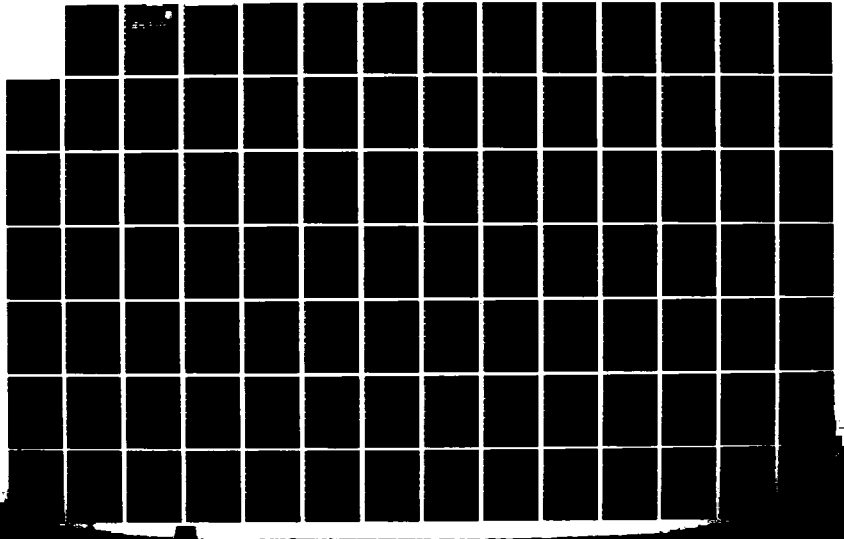
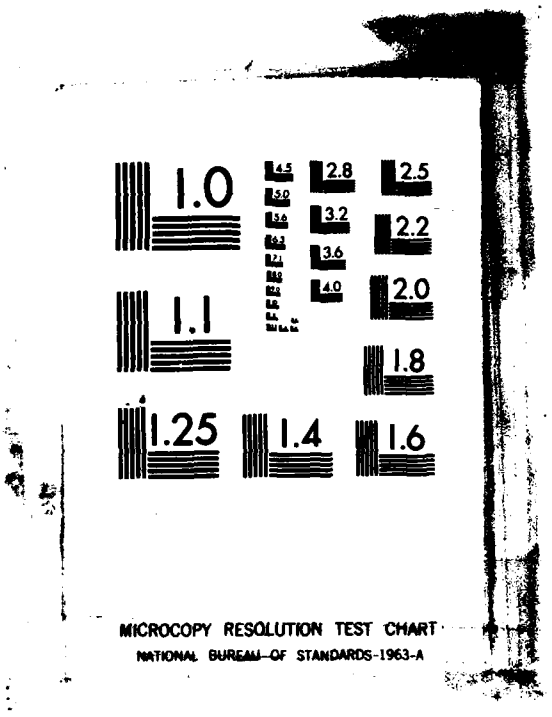
AD-A141 147 ADVANCED APPLICATION OF THE PRINTED CIRCUIT BOARD TESTABILITY DESIGN AND RATING SYSTEM(U) BOEING AEROSPACE CO SEATTLE WA R PPILLMAN ET AL. DEC 83 1/2

UNCLASSIFIED RADC-TR-83-291 F30602-81-C-0224 F/G 9/5 NL




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

RADC-TR-83-291
Final Technical Report
December 1983

ADVANCED APPLICATION OF THE PRINTED CIRCUIT BOARD TESTABILITY DESIGN AND RATING SYSTEM

Boeing Aerospace Company

Richard Spillman and Tom Nicolino

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, NY 13441

This report has been reviewed by the RADC Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-83-291 has been reviewed and is approved for publication.

APPROVED: C. Richard Unkle

C. RICHARD UNKLE
Project Engineer

APPROVED: W. S. Tuthill

W. S. TUTHILL, Colonel, USAF
Chief, Reliability & Compatibility Division

FOR THE COMMANDER: John A. Ritz

JOHN A. RITZ
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (RBET) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document requires that it be returned.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: RADC-TR-83-291	2. GOVT ACCESSION NO: AD-A141147
3. RECIPIENT'S CATALOG NUMBER:	
4. TITLE (and Subtitle): ADVANCED APPLICATION OF THE PRINTED CIRCUIT BOARD TESTABILITY DESIGN AND RATING SYSTEM	5. TYPE OF REPORT & PERIOD COVERED: Final Technical Report 1 Nov 81 - 30 Nov 82
	6. PERFORMING ORG. REPORT NUMBER: N/A
7. AUTHOR(s): Richard Spillman, Tom Nicolino	8. CONTRACT OR GRANT NUMBER(s): F30602-81-C-0224
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Boeing Aerospace Company, PO Box 3999, Seattle WA 98124	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 62702F, 23380269
11. CONTROLLING OFFICE NAME AND ADDRESS: Rome Air Development Center (RBET), Griffiss AFB NY 13441	12. REPORT DATE: December 1983
	13. NUMBER OF PAGES: 118
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office): Same	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: C. Richard Unkle (RBET)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Testability	Controllability
Accessibility	Observability
PCB	Feedback
Electronic Equipment	

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The objectives of this study were to extend the present PCB testability design and rating system developed by Grumman to the module, rack, and subsystem levels and to generate an overall system level testability figure of merit (FOM). The FOM developed is based on the evaluation of a system, modelled as a node-valued weighted graph, (a node being each individual PCB in the system). This model requires as inputs the testability ratings of each individual PCB (outside the system) as a number

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

between 0 and 1. This FOM was validated on 3 of Boeing Aerospaces' own subsystems, each containing a various number of PCB's. The results of the validation and the development of the FOM model are presented. This model has been computerized in the pascal computer language and is provided as part of this report.

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED)

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (CONTINUED)

1.0 INTRODUCTION

The rapid growth in the complexity of digital system designs and the reduction in the size of such systems has increased the scope of the testing problem while reducing the access to interior parts of a system, which is vital to thorough testing. As a result of this advancing technology, system testability has become an important performance and reliability evaluation criterion. As part of the process of quantifying testability, the objectives of this program were to extend the present PCB testability design and rating system developed by Grumman to the module, rack and subsystem levels and to generate an overall system level testability figure of merit. The figure of merit developed during this research effort is highly versatile since it will allow for both the calculation of the testability of an existing system and aid in the top down design of proposed systems.

The remainder of Section 1 will review the need for a testability rating system, outline the research approach, and summarize major results.

1.1 The Need for Testability

The interrelated requirements of high mission effectiveness, high system availability, and low life cycle costs motivates the development of highly reliable and maintainable military electronics. Increasing levels of component integration, the resulting limited access, and the increasing complexity of systems have contributed to the problem of high maintenance costs and low system availability. The ability to adequately test complex electronic systems is a prime requisite for rapid fault isolation and correction. Testability is currently restricted by four major factors.

A. Inadequate Test Equipment

A recent Air Force study (1) concluded that the number one problem for avionic system maintenance is the inadequate fault isolation capabilities of test equipment. Without adequate equipment for fault isolation, maintenance personnel must spend extra time locating and replacing suspected failures. In addition, automatic test equipment (ATE) is subject to its own reliability and maintainability problems.

B. Lack of Design for Testability (DFT)

For most systems, questions of testing and maintenance are considered only after the system design has been frozen. The result is that concurrent (operational) testing becomes difficult-to-impossible and off-line testing becomes more expensive and time consuming because of a less effective test capability. Another result is higher false alarm rates.

C. Inadequate Test Procedures

The move to more reliable systems made up of large scale integration/very large scale integration (LSI/VLSI) components results in higher levels of integration and fewer test points. The classical testing requirements of observability and controllability are severely affected, thus increasing the difficulties of testing systems using these components. In addition, testing problems are increased because of cannot duplicate (CND) faults and black box interface faults that cannot be detected by test procedures. This is a major source of expense in system testing.

D. Complexity and Variety of Equipment

Maintenance tasks must be conducted on a wide variety of complex equipment. Appropriate training to perform these tasks requires a lengthy time investment relative to the average military service period of maintenance personnel.

The result of these factors is that system testing and fault isolation occupy the bulk of maintenance time. Testing and maintenance also represent a high percentage of aircraft support costs and form the limiting factor in military aircraft reliability. Fault isolation is often performed incorrectly and trouble shooting tasks often need to be repeated many times before the isolation process is completed. These testing costs are apparent in the field where fault detection and isolation account for 35% of total maintenance time. In addition, up to 40% of the Line Replaceable Unit (LRU) removals are later determined to be "false alarms" (2). These unnecessary removals increase both maintenance cost and time as well as reduce system availability.

The solution to the electronic system testability problem is the inclusion of adequate test capabilities during each level of the design process.

Overall, design for testability will reduce the following:

- o Engineering redesign costs since fewer designs would be returned because of the lack of testability.
- o Test equipment costs since test equipment need not be as complex.

- o Flow time through production since production testing time would be reduced.

- o Service time in the field since testing would require less time.

- o The number of field failures since there would be fewer latent faults.

Part of this savings would be realized to some degree from any use of DFT rules. The existence of a quantitative measure of testability used concurrently with the DFT rules would have additional advantages. It would provide the following:

- o Help designers choose among alternative designs.

- o Establish standards of testability compliance for the construction of new equipment.

- o Locate testability problem areas within a design before the final product is produced.

1.2 Testability Ratings

Designing for testability involves the careful consideration of three concepts: controllability, observability, and predictability. Controllability refers to the ease in which a system can be directed through its various functions. Observability refers to the ease in which the internal status of

the system can be examined. Predictability is related to controllability and refers to the ease in which a system can be forced into a known state. From these concepts, general design rules can be developed to improve system testability. For example, design rules which enhance circuit predictability require the development of synchronizing digital and analog signal sequences in all systems. Controllability and observability may be enhanced through the use of test points. These test points may provide additional system inputs for control purposes or additional circuit outputs for observation purposes.

These general design rules have led to a variety of specific design implementations. For example, the well known level-sensitive scan design (LSSD) is a specific DFT approach. The method uses shift register latches as sequential circuit storage elements. These latches are threaded to form a shift register allowing all the sequential elements of the circuit to be tested by simply shifting test patterns through the registers. Having done this, the resulting problem is reduced to the more straightforward problem of testing the remaining combinational elements.

In addition to highly structured design procedures such as LSSD, a large number of ad hoc design rules also exist which may improve testability. Most of these design rules operate to increase one of the three basic testability concepts. However, a major problem facing testability researchers is how to evaluate the results of applying these rules. A numerical rating which can then guide both the evaluation and design of large systems is an ideal approach. Several testability measures of this type have been proposed. Dejka (3) suggested a testability measure based on several circuit characteristics including: size, number of I/O pins, and the number of test

vectors or patterns (a test vector or pattern is an applied input condition which can be used to detect fault conditions). Stephenson and Grason (4) developed a testability rating based on controllability and observability characteristics. Their rating is limited to the register transfer level representation of the circuit. These and other existing testability measures suffer from several problems including: (1) most are based on ad hoc procedures, (2) they apply only at the chip or printed circuit board (PCB) level so the special system level testability problems introduced by cables, connectors, etc., are not considered, and (3) they lack general validation since they have been applied to only one or two systems.

Another testability rating system was recently developed by Grumman Aerospace (5). The system rates PCBs on four positive basic testability factors and 30 negative testability factors. The overall PCB testability rating is determined by subtracting the negative factors score from the positive factors score. This rating was validated by applying the procedure to 17 PCBs and comparing the resulting testability score to the independent evaluation of testability made by several expert engineers. This measure, which has generated a great deal of interest and significant use in industry, served as the takeoff point for development of the system level figure of merit (FOM) proposed in this report.

1.3 Research Approach

Preliminary research and review of the technical literature led to the conclusion that the most logical approach for deriving a system level FOM would be based on a signal flow graph model.

The approach to extending the PCB testability FOM to the system level thus involved four tasks. The first task consisted of a survey of Boeing equipment and selection of a set of small digital systems (consisting of multiple PCB's) for use in the FOM verification/validation task. Once the equipment was selected, the Grumman testability FOM was calculated for each PCB in three systems. The second task required the development of guidelines for creating a node-valued weighted graph model to determine testability of items comprised of electronic circuits/functions. This model would be used to extend the PCB testability FOM to the subsystem/system levels (i.e., "black box"). The third task involved the verification and validation of the system testability FOM. Here the model was applied to each system selected in task one. The testability FOM was then compared with actual testing experience in order to validate or suggest changes in the model. Finally, in task four, a higher level system application of the testability FOM was made to serve as a guide to the application of the testability measure at higher levels (where a system consists of subsystems, or "black boxes", each containing multiple PCB's).

1.4 Summary of Results

Several different mathematical approaches to the problem of testability evaluation (using different types of nodal modeling techniques) were explored. The method finally selected, the Accessibility Model, has the following advantages:

1. It works on both analog and digital circuits, provided each component's testability and interaction with other components are known.

2. It works on feedback loops.

3. It allows for the decomposition of complex systems into easy to solve subsystems.

This model was applied to three Boeing systems and produced testability FOM's which were in general agreement with the observed testability experience on the systems.

2.0 TESTABILITY SURVEY

In general, there are two methods of determining system testability: the scorecard method and an algorithmic method. Scorecard methods involve the analysis of circuit features which enhance or detract from testability. Each feature is weighted and the weights are combined to form a testability factor for the circuit. Dejka (3), for example, suggested a testability measure based on several circuit characteristics including: size, number of I/O pins, and the number of test vectors (patterns). Grumman Aerospace (5) developed a PCB testability score for the Rome Air Development Center (RADC). This system rates PCB's on four positive and 30 negative testability factors. The overall PCB testability score is determined by the difference between the negative factors and the positive factors. Grumman validated the procedure on a sample of 17 PCB's by comparing the PCB testability score to an independent evaluation of testability made by expert engineers. Scorecard systems are easily implemented and are tied directly to the circuit design. However, they provide a very coarse measure of testability and cannot be manipulated to study the nature of testability. In addition, scorecard systems do not allow development of a rigorous model for calculating system level testability taking into account the influence of testability.

Algorithmic methods provide a mathematical approach to testability calculation based on an analysis of the topological structure of the circuit. Goldstein (6), for example, describes the testability of a circuit by assigning six values to each circuit element: combinational 0 and 1 controllabilities, sequential 0 and 1 controllabilities and both combinational and sequential observabilities. These values are propagated across the devices in the

circuit and are used as a measure of circuit testability. Dussault (7) analyzes circuit testability based on information theory. Controllability and observability are defined as circuit entropies. Breuer and Friedman (8) using TEST/80 define a cost function which reflects the cost of testing along a given path in the circuit. Based on a D-algorithm like procedure, TEST/80 determines both controllability and observability costs which may be used to determine system testability. British Telecom has also developed a measure of testability called CAMELOT (computer-aided measure of testability) (9). CAMELOT assigns a controllability and an observability value to each circuit node and propagates these values through the circuit using controllability and observability transfer functions which are defined for each circuit element. The testability of a node is determined by the product of the node's controllability and observability. The overall system testability is the average of the testabilities of the circuit nodes. Algorithmic measures of testability are very difficult to calculate for large systems since they require extensive manipulation of transfer functions at the gate level. In addition, they do not respond to the specific details of the circuit implementation which may influence overall testability.

In the following subsections several current methods for testability evaluation that were investigated are reviewed for applicability to this contract and the strengths and weaknesses of the resulting testability figures of merit are evaluated.

2.1 SCOAP

Goldstein developed one of the first testability FOM'S in 1979 (6). His approach involves the analysis of two quantities: a controllability value and an observability value. Six measures are used to characterize the controllability and observability of a circuit:

- o Combinational 0 controllability
- o Combinational 1 controllability
- o Sequential 0 controllability
- o Sequential 1 controllability
- o Combinational observability
- o Sequential observability

SCOAP testability analysis of a circuit assigns a value to each of the six controllability/observability qualities for each circuit element. The values are determined for each device by the direct analysis of the controllability and observability of the input lines and the calculation of the controllability and observability of the output lines.

SCOAP has been programmed and runs efficiently even for large systems. Run times of only 3-4 min for 200 element circuits have been reported. However, SCOAP does not produce a single FOM for overall testability. Goldstein suggests two possible summary analysis methods. First, construct a testability profile. This is a graph of the controllability or observability vs. the number of nodes. The profile forms a visual summary of the overall testability. Second, Goldstein suggests using either the maximum

controllability or the maximum observability value as a summary FOM. Another major drawback of this method is it does not evaluate analog circuits.

2.2 Hybrid Method

Longendorfer developed a testability FOM originally for analog circuits and later extended it to digital circuits (10). Her method involves several steps, the first of which requires reducing the circuit to an equivalent signal flow graph. As a result, circuit elements at the same level are lumped together in the graph as a circuit block. A testability measure for each block is determined by equation 1.

$$t_j = (1 + \log S_j)\, C_j \tag{1}$$

where S_j = sequential level of the block j

C_j = number of nodes in block j

The larger t_j, the less testable the block. The overall testability measure is given by equation 2.

$$T = \left(\frac{PB}{\sum_{j=1}^{B} t_j} \right)^{\frac{1}{2}} \tag{2}$$

Where P = fraction of block inputs or outputs directly driven or sensed during test

B = number of blocks in the circuit

Longendorfer's method is sensitive to controllability, but insensitive to observability. In addition, feedback loops are grouped together into a single block so the method does not respond well to sequential circuits.

2.3 CAMELOT

British Telecom currently uses a testability assessment program called CAMELOT (9). CAMELOT defines a controllability, (CY), and observability, (OY), for each circuit element. The controllability value for a circuit element (or node) is determined by its output controllability given by equation 3.

$$CY\ (output) = CTF \times f\ (CY\ (inputs)) \tag{3}$$

where

CTF = a controllability transfer factor

CY = 1 for primary input

CY = 1 for other nodes

f = a function of input controllabilities

The function, f, is a modified arithmetic mean of the input CY's. The observability, (OY), of a circuit element is determined by its output observability:

$$OY\ (output) = OTF \times OY(inputs) \times g\ (CY\ (inputs)) \tag{4}$$

where

OTF = an observability transfer factor

OY = 1 for primary inputs

OY = 1 for other nodes

g = a function of the input controllabilities

The function, g, is similar to f in that it is a modified arithmetic mean of the input CYs.

A system level measure of testability is determined by first assigning a testability value (TY) to each circuit element:

$$TY_{node} = CY_{node} \times OY_{node} \quad (5)$$

Then, the overall testability is the average of the node testabilities for the circuit.

This method requires a large amount of computational effort for large circuits and using equation 5 tends to underestimate the actual node testabilities.

2.4 A Functional Level Calculus of Testability

Nippon Electric Company has developed a testability FOM based on Goldstein's controllability/observability measures (11). The method requires a gate level description of the network and utilizes the following definitions:

0 - Controllability

The number of circuit elements which need to be set to a known value for obtaining a logical 0 at the specified element.

1 - Controllability

The number of circuit elements which need to be set to a known value for obtaining a logical 1 at the specified element.

Observability

The number of circuit elements which need to be set to a known value for propagating a logical value of a specified element to primary outputs.

