Fall 1979

Simulation Model to Evaluate Performance of Operational Systems and their Impact on Repair Shop Activity at a Navy Field Site

James T. Newell
University of Central Florida, jjnewell@nehp.net

SIMULATION MODEL TO EVALUATE PERFORMANCE OF OPERATIONAL SYSTEMS AND THEIR IMPACT ON REPAIR SHOP ACTIVITY AT A NAVY FIELD SITE

BY

JAMES T. NEWELL
B.S.E., University of Central Florida, 1975

RESEARCH REPORT

Submitted in partial fulfillment of the requirements
for the degree of Master of Science:
Industrial Engineering
in the Graduate Studies Program of the
College of Engineering
at the University of Central Florida; Orlando, Florida

Fall Quarter
1979

ABSTRACT

This paper presents the background and procedures leading to development of a simulation model to analyze the impact of certain decision variables on operational system performance and workloads at the repair facility of a typical Navy field site.

The research examines the impact of maintenance support concepts, as implemented by changes in the decision variables, associated with the broader application of Automatic Test Equipment. The initial effort consisted of data collection and field site surveys which culminated in defining a work flow model illustrating typical repair facility operations.

The work flow model is translated into a computer simulation model. The baseline model contains all the values for failure rates, delay times, and probability decision parameters derived from the available data.

The simulation model is then exercised and the output data recorded for comparison with historical data to validate the model and provide a baseline for comparison as the decision parameters are varied. Of the variables exercised, it appears that the Built-in-Test (BIT), or Self-test capability, is one of the more important design considerations in the original operating systems.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

Background

Whenever new theories, ideas, and approaches are considered for solving existing problem conditions, a background study is usually performed in order to establish a baseline from which to make cost, performance, or other data comparisons. In this way, decisions can be made and refined for either implementing the new approach, modifying it, or discarding it altogether. The maintenance and support of Navy training devices, as well as other sophisticated electronic material, is currently in a critical decision realm as to what kind of test and support equipment is required. In the past, the test equipment has consisted of standard meters, oscilloscopes, and the like, plus some specially designed "automatic" testers for the specific end item being supported. The most common approach for training devices, which are most often one-of-a-kind, has been to treat them as a "self-contained" system with its own special assignment of maintenance and test equipment. Industrial contractors would usually design their own specific "automatic" tester for the end item training device with the government buying the total package deal.

Over a period of time, more training devices (hereafter called operating systems) would begin accumulating at various military installations (training sites). At the same time, the mission requirements of these systems became more comprehensive requiring much more complex equipment. Simple test equipment could no longer satisfy the support requirements, and on the other hand proliferation of "automatic" testers has become increasingly costly. These pressures have focused considerable attention toward relocating common test equipment to special designated repair sites and development of more universal Automatic Test Equipment (ATE) to support a broader range of operating systems. Part of the necessary study would be to measure workloads in a common repair shop supporting several operating systems and the impact on the operating system availability as a result of various decision parameter variations.

It was these concerns that led to a study by the Naval Training Equipment Center (NTEC) in 1977. Part of this study was concerned with the impact of programming the ATE to handle the multitude of various electronic assemblies in the training device inventory. The simulation model developed in this study reproduced existing workload conditions at a repair site and simulated the impact created by the ATE programming requirement.[1] This model was based on historical workload data submitted via the Navy 3-M[2] data collection system. These data were verified by field trips, interviews and observations. In general, there was an excellent correlation of the data which differed by less than ten percent in the quantity of maintenance actions and repairs over a one year period between the projections based on interviews and observations and those culled from the data collection system. This provided a valid foundation from which to develop the simulation model.

In general, the simulation model developed in this study would enable an analyst to observe the flow of repair actions through the repair facility resulting from the various decision parameters pertinent to the present maintenance support concept. From this baseline, observations could be made concerning changes in the basic approach as well as implementing the ATE programming requirement with this subsequent impact on the repair shop

---

[1]George W. Campbell, Intermediate Maintenance Concepts and Use of ATE for Training Devices (Preliminary Study) (Orlando, Fla.: Naval Training Equipment Center, 1977), pp. 10-11, App. A.

[2]3-M is an acronym derived from the Navy data collection system entitled Naval Aviation Maintenance and Material Management System introduced on 1 January 1965. This system is a part of The Naval Aviation Maintenance Program (NAMP), which was originated on 26 May 1959. Although there have been substantial revisions to the program, the term "3-M" is still popularly used when referring to the maintenance data collection portion of the current NAMP introduced on 18 June 1973 by OPNAV Instruction 4790-2A.

workload. The principal objective of this first model was to assess the ATE programming impact. Therefore, sample programming times were collected from observations and from manufacturers of ATE, then used in the model as additional repair cycle delays. The results showed that it was not possible to levy this additional requirement for programming on the repair shop personnel. As a consequence, it was determined that a more comprehensive study concerning the application of ATE was needed. It was during this second study that the more comprehensive simulation model presented herein was developed.

Objectives of Research Project

The principal objective of this research is to develop a simulation model which can be used to evaluate the impact on operating system performance and repair facility workloads due to changes in maintenance support concepts. These support concepts are implemented by several decision variables that could be altered by the introduction of ATE into a typical Navy repair facility supporting trainer systems. To realize this objective required the analysis of the present real-world system, developing the work flow diagram representing this system, developing the simulation model of this system, and then exercising the model with changes in various decision parameters in order to make judgments concerning reactions of the system to these parameter variations.

Specific objectives of the simulation model concerned the ability to observe variations in the performance of the systems and changes in workloads resulting from certain decision parameter changes. These major decision parameters were: (1) Built-in-Test (BIT) or self-test accuracy, (2) probability of having spare parts, (3) performance test accuracy, and (4) diagnostic test accuracy.

The most significant performance measure of an operating system is its operational availability, or percentage of up-status time. Parameters (1), (3) and (4) all concern the ability to correctly detect a failed item and properly restore a downed operating system to an "up" condition, while parameter (2) provides a means of rapidly restoring the system. These parameters will be varied and the resulting changes in operating system availability and workloads observed for the systems and repair shop points.

The following sections describe the development of the simulation model from the beginning of establishing the work flow structure, through the application of programming techniques to solve the required system operations, to the exercising of the model to obtain statistical information related to the operating system performance and the repair shop workloads.

General Description of the Model

After surveying several field sites, a general work flow description of the operating systems and repair activities was developed. All maintenance and repair activities can be generalized into three encompassing categories which are depicted in Figure 1. These activities consist of: (1) The operating systems at each field site, (2) a local repair activity at each field site, and (3) a depot or other remote site from the field site.

Maintenance and support of the operating system is the primary objective of all repair activity. Troubleshooting, maintenance and repair of the operating system consists of using the system's built-in tests and self-diagnostic routines as well as other on-line testing and investigation. This testing is used in an attempt to isolate the fault in the system. Failed or suspect material is then removed from the system and either replaced by a spare, or repaired and then replaced.

The second phase of this maintenance and support consists of those activities normally confined to the established site repair shop. Failed or suspect material (hereafter called "failed material") is funneled into the repair shop from all the operating systems at the site. Decisions must be made as to whether the repair of the

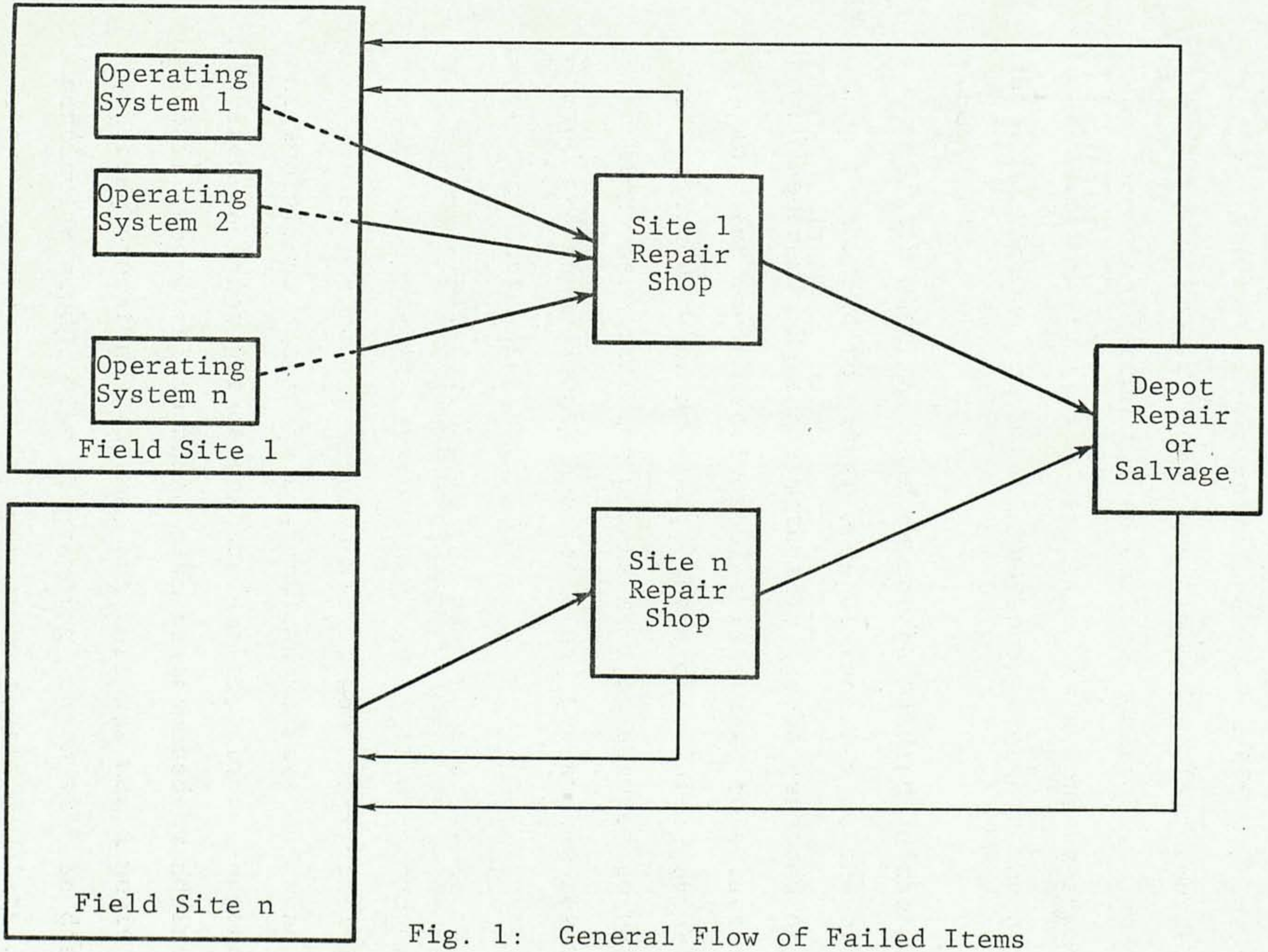

Fig. 1: General Flow of Failed Items

material is within the capability and training of the shop personnel or whether equipment exists at the repair shop to perform the required repair functions. After processing through the repair shop, "good" material is forwarded to the operating system for an on-line verification test while material that is still "defective" is shipped elsewhere. These decisions are made at various points in the normal work flow.

Failed material that cannot be handled in the repair shop is shipped to either a depot facility or the original manufacturer. Here the materials may be repaired or salvaged. For the objectives of this study, this latter is not required to be modeled. Therefore, failed material reaching these decision points will be assumed to have immediate replacements available and the system returned to operational status.

The flow structure is the same regardless of the number of operating systems per site. In this manner, the impact on the repair shop workloads and the sensitivities of individual operating systems can be measured when system decision parameters are varied. Impact on the repair shop workload could also be measured by adding or deleting operating systems, but this was not done in this particular study. The programming, which will be discussed later, was structured in such a way that a complete operating system simulation section could be inserted or deleted without disturbing the repair shop simulation program routine.

In Figure 1, the block labeled Field Site 1 represents a collection of operating systems at a typical Navy site. This block can represent several operating systems and in this study consisted of six individual systems. Failures occurring in any one system could result in a flow of failed material into the block labeled Site 1 Repair Shop. Information needed to establish the baseline operating conditions of each system included the system mean-time-between-failures (MTBF), the average complexity of the failed material, mean-time-to-diagnose the fault on-line, probability of correct fault isolation, probability of available spare, and mean verification time after repair was made. Appendix E gives a facsimile of the field survey sheet for System 1.

From the survey sheet, data on line S1 establishes the interarrival time of failure occurences while line S2 forms the probability of low, medium, or high complexity material being the fault source. In like manner the other data lines form the bases for the program parameters described later. A similar survey sheet provided the data used in the Site 1 Repair Shop for mean delay times associated with performance testing, diagnostic testing, and the actual repair of the failed material. Repaired material is then returned to the operating system for on-line verification tests.

After establishing the baseline parameters and operating logic, which is described more fully in the next section, the model was exercised for four quarterly periods to form the baseline output data. Decision variables selected to alter were those described in the "Objectives" section. Four new quarterly periods were run for each single change in a decision variable. The major observation variable for each data period was the system operational availability. Other observations included the workloads and total expended delay (maintenance) times.

Although the objective of this research was to develop a practical simulation model which could be used to observe the impact of changes due to decision variables, the actual significance of those changes would be of interest in the overall conclusions. As an example, it can be seen that there are random deviations in the quarterly availabilities for any given system. Is the availability deviation due to decision variable change significantly different from the availability deviation due to the random failure pattern? Appendix F presents an Analysis of Variance (ANOVA)[3] test for the data produced for System 2 as a result of changes in self-test accuracy.

---

[3]Isaac N. Gibra, Probability and Statistical Inference for Scientists and Engineers (Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1973) pp. 337-48.

The ANOVA test shows that the hypothesis of equal means (mean availability) is rejected at the .05 level of significance and it is concluded that the mean availability due to the self-test accuracy is significantly different than the mean availability due to random failure for System 2. A similar analysis on System 1 does not reject the hypothesis of equal means. These tests can provide a basis for alternative conclusions for each specific system which are not addressed in any detail in this study. They also help establish a "feel" for the regression equations which were used in forming some basic conclusions concerning the model output data.

In addition, a multi-regression analysis was performed for System 2, as an example, to help verify which decision variable is more significant among those altered during the study. This analysis is presented in Appendix G.

In the comparative analysis of these type tests, it can be seen that the mean availability of the quarterly periods can be the most useful data element for preliminary conclusions. These mean values were then used in single regression applications of each decision variable on the observed mean availability since the treatment of each variable separately leads to the same conclusion as the analysis presented in Appendix G.

II. DEVELOPMENT OF THE MODEL

General Logic of the Model

The general logic of the model consists of two phases; (1) the on-line phase (operating system), and (2) the off-line phase (repair shop). During the on-line phase, the model creates malfunctions at the pre-determined rate computed from the survey data for each operating system. After the malfunction is created, it is assigned a complexity level based on the percentage values from the survey data (line S2 of Appendix E). The malfunction is delayed in transit through the model by the amount of time assigned to the BIT, or self-test procedure, and is then assigned a "true" failure status based upon the accuracy of the self-test. The model then assigns a spare based on the probability values from the surveys. If there were no spare, the failed material goes to the repair shop. If there is a spare, it is installed and a verification test performed. A GO decision at this point sends the failed material to the repair shop while a NO GO causes a recycle in search of another fault source.

In the off-line phase, a decision is first made to destroy the material or attempt repair. If a repair is to be attempted, the material is subjected to a performance verification test. Decisions and delays are implemented by data parameters from the site surveys. The material may pass or fail at this point. Material failing the test goes into the diagnostic test whil those passing go back for on-line tests.

Diagnostic testing decisions and delays are again computed from site survey data and the material either goes in for physical repair or recycled to the performance test. After repair, a second performance test is exercised for verification of the fix. The material then goes back to the on-line phase if it passes, or back to diagnostic tests if it fails.

The detail functional requirements of the model are described in the following sections.

Detailed Operating Requirements of the Model

The Operating System (On-line) Phase

The overall flow of a typical operating system is shown in Figure 2. The initiation of the sequence of work flow events pertaining to each operating system begins with the appearance of a malfunction and its related suspect failure. As previously noted, several operating

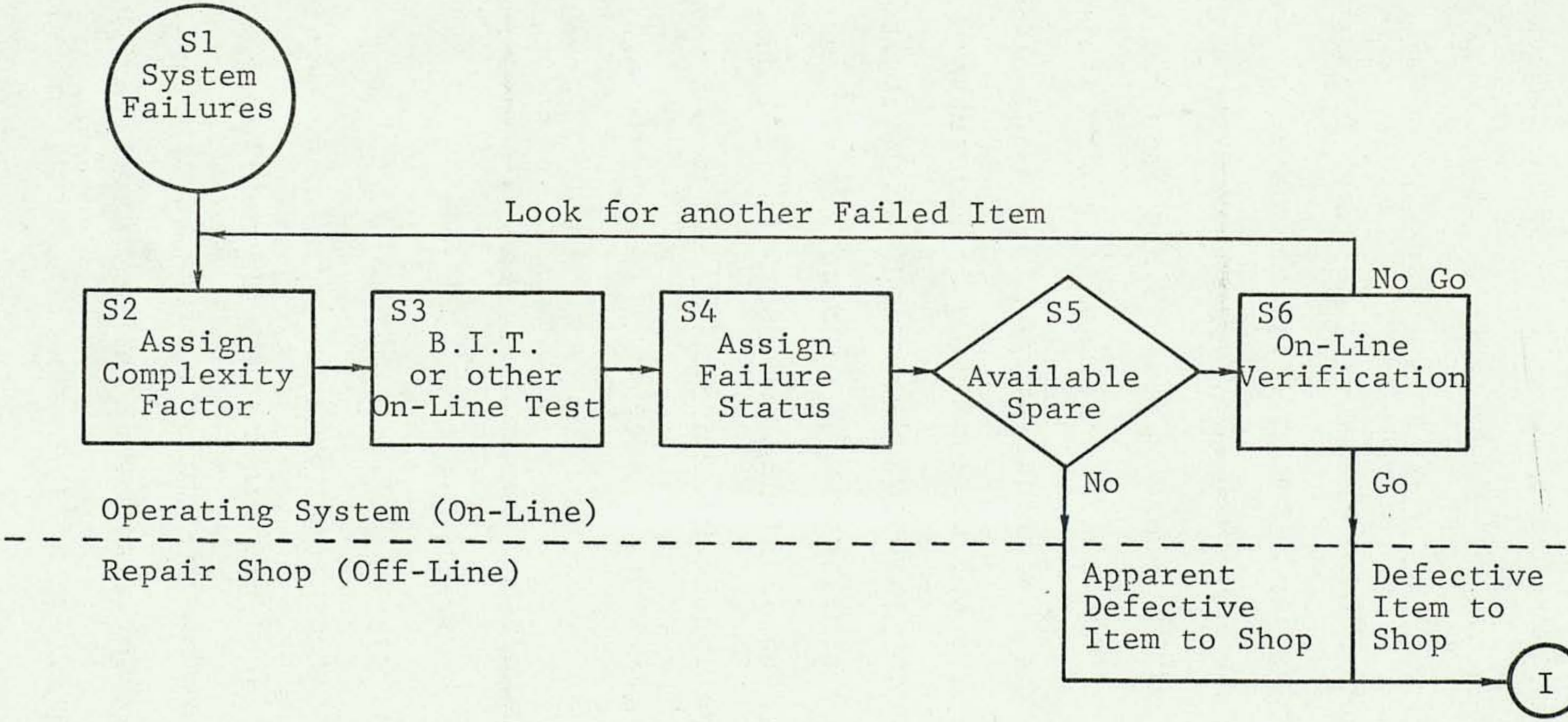

Fig. 2: Operating System Model (Typical)

systems may be on-line simultaneously and feeding failed material (Figure 1) into the repair shop. Block descriptions of the operating system as shown by Figure 2, and required decision parameters associated with these blocks are addressed in the following paragraphs:

Block S1: Block S1 is used to represent the origination of failures and/or malfunctions in the operating system. In normal operations, this tends to be a random process following well-known reliability distributions and becomes a characteristic parameter of the system components, design, and maintenance procedures among other things. The actual values used to represent this random process were taken from site surveys and later substantiated via the Navy 3-M reporting system. Malfunctions represented by Block S1 generally place the operating system in a DOWN status.

Block S2: Block S2 is used to assign a complexity level to each of the transactions generated by Block S1. The complexity of failed material is generally classified as low, medium, or high. In this manner, test and repair times can be differentiated within the model for various complexities such as a simple amplifier module or a high density digital logic module. The basis for assigning these complexity factors was derived from the site surveys and assigned on a percentage basis for each operating system. This assignment, then, becomes one of the required identification parameters carried through the model as part of the failed material specification. Thus, each transaction representing failed material must be labeled with a set of parameter values that will, in general, differ from the values carried by all other transactions. As the model is developed, the identification of required decision conditions then leads to the appropriate transaction parameter labeling. These labels create a unique specification for each transaction moving through the model.

Block S3: Block S3 represents the process whereby a failed item is systematically isolated and identified by various means of on-line testing. Most operating systems have a certain degree of Built-in-Test (BIT) capability and most have some form of diagnostic self-test routines. In addition to these, technician trouble-shooting with various types of test equipment is also represented within this block. The mean delay time represented at this block will generally depend upon the quality of these operating system self-test aids and the complexity level of the failed item signified by the incoming transaction from Block S2.

Block S4: Block S4 labels each entering transaction as a failure or non-failure. Assignment of actual failure to the transaction is a function of the accuracy of the self-test routines denoted by Block S3. The accuracy of the self-test routines is highly dependent on the amount of money allotted for that particular part of the equipment design and does not necessarily reflect faulty design or limited technology. In addition, the complexity level of the failed material has a decided impact on the ability to correctly identify the true failure. In one particular operating system, for example, approximately eight percent of all high complexity material identified as failed during the self-test routine actually had no faults at all. In this case, Block S4 would have labeled ninety-two percent of all incoming high complexity transactions as failures and the remaining eight percent as non-failures. Each transaction would then continue through the model to the subsequent decision points regardless of failure status.

