM MISSION*,  BISECT    94
     *  * TIME*)                                                        BISECT    95
 1004 FORMAT (* MAXIMUM MISSION TIME IS NOT CALCULATED FOR THIS CASE*)  BISECT    96
 1005 FORMAT (* R1 OR R2 CAN NOT BE NEGATIVE*)                          BISECT    97
C                                                                       BISECT    98
      END                                                               BISECT    99
      SUBROUTINE EQUAL (BCOMP,MU,S,N,K,Q,C,Z,W,P,NEQ,I10,RLRV,DELRL)    EQUAL      1
C       ************************************                            EQUAL      2
C       *                                  *                            EQUAL      3
C       * THIS SUBROUTINE CALCULATES THE   *                            EQUAL      4
C       * LOCUS OF RV SUCH THAT THE SYSTEM *                            EQUAL      5
C       * RELIABILITY EQUALS THE UNIT      *                            EQUAL      6
C       * RELIABILITY, R.                  *                            EQUAL      7
C       *                                  *                            EQUAL      8
C       ************************************                            EQUAL      9
      REAL RLRV(1,2)                                                    CRUSH      1
      REAL K,MU                                                         EQUAL     11
      INTEGER S,N,W,Z                                                   EQUAL     12
      LOGICAL BCOMP                                                     EQUAL     13
      DATA RVLAMT/1.0E0/                                                EQUAL     14
      DELRL=5.0E-3                                                      EQUAL     15
      RVT=1.0                                                           EQUAL     16
      RL=1.0E-10                                                        EQUAL     17
      J=1                                                               EQUAL     18
      JT=1                                                              EQUAL     19
      WRITE (6,1001)                                                    EQUAL     20
   20 IF (K.NE.0.0  ) RS=RL**(1.0  /K)                                  EQUAL     21
      CALL RELEQS (NEQ,BCOMP,RL,RS,S,N,K,Q,C,RVT,1,1,P,RVLAMT,EXPR)     EQUAL     22
      RLRV(J,1)=RL                                                      EQUAL     23
      RLRV(J,I10+1)=RL**(1.0/(W*Z))/EXPR                                EQUAL     24
      IF (JT.LE.20) GO TO 110                                           EQUAL     25
      WRITE (6,1002) RLRV(J,1),RLRV(J,I10+1)                            EQUAL     26
      JT=1                                                              EQUAL     27
```

```
  110 J=J+1
      JT=JT+1
      RL=RL+DELRL
      IF (RL.LT.1.00000001  ) GO TO 20
      RETURN
 1001 FORMAT (//,5X,*EXP(-LAMT)*,7X,*RV*)
 1002 FORMAT(5X,F8.5,5X,E14.5)
      END
      FUNCTION FFAC(K)
C       ****************************************
C       *                                      *
C       * THIS FUNCTION COMPUTES FACTORIALS.   *
C       *                                      *
C       ****************************************
C     THIS FUNCTION GIVEN PARAMETER K WILL COMPUTE K FACTORIAL.
C     TO USE THIS FUNCTION STATE FFAC(K).
      IF (K) 20,40,40
C
C          DEFINE FACTORIAL OF A NEGATIVE NUMBER TO BE 1.0D0
   20 FFAC=1.0
      RETURN
   40 IF (K-1) 60,60,70
   60 FFAC=1.0
      RETURN
   70 IF (K-12) 75,75,85
   75 IFAC=K
      L=K-1
      DO 80 I=1,L
      KF=K-I
      IFAC=IFAC*KF
   80 CONTINUE
      FFAC=IFAC
      RETURN
   85 FFAC=K
      L=K-1
      DO 90 I=1,L
      Y=I
      F=K-I
      FFAC=FFAC*F
   90 CONTINUE
      RETURN
      END
      FUNCTION FNCK(NFAC,KFAC)
C       **************************************************
C       *                                                *
C       * THIS FUNCTION COMPUTES BINOMIAL COEFFICIENTS.  *
C       *                                                *
C       **************************************************
C     THIS FUNCTION GIVEN TWO PARAMETERS N,K
C     WILL COMPUTE NCK I.E. NFACT/(N-K)FACT * KFACT
C     THIS IS EQUIV. TO  N(N-1)...(K+1)/(N-K)FACT.
C     TO USE THIS FUNCTION STATE FNCK(N,K).
      NMINK=NFAC-KFAC
      IF (KFAC) 20,30,60
   60 IF (NMINK) 20,30,40
   20 WRITE(6,100)
  100 FORMAT(22H ILLEGAL COMBINATORIAL )
      GO TO 50
   30 FNCK=1.
      GO TO 50
   40 Y1=KFAC+1
      Z1=1.
      FNCK=1.
      DO 10 I=1,NMINK
      FNCK=FNCK*(Y1/Z1)
      Y1=Y1+1.
```

```
      Z1=Z1+1.
   10 CONTINUE
   50 CONTINUE
      RETURN
      END
      SUBROUTINE INTEGR (TIN,BCOMP,LAMBDA,MU,S,N,K,Q,C,RV,Z,W,P,NEQ,B,
     •  MTF,FOFX)
C
C       ********************************
C       *                              *
C       * THIS SUBROUTINE COMPUTES THE *
C       * MEAN TIME TO FAILURE - MTF   *
C       * AND THE RELIABILITY AT MTF   *
C       *                              *
C       ********************************
C
      REAL K ,MTF
      REAL LAMBDA,MU
      INTEGER N,S,Z,W
      LOGICAL TIN,BCOMP,TESTG
      LOGICAL OUTPT1,OUTPT2,OUTPT3,OUTPT4,OUTPT5
      COMMON /OUTPT/ OUTPT1,OUTPT2,OUTPT3,OUTPT4,OUTPT5
      TESTG=.FALSE.
      A=0.0
      HSTAR=0.01
      HMIN=0.0001
      HMAX=10.0
      ERMAX=1.0E-6
      KEY=0
      PMTF=0.0
      D=B
      IF (B.NE.0.0  ) GO TO 3
C
C     *** INPUT UPPER LIMIT OF INTEGRATION ***
C     *** IF IT WAS NOT INPUTED IN SVAR    ***
      WRITE (6,1000)
    1 READ(5,1001)  B1
    2 IF (OUTPT3) WRITE (6,1005) B1
      D=B1
C
C     *** CALL ROMBERG INTEGRATOR ***
    3 CALL ROMBD (A,D,X1,FOFX,HSTAR,HMIN,HMAX,ERMAX,MTF,K1,KEY)
    5 X=X1
      IF (TIN) X=LAMBDA*X1
      RL= EXP(-X)
      IF (K.NE.0.0  ) RS= EXP(-X/K)
      CALL RELEQS (NEQ,BCOMP,RL,RS,S,N,K,Q,C,RV,Z,W,P,X,FOFX)
      IF (TESTG) GO TO 90
      CALL ROM2D(A,D,X1,FOFX,HSTAR,HMIN,HMAX,ERMAX,MTF,K1,KEY)
      IF (K1.EQ.1) GO TO 5
      MTF=MTF+PMTF
      REL=ABS(MTF-PMTF)/ MAX1(ABS(MTF),1.0)
      IF (REL.LT.1.0E-8) GO TO 80
      IF (D.GT.1.0E24) GO TO 80
      PMTF=MTF
      A=D
      D=1.5E0*D
      GO TO 3
   80 IF (OUTPT4) WRITE (6,1002) MTF,D
      X1=MTF
      TESTG=.TRUE.
      GO TO 5
   90 IF (OUTPT4) WRITE (6,1004) FOFX
      RETURN
 9999 STOP
 1000 FORMAT( * THE INTEGRATION UPPER LIMIT -B- WAS NOT INPUTTED IN*,/
```

```
     *,* NAMELIST VAR, THE VALUE OF =B= IS EXPECTED AND MUST BE*,/,
     * * .LE. 1.0E24 AND .GT. 0.0*)
 1001 FORMAT (F20.0)
 1002 FORMAT(//* MEAN TIME TO FAILURE = MTF = *,E15.8,/,
     **  UPPER LIMIT FOR INTEGRATION = B = *,E15.8)
 1004 FORMAT(* RELIABILITY AT MTF = *,E15.8)
 1005 FORMAT (1H ,F20.5)
      END
      SUBROUTINE NEQ2A (RL,RS,S,K,Q,RV,Z,W,P,LAMT,R,C)
C
C        *******************************
C        *                             *
C        * THIS SUBROUTINE COMPUTES THE *
C        * RELIABILITY FOR EQUATION 2A  *
C        *                             *
C        *******************************
C
      REAL K,LAMT
      INTEGER I,S,Z,W
      SUM=0.0
      IF (LAMT.NE.0.0  ) GO TO 5
      R=RV**(W*Z)
      RETURN
    5 IF (S.NE.0) GO TO 7
      R=((RL**(Q/W))*RV)**(W*Z)
      RETURN
    7 DO 10 I=1,S
   10 SUM=SUM+(C*(1.0-RS**(1.0/W)))**I/FFAC(I)*PROD(J,I-1,Q,K)
      R=(RV*RL**(Q/W)*(1.0+SUM))**(W*Z)
      RETURN
      END
      SUBROUTINE NEQ2B (RL,LAMT,S,RV,Z,W,P,R,C,Q)
C
C        *******************************
C        *                             *
C        * THIS SUBROUTINE COMPUTES THE *
C        * RELIABILITY FOR EQUATION 2B  *
C        *                             *
C        *******************************
C
      REAL LAMT
      INTEGER S,Z,W
      SUM=0.0
      IF (LAMT.NE.0.0  ) GO TO 5
      R=RV**(W*Z)
      RETURN
    5 IF (S.NE.0) GO TO 7
      R=((RL**(Q/W))*RV)**(W*Z)
      RETURN
    7 DO 10 I=1,S
   10 SUM=SUM+(((LAMT*C*Q)/W)**I)/FFAC(I)
      R=(RV* EXP((-LAMT*Q)/W)*(1.0+SUM))**(W*Z)
      RETURN
      END
      SUBROUTINE NEQ1A (RL,RS,S,N,K,RV,Z,W,P,LAMT,R)
C
C        *******************************
C        *                             *
C        * THIS SUBROUTINE COMPUTES THE *
C        * RELIABILITY FOR EQUATION 1A  *
C        *                             *
C        *******************************
C
      REAL LAMT,K,KLS
      INTEGER S,S2,ETA,I,N,W,Z ,S22
      SUM=0.0
```

```
      SUM1=0.                                                           NEQ1A  13
      SUM2=0.0                                                          NEQ1A  14
      SUM3=0.0                                                          NEQ1A  15
      S2=S-2                                                            NEQ1A  16
      IF (LAMT.NE.0.0   ) GO TO 3                                       NEQ1A  17
      R=RV**(W*Z)                                                       NEQ1A  18
      RETURN                                                            NEQ1A  19
    3 ETA=2*N+1                                                         NEQ1A  20
      ETAKS=ETA*K+S                                                     NEQ1A  21
      RLAM=RL**(1.0/W)                                                  NEQ1A  22
      RMU=RS**(1.0/W)                                                   NEQ1A  23
      IF (S-1) 5,20,50                                                  NEQ1A  24
    5 R=0.0                                                             NEQ1A  25
      NN=N+1                                                            NEQ1A  26
      DO 10 IX=1,NN                                                     NEQ1A  27
      I=IX-1                                                            NEQ1A  28
   10 R=R+FNCK(ETA,I)*(1.0-RLAM)**I*RLAM**(ETA-I)                       NEQ1A  29
      R=(R*RV)**(W*Z)                                                   NEQ1A  30
      RETURN                                                            NEQ1A  31
   20 NN=N+1                                                            NEQ1A  32
      DO 40 IX=1,NN                                                     NEQ1A  33
      I=IX-1                                                            NEQ1A  34
      SUM=0.0                                                           NEQ1A  35
      II=I+1                                                            NEQ1A  36
      DO 30 LX=1,II                                                     NEQ1A  37
      L=LX-1                                                            NEQ1A  38
   30 SUM=SUM+FNCK(I,L)*(-1)**(I-L)*(1.0/(RMU*RLAM**L)-1.0)/(K*L+1.0)   NEQ1A  39
   40 SUM1=SUM1+FNCK(ETA,I)*SUM                                         NEQ1A  40
      R=RLAM**ETA*RMU*(1.0+(ETA*K+1.0)*SUM1)                            NEQ1A  41
      R=(R*RV)**(W*Z)                                                   NEQ1A  42
      RETURN                                                            NEQ1A  43
   50 NN=N+1                                                            NEQ1A  44
      DO 80 IX=1,NN                                                     NEQ1A  45
      I=IX-1                                                            NEQ1A  46
      II=I+1                                                            NEQ1A  47
      SUM1=0.0                                                          CAREFIX 1
      DO 70 LX=1,II                                                     NEQ1A  48
      L=LX-1                                                            NEQ1A  49
      SUM2=0.0                                                          NEQ1A  51
      KLS=K*L+S                                                         NEQ1A  52
       S22=S2+1                                                         NEQ1A  53
      DO 60 JX=1,S22                                                    NEQ1A  54
      J=JX-1                                                            NEQ1A  55
   60 SUM2=SUM2+RCOMB(KLS,J+1)*(1.0/RMU-1.0)**(J+1)                     NEQ1A  56
   70 SUM1=SUM1+FNCK(I,L)*(-1)**(I-L)*(1.0/(RMU**S*RLAM**L)-1.0-SUM2)/  NEQ1A  57
     *  RCOMB(KLS,S)                                                    NEQ1A  58
   80 SUM=SUM+FNCK(ETA,I)*RCOMB(ETAKS,S)*SUM1                           NEQ1A  59
       S22=S2+1                                                         NEQ1A  60
      DO 90 JX=1,S22                                                    NEQ1A  61
      J=JX-1                                                            NEQ1A  62
   90 SUM3=SUM3+RCOMB(ETAKS,J+1)*(1.0/RMU-1.0)**(J+1)                   NEQ1A  63
      R=RLAM**ETA*RMU**S*(1.0+SUM3+SUM)                                 NEQ1A  64
      R=(R*RV)**(W*Z)                                                   NEQ1A  65
      RETURN                                                            NEQ1A  66
      END                                                               NEQ1A  67
      SUBROUTINE NEQ1B (RL,S,N,RV,Z,W,P,LAMT,R)                         NEQ1B   1
C                                                                       NEQ1B   2
C         *********************************                            NEQ1B   3
C         *                               *                            NEQ1B   4
C         * THIS SUBROUTINE COMPUTES THE  *                            NEQ1B   5
C         * RELIABILITY FOR EQUATION 1B   *                            NEQ1B   6
C         *                               *                            NEQ1B   7
C         *********************************                            NEQ1B   8
C                                                                       NEQ1B   9
      REAL LAMT                                                         NEQ1B  10
      INTEGER S,N,Z,W,ETA,N2,S1   ,S8                                   NEQ1B  11
```

```
      IF (LAMT.NE.0.0  ) GO TO 10                                       NEQ1B     12
      R=RV**(W*Z)                                                       NEQ1B     13
      RETURN                                                            NEQ1B     14
   10 N2=2*N                                                            NEQ1B     15
      ETA=N2+1                                                          NEQ1B     16
      RLAM=RL**(1.0/W)                                                  NEQ1B     17
      S1=S-1                                                            NEQ1B     18
      SUM1=0.0                                                          NEQ1B     19
      SUM2=0.0                                                          NEQ1B     20
      SUM3=0.0                                                          NEQ1B     21
      SUM4=0.0                                                          NEQ1B     22
      IF (S1) 20,30,60                                                  NEQ1B     23
   20 SUM1 = RLAM**ETA                                                  NEQ1B     24
      DO 25 I=1,N                                                       NEQ1B     25
      SUM1=SUM1+((1.0  -RLAM)**I)*(RLAM**(ETA-I))*FNCK(ETA,I)           NEQ1B     26
   25 CONTINUE                                                          NEQ1B     27
      R=(RV*SUM1)**(W*Z)                                                NEQ1B     28
      RETURN                                                            NEQ1B     29
   30 DO 50 I=1,N                                                       NEQ1B     30
      SUM2=0.0                                                          NEQ1B     31
      DO 40 J=1,I                                                       NEQ1B     32
   40 SUM2=SUM2+FNCK(I,J)*(-1)**(I-J)*(1.0  /(RLAM**J)-1.0  )/J         NEQ1B     33
   50 SUM1=SUM1+FNCK(ETA,I)*SUM2                                        NEQ1B     34
      R=(RV*RLAM**ETA*(1.0  +ETA*LAMT*(-1)**N*FNCK(N2,N)                NEQ1B     35
     *  +ETA*SUM1))**(W*Z)                                              NEQ1B     36
      RETURN                                                            NEQ1B     37
   60 DO 90 I=1,N                                                       NEQ1B     38
      SUM2=0.0                                                          NEQ1B     39
      DO 80 J=1,I                                                       NEQ1B     40
      SUM1=0.0                                                          NEQ1B     41
      DO 70 L=1,S1                                                      NEQ1B     42
   70 SUM1=SUM1+LAMT**L/(FFAC(L)*J**(S-L))                              NEQ1B     43
   80 SUM2=SUM2+FNCK(I,J)*(-1)**(I-J)*((1.0  /(RLAM**J)-1.0  )/(J**S)   NEQ1B     44
     *  -SUM1)                                                          NEQ1B     45
   90 SUM3=SUM3+FNCK(ETA,I)*SUM2                                        NEQ1B     46
      SS=S1+1                                                           NEQ1B     47
      DO 100 IX=1,SS                                                    NEQ1B     48
      I=IX-1                                                            NEQ1B     49
  100 SUM4=SUM4+(ETA*LAMT)**I/FFAC(I)                                   NEQ1B     50
      R=(RV*RLAM**ETA*(SUM4+(ETA*LAMT)**S*(-1)**N*FNCK(N2,N)/FFAC(S)+   NEQ1B     51
     *  ETA**S*SUM3))**(W*Z)                                            NEQ1B     52
      RETURN                                                            NEQ1B     53
 1001 FORMAT (* IMPROPER INPUT FOR S - S CANNOT BE LESS THAN 1*)        NEQ1B     54
      END                                                               NEQ1B     55
      SUBROUTINE NEQ3 (RL,RS,S,K,Q,C,RV,Z,W,P,BCOMP,LAMT,R)             NEQ3       1
C                                                                       NEQ3       2
C        ********************************                               NEQ3       3
C        *                              *                               NEQ3       4
C        * THIS SUBROUTINE COMPUTES THE *                               NEQ3       5
C        * RELIABILITY FOR EQUATION 3   *                               NEQ3       6
C        *                              *                               NEQ3       7
C        ********************************                               NEQ3       8
C                                                                       NEQ3       9
      REAL K,LAMT                                                       NEQ3      10
      INTEGER I,S,Z,W ,SS                                               NEQ3      11
      LOGICAL BCOMP                                                     NEQ3      12
      IF (C.LT.1.0  .AND.C.GT.0.C  ) GO TO 5                            NEQ3      13
      WRITE (6,1000) C                                                  NEQ3      14
      GO TO 40                                                          NEQ3      15
    5 IF (LAMT.NE.0.0  ) GO TO 7                                        NEQ3      16
      R=RV**(W*Z)                                                       NEQ3      17
      RETURN                                                            NEQ3      18
    7 SUM=0.0                                                           NEQ3      19
      SS=S+1                                                            NEQ3      20
      DO 10 IX=1,SS                                                     NEQ3      21
      I=IX-1                                                            NEQ3      22
```

```
   10 SUM=SUM+FNCK(S,I)*C**I*(1.0-C)**(S-I)                              NEQ3
      IF (BCOMP) GO TO 20                                                NEQ3
      CALL NEQ2A (RL,RS,S,K,Q,RV,Z,W,P,LAMT,R)                           NEQ3
      GO TO 30                                                           NEQ3
   20 CALL NEQ2B (LAMT,S,RV,Z,W,P,R)                                     NEQ3
   30 R=(SUM*R)**Z                                                       NEQ3
   40 RETURN                                                             NEQ3
 1000 FORMAT (* IMPROPER INPUT FOR C - C = *,E15.8,/,                     NEQ3
     ** MUST HAVE 0.LT.C.LT.1*)                                          NEQ3
      END                                                                NEQ3
      SUBROUTINE NEQ4A (RL,RS,S,K,RV,Z,W,P,LAMT,R)                       NEQ4A
C                                                                        NEQ4A
C       ********************************                                 NEQ4A
C       *                              *                                 NEQ4A
C       * THIS SUBROUTINE COMPUTES THE *                                 NEQ4A
C       * RELIABILITY FOR EQUATION 4A  *                                 NEQ4A
C       *                              *                                 NEQ4A
C       ********************************                                 NEQ4A
C                                                                        NEQ4A
      REAL K,LAMT                                                        NEQ4A
      INTEGER S,Z,W,S1                                                   NEQ4A
      IF (S) 30,40,60                                                    NEQ4A
   30 WRITE (6,1001)                                                     NEQ4A
 1001 FORMAT (* ILLEGAL VALUE FOR S - S CANNOT BE NEGATIVE*)             NEQ4A
      RETURN                                                             NEQ4A
   40 IF (LAMT.NE.0.0  ) GO TO 50                                        NEQ4A
      R=RV**(W*Z)                                                        NEQ4A
      RETURN                                                             NEQ4A
   50 R=((1.5  *RL**(1.0/W)-0.5*RL**(3.0/W))*RV)**(W*Z)                  NEQ4A
      RETURN                                                             NEQ4A
   60 IF (LAMT.NE.0.0  ) GO TO 10                                        NEQ4A
      R=RV**(W*Z)                                                        NEQ4A
      RETURN                                                             NEQ4A
   10 SUM=0.0                                                            NEQ4A
      RES=S                                                              NEQ4A
      S1=S-1                                                             NEQ4A
      RES1=S1                                                            NEQ4A
      NS1=S1+1                                                           NEQ4A
      DO 20 IX=1,NS1                                                     NEQ4A
      I=IX-1                                                             NEQ4A
   20 SUM=SUM+FNCK(S,I)*((-1)**(I))*((1.0  /(RS**((RES-I)/W))-1.0  ))*((NEQ4A
     *3.0  *K**2))/((2.0  *K+I)*(3.0  *K+I))                             NEQ4A
      PROD2=PROD1(1,S,3.0  ,K,0.0  )                                     NEQ4A
      PROD3=PROD1(1,S,3.0  ,K,2.0  *K)                                   NEQ4A
      R=(((RL**(3.0/W))*(RS**(RES/W))*((1.0  )+1.5  *((1.0  /((RL**(2.0/NEQ4A
     *W))*(RS**(RES/W))))-1.0  )*PROD3-PROD2*SUM))*RV)**(W*Z)            NEQ4A
      RETURN                                                             NEQ4A
      END                                                                NEQ4A
      SUBROUTINE NEQ4B (RL,S,RV,Z,W,P,LAMT,R)                            NEQ4B
C                                                                        NEQ4B
C       ********************************                                 NEQ4B
C       *                              *                                 NEQ4B
C       * THIS SUBROUTINE COMPUTES THE *                                 NEQ4B
C       * RELIABILITY FOR EQUATION 4B  *                                 NEQ4B
C       *                              *                                 NEQ4B
      REAL LAMT                                                          NEQ4B
      INTEGER S,Z,W,S1                                                   NEQ4B
      IF (S) 10,20,40                                                    NEQ4B
   10 WRITE (6,1001)                                                     NEQ4B
      RETURN                                                             NEQ4B
   20 IF (LAMT.NE.0.0  ) GO TO 30                                        NEQ4B
      R=RV**(W*Z)                                                        NEQ4B
      RETURN                                                             NEQ4B
   30 R=((1.5  *RL**(1.0/W)-0.5*RL**(3.0/W))*RV)**(W*Z)                  NEQ4B
      RETURN                                                             NEQ4B
   40 IF (LAMT.NE.0.0  ) GO TO 50                                        NEQ4B
```

```
      R=RV**(W*Z)                                                        NEQ4B
      RETURN                                                             NEQ4B
   50 SUM=0.0                                                            NEQ4B
      S1=S+1                                                             NEQ4B
      NS=S+1                                                             NEQ4B
      DO 60 IX=1,NS                                                      NEQ4B
      I=IX-1                                                             NEQ4B
   60 SUM=SUM+(((3.0  *LAMT)**I)*(((1.50  )**(S1-I))-1.0  ))/FFAC(I)     NEQ4B
      R=((((1.50  )**(S1))*(RL**(1.0/W))-((RL**(3.0/W))*SUM))*RV)**(W*Z)NEQ4B
      RETURN                                                             NEQ4B
 1001 FORMAT (* ILLEGAL VALUE FOR S - S CANNOT BE NEGATIVE*)            NEQ4B
      END                                                                NEQ4B
      SUBROUTINE NEQ5 (RL,RV,Z,W,P,LAMT,R)                               NEQ5
C                                                                        NEQ5
C       ********************************                                 NEQ5
C       *                              *                                 NEQ5
C       * THIS SUBROUTINE COMPUTES THE *                                 NEQ5
C       * RELIABILITY FOR EQUATION 5   *                                 NEQ5
C       *                              *                                 NEQ5
C       ********************************                                 NEQ5
C                                                                        NEQ5
      REAL LAMT                                                          NEQ5
      INTEGER W,Z                                                        NEQ5
      RLAM=RL**(1.0/W)                                                   NEQ5
      R=3.0  *RLAM**2-2.0  *RLAM**3+6.0  *P*(1.0  -P)*RLAM               NEQ5
     *  *(1.0  -RLAM)**2                                                 NEQ5
      R=(R*RV)**(W*Z)                                                    NEQ5
      RETURN                                                             NEQ5
      END                                                                NEQ5
      SUBROUTINE NEQ6 (RL,Z,W,P,LAMT,R)                                  NEQ6
C                                                                        NEQ6
C       ********************************                                 NEQ6
C       *                              *                                 NEQ6
C       * THIS SUBROUTINE COMPUTES THE *                                 NEQ6
C       * RELIABILITY FOR EQUATION 6   *                                 NEQ6
C       *                              *                                 NEQ6
C       ********************************                                 NEQ6
C                                                                        NEQ6
      REAL LAMT                                                          NEQ6
      INTEGER Z,W                                                        NEQ6
      RW=W                                                               NEQ6
      IF(RW .EQ. 0.0)WRITE(6,20  )                                       NEQ6
   20 FORMAT(//*EEEEEEEEEEEEERROR, THE PARAMETER - W - IS ZERO *)        NEQ6
      R=RL**(Z/RW)                                                       NEQ6
      RETURN                                                             NEQ6
      END                                                                NEQ6
      SUBROUTINE NEQ7(RL,RS,NS,N,XK,Q,CX,RV,NZ,NW,P,                     INTERP7
     *     BCOMP,XLAMT,DCR)                                              INTERP7
C                                                                        FINAL
C          NEQ7 PROVIDES A SECOND ORDER (PARABOLIC) INTERPOLATION        FINAL
C          FROM THE PRECALCULATED REL. VALUES FOR THE DUAL MODE MODEL    FINAL
C          SAVED IN ARRAY R.                                             FINAL
C                                                                        FINAL
      LOGICAL BCOMP,RSGN                                                 INTERP7
      COMMON /DRIVE/TX,STEP,HSTEP,NUN,RSGN,JMAX,I10,SUMLAM               FINAL
      T=-ALOG(RL)/SUMLAM                                                 FINAL
      COMMON //IDUM(11),FDUM(1),R(121,10)                                CRUSH
      J=T/STEP+1                                                         INTERP7
      IF(NUN.EQ.0.OR.J.GT.(JMAX+1))  GO TO 100                           INTERP7
      J=MAX0(2,J)                                                        INTERP7
      J=MIN0(J,JMAX)                                                     INTERP7
      I=J-1                                                              INTERP7
      K=J+1                                                              INTERP7
      X1=STEP*(I-1)                                                      INTERP7
      X2=X1+STEP                                                         INTERP7
      X3=X2+STEP                                                         INTERP7
```