Given a circuit at the gate level, the 0-controllability, 1-controllability and observability of each gate may be calculated by the direct application of the definitions. In general, in collections of gates from functional blocks (i.e., chips) the controllability of a functional block is given by equation 6:

$$C(FB) = \left[(W_a \cdot A(FB))^2 + (W_b \cdot B(FB))^2 \right]^{\frac{1}{2}} \quad (6)$$

where

$$A(FB) = \sum_i \frac{(WO_i \cdot \text{0-Controllability})}{\text{Number of pins}}$$

$$B(FB) = \sum_i \frac{(W1_i \cdot \text{1-Controllability})}{\text{Number of pins}}$$

W_a, W_b, WO_i, $W1_i$ = weighting factors

The summation is over the inputs and outputs of the functional blocks. The observability of the functional block is given by equation 7:

$$O(FB) = \sum_i \frac{(WO_i \cdot \text{Observability})}{\text{Number of pins}} \quad (7)$$

where WO_i = a weighting factor

The sum is over the inputs and outputs of the functional block. Finally, the overall testability of the circuit is given by equation 8:

$$T(FB) = \left[(W_C \cdot C(FB))^2 + (W_O \cdot O(FB))^2 \right]^{\frac{1}{2}} \quad (8)$$

where W_C and W_O = weighting factors

Clearly, this method has so many weighting factors that it is questionable as to whether the engineer's ability to choose weighting factors or the testability is really being measured.

2.5 TESTSCREEN

TESTSCREEN, developed at the Sperry Research Center, is an attempt to use Goldstein's Controllability/Observability values along with information on network size and the number of input/output pins to determine testability (12). The method begins by applying SCOAP to determine the six C/O values: CC0, CC1, SC0, SC1, CO, SO. For a combinational circuit, TESTSCREEN has a five step procedure for calculation of overall testability:

(1) Find the number of reconvergent fanouts and feedback loops (RAF):

RAF = FANOUTS - NODES + OUTPUTS

(2) Find the number of representative faults

FAULTS = 2 X (INPUTS + RAF)

(3) Find the circuit size

$$NEWSIZE = \frac{FAULTS}{K}$$

where K = a constant (Sperry uses 2.05)

(4) Calculate the average controllability/observability:

$$\frac{CC0}{NODES}, \frac{CC1}{NODES}, \frac{CO}{NODES}, \frac{SC0}{NODES}, \frac{SC1}{NODES}, \frac{CO}{NODES}, \frac{SO}{NODES}$$

(5) Calculate testability

$$T = NEWSIZE + SCALE \left[\frac{CO}{NODES} + MAX \left\{ \frac{CC0}{NODES}, \frac{CC1}{NODES} - 2 \right\} \right]$$

where SCALE = a constant

The major problem with this approach is the need for a choice of constants, although Sperry has found values that work well for them.

2.6 Other Approaches

In addition to the five measures reviewed there are a wide range of other approaches in various stages of development. For example, Breuer and Friedman developed a program called TEST/80 (8). This is actually an automatic test generation program for digital circuits. However, it is based on a cost analysis process which is in effect a controllability/observability analysis. Hence, their method could be used to generate a testability FOM.

2.7 Conclusions

Most testability measures suffer from a series of problems that include:

- Lack of Development

  Many measures have been proposed and procedures for their calculation have been sketched out but they have never been developed to a point where they are useful.

- Theoretically Unsound

  Most testability measures are ad hoc procedures with no real theoretical basis which ties them directly to the concept of testability.

- Computationally Intensive

  Some testability measures require a great deal of computational effort.

- Validity Problems

  Few testability measures have been completely tested to determine their relationship to actual testing costs. In fact, many measures rely on so many arbitrary constants that it is difficult to determine any real relationship between testability and the measure.

- Unable to Guide the Design Process

  Most testability measures are after-the-fact measures that require most of the design effort to be completed before testability can be evaluated. As a result, the testability measure cannot be used to help the designer improve the testability of the circuit during the design phase.

All of the approaches considered suffered from these problems to some degree. Longendorfer's Hybrid Method was the only method that could be used during the design process, and it also easily handles analog circuits. As a result, modification of her approach toward the specific objectives of this effort was attempted. Even with modification, though, it could not be made sensitive

to feedback and observability. Based on this experience, a new concept, "accessibility", was defined and the hybrid model restructured to introduce the concept of accessibility of a node. This resolved the feedback and observability shortcomings of the original model. A full description of the Accessibility Model and the development process that lead to it is given in section 4.0, The Testability Model.

3.0 THE GRUMMAN PCB TESTABILITY FIGURE OF MERIT

The Grumman PCB Testability Figure of Merit System was developed by Grumman Aerospace for RADC as a means of evaluating digital PCB testability by inspection (5). As mentioned in section 2.0 it is a scorecard system which rates a PCB in 4 positive and 30 negative factors. These factors are tallied to produce a score in the range of -infinity to +100. Table 3-1 shows the relationship between Grumman scores and testability. Experience indicates that scores typically fall in a range of -80 to +90. It should be noted that the Grumman system has no provisions for analog circuits, so the testability FOM applies only to digital PCBs or the digital portion of a PCB. The column labeled "Transformed Grumman Score" is a linear transformation of the Grumman scores onto a 0.0 to 1.0 scale. This transformed notation is used for initial card (node) testabilities in the models and examples of sections 4.0 and 5.0.

The four basic positive factors are evaluations of percent of nodes accessible, proper documentation, percent sequential circuits used, and a PCB complexity count. Two of these factors, percentages of nodes accessible and sequential circuits, are closely tied to the testability concepts of controllability, observability and predictability. The other two, documentation and PCB complexity, deserve further explanation.

Proper documentation in the form of schematics, parts lists, board layouts, and detailed I/O signal specifications are felt to improve testability simply by making the test engineer's task easier. This reduces test development time and hence cost. The PCB complexity count considers the type and number of

GRUMMAN SCORE	TRANSFORMED GRUMMAN SCORE	ACTUAL TESTABILITY
+81 to 100	.91 to 1.0	Very Easy
+66 to 80	.83 to .90	Easy
+46 to 65	.73 to .82	Medium/Easy
+31 to 45	.66 to .72	Medium
+11 to 30	.56 to .65	Hard
+1 to 10	.51 to .55	Very Hard
Less than 1	less than .51	Impossible to test without extreme cost penalty

$$\text{transformed Grumman Score}^{*} = \frac{G + 100}{200}$$

where G = Grumman Score when Grumman Score > -100

G = -100 when Grumman Score $\leq$ -100

* Discussed in Section 5.0

Table 3-1 Relationship Between Grumman Score and Testability

sequential circuits. It differs from the percent sequential factor in that it considers overall board size and component types. Small Scale Integration (SSI), Medium Scale Integration (MSI) and "simple" LSI integrated circuits score the best while the use of VLSI circuits (especially memory chips and microprocessors) is heavily penalized because of the large number of inaccessible nodes internal to the chip.

The four positive factors produce a maximum positive score of 100.

The 30 negative factors include such items as use of monostable circuits, buried memory, non-initializable sequential circuits, and VLSI. Other factors address the areas of required test equipment and documentation. Each of the negative factors is weighted in accordance with its impact on cost and difficulty of testing. The maximum negative score is infinite. On a practical basis though, any net negative score indicates that a card is probably difficult to test and a large negative score (e.g., -60) indicates it probably will take much effort (and cost) to adequately test the card.

3.1 Application to BAC Systems

Several Boeing designed and built systems were surveyed for inclusion in this study. The three that were chosen were the Roland Command/Control Computer, the ASAT Flight Control Electronics (FCE), and the 767 Flap/Slat Electronics Unit (FSEU). These systems were chosen to represent typical large, medium, and small complexity aerospace systems. The Grumman technique was applied to the PCBs of each of these systems.

In addition, test engineers responsible for the test development and testing of these PCBs were interviewed to determine their testability ratings by experience. The results of this work, tabulated in tables 3-2, 3-3, and 3-4, are discussed in section 3.3. Both transformed Grumman scores and Boeing test experience transformed into testability scores were used later in our testability model to determine system level testability.

3.2 Summary of Grumman Problem Areas

Our experience indicates that the Grumman testability figure of merit system is quite accurate in rating the testability of simple digital circuit boards. However, it was also discovered that when applied to more complex circuits the accuracy of the technique is diminished. This is demonstrated by the consistently low scores generated by PCBs which use microprocessors, VLSI, or large quantities of memory. The Grumman technique penalizes the mere existence of these devices with little or no regard to their actual testability in the circuit. With the proliferation of microprocessor based systems, the Grumman scores become compressed toward the low end of the scale. Also, many of the PCBs evaluated had analog circuits on them. Since the Grumman system has no provisions for analog, these portions of the boards were ignored in this study.

The Grumman PCB FOM still has merit because of its simplicity and ease of use. Because of this, it should be expanded to be able to distinguish between easily testable and difficult to test microprocessor, VLSI, and memory circuits, as well as analog circuits.

CARD	GRUMMAN	BOEING TEST EXPERIENCE *
Ignition Discrete	31.1 Medium	Very Easy (90.5)
Servo/Amp/BIT	10.2 Very Hard	Medium/Hard (30.5)
Gyro Wheel Supply	19.4 Hard	Hard (20.5)
Serial Data	-27.8 Impossible	Medium (38.0)
Gyro Demod/ Excitation	-10.8 Impossible	Hard (20.5)

* Median Value

ASAT Flight Control Electronics Testability

Table 3-2

CARD	GRUMMAN	BOEING TEST EXPERIENCE *
DISCRETE I/O	63.2 Medium/Easy	Medium/Easy (55.5)
CPU	-0.5 Impossible	Hard/Medium (30.5)
AUX	11.2 Hard	Medium/Easy (55.5)

* Median Value

Table 3-3 767 Flap/Slat Electronics Unit

CARD	GRUMMAN	BOEING TEST EXPERIENCE *
CPU	.9 Impossible	Medium (38.0)
Sequencer	-82.9 Impossible	Impossible/Hard (10.5)
Microcode Adapter	81.8 Very Easy	Easy (73.0)
1K RAM/4K EPROM	-71.2 Impossible	Easy (73.0)
PIM/RTC	49.7 Medium/Easy	Medium/Easy (55.5)
Digital I/O #1	80 Easy	Easy (73.0)
Digital I/O #2	17 Hard	Hard (20.5)
DMA I/O	20.8 Hard	Hard (20.5)

* Median Value

Roland Command/Control Computer Testability
Table 3-4

3.3 Comparison of Grumman PCB Scores to Actual Boeing PCB Test Experience

3.3.1 ASAT Flight Control Electronics

Table 3-2 shows the ASAT Flight Control Electronics PCB testabilities. The Grumman scores showed a general trend of rating each PCB as being somewhat more difficult to test than Boeing experience indicates. The exception to this is the serial data board which Boeing test personnel rate as medium while the Grumman method rates as impossible to test without high cost. Using the Grumman figure of merit, points were deducted for a high percentage of non-initializable sequential circuits. Boeing test personnel agreed with this, however, patterns have been developed which provide an initialized state for these circuits making adequate testing possible. Many points were also deducted for use of memory circuits with one or more non-directly accessible lines. Simple patterns have been developed to access each memory location making testing much easier than predicted by the Grumman technique. Because many of the PCBs in the ASAT system have a high percentage of analog circuitry (which is not addressed by the Grumman system), only five of the ASAT boards were included in this evaluation. These cards have not been production tested since the unit is still in engineering development; however, an engineering model has been built. System testability of this unit is calculated and discussed in section 4.3.1.

3.3.2 767 Flap/Slat Electronics Unit

Table 3-3 shows the 767 Flap/Slat Electronics Unit testabilities. At the time of the initial interview, Boeing test personnel felt the Grumman PCB scores

were quite accurate. However, since that time improved test methods have been developed resulting in better testability scores for the CPU and Aux I/O cards. This improvement was realized through advanced ATE techniques. Boeing engineers test the assembled unit using ATE also. System testability calculations and discussion for the Flap/Slat Electronics Unit is in section 4.3.2.

3.3.3 Roland Command/Control Computer

Table 3-4 shows the Roland Command/Control Computer PCB testabilities. The majority of the PCBs were found to have actual testabilities which correlated quite closely with the Grumman scores. The exceptions are the Central Processor Unit (CPU) and the memory boards. The CPU card contains four four-bit microprocessors in a bit slice arrangement to produce a 16 bit CPU. The Grumman testability measure deducts points for each of these four microprocessors. Interviews with key test personnel indicate that if these four microprocessors are approached functionally as a single 16-bit CPU then adequate testing is not difficult. The 1K RAM/4K EPROM memory board contains 12 separate memory devices. Because each of these devices has at least one line which is not brought out to the edge connector they fall into the Grumman category of buried memory devices and are heavily penalized. In actuality, only simple patterns are needed to access the contents of each memory location through the edge connector. Calculations and discussion of system testability for this unit is in section 4.3.3.

4.0 THE TESTABILITY MODEL

The general approach selected for the development of the testability model was to represent a circuit as a node-valued weighted graph. The nodes represent circuit substructure (e.g., a PCB) for which a testability FOM exists. For example, Figure 4.0-1 illustrates a simple node-valued weighted model of a circuit. An arc is the link between two nodes. The weights on the arcs are given by W_{ij}, which represents the influence of the connection between node i and node j. The node testability represents the testability FOM for node i. In general, there are two types of node testability: (1) Network Dependent Node Testability (NDNT), t_i, which is the testability of the node when it is embedded in the network and (2) Isolated Node Testability (INT), T_i, which is the testability of the node outside the network. A subsystem may have a high isolated node testability rating and a very low network dependent node testability if, for example, it is embedded deep in the network structure. In general, it can be noted that,

$$t_i \leq T_i$$

If the nodes are PC boards then the isolated node testability could be the Grumman PCB testability FOM.

In terms of this graph model the steps associated with the development of a system level testability rating given the INT's and the network structure are:

(1) Calculate the NDNTs and arc weights

(2) Calculate the system FOM

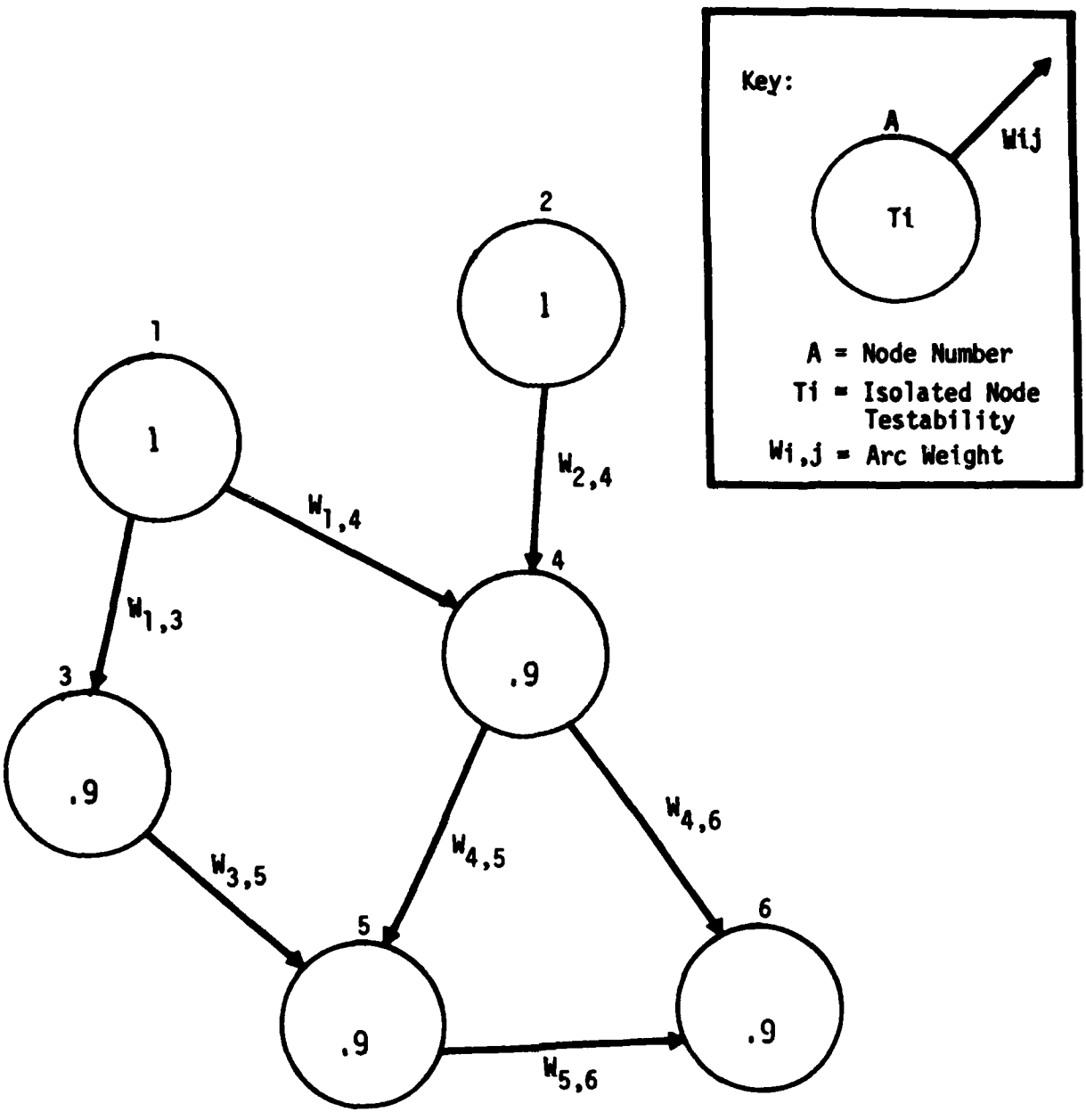


Figure 4.0-1: Simple Node Valued Weighted Model

Several approaches were taken to develop calculation procedures for the graph model. These approaches are summarized in 4.1. The final model is developed in 4.2. Examples of the application of the model are given in 4.3, while a verification of the model is described in 4.4.

4.1 The Model Development Process

Initially, three candidate approaches for the testability model were suggested:

- o Inverse Signal Reliability
- o Pulse Process Analysis
- o Modified Hybrid Systems Analysis

The Inverse Signal Reliability (ISR) method was rejected for development as part of this project since it would not lend itself to incorporating the Grumman PCB testability measure. However, the other two methods were evaluated, and are reviewed in the following two sections.

4.1.1 Pulse Process Method

The pulse process analysis starts with a signal flow graph such as the one shown in Figure 4.1.1-1. The node values are the isolated node testabilities.

The goal of a pulse process analysis is to find the network dependent node testabilities by searching for a stable condition in equation (9).