Block S5: Block S5 is used to determine whether a spare item is available to replace the failed material. The method for implementing this block was influenced by the objectives of the model. Since the operating system and repair shop parameters were to be varied, the spares availability was assigned on a probability basis rather than a deterministic basis. In this way, a pure system response was obtained based on an infinite pool of spares drawn on a historical probability basis only. If a finite pool of spares were originally available, the system response would become discontinuous whenever the spares supply was depleted and this would interrupt the desired response observations.

When the determination is made that there is no spare, the failed material (regardless of true failure status) will exit to the repair shop (Point I). If a spare is available, the flow continues to Block S6.

Block S6: Block S6 represents the function of inserting the replacement spare into the operating system for verification that the correct failed material has been isolated by Block S3. In general, if the original isolated material was the true failure, a GO condition should be present. When a GO is obtained, then the failed material exits to the repair shop at Point I and the operating system returns to UP status. A NO GO will normally be obtained if an error were made in the original isolation procedure of Block S3. When this occurs, the transaction must be cycled back and reenter the flow at Block S2. These re-entering transactions will take priority over malfunctions coming from Block S1. In other words, if the operating system is down due to a malfunction, all attempts to correct that malfunction will take place before attention is diverted to any subsequent malfunction that may occur during the process of operations, troubleshooting, or verification.

Blocks S7 and S8: Blocks S7 and S8 (Figure 3) represent points of access to the operating system for purposes of verification of the findings of the repair shop. In general, they are identical to Block S6. Again, GO, NO GO decisions will depend on the true status of the failed material. At Block S7, the failed material will have gone through Block R2 without showing any fault present. At Block S7, if a GO is obtained, the item will either be returned to RFI (Ready-for-Issue) status, or the operating system returned to UP status. This decision depends on the system status when the GO condition is obtained. If a NO GO is obtained, the failed material reenters the repair shop routine at Block R1. The same operating procedure exists at Block S8 except that if a NO GO is obtained, the material, having cycled through all the repair shop capabilities, is now shipped to depot or salvage (Point II). At this time the operating system is arbitrarily placed in UP status and normal operations continue.

At all times, failed material flowing through the model must be traceable to the original operating system. In other words, Blocks S7 and S8 belong to the original system and cannot be arbitrary test points. Failed material from operating System 2 cannot be forwarded to operating System 1 for verification, etc. Therefore, a system label must also be one of the parameters of the failed material transaction.

The Repair Shop (Off-Line) Phase

The overall flow of the repair shop routine is shown in Figure 3. This phase of the model reflects the sequence

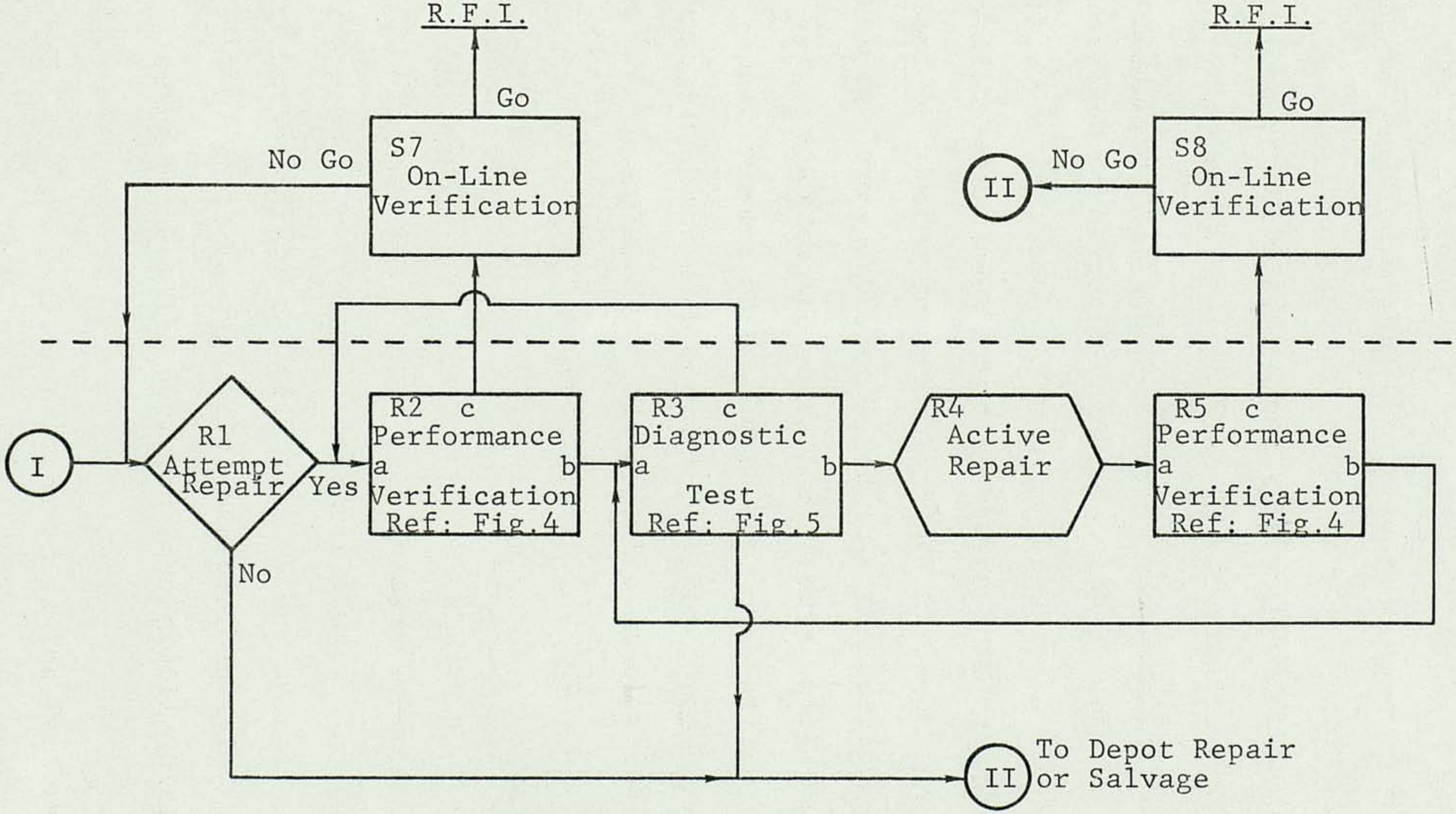

Fig. 3: Repair Shop Model

of events from the entry of failed material into the repair shop (Point I) from the operating systems. The material passes through tests, inspections, and repair actions and will periodically be forwarded to the original operating system for on-line verification tests. Blocks R2, R3 and R5 contain intricate decision conditions and will be addressed more specifically in a later section. The general description of the repair shop blocks and the required decision parameters associated with them are as follows:

Block R1: Decisions made at Block R1 depend on the maintenance philosophy associated with the operating system. A certain percentage of failed material will not even have a repair attempt made. Therefore, probability levels were established for each operating system through site surveys and data analyses. In this way, the model forwards the failed material to either Block R2, or out of the model at Point II, with the established probability values. If the failed material proceeds to Point II, the operating system is assumed to return to an UP status if it had been DOWN because of that particular failure. The decision to attempt repair or not will also depend on the complexity level of the incoming failed material and, in general, different probability decision levels will exist for each complexity class.

Block R2: Block R2 is designated as the Performance Verification Subroutine (PVS). It is shown in more detail in Figure 4 and will be discussed more specifically in the next section. Failed material enters the block at point "a" and may exit either at point "b" or point "c". Specific parameter identifications of failed material entering at point "a" of Block R2 will distinguish it from failed material entering point "a" of Block R5. Block R2 represents testing processes on the incoming failed material in order to establish the overall performance, or non-performance, of the item. If the failed material shows no fault, whether one is present or not, it will exit at point "c". Likewise if it shows a fault, whether present or not, it will exit at point "b" and proceed to Block R3. The exception to this occurs when an item returns from Block R3, is processed through Block R2 again, and is then forced to exit Block R2 at point "b" by a priority decision. Failed material exiting point "c" of Block R2 will always go to Block S7 of the original operating system for on-line verification.

Block R3: Block R3 is designated as the Diagnostic Test Subroutine (DTS) which is more involved than a performance verification test. Here, the attempt to completely isolate and identify the specific failed part, or parts, on an assembly is made. Diagnostics, in general, are more involved and more costly than performance tests. Failed material enters the block at point "a". Exits may occur at points "b", "c", or "d" depending on decisions internal to Block R3. The block is shown in more detail in Figure 5 and will be discussed more specifically in the next section.

Processing decisions will cause the failed material to exit at point "c" if no fault can be found regardless of whether one is present or not. It will normally exit at point "b" if a fault is indicated, whether or not one is actually present. Exits at point "d" usually result when the extent of damage is too great for local repair, or if it is finally determined to be beyond the repair shop's capabilities.

Exits from point "c" are recycled to Block R2. These items are given a higher priority in order to expedite their processing in Blocks R2 and R3. This priority affects the decision parameters and the order of procedure within these blocks. They will not, however, preempt any work or tests in process. This priority assignment will eventually force the failed material to exit point "b" of Block R2 and also point "b" of Block R3 on the subsequent pass through these blocks.

<u>Block R4</u>: Block R4 represents the actual physical repair action on all failed material entering it. At this block, it is assumed that all repair attempts are successful in the fact that specific instructions from the diagnostic testing results are implemented. It is further considered that no new faults will be introduced due to the repair shop activity. In essence, all material exiting Block R4

will be in a non-failure status. The Mean Time to Repair (MTTR) associated with this block was derived from site surveys and analyses of the Navy 3-M data for the different complexity levels.

Block R5: Block R5 represents the final performance verification process which validates the diagnostic and repair actions. It is the same physical activity that is also represented by Block R2. Therefore, transactions entering point "a" of Block R5 require labeling such that on-line verification attempts will go to Block S8 of the correct operating system and not Block S7. In addition, due to the probability conditions within Block R5, a good item may exit point "b", having failed the performance tests, and be returned to Block R3 for additional diagnostic testing. This is a function of the Performance Verification Subroutine capabilities, even though the item was repaired at Block R4. This portion of the flow routine is necessary because the repair technicians have no knowledge of the true failure status of the repaired item, nor can their equipment yield 100 percent accuracy, and the item must pass the tests before being certified as RFI.

Those items returning to Block R3 are assigned a higher priority in order to expedite their processing on the subsequent pass through the blocks. These items are forced to exit point "b" of Block R3 and point "c" of Block R5 on their subsequent pass through the blocks. All repaired items exiting point "c" of Block R5 will always go to Block S8 of the original operating system from which they came for on-line verification tests.

Internal Functional Details of Blocks R2, R3, and R5

Performance Verification Blocks R2 and R5

The Performance Verification Subroutine block is shown in detail in Figure 4. This block contains the actual performance testing routines and several decision subblocks which are based on the accuracy of these tests. The sub-block functions are:

Subblock P1: Performance Verification Tests
Subblock P2: Failure Status Determination
Subblock P3: Failure Indication
Subblock P4: Failure Indication

Subblock P1 represents the process of actually performing those tests necessary to verify the overall performance of the incoming failed material. The mean time to perform these tests will depend on the type of test equipment being used and the complexity level of the failed material being processed.

Subblock P2 examines the incoming failed (or repaired) material to ascertain its true failure status. This information, being unknown to the technicians, is used to route the material to either Subblock P3 or Subblock P4.

Subblock P3 operates on the probability that a true failure will have been properly detected by the tests of

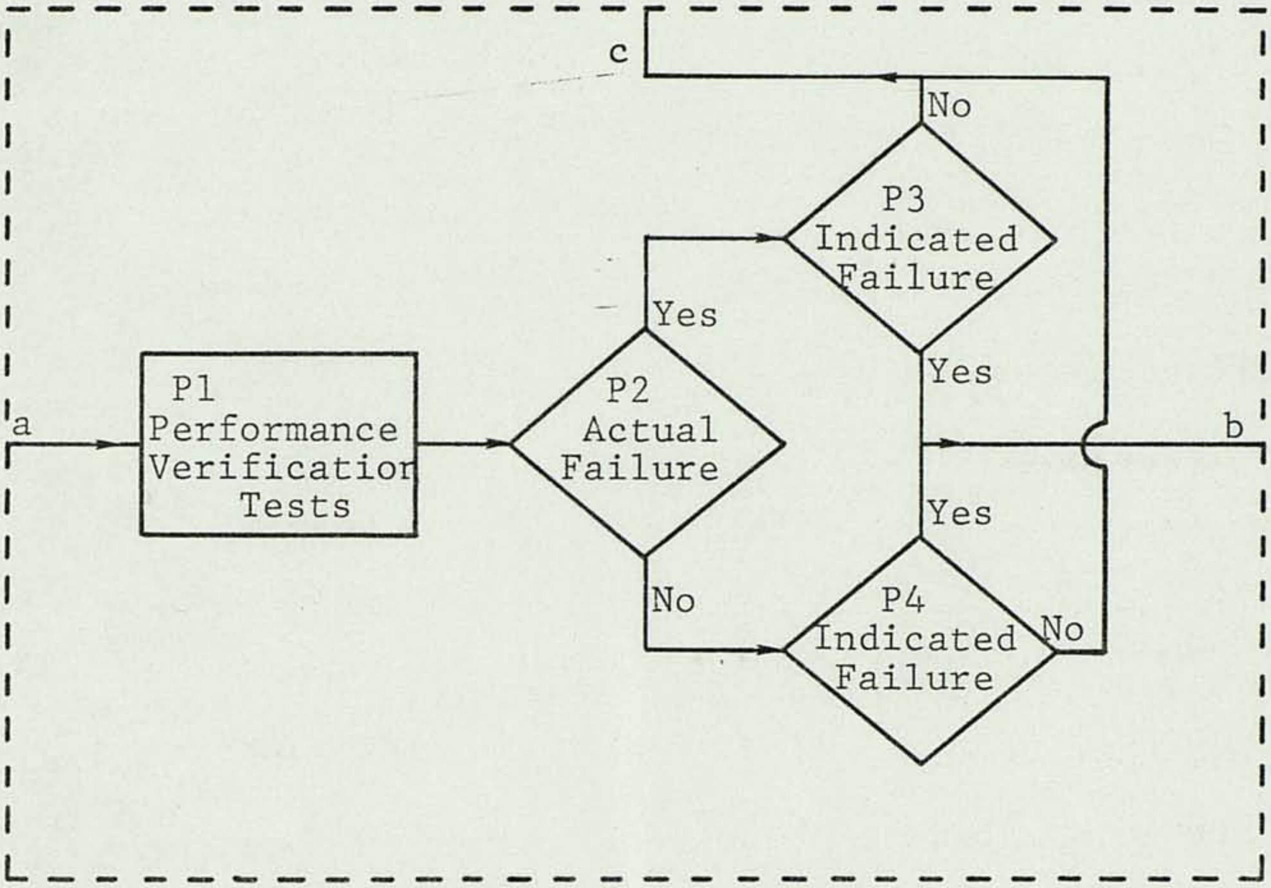

Fig. 4: Performance Verification Routine

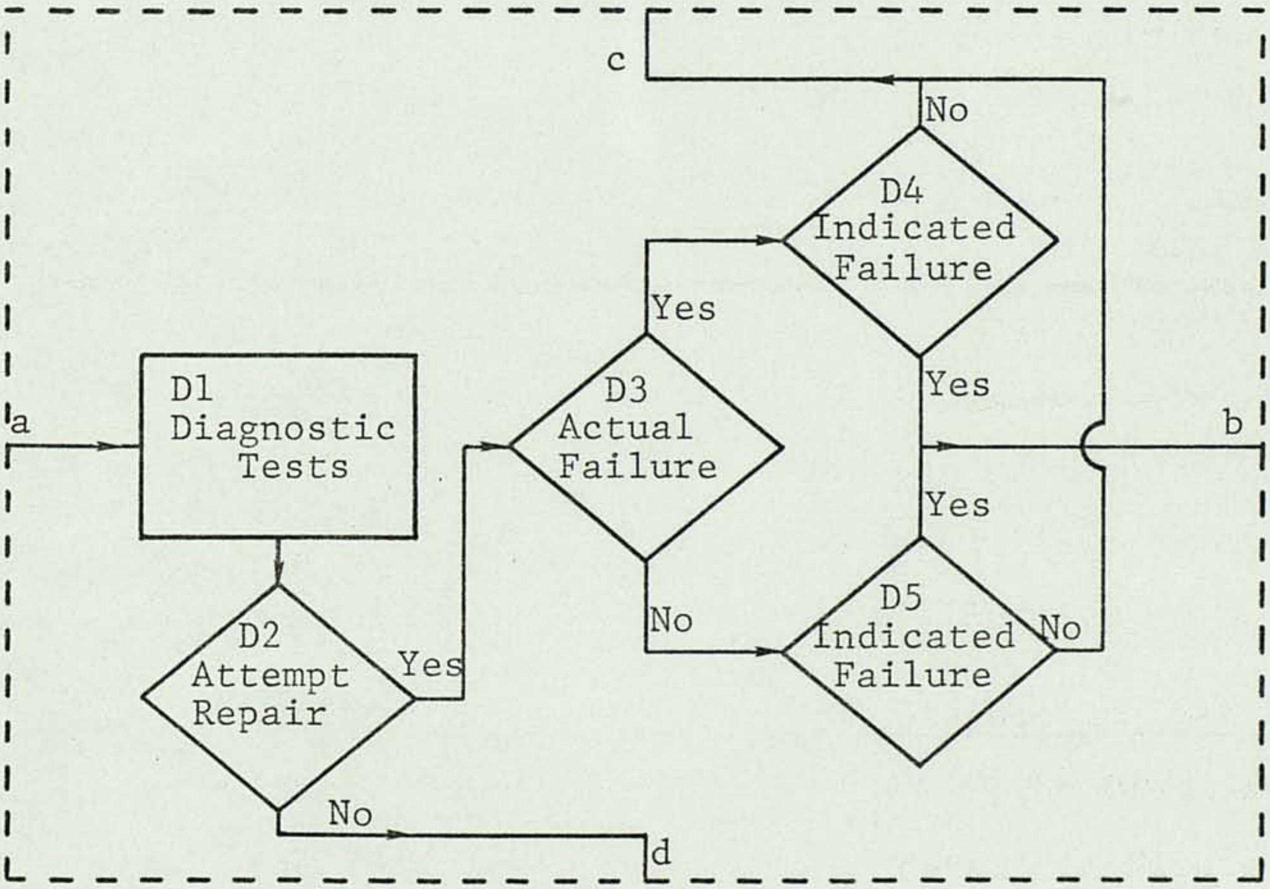

Fig. 5: Diagnostic Test Routine

Subblock P1. The probability levels assigned to this sub-block, as well as Subblock P4, are based on the field survey data and generally reflect the sophistication of the test equipment involved. If a true failure has been identified by Subblock P1 as a failure, the failed material will eventually exit point "b". If it was not detected as "failed", it exits point "c".

Subblock P4 works on the same principle as Subblock P3. However, this subblock operates on the probability that a "non-failed" item may pass through the test showing either a failure or non-failure. In other words, due to the ambiguities of the test equipment, or technician, a non-failed item may exit point "b" due to an erroneous fault indication. If it shows no fault, it will exit point "c".

In general, the majority of failed material will exit point "b" while the majority of non-failed items will exit point "c". This flow routine is maintained for all transactions except those given a higher priority elsewhere in the model. Subblocks P3 and P4 are therefore a measure of performance verification comprehensiveness as far as the capability of eliminating false alarms and undetected failures.

Diagnostic Test Block R3

The Diagnostic Test Subroutine block is show in detail in Figure 5. This block represents the most in-depth testing and diagnostic capability within the repair shop. Whereas the performance tests generally indicate a GO, NO GO decision capability, the diagnostic tests actually isolate and identify the specific failure component within an assembly or group of components in preparation for the physical repair of the failed material. In other words, the performance test may verify that an amplifier has no output, but the diagnostic test will find the open or shorted transistor or IC chips. This block, then, contains the actual test routines and several decision subblocks which are based on the accuracy of these tests. The subblock functions are:

Subblock D1: Diagnostic Tests
Subblock D2: Retention Decision
Subblock D3: Failure Status Determination
Subblock D4: Failure Indication
Subblock D5: Failure Indication

Subblock D1 represents the process of actually performing those tests necessary to isolate the faulty part or component on the incoming failed material. The mean time to perform these tests will generally depend on the type of test equipment used and the complexity level of the failed material being processed.

Subblock D2 represents a closer inspection of the failed material in order to determine whether it can be repaired in the shop. It would also consist of screening out those items too badly damaged to be retained. This subblock works on a probability basis derived from the site surveys. If the item is not retained, it will exit point "d", otherwise, the flow is to Subblock D3. The complexity level of the failed material will influence the probability decision levels; failures in high complexity material are generally more difficult to isolate than those in low complexity material.

Subblock D3 examines the incoming material to ascertain its true failure status. This information, being unknown to the technician, is used to route the material to either Subblock D4 or Subblock D5.

Subblock D4 operates on the probability that a true failure will have been properly identified by the tests of Subblock D1. The probability levels assigned to this subblock, as well as those of Subblock D5, are based on the field survey data and reflects the sophistication of the test equipment and procedures involved. If a true failure has been correctly isolated by Subblock D1, it will exit point "b". If it was incorrectly diagnosed as a non-failure, it will exit point "c".

Subblock D5 works on the same principle as Subblock D4. In contrast to Subblock D4, this subblock operates on the probability that a "non-failed" item may pass through the tests of Subblock D1 indicating either a failure or non-failure. Again, due to ambiguities in test equipment, procedures or technicians, a non-failed item may exit point "b" due to an erroneous fault indication. If it shows no fault, it will exit point "c".