```
      Y1=R(1,I10)                                                       FINAL        8
      Y2=R(J,I10)                                                       FINAL        9
      Y3=R(K,I10)                                                       FINAL       10
      Z1=Y1-2*Y2+Y3                                                     INTERP7     18
      Z2=2.0*STEP**2                                                    INTERP7     19
      A=Z1/Z2                                                           INTERP7     20
      Z3=(Y3-Y1)/(2.0*STEP)                                             INTERP7     21
      B=Z3-2.0*A*X2                                                     INTERP7     22
      C=Y2-A*X2**2-B*X2                                                 INTERP7     23
      DCR=A*T**2+B*T+C                                                  INTERP7     24
      IF(DCR.LT.0.0.OR.DCR.GT.1.0)  GO TO 100                           INTERP7     25
      RETURN                                                            INTERP7     26
  100 WRITE(6,11)                                                       INTERP7     27
   11 FORMAT(/25X,*ERROR IN EQUATION 7 INTERPOLATION*)                  INTERP7     28
      STOP                                                              INTERP7     29
      END                                                               INTERP7     30
      SUBROUTINE PARAR1 (R2,R1MIN,R1MAX,R1STEP,PLOTR2,R2OPT,I10,TMAX)   PARAR1       1
C       *********************************                               PARAR1       2
C       *                               *                               PARAR1       3
C       * THIS SUBROUTINE COMPUTES THE  *                               PARAR1       4
C       * RATIO OF MAX. MISSION TIMES   *                               PARAR1       5
C       *(RATIF) FOR THE VARIOUS SYSTEM*                                PARAR1       6
C       * PARAMETERS SPECIFIED.         *                               PARAR1       7
C       *                               *                               PARAR1       8
C       *********************************                               PARAR1       9
            DIMENSION PARAM(16,10)                                      PARAR1      10
      REAL COMR1(101,4),TMAX1(101),TMAX2(101),RATIF(101)                PARAR1      11
      REAL TMAX(101,15),XAXIS(101)                                      PARAR1      12
      REAL TMX(4),INFI,NITY                                             PARAR1      13
      REAL ENC(20),AB                                                   PARAR1      14
      INTEGER PROD(10),PAR(6),NPROD,NEQ,N10,K1,I10,R2OPT                PARAR1      15
      LOGICAL PRODT,RESLT,PLOTR2,NOPLOT                                 PARAR1      16
      LOGICAL OUTPT1,OUTPT2,OUTPT3,OUTPT4,OUTPT5                        PARAR1      17
      COMMON/GRAPH/ NEQ,PROD,NPROD,PAR,ENC,AB,PRODT                     PARAR1      18
      COMMON /PARM/ PARAM,N10,K1                                        PARAR1      19
      COMMON /OUTPT/ OUTPT1,OUTPT2,OUTPT3,OUTPT4,OUTPT5                 PARAR1      20
      COMMON/CPLOT/COMR1                                                PARAR1      21
      EQUIVALENCE (COMR1(1,1),XAXIS(1)), (COMR1(1,2),TMAX1(1)),         PARAR1      22
     *            (COMR1(1,3),TMAX2(1)), (COMR1(1,4),RATIF(1))          PARAR1      23
      DATA INFI/4HINFI/,NITY/4HNITY/                                    PARAR1      24
      DATA ITEST1/1/                                                    PARAR1      25
      DATA NPT/10/                                                      PARAR1      26
      NPLT=0                                                            PARAR1      27
      NOPLOT=.TRUE.                                                     PARAR1      28
      IF (R2OPT.EQ.1) GO TO 30                                          PARAR1      29
      IF (R2OPT.EQ.3) GO TO 10                                          PARAR1      30
      IR2=I10                                                           PARAR1      31
      IR1=I10-1                                                         PARAR1      32
      GO TO 60                                                          PARAR1      33
   10 WRITE (6,1001) PAR                                                PARAR1      34
   20 READ(5,1002) IR1,IR2                                              PARAR1      35
   21 IF (OUTPT3) WRITE (6,1003) IR1,IR2                                PARAR1      36
      IF (IR1.LE.I10.AND.IR2.LE.I10) GO TO 60                           PARAR1      37
      WRITE (6,1004)                                                    PARAR1      38
      STOP                                                              PARAR1      39
   30 IR1=0                                                             PARAR1      40
   40 IR1=IR1+1                                                         PARAR1      41
      IF (IR1.EQ.N10) GO TO 130                                         PARAR1      42
      IR2=IR1                                                           PARAR1      43
   50 IR2=IR2+1                                                         PARAR1      44
      IF (IR2.GT.N10) GO TO 40                                          PARAR1      45
   60 IF (OUTPT5) WRITE (6,1005) PAR,PARAM(IR1,K1),PAR,PARAM(IR2,K1),   PARAR1      46
     *  NEQ,AB,R2                                                       PARAR1      47
      R1=R1MIN                                                          PARAR1      48
      J=0                                                               PARAR1      49
   90 J=J+1                                                             PARAR1      50
```

```
      XAXIS(J)=R1                                                       PARAR1    51
      IF (R1.NE.0.0  ) GO TO 100                                        PARAR1    52
      ENCODE(8,1006,TMX(1)) INFI,NITY                                   PARAR1    53
      ENCODE(8,1006,TMX(3)) INFI,NITY                                   PARAR1    54
      TMAX1(J)=TMAX(J,IR1)                                              PARAR1    55
      TMAX2(J)=TMAX(J,IR2)                                              PARAR1    56
      RATIF(J)=1.0                                                      PARAR1    57
      GO TO 110                                                         PARAR1    58
  100 ENCODE (14,1007,TMX(1)) TMAX(J,IR1)                               PARAR1    59
      ENCODE (14,1007,TMX(3)) TMAX(J,IR2)                               PARAR1    60
      TMAX1(J)=TMAX(J,IR1)                                              PARAR1    61
      TMAX2(J)=TMAX(J,IR2)                                              PARAR1    62
      IF (TMAX(J,IR1).NE.0.0) GO TO 105                                 PARAR1    63
      RATIF(J)=1.0                                                      PARAR1    64
      GO TO 110                                                         PARAR1    65
  105 RATIF(J)=TMAX(J,IR2)/TMAX(J,IR1)                                  PARAR1    66
  110 IF (OUTPTS) WRITE (6,1008) R1,(TMX(I),I=1,4),RATIF(J)             PARAR1    67
      R1=R1+R1STEP                                                      PARAR1    68
      IF (R1.LE.R1MAX) GO TO 90                                         PARAR1    69
      IF (.NOT.PLOTR2) GO TO 118                                        PARAR1    70
      IF (NPT.LE.15) GO TO 114                                          PARAR1    71
      IF (NOPLOT) WRITE (6,1009)                                        PARAR1    72
      NOPLOT=.FALSE.                                                    PARAR1    73
      GO TO 118                                                         PARAR1    74
  114 NPLT=NPLT+1                                                       PARAR1    75
      CALL PLOTT(J,2)                                                   PARAR1    76
  118 IF (R2OPT.EQ.1) GO TO 50                                          PARAR1    77
  130 IF (PLOTR2) WRITE (6,1014) NPLT                                   PARAR1    78
      RETURN                                                            PARAR1    79
 1001 FORMAT (* IF MAXIMUM MISSION TIME IS WANTED FOR PARAMETER *,6A1,/,PARAR1    80
     *  * INPUT T1 IN COLUMNS 1-2 AND T2 IN COLUMNS 3-4*,/,             PARAR1    81
     *  * OTHERWISE INPUT NO IN COLUMNS 1-2*)                           PARAR1    82
 1002 FORMAT (2I2)                                                      PARAR1    83
 1003 FORMAT (1H ,2I2)                                                  PARAR1    84
 1004 FORMAT(*T1 OR T2 HAS NOT BEEN CALCULATED YET - TRY AGAIN*)        PARAR1    85
 1005 FORMAT(///* MAXIMUM MISSION TIME FOR *,6A1,* = *,G14.7,/,         PARAR1    86
     *  * AND *,6A1,* = *,G14.7,*  FOLLOWS FOR EQUATION  *,I2,A6,/,     PARAR1    87
     *  15X,*REFERENCE R2 = *,F7.5,/,4X,*R1*,9X,*TMAX1*,                PARAR1    88
     *  20X,*TMAX2*,18X,*RATIF*)                                        PARAR1    89
 1006 FORMAT(2A4)                                                       PARAR1    90
 1007 FORMAT(E14.7)                                                     PARAR1    91
 1008 FORMAT(1H ,F7.5,2(4X,2A10),3X,E14.7)                              PARAR1    92
 1009 FORMAT (* A MAXIMUM OF 15 PLOTS HAS BEEN GENERATED FOR MAXIMUM*,/,PARAR1    93
     *  * MISSION TIME CALCULATIONS - NO MORE PLOTS TO BE GENERATED*,/, PARAR1    94
     *  * FOR THESE CALCULATIONS*)                                      PARAR1    95
 1014 FORMAT(//42X,* TMAX1(A),TMAX2(B),RATIF(C)  VS  R1(XAXIS)*//       PARAR1    96
     *40X,I2,* MAXIMUM MISSION TIME PLOT(S) FOR VARYING *,/,40X,        PARAR1    97
     *  * PARAMETER VALUES COMPLETED*)                                  PARAR1    98
      END                                                               PARAR1    99
      SUBROUTINE PLOTRV(I10,DELRL)                                      PLOTRV     1
C       ******************************                                  PLOTRV     2
C       *                            *                                  PLOTRV     3
C       * THIS SUBROUTINE PLOTS THE  *                                  PLOTRV     4
C       * LOCUS OF RV SUCH THAT THE  *                                  PLOTRV     5
C       * SYSTEM RELIABILITY EQUALS  *                                  PLOTRV     6
C       * THE UNIT RELIABILITY, R.   *                                  PLOTRV     7
C       *                            *                                  PLOTRV     8
C       ******************************                                  PLOTRV     9
      COMMON /GRAFF/XAXIS(1),YAXIS(1,1),PSYMBL(16)                      CRUSH      1
      NPTP=1.0  /DELRL                                                  PLOTRV    11
      CALLPLOTN(I10=1,NPTP+1,50,XAXIS,YAXIS,PSYMBL,16,210,.F.)          PLOTRV    12
      CALL ROWPLT                                                       PLOTRV    13
      WRITE (6,1000)                                                    PLOTRV    14
      RETURN                                                            PLOTRV    15
 1000 FORMAT(//40X,* RV  VS  R  PLOT COMPLETED*)                        PLOTRV    16
      END                                                               PLOTRV    17
```

```
      FUNCTION RCOMB (TCOM,BCOM)
C        ******************************
C        *                            *
C        * THIS FUNCTION COMPUTES THE *
C        * GENERALISED BINOMIAL       *
C        * COEFFICIENTS (THOSE NOT    *
C        * NECESSARILY HAVING INTEGER *
C        * VALUES).                   *
C        ******************************
      INTEGER BCOM
      PRDT=1.0
      TCOMT=TCOM
      INT=TCOM-BCOM
      IF (INT) 10,30,30
   10 WRITE (6,1000)
      CALL EXIT
   30 DO 40 I=1,BCOM
      PRDT=PRDT*TCOMT
   40 TCOMT=TCOMT-1.0
      RCOMB=PRDT/FFAC(BCOM)
      RETURN
 1000 FORMAT (* ERROR IN COMBINATORIAL - CHECK INPUT*,/,
     *  * PROGRAM IS TERMINATED*)
      END
      FUNCTION PROD (LIM1,LIM2,Q,K)
C        ****************************
C        *                          *
C        * THIS FUNCTION CALCULATES *
C        * SPECIAL PRODUCT FACTORS  *
C        * FOR THE COMPUTATION OF   *
C        * RELIABILITY EQUATIONS.   *
C        *                          *
C        ****************************
      REAL K
      PROD=1.0
      DO 10 J1=LIM1,LIM2
   10 PROD=(J1+Q*K)*PROD
      RETURN
      END
      FUNCTION PROD1 (LIM1,LIM2,Q,K,DEN)
C        ****************************
C        *                          *
C        * THIS FUNCTION CALCULATES *
C        * SPECIAL PRODUCT FACTORS  *
C        * FOR THE COMPUTATION OF   *
C        * RELIABILITY EQUATIONS.   *
C        *                          *
C        ****************************
      REAL K
      PROD1=1.0
      DO 10 J1=LIM1,LIM2
   10 PROD1=PROD1*(Q*K+J1)/(DEN+J1)
      RETURN
      END
      SUBROUTINE PLOTR (KC,ABSC,XY,G,PLOT,PARAM,K1,IPLT)
C
C        **********************************
C        *                                *
C        * THIS SUBROUTINE IS A DRIVER    *
C        * FOR THE PLOT ROUTINE - PLOTN   *
C        *                                *
C        **********************************
C
      REAL MTF(16),FOFX(16),PARAM(16,10)
      REAL MIN,MAX
      REAL PDT(121 ),SIMPX(121 )
```

```
      DIMENSION XY(1,4),XNAME1(2),XNAME3(2),XNAME4(2),                  CRUSH      1
     * XNAME(2),YNAME(2)                                                 CRUSH      2
      REAL DIFF(1,1),RIF(1,1),GAIN(1,1),ABSC(1,1),                      CRUSH      3
     * G(1,1),FDUM(1),R(121,10)                                          CRUSH      4
      INTEGER IDUM(11),PLOT                                              PLOTR     17
      LOGICAL TIN,XYRANGE,LOG                                            PLOTR     18
      COMMON /PLOTIN/ TIN,MIN,MAX,STEP,DIFF,RIF,GAIN,MTF,FOPX,           PLOTR     19
     *  PDT,SIMPX,XMIN,XMAX,YMIN,YMAX,XYRANGE,IA,IAA,IB,IBB,             PLOTR     20
     *  IC,ICC,IPLT1,IPLT2,IPLT3,IPLT4,IPLT5                             PLOTR     21
      COMMON IDUM,FDUM,R                                                 PLOTR     22
      COMMON /GRAFF/DUM(2),PSYMBL(16)                                    CRUSH      5
       DATA JDIM/121/                                                    PLOTR     24
      DATA XNAME1(1),XNAME1(2),XNAME2,XNAME3(1) ,XNAME3(2),XNAME4(1),    PLOTR     25
     *XNAME4(2),XNAME5,XNAME6,XNAME7,YNAME2,YNAME3,YNAME4,YNAME5,YNAME6,PLOTR     26
     *YNAME7,YNAME8/8HMISSION ,7H TIME T,10HNORMALIZED,8HSIMPLEX ,       CAREFIX    1
     *10HRELYBILITY ,5HLOG10 ,3H(T),6H(LAMT),10HEXP(-LAMT) ,9HPARAMETER,PLOTR     28
     *10H          UN ,4HDIFF,3HRIF,4HGAIN,10HPRODUCT OF,3HMTF,6HAT MTF/  PLOTR     29
      DO 10 I=1,2                                                        PLOTR     30
   10 XNAME(I)=YNAME(I)=1H                                               PLOTR     31
      NPLT=0                                                             PLOTR     32
      NPTP=INT((MAX-MIN)/STEP)+1                                         PLOTR     33
      IPLOT=1                                                            PLOTR     34
      IAAT=IAA                                                           PLOTR     35
      IBBT=IBB                                                           PLOTR     36
      ICCT=ICC                                                           PLOTR     37
      IF (IPLT2.NE.1) IPLOT=2                                            PLOTR     38
   20 GO TO (30,40,50) ,IPLOT                                            PLOTR     39
C                                                                        PLOTR     40
C     *** SET UP INFORMATION FOR RELIABILITY PLOT ***                    PLOTR     41
   30 DO 35 I=1,KC                                                       PLOTR     42
      DO 35 J=1,JDIM                                                     PLOTR     43
   35 G(J,I)=R(J,I)                                                      PLOTR     44
C                                                                        PLOTR     45
C          SET UP SIMREL CURVE FOR REL. PLOTS WHEN X-AXIS IS EXP(-LAMT)  PLOTR     46
      KCEXP=KC+1                                                         PLOTR     47
      DO 38 J=1,JDIM                                                     PLOTR     48
   38 G(J,KCEXP)=SIMPX(J)                                                PLOTR     49
      YNAME(1)=XNAME3(2)                                                 PLOTR     50
      YNAME(2)=10H(EQA,EQB,.                                             PLOTR     51
      GO TO 170                                                          PLOTR     52
   40 IF (IPLT5.NE.1) GO TO 310                                          PLOTR     53
C                                                                        PLOTR     54
C     *** SET UP INFORMATION FOR UNRELIABILITY PLOT ***                  PLOTR     55
      DO 45 I=1,KC                                                       PLOTR     56
      DO 45 J=1,JDIM                                                     PLOTR     57
   45 G(J,I)=1.0-R(J,I)                                                  PLOTR     58
      YNAME(1)=YNAME2                                                    PLOTR     59
      YNAME(2)=XNAME3(2)                                                 PLOTR     60
      GO TO 170                                                          PLOTR     61
   50 WRITE (6,1016) NPLT                                                PLOTR     62
      RETURN                                                             PLOTR     63
C                                                                        PLOTR     64
C        ************************                                        PLOTR     65
C        *                      *                                        PLOTR     66
C        * ENTRY FOR DIFF, RIF, *                                        PLOTR     67
C        * AND GAIN PLOTS       *                                        PLOTR     68
C        *                      *                                        PLOTR     69
C        ************************                                        PLOTR     70
C                                                                        PLOTR     71
      ENTRY PLOTG                                                        PLOTR     72
      KCEXP=KC+1                                                         PLOTR     73
      NPTP=INT((MAX-MIN)/STEP)+1                                         PLOTR     74
      IAAT=IAA                                                           PLOTR     75
      IBBT=IBB                                                           PLOTR     76
      ICCT=ICC                                                           PLOTR     77
      NPLT=0                                                             PLOTR     78
```

```
      IPLOT=1
   70 GO TO (80,90,100,110) ,IPLOT
C
C     *** SET UP INFORMATION FOR DIFFERENCE PLOT ***
   80 DO 85 I=1,KC
      DO 85 J=1,JDIM
   85 G(J,I)=DIFF(J,I)
      YNAME(1)=1H
      YNAME(2)=YNAME3
      GO TO 170
C
C     *** SET UP INFORMATION FOR RELATIVE IMPROVEMENT FACTOR PLOT ***
   90 DO 95 I=1,KC
      DO 95 J=1,JDIM
   95 G(J,I)=RIF(J,I)
      YNAME(1)=1H
      YNAME(2)=YNAME4
      GO TO 170
C
C     *** SET UP INFORMATION FOR GAIN PLOT ***
  100 DO 105 I=1,KC
      DO 105 J=1,JDIM
  105 G(J,I)=GAIN(J,I)
      YNAME(1)=1H
      YNAME(2)=YNAME5
      GO TO 170
  110 WRITE (6,1016) NPLT
      RETURN
C
C       ***************************************
C       *                                     *
C       * ENTRY FOR MTF, RELABILITY AT MTF,   *
C       * AND PRODUCT OF RELIABILITIES PLOTS  *
C       *                                     *
C       ***************************************
C
      ENTRY PLOTPD
      KCEXP=KC+1
      IAAT=IAA
      IBBT=IBB
      ICCT=ICC
      NPLT=0
      IPLOT=1
      IF (PLUT.NE.5) IPLOT=5
  120 CONTINUE
      GO TO (130,140,150,160),IPLOT
C
C     *** SET UP INFORMATION FOR MEAN TIME TO FAILURE PLOT ***
  130 DO 135 I=1,KC
  135 G(I,1)=MTF(I)
      YNAME(1)=1H
      YNAME(2)=YNAME7
      GO TO 300
C
C     *** SET UP INFORMATION FOR RELIABILITY AT MTF PLOT ***
  140 DO 145 I=1,KC
      ABSC(I,1)=MTF(I)
  145 G(I,1)=FUFX(I)
      YNAME(1)=XNAME3(2)
      YNAME(2)=YNAME8
      GO TO 300
  150 IF (IPLT1.NE.1) GO TO 310
C
C     *** SET UP INFORMATION FOR PRODUCT OF RELIABILITIES PLOT ***
      DO 155 J=1,JDIM
  155 G(J,1)=PDT(J)
```

```
      NPTP= INT((MAX-MIN)/STEP)+1
      YNAME(1)=YNAME6
      YNAME(2)=XNAME5(2)
      GO TO 170
  160 WRITE (6,1016) NPLT
      RETURN
  170 CONTINUE
      IF (IA.NE.1) GO TO 210
C
C     *** SET UP INFORMATION FOR PLOTS WITH     ***
C     *** ABSCISSA T OR LOG(T) = BASE 10      ***
      XNAME(1)=XNAME1(1)
      XNAME(2)=XNAME1(2)
      LOG=.FALSE.
      GO TO 200
  190 XNAME1(1)=XNAME4(1)
      XNAME1(2)=XNAME4(2)
C
  200 CALL PLOTN(KC,NPTP,50,XY,G,PSYMBL,16,121,LOG)
C
C     *** PRINT DATA BELOW PLOT ***
      WRITE(6,205) (YNAME(KL),KL=1,2), (XNAME(KL),KL=1,2)
  205 FORMAT(//40X,2A10,2X,*VS*,2X,2A10)
      CALL ROWPLT
      NPLT=NPLT+1
  210 IF (IAA.NE.1) GO TO 220
      IAA=0
      LOG=.TRUE.
      GO TO 190
  220 CONTINUE
      IF (IB.NE.1) GO TO 250
C
C     *** SET UP INFORMATION FOR PLOTS WITH ***
C     *** ABSCISSA LAMT OR LOG(LAMT) = BASE 10 ***
      XNAME(1)=XNAME2
      XNAME(2)=XNAME1(2)
      LOG=.FALSE.
      GO TO 240
  230 XNAME(1)=XNAME4(1)
      XNAME(2)=XNAME5
C
  240 CALL PLOTN(KC,NPTP,50,XY(1,2),G,PSYMBL,16,121,LOG)
C
C     *** PRINT DATA BELOW PLOT ***
      WRITE(6,205) (YNAME(KL),KL=1,2), (XNAME(KL),KL=1,2)
      CALL ROWPLT
      NPLT=NPLT+1
  250 IF (IBB.NE.1) GO TO 260
      IBB=0
      LOG=.TRUE.
      GO TO 230
  260 CONTINUE
      IF (IC.NE.1) GO TO 290
C
C     *** SET UP INFORMATION FOR PLOTS WITH ***
C     *** ABSCISSA EXP(-LAMBDA*T) OR        ***
C     *** LOG(EXP(-LAMBDA*T)) = BASE 10     ***
      XNAME(1)=XNAME3(1)
      XNAME(2)=XNAME3(2)
      LOG=.FALSE.
      GO TO 280
  270 XNAME(1)=XNAME4(1)
      XNAME(2)=XNAME6
  280 CALL PLOTN(KC,NPTP,50,XY(1,3),G,PSYMBL,16,121,LOG)
C
C     *** PRINT DATA BELOW PLOT ***
```

```
      WRITE(6,205) (YNAME(KL),KL=1,2), (XNAME(KL),KL=1,2)               PLOTR    211
      CALL ROWPLT                                                       PLOTR    212
      NPLT=NPLT+1                                                       PLOTR    213
      LOG=.TRUE.                                                        PLOTR    214
  290 IF (ICC.NE.1) GO TO 310                                           PLOTR    215
      ICC=0                                                             PLOTR    216
      GO TO 270                                                         PLOTR    217
  300 XNAME(2)=XNAME7                                                   PLOTR    218
      XNAME(1)=1H                                                       PLOTR    219
      DO 305 I=1,KC                                                     PLOTR    220
  305 ABSC(I,1)=PARAM(I,K1)                                             PLOTR    221
      CALL PLOTN(1,KC,50,XY,G,PSYMBL,16,121,LOG)                        PLOTR    222
      WRITE(6,205) (YNAME(KL),KL=1,2), (XNAME(KL),KL=1,2)               PLOTR    223
      CALL ROWPLT                                                       PLOTR    224
      NPLT=NPLT+1                                                       PLOTR    225
  310 IPLOT=IPLOT+1                                                     PLOTR    226
      IAA=IAAT                                                          PLOTR    227
      IBB=IBBT                                                          PLOTR    228
      ICC=ICCT                                                          PLOTR    229
      GO TO (20,70,120) ,IPLT                                           PLOTR    230
 1016 FORMAT(//40X,I3,* PLOT(S) COMPLETED*)                             PLOTR    231
      END                                                               PLOTR    232
      SUBROUTINE PLOTT(NPP,IBIRTH)                                      PLOTT      1
C       ******************************                                  PLOTT      2
C       *                            *                                  PLOTT      3
C       * THIS SUBROUTINE PLOTS THE  *                                  PLOTT      4
C       * MAX. MISSION TIME FUNCTIONS*                                  PLOTT      5
C       * TMAX, SIMTMAX, SIMTIF AND  *                                  PLOTT      6
C       * RATIF.                     *                                  PLOTT      7
C       *                            *                                  PLOTT      8
C       ******************************                                  PLOTT      9
      DIMENSION PSYMBL(3)                                               PLOTT     10
      COMMON/PPLOT/XAXIS(101),YAXIS(101,3)                              PLOTT     11
      COMMON/CPLOT/XAXISC(101),YAXISC(101,3)                            PLOTT     12
      COMMON /GRAFF/DUM(2),PSYMBL                                       CRUSH      1
      IF(IBIRTH .EQ. 1)                                                 PLOTT     14
     1CALL PLOTN(3,NPP,50 ,XAXIS,YAXIS,PSYMBL,3 ,101 ,.F.)              PLOTT     15
      IF(IBIRTH .EQ. 2) CALL PLOTN(3,NPP,50,XAXISC,YAXISC,PSYMBL,3,101,.PLOTT     16
     2F.)                                                               PLOTT     17
      CALL ROWPLT                                                       PLOTT     18
      RETURN                                                            PLOTT     19
      END                                                               PLOTT     20
      SUBROUTINE PLOTN(NPLT,NBIN,NLIN,XAXIS,YAXIS,PSYMBL,MAXPLT,MAXPT,  PLOTN      1
     1LOG)                                                              PLOTN      2
      DIMENSION XAXIS(MAXPT),PSYMBL(MAXPLT),YAXIS(MAXPT,MAXPLT)         PLOTN      3
      COMMON/PLTWR/NUM,PS(10),ZLINE,BLANK,DUM(3),NHI,UMB,RESTY          PLOTN      4
      LOGICAL LOGN                                                      PLOTN      5
      COMMON /FUN/LOGN                                                  PLOTN      6
       DATA ZLINE/10H----------/                                        PLOTN      7
      LOGN=LOG                                                          PLOTN      8
      IF(.NOT.LOGN) GO TO 91                                            PLOTN      9
      DO 90 KIP=1,NPLT                                                  PLOTN     10
      DO 90 LIP=1,NBIN                                                  PLOTN     11
      YAXIS(LIP,KIP)=ALOG10(YAXIS(LIP,KIP))                             PLOTN     12
   90 CONTINUE                                                          PLOTN     13
   91 CONTINUE                                                          PLOTN     14
      NHI=(NBIN+9)/10                                                   PLOTN     15
      IF (NPLT.GT.0) GO TO  8                                           PLOTN     16
      NPLT=-NPLT                                                        PLOTN     17
      GO TO 18                                                          PLOTN     18
    8 CONTINUE                                                          PLOTN     19
      YMAX=YAXIS(1,1)                                                   PLOTN     20
      YMIN=YMAX                                                         PLOTN     21
      DO 2 J=1,NPLT                                                     PLOTN     22
      DO 2 I=1,NBIN                                                     PLOTN     23
      YMAX=AMAX1(YMAX,YAXIS(I,J))                                       PLOTN     24
```

```
    2 YMIN=AMIN1(YMIN,YAXIS(I,J))
      YSPAN=YMAX-YMIN
      AYMX=YMAX
      HYMX=YMAX
      IF (YMIN.GE.0.0) GO TO 10
      IF ((YMAX.GT.0.0).AND.(YMIN.LT.0.0)) GO TO 11
      HYMX=YMIN
      AYMX=-YMIN
      YMIN=YMAX
      YMAX=HYMX
   10 IF (YSPAN.GT.(AYMX/2.)) GO TO 11
      HYMX=HYMX/2.
      DO 12 J=1,NPLT
      DO 12 I=1,NBIN
   12 YAXIS(I,J)=YAXIS(I,J)-HYMX
      AYMX=AYMX/2.
      GO TO 10
   11 CONTINUE
      NLINES=IABS(NLIN)
      DY=AMAX1(YSPAN,AYMX)/FLOAT(NLINES)
      RESTY=YMAX-HYMX
      HYMX=DIM(HYMX,0.0)
      XMYM=HYMX
   18 CONTINUE
      NIBN=NBIN
      NGROS=(NBIN-1)/100
      IGROS=NGROS
   19 CONTINUE
      HYMX=XMYM
        PRINT 105
      IF (IGROS.EQ.0) GO TO 60
      NBIN=100
      GO TO 61
   60 NBIN=NIBN-NGROS*100
   61 NHI=(NBIN+9)/10
      ICOR=100*(NGROS-IGROS)
      DO 62 J=1,NPLT
      DO 62 I=1,NBIN
   62 YAXIS(I,J)=YAXIS(I+ICOR,J)
      DO 63 I=1,NBIN
   63 XAXIS(I)=XAXIS(I+ICOR)
      LFLG=0
      DO 20 K=1,NLINES
      IF ((HYMX.GT.0.0).OR.(LFLG.NE.0)) GO TO 23
      LFLG=1
      DO 25 I=1,NHI
   25 PS(I)=ZLINE
      PRINT 101,(PS(I),I=1,NHI)
   23 DO 21 I=1,NHI
   21 PS(I)=BLANK
      DO 22 J=1,NPLT
       DO 22 I=1,NBIN
      YI=YAXIS(I,J)
      IF ((HYMX.LT.YI).OR.((HYMX-YI).GT.DY)) GO TO 22
      I1=(I+9)/10
       I2=I-(I1-1)*10
      CALL MOVC(1,PSYMBL(J),I2,PS(I1))
   22 CONTINUE
      CONT=REDUC(HYMX+RESTY)
      PRINT 102,CONT,(PS(I),I=1,NHI)
      PRINT 107
   20 HYMX=HYMX-DY
      CONT=REDUC(HYMX+RESTY)
      DO 30 I=1,NHI
   30 PS(I)=ZLINE
      PRINT 102,CONT,(PS(I),I=1,NHI)
```