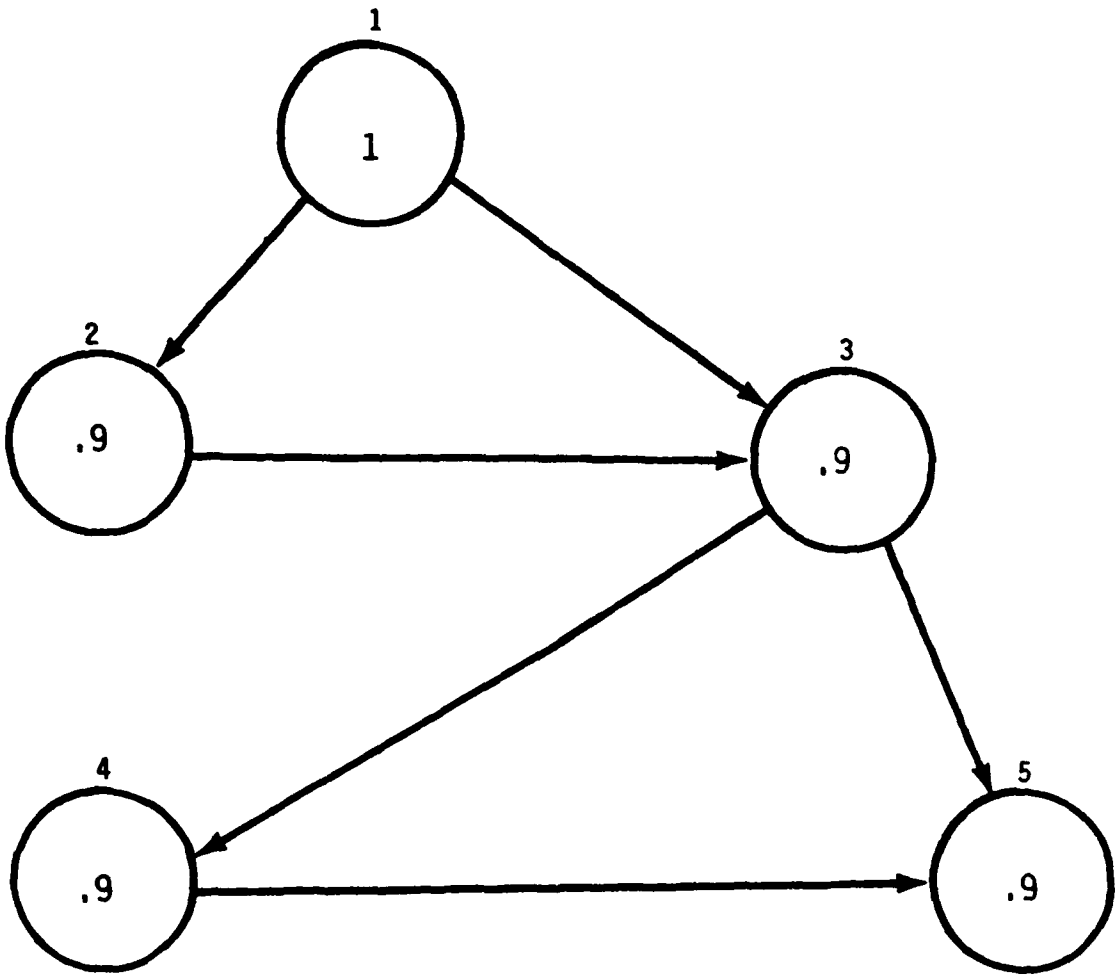


Figure 4.1.1-1: General Signal Flow Graph

$$T_i(K+1) = T_i(K) + \sum_{j=1}^{n} A_{ji}\, P_j(K) \qquad (9)$$

where $T_i(K)$ = the stability criteria at stage K

A = the weighted adjacency matrix

i = the node in question

j = each of the other nodes in the system

$P_j(K)$ = a vector of length n representing the n nodes in the graph where:

$P_j(0) = 1$ if j is an input node

$Pj(0) = 0$ otherwise

It turns out that:

$$P_i(K+1) = \sum_{j=1}^{n} A_{ji}\, P_j(K) \qquad (10)$$

for each i and j = 1, 2, 3...n

Rewriting (10) in matrix terms results in equation (11).

$$P(K) = P(0)\, A^K \qquad (11)$$

The result is that equation (9) becomes the following matrix relationship:

$$T(K+1) = T(K) + P(0)A^K \qquad (12)$$

where T (K) = the initial testabilities vector

A stable condition is reached when $T(K+1) - T(K) \leq E$ for some small $E \geq 0$ at stage K. The new values of T are transformed into the network dependent node testabilities using equation 13.

$$t_i = \left(\frac{1}{T_i(K)} \right)^{\frac{1}{2}} \tag{13}$$

These must be combined to form a single system level measure of testability. One possibility would be to average the network dependent node testabilities. However, a formal method has not been developed.

For the example system in Figure 4.1.1-1, the weighted adjacency matrix is:

$$A = \begin{array}{c} \\ \text{NODE FROM} \\ \\ \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & .6 & .5 \\ 0 & 0 & 0 & 0 & .9 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \begin{array}{c} \\ \text{NODE TO} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}$$

where each element, A_{ij}, is the weight of the connection from node i to node j if there is one.

Given nodes 1 and 2 are inputs, the vector P is given by:

$$P(0) = (1 \quad 1 \quad 0 \quad 0 \quad 0)$$

Using (10) and T(0), the initial testability vector, a stable condition is reached at k = 2. That is, T(4) = T(3) = (1.0 2.0 3.9 2.7 2.4). Using (13), the network dependent node testabilities for this system are:

(1 .71 .51 .61 .65)

Using an averaging method over the network dependent node testabilities the overall system level testability is .70.

While the pulse process method produces some interesting results, it was decided that too much additional research and development was required in order to satisfactorily apply this method. This decision was due, in part, to the fact that the Hybrid Method (see next section) has a more sound theoretical base to tie it directly to the concept of testability.

4.1.2 Hybrid Method

The second approach was a modification of Longendorfer's Hybrid Analysis Method (10). The modified hybrid testability FOM is based on a graphical analysis of the circuit structure. This method combines circuit elements to form interrelated "nodes" so it does not depend on the function of a circuit. As a result, it applies to both analog and digital systems. In the hybrid analysis method the testability FOM is related to the controllability and observability of all the circuit elements. The key concept is that the deeper an element is buried in a circuit the more difficult it is to control and/or observe the operation of that element. In other words, the longer the path from the input to the element or from the element to the output, the lower the testability of the element.

In general the hybrid algorithm requires as inputs:

(a) A signal flow graph of the circuit in the form of a connectivity matrix for the elements.

(b) The initial testability of the circuit elements stored as a vector.

Using these inputs, the algorithm proceeds through a series of five stages leading to a system level testability FOM. These stages are:

(1) Group the indistinguishable elements of the circuit into common nodes (called blocks). This is done by calculating a reachability matrix for the signal flow graph using a procedure developed by Ramamoorthy in 1971 for graph analysis. The reachability matrix is then analyzed to locate elements with indistinguishable signal flow paths. These elements are grouped into blocks and a new connectivity matrix is constructed.

(2) A new initial testability assignment is given to each block which is the average of the initial testabilities of the circuit elements in each block.

(3) The depth of each block in the circuit is determined by examining the new connectivity and reachability matrices. This gives the arc weights, w.

(4) The network dependent testability values are calculated for each block. They are given by the initial testability divided by 1 + ln w. Hence, the testability of elements decreases as their depth in the circuit increases.

(5) A system level figure of merit, T, is determined based on the network dependent testability values and the circuit structure using Longendorfer's methodology. That is,

$$T = \left(\frac{PB}{\sum_{i=1}^{B} S_i t_i} \right)^{\frac{1}{2}}$$

where

P = The percentage of direct I/O nodes in the original circuit

B = The number of blocks in the circuit (found in step 1)

t_i = The network dependent testability value of block i

S_i = Level of block i

For example, given the network shown in Figure 4.1.1-1, the connectivity matrix is:

$$C = \begin{array}{c} \\ \text{Node} \\ \\ \text{FROM (i)} \\ \\ \\ \end{array} \begin{array}{c} \\ \\ \text{To(j)} \\ \\ \\ \\ \end{array} \begin{array}{c} \\ \underline{1} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \\ \underline{2} \\ \\ \\ \\ \\ \\ \end{array} \begin{array}{c} \\ \underline{3} \\ \\ \\ \\ \\ \\ \end{array} \begin{array}{c} \\ \underline{4} \\ \\ \\ \\ \\ \\ \end{array} \begin{array}{c} \\ \underline{5} \\ \\ \\ \\ \\ \\ \end{array} \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

where: C_{ij} = 1 if node i connects to node j

= 0 otherwise

Using C as input along with the initial testability values, the graph reduces to a two node graph as shown in Figure 4.1.3-1. The first node consists of nodes 1, 2, 3, and 4 in the original graph and the second node is node 5 of the original graph. The initial testabilities and NDNT (from step 4, p. 36) for the reduced graph are:

INITIAL	NDNT
1': .925	1': .925
2': .9	2': .53

and the system level testability is .63 (from equation (5), p. 37) which, it should be noted, is close to the .7 value found for the same system using the pulse process method.

4.2 The Accessibility Model

After an extensive analysis of the modified hybrid model it was determined that it had two problems:

(1) It was totally insensitive to feedback.
(2) It was partially insensitive to observability.

That is, given two circuits A and B with identical nodes and forward interconnections yet allowing A to have a feedback loop, the modified hybrid method would assign the same testability values to both circuits. However, it is well known that feedback lines introduce testing problems and as a result

Figure 4.1.3-1: Reduction of Figure 4.1.1-1

such circuits are not as easy to test as the same circuit without feedback. In addition, the testability of a circuit evaluated using the modified hybrid method would only increase slightly with the addition of an observation point while it would increase substantially with the addition of a new control point. This imbalance between the effects of observation points and the effects of control points on system testability does not reflect the real world of testing and actual testing costs.

As a result of the two problems noted above, the modified hybrid model was restructured introducing the concept of accessibility of a node. The accessibility of a node is determined by the accessibility of all the nodes connected to it in a signal path from the input to the output in the following way:

(1) Node N_{ij} is at level i and is node number j. The level is the number of nodes in the longest path from an input to the node, plus 1.

(2) The accessibility (A_{ij}) of all nodes at level 1 (input nodes) is equal to 1.

$$A_{1j} = 1$$

(3) The accessibility of an interior node N_{kl} is the sum of the products of:

(a) The accessibility (A_{ij}) of node N_{ij} in a line segment connected directly to N_{kl} .

(b) The arc weight of node N_{kl} given by Q_{kl}.

$$Q_{kl} = \frac{1}{d_{in}\ (N_{kl})}$$

where $d_{in}\ (N_{kl})$ = the number of arcs entering node N_{kl} for $k \neq 1$

(c) The initial testability (T_{ij}) of node N_{ij}.

In general, the accessibility of an interior node is given by equation 14:

$$A_{kl} = \sum_{ij} A_{ij}\ Q_{kl}\ T_{ij} \tag{14}$$

The sum in equation 14 is over all nodes that are the source of an input arc to node N_{kl} . For example, given the system shown in Figure 4.2-1, the number inside the node is its initial testability (T_{ij}) and the accessibilities (A_{ij}) of nodes 1 and 2 are both 1 since they are level 1 input nodes. In a strict formation of the nodal graph structure, a node of initial testability of 1 is added to indicate input or output to the system. (These nodes are not physical hardware.) For example, node 2 in Figure 4.2-1 indicates that node 4 has inputs from outside the system. Likewise, node 7 shows outputs from node 5 to outside the system. These could be considered controllability and observability points of the system. However, if a node is pure input or output this is not necessary. For example, node 3 of Figure 4.2-1 is pure input (all system arcs lead away from it). In this case, node 1 could be eliminated with no affect on the accessibility and testability calculations.

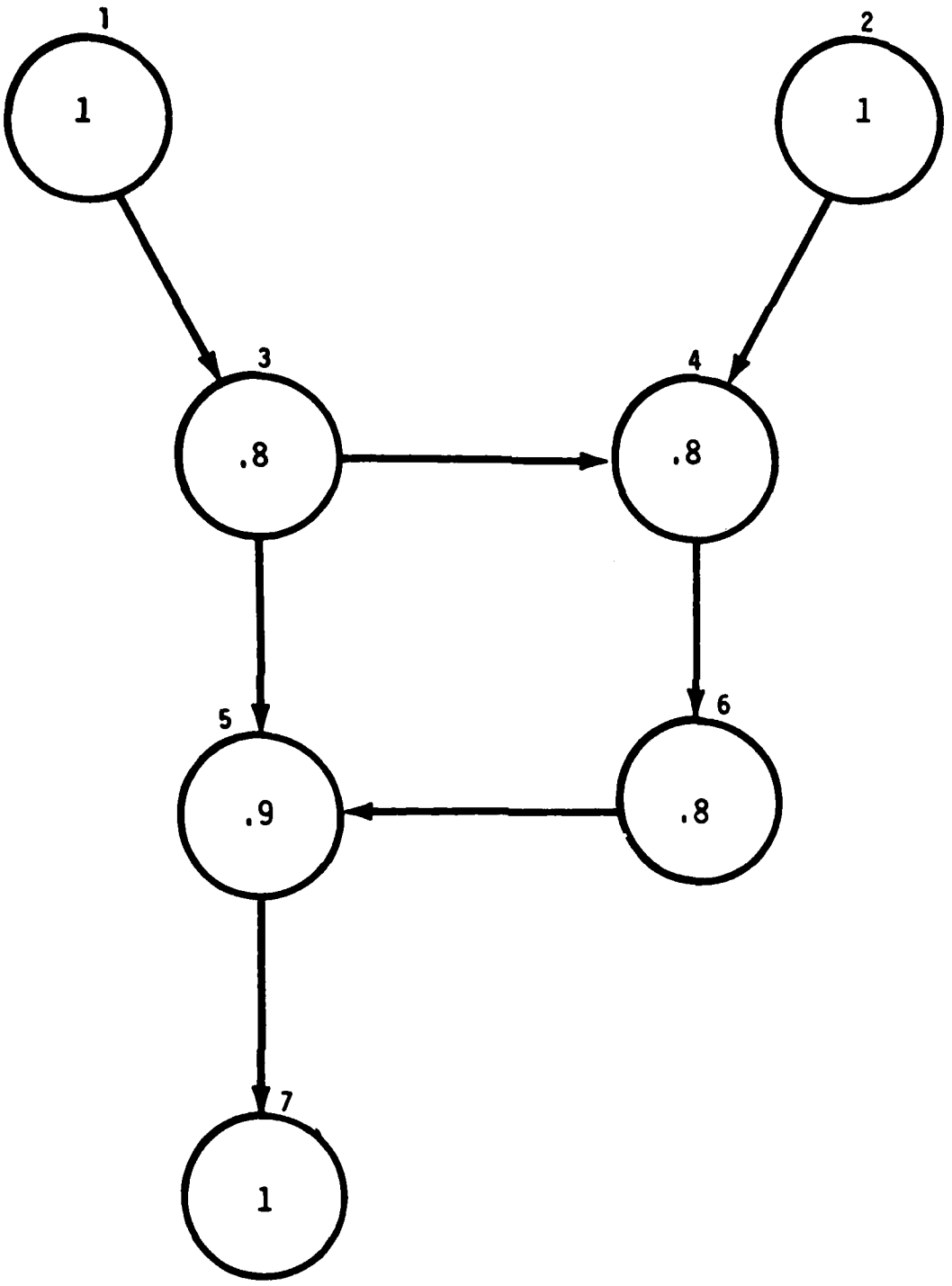


Figure 4.2-1 Example Nodal System

The accessibility of node 3 is given by the following:

$$A_{23} = A_{11} \; Q_{23} \; T_{11} = 1 \times 1 \times 1 = 1$$

The accessibility of node 5 which is at level 5 depends on the accessibilities of nodes 3 and 6 which are connected to node 5:

$$A_{55} = A_{23} \; Q_{55} \; T_{23} + A_{46} \; Q_{55} \; T_{46}$$
$$= 1 \times .5 \times .8 + .72 \times .5 \times .8 = .69$$

The network dependent node testabilities (t_{ij}) are determined by the product of the accessibility of a node and its initial testability.

$$t_{ij} = A_{ij} \; T_{ij}$$

The system level testability calculation method is a three step process:

(a) Find all the paths from the inputs to the outputs.

(b) Calculate a path testability, T_p, for each path using the following:

$$T_p = \left(\frac{1}{M_p} \sum_{n=1}^{M_p} \frac{1}{t_n} \right)^{-1}$$

where M_p = number of nodes in path

t_n = network dependent node testability for each node along path.

(c) Calculate the system testability T_s which is the weighted average of the path testabilities:

$$T_s = \left(\frac{1}{M} \sum_{i=1}^{M} (T_p)_i^{-1} \right)^{-1}$$

M = number of paths

The system testability for the example of Figure 4.2-1 is .70. The network dependent node testabilities (NDNT) and path testabilities are shown in Figure 4.2-2 and Figure 4.2-3.

Another example of the application of this approach is illustrated using the system shown in Figure 4.2-4. The accessibility and NDNT of each node is shown in Figure 4.2-5. The overall testability of Figure 4.2-4 is .62. After an examination of Figure 4.2-4, it is clear that node 6 with an initial testability of .6 and an NDNT of .43 is a problem. One way to increase the system testability then, would be to redesign node 6 such that its initial testability is raised to, for example, .9. The effect on the circuit is shown in Figure 4.2-6. The system testability is raised to .67, almost a 10% increase.

