THE ARINC RESEARCH SYSTEM TESTABILITY AND MAINTENANCE PROGRAM (STAMP)

by

William R. Simpson,. PhD
Harold S. Balaban, PhD

Prepared for Presentation at the

1982 IEEE AUTOTESTCON Conference
Dayton, Ohio

October 12-14, 1982

Copyright © 1982

Institute of Electrical and Electronics Engineers, Inc.

Reproduced with Permission

ARINC Research Corporation
a Subsidiary of Aeronautical Radio, Inc.
2551 Riva Road
Annapolis, Maryland 21401
Publication 6136-2759

THE ARINC RESEARCH SYSTEM TESTABILITY AND MAINTENANCE PROGRAM (STAMP)

William R. Simpson, PhD*, and Harold S. Balaban, PhD**
ARINC Research Corporation
Annapolis, Maryland

Abstract

STAMP is a computer-aided testability design and fault diagnosis system. Using first-order test point and com^{p}onent dependencies as inputs, the model generates all higher order dependencies and their implications. This permits testability assessment through automatic identification of component ambiguity groups, redundant and unnecessary test points, and feedback loops. The model also provides several overall measures of testability and fault isolation. These measures are described in the paper. STAMP also provides a means for developing fault isolation strategies. A choice of several search strategies is provided. One, based on an adaptive, information theoretic approach, appears to offer significant efficiency advantages in lowering both the expected value and the variance of the number of tests required to isolate faults. The detailed analysis of a sample system is presented, including testability redesign and development of a strategy for fault isolation.

Introduction

Achieving a desirable level of maintainability is becoming more difficult and costly for many modern systems because of their complexity and sophistication. In particular, fault isolation at intermediate and depot levels of maintenance presents significant problems despite advances in automatic test equipment.[1][2][3][4] Current system and test design often results in high ambiguity levels for fault isolation and long test time. Recent studies involving the CH-54 helicopter[3] and the F-16 aircraft have shown that trouble-shooting actions can consume as much as 50 percent or more of the total man-hours spent for repair. Those figures suggest that there is a large potential return on an investment in improved testability assessment and fault isolation procedures.

This paper describes one such procedure, called STAMP, an acronym for System Testability and Maintenance Program. STAMP uses a functional dependency input representing the system to develop a complete testability assessment. This assessment provides a number of normative testability measures as well as information necessary for implementing testability design improvements. Once the design has been completed, STAMP is used to develop an efficient fault-isolation strategy.

Logic Modeling and Maintenance Dependency

The procedure that is used by STAMP for analyzing system testability is based on the design topology. That topology is usually represented by the well-known functional block diagram that shows the functional dependencies among the elements of systems. In addition to the functional elements of a system, we also include in the diagram the predetermined or candidate tests to be used for fault isolation. Two types of test are considered:

- Functional Tests - Tests that indicate the correct functioning of all system elements that "feed" the test point
- Special Tests - Tests that have dependencies that are not readily described by a standard functional diagram

A key condition required by STAMP is that each test point provide either a "pass" or "fail" outcome. However, there are ways in which more than two outcomes may be treated, as for relays that may be good, fail open, or fail short.

Fault isolation and testability analysis based on functional block diagrams together with test point dependencies has come to be known generically as logic modeling or logic model analysis. A series of studies on logic modeling was conducted by the U.S. Army Air Mobility Research and Development Laboratory, Ames Research Center[5] These studies were directed toward the development of analytical methods in reliability and maintainability technology. Computer implementations were developed and tested for feasibility[6]. Several DoD evaluations of the logic model analysis approach have been conducted in regard to both desk evaluation[7][8] and troubleshooting development.[9] Each of these evaluations gave logic modeling a high potential value.

For applicability to military systems a specification has been developed[10] to standardize development and documentation practices. This specification introduced a structured representation, of test point and component dependencies using graphic symbols and interconnecting lines[11]. This representation is called the maintenance dependency chart (MDC) and is usable as an input to logic modeling analysis techniques, including STAMP.

----
*Principal Scientist, Advanced Research and Development Group, Member AIAA.
**Manager, Advanced Research and Development Group.