In general, the majority of failed material will exit point "b" while the majority of non-failed material will exit point "c". This flow routine is maintained for all transactions except those assigned a higher priority elsewhere in the model. Subblocks D4 and D5 are therefore a measure of diagnostic comprehensiveness as far as the capability of isolating the true cause of failure and the elimination of false alarms.

III. PROGRAMMING THE SIMULATION MODEL

General Description of the Programming Language

The programming language used in this simulation model is entitled FLOW SIMULATOR, or sometimes FLOWSIM for short. The best general description is that the language consists of: (1) dynamic entities called "transactions" which represent some unit of reality moving through the system, (2) equipment entities that represent elements of the system acted upon by the transaction, (3) statistical entities that are designed for the purpose of measuring the system's behavior, and (4) operational entities called "blocks" that provide the logic for directing the transactions through the system. The language is a free form style containing four fields and requiring a single space to separate the fields in the statements. However, for sake of clarity, the following conventions are generally observed:

(1) Columns 1 - 8 Location or Reference Field

(2) Columns 10 - 18 Operational Field

(3) Columns 20 - xx Variable Field

(4) Columns 40 - 80 Comments or Remarks

The Location Field is used for labels and identifying tags which are called, or addressed, by other program statements. They serve the same general function as FORTRAN numbered statements used for GO TO's, DO's, etc. The simulation automatically assigns sequential block numbers to operational program statements for use in the output report. When the Location Field is used for a label name, this name supercedes the block identification number. Labeling is optional and may be used because of the need to "go to" that particular block from other parts of the program, or simply for user identification in place of block numbers in the output.

The Operational Field is mandatory and provides the basic directional control of the system. The operational field contains such statements as SEIZE, RELEASE, ASSIGN, TRANSFER, and other system control functions.

The Variable Field is used to specify and/or modify the requirements dictated by the Operational Field. It consists of subfields A through G which are used as required by the programmer and have various specifications related to the type of statement in the Operational Field. For instance, Subfield A represents a mean delay time for an ADVANCE statement, but is used as a probability, or percentage value, in a TRANSFER statement. The Variable Field uses as many columns as necessary for complete specification, provided there are no embedded blanks in the field. Subfields not used are set apart by commas until the last required subfield is specified.

The Comment Field uses descriptive information as an aid to the programmer and/or reader. Comments do not enter into the simulation program but are reproduced at the output on the program listing. Additional comments can be entered anywhere in the program by placing an asterisk (*) in column one of the statement. Then all eighty columns are treated as comments.

More specific language details will be discussed as required in the programming of the model. For additional FLOW SIMULATOR information the reader should refer to the reference manual.[4] In addition, brief general descriptions of the more popular GPSS often appear in textbooks on Queueing Theory and adequately apply to the general description of FLOW SIMULATOR.[5]

---

[4]Sperry Rand-Univac, Flow Simulator Reference Manual, Series 70 Publications (Cinnaminson, N.J.: Sperry Rand, 1972).

[5]Donald Gross and Carl M. Harris, Fundamentals of Queuing Theory (New York: John Wiley & Sons, 1974), pp. 401-5.

Programming the Operating System Phase

Establishing the Transaction Specifications

Although normal computer program documentation usually contains a flow chart, a subtle advantage of FLOW SIMULATOR is that one can usually program directly from the system definition or flow diagram. Therefore, no separate flow chart is given and the programming is developed from the system descriptions referred to in Figures 2, 3, 4, and 5.

In Figure 2, the initiation of activity begins at Block S1 which represents system failures. This corresponds directly to the GENERATE statement. Subfield A of the GENERATE contains the mean arrival rate at which a transaction occurs. From the site surveys and later 3-M substantiation, the mean-time-between-failures (MTBF) was computed for the system and entered in terms of minutes to improve accuracy and also because the simulation language cannot work in fractions. Therefore 1.6 hours would be 96 minutes, a whole number. Subfield B is the mean time modifier, and for equipment reliability, follows an exponential distribution. This is defined by a function statement called EXPON which appears near the end of the program listing. So Subfields A and B uniquely define the Poisson arrival (exponential inter-arrival) of transactions representing the failures of each operating system. In order to measure statistics more readily, each operating system program routine is identical except for those system parameters and decision probabilities unique to a specific system. In essence, the corresponding Operational Fields of each operating system simulation program is identical with only their corresponding Variable Fields differing.

After analyzing the system operating requirements it was determined that a minimum of three basic identification parameters was required for each transaction. During the programming it was later found that four more parameters were required for special routines and comparison tests. The following parameter descriptions are given:

Parameter 1: Complexity Level

1 = Low Complexity
2 = Medium Complexity
3 = High Complexity

Parameter 2: Failure Status

0 = Non-failure
1 = Failure

Parameter 3: Subroutine Selection Mode (SBR)

This parameter is reserved for the special SBR TRANSFER function. Whenever this mode is selected, the block number from which the transfer was made is automatically stored in Parameter 3 and the transaction moves to the called SBR block. At the end of the subroutine, the transaction can return to the normal program flow at any desired point with the statement, TRANSFER P,3,X which adds "X" units to the block number stored in Parameter 3. "X" can be any numerical value.

Parameter 4: Operating System Identification.

This parameter is used to uniquely identify the original operating system in which the failure, or transaction, occurred. One of the most beneficial functions used in the simulation is the Parameter Selection Mode (P) of the TRANSFER statements. By storing a block number (similar to the automatic SBR function) in Parameter 4, tests can easily be made such that transactions are returned to any point in the program by the statement TRANSFER P,X,Y where "X" is the parameter number and "Y" is the number of units added to the block number represented by "X". However, extreme caution is required due to the fact that any program change which upsets the sequential block numbers can invalidate the TRANSFER P,X,Y statement. Therefore, most of these statements have been addended with the comment, "BLOCK XFER NUMBER", for rapid identification and review.

Parameter 5: Criticality Identification

0 = Non-critical, spare was available
1 = Critical, no spare available

This parameter was used to identify those transactions entering the repair shop without having a replacement spare. In other words, the operating system would be in DOWN status awaiting the repair of this item. However, no special priority is given to the transaction.

<u>Parameter 6:</u> On-Line Test Block Identification

6 = Block S6
7 = Block S7
8 = Block S8

At appropriate points in the model, the transactions would be labeled with one of the numbers 6, 7, or 8 in this parameter to identify which on-line test position is to be utilized for the on-line verification tests. The parameter is also used to identify appropriate QUEUE's set up to measure backlog at these points in the model.

<u>Parameter 7:</u> Clock Time of Failure

In testing a transaction to determine if it was the specific one causing a DOWN system status, some unique attribute had to be developed. It was determined that: (1) due to the basic definition of the random generator, no two failures within the same system could occur at the exact same clock time, and (2) since the internal transaction number could not be accessed by the user, that this would be the only truly unique parameter among all the possible permutations of identifying parameters of the transaction. This parameter was needed to identify those transactions returning from the repair shop for on-line tests at Blocks S7 and S8 (Figure 3) which may have the system held in a DOWN status.

These seven parameters then, completely define each original transaction and through parameter changes within the model, provide complete and accurate routing instructions to all transactions.

Operating System Program Block Functions (Blocks 1-60)

Program blocks discussed in the following paragraphs refer directly to the program listing in Appendix A. The block numbers appear in the lefthand column of the listing.

Block 1: Having established the parameter requirements, the GENERATE statements could now be completely defined for each operating system. Subfields D and E were not required, so that the value "7" was placed in Subfield F and the letter "F" placed in Subfield G indicating that the parameters had to be FULLWORD (4 Bytes) lengths in order to accommodate the clock time. Even though all the other parameters need only HALFWORD lengths, they cannot be separately specified. A five clock unit (minutes) offset was specified in Subfield C in order to avoid "premature" failures that might conflict with a parameter change value of "1" used in Parameter 7 later in the program. In other words, a failure at one minute after starting wasn't allowed. This could have been changed to a one minute offset and the value "0" used later in Parameter 7, but there does not appear to be any measurable impact on system results. The only impact of the value in Subfield C is to set the limit for the earliest possible occurence of the first transaction.

Block 2: During the development of the programming, periodic use of the TRACE capability was made. This provided a complete audit trail of the movement of a transaction through the system. When the debugging proved satisfactory the TRACE was removed. In order to avoid extensive changes to the Parameter Selection Mode TRANSFER statements, the TRACE was merely replaced by an ADVANCE 0 statement. This maintained the order of the sequential block numbers without causing any simulation change. Execution of an ADVANCE 0 requires no clock time usage and is immaterial to the simulation model.

Block 3: The operating system identification is placed in Parameter 4 by an ASSIGN statement. The numerical value (24) corresponds to the block number for the QUEUE representing items awaiting the use of Block S6 of Figure 2 or Blocks S7 and S8 of Figure 3.

Block 4: This block is a QUEUE to measure the number of transactions entering the system. In addition, if the system is in DOWN status, no new transaction can enter to SEIZE the system and it must remain in the QUEUE.

Block 5: This block tests the system status in order to allow transactions to leave the QUEUE and SEIZE the system only when the system is UP.

Block 6: Transactions SEIZE'ing the system create DOWN status situations and represent failures entering the flow model.

Block 7: This block places the clock time when the transaction was created into Parameter 7.

<u>Block 8</u>: The clock time in Parameter 7 is transcribed into a SAVEVALUE location for later tests. Notice that no change can occur in this value until a new transaction is allowed into the system, or as long as the system is DOWN.

<u>Blocks 10-16</u>: These program blocks perform the function of Block S2 in Figure 2. By using the probability TRANSFER modes in blocks 10 and 11, the transaction is routed to the appropriate ASSIGN statement. After ASSIGN'ing the complexity number in Parameter 1, all transactions are "Unconditionally" TRANSFER'ed to Block 17 in the program. The probabilities used in the TRANSFER blocks (10 and 11) were derived from the site surveys.

<u>Block 17</u>: This block provides the appropriate delay time corresponding to Block S3 of Figure 2. The time and estimated distribution was derived from the site surveys and is implemented by the operational ADVANCE statement. Variable Subfield A is the mean delay time and Subfield B represents the spread ($\pm$). A uniform distribution was used in most ADVANCE statements due to a lack of knowledge of the actual time distributions, which are most likely log-normal.

<u>Blocks 18-21</u>: These blocks ASSIGN the appropriate failure status to the transaction based on the probability of correct fault isolation. The probability value derived from the site surveys is used in the TRANSFER statement and then all transactions move to Block 22 after ASSIGN'ment.

<u>Block 22</u>: This block corresponds to the spares decision Block S5 of Figure 2. The probability value used in the TRANSFER statement was derived from the site survey. If there is no spare, the transaction TRANSFER's to block 61. Otherwise, the transaction proceeds to the next block.

<u>Block 23</u>: This block establishes the appropriate parameter value in Parameter 6 so that distinction can be easily made as to which on-line verification test position the transaction is required to enter. This particular block identifies Block S6 of Figure 2 as the correct on-line verification test position. In similar fashion program block 68 identifies Block S7 of Figure 3 and program block 93 identifies Block S8 of Figure 3.

<u>Block 24</u>: This block establishes the QUEUE for all transactions awaiting on-line verification testing. No transaction can access the operating system while it is in a SEIZE'd status. This QUEUE provides a holding point as required and yields a total count of all attempts to use the on-line verification tests. Block 24 is also the critical system identification point representing the point of access of all returning transactions from the repair shop (see Block 3 explanation).

<u>Block 25</u>: This block represents one of the critical applications of the Parameter TRANSFER function. Variable Subfield C causes twenty units to be added to the value of Parameter 6. Therefore, transactions are sent to either

block 26, 27, or 28 as necessary.

Blocks 26-28: These are merely unconditional TRANSFER blocks to route the transaction to either test Block S6, S7, or S8 as required by the transaction specification. The correct TRANSFER decision was actually made by block 25. These combinations illustrate the power of application of the various modes associated with the TRANSFER statements. Transactions that have moved successively through the model now begin following diverse paths. Some transactions have already diverged from Block S5. Those entering Block S6 are now moved to program block 50 for continuation of the flow described by Figure 2.

Blocks 29-31: These blocks form a small subroutine for holding transactions that failed to gain access to the operating system after returning from the repair shop. The loop can only be entered after the transaction has attempted to enter Block S7 or S8 and failed. It then provides a delay of ten clock units (block 30) before attempting to enter the system again. Some transactions may RECYCLE many times before being able to access an UP status operating system. This loop is analogous to repaired material arriving at a point and waiting for periodic assessment by a technician as to whether the operating system is available for an on-line verification check. It does not affect the statistical values pertaining to the actual repair cycle, nor does it impact those statistics of the operating system that are being evaluated.

Blocks 32-49: These blocks establish the QUEUE's, TEST's, and delays associated with Blocks S7 and S8 of Figure 3. It is part of the operating system, but is applicable to the repair shop cycle only. In other words, no transaction can enter this loop without first entering Point I of the repair shop. Therefore, all transactions returning to this point must wait until the operating system is available before a test can be attempted. In FLOWSIM, QUEUE's may have names or numbers at the discretion of the programmer. Therefore, use is made of the QUEUE number method so that both Blocks S7 and S8 transactions can flow through the same subroutine while being separately enumerated. Block 32 adds forty units to the value of Parameter 6 so that Block S7 transactions enter QUEUE 47 and Block S8 transactions enter QUEUE 48. After passing through the loop, block 43 restores the original Parameter 6 values. Block 34 checks to see if the system is SEIZE'd. If it is, the transaction moves to block 38 for identification. If a transaction SEIZE'd the system at Block S1, had no spare at Block S5, then the system would be DOWN awaiting repair. This returning transaction from the repair shop carries its original clock time at failure in Parameter 7. The transaction originally downing the system had its clock time at failure stored in SAVEVALUE 5. These two values are now compared and then, if and only if they are equal, the transaction is granted immediate access to the operating system at block 35. If they are not equal, some other transaction has the system downed and the current transaction is RECYCLE'd to block 29, q.v., for a later check. In order to avoid disruption of the QUEUE number now in Parameter 7, RECYCLE'd transactions, after being delayed by block 30, will unconditionally TRANSFER over block 32 directly to block 33, retaining the correct Block S7 or S8 identification.

If the system was UP (not SEIZE'd) when checked at block 34, the transaction will move to block 35, SEIZE the system, then move to block 36. Here the clock time becomes unimportant since the system was either restored by a spare (Block S6) or by the original DOWN'ing transaction, and Parameter 7 is set to the value "1", a value that cannot represent a failure time due to the specifications in the GENERATE statement. The transaction is now unconditionally TRANSFER'ed to block 39 where all transactions gaining access to the operating system at either Block S7 or S8 are delayed by the mean time for conducting the on-line verifications. The transactions are tested by block 40 again using the Parameter 7 value against the SAVEVALUE. Those transactions having just SEIZE'd the system at block 35 will pass the NE (not equal) logic and move to block 41 to RELEASE the system. Those transactions failing the NE logic will JUMP to block 42 and eventually all transactions that entered this loop at block 32 then move on to block 45.

At block 45, the true failure status is checked with non-failures moving to block 46, and then to block 148 for further processing. Failure transactions move to block 47 where the correct test Block (S7 or S8) is ascertained. Failures (NO-GO's) exiting Block S7 must reenter the repair shop routine, therefore, these transactions move from program block 48 to block 62. Failures (NO-GO's) exiting Block S8 move to Point II of Figure 3, so these transactions move from block 49 to block 142 for further processing.

Blocks 50-60: Block 50 is the actual continuation point for the flow of transactions represented exclusively by Figure 2, having arrived at this point from program block 26. The intervening blocks from 29-49 are merely a convenience location for the program functions that they perform and could have just as easily been placed between current blocks 60 and 61 provided all parameter TRANSFER mode statements are properly adjusted.

Block 50 provides the on-line verification time delay before the transaction moves to block 51. The mean time was derived from the site surveys and is a common value for Blocks S6, S7, and S8 for a given operating system.

The failure status is ascertained by block 51, sending non-failures to block 52 and failures to block 53. A non-failure at this point in the flow reproduces the condition where a spare item does not verify the original malfunction during verification tests. Therefore, another failure must be looked for and the transaction cycles back to Block S2 at program block 10 via blocks 58-60. Block 58 raises the priority level of the transaction for immediate processing. It is immaterial at this point for model purposes, how the spare is handled. That is, it could be left in the system and the originally pulled item sent to storage, or the spare returned to storage after reinstalling the originally removed item, since either item is assumed to be good. No attempt was made at this time to model those conditions where the spare itself was defective when drawn from storage. Slight modifications to the program can do this, but it was not an objective of this model.

Transactions arriving at block 53 and representing failed material must now perform two distinct functions. One function must be that of the original defective item moving to the repair shop, and the other represents the good spare that restores the system to UP status. This is accomplished at block 54 by the SPLIT operation which makes an identical copy of the original transaction including all the parameter specifications of the original. The original transaction now moves to block 55 while the copy goes to block 62, the repair shop entry point.

Continuing from block 55, the system is RELEASE'd from the originating failure condition and the transaction moves out of the system from block 57 to block 151.

Blocks 61-152 contain the repair shop routine and other "housekeeping" functions and essentially completes the simulation model. Any additional operating systems are added to the model beginning with block 153. Each corresponding operating system function can be found by adding 152 to System 1's block numbers and then multiples of sixty thereafter. For instance, the beginnings, or GENERATE statements, of each operating system are found as follows:

Operating System 1 = Block 1
Operating System 2 = 1 + 152 = Block 153
Operating System 3 = 153 + 60 = Block 213
Operating System 4 = 213 + 60 = Block 273
Operating System 5 = 273 + 60 = Block 333
Operating System 6 = 333 + 60 = Block 393
End of Model = 393 + 60 = Block 453

Block 453 and all following statements would be moved forward or backward as required by the number of operating systems located at the site being modeled.

Programming the Repair Shop Phase

Repair Shop Program Block Functions
(Blocks 61-105)

Blocks 61-62: These blocks correspond to the entry Point I of the repair shop. Block 61 establishes the identification of that failed material coming in which had no available spare and is therefore holding the system DOWN. These transactions and all other transactions then enter the repair shop flow via block 62 which establishes the common non-priority of all work entering the shop for the first time.

<u>Blocks 63-66</u>: Block 63 ascertains the complexity level of the failed material which is then processed, on a probability basis, into the repair shop at block 67 or out of the repair shop via block 142 (Point II). These blocks perform the function of Block R1 of Figure 3.

<u>Blocks 67-78</u>: Transactions now enter the repair shop QUEUE labeled RETAIN at block 67. Parameter 6 is set to represent Block S7 and the transaction now enters the Performance Verification Subroutine (Block R2, point a) at block 69. The internal programming of this subroutine is explained in the next section. Items indicating that a failure is present by the Performance Verification Subroutine (PVS) reenter the flow at block 73. Those items indicating non-failure reenter the flow at block 70, having exited Block R2, point c, and must then move to Block S7 of the original operating system. This is accomplished by block 72 which TRANSFER's to the block number stored in Parameter 4. For System 1, this number is twenty-four.

Transactions indicating failures by Block R2 and those that are given a higher priority returning from Block R3 through Block R2 will continue in the program from block 73 to block 74. Here they enter the Diagnostic Test Sub-Routine (DTS) and may be either passed to Point II, or show failure, or non-failure by the DTS. The internal programming of this subroutine is explained in a later section. Transactions showing non-failure reenter the flow at block 75, have their priority raised by block 76 and TRANSFER'ed to block 69 for reprocessing in the PVS. Those transactions indicating failure by the DTS and those with higher priorities established elsewhere in the model reenter the flow at block 78 and continue to block 79.

Blocks 79-80: These blocks are used to reestablish all transactions coming from Block R2 as non-priority while maintaining the priority established for those returning from Block R5. Block 79 does this by checking Parameter 6 to determine which on-line verification Block, S7 or S8, is stored in the parameter field.

Blocks 81-93: These blocks implement the requirements of Block R4 of Figure 3. The complexity level is assessed by block 82 and the transactions forwarded to the appropriate delay blocks by blocks 83-85. Blocks 86-90 represent the mean repair times for each complexity level which were derived from the site surveys. All transactions then arrive at block 91 which sets the failure status to "0". In this way, all transactions exiting the repair function block are presumed to be good. Block 93 then sets Parameter 6 to the value "8" (Block S8) which would be the next on-line test position for all transactions leaving Block R4.

Blocks 94-105: Block R5 of Figure 3 is implemented by program blocks 94-99 while blocks 100-105 represent those transactions still indicating failure status by the PVS and must go back through Block R3.

Transactions indicating "no failure" by Block R5, and those with higher priority levels established elsewhere in the model, reenter the flow at block 95 and move to the on-line verification Block S8 via block 97. Transactions indicating "failure" by Block R5 reenter the flow at block 98, are elevated in priority by block 99 and are cycled back to the DTS (Block R3) via block 100. Blocks 101-103 are non-functional in this location since no transaction can exit Block R3 at this point in the flow process. By definition of the system requirements, priority transactions must always exit Block R3 at point b which corresponds to program block 104 here. Therefore, merely symmetry was maintained due to the DTS being called from two different locations, and by sending transactions back from the DTS using one logic function. Blocks 101-103 could easily be replaced by ADVANCE 0 statements. So, all transactions exiting Block R3 from this program location reenter the flow at block 104 and move to the repair function (Block R4) at block 81 via block 105.

Performance Verification Subroutine
(Blocks 106-122)

Whenever the statement TRANSFER SBR,PERFVER,3 is encountered, transactions are sent to the PVS section of the program. These program blocks represent both Block R2 and Block R5 of the Repair Shop flow and the correct flow position of the transaction is "remembered" by storing the program block number which called the subroutine in Parameter 3 of the transaction. The subroutine is called from blocks 69 and 94 of the repair shop flow program.