An example of what might be done when the initial testability of a PCB can't be increased for one reason or another is illustrated by Figure 4.2-7, which shows a more complicated system made up of 18 nodes representing PCB's. The results of a testability analysis of this circuit is shown in Figure 4.2-8. The system level testability is .58. If an extra control point (Node 20) is added to PCB 13 and another observation point is placed on PCB 9 (Node 19), the resulting testability calculations are shown in Figure 4.2-9. The new

NODE	ACCESSABILITY (A_{ij})	NDNT (t_{ij})
1	1.0	1.0
2	1.0	1.0
3	1.0	.80
4	.90	.72
5	.69	.62
6	.72	.58
7	.62	.62

Figure 4.2-2: Network Dependent Node Testability of Figure 4.2-1

PATH	PATH TESTABILITY (T_p)
1	.73
2	.68
3	.70

Path 1 1 3 5 7
Path 2 2 4 6 5 7
Path 3 1 3 4 6 5 7

Figure 4.2-3: Path Testability of Figure 4.2-1

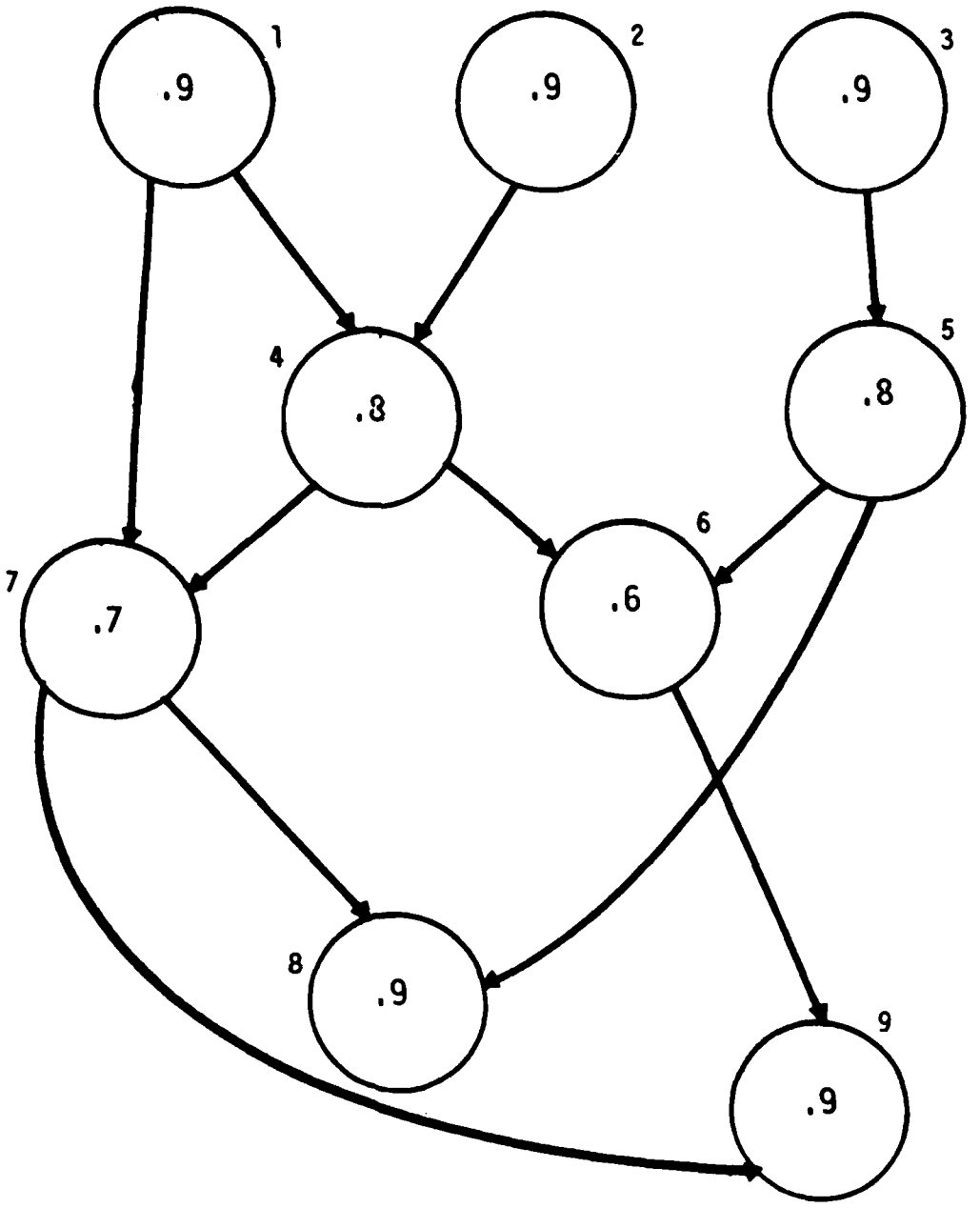


Figure 4.2-4: Complex Example of Nodal System

NODE	T_{ij}	A_{ij}	NDNT
1	.90	1	.90
2	.90	1	.90
3	.90	1	.90
4	.80	.90	.72
5	.80	.90	.72
6	.60	.72	.43
7	.70	.81	.57
8	.90	.64	.58
9	.90	.50	.45

PATH	PATH NODES	T_p
1	1 7 8	.65
2	1 7 9	.59
3	1 4 6 9	.57
4	1 4 7 8	.67
5	1 4 7 9	.62
6	2 4 6 9	.57
7	2 4 7 8	.67
8	2 4 7 9	.62
9	3 5 6 9	.57
10	3 5 8	.71

System Testability (T_S) = .62

Figure 4.2-5: Testability of Figure 4.2-4

Example 3

NODE	T_{ij}	A_{ij}	NDNT
1	.90	1.0	.90
2	.90	1.0	.90
3	.90	1.0	.90
4	.80	.90	.72
5	.80	.90	.72
6	.90	.72	.65
7	.70	.81	.57
8	.90	.64	.58
9	.90	.61	.55

PATH	PATH NODES	T_p
1	1, 7, 8	.65
2	1, 7, 9	.64
3	1, 4, 6, 9	.68
4	1, 4, 7, 8	.67
5	1, 4, 7, 9	.66
6	2, 4, 6, 9	.68
7	2, 4, 7, 8	.67
8	2, 4, 7, 9	.66
9	3, 5, 6, 9	.68
10	3, 5, 8	.71

System Testability (T_S) = .67

Figure 4.2-6: Increased Testability of Figure 4.2-4

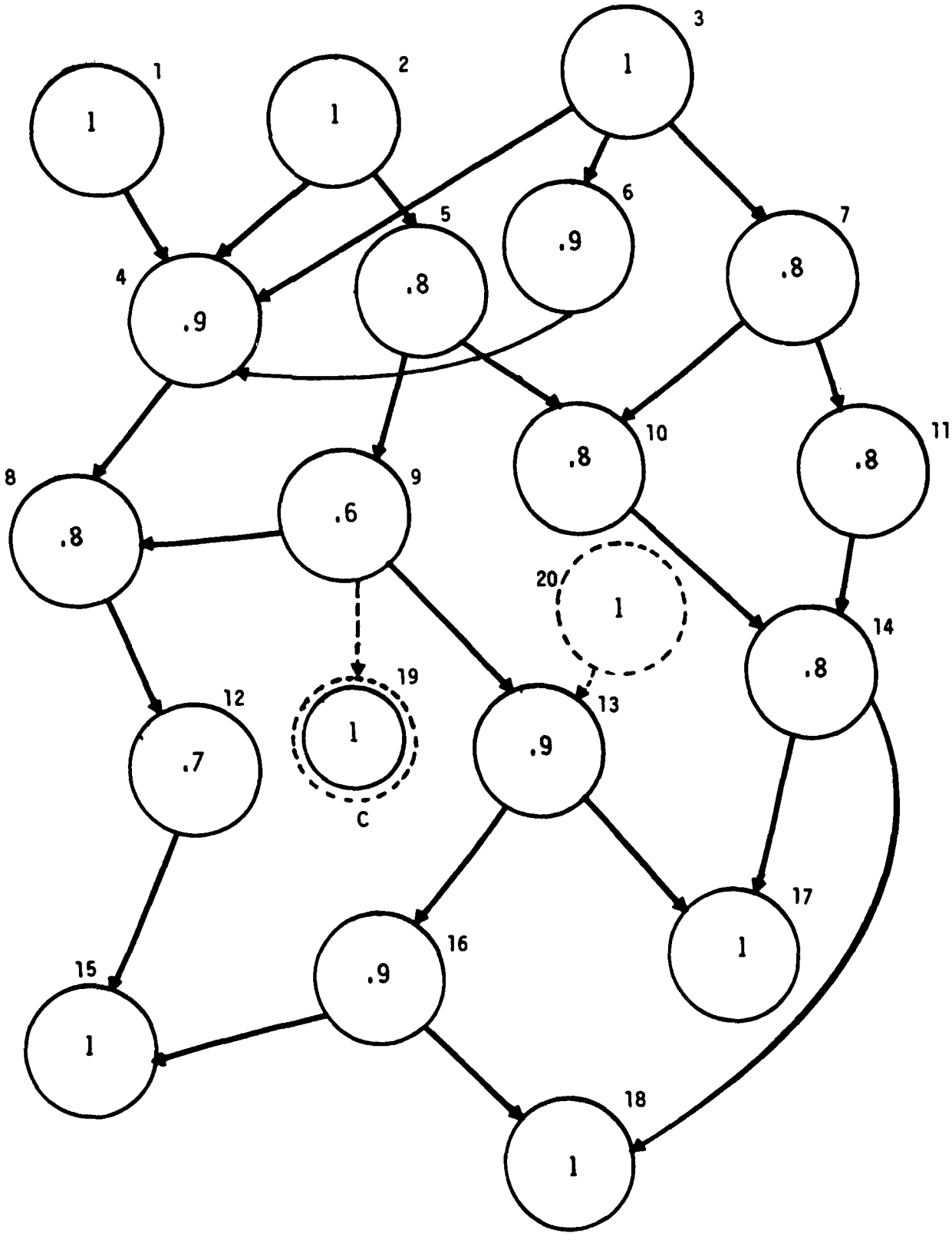


Figure 4.2-7: 18 Node Circuit

Example 4

NODE	T_{ij}	A_{ij}	NDNT
1	1.0	1.0	1.0
2	1.0	1.0	1.0
3	1.0	1.0	1.0
4	.90	.98	.88
5	.80	1.0	.80
6	.90	1.0	.90
7	.80	1.0	.80
8	.80	.68	.54
9	.60	.80	.48
10	.80	.80	.64
11	.80	.80	.64
12	.70	.54	.38
13	.90	.48	.43
14	.80	.64	.51
15	1.0	.38	.38
16	.90	.43	.39
17	1.0	.47	.47
18	1.0	.45	.45

PATH	PATH NODES	T_p
1	1 4 8 12 15	.54
2	2 4 8 12 15	.54
3	2 5 9 8 12 15	.53
4	2 5 9 13 16 15	.51
5	2 5 9 13 16 18	.52
6	2 5 9 13 17	.57
7	2 5 10 14 17	.63
8	2 5 10 14 18	.63
9	3 4 8 12 15	.54
10	3 6 4 8 12 15	.58
11	3 7 10 14 17	.63
12	3 7 10 14 18	.63
13	3 7 11 14 17	.63
14	3 7 11 14 18	.63

System Testability (T_S) = .58

Figure 4.2-8: Testability of Figure 4.2-7

Example 4 With Added Control and Observation Points

NODE	T_{ij}	A_{ij}	NDNT
1	1.0	1.0	1.0
2	1.0	1.0	1.0
3	1.0	1.0	1.0
4	.90	.98	.88
5	.80	1.0	.80
6	.90	1.0	.90
7	.80	1.0	.80
8	.80	.68	.54
9	.60	.80	.48
10	.80	.80	.64
11	.80	.80	.64
12	.70	.54	.38
13	.90	.74	.67
14	.80	.64	.51
15	1.0	.49	.49
16	.90	.67	.60
17	1.0	.59	.59
18	1.0	.56	.56
19	1.0	.48	.48
20	1.0	1.0	1.0

PATH	PATH NODES	T_p
1	1 4 8 12 15	.58
2	2 4 8 12 15	.57
3	2 5 9 8 12 15	.55
4	2 5 9 13 16 15	.63
5	2 5 9 13 16 18	.64
6	2 5 9 13 17	.66
7	2 5 9 19	.62
8	2 5 10 14 17	.67
9	2 5 10 14 18	.66
10	3 4 8 12 15	.57
11	3 6 4 8 12 15	.61
12	3 7 10 14 17	.67
13	3 7 10 14 18	.66
14	3 7 11 14 17	.67
15	3 7 11 14 18	.66
16	20 13 16 15	.64
17	20 13 16 18	.67
18	20 13 17	.71

System Testability (T_S) = .63

Figure 4.2-9: Increased Testability of Figure 4.2-7

system testability value is .63 which registers a slight gain even for a system as large as this one.

Finally, the response of the accessibility model to feedback is illustrated in the following two examples. Consider initially the simple case of five PCB's connected in series as shown in Figure 4.2-10. The testability calculations are given in Figure 4.2-11. The testability of this system is .85. Now if feedback is added between PCB's 4 and 2 as shown in Figure 4.2-12 the NDNT's are reduced as given in Figure 4.2-13. The overall testability is also reduced as shown.

The accessibilities are determined using equation 14.

$$A_{kl} = \sum_{ij} A_{ij}\, Q_{kl}\, T_{ij}$$

$$A_{11} = 1$$

$$A_{22} = A_{11}\, Q_{22}\, T_{11} + A_{44}\, Q_{22}\, T_{44} = .5(.95) + A_{44}\,(.5)\,.95$$

$$A_{33} = A_{22}\, Q_{33}\, T_{22} = A_{22}\,(.95)$$

$$A_{44} = A_{33}\, Q_{44}\, T_{33} = A_{33}\,(.95)$$

$$A_{55} = A_{44}\, Q_{55}\, T_{33} = A_{44}\,(.95)$$

In this case A_{11} is an input, so its accessibility is 1. Since A_{22} is dependent on A_{44}, A_{44} is dependent on A_{33}, and A_{33} is dependent on A_{22}, the accessibilities must be solved as a set of simultaneous equations:

$$A_{44} = A_{33}\,(.95) = \left[A_{22}\,(.95)\right]\,(.95) = A_{22}\,(.90)$$

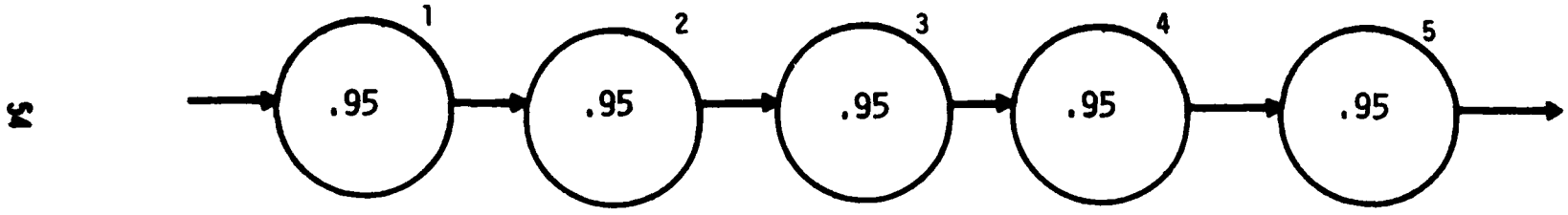


Figure 4.2-10: Simple Five Node System

NODE	T_{ij}	A_{ij}	NDNT
1	.95	1.0	.95
2	.95	.95	.90
3	.95	.90	.86
4	.95	.86	.81
5	.95	.81	.77

	PATH NODES	T_p
1	1 2 3 4 5	.85

System Testability (T_S) = .85

Figure 4.2-11: Testability of Figure 4.2-10

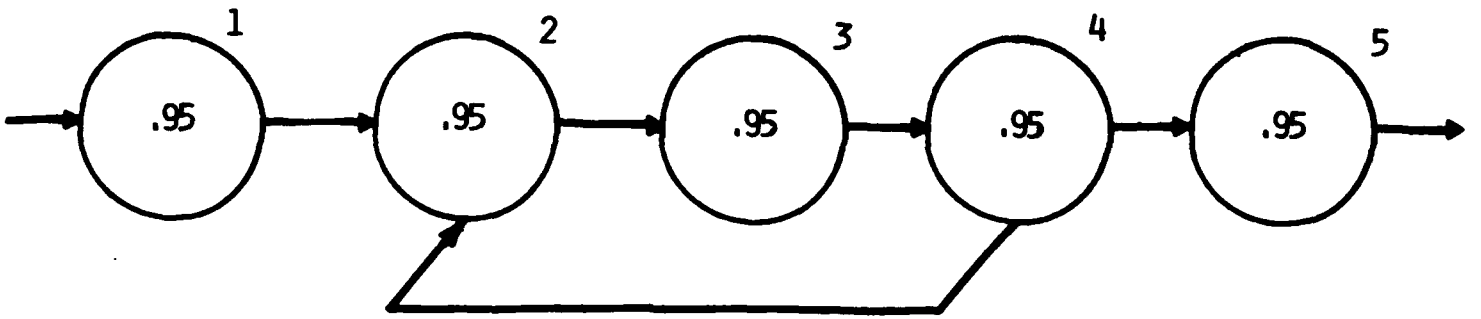


Figure 4.2-12: Five Node System With Feedback

$$A_{22} = .48 + A_{44}\,(.48) = .48 + \left[A_{22}\,(.90)\right](.48) = .84$$

The remaining accessibilities and system testability are given in Figure 4.2-13.

4.3 Application to BAC Equipment

The accessibility model algorithm was programmed in the Pascal computer language and applied to the three BAC systems identified in section 3.0: ASAT, the 767 Flap/Slat Unit, and the Roland Command Computer. Two programs were used; one (Input-Graph) to construct the node graph structure, and the other (Testability) to determine system testability. The source listings of these programs are in Appendix I. The following proposed scale for system testability evaluation is used in the discussions that follow for each of those BAC systems:

BAC System Testability Score	Testability Rating
0 - .15	impossible
.16 - .35	hard
.36 - .70	medium
.71 - .90	easy
.91 - 1.0	very easy

The boundaries for the individual testability rating categories ("impossible," "hard," . . . etc) were derived via means which were similar to the way in which the spread for the Grumman PCB testability technique was derived. The spread for system testability for the categories above "impossible" was set from .16 to 1.0 rather than .51 to 1.0 (see

NODE	A_{ij}	NDNT
1	1	.95
2	.83	.79
3	.79	.75
4	.75	.71
5	.71	.68

	PATH NODES	T_p
1	1 2 3 4 5	.77
2	1 2 3 4 2 3 4 5	.76

System Testability (T_S) = .76

Figure 4.2-13: Testability of Figure 4.2-12

table 3-1), and judgement based on system testing experience, in addition to the character of the system rating process, slightly changed the way individual categories were bounded as well.

The application of the system testability score involved comparing the FOM scores for Boeing designed subsystems, calculated using the Accessibility Model ("BAC System Testability Score"), to independent "Testability Ratings" for the same hardware made by Boeing test engineers. Some care was taken in interpreting these assessments to specifically account for the following two facts:

(1) The use of extensive and expensive BIT and ATE can make even the most untestable design look adequate.

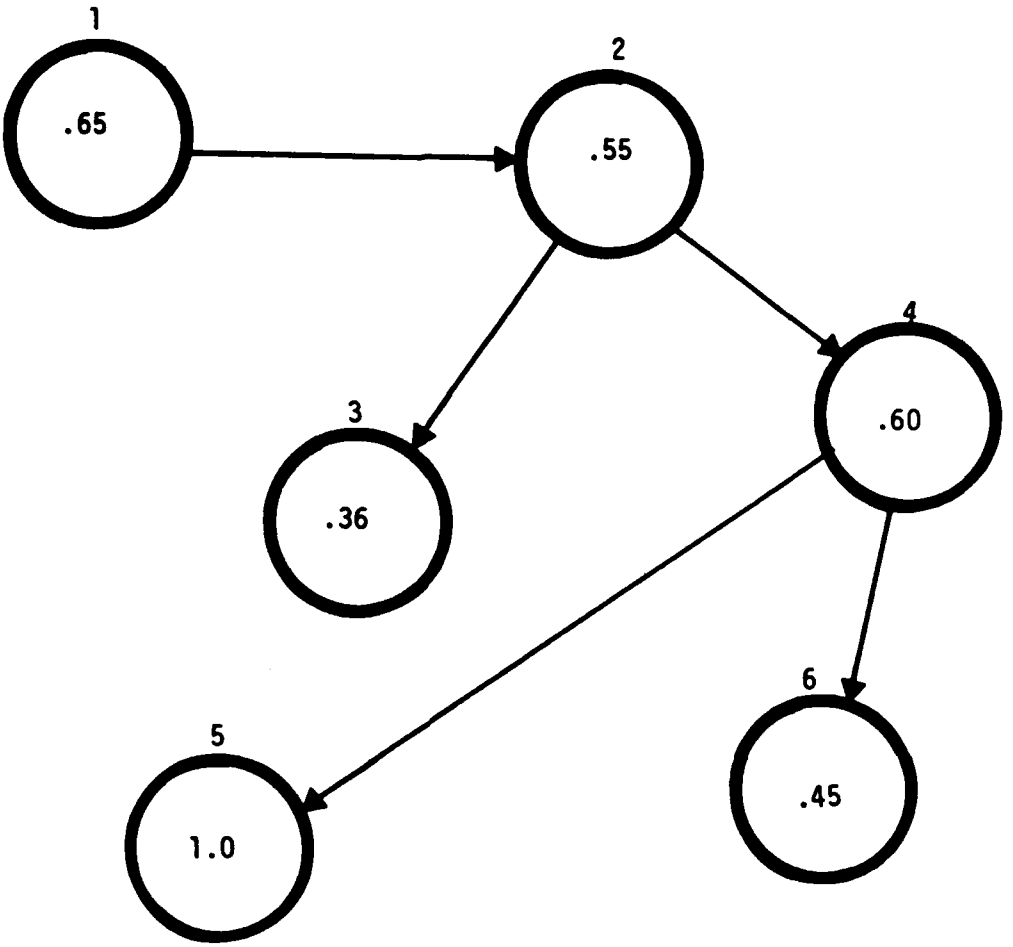
(2) Engineering assessments regarding testability during development (prior to test engineering) are often more pessimistic than the final assessments after the test engineering design phase.