```
      PRINT 107
      DO 40 I=1,NBIN
      I1=(I+9)/10
      I2=I-(I1-1)*10
   40 CALL MOVC(I1,NUM,I2,PS(I1))
      PRINT 107
      CONT=DUM(3)
      PRINT 106,CONT,(PS(I),I=1,NHI)
      DO 45 I=1,NBIN
      I1=(I+9)/10
      I2=I-(I1-1)*10
   45 CALL MOVC(I2,NUM,I2,PS(I1))
      CONT=UMB
      PRINT 106,CONT,(PS(I),I=1,NHI)
      IF (NLIN.LT.0) GO TO 70
      PRINT 104
      CALL WRNR(XAXIS(1),NBIN)
      DO 50 J=1,NPLT
      PRINT 103, PSYMBL(J)
      DO 55 I=1,NBIN
   55 YAXIS(I,J)=REDUC(YAXIS(I,J)+RESTY)
   50 CALL WRNR(YAXIS(1,J),NBIN)
   70 IF (IGROS.EQ.0) GO TO 100
      IGROS=IGROS-1
      GO TO 19
  100 RETURN
  101 FORMAT(1H+,19X,10A10)
  102 FORMAT(1H+,F12.3,7X,10A10)
  103 FORMAT(1H0,*CONTENTS OF CHANNELS FOR HISTOGRAM*3X,A1,3X,
     A  *FOLLOW*/)
  104 FORMAT(*0CHANNEL VALUES FOR X AXIS*/)
  105 FORMAT(1H1)
  106 FORMAT(10X,A5,5X,10A10)
  107 FORMAT(1H )
      END
      SUBROUTINE WRNR(X,N)
      DIMENSION X(1)
      DIMENSION IX(100)
      COMMON/PLTWR/NUM    ,KX(10),MINUS,MESAG(4),NHI,NUMB,RESTY
      DATA NUM/10H0123456789/,NUMB/4HNUMB/
      DATA MESAG/1H ,4HCONT,3H OF,4HCHAN/,IDOT/1H./
      DATA INF/10HRRRRRRRRRR/,IZERO/10HZZZZZZZZZZ/
      XMAX=ABS(X(1))
      DO 10 I=2,N
   10 XMAX=AMAX1(XMAX,ABS(X(I)))
      IF (XMAX.LT.1.E-6) GO TO 150
      IF (XMAX.GE.1.E6) GO TO 160
      X1=ALOG10(XMAX)
      NLIN=6
      IPNT=(X1+ABS(X1))/2. +1.
      NPWR=NLIN-IPNT
      DO 5 I=1,NHI
    5 KX(I)=MESAG(1)
      DO 11 I=1,N
      IF (X(I).GE.0.0) GO TO 11
      I1=(I+9)/10
      I2=I-(I1-1)*10
      CALL MOVC(1,MINUS,I2,KX(I1))
   11 CONTINUE
      MES1=MESAG(1)
      MES2=MES1
      PRINT 1010,MES1,MES2,(KX(I),I=1,NHI)
      DO 20 I=1,N
   20 IX(I)=ABS(X(I))*10.**NPWR+.5
   21 KS=0
      DO 22 I=1,N
```

```
      KXI=MOD(IX(I),10)                                        WRNR
   22 KS=KS+KXI                                                WRNR
      IF (KS.NE.0.OR.NPWR.EQ.0) GO TO 25                       WRNR
      NPWR=NPWR-1                                              WRNR
      NLIN=NLIN-1                                              WRNR
      DO 23 I=1,N                                              WRNR
   23 IX(I)=IX(I)/10                                           WRNR
      GO TO 21                                                 WRNR
   25 CONTINUE                                                 WRNR
      MPNT=NPWR/3                                              WRNR
      NPWR=NLIN-1                                              WRNR
      LIN10=10**NLIN                                           WRNR
      DO 50 J=1,NLIN                                           WRNR
      LTEN=10**NPWR                                            WRNR
      NTEN=LTEN*10                                             WRNR
      MES1=MESAG(1)                                            WRNR
      MES2=MES1                                                WRNR
      DO 30 I=1,N                                              WRNR
      IXI=IX(I)                                                WRNR
      KXI=IXI/LTEN                                             WRNR
      I1=(I+9)/10                                              WRNR
      I2=I-(I1-1)*10                                           WRNR
      IF (KXI.GT.0.OR.IPNT.LE.J) GO TO 35                      WRNR
      CALL MOVC(1,MESAG(1),I2,KX(I1))                          WRNR
      GO TO 30                                                 WRNR
   35 IND=KXI+1 -(IXI/NTEN)*10                                 WRNR
      CALL MOVC(IND,NUM,I2,KX(I1))                             WRNR
   30 CONTINUE                                                 WRNR
      JPNT=J-MPNT                                              WRNR
      IF (JPNT.GT.3) GO TO 51                                  WRNR
      MES1=MESAG(JPNT+1)                                       WRNR
   51 IF (J.EQ.IPNT) MES2=IDOT                                 WRNR
      PRINT 1010,MES1,MES2,(KX(I),I=1,NHI)                     WRNR
   50 NPWR=NPWR-1                                              WRNR
      IF (NLIN.GE.3) GO TO 100                                 WRNR
      DO 65 J=NLIN,2                                           WRNR
      DO 67 I=1,NHI                                            WRNR
   67 KX(I)=MESAG(1)                                           WRNR
      MES1=MESAG(J+2)                                          WRNR
      MES2=MESAG(1)                                            WRNR
   65 PRINT 1010,MES1,MES2,(KX(I),I=1,NHI)                     WRNR
  100 RETURN                                                   WRNR
  150  DO 151 I=1,NHI                                          WRNR
  151 KX(I)=IZERO                                              WRNR
      GO TO 170                                                WRNR
  160 DO 161 I=1,NHI                                           WRNR
  161 KX(I)=INF                                                WRNR
  170 MES1=MESAG(2)                                            WRNR
      MES2=MESAG(1)                                            WRNR
      PRINT 1010,MES1,MES2,(KX(I),I=1,NHI)                     WRNR
      GO TO 100                                                WRNR
 1010 FORMAT(10X,A5,4X,A1,10A10)                               WRNR
      END                                                      WRNR
      SUBROUTINE MOVC(L1,W1,L2,W2)                             CAREFIX
      DIMENSION MASK(10)                                       CAREFIX
      DATA IBLNK/10H          /                                CAREFIX
      DATA MASK/77000000000000000000B,00770000000000000000B,   CAREFIX
     100007700000000000000B,00000077000000000000B,             CAREFIX
     200000000770000000000B,00000000007700000000B,             CAREFIX
     300000000000077000000B,00000000000000770000B,             CAREFIX
     400000000000000007700B,00000000000000000077B/             CAREFIX
      IDATA=AND(MASK(L1),W1)                                   CAREFIX
      IF(L2.GT.L1)  ICNT=60-(L2-L1)*6                          CAREFIX
      IF(L2.LE.L1)  ICNT=6*(L1-L2)                             CAREFIX
      W2=AND(-MASK(L2),W2)                                     CAREFIX
      W2=OR(W2,ISHIFT(IDATA,ICNT))                             CAREFIX
```

```
      RETURN                                                            MOVC
      END                                                               MOVC
      FUNCTION REDUC(X)                                                 REDUC
      LOGICAL LOGN                                                      REDUC
      COMMON /FUN/LOGN                                                  REDUC
      REDUC=X                                                           REDUC
      IF(LOGN) REDUC=(10.)**X                                           REDUC
      RETURN                                                            REDUC
      END                                                               REDUC
      SUBROUTINE ROMBD(A,B,X2,FOFX,HSTAR,HMIN,HMAX,ERMAX,ANS,K,KEY)     ROMBD
C                                                                       ROMBD
C          THE DOCUMENTATION OF THIS SUBROUTINE IS ENTITLED             ROMBD
C      ROMBERG QUADRATURE SUBROUTINES FOR SINGLE AND MULTIPLE INTEGRALS ROMBD
C           SECTION 314 TECH. MEMO NO. 221, JET PROPULSION LAB.         ROMBD
C                              1 JULY, 1969                             ROMBD
C                BY  W. BUNTON, M. DIETHELM, K.HAIGLER                  ROMBD
C                                                                       ROMBD
C                                                                       ROMBD
C     H = CURRENT STEP SIZE                                             ROMBD
C     NAB= 0, H .GT. HMIN,     =1, THEN H .LE. HMIN                     ROMBD
C     NFG= 0  MAY DOUBLE STEP,  =-1 CANT DOUBLE AFTER PREVIOUS CUT      ROMBD
C     K=1, INTEGRATE FROM A TO B,   =2, FINISHED INTEGRATION            ROMBD
C     I = COUNTER ON I TH FUNCTIONAL EVALUATION                         ROMBD
C     X2= INDEPENDENT VARIABLE                                          ROMBD
C     LL=1,  TAKE 16 OR 17 FUNCTIONAL EVALUATIONS                       ROMBD
C     LL=2,  TAKE ONLY 8 FUNCTIONAL EVALUATIONS                         ROMBD
C     X1 = FIRST X VALUE OF EACH SUBINTERVAL                            ROMBD
C     KEY= FLAG ON THE DIAGNOSTIC PRINT                                 ROMBD
C                                                                       ROMBD
      DIMENSION          Y(17),T(5,5)                                   ROMBD
C     FOFX HAS BEEN DIMENSIONED TO OBTAIN PROPER EXECUTION ON THE UNIVACROMBD
C     1108.  IT IS NOT NECESSARY FOR THE USER TO DIMENSION FOFX IN HIS  ROMBD
C     CALLING PROGRAM                                                   ROMBD
      INTEGER FINISH                                                    ROMBD
      IF(A.GE.B.OR.HMIN.GT.HSTAR.OR.HMIN.LE.0..OR.HMAX.LT.HMIN.OR.ERMAX.ROMBD
     *LE.0..OR.HSTAR.GT.HMAX)GO TO 200                                  ROMBD
C                                                                       ROMBD
C     INITIALIZE STARTING VALUES                                        ROMBD
C                                                                       ROMBD
      ANS = 0.                                                          ROMBD
      NAB = 0                                                           ROMBD
      NFG = 0                                                           ROMBD
      X1 = A                                                            ROMBD
      H= HSTAR                                                          ROMBD
      K = 1                                                             ROMBD
      I = 0                                                             ROMBD
      X2 = X1                                                           ROMBD
      LL=1                                                              ROMBD
      FINISH = 0                                                        ROMBD
      IF(X1+16.*H.LT.B)GO TO 10                                         ROMBD
      H=(B-A)/16.                                                       ROMBD
      FINISH=1                                                          ROMBD
   10 RETURN                                                            ROMBD
      ENTRY ROM2D                                                       ROMBD
C                                                                       ROMBD
C     STORE FUNCTIONAL EVALUATIONS IN Y ARRAY                           ROMBD
C                                                                       ROMBD
      IF( LL.GT.1 ) GO TO 140                                           ROMBD
      I = I + 1                                                         ROMBD
      Y(I)=FOFX                                                         ROMBD
      IF(I.EQ.17) GO TO 20                                              ROMBD
      X2 = X2 +H                                                        ROMBD
      IF(I.EQ.16.AND.FINISH.EQ.1)X2=B                                   ROMBD
      RETURN                                                            ROMBD
C                                                                       ROMBD
C     CALCULATE T VALUES                                                ROMBD
```

```
C                                                                          ROMBD
   20 TEMP= .5*(Y(1)+Y(17))                                                ROMBD
      T(1,1)= TEMP*16.*H                                                   ROMBD
      TEMP=TEMP+Y(9)                                                       ROMBD
      T(1,2)= TEMP*8.*H                                                    ROMBD
      TEMP=TEMP+Y(5)+Y(13)                                                 ROMBD
      T(1,3)= TEMP*4.*H                                                    ROMBD
      TEMP=TEMP+Y(3)+Y(15)+Y(7)+Y(11)                                      ROMBD
      T(1,4)= TEMP*2.*H                                                    ROMBD
      TEMP=TEMP+Y(2)+Y(16)+Y(4)+Y(14)+Y(6)+Y(12)+Y(8)+Y(10)                ROMBD
      T(1,5)= TEMP*H                                                       ROMBD
      DO 30 I=2,5                                                          ROMBD
      LUP= 6-I                                                             ROMBD
      DO 30 J=1,LUP                                                        ROMBD
      T(I,J)=T(I-1,J+1)-(T(I-1,J)-T(I-1,J+1))/(4.**(I-1) -1.)              ROMBD
   30 CONTINUE                                                             ROMBD
      IF( ABS(T(5,1)) + ABS(ANS) .EQ. 0.) GO TO 110                        ROMBD
C                                                                          ROMBD
C     TEST D.1, RELATIVE ERROR OVER INTERVAL = H                           ROMBD
C                                                                          ROMBD
      IF(  ABS(T(5,1)) .EQ. 0.) GO TO 40                                   ROMBD
      DENOM= T(5,1)                                                        ROMBD
      IF(  ABS((T(5,1)-T(4,2))/DENOM ) .LE. ERMAX) GO TO 50                ROMBD
      IF( ABS(T(5,1)-T(4,2)).LE.ERMAX)GO TO 50                             ROMBD
C                                                                          ROMBD
C     TEST D.2, RELATIVE ERROR OVER INTERVAL (A,X1)                        ROMBD
   40 IF(ANS.EQ.0.) GO TO 120                                              ROMBD
      DENOM= ANS                                                           ROMBD
      IF( ABS((T(5,1)-T(4,2))/DENOM).GT.ERMAX.AND. ABS(T(5,1)-T(4,2)).GTROMBD
     *.ERMAX)GO TO 120                                                     ROMBD
C                                                                          ROMBD
C     ALL TESTS PASSED, ADD VALUE OF  SUBINTERVAL INTO SUM                 ROMBD
C                                                                          ROMBD
   50 ANS= ANS +T(5,1)                                                     ROMBD
   60 IF(FINISH.EQ.1)GO TO 180                                             ROMBD
      X1=X1+16.*H                                                          ROMBD
C                                                                          ROMBD
C     TEST ON WHETHER TO DOUBLE H OR NOT TO                                ROMBD
C     WHEN NFG= 0, CAN DOUBLE ANTTIME                                      ROMBD
      IF(NFG) 90,70,70                                                     ROMBD
   70 IF( ABS((T(5,1)-T(4,1))/DENOM).GT.ERMAX.AND. ABS(T(5,1)-T(4,1)).GTROMBD
     *.ERMAX) GO TO 100                                                    ROMBD
C                                                                          ROMBD
C     STEP SIZE H CAN NOW BE DOUBLED                                       ROMBD
C                                                                          ROMBD
   80 H= 2.*H                                                              ROMBD
      NAB=0                                                                ROMBD
      IF(H.GT.HMAX) H=HMAX                                                 ROMBD
   90 NFG=0                                                                ROMBD
  100 IF(X1+16.*H.LT.B)GO TO 95                                            ROMBD
      H=(B-X1)/16.                                                         ROMBD
      FINISH=1                                                             ROMBD
      IF(H)180,180,95                                                      ROMBD
   95 CONTINUE                                                             ROMBD
      Y(1)= Y(17)                                                          ROMBD
      X2 = X1 + H                                                          ROMBD
      I = 1                                                                ROMBD
      GO TO 10                                                             ROMBD
C                                                                          ROMBD
C     TRYING TO FIND THE FUNCTION                                          ROMBD
  110 X1= X1 +16.*H                                                        ROMBD
      IF(FINISH.EQ.1)GO TO 180                                             ROMBD
      GO TO 100                                                            ROMBD
C                                                                          ROMBD
C     CUT STEP SIZE IN HALF                                                ROMBD
  120 H= H*.5                                                              ROMBD
```

```
      NFG##1
      IF(H.GT.HMIN) GO TO 130
      IF(NAB.EQ.1) GO TO 160
C
C     HMIN HAS BEEN REACHED
C
      NAB=1
      H=HMIN
      X2 = X1 + H
      I = 1
      GO TO 10
C
C     IN H CUT, SAVE VALUABLE INFO
C
  130 Y(17)= Y(9)
      Y(15)= Y(8)
      Y(13)= Y(7)
      Y(11)= Y(6)
      Y(9)= Y(5)
      Y(7)= Y(4)
      Y(5)= Y(3)
      Y(3)= Y(2)
      I=0
      LL = 2
      X2 = X1 +H
      FINISH=0
      RETURN
C
C     NOW PICK UP ONLY THOSE FUNCTIONAL EVALUATIONS NECESSARY
C
  140 I= I+2
      Y(I)=FOFX
      IF( I .GE. 16) GO TO 150
      X2= X2 +2.*H
      RETURN
  150 LL=1
      GO TO 20
C
C     BOTH TEST D.1 AND D.2 HAVE FAILED, ACCEPT ANSWER
C     AND PROCEED WITH INTEGRATION
C
  160 H= HMIN
      IF(KEY.NE.7) WRITE (6,170) X1,X2
  170 FORMAT(16H IN INTERVAL X= E18.8,7H TO X= E18.8,56H RELATIVE ERROR
     1CANNOT BE REACHED WITH MINIMUM STEP SIZE  )
      GO TO 50
  180 K = 2
      RETURN
  200 WRITE(6,201)A,B,HSTAR,HMIN,HMAX,ERMAX
  201 FORMAT(10X,36HERROR ROMBD,CHECK CALLING PARAMETERS//10X,3HA =,
     *E15.8,5X,3HB =,E15.8,5X,8HHSTART =,E15.8,5X,6HHMIN =,E15.8//10X
     *,6HHMAX =,E15.8,5X,7HERMAX =,E15.8)
      STOP
      END
      SUBROUTINE RIFDIF (I7,OPTION,IRES,N7,TIN,MIN,MAX,STEP,JDIM,R,
     *  PDIFF,PRIF,PGAIN,NG,LPLOT,GPLOT)
C
C        **********************************
C        *                                *
C        * THIS SUBROUTINE COMPUTES THE   *
C        * DIFFERENCE = DIFF, RELATIVE    *
C        * IMPROVEMENT FACTOR = RIF, AND  *
C        * GAIN BETWEEN TWO VALUES OF     *
C        * THE FAMILY OF PARAMETERS       *
C        *                                *
C        **********************************
```

```
C                                                                        RIFDIF    13
           DIMENSION PARAM(16,10 )                                       RIFDIF    14
      REAL PDIFF(1,1),PRIF(1,1),PGAIN(1,1),R(121,1),RF(2)                CRUSH      1
      REAL ENC(20),AB,DIFF,RIF,GAIN,INFI,NITY                            RIFDIF    16
      REAL MIN,MAX                                                       RIFDIF    17
      INTEGER PROD(10),PAR(6),OPTION,NPROD                               RIFDIF    18
      LOGICAL TIN,LPLOT,GPLOT,LRIF                                       RIFDIF    19
      LOGICAL PRODT                                                      RIFDIF    20
      LOGICAL OUTPT1,OUTPT2,OUTPT3,OUTPT4,OUTPT5                         RIFDIF    21
      COMMON/GRAPH/ NEQ,PROD,NPROD,PAR,ENC,AB,PRODT                      RIFDIF    22
      COMMON /PARM/ PARAM,N10,K1                                         RIFDIF    23
      COMMON /OUTPT/ OUTPT1,OUTPT2,OUTPT3,OUTPT4,OUTPT5                  RIFDIF    24
      DATA ITEST1/1/                                                     RIFDIF    25
      DATA INFI/4HINFI/,NITY/4HNITY/                                     RIFDIF    26
      GPLOT=.FALSE.                                                      RIFDIF    27
      IF (OPTION.EQ.1) GO TO 30                                          RIFDIF    28
      IF (OPTION.NE.2) GO TO 10                                          RIFDIF    29
      IR2=I7                                                             RIFDIF    30
      IR1=I7-1                                                           RIFDIF    31
      NG=NG+1                                                            RIFDIF    32
      GO TO 60                                                           RIFDIF    33
   10 WRITE (6,1000) PAR                                                 RIFDIF    34
   20 READ (5,1001)                  IR1,IR2                             RIFDIF    35
   21 IF (OUTPT3) WRITE (6,1009) IR1,IR2                                 RIFDIF    36
      NG=NG+1                                                            RIFDIF    37
      IF (IR1.LE.I7.AND.IR2.LE.I7) GO TO 60                              RIFDIF    38
      NG=NG-1                                                            RIFDIF    39
      WRITE (6,1003)                                                     RIFDIF    40
      GO TO 20                                                           RIFDIF    41
   30 IR1=0                                                              RIFDIF    42
   40 IR1=IR1+1                                                          RIFDIF    43
      IF (IR1.EQ.N7) GO TO 130                                           RIFDIF    44
      NG=NG+1                                                            RIFDIF    45
      IR2=IR1                                                            RIFDIF    46
   50 IR2=IR2+1                                                          RIFDIF    47
      IF (IR2.GT.N7) GO TO 40                                            RIFDIF    48
   60 IF (TIN) GO TO 70                                                  RIFDIF    49
      IF (OUTPT2) WRITE (6,1004) PAR,PARAM(IR1,K1),PAR,PARAM(IR2,K1),    RIFDIF    50
     *  NEQ,AB                                                           RIFDIF    51
      GO TO 80                                                           RIFDIF    52
   70 IF (OUTPT2) WRITE (6,1005) PAR,PARAM(IR1,K1),PAR,PARAM(IR2,K1),    RIFDIF    53
     *  NEQ,AB                                                           RIFDIF    54
   80 CT=MIN                                                             RIFDIF    55
      J=1                                                                RIFDIF    56
      LRIF=.TRUE.                                                        RIFDIF    57
   90 DIFF=R(J,IR2)-R(J,IR1)                                             RIFDIF    58
      RIF=(1.0  -R(J,IR1))/(1.0  -R(J,IR2))                              RIFDIF    59
      IF (R(J,IR2).NE.1.0  ) GO TO 91                                    RIFDIF    60
      ENCODE(8,1012,RF(1)) INFI,NITY                                     RIFDIF    61
      RIF=1.0E35                                                         RIFDIF    62
      GO TO 910                                                          RIFDIF    63
   91 ENCODE(14,1013,RF(1)) RIF                                          RIFDIF    64
  910 GAIN=R(J,IR2)/R(J,IR1)                                             RIFDIF    65
      IF (OUTPT2) WRITE (6,1006) CT,DIFF,(RF(I),I=1,2),GAIN              RIFDIF    66
      IF (OPTION.EQ.0.AND.LPLOT) GO TO 92                                RIFDIF    67
      IF (IR2.NE.(IR1+1).OR..NOT.LPLOT) GO TO 96                         RIFDIF    68
C                                                                        RIFDIF    69
C        *** SET UP ARRAYS FOR DIFF, RIF, AND GAIN PLOTS ***              RIFDIF    70
   92 PDIFF(J,NG)=DIFF                                                   RIFDIF    71
      PRIF(J,NG)=RIF                                                     RIFDIF    72
      IF (LRIF) GO TO 94                                                 RIFDIF    73
C                                                                        RIFDIF    74
C         INSTEAD OF PLOTTING POINT WITH VALUE OF 1.0E35                 RIFDIF    75
C         FOR RIF, PLOT 3.0TIMES THE VALUE OF THE NEXT POINT             RIFDIF    76
C         TO THE RIGHT FOR THAT VALUE                                    RIFDIF    77
      PRIF(J-1,NG)= 3.0*PRIF(J,NG)                                       RIFDIF    78
```

```
      LRIF=.TRUE.                                                       RIFDIF
   94 IF (R(J,IR2).EQ.1.0  ) LRIF=.FALSE.                               RIFDIF
      PGAIN(J,NG)=GAIN                                                  RIFDIF
C                                                                       RIFDIF
      GPLOT=.TRUE.                                                      RIFDIF
   96 J=J+1                                                             RIFDIF
      CT=CT+STEP                                                        RIFDIF
      IF (CT.LE.MAX) GO TO 90                                           RIFDIF
      IF (OPTION.EQ.1) GO TO 50                                         RIFDIF
  130 RETURN                                                            RIFDIF
C                                                                       RIFDIF
C            FORMAT STATEMENTS                                          RIFDIF
C                                                                       RIFDIF
 1000 FORMAT (* IF DIFF, RIF, AND GAIN ARE WANTED FOR PARAMETER *,6A1,/,RIFDIF
     ** INPUT R1 IN COLUMNS 1-2 AND R2 IN COLUMNS 3-4*,/,               RIFDIF
     ** OTHERWISE INPUT NO IN COLUMNS 1-2*)                             RIFDIF
 1001 FORMAT (2I2)                                                      RIFDIF
 1003 FORMAT (* R1 OR R2 HAS NOT BEEN CALCULATED YET - TRY AGAIN*)      RIFDIF
 1004 FORMAT(///* DIFF, RIF, AND GAIN FOR *,6A1,* = *,G14.7,/,          RIFDIF
     *    * AND *,6A1,* = *,G14.7,*    FOLLOWS FOR EQUATION *,I2,A6,//, RIFDIF
     * 11X,*LAMT*,9X,*DIFF*,15X,*RIF*,20X,*GAIN*)                       RIFDIF
 1005 FORMAT(///* DIFF, RIF, AND GAIN FOR *,6A1,* = *,G14.7,/,          RIFDIF
     *    * AND *,6A1,* = *,G14.7,*    FOLLOWS FOR EQUATION *,I2,A6,//, RIFDIF
     *  12X,*T*,11X,*DIFF*,15X,*RIF*,20X,*GAIN*)                        RIFDIF
 1006 FORMAT( 1H ,6X,F9.5,3X,E14.6,4X,2A10,4X,E14.6)                    RIFDIF
 1009 FORMAT (1H ,2I2)                                                  RIFDIF
 1012 FORMAT(2A4)                                                       RIFDIF
 1013 FORMAT(    E14.6)                                                 RIFDIF
C                                                                       RIFDIF
C                                                                       RIFDIF
      END                                                               RIFDIF
      SUBROUTINE RELEQS (NEQ,BCOMP,RL,RS,S,N,K,Q,C,RV,Z,W,P,LAMT,RDP)   RELEQS
      INTEGER S,N,Z,W,NEQ                                               RELEQS
      LOGICAL BCOMP                                                     RELEQS
      REAL K ,LAMT                                                      RELEQS
C                                                                       RELEQS
C       *****************************************                       RELEQS
C       *                                       *                       RELEQS
C       * THIS SUBROUTINE CALLS THE RELIABILITY *                       RELEQS
C       * SUBROUTINE SPECIFIED BY NEQ           *                       RELEQS
C       *                                       *                       RELEQS
C       *****************************************                       RELEQS
C                                                                       RELEQS
      GO TO (10,30,50,60,80,90,100) ,NEQ                                RELEQS
   10 IF (BCOMP) GO TO 20                                               RELEQS
C                                                                       RELEQS
C         COMPUTE RELIABILITY FOR EQUATION 1A                           RELEQS
      CALL NEQ1A (RL,RS,S,N,K,RV,Z,W,P,LAMT,RDP)                        RELEQS
      GO TO 110                                                         RELEQS
C                                                                       RELEQS
C         COMPUTE RELIABILITY FOR EQUATION 1B                           RELEQS
   20 CALL NEQ1B (RL,S,N,RV,Z,W,P,LAMT,RDP)                             RELEQS
      GO TO 110                                                         RELEQS
C                                                                       RELEQS
   30 IF (BCOMP) GO TO 40                                               RELEQS
C                                                                       RELEQS
C         COMPUTE RELIABILITY FOR EQUATION 2A                           RELEQS
      CALL NEQ2A (RL,RS,S,K,Q,RV,Z,W,P,LAMT,RDP,C)                      RELEQS
      GO TO 110                                                         RELEQS
C                                                                       RELEQS
C         COMPUTE RELIABILITY FOR EQUATION 2B                           RELEQS
   40 CALL NEQ2B (RL,LAMT,S,RV,Z,W,P,RDP,C,Q)                           RELEQS
      GO TO 110                                                         RELEQS
C                                                                       RELEQS
C         COMPUTE RELIABILITY FOR EQUATION 3                            RELEQS
   50 CALL NEQ3 (RL,RS,S,K,Q,C,RV,Z,W,P,BCOMP,LAMT,RDP)                 RELEQS
```