Overview of STAMP Application

Figure 1 presents an overview of the STAMP application. A brief summary of the steps is given below with details provided in following sections:

- System Topology. The functional block diagram of the system, including test points, forms the basis for developing the inputs to STAMP.
- STAMP First-Order Dependencies. For each test point, all first-order (immediate) predecessor test points are entered with embedded components.
- Dependency Analysis. Higher-order dependencies are calculated to establish the complete relationship of test points and components.
- Testability Assessment. An analysis of basic testability characteristics inherent in the design along with identification of design improvement areas automatically generates a testability report.
- Redesign or Fault Isolation. A decision by project personnel is made to redesign the system to incorporate testability improvements or proceed to develop a fault isolation strategy.
- Testability Redesign. The information and recommendations provided by the STAMP testability assessment is acted upon by the engineering department to improve testability characteristics.
- Fault Isolation Analysis. On the basis of dependency analysis and weighting factors, an analysis is performed to develop efficient fault isolation strategies.
- Fault Isolation Strategy. The detailed step-by-step sequence of test procedures to follow to isolate faults is determined. A handbook containing the fault isolation strategy in the form of fault-trees with alternatives on initial conditions and an interactive computer-based procedure can guide the maintenance technician.
- Fault Isolation Report. A summary report is automatically generated, describing the testability and fault isolation characteristics of the system, including multiple failure strategies.

System Topology and First-Order Dependency input

Figure 2 is a functional block diagram .of a sample system that we will use in this paper for illustrative purposes. This diagram might represent a full weapon system and the blocks could represent subsystems such as hydraulic, fuel, and environmental control. The diagram could also represent such a low-level item as a PC board with). electronic components being represented by the blocks. For our purposes we will use the terns "components" to represent the individual blocks, C1, C2, ... C9. The nodes represented by T1., T2, . . . T8 are test points, and the signal or dependency flow is indicated by the arrows.

Also shown in the figure are examples of the type of input STAMP requires. For test point: T2, the immediate predecessor test point is T1, Land component C1 is embedded. Similarly, test point T6 has one immediate predecessor test point, T5, and one embedded component, C6. Test point T4 is fed by two branches: one branch has T3 as an immediate predecessor with C4 embedded, and the other branch is fed by T5 with no embedded components.

In addition to these first-order dependency inputs, the user also must enter any weighting information that may be required. STAMP can develop a fault isolation strategy that considers failure rate, test costs, and test times, and such factors must be provided if they are to be considered.