Transactions entering the PVS are checked for complexity level by block 107 and then TRANSFER'ed to the appropriate test time delay block. These mean test times for the complexity levels were derived from the site surveys. All PVS transactions later arrive at block 116 which TEST's for non-zero priority. Zero, or non-priority items, proceed to block 117 (Subblock P2 of Figure 4). Priority items are sent directly to another TEST at block 120.

Block 117 performs the requirements of Subblock P2 by ascertaining the failure status and routing non-failure transactions to block 118 (Subblock P4) and sending failure transactions to block 117 (Subblock P3). At these blocks, the probability levels representing the PVS test accuracy causes non-failure indications to go to block 122 and failure indications to go to block 121 regardless of the true failure condition which cannot be known by the operators.

Block 120 TEST's the non-zero priorities and separates Priority 1, sending them to block 121, from Priority 2 transactions which go to block 122. This is the logic that implements the system operating requirements that priority material must exit Block R2 at point b (block 121) while exiting Block R5 at point c (block 122), and at the same time route non-priority material to either exit based on preassigned probabilities operating on the actual failure status.

Diagnostic Test Subroutine (Blocks 123-141)

Whenever the statement TRANSFER SBR,DIAGNOST,3 is encountered, transactions are sent to the DTS section of the repair shop flow. Although this block (R3) is used in only one position in the repair shop flow, it is called from program blocks 74 and 100. This is done in order to maintain the transaction flow sequence and to "remember" the flow position by storing the program block number which called the subroutine in Parameter 3 of the transaction. In this way, Priority 1 material returning from Block R2 and Priority 2 material returning from Block R5 are easily handled in the logic.

Transactions entering the DTS are checked for complexity level by block 124 and then TRANSFER'ed to the appropriate test time delay block. These mean test times were derived from the site surveys. All DTS transactions eventually arrive at block 133 where the decision is made to keep the item and attempt repair or ship it to the depot or manufacturer (Point II of Figure 3). This decision is based on the probability value derived from the site surveys and is a function of the test equipment capability. Those items shipped out leave the system model via block 140. Those transactions continuing through the model from block 133 move to block 134 where they are TEST'ed for non-zero priority. All priority transactions are required to exit Block R3 at point b and this is accomplished at block 134 by sending non-zero priorities to block 138. Non-priority transactions continue to block 135 for failure status determination.

Non-failure transactions move to block 136 (Subblock D5) while failure transactions move to block 137 (Subblock D4). At these blocks, transactions are routed, on a probability basis, to block 138 if they indicate failure, and to block 139 if they indicate non-failure regardless of the true failure status. These blocks cause the transactions to TRANSFER back to the appropriate flow position in the model by examining Parameter 3. Therefore, blocks 134-139 execute the logic required to send all non-priority transactions out of Block R3 via either points b or c based on failure status and failure indications, while always forcing priority transactions out via point b only.

Miscellaneous Block Functions

Blocks 142-147: Additional program steps are required to handle the transactions reaching Point II of Figure 3.

These are required in order to provide model continuity and implement the assumption that a DOWN system must be returned to UP status at this point. All transactions reaching Point II are sent by the model to block 142. At this time, two basic conditions exist: (1) either the transaction represents a defective item which was replaced by a spare and the operating system is UP, or (2) the transaction represents the actual failed item for which there was no spare, and the operating system is therefore DOWN. Recall that these decisions were made at Block S5 and implemented at program block 22. If there was no spare, the transaction went to block 61, and had Parameter 5 set to the value "1". It is only condition (2) which must be further processed. Condition (1) material is sent to block 151 via block 146 by block 145 and essentially exits the model.

Condition (2) material is now sent to block 147, by block 145, which checks Parameter 4 to find which operating system it came from. In the case of System 1, thirty-two units are added to the Parameter 4 value of twenty-four giving block 56 as the return point. Block 56 correctly RELEASE's the system since no other transaction can be responsible for the DOWN condition. The transaction now moves out of the system via block 57 to block 151.

Blocks 148-150: These blocks implement that part of the model whereby a repaired item has indicated a GO condition at Blocks S7 or S8 of Figure 3. Repaired material that now tests good still has the two conditions explained in the section above and, therefore, require additional handling to restore the DOWN system. Condition (1) material is sent to block 151 via block 149 by block 148 and essentially exits the model. Condition (2) material is sent back to the original operating system by block 150. Actually blocks 148-150 are identical to blocks 145-147, and are not necessary if additional statistics are not needed by the modeler. Removing them, of course, will require adjustments in all the parameter TRANSFER statements following their position in the program.

<u>Blocks 151-152 and Block 453</u>: All transactions GENERATE'd in the model must be TERMINATE'd at the end of their path through the model. The method of termination is at the option of the programmer and depends on the type of system being modeled. Here, the system operates against the clock and is TERMINATE'd at the end of each calendar quarter by the transaction reaching block 151 after the simulation clock reaches 131490, the approximate number of minutes in a quarter year. This is done by a TEST against the clock time. All transactions less than or equal (LE) to the required clock value move to block 152 and are destroyed. That one transaction arriving at block 151 after the system clock exceeds the set value will fail the logic test and move to block 453 where the simulation run is then stopped. More than one quarter year of simulation is accomplished

by placing the appropriate combination of START and RESET controls at the end of the program. In this way, once the model is started, all transactions will move through the system in a continuous fashion while a statistical posture of the model can be obtained for the end of each quarter, in essence a "picture" of the operation on a quarterly basis. These quarterly status reports are contained in the Appendices.

IV. MODEL RESULTS

The simulation model was exercised to produce four quarterly outputs for each simulation year. Several output formats are available to the programmer, but typical standard results include Block Counts, Facility Statistics and Queue Statistics. These simulation model outputs appear in Appendices B through D.

The Block Counts provided in Appendix B allow the programmer to determine how many transactions pass through any given program block in the period of interest. In this way, a useful method is available to measure work load (quantitatively) at any point in the work flow. The program block numbers are sequential, corresponding directly to the program listing in Appendix A. Where the Location Field has been labeled, the label name appears in lieu of the block number. This facilitates rapid identification and visibility of important work flow points. For example, during the first quarter of simulation, 149 transactions representing failed items for which there was no spare available entered the repair shop via the LOCOFF block. Sequentially, this is block number 61. Referring to Appendix A shows that this program block adjusts Parameter 5 of the transaction to establish the necessary decision criterion for later use in the model flow (blocks 145 and

147).

At the top lefthand corner of the first page of each quarterly output in Appendix B are the terms "Relative Clock Time" and "Absolute Clock Time". The Relative Clock keeps track of each simulation period and provides the quarterly cue to produce the periodic "picture", or status, without disturbing the simulation run. This action is controlled by the START 1,,1 and RESET statements at the end of the program listing. After the third RESET, the program encounters only the START 1 and then the END statements. In essence, this is the quarterly simulation control and the Absolute Clock yields the cumulative time since the beginning of the simulation run. This effectively labels the quarter by observing the Absolute Clock to the nearest whole multiple of the Relative Clock Time, i.e., one, two, three, or four.

Appendix C provides the facility utilization statistics which in this model are used to determine the operating systems' availability. Utilization normally measures the percentage of system seizure, or service time in queueing systems. However, a most important function used to measure an operating system's performance is its availability, or percentage of UP time (non-seized by a failure). Therefore;

SYSTEM AVAILABILITY = 1 - FACILITY UTILIZATION

In addition, the "Number Entries" column reveals how many times the system was seized due to failures and on-line verification tests. The "Average Time/Trans" column shows the mean down time in minutes per seizure. The remaining columns are unimportant to this model.

Appendix D contains the Queue Statistics for the model and shows how many transactions passed through designated measuring points in the model. The "Queue ID" column lists the labeled Queues as well as the numbered (unlabeled) Queues. Queues three through six are associated with the repair shop and Queues 47 through 98 relate to the On-Line Verification Blocks S7 and S8 of each operating system. They are associated in pairs; 47 and 48 relate to Blocks S7 and S8 of System 1, 57 and 58 to System 2, etc. Important columns in this output format are the "Total Entries", and the "NZ-Average Time/Trans" which yields the average delay time for passing through certain portions of the model. Proper placement of the QUEUE/DEPART set will allow delay measurements anywhere in the flow. As an example, the queue, REPAIR, reveals that there were ninety-five entries into Block R4 of Figure 3 during the first quarter simulation. The average transit time through Block R4 for these ninety-five transactions was 26.653 minutes. The QUEUE begins in program block 81 and DEPART's at program block 92. However, the queue PERFVER is a little more complicated due to several decision variables in the model. This QUEUE indicates that Blocks R2 and R5, combined, serviced 255 transactions with the average transit time through the Figure 4 block being 87.322 minutes. An easy deduction by looking at Figure 3 would be that 95 of those 255 transactions entered the Block R5 portion of the flow, leaving 160 entering at Block R2. This can be verified by looking at program block 69, labeled SECOND, which represents the entry point of Block R2. In Appendix B, the block labeled SECOND (69) shows 160 entries for the first quarter, which checks with the above deduction. In this manner, the programmer is able to efficiently label and read important decision points in the model for rapid retrieval of needed data.

Appendices B through D then represent the baseline model output statistics as defined by the site surveys and historical data inputs. The model contains several important decision parameters which were subsequently varied to observe system sensitivity, response, and work load changes. Among the most important variables exercised were: (1) Self-test Accuracy represented by Block S3 of Figure 2, (2) Spares Level Probability represented by Block S5 of Figure 2, (3) Performance Verification Accuracy represented by Blocks R2 and R5 of Figure 3 and implemented at Subblocks P3 and P4 of Figure 4, and (4) Diagnostic Test Accuracy represented by Block R3 of Figure 3, and implemented by Subblocks D4 and D5 of Figure 5.

The actual computer outputs of these variations will not be included in this report due to the sheer volume of the data, but the pertinent results are summarized in the following sections.

BIT or Self-Test Accuracy

This variable essentially determines the probability of correctly pulling the original failed material. Its impact on the system should be reflected in the percentage of recycled non-failures from Block S6 to Block S2 and in addition, there should be an impact on the repair shop due to non-failed items exiting to Point I from Block S5. With more recycles and check-outs of non-failed items, it would be expected that system availability should drop with a drop in self-test accuracy.

The statistically random failure rate is also driving the system response which can be seen by the quarterly deviations in the base line data. So an important finding will also be whether changes in decision variables create any perceptive deviation from the random baseline patterns. Generally speaking, lower failure rate systems have higher availabilities, but this parameter is also impacted by delay times within the system for repair and tests. In order to help overcome the random failure deviations, the availability of each system for all four quarters was averaged and then used for comparison with the resulting variations caused by changes in the decision variable.

Table I presents the impact on system availabilities due to varying the self-test accuracy.

The Baseline values for the operating systems were generally different, depending on each specific design. They were: (1) 99, (2)-(4) 90, (5) 70, and (6) 96 percent for Systems 1-6 respectively. In order of the programmed failure rates, the operating systems can be loosely categorized as low, medium and high failure rate systems. In general, System 4 and System 5 are low failure rate, with System 3 high. System 1 is on the high side of a low failure rate system. These categorizations are relative to each other but show interesting groupings when observing responses to parameter variations.

Table I shows a general positive correlation of all systems when the self-test is improved. Note that Systems 1, 4, and 5 are least responsive (small coefficients of X) and Systems 2 and 6 are most responsive.

Another index reflecting system response is the Total System Maintenance Time (TSMT). These values can be extracted from Appendix C data by:

$$TSMT = \sum(\text{Number Entries x Ave. Time/Trans})$$

This index reflects the total down time encountered by the operating systems due to (1) failures, and (2) on-line verification tests. More false alarms will reflect a general increase in work levels throughout the model with the associated increase in total down times of the opera-

TABLE I

Average Systems Availability Changes due to Self-Test Accuracy

	SELF-TEST ACCURACY (%) 80	85	90	BASELINE	REGRESSION EQUATION	CORR. R%
System 1	.938	.940	.941	.945	.909 + .036x	98.41
System 2	.862	.877	.882	.882	.703 + .200x	92.30
System 3	.867	.869	.878	.878	.777 + .110x	88.10
System 4	.978	.986	.979	.978	.980 + .000x	0
System 5	.914	.919	.928	.910	.848 + .085x	87.73
System 6	.873	.876	.893	.902	.713 + .196x	94.28

ting systems. The average self-test accuracy of the baseline systems is ninety percent. The TSMT of this data is 265,059.9 minutes. Table 2 shows the change associated with self-test accuracy.

TABLE 2

Total System Maintenance Time
Changes due to Self-Test Accuracy

SELF-TEST ACCURACY (%)	TSMT (MIN.)	% INCREASE
BASELINE	265,059.90	-
85	279,177.73	5.33
80	299,609.18	13.03

The actual change from the baseline to the eighty percent level is a little less than 576 man hours, or just over one-quarter standard man year.

Note that this is direct active labor and does not reflect the actual manloading due to administrative, paperwork, safety, and other allied duties. These latter activities can often quadruple the direct labor load. This manloading will then need to be evaluated against the costs of designing self-test accuracy into the operating systems to ascertain true benefits.

Another decision variable that was not manipulated during the basic research was associated with Block R1 of Figure 3. For the baseline data, this variable actually allowed approximately one-third of the total failures into the repair shop. This decision variable depends heavily upon maintenance policy as to what repair level capability will be implemented in a particular repair facility. The remaining failures were assumed to be processed to more sophisticated facilities and the systems restored. The subsequent impact on workload levels in the repair shop is then shown in Table 3. This table shows the increasing workloads as Self-test accuracy decreases.

TABLE 3

Repair Shop Workload Changes
Due to Self-Test Accuracy

SELF-TEST ACCURACY (%)	QUANTITIES PROCESSED	
	PERFORMANCE TEST	DIAGNOSTIC TEST
Baseline	1124	584
85	1131	602
80	1191	617

In addition to these results, the percentage of recycled, or "second-looks" for the original failure ranged from 8.2 percent for System 4 to 32.9 percent for System 6 when the accuracy level was eighty percent. At the ninety percent level, these recycles dropped to 6.1 percent for System 4 and 9.9 percent for System 6 as measured against the initial number of failure alerts.

The difference in total expended time between the baseline and eighty percent accuracy level for the Table 3 quantities processed was about 102 man hours for the Performance Tests and about forty man hours for the Diagnostic Tests. Coupled with the systems time yields about 718 man hours of extra direct labor.

Spares Level Probability

The anticipated impact of lower spares levels would be reflected in a larger quantity of failed items going into the repair shop without replacement spares and therefore, expected longer system down times. This would apparently result in lower system availabilities. The baseline percent values for the systems varied widely as follows: System 1 - 80, System 2 - 60, System 3 - 50, System 4 - 40, System 5 - 75, System 6 - 94. From the baseline, the spares probability level was uniformly set for all systems at 75, 80, 85, and 90 percent respectively. The impact on systems availability is shown in Table 4.

The resulting impact on operating system availability shown by Table 4 was somewhat unexpected. There is little response to changes in the spares probability as shown by the small coefficient of X in the regression equations and in fact some systems even displayed a negative response. Further thought will recall that the random failure pattern

TABLE 4

Average Systems Availability Changes
due to Spares Level

	SPARES LEVEL % BASELINE	75	80	85	90	REGRESSION EQUATION	CORR. R%
System 1	.949	.941	.949	.945	.954	.889 + .070x	66.03
System 2	.882	.881	.881	.876	.867	.909 - .041x	58.62
System 3	.878	.898	.886	.896	.892	.861 + .038x	53.45
System 4	.978	.982	.981	.982	.961	.986 - .013x	8.37
System 5	.910	.910	.907	.892	.909	.934 - .036x	7.60
System 6	.902	.900	.893	.907	.899	.882 + .020x	9.79

is independent of the spares supply and appears to be a much stronger driver of availability, at least over the spares range examined. It is recognized that in a finite spare supply, these same results would not be obtained due to discontinuities caused by periodic running out of spares.

No significant changes in the work load was found at any point in the repair shop. This is a logical conclusion since the repair shop will process at least one failed item for each system failure regardless of the spares situation. In fact, one of the few items of interest found with this variable was the quantity of failed items entering the repair shop via the LOCOFF block, that is, no spare available. This is shown in Table 5.

TABLE 5

Failures Entering Repair Shop
With No Spares

SPARES LEVEL (%)	QUANTITY	% DECREASE
75	473	-
80	416	12.05
85	300	36.58
90	204	56.76

The anticipated longer system down times are reflected in the TSMT shown by Table 6.

TABLE 6

Total System Maintenance Time
Changes due to Spares Level

SPARES LEVEL (%)	TSMT (MIN.)	% CHANGE
BASELINE (AVE = 67)	265,059.90	-
80	267,175.80	+0.80
85	264,056.74	-0.38
90	263,221.13	-0.69

Here again, the system randomness seems to be the more powerful influence with less than a one percent savings in manpower (about 31 man hours) from the baseline level to the ninety percent level of sparing.

Performance Verification Accuracy

The Performance Verification Test is basically a GO, NO GO decision test with the intent to make rapid, gross level decisions. The baseline data operated on a 70/30 decision capability, That is, failed material was properly detected 70 percent of the time and good material was claimed to be bad 30 percent of the time. Other levels tested were 80/20 and 90/10. It was not known whether this parameter would be significant since failed material would arrive at the repair shop with a probability of being truly defective essentially controlled by the self-test accuracy. However, it could be expected that lower detection levels by the performance test should cause more on-line tests and subsequently lower system availabilities. The results of the exercise are shown in Table 7.

Varying this parameter results in a consistent response by all operating systems, however, first impression is that it is in the wrong direction. This variable is a prime candidate for further study, which is beyond the scope of this model exercise.

Due to the random failure pattern, there were 7.24 percent more failures during the 90/10 exercise year than experienced by the model for the 70/30 exercise year. This resulted in 5.22 percent more items being processed through the PVS block and could be the major cause of the negative response directions of the operating systems. Here again, it appears that the systems are more sensitive to the random failure pattern than they are to the range of deviations exercised in the model for the PVS variable.

The impact on manloading throughout the repair shop due to varying the Performance Test accuracy is shown in Table 8.

The total manloading increase for all three areas is partly attributable to the 7.24 percent increase in failures during the 90/10 simulation run. However, the DTS manloading increased by 10.79 percent which is partially

TABLE 7

Average Systems Availability Changes
due to PVS Accuracy

	PERFORMANCE ACCURACY (%)			REGRESSION EQUATION	CORR. R%
	70/30	80/20	90/10		
System 1	.945	.948	.942	.957 - .015x	25.00
System 2	.882	.882	.874	.911 - .040x	74.99
System 3	.878	.876	.869	.910 - .045x	90.67
System 4	.979	.978	.984	.997 - .025x	89.28
System 5	.910	.914	.910	.911 + .000x	0
System 6	.902	.901	.898	.916 - .020x	92.30

TABLE 8

Repair Shop Maintenance Time
Changes due to PVS Accuracy

PERFORMANCE ACCURACY (%)	PVS TIME (MIN.)	DTS TIME (MIN.)	REPAIR TIME (MINS.)
70/30	102,732.010	41,806.026	12,346.080
80/20	96,882.940	40,093.760	11,884.692
90/10	108,399.210	46,316.142	13,241.392

accounted for by the change in PVS accuracy. Interestingly, the Repair manloading increased by exactly 7.25 percent which simply implies that eventually failed material would be properly detected and subsequently repaired.

Diagnostic Test Accuracy

The Diagnostic Test is the most intricate troubleshooting procedure used in most repair facilities. Its purpose is to specifically isolate the exact failed component on a module, printed circuit board, or other assembly entering the repair shop. The baseline data was exercised at a 95/5 level, that is, the failed part would be properly isolated ninety-five percent of the time and improperly denoted five percent of the time. The impact on system performance is shown in Table 9.

As previously noted, Systems 1, 4, and 5 are the lower failure rate systems. These are showing the most sensitivity to the diagnostic accuracy. While system availabilities show general positive response (except for System 6) the actual changes are quite small.

The impact on manloading in the repair shop is shown in Table 10.

Manloading generally decreases in the repair shop at all points as a result of increased diagnostic accuracy. Manpower savings are 10.37 percent at the PVS, 16.93 percent at the DTS which is almost uniformly related to the fifteen

TABLE 9

Average Systems Availability Changes due to DTS Accuracy

	DIAGNOSTIC ACCURACY (%) 80/20	85/15	90/10	95/05	REGRESSION EQUATION	CORR. R%
System 1	.938	.947	.947	.945	.907 + .042x	40.27
System 2	.881	.867	.875	.882	.857 + .022x	4.23
System 3	.873	.872	.868	.878	.853 + .022x	11.92
System 4	.974	.971	.984	.978	.933 + .050x	32.98
System 5	.905	.908	.911	.910	.877 + .036x	77.14
System 6	.905	.900	.890	.902	.932 - .038x	14.24

TABLE 10

Repair Shop Maintenance Time Changes due to DTS Accuracy

DIAGNOSTIC ACCURACY (%)	PVS TIME (MIN)	DTS TIME (MIN)	REPAIR TIME (MIN)
80/20	114,614.350	50,324.958	12,997.186
85/15	118,453.100	51,541.920	13,404.196
90/10	106,297.040	45,528.696	12,807.248
95/05	102,732.010	41,806.026	12,346.080 — Repair

69 no/60 Time to process 556 modules

correspondence fig 3 page 20

percent change in the variable, and five percent in the actual Repair activity.

At the 80/20 accuracy level, the total repair shop man-loading is equivalent to 1.43 standard man-years of active, continuous effort. However, this in no way reflects the fact that many simultaneous efforts must be taking place, including removal and delivery of the failed material to the repair shop, administrative effort including paperwork and reports, and various other actions relating to the safety and general housekeeping. At the 95/05 level of accuracy, this direct labor reduces to about 1.26 standard man-years. The actual reduction is approximately 11.56 percent. Since these latter mentioned duties can easily quadruple the direct labor hours, it appears that the diagnostic accuracy level becomes an important variable to consider in future maintenance support concepts.