```
      GO TO 110                                                         RELEQS
C                                                                       RELEQS
   60 IF (BCOMP) GO TO 70                                               RELEQS
C                                                                       RELEQS
C          COMPUTE RELIABILITY FOR EQUATION 4A                          RELEQS
      CALL NEQ4A (RL,RS,S,K,RV,Z,W,P,LAMT,RDP)                          RELEQS
      GO TO 110                                                         RELEQS
C                                                                       RELEQS
C          COMPUTE RELIABILITY FOR EQUATION 4B                          RELEQS
   70 CALL NEQ4B (RL,S,RV,Z,W,P,LAMT,RDP)                               RELEQS
      GO TO 110                                                         RELEQS
C                                                                       RELEQS
C          COMPUTE RELIABILITY FOR EQUATION 5                           RELEQS
   80 CALL NEQ5 (RL,RV,Z,W,P,LAMT,RDP)                                  RELEQS
      GO TO 110                                                         RELEQS
C                                                                       RELEQS
C          COMPUTE RELIABILITY FOR EQUATION 6                           RELEQS
   90 CALL NEQ6 (RL,Z,W,P,LAMT,RDP)                                     RELEQS
      GO TO 110                                                         RELEQS
C                                                                       RELEQS
C          COMPUTE RELIABILITY FOR EQUATION 7                           RELEQS
  100 CALL NEQ7 (RL,RS,S,N,K,Q,C,RV,Z,W,P,BCOMP,LAMT,RDP)               RELEQS
  110 RETURN                                                            RELEQS
      END                                                               RELEQS
      SUBROUTINE ROWPLT                                                 ROWPLT
C        **************************                                     ROWPLT
C        *                        *                                     ROWPLT
C        * THIS SUBROUTINE LABELS *                                     ROWPLT
C        * THE PLOTS GENERATED.   *                                     ROWPLT
C        *                        *                                     ROWPLT
C        **************************                                     ROWPLT
      REAL ENC(20)                                                      ROWPLT
      INTEGER NEQ,PROD(10 ),NPROD,PAR(6)                                ROWPLT
      LOGICAL PRODT                                                     ROWPLT
      COMMON/GRAPH/NEQ,PROD,NPROD,PAR,ENC,AB,PRODT                      ROWPLT
      COMMON /PARM/ PARAM,N10,K1                                        ROWPLT
              DIMENSION PARAM(16,10 )                                   ROWPLT
      IF (PRODT) GO TO 10                                               ROWPLT
      WRITE(6,1001) NEQ                                                 ROWPLT
      WRITE(6,1002) PAR,(PARAM(I,K1),I=1,N10)                           ROWPLT
      GO TO 20                                                          ROWPLT
   10 WRITE(6,1006) (PROD(I),I=1,NPROD)                                 ROWPLT
   20 WRITE(6,1003)                                                     ROWPLT
      WRITE(6,1004)                                                     ROWPLT
      WRITE(6,1005) (ENC(I),I=1,20)                                     ROWPLT
      RETURN                                                            ROWPLT
C                                                                       ROWPLT
C            FORMAT STATEMENTS                                          ROWPLT
C                                                                       ROWPLT
 1001 FORMAT(//60X, * PLOT FOR EQUATION *,I2)                           ROWPLT
 1002 FORMAT (20X,*FAMILY OF PARAMETERS IS *,6A1,* = *,5(G13,8,2X),/,   ROWPLT
     *  (40X,6(G13,8,2X)))                                              ROWPLT
 1003 FORMAT(20X,*(NI MEANS NOT INPUTED OR DEFAULT VALUE USED)*)        ROWPLT
 1004 FORMAT(3X,*LAMBDA*,9X,*MU*,11X,*S*,5X,*N*,6X,*K*,12X,*Q*,         ROWPLT
     *  12X,*C*,11X,*RV*,12X,*Z*,5X,*W*,6X,*P*,11X,*MUT*)               ROWPLT
 1005 FORMAT (1H ,10(2A6,1X))                                           ROWPLT
 1006 FORMAT (67X,*PLOT FOR EQUATIONS *,10(I2,2X))                      ROWPLT
C                                                                       ROWPLT
C                                                                       ROWPLT
      END                                                               ROWPLT
      SUBROUTINE SIMPR1 (TIN,BCOMP,LAMBDA,S,N,K,Q,C,RV,Z,W,P,NEQ,       SIMPR1
     *  R2,R1MIN,R1MAX,R1STEP,PLOTR1,OPTR1,I10,TMAX)                    SIMPR1
C        *********************************************                  SIMPR1
C        *                                           *                  SIMPR1
C        * THIS SUBROUTINE COMPUTES THE MAX. MISSION *                  SIMPR1
C        * TIME (TMAX) AND THE SIMPLEX MISSION TIME  *                  SIMPR1
```

```
C        * (SIMTMAX) AND THE RATIO OF THESE (SIMTIF) *                      SIMPR1
C        *                                           *                      SIMPR1
C        *********************************************                      SIMPR1
      REAL SIMR1(101,4),XAXIS(101),SIMTMX(101),SIMTIF(101),TM(101)          SIMPR1
      REAL TMAX(101,15),TX(4),INFI,NITY                                     SIMPR1
      REAL LAMBDA,K,LAMTR2,LAMTR1,INF                                       SIMPR1
      INTEGER S,N,Z,W,NEQ                                                   SIMPR1
      LOGICAL TIN,BCOMP,PLOTR1,FIRST,SEC,ZERO,OPTR1                         SIMPR1
      LOGICAL OUTPT1,OUTPT2,OUTPT3,OUTPT4,OUTPT5                            SIMPR1
      EQUIVALENCE (SIMR1(1,1),XAXIS(1)),  (SIMR1(1,2),SIMTMX(1)),           SIMPR1
     *            (SIMR1(1,3),SIMTIF(1)), (SIMR1(1,4),TM(1))                SIMPR1
      COMMON /OUTPT/ OUTPT1,OUTPT2,OUTPT3,OUTPT4,OUTPT5                     SIMPR1
      COMMON/PPLOT/SIMR1                                                    SIMPR1
      DATA FIRST/.TRUE./, SEC/.FALSE./, INF/1.0E35/                         SIMPR1
      DATA INFI/4HINFI/,NITY/4HNITY/                                        SIMPR1
      ZERO=.FALSE.                                                          SIMPR1
      IF (.NOT.(OPTR1.AND.OUTPT5)) GO TO 20                                 SIMPR1
      IF (TIN) GO TO 10                                                     SIMPR1
      WRITE (6,1001) R2                                                     SIMPR1
      GO TO 20                                                              SIMPR1
   10 WRITE (6,1002) R2                                                     SIMPR1
   20 CALL BISECT(NEQ,BCOMP,S,N,K,Q,C,RV,Z,W,P,FIRST,LAMTR2,R2)             SIMPR1
      R1=R1MIN                                                              SIMPR1
      J=0                                                                   SIMPR1
   40 J=J+1                                                                 SIMPR1
      XAXIS(J)=R1                                                           SIMPR1
      CALL BISECT(NEQ,BCOMP,S,N,K,Q,C,RV,Z,W,P,SEC,LAMTR1,R1)               SIMPR1
      IF (LAMTR1.NE.INF) GO TO 50                                           SIMPR1
      ENCODE(8,1005,TX(1)) INFI,NITY                                        SIMPR1
      ENCODE(8,1005,TX(3)) INFI,NITY                                        SIMPR1
      SIMTIF(J)=1.0                                                         SIMPR1
      ZERO=.TRUE.                                                           SIMPR1
      JJ=J                                                                  SIMPR1
      GO TO 84                                                              SIMPR1
   50 SIMTMX(J)=-ALOG(R1)                                                   SIMPR1
      IF (SIMTMX(J).GT.1.0E-8) GO TO 60                                     SIMPR1
      SIMTMX(J)=0.0                                                         SIMPR1
      TMAX(J,I10)=0.0                                                       SIMPR1
      SIMTIF(J)=1.0                                                         SIMPR1
      GO TO 80                                                              SIMPR1
   60 TMAX(J,I10)=LAMTR1-LAMTR2                                             SIMPR1
      IF (.NOT.TIN) GO TO 70                                                SIMPR1
      TMAX(J,I10)=TMAX(J,I10)/LAMBDA                                        SIMPR1
      SIMTMX(J)=SIMTMX(J)/LAMBDA                                            SIMPR1
   70 SIMTIF(J)=TMAX(J,I10)/SIMTMX(J)                                       SIMPR1
   80 TM(J)=TMAX(J,I10)                                                     SIMPR1
      ENCODE( 14,1006,TX(1)) SIMTMX(J)                                      SIMPR1
      ENCODE( 14,1006,TX(3)) TMAX(J,I10)                                    SIMPR1
   84 IF (OPTR1.AND.OUTPT5) WRITE (6,1003) R1,(TX(I),I=1,4),SIMTIF(J)       SIMPR1
      R1=R1+R1STEP                                                          SIMPR1
      IF (R1.LE.R1MAX) GO TO 40                                             SIMPR1
      IF (.NOT.ZERO) GO TO 88                                               SIMPR1
      TMAX(JJ,I10)=2.0*TMAX(JJ+1,I10)                                       SIMPR1
      SIMTMX(JJ)=2.0*SIMTMX(JJ+1)                                           SIMPR1
      TM(JJ)=TMAX(JJ,I10)                                                   SIMPR1
   88 IF (.NOT.PLOTR1) GO TO 90                                             SIMPR1
      CALL PLOTT(J,1)                                                       SIMPR1
      WRITE (6,1004)                                                        SIMPR1
   90 RETURN                                                                SIMPR1
 1001 FORMAT (//,6X,*MAXIMUM MISSION TIME*,4X,*REFERENCE R2 = *,F7.5,/,     SIMPR1
     *  4X,*R1*,9X,*SIMLAMTMAX*,16X,*LAMTMAX*,16X,*SIMTIF*)                 SIMPR1
 1002 FORMAT (//,6X,*MAXIMUM MISSION TIME*,4X,*REFERENCE R2 = *,F7.5,/,     SIMPR1
     *  4X,*R1*,8X,*SIMTMAX*,17X,*TMAX*,17X, *SIMTIF*)                      SIMPR1
 1003 FORMAT(1H ,F7.5,2(4X,2A10),3X,E14.7)                                  SIMPR1
 1004 FORMAT(/36X,* SIMLAMTMAX(A),SIMTIF(B),LAMTMAX(C) VS R1(XAXIS)  PLOSIMPR1
     3T COMPLETED *)                                                        SIMPR1
```

```
 1005 FORMAT(2A4)                                                         SIMPR1    73
 1006 FORMAT(E14.7)                                                       SIMPR1    74
      END                                                                 SIMPR1    75
      SUBROUTINE SIMPLE (ELATIN,TLAMT,RL,SIMPX,SIMRF,SG,Q1,R)             SIMPLE     1
C                                                                         SIMPLE     2
C        ***********************************                              SIMPLE     3
C        *                                 *                              SIMPLE     4
C        * THIS SUBROUTINE COMPUTES THE    *                              SIMPLE     5
C        * UNRELIABILITY       = Q1,       *                              SIMPLE     6
C        * SIMPLE RELIABILITY = SIMPX,     *                              SIMPLE     7
C        * SIMPLE RIF          = SIMRF, AND *                             SIMPLE     8
C        * SIMPLE GAIN         = SG        *                              SIMPLE     9
C        *                                 *                              SIMPLE    10
C        ***********************************                              SIMPLE    11
C                                                                         SIMPLE    12
      LOGICAL ELATIN                                                      SIMPLE    13
      Q1=1.0-R                                                            SIMPLE    14
      SIMPX= EXP(-TLAMT)                                                  SIMPLE    15
      IF (ELATIN) SIMPX=RL                                                SIMPLE    16
      SIMRF=1.0E35                                                        SIMPLE    17
      IF(R .EQ. 1.0) GO TO 10                                             SIMPLE    18
      SIMRF=(1.0-SIMPX)/(1.0-R)                                           SIMPLE    19
C                                                                         SIMPLE    20
C     *** WHEN DENOMINATOR IN SIMRF IS ZERO, SIMRF=1.0E35 ***             SIMPLE    21
   10 SG=R/SIMPX                                                          SIMPLE    22
      RETURN                                                              SIMPLE    23
      END                                                                 SIMPLE    24
      SUBROUTINEREADIN (T , LAMT, MUT, B, OPTION , NPROD, PROD, PRODT,    READIN     1
     *MUTIN,                                                              READIN     2
     *  NEQ,RVPLOT,PMIN,PMAX,PSTEP ,ELAMT,ELATIN,LPLOT,                   READIN     3
     *  DEBUG,INPOUT,STGOUT,LSTCH,COVPRC,DMFLG,NUN,                       MAINLOG    1
     *  R2,R1MIN,R1MAX,R1STEP,OPTR1,OPTR2,PLOTR1,PLOTR2,R2OPT)            READIN     4
C                                                                         READIN     5
C        ****************************************                         READIN     6
C        *                                      *                         READIN     7
C        * THIS SUBROUTINE READS IN AND CHECKS  *                         READIN     8
C        * DATA FOR THE RELIABILITY EQUATIONS   *                         READIN     9
C        * AND THE PLOTS AND WRITES INSTRUCTIONS *                        READIN    10
C        *                                      *                         READIN    11
C        ****************************************                         READIN    12
C                                                                         READIN    13
      REAL LAMT,MIN,MAX,LAMDEF,MUDEF                                      MAINLOG    2
      REAL MUT(10),MTF(16),FOFX(16)                                       READIN    17
      REAL DIFF(1,1),RIF(1,1),GAIN(1,1)                                   CRUSH      1
      REAL SIMPX(121 ),PDT(121 )                                          READIN    19
      INTEGER SDEF,WDEF,ZDEF                                              MAINLOG    3
      INTEGER PROD(10),OPTION,R2OPT                                       PARMOD     1
      LOGICAL PRODT,TIN,MUTIN,XYRANGE,LPLOT                               READIN    22
      LOGICAL ELATIN,RVPLOT,OPTR1,OPTR2,PLOTR1,PLOTR2                     READIN    23
      LOGICAL OUTPT1,OUTPT2,OUTPT3,OUTPT4,OUTPT5,DEF                      READIN    24
      LOGICAL DMFLG,DEBUG,INPOUT,STGOUT,COVINT,LSTCH,COVPRC,DEFCHNG       MAINLOG    4
      COMMON /PLOTIN/ TIN,MIN,MAX,STEP,DIFF,RIF,GAIN,MTF,FOFX,            READIN    25
     *  PDT,SIMPX,XMIN,XMAX,YMIN,YMAX,XYRANGE,IA,IAA,IB,IBB,              READIN    26
     *  IC,ICC,IPLT1,IPLT2,IPLT3,IPLT4,IPLT5                              READIN    27
      COMMON /OUTPT/ OUTPT1,OUTPT2,OUTPT3,OUTPT4,OUTPT5                   READIN    29
      COMMON /DEFS/PCW(22),NAMES(22),RLDFS(10),INTDFS(4)                  MAINLOG    5
      COMMON /CVBO/COVINT,IGENC(10),IGENP(10),IFSC(8),FRAC(8)             MAINLOG    6
      EQUIVALENCE (RLDFS(1),Q1DEF),(RLDFS(2),Q2DEF)                       MAINLOG    7
      EQUIVALENCE (RLDFS(3),LAMDEF),(RLDFS(4),MUDEF)                      MAINLOG    8
      EQUIVALENCE (RLDFS(5),GMPDEF),(RLDFS(6),CDEF)                       MAINLOG    9
      EQUIVALENCE (RLDFS(7),CTRDEF),(RLDFS(8),PRCDEF)                     MAINLOG   10
      EQUIVALENCE (RLDFS(9),RVDEF),(RLDFS(10),PDEF)                       MAINLOG   11
      EQUIVALENCE (INTDFS(1),NDEF),(INTDFS(2),SDEF)                       MAINLOG   12
      EQUIVALENCE (INTDFS(3),WDEF),(INTDFS(4),ZDEF)                       MAINLOG   13
      DATA CONV/1.745329251994329E-2/                                     READIN    30
      DATA ITEST1/1/,MB59/4000000000000000000B/                           MAINLOG   14
```

```
      DATA (IGENC(I),I=1,10)/10*0/, (IGENP(I),I=1,10)/10*0/             POSTPCH    1
      DATA (IFSC(I),I=1,8)/8*0/, (FRAC(I),I=1,8)/8*0.0/                 POSTPCH    2
      NAMELIST /VAR/T,LAMT,ELAMT,MUT,MIN,STEP,B,OPTION                  PARMOD     2
      NAMELIST /VAR1/ PROD                                              READIN    34
      NAMELIST/OPTSON/OUTPT1,OUTPT2,OUTPT3,OUTPT4,OUTPT5, PRODT,NEQ,    READIN    35
     *  DEBUG,INPOUT,STGOUT,COVINT,LSTCH,COVPRC,DEFCHNG,                MAINLOG   15
     * LPLOT,RVPLOT,OPTR1,OPTR2,PLOTR1,PLOTR2,IND,XYRANGE               READIN    36
      NAMELIST /DEFAULT/Q1DEF,Q2DEF,LAMDEF,MUDEF,GMPDEF,CDEF,           MAINLOG   16
     *  CTRDEF,PRCDEF,RVDEF,PDEF,NDEF,SDEF,WDEF,ZDEF                    MAINLOG   17
      NAMELIST /COVCAL/COVINT,IGENC,IGENP,IFSC,FRAC                     MAINLOG   18
      IND=0                                                             READIN    37
      NPT=10                                                            READIN    38
      T=0.0                                                             READIN    39
      LAMT=0.0                                                          READIN    40
      ELAMT=0.0                                                         READIN    41
      MIN=0.0                                                           READIN    42
      NPROD=1                                                           MAINLOG   19
      NUN=0                                                             MAINLOG   20
      DMFLG=.F.                                                         MAINLOG   21
      DEBUG=.F.                                                         MAINLOG   22
      INPOUT=.T.                                                        MAINLOG   23
      STGOUT=.F.                                                        MAINLOG   24
      COVINT=.F.                                                        POSTPCH    3
      LSTCH=.F.                                                         MAINLOG   25
      COVPRC=.F.                                                        MAINLOG   26
      DEFCHNG=.F.                                                       MAINLOG   27
      STEP=1.0                                                          READIN    44
      B=0.0                                                             READIN    45
      OPTION=0                                                          READIN    46
      TIN=.FALSE.                                                       READIN    47
      MUTIN=.FALSE.                                                     READIN    48
      ELATIN=.FALSE.                                                    READIN    49
      PRODT=.FALSE.                                                     READIN    50
      XYRANGE=.FALSE.                                                   READIN    51
      LPLOT=.FALSE.                                                     READIN    52
      RVPLOT=.FALSE.                                                    READIN    53
      OPTR1=.FALSE.                                                     READIN    54
      OPTR2=.FALSE.                                                     READIN    55
      PLOTR1=.FALSE.                                                    READIN    56
      PLOTR2=.FALSE.                                                    READIN    57
      OUTPT1=.FALSE.                                                    READIN    58
      OUTPT2=.FALSE.                                                    READIN    59
      OUTPT3=.FALSE.                                                    READIN    60
      OUTPT4=.FALSE.                                                    READIN    61
      OUTPT5=.FALSE.                                                    READIN    62
      IA=0                                                              READIN    63
      IAA=0                                                             READIN    64
      IB=0                                                              READIN    65
      IBB=0                                                             READIN    66
      IC=0                                                              READIN    67
      ICC=0                                                             READIN    68
      IPLT1=0                                                           READIN    69
      IPLT2=0                                                           READIN    70
      IPLT3=0                                                           READIN    71
      IPLT4=0                                                           READIN    72
      IPLT5=0                                                           READIN    73
      DO 20 I=1,10                                                      READIN    87
      MUT(I)=0.0                                                        READIN    88
   20 PROD(I)=0                                                         READIN    89
C                                                                       READIN    90
C     READ THE REQUESTED OPTIONS                                        READIN    91
      READ(5,OPTSON)                                                    READIN    92
C         PRINTED ANSWERS TO QUESTIONS OPTION                           READIN    93
      IF(OUTPT3) WRITE(6,1037) OUTPT3                                   READIN    94
      IF(IND .GT. 0) WRITE(6,OPTSON )                                   READIN    95
C         PRODUCT OF RELIABILITY OPTION                                 READIN    96
```

```
      IF(OUTPT3) WRITE(6,1001) PRODT                                  READIN    97
      IF (.NOT.PRODT) GO TO 120                                       READIN    98
C                                                                     READIN    99
C         READ IN THE EQUATION NUMBERS TO BE                          READIN   100
C         USED IN THE PRODUCT OF RELIABILITIES                        READIN   101
      READ(5,VAR1)                                                    READIN   102
      IF (OUTPT3) WRITE (6,VAR1)                                      READIN   103
C                                                                     READIN   104
C         COUNT THE NUMBER OF EQUATIONS INVOLVED IN THE PRODUCT       READIN   105
      NPROD=NPT                                                       READIN   106
      NPT1=NPT+1                                                      READIN   107
      DO 80 I=1,NPT                                                   READIN   108
      IF (PROD(NPT1-I).NE.0) GO TO 90                                 READIN   109
      NPROD=NPROD-1                                                   READIN   110
   80 CONTINUE                                                        READIN   111
C                                                                     READIN   112
   90 DO 100 I=1,NPROD                                                READIN   113
      IF(PROD(I).LT.1.OR.PROD(I).GT.7)  GO TO 110                     CAREFIX    1
      IF(PROD(I).NE.7)  GO TO 100                                     MAINLOG   28
      DMFLG=.T.                                                       MAINLOG   29
      IF((I-NUN).GT.1)  GO TO 110                                     MAINLOG   30
      NUN=I                                                           MAINLOG   31
  100 CONTINUE                                                        CAREFIX    2
      IF(NUN.LE.8)  GO TO 121                                         MAINLOG   32
  110 WRITE(6,1022)                                                   CAREFIX    4
      WRITE(6,1005) (PROD(I),I=1,NPROD)                               READIN   116
      WRITE( 6,1006)                                                  READIN   117
      STOP                                                            READIN   118
C     WRITE THE RELIABILITY EQUATION NUMBER TO BE USED                READIN   119
  120 IF(OUTPT3) WRITE(6,1039) NEQ                                    READIN   120
      IF(NEQ .GT. 7.OR. NEQ .LT. 1) GO TO 590                         READIN   121
      DMFLG=NEQ.EQ.7                                                  MAINLOG   33
      IF(DMFLG)  NUN=1                                                MAINLOG   34
      PROD(1)=NEQ                                                     MAINLOG   35
C                                                                     READIN   122
C         READ IN THE INPUT VARIABLES FOR THE RELIABILITY EQUATIONS   READIN   123
  121 READ(5,VAR)                                                     READIN   124
C                                                                     READIN   125
      IF( INO  .GT. 0) WRITE(6,VAR)                                   READIN   126
C                                                                     READIN   127
C                                                                     FINAL      1
C          PACK APPLICABILITY FIELD OF EACH PCW ELEMENT,              FINAL      2
C          THEN INITIALIZE DEFAULT VALUES.                            FINAL      3
C                                                                     FINAL      4
      DO 150  J=1,NPROD                                               MAINLOG   36
      DO 150  I=1,22                                                  MAINLOG   37
      MTMP=AND(MB59,ISHIFT(PCW(I),PROD(J)))                           MAINLOG   38
  150 PCW(I)=OR(PCW(I),ISHIFT(MTMP,41-J))                             MAINLOG   39
      Q1DEF=2.0                                                       MAINLOG   40
      Q2DEF=1.0                                                       MAINLOG   41
      LAMDEF=1.0E-6                                                   MAINLOG   42
      MUDEF=1.0E-6                                                    MAINLOG   43
      GMPDEF=1.0E-6                                                   MAINLOG   44
      CDEF=1.0                                                        MAINLOG   45
      CTRDEF=1.0                                                      MAINLOG   46
      PRCDEF=1.0                                                      MAINLOG   47
      RVDEF=1.0                                                       MAINLOG   48
      PDEF=0.5                                                        MAINLOG   49
      NDEF=3                                                          MAINLOG   50
      SDEF=0                                                          MAINLOG   51
      WDEF=1                                                          MAINLOG   52
      ZDEF=1                                                          MAINLOG   53
C                                                                     FINAL      5
C          READ NEW DEFAULT VALUES IF REQUESTED                       FINAL      6
C                                                                     FINAL      7
      IF(DEFCHNG)  READ(5,DEFAULT)                                    MAINLOG   54
```

```
C                                                                      FINAL      8
C           READ INVARIANT COVERAGE DATA IF REQUESTED!                 FINAL      9
C           COVERAGE CALCULATION FLAGS AND LINKAGE DATA                FINAL     10
C                                                                      FINAL     11
      IF(COVPRC)  READ(5,COVCAL)                                       MAINLOG   55
C          TEST TO SEE IF T, LAMT, OR ELAMT WAS INPUTED                READIN   128
      IF(T .NE. 0.0) GO TO 190                                         READIN   129
      IF (LAMT.NE.0.0  ) GO TO 200                                     READIN   130
      IF (ELAMT.LE.1.0  ) GO TO 180                                    READIN   131
      ELAMT=1.0                                                        READIN   132
      WRITE (6,1032)                                                   READIN   133
C                                                                      READIN   134
C          DEFAULT VALUE FOR ELAMT IS 1.0D0                            READIN   135
  180 IF (ELAMT.EQ.0.0  ) ELAMT=1.0                                    READIN   136
      ELATIN=.TRUE.                                                    READIN   137
      GO TO 200                                                        READIN   138
  190 TIN=.TRUE.                                                       READIN   139
C                                                                      READIN   140
C          ARE 2-D RELIABILITY PLOTS TO BE PERFORMED                   READIN   141
  200 IF(OUTPT3) WRITE(6,1013) LPLOT                                   READIN   142
      IF (.NOT.LPLOT) GO TO 280                                        READIN   143
C                                                                      READIN   144
C          2-D PLOT OPTION                                             READIN   145
      WRITE (6,1014)                                                   READIN   146
      READ( 5,1017) IPLT1,IPLT2,IPLT3,IPLT4,IPLT5                      READIN   147
      IF (OUTPT3) WRITE (6,1040) IPLT1,IPLT2,IPLT3,IPLT4,IPLT5         READIN   148
      IF (.NOT.PHODT) IPLT1=0                                          READIN   149
      WRITE (6,1016)                                                   READIN   150
      READ(5,1017) IA,IAA,IB,IBB,IC,ICC                                READIN   151
      IF (OUTPT3) WRITE (6,1040) IA,IAA,IB,IBB,IC,ICC                  READIN   152
      IF (TIN) GO TO 240                                               READIN   153
      IF (IA.EQ.1) WRITE (6,1018)                                      READIN   154
      IF (IAA.EQ.1) WRITE (6,1019)                                     READIN   155
      IA=0                                                             READIN   156
      IAA=0                                                            READIN   157
C                                                                      READIN   158
C          X-AXIS RANGE OPTION                                         READIN   159
  240 IF(.NOT.XYRANGE)GO TO 280                                        READIN   160
      WRITE(6,1020)                                                    READIN   161
      READ (5,1021)                        XMIN,XMAX                   READIN   162
      IF (OUTPT3) WRITE (6,1041) XMIN,XMAX                             READIN   163
C                                                                      READIN   164
C          Y-AXIS RANGE OPTION                                         READIN   165
      WRITE(6,1015)                                                    READIN   166
      READ (5,1021)                        YMIN,YMAX                   READIN   167
      IF (OUTPT3) WRITE (6,1041) YMIN,YMAX                             READIN   168
C                                                                      READIN   169
  280 IF (PRODT) GO TO 430                                             READIN   170
C                                                                      READIN   171
C          RV VERSUS R PLOT OPTION                                     READIN   172
      IF(OUTPT3) WRITE(6,1027) RVPLOT                                  READIN   173
      IF(ELATIN) GO TO 430                                             READIN   174
C                                                                      READIN   175
C         MAXIMUM MISSION TIME OPTIONS IF ELAMT WAS NOT INPUTED        READIN   176
C         MAXIMUM MISSION TIME OPTION 1                                READIN   177
       IF(OUTPT3) WRITE(6,1046)OPTR1                                   READIN   178
      IF( .NOT. OPTR1) GO TO 370                                       READIN   179
      IF(OUTPT3) WRITE(6,1047) PLOTR1                                  READIN   180
  370 IF (PRODT) GO TO 410                                             READIN   181
C         MAXIMUM MISSION TIME OPTION 2                                READIN   182
      IF(OUTPT3) WRITE(6,1048) OPTR2                                   READIN   183
      IF(.NOT.OPTR2) GO TO 410                                         READIN   184
       IF(OUTPT3) WRITE(6,1049)                                        READIN   185
       READ(5,1008) R2OPT                                              READIN   186
      IF(OUTPT3)WRITE(6,1109)R2OPT                                     READIN   187
      IF(OUTPT3)WRITE(6,1050) PLOTR2                                   READIN   188
```

```
  410 IF (.NOT.(OPTR1.OR.OPTR2)) GO TO 430                              READIN
      IF(OUTPT3) WRITE(6,1051)                                          READIN
      READ(5,1053) R2T,R1MINT,R1MAXT,R1STT                              READIN
      IF(OUTPT3) WRITE(6,1054)R2T,R1MINT,R1MAXT,R1STT                   READIN
      R2=R2T                                                            READIN
      R1MIN=R1MINT                                                      READIN
      R1MAX=R1MAXT                                                      READIN
      R1STEP=R1STT                                                      READIN
C                                                                       READIN
C                                                                       READIN
C           OUTPUT OPTIONS                                              READIN
  430 IF(.NOT. OUTPT3) RETURN                                           READIN
      WRITE(6,1034) OUTPT1                                              READIN
                WRITE(6,1036) OUTPT2                                    READIN
                WRITE(6,1045) OUTPT4                                    READIN
                WRITE(6,1052) OUTPT5                                    READIN
C                                                                       READIN
      RETURN                                                            READIN
  590 WRITE(6,1022)                                                     READIN
      STOP                                                              READIN
C                                                                       READIN
C           FORMAT STATEMENTS                                           READIN
C                                                                       READIN
 1001 FORMAT (* DO YOU WISH TO FORM A COMPLEX EQUATION WHICH IS*,/,     READIN
     ** THE PRODUCT OF THE PRIMARY EQUATIONS,*,30X,*PRODT= *,L2)        READIN
 1002 FORMAT (80A1)                                                     READIN
 1005 FORMAT (* INPUT VARIABLES FOR EQUATIONS *,10(I2,2X),/,            READIN
     ** WHICH ARE INVOLVED IN THE PRODUCT OF RELIABILITIES.*)           READIN
 1006 FORMAT (* TYPE IN THE EQUATION NUMBERS INVOLVED IN THE PRODUCT*,/,READIN
     ** USE VAR1 AS THE NAMELIST NAME*,/,                               READIN
     ** EXAMPLE* FOR THE PRODUCT OF THE RELIABILITIES OF EQUATIONS 1, 2,READIN
     * AND 5*,/,* TYPE*,/,                                              READIN
     **    $VAR1 PROD=1,2,5$*,/,                                        READIN
     ** NOTE* NAMELIST INPUT IGNORES COLUMN 1*)                         READIN
 1007 FORMAT (* TYPE IN COLUMN 1 THE NUMBER OF THE RELIABILITY EQUATION*READIN
     *,/,* TO BE USED - 1 THRU 7*,//)                                   READIN
 1008 FORMAT (I1)                                                       READIN
 1013 FORMAT( * DO YOU WISH TO HAVE 2-D RELIABILITY PLOTS- *22X,        READIN
     ** LPLOT= *,L2)                                                    READIN
 1014 FORMAT (* INPUT A 1 IN THE COLUMN SPECIFIED BELOW IF YOU WISH*,/, READIN
     *  * THE CORRESPONDING PLOT OPTION. OTHERWISE INPUT 0.*,/,         READIN
     *  * NOTE* WHEN PERFORMING PRODUCT OF RELIABILITIES, NO OTHER*,/,  READIN
     *  * PLOT OPTION BESIDES PRODUCT OF RELIABILITIES MAY BE*,         READIN
     *  * SPECIFIED.*,/,                                                READIN
     *  * COLUMN 1 - PLOTS PRODUCT OF RELIABILITIES*,/,                 READIN
     *  * COLUMN 2 - PLOTS RELIABILITY*,/,                              READIN
     *  * COLUMN 3 - PLOTS DIFF, RIF, AND GAIN*,/,                      READIN
     *  * COLUMN 4 - PLOTS MTF AND RELIABILITY AT MTF*,/,               READIN
     *  * COLUMN 5 - PLOTS UNRELIABILITY*)                              READIN
 1015 FORMAT (* IF YOU WISH TO PLOT A CERTAIN RANGE OF Y-AXIS VALUES*,/,READIN
     *  * FOR THE 2-D PLOTS, ENTER LEFT-END POINT IN COLUMNS 1-8 WITH*,/READIN
     *  * FORMAT F8.0 AND RIGHT-END POINT IN COLUMNS 9-16 WITH FORMAT*, READIN
     *  * F8.0 *)                                                       READIN
 1016 FORMAT (* FOR ABSCISSA, INPUT 1 IN COLUMN 1 IF ABSCISSA IS T,*,/, READIN
     ** 1 IN COLUMN 2 IF ABSCISSA IS LOG(T) - BASE 10,*,/,              READIN
     ** 1 IN COLUMN 3 IF ABSCISSA IS LAMT,*,/,                          READIN
     ** 1 IN COLUMN 4 IF ABSCISSA IS LOG(LAMT) - BASE 10,*,/,           READIN
     ** 1 IN COLUMN    5 IF ABSCISSA IS EXP(-LAMBDA(T)),*,/,            READIN
     ** 1 IN COLUMN 6 IF ABSCISSA IS LOG(EXP(-LAMT)) - BASE 10,*)       READIN
 1017 FORMAT (6I1)                                                      READIN
 1018 FORMAT (* WARNING ABSCISSA CAN NOT BE T - NO PLOT FOR ABSCISSA T*)READIN
 1019 FORMAT (* WARNING ABSCISSA CAN NOT BE LOG(T) - BASE 10 - NO PLOT*,READIN
     */,* FOR ABSCISSA LOG(T) - BASE 10*)                               READIN
 1020 FORMAT (* IF YOU WISH TO PLOT A CERTAIN RANGE OF X-AXIS VALUES*,/,READIN
     *  * FOR THE 2-D PLOTS, ENTER LEFT-END POINT IN COLUMNS 1-8 WITH*,/READIN
     *  * FORMAT F8.0 AND RIGHT-END POINT IN COLUMNS 9-16 WITH FORMAT*, READIN
```