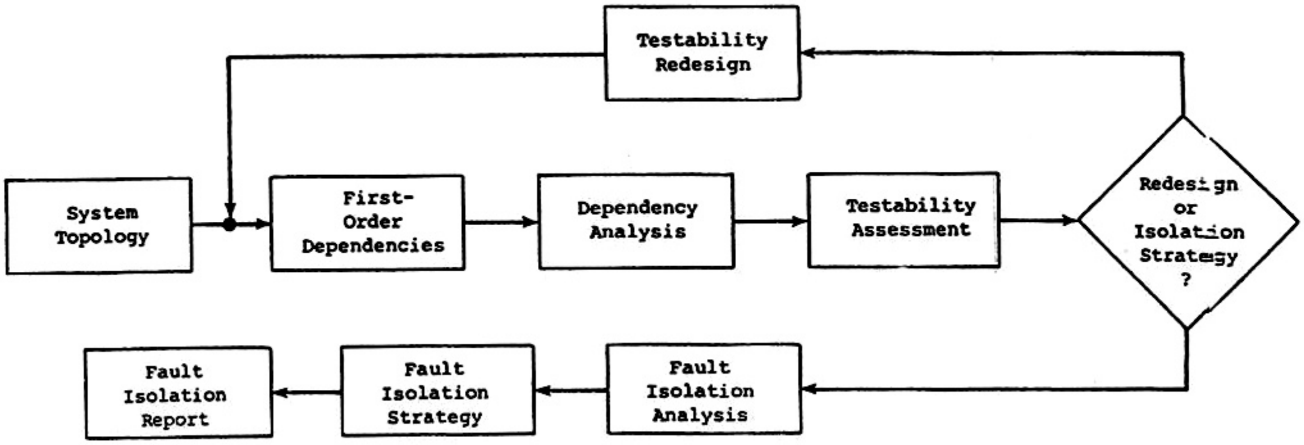


Figure 1. STAMP Application Overview

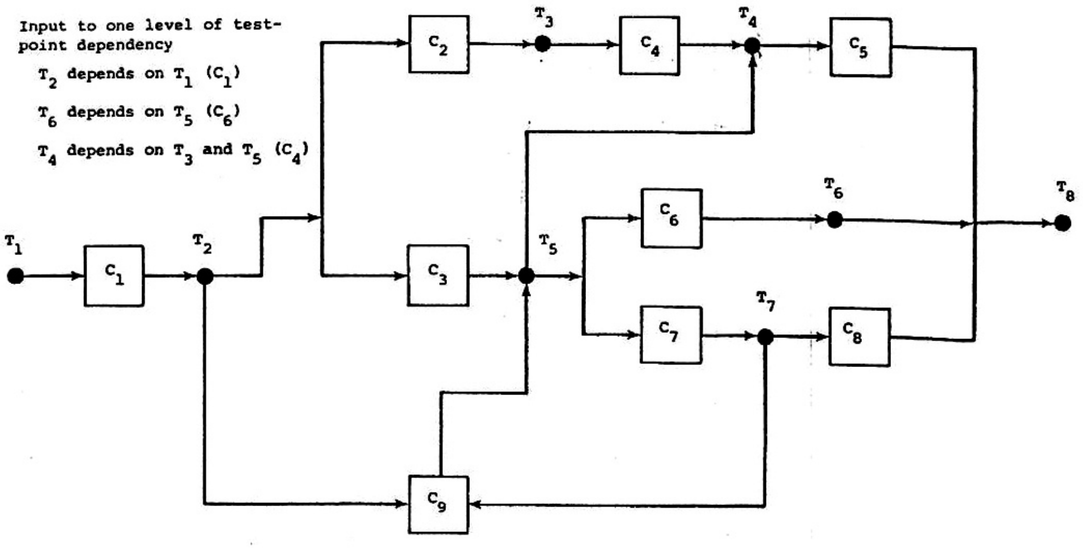

Input to one level of test-point dependency
T_2 depends on T_1 (C_1)
T_6 depends on T_5 (C_6)
T_4 depends on T_3 and T_5 (C_4)
T_1
C_1
T_2
C_2
T_3
C_4
T_4
C_5
C_3
T_5
C_6
T_6
T_8
C_7
T_7
C_8
C_9

Figure 2. Sample System

Dependency and Testability Analysis

Given first-order inputs, STAMP employs a mathematical algorithm to obtain all higher-order dependencies through a manipulation of a matrix representation of the first-order dependency relationship. A full range of testability measures is then generated through analysis of the high-order dependency matrix. Table 1 lists these measures and gives a brief definition of them and shows the values obtained for the sample system.

Table 1. System Testability Measures

Measure	Definition	Sample System Value
Dependency	A measure of test point connectivity	0.56
Component Leverage	Percentage of components uniquely fault isolatable (observability)	0.67
Modified Component Leverage	Percentage of component groups uniquely fault isolatable	0.86
External Dependency Factor	A -measure of dependency on external factors	0.11
Test Point Redundancy Factor	The percentage of test points which contain unique dependencies	0.86
Test Point Leverage	A relative measure of the degree to which the test point set meets theoretical fault isolation limits (controllability)	0.67
Modified Test Point Leverage	Test point leverage for a repackaged system	0.86

The measures are normalized values ranging from 0 to 1. Where applicable, the testability characteristic of a comparably sized full-serial system is used as a basis for normalization or analysis. For example, consider a system consisting of eight components. If all components are directly in series, three tests is the minimum necessary for full fault isolation.* If each component required a separate test, a maximum of eight tests would be required, irrespective of the design. The limits of three and eight then form the basis for developing the test point leverage measure. The normalized measures allow comparison across competing designs and also provide measures of progress as design iterations take place.