V. CONCLUSIONS

This research project presents the development of a simulation model which was used in the production of statistical data for use in analyzing the impact of decision parameter changes on system performance and workloads at a Navy field site.

The collection of data and formulation of a work flow structure to represent a Navy field site consisting of a repair shop in support of several operating systems was accomplished and is represented by Figures 1 through 5. The survey data was reduced to provide significant data elements representing branching decisions, delay times and failure rates in the flow diagram.

Detailed operating instructions applicable to the work flow were developed and basic assumptions established for the simulation model. This was followed by specific programming instructions to formulate the computer based simulation model and application of programming techniques to resolve the actual work flow requirements. The model was then run utilizing the TRACE capabilities for debugging. Data output from the model was compared to historical information to verify correct operations. For example, the total failures represented by the baseline simulation run for one year was 2129. This compares with the historical data of 2239, a difference of only about five percent.

The decision parameters that were exercised in the model were; (1) Built-in-Test (BIT) or self-test accuracy, (2) probability of having spare parts, (3) performance test accuracy, and (4) diagnostic test accuracy. Each of these parameters represent some form of maintenance support that might be altered due to the introduction of ATE into a Navy repair facility in support of training equipment. A set of statistical data similar to Appendices C and D was obtained for each change in the decision variables. These data were analyzed by various statistical methods and presented in Tables 1-10.

While specific systems tended to respond in accordance with their inherent failure rate, some general conclusions may be drawn by summarizing the regression equations of each system against the decision variable. This information is show in Table 11.

TABLE 11

Average Response by Decision Variable

VARIABLE	AVERAGE RESPONSE	RANGE OF X
Self-Test	.822 + .105x	.80 - .90
Spares Level	.910 + .006x	.67 - .90
Performance	.934 - .024x	.70 - .90
Diagnostic	.893 + .022x	.80 - .95

Table 11 indicates that Self-Test Accuracy is more influential where system availability is concerned. It should be noted from Table 1, and from the analyses in Appendices F and G, that high failure rate systems are more responsive to this variable which should lead to a trade-off or optimization study between inherent failure rates and self-test proficiency.

The poor response of the Performance Test Accuracy may be partially due to the specific failure experience during the simulation runs. It would, however, indicate an inability to make a major contribution to system availability and is easily overpowered by the failure rate of the system. Even observing the change from the seventy percent level to the eighty percent level in Table 7 shows insignificant changes in the individual system availabilities except for maybe System 5.

Workload levels are summarized over specific variable ranges in Table 12. The variable and its range are denoted in the first column while the percent change in maintenance times are recorded in column two. These maintenance times represent time expended at the operating system for the Self-Test and Spares variables and changes in the repair shop for the Performance and Diagnostic variables. The last column indicates a rate of change of maintenance time per unit change in the decision variable in order to observe the maximum contributor to labor changes. The actual direct manpower for the baseline model is 2.12 man-years for the operating systems plus 1.26 for the repair shop. Collateral duties can easily quadruple this value.

TABLE 12

Labor Variation due to Decision Variable Change

VARIABLE/RANGE	LABOR CHANGE	RATE OF CHANGE
SELF-TEST/.80-.90	-11.53%	1.15
SPARES/.80-.90	- 1.48%	0.15
PERFORMANCE/.70-.80	- 5.10%	0.51
DIAGNOSTIC/.80-.95	-11.56%	0.77

Table 12 shows the Self-Test parameter to also have the most impact on workloads or time expended as a result of failure alerts. These summarizations then show that, of the decision variables exercised, the Self-Test Accuracy is the most important in terms of impact on system performance and resulting workloads. It should be noted that the results presented in this study represent only the four quarterly variations for each decision variable value. If a complete statistical significance analysis were to be developed concerning the output of data, several replications of the model runs would be required with possible changes in the seed, or random variable generator, for each replication. However, the summaries presented herein do lead to the appropriate conclusions concerning which variables are most sensitive and which systems are most responsive to decision variable changes. The actual cost trade-offs between equipment design and personnel salaries would then have to be evaluated by the design and logistics teams in order to ascertain optimum benefits.

APPENDIX A

PROGRAM LISTING

REPAIR FLOW MODEL

```
        *WORK FLOW MODEL FOR A REPAIR SHOP WITH SEVERAL INPUT SOURCES
        *JAMES T. NEWELL PROJECT PAPER
                 REALLOCATE 200K
00001            GENERATE  2400,FN$EXPON,5,,,7,F      S1
00002            ADVANCE   0
00003            ASSIGN    4,24                     SYSTEM BLOCK I.D.
00004   SYSTEM1  QUEUE     SYSTEM1                  2F87F
00005            GATE NU   SYSTEM1
00006            SEIZE     SYSTEM1
00007            MARK      7
00008            SAVEVALUE 5,P7,F
00009            DEPART    SYSTEM1
00010   REENTER  TRANSFER  .260,OTHER1,HICOMP
00011   OTHER1   TRANSFER  .608,LOCOMP,MECOMP
00012   LOCOMP   ASSIGN    1,1                      LO COMPLEXITY I.D.(S2)
00013            TRANSFER  ,SELFTEST
00014   MECOMP   ASSIGN    1,2                      MED COMPLEXITY I.D.(S2)
00015            TRANSFER  ,SELFTEST
00016   HICOMP   ASSIGN    1,3                      HI COMPLEXITY I.D.(S2)
00017   SELFTEST ADVANCE   60,12                    S3
00018            TRANSFER  .990,NONFAIL,FAILURE
00019   NONFAIL  ASSIGN    2,0                      NONFAILURE I.D.(S4)
00020            TRANSFER  ,SPARE
00021   FAILURE  ASSIGN    2,1                      FAILURE I.D.(S4)
00022   SPARE    TRANSFER  .200,VERIFY,LOCOFF       S5
00023   VERIFY   ASSIGN    6,6                      ONLINE TESTBLOCK I.D.
00024   ONLINE   QUEUE     ONLINE
00025            TRANSFER  P,6,20
00026            TRANSFER  ,CONTINUE                TO S6
00027            TRANSFER  ,CHECKON                 TO S7
00028            TRANSFER  ,CHECKON                 TO S8
00029   RECYCLE  DEPART    P6
00030            ADVANCE   10
00031            TRANSFER  ,WAITING
00032   CHECKON  ASSIGN    6+,40
00033   WAITING  QUEUE     P6                       ONLINE VERIFY S7 - S8
00034            GATE NU   SYSTEM1,COMPARE
00035            SEIZE     SYSTEM1
00036            ASSIGN    7,1
00037            TRANSFER  ,ADVANCE
00038   COMPARE  TEST E    X5,P7,RECYCLE
00039   ADVANCE  ADVANCE   30,6
00040            TEST NE   X5,P7,JUMP
00041            RELEASE   SYSTEM1
00042   JUMP     DEPART    P6
00043            ASSIGN    6-,40
00044            DEPART    ONLINE
00045            TRANSFER  P,2,46                   BLOCK XFER NUMBER
00046            TRANSFER  ,RFI
00047            TRANSFER  P,6,41                   BLOCK XFER NUMBER
```

REPAIR FLOW MODEL

```
00048           TRANSFER  ,OFFLINE
00049           TRANSFER  ,MASTOFF
00050 CONTINUE  ADVANCE   30,6                    S6
00051           TRANSFER  P,2,52
00052           TRANSFER  ,RETEST
00053           ADVANCE   0
00054 UPSTAT1   SPLIT     1,OFFLINE
00055           DEPART    ONLINE
00056           RELEASE   SYSTEM1
00057           TRANSFER  ,CHECK
00058 RETEST    PRIORITY  1
00059           DEPART    ONLINE
00060           TRANSFER  ,REENTER
      *START OF REPAIR SHOP SUBROUTINE
00061 LOCOFF    ASSIGN    5,1                     OFFLINE ROUTINE
00062 OFFLINE   PRIORITY  0
00063           TRANSFER  P,1,63                  BLOCK XFER NUMBER
00064           TRANSFER  .750,RETAIN,MASTOFF     R1
00065           TRANSFER  .500,RETAIN,MASTOFF     R1
00066           TRANSFER  .250,RETAIN,MASTOFF     R1
00067 RETAIN    QUEUE     RETAIN
00068           ASSIGN    6,7                     ONLINE TESTBLOCK I.D.
00069 SECOND    TRANSFER  SBR,PERFVER,3           R2
00070           DEPART    PERFVER
00071           DEPART    RETAIN
00072           TRANSFER  ,*4
00073           DEPART    PERFVER
00074           TRANSFER  SBR,DIAGNOST,3          R3
00075           DEPART    DIAGNOST
00076           PRIORITY  1
00077           TRANSFER  ,SECOND
00078           DEPART    DIAGNOST
00079           TRANSFER  P,6,73                  BLOCK XFER NUMBER
00080           PRIORITY  0
00081 REPAIR    QUEUE     REPAIR
00082           TRANSFER  P,1,82                  R4 BLOCK XFER NO.
00083           TRANSFER  ,FIX1
00084           TRANSFER  ,FIX2
00085           TRANSFER  ,FIX3
00086 FIX1      ADVANCE   22,5
00087           TRANSFER  ,TAG4
00088 FIX2      ADVANCE   32,6
00089           TRANSFER  ,TAG4
00090 FIX3      ADVANCE   32,6
00091 TAG4      ASSIGN    2,0
00092           DEPART    REPAIR
00093           ASSIGN    6,8                     ONLINE TESTBLOCK I.D.
00094           TRANSFER  SBR,PERFVER,3           R5
00095           DEPART    PERFVER
00096           DEPART    RETAIN
```

REPAIR FLOW MODEL

```
00097             TRANSFER  ,*4
00098             DEPART    PERFVER
00099             PRIORITY  2
00100             TRANSFER  SBR,DIAGNOST,3          R3
00101             DEPART    DIAGNOST
00102             DEPART    RETAIN
00103             TRANSFER  ,CHECK
00104             DEPART    DIAGNOST
00105             TRANSFER  ,REPAIR
00106   PERFVER   QUEUE     PERFVER                 PERFORM. VERIF. ROUT.
00107             TRANSFER  P,1,107                 P1  BLOCK XFER NUMBER
00108             TRANSFER  ,DELAY1
00109             TRANSFER  ,DELAY2
00110             TRANSFER  ,DELAY3
00111   DELAY1    ADVANCE   60,6
00112             TRANSFER  ,TAG2
00113   DELAY2    ADVANCE   103,21
00114             TRANSFER  ,TAG2
00115   DELAY3    ADVANCE   162,32
00116   TAG2      TEST E    PR,0,TAG1
00117             TRANSFER  P,2,118                  P2 BLOCK XFER NUMBER
00118   BRAN1     TRANSFER  .700,BRANCHB,BRANCHC     P4
00119   BRAN2     TRANSFER  .300,BRANCHB,BRANCHC     P3
00120   TAG1      TEST E    PR,1,BRANCHC
00121   BRANCHB   TRANSFER  P,3,4                    (PRIORITIES 0/1)
00122   BRANCHC   TRANSFER  P,3,1                    (PRIORITIES 0/2)
00123   DIAGNOST  QUEUE     DIAGNOST                 DIAGNOSTIC ROUTINE
00124             TRANSFER  P,1,124                  D1 BLOCK XFER NUMBER
00125             TRANSFER  ,WORK1
00126             TRANSFER  ,WORK2
00127             TRANSFER  ,WORK3
00128   WORK1     ADVANCE   47,9
00129             TRANSFER  ,TAG3
00130   WORK2     ADVANCE   89,13
00131             TRANSFER  ,TAG3
00132   WORK3     ADVANCE   103,21
00133   TAG3      TRANSFER  .200,RETAINB,ROUTED      D2
00134   RETAINB   TEST E    PR,0,ROUTEB
00135             TRANSFER  P,2,136                  D3 BLOCK XFER NUMBER
00136             TRANSFER  .950,ROUTEB,ROUTEC       D5
00137             TRANSFER  .050,ROUTEB,ROUTEC       D4
00138   ROUTEB    TRANSFER  P,3,4                    (PRIORITIES 0/1/2)
00139   ROUTEC    TRANSFER  P,3,1                    (PRIORITY 0)
00140   ROUTED    DEPART    DIAGNOST
00141             DEPART    RETAIN
        *END OF REPAIR SHOP SUBROUTINE
00142   MASTOFF   PRIORITY  0                         SHIP BACK TO MFG.
00143             QUEUE     MASTOFF
        *ASSUME REPLACEMENT PART OBTAINED FOR THIS MODEL PURPOSE
00144             DEPART    MASTOFF
```

REPAIR FLOW MODEL

```
00145             TRANSFER  P,5,146                   BLOCK XFER NUMBER
00146             TRANSFER  ,CHECK
00147             TRANSFER  P,4,32                    BLOCK XFER NUMBER
00148   RPI       TRANSFER  P,5,149                   BLOCK XFER NUMBER
00149             TRANSFER  ,CHECK
00150             TRANSFER  P,4,32                    BLOCK XFER NUMBER
00151   CHECK     TEST LE   C1,131490,OFF
00152             TERMINATE 0
00153             GENERATE  1300,FN$EXPON,5,,,7,F         S1
00154             ADVANCE   0
00155             ASSIGN    4,176                 SYSTEM BLOCK I.D.
00156   SYSTEM2   QUEUE     SYSTEM2               2P87T
00157             GATE NU   SYSTEM2
00158             SEIZE     SYSTEM2
00159             MARK      7
00160             SAVEVALUE 6,P7,F
00161             DEPART    SYSTEM2
00162   REENTER2  TRANSFER  .030,OTHER2,HICOM2
00163   OTHER2    TRANSFER  .165,LOCOM2,MECOM2
00164   LOCOM2    ASSIGN    1,1                   LO COMPLEXITY I.D.(S2)
00165             TRANSFER  ,SELFTES2
00166   MECOM2    ASSIGN    1,2                   MED COMPLEXITY I.D.(S2)
00167             TRANSFER  ,SELFTES2
00168   HICOM2    ASSIGN    1,3                   HI COMPLEXITY I.D.(S2)
00169   SELFTES2  ADVANCE   115,23                S3
00170             TRANSFER  .900,NOFAIL2,FAIL2
00171   NOFAIL2   ASSIGN    2,0                   NONFAILURE I.D.(S4)
00172             TRANSFER  ,SPAR2
00173   FAIL2     ASSIGN    2,1                   FAILURE I.D.(S4)
00174   SPAR2     TRANSFER  .400,VERIF2,LOCOFF    S5
00175   VERIF2    ASSIGN    6,6                   ONLINE TESTBLOCK I.D.
00176   ONLIN2    QUEUE     ONLIN2
00177             TRANSFER  P,6,172               BLOCK XFER NUMBER
00178             TRANSFER  ,CONTIN2              TO S6
00179             TRANSFER  ,CHECKO2              TO S7
00180             TRANSFER  ,CHECKO2              TO S8
00181   RECYCL2   DEPART    P6
00182             ADVANCE   10
00183             TRANSFER  ,WAITIN2
00184   CHECKO2   ASSIGN    6+,50
00185   WAITIN2   QUEUE     P6                    ONLINE VERIFY S7 - S8
00186             GATE NU   SYSTEM2,COMPAR2
00187             SEIZE     SYSTEM2
00188             ASSIGN    7,1
00189             TRANSFER  ,ADVANC2
00190   COMPAR2   TEST E    X6,P7,RECYCL2
00191   ADVANC2   ADVANCE   30,6
00192             TEST NE   X6,P7,JUMP2
00193             RELEASE   SYSTEM2
00194   JUMP2     DEPART    P6
```

```
00195           ASSIGN    6-,50
00196           DEPART    ONLIN2
00197           TRANSFER  P,2,198
00198           TRANSFER  ,RFI
00199           TRANSFER  P,6,193               BLOCK XFER NUMBER
00200           TRANSFER  ,OFFLINE
00201           TRANSFER  ,MASTOFF
00202  CONTIN2  ADVANCE   30,6                  S6
00203           TRANSFER  P,2,204
00204           TRANSFER  ,RETES2
00205           ADVANCE   0
00206  UPSTAT2  SPLIT     1,OFFLINE
00207           DEPART    ONLIN2
00208           RELEASE   SYSTEM2
00209           TRANSFER  ,CHECK
00210  RETES2   PRIORITY  1
00211           DEPART    ONLIN2
00212           TRANSFER  ,REENTER2
00213           GENERATE  780,FN$EXPON,5,,,7,F       S1
00214           ADVANCE   0
00215           ASSIGN    4,236                SYSTEM BLOCK I.D.
00216  SYSTEM3  QUEUE     SYSTEM3              14B40
00217           GATE NU   SYSTEM3
00218           SEIZE     SYSTEM3
00219           MARK      7
00220           SAVEVALUE 7,P7,F
00221           DEPART    SYSTEM3
00222  REENTER3 TRANSFER  .040,OTHER3,HICOM3
00223  OTHER3   TRANSFER  .312,LOCOM3,MECOM3
00224  LOCOM3   ASSIGN    1,1                  LO COMPLEXITY I.D.(S2)
00225           TRANSFER  ,SELFTES3
00226  MECOM3   ASSIGN    1,2                  MED COMPLEXITY I.D.(S2)
00227           TRANSFER  ,SELFTES3
00228  HICOM3   ASSIGN    1,3                  HI COMPLEXITY I.D.(S2)
00229  SELFTES3 ADVANCE   30,6                 S3
00230           TRANSFER  .900,NOFAIL3,FAIL3
00231  NOFAIL3  ASSIGN    2,0                  NONFAILURE I.D.(S4)
00232           TRANSFER  ,SPAR3
00233  FAIL3    ASSIGN    2,1                  FAILURE I.D.(S4)
00234  SPAR3    TRANSFER  .500,VERIF3,LOCOFF   S5
00235  VERIF3   ASSIGN    6,6                  ONLINE TESTBLOCK I.D.
00236  ONLIN3   QUEUE     ONLIN3
00237           TRANSFER  P,6,232
00238           TRANSFER  ,CONTIN3             TO S6
00239           TRANSFER  ,CHECK03             TO S7
00240           TRANSFER  ,CHECK03             TO S8
00241  RECYCL3  DEPART    P6
00242           ADVANCE   10
00243           TRANSFER  ,WAITIN3
00244  CHECK03  ASSIGN    6+,60
```

REPAIR FLOW MODEL

```
00245   WAITIN3   QUEUE     P6                           ONLINE VERIFY S7 - S8
00246             GATE NU   SYSTEM3,COMPAR3
00247             SEIZE     SYSTEM3
00248             ASSIGN    7,1
00249             TRANSFER  ,ADVANC3
00250   COMPAR3   TEST E    X7,P7,RECYCL3
00251   ADVANC3   ADVANCE   30,6
00252             TEST NE   X7,P7,JUMP3
00253             RELEASE   SYSTEM3
00254   JUMP3     DEPART    P6
00255             ASSIGN    6-,60
00256             DEPART    ONLIN3
00257             TRANSFER  P,2,258
00258             TRANSFER  ,RFI
00259             TRANSFER  P,6,253                      BLOCK XFER NUMBER
00260             TRANSFER  ,OFFLINE
00261             TRANSFER  ,MASTOFF
00262   CONTIN3   ADVANCE   30,6                         S6
00263             TRANSFER  P,2,264
00264             TRANSFER  ,RETES3
00265             ADVANCE   0
00266   UPSTAT3   SPLIT     1,OFFLINE
00267             DEPART    ONLIN3
00268             RELEASE   SYSTEM3
00269             TRANSFER  ,CHECK
00270   RETES3    PRIORITY  1
00271             DEPART    ONLIN3
00272             TRANSFER  ,REENTER3
00273             GENERATE  5880,FN$EXPON,5,,,7,F         S1
00274             ADVANCE   0
00275             ASSIGN    4,296                        SYSTEM BLOCK I.D.
00276   SYSTEM4   QUEUE     SYSTEM4                      14B44
00277             GATE NU   SYSTEM4
00278             SEIZE     SYSTEM4
00279             MARK      7
00280             SAVEVALUE 8,P7,F
00281             DEPART    SYSTEM4
00282   REENTER4  TRANSFER  .050,OTHER4,HICOM4
00283   OTHER4    TRANSFER  .305,LOCOM4,MECOM4
00284   LOCOM4    ASSIGN    1,1                          LO COMPLEXITY I.D.(S2)
00285             TRANSFER  ,SELFTES4
00286   MECOM4    ASSIGN    1,2                          MED COMPLEXITY I.D.(S2)
00287             TRANSFER  ,SELFTES4
00288   HICOM4    ASSIGN    1,3                          HI COMPLEXITY I.D.(S2)
00289   SELFTES4  ADVANCE   48,9                         S3
00290             TRANSFER  .900,NOFAIL4,FAIL4
00291   NOFAIL4   ASSIGN    2,0                          NONFAILURE I.D.(S4)
00292             TRANSFER  ,SPAR4
00293   FAIL4     ASSIGN    2,1                          FAILURE I.D.(S4)
00294   SPAR4     TRANSFER  .600,VERIF4,LOCOFF           S5
```