Both of these factors must be considered and accounted for in any comparison of this type.

The results of the application are summarized in the remainder of this section.

4.3.1 ASAT Flight Control Electronics

The ASAT system consists of five PCB's that result in the graph structure for board interconnection paths shown in Figure 4.3.1-1. The initial



NODE	BOARD
1	Ignition Discrete
2	Servo/amp/bit
3	Serial data
4	Gyro wheel supply
5	Outputs (not physical PCB)
6	Gyro Demod/excitation

Figure 4.3.1-1: Graph Structure for the ASAT Unit

testabilities for the five PCB's (Figures 4.3.1-1 and 4.3.1-2) were determined using the Grumman Method. Node 5, in the graph, represents an output point for the system and has an initial testabilty of 1 (it is not a physical PCB).

The overall system testability using the transformed Grumman PCB testability numbers is .25, or hard to test. (Note that the range of Grumman FOM numbers is changed to a range of 0.0 to 1.0 as shown in Table 3-1). When numbers based on current Boeing PCB test experience are substituted for the Grumman numbers (see Figure 4.3.1-3), the resultant system testability is calculated as .49, or medium in testability difficulty. Test experience on the engineering model of the system (it is not yet in production) has shown the unit to be hard to test, (which is in more agreement with the result using the Grumman PCB numbers). This result is not surprising, since functional test engineering is yet to be completed, and during development, electronic items are often judged more complex to test than when they are in production. This is due to the fact that functional test engineering trails development tasks. A good example of this was in the development of the 767 Flap Slat unit described in Section 4.3.2. The initial data we obtained (before the unit was in full production) resulted in a determination of the system as hard to test. By the end of this contract, with the box in full production and with the support of a modest amount of ATE, this system is now judged medium in difficulty of testing. Because of this example (and others), BAC test engineers believe that when functional test design is complete and ATE is available, the ASAT unit will be in the moderate or medium range of testability difficulty.

4.3.2 767 Flap/Slat Unit

The 767 Flap/Slat Unit is part of the digital control system for the 767 flaps. A basic block diagram is shown in Figure 4.3.2-1. Using the initial testabilities for the three PCB's derived from the Grumman Method, the block

NODE	GRUMMAN FOM	T_{ij}	A_{ij}	NDNT
1	MEDIUM	.65	1.0	.65
2	VERY HARD	.55	.65	.36
3	IMPOSSIBLE	.36	.36	.13
4	HARD	.60	.36	.22
5		1.0	.22	.22
6	IMPOSSIBLE	.45	.22	.10

	PATH NODES	T_p
1	1 2 3	.25
2	1 2 4 5	.29
3	1 2 4 6	.21

System Testability, T_s, = .25

Figure 4.3.1-2: Testability of ASAT Unit

NODE	BAC T_{ij} Experience:	T_{ij}	A_{ij}	NDNT
1	Very Easy	.95	1.0	.95
2	Medium/Hard	.66	.95	.63
3	Medium	.69	.63	.43
4	Hard	.60	.63	.38
5	----	1.0	.38	.38
6	Hard	.60	.38	.23

	PATH NODES	T_p
1	1 2 3	.60
2	1 2 4 5	.50
3	1 2 4 6	.41

System Testability, T_s , = .49

Figure 4.3.1-3: ASAT Testability Using BAC Initial Testabilities

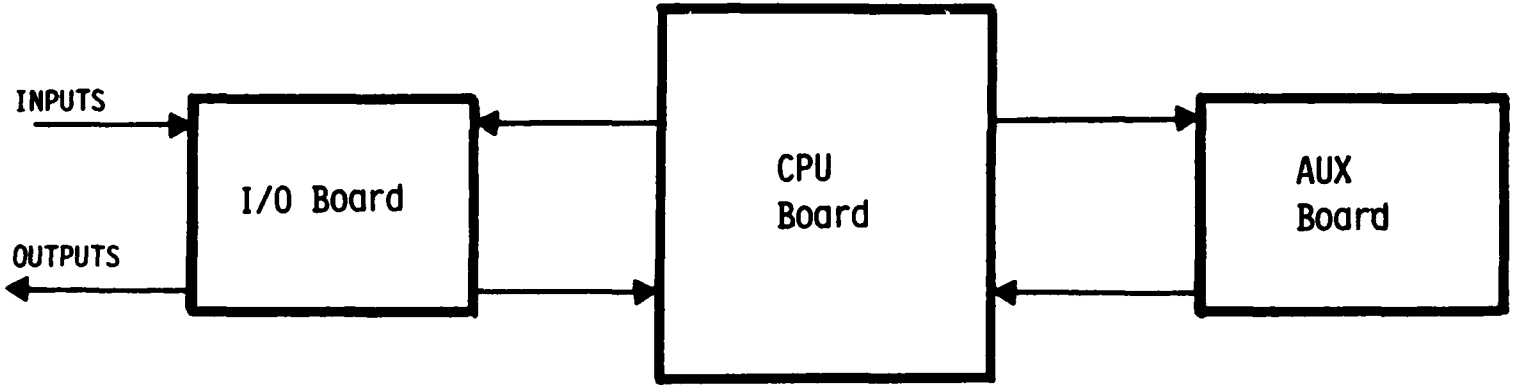


Figure 4.3.2-1: 767 Flap Slat Block Diagram

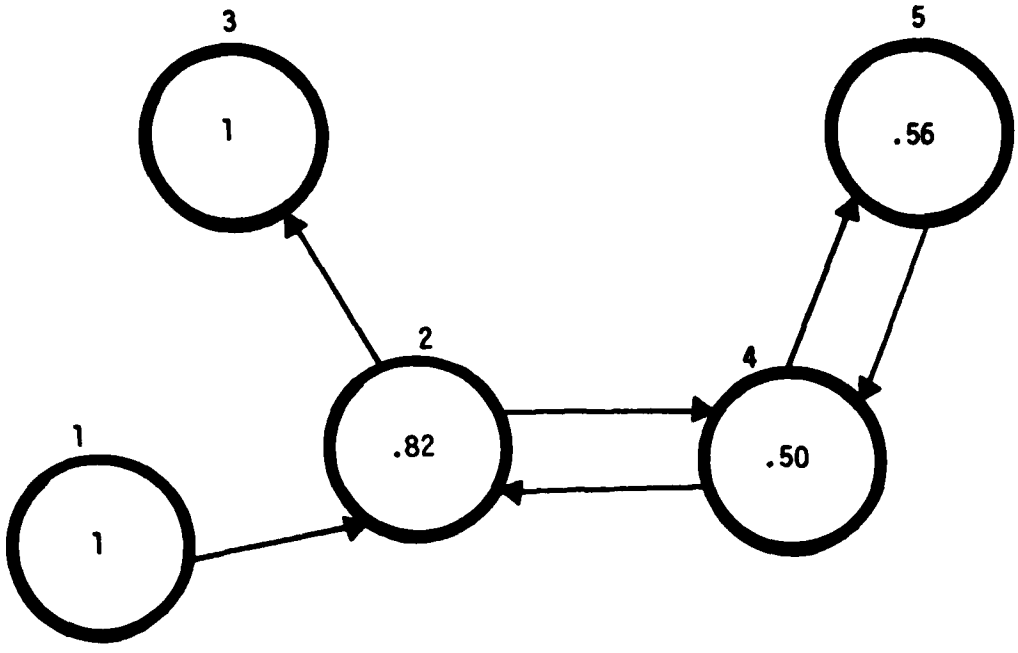
diagram is translated directly into the graph shown in Figure 4.3.2-2. Nodes 1 and 3 in the graph represent the I/O points for the system and have initial testabilities of 1 (they do not represent physical PCB's). Note that this simple unit has feedback links, so most testability FOM's will not work on the system. However, the Accessibility Model produces the results shown in Figure 4.3.2-3.

The overall system testability using the Grumman testability numbers is .31. When numbers based on Boeing PCB testing experience are substituted for the Grumman numbers (Figure 4.3.2-4) the system testability is calculated to be .40. Qualitatively, use of Grumman PCB numbers puts the system on the hard side of the "border" of being hard-to-medium in testability; use of Boeing experience places it at the lower range of medium testability. This result of medium testability was consistent with Boeing test experience for the black box using a modest amount of the factory's Automated Test Equipment.

4.3.3 Roland Command Computer

The Roland Command Computer consists of 8 interconnected digital PCB's which produce the data paths shown in Figure 4.3.3-1. The other 6 cards were analog and didn't apply themselves to the Grumman technique, therefore they were not used for this model. Five I/O point nodes with an initial testability of 1 have been added to the graph to represent the system input/output points.

The system testability FOM using the Grumman PCB variables is calculated as .03 (Figure 4.3.3-2). This results in an "impossible to test without cost penalty" rating; this is to be expected, since, according to the Grumman



Key

NODE #

NODE	BOARD
1	Inputs (not a physical part)
2	Discrete I/O Board
3	Outputs (not physical part)
4	CPU Board
5	AUX I/O Board

Figure 4.3.2-2: Graph Structure For the 767 Flap Slat Unit

NODE	GRUMMAN FOM	T_i	A_{ij}	NDNT
1		1.0	1.0	1.0
2	MEDIUM/EASY	.82	.57	.47
3		1.0	.47	.47
4	IMPOSSIBLE	.50	.27	.14
5	VERY HARD	.56	.14	.08

	PATH NODES	T_p
1	1 2 3	.57
2	1 2 4 2 3	.34
3	1 2 4 5 4 2 3	.20

System Testability, T_S, = .31

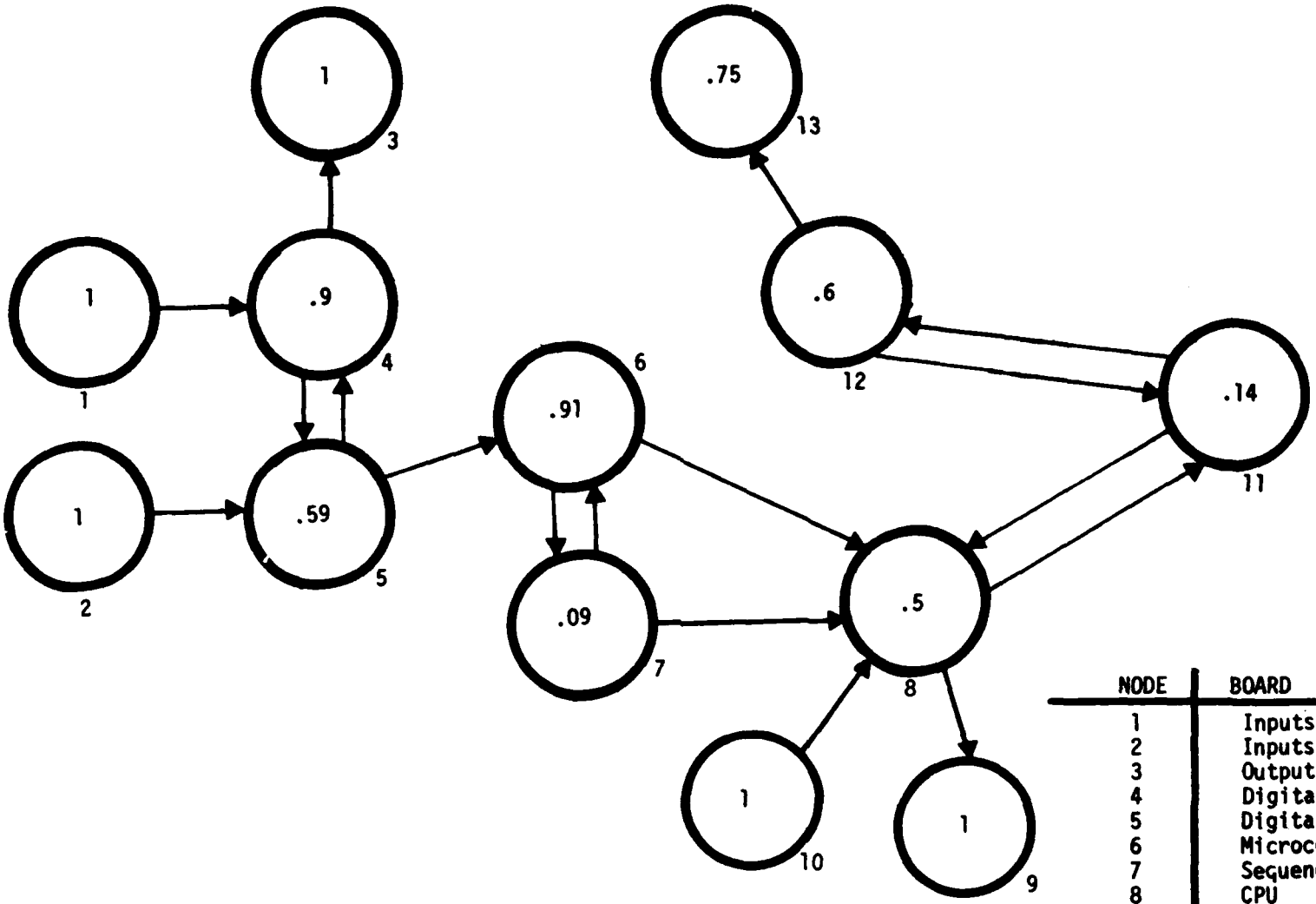
Figure 4.3.2-3: Testability of 767 Flap/Slat Unit

NODE	BAC T_{ij} Experience	T_{ij}	A_{ij}	NONT
1	----	1.0	1.0	1.0
2	Medium/Easy	.78	.60	.47
3	----	1.0	.47	.47
4	Hard/Medium	.66	.32	.21
5	Medium/Easy	.78	.21	.16

	PATH NODES	T_{ij}
1	1 2 3	.57
2	1 2 4 2 3	.41
3	1 2 4 5 4 2 3	.30

System Testability, T_S , = .40

Figure 4.3.2-4: 767 FSU Testability Using BAC Initial Testabilities



NODE	BOARD
1	Inputs
2	Inputs
3	Outputs
4	Digital I/O #1
5	Digital I/O #2
6	Microcode Adapter
7	Sequencer
8	CPU
9	Outputs
10	Inputs
11	1K RAM/4K EPROM
12	DMA I/O
13	PIM/RTC

Figure 4.3.3-1 Graph Structure For the Roland Command Computer

NODE	GRUMMAN FOM	T_{ij}	A_{ij}	NDNT
1	----------	1.0	1.0	1.0
2	----------	1.0	1.0	1.0
3	----------	1.0	.67	.67
4	EASY	.90	.75	.67
5	HARD	.59	.84	.49
6	VERY EASY	.91	.26	.23
7	IMPOSSIBLE	.09	.23	.02
8	IMPOSSIBLE	.50	.32	.16
9		1.0	.16	.16
10		1.0	1.0	1.0
11	IMPOSSIBLE	.14	.08	.01
12	HARD	.60	.01	.01
13	MEDIUM/EASY	.75	.01	.01

	PATH NODES	T_p
1	1 4 3	.75
2	1 4 5 4 3	.67
3	1 4 5 4 5 6 7 6 7 8 9	.09
4	1 4 5 4 5 6 7 6 7 8 11 8 9	.06
5	1 4 5 4 5 6 7 6 7 8 11 8 11 12 13	.02
6	1 4 5 4 5 6 7 6 7 8 11 12 11 8 9	.03
7	1 4 5 4 5 6 7 6 7 8 11 12 11 12 13	.02
8	1 4 5 4 5 6 7 6 7 8 11 12 13	.02
8	1 4 5 4 5 6 7 6 8 9	.13
10	1 4 5 4 5 6 7 6 8 11 8 9	.07
	NOTE: only 20 of the possible 106 paths are shown	
97	2 5 6 8 11 8 11 12 13	.02
98	2 5 6 8 11 12 11 8 9	.03
99	2 5 6 8 11 12 11 12 13	.01
100	2 5 6 8 11 12 13	.02
101	10 8 9	.22
102	10 8 11 8 9	.05
103	10 8 11 8 11 12 13	.01
104	10 8 11 12 11 8 9	.02
105	10 8 11 12 11 12 13	.01
106	10 8 11 12 13	.01

System Testability, T_S = .03

Figure 4.3.3-2: Testability of Roland Command Computer

rating, three of the PCB's are "impossible" to test and two of them are "hard" to test. However, the Grumman FOMs reflect an underrating of the initial testabilities of the CPU and memory units. Using the Boeing test engineer's evaluation of the testabilities of these PCB's (rather than the Grumman FOM) for the initial node (PCB) values and rerunning the Accessibility Model results in a system testability value of .23, a "hard" to test level. Summary data for this calculation is shown in Figure 4.3.3-3.

The calculated "hard-to-test" level for the Roland Command Computer using the Accessibility Model is consistent with both the degree of difficulty of testing the PCB's in the system, and the degree of difficulty that is associated with testing the system without the aid of a sophisticated test system designed especially for the computer. This is especially true considering that one card, the Sequencer, is considered just inside the impossible-to-test-without-cost-penalty range even by the Boeing test personnel. During engineering model checkout it was considered very hard to test by the engineers (no ATE was available). However, those same people now judge the overall commmand computer as easy to test as a system. The reason for this can be attributed to the development of an extensive and relatively expensive test system, which would not have necessarily been needed had the design been endowed with a higher level of testability (the sequencer card in particular). The bottom line is that given enough ATE, sufficient test engineering resources, and some cleverness (insight) in design of the test procedures, many difficulties can be overcome (but at a high cost).

a. When the Roland command computer was initially designed, considerable and difficult troubleshooting was required to detect and isolate faults even when self-test and Special Test Equipment (STE) was used.

NODE:	BAC T_{ij} Experience:	T_{ij}	A_{ij}	NDNT
1	----	1.0	1.0	1.0
2	----	1.0	1.0	1.0
3	----	1.0	.64	.64
4	Easy	.86	.75	.64
5	Hard	.60	.82	.49
6	Easy	.86	.31	.27
7	Impossible/Hard	.50	.27	.13
8	Medium	.69	.39	.27
9	----	1.0	.27	.27
10	----	1.0	1.0	1.0
11	Easy	.86	.18	.16
12	Hard	.60	.16	.09
13	Medium/Easy	.78	.09	.07

	PATH NODES	T_p
1	1 4 3	.73
2	1 4 5 4 3	.65
3	1 4 5 4 5 6 7 6 7 8 9	.29
4	1 4 5 4 5 6 7 6 7 8 11 8 9	.27
5	1 4 5 4 5 6 7 6 7 8 11 8 11 12 13	.20
6	1 4 5 4 5 6 7 6 7 8 11 12 11 8 9	23
7	1 4 5 4 5 6 7 6 7 8 11 12 11 12 13	.18
8	1 4 5 4 5 6 7 6 7 8 11 12 13	.20
9	1 4 5 4 5 6 7 6 8 9	.33
10	1 4 5 4 5 6 7 6 8 11 8 9	.30
	Note: Only 20 of the possible 106 Paths are shown	
97	2 5 6 8 11 8 11 12 13	.18
98	2 5 6 8 11 12 11 8 9	.22
99	2 5 6 8 11 12 11 12 13	.15
100	2 5 6 8 11 12 13	.17
101	10 8 9	.36
102	10 8 11 8 9	.27
103	10 8 11 8 11 12 13	.15
104	10 8 11 12 11 8 9	.20
105	10 8 11 12 11 12 13	.13
106	10 8 11 12 13	.14

System Testability, T_S , = .22

Figure 4.3.3-3: Roland Testability Using Boeing Initial Testabilities

b. Because of the degree of difficulty in testing this subsystem, the Roland Program put their best test engineers on the job designing both ATE software and an interface adapter to go between the ATE and the Roland Command Computer.

c. Because a fairly large production run was anticipated and the contract required that fairly low level maintenance personnel to be able to troubleshoot the subsystem, a relatively costly test development effort was economically justified.

d. The resultant test hardware and software has proven itself in the manufacturing area. Accurate fault diagnosis and isolation to the card level are possible very quickly using the ATE. This is what the manufacturing test engineers base their easy rating on for the box.