STAMP provides an automatically generated testability report. Six major outputs are provided:

- Testability Measures and Discussion
-
- Component Ambiguity Groups
-
- Test Point Redundancies and Excess Test Points
-
- Feedback Loops
-
- Signature Analysis for Hidden Failures
-
- Recommendations for Testability Improvement

Table 1 provided a list of measures and the values for the sample system. Examples of paragraphs discussing these measures developed automatically by STAMP are provided below:

<u>Dependency Measure (DEP = 0.555)</u>

The moderate value of dependency indicates that several gaps in dependency exist. Specialized or adaptive techniques could sharply improve fault isolation. Adaptive techniques may yield the best fault isolation strategies approaching the theoretical limits of between three and four required tests.

<u>Component Leverage (CL = 0.666 MCL - 0.857)</u>

Several component ambiguity groups exist as shown in Table 2. The table lists those components whose faults cannot be individually isolated with the current set of test points. Those components with an asterisk are tied up in one or more feedback loops. They should be packaged together to reduce the number of good components removed. With such packaging the good component removal rate due to component ambiguity is 0.142; without such pack-aging the rate is 0.333. That rate could be reduced further by adding test points to separate those components not tied up in feed-back loops or by repackaging those components.

Table 2. Component Ambiguity Groups

Group Number	Cross Reference†	Component Number
1	T7	C3,* C7,* C9*
2	--	C5, C8

tRefers to a specific feedback loop in Table 3 or an element of that feedback loop.
*Indicates part of feedback loop.

Table 2 illustrates the identification of the component ambiguity groups and feedback loop information. Table 3 shows the information provided through the analysis of test point redundancies and excess test points. A redundant test point is one for which another test point provides identical information. An excess test point is one whose information content is not necessary for fault isolation. As an example, test point T_2 is excess for this design because combinations of other tests (e.g., T_3, T_4, and T_7) can be used to provide the same information. This type of conclusion is not easily reached by inspection even for such a simple design as our illustrative problem.

Table 3. Test Point Redundancies

Group Number	Cross References	Redundancy Group
1	C3	T5,* T7*

Test point analysis indicates one or more of the following test points are not needed:
Test 2, Test 5

tRefers to a specific feedback loop in Table 2 or an element of that feedback loop.
*Indicates part of feedback loop.

<u>Signature Analysis for the Multiple Failure Case</u>

One other analysis performed by STAMP is identification of potential hidden failures and false component failure indications. This is done through analysis of the component failure signatures. A component failure signature as used here can be mathematically defined as a vector

$$K_i = (K_{i1}\ K_{i2}, .. , K_{in})$$

where K_{ij} is equal to 1 if the j^{th} test would fail given C_i has failed, and $K_{ij} = 0$ otherwise. Component C_i is said to dominate C_k if $K_{ij} \geq K_{kj}$ for j = 1, 2, .. , n. Any reasonable

----
*It can be shown that if there are n components in series, at least (log n/log 2) tests are required.[11]

fault isolation procedure will isolate to the dominant component first, which means that if multiple failures were possible, the failure of any dominated components would be hidden. We also have the possibility that failure of two or more components may lead to a false indication that another component has failed. This would occur when there exists a subset of components, say S, such that

$$\bigcup_{j \in S} [\underline{K}_j] = \underline{K}_i$$

That is, the failure signature of a group of components "adds up" to the signature of another component.

STAMP identifies potential hidden and false failures as shown in Table 4 for the sample system. In this case there are no false indication possibilities (an asterisk would be used to identify them), but failures in six of the components as well as the input can hide other failures. The recommended procedure is to determine for which cases it is likely that multiple failures will occur because of physical or environmental dependence. For such cases, it may be desirable to replace both components -- e.g., if the power supply is failed always replace the simple resistor upon which it depends, or else it may be worthwhile to employ a special test to determine if there is a hidden failure of the resistor.