REPAIR FLOW MODEL

```
00295  VERIF4   ASSIGN    6,6                      ONLINE TESTBLOCK I.D.
00296  ONLIN4   QUEUE     ONLIN4
00297           TRANSFER  P,6,292
00298           TRANSFER  ,CONTIN4                 TO S6
00299           TRANSFER  ,CHECK04                 TO S7
00300           TRANSFER  ,CHECK04                 TO S8
00301  RECYCL4  DEPART    P6
00302           ADVANCE   10
00303           TRANSFER  ,WAITIN4
00304  CHECK04  ASSIGN    6+,70
00305  WAITIN4  QUEUE     P6                       ONLINE VERIFY S7 - S8
00306           GATE NU   SYSTEM4,COMPAR4
00307           SEIZE     SYSTEM4
00308           ASSIGN    7,1
00309           TRANSFER  ,ADVANC4
00310  COMPAR4  TEST E    X8,P7,RECYCL4
00311  ADVANC4  ADVANCE   30,6
00312           TEST NE   X8,P7,JUMP4
00313           RELEASE   SYSTEM4
00314  JUMP4    DEPART    P6
00315           ASSIGN    6-,70
00316           DEPART    ONLIN4
00317           TRANSFER  P,2,318
00318           TRANSFER  ,RFI
00319           TRANSFER  P,6,313                  BLOCK XFER NUMBER
00320           TRANSFER  ,OFFLINE
00321           TRANSFER  ,MASTOFF
00322  CONTIN4  ADVANCE   30,6                     S6
00323           TRANSFER  P,2,324
00324           TRANSFER  ,RETES4
00325           ADVANCE   0
00326  UPSTAT4  SPLIT     1,OFFLINE
00327           DEPART    ONLIN4
00328           RELEASE   SYSTEM4
00329           TRANSFER  ,CHECK
00330  RETES4   PRIORITY  1
00331           DEPART    ONLIN4
00332           TRANSFER  ,REENTER4
00333           GENERATE  3600,FN$EXPON,5,,,7,F        S1
00334           ADVANCE   0
00335           ASSIGN    4,356                    SYSTEM BLOCK I.D.
00336  SYSTEM5  QUEUE     SYSTEM5                  2C44
00337           GATE NU   SYSTEM5
00338           SEIZE     SYSTEM5
00339           MARK      7
00340           SAVEVALUE 9,P7,F
00341           DEPART    SYSTEM5
00342  REENTER5 TRANSFER  .010,OTHER5,HICOM5
00343  OTHER5   TRANSFER  .111,LOCOM5,MECOM5
00344  LOCOM5   ASSIGN    1,1                      LO COMPLEXITY I.D.(S2)
```

REPAIR FLOW MODEL

```
00345                   TRANSFER  ,SELFTES5
00346   MECOM5          ASSIGN    1,2                       MED COMPLEXITY I.D.(S2)
00347                   TRANSFER  ,SELFTES5
00348   HICOM5          ASSIGN    1,3                       HI COMPLEXITY I.D.(S2)
00349   SELFTES5        ADVANCE   192,38                    S3
00350                   TRANSFER  .700,NOFAIL5,FAIL5
00351   NOFAIL5         ASSIGN    2,0                       NONFAILURE I.D.(S4)
00352                   TRANSFER  ,SPAR5
00353   FAIL5           ASSIGN    2,1                       FAILURE I.D.(S4)
00354   SPAR5           TRANSFER  .250,VERIF5,LOCOFF        S5
00355   VERIF5          ASSIGN    6,6                       ONLINE TESTBLOCK I.D.
00356   ONLIN5          QUEUE     ONLIN5
00357                   TRANSFER  P,6,352
00358                   TRANSFER  ,CONTIN5                  TO S6
00359                   TRANSFER  ,CHECKO5                  TO S7
00360                   TRANSFER  ,CHECKO5                  TO S8
00361   RECYCL5         DEPART    P6
00362                   ADVANCE   10
00363                   TRANSFER  ,WAITIN5
00364   CHECKO5         ASSIGN    6:,80
00365   WAITIN5         QUEUE     P6                        ONLINE VERIFY S7 - S8
00366                   GATE NU   SYSTEM5,COMPAR5
00367                   SEIZE     SYSTEM5
00368                   ASSIGN    7,1
00369                   TRANSFER  ,ADVANC5
00370   COMPAR5         TEST E    X9,P7,RECYCL5
00371   ADVANC5         ADVANCE   60,12
00372                   TEST NE   X9,P7,JUMP5
00373                   RELEASE   SYSTEM5
00374   JUMP5           DEPART    P6
00375                   ASSIGN    6-,80
00376                   DEPART    ONLIN5
00377                   TRANSFER  P,2,378
00378                   TRANSFER  ,RFI
00379                   TRANSFER  P,6,373                   BLOCK XFER NUMBER
00380                   TRANSFER  ,OFFLINE
00381                   TRANSFER  ,MASTOFF
00382   CONTIN5         ADVANCE   60,12                     S6
00383                   TRANSFER  P,2,384
00384                   TRANSFER  ,RETES5
00385                   ADVANCE   0
00386   UPSTAT5         SPLIT     1,OFFLINE
00387                   DEPART    ONLIN5
00388                   RELEASE   SYSTEM5
00389                   TRANSFER  ,CHECK
00390   RETES5          PRIORITY  1
00391                   DEPART    ONLIN5
00392                   TRANSFER  ,REENTER5
00393                   GENERATE  1440,FN$EXPON,5,,,7,F         S1
00394                   ADVANCE   0
```

REPAIR FLOW MODEL

```
         ASSIGN    4,416                SYSTEM BLOCK I.D.
SYSTEM6  QUEUE     SYSTEM6              2F69D
         GATE NU   SYSTEM6
         SEIZE     SYSTEM6
         MARK      7
         SAVEVALUE 10,P7,F
         DEPART    SYSTEM6
REENTER6 TRANSFER  .040,OTHER6,HICOM6
OTHER6   TRANSFER  .135,LOCOM6,MECOM6
LOCOM6   ASSIGN    1,1                  LO COMPLEXITY I.D.(S2)
         TRANSFER  ,SELFTES6
MECOM6   ASSIGN    1,2                  MED COMPLEXITY I.D.(S2)
         TRANSFER  ,SELFTES6
HICOM6   ASSIGN    1,3                  HI COMPLEXITY I.D.(S2)
SELFTES6 ADVANCE   116,23               S3
         TRANSFER  .960,NOFAIL6,FAIL6
NOFAIL6  ASSIGN    2,0                  NONFAILURE I.D.(S4)
         TRANSFER  ,SPAR6
FAIL6    ASSIGN    2,1                  FAILURE I.D.(S4)
SPAR6    TRANSFER  .060,VERIF6,LOCOFF   S5
VERIF6   ASSIGN    6,6                  ONLINE TESTBLOCK I.D.
ONLIN6   QUEUE     ONLIN6
         TRANSFER  P,6,412
         TRANSFER  ,CONTIN6             TO S6
         TRANSFER  ,CHECK06             TO S7
         TRANSFER  ,CHECK06             TO S8
RECYCL6  DEPART    P6
         ADVANCE   10
         TRANSFER  ,WAITIN6
CHECK06  ASSIGN    6+,90
WAITIN6  QUEUE     P6                   ONLINE VERIFY S7 - S8
         GATE NU   SYSTEM6,COMPAR6
         SEIZE     SYSTEM6
         ASSIGN    7,1
         TRANSFER  ,ADVANC6
COMPAR6  TEST E    X10,P7,RECYCL6
ADVANC6  ADVANCE   18,4
         TEST NE   X10,P7,JUMP6
         RELEASE   SYSTEM6
JUMP6    DEPART    P6
         ASSIGN    6-,90
         DEPART    ONLIN6
         TRANSFER  P,2,438
         TRANSFER  ,RFI
         TRANSFER  P,6,433              BLOCK XFER NUMBER
         TRANSFER  ,OFFLINE
         TRANSFER  ,MASTOFF
CONTIN6  ADVANCE   18,4                 S6
         TRANSFER  P,2,444
         TRANSFER  ,RETES6
```

REPAIR FLOW MODEL

```
00445            ADVANCE    0
00446   UPSTAT6  SPLIT      1,OFFLINE
00447            DEPART     ONLIN6
00448            RELEASE    SYSTEM6
00449            TRANSFER   ,CHECK
00450   RETES6   PRIORITY   1
00451            DEPART     ONLIN6
00452            TRANSFER   ,REENTER6
00453   OFF      TERMINATE  1
        EXPON FUNCTION RN1,C23
        .0,.0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915/.7,1.2/
        .75,1.38/.8,1.6/.84,1.83/.88,2.12/.9,2.3/.93,2.52/.94,2.81/
        .95,2.99/.96,3.2/.97,3.5/.98,3.9/.99,4.6/.995,5.3/.998,6.2/
        .999,8.0
                 START 1,,1
                 RESET
                 START 1,,1
                 RESET
                 START 1,,1
                 RESET
                 START 1
                 END
        /*
```

APPENDIX B

BLOCK COUNTS

RELATIVE CLOCK TIME: 131822

ABSOLUTE CLOCK TIME: 131822

BLOCK COUNTS

BLOCK	CURR,	TOTAL	BLOCK	CURR,	TOTAL	BLOCK	CURR,	TOTAL	BLOCK	CURR,	TOTAL	BLOCK	CURR,	TOTAL
1	0	47	2	0	47	3	0	47	SYSTEM1	0	47	5	0	47
6	0	47	7	0	47	8	0	47	9	0	47	REENTER	0	47
OTHER1	0	36	LOCOMP	0	15	13	0	15	MECOMP	0	21	15	0	21
HICOMP	0	11	SELFTEST	0	47	18	0	47	NONFAIL	0	0	20	0	0
FAILURE	0	47	SPARE	0	47	VERIFY	0	33	ONLINE	0	54	25	0	54
26	0	33	27	0	9	28	0	12	RECYCLE	0	0	30	0	0
31	0	0	CHECKON	0	21	WAITING	0	21	34	0	21	35	0	17
36	0	17	37	0	17	COMPARE	0	4	ADVANCE	0	21	40	0	21
41	0	17	JUMP	0	21	43	0	21	44	0	21	45	0	21
46	0	12	47	0	9	48	0	9	49	0	0	CONTINUE	0	33
51	0	33	52	0	0	53	0	33	UPSTAT1	0	66	55	0	33
56	0	47	57	0	47	RETEST	0	0	59	0	0	60	0	0
LOCOFF	0	149	OFFLINE	0	469	63	0	469	64	0	306	65	0	131
66	0	32	RETAIN	0	156	68	0	156	SECOND	0	160	70	0	88
71	0	58	72	0	58	73	0	102	74	0	102	75	0	4
76	0	4	77	0	4	78	0	78	79	0	78	80	0	78
REPAIR	0	95	82	0	95	83	0	48	84	0	38	85	0	9
FIX1	0	48	87	0	48	FIX2	0	38	89	0	38	FIX3	0	9
TAG4	0	95	92	0	95	93	0	95	94	0	95	95	0	73
96	0	73	97	0	73	98	0	22	99	0	22	100	0	22
101	0	0	102	0	0	103	0	0	104	0	17	105	0	17
PERFVER	0	255	107	0	255	108	0	125	109	0	101	110	0	29
DELAY1	0	125	112	0	125	DELAY2	0	101	114	0	101	DELAY3	0	29
TAG2	0	255	117	0	234	BRAN1	0	81	BRAN2	0	153	TAG1	0	21
BRANCHB	0	124	BRANCHC	0	131	DIAGNOST	0	124	124	0	124	125	0	39
126	0	49	127	0	16	WORK1	0	59	129	0	59	WORK2	0	49

131	0	49	WORK3	0	16	TAG3	0	124	RETAIN3	0	99	139	0	79
136	0	0	137	0	79	ROUTER	0	95	ROUTEC	0	4	ROUTEM	0	25
141	0	25	MASTOFF	0	338	143	0	338	144	0	338	145	0	338
146	0	213	147	0	125	RFI	0	76	149	0	52	150	0	24
CHECK	0	680	152	0	679	153	0	87	154	0	87	155	0	87
SYSTEM2	0	87	157	0	87	158	0	87	159	0	87	160	0	87
161	0	87	REENTER2	0	93	OTHER2	0	90	LOCOM2	0	72	165	0	72
MECOM2	0	18	167	0	18	HICOM2	0	3	SELFTES2	1	92	170	0	92
NOFAIL2	0	11	172	0	11	FAIL2	0	81	SPAR2	0	92	VERIF2	0	46
ONLIN2	0	69	177	0	69	178	0	46	179	0	11	180	0	12
RECYCL2	0	9	182	0	9	183	0	9	CHECKU2	0	23	WAITIN2	0	32
186	0	32	187	0	12	188	0	12	189	0	12	COMPAR2	0	20
ADVANC2	0	23	192	0	23	193	0	12	JUMP2	0	23	195	0	23
196	0	23	197	0	23	198	0	13	199	0	10	200	0	10
201	0	0	CONTIN2	0	46	203	0	46	204	0	6	205	0	40
UPSTAT2	0	80	207	0	40	208	0	86	209	0	86	RETES2	0	6
211	0	6	212	0	6	213	0	146	214	0	146	215	0	146
SYSTEM3	0	146	217	0	146	218	0	146	219	0	146	220	0	146
221	0	146	REENTER3	0	152	OTHER3	0	147	LOCOM3	0	97	225	0	97
MECOM3	0	50	227	0	50	HICOM3	0	5	SELFTES3	0	152	230	0	152
NOFAIL3	0	17	232	0	17	FAIL3	0	135	SPAR3	0	152	VERIF3	0	85
ONLIN3	0	135	237	0	135	238	0	85	239	0	24	240	0	26
RECYCL3	0	77	242	0	77	243	0	77	CHECKU3	0	50	WAITIN3	0	127
246	0	127	247	0	31	248	0	31	249	0	31	COMPAR3	0	96
ADVANC3	0	50	252	0	50	253	0	31	JUMP3	0	50	255	0	50

CONTINUED

BLOCK COUNTS

BLOCK	CURR,	TOTAL	BLOCK	CURR,	TOTAL	BLOCK	CURR,	TOTAL	BLOCK	CURR,	TOTAL	BLOCK	CURR,	TOTAL
256	0	50	257	0	50	258	0	28	259	0	22	260	0	22
261	0	0	CONTIN3	0	85	263	0	85	264	0	6	265	0	79
UPSTAT3	0	158	267	0	79	268	0	146	269	0	146	RETES3	0	6
271	0	6	272	0	6	273	0	25	274	0	25	275	0	25
SYSTEM4	0	25	277	0	25	278	0	25	279	0	25	280	0	25
281	0	25	REENTER4	0	26	OTHER4	0	24	LOCOM4	0	13	285	0	13
MECOM4	0	11	287	0	11	HICOM4	0	2	SELFTES4	0	26	290	0	26
NOFAIL4	0	1	292	0	1	FAIL4	0	25	SPAR4	0	26	VERIF4	0	13
ONLIN4	0	20	297	0	20	298	0	13	299	0	2	300	0	5
RECYCL4	0	0	302	0	0	303	0	0	CHECKU4	0	7	WAITIN4	0	7
306	0	7	307	0	3	308	0	3	309	0	3	COMPAR4	0	4
ADVANC4	0	7	312	0	7	313	0	3	JUMP4	0	7	315	0	7
316	0	7	317	0	7	318	0	5	319	0	2	320	0	2
321	0	0	CONTIN4	0	13	323	0	13	324	0	1	325	0	12
UPSTAT4	0	24	327	0	12	328	0	25	329	0	25	RETES4	0	1
331	0	1	332	0	1	333	0	31	334	0	31	335	0	31
SYSTEM5	0	31	337	0	31	338	0	31	339	0	31	340	0	31
341	0	31	REENTER5	0	39	OTHER5	0	39	LOCOM5	0	33	345	0	33
MECOM5	0	6	347	0	6	HICOM5	0	0	SELFTES5	0	39	350	0	39
NOFAIL5	0	11	352	0	11	FAIL5	0	28	SPAR5	0	39	VERIF5	0	33
ONLIN5	0	40	357	0	40	358	0	33	359	0	4	360	0	3
RECYCL5	0	44	362	0	44	363	0	44	CHECKU5	0	7	WAITIN5	0	31
366	0	51	367	0	6	368	0	6	369	0	6	COMPAR5	0	45
ADVANC5	0	7	372	0	7	373	0	6	JUMP5	0	7	375	0	7
376	0	7	377	0	7	378	0	3	379	0	4	380	0	4
381	0	0	CONTIN5	0	33	383	0	33	384	0	8	385	0	25

UPSTAT3	0	30	387	0	25	388	0	31	389	0	31	RPTES3	0	8
391	0	8.	392	0	8	393	0	80	394	0	80	395	0	80
SYSTEM6	0	80	397	0	80	398	0	80	399	0	80	400	0	80
401	0	80	REENTER6	0	82	OTHER6	0	78	LOCOM6	0	69	405	0	69
MECOM6	0	9	407	0	9	HICOM6	0	4	SELFTES6	0	82	410	0	82
NOFAIL6	0	2	412	0	2	FAIL6	0	80	SPAR6	0	82	VERIF6	0	79
ONLIN6	0	108	417	0	102	418	0	79	419	0	8	420	0	15
RECYCL6	0	25	422	0	25	423	0	25	CHECKU6	0	23	WAITIN6	0	48
426	0	48	427	0	22	428	0	22	429	0	22	COMPARA	0	86
ADVANC6	0	23	432	0	23	433	0	22	JUMP6	0	23	435	0	83
436	0	23	437	0	23	438	0	15	439	0	8	440	0	8
441	0	0	CONTIN6	0	79	443	0	79	444	0	-2	445	0	77
UPSTAT6	1	153	447	0	77	448	0	80	449	0	80	RETES6	0	2
451	0	2	452	0	2	OFF	0	1						

RELATIVE CLOCK TIME: 131340

ABSOLUTE CLOCK TIME: 263362

BLOCK COUNTS

BLOCK	CURR,	TOTAL	BLOCK	CURR,	TOTAL	BLOCK	CURR,	TOTAL	BLOCK	CURR,	TOTAL	BLOCK	CURR,	TOTAL
1	0	58	2	0	58	3	0	58	SYSTEM1	0	58	5	0	58
6	0	58	7	0	58	8	0	58	9	0	58	REENTER	0	59
OTHER1	0	46	LOCMP	0	18	13	0	18	HECOMP	0	28	15	0	28
HICOMP	0	13	SELFTEST	0	59	18	0	59	NONFAIL	0	1	20	0	1
FAILURE	0	58	SPARE	0	59	VERIFY	0	46	ONLINE	0	66	25	0	66
26	0	46	27	0	3	28	0	15	RECYCLE	0	3	30	0	3
31	0	3	CHECKUN	0	20	WAITING	0	23	34	0	23	35	0	12
36	0	12	37	0	12	COMPARE	0	11	ADVANCE	0	20	40	0	20
41	0	12	JUMP	0	20	43	0	20	44	0	20	45	0	20
46	0	13	47	0	5	48	0	5	49	0	0	CONTINUE	0	46
51	0	46	52	0	1	53	0	45	UPSTAT1	0	90	55	0	45
56	0	58	57	0	58	RETEST	0	1	59	0	1	60	0	1
LOCOFF	0	168	OFFLINE	0	519	63	0	519	64	0	356	65	0	124
66	0	39	RETAIN	0	181	68	0	181	SECOND	0	189	70	0	62
71	0	62	72	0	62	73	0	127	74	0	127	75	0	8
76	0	8	77	0	8	78	0	97	79	0	97	80	0	97
REPAIR	0	123	82	0	123	83	0	53	84	0	51	85	0	21
FIX1	0	53	87	0	53	FIX2	0	51	89	0	51	FIX3	0	21
TAG4	0	123	92	0	123	93	0	123	94	0	123	95	0	88
96	0	88	97	0	88	98	0	37	99	0	37	100	0	37
101	0	0	102	0	0	103	0	0	104	0	28	105	0	28
PERFVER	0	314	107	0	314	108	0	146	109	0	115	110	0	33
DELAY1	0	146	112	0	146	DELAY2	0	115	114	0	115	DELAY3	0	33
TAG2	0	314	117	0	278	BRAN1	0	101	BRAN2	0	177	TAG1	0	36
BRANCHB	0	164	BRANCHC	0	150	DIAGNOST	0	164	124	0	164	125	0	76
126	0	62	127	0	26	WORK1	1	75	129	0	75	WORK2	0	62

131	0	68	WORK3	0	26	TAG3	0	163	RETAIN3	0	133	135	0	100
136	0	2	137	0	98	ROUTER	0	125	ROUTEC	0	8	ROUTEB	0	20
141	0	30	HASTOFF	0	368	143	0	368	144	0	368	145	0	368
146	0	229	147	0	139	RFI	0	89	149	0	60	150	0	29
CHECK	0	747	152	0	746	153	0	90	154	0	90	155	0	90
SYSTEM2	0	90	157	0	90	158	0	90	159	0	90	160	0	90
161	0	90	REENTER2	0	94	OTHER2	0	92	LOCOM2	0	81	165	0	81
HECOM2	0	11	167	0	11	HICOM2	0	2	SELFTES2	0	95	170	0	95
NOFAIL2	0	7	172	0	7	FAIL2	0	88	SPAR2	0	95	VERIF2	0	60
ONLIN2	0	86	177	0	86	178	0	60	179	0	14	180	0	12
RECYCL2	0	11	182	0	11	183	0	11	CHECKU2	0	26	WAITIN2	0	37
186	0	37	187	0	17	188	0	17	189	0	17	COMPAR2	0	20
ADVANC2	0	26	192	0	26	193	0	17	JUMP2	0	26	195	0	26
196	0	26	197	0	26	198	0	12	199	0	14	200	0	14
201	0	0	CONTIN2	0	60	203	0	60	204	0	4	205	0	56
UPSTAT2	0	112	207	0	56	208	0	91	209	0	91	RETES2	0	4
211	0	4	212	0	4	213	0	148	214	0	148	215	0	148
SYSTEM3	0	148	217	0	148	218	0	148	219	0	148	220	0	148
221	0	148	REENTER3	0	157	OTHER3	0	154	LOCOM3	0	110	225	0	110
HECOM3	0	44	227	0	44	HICOM3	0	3	SELFTES3	0	157	230	0	157
NOFAIL3	0	17	232	0	17	FAIL3	0	140	SPAR3	0	157	VERIF3	0	81
ONLIN3	0	133	237	0	133	238	0	81	239	0	18	240	0	24
RECYCL3	0	36	242	0	36	243	0	36	CHECKU3	0	52	WAITIN3	0	88
246	0	88	247	0	30	248	0	30	249	0	30	COMPAR3	0	88
ADVANC3	0	52	252	0	52	253	0	30	JUMP3	0	52	255	0	52