An additional sidelight on the Sequencer card also demonstrates the completeness and cleverness of their test approach. That card is so complex and difficult to test that the present card test (theoretically simpler than a subsystem test) sometimes misses faults and passes a card. The subsystem test is good enough that it detects those sequencer faults and isolates them to that PCB.

4.4 Summary

As can be seen from the examples in Section 4.3, use of the Grumman PCB testability numbers tends to give a more pessimistic view of a card's and a system's testability. When VLSI devices are used, it can be considerably more

pessimistic than test experience justifies. Further, another point brought out in comparing the BAC Accessibility Model to actual experience is that the results of using this model will be in fairly close agreement with actual test results when the unit reaches production. However, if more than normal design effort is devoted to test design using ATE and special adapters, better, even much better, testability can be achieved than one would anticipate from the testabilities of the individual cards and the controllability, observability and accessibility of them when interconnected. In the end it is a matter of economics; testability of inherently hard-to-test subsystems can be improved. However, the best economic approach is to design good testability into the subsystem in the first place. Use of Grumman, BAC Accessibility or any F.O.M. approach during the design phase will reduce both production costs during manufacturing test and system life cycle costs in the field.

Further development of F.O.M. techniques so they can be used as practical design tools in a Computer Aided Engineering environment is recommended. This is a necessary first step in providing the design engineer with the tools he needs to design testability into the system as an integral part of the design process.

5.0 System's Application Example

The Accessibility Model developed in section 4.2 meets the initial cri for a testability measure which can extend the Grumman measure to the s level. The entire process is illustrated in the following simplified exam

Given the simple system shown in Figure 5.1-1 consisting of the interc tions of five subsystems, the testability of this system may be calc using the Accessibility Model. Each subsystem consists of an interconne of PCB's as shown in Figures 5.1-2a to 5.1-2e.

As an example, the testability of subsystem 1 (figure 5.1-2a) wi determined.

o Step one: determine i and j

Each node is represented by "N_{ij}". Node N_{ij} is at level i and is node j. The level is the number of nodes in the longest path from an input t node, plus 1. In subsystem 1 (SS1) let j be the PC number of each node.

There are no nodes from the input to node 1, so the level of node 1 There is one node from the input to node 2, so the level of node 2 There are 2 nodes (1 and 2) in the longest path from the input to node node 3 is at level 3. Likewise, node 4 is at level 3. So, PC1 is N_{11}, N_{22}, PC3 is N_{33}, and PC4 is N_{34}.

Step two: determine A_{1j}

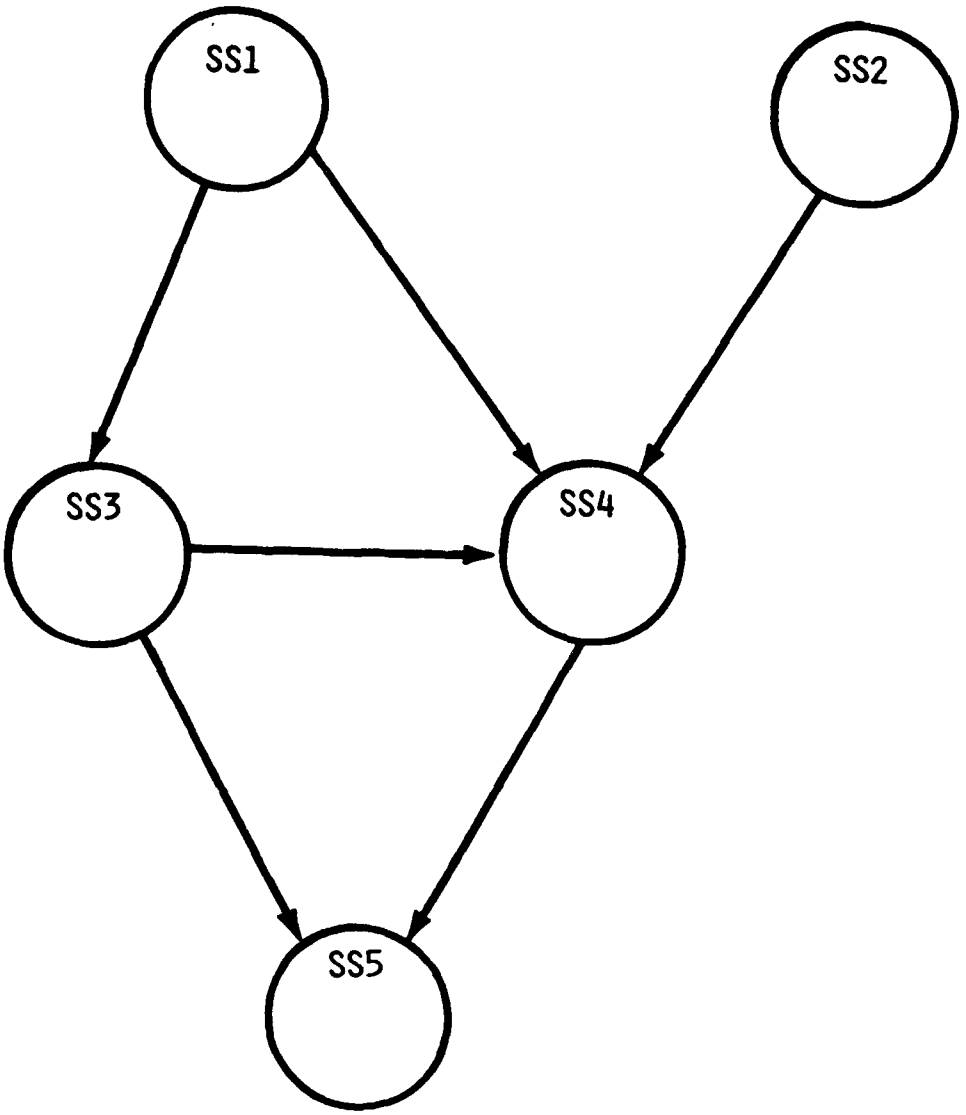


FIGURE 5.1-1: System Level Diagram

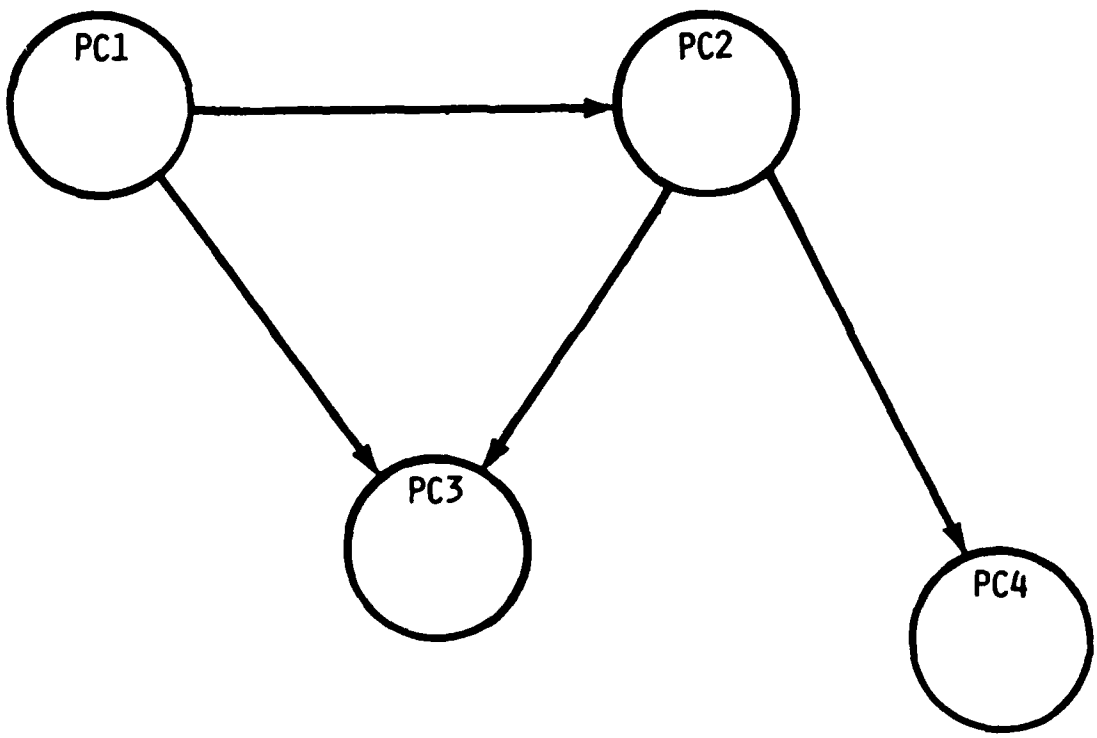


Figure: 5.1-2a Subsystem 1

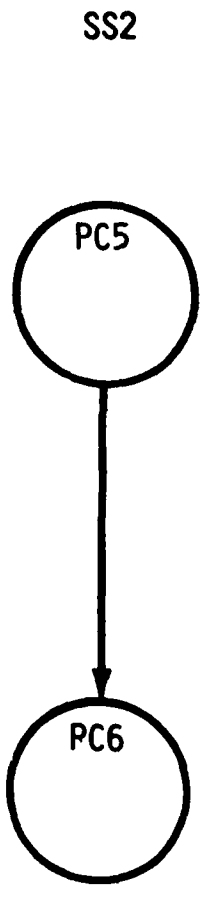


Figure 5.1-2b: Subsystem 2

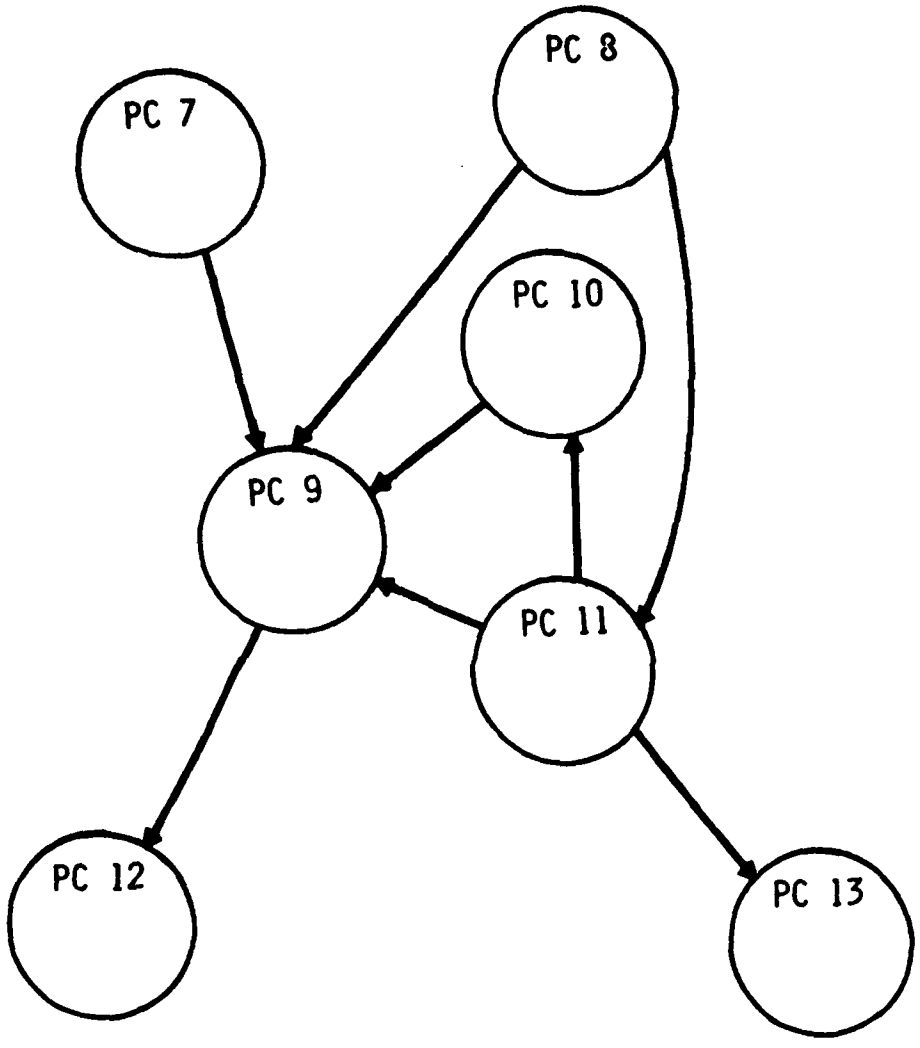


Figure 5.1-2c: Subsystem 3

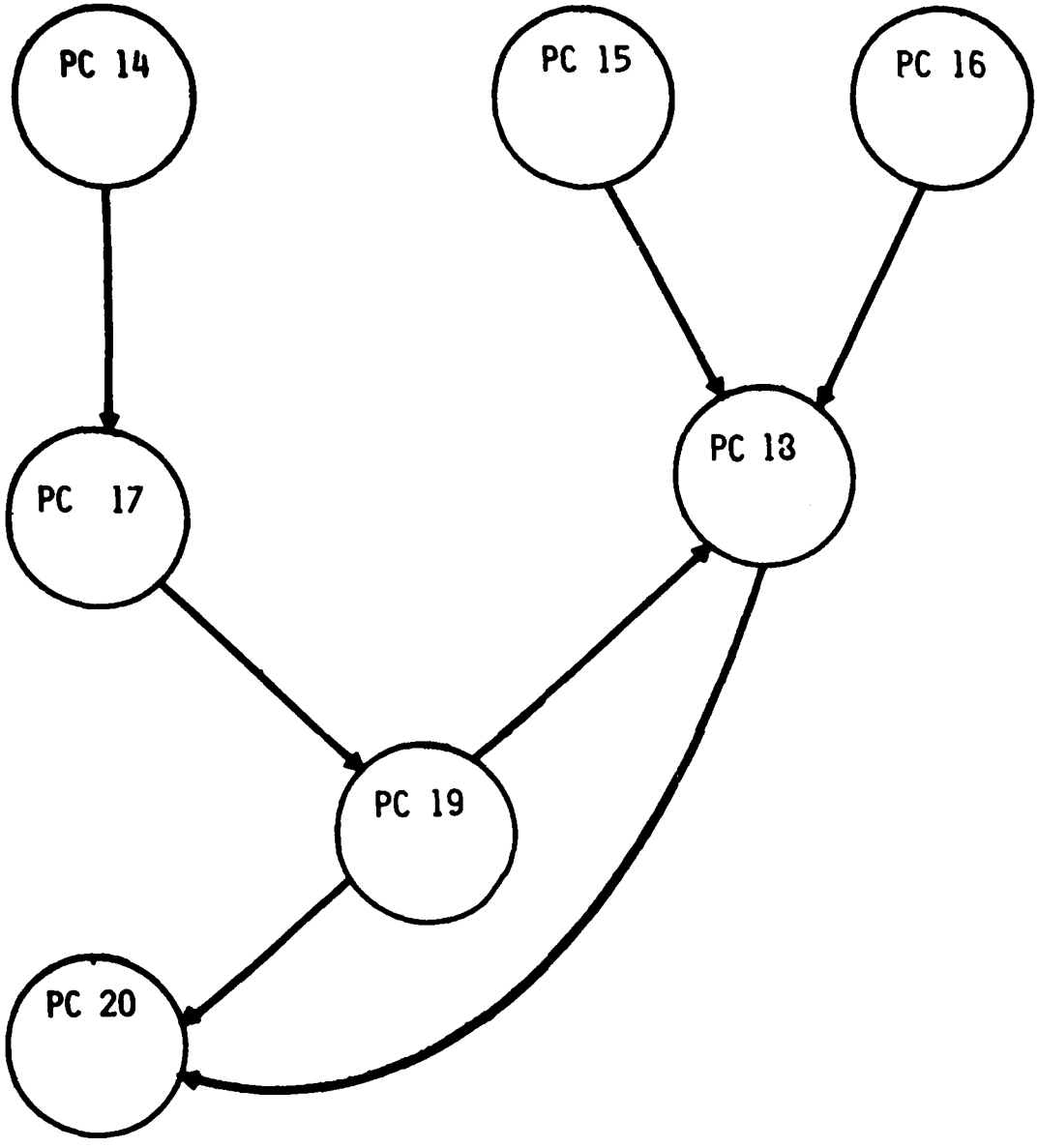


Figure 5.1-2d: Subsystem 4

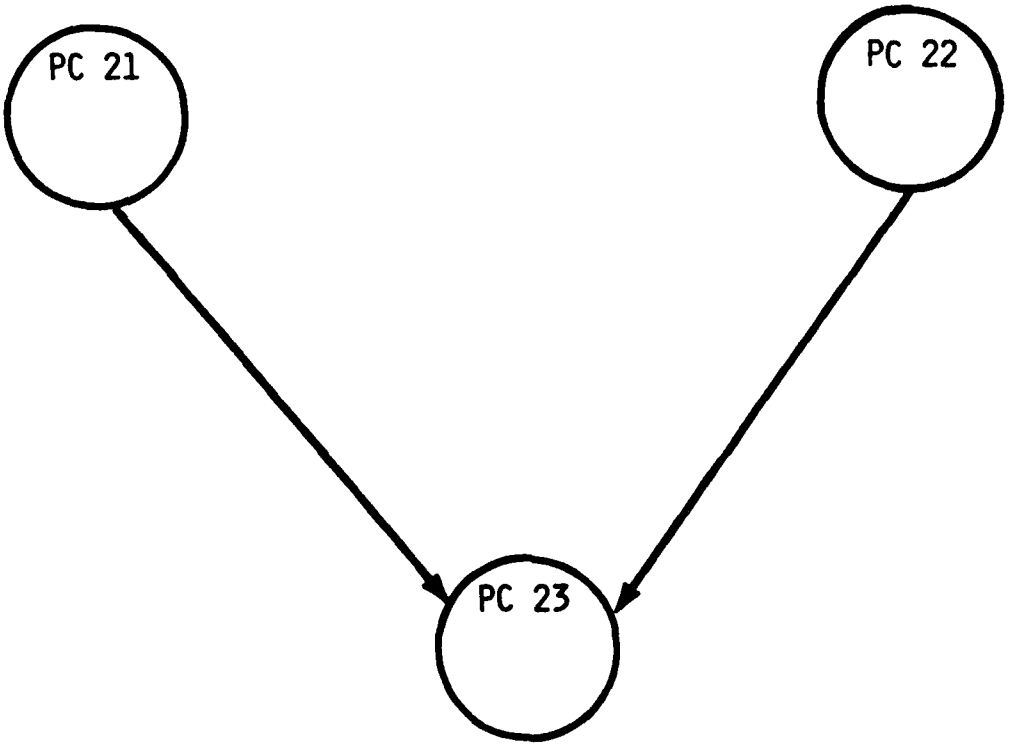


Figure 5.1-2e: Subsystem 5

The accessibility of a node, A_{ij}, at level 1 is equal to one.