Table 4. Sub-signature Equivalence Table

Number	Failure Indication	Dominated Components	Retest with the Following Tests as Initially Given Good
1	T1 (Input)	Cl, C2, C3, C4, C5, C6	Ti
2	C1	C2, C3, C4, C5, C6	T1
3	C2	C4, C5	T6
4	C3	C4, C5, C6	T3
5	C4	C5	T6, T3
6	C6	C5	T4

The table also indicates how fault isolation should proceed given that the indicated failure has been corrected. To illustrate some of these points, consider line 4 of Table 4. It is seen that a failure of component 3 could hide a failure of component 4. Assume that engineering analysis shows that components 3 and 4 are physically dependent in such a manner that failure of component 4 is likely to cause a failure of component 3. Therefore, if through fault isolation C_3 is identified as having failed and is replaced, it will likely fail again if C_4 is a hidden failure. Devising a special test to identify multiple failures may be beneficial if the occurrence probabilities are not insignificant and always replacing both components will be too costly.

Testability Improvement Through Redesign

Referring back to Table 2, we see that components 3, 7, and 9 are in a feedback loop and that components 5 and 8 form an ambiguity group. Assume that it is decided to insert a gate that is opened for test purposes to distinguish between components C_7 and C_9. That will reduce the rate of removal of good components from 0.33 to 0.22. That change will still leave two ambiguity groups of two (C_3, C_9 and C_5, C_8), which may not be satisfactory. Assume that repackaging of C_3 and C_9 into one replaceable unit is satisfactory, leaving only the C_5 and C_8 ambiguity group.

Using our multiple failure example, assume it is decided to eliminate the possibility of a C_3 failure hiding a C_4 failure. To handle this as well as the remaining ambiguity group, a special test may be developed that distinguishes between components 4 and 5.

When these changes are made, the STAMP testability analysis will show that all component groups are now fully testable and that a component 4 failure is not hidden by a component 3 failure. However, STAMP shows that the modified system now is over-specified in that one or more of test points T_2, T_3, and T_5 are not needed.

The design engineer should be asked to identify the best candidate for elimination. Assume it is test T2; e.g., this test may require expensive access hardware or be time consuming to perform. Rerunning STAMP without T2 will show that Test T5 is still not needed so that it can be eliminated as well.

Redesign Summary

Specific redesign actions for the sample system taken as a result of the STAMP analysis are listed below:

- Insertion of a gate to open the feedback loop
- Repackaging of two components
- Addition of a special test
- Deletion of two functional tests

These actions caused the following improvements in testability:

- Component isolation was increased from 6 to 8.
- Component package isolation was increased from 66.6 to 100 percent (component leverage = 1.0).
- An undesirable multiple failure dependency situation was eliminated.

- Testability complexity was reduced in that the modified system had one less test than the original system (test point leverage increased to 0.75).

Fault Isolation

STAMP can also be used to develop fault isolation strategies. Generally, this would be under-taken after all design changes are made or when the design can no longer be changed. STAMP will provide an orderly and efficient sequencing of the tests for fault isolation of specific components through partitioning.

Dependency and Partitioning

The development of search strategies is usually preceded by ordering. This ordering is a general arrangement of upstream/downstream events. A search strategy may exploit this ordering to select test points by use of a sequence of partitions of the test points. A diagnostic strategy defines the sequence of these partitions and, therefore, the sequence of test points. Complications exist with ordering when the ordering in not unique* and certain elements cannot be ordered such as those in feed-back loops, i.e., the interactive mutual dependencies among two or more test events. Table 5 summarizes strategies implemented in STAMP. The adaptive or information theoretic strategy is new and was designed to overcome the problems induced by the orderin[9] requirement. STAMP does no ordering, but where this lack of ordering creates a decision ambiguity, the information theoretics is used to repair the inconsistency and continue the fault path. To date, the adaptive or information theoretic method has given both a lower mean and a lower variance than other methods. This is illustrated in Table 6, which shows a comparison between the various techniques for an ordered system containing 1B test points and 17 components.** The theoretical limit assumes a 50/50 partition at each test.