CONTINUED

BLOCK COUNTS

BLOCK	CURR,	TOTAL	BLOCK	CURR,	TOTAL	BLOCK	CURR,	TOTAL	BLOCK	CURR,	TOTAL	BLOCK	CURR,	TOTAL
256	0	52	257	0	52	258	0	34	259	0	18	260	0	18
261	0	0	CONTIN3	0	81	263	0	81	264	0	9	265	0	72
UPSTAT3	1	143	267	0	72	268	0	148	269	0	148	RETES3	0	9
271	0	9	272	0	9	273	0	28	274	0	28	275	0	28
SYSTEM4	0	28	277	0	28	278	0	28	279	0	28	280	0	28
281	0	28	REENTER4	0	31	OTHER4	0	29	LOCOM4	0	19	285	0	19
HECOM4	0	10	287	0	10	HICOM4	0	2	SELFTES4	0	31	290	0	31
NOFAIL4	0	4	292	0	4	FAIL4	0	27	SPAR4	0	31	VERIF4	0	8
ONLIN4	0	14	297	0	14	298	0	8	299	0	2	300	0	4
RECYCL4	0	0	302	0	0	303	0	0	CHECKU4	0	6	WAITIN4	0	6
306	0	6	307	0	1	308	0	1	309	0	1	COMPAR4	0	5
ADVANC4	0	6	312	0	6	313	0	1	JUMP4	0	6	315	0	6
316	0	6	317	0	6	318	0	4	319	0	2	320	0	2
321	0	0	CONTIN4	0	8	323	0	8	324	0	3	325	0	3
UPSTAT4	0	10	327	0	5	328	0	28	329	0	28	RETES4	0	3
331	0	3	332	0	3	333	0	34	334	0	34	335	0	34
SYSTEM5	0	34	337	0	34	338	0	34	339	0	34	340	0	34
341	0	34	REENTER5	0	48	OTHER5	0	48	LOCOM5	0	41	345	0	41
HECOM5	0	7	347	0	7	HICOM5	0	0	SELFTES5	0	48	350	0	48
NOFAIL5	0	19	352	0	19	FAIL5	0	29	SPAR5	0	48	VERIF5	0	34
ONLIN5	0	45	357	0	45	358	0	34	359	0	4	360	0	7
RECYCL5	0	2	362	0	2	363	0	2	CHECKU5	0	11	WAITIN5	0	13
366	0	13	367	0	8	368	0	8	369	0	8	COMPAR5	0	5
ADVANC5	0	11	372	0	11	373	0	8	JUMP5	0	11	375	0	11
376	0	11	377	0	11	378	0	8	379	0	3	380	0	3
381	0	0	CONTIN5	0	34	383	0	34	384	0	14	385	0	10

UPSTAT5	0	40	387	0	20	388	0	34	389	0	34	RRTES5	0	24
391	0	16	392	0	14	393	0	99	394	0	99	395	0	99
SYSTEM6	0	99	397	0	99	398	0	99	399	0	99	400	0	99
401	0	99	REENTER6	0	105	OTHER6	0	98	LOCOM6	0	86	405	0	86
MECOM6	0	12	407	0	12	HICOM6	0	7	SELFTES6	0	105	410	0	105
NOFAIL6	0	6	412	0	6	FAIL6	0	99	SPAR6	0	105	VERIF6	0	98
ONLIN6	0	133	417	0	133	418	0	98	419	0	19	420	0	16
RECYCL6	0	31	422	0	31	423	0	31	CHECKU6	0	35	WAITIN6	0	66
426	0	66	427	0	35	428	0	35	429	0	35	COMPAR6	0	31
ADVANC6	0	35	432	0	35	433	0	35	JUMP6	0	35	435	0	35
436	0	35	437	0	35	438	0	16	439	0	19	440	0	19
441	0	0	CONTIN6	0	98	443	0	98	444	0	6	445	0	92
UPSTAT6	0	188	447	0	92	448	0	99	449	0	99	RPTES6	0	6
451	0	6	452	0	6	OFF	0	1						

RELATIVE CLOCK TIME: 131534

ABSOLUTE CLOCK TIME: 394806

BLOCK COUNTS

BLOCK	CURR,	TOTAL	BLOCK	CURR,	TOTAL	BLOCK	CURR,	TOTAL	BLOCK	CURR,	TOTAL	BLOCK	CURR,	TOTAL
1	0	58	2	0	58	3	0	58	SYSTEM1	0	58	5	0	58
6	0	58	7	0	58	8	0	58	9	0	58	REENTER	0	58
OTHER1	0	43	LOCOMP	0	19	13	0	19	MECOMP	0	24	15	0	24
HICOMP	0	15	SELFTEST	0	58	18	0	58	NONFAIL	0	1	20	0	1
FAILURE	0	57	SPARE	0	58	VERIFY	0	45	ONLINE	0	72	25	0	72
26	0	45	27	0	12	28	0	15	RECYCLE	0	13	30	0	13
31	0	13	CHECKON	0	27	WAITING	0	40	34	0	40	35	0	23
36	0	23	37	0	23	COMPARE	0	17	ADVANCE	0	27	40	0	27
41	0	23	JUMP	0	27	43	0	27	44	0	27	45	0	27
46	0	16	47	0	11	48	0	11	49	0	0	CONTINUE	0	43
51	0	45	52	0	0	53	0	45	UPSTAT1	0	90	55	0	88
56	0	58	57	0	58	RETEST	0	0	59	0	0	60	0	0
LOCOFF	0	143	OFFLINE	0	495	63	0	495	64	0	337	65	0	127
66	0	31	RETAIN	0	158	68	0	158	SECOND	0	161	70	0	31
71	0	51	72	0	51	73	0	110	74	0	110	75	0	3
76	0	3	77	0	3	78	0	89	79	0	89	80	0	89
REPAIR	0	113	82	0	115	83	0	33	84	0	33	85	0	29
FIX1	0	58	87	0	53	FIX2	0	33	89	0	33	FIX3	0	29
TAG4	0	115	92	0	115	93	0	115	94	0	115	95	0	83
96	0	83	97	0	83	98	0	32	99	0	32	100	0	32
101	0	0	102	0	0	103	0	0	104	0	26	105	0	26
PERFVER	0	276	107	0	276	108	0	127	109	0	94	110	0	35
DELAY1	0	127	112	0	127	DELAY2	0	94	114	0	94	DELAY3	0	35
TAG2	0	276	117	0	247	BRAN1	0	97	BRAN2	0	150	TAG1	0	29
BRANCHB	0	142	BRANCHC	0	134	DIAGNOST	0	142	124	0	142	125	0	64
126	0	44	127	0	34	WORK1	0	65	129	0	65	WORK2	0	44

131	0	44	WORK3	0	34	TAG3	0	·143	RETAIN3	0	118	135	0	89
136	0	1	137	0	87	ROUTER	0	115	ROUTEC	0	3	ROUTEM	0	85
141	0	23	MASTOFF	0	362	143	0	362	144	0	362	145	0	362
146	0	258	147	0	104	RFI	0	89	149	0	48	150	0	41
CHECK	0	753	152	0	754	153	0	89	154	0	89	155	0	89
SYSTEM2	0	89	157	0	89	158	0	89	159	0	89	160	0	89
161	0	89	REENTER2	0	95	OTHER2	0	92	LOCOM2	0	73	165	0	73
MECOM2	0	19	167	0	19	HICOM2	0	3	SELFTES2	0	95	170	0	95
NOFAIL2	0	10	172	0	10	FAIL2	0	85	SPARE	0	95	VERIF2	0	68
ONLIN2	0	89	177	0	89	178	0	68	179	0	10	180	0	21
RECYCL2	0	13	182	0	13	183	0	13	CHECKU2	0	21	WAITIN2	0	34
186	0	36	187	0	16	188	0	16	189	0	16	COMPAR2	0	18
ADVANC2	0	21	192	0	21	193	0	16	JUMP2	0	21	195	0	21
196	0	21	197	0	21	198	0	12	199	0	9	200	0	9
201	0	0	CONTIN2	0	68	203	0	68	204	0	6	205	0	62
UPSTAT2	0	128	207	0	62	208	0	87	209	0	89	RETES2	0	6
211	0	6	212	0	6	213	0	153	214	0	153	215	0	153
SYSTEM3	0	153	217	0	153	218	0	153	219	0	153	220	0	153
221	0	153	REENTER3	0	155	OTHER3	0	148	LOCOM3	0	106	225	0	106
MECOM3	0	42	227	0	42	HICOM3	0	7	SELFTES3	0	155	230	0	155
NOFAIL3	0	8	232	0	8	FAIL3	0	147	SPAR3	0	155	VERIF3	0	78
ONLIN3	0	130	237	0	130	238	0	78	239	0	14	240	0	38
RECYCL3	0	20	242	0	20	243	0	20	CHECKU3	0	52	WAITIN3	0	72
246	0	72	247	0	21	248	0	21	249	0	21	COMPAR3	0	31
ADVANC3	0	52	252	0	52	253	0	21	JUMP3	0	52	255	0	52

CONTINUED

BLOCK COUNTS

BLOCK	CURR,	TOTAL	BLOCK	CURR,	TOTAL	BLOCK	CURR,	TOTAL	BLOCK	CURR,	TOTAL	BLOCK	CURR,	TOTAL
256	0	52	257	0	52	258	0	40	259	0	12	260	0	12
261	0	0	CONTIN3	0	78	263	0	78	264	0	2	265	0	76
UPSTAT3	0	153	267	0	76	268	0	153	269	0	153	RETES3	0	2
271	0	2	272	0	2	273	0	28	274	0	28	275	0	28
SYSTEM4	0	28	277	0	28	278	0	28	279	0	28	280	0	28
281	0	28	REENTER4	0	28	OTHER4	0	26	LOCOM4	0	14	285	0	14
MECOM4	0	12	287	0	12	HICOM4	0	2	SELFTES4	0	28	290	0	28
NOFAIL4	0	2	292	0	2	FAIL4	0	26	SPAR4	0	28	VERIF4	0	17
ONLIN4	0	27	297	0	27	298	0	17	299	0	7	300	0	3
RECYCL4	0	0	302	0	0	303	0	0	CHECKU4	0	10	WAITIN4	0	10
306	0	10	307	0	6	308	0	6	309	0	6	COMPAR4	0	4
ADVANC4	0	10	312	0	10	313	0	6	JUMP4	0	10	315	0	10
316	0	10	317	0	10	318	0	4	319	0	6	320	0	6
321	0	0	CONTIN4	0	17	323	0	17	324	0	0	325	0	17
UPSTAT4	0	36	327	0	17	328	0	28	329	0	28	RETES4	0	0
331	0	0	332	0	0	333	0	34	334	0	34	335	0	34
SYSTEM5	0	34	337	0	34	338	0	34	339	0	34	340	0	34
341	0	34	REENTER5	0	38	OTHER5	0	38	LOCOM5	0	36	345	0	36
MECOM5	0	2	347	0	2	HICOM5	0	0	SELFTES5	0	38	350	0	38
NOFAIL5	0	7	352	0	7	FAIL5	0	31	SPAR5	0	38	VERIF5	0	27
ONLIN5	0	33	357	0	33	358	0	27	359	0	5	360	0	3
RECYCL5	0	20	362	0	20	363	0	20	CHECKU5	0	8	WAITIN5	0	28
366	0	28	367	0	4	368	0	4	369	0	4	COMPAR5	0	24
ADVANC5	0	8	372	0	8	373	0	4	JUMP5	0	8	375	0	8
376	0	8	377	0	8	378	0	4	379	0	4	380	0	4
381	0	0	CONTIN5	0	27	383	0	27	384	0	4	385	0	23

UPSTAT5	0	46	387	0	23	388	0	26	389	0	34	RETES5	0	4
391	0	4	392	0	4	393	0	87	394	0	87	395	0	87
SYSTEM6	0	87	397	0	87	398	0	87	399	0	87	400	0	87
401	0	87	REENTER6	0	89	OTHER6	0	88	LOCOM6	0	78	405	0	78
MECOM6	0	10	407	0	10	HICOM6	0	1	SELFTES6	0	89	410	0	89
NOFAIL6	0	2	412	0	2	FAIL6	0	87	SPAR6	0	89	VERIF6	0	83
ONLIN6	0	99	417	0	99	418	0	83	419	0	3	420	0	13
RECYCL6	0	18	422	0	18	423	0	18	CHECKU6	0	16	WAITIN6	0	34
426	0	34	427	0	13	428	0	13	429	0	13	COMPAR6	0	21
ADVANC6	0	16	432	0	16	433	0	13	JUMP6	0	16	435	0	16
436	0	16	437	0	16	438	0	13	439	0	3	440	0	3
441	0	0	CONTIN6	0	83	443	0	83	444	0	2	445	0	81
UPSTAT6	0	162	447	0	81	448	0	87	449	0	87	RETES6	0	2
451	0	2	452	0	2	OFF	0	1						

RELATIVE CLOCK TIME! 131705

ABSOLUTE CLOCK TIME! 126601

BLOCK COUNTS

BLOCK	CURR;	TOTAL	BLOCK	CURR,	TOTAL	BLOCK	CURR,	TOTAL	BLOCK	CURR,	TOTAL	BLOCK	CURR;	TOTAL
1	0	43	2	0	43	3	0	43	SYSTEM1	0	43	5	0	43
6	0	43	7	0	43	8	0	43	9	0	43	REENTER	0	43
OTHER1	0	32	LOCOMP	0	12	13	0	12	MECOMP	0	20	15	0	20
HICOMP	0	11	SELFTEST	0	43	18	0	43	NONFAIL	0	0	20	0	0
FAILURE	0	43	SPARE	0	43	VERIFY	0	38	ONLINE	0	57	25	0	37
26	0	38	27	0	8	28	0	11	RECYCLE	0	0	30	0	0
31	0	0	CHECKON	0	19	WAITING	0	19	34	0	19	35	0	17
36	0	17	37	0	17	COMPARE	0	2	ADVANCE	0	19	40	0	19
41	0	17	JUMP	0	19	43	0	19	44	0	19	45	0	19
46	0	11	47	0	8	48	0	8	49	0	0	CONTINUE	0	38
51	0	38	52	0	0	53	0	38	UPSTAI1	0	76	55	0	38
56	0	43	57	0	43	RETEST	0	0	59	0	0	60	0	0
LOCOFF	0	161	OFFLINE	0	485	63	0	485	64	0	321	65	0	130
66	0	34	RETAIN	0	156	68	0	156	SECOND	0	159	70	0	43
71	0	43	72	0	43	73	0	116	74	0	116	75	0	3
76	0	3	77	0	3	78	0	92	79	0	92	80	0	92
REPAIR	0	120	82	0	120	83	0	64	84	0	40	85	0	16
FIX1	0	64	87	0	64	FIX2	0	40	89	0	40	FIX3	0	16
TAGA	0	120	92	0	120	93	0	120	94	0	120	95	0	83
96	0	83	97	0	83	98	0	37	99	0	37	100	0	37
101	0	0	102	0	0	103	0	0	104	0	28	105	0	28
PERFVER	0	279	107	0	279	108	0	148	109	0	92	110	0	39
DELAY1	0	148	112	0	148	DELAY2	0	92	114	0	92	DELAY3	0	39
TAG2	0	279	117	0	248	BRAN1	0	95	BRAN2	0	153	TAG1	0	31
BRANCHB	0	153	BRANCHC	0	126	DIAGNOST	0	153	124	0	153	125	0	80
126	0	53	127	0	20	WORK1	0	80	129	0	80	WORK2	0	83

131	0	53	WORK3	0	20	TAG3	0	153	RETAIN3	0	123	135	0	82
136	0	0	137	0	92	ROUTER	0	120	ROUTEC	0	3	ROUTEM	0	80
141	0	30	HASTOFF	0	359	143	0	359	144	0	359	145	0	359
146	0	230	147	0	129	RFI	0	86	149	0	54	150	0	32
CHECK	0	730	152	0	729	153	0	87	154	0	87	155	0	87
SYSTEM2	0	87	157	0	87	158	0	87	159	0	87	160	0	87
161	0	87	REENTER2	0	100	OTHER2	0	99	LOCOM2	0	80	165	0	80
HECOM2	0	19	167	0	19	HICOM2	0	1	SELFTES2	0	100	170	0	100
NOFAIL2	0	13	172	0	13	FAIL2	0	85	SPAR2	0	100	VERIF2	0	67
ONLIN2	0	87	177	0	87	178	0	67	179	0	4	180	0	16
RECYCL2	0	56	182	0	56	183	0	56	CHECKU2	0	20	WAITIN2	0	76
186	0	76	187	0	9	188	0	9	189	0	9	COMPAR2	0	67
ADVANC2	0	20	192	0	20	193	0	9	JUMP2	0	20	195	0	20
196	0	20	197	0	20	198	0	16	199	0	4	200	0	4
201	0	0	CONTIN2	0	67	203	0	67	204	0	13	205	0	54
UPSTAT2	0	108	207	0	54	208	0	87	209	0	87	RNTES2	0	13
211	0	13	212	0	13	213	0	179	214	0	179	215	0	179
SYSTEM3	0	179	217	0	179	218	0	179	219	0	179	220	0	179
221	0	179	REENTER3	0	190	OTHER3	0	180	LOCOM3	0	119	225	0	119
HECOM3	0	61	227	0	61	HICOM3	0	10	SELFTES3	0	190	230	0	190
NOFAIL3	0	19	232	0	19	FAIL3	0	171	SPAR3	0	190	VERIF3	0	83
ONLIN3	0	146	237	0	146	238	0	93	239	0	16	240	0	27
RECYCL3	0	8	242	0	8	243	0	8	CHECKU3	0	53	WAITIN3	0	61
246	0	61	247	0	31	248	0	31	249	0	31	COMPAR3	0	80
ADVANC3	0	53	252	0	53	253	0	31	JUMP3	0	53	255	0	53

CONTINUED

BLOCK COUNTS

BLOCK	CURR,	TOTAL	BLOCK	CURR,	TOTAL	BLOCK	CURR,	TOTAL	BLOCK	CURR,	TOTAL	BLOCK	CURR,	TOTAL
256	0	53	257	0	53	258	0	37	259	0	16	260	0	16
261	0	0	CONTIN3	0	93	263	0	93	264	0	11	265	0	82
UPSTAT3	1	163	267	0	82	268	0	179	269	0	179	RETES3	0	11
271	0	11	272	0	11	273	0	19	274	0	19	275	0	19
SYSTEM4	0	19	277	0	19	278	0	19	279	0	19	280	0	19
281	0	19	REENTER4	0	19	OTHER4	0	18	LOCOM4	0	11	285	0	11
MECOM4	0	7	287	0	7	HICOM4	0	1	SELFTES4	0	19	290	0	19
NOFAIL4	0	3	292	0	3	FAIL4	0	16	SPAR4	0	19	VERIF4	0	11
ONLIN4	0	16	297	0	16	298	0	11	299	0	4	300	0	1
RECYCL4	0	0	302	0	0	303	0	0	CHECKU4	0	5	WAITIN4	0	5
306	0	3	307	0	3	308	0	3	309	0	3	COMPAR4	0	2
ADVANC4	0	3	312	0	5	313	0	3	JUMP4	0	3	315	0	8
316	0	3	317	0	5	318	0	2	319	0	3	320	0	3
321	0	0	CONTIN4	0	11	323	0	11	324	0	0	325	0	11
UPSTAT4	0	22	327	0	11	328	0	19	329	0	19	RETES4	0	0
331	0	0	332	0	0	333	0	46	334	0	46	335	0	46
SYSTEM5	0	46	337	0	46	338	0	46	339	0	46	340	0	46
341	0	46	REENTER5	0	58	OTHER5	0	58	LOCOM5	0	55	345	0	55
MECOM5	0	5	347	0	5	HICOM5	0	0	SELFTES5	0	58	350	0	58
NOFAIL5	0	16	352	0	16	FAIL5	0	42	SPAR5	0	58	VERIF5	0	67
ONLIN5	0	56	357	0	56	358	0	47	359	0	3	360	0	6
RECYCL5	0	18	362	0	18	363	0	18	CHECKU5	0	9	WAITIN5	0	27
366	0	27	367	0	6	368	0	6	369	0	6	COMPAR5	0	21
ADVANC5	0	9	372	0	9	373	0	6	JUMP5	0	9	375	0	9
376	0	9	377	0	9	378	0	8	379	0	1	380	0	1
381	0	0	CONTIN5	0	47	383	0	47	384	0	12	385	0	35

UPSTAT5	0	70	387	0	35	388	0	46	389	0	46	RETES5	0	12
391	0	12	392	0	12	393	0	72	394	0	72	395	0	72
SYSTEM6	0	72	397	0	72	398	0	72	399	0	72	400	0	72
401	0	72	REENTER6	0	75	OTHER6	0	71	LOCOM6	0	59	409	0	59
MECOM6	0	12	407	0	12	HICOM6	0	4	SELFTES6	0	75	410	0	75
NOFAIL6	0	3	412	0	3	FAIL6	0	72	SPAR6	0	75	VERIF6	0	68
ONLIN6	0	88	417	0	88	418	0	68	419	0	8	420	0	12
RECYCL6	0	50	422	0	50	423	0	50	CHECK06	0	20	WAITIN6	0	70
426	0	70	427	0	17	428	0	17	429	0	17	COMPAR6	0	33
ADVANC6	0	20	432	0	20	433	0	17	JUMP6.	0	20	435	0	20
436	0	20	437	0	20	438	0	12	439	0	8	440	0	8
441	0	0	CONTIN6	0	68	443	0	68	444	0	5	445	0	65
UPSTAT6	0	130	447	0	65	448	0	72	449	0	72	RETES6	0	5
451	0	5	452	0	3	OFF	0	1						