N_{11}, by definition of "i" is the only node at level one.

Therefore, A_{11} only is equal to 1.

o Step three: determine A_{ij}

The accessibility of an interior node ($i > 1$) is given by:

$$A_{ij} = \sum_{kl} A_{kl} Q_{ij} T_{kl}$$

The sum in the equation is over all nodes that are the closest source of an input to N_{ij}.

T_{kl} is the initial testability of N_{kl}. T_{kl} is determined using:

$$T_{kl} = \frac{G_{kl} + 100}{200}$$

where: T_{kl} = the transformed Grumman testability of N_{kl}

G_{kl} = the Grumman FOM when the Grumman FOM ≥ -100

G_{kl} = -100 when the Grumman FOM < -100

G_{kl} for each node in SS1 is given in figure 5.1-3. Using these values, T_{11} = $(90 + 100) \div 200 = .95$. The rest are given in figure 5.1-3.

Q_{ij} is the arc weight of N_{ij}. Q_{ij} is determined using the following:

Subsystem #	Node #	Grumman FOM	Transformed FOM
1	1	90	.95
	2	60	.80
	3	78	.89
	4	88	.94
2	5	90	.95
	6	90	.95
3	7	60	.80
	8	50	.75
	9	20	.60
	10	80	.90
	11	40	.70
	12	90	.95
	13	90	.95
4	14	90	.95
	15	90	.95
	16	90	.95
	17	20	.60
	18	0	.50
	19	-20	.40
	20	60	.80
5	21	90	.95
	22	90	.95
	23	90	.95

$$\text{Transformed FOM} = \frac{\text{Grumman FOM} + 100}{200}$$

Figure 5.1-3: Transformed Grumman Scores of Figures 5.1-2a through 5.1-2e

$$Q_{ij} = \frac{1}{d_{in}(N_{ij})}$$

where $d_{in}(N_{ij})$ = the number of arcs entering N_{ij}

Q_{11} is not defined, and not required since A_{11} is determined in step 2. By inspection of figure 5.1-2a, N_{22} has one arc entering it. So $Q_{22} = 1 \div 1 = 1$. N_{33} has 2 arcs entering it, so $Q_{33} = 1 \div 2 = .50$. Also, $Q_{34} = 1.0$.

A_{ij} is the accessibility of N_{ij}. A_{11} was defined in step 2 as one. The other accessibilities are calculated next:

$$A_{22} = A_{11}\, Q_{22}\, T_{11} = 1 \times 1 \times .95 = .95$$
$$A_{33} = A_{11}\, Q_{33}\, T_{11} + A_{22}\, Q_{33}\, T_{22} = 1x.5x.95 + .95x.5x.80 = .86$$
$$A_{43} = A_{22}\, Q_{43}\, T_{22} = .95 \times 1 \times .80 = .76$$

o Step four: determine NDNT

NDNT is the network dependent nodal testability or "t_{ij}".

The NDNT is calculated using the following:

$$t_{ij} = A_{ij}\, T_{ij}$$

Using this equation, $t_{11} = A_{11}\, T_{11} = 1 \times .95 = .95$. The other NDNTs are likewise calculated and shown in figure 5.1-4a.

o Step five: determine T_p

SS 1

NODE	T_{ij}	A_{ij}	NDNT
1	.95	1.0	.95
2	.80	.95	.76
3	.89	.86	.76
4	.94	.76	.71

PATH		T_p
1	1 2 3	.81
2	1 2 4	.80
3	1 3	.85

Subsystem Testability = .82

Figure 5.1-4a

SS 2

NODE	T_{ij}	A_{ij}	NDNT
1	.95	1	.95
2	.95	.95	.90

PATH		T_p
1	1 2	.93

Subsystem Testability = .93

Figure 5.1-4b

T_p is the path testabilities of the system. First, determine all the possible paths through SS1. In this case, there are three:

Path 1 $N_{11}\ N_{22}\ N_{33}$

Path 2 $N_{11}\ N_{22}\ N_{34}$

Path 3 $N_{11}\ N_{33}$

T_p is calculated using the following:

$$T_p = \left(\frac{1}{M_p} \sum_{n=1}^{M_p} \frac{1}{t_n} \right)^{-1}$$

where M_p = number of nodes in the path

t_n = NDNT of each node along the path

For SS1, T_p for Path 1, $(T_p)_1$ is found next:

$$(T_p)_1 = \left[\frac{1}{3} \left(\frac{1}{t_{11}} + \frac{1}{t_{22}} + \frac{1}{t_{33}} \right) \right]^{-1}$$

$$= \left[\frac{1}{3} \left(\frac{1}{.95} + \frac{1}{.76} + \frac{1}{.76} \right) \right]^{-1}$$

$$= .81$$

$(T_p)_2$ and $(T_p)_3$ are determined in the same manner. Their values are given in figure 5.1-4a.

o Step six: determine T_s

T_s is the overall system testability. It is calculated using the following equation:

$$T_s = \left(\frac{1}{M} \sum_{i=1}^{M} \frac{1}{(T_p)_i} \right)^{-1}$$

where M = number of paths in the system.

For SS1, T_s is calculated next:

$$(T_s)_1 = \left[\frac{1}{3} \left(\frac{1}{(T_p)_1} + \frac{1}{(T_p)_2} + \frac{1}{(T_p)_3} \right) \right]^{-1}$$

$$= \left[\frac{1}{3} \left(\frac{1}{.81} + \frac{1}{.80} + \frac{1}{.85} \right) \right]^{-1}$$

$$= .82$$

Using the table in section 4.3.1, SS1 would be easy to test.

The six step procedure is repeated for each of the four remaining subsystems. The results are given in figures 5.1-4b through 5.1-4e. Next, the testability of the overall system comprised of the five subsystems is determined (figure 5.1-1). For this calculation, T_s of each subsystem becomes T_{ij} for the overall system:

$$T_{11} = (T_s)_1$$
$$T_{12} = (T_s)_2$$
$$T_{23} = (T_s)_3$$
$$T_{34} = (T_s)_4$$
$$T_{45} = (T_s)_5$$

SS 3

NODE	T_{ij}	A_{ij}	NDNT
1	.80	1.0	.80
2	.75	1.0	.75
3	.60	.64	.38
4	.90	.52	.47
5	.70	.75	.52
6	.95	.38	.36
7	.95	.52	.50

PATH		T_p
1	1 3 6	.45
2	2 3 6	.44
3	2 5 3 6	.46
4	2 5 4 3 6	.47
5	2 5 7	.57

Subsystem Testability = .48

Figure 5.1-4c

SS 4

NODE	T_{ij}	A_{ij}	NDNT
1	.95	1	.95
2	.95	1	.95
3	.95	1	.95
4	.60	.95	.57
5	.50	.71	.35
6	.40	.57	.23
7	.80	.29	.23

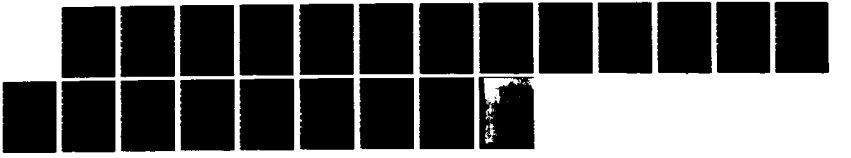
PATH		T_p
1	1 4 6 5 7	.35
2	1 4 6 7	.35
3	2 5 7	.36
4	3 5 7	.36

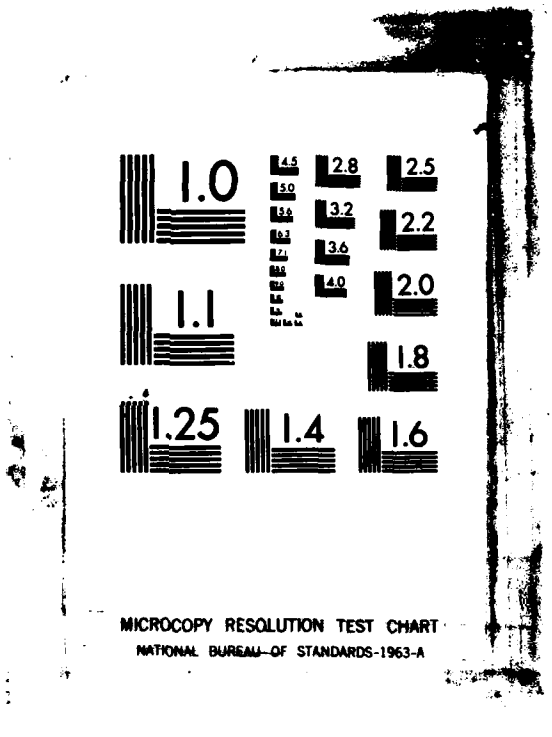
Subsystem Testability = .35

Figure 5.1-4d

AD-A141 147 ADVANCED APPLICATION OF THE PRINTED CIRCUIT BOARD TESTABILITY DESIGN AND RATING SYSTEM(U) BOEING AEROSPACE CO SEATTLE WA R PPILLMAN ET AL. DEC 83 2/2

UNCLASSIFIED RADC-TR-83-291 F30602-81-C-0224 F/G 9/5 NL




1.0
1.1
1.25
1.4
1.6
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.8
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

SS 5

NODE	T_{ij}	A_{ij}	NDNT
1	.95	1.0	.95
2	.95	1.0	.95
3	.95	.95	.90

PATH		T_p
1	1 3	.93
1	2 3	.93

Subsystem Testability = .93

Figure 5.1-4e

System Level

NODE	T_{ij}	A_{ij}	NDNT
SS1	.82	1.0	.82
SS2	.93	1.0	.93
SS3	.48	.82	.39
SS4	.35	.71	.25
SS5	.93	.32	.30

PATH		T_p
1	1 3 4 5	.36
2	1 3 5	.42
3	1 4 5	.41
4	2 4 5	.36

Subsystem Testability = .38

Figure 5.1-5

System Level Testability

The six step procedure is followed and an overall system T_s is found. The result is shown in Figure 5.1-5. This system is difficult to test with a testability value of only .38 on a scale of 0 to 1. An examination of the testability calculations at the system level indicates that subsystems 3 and 4 must be redesigned to improve their testability. A stop gap measure may also be applied as seen from Figure 5.1-5. Subsystem five has a low NDNT and a high initial testability. This results from the fact that the only control over subsystem five occurs through the low testable subsystems three and four. Testability could be improved by placing a test point on the inputs to subsystem five as shown in figure 5.1-6. The results of this modification on testability are shown in Figure 5.1-7. The addition of an added test point increased the NDNT of subsystem five to .51 or by 70%. This resulted in an increase in all the path testabilities and an overall increase in the system testability to .47, or by 24%.

5.1 Future Research Possibilities

While the accessibility model seems to provide an adequate measure of testability which can be used at any system level, the problem of developing a measure of testability which could be generally accepted is far from solved. Among further development possibilities are:

(a) Continued Validation

The accessibility model requires a large number of validation experiments for the results to be statistically acceptable. Doing this would be most meaningful if a PCB level testability rating system approach that accommodated VLSI were available (see item (c)).

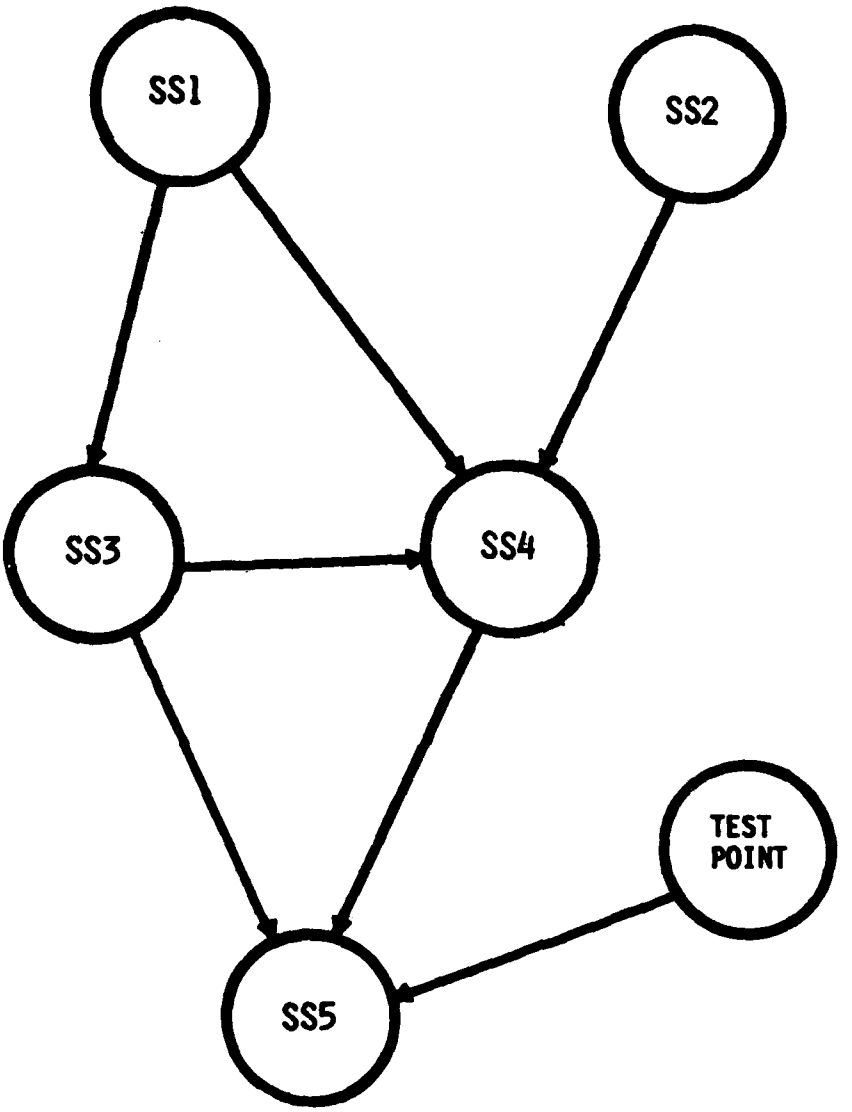


Figure 5.1-6: System With Improved Testability

MODIFIED SYSTEM

NODE	T_{1j}	A_{1j}	NDNT	PATH		T_p
SS1	.82	1.0	.82	1	1 3 4 5	.41
SS2	.93	1.0	.93	2	1 3 5	.52
SS3	.48	.82	.39	3	1 4 5	.42
SS4	.35	.71	.25	4	2 4 5	.43
SS5	.93	.55	.51	5	Tp 5	.68
TEST POINT	1.0	1.0	1.0			

System Testability = .47

Figure 5.1-7: Modified System Testability

(b) A Test Point Location System

One advantage of a testability FOM such as the measure based on the Accessibility Model is its ability to guide the process of test point selection. Research needs to be done on the effects of test points on testability as measured by the FOM. The results of such research would be a set of guidelines for test point location.

(c) Development of an Improved PCB Testability Rating System

The Grumman rating system has many shortcomings when applied to current designs using VLSI components and results in extremely understated testability ratings at times. The Grumman technique does have enough value, however, that it is worth modifying to accommodate VLSI until a rigorous PCB level FOM can be developed. Efforts in both modifying the Grumman technique and research in developing a truly rigorous PCB level FOM should be initiated concurrently.

6.0 SUMMARY

The overall objective of this program was to extend the current PCB testability design and rating system developed by Grumman to the subsystem/system levels and to generate an overall system level testability figure of merit. After an extensive survey of current approaches to testability FOM's it was determined that a graph model would best meet the given objective. Two graph models were constructed based on two different analysis techniques:

- o Pulse Process Analysis
- o Hybrid Systems Analysis

A preliminary analysis of these two techniques demonstrated that a modification of the hybrid systems analysis approach would best satisfy the requirements of an effective testability FOM. Application of that approach to three BAC systems validated applicability of the FOM measure when the baseline Grumman PCB scores were modified to reflect actual PCB test experience for cards containing VLSI memory chips and microprocessors.

Bibliography

1. A Survey of Computer-Aided-Design and Analysis Programs, Electrical and Electronic Systems Department, University of South Florida, Tampa, Florida, April 1976, (AFAPL TR-76-33).

2. Selection Guide for Digital Test Program Generation Sets, Joint DARCOM/NMC/AFLC/AFSC/Commanders. 1981, NAVMATP-9493, DARCOMP-70-9, AFLCP-800-41, AFSCP-800-41, NAVMC-2718.

3. Dejka, W. J., "Measure of Testability in Device & System Design," Proc. 20th Midwest Symposium on Circuits & Systems, 1977, pp. 39-52.

4. Stephenson, J. E. & Grason, J., "A Testability Measure for Register Transfer Level Digital Circuits," 6th FTC Symposium, June 1976, pp. 101-107.

5. Grumman Aerospace, "Design Guidelines & Optimization Procedures for Test Subsystem Design," RADC-TR-80-111, April 1980.

6. Goldstein, L. H., "Controllability/Observability Analysis of Digital Circuits," IEEE Transactions on Circuits & Systems, Vol. 26, Sept. 1979, pp. 685-693.

7. Dussault, J. A., "A Testability Measure," Proc. IEEE Semiconductor Test Conf., 1978, pp. 113-116.

8. Brewer, M. A. & Friedman, A. D., "Test/80-A Proposal for an Advanced Automatic Test Generation System," Proc. IEEE Autotestcon, 1979, pp. 305-312.

9. Bennetts, R. G., et al, "CAMELOT: A Computer-Aided Measure for Logic Testability," IEEE Proc., Vol. 128, Sept. 1981, pp. 177-189.