```
     *    * F8.0 *)                                                  READIN    255
 1021 FORMAT (2F8.0)                                                 READIN    256
 1022 FORMAT (* ERROR IN LAST INPUT - MUST HAVE PROD .GE. 1 AND .LE. 7*,READIN    257
     */,* RESUBMIT EQUATION NUMBERS*)                                READIN    258
 1024 FORMAT (* ENTER MINIMUN VALUE FOR PARAMETER IN COLUMNS 1-8 WITH FOREADIN    259
     *RMAT F8.0*,/,* MAXIMUM VALUE FOR PARAMETER IN COLUMNS 9-16 WITH FOREADIN    260
     *RMAT F8.0*,/,* INCREMENT VALUE FOR PARAMETER IN COLUMNS 16-24 WITHREADIN    261
     * FORMAT F8.0*)                                                 READIN    262
 1025 FORMAT (3F8.0)                                                 READIN    263
 1027 FORMAT (* DO YOU WISH TO PLOT THE LOCUS OF RV SUCH THAT THE*,/,  READIN    264
     ** SYSTEM RELIABILITY EQUALS THE UNIT RELIABILITY,*,19X,        READIN    265
     ** RVPLOT= *,L2)                                                READIN    266
 1032 FORMAT (* WARNING* ELAMT MUST BE GREATER THAN 0.0 AND*,/,       READIN    267
     *   * LESS THAN OR EQUAL TO 1.0  .  ELAMT IS SET EQUAL TO 1.0  *)  READIN    268
 1034 FORMAT (* DO YOU WISH TO HAVE PRINTED TABLE OF RELIABILITY*,    READIN    269
     *     * RESULTS*,11X,*OUTPT1= *,L2)                             READIN    270
 1036 FORMAT (* DO YOU WISH TO HAVE PRINTED TABLE OF DIFF, RIF,*,/,   READIN    271
     *    * AND GAIN RESULTS - *,48X, *OUTPT2= *,L2)                 READIN    272
 1037 FORMAT(//40X,*CARE2 (COMPUTER-AIDED RELIABILITY *,              TITLE       1
     *      *ESTIMATION)*////                                        TITLE       2
     ** DO YOU WISH TO HAVE YOUR RESPONSES TO MY*,/,                 READIN    275
     ** QUESTIONS PRINTED BACK FOR VERIFICATION.*,27X,*OUTPT3=*,L2)  READIN    276
 1038 FORMAT (1H ,80A1)                                              READIN    277
 1039 FORMAT(1H , * EQUATION NUMBER*,51X,*NEQ= *I1)                  READIN    278
 1040 FORMAT (1H ,6I1)                                               READIN    279
 1041 FORMAT (1H ,2F8.3)                                             READIN    280
 1042 FORMAT (1H ,3F8.3)                                             READIN    281
 1043 FORMAT (1H ,F8.3)                                              READIN    282
 1044 FORMAT (* NO*)                                                 READIN    283
 1045 FORMAT (* DO YOU WISH MTF AND RELIABILITY AT MTF RESULTS*      READIN    284
     *      * PRINTED*,13X,*OUTPT4= *,L2)                            READIN    285
 1046 FORMAT (* DO YOU WISH TO CALCULATE MAXIMUM MISSION TIME AND*,   READIN    286
     *    * SIMPLE TIME *,/* FOR GIVEN RELIABILITY-*45X,*OPTR1= *,L2) READIN    287
 1047 FORMAT( * DO YOU WANT PLOTS FOR THESE CALCULATIONS - *,         READIN    288
     * 23X, * PLOTR1= *,L2)                                          READIN    289
 1048 FORMAT (* DO YOU WISH TO CALCULATE MAXIMUM MISSION TIME FOR*,/, READIN    290
     * * GIVEN RELIABILITY AND COMPARE IT AGAINST OTHER PARAMETERS*, 9X,READIN    291
     ** OPTR2= *,L2)                                                 READIN    292
 1049 FORMAT (* INPUT IN COLUMN 1 ONE OF THE FOLLOWING THREE OPTIONS**,/READIN    293
     *    * 1. MAXIMUM MISSION TIME IS COMPARED AGAINST ALL POSSIBLE*,/,  READIN    294
     *    * COMBINATIONS OF THE PARAMETER,*,/,                       READIN    295
     *    * 2. MAXIMUM MISSION TIME IS COMPARED AGAINST THE LAST TWO*,/,  READIN    296
     *    * PARAMETER VALUES,*,/,                                    READIN    297
     *    * 3. THE PROGRAM ASKS THE USER AS TO WHICH PARAMETER VALUES*,/, READIN    298
     *    * MAXIMUM MISSION TIME IS TO BE COMPARED,*)                READIN    299
 1050 FORMAT(* DO YOU WANT PLOTS FOR THESE CALCULATIONS-*,26X, *PLOTR2=*READIN    300
     *,L2, /,* NOTE, WHEN EXERCISING OPTION 1, THE PROGRAM*           READIN    301
     *    * PLOTS*,/,* ONLY THE FIRST 15 PARAMETER COMPARISONS*)     READIN    302
 1051 FORMAT(//* INPUT THE FOLLOWING 4 VARIABLES EACH WITH FORMAT F8.0*,READIN    303
     *    * COLUMNS  1-8  - REFERENCE RELIABILITY R2*,/,              READIN    304
     *    * COLUMNS  9-16 - MINIMUM RELIABILITY R1*,/,                READIN    305
     *    * COLUMNS 17-24 - MAXIMUM RELIABILITY R1*,/,                READIN    306
     *    * COLUMNS 25-32 - RELIABILITY R1 STEP SIZE*)                READIN    307
 1052 FORMAT (* DO YOU WANT PRINTED RESULTS OF THE MAXIMUM MISSION*,/, READIN    308
     *    * TIME CALCULATIONS - *,47X, *OUTPT5= *,L2)                READIN    309
 1053 FORMAT (4F8.0)                                                 READIN    310
 1054 FORMAT (1H ,4F8.3)                                             READIN    311
 1109 FORMAT( 67X, * R2OPT= *,I1)                                    READIN    312
C                                                                    READIN    313
C                                                                    READIN    314
      END                                                            READIN    315
      SUBROUTINE READIN2(NRS,DMFLG,IRES,LVARY,DVAL,DEBUG)            READIN2     1
C                                                                    FINAL       1
C          THIS ROUTINE, CALLED PRIOR TO EACH RUN-SET EXECUTION,     FINAL       2
C          PERFORMS GENERAL DATA BASE MANAGEMENT INVOLVING           FINAL       3
C          VARIABLES OF BOTH THE RELIABILITY AND COVERAGE MODELS     FINAL       4
```

```
C                                                                       FINAL      5
      DIMENSION Q(10),Q1(10),Q2(10), C(10),C1(10),C2(10),               READIN2    2
     $     CD(10),CD1(10),CD2(10), PRC(10),PRC1(10),PRC2(10)            READIN2    3
      DIMENSION NEXT(4),NPROC(4),OUT(10),LINE(5,4),FUNBAD(5,4)          READIN2    4
      COMMON /GRAPH/NEQ,PROD(10),NPROD                                  READIN2    5
      COMMON /PARA1/QQ(10,2),IJK(10),N(10),S(10)                        READIN2    6
      COMMON /PARA2/LAM(10),MU(10),XYZ(20),SLH2,SLH3                    READIN2    7
      COMMON /PARA3/RV(10),Z(10),W(10),P(10)                            READIN2    8
      COMMON /PARA4/CC(10,2),CCD(10,2),CTR(10),CDTR(10),CCSF            READIN2    9
      COMMON /PARA5/ZYX(20),GMP(10),PPRC(10,2)                          READIN2   10
      COMMON /PARM/PARAM(16,10),N10                                     READIN2   11
      COMMON /DRIVE/YXZ(3),NUN,RSGN                                     READIN2   12
      COMMON /DEFS/PCW(22),NAMES(22),RLDFS(10),INTDFS(4)                READIN2   13
      COMMON /CVB0/COVINT,IGENC(10),IGENP(10),IFSC(8),FRAC(8)           READIN2   14
      COMMON /CVB1/NDET(20,8),NISO(20,8),NEPR(20,8),NTLR(20,8)          READIN2   15
      COMMON /CVB2/FDET(7,200),FISO(7,50),FEPR(7,25),FTLR(7,25)         READIN2   16
      COMMON /CVB5/PFDS(8),TFDS(8)                                      READIN2   17
      COMMON /CVB6/LCM,TMINOR,FACTOR                                    READIN2   18
      COMMON /CVB9/NINT                                                 POSTPCH    1
      REAL LAM,MU                                                       READIN2   19
      INTEGER S,Z,W,PROD,PCW,RLFLG                                      READIN2   20
      LOGICAL RSGN,DMFLG,LVARY,COVINT,DEBUG                             POSTPCH    2
      LOGICAL ECARD,FCARD,MDF,IETF,TFLG,MFLG,BPLG                       READIN2   22
      EQUIVALENCE (QQ(1,1),Q(1),Q1(1)),(QQ(1,2),Q2(1))                  READIN2   23
      EQUIVALENCE (CC(1,1),C(1),C1(1)),(CC(1,2),C2(1))                  READIN2   24
      EQUIVALENCE (CCD(1,1),CD(1),CD1(1)),(CCD(1,2),CD2(1))             READIN2   25
      EQUIVALENCE (PPRC(1,1),PRC(1),PRC1(1)),(PPRC(1,2),PRC2(1))        READIN2   26
      EQUIVALENCE (IGFT,ISU),(INTF,MD),(ISCH,IET),(IREP,MKN)            READIN2   27
      NAMELIST /DATA/SLH2,SLH3,CCSF,RSGN,PFDS,TFDS,TMINOR,NINT          POSTPCH    3
      NAMELIST /PARVEC/Q,Q1,Q2,N,S,W,Z,LAM,MU,GMP,C,C1,C2,CD,CD1,CD2,   READIN2   29
     $     CTR,CDTR,PRC,PRC1,PRC2,RV,P                                  READIN2   30
      NAMELIST /VARY/PARAM                                              READIN2   31
      DATA (NDET(I),I=1,160)/160*0/                                     POSTPCH    4
      DATA (NISO(I),I=1,160)/160*0/                                     POSTPCH    5
      DATA (NEPR(I),I=1,160)/160*0/                                     POSTPCH    6
      DATA (NTLR(I),I=1,160)/160*0/                                     POSTPCH    7
      DATA (FDET(I),I=1,1400)/1400*0.0/                                 POSTPCH    8
      DATA (FISO(I),I=1,350)/350*0.0/                                   POSTPCH    9
      DATA (FEPR(I),I=1,175)/175*0.0/                                   POSTPCH   10
      DATA (FTLR(I),I=1,175)/175*0.0/                                   POSTPCH   11
      DATA (PFDS(I),I=1,8)/8*1.0/                                       READIN2   32
      DATA (TFDS(I),I=1,8)/8*0.0/                                       READIN2   33
      DATA TMINOR/1.0/                                                  READIN2   34
      DATA (NEXT(I),I=1,4)/200,50,25,25/                                READIN2   35
      DATA MSKSUB/31/,ECARD,FCARD/2*.F./                                READIN2   36
      DATA MB59/40000000000000000000B/                                  READIN2   37
      DATA (NPROC(I),I=1,4)/9HDETECTION,9HISOLATION,                    READIN2   38
     $     10HE. P. REC.,10HT. L. REC./                                 READIN2   39
      DATA STAR,SLASH/1H*,1H//,B,X/3H   ,3H  X/                         READIN2   40
      DATA NINT/20/                                                     POSTPCH   12
   11 FORMAT(2A3,2I4,2I1,I4,6F10.5)                                     READIN2   41
C                                                                       FINAL      6
C          READ IN A CONTROL CARD, USING THE STANDARD FORMAT.           FINAL      7
C          THE CONTROL CODE IN COL.S 1-3 WILL DETERMINE WHICH           FINAL      8
C          INPUT VALUES WILL ACTUALLY BE USED                           FINAL      9
C                                                                       FINAL     10
    1 READ(5,11)ID,IP,NUM,IGFT,INTF,ISCH,IREP,COEF,TDEL,P1,P2,P3,TDUR   READIN2   42
      IF(EOF,5)1000,2                                                   READIN2   43
    2 IF(NUM.LE.0)  NUM=0                                               READIN2   44
      IF(IGFT.LE.0)  IGFT=0                                             READIN2   45
      IF(INTF.LE.0)  INTF=0                                             READIN2   46
      IF(ISCH.LE.0)  ISCH=0                                             READIN2   47
      IF(IREP.LE.0)  IREP=0                                             READIN2   48
C                                                                       FINAL     11
C          JUMP TO THE START OF CONTROL PROCESSING SEGMENT              FINAL     12
C                                                                       FINAL     13
```

```
      IF(ECARD)  GO TO 65                                         READIN2   49
      IF(ID.EQ.2HID)  GO TO 5                                     READIN2   50
      IF(ID.EQ.2HIB)  GO TO 10                                    READIN2   51
      IF(ID.EQ.2HIV)  GO TO 20                                    READIN2   52
      IF(ID.EQ.1HC)  GO TO 30                                     READIN2   53
      IF(ID.EQ.1HE)  GO TO 50                                     READIN2   54
      IF(ID.EQ.1HM)  GO TO 55                                     READIN2   55
      IF(ID.EQ.1HF)  GO TO 60                                     READIN2   56
      IF(ID.EQ.1HN)  GO TO 70                                     READIN2   57
      IF(ID.EQ.2HPC)  GO TO 80                                    READIN2   58
      IF(ID.EQ.2HPF)  GO TO 90                                    READIN2   59
      IF(ID.EQ.1HG)  GO TO 900                                    READIN2   60
      IF(ID.EQ.1HS)  GO TO 1000                                   READIN2   61
      GO TO 999                                                   READIN2   62
C                                                                 FINAL     14
C          REQUEST FOR NAMELIST DATA                              FINAL     15
C                                                                 FINAL     16
    5 READ(5,DATA)                                                READIN2   63
      GO TO 1                                                     READIN2   64
C                                                                 FINAL     17
C          REQUEST FOR NAMELIST PARVEC                            FINAL     18
C                                                                 FINAL     19
   10 READ(5,PARVEC)                                              READIN2   65
      GO TO 1                                                     READIN2   66
C                                                                 FINAL     20
C          REQUEST FOR ITERATION ON PARAMETER 'NUM.               FINAL     21
C          VARIATION DATA IS READ VIA NAMELIST VARY.              FINAL     22
C                                                                 FINAL     23
   20 IRES=NUM                                                    READIN2   67
      IF(IRES.LT.20)  GO TO 22                                    READIN2   68
      LVARY=.F.                                                   READIN2   69
      GO TO 999                                                   READIN2   70
   22 LVARY=IRES.GT.0                                             READIN2   71
      N10=1                                                       READIN2   72
      IF(.NOT.LVARY)  GO TO 1                                     READIN2   73
      RLFLG=PCW(IRES)                                             READIN2   74
      ISUB=AND(MSKSUB,RLFLG)                                      READIN2   75
      IF(RLFLG.LT.0)  DVAL=RLDFS(ISUB)                            READIN2   76
      IF(RLFLG.GT.0)  DVAL=INTDFS(ISUB)                           READIN2   77
      DO 24  J=1,NPROD                                            READIN2   78
      DO 24  I=2,16                                               READIN2   79
   24 PARAM(I,J)=DVAL                                             READIN2   80
      DO 26  J=1,NPROD                                            READIN2   81
      GO TO(101,102,103,104,105,106,107,108,109,110,              READIN2   82
     *      111,112,113,114,115,116,117,118,119),IRES             READIN2   83
  101 VAL=Q1(J)                                                   READIN2   84
      GO TO 26                                                    READIN2   85
  102 VAL=Q2(J)                                                   READIN2   86
      GO TO 26                                                    READIN2   87
  103 VAL=N(J)                                                    READIN2   88
      GO TO 26                                                    READIN2   89
  104 VAL=S(J)                                                    READIN2   90
      GO TO 26                                                    READIN2   91
  105 VAL=W(J)                                                    READIN2   92
      GO TO 26                                                    READIN2   93
  106 VAL=Z(J)                                                    READIN2   94
       GO TO 26                                                   READIN2   95
  107 VAL=LAM(J)                                                  READIN2   96
       GO TO 26                                                   READIN2   97
  108 VAL=MU(J)                                                   READIN2   98
      GO TO 26                                                    READIN2   99
  109 VAL=GMP(J)                                                  READIN2  100
      GO TO 26                                                    READIN2  101
  110 VAL=C1(J)                                                   READIN2  102
      GO TO 26                                                    READIN2  103
  111 VAL=CD1(J)                                                  READIN2  104
```

```
      GO TO 26                                                          READIN2  105
  112 VAL=C2(J)                                                         READIN2  106
      GO TO 26                                                          READIN2  107
  113 VAL=CD2(J)                                                        READIN2  108
      GO TO 26                                                          READIN2  109
  114 VAL=CTR(J)                                                        READIN2  110
      GO TO 26                                                          READIN2  111
  115 VAL=CDTR(J)                                                       READIN2  112
      GO TO 26                                                          READIN2  113
  116 VAL=PRC1(J)                                                       READIN2  114
      GO TO 26                                                          READIN2  115
  117 VAL=PRC2(J)                                                       READIN2  116
      GO TO 26                                                          READIN2  117
  118 VAL=RV(J)                                                         READIN2  118
      GO TO 26                                                          READIN2  119
  119 VAL=P(J)                                                          READIN2  120
   26 PARAM(1,J)=VAL                                                    READIN2  121
      READ(5,VARY)                                                      READIN2  122
      DO 28  I=1,16                                                     READIN2  123
      DO 28  J=1,NPROD                                                  READIN2  124
      IF(PARAM(I,J).EQ.DVAL)  GO TO 28                                  READIN2  125
      N10=I                                                             READIN2  126
   28 CONTINUE                                                          READIN2  127
      GO TO 1                                                           READIN2  128
C                                                                       FINAL     24
C          CONTROL CARD CONTAINS FUNCTION NUMBER SELECTION              FINAL     25
C          DATA (C CARD)                                                FINAL     26
C                                                                       FINAL     27
   30 IPR=0                                                             READIN2  129
      IF(IP.EQ.1HD)  IPR=1                                              READIN2  130
      IF(IP.EQ.1HI)  IPR=2                                              READIN2  131
      IF(IP.EQ.1HE)  IPR=3                                              READIN2  132
      IF(IP.EQ.1HT)  IPR=4                                              READIN2  133
      IF(IPR.EQ.0)  GO TO 999                                           READIN2  134
      IF(ECARD)  GO TO 52                                               READIN2  135
      IF(NUM.GT.NEXT(IPR))  GO TO 999                                   READIN2  136
      NUMSAV=NUM                                                        READIN2  137
      IF(FCARD)  GO TO 62                                               READIN2  138
   31 MDF=MD.GT.2                                                       READIN2  139
      IF(IET.EQ.0.AND.MD.NE.0)  GO TO 999                               READIN2  140
      IETF=IET.GT.2                                                     READIN2  141
      IF(ISU.EQ.0)  GO TO 999                                           READIN2  142
      IF(ISU.GT.8)  GO TO 32                                            READIN2  143
      ISU1=ISU                                                          READIN2  144
      ISU2=ISU                                                          READIN2  145
      GO TO 34                                                          READIN2  146
   32 ISU1=1                                                            READIN2  147
      ISU2=8                                                            READIN2  148
   34 IF(MKN.EQ.0)  GO TO 999                                           READIN2  149
      IF(MKN.GT.20)  GO TO 36                                           READIN2  150
      MKN1=MKN                                                          READIN2  151
      MKN2=MKN                                                          READIN2  152
      GO TO 38                                                          READIN2  153
   36 MKN1=1                                                            READIN2  154
      MKN2=20                                                           READIN2  155
   38 DO 49  MDT=1,3                                                    POSTPCH   13
      IF(MDF.OR.(MDT.EQ.(MD+1)))  GO TO 40                              READIN2  157
      GO TO 49                                                          POSTPCH   14
   40 DO 48  IETT=1,2                                                   READIN2  159
      IF(IETF.OR.(IETT.EQ.IET))  GO TO 41                               READIN2  160
      GO TO 48                                                          READIN2  161
   41 IF(MDT.EQ.1.AND.IETT.GT.1)  GO TO 48                              READIN2  162
      INDEX=2*IETT+MDT-3                                                READIN2  163
      IF(MDT.EQ.1)  INDEX=5                                             READIN2  164
      DO 47  ISU=ISU1,ISU2                                              POSTPCH   15
      DO 47  MKN=MKN1,MKN2                                              POSTPCH   16
```

```
      GO TO(43,44,45,46),IPR                                            READIN2  167
   43 CALL IPUT(NDET(MKN,ISU),INDEX,NUMSAV)                             READIN2  168
      GO TO 47                                                          POSTPCH   17
   44 CALL IPUT(NISO(MKN,ISU),INDEX,NUMSAV)                             READIN2  170
      GO TO 47                                                          POSTPCH   18
   45 CALL IPUT(NEPR(MKN,ISU),INDEX,NUMSAV)                             READIN2  172
      GO TO 47                                                          POSTPCH   19
   46 CALL IPUT(NTLR(MKN,ISU),INDEX,NUMSAV)                             READIN2  174
   47 CONTINUE                                                          POSTPCH   20
   48 CONTINUE                                                          READIN2  175
   49 CONTINUE                                                          POSTPCH   21
      GO TO 1                                                           READIN2  176
C                                                                       FINAL     28
C           CONTROL CARD IS FIRST OF TWO E CARDS,  DATA,                FINAL     29
C           EXCEPT FOR FUNCTION NUMBER, IS SAME AS ON                   FINAL     30
C           C AND F CARDS.                                              FINAL     31
C                                                                       FINAL     32
   50 ECARD=.T.                                                         READIN2  177
      GO TO 30                                                          READIN2  178
   52 NUMSAV=NEXT(IPR)                                                  READIN2  179
      NEXT(IPR)=NUMSAV-1                                                READIN2  180
      GO TO 31                                                          READIN2  181
C                                                                       FINAL     33
C           ENTER THE MECHANISM DEFINITION LOOP, WHICH                  FINAL     34
C           USES THE SPECIAL INPUT FORMAT BELOW.  (M CARD ONLY)         FINAL     35
C                                                                       FINAL     36
   55 READ(5,15)ID,ISU,MKN,((LINE(I,J),J=1,4),I=1,5)                    READIN2  182
   15 FORMAT(A1,9X,2I5,20I3)                                            READIN2  183
      IF(ID.NE.1H )  GO TO 1                                            READIN2  184
      IF(ISU.LT.1.OR.ISU.GT.8)  GO TO 999                               READIN2  185
      IF(MKN.LT.1.OR.MKN.GT.20)  GO TO 999                              READIN2  186
      DO 57  I=1,5                                                      READIN2  187
      DO 56  J=1,4                                                      READIN2  188
      IF(LINE(I,J).LE.0)  LINE(I,J)=0                                   READIN2  189
   56 CONTINUE                                                          READIN2  190
      CALL IPUT(NDET(MKN,ISU),I,LINE(I,1))                              READIN2  191
      CALL IPUT(NISO(MKN,ISU),I,LINE(I,2))                              READIN2  192
      CALL IPUT(NEPR(MKN,ISU),I,LINE(I,3))                              READIN2  193
   57 CALL IPUT(NTLR(MKN,ISU),I,LINE(I,4))                              READIN2  194
      GO TO 55                                                          READIN2  195
C                                                                       FINAL     37
C           CONTROL CARD CONTAINS FUNCTION SPECIFICATION DATA           FINAL     38
C           (F CARD)                                                    FINAL     39
C                                                                       FINAL     40
   60 FCARD=.T.                                                         READIN2  196
      GO TO 30                                                          READIN2  197
   62 FCARD=.F.                                                         READIN2  198
   65 ECARD=.F.                                                         READIN2  199
      GO TO (66,67,68,69),IPR                                           READIN2  200
   66 CALL SPECIT(FDET,NUMSAV,IGFT,ISCH,IREP,INTF,COEF,                 READIN2  201
     *     TDEL,P1,P2,P3,TDUR)                                          READIN2  202
      GO TO 1                                                           READIN2  203
   67 CALL SPECIT(FISO,NUMSAV,IGFT,0,0,INTF,COEF,                       READIN2  204
     *     TDEL,P1,P2,P3,TDUR)                                          READIN2  205
      GO TO 1                                                           READIN2  206
   68 CALL SPECIT(FEPR,NUMSAV,IGFT,0,0,INTF,COEF,                       READIN2  207
     *     0.0,P1,P2,P3,TDUR)                                           READIN2  208
      GO TO 1                                                           READIN2  209
   69 CALL SPECIT(FTLR,NUMSAV,IGFT,0,0,INTF,COEF,                       READIN2  210
     *     0.0,P1,P2,P3,TDUR)                                           READIN2  211
      GO TO 1                                                           READIN2  212
C                                                                       FINAL     41
C           NORMALIZE RATE FUNCTIONS AND CHECK ALL DEFINED              FINAL     42
C           FUNCTIONS FOR REASONABLENESS                                FINAL     43
C                                                                       FINAL     44
   70 DO 270  IPR=1,4                                                   READIN2  213
```