APPENDIX C

FACILITY STATISTICS

First Quarter Baseline Output

RANGE	FACILITY ID	AVERAGE UTLIZATION	NUMBER ENTRIES	AVERAGE TIME/TRANS	SEIZING TRNSACTION	PREEMPTING TRNSACTION	RANGE
1	SYSTEM1	0.049	64	101.469	0	0	1
2	SYSTEM2	0.110	99	147.101	680	0	2
3	SYSTEM3	0.098	177	73.215	0	0	3
4	SYSTEM4	0.022	28	102.786	0	0	4
5	SYSTEM5	0.076	37	271.676	0	0	5
6	SYSTEM6	0.093	102	120.029	0	0	6

RANGE	FACILITY ID	AVERAGE UTLIEATION	NUMBER ENTRIES	AVERAGE TIME/TRANS	SEIZING TPNSACTION	PREEMPTING TRNSACTION	RANGE
1	SYSTEM1	0.063	70	117.929	0	0	1
2	SYSTEM2	0.114	108	139.065	0	0	2
3	SYSTEM3	0.122	178	89.893	0	0	3
4	SYSTEM4	0.030	29	135.621	0	0	4
5	SYSTEM5	0.095	42	298.357	0	0	5
6	SYSTEM6	0.110	134	108.015	0	0	6

Second Quarter Baseline Output

Third Quarter Baseline Output

RANGE	FACILITY ID	AVERAGE UTLIZATION	NUMBER ENTRIES	AVERAGE TIME/TRANS	SEIZING TRNSACTION	PREEMPTING TRNSACTION	RANGE
1	SYSTEM1	0.064	81	104.481	0	0	1
2	SYSTEM2	0.116	105	144.752	0	0	2
3	SYSTEM3	0.133	174	100.833	0	0	3
4	SYSTEM4	0.020	34	77.118	0	0	4
5	SYSTEM5	0.076	38	262.000	0	0	5
6	SYSTEM6	0.103	100	135.760	0	0	6

RANGE	FACILITY ID	AVERAGE UTLIZATION	NUMBER ENTRIES	AVERAGE TIME/TRANS	SEIZING TRNSACTION	PREEMPTING TRNSACTION	RANGE
1	SYSTEM1	0.044	60	97.650	0	0	1
2	SYSTEM2	0.130	96	177.833	0	0	2
3	SYSTEM3	0.135	210	84.757	0	0	3
4	SYSTEM4	0.014	22	82.773	0	0	4
5	SYSTEM5	0.112	52	282.712	0	0	5
6	SYSTEM6	0.084	89	124.101	0	0	6

Fourth Quarter Baseline Output

APPENDIX D

QUEUE STATISTICS

RANGE	QUEUE ID	MAXIMUM CONTENTS	AVERAGE CONTENTS	TOTAL ENTRIES	ZERO ENTRIES	%-ZERO ENTRIES	AVERAGE TIME/TRANS	NZ AVERAGE TIME/TRANS	TABLE NUMBER	CURRENT CONTENTS	RANGE
1	SYSTEM1	1	0.002	47	43	91.48	5.809	68.250	0	0	1
2	ONLINE	1	0.012	54	0	.00	30.130	30.130	0	0	2
3	RETAIN	4	0.255	186	0	.00	215.499	215.649	0	0	3
4	PERFVER	3	0.169	255	0	.00	87.322	87.322	0	0	4
5	DIAGNOST	2	0.067	124	0	.00	71.056	71.056	0	0	5
6	REPAIR	1	0.019	95	0	.00	26.653	26.653	0	0	6
7	MASTOFF	1	0.000	338	338	100.00	0.000	0.000	0	0	7
8	SYSTEM2	1	0.005	87	76	87.35	8.023	63.455	0	0	8
9	ONLIN2	2	0.016	69	0	.00	31.449	31.449	0	0	9
10	SYSTEM3	2	0.012	146	127	86.98	11.250	86.368	0	0	10
11	ONLIN3	3	0.037	135	0	.00	35.844	35.844	0	0	11
12	SYSTEM4	1	0.000	25	25	100.00	0.000	0.000	0	0	12
13	ONLIN4	1	0.004	20	0	.00	29.100	29.100	0	0	13
14	SYSTEM5	1	0.004	31	28	90.32	17.226	178.000	0	0	14
15	ONLIN5	2	0.022	40	0	.00	71.500	71.500	0	0	15
16	SYSTEM6	1	0.002	80	77	96.24	2.850	76.000	0	0	16
17	ONLIN6	2	0.016	102	0	.00	20.274	20.294	0	0	17
47		1	0.002	9	0	.00	32.333	32.333	0	0	47
48		1	0.003	12	0	.00	29.500	29.500	0	0	48
57		1	0.003	20	9	44.99	16.950	30.818	0	0	57
58		1	0.003	12	0	.00	31.250	31.250	0	0	58
67		2	0.006	36	12	33.33	20.250	30.375	0	0	67
68		2	0.006	91	65	71.42	8.505	29.769	0	0	68
77		1	0.000	2	0	.00	32.000	32.000	0	0	77
78		1	0.001	5	0	.00	29.200	29.200	0	0	78
87		1	0.002	48	44	91.66	5.208	62.500	0	0	87
88		1	0.002	3	0	.00	67.000	67.000	0	0	88
97		1	0.001	29	21	72.41	4.892	17.625	0	0	97
98		1	0.002	19	4	21.05	14.579	18.467	0	0	98

First Quarter Baseline Output

RANGE	QUEUE ID	MAXIMUM CONTENTS	AVERAGE CONTENTS	TOTAL ENTRIES	ZERO ENTRIES	%-ZERO ENTRIES	AVERAGE TIME/TRANS	NZAAVERAGE TIME/TRANS	TABLE NUMBER	CURRENT CONTENTS	RANGE
1	SYSTEM1	2	0.002	58	55	94,82	4,552	88,000	0	0	1
2	ONLINE	2	0.016	66	0	,00	31,015	31,015	0	0	2
3	RETAIN	4	0.339	181	0	,00	246,194	246,144	0	1	3
4	PERFVER	4	0.223	314	0	,00	93,393	93,363	0	0	4
5	DIAGNOST	2	0.090	164	0	,00	72,312	72,312	0	1	5
6	REPAIR	2	0.027	125	0	,00	28,272	28,272	0	0	6
7	MASTOFF	1	0.000	368	368	100,00	0,000	0,000	0	0	7
8	SYSTEM2	1	0.003	90	85	94,44	4,594	81,800	0	0	8
9	ONLIN2	1	0.021	86	0	,00	31,628	31,628	0	0	9
10	SYSTEM3	3	0.025	148	130	87,83	22,379	184,167	0	0	10
11	ONLIN3	2	0.033	133	0	,00	32,872	32,872	0	0	11
12	SYSTEM4	1	0.001	28	27	96,42	4,321	121,000	0	0	12
13	ONLIN4	1	0.003	14	0	,00	30,193	30,143	0	0	13
14	SYSTEM5	2	0.006	34	31	91,17	22,971	260,333	0	0	14
15	ONLIN5	2	0.020	45	0	,00	58,778	58,778	0	0	15
16	SYSTEM6	1	0.004	99	90	90,90	4,808	52,889	0	0	16
17	ONLIN6	2	0.021	133	0	,00	20,308	20,308	0	0	17
47		1	0.001	8	3	37,50	17,375	27,800	0	0	47
48		1	0.003	15	0	,00	29,400	29,400	0	0	48
57		1	0.003	25	11	44,00	16,200	28,929	0	0	57
58		1	0.003	12	0	,00	27,750	27,750	0	0	58
67		1	0.004	48	30	62,50	11,250	30,000	0	0	67
68		1	0.008	40	6	14,99	24,925	29,324	0	0	68
77		1	0.000	2	0	,00	30,000	30,000	0	0	77
78		1	0.001	4	0	,00	31,000	31,000	0	0	78
87		1	0.002	4	0	,00	57,750	57,750	0	0	87
88		1	0.003	9	2	22,72	37,333	48,000	0	0	88
97		1	0.002	33	14	42,42	9,545	16,579	0	0	97
98		1	0.002	33	17	51,51	9,030	18,625	0	0	98

Second Quarter Baseline Output

RANGE	QUEUE ID	MAXIMUM CONTENTS	AVERAGE CONTENTS	TOTAL ENTRIES	ZERO ENTRIES	%-ZERO ENTRIES	AVERAGE TIME/TRANS	NZ AVERAGE TIME/TRANS	TABLE NUMBER	CURRENT CONTENTS	RANGE
1	SYSTEM1	1	0.002	58	53	91.37	3.534	41.000	0	0	1
2	ONLINE	1	0.017	72	0	.00	31.764	31.764	0	0	2
3	RETAIN	3	0.303	159	0	.00	251.044	251.044	0	0	3
4	PERFVER	3	0.200	276	0	.00	95.377	95.377	0	0	4
5	DIAGNOST	2	0.080	143	0	.00	73.427	73.427	0	0	5
6	REPAIR	2	0.024	115	0	.00	26.887	26.887	0	0	6
7	MASTOFF	1	0.000	362	362	100.00	0.000	0.000	0	0	7
8	SYSTEM2	2	0.016	89	77	86.51	23.202	172.083	0	0	8
9	ONLIN2	2	0.021	89	0	.00	31.348	31.348	0	0	9
10	SYSTEM3	2	0.010	153	140	91.50	8.170	96.385	0	0	10
11	ONLIN3	2	0.031	130	0	.00	31.131	31.131	0	0	11
12	SYSTEM4	1	0.000	28	27	96.42	1.821	51.000	0	0	12
13	ONLIN4	1	0.006	27	0	.00	29.259	29.259	0	0	13
14	SYSTEM5	1	0.004	34	32	94.11	13.912	236.500	0	0	14
15	ONLIN5	2	0.018	35	0	.00	68.457	68.457	0	0	15
16	SYSTEM6	2	0.005	87	78	89.65	7.736	74.778	0	0	16
17	ONLIN6	2	0.015	99	0	.00	20.323	20.323	0	0	17
47		1	0.003	12	0	.00	30.167	30.167	0	0	47
48		1	0.003	28	13	46.42	16.179	30.200	0	0	48
57		1	0.003	11	1	9.09	30.273	33.300	0	0	57
58		1	0.002	23	12	52.17	13.595	28.364	0	0	58
67		1	0.003	32	18	56.25	12.844	29.357	0	0	67
68		1	0.008	40	2	4.99	27.475	28.921	0	0	68
77		1	0.002	7	0	.00	31.000	31.000	0	0	77
78		1	0.001	3	0	.00	30.667	30.667	0	0	78
87		1	0.003	25	20	79.99	13.200	66.000	0	0	87
88		1	0.001	3	0	.00	47.333	47.333	0	0	88
97		1	0.000	20	17	84.99	2.850	19.000	0	0	97
98		1	0.002	14	1	7.14	16.357	17.615	0	0	98

Third Quarter Baseline Output

RANGE	QUEUE ID	MAXIMUM CONTENTS	AVERAGE CONTENTS	TOTAL ENTRIES	ZERO ENTRIES	%-ZERO ENTRIES	AVERAGE TIME/TRANS	NZ-AVERAGE TIME/TRANS	TABLE NUMBER	CURRENT CONTENTS	RANGE
1	SYSTEM1	1	0.001	43	42	97.67	1.674	72.000	0	0	1
2	ONLINE	1	0.013	57	0	.00	30.333	30.333	0	0	2
3	RETAIN	2	0.293	156	0	.00	247.538	247.538	0	0	3
4	PERFVER	2	0.188	279	0	.00	88.978	88.978	0	0	4
5	DIAGNOST	2	0.081	153	0	.00	69.301	69.301	0	0	5
6	REPAIR	2	0.024	120	0	.00	26.567	26.567	0	0	6
7	MASTOFF	1	0.000	339	339	100.00	0.000	0.000	0	0	7
8	SYSTEM2	2	0.019	87	72	82.75	29.077	168.333	0	0	8
9	ONLIN2	2	0.024	87	0	.00	36.529	36.529	0	0	9
10	SYSTEM3	2	0.012	179	160	89.38	9.029	85.158	0	0	10
11	ONLIN3	2	0.034	146	0	.00	30.767	30.767	0	0	11
12	SYSTEM4	1	0.000	19	19	100.00	0.000	0.000	0	0	12
13	ONLIN4	1	0.004	16	0	.00	29.625	29.625	0	0	13
14	SYSTEM5	1	0.005	46	43	93.47	14.000	214.667	0	0	14
15	ONLIN5	2	0.027	56	0	.00	62.393	62.393	0	0	15
16	SYSTEM6	1	0.004	72	67	93.05	7.528	108.400	0	0	16
17	ONLIN6	2	0.016	88	0	.00	23.307	23.307	0	0	17
47		1	0.002	8	0	.00	29.500	29.500	0	0	47
48		1	0.002	11	0	.00	29.455	29.455	0	0	48
57		1	0.001	4	0	.00	32.000	32.000	0	0	57
58		2	0.004	72	56	77.77	6.681	30.063	0	0	58
67		1	0.004	16	0	.00	30.250	30.250	0	0	67
68		1	0.008	45	8	17.77	23.911	29.081	0	0	68
77		1	0.001	4	0	.00	31.750	31.750	0	0	77
78		1	0.000	1	0	.00	25.000	25.000	0	0	78
87		1	0.001	3	0	.00	59.000	59.000	0	0	87
88		1	0.003	24	18	75.00	15.417	61.667	0	0	88
97		1	0.001	53	45	84.90	2.755	18.250	0	0	97
98		1	0.002	17	5	29.41	12.000	17.000	0	0	98

Fourth Quarter Baseline Output

APPENDIX E

FIELD SURVEY SHEET
(Facsimile)

Flow Parameter Value

Block	Description	Unit	General	Complexity Level		
				Lo	Med	Hi
S1	Time between failures	Hours	40 ± 8			
S2	Complexity assignment	%		29	45	26
S3	On-Line diagnostic	Min	60 ± 12			
S4	Percent actual failures	%	99			
S5	Percent spares avail	%	80			
S6-10	On-Line verification	Min	30 ± 6			

APPENDIX F

ANALYSIS OF VARIANCE

ANALYSIS OF VARIANCE
System 2 Sample

Availability Observations	Treatments (Self-Test Accuracy-%) 80	85	90
1st QTR	.866	.867	.890
2nd QTR	.863	.886	.886
3rd QTR	.849	.890	.884
4th QTR	.869	.865	.870
T.j	3.447	3.508	3.530
nj	4	4	4
y^2ij	2.970687	3.07701	3.115452

$T.. = 10.485$

$N = 12$

$\Sigma\Sigma y^2 = 9.163149$

$\Sigma \frac{T.^2j}{nj} = \frac{1}{4}(11.881809 + 12.306064 + 12.4609) = 9.162193$

$\frac{T..^2}{N} = \frac{1}{12}(109.93523) = 9.161269$

$SST = \Sigma\Sigma y^2 - \frac{T..^2}{N} = .00188$

$SSG = \Sigma \frac{T.^2j}{nj} - \frac{T..^2}{N} = .000924$

$SSW = SST\text{-}SSG = .000956$

ANOVA SUMMARY TABLE

Source of Variation	Sum of Squares	df	Mean Square
Among Treatments	.000924	2	.000462
Within Treatments	.000956	9	.000106
TOTAL	.00188	11	

$$\frac{MSG}{MSW} > F_{1-\alpha;k-1;N-k} = \frac{462}{106} = 4.358$$

$$4.358 > F_{.95;2;11} = 4.26$$

CONCLUSION: The hypothesis of equal means is rejected at the 5% level of significance and it is concluded that the mean system availability due to changes in self-test accuracy is significantly different than the mean availability due to random failure deviations in the System 2 sample.

APPENDIX G

MULTI-REGRESSION ANALYSIS

MULTI REGRESSION DATA (SYSTEM 2)

X1	X2	X3	X4	Y
.9	.6	.7	.95	.89
.9	.6	.7	.95	.886
.9	.6	.7	.95	.884
.9	.6	.7	.95	.87
.9	.6	.8	.95	.89
.9	.6	.8	.95	.871
.9	.6	.8	.95	.895
.9	.6	.8	.95	.872
.8	.6	.8	.95	.866
.8	.6	.8	.95	.863
.8	.6	.8	.95	.849
.8	.6	.8	.95	.869
.85	.6	.8	.95	.867
.85	.6	.8	.95	.886
.85	.6	.8	.95	.89
.85	.6	.8	.95	.865
.9	.75	.8	.95	.877
.9	.75	.8	.95	.868
.9	.75	.8	.95	.892
.9	.75	.8	.95	.885
.9	.8	.8	.95	.865
.9	.8	.8	.95	.886
.9	.8	.8	.95	.885
.9	.8	.8	.95	.889
.9	.85	.8	.95	.848
.9	.85	.8	.95	.874
.9	.85	.8	.95	.899
.9	.85	.8	.95	.884
.9	.9	.8	.95	.864
.9	.9	.8	.95	.888
.9	.9	.8	.95	.856
.9	.9	.8	.95	.858
.9	.6	.9	.95	.874
.9	.6	.9	.95	.867
.9	.6	.9	.95	.869
.9	.6	.9	.95	.884
.9	.6	.8	.8	.889
.9	.6	.8	.8	.867
.9	.6	.8	.8	.894
.9	.6	.8	.8	.884
.9	.6	.8	.85	.850
.9	.6	.8	.85	.879
.9	.6	.8	.85	.858
.9	.6	.8	.85	.879
.9	.6	.8	.9	.877
.9	.6	.8	.9	.858
.9	.6	.8	.9	.897
.9	.6	.8	.9	.867

Note:

X1 = Self-Test Accuracy
X2 = Spares Probability
X3 = Performance Test Accuracy
X4 = Diagnostic Test Accuracy
Y = Observed Availability

MULTIPLE CORRELATION VAI COOLEY-LOHNES 5 TESTS 48 SUBJECTS

DETERMINANT = .7022568657671

MULTIPLE R SQUARE = .12108924713

MULTIPLE R = .3479788027021

FOR ANALYSIS OF VARIANCE ON R = 1.481048448204

N.D.F. 1 = 4 N.D.F. 2 = 43

PREDICTOR	BETA	BETA SQ	R(CRITERION)	BETA*R	STRUCTURE-R
1	0.35	0.12	0.271	0.09	0.78
2	-0.17	0.03	-0.024	0.00	-0.07
3	-0.14	0.01	-0.140	0.01	-0.40
4	0.14	0.02	0.006	0.00	0.01

TEST	MEAN	S.D
1	0.88	0.02
2	0.67	0.11
3	0.80	0.04
4	0.92	0.04
5	0.87	0.01

WEIGHTS
.1556788848735
-2.08763551E-02
-4.49999999E-02
4.01277749E-02

INTERCEPT CONSTANT = .7501000043023

Predictor 1(Self-Test Accuracy) is the best predictor of the criterion variable (Observed Availability).

The regression equation is:

$$Y' = .750 + .156X1 - .021X2 - .045X3 + .040X4$$

The full correlation matrix is shown on the next page.

5 TESTS 48 NUMBER OF SUBJECTS. RANK = 5

		1.00	2.00	3.00	4.00	5.00
X1	1	0.99	0.28	0.00	-0.21	0.27
X2	2		1.00	0.00	0.35	-0.02
X3	3			1.00	0.00	-0.14
X4	4				1.00	0.00
Y	5					1.00

DETERMINANT OF R = .6172211106002

FOR SPHERICITY TEST, CHI-SQUARE = 21.4724940088 AND N.D.F = 10

FACTOR	EIGENVALUE	PERCENT TRACE	CUM PERCENT	N.D.F	CHI-SQUAR
1	1.39	27.93	27.93	10	21.472
2	1.33	26.79	54.73	6	17.484
3	1.05	21.19	75.92	3	11.397
4	0.82	16.55	92.47	1	6.446
5	0.37	7.52	99.99	0	0.000

FACTOR PATTERN. FACTORS ARE COLUMNS, TESTS ARE ROWS.

ROW	1	2	3	4	5
1	0.74	-0.39	0.37	-0.16	-0.34
2	0.68	0.55	0.18	-0.26	0.34
3	-0.19	0.16	0.82	0.50	0.03
4	0.20	0.82	-0.29	0.30	-0.32
5	0.53	-0.41	-0.35	0.62	0.17

TEST	COMMUNALITY	MULT R SQUARE	
1	0.99999	0.27995	X1
2	1.00000	0.28110	X2
3	0.99999	0.02192	X3
4	1.00000	0.25009	X4
5	0.99999	0.12108	Y

FACTOR SCORE COEFFICIENTS. FACTORS ARE COLUMNS, TESTS ARE ROWS

ROW	1.00	2.00	3.00	4.00	5.00
1	0.53	-0.29	0.35	-0.19	-0.92
2	0.49	0.41	0.17	-0.31	0.92
3	-0.13	0.12	0.77	0.61	0.10
4	0.14	0.61	-0.27	0.36	-0.85
5	0.38	-0.30	-0.33	0.75	0.47

Bibliography

Campbell, George W. Intermediate Maintenance Concepts and Use of ATE for Training Devices (Preliminary Study). Orlando, Fla.: Naval Training Equipment Center, 1977.

Cooley, William W., and Lohnes, Paul R. Multivariate Data Analysis. New York: John Wiley & Sons, 1971.

Gibra, Isaac N. Probability and Statistical Inference for Scientists and Engineers. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1973.

Gross, Donald, and Harris, Carl M. Fundamentals of Queuing Theory. New York: John Wiley & Sons, 1974.

Simulation System Associates, Inc. Study Report: Aviation Trainer Maintenance, Contract Number N00140-77-C-1018. 2 vols. Northport, N.Y.: Simulation Systems Associates, Inc., 1978.

Sperry Rand-Univac. Flow Simulator Reference Manual, Series 70 Publication. Cinnaminson, N.J.: Sperry Rand, 1972.

U. S. Department of the Navy. Office of the Chief of Naval Operations. The Naval Aviation Maintenance Program (NAMP). Washington, D.C., (1973).