Fault Isolation of the Sample System

STAMP offers two methods of fault isolation. The first is interactive and provides the sequence of tests together with information on how to conduct and interpret the tests. When sufficient information has been obtained to isolate the faults, the identification of any failed component is revealed. The second is fault tree development, or a complete set of instructions on how to proceed from a given set of initial conditions. Figure 3 shows the fault tree generated for the redesigned sample system using the adaptive method with no initial conditions. Adaptive fault trees may also be weighted for cost, time, or MTBF data to provide specific isolation objectives. Details of these and other fault isolation data in STAMP are presented in Balaban, Simpson[12]

Table 5. STAMP Implemented Search Strategies

Type of Search	Ordering Required	Strategy
Directed	Yes	Test all test points sequentially from right to left in MDC, beginning with first known fault, except skip test points where result can be inferred from previous tests.
Half-Interval, Directed Combination	Yes	Test at mid-point from left to right of remaining test points until a bad test is encountered. Use directed search from left to right on tests upon which bad test depends, except skip tests whose result can be inferred from previous tests.
Half-Interval, Directed Combination with Pretest of Inputs	Yes	Same as above except that all inputs are tested prior to search.
Exponential, Directed Combination	Yes	Select test closest to 63 percent partition from left to right of remaining test points. Continue until a bad test is encountered, then use directed search from right to left.
Exponential	Yes	Same as above, except continue testing with 63 percent partition.
Adaptive or Information Theoretic	No	For each test point, ask how much information can be inferred from either good or bad tests. Select test points to optimize answer.
Random	No	Randomly choose test points using uniformly distributed random numbers.

Summary

STAMP has been useful in the development of computer-aided design for testability and in the generation of orderly, efficient fault isolation strategies. The analysis is relatively inexpensive and its many automatic features make it extremely attractive. Both mainframe and micro-computer versions have been implemented, the latter having the capability of handling

----

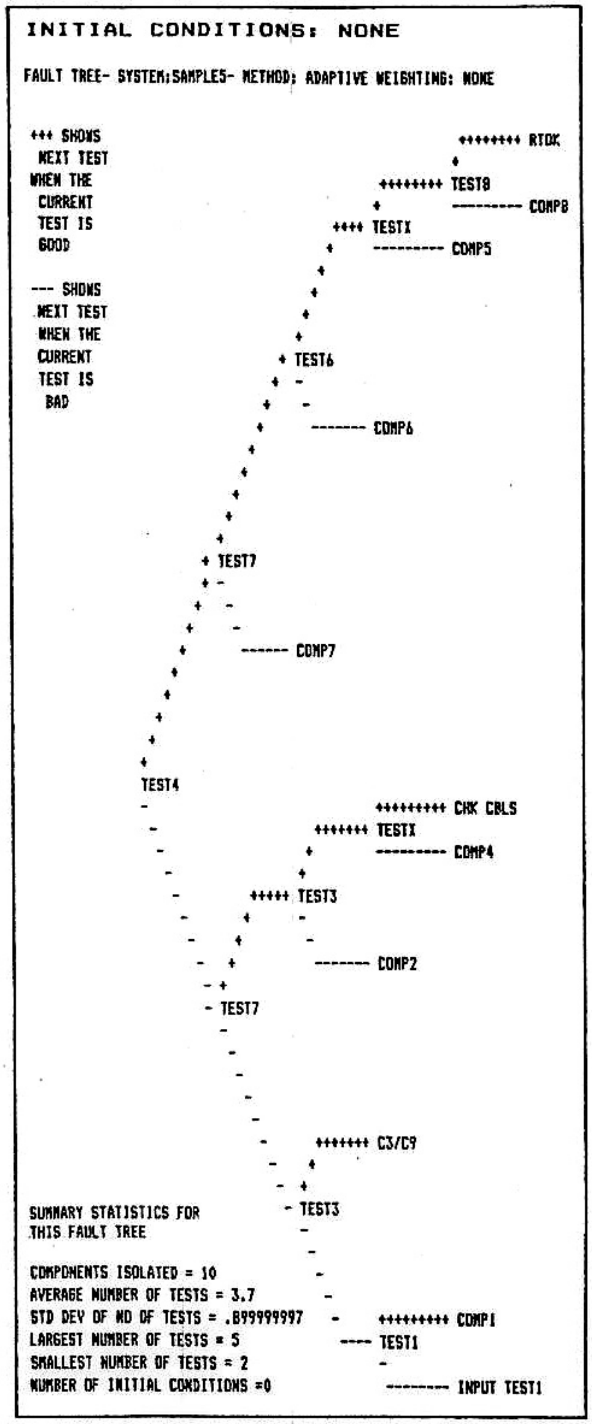
*In mathematical terminology, the test dependencies are partially ordered, not well ordered.
**Example system taken from Cramer, et al.[11]

up to 248 test points and components combined. STAMP has been used on a number of programs including both fielded systems and those undergoing preliminary design.