```
      NAM=NPRUL(IPR)                                                    READIN2  214
      NMAX=NEXT(IPR)                                                    READIN2  215
  200 DO 260  NUM=1,NMAX                                                READIN2  216
      GO TO (210,220,230,240),IPR                                       READIN2  217
  210 COEF=FDET(2,NUM)                                                  FINAL     45
      CODES=FDET(1,NUM)                                                 READIN2  219
      IF(IGET(CODES,4).EQ.0)  GO TO 250                                 READIN2  220
      IF(IGET(CODES,1).EQ.0)  GO TO 250                                 READIN2  221
      TDUR=FDET(7,NUM)                                                  READIN2  222
      TEST=FINTEG(FDET,NUM,TDUR)                                        READIN2  223
      IF(TEST.GT.0.0)  GO TO 212                                        READIN2  224
      WRITE(6,71)NAM,NUM                                                READIN2  225
   71 FORMAT(/10X,*RANGE ERROR IN *,A10,* FUNCTION NUMBER*,I4,          READIN2  226
     $       * - FUNCTION DISABLED*)                                    READIN2  227
      CALL IPUT(CODES,4,0)                                              READIN2  228
      FDET(1,NUM)=CODES                                                 READIN2  229
      GO TO 250                                                         READIN2  230
  212 FDET(4,NUM)=FDET(4,NUM)/TEST                                      READIN2  231
      GO TO 250                                                         READIN2  232
  220 COEF=FISO(2,NUM)                                                  READIN2  233
      CODES=FISO(1,NUM)                                                 READIN2  234
      IF(IGET(CODES,4).EQ.0)  GO TO 250                                 READIN2  235
      IF(IGET(CODES,1).EQ.0)  GO TO 250                                 READIN2  236
      TDUR=FISO(7,NUM)                                                  READIN2  237
      TEST=FINTEG(FISO,NUM,TDUR)                                        READIN2  238
      IF(TEST.GT.0.0)  GO TO 222                                        READIN2  239
      WRITE(6,71)NAM,NUM                                                READIN2  240
      CALL IPUT(CODES,4,0)                                              READIN2  241
      FISO(1,NUM)=CODES                                                 READIN2  242
      GO TO 250                                                         READIN2  243
  222 FISO(4,NUM)=FISO(4,NUM)/TEST                                      READIN2  244
      GO TO 250                                                         READIN2  245
  230 COEF=FEPR(2,NUM)                                                  READIN2  246
      CODES=FEPR(1,NUM)                                                 READIN2  247
      IF(IGET(CODES,4).EQ.0)  GO TO 250                                 READIN2  248
      IF(IGET(CODES,1).GT.0)  GO TO 232                                 READIN2  249
      WRITE(6,73)NAM,NUM                                                READIN2  250
   73 FORMAT(/10X,*IMPULSE NOT ALLOWED FOR REC. FUNCTIONS = *,          READIN2  251
     $       A10,* FUNCTION NUMBER*,I4,* DISABLED*)                     READIN2  252
  232 TDUR=FEPR(7,NUM)                                                  READIN2  253
      DT=TDUR/10.0                                                      READIN2  254
      TFLG=.T.                                                          READIN2  255
      DO 235  ITP1=1,11                                                 POSTPCH   22
      IT=ITP1-1                                                         POSTPCH   23
      T=IT*DT                                                           READIN2  257
      TEST=FEVAL(FEPR,NUM,T)                                            READIN2  258
      IF(TEST.LT.0.0.OR.TEST.GT.1.0001)  TFLG=.F.                       READIN2  259
  235 CONTINUE                                                          READIN2  260
      IF(TFLG)  GO TO 250                                               READIN2  261
      WRITE(6,71)NAM,NUM                                                READIN2  262
  237 CALL IPUT(CODES,4,0)                                              READIN2  263
      FEPR(1,NUM)=CODES                                                 READIN2  264
      GO TO 250                                                         READIN2  265
  240 COEF=FTLR(2,NUM)                                                  READIN2  266
      CODES=FTLR(1,NUM)                                                 READIN2  267
      IF(IGET(CODES,4).EQ.0)  GO TO 250                                 READIN2  268
      IF(IGET(CODES,1).GT.0)  GO TO 242                                 READIN2  269
      WRITE(6,73)NAM,NUM                                                READIN2  270
      GO TO 247                                                         READIN2  271
  242 TDUR=FTLR(7,NUM)                                                  READIN2  272
      DT=TDUR/10.0                                                      READIN2  273
      TFLG=.T.                                                          READIN2  274
      DO 245  ITP1=1,11                                                 POSTPCH   24
      IT=ITP1-1                                                         POSTPCH   25
      T=IT*DT                                                           READIN2  276
      TEST=FEVAL(FTLR,NUM,T)                                            READIN2  277
```

```
      IF(TEST.LT.0.0.OR.TEST.GT.1.0001)  TFLG=.F.
  245 CONTINUE
      IF(TFLG)  GO TO 250
      WRITE(6,71)NAM,NUM
  247 CALL IPUT(CODES,4,0)
      FTLR(1,NUM)=CODES
  250 IF(COEF.GE.0.0.AND.COEF.LE.1.0)  GO TO 260
      WRITE(6,72)NAM,NUM
   72 FORMAT(/10X,*COEFFICIENT ERROR IN *,A10,* FUNCTION NUMBER*,I4,
     $     * - FUNCTION DISABLED*)
      CALL IPUT(CODES,4,0)
      GO TO (251,252,253,254),IPR
  251 FDET(1,NUM)=CODES
      GO TO 260
  252 FISO(1,NUM)=CODES
      GO TO 260
  253 FEPR(1,NUM)=CODES
      GO TO 260
  254 FTLR(1,NUM)=CODES
  260 CONTINUE
  270 WRITE(6,74)NAM
   74 FORMAT(/10X,*NORMALIZATION COMPLETE FOR *,A10,* FUNCTIONS*)
      GO TO 1
C
C          DISPLAY THE COVERAGE DATA BASE, EXCLUSIVE
C          OF FUNCTION SPECIFICATIONS
C
   80 WRITE(6,300)(STAR,I=1,20)
  300 FORMAT(1H1,55X,*COVERAGE DATA BASE*/55X,20A1//)
      WRITE(6,301)
  301 FORMAT(46X,*- FAULT SUBCLASS LINKAGE AND PARAMETERS -*//
     *     29X,*FAULT*,8X,*LINKED*,6X,*FRACTION OF*,5X,
     *     *PROB. OF FAULT*,5X,*TIME FOR FAULT*/28X,*SUBCLASS*
     *     5X,*TO STAGE*,5X,*FAULT RATE*,6X,*DET. IN SPARES*,
     *     5X,*DET. IN SPARES*/)
      DO 310  I=1,8
      IF(IFSC(I).GE.0.AND.IFSC(I).LE.10)  GO TO 310
      IFSC(I)=0
      FRAC(I)=0.0
  310 WRITE(6,302)I,IFSC(I),FRAC(I),PFDS(I),TFDS(I)
  302 FORMAT(28X,I4,10X,I4,9X,F8.6,10X,F8.6,9X,F12.6)
      WRITE(6,308)THINOR
  308 FORMAT(/38X,*MINOR CYCLE DURATION (FOR SCHEDULED*,
     *     * DETECTORS) =*,F12.6)
      WRITE(6,303)
  303 FORMAT(//50X,*- D/I/R MECHANISM DEFINITIONS -*,9X,
     *     *(X MEANS FUNCTION UNDEFINED)*//58X,*FAULT TYPE/MODE*/
     *     17X,*SUBCLASS     MECHANISM*,8X,*PERM/1*,9X,*PERM/2*,8X,
     *     *TRANS/1*,8X,*TRANS/2*,9X,*PERM/0*/44X,*D  I  E  T     *,
     *     *D  I  E  T     D  I  E  T     D  I  E  T     D  I  E  T*/)
      DO 350  ISU=1,8
      IF(IFSC(ISU).EQ.0)  GO TO 350
      DO 350  MEC=1,20
      MFLG=.T.
      DO 320  INDEX=1,5
      NUM=IGET(NDET(MEC,ISU),INDEX)
      IF(NUM.GT.0)  MFLG=.F.
  320 CONTINUE
      IF(MFLG)  GO TO 350
      BFLG=.F.
      DO 330  INDEX=1,5
      DO 330  IPR=1,4
      FUNBAD(INDEX,IPR)=B
      GO TO (321,322,323,324),IPR
  321 NUM=IGET(NDET(MEC,ISU),INDEX)
      IF(NUM.EQ.0)  GO TO 325
```

```
      CODES=FDET(1,NUM)
      GO TO 325
  322 NUM=IGET(NISO(MEC,ISU),INDEX)
      IF(NUM.EQ.0)  GO TO 325
      CODES=FISO(1,NUM)
      GO TO 325
  323 NUM=IGET(NEPR(MEC,ISU),INDEX)
      IF(NUM.EQ.0)  GO TO 325
      CODES=FEPR(1,NUM)
      GO TO 325
  324 NUM=IGET(NTLR(MEC,ISU),INDEX)
      IF(NUM.EQ.0)  GO TO 325
      CODES=FTLR(1,NUM)
  325 LINE(INDEX,IPR)=NUM
      IF(LINE(INDEX,1).EQ.0)  GO TO 330
      IF(NUM.EQ.0)  GO TO 327
  326 IF(IGET(CODES,4).GT.0)  GO TO 330
  327 FUNBAD(INDEX,IPR)=X
      BFLG=.T.
  330 CONTINUE
      WRITE(6,304)ISU,MEC,((LINE(I,J),J=1,4),I=1,5)
  304 FORMAT(/10X,2I12,5X,5(3X,4I3))
      IF(BFLG)  WRITE(6,305)((FUNBAD(I,J),J=1,4),I=1,5)
  305 FORMAT(39X,5(3X,4A3))
  350 CONTINUE
      GO TO 1
C
C           DISPLAY RELEVANT SPECIFICATIONS FOR ALL DEFINED
C           FUNCTIONS IN THE COVERAGE DATA BASE
C
   90 WRITE(6,400)(STAR,I=1,34)
  400 FORMAT(1H1,48X,*COVERAGE FUNCTION SPECIFICATIONS*/48X,34A1/)
      WRITE(6,401)
  401 FORMAT(//59X,*- DETECTION -*//14X,*FUNCTION  IGFT  INTF  ISCH *,
     *      * IREP    COEF*,7X,*TDEL*,9X,*P1*,10X,*P2*,10X,*P3*,9X,
     *      *TDUR*/15X,*NUMBER*/)
      DO 420  NUM=1,200
      CODES=FDET(1,NUM)
      IF(IGET(CODES,4).EQ.0)  GO TO 420
      IGFT=IGET(CODES,1)
      ISCH=IGET(CODES,2)
      IREP=IGET(CODES,3)
      INTF=IGET(CODES,5)
      WRITE(6,406)NUM,IGFT,INTF,ISCH,IREP,(FDET(I,NUM),I=2,7)
  406 FORMAT(11X,2I8,3I6,F11.6,5E12.4)
  420 CONTINUE
      WRITE(6,402)
  402 FORMAT(//59X,*- ISOLATION -*//14X,*FUNCTION  IGFT  INTF  *,
     *      *  COEF*,7X,*TDEL*,9X,*P1*,10X,*P2*,10X,*P3*,9X,*TDUR*/
     *      15X,*NUMBER*/)
      DO 430  NUM=1,50
      CODES=FISO(1,NUM)
      IF(IGET(CODES,4).EQ.0)  GO TO 430
      IGFT=IGET(CODES,1)
      INTF=IGET(CODES,5)
      WRITE(6,407)NUM,IGFT,INTF,(FISO(I,NUM),I=2,7)
  407 FORMAT(11X,2I8,I6,F11.6,5E12.4)
  430 CONTINUE
      WRITE(6,403)
  403 FORMAT(//51X,*- FAULT PROPAGATION RECOVERY -*/)
      WRITE(6,405)
  405 FORMAT(14X,*FUNCTION  IGFT  INTF    COEF*,8X,*P1*,10X,*P2*,10X,
     *      *P3*,9X,*TDUR*/15X,*NUMBER*/)
      DO 440  NUM=1,25
      CODES=FEPR(1,NUM)
      IF(IGET(CODES,4).EQ.0)  GO TO 440
```

```
      IGFT=IGET(CODES,1)                                                READIN2  402
      INTF=IGET(CODES,5)                                                READIN2  403
      WRITE(6,408)NUM,IGFT,INTF,FEPR(2,NUM),                            READIN2  404
     *     (FEPR(I,NUM),I=4,7)                                          READIN2  405
  408 FORMAT(11X,2I8,I6,F11.6,4E12.4)                                   READIN2  406
  440 CONTINUE                                                          READIN2  407
      WRITE(6,404)                                                      READIN2  408
  404 FORMAT(//55X,*= TIME LOST RECOVERY =*/)                           READIN2  409
      WRITE(6,405)                                                      READIN2  410
      DO 450  NUM=1,25                                                  READIN2  411
      CODES=FTLR(1,NUM)                                                 READIN2  412
      IF(IGET(CODES,4).EQ.0)  GO TO 450                                 READIN2  413
      IGFT=IGET(CODES,1)                                                READIN2  414
      INTF=IGET(CODES,5)                                                READIN2  415
      WRITE(6,408)NUM,IGFT,INTF,FTLR(2,NUM),                            READIN2  416
     *     (FTLR(I,NUM),I=4,7)                                          READIN2  417
  450 CONTINUE                                                          READIN2  418
      GO TO 1                                                           READIN2  419
  900 RETURN                                                            READIN2  420
  999 WRITE(6,99)ID,IP,NUM,IGFT,INTF,ISCH,IREP,COEF,                    READIN2  421
     *     TDEL,P1,P2,P3,TDUR                                           READIN2  422
   99 FORMAT(//30X,*$$$$$ ERROR ON CONTROL CARD $$$$$*/                 READIN2  423
     *     2A3,2I4,2I1,I4,6F15.8)                                       READIN2  424
      IF(DEBUG)  GO TO 1                                                READIN2  425
 1000 STOP                                                              READIN2  426
C                                                                       FINAL     54
C         ENTRY FOR RELIABILITY MODEL DATA BASE DISPLAY,                FINAL     55
C         APPLICABLE PARAMETERS ONLY ARE PRINTED                        FINAL     56
C                                                                       FINAL     57
      ENTRY RITERS                                                      READIN2  427
      WRITE(6,500)NRS,(STAR,I=1,23)                                     READIN2  428
  500 FORMAT(1H1,51X,*DATA BASE FOR RUN-SET*/55X,I7/                    READIN2  429
     *  51X,23A1//5X,*PARA=*/5X,*METER*,49X,*STAGE/EQUATION NUMBER*/)   READIN2  430
      WRITE(6,501)((I,SLASH,PROD(I)),I=1,NPROD)                         READIN2  431
  501 FORMAT(/6X,10(I10,A1,I1))                                         READIN2  432
      CALL TRANSFR(Q1,1)                                                READIN2  433
      CALL TRANSFR(Q2,2)                                                READIN2  434
      CALL TRANSFR(N,3)                                                 READIN2  435
      CALL TRANSFR(S,4)                                                 READIN2  436
      CALL TRANSFR(W,5)                                                 READIN2  437
      CALL TRANSFR(Z,6)                                                 READIN2  438
      CALL TRANSFR(LAM,7)                                               READIN2  439
      CALL TRANSFR(MU,8)                                                READIN2  440
      CALL TRANSFR(GMP,9)                                               READIN2  441
      CALL TRANSFR(C1,10)                                               READIN2  442
      CALL TRANSFR(CD1,11)                                              READIN2  443
      CALL TRANSFR(C2,12)                                               READIN2  444
      CALL TRANSFR(CD2,13)                                              READIN2  445
      CALL TRANSFR(CTR,14)                                              READIN2  446
      CALL TRANSFR(CDTR,15)                                             READIN2  447
      CALL TRANSFR(PRC1,16)                                             READIN2  448
      CALL TRANSFR(PRC2,17)                                             READIN2  449
      CALL TRANSFR(RV,18)                                               READIN2  450
      CALL TRANSFR(P,19)                                                READIN2  451
      IF(.NOT.LVARY)  GO TO 530                                         READIN2  452
      RLFLG=PCW(IRES)                                                   READIN2  453
      PCW(IRES)=OR(RLFLG,MB59)                                          READIN2  454
      NAMSAV=NAMES(IRES)                                                READIN2  455
      WRITE(6,502)NAMSAV                                                READIN2  456
  502 FORMAT(///5X,*ITER=*,35X,*PARAMETER TO BE VARIED = *,A5/          READIN2  457
     *     5X,*ATION*)                                                  READIN2  458
      DO 520  I=1,N10                                                   READIN2  459
      ENCODE(10,503,NAMES(IRES))I                                       READIN2  460
  503 FORMAT(I5,5X)                                                     READIN2  461
      DO 510  J=1,NPROD                                                 READIN2  462
  510 OUT(J)=PARAM(I,J)                                                 READIN2  463
```

```
 520 CALL TRANSFR(OUT,IRES)                                              READIN2   464
     PCW(IRES)=RLFLG                                                     READIN2   465
     NAMES(IRES)=NAMSAV                                                  READIN2   466
 530 IF(.NOT.DMFLG)  RETURN                                              READIN2   467
     WRITE(6,504)NUN,SLH2,SLH3,CCBF,RBGN                                 READIN2   468
 504 FORMAT(//49X,*DUAL MODE SYSTEM PARAMETERS*//40X,                    READIN2   469
   *      *NUMBER OF STAGES*,I8/40X,*CHANNEL FAILURE RATE *,F12.8/       READIN2   470
   *      40X,*SYSTEM FAILURE RATE *,F13.8/40X,*CHANNEL FAILURE COV. *,READIN2   471
   *      F12.8/40X,*REASSIGNMENT SWITCH *,L5//)                         READIN2   472
     RETURN                                                              READIN2   473
     END                                                                 READIN2   474
     SUBROUTINE TRANSFR(X,I)                                             TRANSFR     1
C                                                                        FINAL       1
C          USING APPLICABILITY FIELD OF PCW ELEMENT FOR EACH             FINAL       2
C          PARAMETER, TRANSFR PRINTS THE PARAMETER NAME                  FINAL       3
C          AND VALUES FOR EACH STAGE IN WHICH IT IS USED                 FINAL       4
C                                                                        FINAL       5
     DIMENSION X(10),K(10),R(10)                                         TRANSFR     2
     EQUIVALENCE (K,R)                                                   TRANSFR     3
     DATA B/10H          /                                               TRANSFR     4
     COMMON /GRAPH/IJK(11),N                                             TRANSFR     5
     COMMON /DEFS/P(22),M(22)                                            TRANSFR     6
     INTEGER P                                                           TRANSFR     7
     LOGICAL F1,F2                                                       TRANSFR     8
     DO 5  J=1,N                                                         TRANSFR     9
   5 R(J)=X(J)                                                           TRANSFR    10
     F1=.F.                                                              TRANSFR    11
     IF(P(I))30,200,8                                                    TRANSFR    12
   8 DO 20  J=1,N                                                        TRANSFR    13
     IF(ISHIFT(P(I),J+19))9,200,10                                       TRANSFR    14
   9 F1=.T.                                                              TRANSFR    15
     ENCODE(10,11,R(J))K(J)                                              TRANSFR    16
  11 FORMAT(I6,4X)                                                       TRANSFR    17
     GO TO 20                                                            TRANSFR    18
  10 R(J)=B                                                              TRANSFR    19
  20 CONTINUE                                                            TRANSFR    20
     IF(F1)  GO TO 100                                                   TRANSFR    21
     RETURN                                                              TRANSFR    22
  30 F2=.F.                                                              TRANSFR    23
     DO 40  J=1,N                                                        TRANSFR    24
     IF(ISHIFT(P(I),J+19))35,200,40                                      TRANSFR    25
  35 F1=.T.                                                              TRANSFR    26
     IF(R(J).EQ.0.0)  GO TO 40                                           TRANSFR    27
     IF(R(J).LT.0.001.OR.R(J).GT.9.999)  F2=.T.                          TRANSFR    28
  40 CONTINUE                                                            TRANSFR    29
     IF(F1)  GO TO 45                                                    TRANSFR    30
     RETURN                                                              TRANSFR    31
  45 DO 70  J=1,N                                                        TRANSFR    32
     IF(ISHIFT(P(I),J+19))47,200,60                                      TRANSFR    33
  47 IF(F2)  GO TO 50                                                    TRANSFR    34
     ENCODE(10,12,R(J))R(J)                                              TRANSFR    35
  12 FORMAT(F10.8)                                                       TRANSFR    36
     GO TO 70                                                            TRANSFR    37
  50 ENCODE(10,13,R(J))R(J)                                              TRANSFR    38
  13 FORMAT(E10.4)                                                       TRANSFR    39
     GO TO 70                                                            TRANSFR    40
  60 R(J)=B                                                              TRANSFR    41
  70 CONTINUE                                                            TRANSFR    42
 100 WRITE(6,14)M(I),(R(J),J=1,N)                                        TRANSFR    43
  14 FORMAT(/5X,A5,10(2X,A10))                                           TRANSFR    44
     RETURN                                                              TRANSFR    45
 200 WRITE(6,201)                                                        TRANSFR    46
 201 FORMAT(/10X,*ERROR IN TRANSFR*)                                     TRANSFR    47
     STOP                                                                TRANSFR    48
     END                                                                 TRANSFR    49
     SUBROUTINE COVGEN(NRS,NPROD)                                        COVGEN      1
```

```
C                                                                      FINAL       1
C         COVGEN IS THE DRIVER FOR COVAGE.  IT CONTAINS THE            FINAL       2
C         ALGORITHMS NEEDED TO DETERMINE REQUESTS FOR CALCULATION,     FINAL       3
C         APPLICABILITY OF EACH COVERAGE FACTOR, AND TO                FINAL       4
C         IMPLIMENT THE SUBCLASS TO STAGE LINKAGE.                     FINAL       5
C                                                                      FINAL       6
C         THE ORDER OF PROCESSING ALLOWS ANY SET OF FACTORS TO         FINAL       7
C         BE CALCULATED AT ONE TIME.  (EACH MAY BE SELECTED            FINAL       8
C         BY PRESETTING IT TO -1.0.)                                   FINAL       9
C                                                                      FINAL      10
C       *        COVERAGE FACTORS ARE COMPUTED BY STAGE, WITH      *   COVGEN      2
C       *  *HARD* COVERAGE (PERMANENT FAILURES) PRECEEDING         *   COVGEN      3
C       *  *SOFT* COVERAGE (TRANSIENT ERRORS).  THE COMPUTATIONS   *   COVGEN      4
C       *  ARE MADE ONLY IF REQUESTED AND APPLICABLE TO THE        *   COVGEN      5
C       *  APPROPRIATE RELIABILITY EQUATION.                       *   COVGEN      6
      DIMENSION C1(10),CD1(10),C2(10),CD2(10),PRC1(10),PRC2(10)        COVGEN      7
      COMMON /PARA4/CC(10,2),CCD(10,2),CTR(10),CDTR(10)                COVGEN      8
      COMMON /PARA5/XYZ(30),PPRC(10,2)                                 COVGEN      9
      COMMON /DEFS/PCW(22)                                             COVGEN     10
      COMMON /CVBO/COVINT,IGENC(10),IGENP(10),IFSC(8),FRAC(8)          COVGEN     11
      EQUIVALENCE (CC(1,1),C1(1)), (CC(1,2),C2(1))                     COVGEN     12
      EQUIVALENCE (CCD(1,1),CD1(1)), (CCD(1,2),CD2(1))                 COVGEN     13
      EQUIVALENCE (PPRC(1,1),PRC1(1)), (PPRC(1,2),PRC2(1))             COVGEN     14
      LOGICAL CFLG,PFLG                                                COVGEN     15
      DO 40  I=1,NPROD                                                 COVGEN     16
      I19=I+19                                                         COVGEN     17
      CFLG=(IGENC(I).LT.0).OR.(IGENC(I).GT.0.AND.NRS.EQ.1)             COVGEN     18
      IF(.NOT.(CFLG.OR.C1(I).LT.0.0)) GO TO 5                          COVGEN     20
      IF(ISHIFT(PCW(10),I19))1,200,5                                   COVGEN     21
    1 C1(I)=0.0                                                        COVGEN     22
      DO 2  J=1,8                                                      COVGEN     24
      IF(IFSC(J).NE.I)  GO TO 2                                        COVGEN     25
      C1(I)=C1(I)+FRAC(J)*COVAGE(J,1,1,1)                              POSTPCH     1
    2 CONTINUE                                                         COVGEN     27
    5 IF(.NOT.(CFLG.OR.CD1(I).LT.0.0)) GO TO 10                        COVGEN     28
      IF(ISHIFT(PCW(11),I19))6,200,10                                  COVGEN     29
    6 CTMP=0.0                                                         COVGEN     30
      DO 7  J=1,8                                                      COVGEN     31
      IF(IFSC(J).NE.I)  GO TO 7                                        COVGEN     32
      CTMP=CTMP+FRAC(J)*COVAGE(J,1,1,2)                                POSTPCH     2
    7 CONTINUE                                                         COVGEN     34
      IF(C1(I).GT.0.0)  CD1(I)=CTMP/C1(I)                              COVGEN     35
   10 IF(.NOT.(CFLG.OR.C2(I).LT.0.0)) GO TO 15                         COVGEN     37
      IF(ISHIFT(PCW(12),I19))11,200,15                                 COVGEN     38
   11 C2(I)=0.0                                                        COVGEN     39
      DO 12  J=1,8                                                     COVGEN     41
      IF(IFSC(J).NE.I)  GO TO 12                                       COVGEN     42
      C2(I)=C2(I)+FRAC(J)*COVAGE(J,2,1,1)                              POSTPCH     3
   12 CONTINUE                                                         COVGEN     44
   15 IF(.NOT.(CFLG.OR.CD2(I).LT.0.0)) GO TO 20                        COVGEN     45
      IF(ISHIFT(PCW(13),I19))16,200,20                                 COVGEN     46
   16 CTMP=0.0                                                         COVGEN     47
      DO 17  J=1,8                                                     COVGEN     48
      IF(IFSC(J).NE.I)  GO TO 17                                       COVGEN     49
      CTMP=CTMP+FRAC(J)*COVAGE(J,2,1,2)                                POSTPCH     4
   17 CONTINUE                                                         COVGEN     51
      IF(C2(I).GT.0.0)  CD2(I)=CTMP/C2(I)                              COVGEN     52
   20 IF(.NOT.(CFLG.OR.CTR(I).LT.0.0)) GO TO 25                        COVGEN     54
      IF(ISHIFT(PCW(14),I19))21,200,25                                 COVGEN     55
   21 CTR(I)=0.0                                                       COVGEN     56
      DO 22  J=1,8                                                     COVGEN     58
      IF(IFSC(J).NE.I)  GO TO 22                                       COVGEN     59
      CTR(I)=CTR(I)+FRAC(J)*COVAGE(J,0,1,0)                            POSTPCH     5
   22 CONTINUE                                                         COVGEN     61
   25 IF(.NOT.(CFLG.OR.CDTR(I).LT.0.0)) GO TO 30                       COVGEN     62
      IF(ISHIFT(PCW(15),I19))26,200,30                                 COVGEN     63
```

```
26 CTMP=0.0                                                         COVGEN    64
   DO 27  J=1,8                                                     COVGEN    65
   IF(IFSC(J).NE.I)  GO TO 27                                       COVGEN    66
   CTMP=CTMP+FRAC(J)*COVAGE(J,0,1,1)                                POSTPCH    6
27 CONTINUE                                                         COVGEN    68
   IF(CTR(I).GT.0.0)  CDTR(I)=CTMP/CTR(I)                           COVGEN    69
30 PFLG=(IGENP(I).LT.0).OR.(IGENP(I).GT.0.AND.NRS.EQ.1)             COVGEN    70
   IF(.NOT.(PFLG.OR.PRC1(I).LT.0.0)) GO TO 35                       COVGEN    71
   IF(ISHIFT(PCW(16),I19))31,200,35                                 COVGEN    72
31 PRC1(I)=0.0                                                      COVGEN    73
   DO 32  J=1,8                                                     COVGEN    74
   IF(IFSC(J).NE.I)  GO TO 32                                       COVGEN    75
   PRC1(I)=PRC1(I)+FRAC(J)*COVAGE(J,1,2,0)                          POSTPCH    7
32 CONTINUE                                                         COVGEN    77
35 IF(.NOT.(PFLG.OR.PRC2(I).LT.0.0)) GO TO 40                       COVGEN    78
   IF(ISHIFT(PCW(17),I19))36,200,40                                 COVGEN    79
36 PRC2(I)=0.0                                                      COVGEN    80
   DO 37  J=1,8                                                     COVGEN    81
   IF(IFSC(J).NE.I)  GO TO 37                                       COVGEN    82
   PRC2(I)=PRC2(I)+FRAC(J)*COVAGE(J,2,2,0)                          POSTPCH    8
37 CONTINUE                                                         COVGEN    84
40 CONTINUE                                                         COVGEN    85
   RETURN                                                           COVGEN    86
200 WRITE(6,201)                                                    COVGEN    87
201 FORMAT(/10X,*ERROR IN COVGEN*)                                  COVGEN    88
   STOP                                                             COVGEN    89
   END                                                              COVGEN    90
   SUBROUTINE PGET(IR,I,NPROD)                                      PGET       1
C                                                                   FINAL      1
C        TRANSFER A ROW OF ARRAY PARAM TO THE BASE-RUN              FINAL      2
C        VECTOR OF THE PARAMETER OF INTEREST                        FINAL      3
C                                                                   FINAL      4
   COMMON /PARA1/QQ(10,2),IJK(10),N(10),KS(10)                      PGET       2
   COMMON /PARA2/XLAM(10),XMU(10)                                   PGET       3
   COMMON /PARA3/RV(10),KZ(10),KW(10),P(10)                         PGET       4
   COMMON /PARA4/CC(10,2),CCD(10,2),CTR(10),CDTR(10)                PGET       5
   COMMON /PARA5/XYZ(10,2),GMP(10),PPRC(10,2)                       PGET       6
   COMMON /PARM/PARAM(16,10)                                        PGET       7
   DO 20  J=1,NPROD                                                 PGET       8
   VAL=PARAM(I,J)                                                   PGET       9
   GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19),IR       PGET      10
 1 QQ(J,1)=VAL                                                      PGET      11
   GO TO 20                                                         PGET      12
 2 QQ(J,2)=VAL                                                      PGET      13
   GO TO 20                                                         PGET      14
 3 N(J)=VAL                                                         PGET      15
   GO TO 20                                                         PGET      16
 4 KS(J)=VAL                                                        PGET      17
   GO TO 20                                                         PGET      18
 5 KW(J)=VAL                                                        PGET      19
   GO TO 20                                                         PGET      20
 6 KZ(J)=VAL                                                        PGET      21
   GO TO 20                                                         PGET      22
 7 XLAM(J)=VAL                                                      PGET      23
   GO TO 20                                                         PGET      24
 8 XMU(J)=VAL                                                       PGET      25
   GO TO 20                                                         PGET      26
 9 GMP(J)=VAL                                                       PGET      27
   GO TO 20                                                         PGET      28
10 CC(J,1)=VAL                                                      PGET      29
   GO TO 20                                                         PGET      30
11 CCD(J,1)=VAL                                                     PGET      31
   GO TO 20                                                         PGET      32
12 CC(J,2)=VAL                                                      PGET      33
   GO TO 20                                                         PGET      34
13 CCD(J,2)=VAL                                                     PGET      35
```