10. Longendorfer, B., "Computer-Aided Testability Analysis of Analog Circuitry," Autotestcon, 1981, pp. 122-125.

11. Takaski, S., et al, "A Calculus of Testability Measures at the Functional Level," IEEE Test Conf., 1981, pp. 95-101.

12. Kouijanic, P. G., "Testability Analysis," IEEE Test Conf., 1979, pp. 310-316.

APPENDIX I

APPLICATION COMPUTER PROGRAM

```
PROGRAM INGRAPH (INPUT,OUTPUT,OUTFILE);

        CONST MAX=100;
              NEG=-1;

        TYPE  FILETYPE=FILE OF CHAR;

        VAR   OUTFILE: FILETYPE;
              INTEST: ARRAY [1..MAX] OF REAL;
              NODES,LOOPAS,I,J,K,P: INTEGER;
              CIRNAME: PACKED ARRAY[1..40] OF CHAR;

BEGIN

        OPEN( OUTFILE,'DATA',NEW );
        REWRITE(OUTFILE);

        FOR I:=1 TO 25 DO WRITELN;

        WRITELN ('               INPUT INITIAL DATA');
        WRITELN;
        WRITELN;
        WRITELN('ENTER CIRCUIT NAME');
        WRITELN;

        READLN(CIRNAME);
        WRITELN(OUTFILE,CIRNAME);

        WRITELN;
        WRITELN;
        WRITELN('INPUT NUMBER OF FEEDBACK LOOP ITERATIONS');
        READLN(LOOPAS);
        WRITELN(OUTFILE,LOOPAS);
        WRITELN;
        WRITELN ('INPUT NUMBER OF NODES');
        READLN(NODES);
        WRITELN;
        WRITELN(OUTFILE,NODES);

        WRITELN;
        FOR I:=1 TO NODES DO
                BEGIN
                P:=1;
                WRITELN;
                WRITELN('INPUT INITIAL TESTABILITY OF NODE ',I);
                READLN(INTEST[I]);
                WHILE P>0 DO
                  BEGIN
                  WRITELN;
                  WRITELN('NUMBER OF A NODE WITH AN ARC STARTING AT NODE',I);
                  WRITELN('ENTER 0 TO STOP');
                  WRITELN;
                  READLN(P);
                  IF P>0 THEN WRITELN(OUTFILE,I,P);
                END;
        END;
```

```
        WRITELN(OUTFILE,NEG,NEG);
        FOR I:=1 TO NODES DO WRITELN(OUTFILE,INTEST[I]:15:6);
END.
```

```
PROGRAM TESTABILITY(INPUT,OUTPUT,INFILE);

  CONST MAXSIZE=100;

  TYPE FILETYPE=FILE OF CHAR;
       ARRAYTYPE=ARRAY[1..MAXSIZE,1..MAXSIZE] OF REAL;
       ROWTYPE=ARRAY[1..MAXSIZE] OF REAL;
       POINT1=^STACK1;
       STACK1=RECORD
              NODE:INTEGER;
              NEXT:POINT1;
              END;
       POINT2=^STACK2;
       STACK2=RECORD
              NxNODE:INTEGER;
              BP:INTEGER;
              LINK:POINT2;
              END;
       POINT3=^STORExPATH;
       STORExPATH=RECORD
              TESTV:REAL;
              HEADER:POINT1;
              LINKER:POINT3;
              END;

  VAR
    HEADN,TN1,TN2:POINT1;
    HEADP,TP1,TP2:POINT2;
    HEADF,TF1,TF2:POINT3;
    CYCLE:ARRAY [1..MAXSIZE] OF INTEGER;
    NV,CURxNODE:INTEGER;

    TEST,I,J,K,V,W,P,NODES,TOP,CTOP,NPATH:INTEGER;
    INFILE:FILETYPE;
    SUM,PATHxLENGTH:REAL;
    TRANS,CON,TEMPCON:ARRAYTYPE;
    ACCESS,INTxTEST,Q,NDNT,PATHxTEST:ROWTYPE;
    PATHxCOST:ARRAY[0..MAXSIZE,0..1] OF REAL;
    STACK:ARRAY[1..MAXSIZE,0..1] OF INTEGER;
    PATH:ARRAY[1..MAXSIZE,0..1] OF REAL;
    SUM2,TESTABILITY:REAL;
    LOOP:BOOLEAN;
    IPVT:ROWTYPE;
    RESULT:ROWTYPE;
    SUMROW,SUMCOL:ARRAY [1..MAXSIZE] OF REAL;
    CIRNAME: PACKED ARRAY[1..40] OF CHAR;

  PROCEDURE PUSHN(DATA:INTEGER;VAR HEAD:POINT1);
    BEGIN
      NEW(TN1);
      TN1^.NODE:=DATA;
      TN1^.NEXT:=HEAD;
      HEAD:=TN1;
    END;

  PROCEDURE PUSHP(DATA,BRANCH:INTEGER);
    BEGIN
      NEW(TP1);
      TP1^.NxNODE:=DATA;
      TP1^.BP:=BRANCH;
      TP1^.LINK:=HEADP;
      HEADP:=TP1;
    END;
```

```
PROCEDURE POPN(VAR DATA:INTEGER;VAR HEAD:POINT1);
  BEGIN
    DATA:=HEAD^.NODE;
    TN1:=HEAD;
    HEAD:=HEAD^.NEXT;
    DISPOSE(TN1);
  END;

PROCEDURE POPP;
  BEGIN
    TP2:=HEADP;
    HEADP:=HEADP^.LINK;
    DISPOSE(TP1);
  END;

PROCEDURE SETUP;

  BEGIN
    FOR I:=1 TO NODES DO
      BEGIN
        SUM:=0;
        FOR J:=1 TO NODES DO
          BEGIN
            SUMCOL[I]:=SUMCOL[I]+CON[J,I];
            SUMROW[I]:=SUMROW[I]+CON[I,J];
          END;
        IF SUMCOL[I] <> 0 THEN
          FOR K:=1 TO NODES DO
            TEMPCON[K,I]:=CON[K,I]/SUMCOL[I]
         ELSE
           BEGIN
             ACCESS[I]:=-1;
             PUSHP(I,ROUND(SUMROW[I]));
             FOR K:=1 TO NODES DO
                 IF CON[I,K]=1 THEN
                     PUSHN(K,HEADN);
           END;
      END;
    FOR I:=1 TO NODES DO
      BEGIN
        FOR J:=1 TO NODES DO
          TEMPCON[I,J]:=TEMPCON[I,J]*INTXTEST[I];
        TEMPCON[I,I]:=-1;
      END;
    FOR I:=1 TO NODES DO
      FOR J:=1 TO NODES DO
        TRANS[I,J]:=TEMPCON[J,I];

        WRITELN;
        WRITELN('PRIMARY NODES VECTOR:');
        WRITELN;
        WRITELN;
        FOR I:=1 TO NODES DO
            WRITE(ACCESS[I]:5:0);
        WRITELN;


        WRITELN;
        WRITELN;
        WRITELN('CONNECTIVITY MATRIX:');
        WRITELN;
        WRITELN;
        WRITELN('FROM NODE \ TO NODE ');
        WRITELN;
```

```
        WRITE('    ');

        FOR I:=1 TO NODES DO WRITE(I:6);
        WRITELN;
        WRITELN;

        for i:=1 to nodes do
          begin
            WRITE(I:4);
            for j:=1 to nodes do
              write(con[i,j]:6:0);
            writeln;
            WRITELN;
          end;
END;

    PROCEDURE SOLVACC(VAR A:ARRAYTYPE;N:INTEGER;VAR B:ROWTYPE);
         VAR I,J,K:INTEGER;
             SUM,MULTFAC:REAL;
         BEGIN
         FOR J:=1 TO N DO
            BEGIN
            IF A[J,J]<>0 THEN
              BEGIN
              FOR I:=(J+1) TO N DO
                 BEGIN
                 IF A[I,J]<>0 THEN
                    BEGIN
                    MULTFAC:=A[I,J]/A[J,J];
                    FOR K:=1 TO N DO
                       A[I,K]:=A[I,K]-MULTFAC*A[J,K];
                    B[I]:=B[I]-MULTFAC*B[J];
                    END;
                 END;
              END
            ELSE
              WRITELN('ERROR -- ACCESSABILITIES NOT ATTAINED');
            END;

         WRITELN;
         WRITELN;
         WRITELN('FORWARD ELIMINATION OF ACCESSABILITIES COEFFICIENTS');
         WRITELN;
         WRITELN;
         FOR I:=1 TO N DO
            BEGIN
            FOR J:=1 TO N DO
               WRITE(A[I,J]:6:2);
            WRITELN;
            WRITELN;
            END;
         WRITELN;
         WRITELN;

         FOR I:=N DOWNTO 1 DO
           BEGIN
           IF A[I,I]<>0 THEN
             BEGIN
             SUM:=0.0;
             FOR J:=1 TO N DO
               IF I<>J THEN  SUM:=SUM+A[I,J]*B[J];
             B[I]:=(B[I]-SUM)/A[I,I];
             END;
           END;
```

```
            END;

PROCEDURE PATHFINDER;

BEGIN
  WHILE (HEADP<>NIL) AND (HEADN<>NIL) DO
   BEGIN
    POPN(CURxNODE,HEADN);
    HEADP^.BP:=HEADP^.BP-1;
    WHILE (CYCLE[CURxNODE] > NV) AND (HEADP <> NIL) DO
      BEGIN
        IF HEADP^.BP<=0 THEN
          BEGIN
            LOOP:=FALSE;
            WHILE (HEADP<>NIL) AND (NOT LOOP) DO
              BEGIN
                IF HEADP^.BP<=0 THEN
                  BEGIN
                    CYCLE[HEADP^.NxNODE]:=CYCLE[HEADP^.NxNODE]-1;
                    POPP;
                  END
                 ELSE
                    LOOP:=TRUE;
              END;
          END;

            IF HEADN<>NIL THEN
              BEGIN
                POPN(CURxNODE,HEADN);
                HEADP^.BP:=HEADP^.BP-1;
              END;
          END;

            IF HEADP<>NIL THEN
              BEGIN
                CYCLE[CURxNODE]:=CYCLE[CURxNODE]+1;
                PUSHP(CURxNODE,ROUND(SUMROW[CURxNODE]));
                IF SUMROW[CURxNODE]>0 THEN
                  BEGIN
                    FOR I:=1 TO NODES DO
                        IF CON[CURxNODE,I]=1 THEN
                            PUSHN(I,HEADN);
                  END
                 ELSE
                  BEGIN
                    K:=0;
                    NEW(TF1);
                    TN2:=NIL;
                    TF1^.LINKER:=HEADF;
                    TF1^.TESTV:=0;
                    HEADF:=TF1;
                    LOOP:=FALSE;
                    WHILE (HEADP<>NIL) AND NOT(LOOP) DO
                      BEGIN
                        IF HEADP^.BP<=0 THEN
                          BEGIN
                            PUSHN(HEADP^.NxNODE,TN2);
                            TF1^.TESTV:=TF1^.TESTV+(1/NONI[HEADP^.NxNODE]);
                            CYCLE[HEADP^.NxNODE]:=CYCLE[HEADP^.NxNODE]-1;
                            K:=K+1;
                            POPP;
                          END
                         ELSE LOOP:=TRUE;
                      END;
```

```
                        LOOP:=FALSE;
                        TP2:=HEADP;
                        WHILE (TP2 <> NIL) AND NOT(LOOP) DO
                          BEGIN
                            PUSHN(TP2^.NxNODE,TN2);
                            TF1^.TESTV:=TF1^.TESTV+(1/NDNT[TP2^.NxNODE]);
                            K:=K+1;
                            IF SUMCOL[TP2^.NxNODE]>0 THEN
                                TP2:=TP2^.LINK
                              ELSE
                                LOOP:=TRUE;
                          END;
                        TF1^.TESTV:=K/TF1^.TESTV;
                        TF1^.HEADER:=TN2;
                      END;
                  END;
    END;
END;

(*******************************************************)
(*                                                     *)
(*         START OF MAIN PROGRAM                       *)
(*                                                     *)
(*******************************************************)
  BEGIN

    OPEN (INFILE,'DATA',OLD);
    RESET (INFILE);

    READLN(INFILE,CIRNAME);
    WRITELN;
    WRITELN('*************************************************************');
    WRITELN;
    WRITELN;
    WRITE('TESTABILITY ANALYSIS OF ');
    FOR I:= 1 TO 40 DO
        WRITE(CIRNAME[I]);
    WRITELN;
    WRITELN;
    WRITELN('*************************************************************');
    WRITELN;
    WRITELN;

    READLN(INFILE,NV);

    READ(INFILE,NODES);

    READLN(INFILE,I,J);
    WHILE (I>=0) AND (J>=0) DO
      BEGIN
        CON[I,J]:=1;
        TEMPCON[I,J]:=1;
        READLN(INFILE,I,J);
      END;

    FOR I:=1 TO NODES DO
        READLN(INFILE,INTxTEST[I]);

(*INITIALIZE ACCESSABILITY VECTOR*)

    SETUP;

(* NICKI'S ROUTINE *)
```

```
    WRITELN;
    WRITELN('ACCESSABILITY COEFFICIENTS:');
    WRITELN;
    FOR I:=1 TO NODES DO
      BEGIN
       FOR J:=1 TO NODES DO
          WRITE(TRANS[I,J]:6:2);
       WRITELN;
       WRITELN;
      END;

    SOLVACC(TRANS,NODES,ACCESS);

    WRITELN;
    WRITELN;
    WRITELN;
    WRITELN;

    FOR I:=1 TO NODES DO
        NDNT[I]:=ACCESS[I]*INITXTEST[I];



(*OUTPUT NETWORK VALUES*)
    WRITELN;
    WRITELN;
    WRITELN;
    WRITELN('NETWORK VALUES');
    WRITELN;
    WRITELN;
    WRITELN('NODE':5,'INITIAL TESTABILITY':22,'ACCESSABILITY':17,'NDNT':15);
    WRITELN;
    FOR I:=1 TO NODES DO
      WRITELN(I:4,INITXTEST[I]:22:2,ACCESS[I]:17:2,NDNT[I]:15:2);
    WRITELN;
    WRITELN;

 (*FIND THE PATH TESTABILITIES AND THE INPUT NODES*)

    PATHFINDER;

 (*OUTPUT PATH TESTABILITIES*)

    WRITELN('THE NUMBER OF PATH ITERATIONS WAS ',NV:2);
    WRITELN;
    WRITELN('EACH PATH TESTED FOLLOWS:');
    WRITELN;

    TF1:=HEADF;
    WHILE TF1<>NIL DO
      BEGIN
        WRITELN('PATH:');
        TN1:=TF1^.HEADER;
        WHILE TN1<>NIL DO
          BEGIN
            WRITE(TN1^.NODE:3,' -->  : );
            TN1:=TN1^.NEXT;
          END;
        WRITELN;
        WRITELN('PATH TESTABILITY = ',TF1^.TESTV:4:2);
        WRITELN;
        WRITELN;
        TF1:=TF1^.LINKER;
      END;
```

```
    WRITELN;
    WRITELN;
    WRITELN;
    WRITELN;
    WRITELN;


(*CALCULATE SYSTEM TESTABILITY*)
    SUM:=0;
    TF1:=HEADF;
    WHILE TF1<>NIL DO
      BEGIN
        SUM:=SUM+1/TF1^.TESTV;
        NPATH:=NPATH+1;
        TF1:=TF1^.LINKER;
      END;
    TESTABILITY:=NPATH/SUM;
    WRITELN;
    WRITELN;
    WRITELN('      SYSTEM TESTABILITY = ',TESTABILITY:4:2);
  END.
```

Appendix II

List of Acronyms and Abbreviations

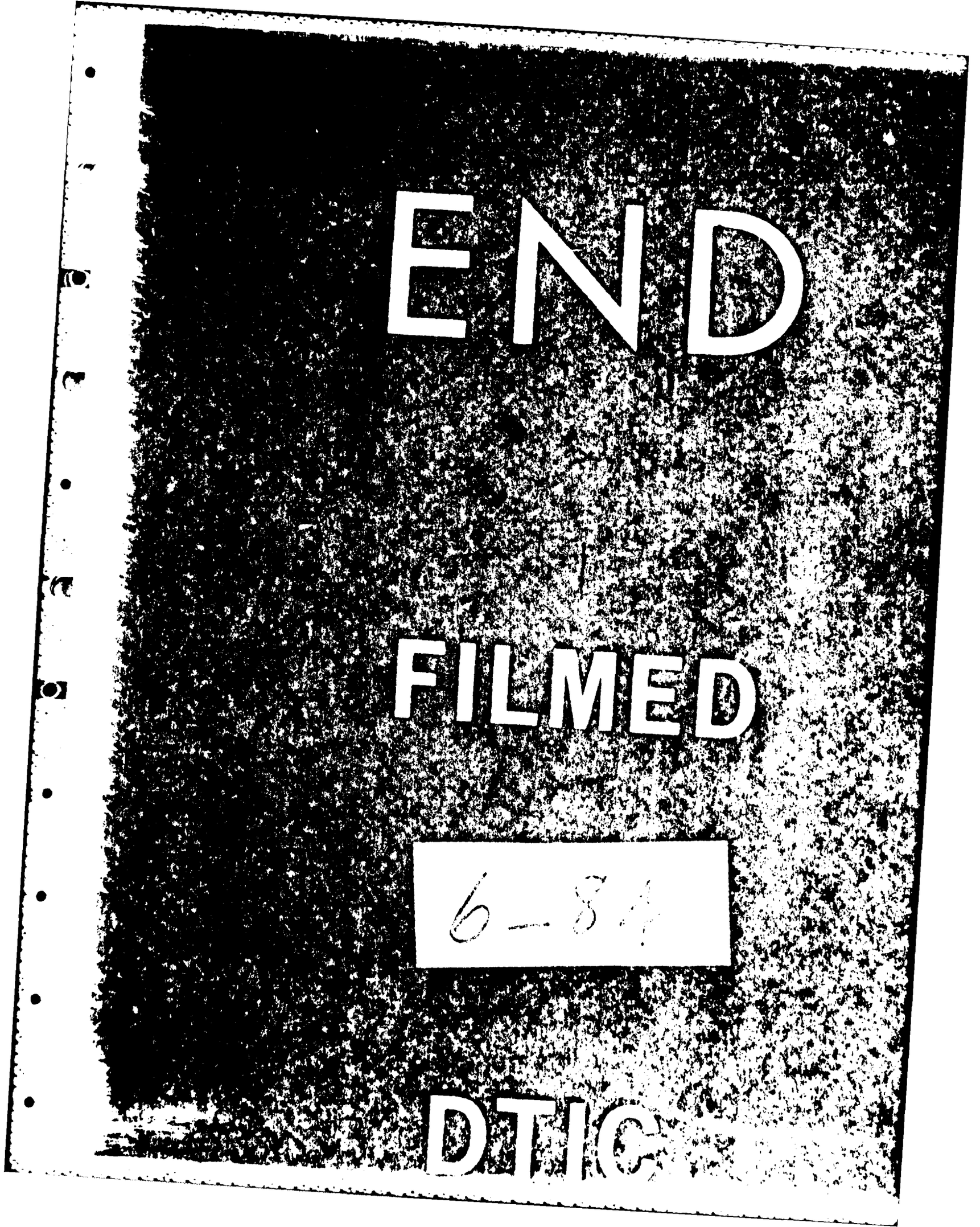
ATE	Automatic Test Equipment
CND	Cannot Duplicate
DFT	Design for Testability
FCE	Flight Control Electronics
FOM	Figure of Merit, or Testability Figure of Merit
FON	Fault Occurrence Networks
FSEU	Flap/Slat Electronics Unit
INT	Isolated Node Testability
ISR	Inverse Signal Reliability
LRU	Line Replaceable Unit
LSI	Large Scale Integration
LSSD	Level Sensitive Scan Design
MSI	Medium Scale Integration
MTBF	Mean Time Between Failures
NDNT	Network Dependent Node Testability
PCB	Printed Circuit Board
RADC	Rome Air Development Center
SSI	Small Scale Integration
TTM	Testability Transfer Matrix
VLSI	Very Large Scale Integration

Explanation of Use of Harmonic Mean Averaging

Since the testability contributions of many nodes are to be considered in obtaining a single subsystem or system figure-of-merit, some sort of averaging means must be used to obtain that value. The harmonic mean (the reciprocal of the sum of the reciprocals of the values) was chosen as the averaging technique. This form of average is relatively insensitive to extreme values, and weights the average value towards the less testable components or nodes. This result intuitively seems to be most logical and consistent with engineering experience.

MISSION
of
Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C^3I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.


END
FILMED
6-84
DTIC