While there have been no direct applications of interactive fault isolation to fielded systems to date, STAMP offers a potential for video tape/disk interface to provide a strong tool for fault isolation by personnel with low skill levels. STAMP applications to software verification, validation, and debugging as well as medical diagnosis are also being examined.

Table 6. Comparison of Search Strategies for an Example System

Search Strategy	Average Number of Tests Required	Test Variance
Directed	6.86	5.26
Half-Interval, Directed Combination	5.14	1.26
Half-Interval, Directed Combination with Pretest	6.50	1.68
Exponential, Directed Combination	4.50	1.68
Exponential	4.43	1.24
Adaptive	4.35	0.23
Random	5.71	4.49
Theoretical Limit	4.09	--



```
INITIAL CONDITIONS: NONE

FAULT TREE- SYSTEM;SAMPLES- METHOD; ADAPTIVE WEIGHTING: NONE
```

```
SUMMARY STATISTICS FOR
THIS FAULT TREE

COMPONENTS ISOLATED = 10
AVERAGE NUMBER OF TESTS = 3.7
STD DEV OF NO OF TESTS = .899999997
LARGEST NUMBER OF TESTS = 5
SMALLEST NUMBER OF TESTS = 2
NUMBER OF INITIAL CONDITIONS =0
```

Figure 3. Sample Fault Tree

References

1. George Smith II, "Testability Analysis: Predict It More Closely," 1979 Proceedings, Annual Reliability and Maintainability Symposium, Washington, D.C., January 1979.
2. William L. Kiener and Anthony Coppola, "Joint Services Program in Design for Testability," 1981 Proceedings Annual Reliability and Maintainability Symposium, Philadelphia, Pennsylvania, January 1981.
3. Thomas N. Cook and John Ariano, "Analysis of Fault Isolation Criteria/Techniques," 1980 Proceedings Annual Reliability and Maintainability Symposium, San Francisco, California. January 1980.

4. Michael L. Labit, G. T. Harrison, and B. L. Retterer, Special Report on Operational Suitability (OS) Verification Study Focus on Maintainability, ARINC Research Corporation Publication 1751-01-2-2395, Annapolis, Maryland, February 1981.

5. James T. Wong and William L. Andre, "Some Generic Properties of a Logic Model for Analyzing Hardware Maintenance and Design Concepts," U.S. Army Air Mobility Research and Development Laboratory, Symposium presentation on Applications of Decision Theory to Problems of Diagnosis and Repair, June 1976.

6. Ralph A. DePaul, Jr., Maintenance Logic Model Analysis Feasibility Study, Tait and Associates, November 1974.

7. Wilton J. Stiegman, LOGMOD Final Report, Interservice Group on Exchange of Technical Manual Technology, U.S. Air Force Deputy Chief of Staff for Logistics, 21 March 1980.

8. William L. Kiener, The Assessment of LOGMOD a:; a Testability Design Tool, Naval Surface Weapons Center, 31 December 1980.

9. B. F. Lacy and J. B. Berry, LOGMOD Diagnostics, Air Force Logistics Management Center, Project 760701, 15 October 1978.

10. Military Specification, Manuals, Technical: Functionally Oriented Maintenance Manuals (FOMM) for Equipment and Systems, MIL-M-24100B, 2 January 1974.

11. M. L. Cramer, et al, Logic Model Analysis and Standard Maintenance Information Display System (SMIDS), USAAVRADCOM-TR-81-D-45, Applied Technology Laboratory, U.S. Army Research and Technology Laboratories, Fort Eustis, Virginia, April 1982.

12. H. Balaban and W. R. Simpson, "Testability/Fault Isolation by Adaptive Strategy," 1983 Proceedings, Annual Reliability and Maintainability Symposium, Orlando, Florida, January 1983.