```
      GO TO 20                                                            PGET   36
   14 CTR(J)=VAL                                                          PGET   37
      GO TO 20                                                            PGET   38
   15 CDTR(J)=VAL                                                         PGET   39
      GO TO 20                                                            PGET   40
   16 PPRC(J,1)=VAL                                                       PGET   41
      GO TO 20                                                            PGET   42
   17 PPRC(J,2)=VAL                                                       PGET   43
      GO TO 20                                                            PGET   44
   18 RV(J)=VAL                                                           PGET   45
      GO TO 20                                                            PGET   46
   19 P(J)=VAL                                                            PGET   47
   20 CONTINUE                                                            PGET   48
      RETURN                                                              PGET   49
      END                                                                 PGET   50
      SUBROUTINE DCR(J,UNITR,RELM1,RSYS)                                  DCR     1
C                                                                         FINAL   1
C           COMPUTE THE DUAL MODE SYSTEM RELIABILITY, RSYS,               FINAL   2
C           FOR THE JTH TIME STEP.  ALSO RETURN VALUES FOR                FINAL   3
C           SYSTEM AND INDIVIDUAL STAGE RELIABILITIES                     FINAL   4
C           ASSUMING MODE 1 OPERATION                                     FINAL   5
C                                                                         FINAL   6
C           NOTE THAT DUE TO THE USE OF STRUCTURED NUMERICAL              FINAL   7
C           INTEGRATIONS IN SUBPROGRAMS DCT2 AND DCT, DCR MAY             FINAL   8
C           NOT BE CALLED FOR AN ARBITRARY VALUE OF CLT=TIME,             FINAL   9
C                                                                         FINAL  10
      DIMENSION UNITR(8)                                                  DCR     2
      COMMON /DRIVE/T,XYZ(2),NUN                                          DCR     3
      COMMON /PARA1/YXZ(20),IJK(20),NS(10)                                DCR     4
      COMMON /PARA2/ZYX(40),SLH2,SLH3                                     DCR     5
      RSYS =1.0                                                           DCR     6
      IF(J.EQ.1) GO TO 100                                                DCR    10
      DO 10  IX=1,NUN                                                     DCR    11
      UNITR(IX)=DCRU(IX,1,NS(IX),T)                                       DCR    12
   10 RSYS=RSYS*UNITR(IX)                                                 DCR    13
      RSYS=RSYS*EXP(-SLH2*T)                                              DCR    14
      ESLH3=EXP(-SLH3*T)                                                  DCR    15
      RELM1=RSYS*ESLH3                                                    DCR    16
      DO 20  IX=1,NUN                                                     DCR    17
   20 RSYS=RSYS+DCT(IX,J)                                                 DCR    18
      RSYS=RSYS+DCT2(J)                                                   DCR    19
      RSYS=RSYS*ESLH3                                                     DCR    20
   99 RETURN                                                              DCR    21
  100 RELM1=RSYS                                                          DCR    22
      DO 110  IX=1,NUN                                                    DCR    23
  110 UNITR(IX)=RSYS                                                      DCR    24
      GO TO 99                                                            DCR    25
      END                                                                 DCR    26
      FUNCTION DCT2(J)                                                    DCT2    1
C                                                                         FINAL   1
C           FOR THE JTH TIME STEP, COMPUTE THE PROBABILITY                FINAL   2
C           THAT THE DUAL MODE SYSTEM WILL HAVE SURVIVED                  FINAL   3
C           AND HAVE DEGRADED DUE TO A CATAGORY 2 SWITCH FAILURE          FINAL   4
C                                                                         FINAL   5
      COMMON /DRIVE/T,STEP,HSTEP,NUN                                      DCT2    2
      COMMON /PARA2/XYZ(40),SLH2                                          DCT2    3
      COMMON /PARA4/ZYX(60),CCSF                                          DCT2    4
      JN=2*J-1                                                            DCT2    5
      DCT2=0.0                                                            DCT2    6
      DO 50  IT=1,JN                                                      DCT2    7
      TAU=HSTEP*(IT-1)                                                    DCT2    8
      PART =1.0                                                           DCT2    9
      DO 10  IX=1,NUN                                                     DCT2   10
   10 PART=PART*DCS(IX,T,TAU)                                             DCT2   11
      PART=PART*EXP(-SLH2*TAU)                                            DCT2   12
      IF(IT.GT.1)  GO TO 30                                               DCT2   13
```

```
20 DCT2=DCT2+PART                                          DCT2      14
   GO TO 50                                                DCT2      15
30 IF(IT.EQ.JN) GO TO 20                                   DCT2      16
   IF(IT.EQ.2*(IT/2)) GO TO 40                             DCT2      17
   DCT2=DCT2+2.0*PART                                      DCT2      18
   GO TO 50                                                DCT2      19
40 DCT2=DCT2+4.0*PART                                      DCT2      20
50 CONTINUE                                                DCT2      21
   DCT2=SLH2*CCSF*HSTEP*DCT2/3.0                           DCT2      22
   RETURN                                                  DCT2      23
   END                                                     DCT2      24
   FUNCTION DCT(IUN,J)                                     DCT        1
C                                                          FINAL      1
C        FOR THE JTH TIME STEP, COMPUTE THE PROBABILITY    FINAL      2
C        THAT THE DUAL MODE SYSTEM WILL HAVE SURVIVED      FINAL      3
C        AND HAVE DEGRADED DUE TO A FAILURE IN STAGE IUN   FINAL      4
C                                                          FINAL      5
   COMMON /DRIVE/T,STEP,HSTEP,NUN,RSGN                     DCT        2
   COMMON /PARA1/XYZ(20),IDQ(10)                           DCT        3
   COMMON /PARA2/ZYX(40),SLH2                              DCT        4
   LOGICAL RSGN                                            DCT        5
   JN=2*J-1                                                DCT        6
   DCT=0.0                                                 DCT        7
   IF(IDQ(IUN).LE.0)  RETURN                               DCT        8
   LR=0                                                    DCT        9
   IF(RSGN)  LR=IDQ(IUN)-1                                 DCT       10
   DO 50  IT=1,JN                                          DCT       11
   TAU=HSTEP*(IT-1)                                        DCT       12
   PART=DCRU(IUN,2,LR,T-TAU)*DCH(IUN,TAU)                  DCT       13
   DO 10  JUN=1,NUN                                        DCT       14
   IF(JUN.EQ.IUN)  GO TO 10                                DCT       15
   PART=PART*DCS(JUN,T,TAU)                                DCT       16
10 CONTINUE                                                DCT       17
   PART=PART*EXP(-SLH2*TAU)                                DCT       18
   IF(IT.GT.1) GO TO 30                                    DCT       19
20 DCT=DCT+PART                                            DCT       20
   GO TO 50                                                DCT       21
30 IF(IT.EQ.JN) GO TO 20                                   DCT       22
   IF(IT.EQ.2*(IT/2)) GO TO 40                             DCT       23
   DCT=DCT+2.0*PART                                        DCT       24
   GO TO 50                                                DCT       25
40 DCT=DCT+4.0*PART                                        DCT       26
50 CONTINUE                                                DCT       27
   DCT=HSTEP*DCT/3.0                                       DCT       28
   RETURN                                                  DCT       29
   END                                                     DCT       30
   FUNCTION DCS(IUN,T,TAU)                                 DCS        1
C                                                          FINAL      1
C        COMPUTE THE CONDITIONAL PROBABILITY THAT STAGE IUN FINAL     2
C        OF THE DUAL MODE SYSTEM CAN SURVIVE IN MODE 2     FINAL      3
C        FROM TAU TO T GIVEN THAT DEGRADING OCCURRED       FINAL      4
C        AT TIME TAU                                       FINAL      5
C                                                          FINAL      6
   COMMON /DRIVE/XYZ(4),RSGN                               DCS        2
   COMMON /PARA1/ZYX(20),IDQ(10),IJK(10),S(10)             DCS        3
   COMMON /PARA2/YXZ(10),MU(10),YZX(10),BCOMPS(10)         DCS        4
   INTEGER S,POW                                           DCS        5
   REAL MU,MUXT                                            DCS        6
   LOGICAL BCOMPS,RSGN                                     DCS        7
   DCS=0.0                                                 DCS        8
   NS=S(IUN)                                               DCS        9
   NSP1=NS+1                                               POSTPCH    1
   DO 50  LDP1=1,NSP1                                      POSTPCH    2
   LD=LDP1-1                                               POSTPCH    3
   NSL=NS-LD                                               DCS       11
   IF(.NOT.BCOMPS(IUN))  GO TO 40                          DCS       12
```

```
      JP=NSL                                                           DCS      13
      IF(RSGN)  JP=JP+IDQ(IUN)                                         DCS      14
      PROD=DCG(IUN,1,LD,TAU)*DCRU(IUN,2,JP,T-TAU)                      DCS      15
      GO TO 50                                                         DCS      16
   40 NSLP1=NSL+1                                                      POSTPCH   4
      DO 50  JDP1=1,NSLP1                                              POSTPCH   5
      JD=JDP1-1                                                        POSTPCH   6
      PROD=FNCK(NSL,JD)                                                DCS      18
      MUXT=TAU*MU(IUN)                                                 DCS      19
      PROD=PROD*EXP(-JD*MUXT)                                          DCS      20
      ONEMX=1.0-EXP(-MUXT)                                             DCS      21
      POW=NSL-JD                                                       DCS      22
      IF(ONEMX.LE.0.0.AND.POW.LE.0)  GO TO 45                          DCS      23
      PROD=PROD*ONEMX**POW                                             DCS      24
   45 PROD=PROD*DCG(IUN,1,LD,TAU)                                      DCS      25
      JP=JD                                                            DCS      26
      IF(RSGN)  JP=JP+IDQ(IUN)                                         DCS      27
      PROD=PROD*DCRU(IUN,2,JP,T-TAU)                                   DCS      28
   50 DCS=DCS+PROD                                                     DCS      29
      RETURN                                                           DCS      30
      END                                                              DCS      31
      FUNCTION DCH(IUN,TAU)                                            DCH       1
C                                                                      FINAL     1
C           COMPUTE THE PROBABILITY DENSITY OF A DEGENERATIVE          FINAL     2
C           FAILURE IN STAGE IUN OF THE DUAL MODE SYSTEM BY            FINAL     3
C           TIME T                                                     FINAL     4
C                                                                      FINAL     5
      COMMON /PARA1/QQ(10,2),IJK(20),S(10)                             DCH       2
      COMMON /PARA2/LAM(10),MU(10),K(10),BCOMPS(10)                    DCH       3
      COMMON /PARA4/XYZ(40),CTR(10),CDTR(10)                           DCH       4
      INTEGER S                                                        DCH       5
      REAL LAM,K,MU                                                    DCH       6
      LOGICAL BCOMPS                                                   DCH       7
      NS=S(IUN)                                                        DCH       8
      IF(BCOMPS(IUN))  GO TO 60                                        DCH       9
      DCH=0.0                                                          DCH      10
      NSP1=NS+1                                                        POSTPCH   1
      DO 50  IDP1=1,NSP1                                               POSTPCH   2
      ID=IDP1-1                                                        POSTPCH   3
      PROD=DCG(IUN,1,NS-ID,TAU)*CTR(IUN)                               DCH      12
      IF(ID.EQ.0)  GO TO 50                                            DCH      13
      PART=CDTR(IUN)*(1.0-EXP(-MU(IUN)*TAU))                           DCH      14
      DO 25  II=1,ID                                                   DCH      15
   25 PROD=PROD*PART                                                   DCH      16
   50 DCH=DCH+PROD                                                     DCH      17
   55 DCH=LAM(IUN)*QQ(IUN,1)*DCH                                       DCH      18
      RETURN                                                           DCH      19
   60 DCH=DCG(IUN,1,NS,TAU)*CTR(IUN)                                   DCH      20
      GO TO 55                                                         DCH      21
      END                                                              DCH      22
      FUNCTION DCRU(IUN,MD,L,TAU)                                      DCRU      1
C                                                                      FINAL     1
C           COMPUTE THE PROBABILITY OF USING AT MOST L SPARE           FINAL     2
C           LRU#S IN STAGE IUN OF THE DUAL MODE SYSTEM                 FINAL     3
C           (OPERATING IN MODE MD) BY TIME TAU                         FINAL     4
C                                                                      FINAL     5
      DCRU=0.0                                                         DCRU      2
      IF(L.LT.0)  RETURN                                               DCRU      3
      LP1=L+1                                                          POSTPCH   1
      DO 50  IDP1=1,LP1                                                POSTPCH   2
      ID=IDP1-1                                                        POSTPCH   3
   50 DCRU=DCRU+DCG(IUN,MD,ID,TAU)                                     DCRU      5
      RETURN                                                           DCRU      6
      END                                                              DCRU      7
      FUNCTION DCG(IUN,MD,I,T)                                         DCG       1
C                                                                      FINAL     1
```

```
C          COMPUTE THE PROBABILITY OF USING EXACTLY I SPARE           FINAL        2
C          LRU#S IN STAGE IUN OF THE DUAL MODE SYSTEM                 FINAL        3
C          (OPERATING IN MODE MD) BY TIME T                           FINAL        4
C                                                                     FINAL        5
      COMMON /PARA1/QQ(10,2)                                          DCG          2
      COMMON /PARA2/LAM(10),MU(10),K(10),BCOMPS(10)                   DCG          3
      COMMON /PARA4/CC(10,2),CCD(10,2)                                DCG          4
      COMMON /PARA5/GGM(10,2)                                         DCG          5
      REAL LAM,K,MU                                                   DCG          6
      LOGICAL BCOMPS                                                  DCG          7
      Q=QQ(IUN,MD)                                                    DCG          8
      DCG=1.0                                                         DCG          9
      IF(BCOMPS(IUN))  GO TO 30                                       DCG         10
      IF(I.LE.0) GO TO 20                                             DCG         11
      PART=CCD(IUN,MD)*(1.0-EXP(-MU(IUN)*T))                          DCG         12
      DO 10  ID=1,I                                                   DCG         13
   10 DCG=DCG*PART                                                    DCG         14
   20 DCG=DCG*DCOMB(K(IUN)*Q*CC(IUN,MD)/CCD(IUN,MD)+(I-1),I)          DCG         15
   25 DCG=DCG*EXP(-Q*LAM(IUN)*T)                                      DCG         16
      DCG=DCG*EXP(-GGM(IUN,MD)*T)                                     DCG         17
      RETURN                                                          DCG         18
   30 IF(I.LE.0)  GO TO 25                                            DCG         19
      PART=Q*CC(IUN,MD)*LAM(IUN)*T                                    DCG         20
      DO 40  ID=1,I                                                   DCG         21
   40 DCG=DCG*PART/ID                                                 DCG         22
      GO TO 25                                                        DCG         23
      END                                                             DCG         24
      FUNCTION DCOMB(TOP,K)                                           DCOMB        1
C                                                                     FINAL        1
C          DCOMB RETURNS THE BINOMIAL COEFFICIENT OF THE              FINAL        2
C          EXPRESSION (TOP C K)                                       FINAL        3
C                                                                     FINAL        4
      COM=1.0                                                         DCOMB        2
      IF(K.LE.0)  GO TO 99                                            DCOMB        3
      DO 30  I=1,K                                                    DCOMB        4
   30 COM=COM*(TOP-I+1)                                               DCOMB        5
      COM=COM/FFAC(K)                                                 DCOMB        6
   99 DCOMB=COM                                                       DCOMB        7
      RETURN                                                          DCOMB        8
      END                                                             DCOMB        9
C                                                                     FINAL        1
C          COMPUTE THE COVERAGE PROVIDED BY UP TO 20 D/I/R/           FINAL        2
C          MECHANISMS COMPETING FOR THE DETECTION AND ISOLATION       FINAL        3
C          OF, AND THE RECOVERY FROM FAULTS OF SUBCLASS ISU,          FINAL        4
C          WHEN JS SPARE LRU#S MUST BE CHECKED OUT AS PART            FINAL        5
C          OF THE RECOVERY PROCESS                                    FINAL        6
C                                                                     FINAL        7
C          MECHANISM CHARACTERISTICS WILL DEPEND ON SYSTEM            FINAL        8
C          OPERATIONAL MODE (MD) AND FAULT TYPE (IET)                 FINAL        9
C                                                                     FINAL       10
      FUNCTION COVAGE(ISU,MD,IET,JS)                                  POSTPCH      1
      DIMENSION PDIRAR(20)                                            COVAGE       2
      COMMON /CVB0/COVINT                                             COVAGE       3
      COMMON /CVB1/NDET(20,8),NISO(20,8),NEPR(20,8),NTLR(20,8)        COVAGE       4
     1      /CVB2/FDET(7,200),FISO(7,50),FEPR(7,25),FTLR(7,25)        COVAGE       5
     2      /CVB3/NDSAV(20),NISAV(20),NR1SAV(20),NR2SAV(20)           COVAGE       6
     3      /CVB7/ICHAR(20),PDETAR(20),IREPAR(20),PERIAR(20),TDELAR(20) COVAGE     7
     4      /CVB4/P1,P2,P3,TDUR,IGFT                                  COVAGE       8
     5      /CVB5/PFDS(8),TFDS(8)  /CVB6/LCM,TMINOR,FACTOR            COVAGE       9
     6      /CVB8/GPFLG,GPAR(20,101) /CVB9/N                          COVAGE      10
      LOGICAL IMPG,IMPH,FOUT,FIN,GPFLG,COVINT                         POSTPCH      2
      NP1=N+1                                                         POSTPCH      3
      INDEX=2*IET+MD-2                                                COVAGE      14
      IF(MD.EQ.0)  INDEX=5                                            COVAGE      15
      FAILFAC=PFDS(ISU)**JS                                           FINAL       11
      LCM=1                                                           COVAGE      16
```

```
C
C         PREPROCESSING LOOP:
C         DETERMINES THE CHARACTERISTIC CODE FOR EACH DETECTOR
C         (0 IS USED FOR ANY MECHANISM WHICH IS NOT PROPERLY
C         DEFINED) AND LOCALIZES OTHER PARAMETERS
C
      DO 1  I=1,20
      NERRS=0
      ICHAR(I)=0
      NUM=IGET(NDET(I,ISU),INDEX)
      IF(NUM.EQ.0)  GO TO 1
      NDSAV(I)=NUM
      CODES=FDET(1,NUM)
      IF(IGET(CODES,4).EQ.0)  NERRS=NERRS+1
      KGFT=IGET(CODES,1)
      KSCH=IGET(CODES,2)
      KREP=IGET(CODES,3)
      IF(KSCH.EQ.1.AND.KREP.EQ.0)  NERRS=NERRS+1
      IREPAR(I)=KREP
      PERIAR(I)=TMINOR*KREP
      PDETAR(I)=FDET(2,NUM)
      TDELAR(I)=FDET(3,NUM)
      PDIRAR(I)=PDETAR(I)
      NUM=IGET(NISO(I,ISU),INDEX)
      IF(NUM.GT.0)  GO TO 310
      NERRS=NERRS+1
      GO TO 320
  310 NISAV(I)=NUM
      CODES=FISO(1,NUM)
      IF(IGET(CODES,4).EQ.0)  NERRS=NERRS+1
      PDIRAR(I)=PDIRAR(I)*FISO(2,NUM)
  320 NUM=IGET(NEPR(I,ISU),INDEX)
      IF(NUM.GT.0)  GO TO 330
      NERRS=NERRS+1
      GO TO 340
  330 NR1SAV(I)=NUM
      CODES=FEPR(1,NUM)
      IF(IGET(CODES,1).EQ.0.OR.IGET(CODES,4).EQ.0)  NERRS=NERRS+1
      PDIRAR(I)=PDIRAR(I)*FEPR(2,NUM)
  340 NUM=IGET(NTLR(I,ISU),INDEX)
      IF(NUM.GT.0)  GO TO 350
      NERRS=NERRS+1
      GO TO 360
  350 NR2SAV(I)=NUM
      CODES=FTLR(1,NUM)
      IF(IGET(CODES,1).EQ.0.OR.IGET(CODES,4).EQ.0)  NERRS=NERRS+1
      PDIRAR(I)=PDIRAR(I)*FTLR(2,NUM)
  360 IF(PDIRAR(I).LE.0.0)  NERRS=NERRS+1
      IF(NERRS.EQ.0)  GO TO 201
      WRITE(6,11)NERRS,I,ISU,MD,IET
   11 FORMAT(I7,* ERROR(S) IN COV. MECH.*,I3,
     1     * WHEN  ISU =*,I2,*, MD =*,I2,*, IET =*,I2)
      GO TO 1
  201 IF(KSCH.NE.1)  GO TO 210
      IF(KREP.GT.LCM)  GO TO 205
      IF(LCM.GT.KREP)  GO TO 206
      GO TO 210
  205 IA=KREP
      IB=LCM
      GO TO 207
  206 IA=LCM
      IB=KREP
  207 M2=IA
      M1=IB
  208 MM=M2/M1
      M0=M2-MM*M1
```

```
      IF(MU.EQ.0)  GO TO 209                                            COVAGE    77
      M2=M1                                                             COVAGE    78
      M1=M0                                                             COVAGE    79
      GO TO 208                                                         COVAGE    80
  209 LCM=IA*IB/M1                                                      COVAGE    81
  210 ICHAR(I)=2*KSCH+1                                                 COVAGE    82
      IF(KGFT.GT.0)  ICHAR(I)=ICHAR(I)+1                                COVAGE    83
    1 CONTINUE                                                          COVAGE    84
      FACTOR=1.0/(LCM*TMINOR)                                           COVAGE    85
      COVAGE=0.0                                                        POSTPCH    4
      IF(COVINT)  WRITE(6,12)ISU,MD,IET,JS                              COVAGE    87
   12 FORMAT(/10X,*COVERAGE CONTRIBUTIONS -  ISU =*,I2,* MD =*,I2,      COVAGE    88
     *      * IET =*,I2,* JS =*,I2/15X,*MECHANISM  CONTRIBUTION*)       COVAGE    89
      SDEL=JS*TFDS(ISU)                                                 COVAGE    90
      GPFLG=.F.                                                         COVAGE    91
C                                                                       FINAL     18
C          EVALUATE SCHEDULED DETECTORS FIRST, WITH GPFLG=.F.           FINAL     19
C          THE GPAR ARRAY CONTAINS N+1 SAMPLES OF THE G-PRIME           FINAL     20
C          FUNCTIONS FOR THESE DETECTORS.  ON THE SECOND                FINAL     21
C          PASS, GPFLG=.T., THE GPAR ARRAY CONTAINS THE CUMULATIVE      FINAL     22
C          INTEGRALS OF ITS FORMAR CONTENTS, AND UNSCHEDULED            FINAL     23
C          DETECTORS ARE EVALUATED.                                     FINAL     24
C                                                                       FINAL     25
      DO 101  IJK=1,2                                                   COVAGE    92
      IF(GPFLG)  GO TO 6                                                COVAGE    93
      DO 5  I=1,20                                                      COVAGE    94
      ICH=ICHAR(I)                                                      COVAGE    95
      IF(ICH.LT.3)  GO TO 5                                             COVAGE    96
      IF(ICH.GT.3)  GO TO 4                                             COVAGE    97
      CALL CVGP3(I)                                                     COVAGE    98
      GO TO 5                                                           COVAGE    99
    4 CALL CVGP4(I)                                                     COVAGE   100
    5 CONTINUE                                                          COVAGE   101
      GO TO 9                                                           COVAGE   102
    6 DO 8  I=1,20                                                      COVAGE   103
      IF(ICHAR(I).LT.3)  GO TO 8                                        COVAGE   104
      DT=PERIAR(I)/N                                                    COVAGE   105
      A=GPAR(I,1)                                                       COVAGE   106
      GPAR(I,1)=0.0                                                     COVAGE   107
      B=GPAR(I,2)                                                       COVAGE   108
      GPAR(I,2)=DT*(A+B)/2.0                                            COVAGE   109
      AREA=0.0                                                          COVAGE   110
      DO 7  IT=2,N                                                      COVAGE   111
      C=GPAR(I,IT+1)                                                    COVAGE   112
      GPAR(I,IT+1)=AREA+DT*(A+4.0*B+C)/3.0                              COVAGE   113
      A=B                                                               COVAGE   114
      B=C                                                               COVAGE   115
    7 AREA=GPAR(I,IT)                                                   COVAGE   116
    8 CONTINUE                                                          COVAGE   117
C                                                                       FINAL     26
C          COMPUTE COVERAGE CONTRIBUTIONS FOR EACH PROPERLY             FINAL     27
C          DEFINED MECHANISM, USING A DOUBLE SIMPSON INTEGRAL.          FINAL     28
C          IMPULSE FUNCTIONS ARE TREATED SPECIALLY IN THE               FINAL     29
C          INTEGRATION ALGORITHM.                                       FINAL     30
C                                                                       FINAL     31
    9 DO 100  I=1,20                                                    COVAGE   118
      ICH=ICHAR(I)                                                      COVAGE   119
      IF(ICH.EQ.0)  GO TO 100                                           COVAGE   120
      IF(ICH.GE.3.AND.GPFLG)  GO TO 100                                 COVAGE   121
      IF(ICH.LE.2.AND..NOT.GPFLG)  GO TO 100                            COVAGE   122
      NW=NDSAV(I)                                                       COVAGE   123
      NX=NISAV(I)                                                       COVAGE   124
      NY=NR1SAV(I)                                                      COVAGE   125
      NZ=NR2SAV(I)                                                      COVAGE   126
      IMPG=ICH.EQ.1                                                     COVAGE   127
      IMP=FIGET(FISO(1,NX),1).EQ.                                       COVAGE   128
```

```
      IF(IMPG)  GO TO 10                                                COVAGE 129
      IF(ICH.LE.2)  TGFIN=TDELAR(I)+FDET(7,NW)                          COVAGE 130
      IF(ICH.GE.3)  TGFIN=PERIAR(I)                                     COVAGE 131
      TR1FIN=FEPR(7,NY)                                                 COVAGE 132
      UPOUT=AMIN1(TGFIN,TR1FIN)                                         COVAGE 133
      BOUT=0.0                                                          COVAGE 134
      IF(ICH.LE.2)  BOUT=TDELAR(I)                                      COVAGE 135
      DOUT=(UPOUT-BOUT)/N                                               COVAGE 136
      FOUT=.F.                                                          COVAGE 137
      SOUT=0.0                                                          COVAGE 138
   10 HDEL=FISO(3,NX)                                                   COVAGE 139
      THFIN=HDEL+FISO(7,NX)                                             COVAGE 140
      TR2FIN=FTLR(7,NZ)                                                 COVAGE 141
C                                                                       FINAL   32
C          BEGIN OUTER INTEGRAL                                         FINAL   33
C                                                                       FINAL   34
      DO 70  IOUTP1=1,NP1                                               POSTPCH  5
      IOUT=IOUTP1-1                                                     POSTPCH  6
      TAU=DOUT*IOUT+BOUT                                                COVAGE 143
      GO TO (21,22,23,23),ICH                                           COVAGE 144
   21 TAU=TDELAR(I)                                                     COVAGE 145
      VOUT=CVG1(I,TAU)                                                  COVAGE 146
      GO TO 25                                                          COVAGE 147
   22 VOUT=CVG2(I,TAU)                                                  COVAGE 148
      GO TO 25                                                          COVAGE 149
   23 VOUT=CVGS(I,TAU)                                                  COVAGE 150
   25 VOUT=VOUT*FEVAL(FEPR,NY,TAU)                                      COVAGE 151
      IF(IMPH)  GO TO 30                                                COVAGE 152
      UPIN=AMIN1((THFIN+SDEL),(TR2FIN-TAU))                             COVAGE 153
      BIN=HDEL+SDEL                                                     COVAGE 154
      DIN=(UPIN-BIN)/N                                                  COVAGE 155
      FIN=.F.                                                           COVAGE 156
      SIN=0.0                                                           COVAGE 157
C                                                                       FINAL   35
C          BEGIN INNER INTEGRAL                                         FINAL   36
C                                                                       FINAL   37
   30 DO 50  IINP1=1,NP1                                                POSTPCH  7
      IIN=IINP1-1                                                       POSTPCH  8
      IF(IMPH)  GO TO 35                                                COVAGE 159
      TAUP=DIN*IIN+BIN                                                  COVAGE 160
      VIN=FEVAL(FISO,NX,TAUP-BIN)                                       COVAGE 161
      GO TO 40                                                          COVAGE 162
   35 TAUP=FISO(3,NX)+SDEL                                              COVAGE 163
      VIN=1.0                                                           COVAGE 164
   40 VIN=VIN*FEVAL(FTLR,NZ,TAU+TAUP)                                   COVAGE 165
      IF(IMPH)  GO TO 55                                                COVAGE 166
      IF(IIN.EQ.0.OR.IIN.EQ.N)  GO TO 45                                COVAGE 167
      IF(FIN)  GO TO 42                                                 COVAGE 168
      SIN=SIN+2.0*VIN                                                   COVAGE 169
      GO TO 50                                                          COVAGE 170
   42 SIN=SIN+4.0*VIN                                                   COVAGE 171
      GO TO 50                                                          COVAGE 172
   45 SIN=SIN+VIN                                                       COVAGE 173
   50 FIN=.NOT.FIN                                                      COVAGE 174
      ANSIN=DIN*SIN/3.0                                                 COVAGE 175
      GO TO 60                                                          COVAGE 176
   55 ANSIN=VIN                                                         COVAGE 177
   60 VOUT=VOUT*ANSIN                                                   COVAGE 178
      IF(IMPG)  GO TO 75                                                COVAGE 179
      IF(IOUT.EQ.0.OR.IOUT.EQ.N)  GO TO 65                              COVAGE 180
      IF(FOUT)  GO TO 62                                                COVAGE 181
      SOUT=SOUT+2.0*VOUT                                                COVAGE 182
      GO TO 70                                                          COVAGE 183
   62 SOUT=SOUT+4.0*VOUT                                                COVAGE 184
      GO TO 70                                                          COVAGE 185
   65 SOUT=SOUT+VOUT                                                    COVAGE 186
```

```
70 POUT=,NUT,POUT                                               COVAGE  187
   ANSOUT=DOUT*SOUT/3.0                                         COVAGE  188
   GO TO 80                                                     COVAGE  189
75 ANSOUT=VOUT                                                  COVAGE  190
80 CONTRIB=ANSOUT*PDIRAR(I)*FAILFAC                             FINAL    38
   IF(COVINT)  WRITE(6,13)I,CONTRIB                             COVAGE  192
13 FORMAT(15X,I5,F15.5)                                         COVAGE  193
   COVAGE=COVAGE+CONTRIB                                        COVAGE  194
100 CONTINUE                                                    COVAGE  195
101 GPFLG=.T.                                                   COVAGE  196
   RETURN                                                       COVAGE  198
   END                                                          COVAGE  199
   FUNCTION CVG1(I,T)                                           CVG       1
C                                                               FINAL     1
C        COMPUTE THE COEFFICIENT OF THE G FUNCTION FOR          FINAL     2
C        UNSCHEDULED IMPULSE DETECTOR I, TIME T.                FINAL     3
C        (THE COMPETITVE RATE OF DETECTION)                     FINAL     4
C                                                               FINAL     5
   DIMENSION LL(20),PP(20)                                      CVG       2
   COMMON /CVB3/NDSAV(20) /CVB7/ICHAR(20),PDETAR(20),XYZ(40),   CVG       3
  1      TDELAR(20) /CVB2/FDET(7,1)                             CVG       4
   ANS=1.0                                                      CVG       5
   NS=0                                                         CVG       6
   DO 10  NU=1,20                                               CVG       7
   IF(NU.EQ.I)  GO TO 10                                        CVG       8
   IF(ICHAR(NU).NE.1)  GO TO 10                                 CVG       9
   IF(TDELAR(NU).NE.T)  GO TO 10                                CVG      10
   NS=NS+1                                                      CVG      11
   PP(NS)=PDETAR(NU)                                            CVG      12
10 CONTINUE                                                     CVG      13
   IF(NS.EQ.0)  GO TO 40                                        CVG      14
   ONE=1.0                                                      CVG      15
   DO 30  NU=1,NS                                               CVG      16
   ONE=-ONE                                                     CVG      17
   SUM2=0.0                                                     CVG      18
   DO 21  M=1,NU                                                CVG      19
21 LL(M)=M                                                      CVG      20
   LL(NU+1)=NS+1                                                CVG      21
22 TERM=1.0                                                     CVG      22
   DO 23  LP=1,NU                                               CVG      23
   LLLP=LL(LP)                                                  CVG      24
23 TERM=TERM*PP(LLLP)                                           CVG      25
   SUM2=SUM2+TERM                                               CVG      26
   LP=NU                                                        CVG      27
24 LT=LL(LP)+1                                                  CVG      28
   IF(LT.GE.LL(LP+1))  GO TO 26                                 CVG      29
   LL(LP)=LT                                                    CVG      30
   IF(LP.EQ.NU)  GO TO 22                                       CVG      31
   LP1=LP+1                                                     CVG      32
   DO 25  M=LP1,NU                                              CVG      33
25 LL(M)=LL(M-1)+1                                              CVG      34
   GO TO 22                                                     CVG      35
26 IF(LP.LE.1)  GO TO 30                                        CVG      36
   LP=LP-1                                                      CVG      37
   GO TO 24                                                     CVG      38
30 ANS=ANS+ONE*SUM2/(NU+1)                                      CVG      39
40 PROD=1.0                                                     CVG      40
   SUM=0.0                                                      CVG      41
   DO 50  K=1,20                                                CVG      42
   IF(K.EQ.I)  GO TO 50                                         CVG      43
   ICH=ICHAR(K)                                                 CVG      44
   IF(ICH.EQ.0)  GO TO 50                                       CVG      45
   IF(ICH.GE.3)  GO TO 47                                       CVG      46
   IF(ICH.GE.2)  GO TO 43                                       CVG      47
   IF(TDELAR(K).EQ.T)  GO TO 50                                 CVG      48
43 PROD=PROD*(1.0-PDETAR(K)*FINTEG(FDET,NDSAV(K),T-TDELAR(K)))  CVG      49
```

```
      GO TO 50                                                          CVG       50
   47 SUM=SUM+PDETAR(K)*CVGPI(K,T)                                      CVG       51
   50 CONTINUE                                                          CVG       52
      CVG1=ANS*PROD*(1.0-SUM)                                           CVG       53
      RETURN                                                            CVG       54
      END                                                               CVG       55
      FUNCTION CVG2(I,T)                                                CVG       56
C                                                                       FINAL      6
C          COMPUTE THE VALUE OF THE G FUNCTION FOR                      FINAL      7
C          UNSCHEDULED FINITE DETECTOR I, TIME T.                       FINAL      8
C          (THE COMPETITIVE RATE OF DETECTION)                          FINAL      9
C                                                                       FINAL     10
      COMMON /CVB3/NDSAV(20) /CVB7/ICHAR(20),PDETAR(20),XYZ(40),        CVG       57
     1     TDELAR(20) /CVB2/FDET(7,1)                                   CVG       58
      PROD=1.0                                                          CVG       59
      SUM=0.0                                                           CVG       60
      DO 10  K=1,20                                                     CVG       61
      IF(K.EQ.I)  GO TO 10                                              CVG       62
      ICH=ICHAR(K)                                                      CVG       63
      IF(ICH.GE.3)  GO TO 5                                             CVG       64
      IF(ICH.EQ.0)  GO TO 10                                            CVG       65
      PROD=PROD*(1.0-PDETAR(K)*FINTEG(FDET,NDSAV(K),T-TDELAR(K)))       CVG       66
      GO TO 10                                                          CVG       67
    5 SUM=SUM+PDETAR(K)*CVGPI(K,T)                                      CVG       68
   10 CONTINUE                                                          CVG       69
      CVG2=PROD*(1.0-SUM)*FEVAL(FDET,NDSAV(I),T-TDELAR(I))              CVG       70
      RETURN                                                            CVG       71
      END                                                               CVG       72
      FUNCTION CVGS(I,T)                                                CVG       73
C                                                                       FINAL     11
C          COMPUTE THE VALUE OF THE G FUNCTION FOR                      FINAL     12
C          SCHEDULED DETECTOR I, TIME T.                                FINAL     13
C          (THE COMPETITIVE RATE OF DETECTION)                          FINAL     14
C                                                                       FINAL     15
      COMMON /CVB3/NDSAV(20) /CVB7/ICHAR(20),PDETAR(20),XYZ(20),        CVG       74
     1     PERIAR(20),TDELAR(20) /CVB2/FDET(7,1)                        CVG       75
      CVGS=0.0                                                          CVG       76
      IF(T.LT.0.0.OR.T.GT.PERIAR(I))  GO TO 20                          CVG       77
      PROD=1.0                                                          CVG       78
      DO 10  K=1,20                                                     CVG       79
      ICH=ICHAR(K)                                                      CVG       80
      IF(ICH.GT.2.OR.ICH.EQ.0)  GO TO 10                                CVG       81
      PROD=PROD*(1.0-PDETAR(K)*FINTEG(FDET,NDSAV(K),T-TDELAR(K)))       CVG       82
   10 CONTINUE                                                          CVG       83
      CVGS=PROD*CVGPI(I,T)                                              CVG       84
   20 RETURN                                                            CVG       85
      END                                                               CVG       86
      SUBROUTINE CVGP3(I)                                               CVG       87
C                                                                       FINAL     16
C          COMPUTE AND SAVE N+1 SAMPLES OF THE G-PRIME FUNCTION         FINAL     17
C          FOR SCHEDULED IMPULSE DETECTOR I, OVER ITS PERIOD.           FINAL     18
C          (RATE OF DETECTION IN COMPETITION WITH OTHER                 FINAL     19
C          SCHEDULED DETECTORS)                                         FINAL     20
C                                                                       FINAL     21
      COMMON /CVB3/NDSAV(20) /CVB6/LCM,TMINOH,FACTOR /CVB7/ICHAR(20),   CVG       88
     1     PDETAR(20),IREPAR(20),PERIAR(20),TDELAR(20) /CVB8/GPFLG,     CVG       89
     2     GPAR(20,101) /CVB9/N /CVB2/FDET(7,1)                         CVG       90
     3     /CVB10/TJLFLG                                                CVG       91
      LOGICAL TJLFLG                                                    CVG       92
      LIM=LCM/IREPAR(I)                                                 CVG       93
      DT=PERIAR(I)/N                                                    CVG       94
      NP1=N+1                                                           POSTPCH    1
      DO 30  ITP1=1,NP1                                                 POSTPCH    2
      IT=ITP1-1                                                         POSTPCH    3
      TAU=DT*IT                                                         CVG       96
      SUM=0.0                                                           CVG       97
```

```
      DO 20  L=1,LIM                                                    CVG       98
      PROD=1.0                                                          CVG       99
      DO 10  J=1,20                                                     CVG      100
      IF(J.EQ.I)  GO TO 10                                              CVG      101
      IF(ICHAR(J).LT.3)  GO TO 10                                       CVG      102
      IF(IT.EQ.1)  TJLFLG=.F.                                           CVG      103
      TJL=CVTJL(J,L,I)                                                  CVG      104
      PROD=PROD*(1.0-PDETAR(J)*(1.0-FINTEG(FDET,NDSAV(J),PERIAR(I)+     CVG      105
     1     TDELAR(I)-TAU-TJL)))                                         CVG      106
   10 CONTINUE                                                          CVG      107
   20 SUM=SUM+PROD                                                      CVG      108
   30 GPAR(I,IT+1)=FACTOR*SUM                                           CVG      109
      RETURN                                                            CVG      110
      END                                                               CVG      111
      SUBROUTINE CVGP4(I)                                               CVG      112
C                                                                       FINAL     22
C          COMPUTE AND SAVE N+1 SAMPLES OF THE G-PRIME FUNCTION         FINAL     23
C          FOR SCHEDULED FINITE DETECTOR I, OVER ITS PERIOD,            FINAL     24
C          (RATE OF DETECTION IN COMPETITION WITH OTHER                 FINAL     25
C          SCHEDULED DETECTORS)                                         FINAL     26
C                                                                       FINAL     27
      COMMON /CVB3/NDSAV(20) /CVB6/LCM,TMINOR,FACTOR /CVB7/ICHAR(20),   CVG      113
     1     PDETAR(20),IREPAR(20),PERIAR(20),TDELAR(20) /CVB8/GPFLG,     CVG      114
     2     GPAR(20,101) /CVB9/N /CVB2/FDET(7,1)                         CVG      115
     3     /CVB10/TJLFLG                                                CVG      116
      LOGICAL FLOP,TJLFLG,TFLG                                          CVG      117
      NP1=N+1                                                           POSTPCH    4
      NUM=NDSAV(I)                                                      CVG      118
      TDUR=FDET(7,NUM)                                                  CVG      119
      TFLG=.F.                                                          CVG      120
      LIM=LCM/IREPAR(I)                                                 CVG      121
      DT=PERIAR(I)/N                                                    CVG      122
      DO 60  ITP1=1,NP1                                                 POSTPCH    5
      IT=ITP1-1                                                         POSTPCH    6
      SUM=0.0                                                           CVG      124
      DE=0.0                                                            FINAL     28
      TAU=DT*IT                                                         CVG      125
      IF(IT.EQ.0)  GO TO 50                                             CVG      126
      UPLIM=TAU                                                         CVG      127
      TFLG=TAU.GT.TDUR                                                  CVG      128
      IF(TFLG)  UPLIM=TDUR                                              CVG      129
      DE=UPLIM/N                                                        CVG      130
      DO 40  L=1,LIM                                                    CVG      131
      FLOP=.F.                                                          CVG      132
      EUM=0.0                                                           CVG      133
      DO 30  IEP1=1,NP1                                                 POSTPCH    7
      IE=IEP1-1                                                         POSTPCH    8
      ETA=DE*IE                                                         CVG      135
      PROD=1.0                                                          CVG      136
      DO 10  J=1,20                                                     CVG      137
      IF(ICHAR(J).LT.3)  GO TO 10                                       CVG      138
      IF(J.EQ.I)  GO TO 10                                              CVG      139
      IF(IT+IE.EQ.1)  TJLFLG=.F.                                        CVG      140
      TJL=CVTJL(J,L,I)                                                  CVG      141
      PROD=PROD*(1.0-PDETAR(J)*(1.0-FINTEG(FDET,NDSAV(J),PERIAR(I)+     CVG      142
     1     TDELAR(I)+ETA-TAU-TJL)))                                     CVG      143
   10 CONTINUE                                                          CVG      144
      PROD=PROD*FEVAL(FDET,NDSAV(I),ETA)                                CVG      145
      IF(IE.EQ.0.OR.IE.EQ.N)  GO TO 20                                  CVG      146
      IF(FLOP)  GO TO 15                                                CVG      147
      EUM=EUM+2.0*PROD                                                  CVG      148
      GO TO 30                                                          CVG      149
   15 EUM=EUM+4.0*PROD                                                  CVG      150
      GO TO 30                                                          CVG      151
   20 EUM=EUM+PROD                                                      CVG      152
   30 FLOP=.NOT.FLOP                                                    CVG      153
```

```
   40 SUM=SUM+EUM                                                   CVG      154
   50 VAL=FACTOR*SUM*DE/3.0                                         CVG      155
      IF(TFLG)  GO TO 60                                            CVG      156
      VAL=VAL+(1.0-FINTEG(FDET,NDSAV(I),TAU))/PERIAR(I)             CVG      157
   60 GPAR(I,IT+1)=VAL                                              CVG      158
      RETURN                                                        CVG      159
      END                                                           CVG      160
      FUNCTION CVTJL(J,L,I)                                         CVG      161
C                                                                   FINAL     29
C        RETURN THE VALUE OF TJL FOR USE BY CVGP3 AND CVGP4         FINAL     30
C                                                                   FINAL     31
      COMMON /CVB7/XYZ(60),PERIAR(20),TDELAR(20)                    CVG      162
     1      /CVB10/TJLFLG                                           CVG      163
      LOGICAL TJLFLG                                                CVG      164
      XNU=(TDELAR(I)+L*PERIAR(I)-TDELAR(J))/PERIAR(J)               CVG      165
      NU=XNU                                                        CVG      166
      YNU=NU                                                        CVG      167
      IF(XNU.EQ.YNU)  GO TO 99                                      CVG      168
      CVTJL=TDELAR(J)+YNU*PERIAR(J)-(L-1)*PERIAR(I)                 CVG      169
      IF(CVTJL.LT.TDELAR(I))  CVTJL=-PERIAR(J)                      CVG      170
   10 RETURN                                                        CVG      171
   99 IF(TJLFLG)  GO TO 10                                          CVG      172
      WRITE(6,11)J,I                                                CVG      173
   11 FORMAT(5X,*ERROR IN CVTJL - 2 DETECTORS (*,I2,1H,,I3,         CVG      174
     1      *) SCHEDULED SIMULTANEOUSLY*)                           CVG      175
      TJLFLG=.T.                                                    CVG      176
      GO TO 10                                                      CVG      177
      END                                                           CVG      178
      FUNCTION CVGPI(I,T)                                           CVG      179
C                                                                   FINAL     32
C          INTERPOLATE THE SAMPLE G-PRIME OR INTEGRAL G-PRIME       FINAL     33
C          VALUES SAVED IN ARRAY GPAR                               FINAL     34
C                                                                   FINAL     35
      COMMON /CVB7/XYZ(60),PERIAR(20)                               CVG      180
     1      /CVB8/GPFLG,GPAR(20,101) /CVB9/N                        CVG      181
      IF(T.LT.0.0)  GO TO 101                                       CVG      182
      PER=PERIAR(I)                                                 CVG      183
      IF(T.LT.PER)  GO TO 1                                         CVG      184
      CVGPI=0.0                                                     CVG      185
      IF(GPFLG.OR.T.EQ.PER)  CVGPI=GPAR(I,N+1)                      CVG      186
      GO TO 100                                                     CVG      187
    1 DT=PER/N                                                      CVG      188
      ID=T/DT                                                       CVG      189
      IT=ID+1                                                       CVG      190
      X=T-DT*ID                                                     CVG      191
      A=GPAR(I,IT)                                                  CVG      192
      B=GPAR(I,IT+1)                                                CVG      193
      CVGPI=A+X*(B-A)/DT                                            CVG      194
  100 RETURN                                                        CVG      195
  101 WRITE(6,11)                                                   CVG      196
   11 FORMAT(5X,*ERROR IN CVGPI - NEGATIVE TIME ARGUEMENT*)         CVG      197
      STOP                                                          CVG      198
      END                                                           CVG      199
      SUBROUTINE SPECIT(FLIST,NUM,IGFT,ISCH,IREP,INTF,COEF,TDEL,    SPECIT     1
C                                                                   FINAL      1
C          SAVE THE SPECIFICATIONS FOR ONE DETECTION, ISOLATION     FINAL      2
C          OR RECOVERY (1 OF 2) FUNTION IN THE DATA BASE            FINAL      3
C                                                                   FINAL      4
     1      P1,P2,P3,TDUR)                                          SPECIT     2
      DIMENSION FLIST(7,1)                                          SPECIT     3
      CALL IPUT(CODES,1,IGFT)                                       SPECIT     4
      CALL IPUT(CODES,2,ISCH)                                       SPECIT     5
      CALL IPUT(CODES,3,IREP)                                       SPECIT     6
      CALL IPUT(CODES,4,1)                                          SPECIT     7
      CALL IPUT(CODES,5,INTF)                                       SPECIT     8
      FLIST(1,NUM)=CODES                                            SPECIT     9
```

```
      FLIST(2,NUM)=COEF                                                 SPECIT  10
      FLIST(3,NUM)=TDEL                                                 SPECIT  11
      FLIST(4,NUM)=P1                                                   SPECIT  12
      FLIST(5,NUM)=P2                                                   SPECIT  13
      FLIST(6,NUM)=P3                                                   SPECIT  14
      FLIST(7,NUM)=TDUR                                                 SPECIT  15
      RETURN                                                            SPECIT  16
      END                                                               SPECIT  17
      FUNCTION FEVAL(FLIST,NUM,T)                                       FEVAL    1
C                                                                       FINAL    1
C          EVALUATE DETECTION, ISOLATION, OR RECOVERY (1 OF 2)          FINAL    2
C          FUNCTION 'NUM AT TIME T.  THE FUNCTION EVALUATES             FINAL    3
C          TO 0.0 OUTSIDE THE RANGE 0.0, TDUR.                          FINAL    4
C                                                                       FINAL    5
      DIMENSION FLIST(7,1)                                              FEVAL    2
      COMMON/CVB4/P1,P2,P3,TDUR,IGFT                                    FEVAL    3
      CODES=FLIST(1,NUM)                                                FEVAL    4
      IGFT=IGET(CODES,1)                                                FEVAL    5
      TDUR=FLIST(7,NUM)                                                 FEVAL    6
      IF(IGFT.GT.0.AND.T.GE.0.0.AND.T.LE.TDUR)  GO TO 10                FEVAL    7
      ANS=0.0                                                           FEVAL    8
      GO TO 100                                                         FEVAL    9
   10 P1=FLIST(4,NUM)                                                   FEVAL   10
      P2=FLIST(5,NUM)                                                   FEVAL   11
      P3=FLIST(6,NUM)                                                   FEVAL   12
      ENTRY EVAL                                                        FEVAL   13
      GO TO (21,22,23,24,25,26),IGFT                                    FEVAL   14
   21 ANS=FN1(T)                                                        FEVAL   15
      GO TO 100                                                         FEVAL   16
   22 ANS=FN2(T)                                                        FEVAL   17
      GO TO 100                                                         FEVAL   18
   23 ANS=FN3(T)                                                        FEVAL   19
      GO TO 100                                                         FEVAL   20
   24 ANS=FN4(T)                                                        FEVAL   21
      GO TO 100                                                         FEVAL   22
   25 ANS=FN5(T)                                                        FEVAL   23
      GO TO 100                                                         FEVAL   24
   26 ANS=FN6(T)                                                        FEVAL   25
  100 FEVAL=ANS                                                         FEVAL   26
      RETURN                                                            FEVAL   27
      END                                                               FEVAL   28
      FUNCTION FINTEG(FLIST,NUM,T)                                      FINTEG   1
C                                                                       FINAL    1
C          EVALUATE THE INTEGRAL OF DETECTION, ISOLATION, OR            FINAL    2
C          RECOVERY (1 OF 2) FUNCTION 'NUM, AT TIME T.                  FINAL    3
C          IF THE INTEGRAL MODEL IS NOT DEFINED, AN N+1                 FINAL    4
C          POINT SIMPSON INTEGRAL IS PERFORMED USING FEVAL.             FINAL    5
C                                                                       FINAL    6
      DIMENSION FLIST(7,1)                                              FINTEG   2
      LOGICAL IFLOP                                                     FINTEG   3
      COMMON /CVB4/P1,P2,P3,TDUR,IGFT                                   FINTEG   4
      DATA N/20/                                                        FINTEG   5
      NP1=N+1                                                           POSTPCH  1
      TDUR=FLIST(7,NUM)                                                 FINTEG   6
      IF(T.GE.0.0)  GO TO 5                                             FINTEG   7
      ANS=0.0                                                           FINTEG   8
      GO TO 100                                                         FINTEG   9
    5 IF(T.LE.TDUR)  GO TO 10                                           FINTEG  10
    6 ANS=1.0                                                           FINTEG  11
      GO TO 100                                                         FINTEG  12
   10 CODES=FLIST(1,NUM)                                                FINTEG  13
      IGFT=IGET(CODES,1)                                                FINTEG  14
      IF(IGFT.EQ.0)  GO TO 6                                            FINTEG  15
      P1=FLIST(4,NUM)                                                   FINTEG  16
      P2=FLIST(5,NUM)                                                   FINTEG  17
      P3=FLIST(6,NUM)                                                   FINTEG  18
```

```
      INTF=IGET(CODES,5)                                                 FINTEG   19
      IF(INTF.GT.0)  GO TO 50                                            FINTEG   20
C                                                                        FINTEG   21
C          START OF SIMPSON INTEGRAL                                     FINTEG   22
C                                                                        FINTEG   23
      DEL=T/N                                                            FINTEG   24
      IFLOP=.F.                                                          FINTEG   25
      SUM=0.0                                                            FINTEG   26
      DO 30  IP1=1,NP1                                                   POSTPCH   2
      I=IP1-1                                                            POSTPCH   3
      TAU=DEL*I                                                          FINTEG   28
      VAL=EVAL(0,0,TAU)                                                  FINTEG   29
      IF(I.EQ.0.OR.I.EQ.N)  GO TO 20                                     FINTEG   30
      IF(IFLOP)  GO TO 15                                                FINTEG   31
      SUM=SUM+2.0*VAL                                                    FINTEG   32
      GO TO 30                                                           FINTEG   33
   15 SUM=SUM+4.0*VAL                                                    FINTEG   34
      GO TO 30                                                           FINTEG   35
   20 SUM=SUM+VAL                                                        FINTEG   36
   30 IFLOP=.NOT.IFLOP                                                   FINTEG   37
      ANS=DEL*SUM/3.0                                                    FINTEG   38
      GO TO 100                                                          FINTEG   39
   50 GO TO (51,52,53,54,55,56),IGFT                                     FINTEG   40
   51 ANS=FN1I(T)                                                        FINTEG   41
      GO TO 100                                                          FINTEG   42
   52 ANS=FN2I(T)                                                        FINTEG   43
      GO TO 100                                                          FINTEG   44
   53 ANS=FN3I(T)                                                        FINTEG   45
      GO TO 100                                                          FINTEG   46
   54 ANS=FN4I(T)                                                        FINTEG   47
      GO TO 100                                                          FINTEG   48
   55 ANS=FN5I(T)                                                        FINTEG   49
      GO TO 100                                                          FINTEG   50
   56 ANS=FN6I(T)                                                        FINTEG   51
  100 FINTEG=ANS                                                         FINTEG   52
      RETURN                                                             FINTEG   53
      END                                                                FINTEG   54
      FUNCTION IGET(IWORD,INDEX)                                         GETPUT    1
C                                                                        FINAL     1
C          RETURN THE VALUE OF THE INDEX-TH 12 BIT FIELD                 FINAL     2
C          OF IWORD, AS A RIGHT JUSTIFIED INTEGER                        FINAL     3
C                                                                        FINAL     4
      DIMENSION MASK(5)                                                  GETPUT    2
      DATA MASK/77770000000000000000B,00007777000000000000B,             GETPUT    3
     100000000777700000000B,00000000000077770000B,00000000000000007777B/GETPUT    4
      ITP=AND(IWORD,MASK(INDEX))                                         GETPUT    5
      IGET=ISHIFT(ITP,12*INDEX)                                          GETPUT    6
      RETURN                                                             GETPUT    7
      END                                                                GETPUT    8
      SUBROUTINE IPUT(IWORD,INDEX,ICODE)                                 GETPUT    9
C                                                                        FINAL     5
C          PACK THE RIGHTMOST 12 BITS OF ICODE INTO THE                  FINAL     6
C          INDEX-TH 12 BIT FIELD OF IWORD                                FINAL     7
C                                                                        FINAL     8
      DIMENSION MASK(5)                                                  GETPUT   10
      DATA MASK/00007777777777777777B,77770000777777777777B,             GETPUT   11
     177777777000077777777B,77777777777700007777B,77777777777777770000B/GETPUT   12
      ITP=ISHIFT(ICODE,12*(5-INDEX))                                     GETPUT   13
      IWORD=AND(IWORD,MASK(INDEX))                                       GETPUT   14
      IWORD=OR(IWORD,ITP)                                                GETPUT   15
      RETURN                                                             GETPUT   16
      END                                                                GETPUT   17
          IDENT ISHIFT                                                   POSTPCH   1
          ENTRY ISHIFT                                                   POSTPCH   2
ISHIFT    PS                                                             POSTPCH   3
          SA1  B1                                                        POSTPCH   4
```

```
      SA2   B2                                                POSTPCH     5
      SB2   X2                                                POSTPCH     6
      LX6   X1,B2                                             POSTPCH     7
      EQ    ISHIFT                                            POSTPCH     8
      END                                                     POSTPCH     9
FUNCTION FN1(T)                                               FN1         1
COMMON /CVB4/A                                                FN1         2
FN1=A                                                         FN1         3
RETURN                                                        FN1         4
END                                                           FN1         5
FUNCTION FN1I(T)                                              FN1         6
COMMON /CVB4/A                                                FN1         7
FN1I=A*T                                                      FN1         8
RETURN                                                        FN1         9
END                                                           FN1        10
FUNCTION FN2(T)                                               FN2         1
COMMON /CVB4/A,B,C                                            FN2         2
FN2=0.0                                                       FN2         3
K=T/C                                                         FN2         4
D=T-K*C                                                       FN2         5
IF(D.GT.B)  RETURN                                            FN2         6
FN2=A                                                         FN2         7
RETURN                                                        FN2         8
END                                                           FN2         9
FUNCTION FN2I(T)                                              FN2        10
COMMON /CVB4/A,B,C                                            FN2        11
K=T/C                                                         FN2        12
D=AMIN1(B,T-K*C)                                              FN2        13
FN2I=A*(K*B+D)                                                FN2        14
RETURN                                                        FN2        15
END                                                           FN2        16
FUNCTION FN3(T)                                               FN3         1
COMMON /CVB4/A,B,C                                            FN3         2
FN3=A*(EXP(-B*T)+C)                                           POSTPCH     1
RETURN                                                        FN3         4
END                                                           FN3         5
FUNCTION FN3I(T)                                              FN3         6
COMMON /CVB4/A,B,C                                            FN3         7
FN3I=A*((1.0-EXP(-B*T))/B+C)                                  FINAL       1
RETURN                                                        FN3         9
END                                                           FN3        10
FUNCTION FN4(T)                                               FN4         1
COMMON /CVB4/A,B,C                                            FN4         2
RETURN                                                        FN4         3
END                                                           FN4         4
FUNCTION FN4I(T)                                              FN4         5
COMMON /CVB4/A,B,C                                            FN4         6
RETURN                                                        FN4         7
END                                                           FN4         8
FUNCTION FN5(T)                                               FN5         1
COMMON /CVB4/ A,B,C                                           FN5         2
RETURN                                                        FN5         3
END                                                           FN5         4
FUNCTION FN5I(T)                                              FN5         5
COMMON /CVB4/ A,B,C                                           FN5         6
RETURN                                                        FN5         7
END                                                           FN5         8
FUNCTION FN6(T)                                               FN6         1
COMMON /CVB4/ A,B,C                                           FN6         2
RETURN                                                        FN6         3
END                                                           FN6         4
FUNCTION FN6I(T)                                              FN6         5
COMMON /CVB4/A,B,C                                            FN6
RETURN                                                        FN6
END                                                           FN6         8
```